

6556

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

Wykonawcy

mgr inż. Zbigniew Stańczak

Stanek

mgr inż. Wanda Ulatowska

Woj.

Konsultant

Nr zlecenia RP-27.3

Cel 27 p.t. "Rozwój modułowych robotów przemysłowych PR-02E"

Temat RP-27.3 p.t. "Opracowanie oprogramowania dla modułowych robotów PR-02E"

Zadanie 6.3 p.t. "Weryfikacja oprogramowania w wersji podstawowej oraz opracowanie dokumentacji oprogramowania wykonanego w zadaniu 6.1"

Zleceńodawca

Pracę rozpoczęto dnia 90.10.01

zakończono dnia 90.10.51

Wykonawca zlecenia

Kierownik Ośrodka

mgr inż. Z. Stańczak

Z-ca Dyrektora d/s  
Automatyki i Pomiarów

dr inż. H. Wrzesień

doc. dr inż. T. Gałęzka

Praca zawiera:

Rozdzielnik - ilość egz:

stron 22

Egz. 1 ROZDZ.

rysunków 4

Egz. 2 OAR

fotografii

Egz. 3 OAE

tabel

Egz. 4 OAP

tablic

Egz. 5 OAP-8

załączników

Egz. 6

Nr rejestr. 6556

4

Analiza deskryptorowa

PRZEMYSŁOWE  
ROBOTY, OPROGRAMOWANIE

Analiza dokumentacyjna

**Sprawozdanie zawiera opis oprogramowania robotów PR-02E w wersji podstawowej.**

Tytuły poprzednich sprawozdań

1. RP-27.3 zad. 2.1 Opracowanie założeń na oprogramowanie.  
Nr rej. 5930.
2. RP-27.3 zad. 3.2 Adaptacja oprogramowania układu sterowania robota IRB-60 dla zbadania modeli.  
Spr. PIAP styczeń 1989.
3. RP-27.3 zad. 6.1 Oprogramowanie umożliwiające sterowanie typu PTP 3 modułów z napędem elektrycznym oraz sterowanie sekwencyjne 10 modułów pneumatycznych i układu 12 wejść/Wyjść.  
Nr rej. 6496.

328 uł 621637.002 1/2 Roboty przemysłowe  
6813 06 Oprogramowanie

UKD

MERA-PIAP/TW 331/78 5000

21

## S P I S T R E Ś C I

1. WPROWADZENIE.
2. BUDOWA PROGRAMU SYMULUJĄCEGO PANEL PROGRAMOWANIA.
3. OPIS PROGRAMU SYMULUJĄCEGO PANEL PROGRAMOWANIA.
  - 3.1. Tworzenie programu użytkowego.
  - 3.2. Sterowanie wykonaniem programu użytkowego.
4. OPIS DZIAŁANIA PROGRAMU SYMULACYJNEGO.
  - 4.1. Synchronizacja.
  - 4.2. Praca automatyczna.
  - 4.3. Poruszanie manipulatorem.
  - 4.4. Programowanie instrukcji.
  - 4.5. Edycja programu użytkowego.
  - 4.6. Ręczne operowanie systemem.
5. OGÓLNA CHARAKTERYSTYKA PROGRAMU STERUJĄCEGO.
  - 5.1. Funkcja 'main()'.
    - 5.3.1. Procedura 'axes\_controller'.
    - 5.3.2. Procedura 'progpanel'.
    - 5.3.3. Procedura 'oppanel'.
  - 5.2. Funkcja 'init\_master()'.
    - 5.3.1. Procedura 'axes\_controller'.
    - 5.3.2. Procedura 'progpanel'.
    - 5.3.3. Procedura 'oppanel'.
  - 5.3. Funkcja 'system()'.
    - 5.3.1. Procedura 'axes\_controller'.
    - 5.3.2. Procedura 'progpanel'.
    - 5.3.3. Procedura 'oppanel'.
  - 5.4. Funkcja 'manipulacja()'.
6. PODSTAWOWE CZYNNOSCI PROGRAMU STERUJĄCEGO.
  - 6.1. Synchronizacja manipulatora robota.
  - 6.2. Ruch quasiliniowy.
  - 6.3. Ręczne operowanie manipulatorem.
  - 6.4. Obsługa panelu operacyjnego.
  - 6.5. Współpraca z panelem programowania.
  - 6.6. Wykonywanie programu użytkowego.
  - 6.7. Współpraca z pamięcią kasetową.
  - 6.8. Obsługa błędów.
7. WNIOSKI KOŃCOWE.
8. LITERATURA.

## 1. WPROWADZENIE.

=====

Niniejsze opracowanie zawiera opis oprogramowania w wersji podstawowej dla sterowania manipulatorem robota PR-02E.

Prace nad opracowaniem programu sterującego robota PR-02E były podzielone na dwa etapy, z których pierwszy dotyczył tworzenia oprogramowania dla badan prototypu manipulatora robota (zad. 8.1), zaś drugi - programu sterującego w wersji podstawowej.

Program sterujący robota PR-02E poprawnie współpracuje z programem symulacyjnym panelu programowania na komputerze IBM w wersji dla tego robota. Program symulacyjny zastępuje rzeczywisty panel programowania i realizuje wszystkie funkcje dotychczasowego panelu programowania. W niniejszym sprawozdaniu zamieszczono opis jego działania oraz sposób wykorzystania go podczas pracy manipulatora, a ponadto opisano zasady ręcznego sterowania manipulatorem robota, w przypadku zastosowania programu symulującego panel programowania robota na komputerze IBM.

Zasady tworzenia, poprawiania i uruchamiania programów użytkowych, ogólne zasady dotyczące działania programu użytkowego, język programowania robota z punktu widzenia użytkownika oraz sposoby zgłaszania przez program sterownika sytuacji błędnych są zawarte w pracy [1].

