

Jihočeská univerzita
Přírodovědecká fakulta
Ústav fyziky a biofyziky

Vytvoření sestavy robotických systémů
pro laboratorní úlohy

Autor: Svatoslav Tošner
Vedoucí práce: Ing. Ladislav Ptáček

České Budějovice 2016

Bibliografické údaje:

Tošner S. ,2016: Vytvoření sestavy robotických systémů pro laboratorní úlohy [Creation of assembly of robotic systems for laboratory, Bc. Thesis, in Czech] – 38p, Faculty of Science, The University of South Bohemia, České Budějovice, Czech Republic

Abstrakt:

Tato bakalářská práce se zabývá řízením robotické sestavy pomocí programovatelného logického automatu neboli PLC (z anglického Programmable Logic Controller).

V první části je popsána teorie, vývoj a historie automatizace. Dále jsou zde popsány veškeré použité komponenty.

Druhá část práce je zaměřena na sestavení a zprovoznění dané soustavy (řídící jednotky Simatic, dotykového panelu, robotické ruky a 2 pracovních linek).

Pomocí této soustavy jsou demonstrovány 3 laboratorní úlohy včetně klíčů k jejich zrealizování.

Klíčová slova:

PLC, automatizace, řízení

Summary:

This bachelor thesis deals with control robotic kit using Programmable Logic Controller (PLC).

The first part describes both the theory and history of an automation and PLC.

The second part of this thesis is focused on the completion and commissioning of the kit (programmable logic controller, touch panel, robotic arm, and two work stations).

The last part deals with three laboratory assignments to demonstrate the system. Each task also includes a teacher manual (so called “key”) for completing the assignment.

Keywords:

PLC, automation, control

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Trhových Svinech 18.4.2016

.....
Svatoslav Tošner

Poděkování

Tímto bych chtěl poděkovat panu Ing. Ladislavu Ptáčkovi za vedení mé bakalářské práce a mému otci, Svatoslavu Tošnerovi, za pomoc s mechanickým upevněním sestavy.

Obsah

1 ÚVOD	1
1 AUTOMATIZACE, PRAKTICKÉ VYUŽITÍ	2
1.1 Od mechanizace k automatizace	2
1.2 Automatizace	2
1.3 Řízení	3
1.3.1 Ovládání	3
1.3.2 Regulace.....	3
1.4 Prostředky automatizace	5
1.4.1 Snímače	5
1.4.2 Regulátory	6
1.4.3 Akční členy	9
1.5 Pojem PLC	9
1.5.1 Vznik a vývoj PLC.....	9
1.5.2 Struktura PLC	10
1.6 Programování PLC	13
2 POPIS KOMPONENTŮ	19
2.1 Rameno	19
2.2 Výrobní linka	21
2.3 Obráběcí stanice	23
2.4 HMI panel	24
2.5 PLC	24
3 ÚLOHY	26
3.1 Úloha 1 – zadání	26
3.1.1 Úloha 1 - řešení	26

3.2	Úloha 2 – zadání	30
3.2.1	Úloha 2 – řešení	30
3.3	Úloha 3 – zadání	32
3.3.1	Úloha 3 – řešení	33
3.4	Úloha 4 – zadání	35
4	ZÁVĚR.....	36
5	ZDROJE	37
6	PŘÍLOHY	1

1 Úvod

Cílem práce je vytvořit sestavu robotických systémů řízených programovatelným automatem (PLC). Na této sestavě budou demonstrovány tři úlohy, které prohloubí znalosti studentů z oblasti automatického řízení strojů. Umožní praktické vyzkoušení programování strojů s využitím profesionálního PLC na robotických modelech.

Základními komponenty této soustavy budou robotická ruka a dvě dopravní linky s pracovními stanicemi. Pro řízení soustavy bude využit Siemens Simatic S7-1200 s dotykovým displejem.

Na sestavě budou vytvořeny čtyři úlohy včetně návodů pro jejich zhotovení.

1 Automatizace, praktické využití

1.1 Od mechanizace k automatizace

Po dlouhá staletí byla většina lidstva odsouzena k tvrdé a namáhavé práci. Proto lidstvo vytvořilo stroje a nástroje, které pracovaly místo něho. První známky o mechanizaci pocházejí z dob faraonů ve starém ve starém Egyptě, kde se využívalo páry k otevírání velkých chrámových vrat.

Dále následovalo dlouhé období zdokonalování výrobních strojů a tím k zvyšování mechanizace neboli k snižování rozsahu fyzicky namáhavých prací. Začaly se objevovat první prvky automatizace výrobních procesů. V této době byla automatizace založena na mechanickém principu, například systém vaček. [1]

1.2 Automatizace

Termín automatizace pochází z řeckého slovo *automátos* (= sám o sobě konající). Představuje důsledek snahy člověka oprostít se od namáhavé fyzické práce a monotónních činností.

Důvody zavádění automatizace:

- snížení výrobních nákladů
- zvýšení produktivity
- odstranění subjektivních chyb obsluhy
- nahrazení lidí v extrémních pracovních podmínkách
- zlepšení jakosti výroby
- zvyšování pohodlí člověka

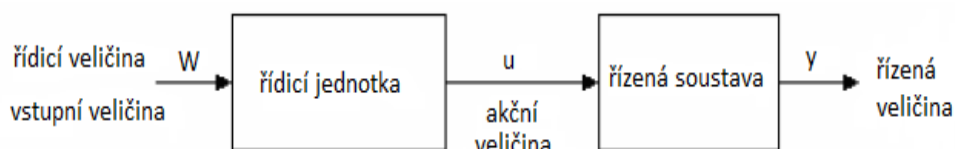
Cílem automatizace je tedy nahrazení činnosti lidí činnostmi vhodně navržených zařízení (automatů). V takovém systému přebírá veškerou činnost soubor přístrojů a čidel, které ovládají činnost strojů.

1.3 Řízení

Řízení - proces, při kterém je zařízení řízeno pomocí řídicích signálů tak, aby se dosáhlo požadovaného výsledku. Řízení je realizováno pomocí zavádění vstupních signálů do soustavy za účelem dosažení požadovaného stavu na výstupu. Řízení můžeme dále rozdělit na ovládání a regulaci.

1.3.1 Ovládání

Jedná se o způsob řízení, který pracuje bez zpětné vazby. Řídicí signály působí na zařízení, aniž by byly upravovány na základě vyhodnocení výstupů řízeného systému.

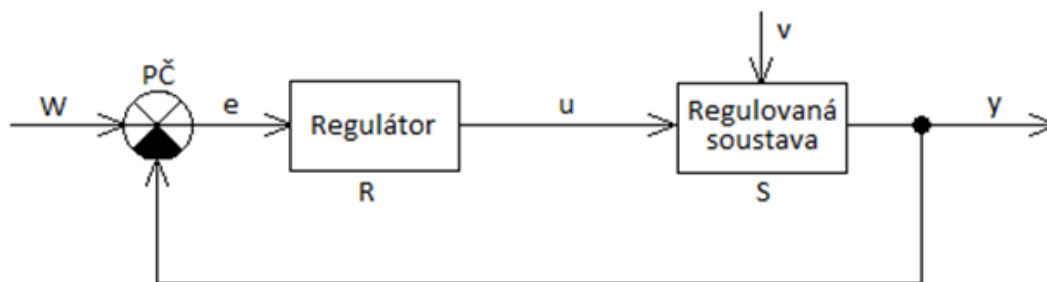


obr.1.: Blokové schéma ovládání [2]

1.3.2 Regulace

Norma DIN IEC 60050-351:2009-06 definuje pojem regulace následovně:

Regulace je proces, při kterém se průběžně ovládá regulovaná veličina jinou proměnnou veličinou (referenční veličina), takovým způsobem, aby došlo k jejich vyrovnání. [18]



obr.2.: Blokové schéma regulačního obvodu

Základní veličiny regulovaného systému:

- **regulovaná veličina (y)** - výstupní veličina regulovaného systému
- **požadovaná veličina, řídicí veličina (W)**
- **porovnávací člen (PČ)** - porovnává skutečnou a žádanou hodnotu regulované veličiny
- **regulační odchylka (e)** – rozdíl mezi regulovanou a požadovanou veličinou
- **regulátor** – zařízení, které samočinně uskutečňuje regulaci
- **akční veličina (u)** – výstupní veličina regulátoru a zároveň vstupní veličina regulované soustavy
- **regulovaná soustava** – zařízení, na kterém se provádí regulace
- **poruchová veličina (v)** – vnější působení (např. změna teploty)

Na obr.2 je znázorněn regulační obvod se zápornou zpětnou vazbou, která slouží k získání regulační odchylky pomocí přivedení výstupního signálu do porovnávacího členu, kde dojde k porovnání signálu mezi výstupem soustavy a požadovanou veličinou. Zpětná vazba musí být nastavena tak, aby působila proti regulační odchylce a musí být nastavena jako záporná zpětná vazba, jelikož potřebujeme potlačit rozdíl mezi požadovanou veličinou (w) a regulační veličinou (y).

Regulační obvod lze popsat rovnicí:

$$e(t) = w(t) - y(t)$$

Z této rovnice vidíme, že regulátor pracuje tak, aby zmenšil či úplně potlačil regulační odchylku.

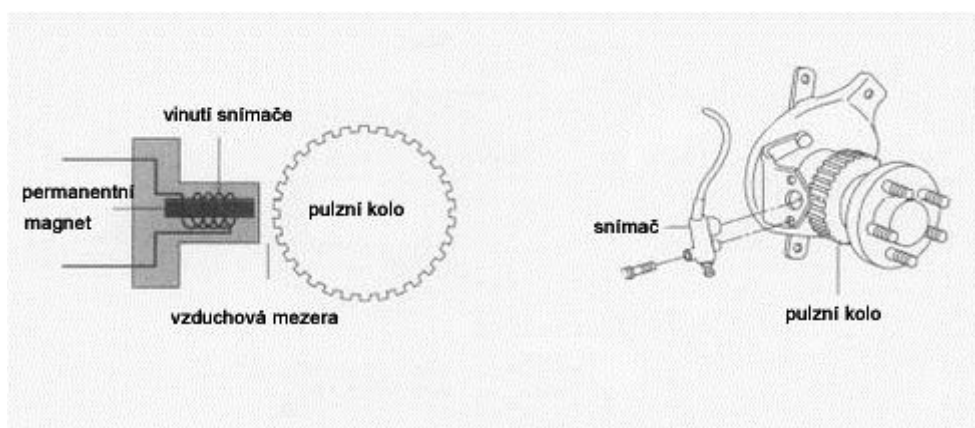
1.4 Prostředky automatizace

Prostředky automatizace lze rozdělit na:

- Snímače
- Regulátory
- Akční členy

1.4.1 Snímače

Jedná se o zařízení informující o stavu a činnosti technického zařízení. Tato prvky jsou určeny pro snímání a detekci různých fyzikálních veličin, vlastností látek a technických stavů. Pracují na principu převádění dané veličiny na vstup na tzv. měřenosnou veličinu na svém výstupu. Snímače tím tvoří vazební článek mezi technických zařízením a jeho řídicím systémem. V praxi můžeme narazit např. na snímače polohy, rychlosti, zrychlení, síly, tlaku, teploty, průtoku atd.



