

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

ZAVEDENÍ ROBOTIZOVANÉHO PRACOVISTĚ DO VÝROBY

IMPLEMENTATION OF ROBOTIZED WORKPLACE IN TO THE PRODUCTION

BAKALÁŘSKÁ PRÁCE
BACHELOR THESIS

AUTOR PRÁCE
AUTHOR

MICHAL CIPRYS

VEDOUcí PRÁCE
SUPERVISOR

ING. TOMÁŠ MARADA, PH.D.

BRNO 2010

ZADÁNÍ ZÁVĚREČNÉ PRÁCE

(na místo tohoto listu vložte originál a nebo kopii zadání Vaš práce)

LICENČNÍ SMLOUVA

(na místo tohoto listu vložte vyplněný a podepsaný list formuláře licenčního ujednání)

ABSTRAKT

Tato práce řeší problém zavedení robotizovaného pracoviště do výroby. Úkolem je seznámení s robotickým manipulátorem, se způsoby programování robotického manipulátoru a jeho plné nasazení do výroby.

ABSTRACT

This work solve the problem introduction robotized workplace to the production. Imposition is acquaintance with robotic manipulator, manners programming robotic manipulator and his full setting to the production.

KLÍČOVÁ SLOVA

Robot, manipulátor, CNC bruska, pásový dopravník, programování manipulátoru, robotizované pracoviště.

KEYWORDS

Robot, manipulator, CNC grinding machine, band conveyor, manipulator programming, robotized workplace.

PODĚKOVÁNÍ

Rád bych poděkoval firmě ZKL Brno a.s. za vstřícnost a ochotu při zpracovávání daného úkolu. Zejména panu Vítězslavu Celému a panu Ondřeji Mágrovi za pomoc při řešení úkolu. Dále bych rád poděkoval vedoucímu práce panu Ing. Tomáši Maradovi, Ph.D. za věcné připomínky k mé bakalářské práci.

Obsah:

	Zadání závěrečné práce.....	3
	Licenční smlouva.....	5
	Abstrakt.....	7
	Poděkování.....	9
1	Úvod.....	13
2	Rozbor problému.....	15
2.1	Výběr manipulátoru.....	15
2.1.1	Dostupná řešení.....	15
2.1.2	Volba manipulátoru.....	16
2.2	Úprava manipulátoru.....	16
3	Programovací prostředí.....	19
3.1	Programování manipulátoru.....	19
3.1.1	Programování pohybu.....	20
3.2	Řídící funkce.....	21
3.3	Registry a práce s nimi.....	21
3.4	Vstupy a výstupy manipulátoru.....	22
4	Program.....	25
4.1	První verze programu.....	25
4.1.1	Rozbor problému.....	25
4.1.2	Řešení.....	25
4.1.3	Rozbor řešení.....	27
4.2	Druhá verze programu.....	27
4.2.1	Návrh modifikací.....	27
4.2.2	Řešení.....	27
4.2.3	Rozbor řešení.....	28
4.3	Konečný program.....	29
4.3.1	Návrh modifikací.....	29
4.3.2	Řešení.....	29
4.3.3	Celkový popis programu.....	30
4.3.4	Rozbor řešení.....	32
4.4	Problémy při zavádění do výroby.....	33
5	Závěr.....	35
	Seznam použité literatury.....	37

1 ÚVOD

Tato práce se zabývá zprovozněním robotického manipulátoru a jeho začlenění do stávající výroby ve firmě ZKL Brno a. s.

Ve výrobním bloku jsou opracovány vnitřní kroužky do ložisek (Obr. 1), které byly předtím soustruženy na požadovaný tvar a dále kaleny na požadovanou tvrdost. Na tomto pracovišti jsou kroužky opracovány na potřebné rozměry. Po opracování se kroužky již montují do ložisek a opouští výrobní areál jako hotový produkt.



Obr. 1 Vnitřní kroužek ložiska.

Manipulátor odebírá kroužky ze dvou pásových dopravníků, kterými zásobuje tři CNC brusky a nakonec opracované výrobky odkládá na EU paletu. Tyto palety má k dispozici také dvě, aby mohl být zabezpečen nepřetržitý provoz. V případě, že dojde k vyprázdnění dopravníků, nebo k naplnění obou palet, manipulátor přivolá obsluhu pomocí zvukového signálu.

2 ROZBOR PROBLÉMU

Problémy se zprovoznováním robotických manipulátorů se dají rozdělit do několika skupin. Je to jednak výběr samotného manipulátoru s ohledem na potřebné parametry jako jsou např. nosnost, délka ramene a snadná údržba. Další důležitá skupina jsou ohledy na připojené periferie, respektive na možnost komunikace s těmito periferiemi. Dalším důležitým parametrem je možnost snadného a efektivního programování.

Manipulátor musí být přizpůsoben přímo na míru každému úkolu zvlášť. Jedná se zejména o uchopovací zařízení, které se liší podle předmětu, který se má přenášet.

2.1 Výběr manipulátoru

2.1.1 Dostupná řešení

Výrobou a distribucí manipulátorů se mimo jiné zabývají následující firmy:

- a) FANUC Robotics Czech s.r.o., Praha
- b) ESY s.r.o., Liberec
- c) Energo - Robotics v.d., Liptovský Mikuláš, Slovenská republika
- d) ARC-H a.s., Plzeň
- e) ABB s.r.o., Praha

Tyto firmy jsou dodavateli japonských robotů FANUC (Obr. 2) a liší se pouze sortimentem dodávaných služeb. Pouze firma Energo - Robotics v.d. má ve své nabídce i roboty vyráběné společností ABB (Obr. 3). Společnost ABB s.r.o. Dodává pouze roboty vlastní výroby.



Obr. 2 FANUC model R-2000iB/210F.[1]



Obr. 3 ABB model IRB 6640.[2]

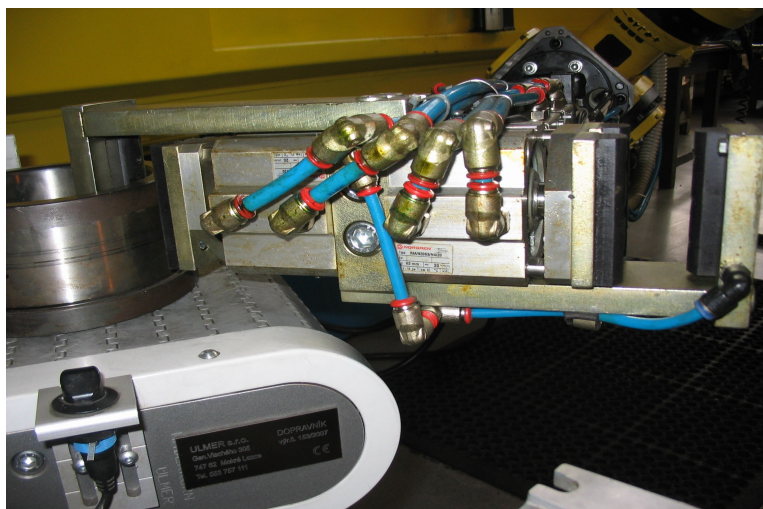
2.1.2 Volba manipulátoru

Všechny firmy dodávají roboty s různou nosností a délkou ramene. Obsluhované pracoviště je vzhledem k velikosti CNC brusek rozsáhlé a obráběné kroužky bývají větších hmotností, proto jsme museli vybírat z větších a odolnějších robotů. Konečný výběr robota prováděla firma ZKL Brno a. s. sama a já jsem již nadále pracoval s nainstalovaným zařízením.

