Fakulta elektrotechnická
Katedra elektrotechniky a informačních technologií

Diplomová práce

Řízení linky E-robot PLC automatem

Autor práce: Bc. David Forejtek
Vedoucí práce: Ing. Jiří Basl, Ph.D

Plzeň 2021

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta elektrotechnická
Akademický rok: 2020/2021

ZADÁNÍ DIPLOMOVÉ PRÁCE
(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Bc. David FOREJTEK**
Osobní číslo:	**E18N0003K**
Studijní program:	**N2644 Aplikovaná elektrotechnika**
Studijní obor:	**Aplikovaná elektrotechnika**
Téma práce:	**Řízení linky E-robot PLC automatem**
Zadávající katedra:	**Katedra elektroenergetiky**

Zásady pro vypracování

1. Rozdělení řídících systémů Siemens Simatic, možnosti jejich uplatnění, technické parametry řídícího systému S7-1515 pro robotické pracoviště E-robot.
2. Naprogramování softwaru PLC automatu Siemens CPU1515-PN robotického pracoviště E-robot.
3. Vytvoření vizualizace robotického pracoviště E-robot pro operátorský panel Siemens TP1900.
4. Rešerše alternativ virtuálního zprovoznění robotických pracovišť a vygenerování podkladů pro simulaci pracoviště.

Rozsah diplomové práce: **40 – 60 stran**
Rozsah grafických prací: **podle doporučení vedoucího**
Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. Technické parametry řídících systémů S7, firemní dokumentace

Vedoucí diplomové práce: **Ing. Jiří Basl, Ph.D.**
Katedra elektroniky a informačních technologií

Datum zadání diplomové práce: **9. října 2020**
Termín odevzdání diplomové práce: **27. května 2021**

L.S.

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

Doc. Ing. Karel Noháč, Ph.D.
vedoucí katedry

V Plzni dne 9. října 2020

Abstrakt

Práce se věnuje návrhu a realizaci software pro PLC a HMI experimentálního robotického pracoviště pro výrobu ptačích budek. Architektura software je založena zejména na objektově orientovaném přístupu a modularitě. Pro popis jednotlivých programových modulů jsou ve velké míře využívány metody abstrakce a generalizace. Výsledkem práce je software otestovaný ve virtuálním prostředí Process Simulate. Potvrzení funkčnosti proběhlo na reálném pracovišti.

Klíčová slova

PLC, HMI, modularita, generalizace, abstrakce

Abstract

The thesis deals with the design and implementation of software for PLC and HMI of the experimental robotic workplace for the production of birdhouse. The software architecture is mainly based on an object-oriented approach and modularity. Abstraction and generalization principles are widely used to describe individual program modules. The implemented software was tested in the virtual environment of the Process Simulate software. The verification of functionality took place at a real workplace.

Keywords

PLC, HMI, Modularity, Generalization, Abstraction

Prohlášení

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

Prohlašuji, že jsem svou závěrečnou práci vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 270 tresního zákona č. 40/2009 Sb.

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

V Plzni dne 27. května 2021

Bc. David Forejtek

. . . . . .

Podpis

Obsah

Seznam obrázků **vii**

Seznam tabulek **viii**

Seznam symbolů a zkratek **ix**

1 Úvod **1**

2 Řídící systém SIMATIC **2**
2.1 Modularita . . . 2
2.2 Dělení série S7 . . . 2
2.3 Porovnání parametrů . . . 3
2.4 Technické parametry CPU1515F-2 PN . . . 5

3 Pracoviště E-Robot **6**
3.1 Výrobek . . . 6
3.2 Aktuátory . . . 7
3.3 Sekvence pracoviště . . . 8
3.3.1 Slovní popis sekvence . . . 10
3.4 Roboti . . . 11
3.4.1 Lemovací robot . . . 12
3.4.2 Manipulační robot . . . 12
3.4.3 Svařovací robot . . . 13
3.5 110010 - lemování podlahy . . . 14
3.6 110020 - lemování střechy . . . 16
3.7 110050 - svařovací stanice . . . 19
3.7.1 110051 - svařovací přípravek podlahy . . . 20
3.7.2 110052 - svařovací přípravek střechy . . . 23

4 Softwarová architektura **26**
4.1 Obecné požadavky: . . . 26
4.2 Komponenty . . . 27
4.2.1 Architektura komponent . . . 27

4.2.2 Synchronizace mezi komponentami: . . . 28
4.2.3 Komponenta PLC programu . . . 28
4.2.4 Komponenta PLC safety programu . . . 30

5 Design komponenty PLC 32
5.1 Actuator . . . 32
5.2 Station . . . 34
5.3 Station Mode . . . 35
5.4 FSM . . . 36
5.5 Sequencer . . . 42
5.6 Robot . . . 43

6 Design komponenty HMI 45
6.1 Obrazovky HMI . . . 45
6.1.1 Hlavní přehled . . . 45
6.1.2 Obrazovka stanice . . . 46
6.2 Ikony . . . 48
6.2.1 Válec . . . 48
6.2.2 Motor . . . 48
6.3 Vyskakovací okna . . . 48
6.3.1 Aktuátor . . . 49
6.3.2 Stanice . . . 49
6.3.3 FSM . . . 50

7 Virtuální zprovoznění 52
7.1 Softwarové alternativy . . . 52
7.2 Process Simulate . . . 53
7.2.1 Simulační sestava . . . 53
7.2.2 Roboti . . . 54

8 Závěr 55

Seznam obrázků

3.1 Model hotového výrobku 7
3.3 Celková výrobní sekvence pracoviště 8
3.2 Layout pracoviště 9
3.4 Jednotlivé kroky procesu lemování podlahy 14
3.5 Stavový automat lemovacího přípravku 15
3.6 Sekvence stanice 110010 15
3.7 Pracovní pozice lemovacího přípravku 110010 16
3.8 Jednotlivé kroky procesu lemování střechy 17
3.9 Stavový automat stanice 110020 17
3.10 Sekvence pracoviště 110020 18
3.11 Pracovní pozice stanice 110020 19
3.12 Automatická sekvence svařovací stanice 20
3.13 Jednotlivé kroky procesu sváření v přípravku 110051 20
3.14 Stavový automat stanice AFO51 21
3.15 Automatická sekvence svařovacího přípravku 110051 21
3.16 Pracovní pozice svařovacího přípravku 110051 22
3.17 Hotová podsestava střechy po zavaření kroužků 23
3.18 Stavový automat stanice AFO52 23
3.19 Pracovní pozice svařovacího přípravku 110052 24
3.20 Automatická sekvence přípravku 110052 24

4.1 Rozdělení na komponenty 27
4.2 Synchronizace mezi komponentami 28
4.3 Komponenta PLC programu 29
4.4 Interface mezi PLC a HMI 29
4.5 Komponenta safety PLC programu 30
4.6 Komponenta HMI programu 31

5.1 Stavový automat aktuátoru 32
5.2 Design modulu stanice 34
5.3 Ukázka stavového automatu lemovacího přípravku 36
5.4 Implementace stavového automatu 37

5.5 Modelový stavový automat automatické sekvence 42
5.6 Komunikace mezi PLC a robotem 43
5.7 Blokování zóny robotem po dokončení činnosti 44
5.8 Ztráta povolení 44

6.1 Hlavní obrazovka 46
6.2 Obrazovka stanice 110010 47
6.3 Obrazovka robota KUKA 47
6.4 Způsob zobrazení válce na hlavní obrazovce 48
6.5 Způsob zobrazení motoru na hlavní obrazovce 48
6.6 Vyskakovací obrazovka pro ovládání aktuátoru 49
6.7 Vyskakovací obrazovka stanice 50
6.8 Vyskakovací obrazovka FSM 51

Seznam tabulek

2.1 Porovnání parametrů PLC různých řad 3
2.2 Podporované programovací prostředí 4
2.3 Hardwarové požadavky na programovací prostředí 4
2.4 Podporované programovací jazyky pro jednotlivé PLC 4
2.5 Technické parametry CPU1515-PN 5

3.1 Typy robotických operací 11
3.2 Operace prováděné robotem KUKA 12
3.3 Operace prováděné robotem FANUC 13
3.4 Operace prováděné robotem FANUC 14
3.5 Stavy zakládacího přípravku 14
3.6 Poloha naklápěcího mechanismu 23

5.1 Provozní módy stanice 35
5.2 Interface FSM 35
5.3 Interface FSM 37
5.4 Řídící signály komunikace PLC robot 43

Seznam symbolů a zkratek

DB	Data Block. Datový blok.
ES	Emergency Stop. Nouzový stop.
FB	Function Block. Funkční blok.
FC	Function. Funkce.
FSM	Finite State Machine. Konečný stavový automat.
gDB	Global Data Block. Globální datový blok.
HMI	Human Machine Interface. Rozhraní mezi člověkem a strojem.
iDB	Instance Data Block. Instanční datový blok.
OB	Organization Block. Organizační blok.
PLC	Programmable Logic Controller. Programovatelný logický automat.

1

Úvod

Závěrečná práce se zabývá popisem softwarové architektury a designu PLC a HMI programu, užitého pro řízení experimentálního robotického pracoviště E-Robot.

Pracoviště je zkonstruované k automatizované výrobě ptačích budek a jeho účelem je praktické ověření trendů v oblasti průmyslu 4.0, zejména virtuálního zprovoznění. Praktickým cílem práce bylo odladění zmíněného software v rámci simulačního programu Process Simulate a následná verifikace správnosti návrhu na reálném pracovišti.

První část práce se věnuje obecně řídícím systémům Siemens Simatic a podrobněji popisuje použitý řídící systém z hlediska technických parametrů.

Konstrukce robotického pracoviště společně s celkovým výrobním cyklem je popsána v kapitole 3. Součástí je i popis všech pracovních stanic pracoviště spolu s jejich pracovním cyklem. Závěry jednotlivých částí kapitoly slouží jako základ pro implementaci řídícího software.

Těžiště práce tvoří kapitoly 4, 5 a 6. Ty detailně popisují způsob jakým byl řídící software navržen a následně implementován. Při návrhu software byla snaha v maximálně možné míře využít principů objektově orientovaného návrhu.

V kapitole je vysvětleno dělení software na komponenty a moduly a je představen způsob výměny informací mezi těmito celky. Každý modul je potom podrobně představen v konkrétní podkapitole.

Poslední kapitola se stručně věnuje virtuálnímu zprovoznění.

2

Řídící systém SIMATIC

Kapitola představí základní vlastnosti a vývojové řady PLC značky Siemens a uvede jejich rozdělení. Následně se zaměří na popis základních technických parametrů v současnosti používaných řad S7-300, S7-1200 a S7-1500. Závěr kapitoly se detailněji věnuje popisu použitého PLC pro řízení pracoviště E-Robot.

2.1 Modularita

Siemens PLC mohou tvořit modulární systém složený z CPU jednotky a dodatečných karet. Karty nejčastěji rozšiřují počet vstupů a výstupů systému nebo zajišťují specializované funkce (komunikace po sériové lince, zpracování hodnot z termočlánku apod.). Maximální možný počet připojených karet je pak různý v závislosti na CPU.

Podpora rozšiřujících karet pro konkrétní PLC se liší podle řady PLC a je nutné ji při návrhu zohlednit. Obecně ale platí, že systém je dostatečně konfigurovatelný a spektrum vyráběných produktů natolik široké, že splnit běžné požadavky pro řízení není problém.

Dalším způsobem rožšíření systému jsou distribuované IO kontroléry. Ty tvoří separátní ostrov rozšiřujících karet připojených k oddělenému specializovanému IO kontroléru. Ten pak s PLC komunikuje po nějakém typu sběrnice (pro Siemens je to typicky ProfiNET).

2.2 Dělení série S7

Na trhu jsou v dnešní době dostupná PLC série S7. Starší série C3, S3 a S5 jsou dnes již zastaralé a v nových aplikacích se nenasazují, proto se jimi tato kapitola nebude zabývat.

Série S7 se dělí do následujících tříd [1]:

- S7-200
  V dnešní době nepoužívaná třída, jejíž výroba byla v roce 2017 ukončena. Je vhodná pro malé aplikace do několika desítek I/O.

- S7-300
  Jedná se o pokročilejší, do dnes používanou třídu PLC, která je vhodná pro řízení náročnějších aplikací. Třída je zde uvedena pro porovnání z hlediska výkonu a nabízených možností oproti moderním typům PLC.

- S7-400
  Oproti S7-300 se jedná o výkonější řadu vhodnou pro datově náročnější aplikace [2]. Uplatnění nalezne například v těžkém průmyslu a pohonářských aplikacích. Protože se tyto aplikace vymykají zaměření práce, nebude řada S7-400 detailněji popisována.

- S7-1200
  Nová třída PLC představená v roce 2012 spolu s S7-1500. Oproti S7-1500 je S7-1200 méně výkoná a nabízí omezenější možnosti programování. S nástupem třídy S7-1200 a S7-1500 se standardní komunikační sběrnicí stává ProfiNET

- S7-1500
  Zatím poslední třída vyráběných PLC. Do jisté míry nabízí zpětnou kompatibilitu s řadou S7-300 a v budoucnu ji má nahradit.

2.3 Porovnání parametrů

Text se zaměřuje na porovnání technických parametrů a programových možností dnes používaných tříd PLC.

Protože množství technických parametrů PLC je široké a porovnání všech mezi sebou by nebylo účelné, byly zvoleny následující reprezentativní parametry pro porovnání jednotlivých tříd z hlediska výpočetního výkonu. Každá třída je v tabulce zastoupena nejvýkonějším CPU.

PLC	Doba zpracování instrukce [ns]			Paměť programu	Paměť dat
	BOOL	WORD	FLOAT		
1217-C [3]	80	1700	2300	150KB	150KB
319-3 [4]	4	10	40	2048KB	2048KB
1518-4 [5]	1	2	6	4MB	20MB

Tab. 2.1: Porovnání parametrů PLC různých řad

Z porovnání je vidět dramatický rozdíl ve výpočetním výkonu třídy S7-1200 a 1500. Třídy S7-300 a 1500 jsou v tomto ohledu srovnatelné.

Podporované programovací prostředí zobrazuje 2.2. Dříve používané prostředí Simatic Manager Step 7 bylo postupně vytlačeno prostředím TIA portal.

TIA portal kombinuje programovací prostředí pro PLC i HMI značky Siemens a je uživatelsky přívětivější. Na druhou stranu je náročnější na hardwarové požadavky a ne-

umožňuje některé funkce, které Simatic Manager Step 7 podporoval (například vyhledávání proměnné jen na základě jména).

PLC	Prostředí
S7-300	Simatic Manager Step 7 TIA Portal
S7-1200 S7-1500	TIA Portal

Tab. 2.2: Podporované programovací prostředí

Srovnání z hlediska hardwarových nároků je znázorněno tabulkou 2.3

Prostředí	RAM	CPU	OS	HDD
Step 7 [7]	512MB	600MHz	Win XP 32 bit	1GB
TIA [6]	8GB	3.4GHz	Win 7 64 bit	50GB

Tab. 2.3: Hardwarové požadavky na programovací prostředí

Dalším zásadním rozdílem mezi ustupující třídou S7-300 a nově nastupivšími třídami S7-1200 a S7-1500 je výrazné potlačení možnosti použití nepřímé adresace a absolutní adresace. V PLC třídy S7-300 měl programátor možnost operovat s adresními a akumulátorovými registry, nad nimiž mohl provádět různé operace v jazyce STL. Bylo tedy možné vypočítat adresu proměnné a potom k ní nepřímo přistupovat. U PLC S7-1200 je tato možnost vyloučena úplně a u S7-1500 je výrazně omezena.

Srovnání podle podporovaných programovacích jazyků zobrazuje tab 2.4. Z uživatelského hlediska se jedná se o klíčovou vlastnost PLC, protože některé operace buď nelze v některých jazycích provést vůbec a nebo jen velmi obtížně (například operace s registry jsou možné pouze v STL).

PLC	LAD	FBD	GRAPH	SCL	STL	HiGraph	CFC
S7-1200 [3]	✓	✓	x	✓	x	x	x
S7-300 [4]	✓	✓	✓	✓	✓	x	x
S7-1500 [5]	✓	✓	✓	✓	✓	✓	✓

Tab. 2.4: Podporované programovací jazyky pro jednotlivé PLC

Pro úplnost je ovšem třeba dodat, že ačkoliv jazyky SCL, LAD a FBD jsou podporovány všemi vyjmenovanými PLC, instrukční sady se pro jednotlivá CPU mohou lišit. To je významný nedostatek z hlediska kompatibility zejména mezi S7-1200 a S7-1500.

Pokud chceme těmto problémům zabránit, je nutné nepoužívat specializované instrukce/funkce/funkční bloky dostupné jen pro určitou řadu PLC.

2.4 Technické parametry CPU1515F-2 PN

Text představuje technické parametry řídícího systému CPU1515-PN [8] užitého pro řízení pracoviště E-Robot. V úvodu jsou pro porovnání uvedeny stejné parametry výpočetního výkonu jako v předchozím textu, v další části jsou pak uvedeny programové možnosti PLC.

PLC	Doba zpracování instrukce [ns]			Paměť programu	Paměť dat
	BOOL	WORD	FLOAT		
1515-PN	30	36	192	750KB	3MB

Tab. 2.5: Technické parametry CPU1515-PN

PLC obsahuje slot na paměťovou kartu. Její maximální velikost může být až 32GB. Programové možnosti jsou následující:

- DB
  Programátor může využít číselný rozsah pro datové bloky od 1 do 59 999. Rozsah od 60 000 do 60 999 je rezervovaný pro safety datové bloky.
  Maximální možná velikost paměti pro přístup absolutní adresací je 64KB, jinak lze využít celý rozsah paměti 3MB.

- FB, FC, OB
  Maximální paměť použitelná pro vyjmenované programové bloky je 500KB.
  V PLC lze použít standardní OB (StartUp, Cyclic interrupt atp.) jako u ostatních PLC. Nejrychlejší perioda cyklického interuptu je 0,5ms.

Maximální počet všech bloků (DB, FB, FC, OB a UDT) je omezen na 6 000.

Z hlediska komunikace PLC s ostatními prvky na pracovišti je důležitým parametrem možnost rozšíření připojení o distribuované IO systémy. PLC jich umožňuje připojit až 64. Navíc k samotnému PLC lze připojit až 31 modulů a rozšířit tak možnosti CPU.

Jak bude zřejmé z dalších kapitol, tyto možnosti PLC jsou zcela dostatené pro potřeby řízení pracoviště E-Robot.

3

Pracoviště E-Robot

Kapitola představuje pracoviště po konstrukční a technologické stránce. V úvodní části je obecně popsána konstrukce výrobku a je naznačeno, jakými výrobními procesy procházejí jednotlivé díly. Následuje popis dělení pracoviště na stanice, způsob propojení jednotlivých celků a popis použitých aktuátorů. Další sekce kapitoly popisuje celý výrobní postup. Následují detailní funkční popisy jednotlivých pracovních stanic; technologie výroby je zde ilustrována na modelech výrobku.

