

PRZEMYSŁOWY INSTYTUT AUTOMATYKI I POMIARÓW
MERA-PIAP
Al. Jerozolimskie 202 02-222 Warszawa Telefon 23-70-81

Ośrodek Automatykacji Procesów Produkcji

440

A

Główny wykonawca mgr inż. Zbigniew Pilat

Wykonawcy mgr inż. Jacek Dunaj
mgr inż. Małgorzata Jacórzyńska-Smigiera
tech. Wacław Grzymała

Konsultant

Nr zlecenia

S 1258

Rozbudowa programu sterującego robotów IRp-6/60 z układem sterowania bazującym na nowej jednostce centralnej PIAP.

Etap 2. Współdział w badaniach robota z nowym układem sterowania, weryfikacja oprogramowania, dokumentacja, koncepcja rozwoju oprogramowania podstawowego robotów.

Zleceniodawca praca statutowa PIAP

Pracę rozpoczęto dnia 15.04.1992r

zakończono dnia 30.06.1992r

Kierownik Ośrodka

Z-ca Dyrektora
d/s. Badawczo-Rozwojowych

dr inż. Marian Wrzesień

dr inż. Jan Jabłkowski

Praca zawiera:

Rozdzielnik - ilość egz: 3

stron 22

Egz. 1 B0INTE

rysunków 2

Egz. 2 OAP

fotografii -

Egz. 3 OAP

tabel -

Egz. 4

tablic -

Egz. 5

załączników 1

Egz. 6

Nr rejestr. 6838

Analiza deskryptorowa

ROBOTY PRZEMYSŁOWE : UKŁAD STEROWANIA, OPROGRAMOWANIE

Analiza dokumentacyjna

Opracowanie zawiera opis oprogramowania układu sterowania robotów IRp-6/60 bazującego na nowej jednostce centralnej PIAP wg stanu na koniec czerwca 1992 roku.

Oprócz syntetycznego opisu zakresu funkcji realizowanych przez to oprogramowanie przedstawiono też propozycje dalszych prac rozwojowych i wnioski z pierwszych badań. Załącznikiem do niniejszego sprawozdania jest uaktualniona wersja podręcznika programowania robotów.

Tytuły poprzednich sprawozdań

UKD

PIAP 41/88 10000

S P I S T R E S C I

1.	WPROWADZENIE.....	4
2.	FUNKCJE REALIZOWANE PRZEZ PROGRAM STERUJĄCY ROBOTA.....	5
2.1.	Rozpoczęcie pracy. Obsługa panelu operacyjnego.....	8
2.2.	Panel programowania.....	8
2.3.	Ręczne poruszanie manipulatorem.....	9
2.4.	Ręczne operowanie systemem.....	9
2.5.	Programowanie robota.....	12
2.6.	Edycja programu użytkowego robota.....	14
2.7.	Start programu użytkowego robota.....	16
2.8.	Praca automatyczna.....	16
2.9.	Obsługa wykrytych błędów pracy robota.....	17
3.	WNIOSKI Z BADAN I PIERWSZYCH PROB EKSPLOATACYJNYCH. WERYFIKACJA OPROGRAMOWANIA PODSTAWOWEGO STEROWNIKA ROBOTA.....	18
4.	PROPOZYCJA DALSZYCH PRAC NAD PROGRAMEM STERUJĄCYM ROBOTA.....	20
5.	PODSUMOWANIE.....	21
	LITERATURA.....	22

1. WPROWADZENIE.

Od początku 1991 roku w Instytucie PIAP są prowadzone prace nad nową generacją układów sterowania dla robotów przemysłowych. Wśród wprowadzanych rozwiązań należy szczególnie podkreślić:

- zastosowanie magistrali AMS (identycznej z magistralą Multibus firmy INTEL) - standard firmy Siemens, przyjęty przez IEC pod symbolem IEC-47B (ew. IEEE-796),
- zastosowanie nowych układów i podzespołów elektronicznych,
- wykorzystanie w konstrukcji pakietów sterownika technologii druku wielowarstwowego,
- wprowadzenie cyfrowych sterowników położenia osi.

Przeprowadzone badania prototypów wykazały znaczny wzrost niezawodności i odporności na zakłócenia elektromagnetyczne nowego układu sterowania w porównaniu z poprzednimi modelami [3]. W tym zakresie uzyskano poziom, który pozwala optymistycznie patrzeć na pracę sterownika w warunkach przemysłowych z punktu widzenia bezawaryjności. O zdolnościach aplikacyjnych robotów obok cech mechanicznych ich manipulatorów i jakości układu sterowania w równym stopniu decydują możliwości funkcjonalne realizowane przez program sterujący. Naturalną więc kolejną rzeczą równoległą z pracami nad konstrukcją sprzętową wykonywane było oprogramowanie podstawowe nowego sterownika. W roku 1991, w ramach kontynuacji prac w CPBR 7.1 pt. "Roboty przemysłowe", wykonano program sterujący w zakresie funkcji umożliwiających przeprowadzenie badań układu sterowania [1], [2]. W bieżącym roku oprogramowanie podstawowe zostało znacznie rozbudowane tak, że możliwe jest już tworzenie programów użytkowych robota realizujących konkretne zadanie aplikacyjne. Przykładem może służyć ekspozycja na tegorocznych targach w Poznaniu.

Niniejsze sprawozdanie zawiera syntetyczny opis zakresu funkcji realizowanych przez aktualną wersję oprogramowania podstawowego układu sterowania robotów. Przedstawiono też dalsze prace rozwojowe, które zdaniem wykonawców powinny być prowadzone w pierwszej kolejności. Integralną częścią tego sprawozdania jest nowa, uaktualniona wersja podręcznika programowania robotów. Te dwa opracowania, wraz z dyskietką zawierającą postać źródłową oprogramowania i wzorcami pamięci EPROM (przekazane do Archiwum PIAP) stanowią komplet materiałów dokumentujących oprogramowanie układu sterowania robotów bazującego na nowej jednostce centralnej PIAP, według stanu na koniec czerwca bieżącego roku.

2. FUNKCJE REALIZOWANE PRZEZ PROGRAM STERUJĄCY ROBOTA.

Dużą zaletą jednostki centralnej MV-52 jest fakt, że dysponuje ona bardzo dużymi zasobami pamięci (pamięć RAM do 256 kB, EPROM do 512 kB). Dzięki temu możliwe było wykorzystanie przy pisaniu programu sterującego dużego modelu pamięci. Możliwe też było jednoczesne posadowienie na pakiecie MV-52 programu monitora operatorskiego. Całe oprogramowanie podstawowe robota zostało napisane w trzech językach:

- asembler procesora 80186 - ASM86
- PL/M-86
- C

Prace rozpoczęto przy wykorzystaniu posiadanego w Instytucie oprogramowania uruchomieniowego firmy Genesis Microsystems. Narzędzie to jest już mocno przestarzałe, zawiera pewne błędy i niedogodności (m. in. różna kolejność danych na stosie przy wywołaniu podprogramu w językach PL/M-86 i C, błędne inicjowanie parametrów procedur bibliotecznych języka C, błędne działanie procedur ze standardowej biblioteki we/wy języka C w zakresie wysyłania ciągów znakowych do urządzenia zewnętrznego, co jest wykorzystywane przy obsłudze panelu programowania), a posiadana dokumentacja nie jest kompletna. Już w trakcie prac zakupiono pakiet oryginalnego oprogramowania firmy Intel. Przeniesienie pod nowy system części pisanych w asemblerze i PL/M-86 okazało się stosunkowo proste i zostało wykonane niejako "z marszu". Różnice pomiędzy intelowskim kompilatorem języka C, a kompilatorem Mark Williams wykorzystywanym w systemie Genesis są dość znaczne. Dlatego też ta część oprogramowania robota nie została przeniesiona pod nowe narzędzia. Jest to jedno z pierwszych zadań, które należy wykonać przy kontynuacji prac nad programem sterującym.