Byl vybrán model FANUC R-2000iB/210F, který má nosnost 210 kg a maximální dosah 2 655 mm (Obr. 2) dodaný firmou ARC-H a.s., Plzeň.

2.2 Úprava manipulátoru

Robot musel být osazen manipulačními kleštěmi pro přenášení materiálu (Obr. 4). Byla zvolena možnost namontování dvojice kleští. Díky tomu je možné měnit obrobky v CNC bruskách bez zbytečných prostojů.



Obr. 4 Manipulační kleště pro přenášení materiálu.

Kleště jsou ovládány pomocí stlačeného vzduchu, který odebírá z rozvodné sítě, která je instalovaná přímo v hale. Stlačený vzduch je použit i pro očištění vřetene brusek. Tryska je také instalována na kleštích.

Dosah samotného robota nebyl dostačující a proto musely být kleště umístěny na další prodloužené rameno (Obr. 5).



Obr. 5 Robotický manipulátor i s potřebnými úpravami.

3 PROGRAMOVACÍ PROSTŘEDÍ

Programování se provádí pomocí ručního ovladače (Obr. 6).

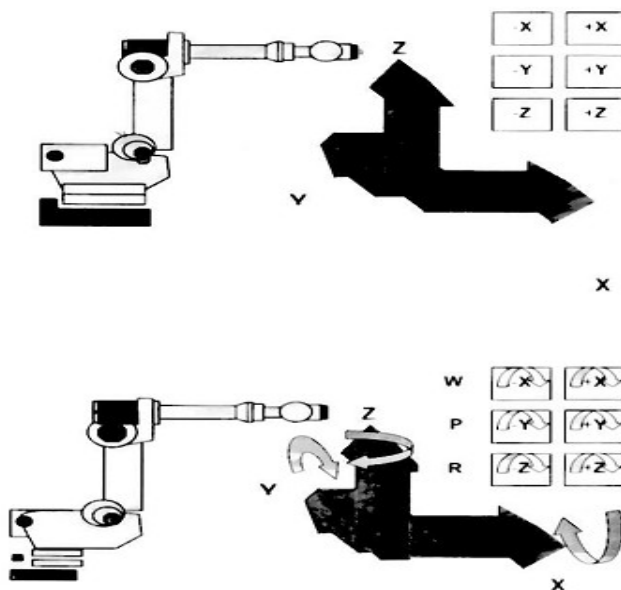


Obr. 6 Ruční ovladač manipulátoru.

3.1 Programování manipulátoru

Manipulátor se pohybuje v absolutních souřadnicích (Obr. 7), které jsou dány montáží manipulátoru.

Program skládáme z jednotlivých pohybů a funkcí pro řízení programu.

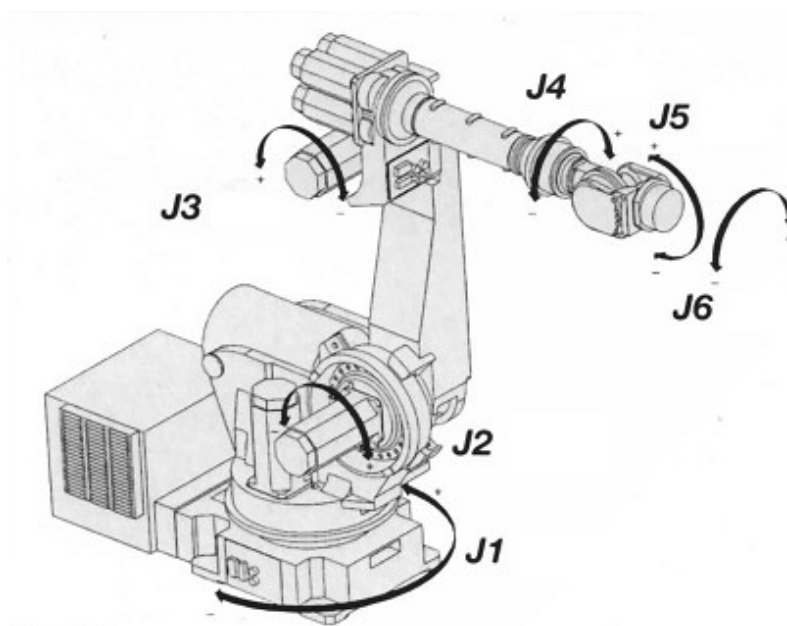


Obr. 7 Absolutní osy manipulátoru.[3]

3.1.1 Programování pohybu

Pohyb je vždy zadán způsobem pohybu, jeho cílovou souřadnicí, rychlostí, přesností dojezdu do bodu a nakonec se mohou upřesňovat další doplňkové informace jako např. rychlost rozjezdu a dojezdu.

Nejdříve se zadává způsob pohybu kde hodnota J je pohyb po osách (Obr. 8). Při tomto pohybu se všechny osy najednou rozjedou a zastaví se jakmile dosáhnou cílového bodu. Hodnota L je pohyb přímý a hodnota C je pohyb po kružnici. Pohyb po kružnici potřebuje navíc pomocný bod, který se zadává před bod cílový, aby byl vymezen směr pohybu. Pohyb jednotlivých os se pokaždé počítá s ohledem na požadovaný pohyb.

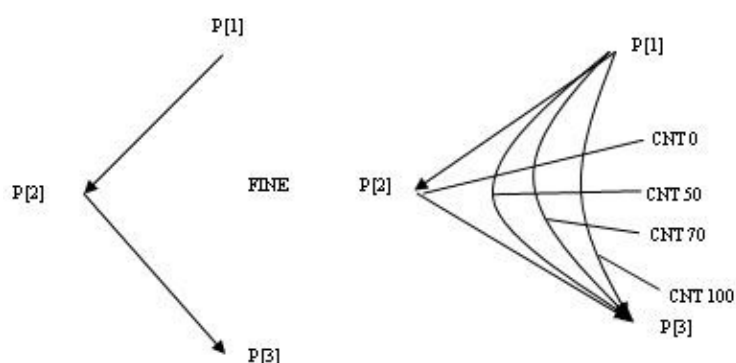


Obr. 8 Osy pohybu robota.[3]

Dále se zadává cílový bod. V programu se cílové body nezadávají přímo, ale pomocí registru ve kterém jsou uloženy jednotlivé souřadnice. Tyto souřadnice se ukládají ručním najetím do požadovaného bodu a uložením do registru.