Layout pracoviště zobrazuje obrázek 3.2. Je funkčně rozdělen do několika stanic, z nichž každá disponuje nějakou formou IO slučovače a průmyslovým switchem. Propojení stanic v rámci Ethernetové sítě je realizované hvězdicovou topologií, po níž prvky komunikují po sběrnici ProfiNET.

Podružné IO slučovače stanic v podobě IO-linkových modulů nebo ventilových terminálů sdružují vstupy a výstupy konkrétní stanice. Součástí jsou i Fail-Safe IO. Každá stanice integruje Fail-Safe výstupy pro spínaní napájení pohonů (je-li na stanici přítomen) a přívodních vzduchových ventilů.

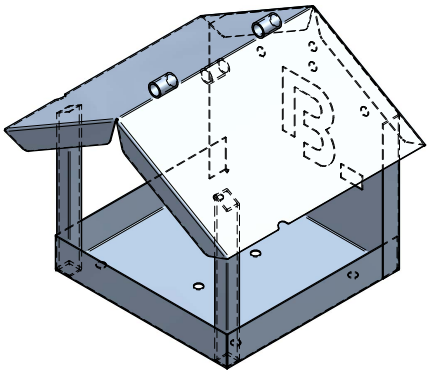
Stanice, s nimiž přijde do kontaktu operátor, jsou navíc vybaveny tlačítkem nouzového vypnutí a odbavovacím tlačítkem. Dalšími bezpečnostními prvky zakládacích stanic jsou pak světelné závory a scannery přítomnosti osob.

3.1 Výrobek

Pracoviště je konstruované k výrobě plechových ptačích krmítek. Motivace pro výběr krmítka jako vyrýběného předmětu je dvojí:

1. Proces výroby je netriviální
   lze na něm tedy dobře ilustrovat výhody virtuálního zprovoznění.

2. Marketingové důvody
   krmítko může sloužit jako reklamní předmět.



Obr. 3.1: Model hotového výrobku

Krmítko se skládá ze dvou dílů: střechy a podlahy. Tyto podsestavy jsou v procesu výroby zvlášť připravovány a jejich svařením vznikne finální výrobek.

Pro samotnou výrobu jsou užité následující technologické operace:

- Robotické lemování (robot KUKA 110030R01)
  Plech střechy i podlahy prochází v první fázi výroby lemovacím procesem. Lemování je u každého dílu dvojí.
- Robotické vybírání a zakládání dílů (robot FANUC 110070R01)
  Kroužky (součást podsestavy střechy) i sloupky (podsestava podlahy) je nutné před samotným svářením vybrat ze zakládacího šuplíku a následně založit do svařovacího přípravku.
- Robotický HandShake podsestav a dílů
  V určitých momentech procesu výroby je nutné, aby si roboti FANUC a KUKA mezi sebou vyměnili díly. Z pohledu průmyslové praxe se nejedná o běžnou situaci.
- Robotické obloukové svařování (robot ABB 110040R01)
  Veškeré svařování na pracovišti (jak podsestav, tak celkové sestavy) je prováděno obloukově svařovacím robotem ABB.

3.2 Aktuátory

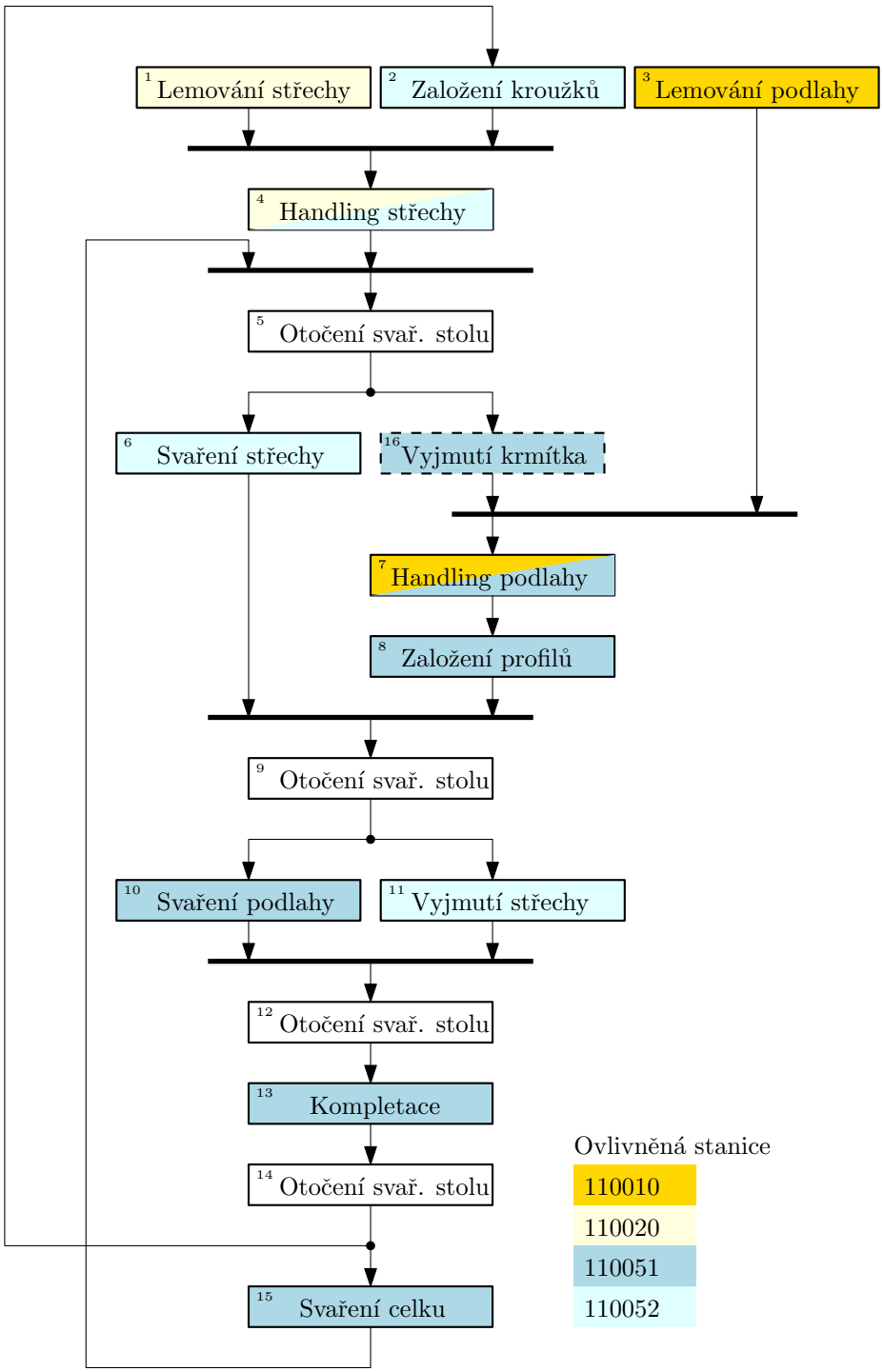
Na pracovišti jsou dva typy aktuátorů. Pneumatické válce ovládané ventily a stykačově řízené asynchronní motory. Z pohledu řízení se tedy jedná výhradně o binární řízení. Stejnou povahu mají i všechny vstupní signály.

U válců rozlišujeme výchozí (VP) a pracovní (PP) polohu na základě zpětné vazby z koncových čidel. U motorů je ZV realizována kontrolními kontakty stykačů. Ty informují o správném sepnutí a tedy směru točení motoru.

Jako zpětná vazba polohy otočných stolů (110020DT1 a 110050DT1) fungují vačkové snímače umístěné na prodloužené hřídeli motoru poháněnjícího stůl. Otočné stoly na pracovišti mají dvě pracovní polohy.

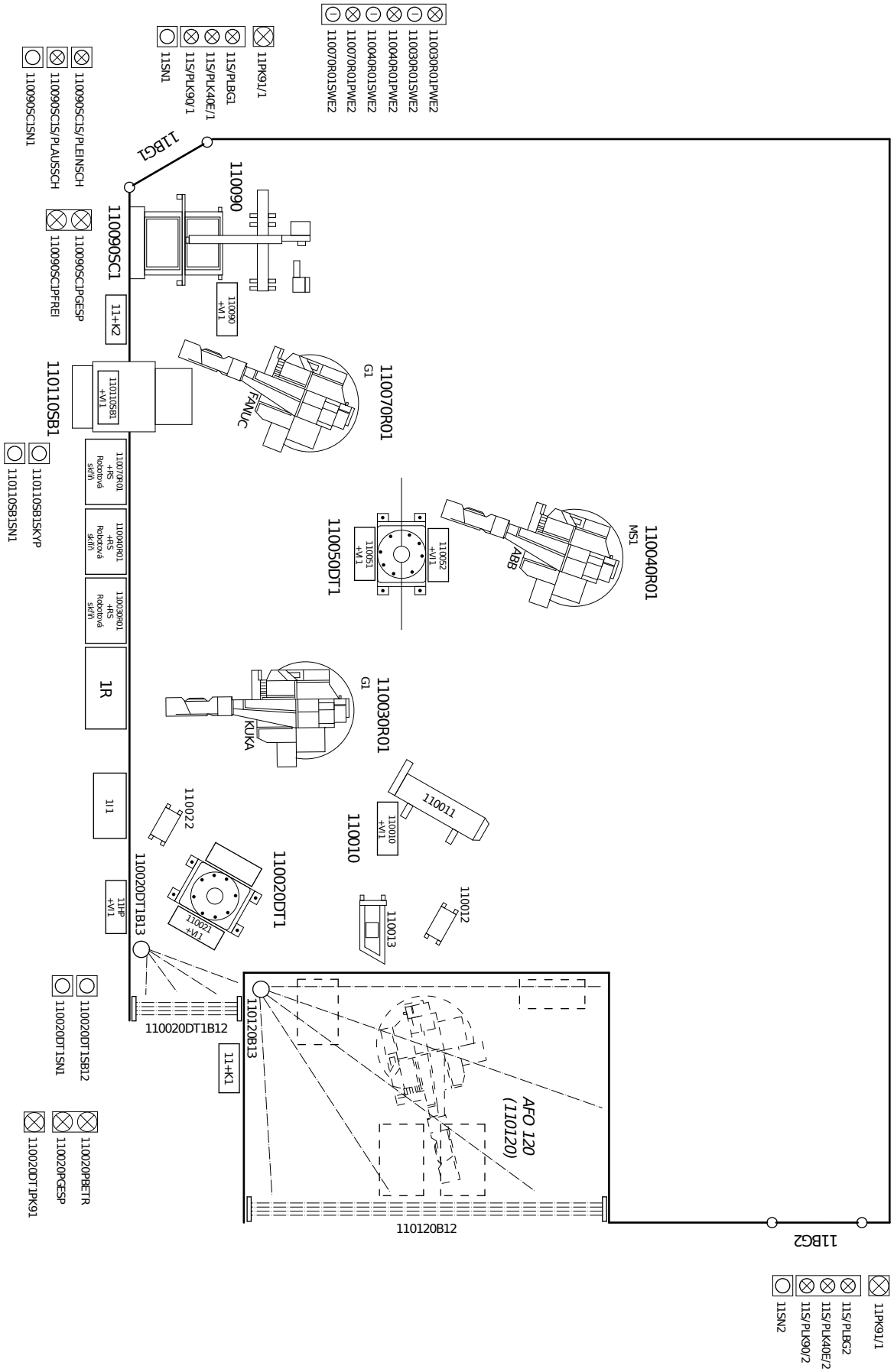
3.3 Sekvence pracoviště

Výrobní sekvence celého pracoviště je rozdělena do několika operací, z nichž některé mohou být prováděny nezávisle na ostatních, zatímco jiné jsou naopak podmíněny dokončením jiné operace popřípadě několika operací. Následující text představuje celkovou sekvenci pracoviště a ilustruje závislost jednotlivých operací.



Pozn.: Po inicializaci je svařovací stůl natočen přípravkem 110051 k svařovacímu robotu.

Obr. 3.3: Celková výrobní sekvence pracoviště



Obr. 3.2: Layout pracoviště

Obrázek znázorňuje výrobní sekvenci pracoviště E-robot z pohledu závislosti jednotlivých operací. Detaily konkrétních operací jsou popsány v kapitolách daných pracovních stanic.

Číslované obdélníky zobrazují operace, jejich dokončení reprezentují z nich vycházející šipky. Vodorovné linie naznačují podmínění pokračování v sekvenci dokončením několika operací. Ústí-li více šipek do vodorovné linie, je pro pokračování nutné dokončení všech těchto operací.

Zpětné vazby naznačené přechodem z operací 14 a 15 se stejně jako operace 16 neuplatní při prvním průchodu pracovním cyklem.

Diagram plyne shora dolů, takže po inicializaci pracoviště mohou být vykonávány operace 1, 2 a 3. Protože se jedná o první cyklus, je pro pokračování do operace 4 nutné splnění pouze operace 1 a 2. Stejně tak pro začátek operace 5 stačí při prvním cyklu dokončit pouze operaci 4. Šipka směřující z operace 14 do operace 2 naznačuje, že pro založení kroužků musí být stůl napolohován svařovacím přípravkem 110051 k svařovacímu robotu.

3.3.1 Slovní popis sekvence

Pro názornost je v dalším odstavci uveden slovní popis sekvence.

V **operaci 1** robot KUKA (110030R01) olemuje střechu ve stanici 110020. Mezitím robot FANUC (110070R01) do svařovacího přípravku 110052 založí trojici kroužků **(operace 2)**. Po dokončení obou operací, odebere robot KUKA olemovanou střechu ze stanice 110020 a provede HandShake s robotem FANUC **(operace 4)**. Robot FANUC následně založí střechu do svařovacího přípravku 110052 a dojde k otočení stolu **(operace 5)**. Během zakládání robot KUKA olemuje plech podlahy ve stanici 110010 **(operace 3)**. Během svařování střechy robotem ABB (110040R01) **(operace 6)**, provede robot KUKA s robotem FANUC HandShake podlahy **(operace 7)** a následně ji robot KUKA založí do svařovacího přípravku 110051. Po založení podlahy robot FANUC odebere dvojici profilů ze zakládacího šuplíku 110090 a založí je k podlaze do svařovacího přípravku 110051 **(operace 8)**. Po dokončení těchto operací a svaření střechy dojde k otočení svařovacího stolu. Svařovací přípravek 110051 je směrem k svařovacímu robotu ABB a může dojít k svaření profilů s podlahou **(operace 10)**. Ze svařovacího přípravku 110052 je robotem KUKA vyjmuta svařená střecha **(operace 11)**. Robot KUKA s robotem FANUC provedou HandShake střechy. Po dokončení těchto operací (10 a 11) dojde k otočení stolu **(operace 12)**. Robot FANUC založí střechu k podlaze **(operace 13)**. Mezitím může robot KUKA začít připravovat novou střechu a podlahu (operace 1 a 3). Po provedení operace 13 se stůl opět otočí **(operace 14)** a robot ABB začne svařovat celkovou sestavu krmítka **(operace 15)**. Protože operace 14 byla dokončena, je robotu FANUC umožněno založit nové kroužky do stanice 110052 **(operace 2)**. Sekvence pokračuje až do dokončení operace 4 stejně jako během prvního cyklu. Narozdíl od něj se ale stůl otočí až po dokončení operace 15. Operace 7 je podmíněna vyjmutím hotového krmítka (operace 16)

robotem FANUC. Hotové krmítko je odloženo na dopravník 110110.

3.4 Roboti

Odstavec je zaměřen na funkční popis robotů pracoviště. V sekcích věnovaných jednotlivým robotům je představen jejich gripper a kompletní výčet operací, které robot provádí.

Ačkoliv je výčet prací robotů v tabulce číslován, pořadí operací nemusí odpovídat reálnému pořadí během výrobního procesu. Identifikátor operace slouží pouze pro přířazení unikátního čísla dané operaci a je s ním dále operováno v rámci automatického cyklu stanice.

Mezi jednotlivými operacemi může robot libovolně přeskakovat (pokud je k tomu vyzván PLC a umožňují to jeho vnitřní programové podmínky).

Typy operací

Operace robota jsou klasikovány typem. Ty jsou rozlišovány na tři: vjezd, výjezd a vjezd/výjezd a odkazují na rozdílné chování řídících signálů během těchto operací (viz 5.6).

Z technologického hlediska je rozdíl mezi operacemi v poloze koncového bodu robota před a po dokončení operace vzhledem k možné kolizní zóně. Rozdíl mezi typy uvádí tabulka.

Typ operace	Poloha robota		Příklad
	Před operací	Po operaci	
Vjezd/výjezd	Mimo kolizi		Olemování dílu
Vjezd	Mimo kolizi	Uvnitř kolize	Zakládání do přípravku vyžadující fixaci v přípravku
Výjezd	Unitř kolize	Mimo kolizi	Vyjmutí dílu zpřípravku vyžadující odepnutí v přípravku

Tab. 3.1: Typy robotických operací

Každé operaci typu výjezd musí předcházet operace typu vjezd. Stejně tak není možné, aby operace typu vjezd neměla přidruženou operaci typu výjezd.

3.4.1 Lemovací robot

Lemovací robot KUKA (110030R01) opracovává díly střechy a podlahy na stanicích 110010 a 110020. Dále provádí Handlingové operace s robotem FANUC a zakládá olemovanou podlahu do přípravku 110051.

Jeho gripper je vybaven dvojicí lemovacích rolen pomocí nichž provádí samotné lemování. K manipulačním operacím využívá magnety jejichž polohu kontroluje pomocí pneumatických válců. Jedna poloha magnetů díl z gripperu uvolní, druhá ho přichytí.

ID	Popis	Typ operace
Operace na 110020		
1	Lemování střechy pozice 1	vjezd/výjezd
2	Přeložení střechy	vjezd/výjezd
3	Založení do lemovací pozice 2	vjezd
4	Odjezd po založení	výjezd
5	Lemování střechy pozice 2	vjezd/výjezd
6	Uchopení olemované střechy	vjezd
7	Odjezd s olemovanou střechou	výjezd
8	HandShake střechy s robotem FANUC	vjezd/výjezd
Operace na 110010		
9	Založení podlahy na pozici 1	vjezd/výjezd
10	Lemování podlahy pozice 1	vjezd/výjezd
11	Přeložení podlahy	vjezd/výjezd
12	Založení do lemovací pozice 2	vjezd/výjezd
13	Lemování podlahy pozice 2	vjezd/výjezd
14	HandShake podlahy s robotem FANUC	vjezd/výjezd
Operace na 110051		
15	Založení podlahy do přípravku	vjezd/výjezd
HandShake polotovaru střechy		
16	HandShake polotovaru střechy s robotem FANUC	vjezd/výjezd

Tab. 3.2: Operace prováděné robotem KUKA

3.4.2 Manipulační robot

Manipulační robot FANUC (110070R01) vybírá kroužky a profily ze zakládacího šuplíku 110090 a zakládá je do svařovacích přípravků 110051 a 52. Do přípravku 110052 zakládá olemovanou střechu, kterou pomocí HandShake přebírá od robota KUKA. Z přípravku 110051 pak odebírá hotovou sestavu krmítka a odkládá ji na pásový dopravník.

