

1. Úvod

Diplomová práce se zabývá návrhem konstrukce modelu robota, který je v rozestavěné podobě k dispozici v učebně K909 na katedře řídicí techniky, dále návrhem programového vybavení k tomuto robotu pro programovatelný automat Simatic S7-300 a konečně návrhem ovládacího a kontrolního rozhraní k tomuto robotu pro vizualizační prostředí technologických procesů InTouch.

Robot, jehož programové vybavení tato diplomová práce navrhuje, se skládá především z otočného talíře, který obsahuje několik otvorů (děr) a z vozíku, do kterého se pomocí pásového dopravníku dopraví kulička a vozík jí pak po talířové ploše může odvalovat. Přitom cílem je dovalit kuličku co nejkratší cestou do cílového otvoru a bezpečně se vyhnout všem ostatním otvorům, do kterých by kulička mohla spadnout. Lze přitom řídit jednak otáčení talířové plochy (doleva/doprava) a jednak pohyb vozíku s kuličkou (od středu otáčení/ke středu otáčení), takže se vlastně jedná o přemísťování kuličky ve dvojrozměrném prostoru. Je přitom známa mapa pracovního prostoru, tj. souřadnice všech děr na povrchu otočného talíře a je stanoveno, jaká z děr je pro danou kuličku cílová, přičemž pro každou kuličku předanou vozíku může být cílová díra jiná.

Kromě toho je stanoveno maximální zrychlení pohonů. Protože robot nebyl při zahájení prací postaven, zabývá se první část této diplomové práce (kap. 2 a 3) konstrukčními podklady nutnými pro jeho mechanické a elektronické sestavení.

Dále se diplomová práce věnuje programovatelným automatům Simatic S7-300, filozofii jejich programování a návrhu řídicího programu robota (kap. 4 a 5). Programovatelné automaty (PA) jsou nasazovány v řídicích systémech v průmyslu pro řízení technologických celků v reálném čase. Jejich přednostmi jsou jednoduché nahrazení klasických logických řídicích obvodů (reléová logika), vysoká spolehlivost a snadná úprava funkce programováním. Ačkoliv PA nejsou určeny pro matematické výpočty ani pro práci s daty, mají zpravidla možnosti zpracování dat podobně jako běžný počítač. Ty umožňují provádět méně náročné výpočty v automatu bez potřeby další techniky. Právě v této diplomové práci jsou tyto nadstandardní možnosti PA plně využity a to pro výpočet optimální trajektorie vozíku s kuličkou. Vyšší využití výpočetní kapacity PA však musí být provedeno se značnou opatrností, neboť při nevhodném provedení hrozí možnost zpomalení či zablokování PA na nepřiměřeně dlouhou dobu. V náročných provozech je takové zablokování nepřijatelné, neboť by potenciálně mohlo znamenat ztrátu kontroly nad řízeným systémem a vést případně i k jeho havárii. Proto je cílem se mu vyhnout i v této úloze.

Kromě řídicího programu robota pro PA Simatic S7-300 je dále v této práci navrženo rozhraní pro řízení robota a vizualizaci celého procesu pro prostředí

InTouch, což je program umožňující vizualizaci technologických procesů. Navržené rozhraní kromě vizualizace a základních operátorských funkcí umožňuje i zadání nové mapy pracovního prostoru a naprogramování postupu práce robota (viz kap. 6).

Jednotlivé kapitoly diplomové práce se tedy věnují těmto tématům:

- Kap. 2: Konstrukce modelu robota – popis mechanické konstrukce modelu robota.
- Kap. 3: Elektrické zapojení modelu robota – návrh zapojení elektronické části modelu robota a jeho připojení k PA Simatic.
- Kap. 4: Programovatelný automat Simatic S7-300 – popis PA Simatic řady S7-300 a filozofie jeho programování.
- Kap. 5: Řídicí program robota pro PA Simatic – popis řídicího programu modelu robota pro PA Simatic.
- Kap. 6: Visualizace v InTouch – popis rozhraní pro řízení PA Simatic a pro vizualizaci celého technologického děje navrženého pro prostředí InTouch.
- Kap. 7: Závěr – shrnutí aktuálního stavu prací a dosavadní výsledky.

2. Konstrukce modelu robota

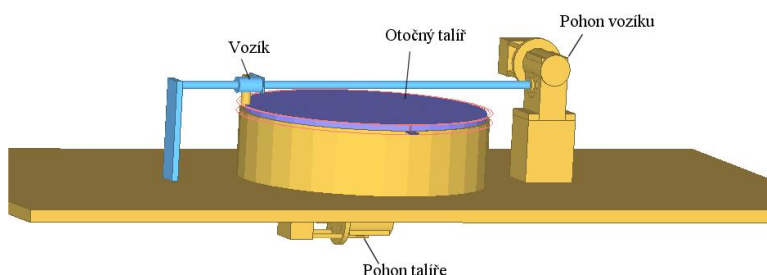
2.1. Výchozí stav modelu

Model robota jsem převzal v rozestavěném stavu, kdy byly hotovy dvě samostatné části, které však nebyly smontovány do jednoho celku. První část byla jednotka karuselu sestavená na dřevěné pracovní desce stolu, druhá část byla jednotka pásu včetně pohonu.

Jednotka karuselu

Celek se skládá z dřevěné pracovní desky (stolu) a z tělesa karuselu, které je k ní připevněno. Horní plochu karuselu tvoří otočný talíř, který je tvořen tělesem z umělé hmoty sevřeným dvěma plechy. Celý talíř je provrtán několika svislými otvory o $\phi 7,7\text{mm}$. Horní plech je spojen se středním umělohmotným tělesem pevně, spodní plech má vůli v otáčení vymezenou dvěma dorazy a je pružinou držen v mírně pootočené poloze oproti poloze střední desky, díky čemuž jsou otvory na spodní straně talíře v klidové poloze uzavřeny. Spodní plech je vybaven jazýčkem, který slouží k detekci polohy talíře a pro mechanické ovládání otevření otvorů. Tělo karuselu je tvořeno plechovým válcem, ve kterém je nakloněná plocha vedoucí propadlé kuličky k výstupnímu otvoru u paty karuselu. Pohon otočného talíře tvoří krokový motor SL23-1012 s převodovkou MRT-28-30. Pohon je umístěný pod plochou stolu.

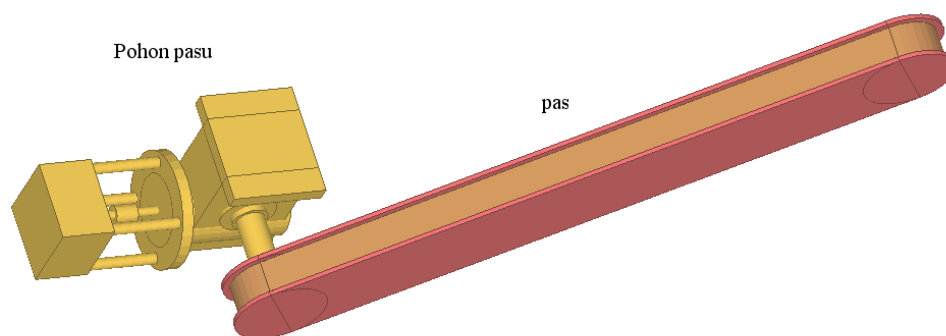
Nad otočným talířem je lineární vedení s vozíkem pro dopravu kuliček. Pohon lineárního vedení je tvořen krokovým motorem SL23-0704 s převodovkou MRT-28-10.



Obr. 1 jednotka karuselu – výchozí stav

Jednotka pásu

Jednotka pásu je tvořena dvěma bočnicemi, mezi nimiž jsou kola napínající gumový ozubený pás. Jedno z kol je poháněno přes převodovku MRT-28-30 krokovým motorkem SL23-0704.



Obr. 2 Jednotka pásu - výchozí stav

2.2. Návrh další konstrukce modelu

Bočnice pásu

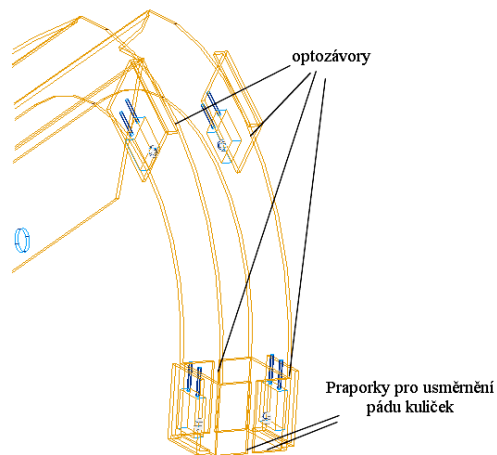
Pásový dopravník je pro naše podmínky zbytečně široký a bez dalších úprav může docházet k volnému příčnému pohybu kuliček. To by mohlo způsobit problémy při navádění kuliček z dopravníku do vozíku a při skulení většího množství kuliček na jednu stranu možná i vyhnutí pásu a propadnutí kuliček. Proto jsem navrhl přidat na boky dopravníku bočnice z plechu. Jejich význam je ve vedení kuliček s minimální vůlí po celou dobu přepravy na dopravníku a zároveň jejich usměrnění při pádu z dopravníku do vozíku. V bočnicích jsou dále připraveny otvory a kryty pro optické závory hlídající přítomnost kuliček na konci pásu a pád kuličky do vozíku, na konci bočnic jsou také praporky pro směřování padající kuličky (Obr. 3; Obr. 4).

Tělo každé z bočnic je naohýbáno z jednoho kusu plechu. Předpokládám strojové řezání plechu podle předlohy a také strojové naohýbání základní části. Kryty optických prvků bude zřejmě nutné ohýbat ručně. Přesné dodržení výkresu v těchto částech není nutné, jen je třeba sledovat, aby nedošlo ke kolizi s pásem. Kryty optických prvků mají být mírně předechnuty aby tlakem držely optické prvky.

