

**ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ**

KATEDRA TECHNOLOGIÍ A MĚŘENÍ

BAKALÁŘSKÁ PRÁCE

**Vytvoření měřicího pracoviště s frekvenčním měničem,
PLC a převodovkou MODACT pro výuku**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jan KOVAROVIČ**
Osobní číslo: **E10B0459P**
Studijní program: **B2612 Elektrotechnika a informatika**
Studijní obor: **Komerční elektrotechnika**
Název tématu: **Vytvoření měřicího pracoviště s frekvenčním měničem, PLC a převodovkou MODACT pro výuku**
Zadávající katedra: **Katedra technologií a měření**

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

1. Navrhnete mechanické uspořádání pracoviště.
2. Realizujte sestavu pracoviště.
3. Zpracujte vzorové úlohy a programy pro PLC a měnič.



Rozsah grafických prací: **podle doporučení vedoucího**

Rozsah pracovní zprávy: **20 - 30 stran**

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

Seznam odborné literatury:

Student si vhodnou literaturu vyhledá v dostupných pramenech podle doporučení vedoucího práce.

Vedoucí bakalářské práce:

Ing. Petr Preuss, CSc.


Katedra teoretické elektrotechniky

Datum zadání bakalářské práce: **15. října 2012**

Termín odevzdání bakalářské práce: **7. června 2013**


Doc. Ing. Jiri Hammerbauer, Ph.D.
děkan




Doc. Ing. Vlastimil Skočil, CSc.
vedoucí katedry

V Plzni dne 15. října 2012

Abstrakt

Předkládaná bakalářská práce popisuje jednotlivé součásti měřicího pracoviště a jejich funkci. Dále je věnována pozornost programování PLC, konkrétně v jazyce Simple3. V práci je také popsána konstrukce měřicího pracoviště.

Klíčová slova

Měřicí pracoviště, servomotor, frekvenční měnič, PLC, řízení motoru, programování PLC, Simple3

Abstract

The bachelor thesis describes individual parts of the measuring workplace and their function. Attention is also paid to the PLC programming, specifically in the language Simple3. The thesis also describes the design and construction of the workplace.

Key words

Measuring workplace, servomotor, frequency converter, PLC, motor control, PLC programming, Simple3

Prohlášení

Prohlašuji, že jsem tuto diplomovou/bakalářskou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této diplomové práce.

Dále prohlašuji, že veškerý software, použitý při řešení této bakalářské práce, je legální.

.....
podpis

V Plzni dne 7.6.2013

Jan Kovarovič

Poděkování

Tímto bych rád poděkoval vedoucímu bakalářské práce Ing. Petru Preussovi, CSc. za metodické vedení práce.

Seznam symbolů a zkratek

PLC	programmable logic controller, programovatelný logický automat
PO, PZ.....	poloha otevřeno/zavřeno – polohové snímače převodovky
SO, SZ.....	směr otevřeno/zavřeno – směrové snímače převodovky
INC.....	inkrementální snímač otáček převodovky
SP1,SP2,SP3	multifunkční vstupy frekvenčního měniče
REW, FWD.....	řídící vstupy frekvenčního měniče
COM	společná svorka frekvenčního měniče
X0 – Xn.....	binární vstupy PLC, případně binární vstupní proměnné jazyka Simple3
Y0 – Yn.....	binární výstupy PLC, případně binární výstupní proměnné
LED.....	light emitting diode, svítivá dioda

Obsah

ÚVOD	10
1 MĚŘÍCÍ PRACOVIŠTĚ	11
1.1 ÚČEL PRACOVIŠTĚ	11
1.2 PRVKY PRACOVIŠTĚ	12
1.2.1 Elektrický servomotor	12
1.2.2 Frekvenční měnič	13
1.2.3 PLC	16
1.2.4 Ovládací prvky	18
2 PROGRAMOVÉ VYBAVENÍ	19
2.1 JAZYK SIMPLE3	20
2.1.1 Vývojové prostředí HypEd4	24
2.2 ZADÁNÍ SEMESTRÁLNÍ PRÁCE	25
3 MECHANICKÁ KONCEPCE A KONSTRUKCE	26
ZÁVĚR	29
SEZNAM LITERATURY A INFORMAČNÍCH ZDROJŮ	30
PŘÍLOHY	31
PŘÍLOHA 1 – UKÁZKA MOŽNÉHO ŘEŠENÍ SEMESTRÁLNÍ PRÁCE	31
PŘÍLOHA 2 – FOTOGRAFIE PRACOVIŠTĚ	33

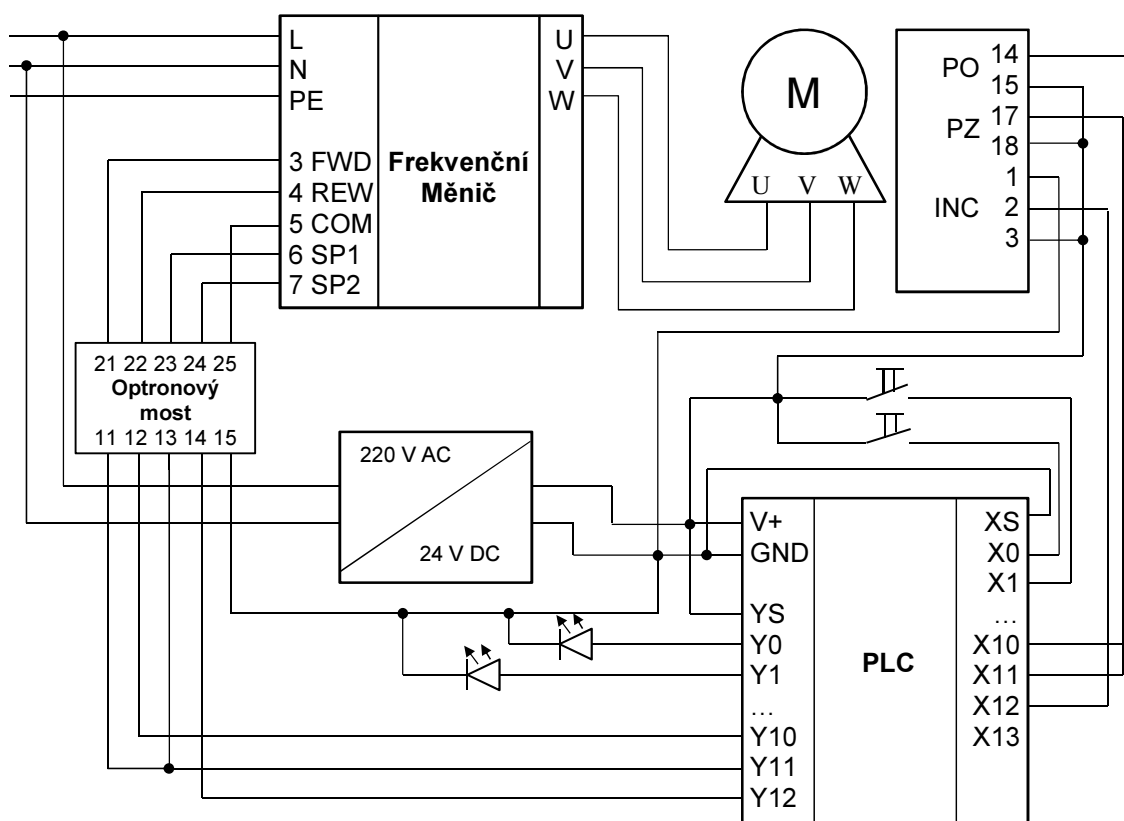
Úvod

Předkládaná práce obsahuje popis vytvořeného pracoviště. A to jak jednotlivých prvků, tak vlastního pracoviště, jeho programového vybavení a stavby pracoviště.

Text práce je rozdělen do tří částí. První část se zabývá teoretickým popisem pracoviště, zde je popsán princip funkce pracoviště a jeho účel. Dále jsou v první kapitole popsány jednotlivé prvky pracoviště, nejprve obecně teoreticky, posléze jejich fungování v rámci pracoviště. Druhá část pojednává o programování PLC, hlavně programování v jazyce Simple3. V této kapitole je také uvedeno zadání semestrální práce pro studenty pracující s tímto pracovištěm. Třetí kapitola popisuje samotnou konstrukci měřicího pracoviště.

1 Měřicí pracoviště

Mým úkolem bylo ze zadaných komponent vytvořit měřicí pracoviště pro výuku, kde budou studenti ovládat servomotor MODACT (viz kap. 1.2.1). Pohon servomotoru zajišťuje třífázový asynchronní motor, jehož rychlost otáčení je regulována pomocí frekvenčního měniče (viz kap. 1.2.2). Převodovka je dále vybavena snímači, které sledují polohu hřídele. PLC (viz kap. 1.2.3) pak na vstupu zpracovává signály z těchto snímačů, na základě těchto údajů a podle svého naprogramování řídí frekvenční měnič.