Program sterujący robota PR-02E w wersji podstawowej jest oparty na standardowym programie sterującym robotów IRp, stosowanym dotychczas. Uwzględniono inną niż w przypadku robotów IRp-6/60 mechanikę manipulatora, dostosowując do niej oprogramowanie układu sterowania poprzez wprowadzenie nowych parametrów matematycznych. Ponadto przystosowano program sterujący do współpracy z manipulatorem o trzech stopniach swobody, jakim jest manipulator robota PR-02E. Struktura programu pozostała w zasadzie bez zmian. Obieg głównej pętli programu jest analogiczny do programu standardowego, również podobne są zasady komunikacji pomiędzy układem sterowania a programem panelu programowania na komputerze IBM oraz sygnalizacja błędów.

Konfiguracja sprzętowa układu sterowania robota PR-02E różni się od standardowej, zmniejszoną o dwa liczbą sterowników osi MA-70 (manipulator robota PR-02E posiada trzy stopnie swobody) oraz liczbą pakietów MC-42. Układ sterowania robota PR 02E zawiera 4 takie pakiety, trzy z nich to pakiety we/wy użytkownika, czwarty - to pakiet we/wy systemowych.

## 2. BUDOWA PROGRAMU SYMULUJĄCEGO PANEL PROGRAMOWANIA.

=====

Program źródłowy symulatora składa się z następujących zbiorów:

- ppsym.c - moduł główny, zajmujący się ponadto obsługą błędów.
- paut.c - moduł zajmujący się obsługą wykonywania programu użytkowego,
- ins.c - moduł zajmujący się obsługą tworzenia programu użytkowego,
- edy.c - moduł zajmujący się edycją programu użytkowego (przeoglądanie, modyfikacje).
- syn.c - moduł zajmujący się synchronizacją robota,
- ope.c - moduł zajmujący się przepisywaniem programu użytkowego z RAM do PK oraz z PK do RAM,
- przes.c - moduł obsługujący odbiór/wysyłanie przesyłek z/do panelu operacyjnego,
- prz.c - moduł umożliwiający ręczne sterowanie manipulatorem robota,
- pc.c - moduł współpracujący z modułem 'p/przes.c' - driver przerwan od interfejsu szeregowego.

Powyzsze moduly programowe zostaly skompilowane i zlinkowane pod systemem TURBO-C. program wykonawczy umieszczony zostal w zbiorze 'panel.exe'.

### 3. OPIS PROGRAMU SYMULUJĄCEGO PANEL PROGRAMOWANIA.

=====

Symulator panelu programowania umożliwia:

- 1). tworzenie i edycję programu użytkowego robota,
- 2). sterowanie wykonaniem programu użytkowego,
- 3). współpracę z panelem operacyjnym (a więc z programem sterującym) w zakresie :
  - ręcznego sterowania robotem we współrzędnych wewnętrznych,
  - synchronizacji robota,
  - przepisywania programu użytkowego z PK do RAM oraz z RAM do PK.

#### 3.1. Tworzenie programu użytkowego.

-----

Tworzenie programu użytkowego robota obejmuje programowanie następujących instrukcji:

- 1). instrukcja wyboru narzędzia,
- 2). instrukcja pozycjonowania typu PTP z programowalną prędkością ruchu i sposobem osiągnięcia położenia zadanego (zgrubne/dokładne),
- 3). instrukcja skoku bezwarunkowego,
- 4). instrukcja czekania bezwarunkowego,
- 5). instrukcja początku pętli programowej,
- 6). instrukcja końca pętli programowej,
- 7). instrukcja ustawienia flagi lub wyjścia,
- 8). instrukcje operacji na chwytakach.

#### 3.2. Sterowanie wykonaniem programu użytkowego.

-----

Sterowanie wykonaniem programu użytkowego obejmuje:

- 1). start od pierwszej instrukcji programu,
- 2). start od bieżącej instrukcji programu,
- 3). pracę krokową,
- 4). zatrzymanie wykonywania programu.

#### 4. OPIS DZIAŁANIA PROGRAMU SYMULACYJNEGO.

=====

Program symulacyjny panelu programowania należy uruchamiać przy włączonym układzie sterowania robota oraz dokonanym połączeniu interfejsu szeregowego pakietu MM-16 w szafie sterowniczej z komputerem. Jest on wywoływany dyrektywą 'panel'.

Wywołanie programu, przy prawidłowej pracy sprzętu robota i prawidłowym połączeniu go z komputerem, powoduje ukazanie się napisu, zawierającego numer wersji programu i typ robota. Po krótkiej chwili ukazuje się główne menu programu. Jeśli wystąpi błąd w pracy sprzętu lub połączenie szafy sterowniczej z komputerem jest nieprawidłowe, program informuje o tym poprzez wyświetlenie komunikatu: "brak komunikacji z układem sterowania". Po zgłoszeniu się programu można załączyć zasilanie napędów robota.

Menu główne programu wygląda następująco :

- 1 Synchronizacja
  - 2 Praca automatyczna
  - 3 Poruszanie manipulatorem
  - 4 Programowanie instrukcji
  - 5 Edycja
  - 6 Ręczne operowanie systemem
- Esc Koniec pracy panelu

Wyboru stanu pracy użytkownik dokonuje poprzez naciśnięcie odpowiedniego klawisza cyfry bądź klawisza 'Esc'.

Wybranie pozycji 1 powoduje rozpoczęcie procesu synchronizowania manipulatora robota, pozycji od 2 do 6 spowoduje przejście do wybranego stanu pracy, natomiast naciśnięcie klawisza 'Esc' (koniec pracy panelu) spowoduje zakończenie działania programu oraz przejście do systemu operacyjnego.

##### 4.1. Synchronizacja.

-----

Kożpoczęcie synchronizowania robota jest możliwe tylko przy załączonych napędach. Synchronizacja robota w przypadku, gdy robot nie był wcześniej zsynchronizowany, rozpoczyna się po naciśnięciu klawisza z cyfrą '1' w menu głównym. Zatrzymania przebiegu tego procesu można dokonać poprzez naciśnięcie klawisza z literą 's' lub klawisza 'spacja' (długi klawisz w dolnej części klawiatury). Naciśnięcie klawisza 'Esc' powoduje powrót do menu głównego.