(modely všech navržených částí jsou na přiloženém CD-ROM).



Obr. 3 Bočnice pásu

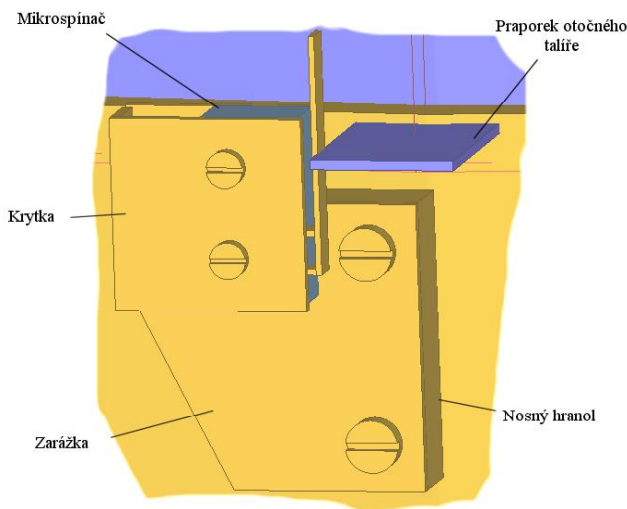


Obr. 4 Bočnice pásu - detail optických závor

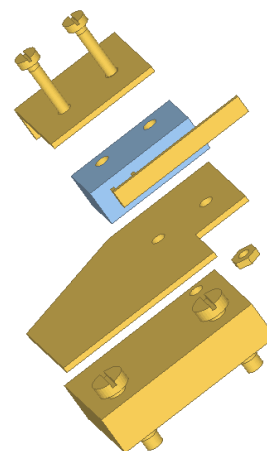
Doraz otočného talíře

Doraz slouží k definici krajní polohy otočného talíře a zároveň je nutnou mechanickou zarážkou pro vysypání kuliček z karuselu. Realizován je kovovým hranolem sloužícím jako distanční podložka, připevněným na bok karuselu. Na něm je přišroubována zarážka z plechu s mikrospínačem. Zarážka slouží pro zastavení spodního plechu talíře a chrání spínač před poškozením nárazy praporku.

Hranol je přišroubován šrouby 3x20mm do závitů v boku karuselu, mikrospínač šrouby 2x12mm, z nichž jeden má závit v hranolku.



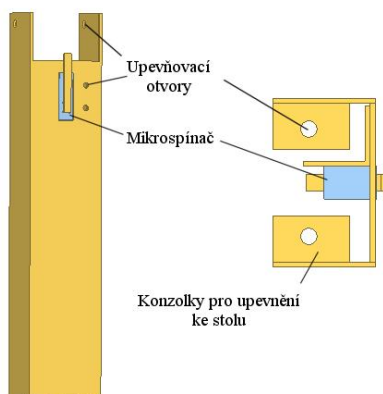
Obr. 5 Zarážka otočného taliře



Obr. 6 Zarážka - detail sestavení

Doraz vozíku a podpěra pásu

Na doraz vozíku nejsou kladeny vysoké nároky po mechanické stránce. Koncová poloha vozíku je zajištěna koncovým spínačem, umístěným na konzoli, která zároveň slouží jako podpěra pásu a je ukotvena dvěma vruty 3x20mm na desce stolu (Obr. 7).



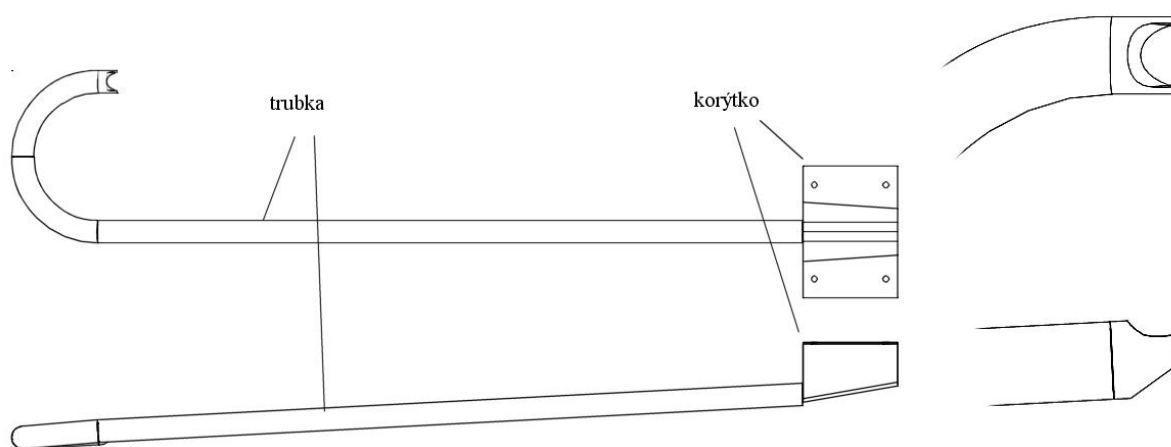
Obr. 7 Doraz vozíku

Doprava kuliček na pás

Přeprava kuliček na pás je realizována hliníkovou nebo měděnou trubkou o vnitřním průměru 10mm. Po celé délce trubky musí být zajištěn sklon větší než cca $0^{\circ}7'$ ($0,2\%$) což je experimentálně ověřená hodnota pro volný pohyb kuliček. Vzhledem ke konstrukci modelu je dosažitelný sklon cca $2,5^{\circ}$. Trubka je vedena od

korýtka zapuštěného v základní desce pod výsypným otvorem karuselu. Kvůli správnému vedení na pás musí být šroubovicově ohnuta. poloměr není kritický a je nutné jej zvolit podle vlastností dostupného materiálu (hliník, měď). Trubka musí být alespoň na 3 místech uchycena k základní desce. Vhodné se jeví úchytky na kabely s tím, že v místech uchycení je nutné mezi trubku a desku stolu vložit distanční podložky pro zajištění sklonu trubky. Konec trubky je vytvarován pro bezpečné vedení kuliček na pás. Zespona je trubka sražena v úhlu 33° aby bylo možno ji co nejvíce přiblížit k pasu, zeshora je vybroušen zářez, který dovolí vynesení kuličky pasem jak je patrné na obrázku (Obr. 8).

Z karuselu jsou kuličky vedeny do trubky s pomocí malého korýtka vytvarovaného z plechu. Korýtko plní funkci trychtýře a směřuje vypadlé kuličky přesně do trubky. Posazeno je v otvoru 50*35mm v pracovní desce a připevněno 4 vruty 3x20mm.



Obr. 8 Korýtko a trubka pro vedení kuliček na pás. Vpravo detail ukočení trubky

Čidla

Čidla jsou nutná pro detekci pohybu kuličky a pro detekci krajních poloh modelu.

Čidla sloužící k monitorování pohybu kuličky jsou tvořena optickými závory, které se skládají vždy z infračervené LED diody a fototranzistoru. V modelu jsou zapotřebí závory pro detekci:

- přítomnosti kuličky na konci pasu (závora v oblouku bočnice)
- propadnutí kuličky do vozíku (na praporku bočnice)

Čidla pro detekci krajních poloh automatu (koncové spínače) jsou realizována pomocí mikrospínačů. Spínač koncové polohy vozíku se nachází v těle podpěry pasu a koncový spínač otočného talíře je zabudován v dorazu otočného talíře na boku karuselu.

Všechny doprovodné obrázky jsou v měřítku 1:1 jako 3D modely na přiloženém CD-ROM

3. Zapojení modelu

Model je řízen PA Simatic. Vazba mezi PA a modelem je realizována signály PA ovládajícími krokové motorky modelu a signály z čidel modelu.

3.1. Připojení výstupů

Krokové motorky všech pohonů jsou řízeny ovládacími moduly SD20M firmy Microcon [4]. Desky jsou řízeny těmito signály:

- směr – směr pohybu motorku
- pulsy – impuls na tomto vstupu je příkazem pro provedení kroku
- disable – umožní odpojit výkonovou část modulu

Vstupy jsou galvanicky odděleny optočleny. Moduly obsahují sequencer řízený těmito vstupními signály a výkonovou část pro buzení cívek krokových motorků. Vstupy modulů jsou napětově kompatibilní s výstupy PA a díky tomu je možné signály „směr“ a „pulsy“ připojit přímo. Signál „disable“ není v modelu potřebný a proto zůstane nepřipojen.

3.2. Připojení vstupů

Optické závory použité jako čidla polohy jsou realizovány infračervenou LED diodou L-NL-7C1 a fototranzistorem L-NP-3C1. Tyto závory není možné připojit přímo k PA, je nutné pro ně navrhnout elektronický modul.

Charakteristické hodnoty pro tyto součástky jsou podle [1] pro LED:

- vlnová délka 940nm
- maximální proud 50mA

pro fototranzistor:

- vlnová délka 940nm
- maximální napětí na kolektoru 30V
- maximální kolektorový proud 5mA
- maximální ztrátový výkon 0,1W

LED dioda i fototranzistor jsou ve stejném plochém průhledném pouzdře BC-1, které má nad čipem malou čočku usměrňující světelný paprsek.