Całe oprogramowanie układu sterowania robota zostało zorganizowane w katalogu o nazwie CTRLPGM. Jego strukturę przedstawiono na rys. 2.1. Oprogramowanie w takiej właśnie postaci zostało upakowane (kompresja danych) i przekazane na dyskietkach do Archiwum Instytutu. Rysunek 2.2 pokazuje wykorzystanie zasobów pamięci jednostki centralnej przez oprogramowanie podstawowe.

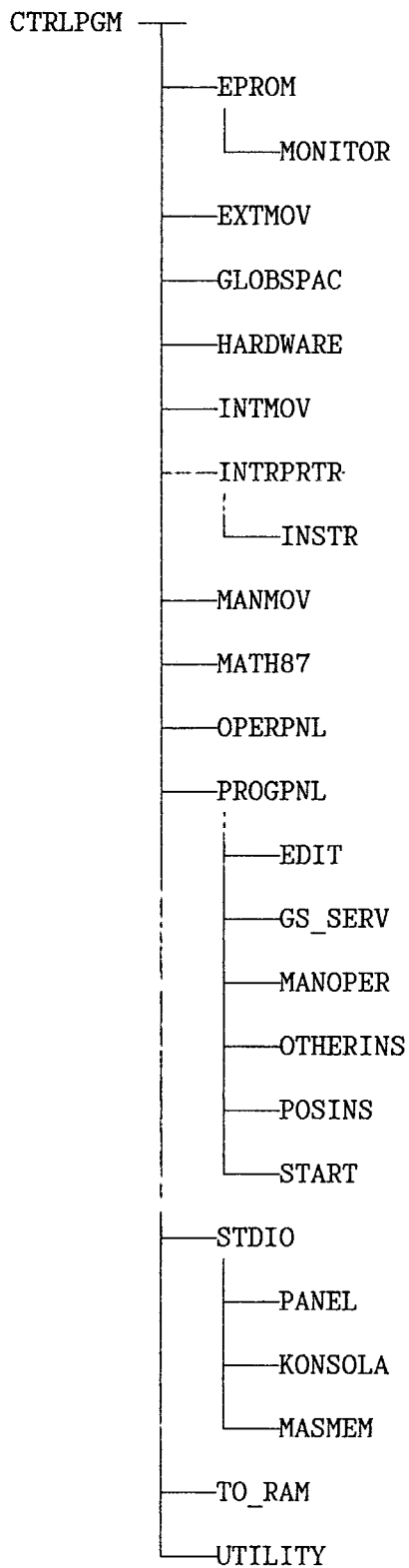
Jednostka centralna może komunikować się z zewnętrznymi urządzeniami inteligentnymi poprzez dwa kanały transmisji szeregowej oznaczone literami A i B. Oba są wykorzystywane przez program sterujący. Kanał A (napięciowy) został oprogramowany jako łącze do komputera zewnętrznego. Do kanału B (prądowy) jest dołączony panel programowania - program sterujący odbiera przysyłane z panelu kody wciśniętych przez operatora przycisków i wysyła ciągi znaków ASCII, które są wyświetlane na panelu.

Bardzo ważną rolę w programie sterującym robota odgrywają przerwania. Obecnie wykorzystuje się dwa przerwania:

- przerwanie 19 - generowane przez wewnętrzny układ timera procesora 80186 - realizacja programowa zegara systemowego o okresie 20ms,
- przerwanie 33 - zgłaszane przez układ transmisji szeregowej Z8530 - obsługa komunikacji z panelem programowania i komputerem zewnętrznym.

Po włączeniu zasilania wykonywana jest procedura restartu. Jest w niej inicjalizowany procesor 80186 oraz układy współpracujące. Następnie sprawdzany jest stan przełącznika TEST na płycie zespołu złącz (z przodu szafy, pod kasetą sterownika). Jeśli przełącznik ten jest włączony, to

sterowanie jest przekazywane do programu monitora. W przeciwnym wypadku rozpoczynany jest właściwy program sterujący robotą.



Rys. 2.1. Struktura katalogu, w którym zawarte jest oprogramowanie podstawowe robota.

00000H	Wektor przerwań
003FFH	
00400H	stos monitora
005FFH	
00600H	dane monitora
006FFH	
	obszar nie wykorzystany
00900H	stos programu sterującego
00DFH	
	obszar nie wykorzystany
01000H	dane programu sterującego
0FFFFH	
	obszar nie wykorzystany
70000H	okno o szerokości 64kbajty, przez które program sterujący ma dostęp do pakietu pamięci masowej MV-62
7FFFFH	
	obszar nie wykorzystany
E0000H	kod i stałe programu sterującego
F6AFFH	
	obszar nie wykorzystany
FA000H	kod i stałe monitora
FF6FFH	
	obszar nie wykorzystany
FFF70H	restart i inicjalizacja procesora 80186
FFFFFH	

Obsada pakietu.

RAM: 00000 - 1FFFFH

EPROM: E0000 - FFFFFH

Rys. 2.2. Wykorzystanie pamięci jednostki centralnej MV-52 przez oprogramowanie sterujące robotą.

2.1. Rozpoczęcie pracy. Obsługa panelu operacyjnego.

Po włączeniu zasilania (przełącznik na przednich drzwiach szafy ustawiony w pozycji "1" i wciśnięty przycisk "GOTOWOSC") i przekazaniu sterowania do części oprogramowania realizującej obsługę robota, wykonywane są programowe czynności inicjujące. O ich prawidłowym zakończeniu operator jest informowany mrugnięciem wszystkich lampek na panelu operacyjnym na okres ok. 1 sekundy. Na panelu programowania pojawia się napis informujący o wersji programu sterującego, typie manipulatora, dla którego wersja ta została przygotowana oraz data generacji wzorców pamięci EPROM. Napis ten znika po ok. 2 sekundach. Jeśli w pamięci RAM sterownika został zachowany program użytkowy (na skutek podtrzymania bateryjnego), to na panelu programowania pojawia się pierwsza instrukcja tego programu. Lampka oznaczająca utratę programu (na panelu operacyjnym) zostaje zgaszona. W przeciwnym wypadku lampka ta świeci się, a na panelu programowania wyświetlany jest domyślny numer pierwszej instrukcji programu (10). Na panelu operacyjnym programowo są obsługiwane także lampki oznaczające:

- automatyczne wykonywanie programu użytkowego,
- wykonywanie synchronizacji robota,
- obsługę pamięci masowej,
- wykrycie błędu w pracy robota.