Obr. 1.1: Schéma zapojení měřicího pracoviště

1.1 Účel pracoviště

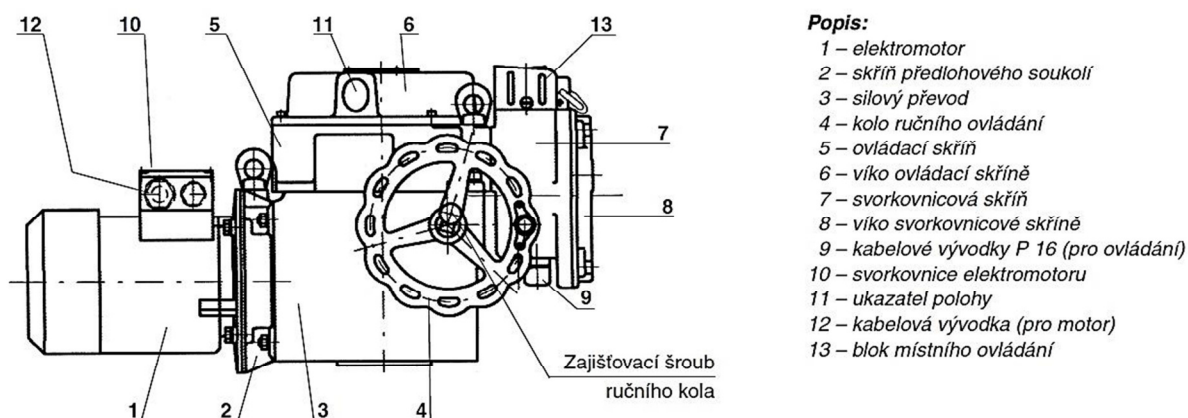
Účelem pracoviště je dát studentům možnost vytvářet řídicí programy a zkoušet jejich funkci na ovládání jednoduchého pohonu. Součástí výuky na tomto pracovišti je také semestrální práce, úkolem studentů bude napsat program podle zadání (viz kap. 2.2). Studenti se tak naučí vytvářet jednoduché programy pro PLC a získají znalosti o ovládání pohonů, konkrétně s asynchronními motory.

1.2 Prvky pracoviště

Pracoviště je zapojeno z těchto prvků: servomotoru s převodovkou, frekvenčního měniče, PLC, vypínacího spínače, dvou tlačítek a dvou signalizačních LED (obr. 1.1).

1.2.1 Elektrický servomotor

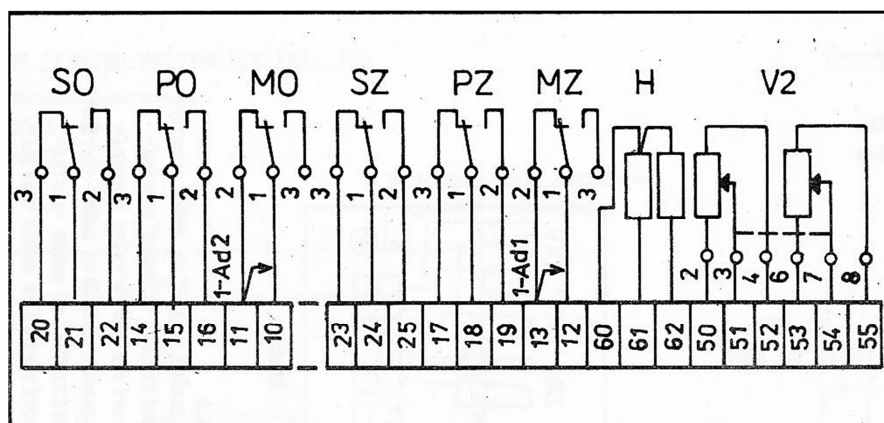
Motorům, které se používají v pohonech s přesně nastavitelnou polohou natočení osy, říkáme servomotory. Nastavování polohy může být buď spojitě, nebo diskretní. Servomotor na tomto pracovišti umožňuje polohování do dvou diskretních poloh, a to „zavřeno“ a „otevřeno“.



Obr. 1.2: Sestava servomotoru (převzato z [2])

MODACT MO 63/110 – 63 je elektrický servomotor otočný, určený pro dálkové ovládání šoupátek a je konstruován k přímé montáži na šoupátko. Třífázový asynchronní motor (1 - obr. 1.2) pohání přes předlohové soukolí (2) silový převod (3), který je umístěn v nosné skříni servomotoru. Výstupní hřídel je napevno připojen k unášeci planetového převodu a prochází do ovládací komory (5), kde jsou umístěny všechny ovládací prvky servomotoru, polohové, signalizační a momentové vypínače a snímače.[1] Činnost vypínačů je přes náhonové mechanismy odvozena od otáčení hřídele, to je ozubenými převody přenášeno k desce se spínači, sepnutí mikrospínačů v dané poloze zajišťují vačky. Tyto spínače jsou celkem čtyři, dva pro určování polohy (poloha „zavřeno“, PZ, a „otevřeno“, PO), které se spínají po natočení hřídele do požadované polohy, a dva pro určování směru otáčení (signalizace „zavírání“, SZ, a „otevírání“, SO). Pro změnu polohy, ve které dojde k sepnutí spínačů, tj. prodloužení či zkrácení pracovního cyklu, se musí ručně upravit nastavení vaček.[1] Pro řízení převodovky jsou v případě tohoto pracoviště využívány jen polohové spínače (PO a SO) a inkrementální snímač otáček (INC). Signálové vodiče můžeme do

svorkovnice ovládací skříně (7) připojit tak, aby sepnutí spínače (např. je-li MODACT v poloze otevřeno, je sepnutý mikropsínač PO) bylo signalizováno spojenými svorkami (log. 1) nebo rozpojenými svorkami (log. 0). V případě spínače PO, vyvedením svorek 14 a 15 (obr. 1.3), bude jeho sepnutí na vstupu PLC signalizováno jako log. 1, pro signalizaci log. 0 by na vstup museli být vyvedeny svorky 15 a 16. Na tomto pracovišti pak bude sepnutí spínačů signalizováno jako log. 1.



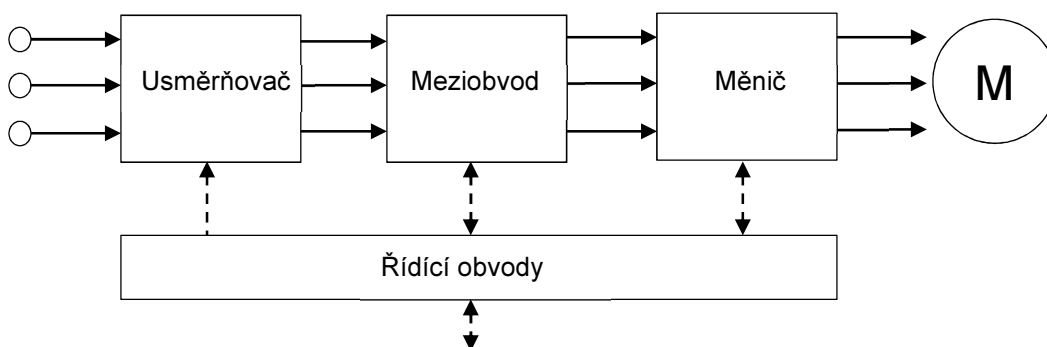
Obr. 1.3: Schéma zapojení svorkovnice ovládací skříně (převzato z [1])

Typové číslo:	52034.2523	Výrobní číslo:	0340491
Napětí:	380/220 V	Výkon motoru:	3 kW
Ot/min:	63	Max. toč. Moment:	630 Nm

Tab. 1.1: Technické specifikace převodovky (zdroj [1 a štítek motoru])

1.2.2 Frekvenční měnič

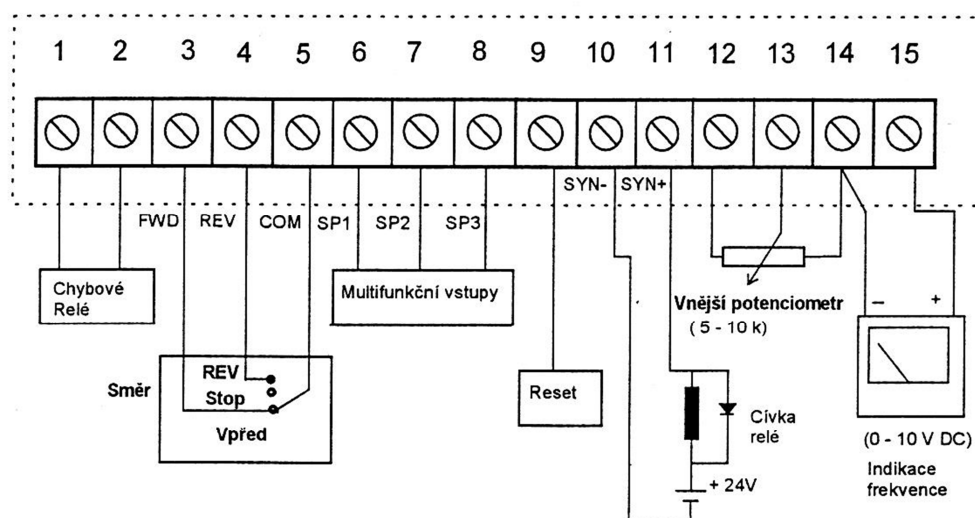
Frekvenční měnič se používá ke změně síťového napětí, které má konstantní amplitudu a frekvenci, na napětí s proměnnou amplitudou a frekvencí. Výhodami regulace elektrických motorů pomocí měničů jsou úspora energie, možnost přesné regulace otáček a nižší provozní náklady vzhledem k menšímu mechanickému opotřebování motoru.