obr.3.:Induktivní snímač pro systém ABS [3]

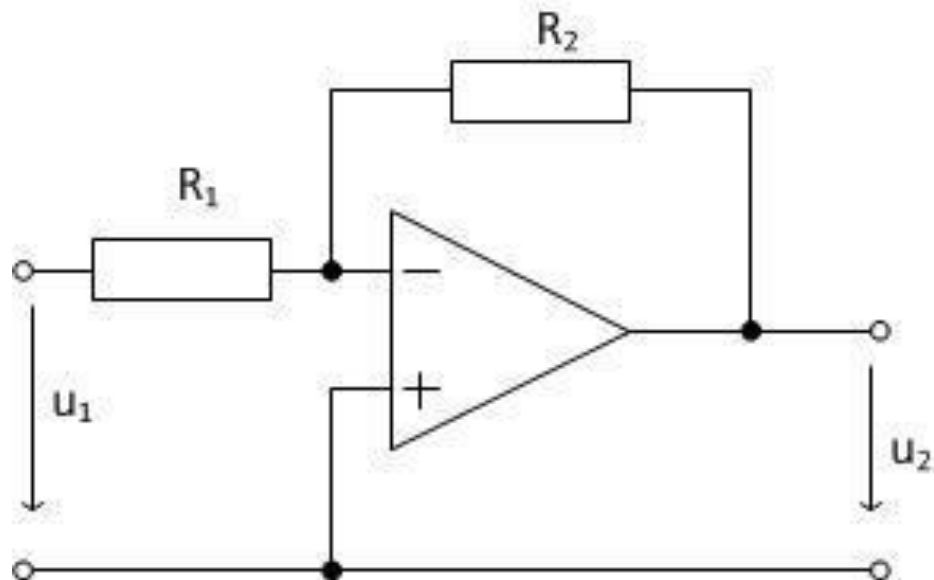
1.4.2 Regulátory

Jak už název napovídá, jedná se o zařízení, která v dané regulační soustavě automaticky udržují veličinu na požadované hodnotě. Regulátory můžeme rozdělit podle několika kritérií např. podle druhu regulované veličiny, podle druhu přenášeného signálu (analogové, digitální), podle průběhu výstupního signálu (spojité, nespojitě), ale také podle přenosových vlastností na tzv.:

- regulátory proporcionální (P)
- regulátory integrační (I)
- regulátory derivační (D)
- sdružené regulátory (PI, PD, PID)

Proporcionální regulátor

Jedná se o základní a nejjednodušší regulátor, u něhož je výstupní signál přímo úměrný vstupnímu signálu. Tento typ regulátoru pouze zesiluje regulační odchylku e . Tento regulátor vytvoříme pomocí invertujícího zesilovače.



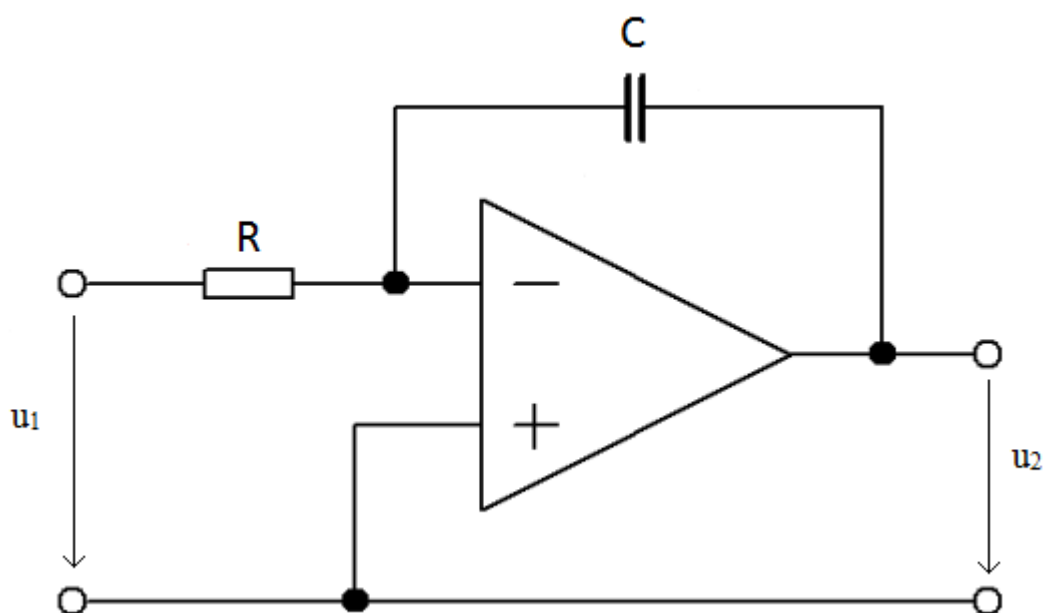
obr.4.: Základní zapojení proporcionálního regulátoru

$$G(p) = -\frac{R_2}{R_1}$$

$$U_2 = -\frac{R_2}{R_1} = K * U_1$$

Integrační regulátor

V soustavě řízené I regulátorem reaguje regulátor na skok postupně narůstajícím akčním zásahem, jelikož akční veličina je úměrná integrálu vstupní veličiny. Tento typ regulátoru jako jediný umožňuje úplné odstranění regulační odchylky e , jelikož je regulátorem integrována. K jejímu úplnému odstranění dojde až za určitý čas, proto se I regulátor hodí tam, kde poruchy nejsou příliš časté.

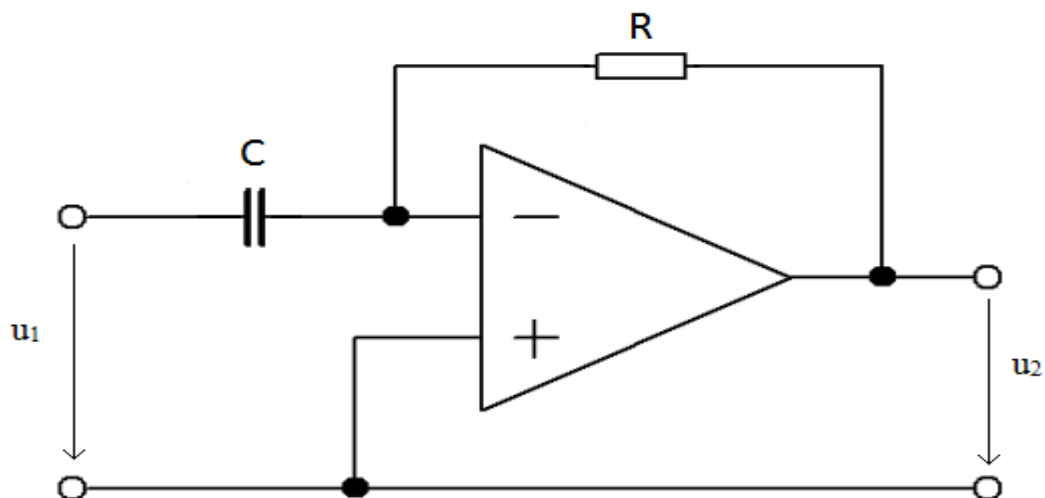


obr.6.: Základní zapojení integračního regulátoru

Integrační regulátor lze realizovat pomocí invertujícího zesilovače.

$$G_{(p)} = \frac{U_2}{U_1} = -\frac{1}{pRC}$$

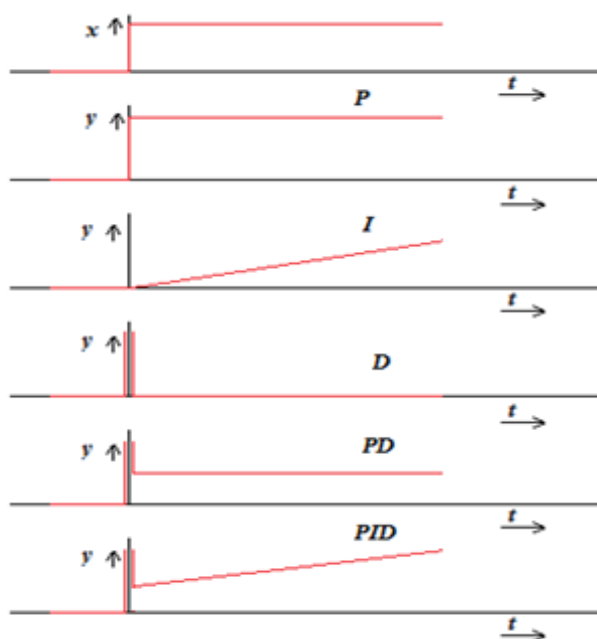
Derivační regulátor



obr.8.: Základní zapojení derivačního regulátoru

Regulátor typu D reaguje pouze na změnu vstupního signálu, tzn. že při konstantním vstupu má nulový přenos. Samotný derivační regulátor nezesiluje regulační odchylku e , a proto musí být vždy kombinován s regulátorem typu P, popř. typu I. Vlastností regulátoru typu D v kombinaci s typem P či I je zrychlení regulace.[1]

Dále se používají různé kombinace výše uvedených základních typů regulátorů.



obr.9.: Idealizované odezvy základních typů regulátorů [4]

1.4.3 Akční členy

Akční členy jsou všechny prvky určené k využití zpracované informace. Akční členy lze rozdělit do dvou skupin:

- **Pohony** - převádějí signály na výchylku konající požadovanou práci s požadovaným výkonem. Rozdělujeme je na pneumatické, hydraulické a elektrické.
- **Regulační orgány** – zařízení pro ovládání toku hmoty nebo energie systémem. Rozdělujeme na ventily, kohouty, klapky atd.

1.5 Pojem PLC

1.5.1 Vznik a vývoj PLC

V roce 1968 podala jedna z divizí General Motors žádost na nalezení alternativního řešení ke komplexním elektromechanickým (reléovým) řídicím systémům, jelikož reléové řízení mělo několik nevýhod, a to:

- pevné propojení
- obtížná změna řídicí funkce
- poměrně nákladné uvádění do provozu

Z těchto nevýhod vzešly i požadavky, které by měl nový řídicí systém splňovat:

- zvýšená odolnost do průmyslového prostředí
- jednoduché uvádění do provozu
- možnost monitorování provozu
- dostatečná rychlost pro možnost řízení
- jednoduchá a levná údržba
- menší, levnější a spolehlivější než předchozí řídicí systém

Vítězný návrh na nový řídicí systém přišel od Bedford Associates. Šlo o první programovatelný logický automat (dále jen „PLC“).

První PLC bylo označeno 084, protože se jednalo o osmdesátý čtvrtý projekt Bedfordské asociace. Po tomto vynálezu založila tato asociace novou firmu, která byla pojmenována Modicon (**MOD**ular **DI**gital **CON**troller). Firma Modicon nadále vyrábí a prodává PLC, avšak pod novým majitelem, a to francouzským Schneider Electric.

Jeden z úplně prvních modelů 084 vydržel ve výrobním koncernu GM téměř 20 let nepřetržitého provozu. Nyní je vystaven v Modiconské ústředně v Andoveru v Massachussets.