Pozostałe lampki są zaświecane i gaszone sprzętowo. Na panelu operacyjnym znajduje się również grupa przycisków (w większości przyciski są zintegrowane z lampkami w jednej obudowie, z punktu widzenia oprogramowania są to jednak oddzielne urządzenia). Programowo są obsługiwane przyciski służące do:

- rozpoczęcia synchronizacji robota,
- rozpoczęcia automatycznego wykonywania programu użytkowego,
- rozpoczęcia odczytu programu użytkowego z pamięci masowej,
- zatrzymania automatycznego wykonywania programu użytkowego.

Przyciski przełączające robota ze stanu "GOTOWOSC" w stan "PRACA" i z powrotem są obsługiwane sprzętowo. Program sterujący może określać aktualny stan pracy robota odczytując sygnał z odpowiedniego stycznika. Podobnie wygląda sytuacja ze stanem STOP AWARYJNY. Sygnał ze stycznika w obwodzie załączania tego stanu jest odczytywany przez program sterujący.

2.2. Panel programowania.

Koncepcja komunikacji człowiek-maszyna za pomocą panelu programowania została zachowana taka jak w poprzednich modelach robotów IRp. Podstawową cechą tej koncepcji jest konwersacyjny system pracy z robotem. Znakomicie pomaga w tym pięciopozycyjne menu, zorganizowane w dolnej linii wyświetlacz panelu. Z punktu widzenia realizowanych funkcji wyróżnia się pięć możliwych stanów pracy panelu programowania:

- stan 1 - programowanie instrukcji innych niż pozycjonowanie,
- stan 2 - programowanie instrukcji pozycjonowania,
- stan 3 - start programu,

- stan 4 - edycja programu,
- stan 5 - ręczne operowanie systemem.

W następujących podrozdziałach opisano funkcjonowanie robota w poszczególnych stanach.

2.3. Ręczne poruszanie manipulatorem.

Poruszanie robotem jest jedną z najczęściej wykonywanych przez operatora czynności. Wykorzystuje on przy tym panel programowania i znajdujące się tam przyciski. Obecny etap prac nad układem sterowania robotów z nową jednostką centralną, obejmuje tylko jeden wariant wykonania panelu - z przyciskami do ręcznego poruszania manipulatorem. W poprzednich opracowaniach IRp stosowano również joystick, rozwiązanie to jednak nie sprawdziło się (głównie z powodu braku dostępu do joysticków odpowiedniej jakości). Ponieważ powstało wówczas oprogramowanie uwzględniające obsługę joysticka, realizujące to procedury zostały włączone do omawianego, aktualnego oprogramowania. Procesor sterujący panelu programowania rozpoznaje, jaki typ urządzenia do poruszania robotem ma zamontowany. Podczas inicjalizacji komunikacji panel-jednostka centralna, program sterujący za pomocą odpowiedniej sekwencji przesyłek identyfikuje czy ma do czynienia z panelem z przyciskami, czy z joystickiem. Obecnie realizowany jest ten pierwszy wariant. Jeśli jednak pojawi się potrzeba wykonania robota sterowanego joystickiem, z punktu widzenia oprogramowania zadanie to sprowadzi się do sprawdzenia działania tej części oprogramowania.

Operator może poruszać robotem we wszystkich wymienionych w poprzednim punkcie stanach panelu programowania. Przy ręcznym sterowaniu manipulator robota IRp może się poruszać w dwóch różnych układach współrzędnych:

- układ współrzędnych wewnętrznych (nazywanych też współrzędnymi robotowymi, związane są one z osiami robota),
- układ bazowy współrzędnych kartezjańskich związany z podstawą robota.

Przed dokonaniem synchronizacji operowanie robotem możliwe jest tylko w układzie współrzędnych wewnętrznych. Po wykonaniu synchronizacji można poruszać manipulatorem również we współrzędnych kartezjańskich.

Jeśli zachodzi konieczność bardzo dokładnego ustawienia robota, to można w tym celu posłużyć się ruchem w inkrementach. Jest on realizowany tylko we współrzędnych wewnętrznych. Każda oś jest poruszana osobno.

Na panelu znajduje się dwanaście przycisków do poruszania robotem: sześć, których naciskanie umożliwia ruch manipulatora w kierunku dodatnim oraz sześć, których naciskanie wymusza ruch w kierunku ujemnym. Przyciskom tym mogą być przyporządkowane różne ruchy robota lub jego osi zewnętrznych. Przyporządkowania tego dokonuje się wybierając układ współrzędnych i osie sterowane. Ruch elementu sterowanego odbywa się tak długo, jak długo pozostaje naciśnięty odpowiedni przycisk. Ruch jest wykonywany ze stałą prędkością. Wartość tej prędkości może być zmieniana za pomocą funkcji V_{MAN} (p.2.4).

2.4. Ręczne operowanie systemem.

Poniżej zostaną wyszczególnione funkcje realizowane przez oprogramowanie podstawowe robota w stanie ręcznego operowania systemem.

- Zapamiętanie aktualnego położenia jako punktu odniesienia - funkcja

PKTODN - służy do określenia pozycji odniesienia, względem której zostaną obliczone względne współrzędne ruchu przy programowaniu instrukcji pozycjonowania z pozycją względną;

- Odczyt lub zmiana parametrów narzędzia oraz wybór narzędzia aktywnego - funkcja NARZĘDZIE - można zdefiniować do 9 narzędzi (o numerach od 1 do 9). System sterowania robota zapewni trzy sposoby definiowania narzędzia:

- ręczny pełny (podanie współrzędnych TCP i orientacji narzędzia),
- ręczny skrócony (podanie tylko współrzędnych TCP narzędzia)
- automatyczny.

Definiowanie ręczne polega na wprowadzeniu przez operatora wartości parametrów narzędzia o zadeklarowanym numerze za pomocą klawiatury numerycznej. Korzystając z systemu menu operator wybiera, jaką definicję chce podać: pełną czy skróconą. W zależności od tego wyboru, system zażąda podania odpowiednich parametrów. Definiowanie automatyczne polega na wyznaczeniu współrzędnych punktu roboczego TCP narzędzia definiowanego drogą porównania przez system sterowania z narzędziem o numerze 0 (domyślne - patrz niżej). Operator doprowadza środek kołnierza przegubu robota do ustalonej pozycji w przestrzeni, co zapamiętane jest przez system jako punkt startowy, a następnie doprowadza do tego miejsca punkt roboczy TCP narzędzia definiowanego, co zapamiętane jest jako punkt końcowy. Z różnicy obu pozycji system wylicza parametry narzędzia definiowanego.

W systemie istnieje wyróżnione narzędzie o numerze 0 - tzw. narzędzie domyślne, hipotetyczne, systemowe. Jest ono zamocowane w środku kołnierza robota. Definicji tego narzędzia użytkownik nie może zmienić.

