

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

Ośrodek Automatyzacji Procesów Produkcji

Pracownia Robotyzacji Procesów Produkcji

074 Główny wykonawca mgr inż. A. Aderek

Wykonawcy mgr inż. Z. Pilat

Konsultant

Nr zlecenia RP-77.3

Robot z napędami elektrycznymi i sterowaniem MP i CP o udźwigu 120kg. Wykonanie programu sterującego.

Zad. 1.1. Opracowanie założeń.

Zleceniodawca

Pracę rozpoczęto dnia 87.03.10

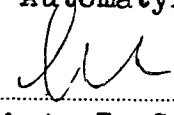
zakończono dnia 87.09.30

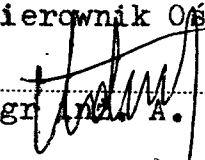
Kierownik Pracowni

Z-ca dyr.
d/s Automatyki

Kierownik Ośrodka


mgr inż. Z. Pilat


dr inż. T. Gałazka


mgr inż. A. Aderek

Praca zawiera:

Rozdzielnik - ilość egz:

stron 21

Egz. 1 BOINTE

rysunków 2

Egz. 2 OAP

fotografii

Egz. 3 OAR

tabel

Egz. 4 OAE

tablic

Egz. 5

załączników

Egz. 6

Nr rejestr. 5925

Analiza deskryptorowa

ROBOTY PRZEMYSŁOWE, UKŁAD STEROWANIA, OPROGRAMOWANIE

i 5 0 i 5 0 i 5 0 i 5 0

Analiza dokumentacyjna

Krótką charakterystyką układu sterowania robotów IRP.
Założenia na oprogramowanie robota IRP-120.

Tytuły poprzednich sprawozdań

338.45:62/63].002 1/.2 Roboty przemysłowe
684.3.06 Oprogramowanie

UKD

PIAP-252/63-6000

Spis treści.

	str.
1. Wstęp	4
2. Wymagania funkcjonalne układu sterowania robota RP-120 od strony oprogramowania.	4
2.1. Parametry ruchu.	4
2.2. Tryby pracy	5
2.3. Programowanie robota RP-120	8
2.3.1. Edycja programu	8
2.3.2. Typy instrukcji	9
2.4. Dodatkowe funkcje oprogramowania.	15
2.5. Obsługa błędów układu sterowania robota RP-120.	15
3. Układ sterowania robota RP-120.	16
4. Koncepcja realizacji oprogramowania robota RP-120.	18
5. Podsumowanie.	20.

1. Wstęp.

Tematem opracowania są założenia na oprogramowanie robota przemysłowego o udźwigu 120 kg /roboczy symbol RP-120/.

Jest to nowa konstrukcja. Robot ten ma wejść do produkcji po roku 1990. Jego parametry użytkowe powinny być więc nowoczesne na miarę lat 90-tych. Założenia na część mechaniczną przedstawiono w /1/, a na część sprzętową układu sterowania w /2/. Robot jest sześciooosiowy. Zastosowano w nim takie same silniki jak w modelu IRp-60: PZTK 13 07TRR o mocy 1KW, 3000 obr/min /1/ /we wszystkich osiach/. Do napędu osi wykorzystano dwa takie silniki. Umożliwiło to wykorzystanie układu sterowania takiego jak dla robotów IRp-60 tyle, że w wersji siedmioosiowej.

Robot IRp-120 będzie wykonywany w dwóch wersjach:

- A. z większymi przyspieszeniami osi a mniejszymi szybkościami maksymalnymi - dla zastosowań zgrzewalniczych
- B. z mniejszymi przyspieszeniami a większymi prędkościami maksymalnymi - dla zastosowań manipulacyjnych.

Zakłada się, że oprogramowanie powinno zapewnić funkcje jak dla robotów IRp-60, IRb-60/60Z.

2. Wymagania funkcjonalne układu sterowania robota RP-120 od strony oprogramowania.

2.1. Parametry ruchu.

Oprogramowanie zapewni uzyskiwanie założonej w /1/ powtarzalności pozycjonowania ± 1 mm. Maksymalne prędkości poszczególnych osi dla dwóch wersji robota będą wynosiły:

Wersja A

: $90^{\circ}/s$

: $34^{\circ}/s$

: $34^{\circ}/s$

: $90^{\circ}/s$

: $90^{\circ}/s$

: $90^{\circ}/s$

Wersja B

: $90^{\circ}/s$

: $55^{\circ}/s$

: $55^{\circ}/s$

: $90^{\circ}/s$

: $90^{\circ}/s$

: $90^{\circ}/s$

Zostanie zapewnione uzyskanie czasu 0,4s przy pozycjonowaniu pomiędzy dwoma sąsiednimi punktami odległymi o 40 mm /przy obciążeniu nominalnym/.