W przypadku wystąpienia błędów w trakcie trwania synchronizacji podany zostanie odpowiedni komunikat. Jeśli natomiast synchronizacja zakończy się prawidłowo pojawi się napis :

"Koniec synchronizacji."

#### 4.2. Praca automatyczna.

-----

Pracę automatyczną można rozpocząć w przypadku, gdy :

- robot jest zsynchronizowany,
- w pamięci RAM systemu znajduje się program użytkowy.

Program użytkowy może zostać wprowadzony do pamięci za pomocą klawiatury lub przy wykorzystaniu pamięci kasetowej PK (w G stanie pracy). Podczas pracy robota, na monitorze w okienku edycyjnym, wyświetlana jest aktualnie wykonywana instrukcja.

Menu stanu 'praca automatyczna' wygląda następująco :

- 1 Start od pierwszej instrukcji programu
- 2 Start od aktualnie pokazanej instrukcji
- 3 Wykonanie aktualnie pokazanej instrukcji
- + Zwiększenie szybkości wykonywanego programu
- Zmniejszenie szybkości wykonywanego programu
- pace Stop programu
- Esc Koniec pracy panelu

Wcisnięcie odpowiednio :

- cyfry '1' -- powoduje wystartowanie programu użytkowego (jeżeli taki istnieje) od jego pierwszej instrukcji,
- cyfry '2' - powoduje wystartowanie programu od instrukcji, która jest aktualnie wyświetlona w okienku edycyjnym,
- cyfry '3' - wykonanie instrukcji wyświetlonej w okienku edycyjnym (praca krokowa),
- litery 's' - powoduje zatrzymanie wykonywania programu,
- klawisza 'Esc' - powoduje powrót do menu głównego (klawisz jest nieaktywny. podczas pracy manipulatora).

#### 4.3. Poruszanie manipulatorem.

-----

Menu stanu "poruszanie manipulatorem" wygląda następująco:

- 1 Ruch w pojedynczych inkrementach
- 2 Układ współrzędnych wewnętrznych
- Esc powrót do menu głównego

Nacisnięcie klawisza '1' lub '2' powoduje wejście do wewnętrznego, identycznego w obu przypadkach menu, opisującego klawisze klawiatury, służące do poruszania manipulatorem. Pojawia się jeden z dwóch możliwych napisów :

RUCH W POJEDYNCZYCH INKREMENTACH

lub

UKŁAD WSPÓLRZĘDNYCH WEWNĘTRZNYCH



a następnie :

Poruszanie manipulatorem - klawiatura funkcyjna.

F1	OS 1 -
F2	OS 1 +
F3	OS 2 -
F4	OS 2 +
F5	OS 3 -
F6	OS 3 +
Tab	Wł./wył. ruchu ciągłego
Esc	Powrót do głównego menu

Znak '+' oznacza ruch manipulatora w kierunku zgodnym z kierunkiem ruchu w trakcie trwania procesu synchronizacji, znak '-' natomiast - oznacza ruch w kierunku przeciwnym do kierunku przebiegu procesu synchronizacji. Nacisnięcie klawisza z cyfrą '1' umożliwia poruszanie manipulatorem robota pojedynczymi inkrementami. W praktyce oznacza to możliwość wolnego, dokładnego dojścia do zadanego punktu przestrzeni roboczej. Wielkość inkrementów ruchu została zainstalowana programowo. Nacisnięcie klawisza z cyfrą '2' umożliwia poruszanie manipulatorem robota w układzie współrzędnych wewnętrznych, oznacza to ruch, w którym jednostka ruchu jest wielokrotnością jednego inkrementu. Wielkość tę można dobrać w zależności od potrzeb użytkownika.

#### 4.4. Menu programowania instrukcji.

-----

Menu programowania instrukcji programu użytkowego jest pokazane poniżej:

- 1 - pozycjonowanie
- 2 - skok bezwarunkowy
- 3 - oczekiwanie bezwarunkowe
- 4 - początek pętli programowej
- 5 - koniec pętli programowej
- 6 - ustawienie flagi lub wyjścia
- Esc - powrót do menu głównego

Wcisnięcie odpowiednio :

- cyfry '1' - umożliwia zaprogramowanie instrukcji pozycjonowania. Dokonuje się określenia wartości prędkości (% prędkości podstawowej) i sposobu dojścia do zadanego położenia,
- cyfry '2' - umożliwia określenie numeru instrukcji programu, do której ma nastąpić skok,
- cyfry '3' - określenie czasu czekania (w sek.).
- cyfry '4' - umożliwia otwarcie pętli programowej,
- cyfry '5' - umożliwia zamknięcie pętli,
- cyfry '6' - umożliwia ustawienie flagi lub wyjścia na wartość stałą (0/1), wejście lub flagę,
- klawisza 'Esc' - umożliwia powrót do menu głównego.

#### 4.5. Menu edycji.

-----  
Wejście w stan edycji programu, powoduje wypróbowanie na ekran monitora :

- napisu " EDYCJA PROGRAMU ",
- informacji o bieżącej instrukcji programu w postaci :  
numer instrukcji, kod instrukcji, [parametry instrukcji],
- menu stanu edycji programu.

Menu stanu edycji programu jest następujące :

- 1 - instrukcja w przód
- 2 - instrukcja w tył
- 3 - usun instrukcję
- 4 - wstaw instrukcję
- 5 - początek programu
- 6 - koniec programu
- 7 - zmiana instrukcji
- 8 - ustawianie numeru instrukcji
- Esc - powrot do menu głównego.