Optické závory je možné pro malé vzdálenosti mezi vysílačem (LED) a přijímačem (fototranzistor, fotodioda) řešit s pomocí jednoduchého zapojení, kde LED je napájena konstantním proudem a signál z fototranzistoru je upraven jednoduchým zesilovačem. Pro větší vzdálenosti klesá hodnota záření z LED dopadajícího na fototranzistor. Záření z LED dopadající na fototranzistor může dosahovat i o několik řádů nižších hodnot, než je záření dopadající na fototranzistor z okolí (běžné osvětlení okolí). V takovém případě je nutné použít složitější zapojení, kde na straně vysílače je generátor frekvence (oscilátor) a na straně přijímače je dekodér (pásmová propust). Tak je zaručeno, že přijímač reaguje pouze na signál z LED a je necitlivý proti pomalu se měnícím hodnotám okolního osvětlení. Zde ale nastává problém, pokud používáme větší množství závor s oscilátorem. Přijímač může reagovat i na signál z jiného vysílače. V tom případě je nutné buď pro každou závoru vyhradit jedinečnou frekvenci značně se lišící od ostatních, či signál z oscilátoru vést zároveň do vysílače i přijímače a v přijímači jej využít v řízeném dekodéru (řízený usměrňovač).

Pro rozhodnutí o typu přijímače jsem provedl experimentální měření odezvy fototranzistoru na různé světelné podmínky. (Tab. 1) Na základě tohoto experimentu jsem se rozhodl pro nejjednodušší typ přijímače, protože u tohoto modelu nepředpokládám expozici na přímém slunci a všechny optické závory jsou alespoň částečně kryté. V případě nutnosti je možné všechny závory zakrýt.

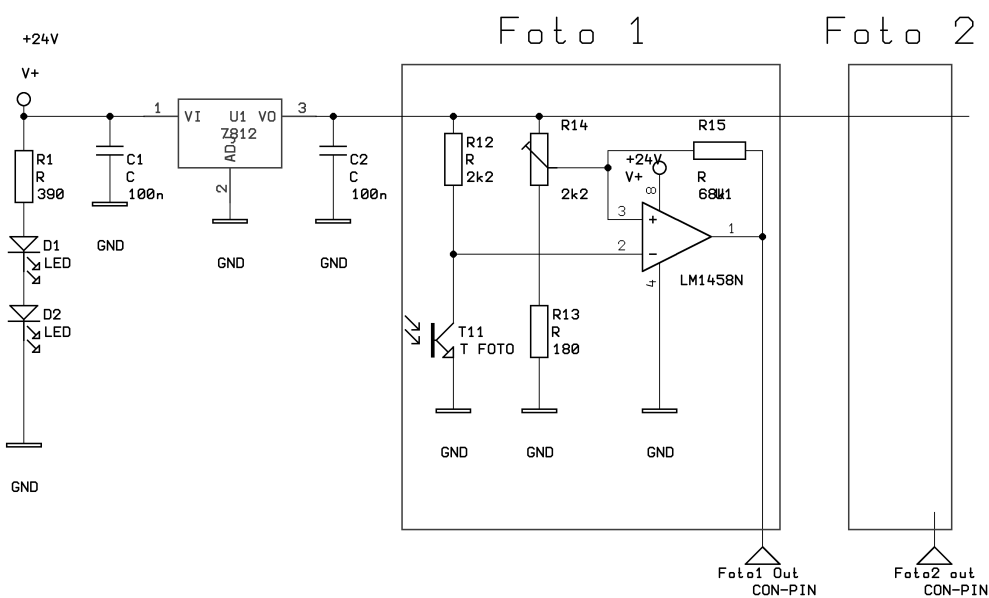
světelné podmínky	proud I_c [mA]
přímé slunce (orientace čidla do slunce)	$\sim 25^1$
přímé slunce (orientace čidla kolmo ke slunci)	6
expozice LED diodou (vzdálenost 12mm, $I_f = 50$ mA)	3
osvětlená místnost za jasného dne (orientace k oknu)	1,7
osvětlená místnost za jasného dne (orientace ke zdi)	0,5
místnost s umělým osvětlením	<0.1

Tab. 1 Odezva fototranzistoru na světelné podmínky

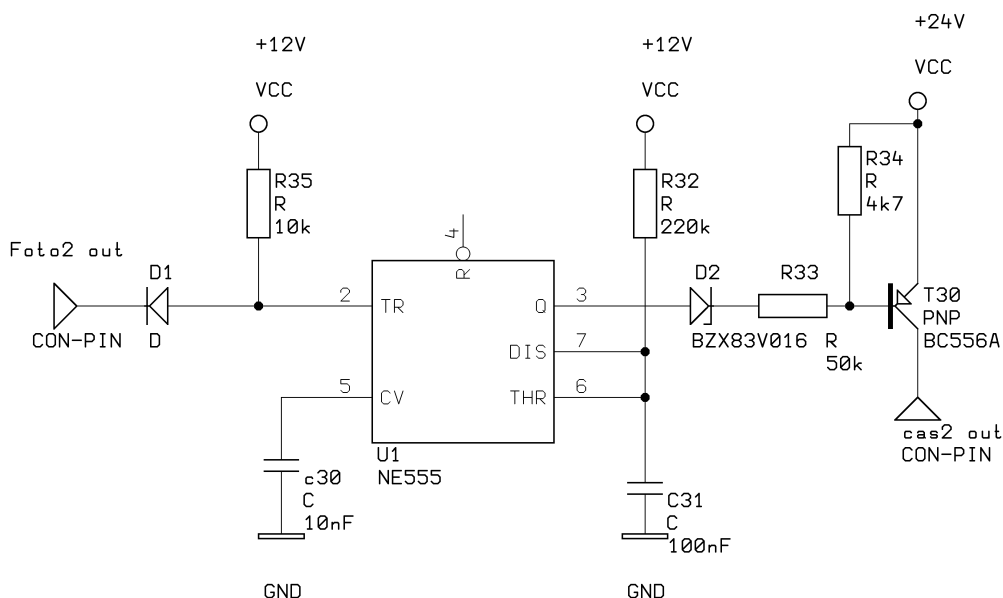
Modul optických závor jsem navrhl s ohledem na dostupné napájecí napětí 24V (používá se pro PA). Všechny LED diody jsou řazeny sériově, čímž se omezí ztráty v předřadném rezistoru R1 (Obr. 9). Každý fototranzistor je zapojen v napěťovém

¹ Vzhledem ke značnému překročení mezních hodnot bylo toto měření zkráceno na minimum a je tedy možná značná nepřesnost. Vzhledem k zajímavosti tohoto měření jsem jej provedl bez omezení proudu i za cenu rizika poškození součástky.

dělič s rezistorem R12, který při napájecím napětí 12V zajistí dodržení předepsaných hodnot fototranzistoru. Operační zesilovač porovnává napětí na fototranzistoru s nastavitelnou referenční hodnotou. Je-li fototranzistor otevřen, je na výstupu operačního zesilovače úroveň H. Odpor R15 zavádí do zapojení hysterezi a tím necitlivost vůči malým poruchám (el-mag a světelné rušení)



Obr. 9 Zapojení modulu optických závor



Obr. 10 Zapojení modulu časovače - monostabilní KO pro prodloužení impulsu

Kulička padající z pasu má pro naše potřeby příliš velkou rychlost. Propadnutí skrz optozávoru při volném pádu trvá při její velikosti méně než 1ms. V tom případě je

značné nebezpečí, že by její přítomnost PA vůbec nezaznamenal. Proto jsem vložil mezi PA a modul optické závory monostablní klopný obvod s IO 555 (Obr. 10). Ten prodlouží impuls na cca 25ms. Výstupní tranzistor je použit pro přizpůsobení signálu pro vstup PA.

3.3. Součástky:

optozávora:

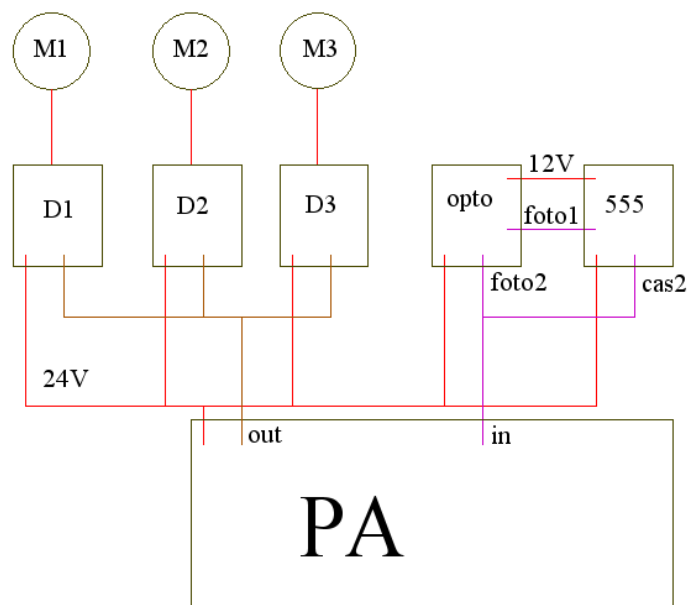
Diody		
D1, D2	L-NL-7C1	infra LED
Rezistory		
R1	390R, 2W	
R12, 22	2k2	
R13, 23	180	
R14, 24	2k5 PTC 10VK002.5	Trimr
R15, 25	68k	
Tranzistory		
T11, 21	L-NP-3C1	fototranzistor
Kondenzátory		
C1, 2	100nF	keram.
Integrované obvody		
U1	7812	stabilizátor 12V
U2	LM1458N	2x oper. zesilovač

časovač

diody		
D1	1N4007	
D2	BZX83V015	zener. dioda 15V
rezistory		
R32	220k	
R33	50k	
R34	4k7	
R35	10k	
kondenzátory		
C30	10nF	keram.

C31	100nF	keram.
tranzistory		
T1	BC556A	PNP
Integrované obvody		
U1	časovač	

Blokové schéma



Obr. 11 Blokové elektrické schéma robota

Všechny moduly jsou napájeny z automatu napětím 24V. Toto napájení je externí 24V/10A z místních rozvodů strojovny. Modul časovače 555 je napájen navíc napětím 12V ze stabilizovaného zdroje desky optických závor.