- Przepisywanie programu z pamięci wewnętrznej do pamięci masowej - funkcja ZAPIS - przeznaczona jest do zapamiętania w pamięci masowej obrazu obszaru programów użytkowych, tzn. programu i zdefiniowanych przez użytkownika parametrów pracy robota. W pamięci masowej może być zapisana liczba programów ograniczona jedynie pojemnością pamięci (pakiet EEPROM o pojemności do 512kbajtów). Aby zapisane programy można było rozróżnić, wprowadzono ich nazwy - ośmioznakowe ciągi znaków klawiatury numerycznej panelu programowania. Dlatego też przy zapisie programu do pamięci masowej trzeba podać jego nazwę.
- Przepisywanie programu z pamięci masowej do pamięci wewnętrznej - funkcja ODCZYT - przeznaczona jest do odtworzenia, uprzednio zapamiętanego w pamięci masowej programu i zdefiniowanych przez użytkownika parametrów pracy robota. Podobnie jak przy zapisie trzeba podać nazwę odczytywanego programu.
- Odczyt wielkości wolnego obszaru pamięci użytkowej - funkcja PAMIĘĆ - pozwala na odczytanie wielkości pamięci wewnętrznej (RAM) jaka jest jeszcze do wykorzystania. Wielkość ta podawana jest jako procentowy stosunek wielkości pamięci jeszcze nie zajętej do całkowitej wielkości pamięci dostępnej użytkownikowi.
- Odczyt aktualnego położenia robota - funkcja AKTPOZ - pozwala na odczytanie aktualnego położenia manipulatora robota - pozycji wewnętrznej lub zewnętrznej (we współrzędnych kartezjańskich). Pozycja wewnętrzna podana jest w inkrementach, pozycja zewnętrzna - odpowiednio w milimetrach i stopniach.

- Zmiana prędkości ruchu manipulatora przy sterowaniu ręcznym - funkcja V_MAN - umożliwia zmianę prędkości ruchu manipulatora przy sterowaniu ręcznym. Zadawana prędkość ruchu jest podawana w procentach maksymalnej prędkości przy sterowaniu ręcznym robota (wynosi ona 25% maksymalnej prędkości konstrukcyjnej robota. Można wybrać jedną z pięciu wartości: 100, 50, 25, 10 lub 5%.
- Odczyt stanu wejść, odczyt i zmiana stanu wyjść dwustanowych - funkcja WE/WY - dzięki niej użytkownik może oglądać aktualny stan wejść i wyjść dwustanowych robota. Może również ustawić konkretną wartość 0 lub 1 na wybranym wyjściu.
- Obsługa pamięci masowej - menu drugiego poziomu MEMSERV - obejmuje ono cztery funkcje zapewniające pełną współpracę z pakietem pamięci masowej. Są to:
 - formatowanie pakietu pamięci masowej - funkcja FORMAT - sprawdzana jest obsada pakietu układami EEPROM (muszą być przynajmniej dwa takie układy), testowane są wszystkie komórki pamięci na zapis/odczyt, sporządzana jest mapa pamięci dostępnej i zapamiętywany jest jej rozmiar.
 - przeglądanie pamięci masowej - funkcja DIR - wyświetlane są nazwy i długości poszczególnych programów zapisanych w pakiecie pamięci masowej. Dla celów diagnostycznych podawany jest również numer sektora (pamięć jest programowo podzielona na sektory po 256 bajtów), w którym zapisany jest początek programu. Przewidziano opcję jednoczesnego wysyłania tych informacji poprzez komputerowy kanał szeregowy do zewnętrznego komputera lub drukarki. Po przekazaniu danych o wszystkich programach, na panelu pokazuje się napis, informujący o wielkości wolnego jeszcze obszaru pamięci masowej.
 - zmiana nazwy programu w pamięci masowej - funkcja RENAME.
 - usunięcie programu z pamięci masowej - funkcja DELETE.
- Obsługa współpracy z komputerem - menu drugiego poziomu KOMPUTER - obejmuje ono trzy funkcje realizujące obsługę kanału A transmisji szeregowej na jednostce centralnej. Są to:
 - ustawienie parametrów transmisji łącza - funkcja PARAMETR - możliwe jest przeprogramowanie układu transmisji szeregowej i zmiana prędkości przesyłania, parzystości, długości słowa i liczby bitów stopu. Umożliwia to współpracę układu sterowania robota praktycznie z dowolnym urządzeniem posiadającym kanał typu V-24.
 - przesłanie programu użytkowego do komputera zewnętrznego - funkcja SEND - można przesłać program będący aktualnie w pamięci RAM jednostki centralnej lub wskazany program z pamięci masowej. Dzięki tej funkcji możliwe już jest tworzenie archiwum programów użytkowych robota na dysku komputera lub na dyskietkach.
 - załadowanie programu użytkowego z komputera zewnętrznego - funkcja LOAD - można przesłać program z komputera zewnętrznego do pamięci RAM jednostki centralnej. Podczas wykonywania tej funkcji jednostka jest w stanie gotowości do odczytu. Od strony komputera

można więc realizować przesłanie zwykłą komendą DOS-a np. 'copy'. Funkcje SEND i LOAD umożliwiają praktycznie przenoszenie programów użytkowych między różnymi robotami.

2.5. Programowanie robota.

Ogólne zasady programowania robota zostały zachowane zgodnie z koncepcją przyjętą we wcześniejszych pracach nad robotami IRp. Program użytkowy tworzony jest z instrukcji. Każda instrukcja ma swój numer i typ. Niektóre instrukcje posiadają także parametry określające sposób wykonania. Parametry mogą być numeryczne tj. liczbowe (np. numer chwytaka) i nienumeryczne (np. zadany stan chwytaka - otwarty lub zamknięty). Numer instrukcji może być dowolną liczbą całkowitą z przedziału od 1 do 9999. W trakcie programowania kolejnych, nowych instrukcji numery automatycznie zwiększane są o 10. Za numerem instrukcji znajduje się mnemonik (skrót nazwy) określający typ instrukcji. Za mnemonikiem znajdują się parametry - jeśli instrukcja takowe posiada.

Programowanie rozpoczyna się od wybrania typu instrukcji (przez naciśnięcie przycisku wprowadzającego panel programowania w stan programowania instrukcji i wybranie z menu związanego z przyciskiem odpowiedniego typu). W górnej linii wyświetlacza za numerem instrukcji zostaje wpisany mnemonik określający jej typ. Dalsze postępowanie zależy od typu instrukcji. Jeśli potrzebne są parametry numeryczne, to wprowadza się je korzystając z klawiatury numerycznej.

Wszystkie instrukcje programowane przez użytkownika podzielono na dwie grupy:

- instrukcje pozycjonowania (związane z ruchem)
- inne instrukcje (sterujące, deklaracyjne, wejścia/wyjścia).

Zostanie teraz przedstawiony zestaw programowanych instrukcji robota realizowanych przez program sterujący.

Instrukcje pozycjonowania.