Wcisnięcie odpowiednio :

- cyfry '1' - powoduje pokazanie następnej, względem bieżącej, instrukcji. Jeżeli następna instrukcja nie istnieje pojawi się komunikat :  
"nie ma następnej instrukcji - koniec programu",
- cyfry '2' - powoduje pokazanie poprzedniej, względem bieżącej, instrukcji programu. Jeżeli instrukcja ta nie istnieje, pojawi się komunikat :  
"nie ma poprzedniej instrukcji - początek programu",
- cyfry '3' - powoduje usunięcie grupy instrukcji od bieżącej instrukcji programu do instrukcji o podanym numerze,
- cyfry '4' - powoduje wstawienie numeru instrukcji pomiędzy numerem instrukcji bieżącej i następnej. Nowy numer instrukcji jest wyświetlony na ekranie. Jeśli w okienku edycyjnym była wyświetlona ostatnia instrukcja programu, to wcisnięcie cyfry '4' umożliwia zaprogramowanie kolejnej instrukcji,
- cyfry '5' - powoduje skok do początku programu,
- cyfry '6' - powoduje skok do końca programu,
- cyfry '7' - pozwala na zmianę parametrów numerycznych bieżącej instrukcji,
- cyfry '8' - powoduje podanie zawartości instrukcji o podanym numerze,
- klawisza 'Esc' - umożliwia powrot do menu głównego.

#### 4.6. Menu ręcznego operowania systemem.

---

Menu ręcznego operowania systemem jest następujące:

- 1 - przepisanie programu z PK do RAM
- 2 - przepisanie programu z RAM do PK
- s - zatrzymanie przepisywania,
- Esc - powrót do menu głównego.

Wcisnięcie odpowiednio :

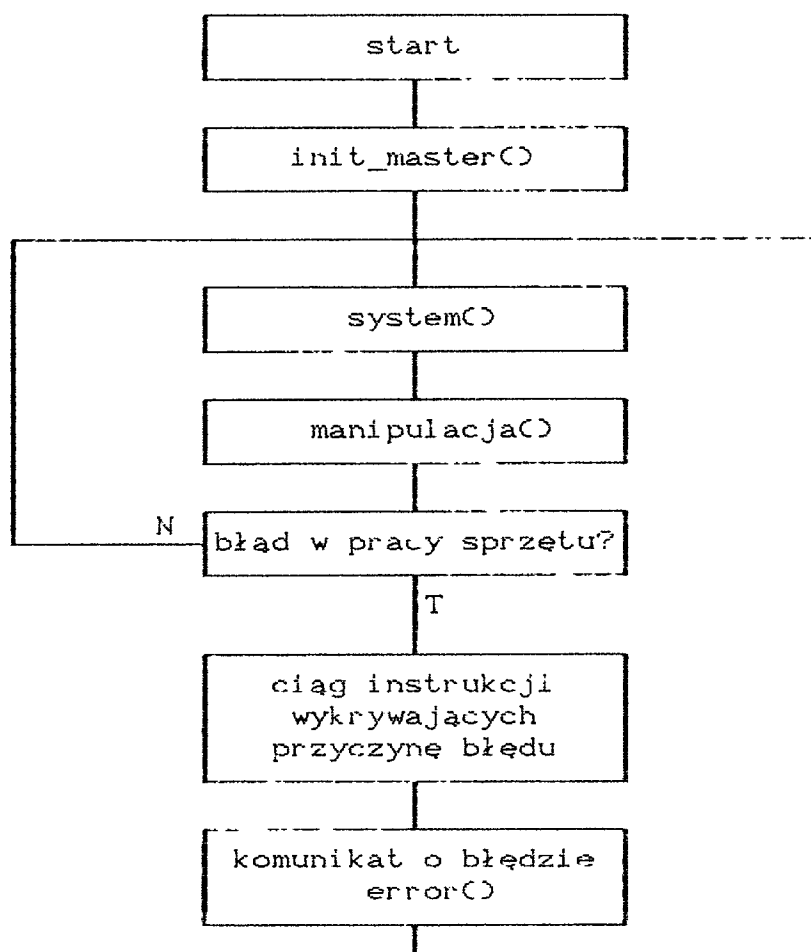
- cyfry '1' - powoduje rozpoczęcie przepisywania programu użytkowego z pamięci kasetowej do pamięci RAM systemu sterowania,
- cyfry '2' - powoduje rozpoczęcie przepisywania programu użytkowego z pamięci RAM do pamięci kasetowej,
- litery 's' - powoduje zatrzymanie przepisywania programu,
- klawisza 'Esc' - umożliwia powrót do menu głównego.

## 5. OGÓLNA CHARAKTERYSTYKA PROGRAMU STERUJĄCEGO.

Standardowy program sterujący, jego budowa i działanie zostały dokładnie opisane w literaturze [2]. W niniejszym rozdziale zajmujemy się wyjaśnieniem różnic, jakie w stosunku do tego programu, wykazuje program opracowany dla robota PR-02E.

Poniżej zamieszczono schematy blokowe pętli głównej oraz podstawowych funkcji programu oraz krótki ich opis.

### 5.1. Funkcja 'main()'. -----



Główną procedurą programu sterującego robota PR-02E jest funkcja 'main'. Funkcja ta, wywoływana jako 'main()', inicjalizuje system (program sterujący i sprzęt). Inicjalizacja wykonywana jest sekwencyjnie, w efekcie wywoływania z funkcji 'main' następujących funkcji :

