

PRZEMYSŁOWY INSTYTUT AUTOMATYKI I POMIARÓW

MERA-PIAP

Al. Jerozolimskie 202

02-222 Warszawa

Telefon 23-70-81

Ośrodek Automatyki Elektrycznej

074

Zespół Budowy Cyfrowych Urządzeń Systemowych

Główny Wykonawca mgr inż. Janusz Zakolski

Wykonawcy dr inż. Wiesław Stańczak

Konsultant

Nr zlecenia

RP.53.3

Zad. 1.6

Opracowanie pakietu inteligentnych wejść analogowych dla sprzężenia robota z otoczeniem.

Zadanie 1.6. Uruchomienie prototypu

Instrukcja oprogramowania wewnętrznego pakietu MA-30.

Zleceniodawca CPBR 7.1 cel 53.

- Prace rozpoczęto dnia 87.07.01

Kierownik pracowni

mgr inż. K. Stefański

Kierownik Ośrodka

prof. dr inż. T. Missala

Z-ca Dyrektora d/s Automatyki

dr inż. F. Gałązka

zakończono dnia 87.10.15

Kierownik zespołu

dr inż. A. Syrczyński

Praca zawiera:

stron 15

rysunków

fotografii

tabel

tablic

załączników 2

Rozdzielnik - ilość egz:

Egz. 1 BOINTE

Egz. 2 OAE

Egz. 3 OAE

Egz. 4 OAE

Egz. 5 OAP

Egz. 6

Nr rejestr. 5956

Analiza deskrytorowa

URZĄDZENIA AUTOMATYCZNEJ REGULACJI I STEROWANIA:
KSAP + MIKROPROCESOR + INTELDIGIT-PROWAY + ROBOT +
+ INSTRUKCJA OPROGRAMOWANIA.

Analiza dokumentacyjna

Opracowanie zawiera opis oprogramowania wewnętrznego pakietu inteligentnych wejść analogowych MA-30 i opis trybu jego współpracy z jednostką centralną kasety INTELDIGIT-PROWAY.

Tytuły poprzednich sprawozdań

Opracowanie algorytmu i oprogramowania wewnętrznego
Nr rej. 5753.

UKD

SIAP-252/82-6000

681.5 Technika sterowania automatycznego
338.45:62/69].002.1/2 Roboty pneumatyczne

Spis treści

1. Wstęp
2. Zasada współpracy
3. Komunikat o zadaniu
4. Procedury
5. Opcje proceduralne

Załącznik 1: Kody wysyłane do procesora
nadrzędnego

Załącznik 2: Wykaz stałych miejsc pamięci
dwudostępnej wykorzystywanych
przy współpracy pakietu MA30
z procesorem nadrzędnym

1. Wstęp

Algorytm pakietu MA30 ma za zadanie umożliwienie odczytania poleceń przekazywanych z procesora nadrzędnego do Z80, zbadanie poprawności ich formy, ewentualne wygenerowanie kodu wynikowego programu odpowiadającego zadaniu do wykonania przesłanemu w poleceniu, uruchomienie i realizację wspomnianego programu oraz przekazanie do wglądu procesora nadrzędnego wyników tegoż programu.

Ogólna idea współpracy pakietu MA30 z procesorem nadrzędnym została już omówiona w pracy: "Opracowanie pakietu inteligentnych wejść analogowych dla sprzężenia robota z otoczeniem. Opracowanie algorytmu i oprogramowania wewnętrznego" nr rej. 5753. W trakcie dalszych prac przy pakiecie MA30 rozszerzono możliwości algorytmu i zwiększono jego efektywność. Wiązało się to ze zmianami niektórych szczegółów /jak np. numery błędów i podział procedur na grupy/, z zachowaniem jednakże wspomnianej koncepcji ogólnej. Dlatego też te informacje zostaną raz jeszcze przedstawione w prezentowanej pracy. Główny jednak nacisk kładzie się tu na dokładne opisanie trybu współpracy pakietu MA30 z procesorem nadrzędnym.

2. Zasada współpracy

Pakiet MA30 komunikuje się z procesorem nadrzędnym poprzez pamięć dwudostępną. Ponadto w sytuacjach awaryjnych oraz po zakończeniu wykonywania zadania do procesora nadrzędnego jest dodatkowo kierowane przerwanie.

