

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

Ośrodek Automatyki Elektrycznej
Pracownia Oprogramowania Wieloprocessorowych Systemów

074 Automatyki i Robotów A

Główny wykonawca

Wykonawcy : mgr inż. Bożena Siedlarska
mgr inż. Stanisław Woltański

Konsultant : mgr inż. Tomasz Wański

Nr zlecenia UR 010403

Opracowanie sekwencyjnego mikroprocesorowego
układu sterowania SP grupą robotów przemysłowych

Etap 7. Opracowanie programu sterującego
dla prototypowego zestawu robota.

Zleceniodawca problem węzłowy 06.1

Pracę rozpoczęto dnia 5.01.84

zakończono dnia 28.09.84

Kierownik Pracowni

p.o. Z-ca Dyrektora
d/s Automatyki

Kierownik Ośrodka

mgr inż. A. Aderek

dr inż. T. Gałązka

prof. dr inż. T. Missala

Praca zawiera:

Rozdzielnik - ilość egz: 2

stron 10

Egz. 1 BOINTE

rysunków 4

Egz. 2 OAE-83

fotografii

Egz. 3

tabel 3

Egz. 4

tablic

Egz. 5

załączników 1 /tabulogram
programu/

Egz. 6

Nr rejestr. 5273

Analiza deskryptorowa

ROBOT + KOMPUTER + MIKROPROCESOR + PROGRAMOWANIE +
JEZYK PROGRAMOWANIA

Analiza dokumentacyjna

Praca zawiera opis sposobu programowania robotów prostych PR-02 przy użyciu programu sterującego dla mikroprocesorowego układu sterowania grupą takich robotów. Załączono tabulogram nieuruchomionego programu sterującego ROBOT.

Tytuły poprzednich sprawozdań

Etap 3. Opracowanie struktury programu sterującego, opracowanie programu assemblera. /Nr rejestr. 5179 /.

338.45:62/69]. 002.1/2 roboty programowane
681.3.06 -oprogramowanie

UKD

MERA-PIAP/TW 331/78 5000

2

STRONA... 1
STRON.... 10
NR REJ. 5273

SPIS TRESCI

1. Przeznaczenie programu
2. Stany pracy programow uzytkowych i sposob ich sygnalizacji
3. Programowanie robota
4. Sygnalizowane bledy
5. Inicjalizacja programu
6. Konfiguracja prototypowego zestawu robotow

Zalacznik nr 1. Tabulogram programu sterujacego ROBOT

1. Przeznaczenie programu

Zadaniem programu ROBOT jest umożliwienie obsługi od jednego do ośmiu niezależnych robotów prostych PR-02, z których każdy składa się z co najwyżej 32 modułów dwupolozeniowych. Moduły o większej liczbie pól można traktować jako złożenie kilku modułów dwupolozeniowych (tzn. moduł trypolozeniowy odpowiada dwóm modułom dwupolozeniowym, moduł czterypolozeniowy - trzem dwupolozeniowym itd.) w sensie rezerwacji we/wy sterujących oraz zajętości pamięci RAM przeznaczonej na zapamiętanie kolejnych pól. Moduły mogą posiadać lub nie sygnalizację realizacji zadanego pola (sprzeżenie zwrotne).

Aktualna wersja programu ROBOT, zrealizowanego dla mikroprocesorowej jednostki centralnej 8-bitowej MM80, umożliwia obsługę modułów dwu- i trypolozeniowych.

Liczba modułów, ich typ oraz fizyczne adresy we/wy sterujących należy określić na etapie s e n e r a c j i programu dla danego zestawu robotów.

Przyjęto, że jednemu robotowi odpowiada jeden program, którego stan jest sygnalizowany na panelu programowania (diody "Rodzaj pracy" świeci się na zielono, na żółto, na czerwono lub jest zgaszona).

Sterowanie modułami robota polega na wykonywaniu uruchomionego programu użytkowego, zbudowanego z dziewięciu typów instrukcji wprowadzonych uprzednio z panelu programowania.