2.2. Tryby pracy

Oprogramowanie zapewni dwa podstawowe tryby pracy

- automatyczny
- ręczny

Tryb pracy automatycznej jest przeznaczony do wykonywania programu zapisanego w pamięci robota. Możliwe będzie rozpoczęcie wykonywania programu:

- od pierwszej instrukcji /zarówno z panelu programowania jak i z panelu operacyjnego/
- od wybranej instrukcji programu /tylko z panelu programowania/.

Zostanie wprowadzony mechanizm zmiany prędkości dla wszystkich instrukcji ruchu w programie. Prędkość ta będzie ustalana jako procent prędkości zadeklarowanej w trakcie uczenia robota w zakresie od 1 do 100%.

Wykonanie programu można przerwać na dwa sposoby:

- przyciskiem STOP PROGRAMU
- przyciskiem STOP AWARYJNY

Przyciski obu STOP-ów będą zdublowane na panelach: operacyjnym i programowania. W przypadku STOP PROGRAMU program zostaje zawieszony na aktualnie wykonywanej instrukcji. Możliwa jest jego kontynuacja przy pomocy funkcji startu od wybranej instrukcji lub wykonanie od początku przy pomocy funkcji startu od pierwszej instrukcji. Program może być również wykonywany dalej krokowo. W drugim przypadku /STOP AWARYJNY/ program zostaje przerwany równocześnie wyłączane są napędy wszystkich osi. Robot rozsynchronizowuje się. Przed ponownym uruchomieniem programu należy skasować stan STOP AWARYJNY /p. 2.4./ i ponownie zsynchronizować robota.

Szczególnym rodzajem pracy automatycznej jest synchronizacja robota. Czynność tę wykonuje się zawsze po włączeniu robota a także jak wspomniano po wystąpieniu stanu STOP AWARYJNY.

Do obsługi synchronizacji przeznaczony jest jeden z podświetlanych przycisków na panelu operacyjnym. Mruganie lampki oznacza brak synchronizacji. Po naciśnięciu jej uruchamiana jest procedura synchronizacji /zapisana na stałe w pamięci układu sterowania/.

Informuje o tym lampka świecąca się światłem ciągłym. Zakończenie synchronizacji jest sygnalizowane przez zgaszenie lampki.

Jeśli program został przerwany i wystąpił STOP AWARYJNY, to po wyjściu z tego stanu, a więc po synchronizacji programu może być kontynuowany lub wykonywany od początku. Podobnie po pierwszej synchronizacji po włączeniu programu może być wykonywany od początku lub od wybranej instrukcji.

Tryb pracy ręcznej umożliwia:

- krokowe wykonywanie programu /instrukcja po instrukcji/
- ręczne poruszanie robotem przy pomocy joysticka lub klawiszy
- programowanie robota /tworzenie programu użytkowego/
- współpracę z pamięcią kasetową
- realizację funkcji dodatkowych.

Robot będzie poruszany ręcznie we współrzędnych wewnętrznych /każda oś osobno/ lub we współrzędnych kartezjańskich. W drugim przypadku mamy do czynienia z ruchem wzdłuż osi ~~xxx~~ prostokątnego układu współrzędnych /x, y, z/.Przewiduje się wprowadzenie dwóch układów kartezjańskich:

- jeden związany z podstawą robota /bazowy/ - oś^Z pokrywa się z osią obrotu
- związany z narzędziem, którym manipuluje robot - ośZ pokrywa się z osią narzędzia, początek układu współrzędnych z punktem roboczym narzędzia /TCP/.

Prędkość ruchu robota przy sterowaniu klawiszami będzie ustalana przez podanie wartości z klawiatury numerycznej panelu programowania. Dla ruchu we współrzędnych wewnętrznych będzie możliwa zmiana położenia osi o pojedyncze inkreментy. Przy sterowaniu za pomocą joysticka prędkość ruchu będzie funkcją wychylenia drążka sterowniczego przy czym prędkość maksymalna po sterowaniu joystickiem lub klawiszami powinna być ograniczona do 25% prędkości wynikającej z konstrukcji robota.

Robot IRp-120 będzie współpracował z pamięcią kasetową PK-3. Możliwy będzie zapis programu na kasetę i załadowanie programu z kasety do pamięci układu sterowania /odczyt/. Przewiduje się również współpracę ze stacją dysków elastycznych.

W trybie pracy ręcznej możliwe są następujące funkcje dodatkowe:

- testowanie panelu programowania /kontrola segmentów wyświetlaczy
- odczyt aktualnej pozycji we współrzędnych wewnętrznych
- odczyt pozycji synchronizacji
- deklaracja narzędzia - określanie jego parametrów: współrzędnych x, y, z punktu TCP narzędzia /punkt roboczy narzędzia/ w kartezjańskim układzie współrzędnych związanym z kołnierzem robota oraz trzech kątów Eulera osi narzędzia w tym samym układzie współrzędnych
- deklaracja wektora korekcji - zaprogramowanie jego minimum i wartość /czyli prędkość/ /dotyczy funkcji adaptacyjnych/
- deklaracja palety - określenie jej parametrów: liczby elementów w wierszu i kolumnie palety oraz zaprogramowanie punktu początku obsługi palety i położenia trzech narożnych elementów obsługiwanej palety.