Obrázek 1.4: Blokové schéma frekvenčního měniče (zdroj [3])

Pro popis můžeme frekvenční měnič rozdělit na čtyři části - usměrňovač, střední část (meziobvod), silovou část (střídač) a řídicí obvody (*obr. 1.4*). Usměrňovač převádí střídavé napětí z jedno- nebo třífázové sítě na pulzující stejnosměrné napětí. Usměrňovač může být sestaven buď pomocí diod, potom se jedná o neřízený usměrňovač, nebo tyristorů, který se nazývá plně řízený usměrňovač. Případně může být usměrňovač sestaven z kombinace diod i tyristorů, v tom případě jde o polořízený usměrňovač. Meziobvod dále zpracovává výstupní napětí usměrňovače pro samotný měnič. Využívány jsou následující tři typy meziobvodů, a to: Meziobvod měnící napětí usměrňovače na stejnosměrný proud, meziobvod stabilizující a vyhlazující pulzující napětí, a meziobvod měnící konstantní napětí usměrňovače na proměnné napětí. Použitý typ meziobvodu závisí na druhu usměrňovače a na měniči, který následuje za meziobvodem. Silová část, měnič, na svém výstupu dodává střídavé napětí o požadované frekvenci. Způsoby, jakými se dosahuje požadovaného výstupu, jsou různé, ale konstrukce měničů je ve všech případech podobná. Hlavní složkou jsou řízené polovodiče, nejčastěji IGBT tranzistory, spojené po dvojicích ve třech větvích. Nejrozšířenější způsob generování střídavého napětí s požadovanou frekvencí je pulzně šířková modulace, zkratka PWM z anglického Pulse Width Modulation, při kterém se střídavě spínají zmíněné polovodiče. Výsledné napětí závisí na šířce a frekvenci těchto pulzů. Poslední částí jsou řídicí obvody, které přijímají a odesílají informace do ostatních částí zařízení, např. podle požadavku obsluhy ovlivňují spínání polovodičů měniče a tak generují požadovanou frekvenci. Řídicí obvody také zajišťují regulaci výstupu, aby byl zajištěn správný běh motoru (např. konstantní otáčky při změnách momentu, přesné řízení otáček, atd.), způsobů regulace také existuje několik. Můžeme je rozdělit do dvou skupin, s otevřenou nebo uzavřenou regulační smyčkou. Systémy s otevřenou smyčkou nemají žádnou zpětnou vazbu na skutečný počet otáček a moment na hřídeli motoru. Regulace je založena na matematickém modelu, který popisuje chování motoru, přesnost takovéto regulace je 0,1 - 5% v závislosti na použitém modelu. Systémy s uzavřenou smyčkou jsou vybaveny čidlem, které snímá počet otáček na hřídeli nebo proud a napětí v motoru. Můžeme tak porovnávat požadavek se skutečností a tím řídit motor mnohem přesněji, odchylka je v tomto případě 0,01 – 0,05%. Pro řízení motoru na pracovišti je použita U/f regulace, jedná se o regulaci v otevřené smyčce, kdy napětí a frekvence rostou lineárně. Tento způsob regulace nerespektuje dynamické jevy v motoru a je tak vhodný pouze pro jednoduché aplikace, jako jsou čerpadla. Typická přesnost této regulace je 1 – 5%.[3]

Frekvenční měnič TAIAN N2 – 202 – M byl vybrán s ohledem na to aby, mohl být napájen jednofázově ze sítě 220V/50Hz. Bohužel měniče, které toto umožňují, jsou oproti ovládanému motoru výkonově poddimenzované. Vzhledem k tomu, že k motoru nebude připojena žádná zátěž, ten poběží na prázdko, nebude nižší výkon měniče představovat problém. Měnič využívá jako silový prvek IGBT tranzistory a jeho výstupní napětí je generováno pulzní šířkovou modulací a výstupní proud se blíží sinusovému průběhu, frekvenční rozsah měniče je 0 – 400Hz. Frekvenční měnič bude předem nakonfigurován pro dálkové ovládání pomocí PLC, které bude připojeno k řídicí svorkovnici TM2 (obr. 1.5).



Obr. 1.5: Řídicí svorkovnice frekvenčního měniče (převzato z [4])

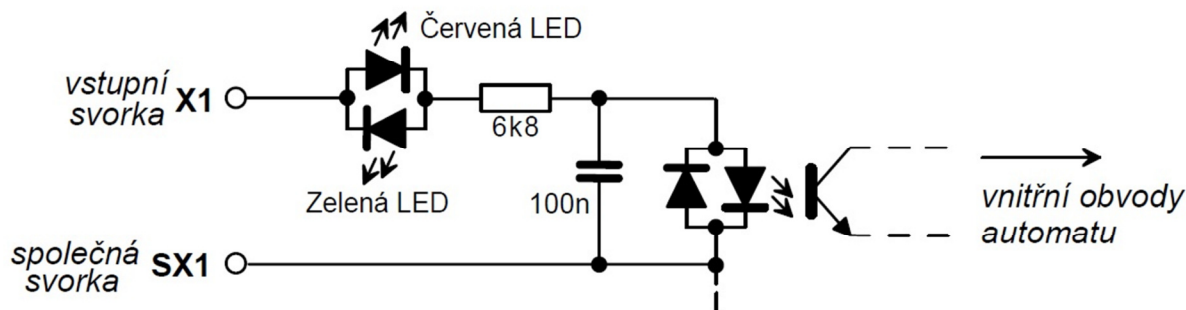
K řízení frekvence budou sloužit multifunkční vstupy SP1 – SP3 (svorky 6,7 a 8), ke každé z celkem osmi kombinací sepnutí těchto vstupů je pak v měniči nastavena jiná frekvence. Kombinaci, kdy jsou všechny vstupy odpojeny, je přiřazena tzv. referenční master frekvence; jedná se základní frekvenci, na kterou se měnič vždy nastaví, pokud nedostane jiný příkaz.[4] Pracoviště bude využívat jen dvě frekvence, pro pomalý chod a pro rychlý chod, takže budou zapojeny dva multifunkční vstupy (pomalý chod pro sepnutý vstup SP1 a rychlý chod pro sepnuté SP1 a SP2 zároveň). Běh motoru budou řídit vstupy FWD a REV (svorky 3 a 4), vstupem FWD (3) se bude řídit rozběh a zastavení motoru, log. 1 pro běh a log. 0 pro zastavení motoru, a vstupem REV (4) zapínat reverzace, log. 1 pro zpětný chod. Vstup FWD bude společně se vstupem SP1 připojen ke stejné výstupní svorce PLC, takže pokyn k běhu motoru zároveň nastaví nízkou rychlost otáčení. Všechny tyto vstupy (3. 4. 6, 7, 8) jsou spínány proti společné svorce COM.

1.2.3 PLC

PLC, z anglického programmable logical controller, česky programovatelný logický automat, je číslicový počítač pro řízení elektromechanických procesů. PLC původně vznikly jako náhrada za systémy reléové logiky a dnes jsou široce využívány v celém průmyslu. Protože jsou použity převážně k řízení strojního vybavení, je jejich konstrukce odolná proti nepříznivým vlivům prostředí, jako je prach, vysoká či nízká teplota, elektrický šum, vibrace a rázy. Jejich hlavní výhodou oproti mikrokontrolerům a jednoúčelovým počítačům je variabilita. Jsou totiž vybaveny řadou digitálních i analogových vstupů, jejichž množství je možno zvýšit dalšími moduly, a jsou snadno přeprogramovatelné.

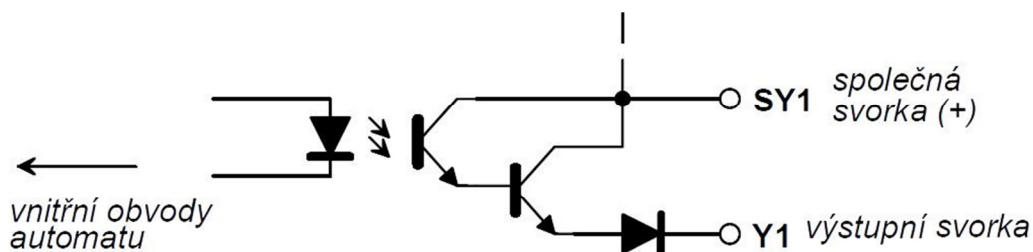
PLC pracují v tzv. „tvrdém real-time“ režimu, což znamená, že je kladen velký důraz na rychlost operací. Doba, za kterou musí být program vykonán, může být v závislosti na aplikaci omezená a pokud by prodleva mezi načtením vstupů a odesláním výstupů byla příliš dlouhá, může dojít k problémům. Program nahraný v paměti PLC se opakuje, dokud je automat zapnut, doba jednoho opakování je obvykle několik milisekund. Na začátku programové smyčky jsou zkontrolovány stavy vstupů, tyto hodnoty jsou nahrány do vyhrazené části paměti. Poté je spuštěn vlastní program, který pracuje s hodnotami vstupů a do paměti zapisuje nové hodnoty výstupů. Poté co byl vykonán celý program, jsou hodnoty z výstupního registru odeslány na fyzické výstupy. Další opakování programové smyčky je opět zahájeno přečtením hodnot vstupů.[5] Programy pro PLC jsou běžně tvořeny na osobních počítačích. Software potřebný pro tvorbu programů a komunikaci automatu s počítačem (viz kap. 2) zpravidla dodává výrobce konkrétního zařízení, proto jsou programy, i přes stejný základ, napsané na jedno zařízení nepřenositelné na zařízení jiného výrobce nebo i typu (problémem je např. jiná adresace paměti, proprietární funkce, atd.). Hardwarově je komunikace zajištěna v PLC sériovým portem buď RS – 232, RS – 485, RS – 422 nebo, u novějších modelů, připojením k síti ethernet. PLC může být také vybaveno vlastním uživatelským rozhraním, tedy klávesnicí nebo tlačítky a zobrazovací jednotkou, rozhraní může sloužit např. k zobrazování zpráv nebo k editaci programu.

PLC na pracovišti má označení AnneX-F a jeho výrobcem je firma Hypel. Je napájeno z 24V zdroje, odběr proudu je při tomto napětí 30mA. Automat je vybaven patnácti binárními vstupy, šestnácti binárními výstupy, osmi analogovými vstupy a dvěma analogovými výstupy.