Protože gripper robota FANUC je poměrně komplikovaný a jeho popis by byl vyčerpávající, je pouze konstatováno, že pro pro handlingové operace je FANUC narozdíl od KUKA robota vybaven přísavkami a operace vybírání kroužků a profilů 110090 umožňuje buď malá přísavka a nebo rozpěrný mechanismus. Gripper je dále vybaven kleštěmi s roztečí rovnou rozteči profilů hotové sestavy krmítka.

ID	Popis	Typ operace
Operace na 110090		
1	Odebrání kroužku	vjezd/výjezd
2	Odebrání profilu	vjezd/výjezd
Operace na 110052		
3	Založení kroužku 1	vjezd/výjezd
4	Založení kroužku 2	vjezd/výjezd
5	Založení kroužku 3	vjezd/výjezd
6	HandShake střechy	vjezd/výjezd
7	Založení střechy	vjezd/výjezd
8	Vyjmutí svařené střechy	vjezd/výjezd
9	HandShake střechy	vjezd/výjezd
Operace na 110051		
10	Založení střechy k podlaze	vjezd/výjezd
11	Založení profilů k podlaze	vjezd
12	Odjezd po založení profilů	výjezd
13	Vyjmutí hotového krmítka	vjezd/výjezd
Operace na 110110		
14	Odložení hotového krmítka na dopravník	vjezd/výjezd

Tab. 3.3: Operace prováděné robotem FANUC

3.4.3 Svařovací robot

Svařovací robot ABB (110040R01) v přípravku 110052 přivařuje kroužky k plechu střechy a v přípravku 110051 sloupky k plechu podlahy a nakonec celou sestavu krmítka.

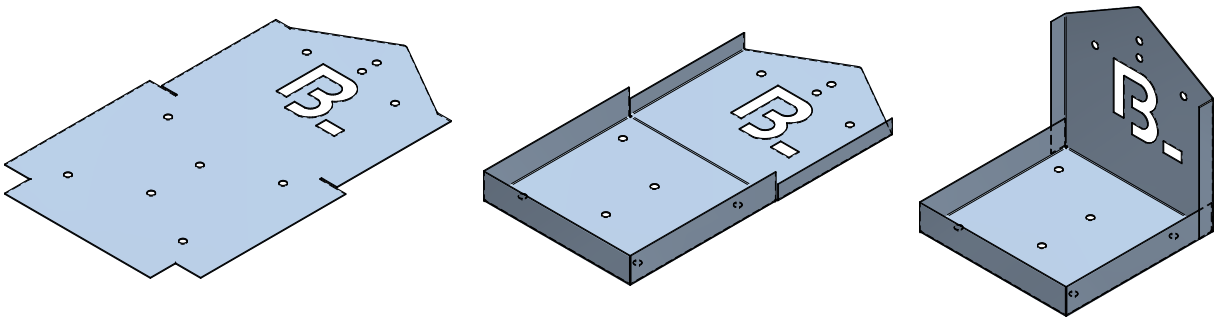
Robot je vybaven svařovací tryskou a navijákem drátu. Součástí robotického pracoviště je i štípačka a čistička drátu. Štípačka a čistička je sice formálně řízená pomocí PLC, ale povely pro spínání jejích aktuátorů si řídí robot.

ID	Popis	Typ operace
Operace na 110052		
1	Přivaření kroužků strana A	vjezd/výjezd
2	Přivaření kroužků strana B	vjezd/výjezd
Operace na 110051		
3	Přivaření profilů k podlaze	vjezd/výjezd
4	Přivaření střechy k podlaze	vjezd/výjezd

Tab. 3.4: Operace prováděné robotem FANUC

3.5 110010 - lemování podlahy

Na stanici 110010 dochází k robotickému lemování plechu podlahy (operace 3 na obr. 3.3). Stanice se skládá z zakládacího přípravku 110013, lemovacího přípravku 110011 a překládacího přípravku 110012.



Obr. 3.4: Jednotlivé kroky procesu lemování podlahy

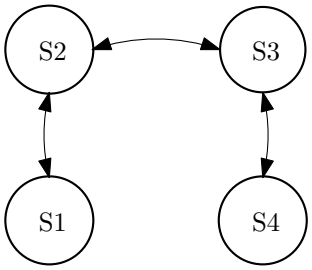
Zakládací přípravek slouží pro založení plechu podlahy. Přípravek je vybaven dvěma čidly přítomnosti. Na základě kombinace jejich stavů je detekována správnost založení a orientace dílu. Ta rozhoduje o způsobu odebírání dílu robotem.

Stav čidel		Stav přípravku	Způsob odebrání
ST02	ST01		
0	0	Díl nezaložen	x
0	1	Díl založen	”Shora”
1	0	Chybový stav	x
1	1	Díl založen	”Sdola”

Tab. 3.5: Stavy zakládacího přípravku

Lemovací přípravek obsahuje dvě lemovací pozice, na nichž je upínán díl. Na první pozici je dílu nejprve olemováno pět hran, na druhé pozici je díl olemován do finálního tvaru. Přípravek se skládá ze čtyř válců. Upínací statický válec ZS01, pohyblivý upínací

válec ZS02 a dva válce přesunu ZS03a a ZS03b. Čtyři přípustné koncové stavy přípravku ilustruje statový automat na obrázku 3.5.

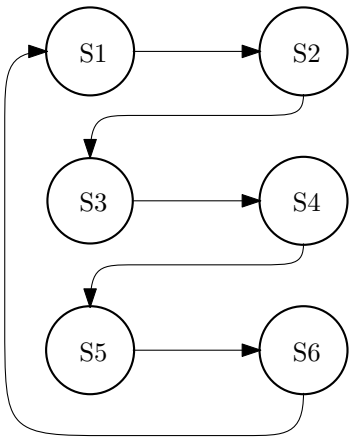


Stav FSM		Požadovaný stav aktuátorů		
		Upínací válce		Válce přesunu
ID	Popis	Statický pozice 1	Pohyblivý pozice 2	
1	Upnuto poz. 1	PP	PP	VP
2	Pozice 1	VP	VP	VP
3	Pozice 2	VP	VP	PP
4	Upnuto poz. 2	VP	PP	PP

Obr. 3.5: Stavový automat lemovacího přípravku

V **překládacím přípravku** dochází k přeuchopení dílu po dokončení prvního lemování. Vzhledem k tomu, že překládání probíhá roboticky a je definována orientace dílu, je detekce zajištěna jen jedním čidlem přítomnosti.

Automatická sekvence stanice je znázorněná diagramem 3.6.



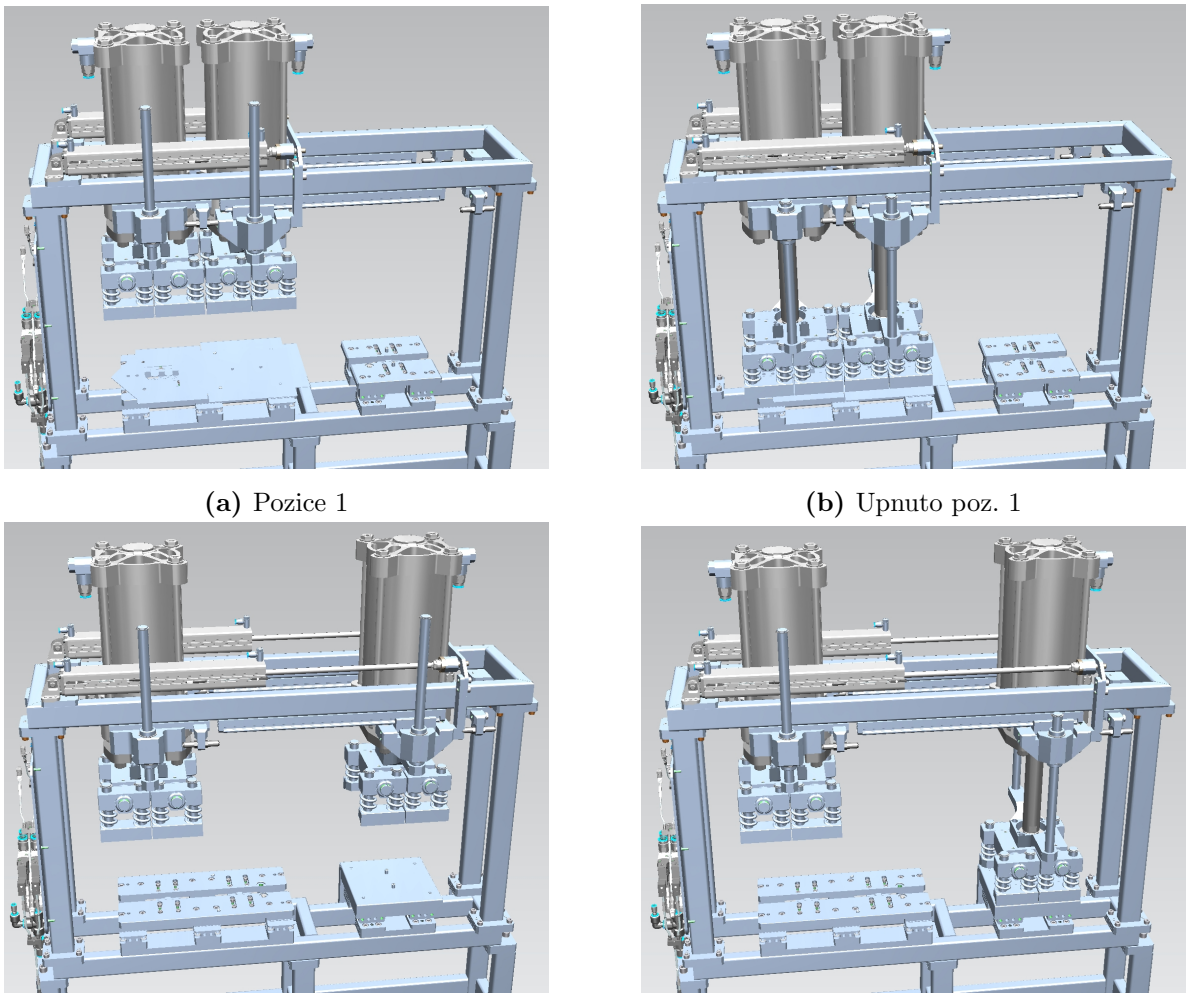
Krok sekvence		Podmínka přechodu	Požadovaný stav FSM
ID	Popis		
1	Založení 110013	Založeno	2
2	Založení 110011 poz. 1	Založeno	2
3	Lemování 1	Lemování 1 hotovo	1
4	Založení 110011 poz. 2 (překlad 110012)	Založeno	3
5	Lemování 2	Lemování 2 hotovo	4
6	Odberání dílu	Díl odebrán	2

Obr. 3.6: Sekvence stanice 110010

Podmínky přechodu stanice jsou řízeny převážně signály *Hotovo* robota KUKA. Je to dáno tím, že většinu technologických operací provádí právě robot. Výjimku tvoří přechod z kroku 1 a 6.

Přechod z prvního kroku je závislý na založení dílu do 110013. Podle konfigurace pracoviště může zakládání probíhat roboticky (zakládací robot stanice AFO120) a nebo je uskutečněno obsluhou. V případě robotického zakládání je tedy přechod řízen signálem *Hotovo* zakládacího robota, v případě ručního zakládání signálem odbavovacího tlačítka.

Přechod z kroku 6 je naproti tomu podmíněný stavem ostatních operací pracoviště (viz. 3.3 operace 3).



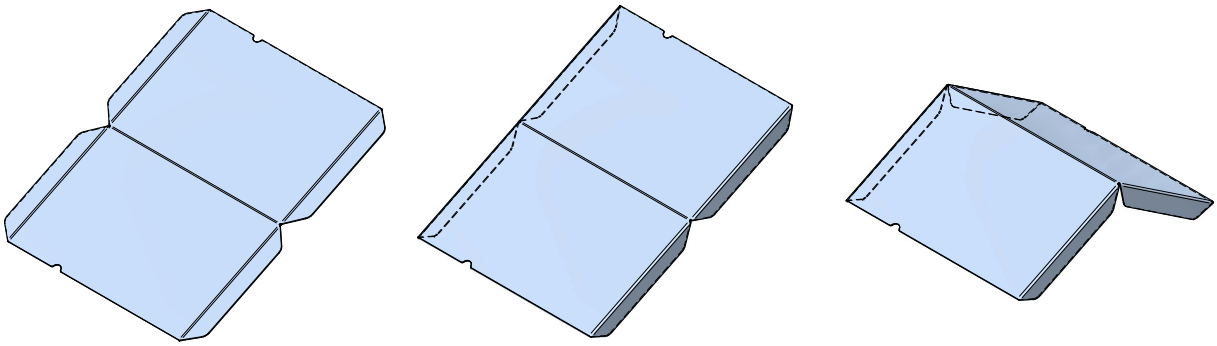
(a) Pozice 1 **(b)** Upnuto poz. 1

(c) Pozice 2 **(d)** Upnuto poz. 2

Obr. 3.7: Pracovní pozice lemovacího přípravku 110010

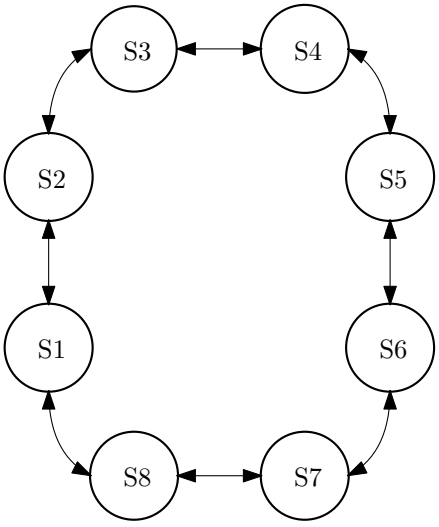
3.6 110020 - lemování střechy

Na stanici 110020 dochází k robotickému lemování plechu střechy (obr. 3.3 operace 1). Stanice se skládá z lemovacího přípravku 110021, překládacího přípravku 110022 a otočného stolu 110020DT1. Bezpečnostními prvky stanice jsou světelná závora 110020DT1B12 a scanner přítomnosti osob v zakládacím prostoru 110020DT1B13. Pracoviště je dále vybaveno tří segmentovým signalizačním majákem a odbavovacím tlačítkem 110020DT1SN1.



Obr. 3.8: Jednotlivé kroky procesu lemování střechy

Lemovací přípravek je konstrukčně i funkčně téměř shodný s lemovacím přípravkem 110011. Přípravek 110021 má navíc fixační válec pro uchycení dílu po přeložení v překládací stanici. Po založení robotem do druhé pozice válec přídrží díl a k samotnému upnutí lemovacím přípravkem dojde až po odjezdu robota. Dalším rozdílem je způsob zakládání dílu do přípravku. Zatímco do přípravku 110011 zakládá robot, do přípravku 110021 zakládá obsluha. Vhzhledem k těmto rozdílům jsou rozdílné i sekvence stanic.



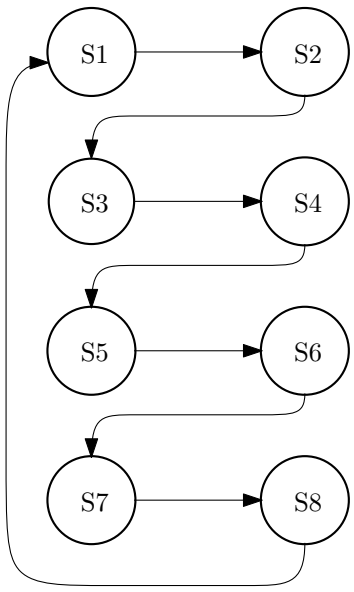
Stav FSM		Požadovaný stav aktuátorů		
ID	Popis	Poloha otočného stolu	Lemovací přípravek	Fixační válec
1	Zakládací poloha	Obsluha	2	VP
2	Upnutí poz. 1	Obsluha	1	VP
3	Lemování 1	Robot	1	VP
4	Příprava na fixování poz.2	Robot	3	VP
5	Fixováno poz. 2	Robot	3	PP
6	Lemování 2	Robot	4	PP
7	Pracovní poloha 1	Robot	2	PP
8	Pracovní poloha 1	Robot	2	VP

Obr. 3.9: Stavový automat stanice 110020

Otočný stůl nese lemovací přípravek a umožňuje jeho polohování do zakládací (směrem k obsluze) a pracovní polohy (směrem k robotu). Otočný stůl je poháněn tří fázovým asynchronním motorem a jeho pozice je detekována pomocí vačkových snímačů umístěných na prodloužené hřídeli rotoru. Stůl navíc obsahuje FailSafe čidla polohy.

Překládací stanice plní stejnou funkci jako překládací stanice na stanici 110010 a je vybavena stejnou senzorikou.

Automatická sekvence stanice je znázorněna diagramem 3.10.