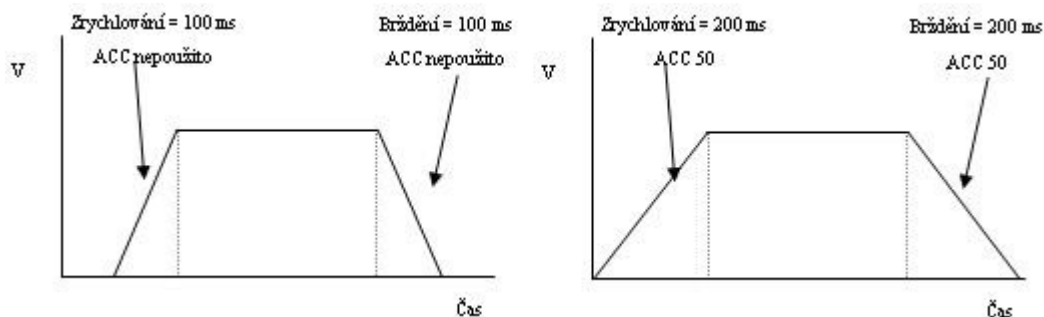
Potom se zadává rychlost pohybu. Tato hodnota se může zadávat v procentech maximální rychlosti, mm/s, cm/min, inch/s nebo pomocí registru ve kterém je zadána požadovaná hodnota.

Dalším parametrem je přesnost průjezdu cílovým bodem (Obr. 9). Tento parametr může nabývat hodnot FINE nebo CNT xxx. Za xxx se dosazuje požadovaná hodnota průjezdu přičemž CNT 0 odpovídá FINE. FINE je přesný průjezd bodem.



Obr. 9 Přesnost průjezdu cílovým bodem.[3]

Posledním parametrem je možnost upravit rychlost rozjezdu a dojezdu (Obr. 10). Hodnota může být v rozmezí 20-500%. Tímto parametrem se dá optimalizovat program.



Obr. 10 Diagram zrychlování a zpomalování.[3]

3.2 Řídící funkce

Při programování můžeme využít základní řídicí funkce. WAIT je funkce, která čeká dokud není splněná podmínka, nebo po stanovenou dobu. JMP_LBL je funkce pro skok v programu. LBL je návěstí na které se v programu přeskakuje. CALL je funkce volání podprogramu. Po ukončení podprogramu se pokračuje další instrukcí. IF je rozhodovací instrukce a SELECT je výběrový příkaz umožňující reagovat na více stavů daného registru.

3.3 Registry a práce s nimi

Registrů máme k dispozici celkem 255 a můžeme do nich ukládat pouze čísla. Registr označujeme R[1], kde 1 určuje číslo registru. Registr můžeme určovat buďto přímo jako je v předchozím příkladu, nebo nepřímo kdy jeden registr ukazuje na druhý R[R[2]]. Druhý příklad je totožný s prvním pokud v R[2] bude uložena hodnota 1. S registry můžeme provádět operace popsané v tabulce (Tab. 1).

Operace s registry	Příklad použití
Zadání	$R[1]=1$
Sčítání	$R[1]=R[1]+R[2]$
Odečítání	$R[1]=R[1]-R[2]$
Násobení	$R[1]=R[1]*R[2]$
Dělení	$R[1]=R[1]/R[2]$
Celočíselné dělení	$R[1]=R[1]DIV R[2]$
Zbytek po celočíselném dělení	$R[1]=R[1]MOD R[2]$

Tab. 1 Seznam operací při práci s registry.

Máme k dispozici i speciální druh registru, takzvaný poziční registr. Těchto registrů máme 200 a ukládáme do jich pomocné souřadnice bodů. Můžeme je zadávat buď přímo ve tvaru $PR[1]=(\text{slovo})$, nebo po jednotlivých osách ve tvaru $PR[1,1]=\text{číslo}$. S pozičními registry můžeme provádět stejné operace jako s registry (Tab. 1).

3.4 Vstupy a výstupy manipulátoru

Robot je opatřen řadou digitálních i analogových vstupů a výstupů. Digitálních vstupů a výstupů máme k dispozici 32. Analogové vstupy nejsou v tomto programu využity a proto zde nebudou popsány.

Rozlišujeme následující skupiny digitálních rozhraní robotové vstupy RO, robotové výstupy RI, digitální vstupy DO a digitální výstupy DI.

Robotové vstupy jsou využity k ovládání kleští a ofukovací trysky. Digitální vstupy a výstupy jsou použity ke komunikaci s CNC bruskami a obsluhou. Seznam vstupů a jejich využití je uvedeno v tabulce (Tab. 2).

Číslo	DO[Číslo]	DI[Číslo]
1	Home pozice	Home pozice
2	Fault	Reset
3	Prg running	Cykle start
4	Zalozit paletu 1	Paleta 1 tl. zal
5	Zalozit paletu 2	Paleta 2 tl. zal
6	E_STOP	Paleta 1 sp. zal
7		Paleta 2 sp. zal
8		JUNKER DOKONCENO
9	ODMAGNETOVAT JUN	JUNKER DVERE OTE
10	UPNOUT JUNKER	JUNKER ODEPNITO
11		JUNKER UPNUTO
12	ZALOZENOK	JUNKER AUT.
13		
14		
15		
16		
17	BDA 2 ODEPNI	BDA 2 AUT.
18	BDA 2 UPNI	BDA 2 DOKONCENO
19	BDA 2 ZALOZENOK	BDA 2 DVERE OTEV.
20	BDA ROBOT V AUT	BDA 2 ODEPNUTO
21	BDA 1 ODEPNI	BDA 2 UPNUTO
22	BDA 1 UPNI	BDA 1 AUT.
23	BDA 1 ZALOZENOK	BDA 1 DOKONCENO
24	BDA 1 REZ.	BDA 1 DVERE OTEV
25	Dopravnik 1	BDA 1 ODEPNUTO
26	Dopravnik 2	BDA 1 UPNUTO
27		Dopravnik 1praz
28		Dopravnik 2 praz
29		
30		
31		
32		

Tab. 2 Seznam digitálních vstupů a výstupů robota.