Zarówno narzędzie, paleta jak i wektor korekcji mają przyporządkowane numery / \emptyset do 63/.

2.3. Programowanie robota RP-120.

2.3.1. Edycja programu.

Do wprowadzania instrukcji programu do pamięci robota służy panel programowania. Daje on w trybie edycji następujące możliwości:

- zapisanie kolejnej instrukcji programu
- przeglądanie programu /po jednej instrukcji w przód lub w tył, skok na początek lub koniec programu, przejście do instrukcji o wybranym numerze/
- kasowanie instrukcji
- wstawianie nowych instrukcji między instrukcje wcześniej zaprogramowane
- zmiana parametrów numerycznych instrukcji
- kasowanie całego programu.

Każda instrukcja ma swój numer. Może to być dowolna liczba całkowita z przedziału 1 + 9999. Pierwsza instrukcja w nowym programie ma numer 10, następna 20, 30 itd.

Jeśli wstawiamy nową instrukcję między już istniejące, otrzymuje ona numer obliczony jako średnia arytmetyczna numerów instrukcji sąsiednich /zaokrąglona w górę/. Przykładowo instrukcja wstawiona między ~~10~~ instrukcje o numerach 10 i 20 będzie miała numer 15, między 14 a 20 - numer 18. Każdej instrukcji można nadać dowolny numer przez wpisanie go przy pomocy klawiatury numerycznej. Za numerem instrukcji znajduje się skrót /mnemonik/ określający jej typ, a następnie parametry instrukcji. Nie wszystkie instrukcje mają parametry, decyduje o tym typ instrukcji. Parametry mogą być numeryczne /np. numer ustawianego wyjścia/ lub nienumeryczne /np. rodzaj skoku warunkowy lub bezwarunkowy/.

2.3.2. Typy instrukcji.

Zostaną wprowadzone następujące typy instrukcji:

- deklaracja prędkości
- wybór narzędzia
- deklaracja masy
- + pozycjonowanie
- ustawianie wyjścia lub flagi
- skok
- czekanie
- chwytak
- paletyzacja
- powtarzanie wycinka programu
- koniec powtarzanego wycinka programu
- deklaracja początku podprogramu
- deklaracja końca podprogramu
- wywołanie podprogramu
- deklaracja końca programu.

Deklaracja prędkości.

Instrukcja ta służy do ustalenia maksymalnej prędkości ruchu osi robota i nominalnej prędkości liniowej punktu roboczego w ruchach z kontrolą trajektorii. Ma więc dwa parametry. Pierwszy z nich wyrażony w procentach jako część maksymalnej prędkości wynikającej z konstrukcji robota /tę wartość układ sterowania ma zapisaną w pamięci stałej/. Jego zakres jest od 1 do 100%. Drugi parametr jest podany w milimetrach na sekundę.

Wybór narzędzia.

Robot może operować różnymi narzędziami, o różnych parametrach geometrycznych zdefiniowanych przy deklaracji narzędzia /p.2.2./. Ta instrukcja wskazuje, które z zadeklarowanych narzędzi jest aktualnie używane /parametrem jest numer narzędzia/.

Deklaracja masy.

Funkcja ta pozwoli zmienić parametry regulacji w zależności od obciążenia robota. Potrzeba wprowadzenia tej instrukcji wiąże się z faktem, że bezwładność obciążenia ma duży udział w bezwładności całego manipulatora. Parametrem instrukcji będzie wielkość masy narzędzia. Przewiduje się uwzględnienie trzech przypadków:

- obciążenie małe
- obciążenie średnie
- obciążenie duże.

Zakresy mas dla poszczególnych przypadków zostaną określone przez autorów części mechanicznej robota.

Pozycjonowanie.

Instrukcja ta służy do przemieszczania manipulatora. Będzie ona miała pięć parametrów określających:

- rodzaj śledzenia trajektorii
- dokładność osiągnięcia punktu docelowego
- szybkość ruchu
- postać przechowywania pozycji
- reakcję na sygnały z czujników.

Robot będzie mógł się przemieszczać bez kontroli trajektorii /PTP/ na dwa sposoby:

- pozycjonowanie świadome /każda oś porusza się z maksymalną zadeklarowaną dla niej prędkością, ruch jest zakończony po osiągnięciu położenia docelowego przez wszystkie osie/
- pozycjonowanie quasiliniowe /odpowiednik instrukcji LINIOWO. W robotach IRb-6*60 /3/, prędkości dla poszczególnych osi dobrane są tak, aby ruchy poszczególnych stopni swobody kończyły się w tym samym czasie/.