- Ruch z interpolacją liniową - instrukcja POZ LIN - jest to taki ruch, w którym punkt roboczy narzędzia (TCP) porusza się po linii prostej od pozycji startowej (gdzie znajduje się w momencie rozpoczęcia wykonywania instrukcji) do pozycji zapamiętanej w instrukcji, z zaprogramowaną prędkością. Podczas wykonywania ruchu orientacja narzędzia zmienia się ze stałą prędkością kątową. Jeśli orientacja narzędzia w punkcie startowym jest taka jak w docelowym, jest ona stała na całym odcinku ruchu. W trakcie programowania instrukcji pozycjonowania z interpolacją liniową operator określa następujące parametry:

- prędkość ruchu,
 - sposób dojścia do punktu końcowego (dokładnie lub zgrubnie),
 - sposób pamiętania pozycji (bezwzględna lub względna).
- Ruch z interpolacją quasiliniową - instrukcja POZ QLIN wykonanie tej instrukcji polega na przeprowadzeniu punktu roboczego narzędzia z aktualnego położenia do położenia zapamiętanego w instrukcji w taki sposób, aby poszczególne osie manipulatora poruszały się ze stałymi prędkościami kątowymi, przy czym wszystkie osie jednocześnie rozpoczynają i jednocześnie kończą ruch. W trakcie programowania

instrukcji pozycjonowania z interpolacją quasiliniową operator określa następujące parametry:

- prędkość lub czas ruchu,
- sposób dojścia do punktu końcowego (tak jak w POZ LIN),
- sposób pamiętania pozycji (tak jak w POZ LIN),

Instrukcje inne niż pozycjonowanie.

- Wybór definicji narzędzia - instrukcja NARZĘDZIE - wskazuje systemowi sterowania robota, jakim narzędziem będzie operować robot. Jedynym parametrem instrukcji jest numer narzędzia. Jeśli w programie użytkowym nie występuje instrukcja NARZĘDZIE, system sterowania przyjmuje domyślnie, że pracuje z narzędziem hipotetycznym (por. p.2.4).
- Skok bezwarunkowy i warunkowy - instrukcja SKOK - skok bezwarunkowy wykonywany jest do wskazanej instrukcji bez względu na okoliczności (oczywiście jeśli program zawiera tę instrukcję). Skok warunkowy wykonuje się, jeśli spełniony jest zaprogramowany warunek. W przeciwnym przypadku wykonywanie programu jest kontynuowane od instrukcji znajdującej się bezpośrednio za instrukcją skoku. Skok warunkowy może być uwarunkowany stanem flagi, wyjścia lub wejścia.
- Deklaracja prędkości maksymalnej i podstawowej - instrukcja PRĘDKOSC - prędkość podstawowa i maksymalna może być zmieniana w dowolnym miejscu programu instrukcją PRĘDKOSC. Podczas wykonywania instrukcji pozycjonowania liniowego sprawdzane jest, czy prędkość ruchu nie jest większa od aktualnie zadeklarowanej prędkości maksymalnej. Prędkość podstawową i maksymalną podaje się w milimetrach na sekundę (mm/s), a zakres zmienności jest od 1 do 9999 mm/s. Prędkość podstawowa musi być przy tym nie większa niż prędkość maksymalna.
- Czekanie bezwarunkowe i warunkowe - instrukcja CZEKAJ - czekanie bezwarunkowe polega na wstrzymaniu wykonywania ciągu instrukcji na przeciąg zadeklarowanego czasu. Czekanie warunkowe polega na oczekiwaniu na spełnienie zaprogramowanego warunku. Oznacza to, że wykonywanie programu użytkowego będzie wstrzymane aż do chwili, w której warunek będzie spełniony. Czekanie warunkowe może być uzależnione od stanu flagi, wyjść lub wejść.
- Ustawianie wyjścia lub flagi - instrukcja WY/FLAG - ustawia wartość wyjścia lub flagi na wartość podaną wprost lub wartość reprezentowaną przez stan flagi, wyjścia lub wejścia. W programie można wykorzystać jednocześnie do 64 flag. Wejść i wyjść dwustanowych w podstawowej wersji robota jest po 16. Liczba ta może być zwiększona do 64.
- Operacje na chwytakach - instrukcja CHWYTAK - jej parametrem jest numer chwytaka (1 lub 2) oraz stan, jaki ma przyjąć chwytak (zamknięty lub otwarty). Dodatkowym parametrem jest czas wykonania operacji na chwytaku - czas na zareagowanie chwytaka.
- Tworzenie pętli programowej - instrukcje POWTORZ, KONPOWT - pętla programowa umożliwia powtarzanie wykonywania dowolnego ciągu instrukcji umieszczonego wewnątrz tej pętli. Każda pętla programowa składa się z:
 - instrukcji początku pętli programowej (POWTORZ),

- ciągu instrukcji wewnątrz pętli, których wykonanie jest powtarzane,
- instrukcji końca pętli programowej (KONPOWT).

Pętla programowa może zawierać w sobie inne pętle programowe, przy czym instrukcja końca pętli wewnętrznej musi znajdować się przed instrukcją końca pętli zewnętrznej. Dopuszcza się 8 poziomów zagnieżdżenia, tzn. że można zaprogramować do ośmiu pętli umieszczonych jedna w drugiej. Niedozwolone jest wykonanie instrukcji końca pętli bez uprzedniego wykonania instrukcji początku pętli.

- Tworzenie podprogramów - instrukcje PODPROG, POWROT - w programach użytkowych mogą znajdować się podprogramy, tj. bloki instrukcji, które mogą być wykonywane w programie za każdym razem, kiedy zostaną wywołane. Podprogram składa się z:
 - instrukcji początku podprogramu (POPLOG),
 - ciągu instrukcji wewnątrz podprogramu,
 - instrukcji powrotu z podprogramu (POWROT), czyli instrukcji kończącej podprogram.

Obie instrukcje PODPROG i POWROT nie mają parametrów. Liczba podprogramów w programie użytkownika może być nie większa, niż 255, a maksymalna liczba zagnieżdżenia wynosi 10.

- Wywołanie podprogramów - instrukcje WYKONAJ, POCZĄTEK i KONIEC - parametrem tych instrukcji jest numer pierwszej instrukcji wywoływanego podprogramu. Wprowadzenie trzech sposobów wywołania podprogramu wynika z uwzględnienia różnych okoliczności, w których sięgnięcie do podprogramu jest uzasadnione. Zadania poszczególnych instrukcji są następujące:
 - instrukcja WYKONAJ - pozwala na wywołanie podprogramu w dowolnym miejscu programu użytkowego, dowolną liczbę razy. Po wykonaniu podprogramu, system przechodzi do instrukcji następnej za WYKONAJ.
 - instrukcja POCZĄTEK - pozwala na jednorazowe wykonanie unikalnego podprogramu (nie mogą się w nim znaleźć instrukcje pozycjonowania) na początku pracy robota z danym programem użytkowym w trybie automatycznym (por. p. 2.7., 2.8.).
 - instrukcja KONIEC - pozwala na jednorazowe wykonanie unikalnego podprogramu (nie mogą się w nim znaleźć instrukcje pozycjonowania) na zakończenie pracy automatycznej robota z danym programem użytkowym.