4 PROGRAM

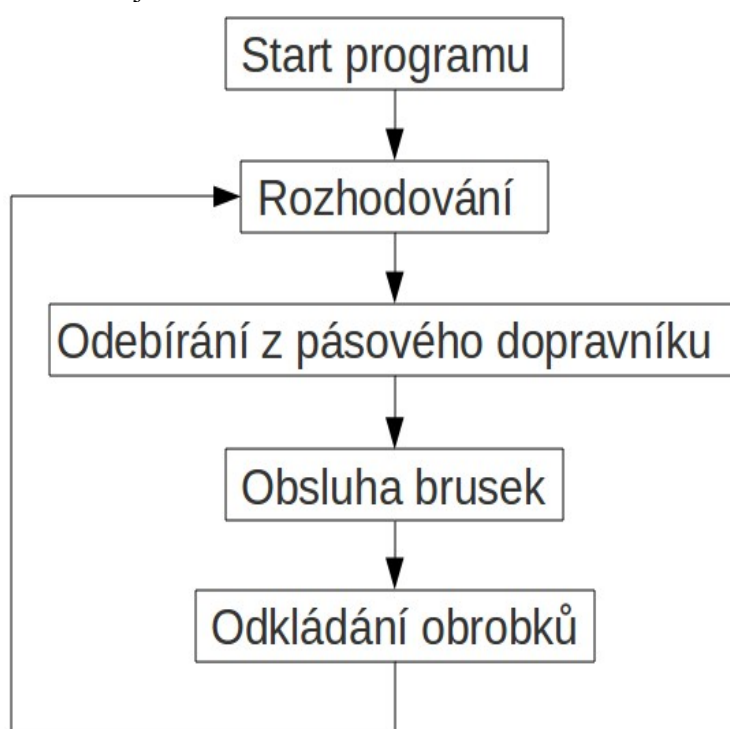
4.1 První verze programu

4.1.1 Rozbor problému

Je nutné vytvořit nový program, který bude odebírat obrobky z pásového dopravníku a obsluhovat CNC brusky. Obrobené součástky bude odkládat na připravenou paletu. Program musí být jednoduše modifikovatelný.

4.1.2 Řešení

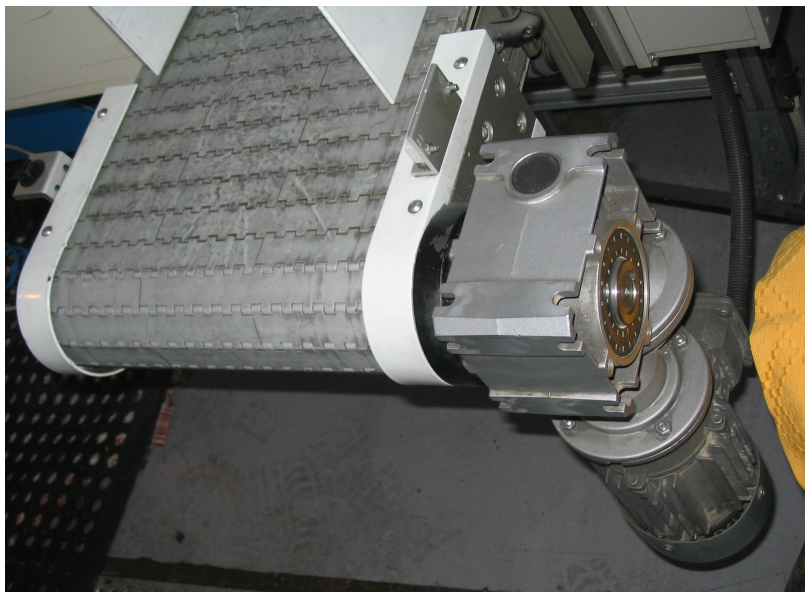
Celý program jsem rozdělil do následujících bloků: Rozhodovací část, odebírání z pásového dopravníku, obsluha brusek a odkládání obrobků (Obr. 11). Tyto bloky jsem označil návěstími a po dokončení každé dílčí akce jsem na toto návěstí odkazoval.



Obr. 11: Diagram programu.

Rozhodovací část pouze vyčkává na dokončení práce brusky, číslo této brusky je poté uloženo do registru na základě kterého se v dalších částech rozhoduje podmínkou, které zařízení bude obsluhováno.

Odebírací část zajišťuje odebrání neobrobeného obrobku z pásového dopravníku. Manipulátor přijede k dopravníku, otevře příslušné kleště a počká dokud nedojde k zastavení dopravníku. Dopravník je ovládán světelnou závorou tak, že popojíždí dokud není závora přerušena a jakmile obrobek přeruší světelný paprsek tak se dopravník zastaví (Obr. 12). Potom manipulátor najede do pozice ve které může uchytit obrobek a spustí program pro zavření kleští. Obrobek je potom zvednut z pásového dopravníku a manipulátor se přesune do výchozí pozice.



Obr. 12: Pásový dopravník v místě odběru obrobků.

Nejsložitější část programu s ohledem na přesnost a co do počtu prováděných kroků je část vyměňovací. Tato část provádí výměnu obrobků přímo v bruskách. Program nejdříve najede manipulátorem k brusce, ve které chce vyměnit obrobek, a potom provede sérii bezpečnostních kontrol. Nejdříve testuje zda je práce brusky ukončena, dále testuje zda již byl obrobek založen a nakonec se testuje zda jsou otevřené dveře. Po provedených kontrolách manipulátor najede blíže k brusce a otevře druhé kleště, než ve kterých má umístěný obrobek a potom najede do odebírací pozice dá příkaz k zavření kleští. Nyní musí dát brusce příkaz k odmagnetování a odepnutí obrobku. Jakmile manipulátor dostane informaci, že již bylo odmagnetování dokončeno tak se vzdálí od vřetene a provede ofouknutí upínací plochy stlačením vzduchem čímž se odstraní nečistoty. Díky tomu je nový obrobek dobře uchycen a nedochází k nepřesnostem vlivem špatného uchycení. Nyní manipulátor s oběma výrobky vyjede z brusky ven a otočí kleště, by bylo možné usadit nový obrobek. Nyní vjede opět do brusky a co nejpřesněji usadí nový obrobek na přidržovací trny. Čím bude usazení přesnější, tím se budou trny méně opotřebovávat a prodlouží se tím doba provozu mezi seřizovacími odstávkami. Po usazení se zavolá program pro otevření příslušných kleští a manipulátor může opustit brusku. Dále program oznámí brusce, že již bylo založeno a může spustit obráběcí operace. Manipulátor se nyní přemístí do výchozí pozice a program se přesune na návěstí další části.