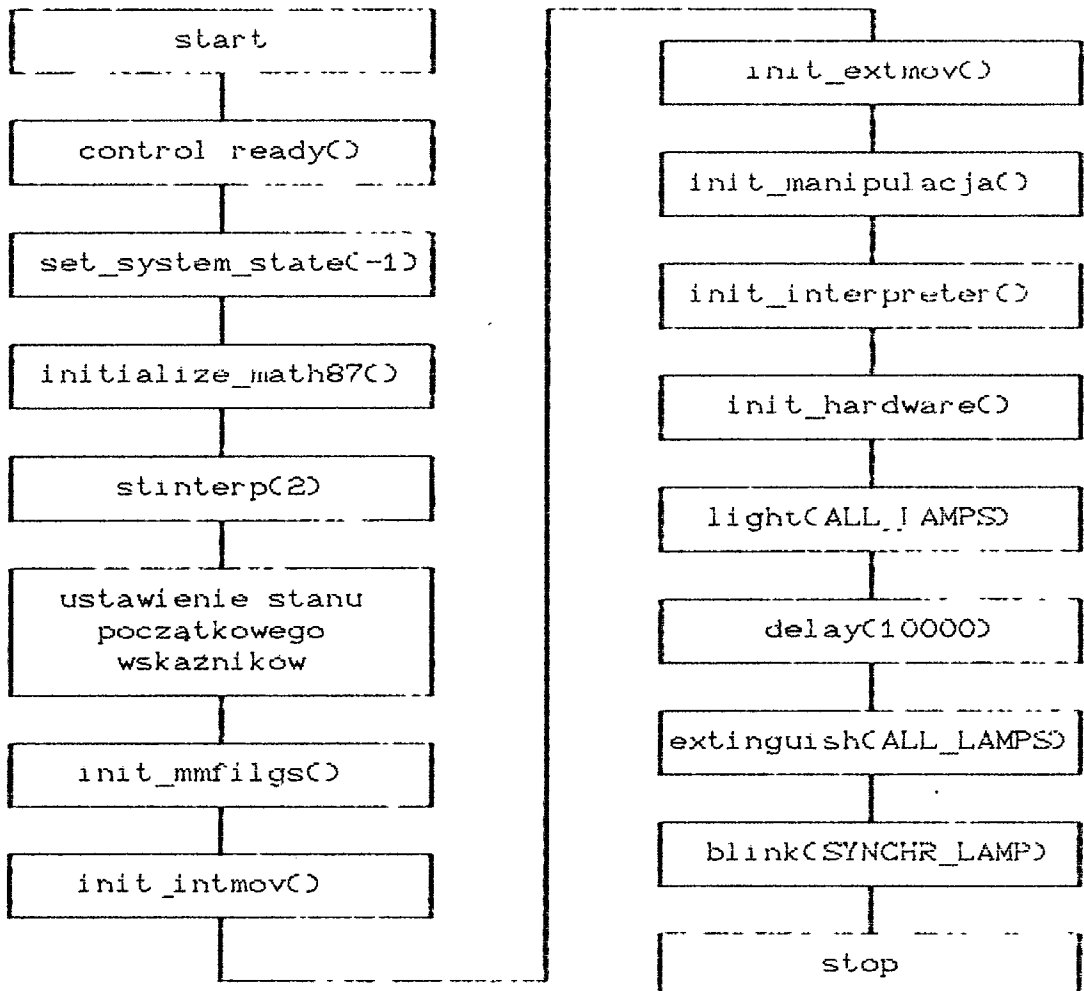
- init\_master() - inicjalizacja programu głównego,
- system() - obsługa sterowników osi, panelu programowania i panelu operacyjnego,
- manipulacja() - obsługa ręcznego sterowania manipulatorem.

Po inicjalizacji działania systemu następuje testowanie, czy nie wystąpił błąd związany z pracą sprzętu :

- brak gotowosci osi,
- stop awaryjny,
- zbyt duzy błąd położenia,
- wyjście ze stanu 'praca',
- za wysoka temperatura w sterowniku,
- utrata synchronizacji,
- przeciążenie,
- inne błędy.

Wystąpienie któregokolwiek z wymienionych błędów powoduje wywołanie funkcji 'error(n)', (gdzie n - numer błędu), która ustawia system w stan ERROR\_STATE, załącza miganie lampki BŁĄD oraz przekazuje kod błędu do panelu programowania.

## 5.2. Funkcja 'init\_master()'. -----



Przedstawiona wyżej funkcja wywoływana jest jako 'init\_master()'. Jej rola polega na zainicjowaniu pracy całego systemu robota.

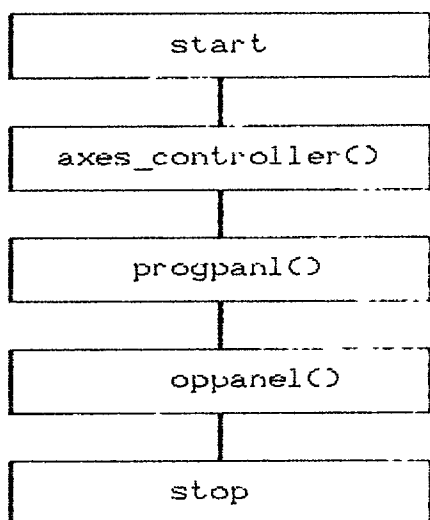
Wykonywane przez tę funkcję kolejne kroki są następujące :

- 1). załączenie sygnału control\_ready(),
- 2). ustawienie systemu w stan pracy ręcznej,
- 3). inicjalizacja koprocessora arytmetycznego Intel 8087 oraz jego biblioteki,
- 4). zdefiniowanie czasu makrointerpolacji na wartość 32 ms,
- 5). ustawienie stanu początkowego wskaźników :
  - włączony jest stop awaryjny,
  - sygnalizowane jest przekroczenie limitu prądowego,
  - brak utraty pozycji,
  - nie ma sygnalizacji błędów,
  - robot nie jest zsynchronizowany,
  - nie jest włączony przycisk praca,
  - zerowany jest wskaźnik stosu pozycji,
  - uwzględniany jest przewijak pamięci kasetowej nr 2,
- 6). wywołanie funkcji inicjalizujących :
  - init\_mmfilgs() - inicjalizacja funkcji wypełniających obszar użytkowy,
  - init\_intmov() - inicjalizacja funkcji obsługujących ruch we współrzędnych wewnętrznych,
  - init\_extmov() - inicjalizacja funkcji obsługujących ruch we współrzędnych zewnętrznych,
  - init\_manipulacja() - inicjalizacja funkcji obsługujących ręczne operowanie systemem,
  - init\_interpreter() - inicjalizacja interpretera,
  - init hardware() - inicjalizacja funkcji obsługi sprzętu,
- 7). zapalenie wszystkich lampek panelu operacyjnego,
- 8). zgaszenie lampek panelu operacyjnego (z wyłączeniem lampki 'gotowosc').