Zostaną wprowadzone również dwa rodzaje pozycjonowania z kontrolą trajektorii /CP/:

- pozycjonowanie liniowe /ruch punktu TCP odbywa się po linii prostej w przestrzeni z kontrolą orientacji osi z kontrolą osi narzędzia/.

Ze względu na dokładność osiągnięcia punktu docelowego wyróżnia się:

- pozycjonowanie dokładne /następna instrukcja programu jest wykonywana po otrzymaniu potwierdzenia z układu pomiarowego osiągnięcia punktu docelowego/

- pozycjonowanie zgrubne /gdy następną instrukcją jest postój w miejscu przez pewien czas dokładność jest taka jak przy pozycjonowaniu dokładnym, jeśli nie - gorsza; układ sterowania przechodzi do wykonania kolejnej instrukcji programu nie czekając na potwierdzenie osiągnięcia zaprogramowanego punktu.

Szybkość ruchu jest ustalana różnie w zależności od rodzaju śledzenia trajektorii.

- dla pozycjonowania CP można podać prędkość liniową /jako procent zadeklarowanej wcześniej prędkości nominalnej/ lub czas przejścia trajektorii,
- dla ruchu guasiliniowego podaje się czas przemieszczenia robota,
- dla ruchu swobodnego podaje się maksymalną prędkość osi /jako procent zadeklarowanej wcześniej maksymalnej prędkości osi dla całego programu/.

Ze względu na postać przechowywania pozycji będzie możliwe

- pozycjonowanie bezwzględne /pozycja jest przechowywana jako przyrost impulsów rezolwera względem pozycji synchronizacji/.
- pozycjonowanie względne /pozycja jest przechowywana jako przyrost impulsów rezolwera względem ostatnio zaprogramowanej pozycji robota/.

Ostatni parametr mówi o sposobie reakcji na sygnały z czujników. Wyróżnia się:

- pozycjonowanie zwykłe /sygnały z czujników nie są uwzględniane/.
- pozycjonowanie adaptacyjne /zmiana ruchu w zależności od sygnałów z czujników/.

Przewiduje się możliwość zaprogramowania następujących ruchów adaptacyjnych:

- szukanie zwykłe /ruch jest przerwany po otrzymaniu sygnału z czujnika, wykonywany jest ruch korekcyjny do punktu zapamiętanego w momencie przyjęcia sygnału /uwzględnienie opóźnienia serwomechanizmów/ ustawiana jest flaga i jest wykonywana kolejna instrukcja programu/.
- szukanie z opóźnieniem /podobnie jak przy szukaniu zwykłym, z tym, że sygnał z czujnika musi trwać ok. 0.5 sek./.
- nadzór /po otrzymaniu sygnału z czujnika przerwanie ruchu, usta-

- wienie flagi i przejście do wykonywania następnej instrukcji/.
- korekcja prędkości /po otrzymaniu sygnału z czujnika, ruch jest przerwany, wykonywany jest ruch korekcyjny jak przy szukaniu zwykłym a następnie ruch jest kontynuowany ze zmniejszoną prędkością, jeśli sygnał z czujnika zniknie na więcej niż 0.5 sek. robot ponownie uzyskuje prędkość zaprogramowaną podczas uczenia/.
 - Konturowanie /do ruchu zaprogramowanego dodawany jest ruch korekcyjny w kierunku i z prędkością zgodną z zadeklarowanym wektorem korekcji związanym z aktywnym czujnikiem; przy konturowaniu jest możliwe uwzględnienie więcej niż jednego czujnika, gdy żaden z czujników nie podaje sygnału ruch odbywa się po trajektorii zaprogramowanej/.

Instrukcja pozycjonowania adaptacyjnego ma dodatkowy parametr w postaci numeru czujnika /numer wejścia, do którego czujnik jest dołączony/. W przypadku konturowania jest to jedna lub więcej par parametrów /numer czujnika- numer wektora korekcji/.

Ustawianie wyjścia lub flagi.

Instrukcja ta pozwala na włączanie i wyłączanie wyjść dwustanowych robota oraz flag /wskaźników/, które mogą być wykorzystane do sterowania wykonaniem programu /skoki warunkowe/.

Skok.

W robocie RP-120 będą dostępne skoki warunkowe i bezwarunkowe. Warunkiem może być stan jednego z wyjść dwustanowych lub jednej z flag. Drugim parametrem tej instrukcji jest adres skoku /numer instrukcji, do której ma być wykonany skok/.

Czekanie.

Można zaprogramować czekanie na upływ określonego odcinka czasu, na określony stan określonego wejścia lub flagi /np. czekanie na ustawienie flagi 10' w stan 0/. Czas czekania podaje się w sekundach /od 0.1 do 999.9 sek. co 0.1 sek./.

Chwytnak.