Po włączeniu pakietu automatycznie wykonywane są niezbędne czynności wstępne. W trakcie ich trwania między innymi zerowane są następujące obszary pamięci dwudostępczej:

4

8000H-87FFH, 8900H-8DFFH i 8FO0H-8FFFH.

W przypadku stwierdzenia błędu EPROMu /umożliwiającego jednak dalej opisaną reakcję/ lub błędu fragmentu pamięci dwudostępnej od 8FO0H do 8FFF /gdzie są właśnie kierowane komunikaty z Z80/ procesor Z80 jest zatrzymywany /wykonywana jest instrukcja HALT/ po czym działa mechanizm budzika /WATCHDOG/. Warto tu zwrócić uwagę, że w omawianym przypadku zawartość komórki KOM /adres 8FFEh/ jest nieokreślona.

Reakcja na błąd pamięci jedno- lub dwudostępnej /w zakresie 8000H-87FFH i 8900H-8DFFH/ jest analogiczna, wysyłany jest jednakże dodatkowo numer błędu do komórki KOM /odpowiednio 255 lub 253/ z następującym po nim przerwaniem skierowanym do procesora nadrzędnego.

W przypadku prawidłowego przebiegu inicjacji pakietu MA30 procesor nadrzędny otrzymuje kolejno dwa przerwania poprzedzone następującymi wpisami do komórki KOM: 11 - pozytywny wynik testu pamięci dwudostępnej; 9 - pozytywny wynik testu pamięci jednodostępnej, a także w 8FFAH i 8FFBH odnotowany jest fizyczny koniec pamięci jednodostępnej /pierwszy - mniej znaczący bajt - powinien zawierać FFH/. Pakiet MA30 jest teraz gotowy do współpracy z procesorem nadrzędnym i oczekuje na zadanie.

Ogólna zasada dalszej współpracy procesora nadrzędnego z pakietem MA30 jest następująca. Mianowicie komunikaty do procesora nadrzędnego są zapisywane do komórki KOM, po czym generowane jest przerwanie. Zaś MA30 w oczekiwaniu na zadanie periodycznie bada zawartość komórki KOM1 /adres 8FFCH/, a po wygenerowaniu kodu wynikowego programu realizującego zadanie oraz w trakcie przetwarzania tego programu /wyjątek może tu

stanowić niewłaściwie skonstruowana opcja proceduralna pierwszego rodzaju - patrz p. "Opcje proceduralne"/ - sprawdza zawartość komórki LSTR /adres 8FFFFH/.

Oczekujący na zadanie pakiet MA30 wysyła przerwanie do procesora nadrzędnego wpisując uprzednio 5 do komórki KOM, następnie zeruje zawartość KOM1 i periodycznie bada KOM1. Po przygotowaniu zadania procesor nadrzędny redaguje je w formie komunikatu /patrz p. "Komunikat o zadaniu"/, a następnie do KOM1 przesyła liczbę bloków zawartych w pierwszej części komunikatu /od 1 do 255/. Po stwierdzeniu niezerowej zawartości KOM1 pakiet MA30 przystępuje do sprawdzania poprawności komunikatu, a następnie do generacji kodu wynikowego programu realizującego wprowadzone zadanie. Wykrycie błędu jest sygnalizowane przerwaniem skierowanym do procesora nadrzędnego poprzedzonym wpisaniem numeru błędu do KOM /patrz - "Załącznik 1"/, po czym pakiet MA30 znowu przechodzi do oczekiwania na zadanie.

Utworzenie kodu wynikowego sygnalizowane jest wpisaniem kodu 7 do KOM, po czym pakiet MA30 rozpoczyna periodyczne testowanie LSTR. Stwierdzenie niezerowej zawartości LSTR powoduje rozpoczęcie wykonywania programu. Realizacja programu może zakończyć się na skutek wyzerowania wszystkich liczników powtórzeń poszczególnych podzadań /procedur/ lub też zostaje wymuszona przez procesor nadrzędny, który w tym celu wpisuje dowolną niezerową wartość do LSTR /wyjątek może tu stanowić niewłaściwie skonstruowana opcja proceduralna pierwszego rodzaju - patrz p. "Opcje proceduralne"/. Po zakończeniu przetwarzania programu wynikowego generowane jest przerwanie skierowane do procesora nadrzędnego poprzedzone wpisaniem odpowiedniego kodu do KOM /1 - w przypadku zakończenia programu

6

wskutek wyzerowania wszystkich liczników powtórzeń; 3 - zakończenie spowodowane niezerową zawartością komórki LSTR/, po czym znowu pakiet MA30 przechodzi do oczekiwania na kolejne zadanie.