Jednotlivé moduly jsou:

M1	SL23 1012	pohon karuselu
M2, 3	SL23 0704	pohon vozíku a pasu
D1 – D3	SD20M	výkonové stupně
opto		2x IR závora
555		časovač
PA	Simatic 315-2DP	Programovatelný automat

Přiřazení vstupů a výstupů

robot	PA	
D1 – směr	O1.0	směr karuselu
D1 – krok	O1.1	krok karuselu
D2 – směr	O1.2	směr vozíku
D2 – krok	O1.3	krok vozíku
D3 – směr	O1.4	směr pasu
D3 – krok	O1.5	krok pasu
opto – foto2	I1.0	kulička na pase
cas2	I1.1	kulička z pasu

4. Programovatelný automat Simatic S7-300

Programovatelné automaty (dále jen PA) Simatic řady S7-300 jsou určeny pro nasazení v průmyslu pro řízení technologických procesů v reálném čase. Jsou navrženy tak, aby mohly zastoupit dřívější řízení těchto procesů reléovou logikou. Tomu odpovídá i jejich filosofie.

PA se v nižších třídách vyrábějí jako kompletní malé bloky s několika vstupy a výstupy a jsou určeny k montáži přímo na lišty rozvaděčů. Ve vyšších třídách se vyrábějí v modulárním provedení, kde základní blok s procesorovou jednotkou a zdrojem lze rozšiřovat o moduly vstupů a výstupů. Tyto moduly se vyrábějí pro různé napěťové i výkonové úrovně a tak lze PA vyšší třídy použít ve velkém rozsahu aplikací.

Nejjednodušším způsobem programování PA (zvláště pro inženýry zběhlé v reléové logice) je kreslení kontaktního plánu. Zobrazení programu se v tomto případě velice podobá tradičnímu plánu zařízení s reléovou logikou. Další možností je zobrazení v podobě logického schémata. Zde jsou prvky zobrazeny jako logická hradla či další TTL obvody. Poslední způsob je zobrazení vlastních instrukcí PA (obdoba assembleru). Mezi těmito způsoby zobrazení lze téměř libovolně přepínat, takže každý programátor si může zvolit jemu nejbližší a nejsrozumitelnější zobrazení. Díky těmto možnostem se programovatelné automaty zvláště od firmy Siemens dobrali opravdu masového rozšíření.

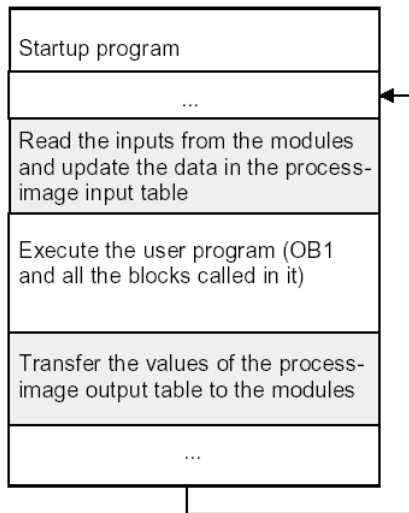
Ideální logický obvod reaguje na změnu vstupních hodnot okamžitě. Reálný logický obvod reaguje se zpožděním daným zejména technologií, ale za provozu vždy s přibližně stejným časem odezvy. Program v PA se ale provádí cyklicky a diskrétně, podle typu automatu a složitosti programu řádově 100x za sekundu.

Celý cyklus PA se skládá z několika fází, jak je znázorněno na (

Obr. 12) [8]. První fáze je načtení vstupů do oblasti vstupů paměti automatu. Tato oblast, resp. data v ní jsou značena písmenem I – input (v německé verzi E – eingabe). Druhou fází je provedení uživatelského programu zapsaného v bloku OB1, což je vyhrazený blok pro cyklické provádění. Tento blok zpravidla obsahuje odkazy pro spouštění dalších programových bloků, takže celý program může být poměrně členitý. V poslední části se provádí přepis oblasti výstupů v paměti automatu na skutečné výstupní moduly. Data v této oblasti se běžně značí Q – output (či A – ausgabe). Po dobu běhu programu se zpravidla nepřístupuje na skutečné vstupy a výstupy, ale pouze k jejich obrazům ve vnitřní paměti. To je pro zamezení kolizí –

pokud by se přistupovalo ke vstupům a výstupům přímo, mohlo by se stát že blok který čte vstup dvakrát by jej jednou přečetl jako aktivní a podruhé neaktivní.

Scan cycle on existing CPUs



Obr. 12 Free scan cycle [8]

Díky zde popsanému způsobu zpracování dat je reakce automatu na vstupní signály zpožděna o čas nutný na provedení cyklu a o náhodnou dobu od změny vstupního signálu do nejbližšího čtení vstupů automatem. Pro zjednodušení můžeme předpokládat, že reakční doba automatu T_r je v rozmezí:

$$T_s \leq T_r \leq 2T_s \quad (4.1)$$

Kde T_s je doba nutná k proběhnutí jednoho cyklu PA. Takto omezená velikost reakční doby zpravidla není na závadu, ale při návrhu řízení s PA je nutné ji uvažovat. To znamená, že je např. nutné uvážit použití PA v kritických aplikacích, kde by zpoždění T_r mohlo způsobit havárii. Dále je pak nutné zajistit, aby prováděný program nebyl zbytečně náročný a periodu T_s neúměrně neprodlužoval.

5. Řídicí program robota pro PA Simatic

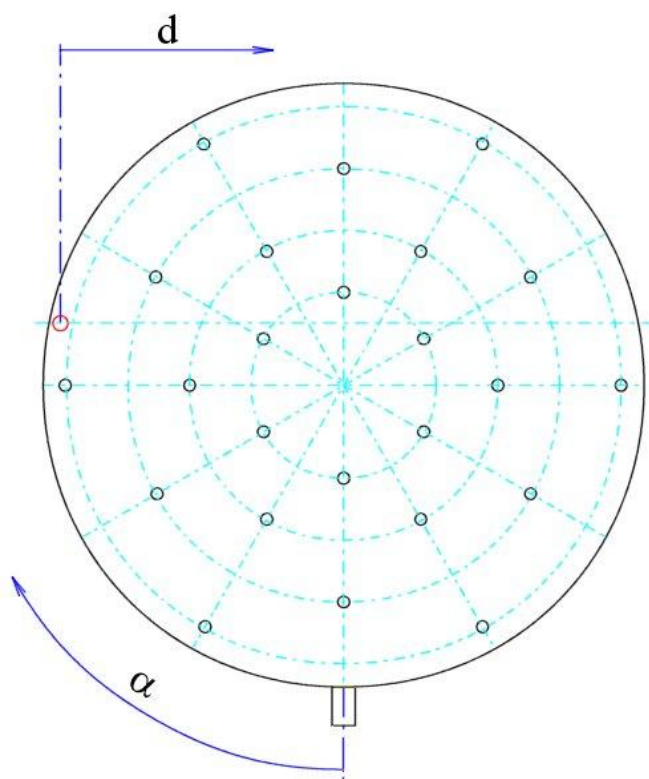
Robot řízený PA Simatic musí v tomto projektu bezchybně dopravit každou kuličku do patřičného cílového otvoru (zásobníku) a zároveň musí řídit tyto další činnosti: navážení kuliček do vozíku pomocí pásového dopravníku, vysypání kuliček ze zásobníků na nakloněnou plochu uvnitř karuselu, reagování na pokyny obsluhy. V této kapitole ukáží nejprve způsob reprezentace prostředí robota, vytvoření mapy prostředí a nakonec nalezení a výpočet optimální trajektorie vozíku s kuličkou.

5.1. Reprezentace prostředí robota

Skutečné souřadnice

Prostředím robota je míněn „svět robota“, čili prostor, který je pro robota relevantní (podstatný), což je v našem případě prostor, ve kterém se může pohybovat vozík robota.

Reprezentace prostředí robota tedy znamená reprezentaci polohy vozíku ve 2D prostoru. Skutečné souřadnice polohy vozíku jsou naznačeny na obrázku (Obr. 13).



Obr. 13 Souřadnice polohy vozíku

Souřadnice polohy vozíku robota jsou tedy dány úhlem natočení α karuselu (praporku otočného talíře) oproti dorazu otočného talíře a vzdáleností d vozíku od jeho základní polohy, která je na okraji otočného talíře. Omezíme-li úhel α na cca 0-300° (to je nutné a samozřejmé i z konstrukce automatu – dorazy) a vzdálenost d na cca 0-180mm (tedy volnost pohybu pouze v levé polovině karuselu), dostaneme vzájemně jednoznačné zobrazení polohy vozíku dané $[\alpha, d]$ na bod na povrchu otočného talíře.

Souřadná soustava $[\alpha, d]$ reprezentuje pozici robota v běžných (metrických) mírách. Při programování PA však raději použijeme souřadnou soustavu odpovídající možnostem řízení modelu, kdy je přirozené jako jednotku pro úhel α a pro vzdálenost d použít krok pohonu. Převodními vztahy se budeme zabývat v následujících odstavcích.

Pohon karuselu

Karusel je poháněn krokovým motorkem SL23-1012. Ten má stejně jako ostatní použité motorky velikost kroku 1,8°. Motorek je připojen převodovkou MRT 28 s převodovým poměrem 1:30. Velikost jednoho kroku pohonu je tedy 0,06°, takže celému pracovnímu rozsahu úhlu natočení karuselu $\alpha \in \langle 0; 300 \rangle$ [°] odpovídá rozsah $x \in \langle 0; 5000 \rangle$, kde x je úhel α vyjádřený v krocích pohonu.

Pohon vozíku

Vozík je poháněn krokovým motorkem SL23-0704 s krokem 1,8° připojeným převodovkou MRT28 s převodovým poměrem 1:10 k lineárnímu vedení KERK SRZ se stoupáním posuvného šroubu 10mm/ot. Velikost jednoho kroku pohonu je tedy 5 μ m, takže celému pracovnímu rozsahu vzdálenosti vozíku od krajní polohy $d \in \langle 0; 180 \rangle$ [mm] odpovídá rozsah $y \in \langle 0; 36000 \rangle$, kde y je vzdálenost d vyjádřená v krocích pohonu.