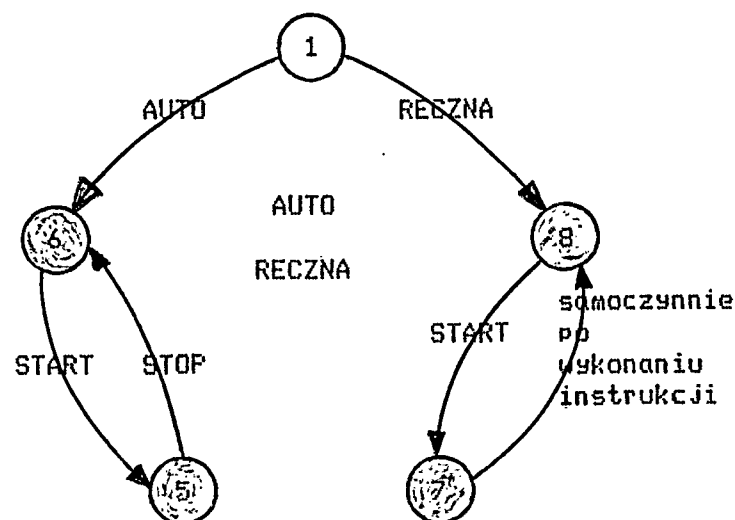
Program obsługuje p a n e l p r o g r a m o w a n i a, p a n e l s t e r o w a n i a, we/wy sterujące modułami robotów, we/wy obiektowe, zegar jednostki centralnej MM80.

Przewiduje się dołączenie procedur zapisu/odczytu programów użytkowych do/z pamięci zewnętrznej PK1 oraz wykorzystanie fragmentów programu ROBOT do testowania panelu programowania i kanału transmisji.

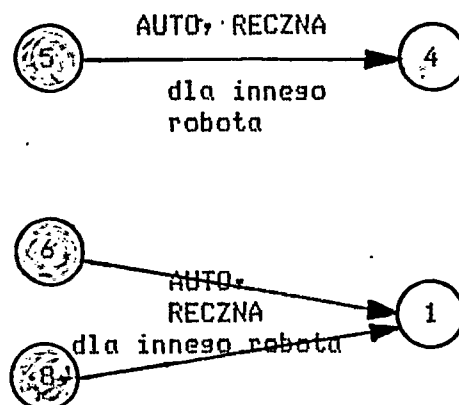
2. Stany pracy programow uzytkowych robota i sposob ich sygnalizacji

O stanie pracy kazdego z programow informuje wartosc slowa STAN. Program okreslamy jako o b s l u s i w a n y, gdy znajduje sie w stanie 5÷8 (znaczenie poszczegolnych stanow programu i sposob ich sygnalizacji zdefiniowano w tabeli 1.) Graf stanow takiego programu przedstawia rys.1.

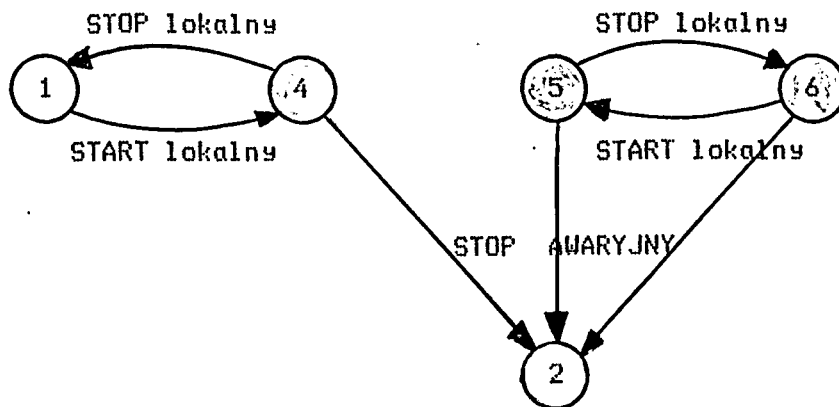
Zmiany stanow danego programu przy przejsciui do obslugi innego ilustruje rys.2, a rys.3 obrazuje zmiany stanow podczas korzystania z przyciskow panelu sterowania robota.