Obr. 1.7: Binární vstupy (převzato z [6])

Komunikace s počítačem je zajištěna po lince RS – 485. Binární vstupy (obr 1.6) jsou zapojeny se společnou svorkou (SX) a vstupními svorkami (X0 – X14) a jsou galvanicky odděleny optickými členy od vnitřních obvodů. Vstupy jsou bipolární a jejich aktivita je signalizována dvoubarevnou LED, teče-li proud do vstupu, tj. napětí na vstupní svorce Xn je vyšší než na společné svorce, svítí červená LED, v opačném případě, proud teče ze vstupu, svítí zelená LED. Pro napětí na X proti SX v rozmezí 0 – 2V je na vstupu detekována log. 0, pro log. 1 musí být napětí mezi svorkami větší než 10V. Na binární vstup PLC budou přivedeny signály z čidel převodovky a ovládacích tlačítek. Binární výstupy (obr 1.7) jsou



Obr. 1.6: binární výstup PLC (převzato z [6])

tranzistorové, v zapojení se společným kolektorem, galvanicky oddělené a chráněny proti přepólování. Při log. 0, vypnutý výstup, je mezi společnou svorkou SY a výstupní svorkou Yn pracovní napětí 30V. Při log. 1, sepnutý výstup, dochází k úbytku napětí, který je maximálně 2V a svorkami protéká proud, maximální spínaný proud je 200mA. Na binární výstup budou připojeny signalizační diody a řízení frekvenčního měniče. Protože jsou vstupy měniče spínány napěťově, kdežto výstupy PLC proudově, musíme výstup automatu a vstup měniče oddělit pomocí optronů. Analogové vstupy ani výstupy nebudou zapojeny.[6]

Horní řada svorek			Dolní řada svorek		
1	V+	Kladný pól napájení	25	I0	Analogový vstup I0
2	GND	Záporný pól napájení	26	I1	Analogový vstup I1
3	LB	Vstup linky RS485 – vodič B	27	I2	Analogový vstup I2
4	LA	Vstup linky RS485 – vodič A	28	I3	Analogový vstup I3
5	GND	POWER – zem napájení	29	I4	Analogový vstup I4
6	O0	Analogový výstup O0	30	I5	Analogový vstup I5
7	O1	Analogový výstup O1	31	I6	Analogový vstup I6
8	SY	Společný kolektor výstupů	32	I7	Analogový vstup I7
9	Y0	Emitor výstupu Y0	33	SX	Společná svorka vstupů
10	Y1	emitor výstupu Y1	34	X0	Bipolární vstup X0
11	Y2	emitor výstupu Y2	35	X1	Bipolární vstup X1
12	Y3	emitor výstupu Y3	36	X2	Bipolární vstup X2
13	Y4	emitor výstupu Y4	37	X3	Bipolární vstup X3
14	Y5	emitor výstupu Y5	38	X4	Bipolární vstup X4
15	Y6	emitor výstupu Y6	39	X5	Bipolární vstup X5
16	Y7	emitor výstupu Y7	40	X6	Bipolární vstup X6
17	Y8	emitor výstupu Y8	41	X7	Bipolární vstup X7
18	Y9	emitor výstupu Y9	42	X8	Bipolární vstup X8
19	Y10	emitor výstupu Y10	43	X9	Bipolární vstup X9
20	Y11	emitor výstupu Y11	44	X10	Bipolární vstup X10
21	Y12	emitor výstupu Y12	45	X11	Bipolární vstup X11
22	Y13	emitor výstupu Y13	46	X12	Bipolární vstup X12
23	Y14	emitor výstupu Y14	47	X13	Bipolární vstup X13
24	Y15	emitor výstupu Y15	48	X14	Bipolární vstup X14

Tab. 1.2: Svorky PLC (zdroj [6])

1.2.4 Ovládací prvky

Posledními prvky pracoviště jsou ovládací a signalizační prvky. Jde o hlavní vypínač, dvě signalizační LED a dvě ovládací tlačítka. Hlavní vypínač vypíná proud do frekvenčního měniče a tím pádem do celého pracoviště. Na místo hlavního vypínače byl použit dvupólový kolébkový přepínač dimenzovaný na proud 15A. Signalizační LED jsou červená a zelená. Napětí na LED v propustném směru je 2V pro červenou a 2.2V pro zelenou, odběr proudu je u obou LED cca 20mA. LED budou připojeny na binární výstup PLC. Dvojice spínacích tlačítek bude připojena na vstup PLC.

2 Programové vybavení

Programové vybavení PLC AnneX – F se skládá ze dvou vrstev. První vrstvou je firmware, umístěný výrobcem v pevné paměti automatu, který řeší veškeré vazby na obvodové vybavení PLC. Druhou vrstvou je pak konkrétní uživatelský program, který byl překladačem přeložen do strojového kódu a nahrán do paměti PLC. Firmware automatu se spouští automaticky po zapojení napájení, načež čeká přibližně 1 – 2 sekundy na komunikaci po lince RS – 485 z počítače. Pokud během této doby nedojde k žádnému pokusu o komunikaci, zahájí automat přípravy ke spuštění programu. Nejprve zkontroluje, zda se uživatelský program nachází v paměti PLC, poté ověří jeho kontrolní součet a zda je pro program povoleno automatické spuštění. Proběhnou-li tyto tři kroky úspěšně, dojde ke spuštění programu. Poté co program vykoná poslední instrukci, převezme řízení automatu opět firmware, který přepíše fyzické výstupy a načte vstupy, následuje opětovné vykonání programu atd.[7]

Aby bylo možno programy pro jakékoliv PLC vytvářet a následně je nahrát do jeho paměti, je zapotřebí k tomu určený software. Jak bylo zmíněno v předchozí kapitole, software k PLC je dodáván výrobcem daného automatu, obvykle jako součást jednoho balíčku. Jedná se především o komunikační software a vývojové prostředí. Hlavním úkolem komunikačního softwaru je nahrání uživatelského programu, k jehož vytvoření bylo využito vývojové prostředí, do paměti PLC. Kromě vývojového prostředí a softwaru pro komunikaci může výrobce poskytovat také simulační program, pro bezpečné testování funkčnosti uživatelského programu, nebo nástroje pro diagnostiku. Ty mohou být například integrovány ve vývojovém prostředí nebo dodávány jako samostatný software.

Programy pro PLC mohou být tvořeny pomocí několika typů jazyků, které jsou definovány normou IEC 61 131 – 3, jedná se o třetí část osmidílného standartu IEC 61 131 pro PLC. V ČR byla tato norma převzata, ale nepřeložena, jako ČSN EN 61 131 – 3. V současné době norma definuje pět jazyků, tři grafické a dva textové. Mezi grafické jazyky patří jazyk příčkového diagramu (LD – ladder digram), který vychází z reléové logiky a její grafické interpretace, a je určený převážně ke zpracování booleovských signálů. Dalším je jazyk funkčního blokového schématu (FBD – function block diagram), který je založen na propojování funkčních bloků a funkcí, na rozdíl od LD může přenášet i nebooleovské hodnoty. Posledním grafickým jazykem je jazyk sekvenčního funkčního diagramu (SFC –

sequential function chart). Jedním ze dvou textových jazyků je jazyk seznamu instrukcí (IL – instruction list), nízkourovňový jazyk typu assembler. Druhý je jazyk strukturovaného textu (ST – structured text), programovací jazyk vyšší úrovně, který je odvozen od jazyků C a Pascal, obsahující prvky moderních programovacích jazyků jako větvení a iterační cykly. Pomocí jazyka ST lze definovat komplexní funkční bloky, které je možno využívat i v jiných programovacích jazycích.[8] Programovacích jazyků jako takových existuje mnohem více (opět se jedná o záležitost výrobce automatu), ale všechny plní požadavky na standartu. Například bude-li programátor programovat PLC určitého výrobce jazyce typu LD a potom přejde k programování automatu jiného výrobce, ale také v LD. Tak přestože se bude jednat o odlišné, vzájemně nekompatibilní jazyky, tak princip programování zůstane stejný.

2.1 Jazyk Simple3

Programovací jazyk Simple3, určený k programování PLC a jiných řídicích systémů vyráběných firmou Hypel, je jazykem strukturovaného textu. Tento jazyk, v rozporu s informací uvedenou o podkapitolu výše (možnost větvení a cyklů u ST jazyků) nepodporuje tvorbu iteračních cyklů. Cykly jsou v tomto programovacím jazyce záměrně zakázány, aby nemohlo docházet k tzv. „zatužení“ programu, kdy uživatelský program zůstane v nekonečném cyklu a nedojde k předání řízení zpět firmwaru. Zpracování uživatelského programu pak probíhá ve třech krocích. Vytvoření zdrojového kódu, a to buď v integrovaném vývojovém prostředí dodávaného firmou Hypel (viz kap. 2.1.1) nebo v jakémkoliv textovém editoru např. Notepad. Druhým krokem je překlad zdrojového kódu pomocí překladače. Po úspěšném překladu dojde k vygenerování souboru *.DNL, který obsahuje kód pro automat. Posledním krokem je natažení obsahu tohoto souboru do paměti PLC, to se provede buď pomocí programu LOADER3 nebo z menu vývojového prostředí. Fyzicky je natažení provedeno po již několikrát zmíněné sériové lince. Tvorba uživatelského programu probíhá stejně jako u jiných jazyků strukturovaného textu, posloupnost direktiv, deklarácí, definicí a příkazů jazyka. Direktivy jsou ve zdrojovém textu napsány jako první a modifikují způsob překladu textu. Určují např. pro jaký typ PLC bude zdrojový text překládán nebo odkazují překladač na další soubory připojené k hlavnímu programu. Překlad zdrojového textu pro řídicí jednotku pracoviště upřesňuje direktiva `#Configuration "ANNEX F"`. Deklarace zajišťují vytvoření místa v paměti pro objekt daného druhu a názvu, typicky se jedná o deklaráce proměnných, vlastnosti objektu jsou určeny definicemi, např. přiřazení hodnoty do proměnné. Informace o objektu a adresu přiděleného místa v paměti překladač bezprostředně

nezapíše do kódu pro automat, ale uloží do svých tabulek. Tělo programu je tvořeno posloupností příkazů a programových struktur, každý příkaz je ukončen středníkem. Zdrojový text může také být pro větší přehlednost doplněn komentáři, které jsou od textu programu odděleny uvozovacími znaky (překladač potom tyto části přeskakuje). Mohou být dvojího druhu – řádkové, uvozené znaky „/“, nebo strukturované, ohraničené složenými závorkami „{“.[7]