První PLC byly programovány v tzv. „*ladder logic*“ neboli „jazyku příčkového diagramu“, který se silně podobá diagramu reléové logiky. Hlavním důvodem pro zvolení jazyku příčkového diagramu bylo zkrácení doby nutné pro přeškolení stávajícího personálu, který byl zvyklý na reléovou logiku. Moderní PLC mohou být programovány v několika rozdílných jazycích, avšak na programování PLC se zaměříme později. [6]

V dnešní době lze použít PLC na opravdu rozličné druhy aplikací, jako jsou řízení výrobních strojů, regulace chlazení a vytápění, řízení osvětlení, kontrolní úlohy, zabezpečovací systémy, vzdálené řízení vybavení domů a bytů atd.

1.5.2 Struktura PLC

PLC se skládá z několika základních částí:

- **základní řídicí (CPU) jednotka zahrnuje:**
 - zdrojovou část – napájení jednotky či dalších modulů
 - procesorovou část – tvořená CPU
 - komunikační rozhraní – základní propojení s PC či dalšími PLC
 - vstupy / výstupy – analogové či digitální

- **rozšiřující moduly**

- rozšiřují základní jednotku o další prvky, jako jsou komunikační moduly, vstupní či výstupní moduly a zabezpečovací moduly.

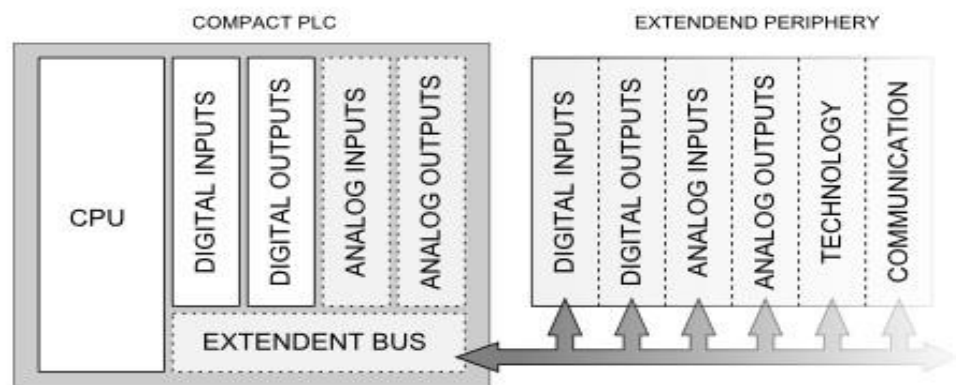
Základem principu PLC jsou v podstatě 3 funkční bloky:

- zpracování informace
- vstupy/výstupy
- paměť

Z hlediska konstrukce můžeme PLC rozdělit do dvou kategorií:

- **Kompaktní** - v jednom modulu je obsažen CPU, digitální a analogové vstupy/výstupy, konektory pro základní komunikaci, někdy i zdroj či displej.

- největší výhodou kompaktních systémů je jejich rychlost přístupu k perifériím, jelikož jsou integrovány v PLC a signály nemusí procházet přes řadiče sběrnic.[7]



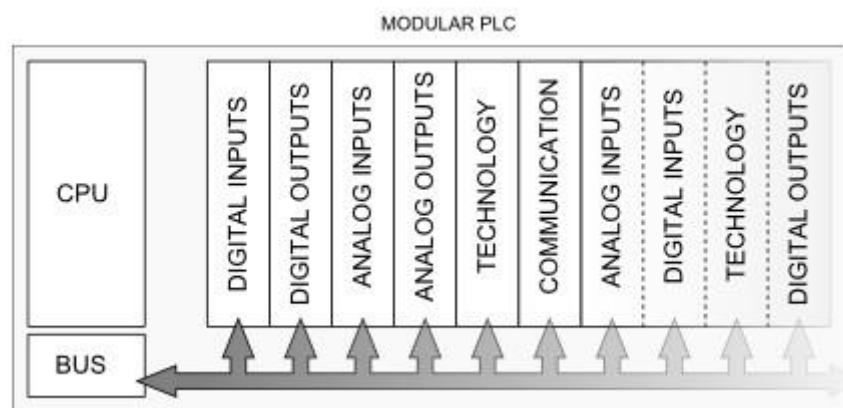
obr.9.: Princip kompaktního PLC s možným rozšířením [7]



obr.10.: Kompaktní PLC easy Control EC4-200 [8]

- **Modulární** – systém je rozdělen do několika modulů, z nichž jeden musí být CPU a k němu lze dle potřeby připojit další rozšiřující moduly, jako vstupní/výstupní moduly, komunikační moduly či funkční moduly.

- výhodou tohoto typu systému je možnost jeho sestavení přesně podle dané úlohy či jeho pozdější upgradování, které se provede pouze dokoupením a instalací potřebného modulu do stávajícího systému.



obr.11.: Princip modulárního PLC [7]



obr.12.: Modulární PLC Siemens S7-1200 s rozšiřujícími moduly [9]

1.6 Programování PLC

Podle normy IEC 61131-3 lze programovací jazyky rozdělit do dvou skupin, a to:

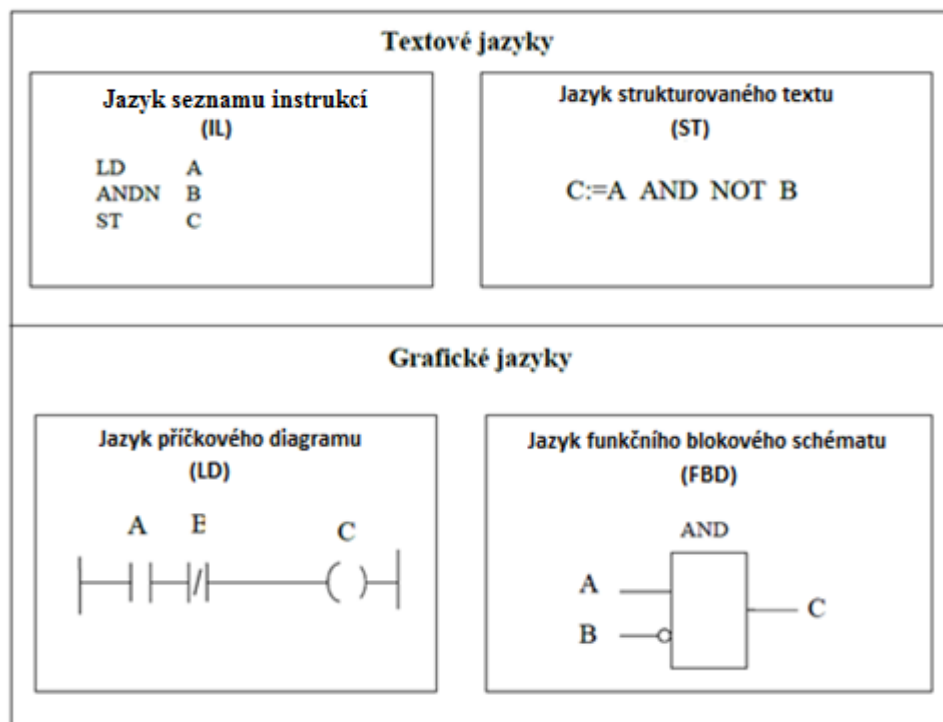
Textové jazyky:

- IL** - Instruction List – jazyk seznamu instrukcí
- ST** - Structured Text – jazyk strukturovaného textu

Grafické jazyky:

- LD** - Ladder Diagram – jazyk příčkového diagramu
- FBD** - Function Block Diagram – jazyk funkčního blokového schématu

Každý z těchto programovacích jazyků je vhodný pro jiný druh aplikace, a proto si tyto jazyky podrobněji popíšeme:



obr. 14: Rozdělení programovacích jazyků dle normy IEC 61131-3 [10]

Jazyk seznamu instrukcí

IL – jedná se o nejzákladnější možný zápis programu, který poměrně připomíná assembler. Kód se skládá z několika instrukcí, přičemž každá instrukce začíná na novém řádku. Instrukce obsahuje název dané instrukce a případně i název použitého operandu. Například načtení hodnoty do zásobníku se provede příkazem „**LD**“. Načtení proměnné „**VSTUP**“ tedy zapíšeme:

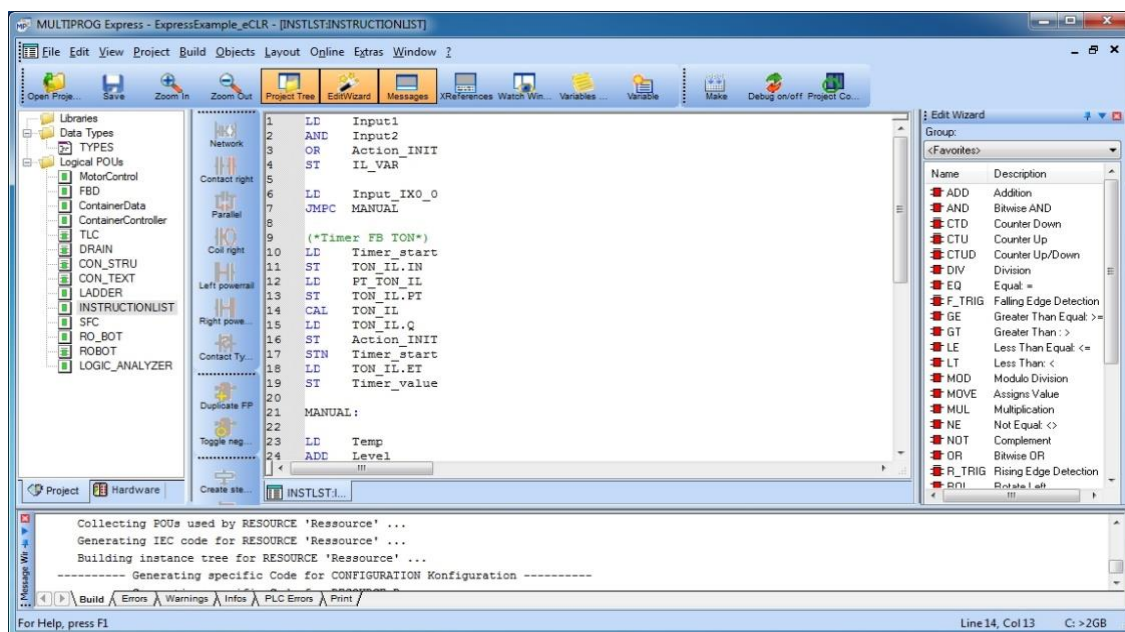
LD VSTUP

Soubor všech možných instrukcí daného automatu se nazývá instrukční sada. [11]

Výhodou toho programovacího jazyka je, že výsledný program je úsporný jak na čas potřebný pro zpracování úlohy, tak také na paměťovou kapacitu potřebnou pro uložení programu.

Mezi nevýhody patří: nutnost dobré orientace v příkazech a registrech, horší přehlednost a orientace u složitějších programů.