Poslední část programu zajišťuje odkládání obrobků na paletu. Hned na začátku programu si ukládáme do registru průměr obrobku potřebný pro výpočty odkládacích souřadnic a potom přesuneme manipulátor k paletě. Nyní se provádí testování přítomnosti palety a zda-li již není paleta plná. Zde manipulátor najede do pozice ve které otevřením příslušných kleští provede odložení obrobku a dále odjede do bezpečné vzdálenosti od palety. Nyní je potřeba spočítat nové souřadnice pro odkládání obrobku. To je prováděno tak, že se k současné odkládací pozici připočte průměr obrobku z registru. Takto vypočítané souřadnice se porovnají s velikostí EU palety a pokud již není dostatek místa, tak je založena nová řada přičtením průměru do příslušného směru a pomocný registr je vynulován. Nyní program přeskočí na poslední návěstí této části a přesune manipulátor do základní pozice. Pokud ovšem program podmínkou na začátku zjistí, že paleta je již plná nejdříve přivolá obsluhu, která musí paletu vyměnit a poté zmáčknout tlačítko PALETA ZALOŽENA (Obr. 13). Nyní dojde k vynulování všech registrů a manipulátor začne skládat obrobky opět od začátku. Manipulátor se přesune do výchozí pozice a program se začne provádět zase od začátku.



Obr. 13: Tlačítko pro potvrzení založení nové palety.

4.1.3 Rozbor řešení

Klady:

- Velmi dobře navrhnutá část pro výměnu obrobků v bruskách.
- Návrat manipulátoru po každé provedené akci do výchozí pozice se projevil jako výhodný s ohledem na další vývoj programu.
- Osvědčila se několikanásobná kontrola před vstupem do brusky, kdy vlivem rušení z okolních starých strojů je občas detekován nesprávný signál.

Zápory:

- Program je příliš dlouhý a tím nepřehledný.
- Je složitá modifikace programu.
- Příliš mnoho často se opakujících částí programu.
- Paleta byla vždy příliš brzy plná.

Závěr:

Tento program nikdy nebyl plně nasazen do výroby a obsluhoval pouze jednu brusku. Pokus o jeho rozšíření na další brusky bylo příliš složité a neefektivní, a proto skončil již ve stádiu vývoje.

4.2 Druhá verze programu

4.2.1 Návrh modifikací

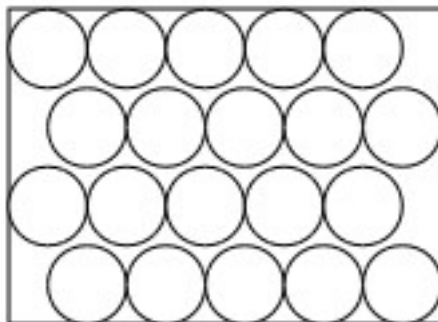
Je bezpodmínečně nutné rozdělit program na více menších programů a tyto programy volat z programu hlavního. Je vhodné zbavit se často opakující částí. Dále je potřeba vyřešit problém s brzkým naplněním palety.

4.2.2 Řešení

Nejdříve jsem výše popsany program přepracoval na ovládání pouze jedné brusky, tak že jsem odmazal odebrání z dopravníku pro druhou brusku a rozhodovací část.

Nyní jsem musel vyřešit problém s neefektivním odkládáním na paletu. Jediné možné řešení, které mohlo být zvoleno, je posunout souřadnici o poloměr vedle a potom je možné posunout obrobek kousek nahoru. (Obr. 14) Tato varianta mi umožnila založit další řadu a tím zvýšit obsazení palety. Po zvážení a odzkoušení několika variant s připočítáváním různých konstant jsem došel k závěru, že nejjednodušší bude uložit si do registru číslo právě plněné řady a dále se rozhodovat pomocí

příkazu SELECT, kterou řadu plnit. Každá z řad má vlastní počáteční pozici a další pozice v řadě jsou již dopočítávány s pomocí průměru.

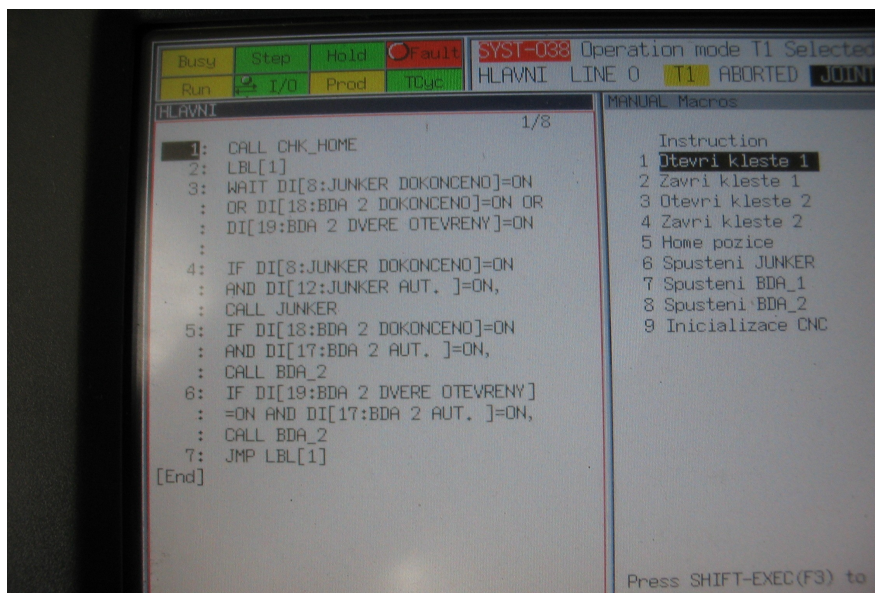


Obr. 14: Uspořádání obrobků na paletě.

Další na řadě byl problém často se opakujících kódů. Jednalo se zejména o návrat do výchozí polohy, dále jsem se rozhodl oddělit části pro obsluhu kleští a brusek. Veškerý kód jsem zkopíroval do nových programů a ty dále volal z programu pro obsluhu brusky. Touto operací se mi program značně zpřehlednil a proto jsem mohl přistoupit k dalšímu kroku.

Program jsem zkopíroval a jeho kopii jsem modifikoval tak, aby obsluhovala druhou brusku. Úprava byla jednoduchá, protože jsem pouze změnil body pohybu a registry použité při výpočtu odkládacích souřadnic.

Nyní již zbývalo pouze vytvořit řídicí funkci (Obr. 15). Jejím úkolem je vyčkávat na dokončení obráběcích operací a otevření dveří příslušné brusky. V okamžiku, kdy jsou tyto signály přijaty, jsou spuštěny předchozí programy, které již zajišťují samotnou výměnu obrobků. Je nutné, aby celý program byl umístěn v nekonečné smyčce z důvodu zajištění nepřetržitého provozu manipulátoru.



Obr. 15: Ukázka funkce která čeká na dokončení obráběcí operace brusky a dále spouští program na výměnu obrobků.