Pohon pásu

Pás je poháněn motorkem SL23-0704 s krokem 1,8° s převodovkou MRT 28 s převodovým poměrem 1:30. Při průměru poháněcího kola 55mm bude jeden krok motorku pohonu odpovídat 0,029mm posunu pásu. Tento údaj uvádím jen pro úplnost, neboť přesná hodnota není pro řízení důležitá.

Vnitřní reprezentace prostředí robota

Souřadnice děr

PA musí mít informace o souřadnicích děr v otočném talíři karuselu. Tyto souřadnice jsou naprogramovány v bloku DB100 a po startu programu jsou zkopírovány a setříděny do pole DB10, kde každý řádek obsahuje prvky **X**, **Y**, **I**, přičemž:

X je souřadnice odpovídající natočení karuselu,

Y je souřadnice odpovídající posuvu vozíku,

I je index díry.

Toto pole je setříděno podle hodnoty **X**.

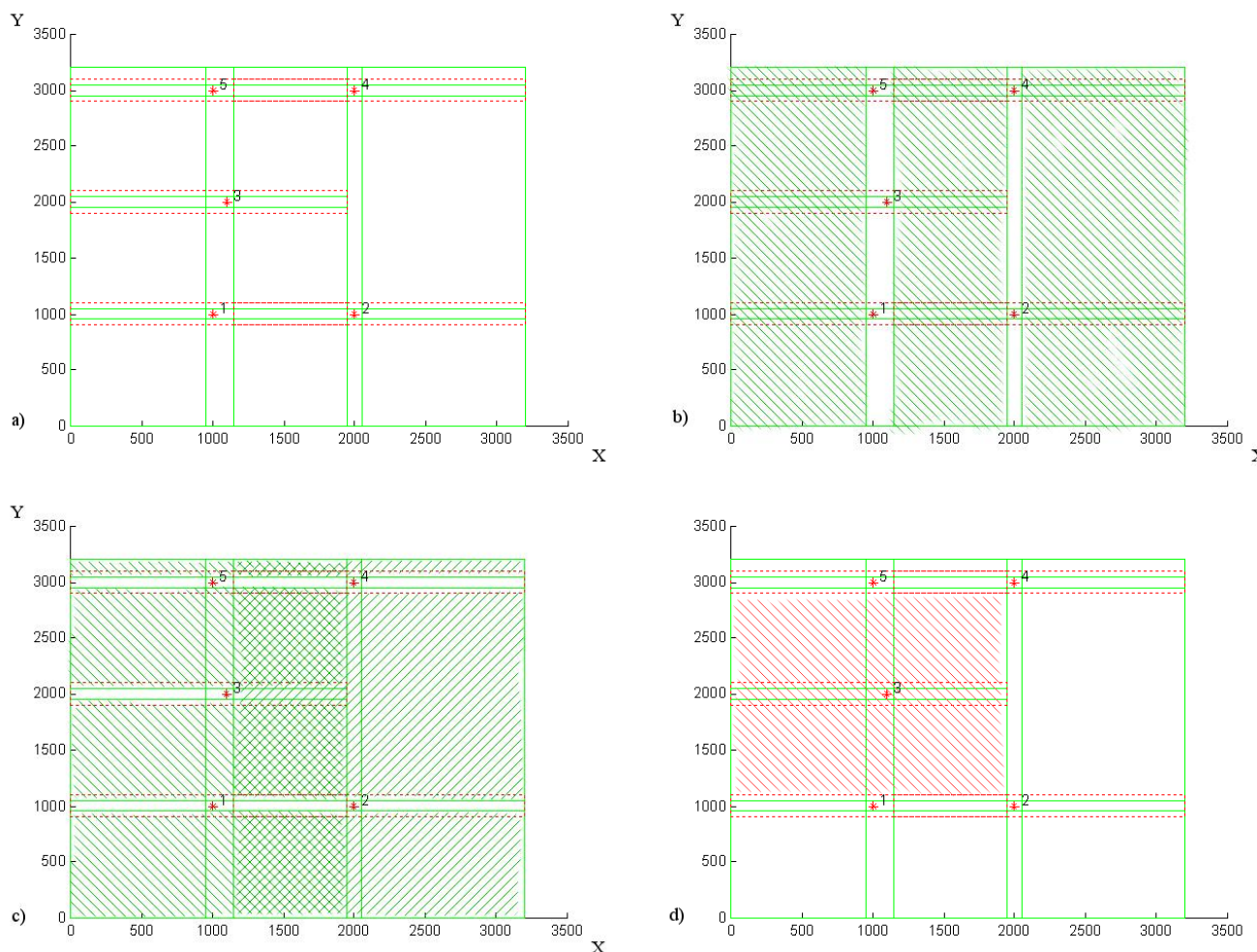
Souřadnice **X** a **Y** jsou na sobě nezávislé. Pro zjednodušení s nimi tedy budu dále pracovat jako s kartézskými

Popis prostoru

Ačkoliv známe souřadnice děr, nestačí sama tato informace k efektivnímu plánování trajektorie vozíku. Jsou různé způsoby, jak plánování vyřešit. My jsme zde ale omezeni možnostmi PA, které neumožňují vyhledávat obecně cestu ve 2D prostoru v reálném čase. Je tedy zapotřebí úlohu pro automat co nejvíce zjednodušit. Jedna z účinných možností je připravit automatu mapu oblastí, takže pak nemusí hledat cestu celým prostorem – pouze rozhodne přes které oblasti cesta povede. Ovšem i oblasti je možné hledat podle různých kritérií, která se budou lišit jak časovou či paměťovou náročností, tak složitostí implementace. Pro nižší výpočetní náročnost jsem zvolil způsob, kde se otvory rozdělí do skupin (pásů), kde v každé skupině jsou otvory s podobnými hodnotami souřadnice **X** (Obr. 14 a). Pak mohu snadno nalézt volné oblasti.

Jeden typ volné oblasti je volný pás mezi dvěma skupinami otvorů (nebo také vlevo, či vpravo od krajních pásů děr) (Obr. 14 b). Druhým typem jsou oblasti mezi jednotlivými děrami ve skupině (Obr. 14 c). Tyto dva typy oblastí se vzájemně překrývají, což umožňuje přes ně plánovat trajektorii.

V mapě ovšem můžeme najít ještě jeden užitečný typ oblasti a to je oblast díry (Obr. 14 d). V této oblasti leží právě jedna díra. Tato oblast vlastně vznikne propojením volných oblastí sousedících s danou dírou. Pokud chceme zavést kuličku do nějaké díry a dostaneme se do její oblasti, můžeme zvolit přímou cestu bez obav z kolize s jinou dírou.



Obr. 14 Reprezentace prostředí a) Základní informace o poloze děr; b) Volné pásy; c) Volné oblasti v páscech děr; d) Oblast díry „3“

5.2. Nalezení jednotlivých oblastí

Nalezení dříve popsaných oblastí je pro PA stále poměrně náročný úkol. Nalezení oblastí je nutné provést při startu automatu, či při přeprogramování prostředí (zadání nových souřadnic děr. Zde ukáži postup vymezení jednotlivých oblastí a nutné úpravy iteračních algoritmů (algoritmů s cykly) pro zachování plné funkce PA.

Nalezení volných oblastí

Pro nalezení volných oblastí potřebujeme setříděný seznam děr, které pak rozdělíme do skupin tak, že ve skupině budou díry s podobnými hodnotami souřadnic X. Volné oblasti pak najdeme jako oblasti mezi skupinami děr a oblasti mezi jednotlivými dírami ve skupině.

Setřídění souřadnic děr

PA na třídění dat nemá vhodné nástroje, neboť nebyl k práci s daty projektován. Obvyklý program PA pracuje cyklicky a v každém cyklu běží téměř bez výjimek „shora dolů“ – tzn. téměř bez skoků, cyklů a podobných konstrukcí, které představují riziko pro včasné dokončení cyklu. Pro třídění ovšem cykly potřebujeme. Je nutné cykly naprogramovat tak, aby se riziko nadměrného brždění automatu minimalizovalo.

Jako základní algoritmus pro třídění jsem zvolil hledání nejmenšího prvku, který sice nemá obecně nejnižší časovou složitost ($O=n^2$), nicméně to vzhledem k předpokládané velikosti pole (max. desítky děr) nevadí. Naopak výhodou je zejména jednoduchost algoritmu a jeho malá náročnost na paměťové přesuny ($O_t=n$). Originální algoritmus třídění (Obr. 15) jsem upravil tak, že hlavní smyčka není uvnitř algoritmu, ale každý průchod smyčkou je proveden při jednom scan cyklu automatu. Tím odpadne problém s tím, že zdržení v cyklu by na delší dobu vyřadilo PA z činnosti (resp. podstatně zvýšilo dobu odezvy). Řízení cyklu je v tomto případě trochu neobvyklé. Inicializace a spuštění cyklu je realizováno vně vlastního algoritmu pro třídění. Také je zde nastaven příznak, že tento cyklus má být prováděn. Příznak je zde realizován proměnnou **krok**, které je přiřazena symbolická konstanta **Tříd'**. Proměnná **krok** je společná pro všechny cykly nejvyšší úrovně. Tím je systematicky zajištěno, že nebude spuštěno více cyklů v jednom okamžiku. Uvnitř cyklu je po dokončení posledního průchodu nastaven příznak **konec**, který je pak vyhodnocen v nadřazeném programovém bloku a řízení je předáno další funkci. Tento systém je v programu použit pro všechny časově náročnější cykly, takže doba jejich provádění je rozložena mezi řadu scan cyklů automatu a základní funkce PA tak nejsou dotčeny.