### 5.3. Funkcja 'system()'. -----

Funkcja, której schemat blokowy przedstawiono poniżej, wywoływana jest jako 'system()'. Inicjuje ona obsługę istotnych elementów systemu sterowania. Są tu kolejno wywoływane następujące funkcje :

- 1). axes\_controller() - kontrola stanu sterowników osi.
- 2). progpanl() - obsługa panelu programowania,
- 3). oppanel() - obsługa panelu operacyjnego.



### 5.3.1. Procedura `'axes_controller'`.

Procedura `'axes_controller'` odczytuje słowa stanów sterowników osi i sprawdza, czy słowo stanu osi obrotowej wskazuje na wystąpienie przekroczenia maksymalnego błędu lub utratę synchronizacji oraz czy jest ustawiony sygnał INPOS. Jeśli tak, to zostają zacisnięte hamulce.

### 5.3.2. Procedura `'progpanl'`.

Procedura `'progpanl()`' jest wywoływana w podprogramie `'system'`, wykonuje się ona cyklicznie, co okres obliczeniowy programu sterującego. Kolejno wykonywane są następujące czynności :

- 1). wywołanie podprogramu odczytu adresu przesyłki.
- 2). wywołanie podprogramu obsługi przesyłki,
- 3). powrót z `'progpanl'` do procedury wywołującej.

Podprogramy bezpośrednio obsługujące przesyłki wykorzystują dwie procedury pomocnicze :

- `pkfault()` - sygnalizuje wystąpienie błędu podczas transmisji z/do pamięci kasetowej PK,
- `startper()` - sprawdza, czy można rozpocząć wykonywanie programu użytkowego.

Funkcje wykorzystywane przez procedurę `'progpanl()`' to :

- funkcja `cont()`,
- funkcja `cont_step()`,
- funkcja `cvecend()`,
- funkcja `cvecstrt()`,
- funkcja `get_geom()`,
- funkcja `getinter()`,
- funkcja `gripoper()`,
- funkcja `hold_pk()`,

- funkcja no\_parcel(),
- funkcja pgmstart(),
- funkcja pgmstop(),
- funkcja restrpgm(),
- funkcja savepgm(),
- funkcja sendvelo(),
- funkcja seterror(),
- funkcja setgeom(),
- funkcja storeref(),
- funkcja switchfr(),
- funkcje interfejsowe.

### 5.3.3. Procedura 'oppanel'.

Procedura 'oppanel()' obsługuje sygnały podawane na wejścia systemowego bloku wejść/wyjść dwustanowych pakietu MC-42 (w układzie sterowania jest również blok wejść/wyjść użytkownika). Do tego bloku są dołączone trzy przyciski umieszczone na panelu operacyjnym ( START PROGRAMU, ODCZYT KASETY, STOP PROGRAMU ) oraz dwa styczniki ( PRACA, STOP AWARYJNY). Procedura ta jest wykonywana cyklicznie, co okres obliczeniowy programu sterującego.

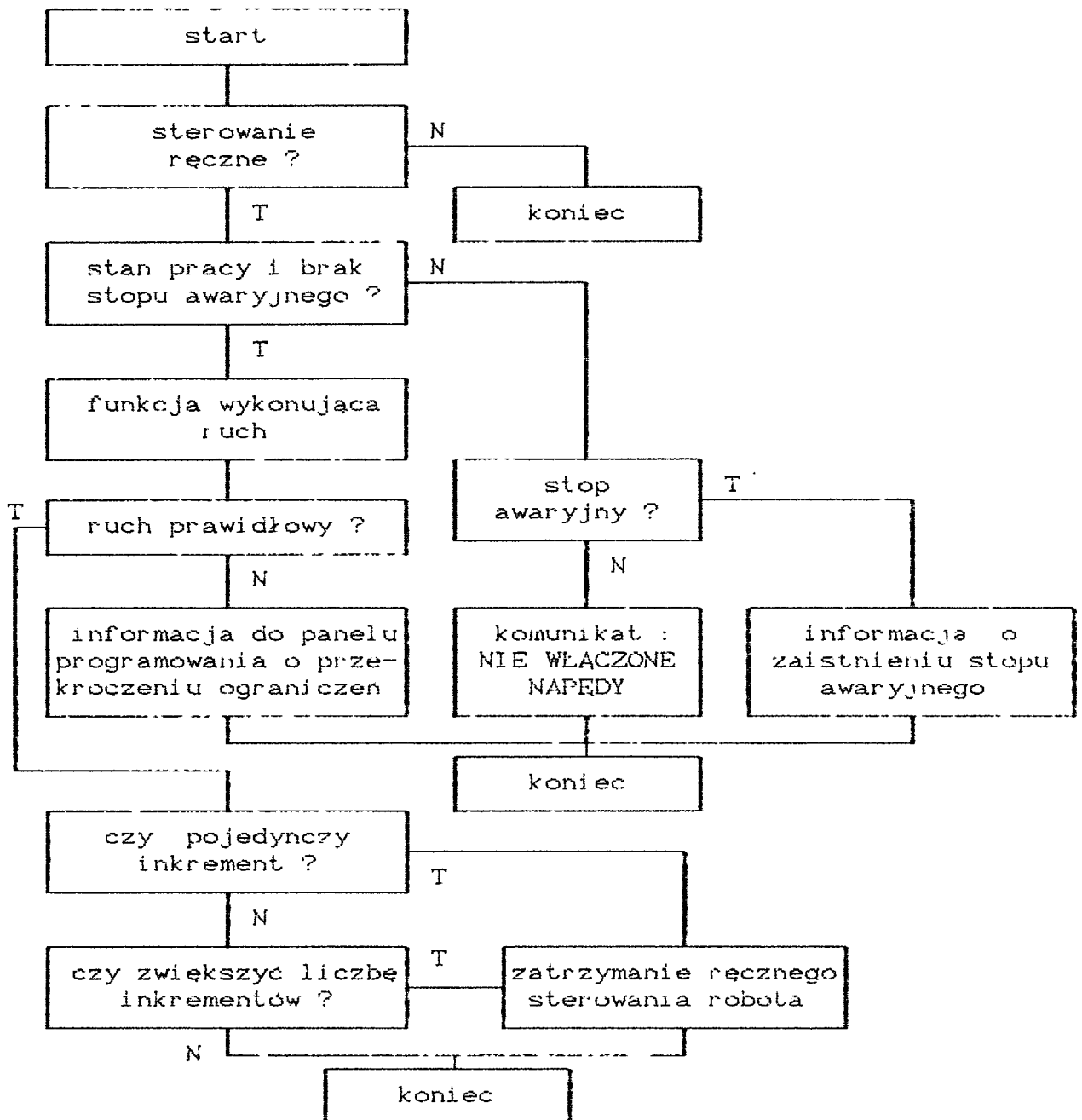
W procedurze 'oppanel' wykonywane są kolejno następujące kroki :