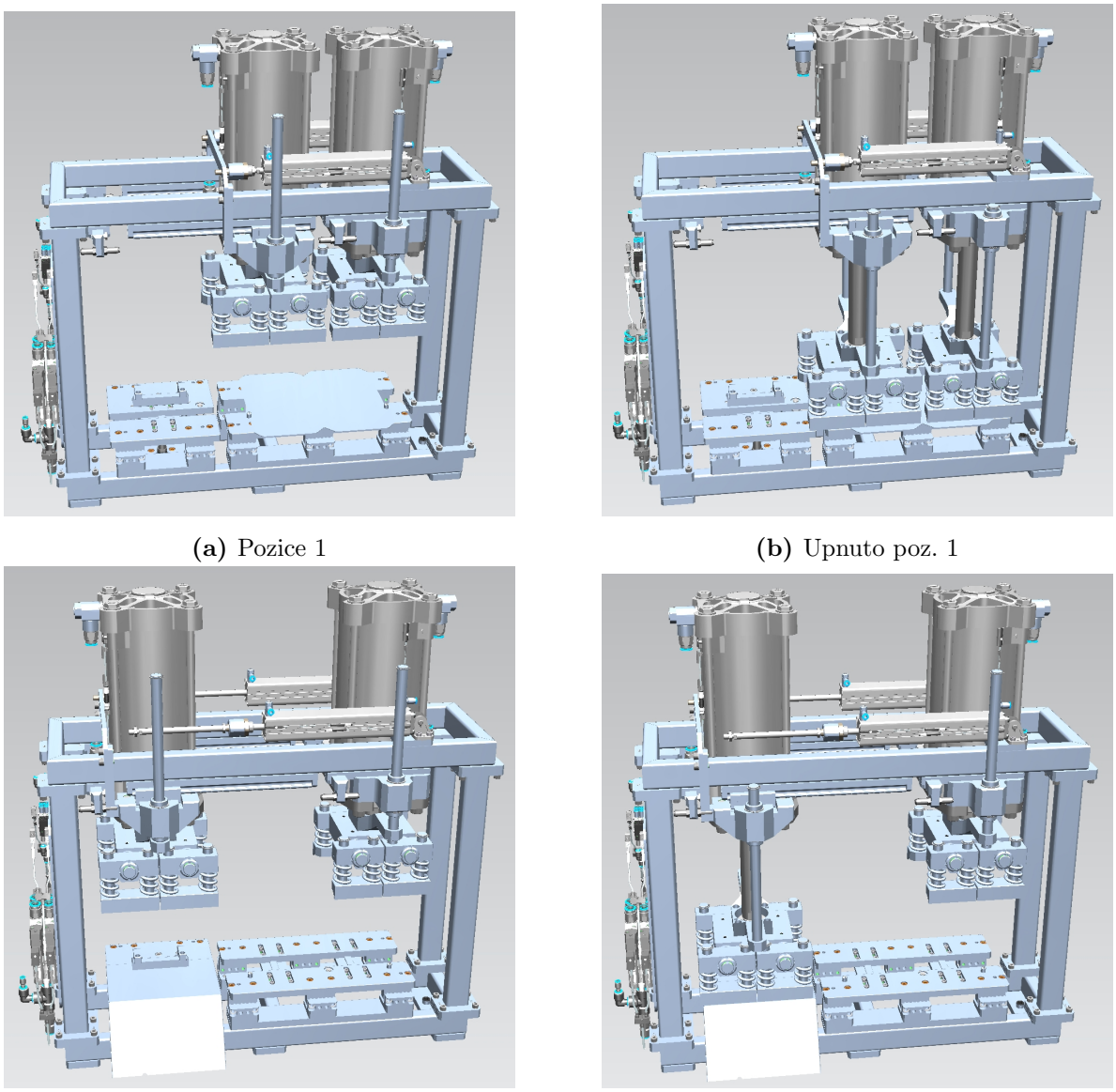


Krok sekvence ID	Krok sekvence Popis	Podmínka	Požadovaný stav FSM
1	Založení obsluhou	Založeno a odbaveno	1
2	Upnutí poz 1	Upnuto poz 1	2
3	Lemování 1	Lemování 1 hotovo	3
4	Založení poz. 2	Založeno 110021 poz.2 (překlad 110022)	4
5	Fixace poz. 2	Fixováno poz. 2	5
6	Lemování 2	Lemování 2 hotovo	6
7	Příprava k odebrání dílu	Odepnuto	7
8	Odebrání dílu	Díl odebrán	8

Obr. 3.10: Sekvence pracoviště 110020

Podmínky přechodu jsou převážně řízeny signály *Hotovo* robota KUKA. Výjimku tvoří pouze přechod z kroku 1. Protože zakládání provádí operátor, je přechod řízen odbavovacím tlačítkem.

Přechod z kroku 8 je naproti tomu podmíněný stavem ostatních operací pracoviště (viz. 3.3 operace 1)



(a) Pozice 1 **(b)** Upnuto poz. 1

(c) Pozice 2 **(d)** Upnuto poz. 2

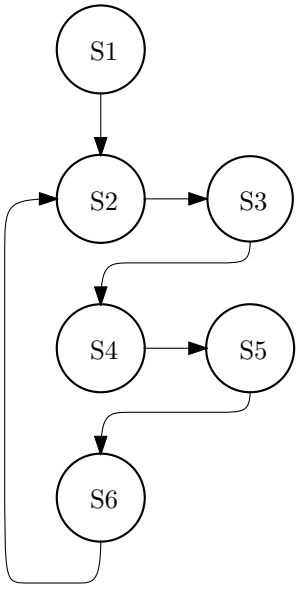
Obr. 3.11: Pracovní pozice stanice 110020

3.7 110050 - svařovací stanice

Ve stanici jsou dokončeny podsestavy podlahy a střechy a dochází v ní i ke kompletaci celé sestavy (viz. obr. 3.3 operace 2, 4, 6, 7, 8, 10, 11, 13, 15 a 16). Z pohledu technologie se tak jedná o poslední článek výrobního řetězce. Svařovací stanice 110050 se skládá z otočného stolu 110050DT1 a dvou svařovacích přípravků (110051 a 110052). Podle polohy otočného stolu je jeden přípravek natočen k svařovacímu robotu ABB a druhý do pracoviště.

Oba svařovací přípravky v rámci stanice mají vlastní automatickou sekvenci a jsou popsány svým stavovým automatem. Úkolem automatické sekvence stanice je synchronizace těchto dvou podružných sekvencí a řízení otočného stolu.

Automatická sekvence stanice přechází do dalšího kroku v případě, že automatické



Krok sekvence		Natočení stolu		Podmínka přechodu Krok sekvence	
ID	Popis	Svař.	Zakl.	AFO51	AFO52
1	Kompletace dílů střechy	AFO51	AFO52	1	3
2	Kompletace podlahy Svařování střechy	AFO52	AFO51	5	5
3	Vyjmutí střechy Svařování podlahy	AFO51	AFO52	6	6
4	Kompletace krmítka	AFO52	AFO51	7	6
5	Kompletace střechy Svařování krmítka	AFO51	AFO52	8	3
6	Vyjmutí hotového krmítka	AFO52	AFO51	1	5

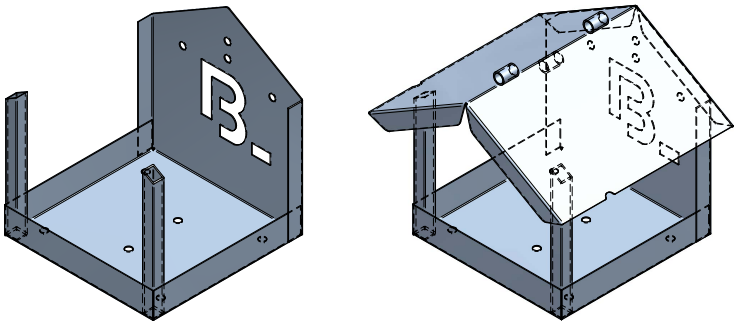
Obr. 3.12: Automatická sekvence svařovací stanice

sekvence obou svařovacích přípravků se nacházejí v požadovaném kroku a otočný stůl je správně napolohován. Tím je zajištěna synchonizace sekvencí a je zaručeno, že jedna nepředběhne druhou.

Podmínky přechodu obou přípravků jsou řízeny výhradně na základě stavů ostatních operací pracoviště a signálů Hotovo robotů. Pro přechody v rámci jiných operací opět viz obr. 3.3

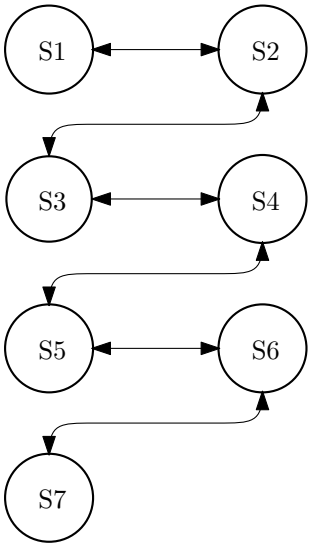
3.7.1 110051 - svařovací přípravek podlahy

Ve svařovacím přípravku 110051 je nejprve dokončena podsestava podlahy (viz. obr. 3.3 operace 7, 8 a 10) a následně celá sestava krmítka (viz. obr. 3.3 operace 13, 15 a 16). Přípravek je vybaven několika sadami pneumatických válců pro uchycení různých částí sestavy. Některé válce fixují samotnou podsestavu podlahy, popřípadě profily a podlahu. Další sada fixuje celkovou sestavu (podlahu spolu se střechou). Další skupina válců slouží k přesunu těchto fixačních prvků.



Obr. 3.13: Jednotlivé kroky procesu sváření v přípravku 110051

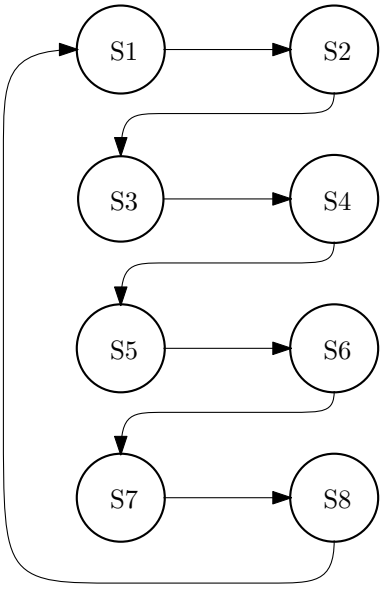
Svařovací přípravek 110051 je objektivně nejsložitějším celkem pracoviště. Vzhledem k množství aktuátorů, je pro přehlednost v tabulce stavového automatu uveden pouze slovní popis stavů bez výčtu jmen a požadovaných stavů jednotlivých aktuátorů. Stavy, jejichž popis obsahuje slovo *příprava*, odkazují na situaci, kdy je nutné určitou skupinu fixačních válců přesunout do pracovní pozice pomocí válců přesunu. Z tabulky a diagramu je zřejmé, že pro přechod mezi stavy existuje jen jedna povolená cesta. Nedodržení této cesty by vedlo ke kolizím.



Stav FSM	
ID	Popis
1	Odepnuto vše
2	Příprava upnutí podlahy
3	Upnutí podlahy
4	Příprava upnutí sloupků
5	Upnutí sloupků
6	Příprava upnutí celku
7	Upnutí celku

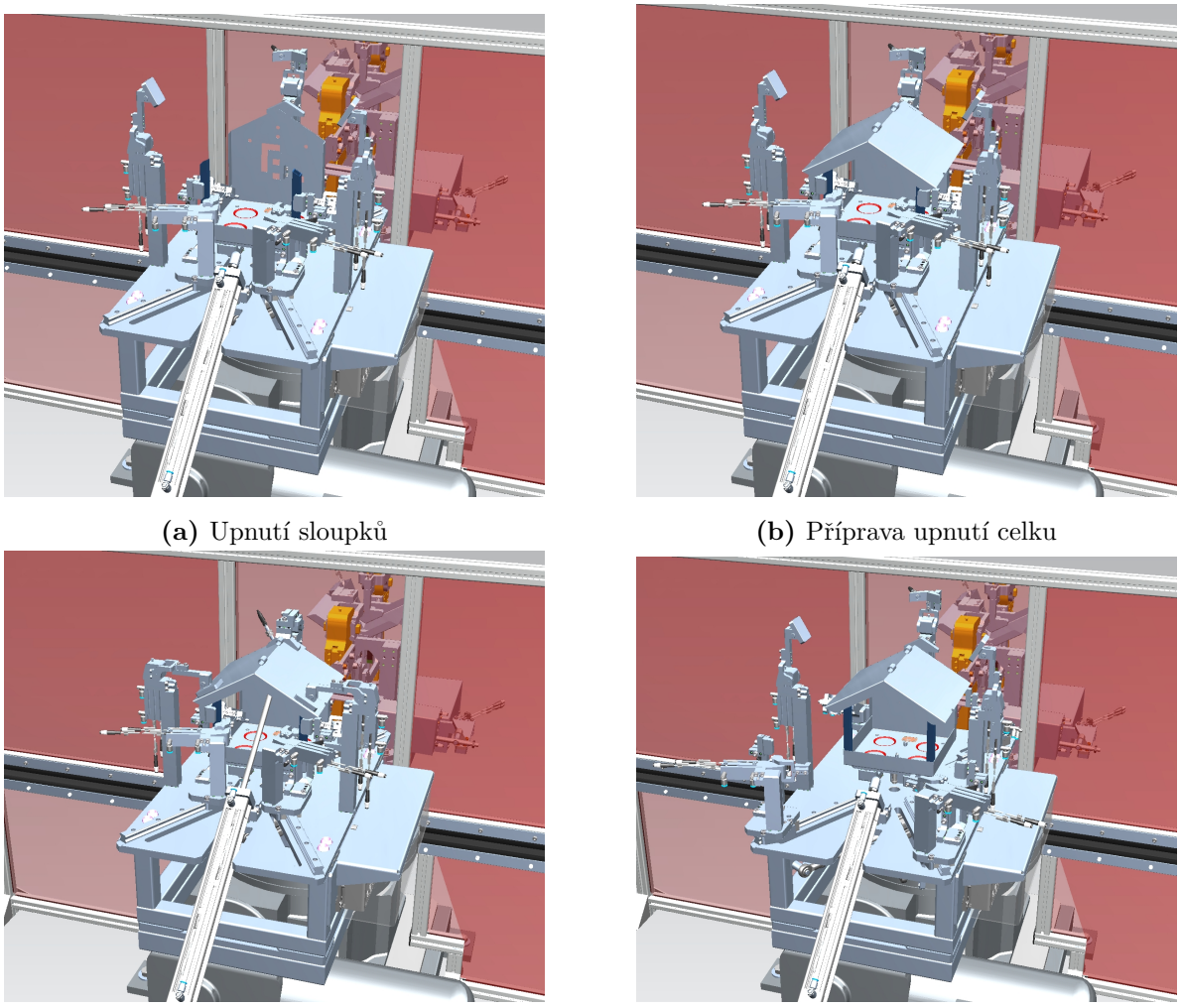
Obr. 3.14: Stavový automat stanice AFO51

Automatická sekvence přípravku



Krok sekvence		Podmínka přechodu	Požadovaný stav FSM
ID	Popis		
1	Založení podlahy	Založeno	1
2	Upnutí podlahy	Upnuto	3
3	Založení profilů	Profily založeny	3
4	Upnutí profilů	Upnuto	5
5	Přivaření profilů k podlaze	Svařeno	5
6	Založení střechy k podlaze	Založeno	5
7	Svaření sestavy	Svařeno	7
8	Odebrání hotové sestavy	Odebráno	1

Obr. 3.15: Automatická sekvence svařovacího přípravku 110051



(a) Upnutí sloupků

(b) Příprava upnutí celku

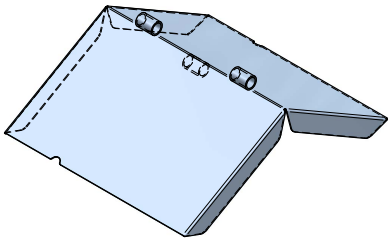
(c) Upnutí celku

(d) Odepnuto vše

Obr. 3.16: Pracovní pozice svařovacího přípravku 110051

3.7.2 110052 - svařovací přípravek střechy

V přípravku 110052 je dokončena podsestava střechy. Přípravek fixuje kroužky a olemovanou střechu a umožňuje jejich svaření.

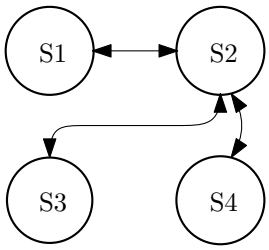


Obr. 3.17: Hotová podsestava střechy po zavaření kroužků

Svařovací přípravek 110052 upíná střechu a kroužky pomocí sady upínacích válců. Další dvojice válců (ZS02 a ZS03) tvoří naklápěcí mechanismus. Kombinace poloh těchto válců dovoluje naklopit celý přípravek do tří pozic: 0°, -45° a +45°. To umožňuje přivařit jednotlivé kroužky ke střeše dostatečně kvalitně.

Stav válců		Stav
ZS02	ZS03	náklonu
VP	VP	Chybový stav
VP	PP	-45°
PP	VP	0°
PP	PP	+45°

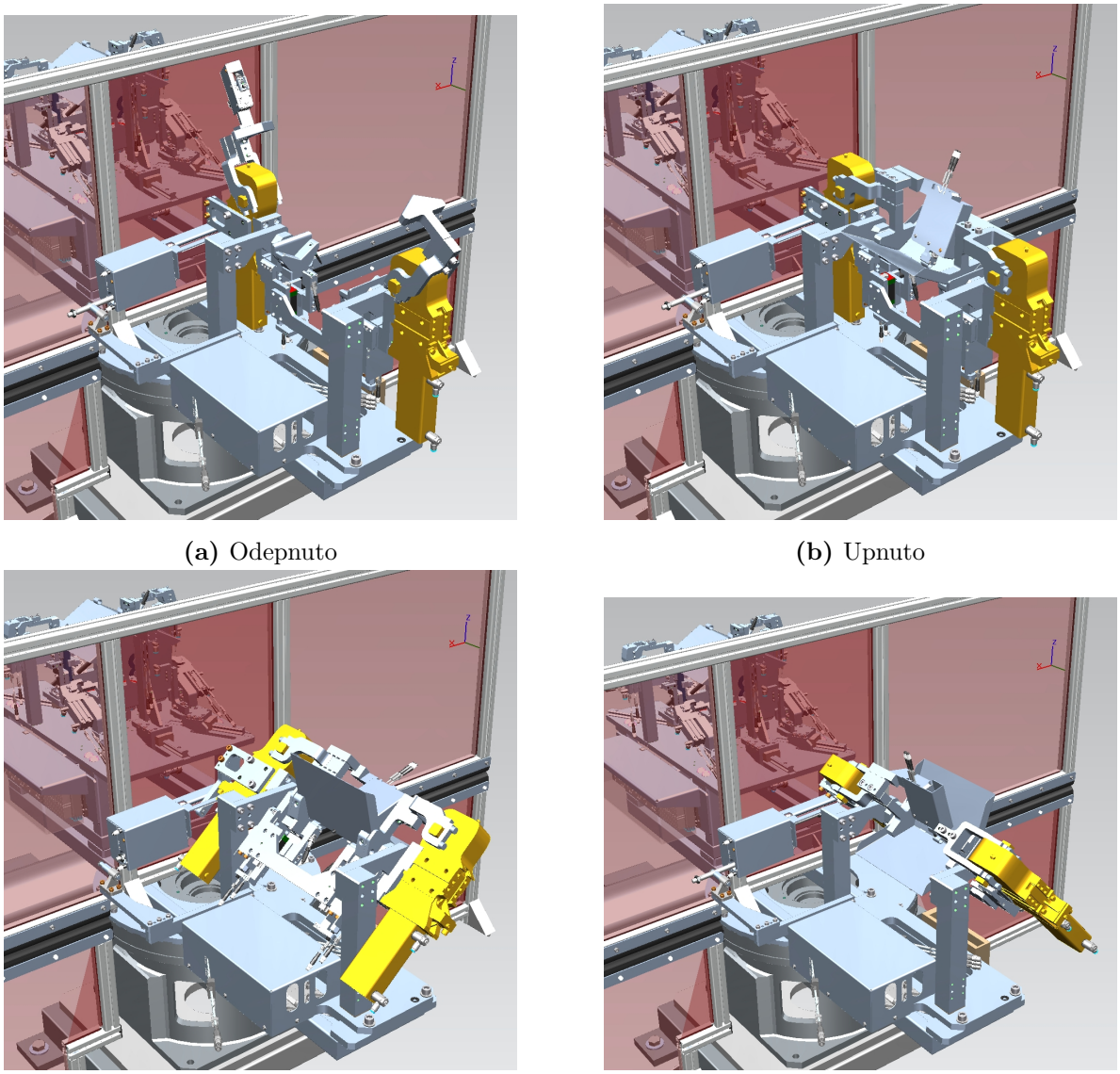
Tab. 3.6: Poloha naklápěcího mechanismu



Stav FSM		Požadovaný stav aktuátorů			
		Upínací válec			Náklon přípravku
ID	Popis	střechy	vnějších kroužků	vnitřního kroužku	
1	Odepnuto	VP	VP	VP	0°
2	Upnuto	PP	PP	PP	0°
3	Pozice A	PP	PP	PP	45°
4	Pozice B	PP	PP	PP	-45°

Obr. 3.18: Stavový automat stanice AFO52

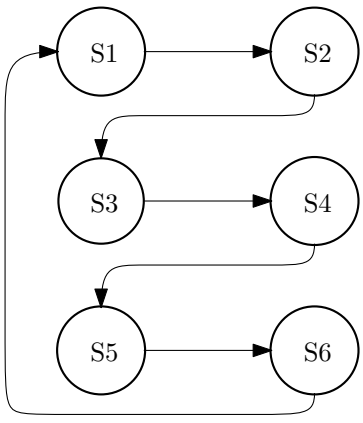
Automatická sekvence



(a) Odepnuto **(b)** Upnuto

(c) Pozice A **(d)** Pozice B

Obr. 3.19: Pracovní pozice svařovacího přípravku 110052



Krok sekvence		Podmínka přechodu	Požadovaný stav FSM
ID	Popis		
1	Založení kroužků	Kroužky založeny	1
2	Založení střechy	Střecha založena	1
3	Svaření kroužků strana A	Zavařeno	3
4	Svaření kroužků strana B	Zavařeno	4
5	Odebrání podsestavy	Odebráno	1
6	Založení střechy k podlaze	Založeno	1

Obr. 3.20: Automatická sekvence přípravku 110052

Podmínky přechodu přípravku jsou řízeny výhradně na základě stavů ostatních operací pracoviště a signálů Hotovo robotů. Pro přechody v rámci jiných operací viz obr. 3.3

4

Softwarová architektura

Cílem kapitoly je představit zvolenou softwarovou architekturu. Popis je v každé podkapitole strukturovaný způsobem shora dolů: nejprve jsou definované obecné požadavky, následně interface a nakonec implementace. Implementace je znázorněna zejména formou vývojových diagramů, stavových automatů a blokových schémat.