Instrukcja ta służy do operowania chwytnakami. Jej parametrami są numer chwytnaka /1 lub 2/ i jego stan pożądany /otwarty lub zamknięty/. Instrukcja ta służy do zaprogramowania obsługi palety.

Jej parametrami są: numer palety, prędkość dojścia do punktu początku obsługi palety, prędkość dojścia do elementu obsługiwanego i numer podprogramu obsługi elementu. Opcjonalnie są podawane: współrzędne pierwszego obsługiwanego punktu /domyślnie przyjmowany element /1,1/ i współrzędne ostatniego obsługiwanego punktu /domyślnie przyjmowany jest ostatni element palety/. Wykonanie instrukcji paletyzacji powoduje ruch prostoliniowy do punktu początku obsługi /p.2.2./, ruch prostoliniowy do elementu obsługiwanego, wykonanie wskazanego podprogramu obsługi, ruch do punktu początku obsługi palety. Jeśli nie jest to ostatni obsługiwany element cykl jest powtarzany. Po obsłużeniu ostatniego elementu program przechodzi do wykonania następnej instrukcji.

Powtarzanie wycinka programu.

Określa początek fragmentu, który ma być powtarzany, a jej parametr ustala wielokrotność powtarzania.

Koniec powtarzanego wycinka programu.

Ta instrukcja kończy fragment programu przeznaczony do powtarzania. Wraz z poprzednią są one dokładnym odpowiednikiem funkcji POWT., KONIEC POWT. w robotach IRb-6/60.

Instrukcje związane z podprogramami.

Mechanizm podprogramów jest rozwinięciem koncepcji funkcji WZÓR, MODYF w robotach IRb-6/60. W pełni je zatem zastępuje dając dodatkowe możliwości. Zarówno deklaracja początku jak i końca podprogramu nie mają parametrów.

Parametrem instrukcji wywołania podprogramu jest numer instrukcji deklaracji początku wywoływanego podprogramu.

Deklaracja końca programu.

Jest to instrukcja bezparametrowa. Po jej wykonaniu w pracy automatycznej układ sterowania przechodzi do pierwszej instrukcji programu. Układ sterowania "nie widzi" instrukcji o numerach wyższych od numeru instrukcji końca programu. Wstawienie tej instrukcji w środek istniejącego programu powoduje usunięcie wszystkich instrukcji o numerach wyższych.

Dla zgrzewalniczej wersji robota Rp-120 zostaną wprowadzone dodatkowe dwa typy instrukcji związane z procesem technologicznym:

- deklaracja parametrów zgrzewania,
- pozycjonowanie ze zgrzewaniem.

Pierwsza z nich umożliwi wybór numeru programu zgrzewania i wielkości strefy zerowej przy pozycjonowaniu w punkcie zgrzewania. Instrukcja ta będzie więc miała dwa parametry numeryczne.

Pozycjonowanie ze zgrzewaniem będzie instrukcją złożoną. W pierwszej fazie jej wykonania robot będzie realizował ruch do zaprogramowanego punktu zgrzewania tak jak przy pozycjonowaniu swobodnym. Prędkość maksymalną poszczególnych osi ogranicza tylko instrukcja deklaracji prędkości. Po wejściu na strefę zerową, której wielkość określa instrukcja deklaracji parametrów zgrzewania robot rozpoczyna hamowanie, wysyłając jednocześnie rozkaz uruchomienia odpowiedniego programu zgrzewania /do układu sterowania zgrzewarki - zakłada się, że robot będzie współpracował ze zgrzewarkami wyposażonymi w układy sterowania typu NUS produkcji ASPA Wrocław/. Po otrzymaniu potwierdzenia rozwarcia kleszczy zgrzewadła robot przechodzi do wykonania kolejnej instrukcji programu. W trakcie ~~wykonywania~~ wykonywania pozycjonowania ze zgrzewaniem są sprawdzane warunki kolizji robota i przygrzania szczęk /programowo-kontrola uchybu pozycji podczas ruchu/. Jest możliwa również kontrola zewnętrznego sygnału pochodzącego np. od mechanicznego sprzęgła przeciążeniowego. W razie wykrycia stanu awaryjnego /kolizja, przygrzanie, przeciążenie/ robot przechodzi w stan STOP SYSTEM. Stan ten objawia się zawieszeniem wykonywania programu, zatrzymaniem robota, ograniczeniem maksymalnego prądu silników do 50%. Po usunięciu przyczyny błędu i skasowaniu stanu STOP SYSTEM /2.5./ program może być kontynuowany lub wykonywany od początku.

