

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

ŘÍZENÍ MODELŮ PROCESŮ EDU-MOD PROGRAMOVATELNÝM AUTOMATEM ŘADY S7-1200

EDU-MOD MODEL CONTROL BY PROGRAMMABLE LOGIC CONTROLLER S7-1200

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JIŘÍ KROUPA

VEDOUcí PRÁCE
SUPERVISOR

ING. TOMÁŠ MARADA, PH.D.

BRNO 2013

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav automatizace a informatiky

Akademický rok: 2012/2013

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

student(ka): Jiří Kroupa

který/která studuje v bakalářském studijním programu

obor: **Aplikovaná informatika a řízení (3902R001)**

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:

Řízení modelů procesů EDU-mod programovatelným automatem řady S7-1200

v anglickém jazyce:

EDU-mod model control by programmable logic controller S7-1200

Stručná charakteristika problematiky úkolu:

V laboratoři programovatelných automatů A1/731a jsou čtyři modely soustav EDU-mod. Jedná se o soustavu posuvové jednotky, mísicí jednotky, automatické pračky a křižovatky.

Cílem této BC práce je navrhnout a realizovat řízení modelu posuvové a mísicí jednotky programovatelným automatem Siemens Simatic S7-1200. Dalším cílem je vytvořit vzorová zadání a vypracování. Funkčnost ověřit řízením pomocí PLC.

Cíle bakalářské práce:

1. Seznamte se s modely soustav EDU-mod v laboratoři A1/731a.
2. Seznamte se s programovatelným automatem Siemens S7-1200.
3. Vytvořte vzorová zadání s vypracováním řízení úloh.
4. Ověřte funkčnost řízením pomocí PLC.

Seznam odborné literatury:

- [1] Šmejkal, L., Martinásková, M., PLC a automatizace, Praha: BEN, 1999.
- [2] Firemní materiály o programovatelných automatech fy Siemens pro Simatic S7-1200.

Vedoucí bakalářské práce: Ing. Tomáš Marada, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2012/2013.

V Brně, dne 20.11.2012

L.S.

Ing. Jan Roupec, Ph.D.
Ředitel ústavu

prof. RNDr. Miroslav Doupovec, CSc., dr. h. c.
Děkan fakulty

ABSTRAKT

Tato bakalářská práce byla vytvořena jako podpora pro výuku předmětu Programovatelné automaty na Ústavu automatizace a informatiky fakulty Strojního inženýrství Vysokého učení technického v Brně. Cílem práce je seznámit se s programovatelným automatem Siemens Simatic S7-1200, HMI panelem Siemens Simatic KTP400 Basic color, vývojovým prostředím TIA Portal v11 a modely soustav EDU-mod. Poté pro tyto modely navrhnout a realizovat vzorové úlohy a ověřit funkčnost pomocí PLC.

ABSTRACT

This bachelor's thesis was created as a support for Programmable Logic Controllers course at the Institute of Automation and Computer Science of the Faculty of Mechanical Engineering at Brno University of Technology. The aim of this work is to get acquainted with programmable logic controller Siemens Simatic S7-1200, HMI panel Siemens Simatic KTP400 Basic color, development environment TIA Portal v11 and EDU-mod models. Subsequently, for these models to design and implement sample task and verify their functionality using a PLC.

KLÍČOVÁ SLOVA

Siemens Simatic S7-1200, Siemens Simatic HMI KTP400 Basic color, TIA Portal, EDU-mod.

KEYWORDS

Siemens Simatic S7-1200, Siemens Simatic HMI KTP400 Basic color, TIA Portal, EDU-mod.

PROHLÁŠENÍ O ORIGINALITĚ

Prohlašuji, že jsem tuto práci vypracoval samostatně, s využitím uvedené literatury a podkladů, na základě konzultací a pod vedením vedoucího bakalářské práce.

V Brně dne 24.5.2013

.....
Jiří Kroupa

BIBLIOGRAFICKÁ CITACE

KROUPA, J. *Řízení modelů procesů EDU-mod programovatelným automatem řady S7-1200*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2013. 64 s. Vedoucí bakalářské práce Ing. Tomáš Marada, Ph.D.

PODĚKOVÁNÍ

Děkuji tímto svému vedoucímu Ing. Tomáši Maradovi, Ph.D. za rady a veškerou pomoc při tvorbě této bakalářské práce.

Obsah:

Zadání závěrečné práce.....	3
Abstrakt.....	5
Prohlášení o originalitě.....	7
Poděkování.....	9
Úvod	13
1 Modely soustav EDU-mod.....	15
1.1 Hydraulická posuvná jednotka.....	15
1.2 Mísicí jednotka.....	16
2 Siemens Simatic S7-1200.....	17
2.1 Technické parametry.....	17
2.2 Rozšiřitelnost.....	18
2.2.1 Komunikační moduly.....	18
2.2.2 Signální moduly.....	18
2.2.3 Signální karty.....	18
2.2.4 Ostatní příslušenství.....	18
3 Siemens HMI KTP400 Basic color.....	19
3.1 Technické parametry.....	19
4 Deska plošných spojů.....	21
4.1 Návrh.....	21
4.2 Osvit.....	23
4.3 Vyvolávání.....	23
4.4 Leptání.....	23
4.5 Vyčištění a osazení desky.....	23
5 Siemens TIA Portal.....	25
5.1 Instalace.....	25
5.2 Tvorba projektu a nastavení zařízení.....	27
5.3 Tvorba programu.....	30
6 Řešené úlohy.....	35
6.1 Hydraulická posuvná jednotka.....	35
6.1.1 Zadání příkladu č. 1.....	36
6.1.2 Vypracování příkladu č. 1.....	36
6.1.3 Zadání příkladu č. 2.....	37
6.1.4 Vypracování příkladu č. 2.....	37
6.1.5 Zadání příkladu č. 3.....	40
6.1.6 Vypracování příkladu č. 3.....	40
6.1.7 Zadání příkladu č. 4.....	43
6.1.8 Vypracování příkladu č. 4.....	43
6.2 Mísicí jednotka.....	48
6.2.1 Zadání příkladu č. 1.....	49
6.2.2 Vypracování příkladu č. 1.....	49
6.2.3 Zadání příkladu č. 2.....	50
6.2.4 Vypracování příkladu č. 2.....	50
6.2.5 Zadání příkladu č. 3.....	52
6.2.6 Vypracování příkladu č. 3.....	52
6.2.7 Zadání příkladu č. 4.....	58
6.2.8 Vypracování příkladu č. 4.....	58
Závěr.....	61
Seznam použité literatury.....	63

ÚVOD

Cílem práce je vytvořit programy pro řízení dvou modelů technologických procesů EDU-mod. Konkrétně se jedná o model hydraulické posuvové jednotky a mísící jednotky. Pro každou jednotku vytvoříme čtyři programy, které budou napsány v jazyce strukturovaného textu (SCL). Čtvrtý program přitom bude rozšíření třetího programu o HMI panel Siemens Simatic KTP400 basic color, pomocí kterého budeme jednotku ovládat a který bude zobrazovat stav ve kterém se právě nachází. Programy budou vytvářeny pro programovatelný automat Siemens Simatic S7-1200 ve vývojovém prostředí TIA Portal v11.

1 MODEL Y SOUSTAV EDU-MOD

Jedná se o modely "technologických procesů", které slouží k výuce logických systémů realizovaných programovatelnými automaty, řídicími počítači, stavebnicemi logických obvodů. Vstupní a výstupní signály EDU-mod jsou vyvedeny na 20-ti pólový konektor. Modely můžeme rozlišit podle napěťové úrovně logických signálů na dvě kategorie.

- Moduly EDU-mod řady 24V
- Moduly EDU-mod řady 5V

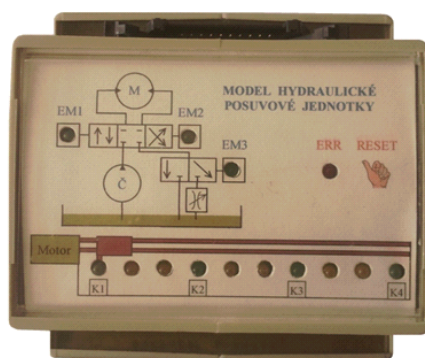
V našem případě se jedná o modely s napěťovou úrovní 24V [1]

1.1 Hydraulická posuvná jednotka

Jedná se o model simulující posuv hydraulické jednotky, který je znázorněn pomocí deseti LED diod. Čtyři z těchto diod představují snímače polohy, rozmístěné rovnoměrně po délce lišty. Dále model obsahuje tři LED diody znázorňující elektromagnety ovládající pohyb suportu vpravo nebo vlevo, případně jeho zpomalení.

Po zapnutí napájení nebo po restartu (tlačítko RESET) se suport automaticky nastaví do inicializačního stavu - pozice na snímači K1.

Na modelu se dále nachází červená LED ERR signalizující chybový stav. Ten může nastat při dvou případech. Prvním případem je přejezd krajního snímače (K1, K4). V tomto stavu se rozsvítí LED ERR a dále všechny LED snímačů polohy (K1 až K4). Do výchozího stavu se lze dostat po stisku tlačítka RESET. Druhým případem je současné sepnutí elektromagnetů EM1 a EM2. V tomto stavu LED ERR bliká, po odstranění chybového stavu systém pokračuje bez restartu v činnosti. [2]



Obr. 1: Model hydraulické posuvné jednotky

Vodič	Význam	PLC	Vodič	Význam	PLC
1	GND	M	11	---	O0.6
2	+24V	I+	12	---	O0.7
3	EM1	O0.0	13	---	I0.0
4	EM2	O0.1	14	---	I0.1
5	EM3	O0.2	15	---	I0.2
6	---	O0.3	16	---	I0.3
7	GND	1M	17	K1	I0.4
8	+24V	1L+	18	K2	I0.5
9	---	O0.4	19	K3	I0.6
10	---	O0.5	20	K4	I0.7

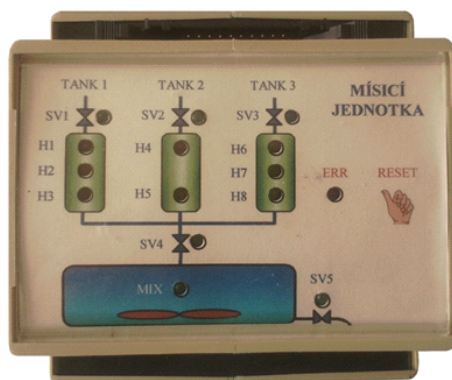
Tab. 1: Popis vývodů 20-ti pólového konektoru

1.2 Mísící jednotka

Jedná se o model, který simuluje proces napouštění, přepouštění a mísení. Jednotka se skládá ze tří tanků s objemem 84 litrů a mísící nádrže s objemem 253 litrů, propojené potrubím o průtoku 18 l/s. Tanky se plní pomocí tří ventilů (SV1 - SV3), jejichž stav je signalizován LED diodami. Nádrže se plní rychlostí 6 l/s. O výšce hladiny v tankách informují hladinoměry, přičemž snímače hladiny H1, H4, H6 signalizují plnou nádrž, snímače H2, H7 polovinu nádrže a snímače H3, H5, H8 minimální množství kapaliny. Dále se na jednotce nachází ventily SV4 pro přepuštění obsahu tanků do mísící nádrže, SV5 pro vypuštění nádrže a mixer.

Po zapnutí nebo po restartu (tlačítko RESET) se jednotka automaticky nastaví do inicializačního stavu, kdy jsou všechny nádoby prázdné. Zároveň se rozblíká červená LED ERR, která zhasne po prvním vybuzení některého z ventilů.

Chybový stav nastane v případě přetečení kterékoliv nádrže a je signalizován rozsvícením LED ERR. Systém se vrátí do výchozího stavu pomocí stisku tlačítka RESET. [3]



Obr. 2: Mísící jednotka

Vodič	Význam	PLC	Vodič	Význam	PLC
1	GND	M	11	MIX	O0.6
2	+24V	L+	12	---	O0.7
3	SV1	O0.0	13	H6	I0.0
4	SV2	O0.1	14	H7	I0.1
5	SV3	O0.2	15	H8	I0.2
6	---	O0.3	16	H5	I0.3
7	GND	M	17	H4	I0.4
8	+24V	1L	18	H3	I0.5
9	SV5	O0.4	19	H2	I0.6
10	SV4	O0.5	20	H1	I0.7

Tab. 2: Popis vývodů 20-ti pólového konektoru

2 SIEMENS SIMATIC S7-1200

Je malý programovatelný automat od firmy siemens vhodný pro menší aplikace. Řada S-1200 obsahuje v současné době 5 modelů, lišících se počtem vstupů a výstupů, velikostí paměti, rozšiřitelností a počtem čítačů. V našem případě se jedná o model s CPU 1214C. Technické parametry jsou uvedeny v kap. 2.1 [4]



Obr. 3: Siemens Simatic S7-1200, CPU1214C [5]

2.1 Technické parametry

Rozměry (mm)	110 x 100 x 75
Zdroj napájení	24V DC
Paměť	<ul style="list-style-type: none"> • Pracovní 75 kB • Zaváděcí 4 MB • Zálohovací 10 kB
Bitová paměť (M)	8192 Bytes
Vstupy / výstupy	<ul style="list-style-type: none"> • Digitální 14 vstupů / 10 výstupů • Analogové 2 vstupy
Typ výstupů	Tranzistor
Vysokorychlostní čítače	<ul style="list-style-type: none"> • Počet 6 • 1F 3/100kHz, 3/30kHz • 2F 3/80kHz, 3/20kHz
Komunikační porty	1 x RJ-45
Rozhraní	Profinet

Tab. 3: Technické parametry S7-1200 CPU1214C [6]

2.2 Rozšiřitelnost

Model používaný v této bakalářské práci je možné rozšířit o tři komunikační moduly, jeden signální modul a až osm signálních karet.

2.2.1 Komunikační moduly

Slouží k zajištění komunikace mezi automatem a dalšími zařízeními jako jsou např. tiskárny, RFID zařízení, GSM modemy nebo ke komunikaci po síti Profibus a to buď jako master nebo slave.

- CM 1241 RS232
- CM 1241 RS422/485
- CM 1243-2 AS-i Master
- DCM 1271 AS-i Data Decoupling
- CM1242-5 PROFIBUS DP-Slave
- CM 1243-5 PROFIBUS DP-Master
- CP 1247-7 GPRS

2.2.2 Signální moduly

Umožňují rozšířit počet digitálních a analogových vstupů/výstupů bez přidání signálních karet. Pomocí signálních modulů lze také zvýšit frekvenci na vstupech/výstupech až na 200 kHz.

- SB 1221 DC 200 kHz
- SB 1222 DC 200 kHz
- SB 1223 DC/DC
- SB 1223 DC/DC 200 kHz
- SB 1232 AQ
- SB 1231 AI
- SB 1231 RTD
- SB 1231 TC
- CB 1241 RS485
- BB 1297

2.2.3 Signální karty

Pomocí signálních karet je možné rozšířit počet digitálních a analogových vstupů/výstupů

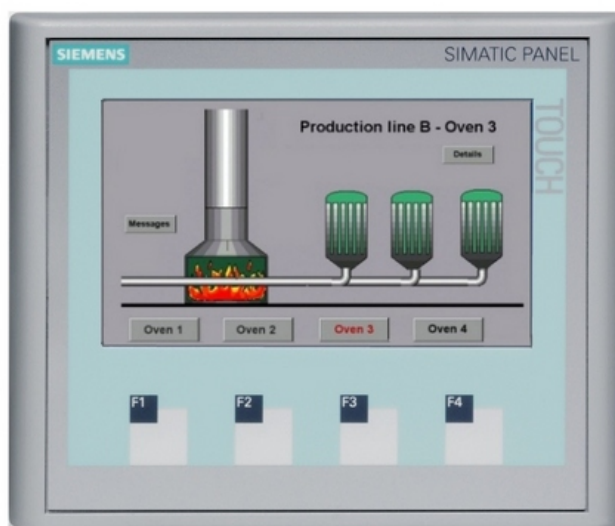
- SM 1221 DC
- SM 1222 DC
- SM 1222 RLY
- SM 1223 DC/DC
- SM 1223 DC/RLY
- SM 1223 AC/RLY
- SM 1231 AI
- SM 1231 RTD
- SM 1231 TC
- SM 1232 AQ
- SM 1234 AI/AQ

2.2.4 Ostatní příslušenství

Mezi další příslušenství patří napájecí zdroj, komunikační procesor umožňující monitorování a řízení na dálku, modemový modul, modul ISDN, paměťové karty, simulátor vstupů, prodlužovací kabely a 4 pásmová GSM anténa. [7]

3 SIEMENS SIMATIC HMI KTP400 BASIC COLOR

Jedná se o HMI panel patřící do kategorie Basic panelů určených pro méně náročné aplikace. Panel je možné použít s celou řadou programovatelných automatů a to jak od firmy Siemens tak od firem jako Allen Bradley, Modicon, Mitsubishi nebo Omron. Po síti komunikuje pomocí protokolu Profinet. Aplikace je možné psát ve vývojovém prostředí WinCC Basic V11 nebo TIA Portal. Specifikace panelu jsou uvedeny v kap. 3.1. [8]



Obr. 4: Siemens Simatic HMI KTP400 Basic color [10]

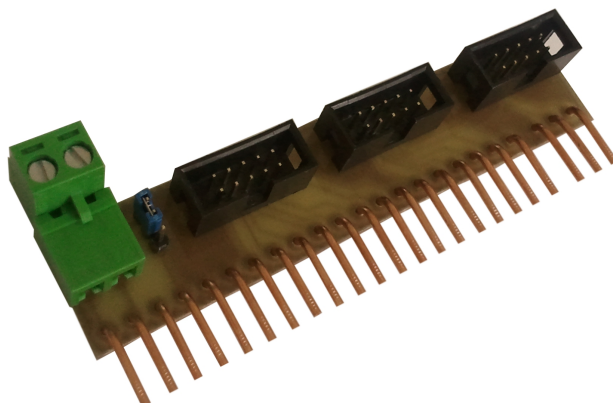
3.1 Technické parametry

Display	4.3“ TFT širokoúhlý display, 256 barev
Rozlišení	480 x 272 pixelů
Ovládací prvky	Dotykový rezistivní display 4 konfigurovatelné klávesy
Uživatelská paměť	512 kB
Rozhraní	Profinet RJ-45
Napájení	24V DC

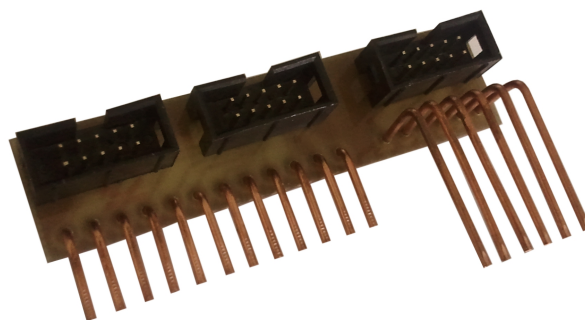
Tab. 4: Technické parametry [9]

4 DESKA PLOŠNÝCH SPOJŮ

Aby bylo možno bezšroubově propojit EDU-mod modely s programovatelným automatem, bylo potřeba vytvořit dvě propojovací desky plošných spojů (DPS), přičemž jedna slouží pro připojení vstupů a druhá k připojení výstupů. DPS byly vytvořeny ve spolupráci s jinou bakalářskou a diplomovou prací, jejímiž autory jsou pan Polák a Jirčák.