Instrukcje POCZĄTEK i KONIEC są bardzo przydatne w aplikacjach, w których robot steruje zewnętrznymi urządzeniami technologicznymi (np. spawarka). Można dzięki tym instrukcjom ustalać stan urządzeń zewnętrznych na początku programu jak też w sytuacjach przerwania pracy automatycznej. Dotyczy to również nieoczekiwanego wstrzymania wykonywania programu użytkowego, np. na skutek awarii. Wprowadzenie tych instrukcji znacznie podnosi bezpieczeństwo pracy robota na stanowisku produkcyjnym.

2.6. Edycja programu użytkowego robota.

Mechanizm edycji programu użytkowego robota nie budził zastrzeżeń użytkowników poprzednich modeli IRp. Dlatego też został w całości

przeniesiony do oprogramowania układu sterowania opartego na nowej jednostce centralnej. Dla porządku poniżej wyszczególniono funkcje dostępne w edytorze robota.

- Wyświetlenie następnej instrukcji - komenda PRZOD - umożliwia przejście do następnej zaprogramowanej instrukcji programu. Jeżeli istnieje w programie instrukcja o numerze większym od aktualnie wyświetlonego, to jej tekst pojawi się na górnym wyświetlaczu. W przeciwnym wypadku na około 1 sekundę w górnym wyświetlaczu pojawi się komunikat "KONIEC PROGRAMU".
- Wyświetlenie poprzedniej instrukcji - komenda TYL - umożliwia przejście do instrukcji poprzedzającej aktualnie wyświetlaną instrukcję. Jeżeli jest taka instrukcja, to jej tekst zostaje wyświetlony na wyświetlaczu. Jeśli nie, to na około 1 sekundę na górnym wyświetlaczu pojawi się komunikat: "POCZĄTEK PROGRAMU".
- Usunięcie instrukcji - komenda USUN - daje możliwość usunięcia instrukcji, której tekst pokazany jest na wyświetlaczu, bądź też usunięcia bloku instrukcji (od numeru instrukcji pokazanej na wyświetlaczu aż do instrukcji o numerze wskazanym przez operatora).
- Wstawienie instrukcji między dwie inne - komenda WSTAW - umożliwia szybkie wstawienie instrukcji pomiędzy instrukcję aktualnie wyświetloną a następną. Wartość numeru instrukcji, który zostanie wyświetlony na górnym wyświetlaczu, zależy od wartości numerów instrukcji: bieżącej i następnej. Numer ten może być:
 - średnią arytmetyczną (z zaokrągleniem w dół) numerów instrukcji bieżącej i następnej,
 - najbliższym numerowi instrukcji bieżącej numerem podzielonym przez 10 (jeśli instrukcja bieżąca jest ostatnią w programie lub numer instrukcji następnej jest większy od numeru instrukcji bieżącej o więcej niż 10).
- Wyświetlenie pierwszej instrukcji programu - komenda POCZĄTEK - pozwala na szybkie przejście na początek tworzonego lub edytowanego programu. Jeśli program jest pusty, tzn. nie ma żadnej zaprogramowanej instrukcji, to w górnym wyświetlaczu wyświetlany jest numer 10. W przeciwnym przypadku wyświetlona jest instrukcja o najniższym numerze.
- Wyświetlanie ostatniej instrukcji programu - komenda KONIEC - pozwala na szybkie przejście na koniec tworzonego lub edytowanego programu. Jeżeli program jest pusty tzn. nie ma żadnej zaprogramowanej instrukcji, to na górnym wyświetlaczu wyświetlany jest numer 10. W przeciwnym przypadku wyświetlona jest instrukcja o najwyższym numerze.
- Zmiana parametrów numerycznych instrukcji - komenda ZMIANA - pozwala na zmianę parametrów numerycznych instrukcji bez konieczności ponownego programowania całej instrukcji. Parametr, którego wartość można w danej chwili zmieniać, "mruka". Po zmianie dowolnego parametru nowa wersja instrukcji wpisywana jest do programu w miejsce starej.
- Ustawienie numeru instrukcji - komenda NRINSTR - przejście do instrukcji o wybranym numerze. Po wpisaniu nowego numeru instrukcji

na górnym wyświetlaczu ukazuje się tekst instrukcji (o ile pod tym numerem jest zaprogramowana jakaś instrukcja) lub wyświetlany jest tylko numer instrukcji, gdy program nie zawiera instrukcji o tym numerze.

- Przenumerowanie instrukcji - komenda PRZENUM - umożliwia uporządkowanie numeracji programu polegające na zmianie numerów kolejnych instrukcji programu w taki sposób, aby program rozpoczynał się od instrukcji o numerze 10, a numer każdej kolejnej instrukcji zwiększał się o 10.

2.7. Start programu użytkowego robota.

W tym stanie pracy, podobnie jak przy edycji, nie wprowadzono zmian w stosunku do oprogramowania poprzedniej wersji robota IRp. Menu tego stanu obejmuje pięć pozycji. Zostaną one teraz pokrótce przedstawione.

- Start programu od pierwszej instrukcji - polecenie START - po wybraniu z menu opcji START rozpoczyna się automatyczne wykonywanie programu użytkowego, poczynając od pierwszej instrukcji (instrukcji o najmniejszym numerze). Następuje przejście w tryb automatyczny. Polecenie START zeruje stan wyjść i flag, kasuje znaczniki otwartych pętli i wywołanych podprogramów.
- Start programu od instrukcji aktualnie pokazanej na wyświetlaczu - polecenie STARTINS - po wybraniu opcji STARTINS rozpoczyna się automatyczne wykonywanie programu, poczynając od instrukcji aktualnie pokazanej na wyświetlaczu. Następuje przejście w tryb automatyczny.
- Wykonanie aktualnie wyświetlonej instrukcji - polecenie KROKOWO - wybranie opcji KROKOWO powoduje wykonanie jednej instrukcji, tej wyświetlonej aktualnie na wyświetlaczu. Po wykonaniu instrukcji na wyświetlaczu ukazuje się kolejna instrukcja programu.
- Zmiana prędkości dla pracy automatycznej - polecenie V_PROG - umożliwia zmianę prędkości wykonywania programu w pracy automatycznej przed wystartowaniem programu. Prędkość ruchu zaprogramowana w poszczególnych instrukcjach ruchu jest przed wykonaniem tych instrukcji mnożona przez wartość ustawioną poleceniem V_PROG. Jeśli wartość ta równa jest 100%, to program (ściślej instrukcje ruchu) będzie wykonywany z prędkościami zaprogramowanymi w instrukcjach. Prędkość można zwiększać lub zmniejszać z kwantem równym 5%, w zakresie od 5% do 400%.
- Praca z symulacją wejść lub wyjść obiektowych - polecenie SYMUL - umożliwia ustawienie w pracy automatycznej bądź krokowej symulacji sygnałów obiektowych wejściowych lub wyjściowych (bądź jednych i drugich). Opcja ta jest przydatna na etapie uruchamiania programu użytkowego.