První část textu představí obecné požadavky kladené na architekturu bez popisu jejich implementace a stručně zdůvodňuje výhody jejich zavedení.

Druhá podkapitola v obecných rysech popisuje členění softwaru na komponenty a způsob jakým mezi nimi dochází k výměně dat.

4.1 Obecné požadavky:

1. ***Modularita programu***

Modulární program se skládá z na sobě nezávislých modulů, které mezi sebou komunikují pomocí interface. Omezují se tím vazby mezi částmi programu a kód se pak stává lépe čitelný. Jednotlivé moduly jsou kompaktní a zaměřené na jednu konkrétní funkcionalitu, což přispívá k jejich snadné údržbě a testovatelnosti [9] Jak bude ukázáno následně, jedná se o klíčovou vlastnost softwarové architektury.

2. ***Error Handling***

Architektura musí být připravena zohlednit možné chybové stavy na nejrůznějších úrovních řízení a vhodně na ně reagovat. Z toho pohledu je žádoucí rozlišovat několik kategorií chyb (safety-critical error, alarm, informační hláška, varování) a na každou být schopen reagovat odlišně: u některých například vyžadovat kvitování a u jiných ne.

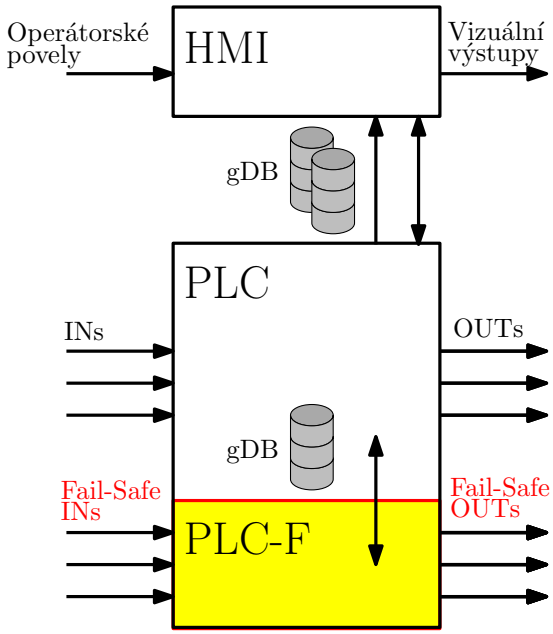
3. ***Vizualizace a režimy pracoviště***

Nedílnou součástí pracoviště je operátorský panel, informující obsluhu o stavech jednotlivých prvků linky. Panel zároveň slouží k přepínání mezi různými režimy a k jejich obsluze.

Výše popsané vlastnosti tvoří základ celé architektury a jsou esenciálními prvky pro všechny softwarové konstrukce. Jejich konkrétní implementace je detailně popsána v následujících odstavcích.

4.2 Komponenty

Dělení software na komponenty výchází z různé povahy zpracovávaných vstupů a poskytovaných výstupů, a dále z dělení software na PLC a HMI část. Pro výměnu informací mezi komponentami slouží gDB.



Obr. 4.1: Rozdělení na komponenty

gDB tedy zastávají roli interface a podle povahy vyměňovaných informací mohou zajišťovat obousměrnou, popřípadě jednosměrnou komunikaci.

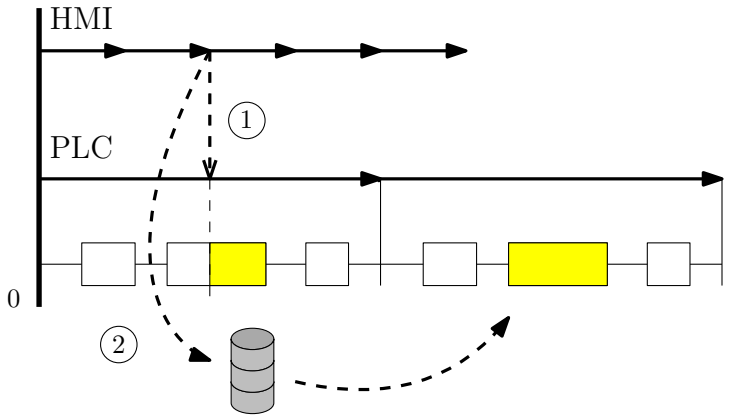
4.2.1 Architektura komponent

V obrázku je naznačeno, že v rámci obou hlavních komponent (PLC a HMI) je snaha dodržet princip modularity, a to i přes různý způsob implementace. V zásadě platí, že modul vytvořený v komponentě PLC má svůj obraz i v komponentě HMI.

Tento přístup tak umožňuje monitorovat a ovládat každý prvek pracoviště, který má v programu vytvořený svůj modul. Další výhodou je snadná modifikovatelnost a replikovatelnost; jednotlivé moduly mají v praxi několik instancí, takže pro změnu všech instancí modulu (například změna barevného schématu prvku válce na HMI) stačí modifikovat kód na jednom místě v programu.

4.2.2 Synchronizace mezi komponentami:

Protože PLC program i program operátorského panelu mají různě rychlý oběh programu, mohlo by docházet k hazardním stavům při nekoordinovaném čtení z HMI interface na straně PLC (situace (1)). Modul by pak nemusel mít platnou hodnotu signálu po celou dobu oběhu své programové smyčky, což by mohlo působit problémy například při ručním ovládání aktuátorů.



Obr. 4.2: Synchronizace mezi komponentami

Z tohoto důvodu jsou hodnoty z interface DB čteny vstupními interface jednotlivých modulů (situace (2)); tím je zaručeno, že uvnitř modulu, který k interface přistupuje, je hodnota platná po celou dobu vykonávání podprogramu modulu. V opačné situaci, kdy výstupní signály z PLC modulů vstupují přes interface do HMI, není synchronizaci nutné řešit, protože v takovém případě slouží HMI jen jako zobrazozovací člen.

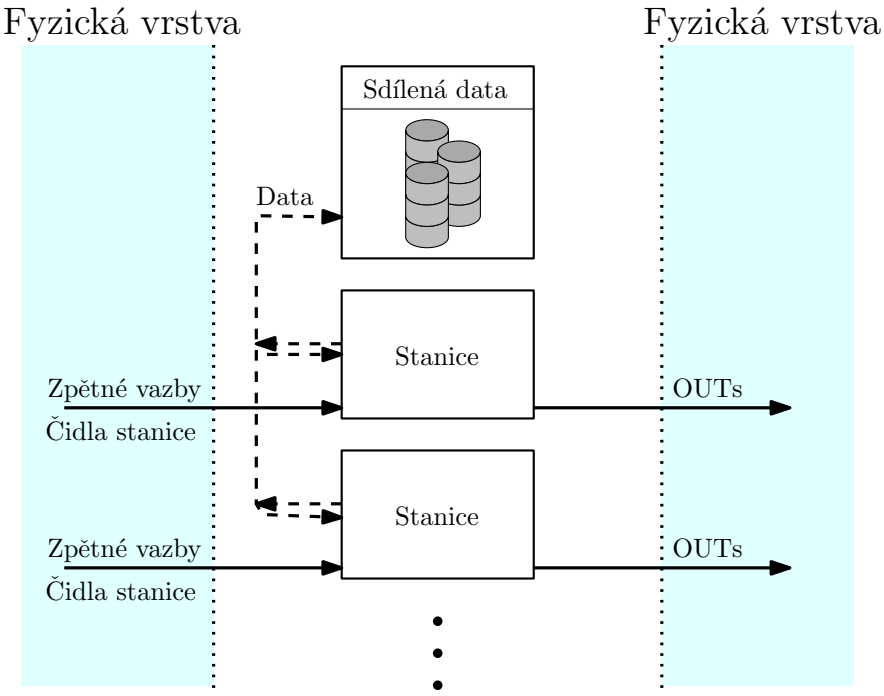
Stejně tak není nutné řešit synchronizaci komunikace mezi subkomponentami PLC programu (i přes různě dlouhé oběhy cyklů). Je to dané tím, že výstupy z safety části programu jsou v safety kritický situacích závislé pouze na safety vstupech jejichž platnost je v této subkomponentě zaručena.

4.2.3 Komponenta PLC programu

Architektura komponenty vychází z funkčního rozdělení linky na stanice. Stanice jako celek se tak stává největším modulem komponenty PLC programu.

Jak je patrné z blokového schématu, na fyzické úrovni musí interface modulu zajistit spínání výstupních cívek aktuátorů a kontrolu jejich zpětných vazeb. Modul stanice dále zpracovává ostatní signály stanice (čidla přítomnosti dílů, odbavovacího tlačítka ...). S ostatními moduly je zajištěna komunikace formou dat v Sdíleném datovém prostoru. Do tohoto prostoru přispívají všechny stanice informacemi o stavu svých sekvencí.

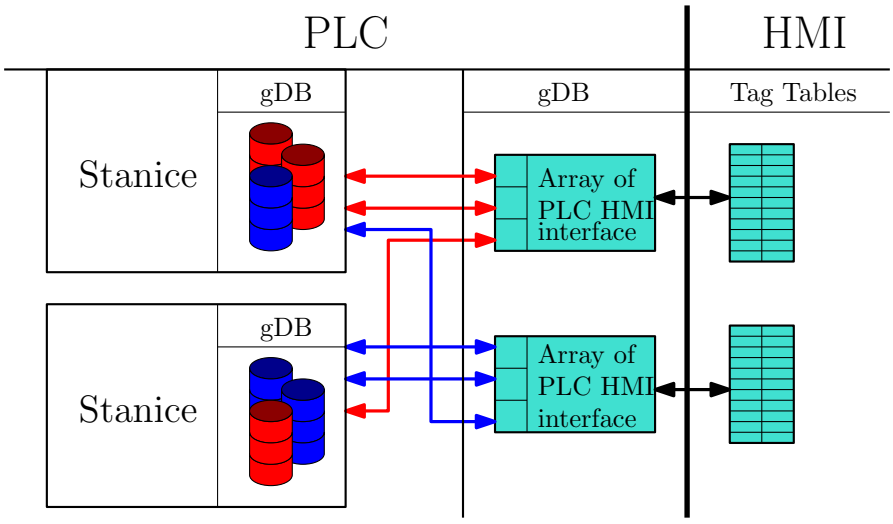
Moduly v rámci komponenty komunikují pomocí iDB jednotlivých modulů. I když jsou tyto bloky paměti v rámci softwaru (tzn i napříč všemi komponentami) viditelné a přístupné celé, jsou jako interface striktně využívány pouze vstupní, výstupní a vstupně-



Obr. 4.3: Komponenta PLC programu

výstupní parametry iDB. Statické proměnné jsou rezervované jako privátní paměťový prostor modulu.

Jak již bylo zmíněno, jako interface mezi komponentou PLC a HMI slouží strukturované gDB. GDB mají formu pole sdílených informací (není nutné sdílet veškerý datový obsah modulu) konkrétní instance, jejíž data jsou sdílena mezi komponentami. Pro každý modul komunikující s HMI komponentou tak existuje jedno gDB obsahující pole o celkovém množství daných prvků použitých na lince.



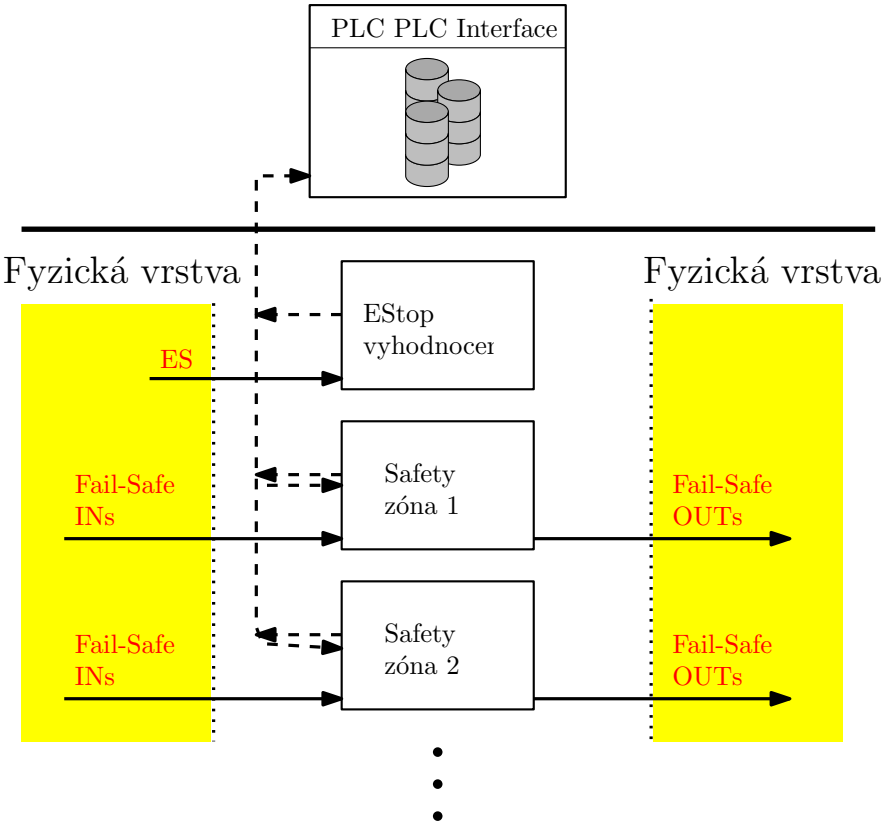
Obr. 4.4: Interface mezi PLC a HMI

Důvod pro zvolení této formy je možnost indexace jednotlivých prvků pole, čehož využívá implementace vizualizace.

Komponenta obsahuje ještě jednu sadu gDB (každá stanice po jedné) jejíž účel je popsat stavový automat formou dat.

4.2.4 Komponenta PLC safety programu

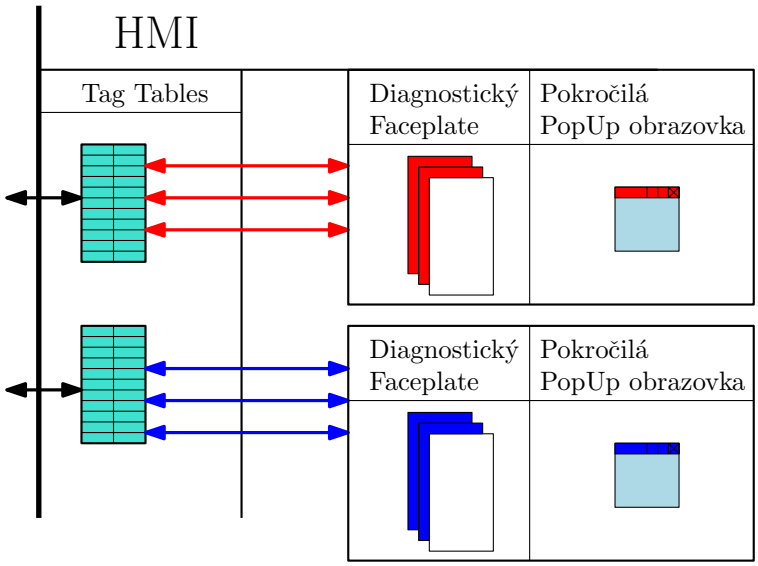
Komponenta zajišťuje spínání bezpečných potenciálů pracovišť na základě signálů od Fail-Safesenzorů a signálů z nouzových okruhů. Protože implementace je v tomto případě poměrně přímočará a k ovládání výstupů slouží standardizované certifikované bloky poskytované protředím TIA Portal, tak není přistoupeno k členění komponenty na moduly. Místo toho je komponenta dělena na sekce podle spínaného potenciálu. Pro komunikaci v rámci komponenty a mezi komponentami platí stejné principy popsané v podkapitole výše.



Obr. 4.5: Komponenta safety PLC programu

Komponenta HMI programu

Jak vyplývá z [předchozího odstavce o komponentě PLC programu], architektonicky jsou si komponenty HMI a PLC velmi podobné (viz obrázek). Největší rozdíl je několikanásobná forma implementace modulů v komponentě HMI. Několikanásobná implementace vyplývá z potřeby více forem zobrazení jednoho modulu. Důvodem k tomu je premisa, že vizualizace má ve výchozím stavu poskytovat pouze přehled o stavech prvků na lince a pokročilé funkce uživateli nabídnout až při jejich vyžádání.