2.4. Dodatkowe funkcje oprogramowania.

Oprogramowanie powinno uwzględniać możliwość pracy robota w większych zautomatyzowanych systemach produkcyjnych /gniazda, linie/. Musi więc zabezpieczyć możliwość połączenia układu sterowania robota RP-120 z wyższą warstwą sterowania. Warunkiem realizacji tego wymagania są zmiany w części sprzętowej sterownika robota /dodatknie pakietu kontrolera komunikacyjnego/. Kolejną funkcją, jaką powinno spełniać oprogramowanie jest współpraca z komputerem zewnętrznym przez łącze V-24. Kanałem takim dysponuje dziś większość komputerów osobistych biurowych i na takich maszynach będą z pewnością powstawały pierwsze systemy programowania off-line przy użyciu języków wysokiego poziomu.

Oprogramowanie musi też uwzględniać możliwość operowania sygnałami analogowymi.

W obu tych przypadkach przedstawiona koncepcja sprzętowa układu sterowania /2/ jest zbyt uboga. Oprogramowanie musi jednak sprawy te uwzględniać.

2.5. Obsługa błędów w pracy układu sterowania robota RP-120.

Wyświetlacz alfanumeryczny na panelu programowania umożliwia wprowadzenie znacznie rozbudowanego systemu informowania użytkownika o błędach w pracy układu sterowania. Wykonawcy proponują podobnie jak to rozwiązano w robotach IRp-6, IRp-60, wprowadzenie listy błędów. W konkretnej sytuacji w górnej linii wyświetlacza pojawi się numer błędu, a w dolnej krótki komentarz /24 znaki/. Do instrukcji obsługi robota lub do podręcznika programowania zostanie dołączona lista z obszernym opisem poszczególnych błędów. Jeden z klawiszy panelu programowania zostanie przewidziany do wprowadzania rozkazu kasowania komunikatu o błędzie i powrotu do normalnej pracy. Wykrycie błędu jest również sygnalizowane na panelu operacyjnym - mruganie lampki BLA /oznaczonej znakiem ?/. Po skasowaniu błędu lampka gaśnie.

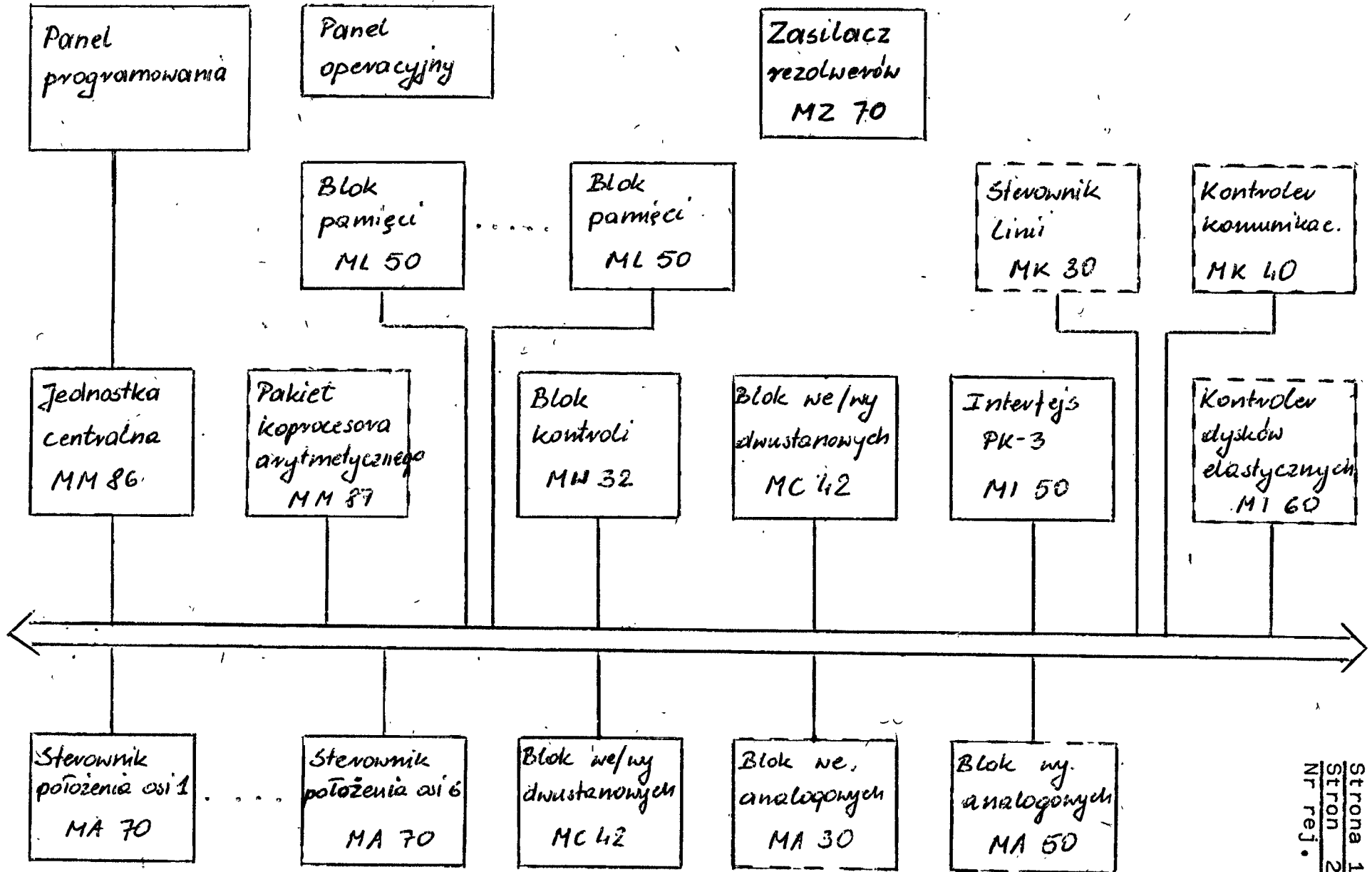
Osobno będą sygnalizowane błędy wprowadzające układ sterowania w stan STOP SYSTEM i STOP AWARYJNY. W pierwszym przypadku lampka BLA świeci światłem ciągłym. Wyjście z tego stanu odbywać się będzie przez naciśnięcie klawisza /wspomnianego wyżej/ na panelu programowania. W drugim przypadku wykorzystana zostanie lampka-przycisk na panelu operacyjnym. Świecenie się jej światłem ciągłym oznacza stan STOP AWARYJNY. Po usunięciu przyczyny błędu, za pomocą tego przycisku stan STOP AWARYJNY będzie kasowany. Lampka gaśnie. Oczywiście dla tych błędów również będzie wyświetlany komunikat na panelu programowania.