- 1). odczytanie stanu wejść bloku MC-42,
- 2). wykonanie procedur obsługujących sygnały :
  - procedura PROGRAM\_START\_BUTTON,
  - procedura READ\_CASSETTE\_BUTTON,
  - procedura PROGRAM\_STOP\_BUTTON,
  - procedura SEARCH\_STOP\_BUTTON,
  - procedura EMERGENCY\_STOP\_BUTTON,
  - procedura WORK\_BUTTON,
  - procedura PROG\_TEST\_BUTTON,
  - procedura OVERCURRENT\_SIGNAL.



#### 5.4. Funkcja manipulacja().

---



Powyzsza funkcja wywoływana jest jako 'manipulacja()'. Jej zadanie to realizacja sterowania ramieniem robota za posrednictwem joystick'a. Funkcja ta wykonuje kolejno następujące czynności :

- 1). sprawdzenie, czy jest stan pracy 'ręczne operowanie manipulatorem',  
jeśli odpowiedź jest negatywna, to funkcja kończy swoje działanie,
- 2). sprawdzenie, czy jest aktywny stan stopu awaryjnego oraz czy są włączone napędy,

jeśli stop awaryjny jest włączony lub napędy nie są włączone generowane są odpowiednie komunikaty o błędach i funkcja kończy działanie,

3). umożliwienie operatorowi poruszania manipulatorem robota (przy włączonych napędach i braku stopu awaryjnego),

4). zbadanie prawidłowości wykonania ruchu (po powrocie do funkcji wywołującej),

jeśli wystąpił błąd, to do panelu programowania przekazywany jest odpowiedni komunikat i funkcja kończy działanie; jeśli ruch został wykonany prawidłowo, to funkcja umożliwia operatorowi wybór ilości inkrementów ruchu do realizacji lub rezygnację z ruchu inkrementami; funkcja kończy działanie.

## 6. PODSTAWOWE CZYNNOŚCI PROGRAMU STERUJĄCEGO.

==== :=====

Podstawowe czynności programu sterującego to :

- umożliwienie synchronizowania manipulatora robota,
- spowodowanie quasiliniowego ruchu punktu końcowego manipulatora robota i ruchu pojedynczymi inkrementami we współrzędnych wewnętrznych,
- umożliwienie operowania manipulatorem robota za pomocą panelu programowania, (programu symulacyjnego),
- zapewnienie możliwości wykorzystywania panelu operacyjnego,
- zapewnienie możliwości współpracy z panelem programowania,
- interpretacja wprowadzonego przez użytkownika programu użytkowego,
- zapewnienie możliwości współpracy z pamięcią kasetową,
- obsługa błędów.

### 6.1. Synchronizacja manipulatora robota.

-----

Synchronizowanie robota, może być dokonywane przy pomocy odpowiedniego przycisku z panelu programowania (programu symulacyjnego). Procedurą umożliwiającą dojście do punktu synchronizacji jest 'getgeom'.

### 6.2. Ruch quasiliniowy.

-----

Ruch robota przy jednoczesnej pracy wszystkich sterowników zostaje nazwany pozycjonowaniem quasiliniowym. Wartość każdej współrzędnej wewnętrznej robota zmienia się liniowo w funkcji czasu. W efekcie uzyskuje się ruch liniowy w przestrzeni współrzędnych wewnętrznych. Nie jest to jednak ruch liniowy w przestrzeni kartezjańskiej, ze względu na nieliniowe przekształcenie współrzędnych wewnętrznych na współrzędne kartezjańskie.

### 6.3. Ręczne operowanie manipulatorem.

-----

Poruszanie manipulatorem realizowane jest w sposób typowy, tzn. deszyfrowane są standardowe przesyłki od panelu programowania (w tym przypadku od programu symulacyjnego, w którym odpowiednie przyciski klawiatury, wymuszają ruch poszczególnych osi manipulatora). Inicjalizacja ruchu manipulatorem robota jest wykonywana przez funkcję 'init\_manipulacja()', wywoływaną przez funkcję 'init\_master()'. Opis funkcji 'init\_manipulacja()' zamieszczono przy okazji opisu funkcji 'master()'. Następną wywoływaną funkcją jest funkcja wykonująca ruch. Jeśli program sterujący znajduje się w stanie pracy 'ręczne operowanie manipulatorem', to funkcja ta spowoduje wykonanie zadanego ruchu.

#### 6.4. Obsługa panelu operacyjnego.

---

Podprogram 'oppanel' obsługuje sygnały podawane na wejścia systemowych pakietów we/wy dwustanowych MC 42. Dołączone są tu trzy przyciski panelu operacyjnego (START PROGRAMU, ODCZYT KASETY oraz STOP PROGRAMU), dwa styczniki (PRACA i STOP AWARYJNY), a także sygnał informujący o przekroczeniu ograniczenia prądowego. Procedura 'oppanel' jest wywoływana w podprogramie 'system', jest więc wykonywana cyklicznie, co okres obliczeniowy programu sterującego.