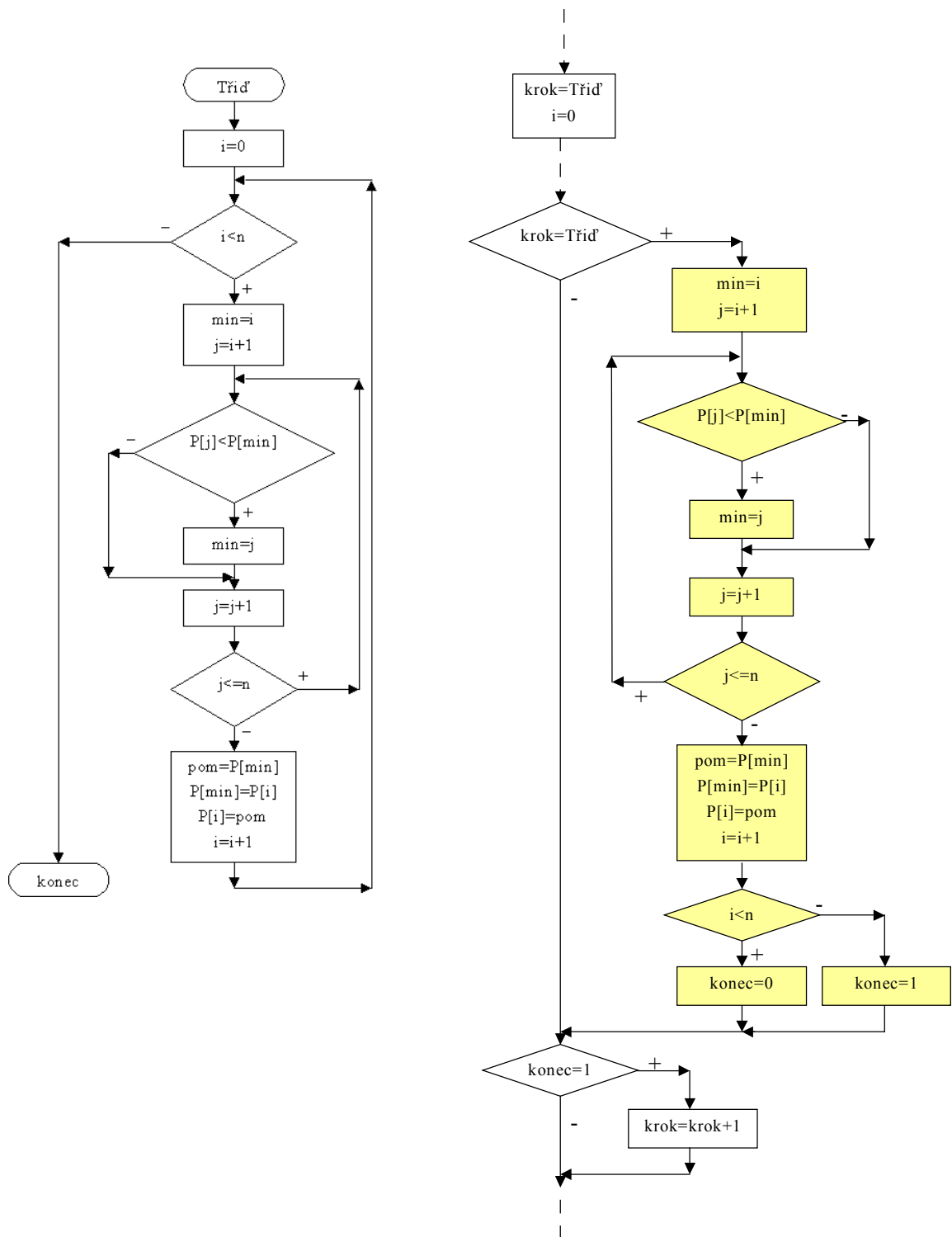
Generování seznamu oblastí

S připraveným setříděným seznamem děr již můžeme nalézt volné oblasti podle následujícího algoritmu.

Předem musíme znát následující hodnoty:

- DX, DY – vzdálenost vlivu díry v obou směrech. Vzdálenost na jakou se vozík s kuličkou nesmí přiblížit k díře, pokud do ní kulička nemá spadnout. Při větším přiblížení vzniká riziko propadnutí kuličky, či její vzpříčení mezi hranou díry a vozíkem. Také definuje minimální vzdálenost mezi dírou a hranicí volné oblasti.
- VX – minimální vzdálenost mezi pásy děr. Také definuje minimální šířku volné oblasti mezi pásy.

Předpokládáme, že hodnota souřadnice X žádné díry není menší než DX a žádná souřadnice Y není menší než DY .



Obr. 15 Algoritmus pro třídění. Vlevo klasická podoba, vpravo úpravy pro PS Simatic

Nejprve vytvoříme tři pole. První (tabV) bude obsahovat tabulku volných oblastí, druhé pole (tabP) bude obsahovat indexy volných pásů v této tabulce a třetí (tabO) bude obsahovat oblasti děr. V prvních dvou polích nastavíme první položky. První volnou oblastí bude pás zcela vlevo od děr. Jeho index zaznameneáme do tabP. Pak do množiny děr v pásu postupně přidáváme díry ze seznamu, dokud nenajdeme takovou díru, jejíž souřadnice X je od předchozí díry vzdálenější, než $2*DX+VX$, nebo dokud nevyčerpáme všechny díry ze seznamu. Tzn je mezi nimi místo na volný pás. Nyní musíme nalézt všechny volné oblasti v tomto pásu děr. Opět je setřídění značnou výhodou. Ovšem nyní potřebujeme seznam děr setříděný podle souřadnice Y. Proto vytvoříme tabulku podobnou tabulce se seznamem děr a zkopírujeme do ní řádky s dírami uzavřené skupiny. Pro setřídění teď můžeme použít klasický algoritmus, protože jednotlivé skupiny budou oproti celému seznamu děr značně menší (na našem talíři maximálně 3 díry, obecně pouze jednotky děr). Po dokončení třídění ještě zkontrolujeme, jestli nejsou nějaké díry příliš blízko u sebe a pokud ano, automat odmítne zadaná vstupní data. V tom případě by totiž nebylo možné vytvořit mezi nimi volný pás. Do tabulky tabV přidáme postupně oblasti pod 1. dírou ve skupině, mezi jednotlivými dírami a nakonec oblast nad poslední dírou. Šířka těchto oblastí bude od počátku posledního zaznamenaného volného pásu až k souřadnici X díry, která již nepatří do skupiny anížené o vzdálenost DX, či k maximální hodnotě X (X_{max}) pokud již nejsou žádné další díry v seznamu. Nakonec do tabulky volných oblastí přidáme volný pás a jeho index uložíme i do tabP. Průběžně doplňujeme data i do tabO. V tabulkách (Tab. 2 - Tab. 5) jsou vidět výsledná data pro ilustrační příklad (Obr. 16)

	X	Y	I
1	1000	1000	1
2	1000	3000	5
3	1100	2000	3
4	2000	1000	2
5	2000	3000	4

Tab. 2 Setříděný seznam děr

	Xmin	Xmax	Ymin	Ymax
1	0	950	0	3200
2	0	1950	0	950
3	0	1950	1050	1950
4	0	1950	2050	2950
5	0	1950	3050	3200
6	1150	1950	0	3200
7	1150	3200	0	950
8	1150	3200	1050	2950
9	1150	3200	3050	3200
10	2050	3200	0	3200

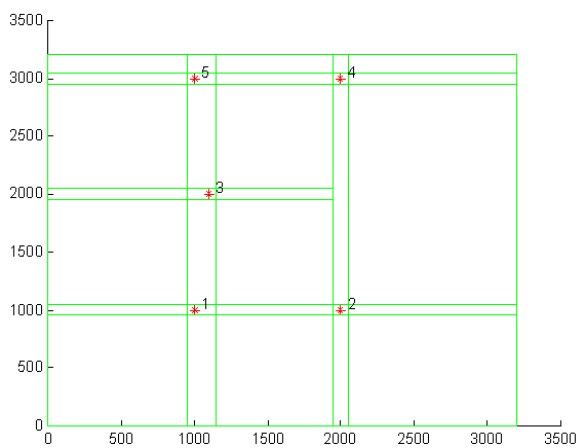
Tab. 3 Seznam volných oblastí

	1	2	3
1	1	6	10

Tab. 4 Indexy pásů v tabulce volných oblastí

	Xmin	Xmax	Ymin	Ymax	X	Y	pas	poradi
1	0	1950	0	1950	1000	1000	1	1
2	1150	3200	0	2950	2000	1000	2	1
3	0	1950	1050	2950	1100	2000	1	2
4	1150	3200	1050	3200	2000	3000	2	2
5	0	1950	2050	3200	1000	3000	1	3

Tab. 5 Oblasti děr. Rozšířená tabulka. Udává i příslušnost díry k pásu děr a pořadí v něm



Obr. 16 Přehledový obrázek prostředí

5.3. Hledání trajektorie pohybu

Vyhledávání oblastí trajektorie

Při zavážení kuličky vozíkem je třeba dbát na to, aby se vozík vyhnul všem ostatním dířům. Proto je nutné předem naplánovat dráhu vozíku. Dráhu tedy hledáme jako sled volných oblastí, kde se pravidelně střídají volné pásy a volné oblasti v pásech děr. Ve volných pásech je možný libovolný pojezd ve směru Y, ale k projetí na sousední volný pás je nutno najít volnou oblast v pásu děr. Do seznamu oblastí trajektorie vložíme tedy všechny volné pásy přes které je nutno projet. Hledáme pásy které zasahují do intervalu $\langle X_{start}, X_{cíl} \rangle$.