Obr. 5: Propojovací DPS pro vstupy

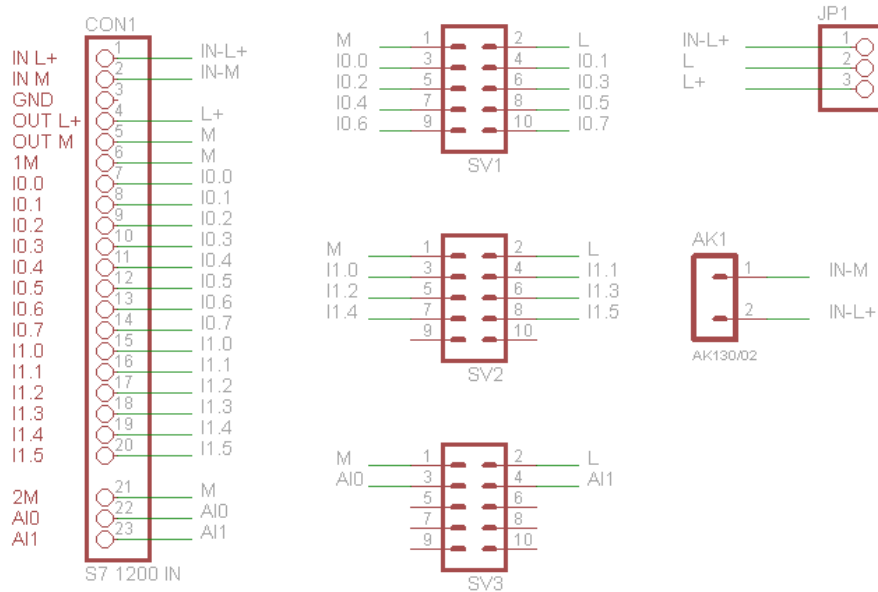


Obr. 6: Propojovací DPS pro výstupy

4.1 Návrh

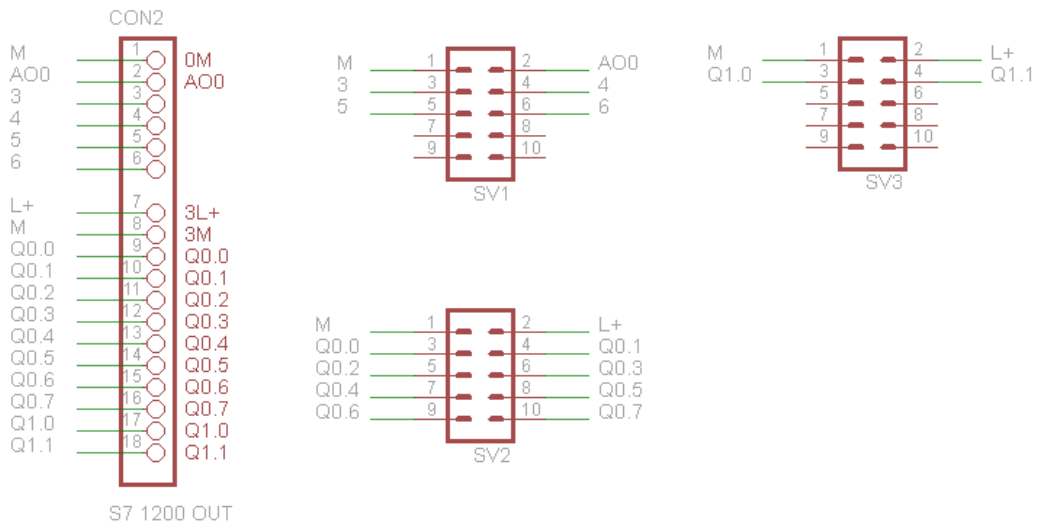
DPS byly navrženy v programu Eagle, což je editor plošných spojů od firmy CadSoft. Nejdříve bylo nutné vytvořit součástky, které se v knihovně nenachází. Jedná se o konektory, které mají rozteče děr shodné s roztečí děr svorkovnice na programovatelném automatu. Protože svorkovnice na programovatelném automatu pro vstupy a výstupy mají rozdílnou rozteč a počet děr, bylo nutné vytvořit dva konektory.

Na DPS pro vstupy se nachází tři deseti-pinové konektory Molex, konektor pro napájení a jeden tří-pinový jumper. Pomocí tohoto jumperu se přepíná způsob napájení EDU-mod modelů. Modely je možno napájet z automatu, abychom ale omezili jeho proudovou zátěž můžeme přepnout na napájení modelů přímo ze zdroje. Všechny součástky jsou propojeny s vytvořeným konektorem podle obr. 7.



Obr. 7: Schéma zapojení přechodové desky pro vstupy

Přechodová deska pro výstupy obsahuje pouze tři deseti-pinové konektory Molex, které jsou propojeny s vytvořeným konektorem pro výstupy podle obr. 8.



Obr. 8: Schéma zapojení přechodové desky pro výstupy

4.2 Osvit

Před samotným osvitom je nutné si vytisknout předlohu, v našem případě byla předloha vytisknuta na fólii. Vytištěnou předlohu přiložíme stranou s potiskem na DPS s fotoemulzí. Aby jsme zamezili rozostření obrazu při osvitom, je nutné aby byla předloha dobře přitížena k DPS. Proto předlohu překryjeme sklem, které důkladně zatížíme. Osvit byl prováděn osvitovou jednotkou osazenou UV LED diodami, po dobu 5,5 minuty, ze vzdálenosti cca 20 cm.

4.3 Vyvolávání

Po osvětlení desku vyvoláme v roztoku NaOH (hydroxid sodný). Princip spočívá v tom, že osvětlená fotoemulze se v roztoku rozpustí, kdežto neosvětlený lak zůstane na DPS. Vyvolávání je hotovo jakmile se z DPS přestane smývat rozpuštěná fotoemulze a zřetelně vyniknou všechny cesty. V našem případě trvalo vyvolávání zhruba 3 minuty. Po vyvolání je nutné desku opláchnout vodou aby se odstranily zbytky vývojky a fotoemulze.

4.4 Leptání

Leptání provádíme pomocí FeCl_3 (Chlorid železitý). Opláchnutou a osušenou desku položíme na hladinu leptadla a to vrstvou mědi směrem dolů tak aby odleptaná měď odpadávala ke dnu nádoby.

Doba leptání se odvíjí od čerstvosti chlorydu železitého a také jeho teploty. Pokud je chlorid železitý nový trvá leptání zhruba 10 minut, používáním se potřebná doba prodlužuje. Se vzrůstající teplotou roztoku se naopak potřebná doba snižuje. V našem případě trvalo leptání 20 minut. Po leptání je nutné desku opět opláchnout vodou abychom smyli roztok chloridu železitého.

4.5 Vyčištění a osazení desky

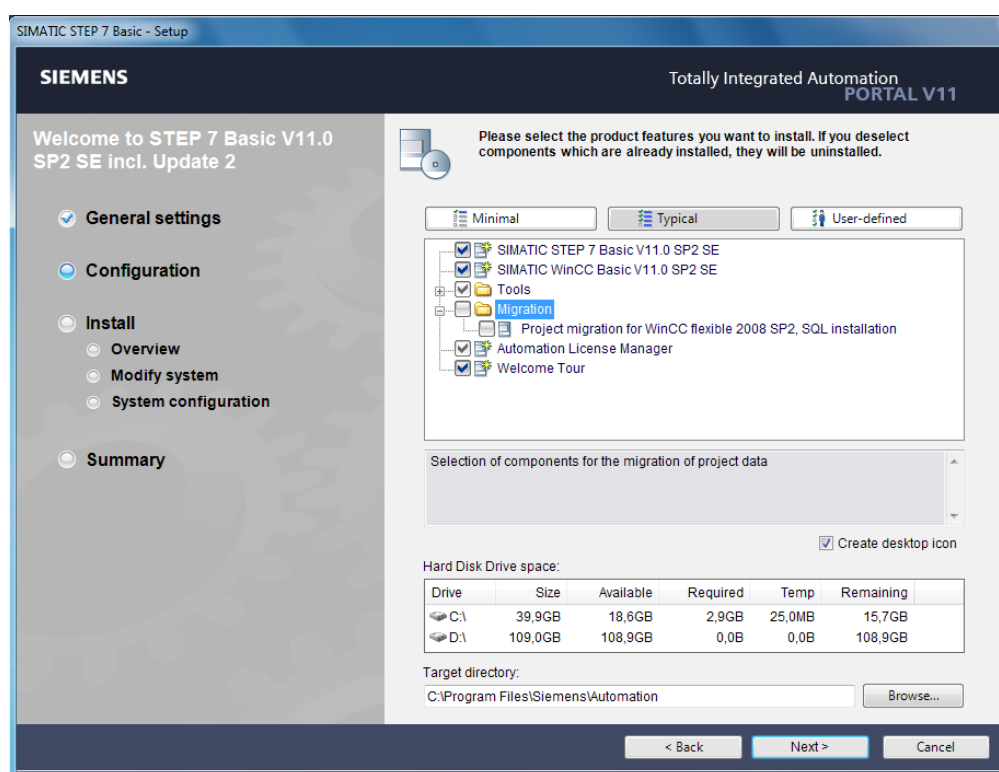
Po vyleptání desky je potřeba odstranit zbylou fotoemulzi. To lze provést dvěma způsoby. Buď desku znovu osvětlíme pod osvitovou jednotkou a vyvoláme čímž se všechny lak odstraní, nebo jako v našem případě desku vyčistíme acetonem, čímž dosáhneme stejného výsledku. Nyní vyvrtáme pomocí stojanové vrtačky díry a poté desku přebrousíme jemným smirkovým papírem. Desku ještě jednou očistíme acetonem a nastříkáme pájitelným lakem, který zlepšuje pájení a zároveň zabráňuje oxidaci mědi. Po zaschnutí laku, který schne přibližně 12 hodin můžeme napájet součástky.

5 SIEMENS TIA PORTAL

TIA Portal je software od firmy Siemens, který v sobě integruje vývojové prostředí pro programování PLC (Step7 V11) a prostředí pro vytváření HMI aplikací (WinCC V11). Software existuje ve dvou verzích a to Basic a Professional. Tyto verze se liší podporou automatů. Zatím co verze Basic umožňuje programovat pouze automaty řady S7-1200, verze Professional podporuje i řady S7-300, S7-400 a WinAC. Mezi podporované jazyky patří jazyk FBD, LAD, STL, S7-SCL, S7-GRAPH. [11]

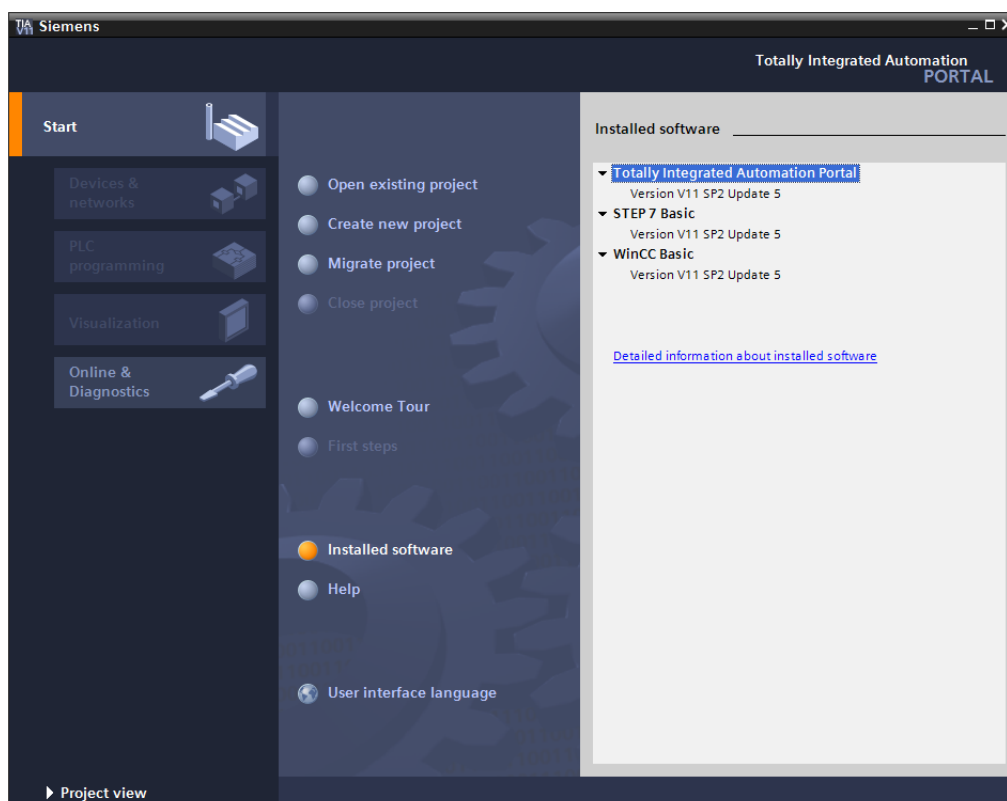
5.1 Instalace

V našem případě se jedná o TIA Portal V11 SP 2 s přiloženým updatem 2. Verze programů Step7 i WinCC je Basic. Během instalace jsem zvolil typickou instalaci, tím se nainstaluje vše kromě migračních nástrojů pro PLC programy, které slouží k přenosu programů ze starších verzí TIA Portal nebo Step7. U migrace ze Step7 je podmínkou minimální verze V5.4 s SP5.



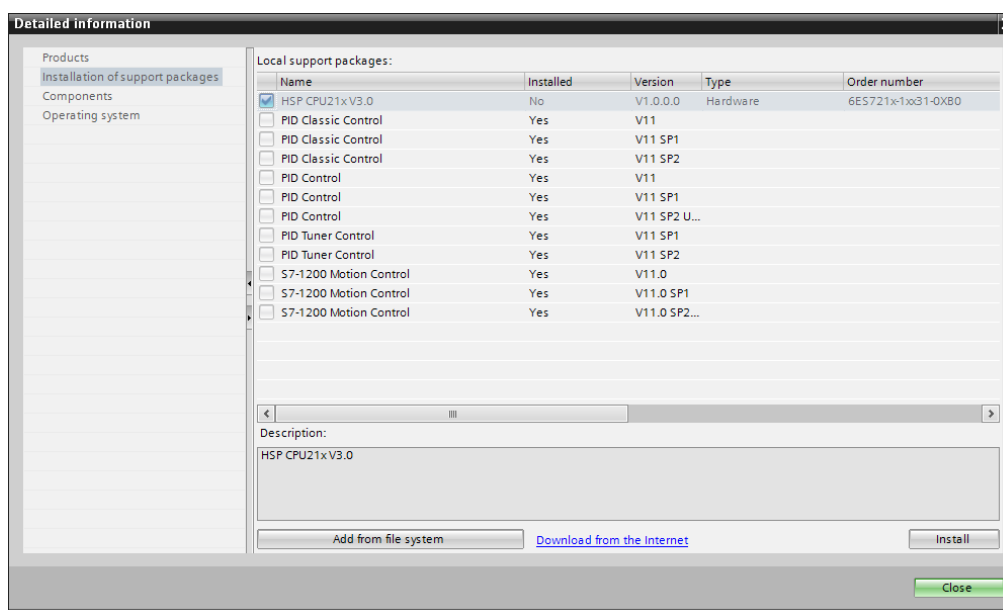
Obr. 9: Instalace TIA Portal

Po úspěšné instalaci je nutné zkontrolovat aktualizace. V našem případě je k dispozici update 5. Tento update zlepšuje stabilitu programu při migraci projektů, v editoru SCL a při navrhování HMI aplikací [12]. Nainstalované updatey můžeme zkontrolovat ve spuštěném programu po odkliknutí tlačítka „Installed software“. Detailní výpis komponentů zobrazíme kliknutím na odkaz „Detailed information about installed software“.