4.2.3 Rozbor řešení

Klady:

- Velmi se zpřehlednil kód programu.
- Vytvoření programu pro obsluhu další brusky by již nebylo příliš složité.
- Odkládání obrobků na paletu je již efektivní.

Zápory:

- Program byl v některých fázích příliš pomalý.

Požadavky:

- Seřizovači manipulátoru si přejí další rozdělení programu.
- Znemožnit obsluhu zvolení špatného programu.

Závěr:

Program byl nasazen do výroby a po několik týdnů testován. Během provozu došlo několikrát ke spuštění špatného programu obsluhou. Dále se objevil problém se seřizovači, pro které byl program stále příliš dlouhý a nepřehledný, proto je nutné jej ještě dále rozdělit na menší části.

4.3 Konečný program

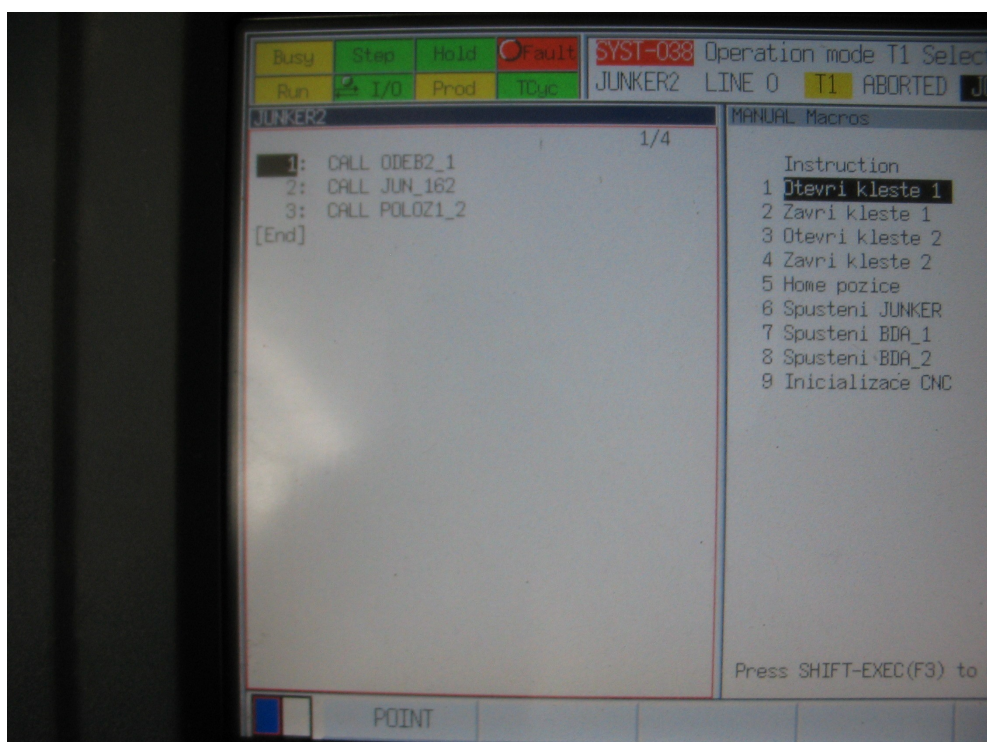
4.3.1 Návrh modifikací

Nyní je nutné ještě odstranit zápory a zapracovat požadavky popsané v kapitole 4.2.3.

4.3.2 Řešení

Při zapnutí manipulátoru je obsluze nabízen vždy pouze program s přednastaveným jménem, které je ve tvaru PNSxxxx kde x zastupuje číslici. Ostatní programy jsou obsluze skryty, aby nemohlo docházet k chybnému zvolení programu. Proto jsem vytvořil pouze jeden program s názvem PNS0001 a do něj jsem umístil pouze příkaz CALL HLAVNI, který zajišťuje spuštění řídicí funkce z předchozí verze programu (Obr. 15). Tento postup znemožnil obsluhu zvolit chybný program a zároveň umožňuje seřizovači pohodlné nastavování požadovaného programu.

Nyní bylo nutné program pro obsluhu brusky rozdělit na další části. Program jsem rozdělil na část pro odebrání z pásového dopravníku, část zajišťující výměnu obrobku v brusce a na závěrečnou část, která zajišťuje odkládání obrobků na paletu. Dále bylo nutno vytvořit program, který bude tyto programy postupně spouštět (Obr. 16). Tento jsem nazval stejně, jako je označení obsluhované brusky.

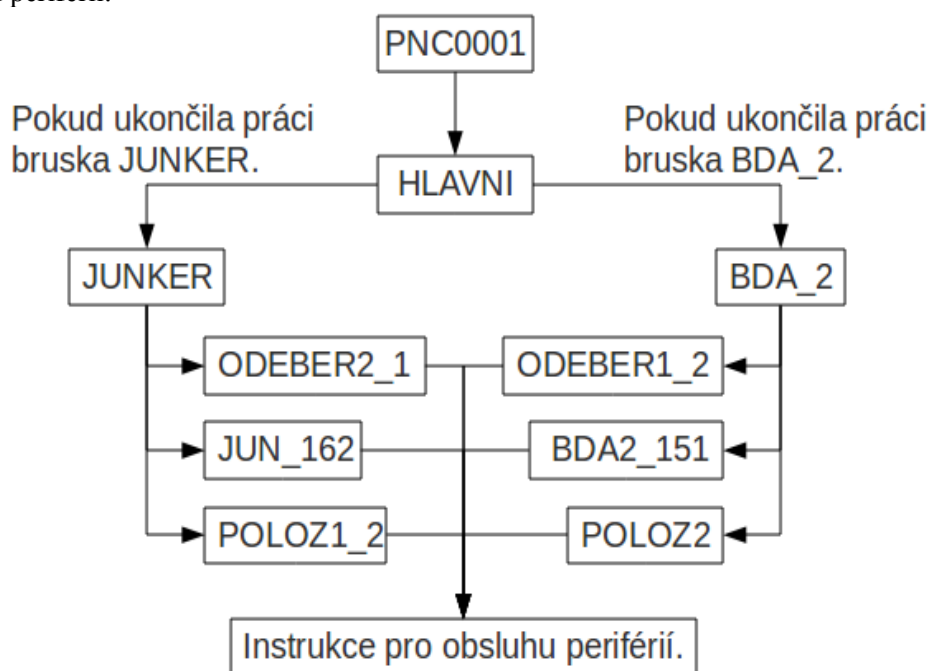


Obr. 16: Ukázka programu JUNKER2

Posledním úkolem bylo provést celkovou optimalizaci programu. Tento úkol byl proveden lepší volbou potřebných referenčních bodů a vymazáním nadbytečných pohybů manipulátoru. Další optimalizace byla provedena zvýšením rychlosti manipulátoru v místech, kde není nutný přesný pohyb. Zrychlení bylo nejcitelnější zejména u přesunu do základní polohy.