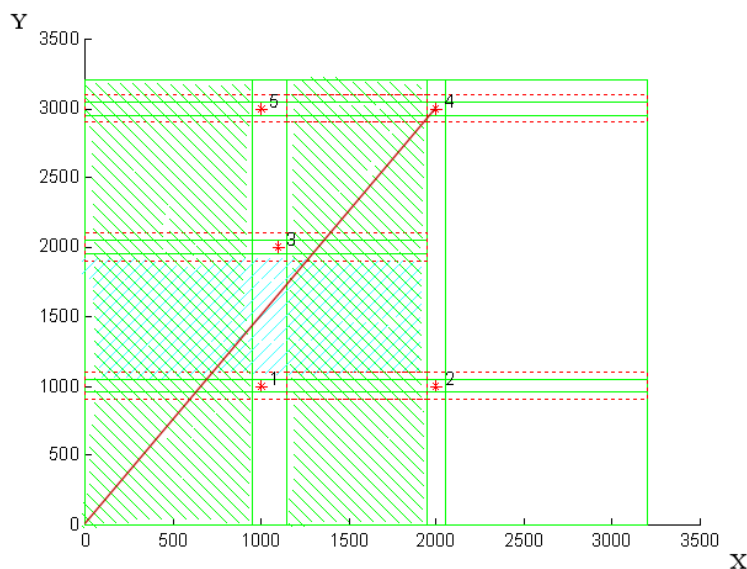
V pásech děr pak hledáme ty oblasti, které leží nejbližší požadovanému průchodu mezi pásy. Hledáme tedy takovou oblast, která pokud možno obsahuje, nebo je nejbližší hodnotě

$$y = \frac{y_{cíl} - y_{start}}{x_{cíl} - x_{start}} \cdot (x - x_{start}) + y_{start} \quad (5.1)$$
$$x = \frac{x_{p1end} + x_{p2start}}{2}$$

kde

x_{start}, y_{start}	je souřadnice počátku pohybu,
$x_{cíl}, y_{cíl}$	souřadnice cílové díry,
x_{p1end}	konec 1. pásu,
$x_{p2start}$	začátek následujícího pásu

X-ové souřadnice bodu přechodu mezi jednotlivými oblastmi volíme vždy na krajích volných pásů, souřadnici Y buďto ideálně jako průsečík přímky mezi počátečním a koncovým bodem trajektorie (Obr. 17), nebo jako nejbližší bod uvnitř volné oblasti pokud by průsečík ležel mimo volnou oblast.



Obr. 17 Ideální trajektorie

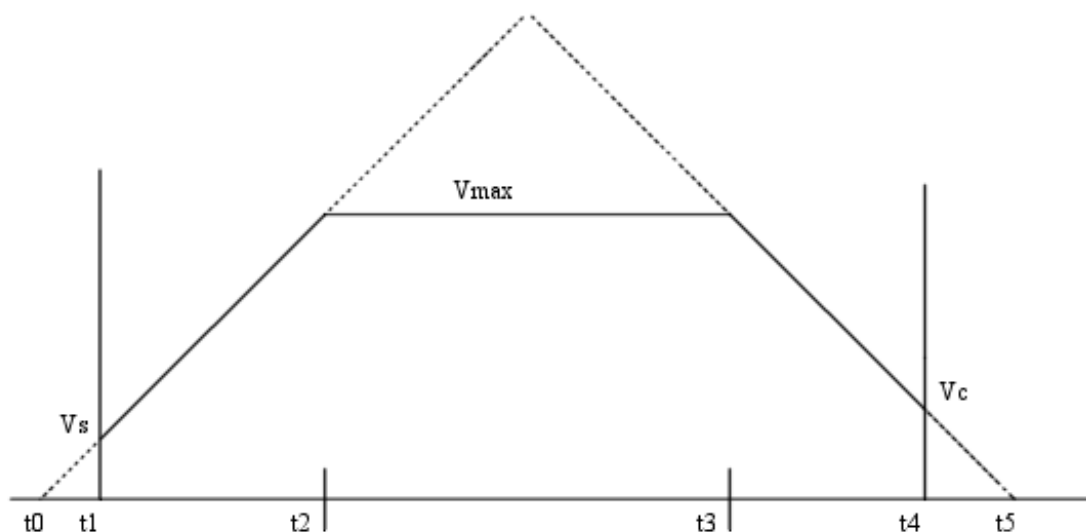
Ke každému bodu přechodu spočteme maximální dovolenou rychlost. Není možné kontrolovat celou trajektorii, tak rychlost odhadneme jako takovou, jaké je možné dosáhnout při zadaném zrychlení z počátečního bodu v oblasti a zároveň ji bude možné daným zpomalením ubrzdít v následující oblasti. Samozřejmě s respektováním omezení rychlosti a zadaného maximálního zrychlení.

Výpočet pohybových rovnic

Po nalezení všech oblastí trajektorie spočteme ideální rovnice pro výpočet pohybu ve směru X a Y pro každou oblast. Předpokládáme, že v každé oblasti je možné pohyb rozložit do 3 dob – zrychlení, rovnoměrný pohyb, zpomalení (Obr. 18). Při hledání nejlepší trajektorie při daném zrychlení, maximální rychlosti, maximální rychlosti na začátku a konci úseku a požadované dráhy, můžeme vycházet z následujících výpočtů.

Pro rovnoměrně zrychlený pohyb od nulové rychlosti platí:

$$\begin{aligned}
 x &= \frac{1}{2} at^2 \\
 t &= \frac{v}{a} \\
 x &= \frac{1}{2} \frac{v^2}{a}
 \end{aligned}
 \tag{5.2}$$



Obr. 18 Průběh rychlosti při rovnoměrně zrychleném, rovnoměrném a rovnoměrně zpomaleném pohybu

Pak můžeme spočítat dráhu, jakou vozík ujede při maximálním zrychlení na maximální rychlost a pak zpomalení. Tento vztah při uvažování počáteční a koncové rychlosti bude mít podobu

$$x = \frac{v_{\max}^2}{a} - \frac{1}{2} \frac{v_s^2}{a} - \frac{1}{2} \frac{v_c^2}{a} \quad (5.3)$$

Na základě těchto vztahů snadno zjistíme minimální dobu nutnou k ujetí požadované vzdálenosti. Výpočet provedeme vždy pro oba směry. Je téměř jisté, že v jednom směru bude potřebná doba podstatně kratší. V tom případě vypočteme parametry i pro tento směr tak, abychom se dosatali na požadované místo najednou.

V těchto výpočtech se ovšem neobejdeme bez výpočtů odmocniny. PA ji dokáže spočítat, ovšem doba výpočtu je až 1ms. Pokud bychom počítali parametry celé trajektorie najednou, byla by doba zpracování podle členitosti prostředí řádově až desetiny sekundy. Není tedy jiná možnost, než tyto výpočty naprogramovat jako otevřenou smyčku, podobně jako jsme si ukazovali na algoritmu pro třídění (Obr. 15). Tyto výpočty je nutné provést před vyjetím vozíku, protože při pohybu vozíku by značně zdržovali. S výhodou je můžeme provést v době nakládání kuličky na vozík. Pro cestu vyloženého vozíku zpět stačí jen jednoduchý výpočet, protože už není nutné vyhýbat se dířám.

Při samotné jízdě vozíku stačí postupně číst parametry zrychleného či rovnoměrného pohybu, na základě jednoduchých vztahů (viz. 5.4) počítat požadovanou

polohu a tuto polohu sledovat pohony. V těchto výpočtech se již používají jednodušší operace, které zdaleka nejsou tak náročné na čas.

$$\begin{aligned}x &= x_0 + (t - t_0) * v \\x &= x_0 + \frac{a(t - t_0)^2}{2}\end{aligned}\tag{5.4}$$

6. Visualizace pomocí InTouch

InTouch je známý systém pro vizualizaci procesů. Běžně se využívá jako rozhraní operátora v řadě technologických provozů. Mezi jeho výhody patří:

- jednoduchý objektový grafický návrh
- práce pod systémem MS Windows
- existují rozhraní pro téměř všechny řídicí platformy (Simatic, Alan-Bradley,...) s jednoduchým přístupem k datům řídicího systému

Proto se aplikace v InTouch velmi rozšířily a rychle vytlačují klasické rozsáhlé operátorské panely (velíny) známé zejména v provozech sedmdesátých a osmdesátých let.

InTouch se výborně hodí i pro různé výukové účely, protože umožňuje rychlou a levnou vizualizaci různých studentských řídicích aplikací.

Z těchto důvodů ho navrhuji i já jakožto operátorský panel k tomuto modelu.

6.1. Návrh operátorského panelu

Od operátorského panelu tohoto robota požadujeme plnou kontrolu nad robotem, tzn. přístup k jeho funkcím a kontrolu stavu.

Proto jsem navrhl rozhraní kde je vidět poloha reálná poloha vozíku jak graficky tak číselně. Dále jsou zobrazeny stavy koncových spínačů, kde poloha v koncové poloze je indikována rozsvícením patřičných světel. A přítomnost kuličky v závorách je indikována zobrazením kuličky na konci pasu.

Ovládání robota navrhuji pomocí těchto řídicích prvků:

režim

- AUTO – automatický chod podle zadaného programu
- MAN – ruční řízení pomocí dalších ovládacích tlačítek

v režimu MAN jsou aktivní tyto prvky

- MAP – Robot smaže seznam děr čeká na zadání nových
- PROG – Programování postupu zavážení děr
- NEXT – Při programování zvolí další díru
- ENTER – Potvrdí současnou pozici
- další tkačítka – ruční ovládání pohonů

AUTO

Automat pracuje samostatně podle zadání, tj, zaveze díry podle programu a po splnění programu vše vysype a začne znovu. Není-li automat správně naprogramován, zjistí to při inicializaci tabulek prostředí a režim auto vypne.