Obr. 10: Nainstalovaný software

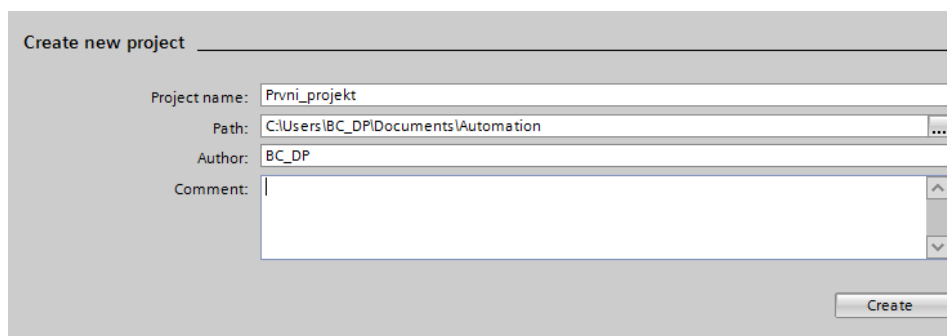
Poslední věcí, kterou je nutné provést je aktualizace hardwarového katalogu a to jak PLC tak i HMI. Po stažení potřebných souborů otevřeme v TIA Portal již zmiňovaný odkaz „Detailed information about installed software“ a v něm záložku „Installation of support packages“. Tlačítkem „Add from file system“ vybereme stažený update, čímž ho přidáme do seznamu. Instalaci provedeme označením updatu v seznamu a stiskutím tlačítka „install“. Aby se instalace spustila je nutné TIA Portal ukončit. U aktualizace HW katalogu pro HMI postupujeme stejným způsobem.



Obr. 11: Aktualizace HW katalogu

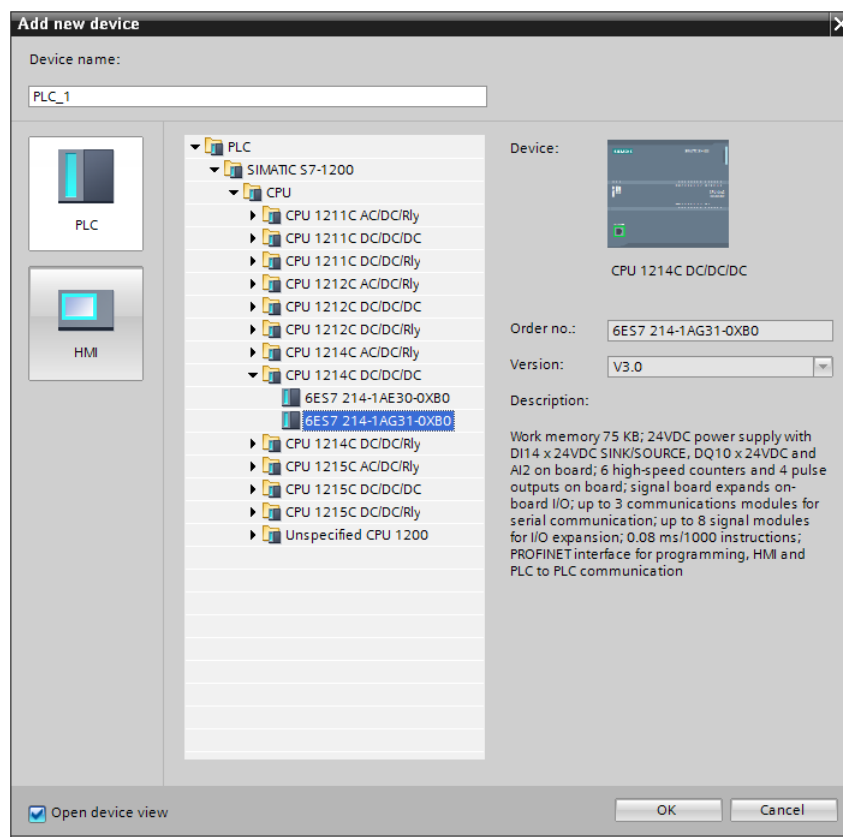
5.2 Tvorba projektu a nastavení zařízení

Po spuštění TIA Portal klikneme na „Create new project“. Vyplníme název projektu, jméno autora, popřípadě komentář a zvolíme si místo uložení projektu. Tlačítkem „Create“ projekt vytvoříme. Pro zobrazení projektu slouží tlačítko „Project view“.



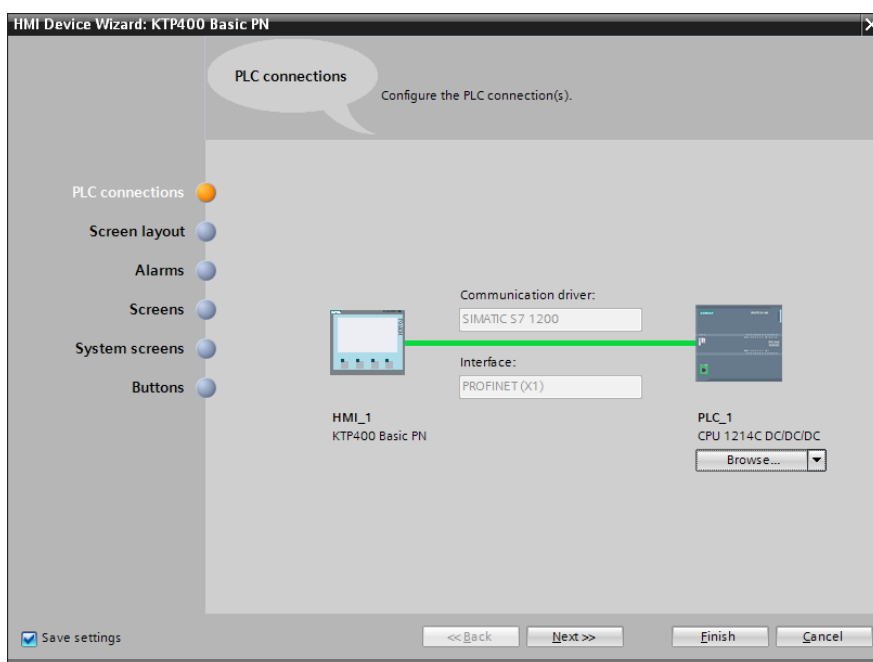
Obr. 12: Tvorba projektu

První věc, kterou musíme udělat po vytvoření projektu je přidat a nakonfigurovat zařízení, to provedeme tlačítkem „Add new device“ v okně Project tree. Z HW katalogu vybereme odpovídající zařízení. V našem případě se jedná o PLC CPU1214 DC/DC/DC (6ES7 214-1AG31-0XB0) a HMI panel KTP400 Basic PN (6AV6647-0AK11-3AX0).




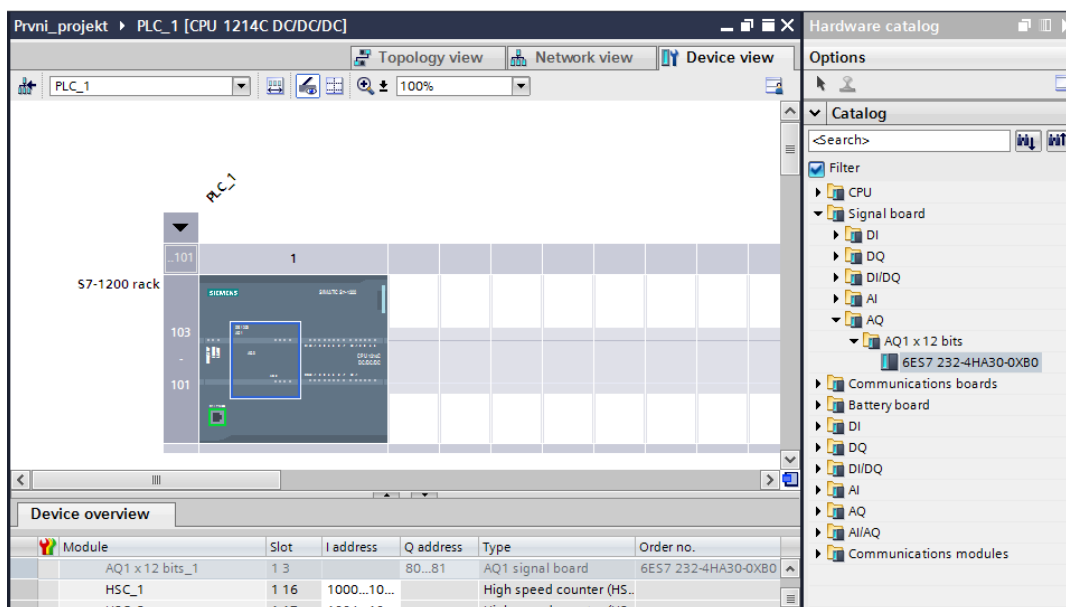
Obr. 13: Výběr zařízení z HW katalogu

Po přidání HMI panelu se otevře konfigurační okno. Hned na první stránce se nachází nejdůležitější nastavení a to, se kterým zařízením chceme HMI panel propojit. Vyhledáme tedy naše PLC a potvrdíme, čímž se nám obě zařízení navzájem propojí. Na dalších stránkách si můžeme nastavit vzhled základní obrazovky, tlačítka, popřípadě počet obrazovek. Všechny tyto vlastnosti jde ovšem nastavit i v průběhu vytváření projektu.



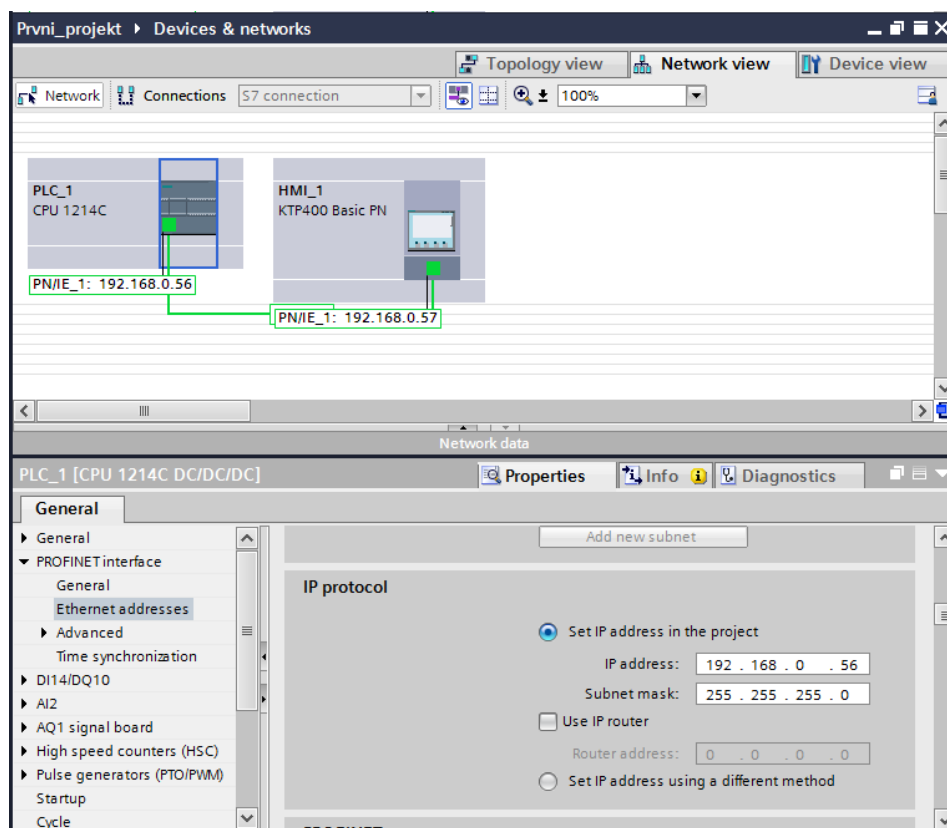
Obr. 14: Konfigurační okno HMI panelu

Propojení PLC a HMI panelu můžeme zkontrolovat otevřením položky „Devices & networks“ v okně Project tree. Tlačítkem „Show address label“  si můžeme zobrazit IP adresy zařízení. Další věcí, kterou musíme udělat je přiřazení signálního modulu k automatu. V okně „Devices & networks“ dvojklikem poklepeme na PLC. V nově otevřeném okně vybereme z HW katalogu po pravé straně signal board označený na obr. 15 a myší jej přetáhneme na modře vyznačené místo na automatu.




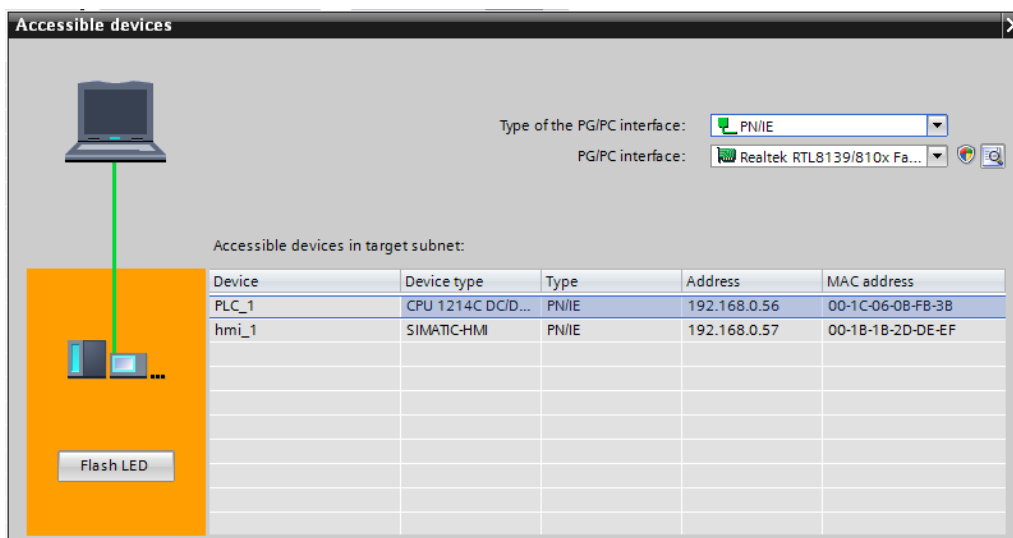
Obr. 15: Výběr a přiřazení signálního modulu

Poslední věcí, kterou musíme udělat je nastavení IP adres. To se provádí v okně „Devices & networks“, kde si označíme zařízení, u kterého chceme změnit adresu. V okně Network data, které se nachází ve spodní části obrazovky vybereme záložku „Properties“. Ze seznamu na levé straně otevřeme položku „PROFINET interface“ a v ní vybereme záložku „Ethernet addresses“.



Obr. 16: Nastavení IP adresy

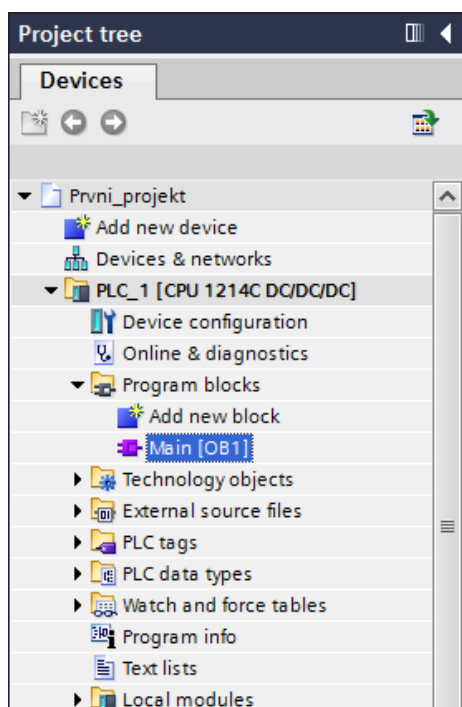
Pomocí tlačítka „Accessible devices“  můžeme zkontrolovat komunikaci mezi PC a dostupnými zařízeními. První ale musíme nastavit typ PG/PC interface kde vybereme „PN/IE“, po vybrání síťové karty se spustí hledání zařízení. Správnou komunikaci můžeme otestovat vybráním zařízení a zmačknutím tlačítka „Flash LED“, které nám v případě PLC rozbliká diody na automatu nebo v případě HMI panelu rozsvítí display.