3. Komunikat o zadaniu

Format komunikatu jest ściśle ustalony. Komunikat składa się z dwóch części, przy czym pierwsza z nich musi zawsze występować. Pojawienie się zaś drugiej części komunikatu warunkowane jest zawartością części pierwszej. Obydwie części składają się z następujących po sobie kolejnych bloków, z których każdy odpowiada pojedynczej procedurze. Formaty bloków części pierwszej komunikatu zostaną przedstawione w bieżącym punkcie, zaś formaty bloków drugiej części komunikatu omówiono w p. "Opcje proceduralne" oraz p. "Procedury".

Pierwsza część komunikatu wpisywana jest w kolejne komórki począwszy od 8000H i musi być zakończona pojedynczym bajtem o kodzie FFH. Liczba występujących w tej części bloków powinna się równać zawartości komórki KOM1. W przypadku wystąpienia opcji proceduralnych typu pierwszego /TOP1/ komunikat musi składać się tylko z pojedynczego bloku, zaś opcja proceduralna typu drugiego /TOP2/ może być poprzedzona pewną liczbą procedur współpracy z przetwornikiem /T0/ /patrz p. "Opcje proceduralne" i p. "Procedury"/. Jednakże blok odpowiadający TOP2 powinien być ostatnim blokiem pierwszej części komunikatu.

Liczba procedur typu T0, T2, T3, T4 i TPP /patrz p. "Procedury"/ jest dowolna /od 1 do 255/, musi się jednak równać zawartości komórki KOM1. Jedynym ograniczeniem jest zastrzeżenie, że procedura T0 może być tylko raz wywoływana

dla każdego kanału z osobna.

Bloki dotyczące opcji proceduralnych są jednobajtowe - zawierają one wyłącznie numer procedury /1 lub 2/. Bloki dla T0, T2, T3 i T4 są ośmiobajtowe i składają się z czterech pól /patrz rys. 1/.

Pole 1	numer procedury licznik powtórzeń
Pole 2	adres danych
Pole 3	częstotliwość wykonania
Pole 4	adres wyników

Rys. 1. Format bloku cz. 1 komunikatu w przypadku procedur typu T0, T2, T3 i T4.

Pole 1 zawiera numer procedury - tzn. liczbę z zakresu od 0 do 127 odpowiadającą zimplementowanej procedurze bibliotecznej typu T0, T2, T3 i T4 /w przypadku próby wywołania procedury niezimplementowanej lub typu T1 sygnalizowany jest błąd/; oraz licznik powtórzeń - tzn. liczbę z zakresu od 1 do 255 wskazującą na to ile razy procedura będzie wykonywana lub 0 - co odpowiada nieokreślonej liczbie wykonań, W przypadku T0 licznik powtórzeń jest ignorowany.

Pole 2 zawiera adres danych dla procedury. W przypadku T0 musi to być liczba z zakresu 0 - 7 - odpowiada ona numerowi kanału.

Pole 3 zawiera proponowaną częstotliwość /z zakresu 2 Hz + 10 kHz/, z którą powinna być wykonywana dana procedura. W przypadku procedur typu T4 omawiane pole musi być wyzerowane /w przeciwnym razie generowany jest sygnał błędu/. Procedury typu T3 powinny mieć zadawaną niezerową częstotliwość wykonania /jeżeli zdarzy się inaczej, to generowany jest sygnał błędu/.

W przypadku procedur typu T0 i T2 wyzerowanie omawianego pola odpowiada przyjęciu tzw. opcji częstotliwościowej, czyli przyjęciu, że dana procedura jest wykonywana z największą możliwą częstotliwością wynikającą z czasu przetwarzania całego zadania oraz z częstotliwości określonych dla innych procedur.

Należy tu jeszcze wspomnieć, iż zawartość pola 3 procedur typu T0 jest ignorowana, jeśli po nich pojawia się TCP2.

Pole 4 zawiera adres wyników. W przypadku T0 jest ignorowane.

Dane w polach 2, 3 oraz 4 zapisywane są w ogólnie przyjętej kolejności, tzn. pierwszy występuje mniej znaczący bajt.