3. Układ sterowania robota RP-120.

Układ sterowania robota RP-120 jest systemem wieloprocesorowym /rys. 3.1./. Wykorzystano w nim moduły systemu INTEL DIGIT PROWAY. Są one umieszczone w 19-calowej kasecie i połączone magistralą wg standardu PROWAY. Jednostką centralną jest pakiet MM86 z mikroprocesorem INTEL 8086. Dla realizacji założonych funkcji oprogramowanie niezbędny jest również moduł koprocatora arytmetycznego MM87 /tak jak inne nieuwzględnione w /2/- oznaczony na rysunku linią przerywaną/. Bloki MA70 /sterowniki położenia osi/ posiadają również jednostki obliczeniowe -INTEL 8080. Funkcją każdego z nich w układzie sterowania robota jest przeliczanie otrzymanej /zadanej do przebycia/ wartości drogi dla jego osi na analogowy sygnał prędkości tej osi na najbliższy okres próbkowania.

Oprogramowanie bloków MA70 jest wykonywane w ramach tematu 77.4. /opracowanie /2/ jest jego pierwszą częścią/. Należy w nim uwzględnić zmiany charakterystyki regulatora w zależności od obciążenia manipulatora wg rozkazu głównego programu sterującego z MM86 /realizacja funkcji deklaracji masy 2.3.2./. W układzie sterowania musi się też znaleźć kontroler dysków elastycznych. W skład układu sterowania muszą też wejść i wyjść analogowych pakiet z dodatkowym kanałem transmisji V-24 oraz kontroler komunikacyjny. Stosowane do tej pory bloki pamięci ML50 powinna być zastąpione pakietami ML16 produkcji MERA-ZAP lub innymi /o większej pojemności pamięci/. Sprawy te powinny być uwzględnione w ramach tematu 77.4. na etapie projektu technicznego /ewentualnie jako uruchomienie /2//.