Z těchto vlastností vyplývá, že tento programovací jazyk je vhodný pro psaní krátkých a vysoce optimalizovaných částí programu. [12]



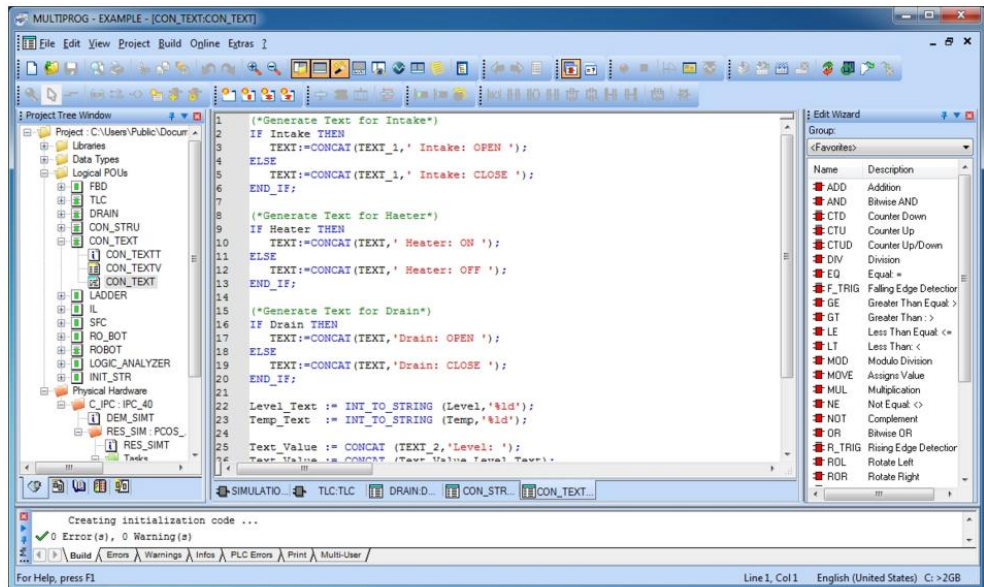
obr. 15: Příklad programovacího jazyka IL pomocí softwaru MULTIPROG Express [13]

Jazyk strukturovaného textu

ST - jedná se o vyšší programovací textový jazyk, jehož základy jsou v jazycích Pascal a C. Jde o objektově orientovaný jazyk, který obsahuje všechny důležité prvky moderního programovacího jazyka, jako je větvení (IF-THEN-ELSE a CASE OF) a cykly (FOR, WHILE a REPEAT).

Jazyk strukturovaného textu je určen pro složité výpočty a algoritmy, jejichž zápis pomocí instrukcí by byl velmi složitý. Tento programovací jazyk je zcela obecný a má poměrně obsáhlou nabídku příkazů pro různé typy úloh.

Výhodou tohoto programovacího jazyka je to, že i pro složité úlohy lze napsat program, který je přehledný a snadno opravitelný.



obr. 16: Příklad programovacího jazyka ST pomocí softwaru MULTIPROG[14]

Jazyk příčkového diagramu

LD - jazyk příčkového diagramu spadá do skupiny grafických jazyků. Lze říci, že se jedná o grafické znázornění výše popsaného jazyku systémových instrukcí (IL).

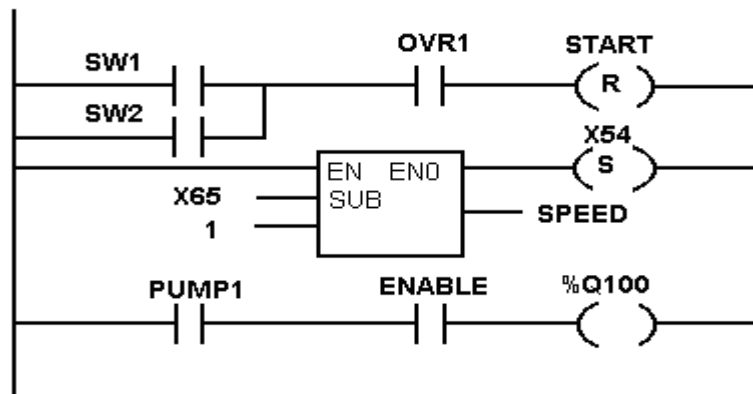
Výsledný program má formu kontaktního schématu v elektronice. Kontaktní schéma je ohraničeno dvěma svislými čarami (sběrnicemi) a jednotlivé funkce se nacházejí na vodorovných spojnicích těchto dvou sběrnic (připomínající příčky v pomyslném žebříčku). Bitové operace jsou realizovány pomocí ovládacích a výstupních kontaktů relé. Složitější příkazy a funkce (například generátory impulzů, čítače, časovače a další) jsou vkládány pomocí „obdelníků“ s vývody.

Funkce	Anglický název	Symbol
spínací kontakt	<i>open contact</i>	- -
rozpínací kontakt	<i>closed contact</i>	- / -
cívka s přímou funkcí	<i>coil</i>	-()-
cívka s negovanou funkcí	<i>negated coil</i>	-(/)-
cívka s funkcí set	<i>set coil</i>	-(S)-
cívka s funkcí reset	<i>reset coil</i>	-(R)-

obr. 17: Základní prvky jazyka příčkového diagramu [15]

Tento programovací jazyk vychází z dob, kdy se místo logických obvodů NAND, NOR a XOR využívala sestava vzájemně propojených relátek (= reléová logika).

Program je vykonáván zleva doprava a shora dolů.



obr. 18 :Příklad programu v jazyku příčkového diagramu [16]

Mezi výhody tohoto typu programování patří: přehlednost zápisu, poměrně rychlé programování logických operací a časování.

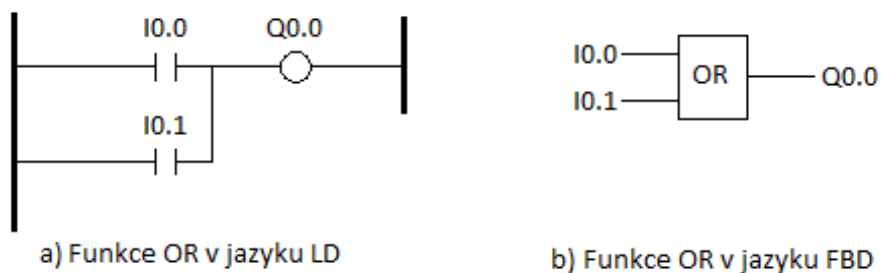
Mezi nevýhody patří: velká délka programu s narůstající složitostí, tento jazyk je hůře pochopitelný pro „klasické programátory“.

Jazyk funkčního blokového schématu

FBD - jazyk funkčních blokových schémat je další ze skupiny grafických jazyků popsán normou IEC 61131 – 3.

Posloupnost programu je reprezentována soustavou za sebou propojených bloků, které představují dané funkce. Režim FBD si lze představit jako výše popsáný LD režim, avšak rozdíl je v tom, že místo sériově-paralelního propojování kontaktů relé se zde používají klasické značky hradel AND, OR, ADD, avšak můžeme zvolit i funkční bloky, jako jsou časovače, prvky pro detekci náběžné/sestupné hrany, čítače a další.

FBD režim je vhodný pro uživatele, kteří požadují velikou přehlednost programu, ale nevyhovuje jim jazyk kontaktních („reléových“) schémat.



obr. 19: Srovnání funkce OR v jazyku LD a FBD

2 Popis komponentů

Jedná se o stavebnici Fischertechnik, kterou dodala firma Helago. Firma Fishertechnik vyrábí stavebnice ve dvou verzích, a to 9 V a 24 V verzi. Při této práci je použita verze s 24 V napájením.

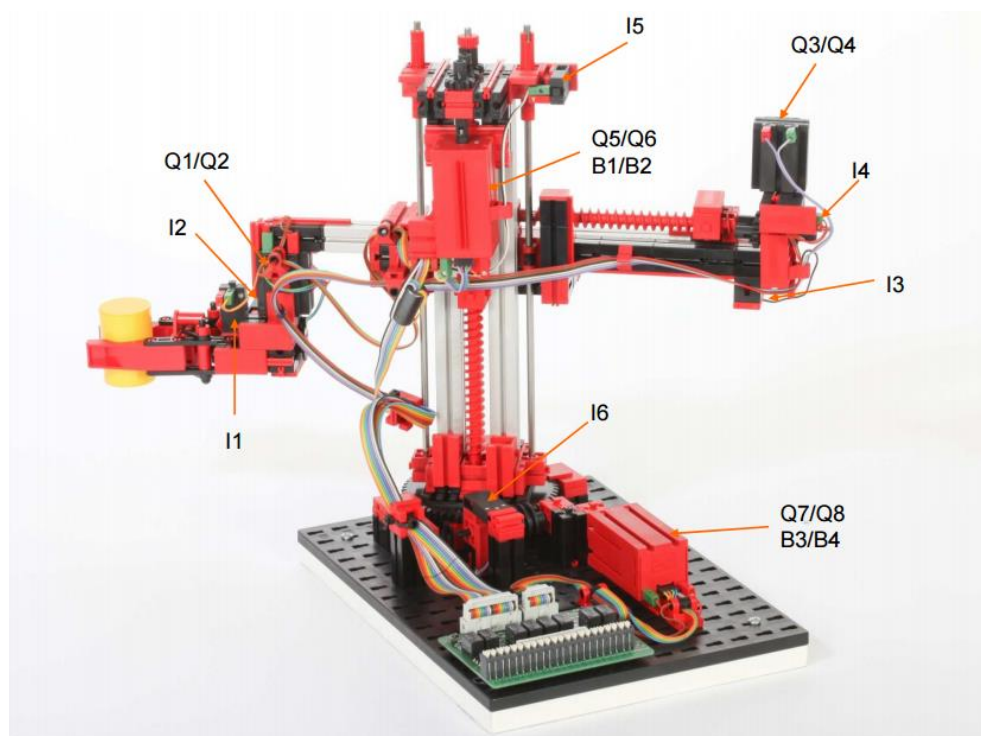
2.1 Rameno

Tento robot má 3 hřídele. První hřídel umožňuje rotaci o 180°. Druhá hřídel slouží k posunu uchopovacího mechanismu směrem vpřed či vzad o 90mm. Třetí hřídel slouží k pohybu ve vertikálním směru v rozsahu 150mm.

Pohyb zajišťují 4 DC motory, 3 slouží k pohybu ramene a 1 k ovládání uchopovacího zařízení.

Stavebnice dále obsahuje 4 koncové spínače k ošetření koncových poloh. Na plošném spoji je vyvedeno 8 digitálních vstupů, 8 digitálních výstupů a piny pro 24V napájení.

Celá tato stavebnice je připevněna na dřevěné desce o rozměrech (480x400x220 mm).