Výše zmiňované objekty jsou v případě jazyku Simple3 konstanty, proměnné, procedury a funkce. Deklarace objektu je uvozena klíčovým slovem podle druhu objektu a ukončena klíčovým slovem **END** nebo **RETURN**. Při jejich vytváření jim uživatel přiřazuje identifikátor, tzn. pojmenování objektu, ten nesmí být shodný s žádným dříve použitým ani s identifikátory předdefinovaných objektů, při porovnávání nejsou rozlišována velká a malá písmena. Pojmenování objektu musí začínat písmenem, dále mohou následovat číslice, písmena bez diakritiky nebo znak „_“, délka identifikátoru nesmí přesáhnout osmdesát znaků. U konstant, proměnných a funkcí uživatel definuje také jejich datové typy. Tento jazyk rozeznává čtyři datové typy, jde o:

- **BIT**: jednobitová logická informace o dvou hodnotách 1 a 0
- **WORD**: 16 – bitové kladné celé číslo v rozsahu 0 až 65535
- **INT**: 16 – bitové celé číslo v rozsahu -32768 až 32767
- **REAL**: 32 – bitové reálné číslo, rozsah hodnot $\pm 3,4 \cdot 10^{38}$ až $\pm 1,5 \cdot 10^{45}$ a 0
- variace typů **BIT** a **WORD**: **IN_BIT** a **IN_WORD** určené pouze pro čtení

Klíčovým slovem pro deklaraci konstant je **CONST**, po něm následuje definice jednotlivých konstant, oddělených středníkem, deklarace je ukončena klíčovým slovem **END**. V definici je nejdříve zapsána hodnota konstanty, které může být datového typu **WORD**, **INT** nebo **REAL**, následuje znak „#“ a identifikátor.[7] Zápis tedy vypadá např. takto:

```
CONST
3,14 # PI;
10 # Pocitej_Do;
END
```

Deklarace proměnných je uvozena slovem **VAR** a ukončena rovněž slovem **END**. V definici je nejprve zapsán identifikátor proměnné, poté znak „:“ a její datový typ, jednotlivé definice jsou opět odděleny středníkem. Simple3 také obsahuje předdefinované proměnné, které reprezentují fyzické vstupy a výstupy, speciální funkční bity, „vnitřní relé“, obecné

proměnné a funkční registry automatu, jejichž deklarace je součástí překladače. Pro binární vstupy a výstupy jsou definovány identifikátory X0 – X255 (IN_BIT) a Y0 – Y255 (BIT), I0 – I63 (IN_WORD) a O0 – O63 (WORD) pro analogové. B0 – B127 (BIT) jsou speciální funkční bity a M0 – 127 (BIT) „vnitřní relé“. Identifikátory pro obecné proměnné jsou D0 – D63 (WORD) a W0 – W255 (WORD) pro funkční registry. Pro lepší orientaci v programu je těmto proměnným možno přiřadit další identifikátor, definice je ve tvaru původní identifikátor, znak „#“, nový identifikátor, „:“, datový typ.[7] Příklad zápisu:

```
VAR
bin1, bin2      :      BIT
pocet           :      INT
X0              #      tlacitko1      :      BIT
Y0              #      cervena_LED    :      BIT
END
```

Procedury a funkce, umožňují opakovaně používat určitou část programu, výhodou jejich používání je menší paměťová náročnost programu a přehlednější zdrojový text. Procedura je deklarována klíčovým slovem **PROCEDURE** nebo **SUBROUTINE** a funkce slovem **FUNCTION**. Za klíčovým slovem následuje identifikátor funkce nebo procedury, po něm jsou v závorkách uvedeny parametry. Rozdíl mezi procedurou a funkcí je, že funkce má návratovou hodnotu, u funkce je tedy nutné definovat její datový typ. Tělo procedury či funkce tvoří posloupnost příkazů a programových struktur, obě jsou ukončeny klíčovým slovem **RETURN**. Při volání jejich se uvede její identifikátor a dosadí parametry, návratová hodnota funkce je předávána přiřazovacím příkazem (viz níže).[7] Příklad zápisu:

```
PROCEDURE Napr (a : INT; b : BIT)  FUNCTION Napr2 (a,b : INT)
<tělo procedury>                  <tělo funkce>
RETURN                             RETURN
...                                 ...
Napr(10, bin1);                    pocet = Napr2(a,b);
```

Součástí jazyka Simple3 jsou také speciální funkce pro práci časovačem, čítačem, volbu rychlosti systému, A/D převodník pro obsluhu analogových vstupů a výstupů, atd. Tyto funkce pracují se speciálními funkčními bity a funkčními registry. Pro pracoviště je důležitá funkce čítače, ten danou rychlostí zvyšuje hodnotu 16 – bitového registru a je ovládán následujícími proměnnými: **TEN** – povoluje čítač, pro hodnotu 1 je čítač povolen; **TPA** – nastavuje rychlost čítání, 0 pro inkrementaci čítače každou milisekundu a 1 pro zvýšení každou sekundu; **TOE** – nastavuje chování čítače při přetečení, pro 0 se čítač po dosažení konečné hodnoty zastaví, pro 1 se vrátí zpět na výchozí hodnotu; **TDM** – nastavuje směr čítání, pro 0 čítač čítá nahoru, pro 1 směrem dolů.

Základní příkazy v tomto jazyce jsou přiřazení a nastavení. Přiřazení přiřazuje do proměnné na levé straně příkazu hodnotu výrazu na pravé straně, např:

```
pocet = a+b;           nebo           bin2 = a > b;
```

Nastavení slouží pro práci s bitovými proměnnými, tímto příkazem nastavujeme do bitové proměnné hodnotu 1 nebo 0, nebo její negaci. Nastavení hodnoty 1 do bitové proměnné se provede uvedením identifikátoru proměnné, nastavení hodnoty 0 napsáním identifikátoru s apostrofem „ ‘ “. Negování bitové proměnné se provede napsáním vykřičníku „!“ před identifikátorem proměnné,[7] tj.:

```
bin1; - pro nastavení 1           bin1'; - pro nastavení 0           !bin1; - negace proměnné
```

Programové struktury slouží k větvení programu a tvorbě cyklů, jak bylo řečeno, jsou cykly zakázané. Jazyk Simple3 využívá programové struktury IF a CASE. Struktura IF větví program na základě splněné podmínky, začíná klíčovým slovem IF a končí klíčovým slovem ENDIF a můžeme jí psát několika způsoby, a to pomocí klíčových slov: IF – THEN, IF – THEN – ELSE a IF – THEN – ELSIF – THEN – ELSE. Struktury IF lze dále vnořovat do sebe. Syntaxe může být následující:

```
IF <podmínka>
THEN <příkazy> - jsou vykonány v případě splnění podmínky
ELSEIF <druhá podmínka>
THEN <příkazy>
ELSE <příkazy> - jsou vykonány v případě nesplnění ani jedné podmínky
ENDIF
```

Struktura CASE, uvedena klíčovým slovem CASE a ukončena slovem ENDCASE, větví program na základě hodnoty celočíselného výrazu. K možným hodnotám tohoto výrazu jsou přiřazeny příkazy, které se vykonají v případě, že celočíselný výraz těchto hodnot nabyde. Součástí této struktury je také příkaz pro případ, že celočíselný výraz nenabyde žádných z určených hodnot.

```
CASE <celočíselný výraz nebo proměnná> -
OF <hodnota celočíselného výrazu> THEN <příkazy>
ELSE <příkazy> - vykonané v případě, že nedojde ke shodě
ENDCASE
```

Jednoduchý program, s užitím jednoduché programovací struktury IF – THEN, by potom vypadal takto:

```
{při držení tlačítka bliká LED frekvencí 1 Hz}
#Configuratin „ANNEX F“
```

```

VAR
X0 # tlacitko : BIT; // definování proměnných
Y0 # led : BIT;
END

```

```

IF RESET THEN TEN0'; TPA0'; TDM'; led'; T0 = 0; //počáteční inicializace, jedná – li
se o první spuštění programu po zapnutí PLC
ENDIF

```

```

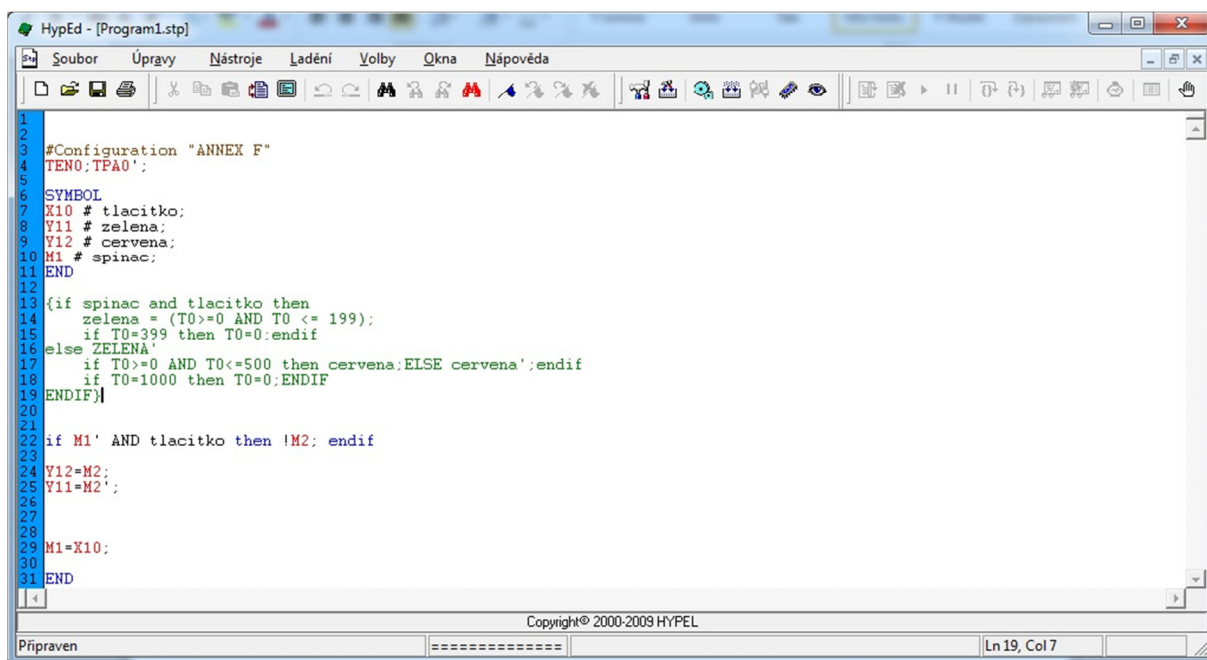
TEN0 = tlacitko; // při držení tlačítka je spuštěn čítač
IF (TEN' OR T0 > 999)
THEN T0 = 0; // nulování časovače při jeho neaktivitě nebo po jedné sekundě
ENDIF
led = (T0 > 499); // LED svítí pro druhou polovinu vteřinové periody
END