Pole 0	03H typ procedury
Pole 1	numer procedury licznik powtórzeń
Pole 2	adres danych
Pole 3	częstotliwość wykonania
Pole 4	adres wyników

Rys.2. Format bloku cz.2 komunikatu w przypadku procedur typu TP

Bloki dotyczące TPP są dziesięciobajtowe i składają się z pięciu pól. Ich format pokazano na rys. 2.

Pole 0 zawiera numer /identyfikator/ procedury TPP, a następnie typ procedury, na którą ma być przekształcona specyfikowana dalej procedura biblioteczna. Dozwolona jest konwersja na typ T2 /kod 3FH/ i na typ T3 /kod 5FH/. Próba innych przekształceń powoduje wygenerowanie sygnału błędów.

Pola 1, 2, 3 i 4 mają taką samą interpretację jak w przypadku procedur typu T0, T2, T3 i T4. Jednakże częstotliwość rozpatrywana jest jednak w kontekście nowego typu procedury, a nie zadanego w polu 1.

4. Procedury

Procedury są specyfikowane przez podanie ich numerów. Numer /nazwa/ procedury musi być liczbą z zakresu 0 - 127 /00H-7FH/. Poza ograniczeniami omówionymi w poprzednim punkcie nie ma żadnych zastrzeżeń ani do tworzenia dowolnych zestawów procedur, ani też do wielokrotnego wywoływania tej samej procedury /być może z innymi parametrami/ w komunikacie. Na przykład chcąc utworzyć filtr kaskadowy składający się z pewnej liczby członów bikwadratowych należy określić dane członu wejściowego jako wyjście z badanego kanału, zaś adres wyników omawianej procedury podać jako adres danych następnego członu bikwadratowego /być może o innych parametrach/ itd.

Numer procedury określa jej przynależność do konkretnego typu, który z kolei implikuje podejście do omawianej procedury podczas generowania kodu wynikowego programu realizującego zadanie. Nie wdając się bliżej w szczegóły implementacyjne

wspominamy tu tylko, że chodzi tu przede wszystkim o dwa aspekty zagadnienia, a mianowicie o to, czy przetwarzane są wyłącznie procedury biblioteczne, czy też dopuszcza się programy własne użytkownika oraz sposób w jaki realizuje się częstotliwość wykonywania danej procedury narzuconą uprzednio w komunikacie. Rozróżnia się następujące typy procedur:

TO - numer 0 - oznacza procedurę biblioteczną pomiaru zadanego kanału bez jakichkolwiek czynności dodatkowych. Może ona współpracować z dowolnym zestawem innych procedur dodatkowych oraz z TOP2. Zabronione jest wywoływanie kilkakrotnie TO dla tego samego kanału.

Wyniki procedury TO przekazywane są w ustalone miejsce pamięci dwudostępnej:

kanał 0 - 8000H, 8001H
kanał 1 - 8002H, 8003H
kanał 2 - 8004H, 8005H
kanał 3 - 8006H, 8007H
kanał 4 - 8008H, 8009H
kanał 5 - 800AH, 800BH
kanał 6 - 800CH, 800DH
kanał 7 - 800EH, 800FH

/pierwszy bajt - mniej znacząca część wyniku/.

TOP1 - numer 1 - opcja proceduralna pierwszego rodzaju, patrz p. "Opcje proceduralne".

TOP2 - numer 2 - opcja proceduralna drugiego rodzaju, patrz p. "Opcje proceduralne".

TPP - numer 3 - pseudoprocedura - oznacza przekształcenie typu procedury bibliotecznej z TO, T2, T3 lub T4 na T2 ewentualnie T3 z punktu widzenia realizacji narzuconej częstotliwości wykonania.

- T1 - numery od 04H do 0FH - numery procedur zastrzeżone dla ewentualnych dalszych rozszerzeń. Obecnie ich wywołanie traktowane jest jako błąd.
- T2 - numery od 10H do 3FH - procedury biblioteczne nie wymagające ścisłego przestrzegania żądanej częstotliwości wykonania /patrz p. 3/. Częstotliwość wykonania może być tu definiowana lub podawana jako opcja.
- T3 - numery od 40H do 5FH - procedury biblioteczne wymagające ścisłego przestrzegania zadanej częstotliwości wykonania. W ich przypadku nie wolno stosować mechanizmu opcji częstotliwościowych.
- T4 - numery od 60H do 7FH - procedury biblioteczne wymagające ścisłego przestrzegania uprzednio ustalonej częstotliwości wykonania. W przypadku procedur typu T4 częstotliwość ich wykonywania jest na stałe zapisana w EPROMie, zaś w komunikacie pole 3 powinno być wyzerowane /w przeciwnym przypadku generowany jest sygnał błędu/.