2.8. Praca automatyczna.

Przejście w tryb pracy automatycznej możliwe jest poleceniem z panelu programowania (w stanie "START PROGRAMU") lub przez naciśnięcie lampki-przycisku STR na panelu operacyjnym. Podczas pracy automatycznej na górnym wyświetlaczu wyświetlony jest numer i typ aktualnie wykonywanej

instrukcji, a na dolnym - aktualna prędkość w ruchu automatycznym. Prędkość ta może być zmieniana podczas automatycznego wykonywania programu przy wykorzystaniu przycisków ZMV i ZWV na panelu programowania z kwantem 5%.

Program wykonywany jest cyklicznie, tzn. po wykonaniu instrukcji o najwyższym numerze (ostatniej), system przechodzi do wykonania instrukcji o numerze najmniejszym (pierwszej), o ile ostatnią instrukcją nie jest instrukcja skoku.

Wykonanie programu może być zatrzymane z panelu programowania lub z panelu operacyjnego, po naciśnięciu przycisku STP (stop). Na panelu następuje wtedy przejście do stanu "START PROGRAMU", a na jego wyświetlaczu pojawia się pełny tekst instrukcji ostatnio wykonywanej, o ile jej wykonywanie zostało przerwane, lub instrukcji następnej do wykonania, jeśli wykonanie instrukcji, przy której wciśnięto przycisk STP, zostało zakończone.

2.9. Obsługa wykrytych błędów pracy robota.

W trakcie pracy system sterowania cały czas nadzoruje prawidłowość działania całego robota. Wykrycie przez system sytuacji błędnej sygnalizowane jest przez wyświetlenie na wyświetlaczu numeru błędu z komentarzem (na panelu operacyjnym zaczyna mrugać lampka BLA - błąd). Numer błędu pozwala na odszukanie na liście błędów (dołączonej do podręcznika programowania robota) szerszego opisu zaistniałej sytuacji, z ewentualną sugestią przyczyny, która doprowadziła do powstania błędu. Wyjście ze stanu "BŁĄD" może nastąpić tylko w wyniku interwencji operatora, który poprzez naciśnięcie przycisku funkcyjnego pod napisem KASUJ potwierdza przyjęcie komunikatu o błędzie. Po skasowaniu błędu następuje powrót do stanu sprzed błędu. Skasowanie błędu wykrytego przy pracy automatycznej powoduje przejście panelu do stanu "START PROGRAMU".

3. WNIOSKI Z BADAN I PIERWSZYCH PROB EKSPLOATACYJNYCH. WERYFIKACJA OPROGRAMOWANIA PODSTAWOWEGO STEROWNIKA ROBOTA.

Omawiane, pełne oprogramowanie było wykorzystywane w badaniach robota od końca maja b.r. Dla badań wcześniejszych wykonano na początku roku specjalną wersję programu sterującego realizującego ograniczony zakres podstawowych funkcji robota. Oprogramowanie pełne było również zaimplementowane w robocie prezentowanym podczas tegorocznych targów w Poznaniu. Dodatkowym źródłem uwag były własne próby, przeprowadzane przez wykonawców programu sterującego w ramach uruchamiania i sprawdzania kolejnych funkcji. Wszystkie zebrane sygnały można podzielić na dwie grupy: zastrzeżeń do działania programu sterującego (w tym zakresie program był poprawiany w miarę możliwości na bieżąco) i uwag, wniosków, których intencją jest ukierunkowanie przyszłych prac. Dla porządku poniżej przedstawione są zebrane do tej pory zastrzeżenia (tylko usterki jeszcze nie usunięte) i uwagi, wraz z krótkim komentarzem.

ZASTRZEZENIA:

- W sytuacji zakłócenia pracy jednostki centralnej program sterujący zawiesza się i operator nie ma żadnej informacji o przyczynie tego stanu. Jest to bardzo trudny i jednocześnie bardzo ważny problem. Wyniknął on podczas badania odporności układu sterowania na zakłócenia elektromagnetyczne. W uzgodnieniu z konstruktorami sterownika dobudowano część programowej diagnostyki on-line jednostki centralnej. Modyfikacja polega na oprogramowaniu obsługi wszystkich przerw. Jak wspomniano w p. 2. program sterujący korzysta tylko z dwóch przerw. Dla wszystkich pozostałych, a więc w normalnych warunkach pracy nieprzewidzianych przerw, wprowadzono wspólny mechanizm postępowania. Do panelu programowania jest wysyłany komunikat z informacją o numerze przerwania, które wystąpiło, a wszystkie lampki panelu operacyjnego migają. Wyjście z tego stanu jest możliwe tylko na skutek świadomej interwencji operatora (naciśnięcie odpowiedniego przycisku). Następuje wtedy programowy restart jednostki centralnej. Opisany mechanizm został przetestowany na pamięci RAM. Aby sprawdzić go na pamięci EPROM konieczne jest powtórzenie badań zakłóceńowych.
- Przy czekaniu warunkowym robot nie reaguje na "stop" z panelu programowania, a tylko z panelu operacyjnego - procedury związane z realizacją instrukcji czekania rozbudowano o części generujące odpowiednie komunikaty testowe. Obecnie przystąpiono do prób. Zidentyfikowanie miejsca wystąpienia i poprawienie błędu powinno zająć nie więcej niż kilka dni.
- Po wczytaniu programu z pamięci masowej robot nie reaguje na przyciski ręcznego sterowania ruchem. Wykonanie jakiegokolwiek niewłaściwej operacji, które kończy się sygnalizacją przez system błędu powoduje przywrócenie prawidłowego funkcjonowania mechanizmu ręcznego poruszania robotem. Problem ten został zauważony dopiero podczas prac przygotowawczych do targów w Poznaniu, nie było więc czasu aby się nim dokładnie zająć. Dlatego też trudno jest obecnie diagnozować jego przyczynę.
- Przy ruchu ręcznym jedną osią oraz chwilowym naciśnięciu i puszczeniu przycisku innej osi robot zatrzymuje się - problem jest związany z zakłóceniem repetycyjnego przysyłania przez panel informacji o stale wciśniętym przycisku, w sytuacji gdy inny przycisk został zwolniony. Dla usunięcia tej niedogodności konieczne jest dokładniejsze

zaznajomienie się z oprogramowaniem panelu programowania, co już zostało przedsięwzięte. Gromadzenie wydruków testowych, które mogą ułatwić diagnozę sytuacji, jest utrudnione, gdyż odbieranie znaków z panelu odbywa się w trybie przerwanym. Wstawienie w tych miejscach drukowania komunikatów zakłóca czasowy przebieg zdarzeń, co w zasadniczo zmienia istotę całego mechanizmu ręcznego poruszania robotem. Rozwiązanie problemu wymaga po prostu czasu.