MAN

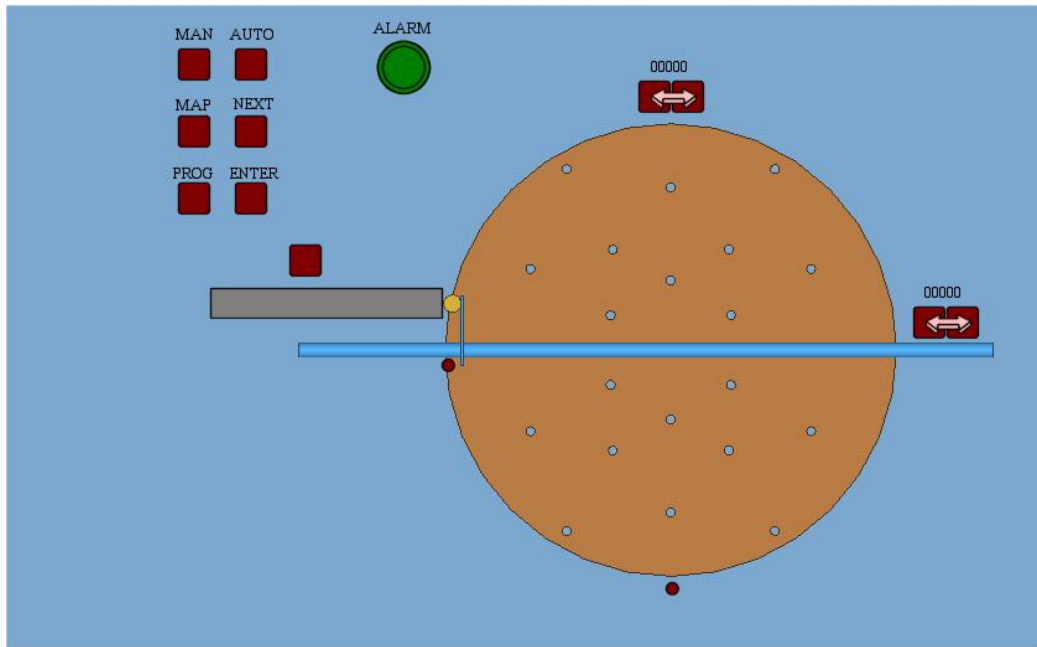
Umožní ruční řízení robota, nebo jeho programování po zvolení funkce.

MAP

Po přepnutí automat „zapomene“ původní mapu a čeká na zadání nových souřadnic děr. Pořadí zadávání děr určuje očíslování děr. Pro zadání polohy díry je nutné v ryčném režimu dojet vozíkem nad díru a stisknout ENTER. Poté je možné zadat polohu další díry, nebo odkliknutím režimu MAP potvrdit platnost zadaných dat.

PROG

Programování postupu činnosti automatu. Po zapnutí režimu automat „zapomene“ pořadí zavážení děr a očekává zadání nového. Stisknutím tlačítka NEXT automat najede nad další díru dle svého seznamu. Při stisku ENTER si automat uloží vybranou díru jako další do seznamu pro zavážení. Pro potvrzení a ukončení zadání je nutné opět vypnout režim PROG.



Obr. 19 Operátorský panel

6.2. Rozhraní PA Simatic – InTouch

Pro komunikaci mezi automatem a vizualizačním panelem je nutné navrhnout protokol. Naznačení funkce panelu je vidět na Tab. 6

proměnná PA	režim přístupu z panelu	význam
M0.0	R/W	režim AUTO
M0.1	W	režim MAN
M0.2	W	ruční posun vozíku vlevo
M0.3	W	ruční posun vozíku vpravo
M0.4	W	ruční otáčení talíře vlevo
M0.5	W	ruční otáčení talíře vpravo
M0.6	R	doraz vozíku
M0.7	R	doraz talíře
M1.4	R	kulička na páse
M1.5	R	kulička spadla z pasu
M2.2	W	režim MAP
M2.3	W	režim PROG
M2.4	W	signál NEXT
M2.5	W	signál ENTER
M2.6	W	ruční spuštění pasu
MW60	R	poloha vozíku
MW62	R	poloha talíře

Tab. 6 přiřazení proměnných pro komunikaci mezi PA a vizualizačním panelem

7. Závěr

Cílem diplomové práce bylo navrhnout konstrukci výše popisovaného robota a tuto konstrukci pokud možno provést, dále sestavit k robotu řídicí program pro programovatelný automat (PA) Simatic a nakonec navrhnout ovládací a kontrolní rozhraní tohoto řídicího programu pro prostředí InTouch. Závěrem ukažme, jak se jednotlivé dílčí cíle diplomové práce podařilo naplnit.

V současné době je hotova a připravena k použití kompletní technická dokumentace, která je nutná ke konstrukci robota a k vyhotovení některých jeho mechanických částí. Veškeré potřebné technické výkresy (ve 3D zobrazení, zhotovené v programu VariCAD) jsou k dispozici na cd, které je přiloženo k diplomové práci. Z elektronické části je potřebné pouze schéma k optočidlům, přičemž toto je uvedeno v kap. 2. i na přiloženém CD. Vlastní sestavení robota však z časových důvodů nebylo provedeno.

K dispozici je dále řídicí program robota pro PA Simatic S7-300, který je hotov z převážné části, avšak z časových důvodů ne úplně. Nejzákladnější a nejobtížnější částí tohoto programu je výpočet optimální trajektorie pohybu chapadla s kuličkou, přičemž návrh této programové části je podložen simulacemi v programu Matlab. Přitom jak návrh výpočtu optimální trajektorie, tak jeho simulace v Matlabu byly částečně dokončeny a jsou k dispozici na přiloženém cd – program ještě obsahuje chyby, které se projevují za jistých podmínek.

Co se týče ovládacího a kontrolního rozhraní řídicího programu pro prostředí InTouch, bylo zde provedeno z časových důvodů nejméně prací. Hotov je návrh komunikace mezi řídicím programem PA Simatic a rozhraním a dále návrh obrazovky tohoto rozhraní.

Aktuální stav prací na robotu pro „třídění předmětů“ je tedy takový, že jsou dokončeny dvě nejobtížnější práce, a to funkční návrh robota, vč. technické dokumentace pro vyhotovení chybějících konstrukčních dílů a návrh hlavní součásti řídicího programu robota, kterou je algoritmus pro výpočet optimální trajektorie chapadla robota k cílovému otvoru. Nedokončeny zůstávají tyto úkoly: sestavení robota vč. předchozí výroby chybějících konstrukčních dílů, dotažení a oživení jeho řídicího programu pro PA Simatic a naprogramování vizualizačního a řídicího rozhraní v prostředí InTouch, které bude ovládat řídicí program Simaticu a graficky znázorňovat celý „technologický“ proces.

8. Přehled použité literatury

- [1] *ELE – Internetový obchod [on-line]*; ELE; <http://www.ele.cz>
- [2] *LM1458/LM1558 - Dual Operational Amplifier [file]*; National Semiconductor; <http://www.national.com/ds.cgi/LM/LM1458.pdf>
- [3] *MICROCON: Krokové motory [online]*; Microcon, <http://www.microcon.cz/pdf2004/19-20.pdf>
- [4] *MICROCON: Moduly SD [online]*; Microcon, <http://www.microcon.cz/pdf2004/12-13.pdf>
- [5] *Programmable Logic Controllers S7-300: Module Data [online]*; Siemens, http://www.ad.siemens.de/simatic/controller/html_76/support/techdoku.htm
- [6] *Siemens SG: S7-300 Instruction List [file]*; Siemens, soubor S7300Instr.pdf, zdroj: <http://dce.felk.cvut.cz/rs>
- [7] *Simatic: System Software for S7-300/400: System and Standard Functions [file]*; Siemens, soubor sfc_e.pdf, zdroj: <http://dce.felk.cvut.cz/rs>
- [8] *SIMATIC S7: Programming with STEP 7 V5.0*; Siemens, soubor Prog50_e.pdf, zdroj: <http://dce.felk.cvut.cz/rs>
- [9] *Součástky pro elektroniku*; Praha: GM electronic, 2004. 537 s.
- [10] *Výkres převodovky MRT 28A [file]*; Znojmo: TOS Znojmo, soubor mrt28_a.dwb, zdroj: <http://www.tos-znojmo.cz>

9. Přílohy

9.1. CD-ROM

Výkresy	/vykresy
Schémata zapojení	/schema
Texty DP	/texty
Programy v Matlabu	/matlab

Obsah

1.	Úvod	1
2.	Konstrukce modelu robota	3
2.1.	Výchozí stav modelu.....	3
	Jednotka karuselu.....	3
	Jednotka pásu	4
2.2.	Návrh další konstrukce modelu	4
	Bočnice pásu	4
	Doraz otočného talíře	5
	Doraz vozíku a podpěra pásu.....	6
	Doprava kuliček na pás.....	6
	Čidla.....	7
3.	Zapojení modelu	9
3.1.	Připojení výstupů	9
3.2.	Připojení vstupů	9
3.3.	Součástky:	12
4.	Programovatelný automat Simatic S7-300	15
5.	Řídicí program robota pro PA Simatic	17
5.1.	Reprezentace prostředí robota.....	17
	Skutečné souřadnice.....	17
	Vnitřní reprezentace prostředí robota	19
5.2.	Nalezení jednotlivých oblastí.....	20
	Nalezení volných oblastí.....	21
5.3.	Hledání trajektorie pohybu	26
	Vyhledávání oblastí trajektorie.....	26
	Výpočet pohybových rovnic	27
6.	Visualizace pomocí InTouch	30
6.1.	Návrh operátorského panelu	30
6.2.	Rozhraní PA Simatic – InTouch.....	33
7.	Závěr	34
8.	Přehled použité literatury.....	35
9.	Přílohy.....	36
	9.1. CD-ROM.....	36
	Obsah	37