W obecnej wersji pakietu MA30 realizowane są: procedura T0, TOP1, TOP2, TPP, pięć procedur typu T2 o numerach od 3AH do 3EH i jedna procedura typu T3 oznaczona 5FH. Próba wywołania innych procedur kończy się zasygnalizowaniem błędu.

Wymienione procedury realizują, odpowiednio, następujące funkcje:

- 3AH - przepisuje pojedynczą daną /2 bajty/, np. wynik pomiaru ewentualnie wynik działania innej procedury z zadanego miejsca w inne, również zadane miejsce;
- 3BH - dokonuje przekształcenia dyskretnego /wg zadanej tabelki ujmującej końce przedziałów argumentów na odpowiadające tym przedziałom wartości wyjściowe/ pojedynczej danej i wynik wpisuje w określone miejsce,
- 3CH - począwszy od adresu wyniku zadanego w cz. A komunikatu aż do wyzerowania licznika cykli /umieszczonego w cz. 2 komunikatu/ w sposób cykliczny wypełnia obszar pamięci dwudostępnej kolejnymi danymi czerpanymi spod adresu określonego w cz. 1 komunikatu;
- 3DH - wyznacza wartość minimalną przebiegu w czasie, tzn. pobiera daną spod wskazanego adresu, porównuje z dotychczasową wartością minimalną i mniejszą spośród tych dwu liczb umieszcza pod wskazanym adresem jako wynik;
- 3EH - wyznacza wartość maksymalną przebiegu w czasie, działając analogicznie do procedury 3EH.
- 5FH - człon bikwadratowy filtru cyfrowego: pobiera dane spod określonego adresu i po dokonaniu przekształcenia wpisuje otrzymany wynik pod wskazany adres. Parametry filtracji zadaje się w cz. 2 komunikatu.

Procedura TO i procedury o numerach 3AH, 3DH oraz 3EH nie mają żadnego odzwierciedlenia w cz. 2 komunikatu. Procedury 3BH, 3CH i 5FH korzystają z parametrów przekazywanych w cz. 2 komunikatu. Format parametrów pozostaje w ścisłym związku

z numerem procedury i we wspomnianych przypadkach jest pokazany poniżej.

1 bajt	liczba przedziałów $\neq n$	}	2/n-1/ bajtów
2 bajty	kres górny przedziału 1 ...		
2 bajty	kres górny przedziału n-1	}	2n bajtów
2 bajty	wartość odp. przedziałowi 1 ...		
2 bajty	wartość odp. przedziałowi n		

Rys. 3. Format bloku cz. 2 komunikatu odpowiadający procedurze 3CH.

2 bajty	długość wypełnionego obszaru pamięci
---------	--------------------------------------

Rys. 4. Format bloku cz.2 komunikatu odpowiadający procedurze 3DH.

2 bajty	mnożnik przy z^{-2} mianownika
2 bajty	mnożnik przy z^{-1} mianownika
2 bajty	mnożnik przy z^{-2} licznika
2 bajty	mnożnik przy z^{-1} licznika
2 bajty	współczynnik skalujący

Rys. 5. Format bloku cz. 2 komunikatu odpowiadający procedurze 5FH.

5. Opcje proceduralne

Opcje proceduralne /TOP1 i TOP2/ są to procedury w całości definiowane i wprowadzane do pamięci dwudostępnej przez użytkownika /na jego wyłączną odpowiedzialność/. Różnią się one między sobą stopniem wykorzystania rejestrów procesora wewnętrznego pakietu MA30 oraz sposobem współpracy z procedurami bibliotecznymi. W przypadku TOP2 w pełni wykorzystuje się mechanizm pętli czasowej /przerwanie niemaskowalne przychodzące ze z góry określoną częstotliwością/ i można wywoływać procedurę współpracy z przetwornikiem /TO/. Wywołanie innych procedur bibliotecznych jest zabronione. Ponadto w treści TOP2 nie wolno wykorzystywać rejestrów alternatywnych i rejestru IX. Poza tym należy pamiętać, że po każdym przerwaniu niemaskowalnym /przychodzącym cyklicznie z zadaną częstotliwością/ niszczone są zawartości rejestrów HL, DE i SP /HL jest ustawiane na 47B1H, SP - na 407CH, zaś D - na 8FH/. Format bloku cz. 2 komunikatu towarzyszący TOP2 pokazano na rys. 6.