```

Vykonání takto jednoduchého programu trvá automatu několik málo milisekund, u složitějších programů musíme počítat s tím, že vykonání programu nějakou dobu trvá a respektovat tuto vlastnost při psaní zdrojového textu. Dále by měl mít uživatel na paměti základní princip funkce automatu. Při běhu programu se nemění vstupy ani výstupy, teprve po ukončení programu se řízení automatu chopí firmware, který aktualizuje hodnoty výstupů a načte nové vstupy.

2.1.1 Vývojové prostředí HypEd4

Vývojové prostředí HypEd4 je určeno speciálně pro programování v jazyce Simple3 a práci s automaty společnosti Hypel. Toto vývojové prostředí umožňuje jak tvorbu a editaci



Obr. 2.1: Vývojové prostředí HypEd4

zdrojových textů, což je samozřejmost, tak i jejich kompilaci, překlad a následné nahrání zdrojového kódu do paměti PLC. Do vývojového prostředí je také integrován simulátor automatů, který v kombinaci s dalšími nástroji pro ladění programu, jako jsou např. breakpointy nebo sledování proměnných, umožňuje uživateli program co nejlépe optimalizovat a zbavit chyb předtím než je v automatu uveden do chodu. Alternativou k tomuto vývojovému prostředí je použití ke tvorbě a editaci zdrojového textu libovolného textového editoru a ke kompilaci a nahrání kódu do PLC pak speciální nástroje, určené pro operační systém MS – DOS, které jsou společností Hypel poskytovány zdarma a jsou volně šiřitelné. Avšak tato metoda je mnohem méně komfortní a neumožňuje tak dobré odladění programu jako vývojové prostředí se simulátorem.

2.2 Zadání semestrální práce

Jak bylo řečeno v kapitole 1.1, součástí výuky na tomto pracovišti bude také semestrální práce. Její zadání je pak napsáno zde. Vzorové řešení této práce je pak uvedeno v příloze A.

V jazyce Simple3 napište program pro řídicí automat AnneX, který zajistí následující činnosti: Po stisku tlačítka se rozběhne pracovní cyklus, ve kterém se hřídel převodovky otočí z polohy „zavřeno“ do polohy „otevřeno“ a zpět, v koncové poloze „otevřeno“ zůstane převodovka stát přesně 2 sekundy. Hřídel se v obou směrech nejprve otáčí maximální rychlostí, poté zpomalí a dojíždí do koncové polohy. Dobu, po kterou se hřídel otáčí maximální rychlostí, nastavte tak, aby dojezd do koncové polohy trval přibližně 1 až 2 sekundy.

Dále řešte:

- 1) Hřídel převodovky se po zapnutí napájení, nebo za jiných okolností, nemusí nacházet ve výchozí poloze, je tedy třeba hřídel natočit do výchozí polohy.
- 2) Nespouštějte cyklus stavem tlačítka, ale změnou jeho stavu (derivace signálu)
- 3) Vstup tlačítka je také třeba filtrovat, aby se odlišily nahodilé impulsy (integrace signálu)
- 4) Zelená LED indikuje připravenost zařízení, tzn. hřídel je ve výchozí poloze „zavřeno“.
- 5) Červená LED svítí v intervalech zpomaleného pohybu, tedy v době kdy se hřídel blíží ke koncové poloze a pohybuje se sníženou rychlostí.

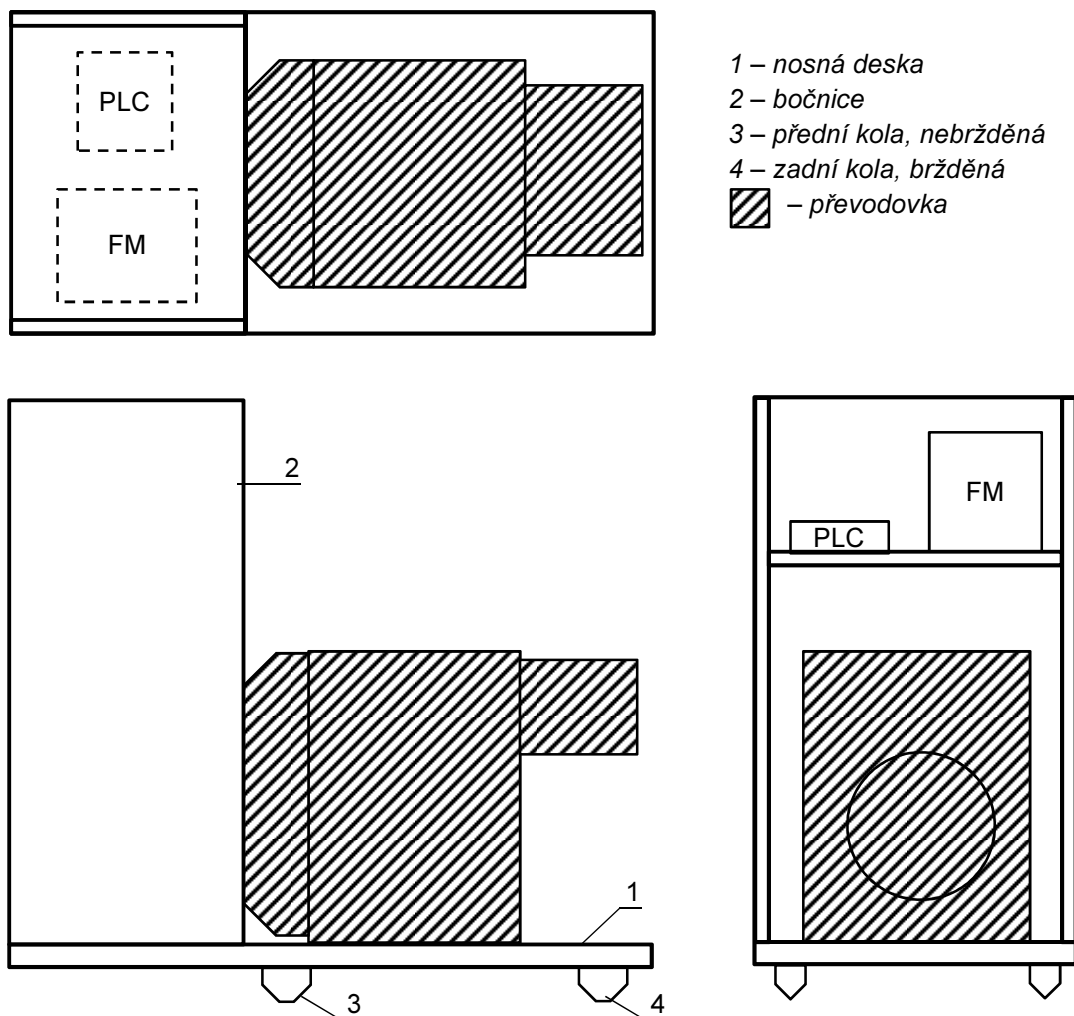
3 Mechanická koncepce a konstrukce

Mým hlavním úkolem bylo navrhnout a sestavit z výše popsaných komponent pracoviště pro výuku. Důležitými kritérii při návrhu pracoviště byly pohyblivost, rozmístění jednotlivých prvků, bezpečnost a snadná ovladatelnost pracoviště. Bylo tedy třeba vytvořit pohyblivou konstrukci, které ponese všechny prvky. Všechny prvky by navíc měly být viditelné, ale PLC a měnič musí být také nepřístupné. Jednotlivé prvky jsou rozmístěny tak, aby znázorňovaly posloupnost řízení, kde na základě uživatelem dodaných vstupů PLC ovládá frekvenční měnič a tím je řízen běh motoru.

Výchozím návrhem bylo uložit převodovku do skříně na pojezdových kolečkách, jejíž horní stěnu bude tvořit deska z průhledného materiálu, nejlépe plexiskla. Rozměry této skříně pak odpovídají krajním rozměrům převodovky, s rozdílem že horní stěna se nachází ve výšce 75 cm od podlahy. Na průhledné horní desce budou umístěny ovládací prvky (tlačítka, LED a spínač) a sériový port PLC. Pod ovládacími prvky, na polici v prostoru skříně, budou umístěny PLC a frekvenční měnič. Celá tato řídicí sekce se bude nacházet v prostoru nad motorem převodovky (viz (1) *obr. 1.2*). Signalizace otáčení převodovky je zajištěna hřídelovým nástavcem, na kterém je umístěn ukazatel. Hmotnost celého zařízení je přibližně 120 kg, skříň tedy musí být sestavena tak, aby její dno a kolečka měly dostatečnou nosnost. Pro stavbu skříně byly v úvaze dvě možnosti, buď z kovových profilů svařit pojízdnou konstrukci, která ponese hmotnost převodovky a ke které se připevní stěny z vybraného materiálu (plast nebo laminovaná dřevotříska). Nebo vyrobit skříň přímo z dřevěných desek, kde jsou na nosnou desku, která je vybavena kolečky, připevněn motor a boční stěny skříně. U druhé možnosti je problém v tom, že převodovka bude dosedat na dno skříně poměrně malou plochou a protože nosným prvkem je samotná deska, může dojít k jejímu přelomení. Aby se omezila délka páky, na kterou působí hmotnost převodovky, bylo nutné umístit kolečka blíže k místu, kde převodovka dosedá na nosnou desku.