Obr. 17: Vyhledání dostupných zařízení

5.3 Tvorba programu

Po přidání programovatelného automatu z HW katalogu se nám automaticky vytvořil programový blok Main, který je v jazyce kontaktních schémat. Tento blok můžeme najít v okně „Project tree“ po otevření záložky s názvem programovatelného automatu a v ní záložky „Program blocks“ viz. obr. 18



Obr. 18: Okno Project tree

Protože v této práci bude používán jazyk strukturovaného textu, firmou Siemens značený jako SCL, tak blok Main smažeme a vytvoříme nový blok v jazyce SCL. To provedeme tlačítkem „Add new blocks“. Po jeho stisknutí se nám otevře okno s výběrem typu bloku. Na výběr máme čtyři možnosti.

Organization block (Programový blok)

Jedná se o blok, ve kterém se nachází hlavní program. Obsahuje statickou paměť. Z programového bloku je možné volat jak funkční bloky tak funkce.

Function block (Funkční blok)

Funkční bloky obsahují paměť v podobě instance datových bloků. Po vykonání programu ve funkčním bloku se data uloží do příslušného datového bloku. Z funkčních bloků je možné volat další funkční bloky, kromě sama sebe a funkce.

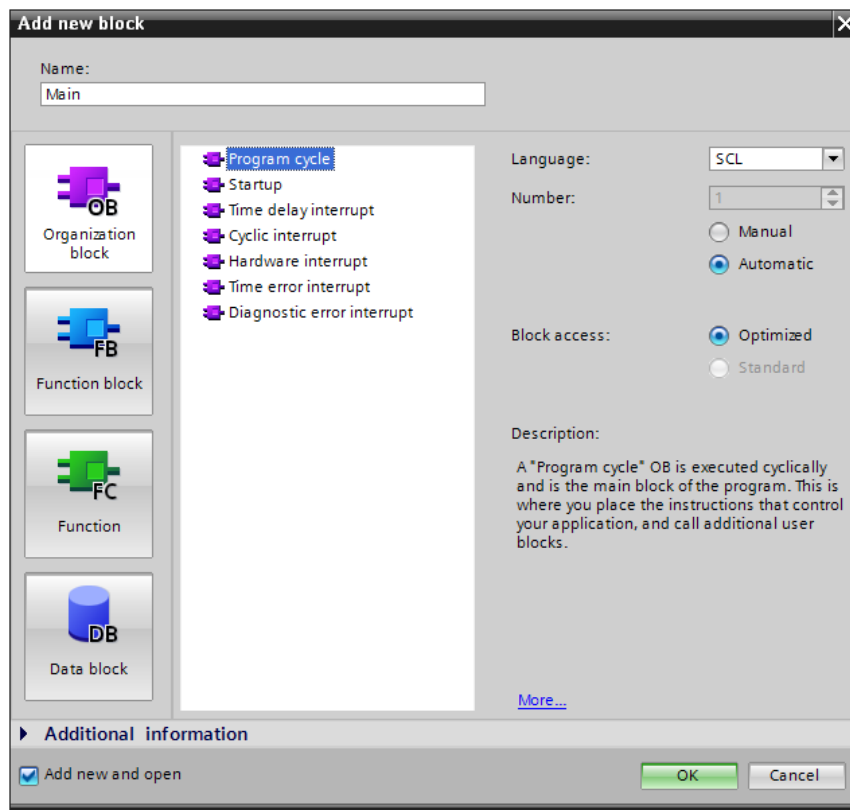
Function (Funkce)

Blok Funkce se odlišuje od ostatních bloků tím, že neobsahuje paměť. Výstup funkce můžeme uložit do bitové paměti M nebo do globálního datového bloku.

Data block (Datový blok)

Datový blok slouží jako paměť. Tento blok neobsahuje žádný kód. [13]

V našem případě vybereme programový blok, který nazveme Main. Ze seznamu vybereme „Program cycle“ což je cyklicky spouštěný program a jako jazyk vybereme jazyk strukturovaného textu SCL.



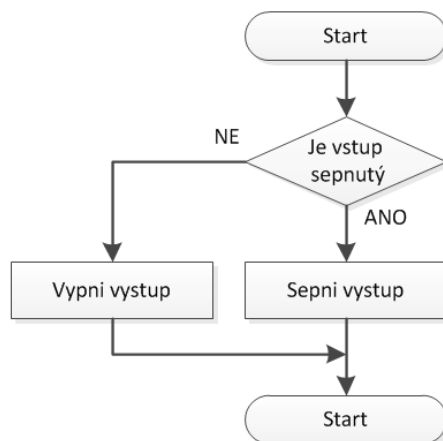
Obr. 19: Přidání nového bloku

Zadání programu

Program bude obsahovat jeden vstup a výstup. Po sepnutí vstupu se aktivuje výstup v opačném případě se výstup vypne.

Řešení

Princip programu je velmi jednoduchý a je patrný z vývojového diagramu na obr. 20



Obr. 20: Vývojový diagram

Nejprve si vyplníme tabulku symbolických proměnných, kterou nalezneme v okně „Project tree“ pod záložkou „PLC tags“. V tomto případě nebudeme vytvářet novou tabulku, ale využijeme tabulku defaultní, kterou si otevřeme poklikáním. Do tabulky si přidáme proměnnou „Vstup“ datového typu „Bool“ a přiřadíme jí adresu vstupu I0.0 na kterém bude přivedeno tlačítko. Další proměnnou bude „Vystup“ se stejným datovým typem a adresou výstupu Q0.0.

	Name	Tag table	Data type	Address	Retain	Visible..	Acces...	Comment
1	Vstup	Default tag table	Bool	%I0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Vstup I0.0
2	Vystup	Default tag table	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Výstup Q0.0

Obr. 21: Tabulka symbolických proměnných

Do programového bloku Main zapíšeme následující kód.

```
IF "Vstup" THEN           // Pokud je sepnutý Vstup
  "Vystup" := true;      // Aktivuj Vystup
ELSE                       // Jinak
  "Vystup" := false;    // Vypni Vystup
END_IF;                   // Konec podmínky IF
```

Nyní můžeme program nahrát do automatu pomocí tlačítka „Download to device“ . Po stisknutí tlačítka se program automaticky zkompileje.

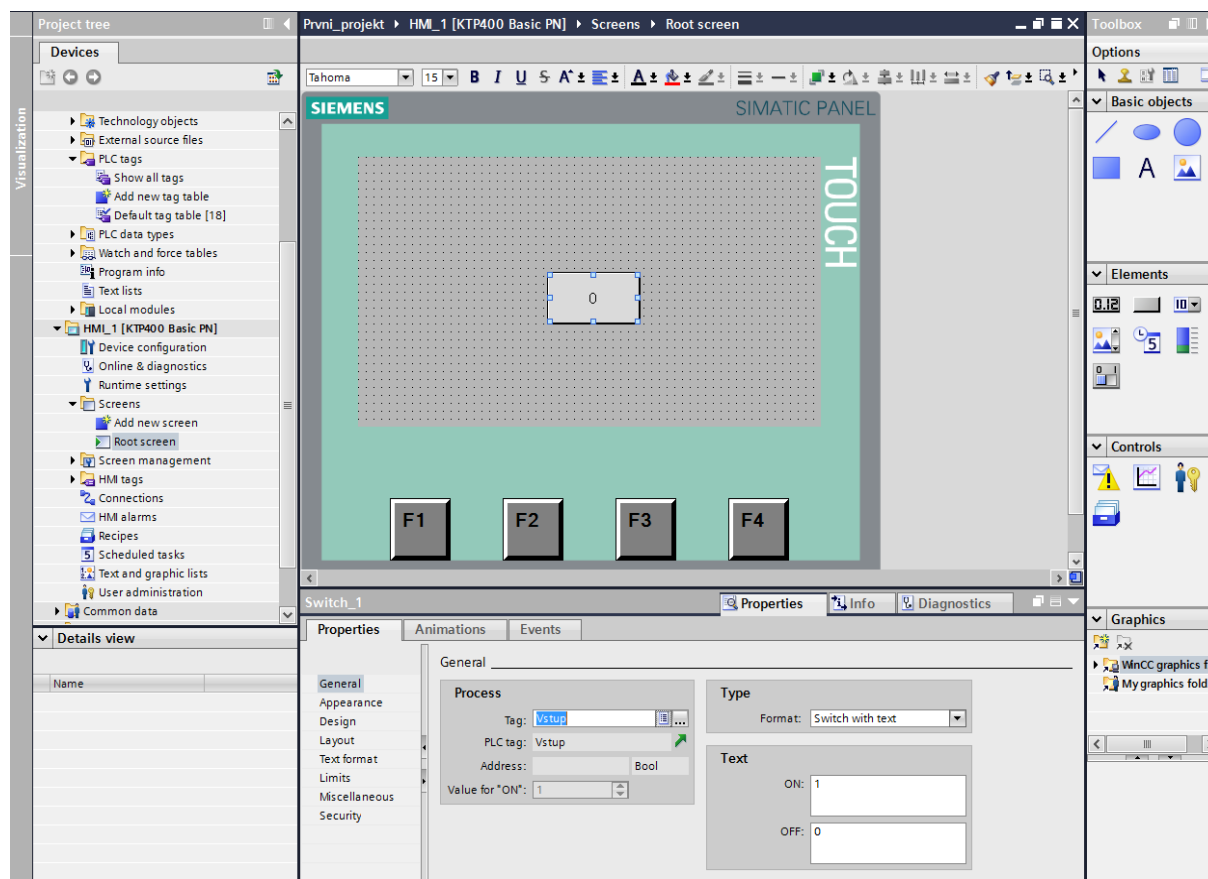
Rozšíření o HMI panel

Zadání si rozšíříme o HMI panel pomocí kterého budeme spouštět či vypínat výstup. Protože k ovládání budeme využívat tlačítko na HMI panelu, upravíme tabulku symbolických proměnných a to konkrétně adresu u proměnné „Vstup“ na M0.0. Jedná se o bitovou paměť M uvnitř automatu.

	Name	Tag table	Data type	Address	Retain	Visible..	Acces...	Comment
1	Vstup	Default tag table	Bool	%M0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Vstup M0.0
2	Vystup	Default tag table	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Výstup Q0.0

Obr. 22: Tabulka symbolických proměnných upravená pro použití HMI panelu

V okně „Project tree“ otevřeme HMI zařízení a v něm vybereme záložku „Screens“. V této záložce se nachází obrazovka „Root screen“ což je obrazovka, která se spustí po zapnutí programu. Z okna „Toolbox“, které se nachází na pravé straně obrazovky vybereme ze záložky „Elements“ přepínač (switch) a přetáhneme ho na obrazovku. Ve spodním okně otevřeme záložku „Properties“, poté záložku „General“ a u položky „Tag“ přiřadíme proměnnou „Vstup“. Po tomto nastavení se při sepnutí tlačítka nastaví proměnná „Vstup“ na logickou hodnotu 1 a při vypnutí na logickou hodnotu 0.



Obr. 23: Návrh obrazovky HMI panelu

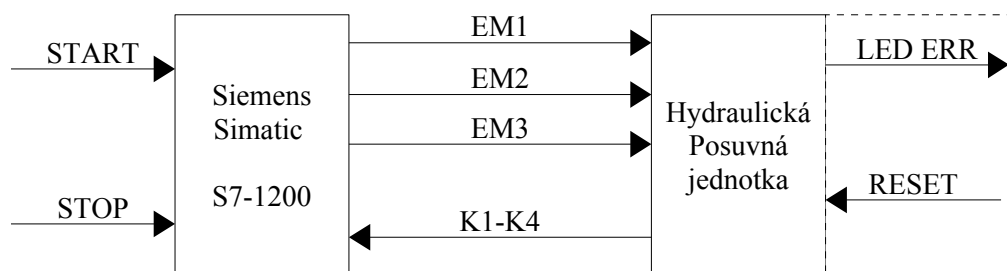
Nyní stačí pouze nahrát program do zařízení. Nesmíme ovšem zapomenout nahrát i program do PLC a to proto, že jsme měnili tabulku symbolických proměnných.

6 ŘEŠENÉ ÚLOHY

Protože je tato bakalářská práce pojatá jako podpora k výuce programovatelných automatů, rozdělil jsem základní úlohy na několik podúloh pro snazší pochopení problematiky. V této kapitole se budu věnovat pouze návrhu a řešení zadaných úloh.

6.1 Hydraulická posuvná jednotka

Hydraulickou posuvnou jednotku budeme ovládat pomocí tlačítek START a STOP. Schéma zapojení můžete vidět na obrázku obr. 24. Detailnější popis jednotky se nachází v kapitole 1.1.



Obr. 24: Schéma propojení PLC s EDU-mod modelem

Adresy konkrétních symbolických proměnných zjistíme pomocí tab. 1.

Symbolická proměnná	Adresa	Popis
K1	I0.4	Snímač polohy K1
K2	I0.5	Snímač polohy K2
K3	I0.6	Snímač polohy K3
K4	I0.7	Snímač polohy K4
START	I1.0	Tlačítko START
STOP	I1.1	Tlačítko STOP
EM1	Q0.0	Pohyb vpravo
EM2	Q0.1	Pohyb vlevo
EM3	Q0.2	Zpomalení posuvu

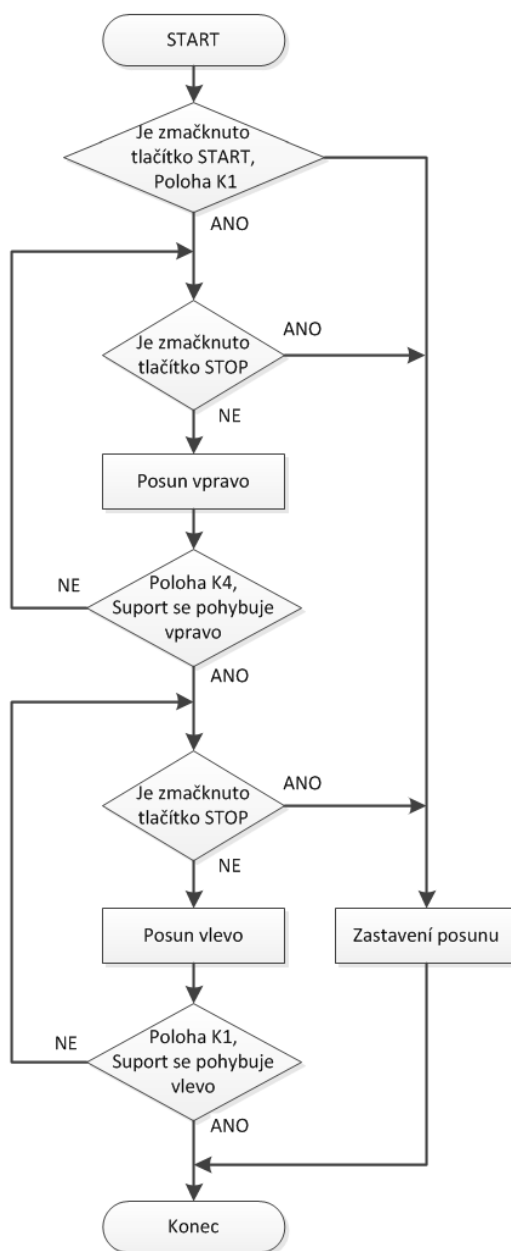
Tab. 5: Adresy symbolických proměnných

6.1.1 Zadání příkladu č. 1

Naprogramujte program pro řízení hydraulické posuvové jednotky tak, aby při stisknutí tlačítka START vykonal suport pohyb rychloposuvem z K1 do K4 a zpět. Tlačítkem STOP lze suport kdykoliv zastavit.

6.1.2 Vypracování příkladu č. 1

Pokud se suport nachází v poloze K1 a stiskneme tlačítko START, sepne se elektromagnet EM1, který způsobí posun suportu vpravo. Po dosažení koncové polohy K4 musíme vypnout elektromagnet EM1 a změnit směr posuvu sepnutím elektromagnetu EM2. Po dosažení počáteční polohy K1 ukončíme posuv vlevo a čekáme na opětovný stisk tlačítka START. Pokud je kdykoliv stisknuto tlačítko STOP, vypnou se oba elektromagnety a tím se zastaví pohyb.



Obr. 25: Vývojový diagram

Nejprve si vytvoříme tabulku symbolických proměnných. U tohoto zadání si vystačíme se snímači polohy K1 a K4, elektromagnety EM1 a EM2, a tlačítka START a STOP.