2 bajty	częstotliwość wykonania
2 bajty	długość programu użytkownika
2 bajty	adres początkowy kodu programu
	PROGRAM UŻYTKOWNIKA /TOP2/
2 bajty	adres startu programu
1 bajt	FFH
2 bajty	adres inicjacji programu
1 bajt	FFH

Rys. 6. Format bloku cz. 2 komunikatu towarzyszący TOP2.

Opcja proceduralna TOP1 jest pomyślana przede wszystkim w celu umożliwienia efektywnego wykorzystania mocy obliczeniowej mikroprocesora pakietu MA30 w czasie wystąpienia braku potrzeby współpracy z przetwornikiem. TOP1 w odróżnieniu od TOP2 może wykorzystywać wszystkie rejestry Z80 będącego mikroprocesorem pakietu MA30. Nie może jednak w sposób jawny /tzn. przez specyfikowanie w komunikacie/ ani wykorzystywać mechanizmu przerwania niemaskowalnego przychodzącego cyklicznie z zadaną częstotliwości, ani też współpracować z żadną z procedur bibliotecznych. Ponadto w przypadku wykorzystania TOP1 należy pamiętać o tym, aby program użytkownika zapewniał uaktywnienie WATCHDOGa /instrukcją OUT /8/,A - z dowolną

zawartością rejestru A/ w nie mniej niż 1,5 s. Format bloku cz. 2 komunikatu towarzyszący TOP1 przedstawiono na rys. 7.

2 bajty	długość programu użytkownika
2 bajty	adres początkowy kodu programu
	PROGRAM UŻYTKOWNIKA /TOP1/
2 bajty	adres startu programu
1 bajt	FFH

Rys. 7. Format bloku cz. 2 komunikatu towarzyszący TOP1.

Załącznik 1

Kody wysyłane do procesora nadrzędnego

I. Pytania i potwierdzenia

- 1 - pozytywny koniec wykonania zadania
- 3 - pozytywny koniec wykonania zadania wymuszony jednak przez procesor nadrzędny
- 5 - oczekiwanie na komunikat /gotowość do przyjęcia komunikatu/
- 7 - pozytywne przyjęcie komunikatu zakończone wygenerowaniem kodu wynikowego zadania i oczekiwanie na sygnał startu przetwarzania
- 9 - pozytywny wynik testu pamięci jednodostępnej
- 11 - pozytywny wynik testu pamięci dwudostępnej

II. Zawiadomienia o błędach

- 255 - błąd pamięci jednodostępnej
- 253 - błąd pamięci dwudostępnej
- 251 - trudny do rozpoznania błąd komunikatu, np. brak znacznika FFH na końcu komunikatu, zbyt duży numer procedury, itp.
- 249 - wywołanie niezimplementowanej procedury bibliotecznej
- 247 - błędny parametr T0 lub próba kilkukrotnej komunikacji z tym samym kanałem w trakcie jednego cyklu
- 245 - błąd konwersji typu /TPP/ lub wywołanie procedury z grupy T1
- 243 - opcja częstotliwościowa przy procedurze typu T3 lub próba wymuszenia częstotliwości w przypadku procedury typu T4
- 241 - zbyt mała częstotliwość wykonania programu realizującego zadanie

- 239 - przekroczenie dopuszczalnej wartości mnożnika /255/
przy ustalaniu najmniejszej wspólnej częstotliwości
realizacji procedur typu T3 i T4
- 237 - zbyt duża częstotliwość /zadana lub wyliczona/
- 235 - kod wynikowy zadania nie mieści się w pamięci jedno-
dostępnej

Załącznik 2

Wykaz stałych miejsc pamięci dwudostępnej wykorzystywanych przy współpracy pakietu MA30 z procesorem nadrzędnym

- 8000H - początek pola przeznaczonego na wpisanie komunikatu
- 87FFH - koniec pola przeznaczonego na wpisanie komunikatu
- 8FFAH - mniej znaczący bajt ostatniego adresu pamięci jednodostępnej
- 8FFBH - bardziej znaczący bajt ostatniego adresu pamięci jednodostępnej
- 8FFCH - KOM1 - długość komunikatu
- 8FFEH - KOM - wiadomości z MA30 przesyłane do procesora nadrzędnego
- 8FFFH - LSTR - sygnał startu/stopu przetwarzania zadania w pakiecie MA30