UWAGI:

- Po przełączeniu na "GOTOWOSC" robot nie powinien tracić synchronizacji - obecnie jeśli w stanie "GOTOWOSC" nastąpi ruch robota, synchronizacja jest tracona.
- Szata graficzna podręcznika programowania pozostawia wiele do życzenia.
- Przydatnym byłoby wprowadzenie możliwości zatrzymania programu użytkowego na końcu, np. jednym z przycisków na panelu operacyjnym.
- Przy rozsynchronizowaniu jednej osi, ponowna synchronizacja powinna dotyczyć wszystkich osi. Obecnie osie, które synchronizacji nie utraciły, nie są ponownie synchronizowane.
- Dobrze by było, aby program robota przesłany do komputera można było obejrzeć na ekranie monitora.
- Bardzo przydałaby się możliwość wydrukowania programu robota. Najlepiej, gdyby można było podłączyć drukarkę wprost do szafy sterowniczej. Można ewentualnie wydrukować program przesłany do komputera.
- Przydatna byłaby możliwość kodowania dostępu do programu zapisanego w pamięci masowej, tzn., aby osoba postronna nie mogła tego programu zmienić lub zniszczyć.
- Należy się zastanowić nad możliwością zmiany programowych zakresów ruchów osi robota przy pomocy panelu programowania, tzn. bez konieczności robienia nowej wersji programu i przepalania nowych pamięci EPROM (taką "technologię" zastosowano podczas przygotowywania wystawy w Poznaniu).

4. PROPOZYCJA DALSZYCH PRAC NAD PROGRAMEM STERUJĄCYM ROBOTA.

Przebieg dotychczasowych prac, uwagi po pierwszych próbach i badaniach robotów z nowym programem sterującym oraz oczekiwania aplikatorów i przyszłych użytkowników robotów pozwalają skonkretyzować kierunki dalszych działań w tym zakresie na najbliższy okres. Można w nich wyróżnić dwie zasadnicze grupy zagadnień. Pierwsza obejmuje testowanie, poprawianie, modernizację i rozwój przedstawionego programu sterującego. Zgodnie z przyjętymi na początku tych prac założeniami o dążeniu do unifikacji programów sterujących dla różnych typów robotów, oprogramowanie robota o udźwigu 6kg ma stanowić bazę dla wszelkich dalszych prac. Dlatego też prace rozwojowe powinny skupić się na tym typie robota. Druga grupa działań obejmuje właśnie wykonanie wersji programu sterującego dla kolejnych typów manipulatorów opracowanych w PIAP. Poniżej wymieniono konkretne prace, które zdaniem autorów opracowania powinny być wykonane w najbliższej przyszłości.

- Całkowite przejście na oprogramowanie narzędziowe firmy Intel - jest to krok niezbędny dla dalszych prac, co szerzej wyjaśniono w p.2.
- Usunięcie istniejących jeszcze zastrzeżeń do obecnej wersji programu sterującego.
- Rozbudowa programu sterującego o oscylacje, funkcje adaptacyjne i interpolację kołową.
- Rozbudowa mechanizmów programowej autodiagnostyki robota.
- Wykonanie wersji programu sterującego realizującego sterowanie typu PTP dla robota URP-60 oraz robota sześciooosiowego (z przeznaczeniem dla URP-3, URP-60Z, URP-120).
- Implementacja sterowania typu CP dla robota URP-60.
- Przeniesienie podręcznika programowania pod wybrany system przetwarzania tekstów lub wręcz składu komputerowego (DTP) - umożliwi to przy wykorzystaniu drukarki laserowej i pomocy BIUTECH-u wydawanie w jednostkowych nakładach, małym kosztem podręcznika, którego jakość nie będzie odbiegać od podobnych materiałów zagranicznych,
- Prace w kierunku stworzenia systemu komputerowego wspomaganie programowania robotów - początek tym pracom dała realizacja przesyłania programów między komputerem a układem sterowania robota. Kolejnym krokiem jest wprowadzenie możliwości edycji programów użytkowych na komputerze osobistym - w tym celu należy wykonać asembler i dezassembler programu użytkowego robota, pracujący na komputerze PC.

5. PODSUMOWANIE.

Wykonane do tej pory oprogramowanie stanowi zamknięcie pewnego etapu prac nad oprogramowaniem robotów pracujących z układem sterowania bazującym na nowej jednostce centralnej PIAP. Nie oznacza to oczywiście, że zostało zrobione wszystko, co było od tej strony możliwe, aby robot spełniał najróżniejsze wymagania aplikacyjne. Możliwości, jakie stwarza nowy układ sterowania, w zakresie rozbudowy funkcjonalnej robota na drodze software'owej, są bardzo duże. Wykonawcy programu sterującego są tego świadomi i widzą logiczne kierunki dalszych działań. Oczywiście nakreślony w poprzednim punkcie porządek dalszych prac nie należy traktować jako jedynej, wyznaczonej "na sztywno" drogi. Cechą prac nad oprogramowaniem być ich elastyczność, zdolność do zmiany zakresu czy kolejności wykonywanych zadań. Tak było przy pisaniu programu sterującego dotychczas. Zmiana koncepcji pamięci masowej, zresztą bardzo udana, pociągnęła za sobą konieczność wykonania bardzo dużej części oprogramowania od nowa. Dotyczy to nie tylko funkcji zapisu i odczytu programów użytkowych, lecz także organizacji programów w pamięci masowej (możliwość przechowywania dużej liczby programów na jednym nośniku). Wiązał się z tym także problem przenoszenia programów użytkowych pomiędzy robotami, który rozwiązano oprogramowując komunikację robota z komputerem PC. W efekcie otrzymano bardzo funkcjonalne i ze wszech miar nowoczesne rozwiązanie pamięci masowej robota. W zamian za to trzeba było zrezygnować z niektórych funkcji przewidzianych pierwotnie do realizacji. Proponuje się zaimplementować je w pierwszej kolejności, w ramach kontynuacji tematu.

Celem dotychczasowych prac było doprowadzenie oprogramowania podstawowego robotów do postaci zapewniającej im zdolność wdrożeniową. Zdaniem wykonawców cel ten osiągnięto. Program sterujący w obecnym stanie umożliwia realizację wielu aplikacji. W razie zainteresowania konkretnego klienta możliwa jest też szybka rozbudowa programu o nowe, dodatkowe funkcje, również te, które nie zostały ujęte w przedstawionych propozycjach rozwojowych.

L I T E R A T U R A

1. Z. Pilat [i in.]: Program sterujący robotów IRp-6/60 z układem sterowania bazującym na nowej jednostce centralnej PIAP. Etap 1. Opracowanie założeń funkcjonalnych robotów z uwzględnieniem zastosowań. Spraw. PIAP nr rej. 6613, 1991r.
2. Z. Pilat [i in.]: Program sterujący robotów IRp-6/60 z układem sterowania bazującym na nowej jednostce centralnej PIAP. Etap 2. Zapoznanie się z narzędziami i projekt oprogramowania (w tym model kinematyczny robotów IRp-6/60). Spraw. PIAP nr rej. 6637, 1991r.
3. W. Klimasara [i in.]: Wykonanie układów sterowania dla robotów 6kg i 2,5kg z magistralą AMS-BUS i cyfrowymi sterownikami mocy. Etap 1. Zmiany w układzie sterowania oraz badania pełne robota 6kg. Spraw. PIAP nr rej. 6849, 1992r.