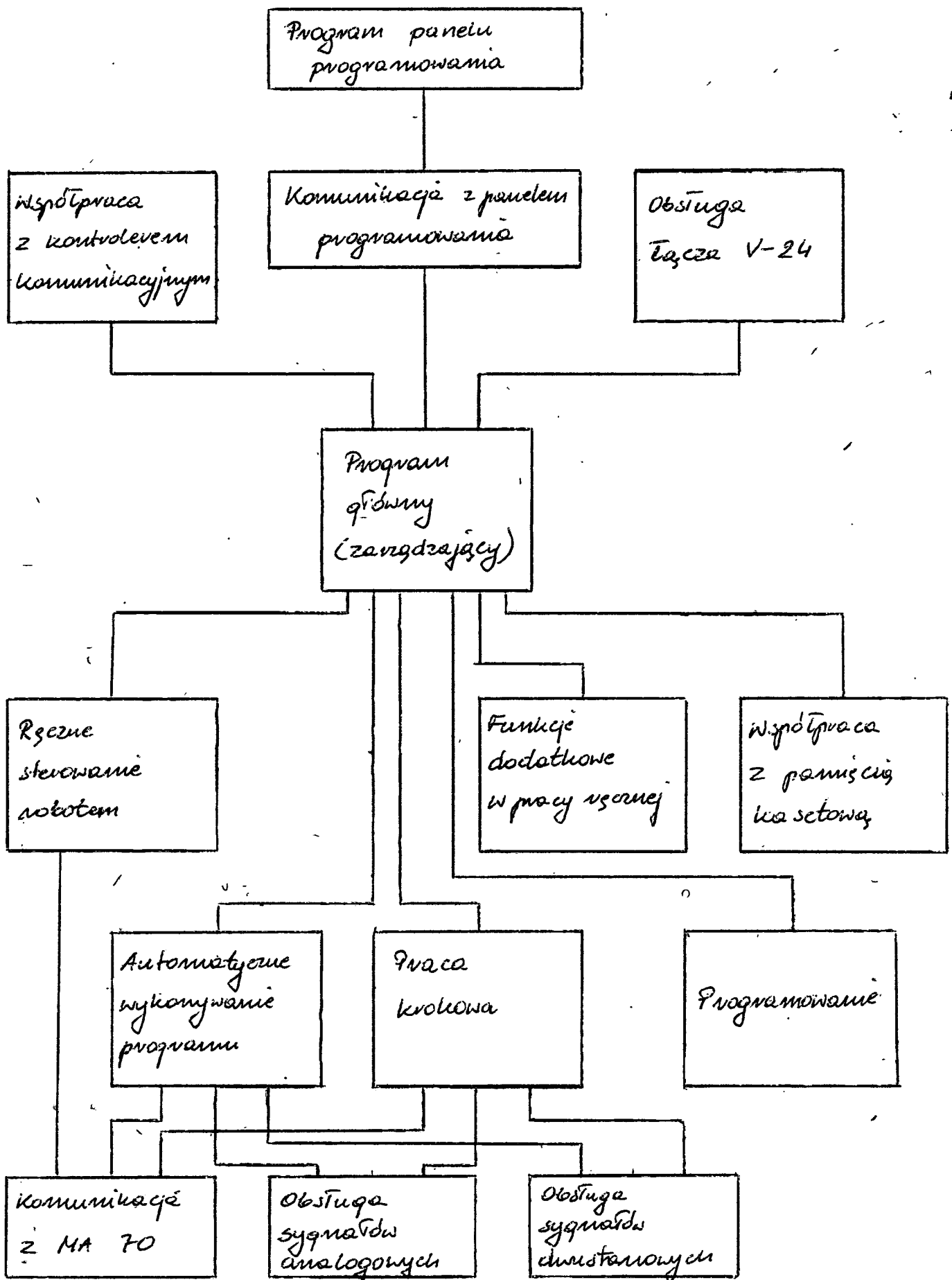
Rys. 3.1. Schemat blokowy układu sterowania robota Irp-120.



AK

4. Koncepcja realizacji oprogramowania robota RP-120.

Schemat blokowy oprogramowania przedstawia rys. 4.1. Jego podstawową częścią jest program zarządzający. Będzie on koordynował pracę całego systemu rozdzielając zadanie do poszczególnych bloków programu. Struktura programu zarządzającego musi zapewnić otwartość systemu rozumianą jako możliwość rozbudowy o nowe moduły bez zmiany oprogramowania istniejącego.



Rys. 4.1. Ogólny schemat blokowy oprogramowania robota RP-120

5. Podsumowanie.

Od strony funkcjonalnej robot RP-120 będzie bardzo zbliżony do robotów IRp 6/60/60Z. Dlatego też wykonawcy uważają, że najlepszą drogą będzie wykorzystanie istniejącego oprogramowania dla tych właśnie robotów. Praca będzie polegała na modyfikacjach i rozbudowie, co jest tańsze i szybsze od budowy oprogramowania od podstaw.

L I T E R A T U R A

1. A. Socha "Wymagania i założenia dla części mechanicznej robota o udźwigu do 120 kg".
Sprawozd. PIAP nr rej. 5764
2. G. Heszen "Robot z napędem elektrycznym i sterowaniem MI i CP o udźwigu 120 kg".
Opracowanie układu sterowania robota IRp-120.
Etap 1.1. Opracowanie założeń"
Sprawozd. PIAP nr rej. 5673
3. DTR robotów IRb-6, IRb-60, IRb-60Z.
4. DTR robotów IRp-6, IRp-60.
5. Podręcznik programowania robotów IRp.
6. A. Aderek "Układ sterowania dla robota zgrzewalniczego IRp-60Z".
Założenia techniczne". Sprawozd. PIAP nr rej.
7. Z. Pilat "Układ sterowania dla robota zgrzewalniczego IRp-60Z.
Wstępna wersja oprogramowania".
Sprawozd. PIAP nr rej. 5896.
8. A. Aderek "Opracowanie układu bazowego rodziny zunifikowanych układów sterowania dla różnych robotów krajowych, opracowanie układu sterowania dla obecnie istniejących części manipulacyjnych robotów IRb-6 i IRb-60 w zakresie programu sterującego.
Etap 1 Opracowanie założeń na program sterujący".
Sprawozd. PIAP nr rej. 5214.