Obr. 4.6: Komponenta HMI programu

Komponenta tedy definuje dvě formy zobrazení prvku: diagnostický a pokročilý. Úlohou diagnostické formy je poskytnout pouze základní diagnostiku o stavu prvku. Tato forma je úsporná a umožňuje diagnostikovat jen základní stavy. Funkce pro servisní ovládání a pokročilou diagnostiku jsou pak součástí pokročilého režimu. Hlavní snímek tedy zobrazuje prvky pouze v diagnostickém režimu. Pro aktivaci pokročilého režimu je nutné na prvek kliknout, tím se zobrazí vyskakovací okno s dalšími možnostmi. Uplatnění tohoto principu přispívá k přehlednější a čistší vizualizaci.

5

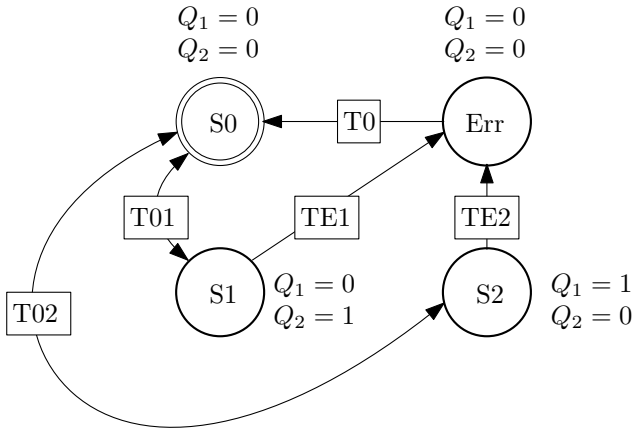
Design komponenty PLC

Kapitola pojednává o designu jednotlivých modulů a programových bloků v rámci komponenty PLC. Detailně popisuje koncept Stanice (obecně popsaný v odstaci Komponenta PLC programu) a jejích jednotlivých programových bloků.

5.1 Actuator

Blok pro řízení aktuátoru představuje nejmenší ucelený progamový celek. Jak je vidět z kapitoly 3.2, škála použitých aktuátorů na pracovišti je omezená (a jejich řízení formálně stejné), což umožňuje vytvořit jeden univerzální blok.

Stavový automat popisující takový aktuátor lze znázornit následujícím způsobem:



Přechod	Význam
T0	Kvitace chyby
T01	Pracovní stav 1
T02	Pracovní stav 2
TE1	Chyba
TE2	Chyba

Obr. 5.1: Stavový automat aktuátoru

Ve výchozím stavu S0 jsou oba výstupy aktuátoru resetované. Situace trvá do doby, než dojde k přechodu T01 nebo T02. Tyto přechody představují povel pro určitý pracovní stav. Ze schématu je zároveň vidět, že přechod je možné učinit i opačným směrem (z pracovního do výchozího stavu). To umožňuje vypnutí aktuátoru.

V případě chyby jsou vynuceny chybové přechody TE1 nebo TE2. Jako chybé situace jsou kvalifikovány následující události:

1. Překročení kritické doby nesouladu zpětné vazby
2. Ztráta zpětné vazby
3. Nevyžádaná zpětná vazba
   Je-li aktivována zpětná vazby pracovního stavu 2 při požadavku na pracovní stav 1 nebo obráceně. Tato chyba je kvalifikována jako kritická a dochází k okamžitému vynucení přechodu TE1 nebo TE2.

Přechod T0 je umožněn pouze při odstranění poruchy a kvitaci chyby prvku. Do té doby zůstává aktuátor pasivován.

Parametrizace

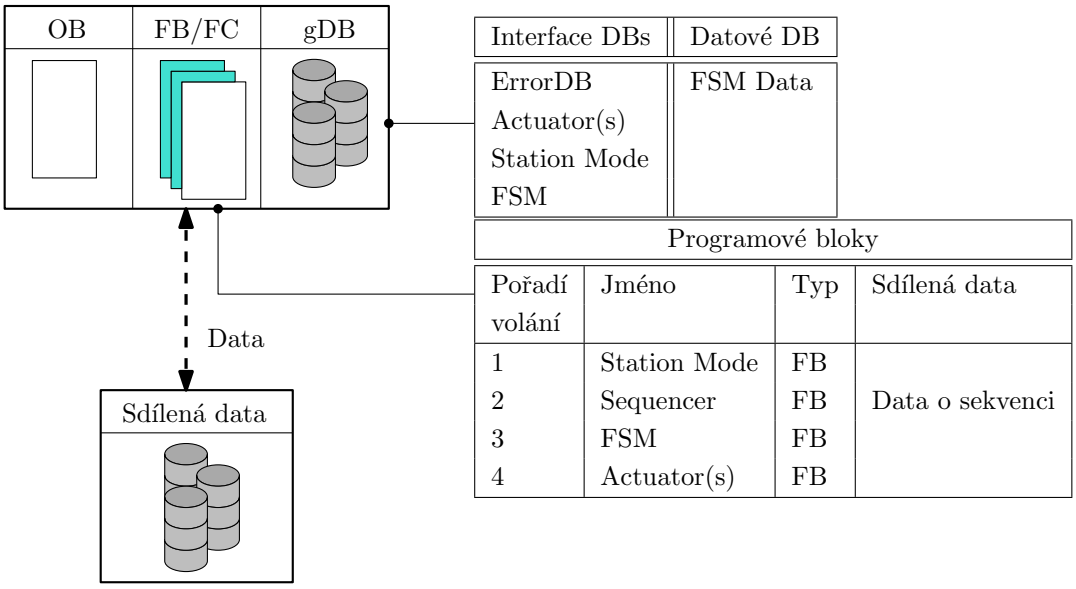
Uvedené chování aktuátorů je sice univerzálně použitelné, nicméně je možné setkat se s požadavkem, aby se různé aktuátory (nebo skupiny aktuátorů) chovali odlišně, například nevyžadovali kvitování. Proto je přistoupeno k parametrizaci aktuátoru. Tou lze nastavit:

1. Nutnost kvitování aktuátoru
   Během chyby se aktuátoru resetují výstupy, ale jeho opětovné zprovoznění nevyžaduje kvitaci.
2. Resetování výstupů při chybě
   Lze nastavit, že při chybě 1 a 2 nedojde k resetování výstupů. Na chybu 3 parametrizace nemá vliv, protože tato chyba je kvalifikovaná jako kritická a proto ji nelze přemostit.
3. Kritická doba nesouladu zpětné vazby s pracovním stavem 1
4. Kritická doba nesouladu zpětné vazby s pracovním stavem 2

5.2 Station

Stanice jsou koncipované jako nezávislé celky, jejichž primární úlohou je řízení aktuátorů podle sekvence výrobního procesu. Všechny stanice jsou formálně shodné, liší se pouze v počtu instancí aktuátorů a datech určujících výrobní sekvenci.

Základní blokové schéma modulu ilustruje obrázek 5.2.



Obr. 5.2: Design modulu stanice

Ze schématu je vidět, že se stanice sestává z:

- OB
  V daném pořadí postupně volá všechny programové bloky stanice. Priorita volání samotného OB v rámci hlavní programové smyčky je daná ID číslem stanice (OB).
- FB/FC programových bloků
  Implementují všechny potřebné funkcionality stanice. Jednotlivým blokům se věnují následující podkapitoly.
- gDB
  Jedna část tvoří interface mezi komponentami (viz odstavec 4.2.3 obrázek 4.4), druhá obsahuje data popisující sekvenci (viz)

Protože z hlediska výrobního procesu je nutné mezi stanicemi vyměňovat informace, poskytuje každá stanice Data o stavu své sekvence do sdíleného datového prostoru (viz obrázek 5.2 nebo 4.3). Odtud jsou přístupná ostatním stanicím, které na jejich základě mohou řídit svoje sekvence.

Uvedeným způsobem dochází k sdílení pouze dat bloku Sequencer. Tím je minimalizováno ovlivňování mezi stanicemi.

5.3 Station Mode

Station Mode poskytuje ostatním blokům PLC (a obsluze formou diagnostiky) aktivní provozní režim. Protože provozní režim je klíčový parametr pro řízení všech procesů na lince, jedná se o první volaný blok. Na straně HMI blok zpracovává požadavky na změnu režimu a poskytuje přehled o stavu safety prvků na stanici.

Design rozlišuje mezi třemi módy ve dvou kategoriích. Kategorii je určena globálně polohou ***servisního klíčku***, dodatečná volba je pro modul individuální.

Kategorie	Výchozí	Pokročilý
Automat	Plně automatický	StepMode
Ruční režim	Údržba	

Tab. 5.1: Provozní módy stanice

Charakteristika módů je následující:

1. Plně automatický
   Výchozí mód v kategorii Auto. Stanice se chová plně autonomně a následuje předepsanou výrobní sekvenci. HMI poskytuje pouze diagnostiku a neumožňuje ovládat žádný aktuátor nebo FSM.

2. Semi-automatický (krokový)
   Narozdíl od Plně automatického režimu, je přechod do dalšího kroku sekvence nutné potvrdit stisknutí tlačítka na HMI nebo fyzického odbavovacího tlačítka pracoviště.

3. Režim údržby
   HMI umožňuje udělovat povely jak FSM, tak jednotlivým aktuátorům. Aktuátory nejsou v tomto módu blokované, takže je možné dostat se do kolizního stavu. To neplatí pro blok FSM, který následuje zapsanou sekvenci.

Protože některé podmínky startu režimů závisí na stavu safety prvků stanice, jsou tyto informace součástí interface bloku. Interface viz tab 5.2

Station Mode		
Směr	Jméno	Datový Typ
IN	bKeyAuto	Bool
IN	bKeyMT	Bool
IN	Safety	Safety_UDT
IN/OUT	HMI	StationHMI_UDT
OUT	CurrMode	Mode_UDT

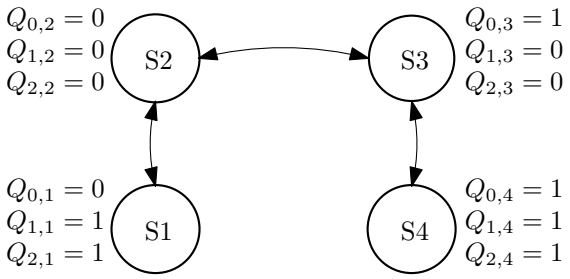
Tab. 5.2: Interface FSM

Stavový automat implementace bloku je na obrázku.

5.4 FSM

Blok implementuje obecný konečný stavový automat. Design je navržený tak, že umožňuje realizovat libovolný automat o jakémkoliv množství výstupů a stavů (určeno globálními konstantami).

Způsob implementace ilustruje stavový automat lemovacího přípravku na obr 5.3. Jak je z něj patrné, automat je plně určen dvěma tabulkami (výstupní a přechodovou).



Přechodová tabulka				
Výchozí stav	Požadovaný stav			
	S1	S2	S3	S4
S1		S2	S2	S2
S2	S1		S3	S3
S3	S2	S2		S4
S4	S3	S3	S3	

Výstupní tabulka				
Index aktuátoru	Stav			
	S1	S2	S3	S4
0	0	0	1	1
1	1	0	0	1
2	1	0	0	1

Obr. 5.3: Ukázka stavového automatu lemovacího přípravku

Přechodová tabulka popisuje všechny přechody mezi stavy a úlohou modulu je následování předpsané cesty od výchozího stavu *Start State* do požadovaného *Next State*. Například při požadavku na přechod ze stavu 1 do stavu 4, projde automat postupně stavy 2 a 3. Výstupy aktuátorů v jednotlivých stavech určuje výstupní tabulka.

Ze znázorněného algoritmu je patrné, že v případě chyby jsou výstupy aktuátorů resetovány.

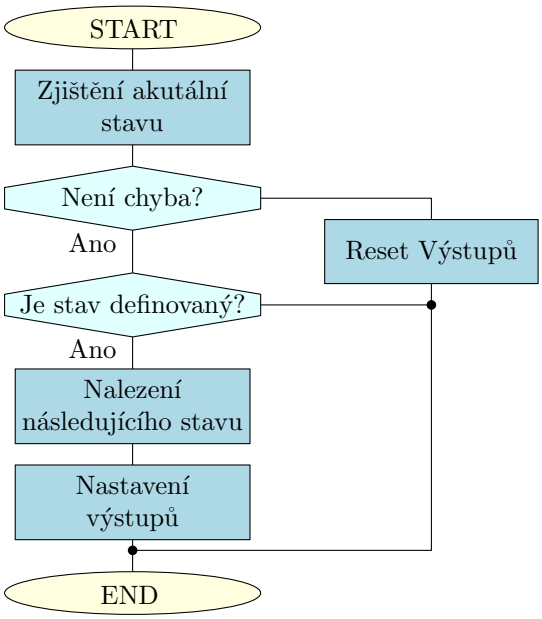
Pro splnění požadované funkcionality je nutné modulu poskytnout informace o aktuálních stavech aktuátorů a požadovaném stavu. Modul následně poskytuje informaci o aktuálním stavu a povely pro jednotlivé aktuátory. Interface je zobrazený v tab 5.3

V porovnání s naprogramovaným stavovým automatu přímo v kódu PLC programu je jasné, že uvedená implementace je výpočetně náročnější. Důvodem je zjevně fakt, že pro identifikaci aktuálního stavu musí blok zahrnovat algoritmus prohledávání dvourozměrného pole o rozměrech [počet stavů ; počet aktuátorů]

Oproti standardnímu přístupu, je tedy složitost větší minimálně přímo úměrně počtu stavů automatu.

Další zjevnou nevýhodou je, že aby mohl být FSM popsaný touto formou provozován, musí se nacházet v definovaném stavu. To nemusí být splněno například po startu linky. V takovém případě je tedy nutná kooperace obsluhy.

Výhodou je na druhou stranu univerzálnost bloku a snadná změna stavového automatu pouze změnou dat.



Obr. 5.4: Implementace stavového automatu

FSM Module		
Směr	Jméno	Datový typ
IN	CurrMode	Mode_UDT
IN	SpState	USInt
IN	StateTable	Array [1..#, 1..#]of USInt
IN	TransitionTable	Array [1..#, 1..#]of USInt
IN	CurrStates	Array [1..#, 1..#]of USInt
IN/OUT	HMI	FSMHMI_UDT
OUT	CurrStat	USInt
OUT	SpStates	Array [1..#, 1..#]of USInt

Tab. 5.3: Interface FSM

V následující části je ukázka implementovaného kódu FB FSM. Možnosti bloku mohou být rozšířeny o signál *Uvolněno* (lze vyřadit nastavení na TRUE) a IMSTP (okamžitý stop).

V horní části kódu jsou definované použité datové typy (Mode_UDT a FSM) a pomocná funkce ValidityStateCheck.

```
TYPE "Mode_UDT"
VERSION : 0.1
STRUCT
  Auto : Bool;    // Rezim automat
  Rucne : Bool;    // Rezim rucne
END_STRUCT;

END_TYPE

TYPE "FSM"
VERSION : 0.1
STRUCT
  SpState : USInt;
END_STRUCT;

END_TYPE

FUNCTION "FC_ValidityStateCheck" : Bool
{ S7_Optimized_Access := 'TRUE' }
VERSION : 0.1
   VAR_INPUT
      SpState : USInt;    // Pozadovany stav
      Table : Array[*, *] of USInt;    // Tabulka stavu
   END_VAR



BEGIN
        #FC_ValidityStateCheck :=
        (#SpState >= LOWER_BOUND(ARR := #Table, DIM := 1)) &
        (#SpState <= UPPER_BOUND(ARR := #Table, DIM := 1));
END_FUNCTION

FUNCTION_BLOCK "FB_FSM"
{ S7_Optimized_Access := 'TRUE' }
VERSION : 0.1
   VAR_INPUT
      Mode : "Mode_UDT";
      IMSTP : Bool;
      SpState : USInt;
```

```
        Uvolneno : Bool;
        ACK_IMSTP : Bool;
    END_VAR

    VAR_OUTPUT
        CurrStateO : USInt;    // Aktualni stav pripravku
    END_VAR

    VAR_IN_OUT
        StateTable : Array[*, *] of USInt;
        TransitionTable : Array[*, *] of USInt;
        SpStates : Array[*] of USInt;
        CurrState : Array[*] of USInt;
        HMI : "FSM";
    END_VAR

    VAR
        _IMSTP : Bool;
    END_VAR

    VAR_TEMP
        i : Int;
        j : Int;
        CurrState : USInt;
        SpStateTmp : USInt;
        SpStateMux : USInt;
    END_VAR


BEGIN
IF #IMSTP THEN
  #_IMSTP := TRUE;
ELSE
  IF #ACK_IMSTP THEN
    #_IMSTP := FALSE;
  END_IF;
END_IF;

IF #_IMSTP OR (#Mode.Auto & NOT #Uvolneno) THEN
  GOTO IMSTP;
```

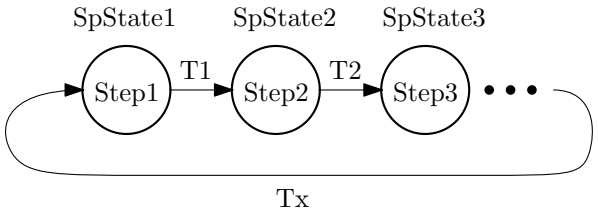
```
END_IF;

REGION CURR_STATE_CHECK

FOR #i :=
  LOWER_BOUND(ARR := #StateTable, DIM := 1) TO
  UPPER_BOUND(ARR := #StateTable, DIM := 1)
DO
  FOR #j :=
    LOWER_BOUND(ARR := #StateTable, DIM := 2) TO
    UPPER_BOUND(ARR := #StateTable, DIM := 2)
  DO
    #CurrState := "__AktuatorIdle";

    IF #CurrStates[#j] <> #StateTable[#i, #j] THEN
      IF ((#CurrStates[#j] AND "__AktuatorErr") <> 0) THEN
        #CurrState := "__AktuatorErr";
      END_IF;
    EXIT;
    ELSE
      #CurrState := INT_TO_USINT(#i);
    END_IF;
  END_FOR;

  IF #CurrState <> 0 THEN
    EXIT;
  END_IF;

END_FOR;

#CurrState := #CurrState;

END_REGION

REGION ERROR_STATE_CHECK

IF #Mode.Auto THEN
  #SpStateMux := #SpState;
ELSIF #Mode.Rucne THEN
  #SpStateMux := #HMI.SpState;
```

```
ELSE
  RETURN;
END_IF;

IF NOT "FC_ValidityStateCheck"
       (SpState := #SpStateMux, Table := #StateTable)
        OR
   NOT "FC_ValidityStateCheck"
        (SpState := #CurrState, Table := #StateTable)
  THEN
  RETURN;
END_IF; //Kontrola platnosti povelu

END_REGION Kontrola chyb a platnosti povelu

#SpStateTmp := #TransitionTable[#CurrState, #SpStateMux];

FOR #i :=
  LOWER_BOUND(ARR := #SpStates, DIM := 1) TO
  UPPER_BOUND(ARR := #SpStates, DIM := 1)
DO
  #SpStates[#i] := #StateTable[#SpStateTmp, #i];
END_FOR;

RETURN;

IMSTP:

FOR #i :=
  LOWER_BOUND(ARR := #SpStates, DIM := 1) TO
  UPPER_BOUND(ARR := #SpStates, DIM := 1)
DO
  #SpStates[#i] := "__AktuatorIdle";
END_FOR;
END_FUNCTION_BLOCK
```

5.5 Sequencer

Blok sequencer řídí automatickou sekvenci stanice. V jednotlivých krocích určuje požadovaný stav FSM a po splnění přechodové podmínky přechází do dalšího kroku. Ačkoliv se konkrétní implementace bloku sequencer pro každou stanici liší (rozdílný počet kroků, jiné přechodové podmínky), všechny obsahují určité společné prvky, které popisuje následující text.