#### 6.5. Współpraca z panelem programowania.

---

Komunikacja pomiędzy panelem programowania (czy też programem symulacyjnym) a sterownikiem odbywa się przez złącze szeregowe pakietu MM-16 i analogiczne złącze w panelu programowania.

Informacje są przekazywane w postaci przesyłek. Program sterujący nadaje przesyłki procedurą 'putparcl', a odbiera procedurą 'getparcl'. Przed wysłaniem przesyłki program sterujący musi przygotować bufor zawierający przesyłkę. Następnie poprzez wywołanie procedury 'putparcl' inicjuje proces wysyłania przesyłki do panelu programowania. Wszystkie przesyłki są przesyłane w postaci zakodowanej, co ma na celu zwiększenie możliwości wykrywania błędów w odbieranych przesyłkach oraz ułatwienie modyfikacji programu i diagnostyki sprzętu (możliwość bezpośredniego podłączenia złącz szeregowych do monitora z klawiaturą i obserwowania oraz symulowania przesyłek).

Procedura 'progpanl' jest wywoływana w podprogramie 'system', a więc wykonuje się cyklicznie, co okres obliczeniowy programu sterującego (podobnie jak procedura obsługi panelu operacyjnego 'oppanel'). Po wejściu do podprogramu 'progpanl' jest wywoływany podprogram odczytu adresu przesyłki. Wartość przez niego zwracana jest podstawiana pod zmienną, która wskazuje typ przesyłki oraz tablice jej parametrów. Jeśli zmienna ta zawiera wartość '0', oznacza to, że w buforze nie ma nieobsłużonej przesyłki.

Po odczytaniu adresu przesyłki wywoływany jest podprogram jej obsługi. Zakonczenie procesu obsługiwanego aktualnej przesyłki powoduje powrót z 'progpanl' do procedury wywołującej.

Program sterujący zawiera ponadto dwie procedury pomocnicze dla procedur bezpośrednio obsługujących przesyłki. Są to :

- procedura 'pkfault' - sygnalizująca wystąpienie błędu podczas transmisji z/do pamięci kasetowej,
- procedura 'startper' - sprawdzająca, czy można rozpocząć wykonywanie programu użytkowego (sprawdzenie synchronizacji, istnienia programu użytkowego).

## 6.6. Wykonywanie programu użytkowego.

-----

Program sterujący zawiera procedury, umożliwiające wykonywanie programu użytkowego, wprowadzonego przez użytkownika, za pomocą pamięci kasetowej systemu, bądź też poprzez ręczne wpisanie go z klawiatury panelu programowania (komputera, w przypadku pracy z programem symulacyjnym). Należy tu :

- procedura 'pgmstart' realizująca rozpoczęcie wykonywania programu użytkowego od pierwszej instrukcji,
- procedura 'cont' - realizująca rozpoczęcie wykonywania programu użytkowego od instrukcji bieżącej,
- procedura 'cont\_step' - realizująca wykonanie pojedynczej instrukcji programu (pracę krokową),
- procedura 'pgmstop' - obsługująca przesyłkę przerywającą wykonywanie programu.

## 6.7. Współpraca z pamięcią kasetową.

-----

Program sterujący robota PR-02E w wersji podstawowej umożliwia także korzystanie z pamięci kasetowej PK. Są tu wykorzystane następujące procedury :

- procedura 'restrpgm' - obsługująca przesyłkę z panelu programowania, nakazującą wczytanie programu z pamięci kasetowej,
- procedura 'savepgm' - przepisująca program użytkowy z pamięci RAM układu sterowania do pamięci kasetowej,
- procedura 'hold\_pk3' - realizująca zatrzymanie pamięci kasetowej w trakcie trwania transmisji.

## 6.8. Obsługa błędów.

-----

Do obsługi błędów powstałych w trakcie pracy służy procedura 'seterror'. Obsługuje ona przesyłkę z panelu programowania, informującą o wystąpieniu błędu.

## 7. WNIOSKI KOŃCOWE.

Dotychczasowy przebieg prac nad oprogramowaniem robota PR-02E nasuwa kilka spostrzeżeń :

- 1). Oprogramowanie w wersji handlowej powinno zawierać dodatkowo kilka funkcji poprawiających komercyjność obsługi robota, tj. możliwość budowy podprogramów w programie użytkownika, możliwość wykonywania ruchów względnych, umiejętność dokonywania pewnych prostych obliczeń na zmiennych logicznych oraz możliwość działania programu użytkowego w warunkowych pętlach logicznych. Należałoby więc rozbudowywać język użytkownika, kierując się w stronę popularnie znanego języka BASIC. Warto też rozpatrzyć możliwość opisu ruchu końcówki manipulatora we współrzędnych kartezjańskich, z uwzględnieniem opisu ewentualnych przeszkód znajdujących się w przestrzeni roboczej manipulatora.
- 2). W wypadku prowadzenia dalszych prac rozwojowych należałoby :
  - poprzez poprawę technologii wykonania elementów układu sterowania zwiększyć jego niezawodność,
  - przestarzały i zawodny układ zapisu/odczytu programów użytkowych na taśmie magnetycznej zastąpić zapisem/odczytem poprzez interfejs szeregowy na komputerze PC lub bezpośrednio na stacji dysków elastycznych w szafie sterowniczej,
  - zastosować docelowo perspektywiczny panel programowania lub komputer inżynierski PV25 firmy WINUEL w miejsce dotychczasowego komputera IBM PC.

## 8. LITERATURA.

=====

- [1] - Wykonanie programu sterującego dla robota IRp-2,5 kg. Opracowanie założeń na oprogramowanie. - Sprawozdanie PIAP - nr rej. 5957.
- [2] - Opis programu sterującego robotów IRp-6/60. Dokumentacja eksploatacyjna PIAP.