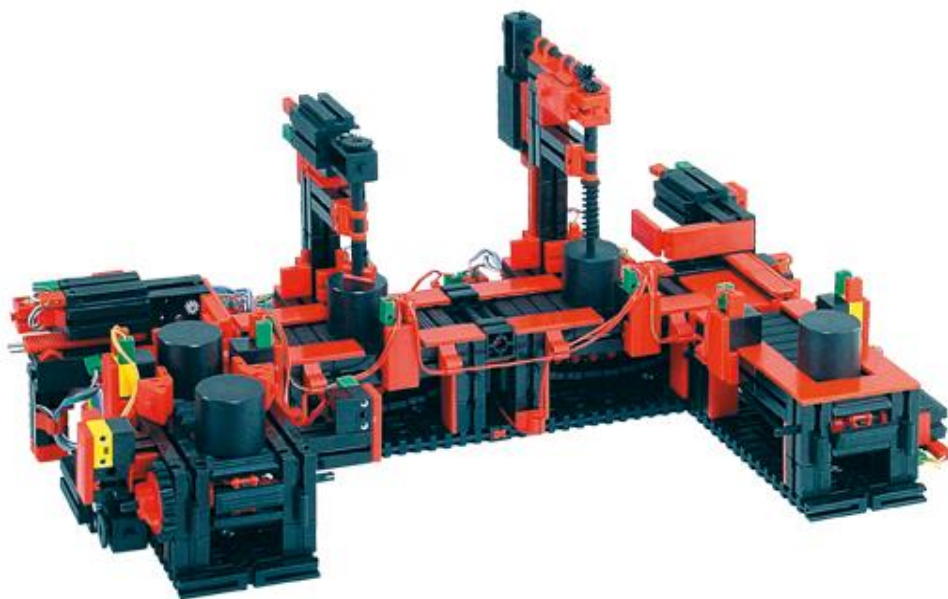
obr.20.: Robotické rameno výrobce Fischertechnik

Pin	Funkce	Vstup/výstup
1	Napájení (+) motorů	24VDC
2	Napájení (+) senzorů	24VDC
3	Napájení (-)	0V
4	Napájení (-)	0V
5	Referenční přepínač svěráku	I0.4
6	Pulzní počítadlo svěráku	I0.2
7	Referenční přepínač paže svěráku	I0.7
8	Pulzní počítadlo paže svěráku	I0.3
9	Referenční přepínač vertikální osy	I0.6
10	Referenční přepínač otáčení	I0.5
11	Enkodér vertikální osy impulzu 1	I0.0
12	Enkodér vertikální osy impulzu 2	I0.1
13	Enkodér otáčení impulzu 1	I1.4
14	Enkodér otáčení impulzu 2	I1.5
15	Motor otevírání svěráku	Q0.4
16	Motor zavírání svěráku	Q0.7
17	Motor paže svěráku dopředu	Q0.6
18	Motor paže svěráku dozadu	Q0.5
19	Motor vertikální osy dolů	Q0.0
20	Motor vertikální osy nahoru	Q0.1
21	Motor otáčení doprava	Q0.2
22	Motor otáčení doleva	Q0.3

Tab.1: Rozvržení obvodu robotického ramene

2.2 Výrobní linka

Dopravní pás ve tvaru U pro přerušovaný pohyb a pro zpracování dílů na několika místech. Tato výrobní linka se skládá z 2 obráběcích stanic, 4 dopravních pásů, 8 DC motorů, 4 koncových spínačů a 5 světelných závor. Model je připevněn na dřevěné desce o rozměrech 450x410x190mm(d x š x v).



obr.21: Výrobní linka se 2 stanicemi

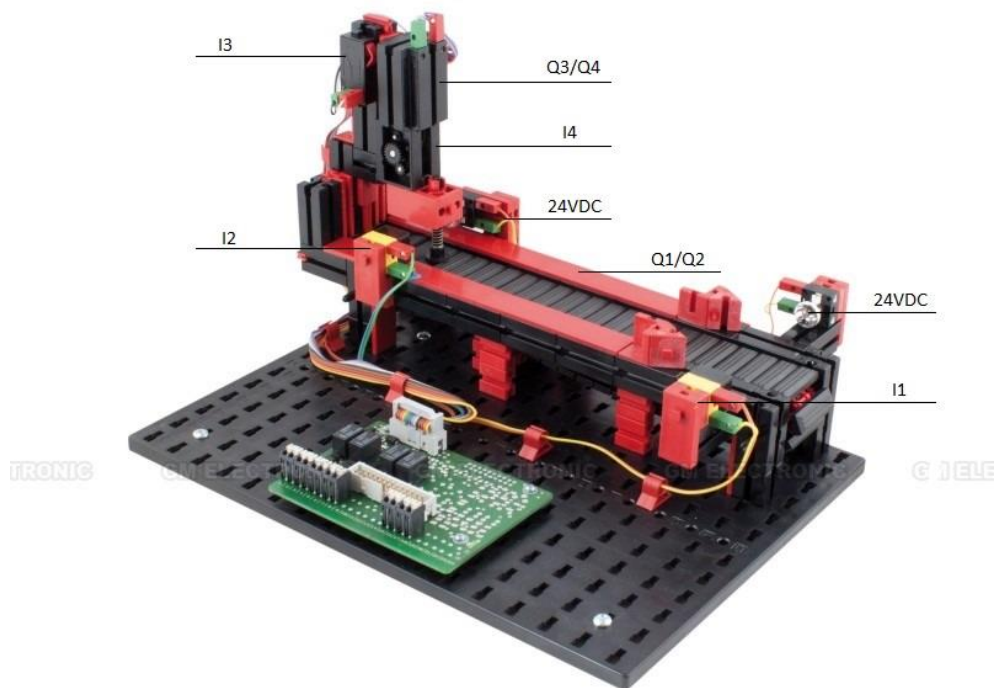
Pin	Funkce	Vstup/výstup
1	Napájení (+) akčních členů	24VDC
2	Napájení (+) senzorů	24VDC
3	Napájení (-)	0V
4	Napájení (-)	0V
5	Tlačítko 1, přední strana	I2.0
6	Tlačítko 1, zadní strana	I2.1
7	Tlačítko 2, přední strana	I2.2
8	Tlačítko 2, zadní strana	I2.3
9	Přepínač fototranzistoru	I2.4
10	Fototranzistor mlýnku	I2.5

11	Fototranzistor nakládací stanice	I2.6
12	Fototranzistor drtiče	I2.7
13	Fototranzistor výměny pásu	I3.0
15	Přepínač motoru 1 dozadu	Q2.0
16	Přepínač motoru 1 dopředu	Q2.1
17	Přepínač motoru 2 dozadu	Q2.2
18	Přepínač motoru 2 dopředu	Q2.3
19	Motor pásového podavače	Q2.4
20	Motor pásu k mlýnku	Q2.5
21	Motor mlýnku	Q2.6
22	Motor pásu k drtiči	Q2.7
23	Motor drtiče	Q3.0
24	Motor výměny pásu	Q3.1

Tab.2.: Rozvržení obvodu výrobní linky

2.3 Obráběcí stanice

Dopravní pás se dvěma světelnými závory a značkovací stanicí. Tato stavebnice obsahuje 2 DC motory, 2 koncové spínače, 2 světelné závory složené z fototranzistoru a žárovky s čočkou. Na plošný spoj jsou vyvedeny piny pro 4 digitální vstupy, 4 výstupy a napájení.



obr.22: Značkovací stanice s dopravním pásem

Pin	Funkce	Vstup/výstup
1	Napájení (+) akční členy	24VDC
2	Napájení (+) senzory	24VDC
3	Napájení (-)	0V
4	Napájení (-)	0V
5	Světelný senzor příchozího/odchozího předmětu	I1.0
6	Světelný senzor obrobku	I1.1
7	Vypínač pro pohyb obrobku nahoru	I1.2
8	Vypínač pro pohyb obrobku dolů	I1.3
15	Motor pro pohyb pásu dopředu	Q1.0
16	Motor pro pohyb pásu dozadu	Q1.1
17	Motor pro pohyb obrobku nahoru	Q3.2
18	Motor pro pohyb obrobku dolů	Q3.3

Tab.2.: Rozvržení obvodu obráběcí stanice

2.4 HMI panel

Barevný dotykový panel značky Siemens s označením SIMATIC HMI KTP1000 BASIC COLOR PN. Jedná se o 10" TFT displej s rozlišením 640x480 pixelů. Na displeji lze nastavit 8 funkčních kláves. Komunikace s řídicí jednotkou je realizována pomocí rozhraní PROFINET.

2.5 PLC

Série S7-1200 je řada programovatelných logických automatů (PLC), které mohou řídit různorodé automatické aplikace. Kompaktní konstrukce, nízká cena a výkonná instrukční sada činí z S7-1200 perfektní řešení pro řízení široké škály aplikací.

Toto PLC kombinuje mikroprocesor, integrované napájení, vstupní a výstupní obvody, zabudovaný PROFINET.

PROFINET port slouží ke komunikaci přes PROFINET síť. Komunikaci lze rozšířit připojením přídatných modulů o komunikaci pomocí PROFIBUS, GPRS, RS485, RS232, IEC a WDC sítě.



obr.23: Použité PLC Simatic S7-1200, CPU 1215C

V rámci této bakalářské práce je použita řídicí jednotka s označením PLC SIMATIC S7-1200,CPU 1215C (AC/DC/RLY). Tato jednotka obsahuje 2 PROFINET

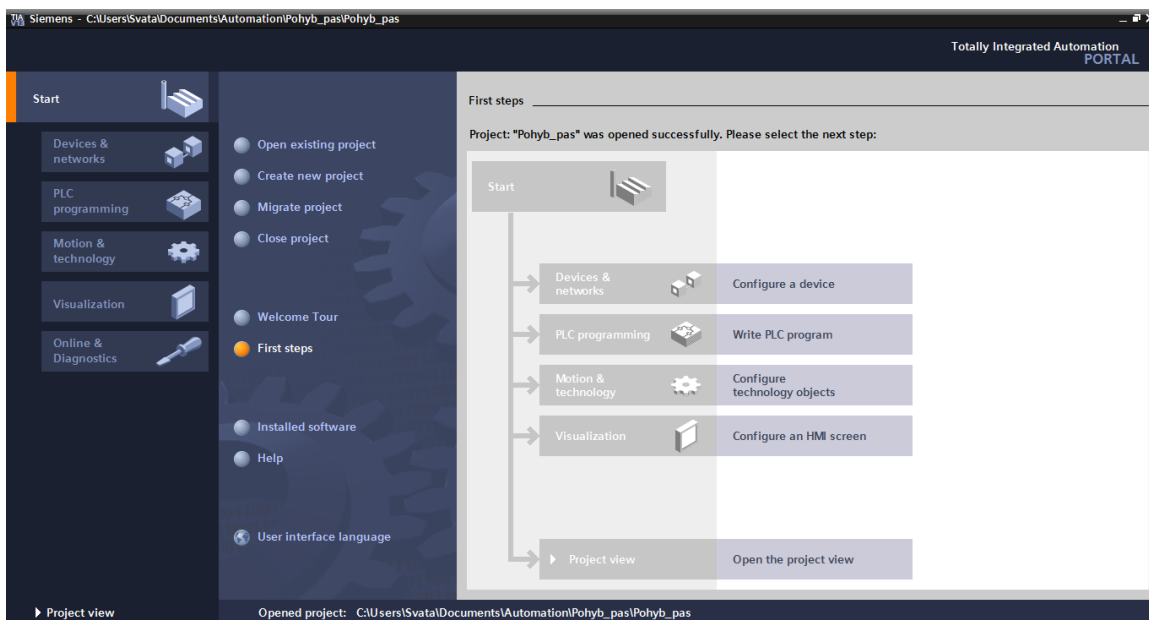
porty, 14 digitálních vstupů, 6 HSC, 10 digitálních výstupů, 2 analogové vstupy a 2 analogové výstupy. Tuto jednotku lze rozšířit až o 3 moduly pro sériovou komunikaci a až o 8 signálních modulů pro rozšíření počtu vstupů/výstupů.