	Name	Tag table	Data type	Address	Retain	Visi...	Acce..	Comment
1	K1	Posuvna jednotka	Bool	%I0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Poloha K1
2	K4	Posuvna jednotka	Bool	%I0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Poloha K4
3	EM1	Posuvna jednotka	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Elektromotor EM1
4	EM2	Posuvna jednotka	Bool	%Q0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Elektromotor EM2
5	Start	Posuvna jednotka	Bool	%I1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Tlačítka START
6	Stop	Posuvna jednotka	Bool	%I1.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Tlačítka STOP

Obr. 26: Tabulka symbolických proměnných

```

IF "K1" AND "Start" THEN // Pokud je aktivován snímač K1 a je stisknuto tlačítka
    Start
    "EM1" := true; // Zapne se pohyb vpravo
END_IF; // Konec podmínky IF

IF "K4" AND "EM1" THEN // Pokud je aktivován snímač K4 a posun vpravo
    "EM1" := false; // Vypne se posun vpravo
    "EM2" := true; // Zapne se posun vlevo
END_IF; // Konec podmínky IF

IF "K1" AND "EM2" THEN // Pokud je aktivován snímač K1 a posun vlevo
    "EM2" := false; // Vypne se posun vlevo
END_IF; // Konec podmínky

IF "Stop" THEN // Jestliže je stisknuto tlačítka Stop
    "EM1" := false; // Vypne se posun vpravo
    "EM2" := false; // Vypne se posun vlevo
END_IF; // Konec podmínky IF

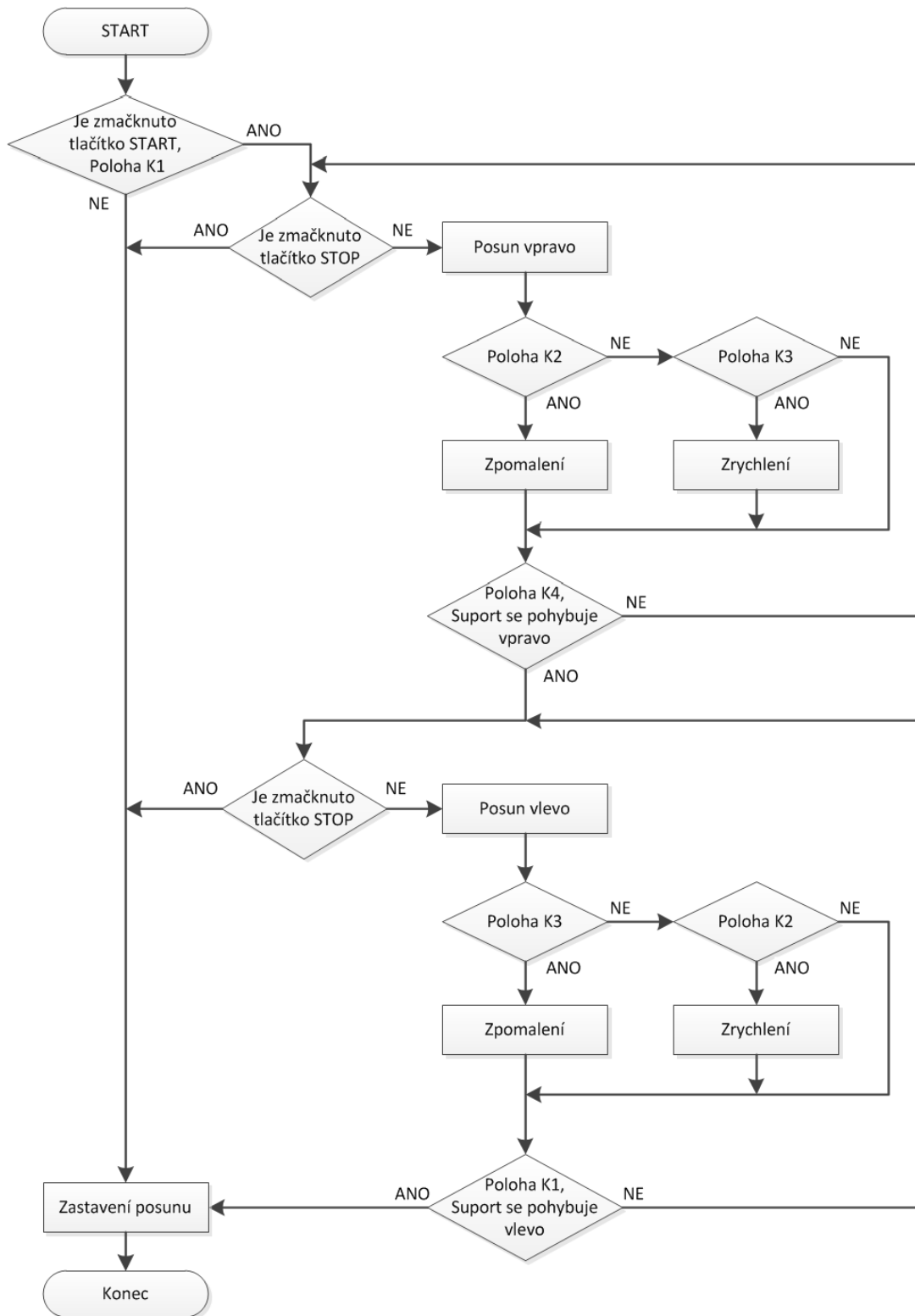
```

6.1.3 Zadání příkladu č. 2

Rozšířte zadání z prvního příkladu tak, že v úseku mezi snímači K2 až K3 se bude suport pohybovat zpomaleným pracovním posuvem a to v obou směrech. Tlačítkem STOP lze suport kdykoliv zastavit.

6.1.4 Vypracování příkladu č. 2

Zadání první úlohy rozšíříme tak, že při posuvu vpravo tzn. při sepnutí elektromagnetu EM1 a současném dosažení polohy K2 zpomalíme posuv pomocí sepnutí elektromagnetu EM3. Po dosažení polohy K3 elektromagnet EM3 opět vypneme. Stejným způsobem budeme postupovat v opačném směru a to pouze s tím rozdílem, že elektromagnet EM3 sepneme tehdy, když je sepnutý elektromagnet EM2 a je dosaženo polohy K3. Zpomalený posuv zrušíme vypnutím EM3 při dosažení polohy K2.



Obr. 27: Vývojový diagram

Tabulku symbolických proměnných si doplníme o snímače polohy K2, K3 a elektromagnet EM3. Protože se nám počet výstupů rozšířil, zavedeme si proměnnou „Vystupy“. Datový typ nastavíme na Byte a adresu na QB0. Tím jsme vytvořili proměnnou o velikosti 8 bitů, kde jednotlivé bity představují prvních osm výstupů od adresy Q0.0, přičemž nejnižším bitem (LSB) je adresa Q0.0 a nejvyšším bitem (MSB) adresa Q0.7. Pro vyresetování všech výstupů potom stačí pouze přiřadit proměnné „Vystupy“ hodnotu 0.

	Name	Tag table	Data type	Address	Retain	Visible..	Acces...	Comment
1	K1	Posuvna jednotka	Bool	%I0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Poloha K1
2	K2	Posuvna jednotka	Bool	%I0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Poloha K2
3	K3	Posuvna jednotka	Bool	%I0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Poloha K3
4	K4	Posuvna jednotka	Bool	%I0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Poloha K4
5	EM1	Default tag table	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Elektromagnet EM1
6	EM2	Posuvna jednotka	Bool	%Q0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Elektromagnet EM2
7	EM3	Posuvna jednotka	Bool	%Q0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Elektromagnet EM3
8	Start	Posuvna jednotka	Bool	%I1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Tlačítko START
9	Stop	Posuvna jednotka	Bool	%I1.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Tlačítko STOP
10	Vystupy	Posuvna jednotka	Byte	%QB0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Proměnná Vystupy

Obr. 28: Tabulka symbolických proměnných

```

IF "K1" AND "Start" THEN // Pokud je aktivován snímač K1 a je stisknuto tlačítko
    Start
    "EM1" := true; // Zapne se pohyb vpravo
END_IF; // Konec podmínky IF

IF "K2" THEN // Jestliže je aktivován snímač K2
    IF "EM1" THEN // Pokud se suport pohybuje vpravo
        "EM3" := true; // Zpomalí se posuv
    END_IF; // Konec podmínky IF

    IF "EM2" THEN // Pokud se suport pohybuje vlevo
        "EM3" := false; // Zrychlí se posuv
    END_IF; // Konec podmínky IF
END_IF; // Konec podmínky IF

IF "K3" THEN // Jestliže je aktivován snímač K3
    IF "EM1" THEN // Pokud se suport pohybuje vpravo
        "EM3" := false; // Zrychlí se posuv
    END_IF; // Konec podmínky IF

    IF "EM2" THEN // Pokud se suport pohybuje vlevo
        "EM3" := true; // Zpomalí se posuv
    END_IF; // Konec podmínky IF
END_IF; // Konec podmínky IF

IF "K4" AND "EM1" THEN // Pokud je aktivován snímač K4 a posun vpravo
    "EM1" := false; // Vypne se posun vpravo
    "EM2" := true; // Zapne se posun vlevo
END_IF; // Konec podmínky IF

IF "K1" AND "EM2" THEN // Pokud je aktivován snímač K1 a posun vlevo
    "EM2" := false; // Vypne se posun vlevo
END_IF; // Konec podmínky IF

IF "Stop" THEN // Jestliže je stisknuto tlačítko Stop
    "Vystupy" := 0; // Vypnou se všechny elektromagnety
END_IF; // Konec podmínky IF

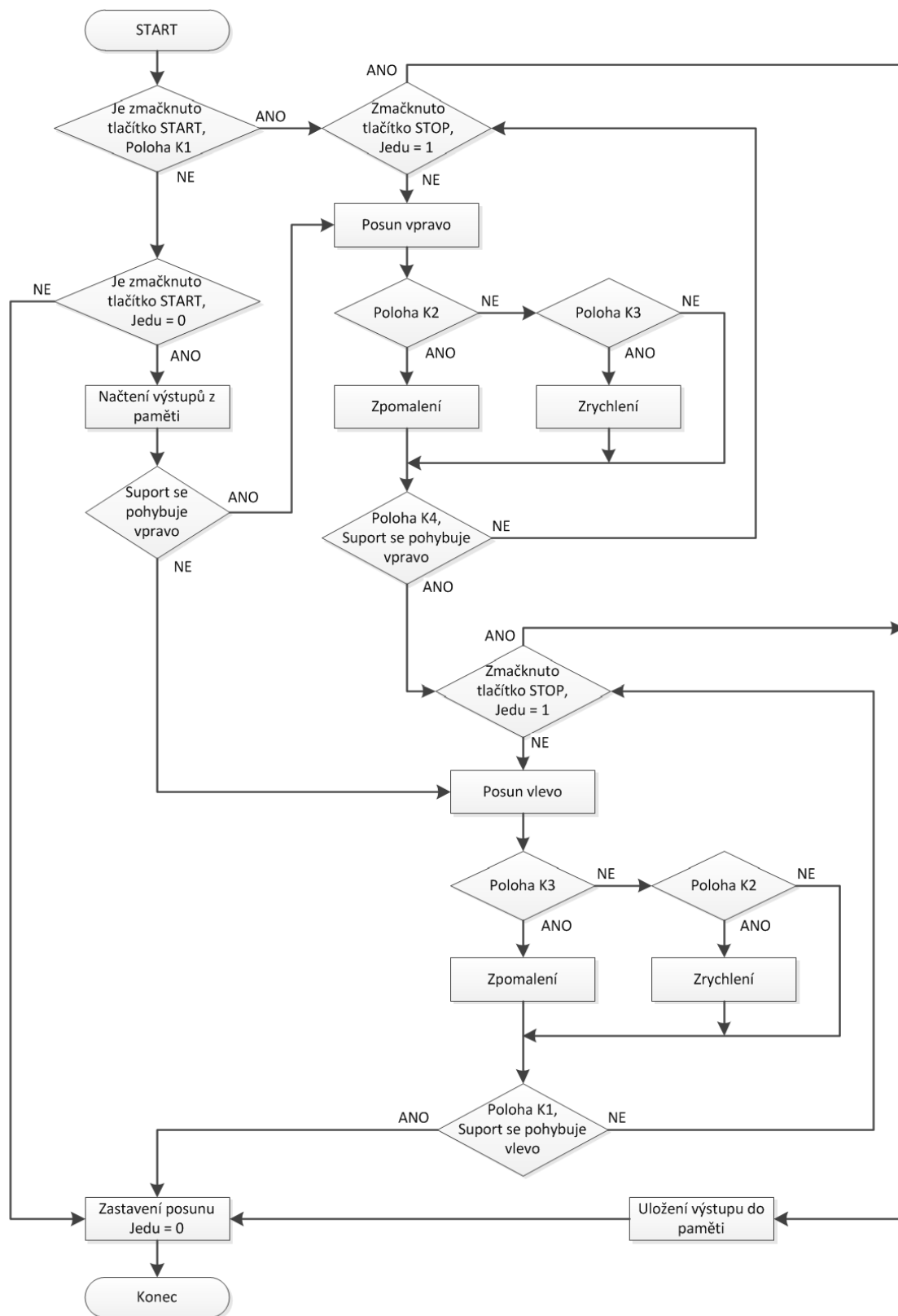
```

6.1.5 Zadání příkladu č. 3

Rozšířte zadání z druhého příkladu tak, aby po přerušení tlačítkem STOP a po opětovném spuštění tlačítkem START, jednotka pokračovala v přerušeném pohybu.

6.1.6 Vypracování příkladu č. 3

Aby bylo možné při opětovném spuštění pokračovat v přerušené činnosti, je nutné si při zastavení jednotky tlačítkem STOP uložit stavy výstupů do paměti. Po opětovném spuštění tlačítkem START automat načte stav výstupů z paměti a odešle je na výstupy, čímž se sepnou požadované elektromagnety ovládající pohyb. Dále je potřeba zavést bitovou proměnnou, v našem případě pojmenovanou „Jedu“, která indikuje pohyb suportu. Pokud je logická hodnota rovna 1, znamená to, že je právě suport v pohybu. V opačném případě je jednotka zastavena. Tato proměnná je zavedena z toho důvodu aby zamezila uložení výstupů do paměti při vícenásobném použití tlačítka STOP, kdy jsou výstupy vyresetovány.



Obr. 29: Vývojový diagram

Do tabulky symbolických proměnných nám přibude bitová proměnná „Jedu“, která je uložena v bitové paměti M. Dále si do tabulky přidáme proměnnou „Pamet“, do které se budou ukládat stavy výstupů. Protože v úloze máme tři výstupy, musíme alokovat pamět minimálně o velikosti 1 Byte. Jako adresu si zvolíme MB0, tedy prvních 8 bitů počínající adresou M0.0.

	Name	Tag table	Data type	Address	Retain	Visible..	Acces...	Comment
1	K1	Posuvna jednotka	Bool	%I0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Poloha K1
2	K2	Posuvna jednotka	Bool	%I0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Poloha K2
3	K3	Posuvna jednotka	Bool	%I0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Poloha K3
4	K4	Posuvna jednotka	Bool	%I0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Poloha K4
5	EM1	Posuvna jednotka	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Elektromagnet EM1
6	EM2	Posuvna jednotka	Bool	%Q0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Elektromagnet EM2
7	EM3	Posuvna jednotka	Bool	%Q0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Elektromagnet EM3
8	Start	Posuvna jednotka	Bool	%M1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Tlačítko START
9	Stop	Posuvna jednotka	Bool	%M1.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Tlačítko STOP
10	Jedu	Posuvna jednotka	Bool	%M1.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Příznak pohybu
11	Vystupy	Posuvna jednotka	Byte	%QB0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Proměnná Vysupy
12	Pamet	Posuvna jednotka	Byte	%MB0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Pamět pro uložení stavu výstupů

Obr. 30: Tabulka symbolických proměnných

```

IF "K1" AND "Start" THEN // Pokud je aktivován snímač K1 a je stisknuto tlačítko
                          // Start
    "EM2" := true;       // Vypne se pohyb vlevo
    "EM1" := true;       // Zapne se pohyb vpravo
    "Jedu" := true;      // Nastaví se příznak pohybu
END_IF;                 // Konec podmínky IF

IF "K2" THEN           // Jestliže je aktivován snímač K2
    IF "EM1" THEN      // Pokud se suport pohybuje vpravo
        "EM3" := true; // Zpomalí se posuv
    END_IF;            // Konec podmínky IF

    IF "EM2" THEN      // Pokud se suport pohybuje vlevo
        "EM3" := false; // Zrychlí se posuv
    END_IF;            // Konec podmínky IF
END_IF;                // Konec podmínky IF

IF "K3" THEN           // Jestliže je aktivován snímač K3
    IF "EM1" THEN      // Pokud se suport pohybuje vpravo
        "EM3" := false; // Zrychlí se posuv
    END_IF;            // Konec podmínky IF

    IF "EM2" THEN      // Pokud se suport pohybuje vlevo
        "EM3" := true;  // Zpomalí se posuv
    END_IF;            // Konec podmínky IF
END_IF;                // Konec podmínky IF

IF "K4" AND "EM1" THEN // Pokud je aktivován snímač K4 a posun vpravo
    "EM1" := false;    // Vypne se posun vpravo
    "EM2" := true;     // Zapne se posun vlevo
END_IF;                // Konec podmínky IF

IF "K1" AND "EM2" THEN // Pokud je aktivován snímač K1 a posun vlevo
    "EM2" := false;    // Vypne se posun vlevo
    "Jedu" := false;   // Vynuluje příznak pohybu
END_IF;                // Konec podmínky

```

```

IF "Stop" AND "Jedu" THEN // Pokud je stisknuto tlačítko Stop a Jedu = 1
  "Pamet" := "Vystupy"; // Výstupy se uloží do paměti
  "Jedu" := false; // Vynuluje se příznak pohybu
  "Vystupy" := 0; // Vypnou se všechny elektromagnety
END_IF; // Konec podmínky IF

IF "Start" AND NOT "Jedu" THEN // Pokud je stisknuto tlačítko Start a Jedu = 0
  "Vystupy" := "Pamet"; // Obnoví se výstupy z paměti
  "Jedu" := true; // Nastaví se příznak pohybu
END_IF; // Konec podmínky IF

```

6.1.7 Zadání příkladu č. 4

Rozšířte zadání ze třetího příkladu přidáním HMI panelu. Tento panel bude sloužit k ovládání jednotky a k zobrazení stavu, ve kterém se právě nachází. Podmínkou je, aby na sobě byly aplikace pro HMI a program pro řízení jednotky nezávislé a vzájemně do sebe nezasahovaly.