Rys.1. Zmiany stanow programu N przyciskami "Rozkazy robota" na panelu programowania.



Rys.2. Zmiany stanow programu N przy przejsciui do obslugi innego robota.



Rys.3. Zmiany stanów programów przy korzystaniu z przycisków panelu sterowania.

! kod ! STAN-u !	Znaczenie	Sygnalizacja	Uwagi
0	brak programu	N zgaszona	
1	STOP	N zgaszona	
2	STOP AWARYJNY	N czerwona↑	} Wykonywanie programu awaryjnie zawieszone
3	ZACIECIE	N czerwona↑	
4	AUTO	N zolta↑	
5	AUTO START	auto↑ start↑	} Tylko jeden program
6	AUTO STOP	auto↑	
7	RECZNA START	reczna↑ start↑	
8	RECZNA STOP	reczna↑	
		!N zielona↑ dla 5÷8 !	

Tabela 1. Stany programu i ich sygnalizacja.

Komunikacja programu z panelem programowania odbywa się na zasadzie interpretacji przerw, natomiast z panelami sterowania na zasadzie cyklicznego czytania odpowiednich wejść.

3. Programowanie robota

Czynność programowania robota, wykonywana wyłącznie w stanie OBSŁUGA, składa się z trzech faz: wprowadzania instrukcji, pracy krokowej i pracy automatycznej.

Wprowadzanie instrukcji polega na wpisywaniu z panelu programowania (STAN=8) następujących instrukcji:

- pozycjonowanie, czyli zapamiętanie w pamięci RAM aktualnego położenia wszystkich modułów robota;
- załączenie wyjścia, czyli zwarcie wyjścia o numerze 1÷16 lub ustawienie flagi programowej (arg.=101÷132);
- wyłączenie wyjścia, czyli rozwarcie wyjścia o numerze 1÷16 lub wyzerowanie flagi programowej (arg.=101÷132);
- czekanie przez określony czas (argument=1÷99 określa ilość jednostek 0.1 sekundowych);
- czekanie warunkowe na włączenie określonego wejścia (nr wej.=1÷13) lub flagi programowej (argument=101÷132);
- powtarzanie wybranej sekcji programu (argument=1÷99 na początku sekcji oraz arg.=0 na końcu sekcji);
- skok do instrukcji o zadanym numerze (arg.=10÷999);
- skok warunkowy (arg.=1÷13 lub arg.=101÷132)-omija następną instrukcję, gdy wejście jest zwarte lub flaga programowa ustawiona;
- koniec, bezargumentowa instrukcja zamykająca program użytkowy.

W tabeli 2 zestawiono wszystkie typy instrukcji oraz podano zakres ich argumentów. Na rys. 4 opisano przyciski panelu programowania i elementy sygnalizacyjne (podane liczby, w zapisie heksadecymalnym lub oktalnym, oznaczają kod przerywania z przycisku lub argument instrukcji OUT).

! Typ instrukcji !	Oznaczenie	! Zakres argumentu !
1	POZ.	-----
2	WY.ZAL.	1÷16 , 101÷132
3	WY.WYL.	1÷16 , 101÷132
4	CZEK.	1÷99
5	CZEK.WAR.	1÷13 , 101÷132
6	POWT.	0÷99
7	SKOK	10÷999
8	SKOK WAR.	1÷13 , 101÷132
9	KONIEC	-----

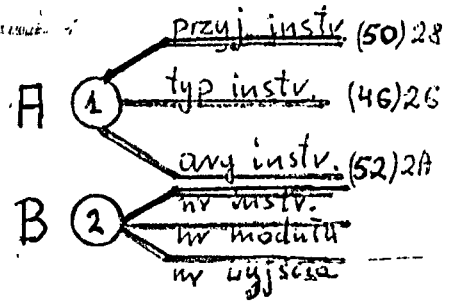
Tabela 2. Typy instrukcji i dopuszczalne wartości ich argumentów.

RODZAJ PRACY

30(60) 2E(56) 2C(54)

☒ OBSŁUGA ☐ AUTO ☐ STOP ☒ STOP AW.

N° ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8



PAMIĘĆ ZEWNĘTRZNA

☐ ZAPIS 26(46) ☐ ODCZYT

☐ (13) B ☐ (14) C

STEROWANIE MODUŁÓW

☐ (15) D

24(44) 22(42) 20(40)

☐ ☐ ☐

INSTRUKCJE

1. POZ. 2. WY. ZAK. 3. WY. WYŁ.

☐ (21) 11 ☐ (22) 12 ☐ (23) 13

4. CZEK. 5. CZEK. WAR. 6. POWT?

☐ (24) 14 ☐ (25) 15 ☐ (26) 16

7. SKOK 8. SKOK WAR. 9. KONIEC

☐ (27) 17 ☐ (30) 18 ☐ (31) 19

7 8 9

4 5 6

1 2 3

0 R (12) A

SYMULACJA

☐ (32) 1A

KASOWANIE SPR. INSTR.

☐ (33) 1B

☐ (34) 1C

NR INSTR.

☐ (35) 1D

NR MODUŁU

☐ (36) 1E

NR WYJŚCIA

☐ (37) 1F

STER. RECZNE

☐ - 26(46) ☐ +

☐ (17) F ☐ (16) E

ROZKAZY ROBOTA

28(50)

☐ AUTO ☐ RECZNA ☐ START ☐ STOP

☐ (40) 20 ☐ (41) 21 ☐ (42) 22 ☐ (43) 23

Rys. 4. Panel programowania robota PR-02.

W stanie RECZNA STOP dana instrukcje mozna skasowac (przycisk "Kasowanie") lub zastapic inna instrukcja.

Wykorzystujac przyciski "+" lub "-" ("Ster. reczne") oraz "Nr modulu" lub "Nr wyjscia" ustawiamy dany modul lub wyjscie w okreslone polozenie.

Poprawnosc instrukcji sprawdzamy przyciskiem "Spr. instr."
Po pierwszym nacisnieciu wyswietla sie kod typu instrukcji (liczba 1-9), po drugim argument (nie dotyczy POZ. i KONIEC).

Podczas testowania instrukcji SKOK WAR. i CZEK. WAR. mozna wykorzystac przycisk "Symulacja", ktorego uzycie sprawia, ze w czasie najblizszego testowania stanu danego wejścia (lub flasi programowej) program traktuje je jako wlaczone.

Praca krokowa polega na wykonywaniu jednej instrukcji o zadany numerze po wcisnieciu przycisku "Start", gdy program znajdowal sie w stanie 8. Na wyswietlaczu pojawi sie numer nastepnej instrukcji lub 10, jezeli wykonana instrukcja byl KONIEC. Stan programu zmienia sie na 7, a po wykonaniu instrukcji wraca na 8.

UWAGA: Przed wykonaniem pierwszej instrukcji wszystkie moduly powinny znajdowac sie w polozeniu skrajnym, wyjscia powinny byc rozwarte a flasi programowe wyzerowane. Te wstepna synchronizacje modulow i wyjsc realizujemy przy pomocy przycisku "Sterowanie modulow" (po wlaczeniu zasilania jest ona wykonywana automatycznie).

Praca automatyczna polega na wykonywaniu ciagu instrukcji programu uzytkowego poczynajac od pierwszej (o numerze 10) lub od zadanej na wyswietlaczu. Program moze znajdowac sie w stanie 4 lub 5; w tym drugim przypadku na wyswietlaczu jest widoczny numer aktualnie wykonywanej instrukcji. Mozemy wowczas sprawdzac synchronizacje dwu programow uzytkowych, z ktorych jeden zatrzymuje sie na instrukcji CZEK. WAR. czekajac na ustawienie flasi programowej przez drugi program.

Po wykonaniu instrukcji KONIEC program realizuje ponownie pierwsza instrukcje.

Wcisniecie przycisku "Stop" na panelu programowania lub sterownia zawiesza wykonywanie danego programu.