Po konzultaci s vedoucím práce, kde jsem dověděl, že výroba svařované konstrukce není možná a v úvahu tak přichází stavba skříně přímo z desek, jsme se rozhodli, že stavba celé skříně bude příliš finančně a časově náročná. Konečná podoba pracoviště (*obr. 3.1*), sice bude vycházet z druhé možnosti konstrukce skříně, ale místo bočních stěn a horní průhledné stěny bude k nosné desce připevněna pouze police nesoucí měnič a automat. Police, která je široká 30 cm, je nesena dvěma bočnicemi a umístěna 20 cm pod jejich horním okrajem. Přední a

zadní strana prostoru, který je tvořen policí a bočnicemi, bude zakryta, aby se zamezil přístup k těmto dvěma částem pracoviště. Horní deska překrývající tento prostor bude, stejně jako ve výchozím návrhu, průhledná a budou v ní umístěny ovládací prvky.



Obr. 3.1: Zjednodušený náčrt pracoviště, rozložení prvků pracoviště

Na nosnou desku (1) (obr. 3.1) je použita pracovní deska z dřevotřísky o tloušťce 30 mm a rozměrech 850 x 425 mm, pohyblivost základny obstarávají čtyři otočná kolečka, přední pár (3) a zadní pár vybavený brzdou (4). Vhodnější materiál pro základnu by byla překližková nebo laťová deska, dřevotřísková deska byla zvolena z finančních důvodů. Převodovka je na desku uložena na podložce z překližky a k základně je pevně uchycena čtyřmi šrouby. Bočnice (2) jsou přidělané na přední straně pracoviště a slouží k upevnění police pro měnič a PLC, použitý materiál je laminovaná dřevotřísková deska o tloušťce 18 mm. Výška bočnic je upravena tak aby jejich horní hrana, v jejíž výšce budou umístěny ovládací prvky, byla

vzdálená 75 cm od povrchu podlahy. Protože je základna ve výšce 15 cm od země, jsou rozměry bočnic 600 x 320 mm. Prostor, kde je umístěn měnič a PLC, je zakryt deskami z měkčeného PVC ze stran a průhlednou deskou na horní straně, která je kvůli vysoké ceně plexiskla vyrobena z průhledného polyesteru, do které jsou zasazeny ovládací prvky. Napájení ze sítě je přivedeno přes 15A vypínač do frekvenčního měniče, ze svorek vstupního napájení měniče je také vyvedeno napětí pro zdroj PLC. Napájecí vodič a signální vodiče ze skříně převodovky jsou do ovládací sekce přiveden otvory vyvrtanými v polici.

Konstrukce pracoviště začala vytvořením pojízdné základny, tedy montáže koleček k nosné desce a vytvořením otvorů pro přídržné šrouby a vyčnívající část duté hřídele převodovky. Jako další krok byly k nosné desce pomocí truhlářských úhelníků připojeny bočnice. Po umístění a upevnění motoru byla mezi bočnice vložena deska police s připevněným automatem a frekvenčním měničem. Horní průhledná deska byla nejprve osazena ovládacími prvky, na jejichž vývody se připájejí vodiče. Poté byla horní deska přišroubována k horním hranám bočnic a zapojeny všechny prvky. Po realizaci zapojení byl prostor řídicích prvků uzavřen dvojicí desek z PVC.

Závěr

V první kapitole je popsána teoretická funkce pracoviště (kap. 1.1) a jednotlivé prvky (kap. 1.2), ze kterých se pracoviště skládá. Jednotlivé podkapitoly, které popisují prvky pracoviště (kap. 1.2.1 – 1.2.4), jsou děleny do dvou částí. První část každé této podkapitoly popisuje dané zařízení v obecné rovině, tedy jeho účel a princip fungování. V těchto částech jsem se snažil jednotlivé prvky popsat tak, aby i čtenář, který není s těmito zařízeními obeznámen, si mohl udělat představu o tom, k čemu jsou a jak fungují. Druhá část podkapitoly potom popisuje konkrétní typ zařízení použitý při konstrukci pracoviště a vysvětluje, jak bude prvek na pracovišti zapojen a jakým způsobem bude v rámci pracoviště fungovat. Ve druhé kapitole jsem se zaměřil na softwarovou stránku pracoviště. V úvodní části kapitoly je popsán způsob jakým se PLC programují obecně a je zmíněno pět typů programovacích jazyků pro automaty. Další část této kapitoly (kap. 2.1) je opět věnována výše zmíněné problematice pro potřeby pracoviště. V této části textu je popsán programovací jazyk Simple3, který je určen pro programování automatů firmy Hypel. Třetí část kapitoly o programovém vybavení pracoviště (kap. 2.2) obsahuje zadání semestrální práce, kterou by studenti měli zpracovat v rámci výuky na tomto stanovišti.

Hlavním cílem této bakalářské práce byla konstrukce měřicího pracoviště (kap. 3), které bude využíváno studenty při výuce. Tento cíl se mi podařilo splnit a pracoviště bylo sestaveno. Hotové pracoviště bude sloužit studentům k tomu, aby si v praxi vyzkoušeli programování PLC. Nejprve jednoduché aplikace, kde např. jsou pomocí tlačítek ovládány signalizační LED, a posléze i řízení servomotoru. Další význam pracoviště je jako praktická ukázka řízení asynchronního stroje pomocí frekvenčního měniče, který může být ovládán elektronicky, jako je tomu v případě pracoviště, nebo ručně. Vzhledem ke konstrukční jednoduchosti asynchronních motorů a dostupnosti polovodičových měničů jsou takto konstruované pohony velmi rozšířené. Stejně tak je čím dál více procesů řízeno elektronicky pomocí logických automatů nebo mikropočítačů. Přínosem pro studenty bude možnost vyzkoušet si programování takového automatu a řízení jednoduchého systému. Dalším přínosem by mělo být zvýšení kvality výuky, právě díky této praktické ukázce.

Seznam literatury a informačních zdrojů

- [1] ZPA Pečky a.s. *MODACT otočný – montážní návod*. 10. vyd. Pečky. 1987. 30 s.
- [2] ZPA Pečky a.s. *Elektrické motory víceotáčkové*. Servomoty MODACT. [online] [cit. 1.5.2013]. Dostupné z: http://www.zpa-pecky.cz/editor/image/stranky3_soubory/k_mon_mop_obalka_net.pdf
- [3] Pohonná technika. *Frekvenční měniče*. Pohannatechnika.cz. [online]. © 2007-2013 [cit. 1.5.2013]. <http://www.pohonnatechnika.cz/frekvencni-menice>
- [4] HANÁK, Z. *Uživatelský manuál frekvenčních měničů TALAN*. Únor 2004. Praha: KONZULT, 2004. 86 s.
- [5] Wikipedia Foundation, Inc. *Programmable logic controller*. Wikipedia. [online] 2.5.2013. [cit 4.5.2013]. http://en.wikipedia.org/wiki/Programmable_logic_controller
- [6] HYPEL. *Dokumentace k programovatelným logickým automatům*. Automatizační technika. [online] Březen 2004 [cit 1.5.2013]. Dostupné z: <http://www.hypel.cz/data/plc.pdf>
- [7] HYPEL. *Simple3 – popis jazyka*. Software. [online] Březen 2001. [cit. 10.5.2013]. Dostupné z: <http://www.hypel.cz/data/simple3.pdf>
- [8] TECO a.s. *Programování PLC podle normy IEC 61 131 – 3 v prostředí Mosaic*. SW příručky - ke stažení. [online] 10. vyd. Listopad 2007 [cit. 10.5.2013]. Dostupné z: http://www.tecomat.com/wpimages/other/DOCS/cze/TXV00321_01_Mosaic_ProgIE_C_cz.pdf

Přílohy

Příloha 1 – Ukázka možného řešení semestrální práce

```
#Configuration "Annex F"
{předpoklady: zavřeno - výchozí poloha; otevřeno - konečná poloha; zavřeno -> otevřeno - chod;
Otevřeno -> zavřeno chod zpět}
VAR //V první části definic proměnných jsou uvedeny vstupy, v druhé výstupy.
X0 # tlacitko1 : BIT
X10 # PO : BIT //snímač polohy - otevřeno
X11 # PZ : BIT //zavřeno
D0 # faze : WORD //řídící proměnná struktury CASE – OF
//-----
Y0 # zelena_LED : BIT
Y1 # cervena_LED : BIT
Y10 # zpet : BIT //reverzace chodu
Y11 # chod : BIT //chod motoru pomalou rychlostí
Y12 # rychle : BIT //zapnutí rychlého chodu
END

PROCEDURE otaceni (var chod, rychle, koncova_poloha, priznak_casovace, LED : BIT; var casovac,
faze : WORD; pristi_faze : WORD)
priznak_casovace; rychle; chod;
IF (casovac > 3999 AND koncova_poloha) THEN rychlost'; LED;
ELSEIF koncova_poloha THEN rychlost'; chod'; LED'; priznak_casovace'; casovac = 0; faze =
pristi_faze;
ENDIF
RETURN
{Tato procedura zajišťuje otáčení hřídele. Nejprve se hřídel roztočí vyšší rychlostí a po uplynutí
nastavené doby zpomalí a dojíždí do koncové polohy, kde se zastaví. Součástí procedury „otaceni“ je
nastavení proměnné „faze“ pro příští fázi programu.}
IF RESET THEN zelena_LED'; cervena_LED'; zpet'; chod'; rychle'; TEN0'; TEN1'; TPA0'; TPA1';
TDM0'; TDM1'; TOE0'; TOE1'; T0 = 0; T1 = 0; M0 = tlacitko1'; faze = 0; RESET';
ENDIF
{Počáteční nastavení proměnných. Provede se pouze při prvním běhu programu po zapnutí napájení
PLC.}
IF (M0 AND tlacitko1) THEN !TEN0; T0 = 0; //Sledování změny stavu vstupu (tlačítka)
ENDIF //Při změně stavu tlačítka z 0 na 1 se zapne časovač, při změně z 1 na 0 se vypne
M0=tlacitko1';
CASE faze OF 0 THEN //Příprava na chod motoru
IF PZ' THEN chod; zpet; //kontrola polohy
```

```
ELSE chod'; zpet'; zelena_LED; //signalizace připravenosti
    IF (T0 > 100) THEN T0 = 0; faze = 1; zelena_LED'; ENDIF //integrace signálu vstupu
ENDIF
```