Automatické sekvence jednotlivých pracovních stanic jsou popsány v kapitole 3. Jedná se o jednoduché stavové automaty s jedinou možnou cestou: v případě splnění přechodové podmínky postupují do následného kroku, popřípadě se vrací do některého kroku předchozího). Každý blok sequencer stanice pouze realizuje jeden tento automat.



Obr. 5.5: Modelový stavový automat automatické sekvence

Konkrétní softwarová realizace je uvedena na zobecněném příkladu z obrázku 5.5. Pro přechod do dalšího stavu sekvence je nutné splnit podmínky:

1. Úkolovaný FSM je v požadovaném stavu
   Každý krok automatické sekvence **musí** definovat požadovaný stav FSM. V kroku StepX se tedy kontroluje, zda skutečný stav FSM stanice odpovídá SpStateX. Splnění této podmínky zároveň znamená, že **krok sekvence je validní**, její nesplnění, značí chybu a krok automatické sekvence stanice je označen jako nevalidní. Podmínka má smysl zejména při spolupráci několika stanic.

2. Čidla stanice (ne zpětné vazby aktuátorů) splňují zadané podmínky
   Každý krok **musí** zajistit kontrolu všech čidel přítomnosti dílů stanice. V případě, že se nacházejí v předepsaném stavu je podmínka splněna. Kontrola zpětných vazeb aktuátorů není nutná, protože je zahrnuta v implementaci bloku FSM. V případě chyby aktuátoru je vyvolána i chyba příslušného FSM a to způsobí, nesplnění podmínky 1.

3. Automatická sekvence jiné stanice je v předepsaném kroku a tento krok je validní
   Podmínka **může** být zahrnuta v případě spolupráce několika stanic. Je jí nepřímo kontrolovaný požadovaný stav spolupracující pracovní stanice.

Podmínka 1 a 2 zajišťuje, že pro přechod do dalšího kroku se stanice musí nacházet v definovaném stavu z pohledu stavu veškerých aktuátorů a veškeré senzoriky. V opačném případě se jedná o chybu, která neumožní pokračování ve výrobní sekvenci. Zahrnutí

podmínky 3 je nepovinné. Je zahrnuta nejčastěji v případě, kdy je sekvence řízena robotickými signály *Hotovo*.

5.6 Robot

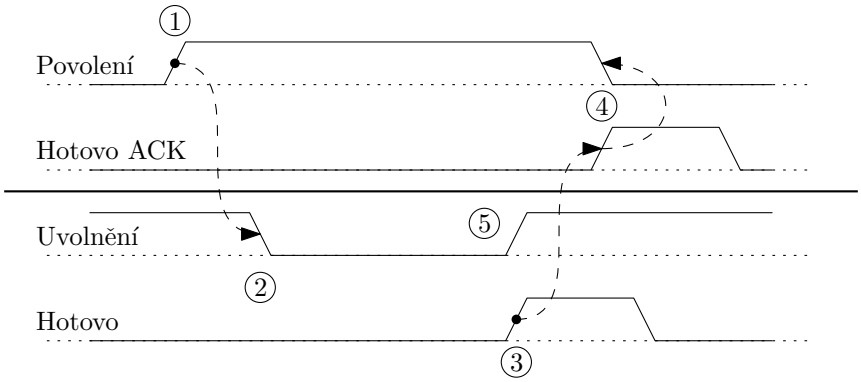
Blok zprostředkovává komunikaci s robotem, zajišťuje startovací sekvenci a v průběhu výroby řídí jeho činnost. Řízení činnosti probíhá na principu Master Slave, kdy robot je vůči PLC slave. Následující kapitola popisuje princip řízení robota z pohledu komunikace mezi PLC a robotem.

Jsou definovány následující čtyři řídící signály. Ty jsou použité pro řízení každé jedné operace robota.

Jméno	Význam	Směr komunikace
Povolení	Požadavek na odstartování práce	PLC →Robot
Uvolnení	Práce přijata blokování pracovní zóny	Robot →PLC
Hotovo	Práce dokončena	Robot →PLC
Hotovo ACK	Potvrzení od PLC	PLC →Robot

Tab. 5.4: Řídící signály komunikace PLC robot

Jejich časová souslednost v bezchybovém stavu je zobrazena na diagramu.



Obr. 5.6: Komunikace mezi PLC a robotem

Význam označených časových okamžiků je následující:

① PLC povoluje robotu činnost

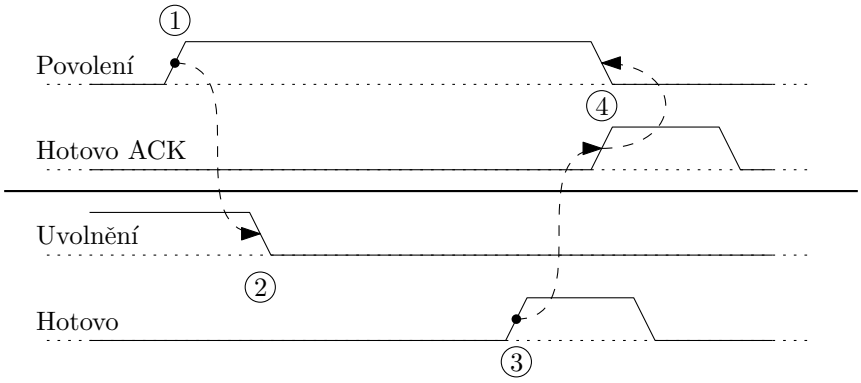
② Robot informuje o zahájení činnosti a blokuje pracovní zónu

③ Dokončení činnosti

④ HandShake signálu Hotovo, reset signálu Povolení

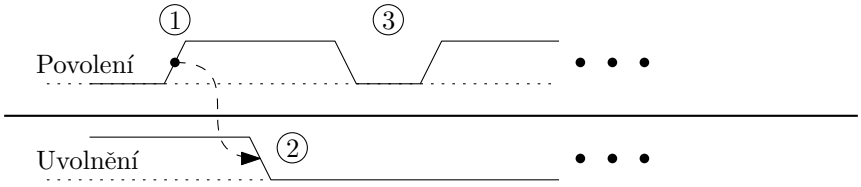
⑤ Uvolnění pracovní zóny

Diagram 5.6 zobrazuje situaci, kdy robot po dokončení činnosti uvolní pracovní zónu. V případě, že po dokončení činnosti pracovní zónu robot stále blokuje, zůstává příslušný signál uvolnění nadále resetovaný. Signál Uvolnění se vrací do původního stavu až ve chvíli skutečného uvolnění pracovní zóny.



Obr. 5.7: Blokování zóny robotem po dokončení činnosti

Signál Povolení je robotem monitorován cyklicky. V případě, že v průběhu provádění operace dojde k jeho resetu (okamžik ③ v obr. 5.8), robot zastavuje činnost. Pokud se následně úroveň signálu obnoví a robot zůstal na pracovní dráze, může pokračovat v činnosti.



Obr. 5.8: Ztráta povolení

6

Design komponenty HMI

Kapitola popisuje vizualizaci pracoviště E-Robot. V úvodní části je představena hlavní obrazovka a stručně popsány ovládací prvky na ní obsažené. Následně jsou pro ilustraci uvedeny další dvě obrazky pracovišť. Další část je dělena na sekce podle jednotlivých programových modulů popsaných v kapitole 5. V těchto sekcích je prezentován způsob vizualizace těchto modulů.

Všechny prezentované objekty modulů jsou dynamické. Jejich symbolická pojmenování, pojmenování čidel, diagnostické hlášky a podobně jsou uloženy v interface gDB na straně PLC. Objekty jsou díky tomu schopny tvořit instance, zobrazující stav konkrétního modulu pracoviště. Forma interface gDB je realizována formou pole (viz 4), konkrétní modul tedy předsatuje jeden prvek pole.

6.1 Obrazovky HMI

HMI panel obsahuje sadu obrazovek a sadu vyskakovací obrazovek. Účelem obrazovek je zobrazovat stavy čidel přítomnosti dílů jednotlivých stanic a sdružovat ikony aktuátorů na stanici použitých. Vyskakovací obrazovky na druhou stranu slouží k dynamickému linkování informací modulů aktuátor, FSM a Stanice.

V dalším textu jsou pro ilustraci uvedeny pouze tři obrazovky. Zbytek obrazovek je vytvořen analogicky.

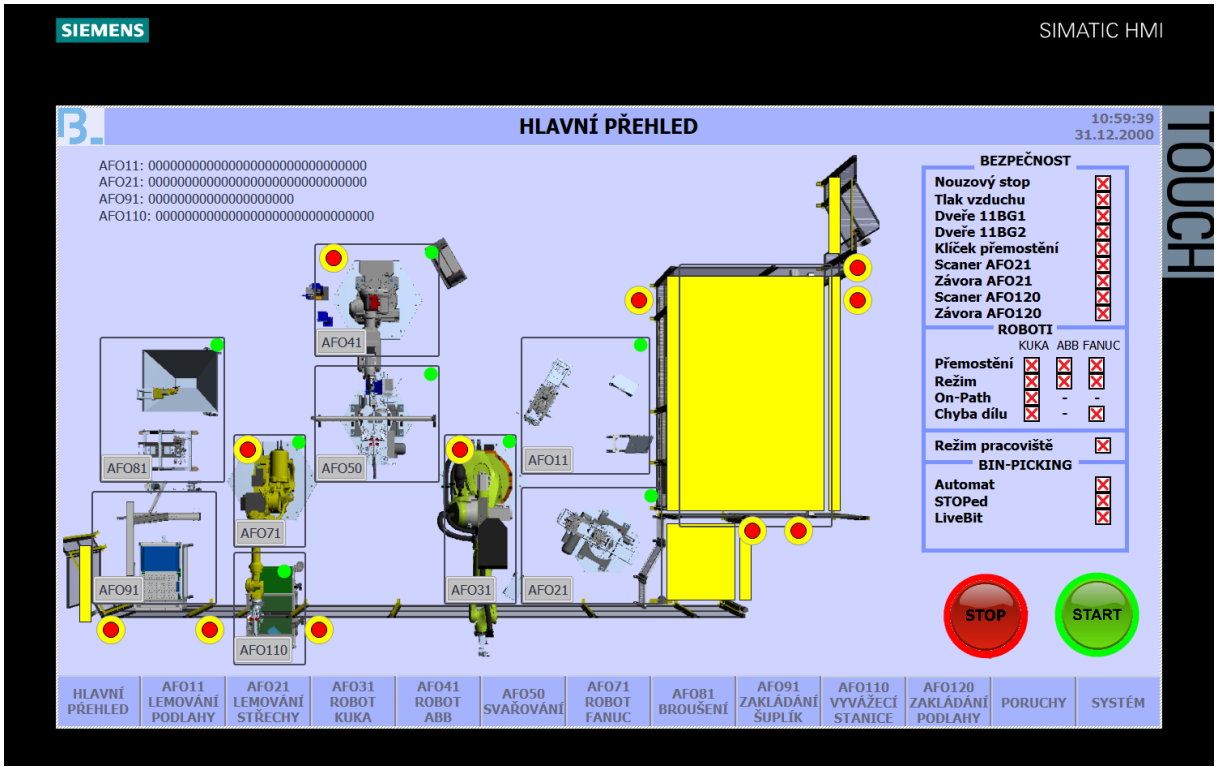
6.1.1 Hlavní přehled

Obrazovka slouží jako rozcestník na další obrazovky a zobrazuje základní přehled pracoviště, zejména stavy ES a ostatních bezpečnostních prvků. Je inicializována po startu HMI.

Na hlavní obrazovce je operátorovi umožněno jednak odstartování automatické sekvence všech stanic, dále aktivování žádosti o její zastavení. Další možností hlavního přehledu je aktivování vyskakovacího okna vybrané stanice.

Stanice jsou graficky odděleny obdélníky, které slouží zároveň jako tlačítka. Jejich stiskem je vyvoláno vyskakovací okno Stanice (viz dále vyskakovací okna). To obsahuje diagnostiku a speciální ovládací prvky.

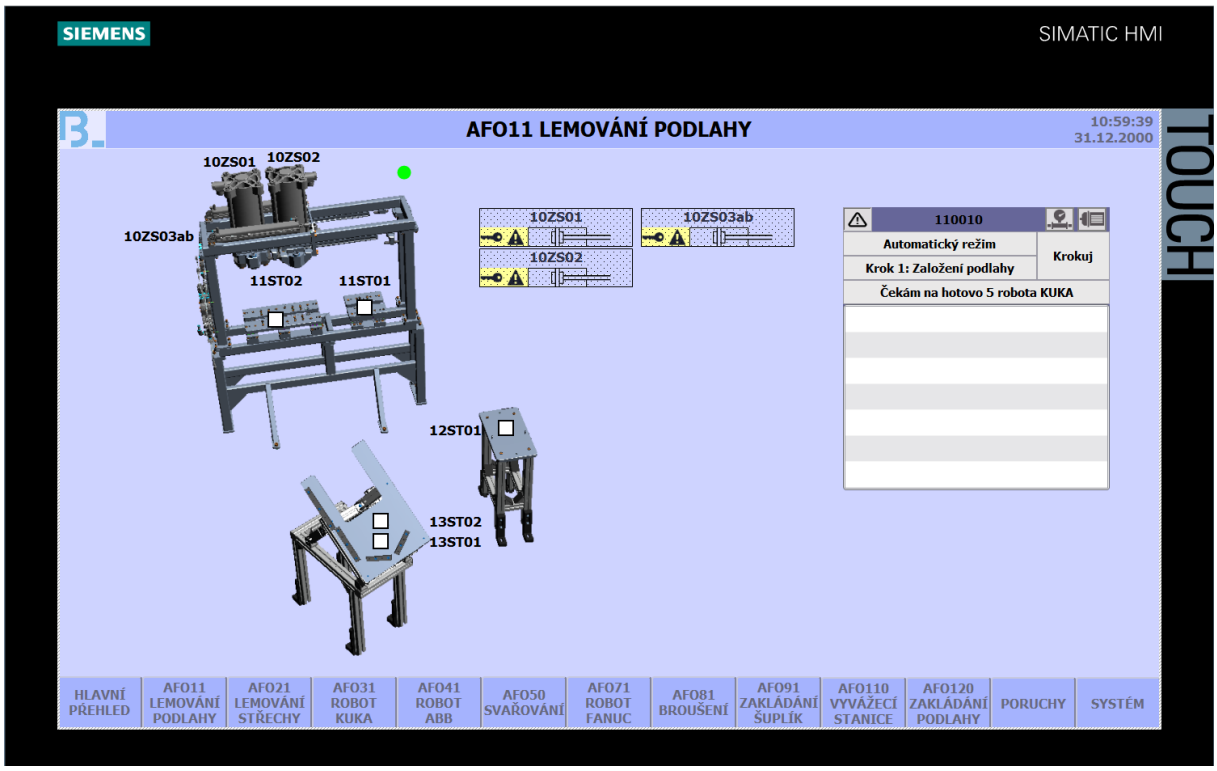
Spodní menu umožňuje přepínat mezi obrazovkami.



Obr. 6.1: Hlavní obrazovka

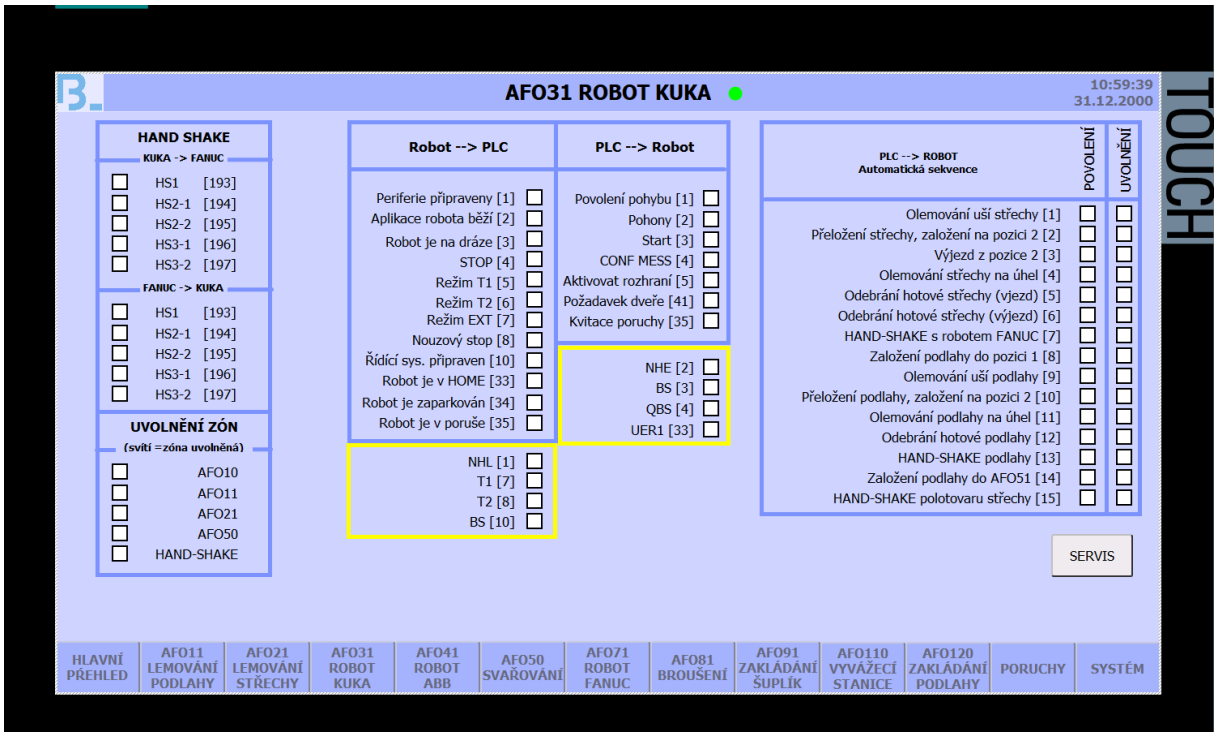
6.1.2 Obrazovka stanice

Obrazovka konkrétní stanice je zobrazena na obrázku níže. Sdružuje informace o čidlech přítomnosti dílů, ikony použitých aktuátorů a podobrazovku Stanice. Podobrazovka stanice je v tomto případě umístěna přímo na obrazovce a není vyvolána formou vyskakovacího okna.