Pro konfiguraci a následné programování řídicích jednotek značky Simatic se využívá software STEP 7. Tento software poskytuje uživatelsky příjemné prostředí pro vytváření, úpravu a monitoring potřebný k ovládání aplikací, včetně nástrojů pro správu a konfiguraci všech zařízení v projektu, jako je například HMI panel.

Step 7 podporuje několik programovacích jazyků. Změnu programovacího jazyku provedete následujícím způsobem:

1. Kliknutím pravého tlačítka na programový blok (např. Main)
2. Z otevřeného menu vybereme „Switch programming language“.
3. Zvolíme požadovaný programovací jazyk.

Step 7 je součástí tzv. TIA portalu („Totally Integrated Automation“). TIA portal dále obsahuje WinCC, což je software sloužící pro vizualizaci procesů, do kterého přejdeme pomocí nabídky „Visualization“ na hlavní stránce TIA portalu.[17]



obr.24.: Screen vývojového systému TIA portal.

3 Úlohy

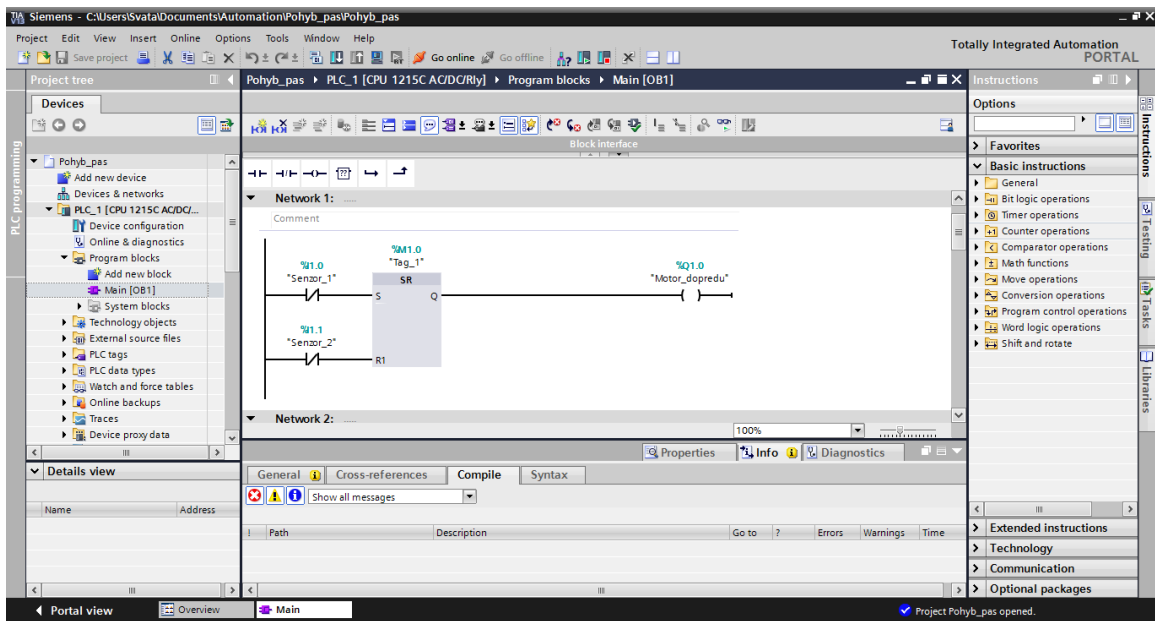
3.1 Úloha 1 – zadání

První úloha slouží k osvojení si základních principů programování řídicí jednotky Simatic S7-1200 pomocí softwaru STEP 7 Basic V13.

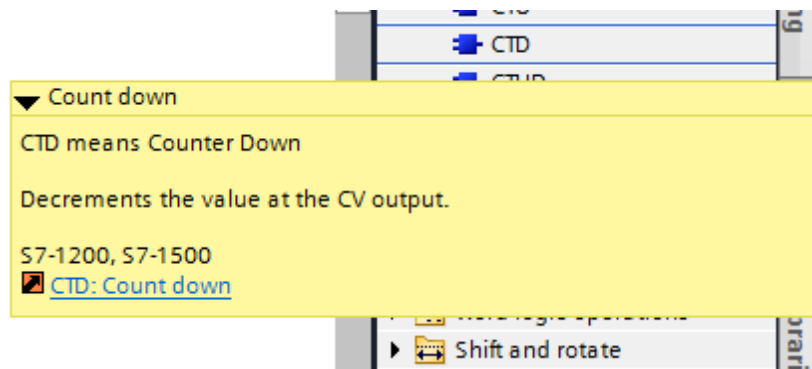
- 1) Vytvořte program, který dopraví obrobek k razítkovací stanici. Pás by se měl dát do pohybu po vložení obrobku před optické čidlo příchozích/odchozích předmětů a zastavit, pokud předmět dorazí před optické čidlo obrobku.
- 2) Naprogramujte pohyb razítkovače. Využijte koncových spínačů pro ošetření pohybů tak, aby razítkovač nevyjel ze své dráhy.
- 3) Upravte program tak, aby se po orazítkování předmětu dal pás do pohybu směrem k optickému čidlu pro příchozí/odchozí předměty.

3.1.1 Úloha 1 - řešení

- 1) Po spuštění TIA portalu vytvoříme nový projekt.
- 2) Ve složce „Devices & networks“ přidáme PLC a rozšiřující moduly. Nejrychlejší způsob přidání celé konfigurace je výběr Unspecified CPU 1200 verze 4.0.
Po zobrazení oranžového rámečku zvolíme možnost „detect“.

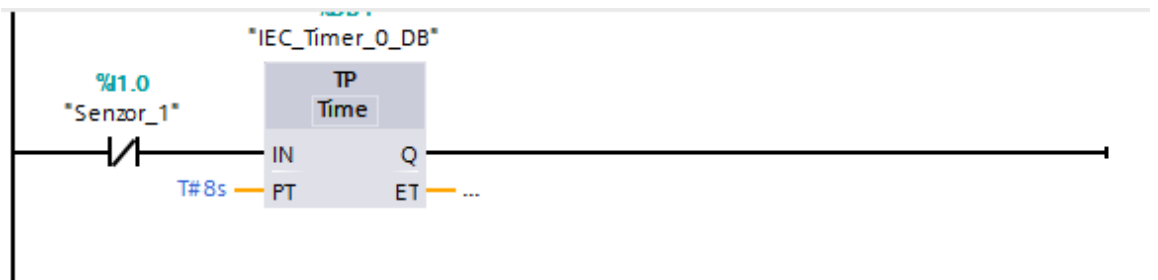


- 3) Do programovacího prostředí se dostaneme pomocí: „PLC_1[CPU 1215C AC/DC/Rly] -> Program blocks -> Main [OB1]
- 4) Instrukce jednoduše přetahujeme ze seznamu instrukcí do jednotlivých networků. Každá instrukce je podrobněji popsána v její nápovědě, do které se dostaneme tak, že najedeme myší na danou instrukci a po zobrazení oranžového rámečku klikneme na odkaz.



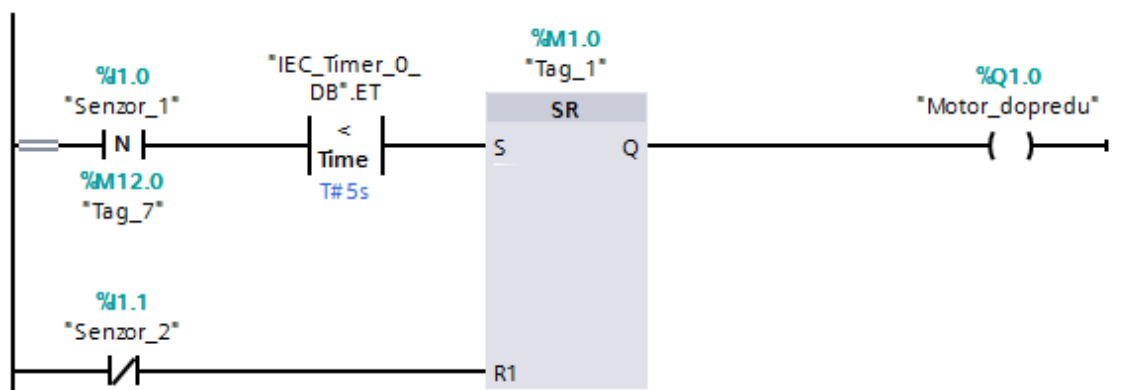
- 5) Jednou z možností ovládní pohybu pásu je použití klopného obvodu SR a obou optických čidel.
- Do networku vložíme klopný obvod SR.
 - Na vstup S zapojíme rozpínací kontakt („Normally closed contact“) čidla příchozích/odchozích předmětů, jelikož příchozí předmět nám změní hodnotu čidla z log1 na log0.
 - Na vstup R zapojíme rozpínací kontakt pro senzor obrobku.
 - Na výstup klopného obvodu zapojíme výstup Q1.0, který reprezentuje pohyb motoru vpřed.
- 6) Pro ošetření pohybu razítkovače lze opět využít klopného obvodu SR a koncových spínačů.
- Do networku vložíme klopný obvod SR.
 - Na vstup S zapojíme spínací kontakt („Normally open contact“) horního koncového spínače.
 - Na vstup R zapojíme spínací kontakt („Normally open contact“) dolního koncového spínače.
 - Na výstup klopného obvodu zapojíme výstup pro motor razítkovače dolu (Q3.3).
 - Stejným způsobem ošetříme i pohyb razítkovače směrem nahoru.
- 7) Pro pohyb pásu zpět opět použijeme SR klopný obvod a přidáme časovač stání, který slouží k tomu, aby se nám pás nedal do pohybu vpřed hned po dopravení obrobku před čidlo příchozích/odchozích předmětů.
- Do networku přidáme časovač (TP), který budeme spouštět senzorem příchozích/odchozích předmětů.
 - Do dalšího networku přidáme SR klopný obvod.
 - Na výstup zapojíme motor pásu pro pohyb vpřed.
 - Na vstup S do série zapojíme kontakt na sestupnou hranu a porovnávací kontakt.

e) Na vstup R zapojíme rozpínací kontakt senzoru přichozích/odchozích předmětů.



Network 2:

Comment



3.2 Úloha 2 – zadání

Druhá úloha slouží k seznáení studenta se softwarovým nastavením HSC (High speed counter) u PLC. Dále k propojení Touch panelu s PLC pomocí ETHERNET rozhraní a následnému naprogramování a vytvoření grafiky.

- 1) Naprogramujte automatický pohyb robotické ruky včetně ovládání kleští.
- 2) Propojte HMI panel s PLC přes ETHERNETOVÉ rozhraní.
- 3) Vytvořte softwarová tlačítka na HMI panelu pro ovládání pohybu robotické ruky a nastavení do referenční polohy.

3.2.1 Úloha 2 – řešení

Doporučuji začít s nastavením HMI panelu a následným naprogramováním ručního ovládání, jelikož během programování automatického pohybu pravděpodobně nastane situace, kdy se budete potřebovat vrátit do referenční polohy nebo budete potřebovat udělat jiný pohyb například pro naměření potřebných pulzů.