4. Sygnalizowane błędy

W trakcie pracy krokowej lub automatycznej na wyświetlaczu mogą pojawić się następujące kody wykrytych przez program sterujący błędów:

- 0 - zła kolejność przerwania z panelu programowania;
- 1 - wcisnięcie "+" lub "-", gdy wcześniej nie określono numeru wyjścia lub numeru modułu, lub gdy STAN nie był RECZNA STOP;
- 2 - próba skasowania, sprawdzenia lub wykonania nieistniejącej instrukcji;
- 4 - przekroczenie dostępnej pamięci RAM (za duży program użytkowy);
- 6 - próba wprowadzenia więcej niż 3 cyfr na wyświetlacz;
- 7 - zły argument instrukcji;

dla CZEK. poza [1,99],
dla POWT. poza [0,99],
dla WY.WYL. lub WY.ZAL. poza {[1,16],[101,132]},
dla CZEK.WAR. lub SKOK WAR. poza {[1,13],[101,132]},

lub próba wykonania "Reczna", "Auto", "Sterowanie modułów"
z numerem robota wyższym od zadeklarowanej ich liczby
podczas generacji programu ROBOT;

- 8 - instrukcja POWT. zamykająca petle ma arg. ≠ 0 lub otwierająca ma arg. = 0;
- 9 - numer instrukcji SKOK równy numerowi instrukcji docelowej (skok "na siebie");
- 10 - brak instrukcji KONIEC lub brak zamykającej petle instrukcji POWT.;
- 11 - błędne parametry generacji programu sterującego(*);
- 13 - niezidentyfikowany kod przerwania z panelu programowania(*);
- 21 - "Auto", gdy program był w innym stanie niż STOP lub RECZNA STOP;
- 22 - "Reczna", gdy program był w innym stanie niż STOP lub AUTO STOP;
- 23 - "Start", gdy program był w stanie różnym od AUTO STOP lub RECZNA STOP;

a po dołączeniu procedur obsługi pamięci kasetowej PK1:

- 31 - błąd danych na taśmie lub brak programu;
- 32 - pamięć nie włączona do sieci lub brak kasy;
- 33 - włączona blokada zapisu;
- 34 - pamięć niesprawna np. zablokowana taśma (*).

UWAGA: Na wyświetlaczu kod błędu jest zawsze poprzedzony literą E (tabela 3 podaje zawartość akumulatora instrukcji OUT sterującej wyświetlaczem)

Błędy oznaczone gwiazdka (*) nie są spowodowane pomyłką użytkownika.

! Wswietlany znak !	Kod-heksadecymalnie !	Kod-oktalnie !
nie swieci sie	0	0
0	3F	77
1	6	6
2	5B	133
3	4F	117
4	66	146
5	6D	155
6	7D	175
7	7	7
8	7F	177
9	6F	157
A	77	167
B	7C	174
C	39	71
D	5E	136
E	79	171
"spacja"	1C	34

Tabela 3. Dane sterujace siedmiosegmentowymi wskaźnikami na panelu programowania.

5. Inicjalizacja programu

Po wlaczeniu zasilania program sterujacy blokuje przerwania, gasi wswietlacz oraz diody sygnalizacyjne, wysterowuje zegar jednostki centralnej, ustawia wszystkie moduly w polozenie skrajne, zeruje wyjscia obiektowe oraz robocza pamiec RAM, a nastepnie bada stany kolejnych programow i realizuje wykonanie instrukcji.

6. Konfiguracja prototypowego zestawu robotow

Program ROBOT, sterujacy grupa robotow prostych PR-02, bedzie uruchamiany i testowany na prototypowym zestawie skladajacym sie z nastepujacej konfiguracji modulow:

- 1-szy robot: trzy moduly dwupolozeniowe ze sprzezeniem i Jeden bez sprzezenia,
- 2-gi robot: trzy moduly dwupolozeniowe bez sprzezenia i Jeden ze sprzezeniem,
- 3-ci robot: dwa moduly dwupolozeniowe ze sprzezeniem i Jeden obrotowy (trojpolozeniowy) ze sprzezeniem.
- 4-ty robot: dwa moduly dwupolozeniowe ze sprzezeniem i trzy moduly dwupolozeniowe bez sprzezenia.

16