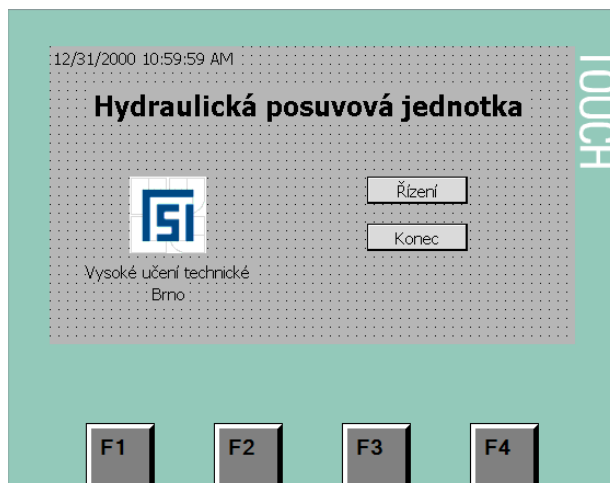
6.1.8 Vypracování příkladu č. 4

Protože jednotku budeme ovládat pomocí HMI panelu, je nutné si upravit tabulku symbolických proměnných. Změníme tedy adresy u proměnných START a STOP na M1.0 a M1.1. Nyní nejsou proměnné vázány na fyzické vstupy automatu, ale odkazují do bitové paměti M, kterou budeme nastavovat pomocí tlačítek na HMI panelu.

	Name	Data type	Address	Retain	Visible..	Acces...	Comment
1	K1	Bool	%I0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Poloha K1
2	K2	Bool	%I0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Poloha K1
3	K3	Bool	%I0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Poloha K1
4	K4	Bool	%I0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Poloha K1
5	EM1	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Elektromagnet EM1
6	EM2	Bool	%Q0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Elektromagnet EM2
7	EM3	Bool	%Q0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Elektromagnet EMB
8	Start	Bool	%M1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Tlačítko START
9	Stop	Bool	%M1.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Tlačítko STOP
10	Vystupy	Byte	%QB0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Proměnná Vystupy
11	Jedu	Bool	%M1.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Příznak pohybu
12	Pamet	Byte	%MBO	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Paměť pro uložení stavu výstupů

Obr. 31: Tabulka symbolických proměnných

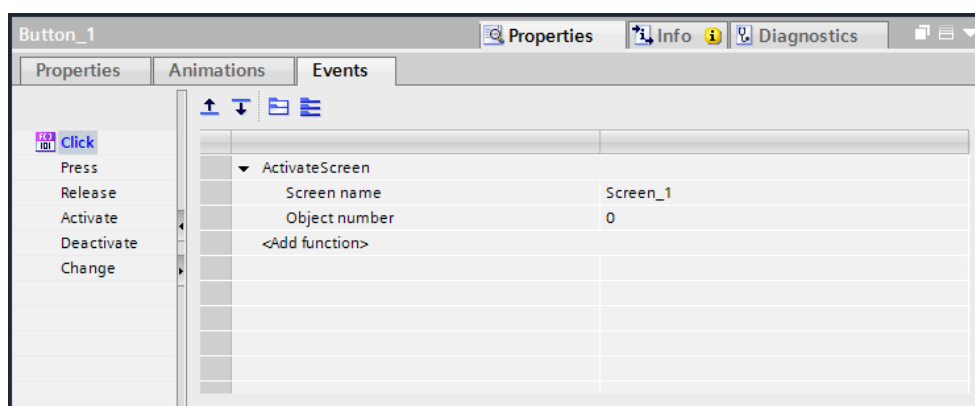
Pro HMI panel si připravíme dvě obrazovky. Jedna obrazovka bude sloužit jako menu, druhá potom k ovládání a zobrazení stavu, ve kterém se jednotka nachází. Rozložení obrazovek můžete vidět na obr. 32 a obr. 34.



Obr. 32: Obrazovka sloužící jako menu

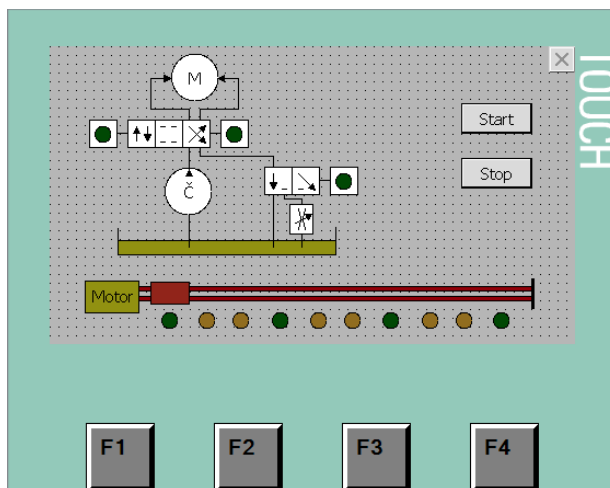
Obrazovka sloužící jako menu obsahuje dvě tlačítka. Tlačítko „Řízení“ slouží k přepnutí na obrazovku s ovládáním jednotky, tlačítko „Konec“ potom k ukončení HMI aplikace.

Nyní si popíšeme jejich nastavení. Tlačítko „Řízení“ můžeme nastavit dvěma způsoby. Prvním způsobem je přetažení obrazovky z okna „Projekt tree“ na obrazovku menu. Tím se nám automaticky vytvoří tlačítko odkazující na danou obrazovku. Druhým způsobem je vložení tlačítka a následně ruční nastavení události, která se má vykonat při určité činnosti. Tuto událost nastavíme ve vlastnostech tlačítka, pod záložkou „Events“. Po levé straně si vybereme při jaké činnosti se má událost provést. V našem případě vybereme při kliknutí. Jako další věc musíme nastavit jaká událost se má provést. V tabulce přidáme novou funkci „ActivateScreen“ sloužící k aktivaci obrazovky. Tuto funkci najdeme v záložce „Screen“. Jako parametr funkce „Screen name“ nastavíme obrazovku, na kterou se po stisknutí tlačítka chceme přepnout.



Obr. 33: Nastavení události po stisknutí tlačítka "Řízení"

V případě tlačítka „Konec“ postupujeme stejně pouze vybereme jinou funkci, která bude sloužit pro vypnutí programu. Tato funkce se jmenuje „StopRuntime“ a nalezneme ji v záložce „Other functions“.

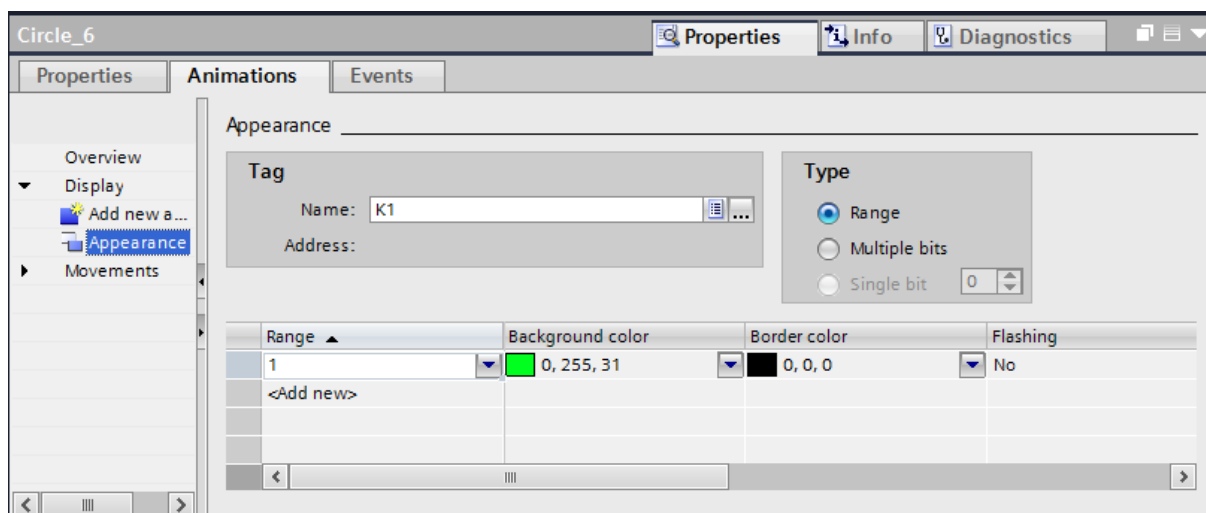


Obr. 34: Obrazovka sloužící k ovládání a zobrazení stavu jednotky

Druhá obrazovka obsahuje schéma hydraulické posuvové jednotky, na kterém se zobrazuje vykonávaná činnost a dále dvě tlačítka pro ovládání jednotky. Tlačítka jsou propojena s příslušnými proměnnými START a STOP nacházející se v bitové paměti M. Způsob propojení je vysvětlen v kapitole 5.3. Schéma je nakresleno pomocí základních tvarů přímo ve vývojovém prostředí TIA Portal. Probíhající stav je signalizován stejně jako u fyzického modelu pomocí „LED diod“. Ty jsou vytvořeny pomocí kružnic vyplněných tmavě zelenou, v případě mezipoloh žlutou barvou. Indikace

stavu je vytvořena pomocí animace. Způsob animace objektů je stejný pro všechny kružnice, proto si zde popíšeme pouze vytvoření animace pro polohu K1.

Nejprve si označíme kružnici znázorňující polohu K1. V okně nacházejícím se ve spodní části obrazovky otevřeme postupně záložky „Properties“ a „Animation“. Zde si vytvoříme novou animaci typu „Appearance“. Jedná se o animaci, která nám změní barvu vybraného objektu při dosažení zadaných podmínek. Protože očekáváme reakci na změnu snímače K1, vybereme jako „Tag“ proměnnou K1. Typ animace ponecháme nastavený na „Range“. Do tabulky přidáme nový řádek, kde buňku „Range“ nastavíme na hodnotu 1 a „Background color“ nastavíme na světle zelenou barvu. Tímto jsme nastavili, že při dosažení logické hodnoty 1 u proměnné K1, se změní barva výplně kružnice na světle zelenou.



Obr. 35: Nastavení animace u snímače polohy K1

Protože mezipolohy znázorněné žlutou LED diodou nejsou indikovány žádnými snímači polohy, je nutné si vytvořit program, pomocí kterého bude automat schopen rozeznat, na kterém místě se suport právě nachází. Tento program bude vytvořen v novém programovém bloku a nebude žádným způsobem zasahovat do již vytvořeného programu pro řízení jednotky.

Princip tohoto programu spočívá ve vytvoření časově spouštěného programového bloku, který se bude spouštět s periodou, která se rovná času, potřebného ke skoku z jedné polohy suportu do druhé polohy a to při rychloposuvu. V případě pracovního zpomaleného posuvu budeme uvažovat, že čas skoku mezi jednotlivými polohami bude roven násobku této základní periody. V programu se bude nacházet pomocná proměnná „i“ typu integer, která se bude zvětšovat při každém spuštění programu o 1 a to tehdy, když bude jednotka v pohybu, což poznáme podle příznaku pohybu „Jedu“. Pomocná proměnná „i“ je tedy násobkem periody, kterou je program spouštěn. Protože víme v jakých násobcích této periody jsou aktivovány jednotlivé mezipolohy, můžeme pomocí proměnné „i“ určit v jaké poloze se suport právě nachází.

Pro následující program si vytvoříme novou tabulku symbolických proměnných. Tato tabulka bude obsahovat pomocnou proměnnou „i“ typu integer, která bude sloužit k určení násobku periody spouštění programu. Dále obsahuje mezipolohy P1 až P6, které jsou na hydraulické posuvové jednotce znázorněny žlutou LED diodou. Poslední proměnnou je proměnná „Mezipolohy“, která slouží k vyresetování všech mezipoloh.

	Name	Data type	Address	Retain	Visible..	Acces...	Comment
1	i	Int	%MW2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Pomocná proměnná
2	P1	Bool	%M4.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Mezipoloha 1
3	P2	Bool	%M4.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Mezipoloha 2
4	P3	Bool	%M4.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Mezipoloha 3
5	P4	Bool	%M4.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Mezipoloha 4
6	P5	Bool	%M4.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Mezipoloha 5
7	P6	Bool	%M4.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Mezipoloha 6
8	Mezipolohy	Byte	%MB4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Všechny mezipolohy

Obr. 36: Tabulka symbolických proměnných

```

IF "Jedu" THEN // Pokud je nastaven příznak pohybu
  "i" := "i" + 1; // Zvyš proměnnou i o 1
END_IF; // Konec podmínky IF

IF "K1" OR "K4" THEN // Jestliže je aktivován snímač K1 nebo K4
  "i" := 0; // Vynuluj proměnnou i
  "P6" := false; // Vypni mezipolohu P6
  "P1" := false; // Vypni mezipolohu P1
END_IF; // Konec podmínky IF

CASE "i" OF // Jestliže je i rovno
1: IF "EM1" THEN // 1 a pokud se suport pohybuje vpravo
  "P1" := true; // Aktivuj mezipolohu P1
  "P2" := false; // Vypni mezipolohu P2
END_IF; // Konec podmínky IF

  IF "EM2" THEN // Pokud se suport pohybuje vlevo
  "P6" := true; // Aktivuj mezipolohu P6
  "P5" := false; // Vypni mezipolohu P5
END_IF; // Konec podmínky IF

3: IF "EM1" THEN // 3 a pokud se suport pohybuje vpravo
  "P2" := true; // Aktivuj mezipolohu P2
  "P1" := false; // Vypni mezipolohu P1
END_IF; // Konec podmínky IF

  IF "EM2" THEN // Pokud se suport pohybuje vlevo
  "P5" := true; // Aktivuj mezipolohu P5
  "P6" := false; // Vypni mezipolohu P6
END_IF; // Konec podmínky IF

11: IF "EM1" THEN // 11 a pokud se suport pohybuje vpravo
  "P3" := true; // Aktivuj mezipolohu P3
  "P2" := false; // Vypni mezipolohu P2
END_IF; // Konec podmínky IF

  IF "EM2" THEN // Pokud se suport pohybuje vlevo
  "P4" := true; // Aktivuj mezipolohu P4
  "P5" := false; // Vypni mezipolohu P5
END_IF; // Konec podmínky IF

```

```
27: IF "EM1" THEN // 27 a pokud se suport pohybuje vpravo
    "P4" := true; // Aktivuj mezipolohu P4
    "P3" := false; // Vypni mezipolohu P3
END_IF; // Konec podmínky IF

IF "EM2" THEN // Pokud se suport pohybuje vlevo
    "P3" := true; // Aktivuj mezipolohu P3
    "P4" := false; // Vypni mezipolohu P4
END_IF; // Konec podmínky IF

50: IF "EM1" THEN // 50 a pokud se suport pohybuje vpravo
    "P5" := true; // Aktivuj mezipolohu P5
    "P4" := false; // Vypni mezipolohu P4
END_IF; // Konec podmínky IF

IF "EM2" THEN // Pokud se suport pohybuje vlevo
    "P2" := true; // Aktivuj mezipolohu P2
    "P3" := false; // Vypni mezipolohu P3
END_IF; // Konec podmínky IF

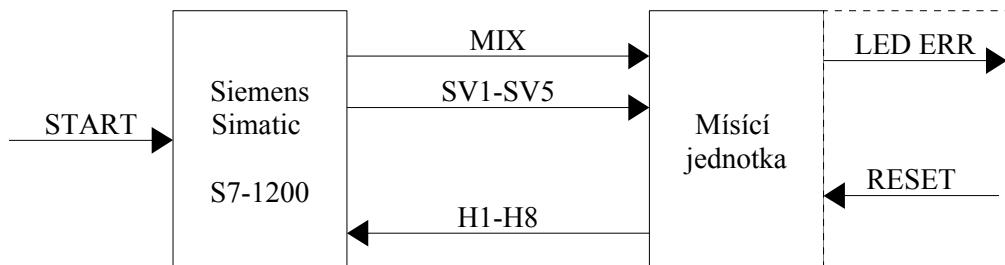
52: IF "EM1" THEN // 52 a pokud se suport pohybuje vpravo
    "P6" := true; // Aktivuj mezipolohu P6
    "P5" := false; // Vypni mezipolohu P5
END_IF; // Konec podmínky IF

IF "EM2" THEN // Pokud se suport pohybuje vlevo
    "P1" := true; // Aktivuj mezipolohu P1
    "P2" := false; // Vypni mezipolohu P2
END_IF; // Konec podmínky IF

ELSE // Jinak
    IF "K1" OR "K2" OR "K3" OR "K4" THEN // Pokud je aktivován kterýkoliv
        "Mezipolohy" := 0; // snímač polohy
    END_IF; // Vypnou se všechny mezipolohy
END_CASE; // Konec podmínky IF
// Konec podmínky CASE
```

6.2 Mísící jednotka

Mísící jednotka bude ovládána pomocí tlačítka START. Schéma zapojení můžete vidět na obrázku obr. 37. Detailnější popis jednotky se nachází v kapitole 1.2.