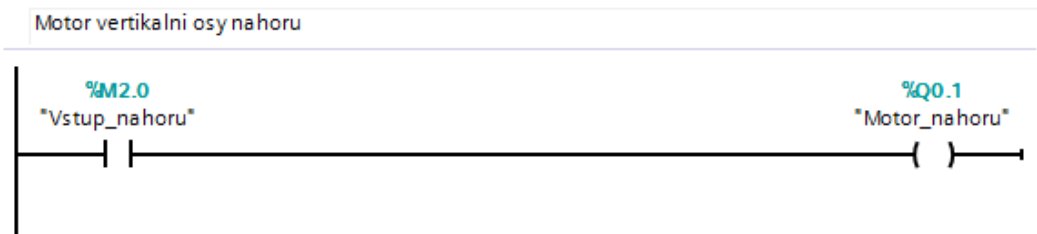
- 1) Pro automatické ovládání robotické ruky budeme potřebovat tzv. High speed counter (HSC) pro čítání pulzů enkodérů.
 - a) Pro každý enkodér povolíme HSC čítač v hardwarovém nastavení řídicí jednotky. Do nastavení se dostaneme touto cestou: „PLC_1[CPU 1215C AC/DC/Rly] -> Device configuration -> Properties -> High speed counter (HSC).”
 - b) Čítače nastavíme tímto způsobem:
 - Type of counting : Count up
 - Operating phase : A/B counter
 - Initial count. direction: Count up
 - c) Vytvoříme nový funkční blok, do kterého budeme psát kód automatického ovládání.
 - d) Načítáme si požadovaný počet impulzů pomocí HSC.

- e) Do dalšího networku vložíme SR klopný obvod. Na vstup S přivedeme bit pro zapnutí automatického ovládání. Na vstup R zapojíme porovnávací kontakt, na který dáme adresu HSC čítače, kterou najdeme v hardwarovém nastavení. Na vstup R doporučuji zapojit kontakt „větší/rovno“, jelikož rychlost čítání je větší, než rychlost porovnání a může se stát, že kontakt pro porovnání nestihne zareagovat a vyresetovat klopný obvod.
- f) Stejným způsobem pokračujeme pro druhý enkodér.
- g) Na čítání impulzů paže nám stačí obyčejný čítač CTU.

2) HMI panel propojíme s PLC pomocí LAN a softwarově ho přidáme tímto způsobem: „Devices & network -> Add new devices -> HMI panel -> 10" Display -> KTP1000 Basic -> 6AV6 647-0AF11-3AX0 -> version 12.0.0.0“

3) Ruční ovládání budeme řešit pomocí nastavení daného bitu přes HMI panel. Pro ruční ovládání doporučuji vytvořit nový funkční blok, který se následně přidá do bloku Main.

- a) Nový funkční blok přidáme tímto způsobem: „PLC_1[CPU 1215C AC/DC/Rly] -> Program blocks -> Add new block -> Function block.
- b) Do série zapojíme spínací tlačítko a výstup. Spínací tlačítko nastavíme na libovolný nepoužívaný bit a výstup nastavíme na pohyb motoru.



- c) Tímto způsobem pokračujeme pro všechny možné pohyby robotické ruky.
- d) Pomocí Project tree (nabídka na levé straně) se přepneme do HMI panelu.
- e) Ve složce Screens vytvoříme nový screen.
- f) Do vytvořeného screenu vložíme tlačítka tímto způsobem:
 - Z toolboxu ze složky elements vhodně umístíme na panel tlačítka („Button“).
 - Klikneme na dané tlačítko.
 - V podsložce „Events“ zvolíme možnost „Press“.
 - Přidáme funkci „SetBit“ a zvolíme požadovaný bit.
 - Dále zvolíme možnost „Release“.
 - Přidáme funkci „ResetBit“ a zvolíme požadovaný bit.
 - Stejným způsobem pokračujeme pro zbylá tlačítka.

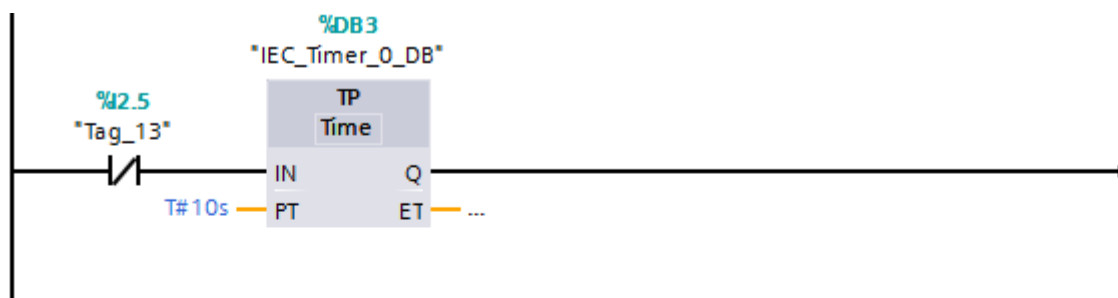
3.3 Úloha 3 – zadání

Třetí úlohou bude zprovoznění montážní linky se dvěma pracovními stanicemi včetně ručního ovládání.

- 1) Napište program pro řízení pohybů montážní linky. Opět využijte optických čidel a koncových spínačů pro spínání či vypínání pohybů pásů a obráběcích stanic.
- 2) Přidejte nový screen do HMI panelu a vytvořte ruční ovládání pro montážní linku se dvěma pracovními stanicemi.

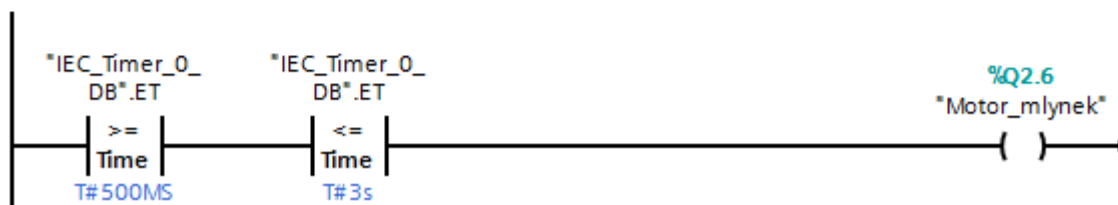
3.3.1 Úloha 3 – řešení

- 1) Vytvoříme nový blok, do kterého budeme psát program pro automatický pohyb.
 - a) Na pohyb prvního pásu použijeme SR klopný obvod. Na vstup S zapojíme rozpínací tlačítko prvního čidla. Na vstup R zapojíme vypínač (I2.1).
 - b) Na další pásy opět použijeme SR klopné obvody, které budeme překlápět pomocí optických čidel.
 - c) Do série zapojíme rozpínací kontakt a časovač.
 - d) Do dalšího networku vložíme vstupy pro porovnání a nastavíme dobu, kdy bude aktivní mlýnek.
 - e) Stejný časovač využijeme i pro dokončení pohybu pásu po skončení pohybu mlýnku. To zrealizujeme tak, že paralelně ke klopnému obvodu zapojíme 2 porovnávací kontakty a nastavíme hodnotu tak, aby pás pokračoval v pohybu, dokud nedojede do cíle.



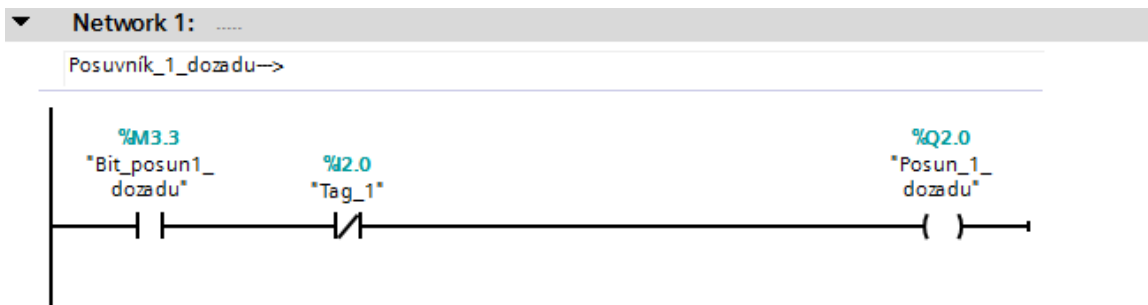
Network 6:

Motor mlýnek

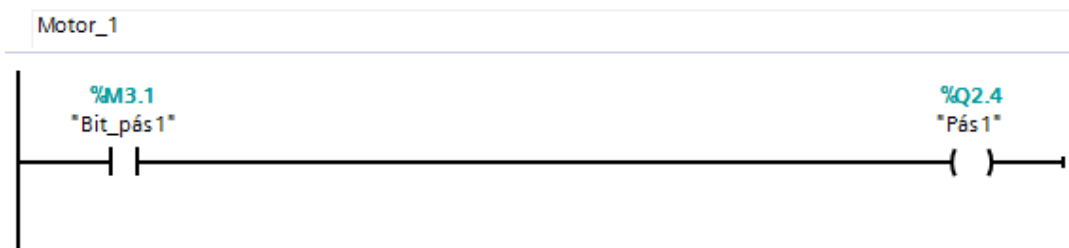


- f) Stejným způsobem ošetříme pohyb vrtáku.
- g) Na přepínače motoru použijeme opět klopné obvody, avšak na vstupy S a R přivedeme koncová tlačítka.

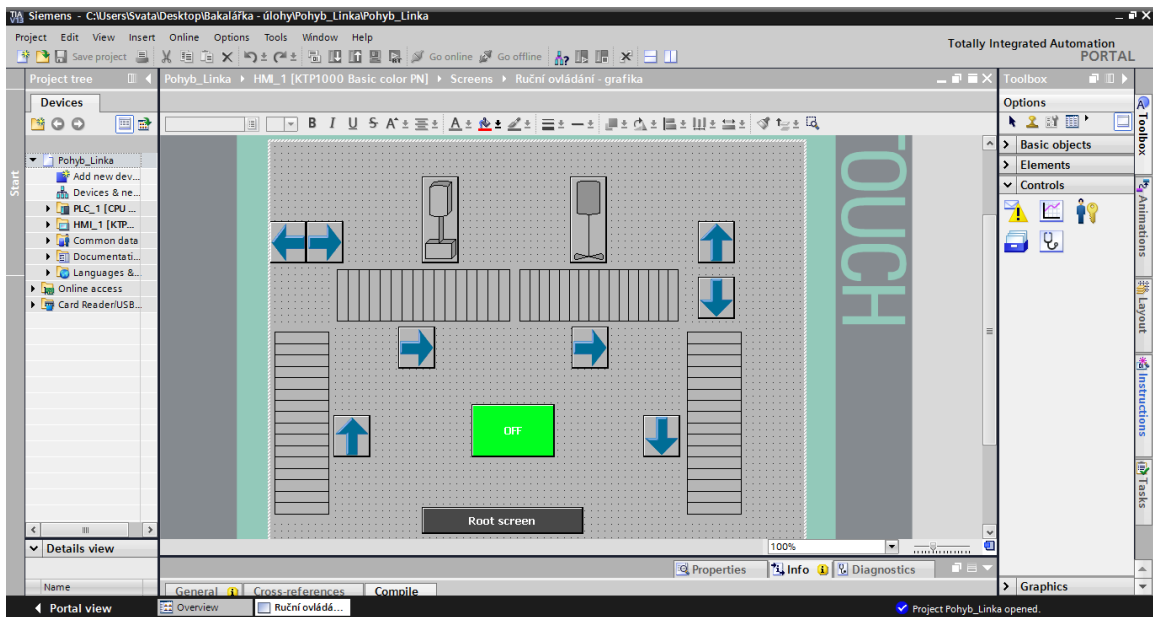
2) Ruční ovládání programujeme stejným způsobem jako v úloze 2.