4.3.3 Celkový popis programu

Celý program je složen z několika dílčích programů, které jsou postupně volány (Obr. 17). Tyto programy se dělí na tři základní typy. Jedná se o typ programů řídicí, vykonávací a instrukce pro obsluhu periférií.

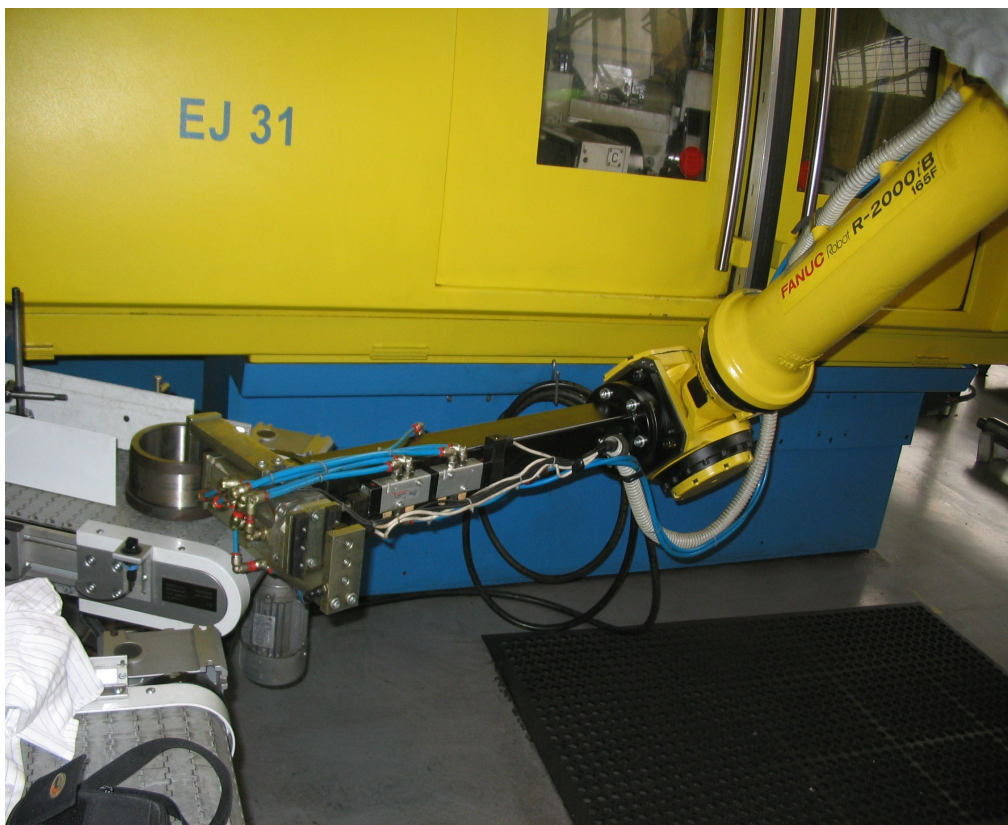


Obr. 17: Diagram spouštění dílčích programů.

Prvním typem jsou programy řídicí, které samy o sobě neprovádí žádný pohyb s manipulátorem. Sem patří program PNC0001, který je spuštěn obsluhou manipulátoru. Tento program spustí náš program HLA VNI (Obr. 15), jehož úkolem je rozhodnout, která bruska bude obsluhována. Nejdříve čeká na dokončení práce některé z brusek a dále spustí program pro brusku, která práci skončila. Při popisování budeme předpokládat, že práci ukončila bruska JUNKER, proto je spuštěn příkaz CALL JUNKER, který spustí program JUNKER (Obr. 16). Pro druhou brusku je popis všech programů stejný, protože programy se liší pouze souřadnicemi a použitými registry. Úkolem tohoto programu je pouze postupné spouštění programů vykonávacích.

Druhým typem programů jsou vykonávací. Jejich kód přímo ovládá manipulátor a obsahuje pouze nezbytně důležité množství rozhodovacích instrukcí. Tyto programy jsou již dobře popsány v kapitole 4.1.2, proto je budu popisovat jenom zběžně a zaměřím se hlavně na zajímavé detaily.

Nejdříve je spuštěn program pro odběr obrobku z pásového dopravníku (Obr. 18). V názvu programu mám zavedeny i dvě číslice. První z nich udává číslo pásového dopravníku, ze kterého se bude odebírat, a druhé číslo udává kleště do kterých se bude obrobek upínat. Z každého pásového dopravníku se odebírá do jiných kleští. Toto je výhodnější, protože na každém pásu bývá jiný typ obrobků, na jednom pásu jsou vnitřní a na druhém vnější kroužky. Značně to znesnadňuje chybu seřizovače při úpravě programu. Před vstupem do brusky se otevírají kleště, kterými budeme odebírat opracovanou součást a pokud by manipulátor vzal obrobek ze špatného dopravníku, došlo by k vypadnutí obrobku na zem. V opačném případě by mohlo dojít k poškození zařízení. Manipulátor by se pokusil odebrat obrobek již obsazenými kleštěmi.



Obr. 18: Manipulátor při odebrání obrobku z pásu.

Dalším spouštěným programem je JUN_162, který se stará o výměnu obrobku v brusce. Číslo v názvu programu určuje rozměr obrobku. Do budoucna by měl mít každý typ obrobku vlastní program. Tyto programy se budou vytvářet dle potřeby. Ukázka programu (Obr. 19) je velmi dobře komentovaná a proto není třeba žádný další popis.

```
J P[1] 80% CNT100
WAIT DI[8:JUNKER DOKONCENO]=ON           //Několik bezpečnostních kontrol.
DO[12:ZALOŽENO OK]=OFF
WAIT DI[9:JUNKER DVERE OTEVRENY]=ON
L P[2] 1000mm/sec FINE
Otevri kleste 2
...                                     //Nájezd manipulátoru do odebírací pozice.
Zavri kleste 2
DO[9:ODMAGNETOVAT JUNKER]=ON             //Odmagnetování a odepnutí obrobku.
WAIT DI[10:JUNKER ODEPNUTO]=ON
DO[9:ODMAGNETOVAT JUNKER]=OFF
... //Odebrání obrobku.
RO[2:Ofuk]=PULSE,8.0sec                 //Ofouknutí upínací plochy.
...                                     /* Výjezd z brusky, otočení a co nejpřesnější
                                         usazení nového obrobku. */
DO[10:UPNOUT JUNKER]=ON                  //Upnutí obrobku.
WAIT DI[11:JUNKER UPNUTO]=ON

Otevri kleste 1
DO[10:UPNOUT JUNKER]=OFF
...                                     //Opuštění brusky.
DO[12:ZALOŽENO OK]=ON                   //Spuštění obráběcího procesu.
CALL HOME_BOD                           //Návrat do výchozí pozice.
```