Obr. 37: Schéma propojení PLC s EDU-mod modelem

V následující tabulce se nacházejí symbolické proměnné přiřazené ke konkrétním adresám programovatelného automatu

Symbolická proměnná	Adresa	Popis
H1	I0.7	Snímač hladiny H1
H2	I0.6	Snímač hladiny H2
H3	I0.5	Snímač hladiny H3
H4	I0.4	Snímač hladiny H4
H5	I0.3	Snímač hladiny H5
H6	I0.0	Snímač hladiny H6
H7	I0.1	Snímač hladiny H7
H8	I0.2	Snímač hladiny H8
START	I1.0	Tlačítko START
SV1	Q0.0	Ovládání ventilu SV1
SV2	Q0.1	Ovládání ventilu SV2
SV3	Q0.2	Ovládání ventilu SV3
SV4	Q0.5	Ovládání ventilu SV4
SV5	Q0.4	Ovládání ventilu SV5
MIX	Q0.6	Ovládání mixéru MIX

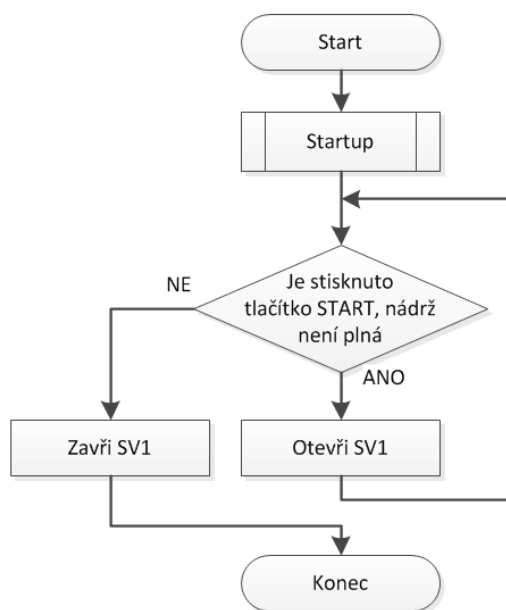
Tab. 6: Adresy symbolických proměnných

6.2.1 Zadání příkladu č. 1

Vytvořte program, který po stisku tlačítka START naplní tank 1 na 50 procent objemu

6.2.2 Vypracování příkladu č. 1

Aplikace se skládá ze dvou programových bloků a to Startup a Main. Programový blok Startup se vykoná pouze při prvním programovém cyklu a obsahuje počáteční inicializaci. V programovém bloku Main se nachází hlavní program. Pokud je stisknuto tlačítko START a snímač hladiny H2 není aktivní, otevře se ventil SV1. V opačném případě se ventil zavře a program skončí.



Obr. 38: Vývojový diagram

Do tabulky symbolických proměnných si vložíme snímače hladiny H1, H2 a H3, ventil SV1 a tlačítko START.

	Name	Tag table	Data type	Address	Retain	Visible..	Acces...	Comment
1	H1	mísící jednotka	Bool	%I0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Snímač hladiny H1
2	H2	mísící jednotka	Bool	%I0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Snímač hladiny H2
3	H3	mísící jednotka	Bool	%I0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Snímač hladiny H3
4	SV1	mísící jednotka	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Ventil SV1
5	START	mísící jednotka	Bool	%I1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Tlačítko START

Obr. 39: Tabulka symbolických proměnných

Zdrojový kód programového bloku Startup

```
"SV1" := false; // Při prvním programovém cyklu zavři ventil SV1
```

Zdrojový kód programového bloku Main

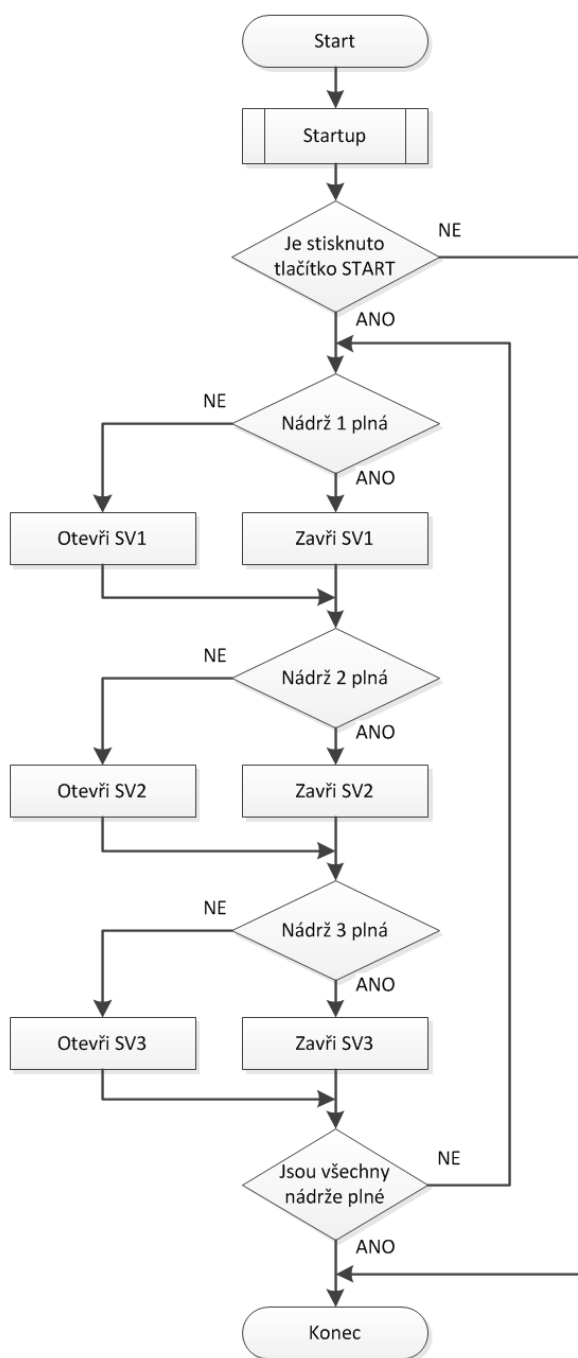
```
IF "Start" AND NOT "H2" THEN // Pokud je stisknuté tlačítko Start a není
    // dosaženo hladiny H2
    "SV1" := true; // Otevři ventil SV1
ELSE // Jinak
    "SV1" := false; // Zavři ventil SV1
END_IF; // Konec podmínky IF
```

6.2.3 Zadání příkladu č. 2

Rozšířte zadání z prvního příkladu tak, aby se po stisknutí tlačítka START naplnil tank 1 na 50%, tank 2 na 100% a tank 3 na 50%.

6.2.4 Vypracování příkladu č. 2

Příklad se skládá ze dvou programových bloků Startup a Main. Stejně jako v předchozím příkladě slouží programový blok Startup pro vyresetování výstupů. V programovém bloku Main se opět nachází hlavní program. Pokud zmačkneme tlačítko START a tanky nejsou naplněny, otevřou se ventily, čímž se začnou tanky plnit. Pokud hladina dosáhne zadané úrovně příslušný ventil se vypne.



Obr. 40: Vývojový diagram

Tabulku symbolických proměnných si rozšíříme o snímače hladiny H4 – H8, ventily SV2 a SV3, a proměnnou Vystupy. Tato proměnná má stejný účel jako v hydraulické posuvné jednotce viz. kap. 6.1.4.

	Name	Tag table	Data type	Address	Retain	Visible..	Acces...	Comment
1	H1	mísící jednotka	Bool	%I0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Snímač hladiny H1
2	H2	mísící jednotka	Bool	%I0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Snímač hladiny H2
3	H3	mísící jednotka	Bool	%I0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Snímač hladiny H3
4	H4	mísící jednotka	Bool	%I0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Snímač hladiny H4
5	H5	mísící jednotka	Bool	%I0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Snímač hladiny H5
6	H6	mísící jednotka	Bool	%I0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Snímač hladiny H6
7	H7	mísící jednotka	Bool	%I0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Snímač hladiny H7
8	H8	mísící jednotka	Bool	%I0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Snímač hladiny H8
9	SV1	mísící jednotka	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Ventil SV1
10	SV2	mísící jednotka	Bool	%Q0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Ventil SV2
11	SV3	mísící jednotka	Bool	%Q0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Ventil SV3
12	START	mísící jednotka	Bool	%I1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Tlačítko START
13	Vystupy	mísící jednotka	Byte	%QB0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Proměnná Vystupy

Obr. 41: Tabulka symbolických proměnných

Zdrojový kód programového bloku Startup

```
"Vystupy" := 0;           // Při prvním programovém cyklu uzavře všechny ventily
```

Zdrojový kód programového bloku Main

```
IF "Start" THEN           // Pokud je stisknuto tlačítko Start
  IF "H2" THEN             // Pokud je tank 1 plný
    "SV1" := false;       // Zavři ventil SV1
  ELSE                     // Jinak
    "SV1" := true;        // Otevři ventil SV1
  END_IF;                 // Konec podmínky IF

  IF "H4" THEN             // Pokud je tank 2 plný
    "SV2" := false;       // Zavři ventil SV2
  ELSE                     // Jinak
    "SV2" := true;        // Otevři ventil SV2
  END_IF;                 // Konec podmínky IF

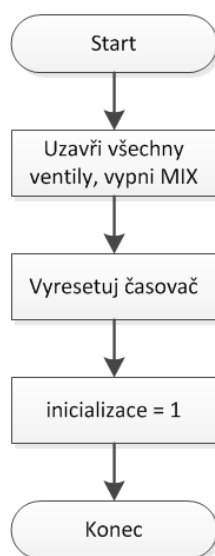
  IF "H7" THEN             // Pokud je tank 3 plný
    "SV3" := false;       // Zavři ventil SV3
  ELSE                     // Jinak
    "SV3" := true;        // Otevři ventil SV3
  END_IF;                 // Konec podmínky IF
END_IF;                   // Konec podmínky IF
```

6.2.5 Zadání příkladu č. 3

Rozšířte zadání ze druhého příkladu tak, aby se po stisknutí tlačítka START naplnil tank 1 na 50%, tank 2 na 100% a tank 3 na 50%. Po dosažení požadovaných hladin se otevře přepouštěcí ventil SV4 a zapne mixer MIX. Po vyprázdnění všech tanků se uzavře ventil SV4, vypne mixer MIX a otevře se přepouštěcí ventil SV5 po dobu 15 sekund.

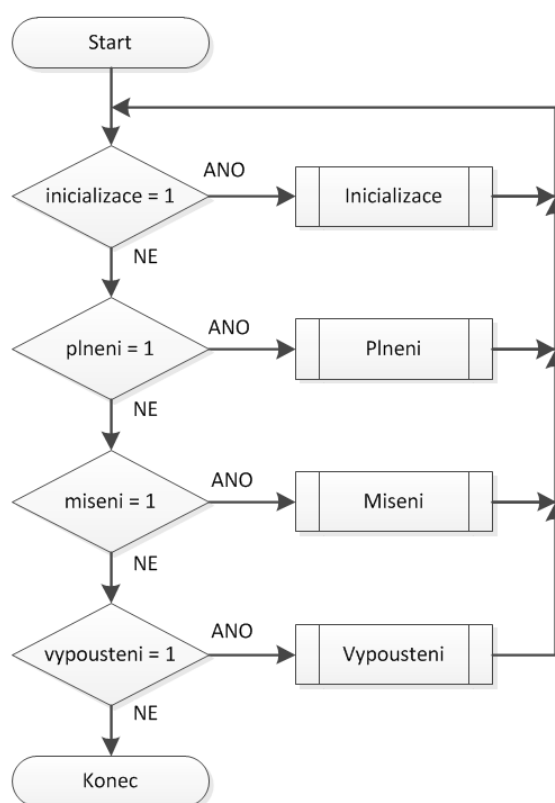
6.2.6 Vypracování příkladu č. 3

Vhodným řešením je rozdělit provoz do několika bloků. V našem případě se bude jednat o programové bloky Startup, Main a funkce Inicializace, Plnění, Mísení a Vypouštění, které se budou starat o samotné fáze procesu. Pro ukázkou si v následujícím příkladě ukážeme práci s data bloky, do kterých budeme ukádat pomocné proměnné. Funkce starající se o jednotlivé fáze budeme volat v programovém bloku Main a to na základě pomocných proměnných „inicializace“, „plnění“, „mísení“ a „vypouštění“. Při prvním spuštění programu se nejříve provede programový blok Startup, ve kterém vyresetujeme výstupy, časovač a proměnnou „inicializace“ nastavíme na logickou hodnotu 1.



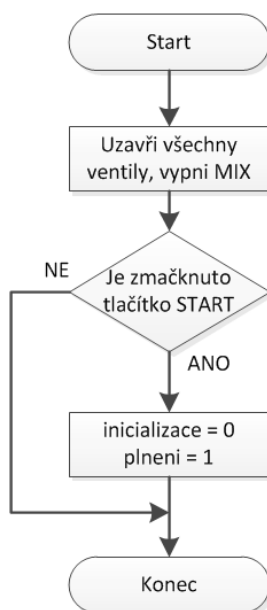
Obr. 42: Vývojový diagram programového bloku Startup

V programovém bloku Main se nyní pomocí pomocných proměnných vyhodnotí, která funkce se má zavolat.



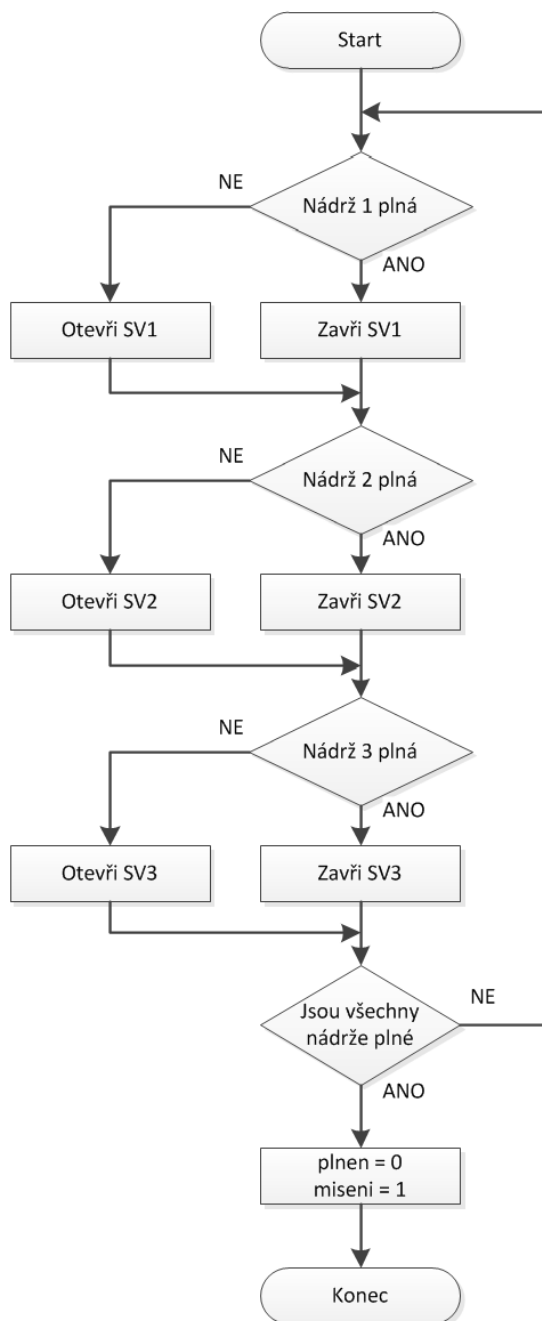
Obr. 43: Vývojový diagram programového bloku Main

Protože jediná proměnná nastavená na logickou hodnotu 1 je „inicializace“ zavolá se funkce Inicializace. V této funkci se vyresetují všechny výstupy. Dále se zde nachází podmínka testující stisk tlačítka START. Pokud je tlačítko stisknuto, proměnná „inicializace“ se vyresetuje a proměnná „plneni“ se nastaví na logickou hodnotu 1.