- Do série zapojíme spínací kontakt, rozpínací kontakt a výstup. První kontakt slouží k nastavení bitu pomocí HMI panelu. Druhý kontakt slouží k ošetření pohybu pomocí koncového spínače tak, aby nám „posuvník“ nevyjel ze své dráhy. Na výstup zapojíme přepínač motoru dopředu či dozadu.
- Stejným způsobem pokračujeme pro všechny přepínače motoru (Q1,Q2,Q3 a Q4).
- Pohyb pásů naprogramujeme tak, že zapojíme do série spínací kontakt a výstup motoru daného pásu.



- Stejným způsobem pokračujeme pro všechny pásy i pro motor mlýnku a vrtáku.
- S HMI panelem pracujeme stejným způsobem jako u úlohy číslo 2.



Screen ručního ovládání výrobní linky

3.4 Úloha 4 – zadání

Zkompletujte všechny tři části stavebnice a modifikujte kód tak, aby výrobní linka byla co nejefektivnější.

4 Závěr

Cílem této bakalářské práce bylo zkompletovat a zprovoznit robotickou sestavu řízenou pomocí PLC řady Simatic. A dále na této sestavě vytvořit 3 laboratorní úlohy. K úlohám je přiložen i návod k řešení, avšak každý student může dané úlohy řešit odlišným způsobem.

Dle mého názoru je kvalita robotických systémů Fischertechnik velmi nízká. Firma nepoužívá kvalitní komponenty, motory dopravníkových pásů a robotického ramena prokluzují, obrobky (=plastové válečky) se často zasekávají na spojích mezi jednotlivými pásy. Další problém nastává v mechanické pevnosti součástek. Například rameno robotické ruky se po vysunutí horizontální osy poměrně ohýbá.

S komponenty od firmy Siemens jsem naprosto spokojený. Jedná se o světovou špičku jak v kvalitě součástek, tak v rozsahu technické podpory. Velkou výhodou při práci se Simaticem je možnost vyhledání odborných rad na fórech firmy Siemens. Další výhodou vidím ve vývojovém prostředí TIA portal, které je přehledné a nabízí velké množství možností jak v programování, tak ve vizualizaci pomocí SCADA systému.

O kvalitě řídicích jednotek Siemens vypovídá také to, že je používají firmy jako Robert Bosch, Continental, Sinop a další podniky po celém světě.

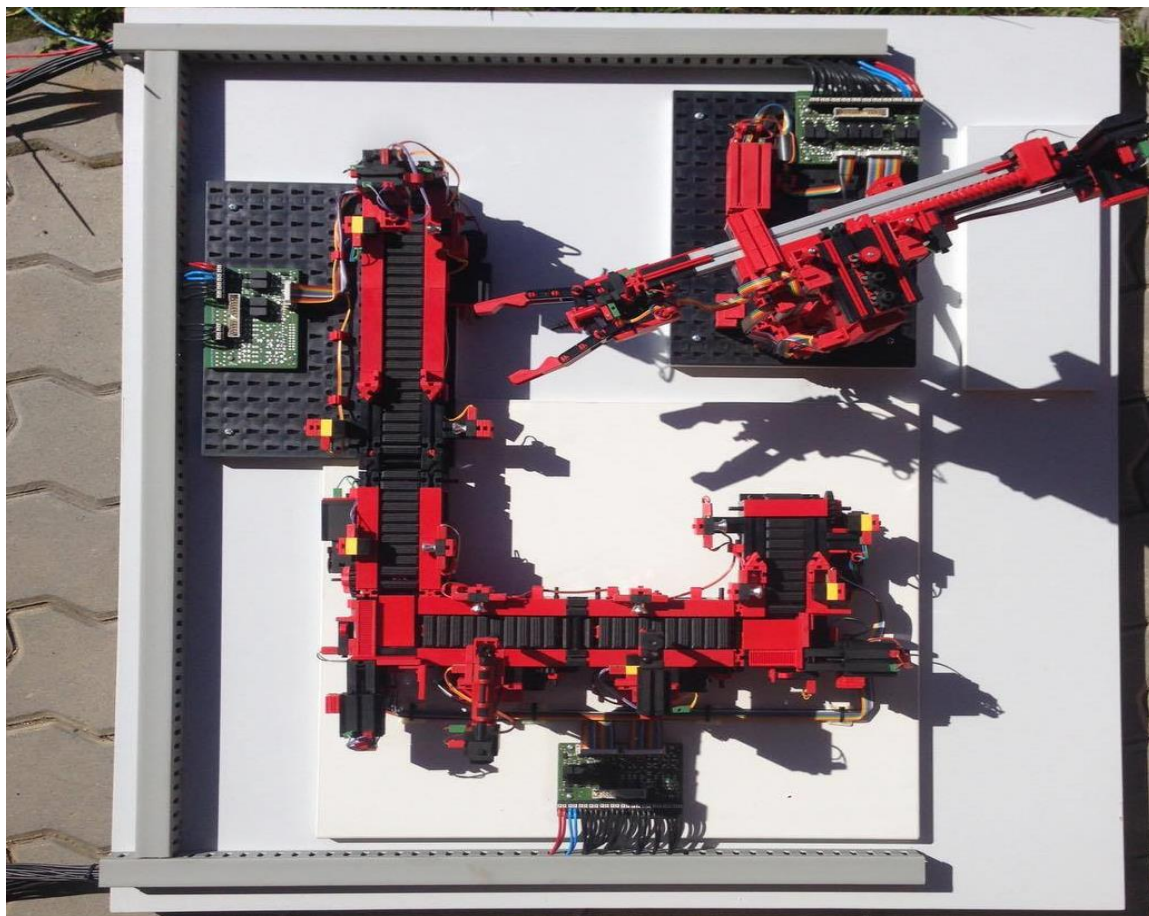
Návody k jednotlivým úlohám jsou psány v jazyce LD, avšak v příloze najdeme programy v jazyce FBD. Dle mého názoru je jazyk LD vhodný pro začínající programátory řídicích jednotek PLC, ale jazyk FBD je celosvětově uznávaný standart. Složitější programy doporučuji rozdělovat na funkční bloky, zvýšíte tím přehlednost celého programu a snadněji najdete chyby, kterých jste se během programování mohli dopustit.

5 Zdroje

- [1] VORÁČEK, Rudolf. *Automatizace a automatizační technika 2* Vyd. 1. Praha: Computer Press, 2000. ISBN 80-7226-247-5.
- [2] Obrázek *Blokové schéma ovládání*
<<https://eluc.kr-olomoucky.cz/verejne/lekce/930>>
- [3] Obrázek *Induktivní snímač pro systém ABS*
<<http://files.h-diag.webnode.cz/200000507-15ba516b44/n%C3%A1boj%20+%20sn%C3%ADma%C4%8D.jpg>>
- [4] Obrázek *Idealizované odezvy základních typů regulátorů*
<biofyzika.upol.cz/userfiles/file/biokybernetika_2_regulace.doc>
- [5] *Programmable logic controller*
<https://en.wikipedia.org/wiki/Programmable_logic_controller>
- [6] *PLC automatizace*
<<http://www.plc-automatizace.cz/knihovna/plc/plc-hw-sestava.htm>>
- [7] Obrázek *Kompaktní PLC easy Control EC4-200*
<<http://www.eatonelektrotechnika.cz/cz/kompaktni-plc-01.html>>
- [8] Obrázek *Modulární PLC Siemens S7-1200 s rozšiřujícími moduly*
<http://w3.siemens.com/mcms/programmable-logic-controller/en/basic-controller/s7-1200/communication/pages/default_vor_tabs.aspx>
- [9] Obrázek *Rozdělení programovacích jazyků dle normy IEC 61131-3*
<http://www.spszl.cz/soubory/plc/programovani_plc.pdf>

- [10] *Lehký úvod do programování PLC v jazyce IL*
<http://vlab.fs.cvut.cz/navody/files/Uvod_programovani_PLC-IL.pdf>
- [11] *Programovací režimy pro PLC dle IEC 61131-3*
<<http://automatizace.hw.cz/programovaci-rezimy-pro-plc-dle-iec-61131-3-codesys>>
- [12] *Obrázek Příklad programovacího jazyka IL pomocí softwaru MULTIPROG Express*
<https://www.phoenixcontact-software.com/files/standard/images/produkte/iec-61131/multiprog%20express/MP50expr_il.jpg>
- [13] *Obrázek Příklad programovacího jazyka ST pomocí softwaru MULTIPROG*
<<https://www.phoenixcontact-software.com/en/iec-61131-control/iec-61131-basics/structured-text-st>>
- [14] *Obrázek Základní prvky jazyka kontaktních schémat*
<<http://automa.cz/res/pdf/45413.pdf>>
- [15] *Obrázek Příklad programu v jazyku kontaktních schémat*
<<http://www.amci.com/tutorials/tutorials-what-is-programmable-logic-controller.asp>>
- [16] *Simatic S7-1200 System manual*
<http://www1.siemens.cz/ad/current/content/data_files/automatizacni_systemy/mikrosystemy/simatic_s71200/manual_s7-1200_2015-01_en.pdf>
- [17] *Regulace (technika)*
<[https://cs.wikipedia.org/wiki/Regulace_\(technika\)](https://cs.wikipedia.org/wiki/Regulace_(technika))>

6 Přílohy



Příloha 1: Fotografie kompletní sestavy



Příloha 2: Uložení kabelů do rozvaděčového kanálu.

Totally Integrated Automation Portal			
Pohyb_pas_funkční / PLC_1 [CPU 1215C AC/DC/Rly] / Program blocks			
Main [OB1]			
Main Properties			
General			
Name	Main	Number	1
Type	OB	Language	FBD
Numbering	automatic		
Information			
Title	"Main Program Sweep (Cy-ckly)"	Author	
Version	0.1	User-defined ID	
Comment		Family	
Network 1:			
Časovač stání			
Network 2:			
Pohyb pásu vpřed. Start pomocí čidla 1. Stop pomocí čidla 2. Časovač pro ošetření startu.			
Network 3:			
Pás pohybu dozadu. Start pomocí cíla 2 & horního vypínače. Stop pomocí čidla 1.			
Network 4:			
Pohyb razítkovače dolů. Start pomocí sestupné hrany senzoru 2. Stop pomocí dolního vypínače.			
Network 5:			
Pohyb razítkovače nahoru. Start pomocí dolního vypínače. Stop pomocí horního vypínače.			

Příloha 3: Kód programu pro úlohu č.1 v jazyce FBD

Příloha 4: Funkční programy k jednotlivým úlohám, psaných jazykem LD pomocí softwaru Simatic STEP7 V12