Obr. 19: Ukázka programu JUN_162

Nyní je třeba obrobek odložit a proto je zavolán program POLOZ1_2 (Obr. 20). Obdobně jako

u dopravníku zde první číslo ve jméně znamená paletu na kterou se odkládá a druhé číslo jsou kleště, které drží obrobek. V tomto programu je nutné použít pohyb s proměnnými souřadnicemi. Programování takového pohybu je obdobné jako u pohybu s pevnými souřadnicemi, ale za posledním parametrem je ještě použit speciální parametr Offset za kterým je označení pozičního registru (např.: PR[1]). Tento poziční registr se přičítá k pevnému bodu.

```

R[5:Offset paleta]=95 //Nastavení průměru obrobku.
J P[1] 100% CNT100 //Příjezd k paletě.
L P[2] 150mm/sec CNT100 ACC50
LBL[5]
IF R[1:Paleta plna]=2,JMP LBL[6] //Test zda je paleta plná. Pokud ano přejde na LBL[6].
WAIT DI[6:Paleta 1 sp. zalozeni]=ON //Čeká na založení palety.
PR[1:Offset pal.1]=PR[1:Offset pal.1]-PR[1:Offset pal.1] //Nastavení pozičního registru.
PR[1,2:Offset pal.1]=R[7:Pomocny offs.1]
SELECT R[8:Rada pal.1]=1,JMP LBL[1] //Volba řady do které se bude odkládat.
=2,JMP LBL[2]
=3,JMP LBL[3]
=4,JMP LBL[4]
LBL[6]
R[1:Paleta plna]=1 //Vynulování registrů.
R[7:Pomocny offs.1]=0
R[8:Rada pal.1]=1
UALM[3] //Alarm pro přivolání obsluhy.
JMP LBL[5]
LBL[1]
J P[4] 25% CNT100 Offset,PR[1:Offset pal.1] //Pokud plníme první řadu.
L P[3] 100mm/sec FINE Offset,PR[1:Offset pal.1] //Příjezd na odkládací pozici.
Otevri kleste 2
L P[5] 50mm/sec CNT100 Offset,PR[1:Offset pal.1] //Opuštění odkládací pozice.
L P[6] 200mm/sec CNT100 Offset,PR[1:Offset pal.1]
L P[7] 500mm/sec CNT100 Offset,PR[1:Offset pal.1]
R[7:Pomocny offs.1]=R[7:Pomocny offs.1]+R[5:Offset paleta] //Výpočet nové souřadnice.
IF R[7:Pomocny offs.1]<750,JMP LBL[7] //Pokud se nová pozice vejde na paletu tak přechod na LBL[7].
R[7:Pomocny offs.1]=0 //Přechod na další řadu.
R[8:Rada pal.1]=R[8:Rada pal.1]+1
JMP LBL[7] //Přeskočí instrukce pro další řady a přeskočí na konec.
... //Plnění dalších řad obdobné jako u první řady.
LBL[7]
J P[2] 50% CNT100 //Vzdálení do bezpečné polohy.
L P[1] 1000mm/sec CNT100
CALL HOME_BOD //Návrat do základní polohy.

```

Obr. 20: Ukázka programu POLOZI_2

Posledním typem programů je instrukce pro obsluhu periférií. Jejich jediným úkolem je zapisovat signály na digitální výstupy manipulátoru na které reagují brusky a kleště (Tab. 2). Jedinou výjimkou je program pro návrat do výchozí pozice, který zajišťuje přesun manipulátoru do základní polohy bez ohledu na předchozí pozici. Před spuštěním programu je pouze nutné, aby se manipulátor nenacházel uvnitř zařízení, proto vždy musíme přemístit manipulátor do bezpečné vzdálenosti.

4.3.4 Rozbor řešení

Klady:

- Program je plně funkční a nasazen do výroby.
- Program obsahuje dostatečné zajištění proti chybě obsluhy, seřizovače i chybné detekci signálu.
- Seřizovači jsou nyní schopni velmi efektivně měnit nastavení pro jiný rozměr obrobků.

Zápory:

- Manipulátor neobsluhuje brusku BDA_1.

Závěr:

Manipulátor je již plně nasazen do výroby a značně zefektivnil výrobu daného střediska. Pro brusku BDA_1 zatím není kapacitní využití a proto firma ZKL Brno a. s. nepožadovala její zprovoznění.

4.4 Problémy při zavádění do výroby

Během vytváření první verze programu se objevil problém s chybnou detekcí signálů. Sledováním tohoto problému jsem zjistil, že rušení pochází od několika starých strojů. Tyto stroje byly nadále prohlédnuty údržbáři firmy a bylo nalezeno několik drobných závad. Bohužel mi tyto závady nebyly blíže popsány.

Po opravách těchto strojů již k rušení nedocházelo, přesto jsem se rozhodl ponechat před vjezdem do brusky několik kontrol, aby v případě opakování problému nemohlo dojít k poškození brusky.

5 ZÁVĚR

Tato práce byla zaměřena na výběr a zprovoznění robotického manipulátoru. Manipulátor zajišťuje nepřetržitý provoz daného pracoviště s minimálními nároky na obsluhu. Zkrácením doby prostoje CNC brusek se podstatně zvýšila produktivita pracoviště.

Obsluha tohoto pracoviště se stará pouze o dostatek obrobků na zásobovacích dopravnících a o výměnu obsazených odkládacích palet za prázdné. V období mezi těmito úkony má obsluha čas se věnovat dalším úkolům na jiných strojích a manipulátor v případě potřeby obsluhu přivolá zvukovým signálem. Díky tomu je práce manipulátoru velmi efektivní.

Tento program se bude dále ještě rozšiřovat pro obsluhu CNC brusky BDA 1, která není dosud plně kapacitně vytížena.

Celý projekt vznikl za finanční podpory Evropské unie v rámci rozvoje průmyslové výroby.

SEZNAM POUŽITÉ LITERATURY

- [1] FANUC Robotics Czech s.r.o. *Katalog výrobků* [online]..[cit. 26.5.2009]. Dostupné z:<http://www.fanucrobotics.cz/products/robots/detail.asp?id=317>
- [2] ABB s.r.o. *Katalog výrobků* [online]..[cit. 26.5.2009]. Dostupné z:
<http://www.fanucrobotics.cz/products/robots/detail.asp?id=317>
- [3] Podklady pro školení:Obsluha a programování RJ-3. ARC-H a.s., FANUC Robotics, 2006