{V této části je nejprve provedena kontrola, zda se hřídel nachází ve výchozí poloze. Pokud tomu tak není, provede se natočení hřídele do polohy „zavřeno“. Poté co zajistíme natočení hřídele do požadované polohy, se rozsvítí zelená LED, která signalizuje připravenost pracoviště, a čeká se na stisknutí tlačítka. Bylo-li tlačítko stisknuto déle než 100ms (časovač T0), vykoná se hlavní část programu}

```
OF 1 THEN //chod vpřed
    otaceni (chod, rychle, PO, TEN1, cervena_LED, T1, faze, 2); //volání procedury „otaceni“
OF 2 THEN //čekání v poloze otevřeno
    TEN1; //dle zadání hřídel zůstane v poloze otevřeno po dobu 2 sekund
IF (T1 > 1999) THEN TEN1'; T1 = 0; faze = 3; ENDIF
```

```
OF 3 THEN //chod zpět
    zpet;
    otaceni (chod, rychle, PZ, TEN1, cervena_LED, T1, faze, 0)
IF PZ THEN zpet'; ENDIF
{Po návratu zpět do výchozí polohy je proměnná „faze“ nastavena na 0, program se tedy vrací
do přípravné fáze a čeká na stisknutí tlačítka.}
```

```
ENDCASE
```

```
END
```

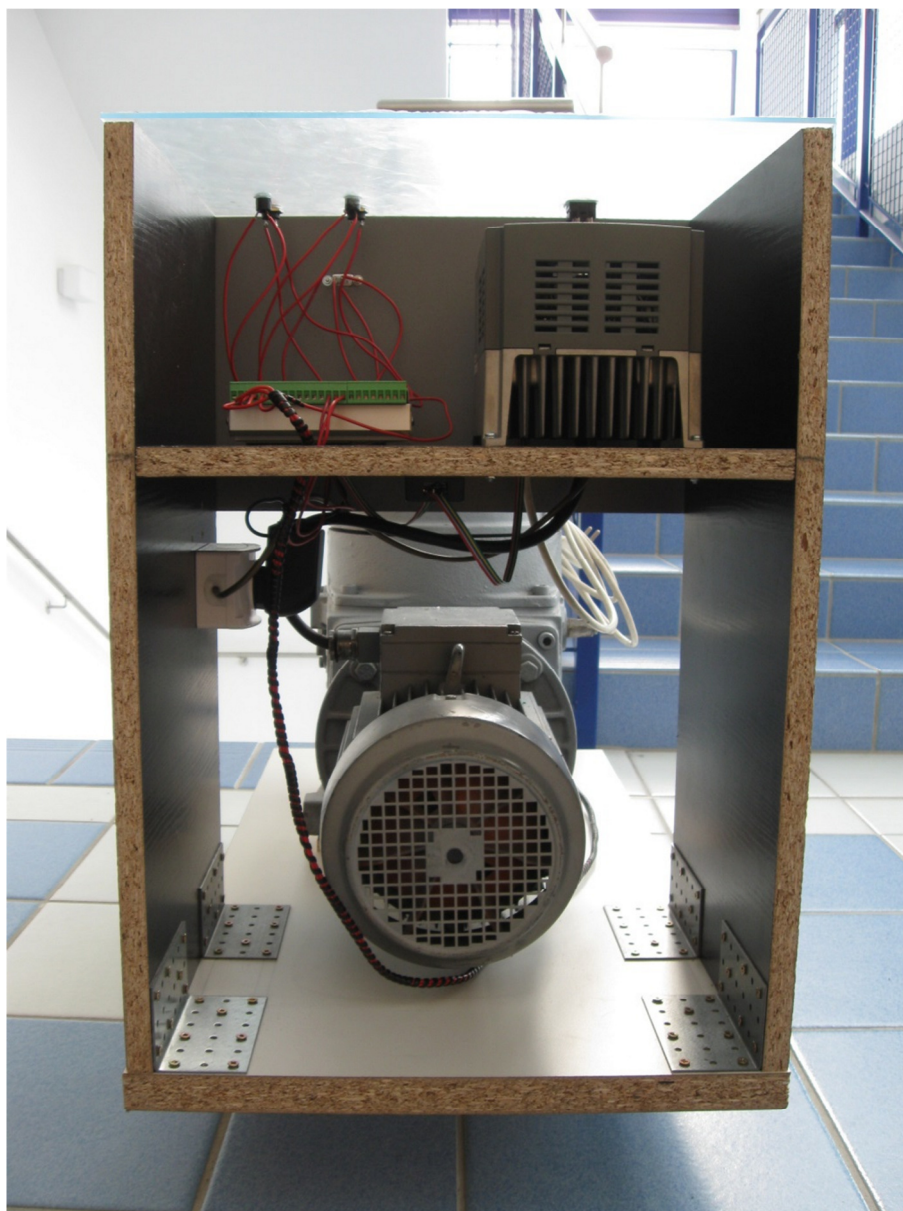

Příloha 2 – Fotografie pracoviště



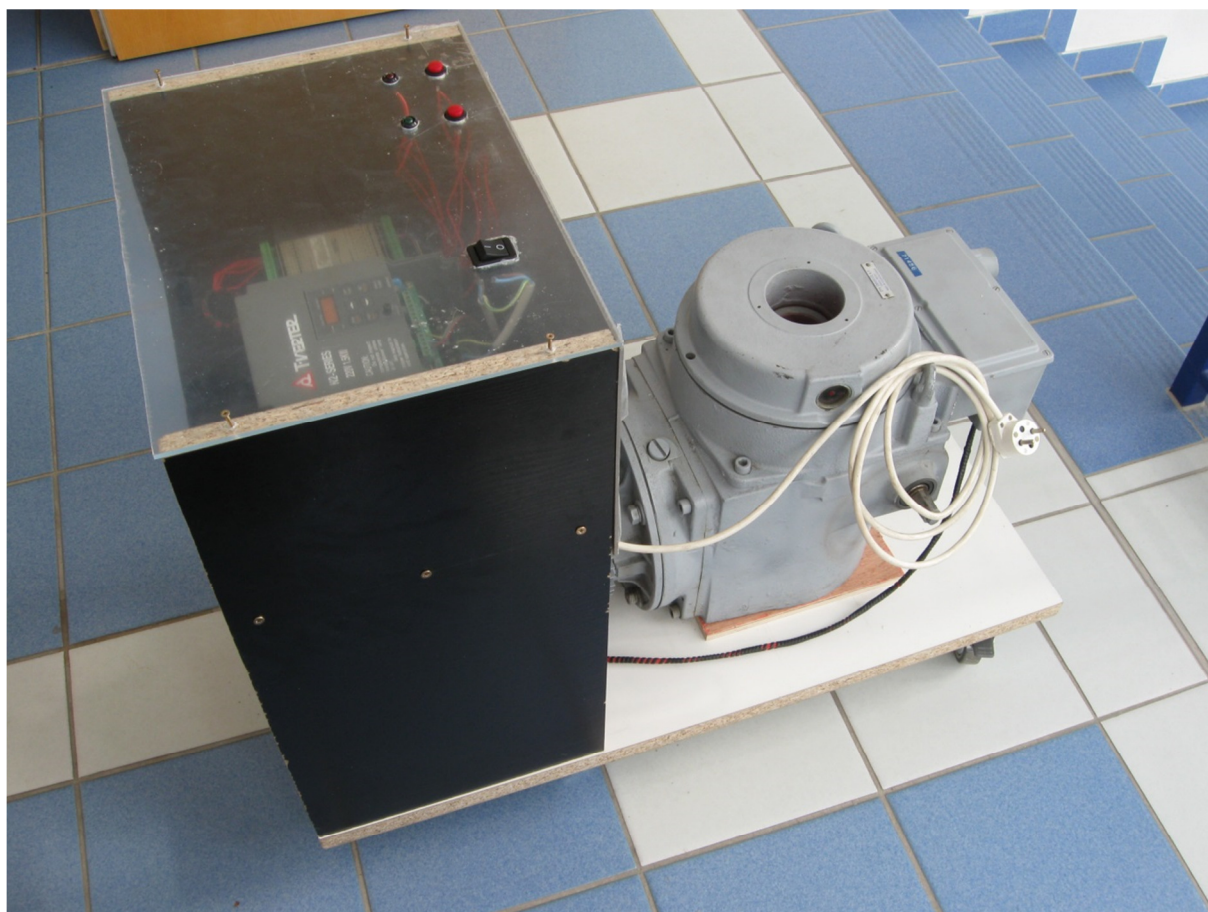
Obr. A: Pracoviště - pohled ze strany



Obr. B: Uložení PLC a FM



Obr. C: Pracoviště – před uzavřením prostoru pro PLC a FM



Obr. D: Pracoviště