Obr. 6.2: Obrazovka stanice 110010

V jistém smyslu vymykajícími se obrazovkami je trojice obrazovek zobrazujících systémové signály pro komunikaci s roboty. Na obrázku je zobrazena obrazovka robota KUKA.



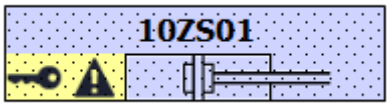
Obr. 6.3: Obrazovka robota KUKA

6.2 Ikony

Implementace bloku aktuátoru je v PLC programu realizována shodně nezávisle na tom, zda se jedná o motor nebo válec. Pro potřeby vizualizace je ovšem vhodné mezi nimi rozlišovat a to z hlediska přehlednosti. Liší se tedy forma zobrazení, přenášená data zůstavají stejná.

6.2.1 Válec

Každý válec pracoviště je na snímku stanice zobrazen následujícím způsobem:



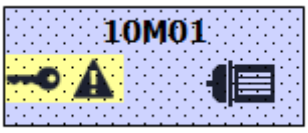
Obr. 6.4: Způsob zobrazení válce na hlavní obrazovce

Ze zobrazení je patrné symbolické pojmenování válce a jeho aktuální stav. Ikona válce se mění v závislosti na poloze pístu válce daného koncovými čidli. V případě chyby (signál z obou čidel najednou) nebo absence těchto signálů je poloha válce zobrazena tak, jak uvádí obrázek.

Dalšími prvky jsou výstražný trojúhelník a ikona klíčku. Podbarvení trojúhelníku symbolizuje chybu válce, podbarvení klíčku jeho přesouvání (jeden z výstupů ovládající ventil je aktivní).

6.2.2 Motor

Stejná situace jako pro válec platí pro motor. Jediný rozdíl je v grafické reprezentaci ikony umístěné na hlavním snímku:



Obr. 6.5: Způsob zobrazení motoru na hlavní obrazovce

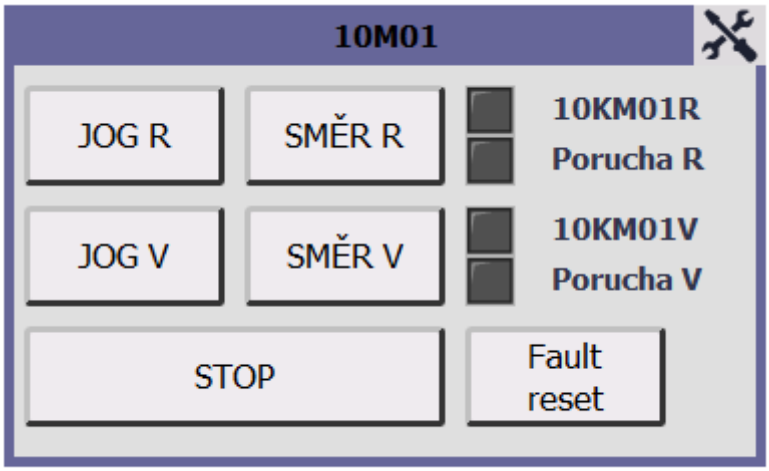
Každá ikona válce i motoru zobrazená na hlavní obrazovce funguje zároveň jako tlačítko. Jeho stiskem je vyvolána vyskakovací obrazovka Aktuátor.

6.3 Vyskakovací okna

Vyskakovací obrazovky jsou vyvolány stiskem některého z virtuálních tlačítek na HMI.

6.3.1 Aktuátor

Obrazovka je vyvolána stiskem ikony válce nebo motoru.



Obr. 6.6: Vyskakovací obrazovka pro ovládání aktuátoru

Umožňuje v ručním režimu kontrolovat aktuátor a obsahuje diagnostické informace. Symbol v pravém horním rohu informuje o aktuálně zvoleném režimu. Nabývá dvou hodnot a odráží stav klíčku režimu pracoviště.

Dalšími diagnostickými informacemi jsou signály senzorů zpětných vazeb (XXKMYYV a R) a informace o případné poruše. Pětice tlačítek umožňuje ruční ovládání prvku.

Tlačítko JOG spíná konkrétní výstup jen pokud je stisknuto a jeho uvolněním dojde k resetování příslušného výstupu. Tlačítko SMĚR naproti tomu sepne daný výstup trvale. STOP tlačítko vynutí okamžitý reset obou výstupů.

Tlačítko Fault reset se uplatňuje jen u aktuátorů, které jsou nakonfigurované tak, že při chybě vyžadují kvitaci. U ostatních se toto tlačítko nezobrazuje.

6.3.2 Stanice

Vyskakovací okno stanice je vyvolána stiskem ohraničujícího obdélníku stanice na hlavním snímku.

Zobrazení má zejména informovat obsluhu o aktuálním prováděném kroku automatické sekvence, chybách a případné očekávané akci.

Obr. 6.7: Vyskakovací obrazovka stanice

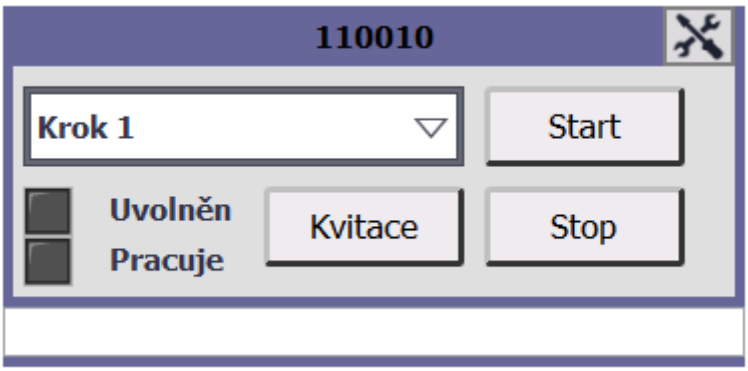
Horní část obrazovky obsahuje název stanice, výstražný trojúhelník a dvojici ikon. Trojúhelník stejně jako u aktuátoru informuje o přítomnosti chyby (chyba jakéhokoliv prvku stanice). Dvojice ikon odkazuje na stav Fail-Safevýstupů stanice. Ikony jsou podbarvené zeleně v případě sepnutí bezpečnostního přívodního ventilu stlačeného vzduchu (levá ikona) nebo pohonů motorů (pravá).

Další prvky okna informují o aktuálně navoleném režimu a umožňují ho změnit (viz 5.3. Tlačítko Krokuj slouží pro krokování sekvence v krokovém režimu. Dále obrazovka obsahuje popis právě prováděného kroku a diagnostickou hlášku přiřazenou konkrétní situaci. Tyto hlášky jsou popsané individuálně v PLC pro každou stanici / sekvenci.

Spodní část obrazovky obsahuje soupis aktivních alarmů stanice. Jejich filtrování je zajištěno na základě jména stanice.

6.3.3 FSM

Grafická reprezentace bloku FSM zajišťuje možnost ručního ovládání a diagnostiku. Obrazovky s ovládacími prvky jsou přístupné po aktivaci příslušné ikony FSM.



Obr. 6.8: Vyskakovací obrazovka FSM

Požadovaný stav se volí pomocí rozbalovacího kontextového menu a aktivuje tlačítkem Start. Obsah kontextového menu je shodný s popisem stavů v kapitole 3.

Tlačítkem Stop dojde k vypnutí výstupů všech aktuátorů FSM bloku. Kvitačním tlačítkem lze najednou kvitovat veškeré chyby vyskytující se na akutátorech FSM.

Spodní část obrazovky informuje obsluhu o aktuálním stavu FSM. Stejně jako kontextové menu jsou příslušná textová hlášení vytvořená individuálně na základě konkrétního FSM.

7

Virtuální zprovoznění

Kapitola se věnuje tématu virtuálního zprovoznění robotického pracoviště. V úvodní části obecně popisuje koncept virtutálního zprovoznění a virtuálního dvojčete, v další části se pak věnuje rešerži softwarových produktů, které virtuální zprovoznění umožňují. Závěr kapitoly je zaměřen na podrobnější popis použitého řešení.

Virtuální zprovoznění umožňuje simulaci chování pracoviště nebo jeho celku v počítačovém prostředí. Nad touto simulací lze provádět různé testy a zkoumat, zda se model pracoviště chová podle očekávání. Zjevnou výhodou tohoto přístupu je možnost včasného objevení chyb a jejich náprava ještě ve fázi návrhu. Nevýhodou je ovšem časová náročnost přípravy modelu a jeho netriviální obsluha.

Pro simulaci jakéhokoliv celku je nejprve nutné vytvoření jeho modelu - virtuálního dvojčete - v počítačovém prostředí. Tento model je nejčastěji tvořen na základě 3D dat z nějakého CAD softwaru, ale prakticky se může jednat i o jakoukoliv jinou abstrakci. Podle požadovaných cílů simulace se tedy může podoba digitálního dvojčete lišit. Například virtuální dvojče robotického pracoviště bude mít jinou podobu v simulaci zaměřující se na dynamické chování a v simulaci zaměřující se na tok materiálu.

V různých typech simulací je také možné pracoviště do určité míry zjednodušit a chování některých prvků zanedbat. V příkladu simulace materiálového toku bude možná účelnější simulovat chování výrobních stanic pracoviště jako celku (například z hlediska časové náročnosti operací), než simulovat dopodrobna dynamiku válce.

7.1 Softwarové alternativy

Následující seznam uvádí různé alternativy softwaru podporujícího virutální zprovoznění.

- KUKA.Sim
  Simulační nástroj výrobce robotů KUKA, který umožňuje pomocí virtuálního kontroléru programovat roboty stejné značky. Software zahrnuje možnost importování CAD dat a obsahuje knihovnu používaných prvků v automatizačním průmyslu (dopravník, válec ...). Software je možné pomocí OPC-UA rozhraní propojit s PLC

různých značek (Siemens, Beckhoff) Jedná se o podpůrný testovací nástroj, který není určený pro simulování větších celků. Pro aplikace malého rozsahu umožňuje ověřit základní předpoklady (kolize, takt). Další nevýhodou je nemožnost virtuálního zprovoznění robotů jiných značek.

- RobotStudio
  Jedná se o softwarový produkt společnosti ABB a nabízí podobné možnosti jako KUKA.Sim. Stejnou nevýhodou je nemožnost simulace jiných robotických kontroléru než ABB.

- Virtual Universe Pro
  Software umožňuje simulovat rozsáhlejší celky než výše jmenované nástroje. Stejně jako předchozí jmenované umožňuje importovat CAD data a obsahuje zabudovanou knihovnu základních prvků. Samozřejmostí je propojení s reálným nebo simulovaným PLC a HMI. Nevýhodou software je, že neobsahuje simulace robotického kontroléru a proto není možné detailní simulace chování robotického programu.

- Process Simulate
  Profesionální nástroj vyráběný společností Siemens. Integruje všechny vlastnosti výše jmenovaných softwarů a jako jediný z nich umožňuje simulovat robotické kontroléry široké řady výrobců. V prostředí process simulate je tedy možné provést detailní studii robotického pracoviště z pohledu robotického programu i programu PLC.

7.2 Process Simulate

Odstavec představuje použitý způsob simulace pracoviště E-Robot ve virtuálním prostředí Process Simulate.

Digitální dvojče pracoviště E-Robot zahrnuje 3D modely všech akčních prvků a senzorů pracoviště. Jednotlivé modely aktuátorů mají nasimulovanou dynamiku a umožňují tedy vykonávání stejných pohybů jako na reálném pracovišti.

Součástí simulace aktuátorů je i simulace senzorů zpětných vazeb. Koncová čidla válců reagují stejně jako v reálném prostředí, takže k jejich sepnutí dochází až při vyjetí pístu válce do určité úrovně. Stejným způsobem se chovají i vačkové snímače na hřídelích motorů. Díky tomu je možné věrně ověřit chování jednotlivých FSM a posloupnot vykonávaných akcí.

7.2.1 Simulační sestava

Simulační sestava se skládá z reálného PLC, simulační jednotky účastníků na ProfiNET sběrnici SIMULATION UNIT a stolního počítače se simulací v prostředí Process Simulate.

Na stolním počítači je ještě navíc spuštěna simulace HMI. Sestava tak obsahuje všechny prvky a zpracovávané vstupy a výstupy.

Z hlediska PLC programu je pro potřeby simulace nutný pouze export dat o hardwarové konfiguraci sítě. Tato data jsou nahrána do simulační jednotky SIMULATION UNIT, která emuluje chování těchto účastníků.

Reálné PLC se potom díky propojení s jednotkou nachází ve stejné síti a může přistupovat k fyzickým adresám ostatních účastníků stejně jako reálné PLC na pracovišti. Zapíše-li se tedy v prostředí Process simulate na adresu účastníka ProfiNET sítě nějaká informace je díky SIMULATION UNIT přístupná i pro PLC.

7.2.2 Roboti

Ačkoliv simulační prostředí umožňuje import robotických kontrolérů všech fyzicky přítomných robotů, z důvodů výpočetní náročnosti bylo nakonec přistoupne k simulaci všech robotů pomocí univerzálního kontroléru. Tím vznikl největší rozdíl mezi softwarem simulovaným ve virtuálním a reálném prostředí.

Dalším rozdílem byla nutnost dodržení pevně dané sekvence kroků v simulovaném prostředí. Potřeba pevně daného lineárního procesu vyvstala z čistě pragmatických důvodů jednoduší tvorby robotického programu ve virtuálním prostředí.

8

Závěr

Po seznámení s jednotlivými užívanými třídami Simatic se práce věnuje konstrukci pracoviště. Cílem kapitoly je vysvětlit technologii pracovního cyklu a popsat pracovní stavy každé stanice do té míry, aby byl způsob implementace programových bloků (zejména blok FSM a jeho tabulky) pochopitelný.

Kapitola o architektuře a designu komponenty PLC představuje nejobsáhlejší část práce. Pomocí stavových automatů a diagramů vysvětluje chování programových modulů a přibližuje způsob výměny dat mezi komponentami.

Největším praktickým přínosem z pohledu implementace se stává důsledné dělení funkce na moduly a snaha o dodržení objektově orientovaného přístupu. Ta se promítá i do designu HMI.

Vzhledem k experimentální povaze pracoviště bylo přistoupeno k popisu stavových automatů pomocí dat (stavové a přechodové tabulky). Popis chování skupiny aktuátorů touto formou umožňuje snadnou modifikaci chování stavového automatu a zaručuje bezkolizní provoz. Jeho nevýhodou v porovnání s klasickým přístupem popisu stavového automatu jsou výpočetní náročnost a větší nároky na datový prostor. Blok pro svůj provoz také vyžaduje definovaný počáteční stav. To může působit potíže při prvním spuštění pracoviště a je nutné blok uvést do nějakého definovaného stavu. Pro popsání jednoduché (a v čase neměnné) sekvence se tedy tento způsob implementace nejeví jako vhodný.

Praktickým testováním ve virtuálním prostředí byly ověřeny funkce jednotlivých programových modulů v komponentě PLC i HMI. Z tohoto pohledu byl přínos virtuálního oživení přínosný, protože se podařilo v poměrně krátkém čase (asi dva týdny) odladit jednotlivé programové moduly a potvrdit jejich funkci. Implementace se díky tomu na reálném pracovišti neměnila.

Na druhou stranu se nepodařilo plně otestovat automatickou sekvenci celého pracoviště a to z několika důvodů

1. Simulace robotických kontrolérů neodpovídala reálnému chování
   Startovací sekvence a systémové řídící signály konkrétních robotů musely být upravovány na reálném pracovišti.

2. Roboti v simulačním prostředí následovali pouze lineární proces
   Vzhledem k tomuto zjednodušujícímu předpokladu (který usnadnil konfiguraci na straně Process Simulate) bylo možné otestovat pouze pevně danou posloupnost akcí. Na reálném pracovišti mohou ale jednotlivé sekvence předbíhat jedna druhou a proces může být do jisté míry nelineární.

3. Nedostatečné projektové řízení
   Projekt nebyl dostatečně kvalitně řízen, což vedlo k několikanásobným změnám v konceptu SW jak pro roboty i PLC. Navíc některé prvky pracoviště nebyly nikdy zprovozněny (nejsou uvedeny v této práci)

Literatura

[1] Chen, Anna *History Of Siemens PLC.* Shenzhen Digital Life Electric Co.,Ltd 6.11.2018 [cit 26.5.2021] Dostupné z: https://www.cnsiemensplc.com/news/history-of-siemens-plc-18853727.html

[2] Siemens, s.r.o. *Simatic S7-400.* Siemens, s.r.o. [cit 26.5.2021] Dostupné z: https://new.siemens.com/cz/cs/products/automation/systems/industrial/plc/simatic-s7-400.html

[3] Siemens, s.r.o. *Data sheet 6ES7217-1AG40-0XB0.* [cit 26.5.2021] Dostupné z: https://mall.industry.siemens.com/mall/cs/cz/Catalog/Product/6ES7217-1AG40-0XB0

[4] Siemens, s.r.o. *Data sheet 6ES7318-3EL01-0AB0.* [cit 26.5.2021] Dostupné z: https://mall.industry.siemens.com/mall/cs/cz/Catalog/Product/6ES7318-3EL01-0AB0

[5] Siemens, s.r.o. *Data sheet 6ES7518-4AX00-1AC0.* [cit 26.5.2021] Dostupné z: https://mall.industry.siemens.com/mall/cs/cz/Catalog/Product/6ES7518-4AX00-1AC0

[6] Siemens, s.r.o. *Delivery Release SIMATIC STEP 7 Professional / Basic V16.* 12.2.2019 [cit 26.5.2021] Dostupné z: https://support.industry.siemens.com/cs/document/109771628/delivery-release-simatic-step-7-professional-basic-v16-?dti=0lc=en-AT

[7] Siemens, s.r.o. *What are the requirements for installing STEP 7 version V5.5?* 19.6.2013 [cit 26.5.2021] Dostupné z: https://support.industry.siemens.com/cs/document/24059046/what-are-the-requirements-for-installing-step-7-version-v5-5-?dti=0dl=enlc=nl-NL

[8] Siemens, s.r.o. *Data sheet 6ES7515-2FM02-0AB0.* [cit 26.5.2021] Dostupné z: https://mall.industry.siemens.com/mall/cs/cz/Catalog/Product/6ES7515-2FM02-0AB0

[9] Thomas, D., Hunt, A. *The Pragmatic Programmer.* 20th Anniversary edition, Pearson Education, Inc. ISBN-10: 0-13-595705-2