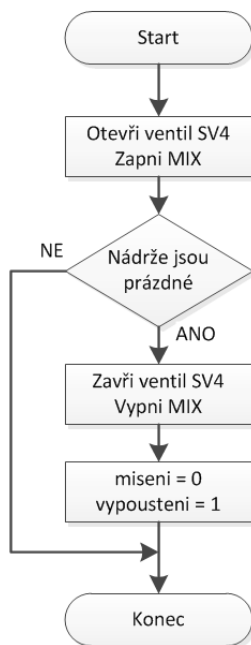
Obr. 44: Vývojový diagram funkce Inicializace

Další spuštěnou funkcí je Plnění. Tato funkce se stará o naplnění tanků na požadovanou výšku hladiny. Po dosažení potřebných hladin se vyresetuje proměnná „plnení“ a proměnná „mísení“ se nastaví na logickou hodnotu 1.



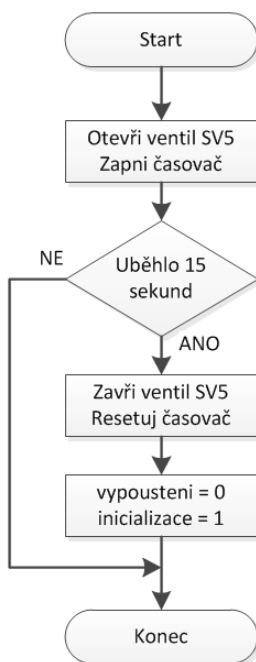
Obr. 45: Vývojový diagram funkce Plnění

Funkce Miseni se stará o přepuštění tanků do mísící nádrže pomocí otevření ventilu SV4. Zároveň s otevřením ventilu se zapne mixer. Pokud se nádrže vyprázdní uzavře se ventil SV4, vypne se mixer a nastaví se pomocné proměnné „miseni“ na logickou hodnotu 0 a „vypousteni“ na logickou hodnotu 1.



Obr. 46: Vývojový diagram funkce Miseni

Funkce Vypousteni vypustí směs s mísící nádobou mimo mísící jednotku pomocí ventilu SV5. Tento ventil je otevřen po dobu 15 sekund. Po uplynutí stanovené doby se uzavře ventil SV5 a časovač se vyresetuje. Posledním krokem je nastavení proměnných „vypousteni“ na logickou hodnotu 0 a „inicializace“ na log hodnotu 1.



Obr. 47: Vývojový diagram funkce Vypousteni

Do tabulky symbolických proměnných si přidáme ventily SV4, SV5 a mixer MIX.

	Name	Tag table	Data type	Address	Retain	Visible..	Acces...	Comment
1	H1	mísící jednotka	Bool	%I0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Snímač hladiny H1
2	H2	mísící jednotka	Bool	%I0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Snímač hladiny H2
3	H3	mísící jednotka	Bool	%I0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Snímač hladiny H3
4	H4	mísící jednotka	Bool	%I0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Snímač hladiny H4
5	H5	mísící jednotka	Bool	%I0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Snímač hladiny H5
6	H6	mísící jednotka	Bool	%I0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Snímač hladiny H6
7	H7	mísící jednotka	Bool	%I0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Snímač hladiny H7
8	H8	mísící jednotka	Bool	%I0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Snímač hladiny H8
9	SV1	mísící jednotka	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Ventil SV1
10	SV2	mísící jednotka	Bool	%Q0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Ventil SV2
11	SV3	mísící jednotka	Bool	%Q0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Ventil SV3
12	SV4	mísící jednotka	Bool	%Q0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Ventil SV4
13	SV5	mísící jednotka	Bool	%Q0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Ventil SV5
14	MIX	mísící jednotka	Bool	%Q0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Mixer MIX
15	START	mísící jednotka	Bool	%I1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Tlačítko START
16	Vystupy	mísící jednotka	Byte	%QB0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Proměnná Vystupy

Obr. 48: Tabulka symbolických proměnných

Dále si vytvoříme datový blok. Klikneme tedy na tlačítko „add new block“, v novém okně vybereme „Data block“ a položku „Type“ necháme nastavenou na „Global DB“. Potvrzení tlačítkem „OK“ se nám vytvoří globální datový blok, ke kterému můžeme přistupovat z kteréhokoliv programového bloku, funkčního bloku nebo funkce. Do datového bloku si přidáme proměnné „inicializace“, „plnění“, „mísení“, „vypouštění“, jejichž funkce jsme si již popsali. Poslední přidanou proměnnou bude „timer“. Tuto proměnnou bude využívat časovač. Pokud časovač odpočítá zadaný čas, nastaví proměnnou na logickou hodnotu 1, v opačném případě na 0. K proměnným z datového bloku budeme přistupovat jako k "navez_datoveho_bloku".promenna.

	Name	Data type	Start value	Retain	Visible in ...	Comment
1	Static					
2	inicializace	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
3	plneni	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
4	miseni	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
5	vypousteni	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
6	timer	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	

Obr. 49: Obsah datového bloku

Zdrojový kód programového bloku Startup

```
"Vystupy" := 0; // Při prvním programovém cyklu vyresetuj všechny výstupy
"DB1".inicializace := 0; // Nastav proměnnou inicializace

"Timer".TON{ // Spust časovač zpožděného zapnutí Timer
  IN := false, // Vyresetuj časovač
  PT := 15000 // Povinná proměnná, určuje odpočítávaný čas v ms
}; // Konec časovače
```

V tomto příkladě je použit časovač TON. Jedná se o časovač zpožděného zapnutí. Tlačítkem „add new block“ si vytvoříme instanci datového bloku typu IEC_TIMER, který si pojmenujem Timer. Časovač vyresetujeme pokud je při zapnutí časovače stav povolovacího vstupu (IN) roven logické hodnotě 0.

Zdrojový kód funkce Vypousteni

```

"SV5" := true; // Otevři ventil SV5

"Timer".TON{ // Spust' časovač zpožděného zapnutí Timer
  IN := true, // Spust' odpočet
  PT := 15000 // Počítej do 15s
  Q => "DB1".timer // Pokud uplynulo 15s nastav proměnnou timer
}; // Konec časovače

IF "DB1".timer THEN // Pokud časovač odpočítal 15s
  "SV5" := false; // Zavři ventil SV5

  "Timer".TON{ // Spust' časovač zpožděného zapnutí Timer
    IN := false, // Vyresetuj časovač
    PT := 15000 // Povinná proměnná, určuje odpočítávaný čas v ms
  }; // Konec časovače

  "DB1".vypousteni := false; // Nastav proměnnou vypousteni
  "DB1".inicializace := true; // Nastav proměnnou inicializace
END_IF; // Konec podmínky IF

```

6.2.7 Zadání příkladu č. 4

Rozšířte zadání ze třetího příkladu přidáním HMI panelu. Tento panel bude sloužit k ovládání jednotky a k zobrazení stavu, ve kterém se právě nachází. Podmínkou je aby na sobě byly aplikace pro HMI a program pro řízení jednotky nezávislé a vzájemně do sebe nezasahovaly.

6.2.8 Vypracování příkladu č. 4

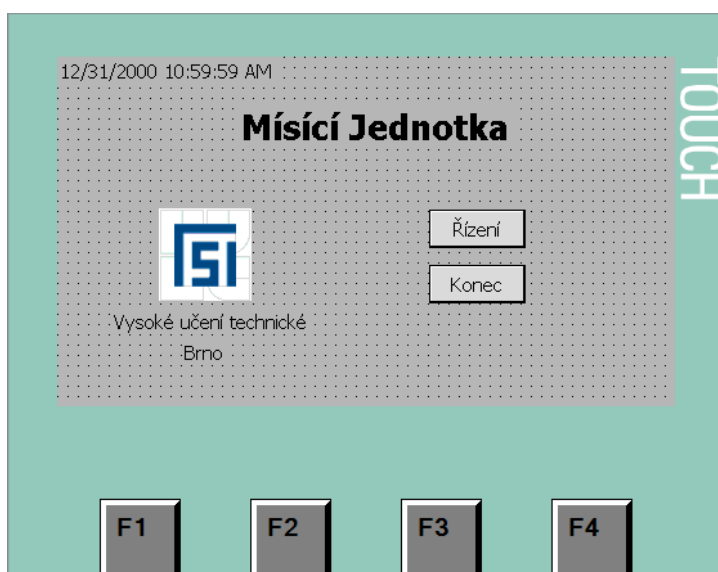
Mísící jednotka bude na rozdíl od hydraulické posuvové jednotky ovládána pouze tlačítkem START. Protože v této úloze ukládáme pomocné proměnné do datového bloku, vymažeme tlačítko START z tabulky symbolických proměnných a vložíme ho do datového bloku.

	Name	Data type	Start value	Retain	Visible in ...	Comment
1	Static					
2	inicializace	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
3	plneni	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
4	miseni	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
5	vypousteni	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
6	timer	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
7	START	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	

Obr. 50: Datový blok

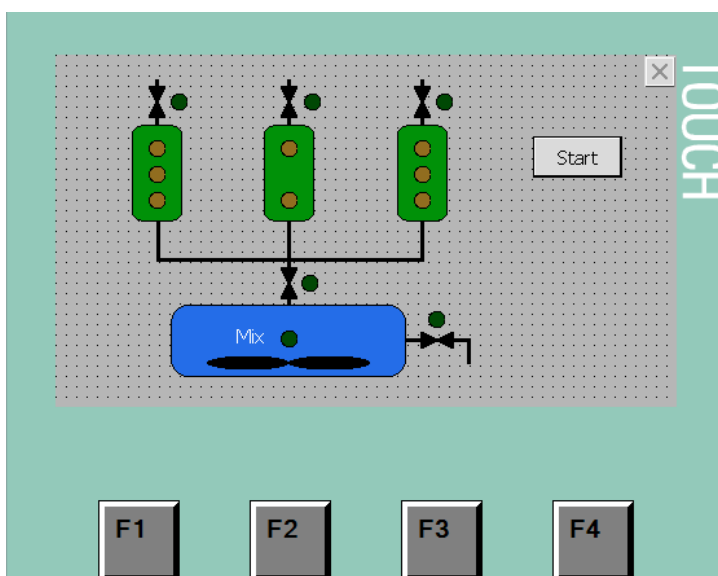
Stejně jako u úlohy hydraulické posuvové jednotky vytvoříme dvě obrazovky. Jedna bude sloužit jako menu a druhá k ovládání a zobrazení stavu jednotky.

Obrazovka sloužící jako menu (obr. 51) obsahuje tlačítka „Řízení“ a „Konec“. Tyto tlačítka jsou nastavena stejným způsobem jako v případě hydraulické posuvové jednotky v kapitole 6.1.8.



Obr. 51: Obrazovka sloužící jako menu

Druhá obrazovka (obr. 52) obsahuje opět schéma jednotky, obsahující „LED diody“ zobrazující stav. Ty jsou nastaveny stejným způsobem jako v případě hydraulické posuvové jednotky v kapitole 6.1.8 a to pomocí animace. Dále se zde nachází tlačítko START, které je propojeno s proměnnou START nacházející se v datovém bloku.



Obr. 52: Obrazovka sloužící k ovládní a zobrazení stavu jednotky

ZÁVĚR

Cílem této bakalářské práce bylo seznámit se s modely soustav technologických procesů, programovatelným automatem Siemens Simatic S7-1200 a vývojovým prostředím TIA Portal v11. Dále pro tyto modely vytvořit programy pro řízení a následně ověřit jejich funkčnost pomocí výše uvedeného programovatelného automatu. Úlohy měly být řešeny pomocí programovacího jazyka strukturovaného textu, u firmy Siemens označeného jako SCL. Dále byla práce rozšířena o začlenění HMI panelu, který slouží jako ovládací a informační prvek.

V první kapitole byly popsány funkce, vstupy a výstupy modelů soustav technologických procesů EDU-mod. Jedná se o model hydraulické posuvové jednotky a mísící jednotky.

Ve druhé kapitole byl představen programovatelný automat Siemens Simatic S7-1200, pomocí kterého jsou modely EDU-mod řízeny. Tato kapitola obsahuje technické parametry a dále popisuje možná rozšíření.

Ve třetí kapitole byl popsán HMI panel Siemens KTP400 Basic color a jeho technické parametry.

Čtvrtá kapitola se zabývá návrhem a realizací desek plošných spojů, pomocí kterých je umožněno bezšroubové připojení EDU-mod modelů k programovatelnému automatu.

Pátá kapitola obsahuje popis vývojového prostředí TIA Portal v11. V této části je popsána instalace samotného programu a jeho updatů včetně aktualizace hardwarového katalogu. Dále je popsána tvorba nového projektu s nastavením všech zařízení a také tvorba jednoduchého programu a jeho nahrání do programovatelného automatu.

V poslední kapitole byly popsány řešení pro zadané EDU-mod modely. Tento popis obsahuje vývojový diagram, symbolickou tabulku a okomentovaný kód programu dané úlohy. Protože tato bakalářská práce byla psána jako podpora k výuce programovatelných automatů, rozhodl jsem se pro lepší pochopení vytvořit ke každé úloze čtyři zadání, které se svou obtížností stupňují od nejlehčího po nejtěžší, přičemž úkolem posledního zadání je rozšíření předchozího zadání o přidání HMI panelu.

SEZNAM POUŽITÉ LITERATURY

- [1] KOHOUT, Luděk. *Edumat.cz - Autorizované kurzy Teco a.s. , pomůcky do odborných učeben a laboratoří* [online]. [cit. 2013-03-14]. Dostupné z :
<<http://www.edumat.cz/produkty.php?produkt=edumod>>
- [2] KOHOUT, Luděk. *Edumat.cz - Autorizované kurzy Teco a.s. , pomůcky do odborných učeben a laboratoří* [online]. [cit. 2013-03-14]. Dostupné z :
<<http://www.edumat.cz/produkty.php?produkt=support>>
- [3] KOHOUT, Luděk. *Edumat.cz - Autorizované kurzy Teco a.s. , pomůcky do odborných učeben a laboratoří* [online]. [cit. 2013-03-14]. Dostupné z :
<<http://www.edumat.cz/produkty.php?produkt=mixer>>
- [4] *S7-1200 CPUs – PLC - Siemens* [online]. [cit. 2013-03-28]. Dostupné z :
<<http://www.automation.siemens.com/mcms/programmable-logic-controller/en/simatic-s7-controller/s7-1200/cpu/Pages/Default.aspx>>
- [5] *Siemens SIMATIC CPU 1214C DC/DC/DC 6ES7214-1HE30-0XB0 20.4 - 28.8 V/DC im Conrad Online Shop | 197494* [online]. [cit. 2013-03-28]. Dostupné z
<http://www.conrad.at/medias/global/ce/1000_1999/1900/1970/1974/197468_LB_00_FB.EPS_1000.jpg>
- [6] *Siemens S7-1200 Programmable controller System Manual* [PDF dokument]. Duben 2012 [cit. 2013-03-28]. Dostupné z :
<http://stest1.etnetera.cz/ad/current/content/data_files/automatizacni_systemy/mikrosystemy/simatic_s71200/manual_s7-1200_2012-04_en.pdf>
- [7] *Prospekt Simatic S7-1200* [PDF dokument]. [cit. 2013-03-28]. Dostupné z :
<http://stest1.etnetera.cz/ad/current/content/data_files/automatizacni_systemy/mikrosystemy/simatic_s71200/brochure_s7-1200_2012_cz.pdf>
- [8] *Simatic HMI Panels, Operator panels to suit all demands* [PDF dokument]. [cit. 2013-04-02]. Dostupné z :
<http://www.automation.siemens.com/salesmaterial-as/brochure/en/brochure_panels_en.pdf>
- [9] *Basic panely – Industry Automation & Drive Technologies - Siemens* [online]. [cit. 2013-04-02]. Dostupné z :
<<http://stest1.etnetera.cz/ad/current/index.php?vw=0&ctxnh=2bc5272a29&ctxp=home>>
- [10] *Simatic HMI KTP400 Basic color* [online]. [cit. 2013-04-02]. Dostupné z :
<http://www.automatyka.siemens.pl/Images_industry/Images_HAIDT/ktp400-basic-color-600.jpg>
- [11] *Totally Integrated Automation Portal – Industry Automation & Drive Technologies - Siemens* [online]. [cit. 2013-04-09]. Dostupné z :
<<http://stest1.etnetera.cz/ad/current/index.php?vw=0&ctxnh=2416f2e791&ctxp=home>>
- [12] *Siemens Industry Online Support* [online]. [cit. 2013-04-09]. Dostupné z :
<<http://support.automation.siemens.com/WW/llisapi.dll?func=cslib.csinfo&lang=en&objid=58112582&caller=view>>

- [13] MARADA, Tomáš. *Programovatelné automaty, časový chod programu, norma IEC-61131, jazyky* [PDF dokument]. Říjen 2011 [cit. 2013-04-09]. Dostupné z : <<https://www.vutbr.cz/elearning/mod/resource/view.php?id=196170>>