

0411

A

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

Ośrodek Automatyki Elektrycznej

Pracownia Systemów Wizyjnych

Główny wykonawca

dr inż. Bohdan Kontrymowicz



Wykonawcy

mgr inż. Andrzej Wojtych
 dr inż. Jacek Frontczak
 mgr inż. Dariusz Okrasa
 inż. Leszek Kołodziejczyk

Konsultant

Nr zlecenia 1117

Czujnik wizyjny do wykrywania,
 zliczania i pomiarów obiektów
 w zakresie do 0,02 m

Etap 3 - Oprogramowanie użytkowe,
 badania funkcjonalne i rewizja
 dokumentacji po badaniach.

Zleceniodawca

Pracę rozpoczęto dnia

02.05.89

zakończono dnia

31.10.89

Kierownik Pracowni

Kierownik Ośrodka

dr inż. J. Frontczak

Z-ca Dyrektora d/s
 Automat. i Pomiar.

dr inż. B. Kontrymowicz

doc. dr inż. T. Gałązka

Praca zawiera:

Rozdzielnik - ilość egz:

stron 13
 rysunków
 fotografii
 tabel
 tablic
 załączników 1

Egz. 1 BOINTE
 Egz. 2 ORGMASZ
 Egz. 3 OAE
 Egz. 4
 Egz. 5
 Egz. 6

Nr rejestr. 6385

Analiza deskryptorowa

URZĄDZENIE
POMIAR + SYSTEMY WIZYJNE, PROGRAM

Analiza dokumentacyjna

Sprawozdanie zawiera opis oprogramowania użytkowego oraz listę programu.

Tytuły poprzednich sprawozdań

Założenia i projekt układu
Nr rejestr. 6032

621.397 6 Urządzenia w cyfry
681.3.08 Pomiar

UKD

OPROGRAMOWANIE UKŁADU POMIAROWEGO WIZYJNEGO ZESTAWU
POMIAROWEGO WZP - 3/89

Spis treści

1. Informacje ogólne
2. Przeznaczenie oprogramowania
3. Środowisko sprzętowe programu
4. Skrócony opis działania programu

1. Informacje ogólne

Oprogramowanie układu pomiarowego wizyjnego zestawu pomiarowego WZP w wykonaniu 3/89 jest napisane w asemblerze mikroprocesora Intel 8048. Jest związane z urządzeniem i nie może być przenoszone do innego środowiska sprzętowego.

Na postać źródłową programu składają się pliki:

S.S48 (plik pierwotny) oraz BCD.PP, INI.ALL, IF, IWR.PP, MACRO.VV, STALE.VV.

Wynikowy plik binarny BIN uzyskano z postaci źródłowej za pomocą systemu asemblacji skróśnej REL48 (programy makroassemblera skróśnego REL48 i linkera LINK48) oraz programu tłumaczącego postać heksodecymalną na binarną HEXOBJ.EXE.

System asemblacji skróśnej REL48 generuje pliki *LST (listing), *.R48 (kod pośredni), *.MAP (mapa programu), *.A48 (kod heksadecymalny), przy czym nazwa * może być dowolnie określona przez użytkownika.

Do wsadowego przetwarzania programu źródłowego wykonano plik RUN.BAT. Program układu pomiarowego jest wpisany do pamięci ROM typu 2732, która zajmuje cały dostępny obszar 0000H-0FFFH adresowania mikroprocesora Intel 8048. Program zajmuje ok. 80% tego obszaru.

2. Przeznaczenie oprogramowania

Przeznaczenie programu wynika z wymagań funkcjonalnych wizyjnego zestawu pomiarowego WZP w wykonaniu 3/89. W szczególności zadaniem programu jest komunikacja z urządzeniem sterującym zestawu pomiarowego i wykonywanie jego poleceń związanych z obsługą pomiarowych torów kamer linijkowych CCD i zewnętrznych urządzeń rejestrujących.

3. Środowisko sprzętowe programu

Sprzętowe środowisko programu jest specyficzne dla urządzenia. Wynika z typu mikroprocesora jednostki centralnej układu pomiarowego, konfiguracji sprzętowej jednostki centralnej i otoczenia jednostki centralnej, które stanowią pakiety układu pomiarowego.

3.1 Wewnętrzne środowisko mikroprocesora.

- 128 rejestrów procesora, z których:

R0-R7 i R24-R31 stanowią przełączne banki 0 i 1 adresowane bezpośrednio,

R8-R23 stanowią ośmio poziomowy stos automatycznie używany przez mikroprocesor przy wykonywaniu instrukcji CALL, RET i RETR,

R32-R127 stanowią wewnętrzną pamięć RAM użytkownika adresowaną pośrednio poprzez rejestry R0/R1 aktywnego banku 0 lub 1. Opis wykorzystywania rejestrów R32-R127 w programie zawiera plik STALE.VV (zał.1)

- port P1

- port P2

linia P23 wykorzystywana do przełączania układów adresowanych jako IO:

P23=0 zewnętrzna pamięć RAM

P23=1 układy wejściowe-wyjściowe

linie P22-P20 używane do przełączania stron zewnętrznej pamięci RAM.

3.2. Środowisko jednostki centralnej

- pamięć programu (ROM) o adresach 000H-OFFFH stanowiących dwa banki 000H-7FFFH i 800H-OFFFH,
- zewnętrzna pamięć RAM mikroprocesora: osiem stron o adresach 000H-OFFFH wybieranych programowo a liniami P22-P20 portu P2 mikroprocesorowe, adresowana jak układy IO,
- układ generujący sygnał RESOUT (adres RESA=20H),
- rejestr RQR przerwań (adres RQA=24H),
- przełączniki SW ogólnego przeznaczenia (adres SwA=20H),
- programowany licznik - zegar (układ 8253) TC o adresach:
TC0 = 28H - licznik 0
TC1 = 29H - licznik 1
TC2 = 2AH - licznik 2
TCC = 2BH - programowanie układu

3.3. Środowisko systemu

Sprzętowe środowisko systemu stanowią dla programu pakiety układu pomiarowego:

Pakiet_RIR interfejsów równoległych posiada dwa układy 8255

6

przeznaczone do komunikacji z układami lokalnymi (pulpit) i urządzeniami zewnętrznymi (komputer, drukarka).

Adresy do komunikacji z pulpitem:

RPA = 90H - port A
RPB = 91H - port B
RPC = 92H - port C
RRP = 93H - programowanie/status

Adresy do komunikacji zewnętrznej

RXA = 98H - port A
RXB = 99H - port B
RXC = 9AH - port C
RXP = 9BH - programowanie/status

Pakiet RIS interfejsu szeregowego posiada układ komunikacyjny 8251 oraz programowany licznik-zegar 8253.

Adresy układu 8251

SID = 84H - zapis/odczyt danych
SIC = 85H - programowanie/status

Adresy układu 8253

SC0 = 80H - licznik 0
SC1 = 81H = licznik 1
SC2 = 82H = licznik 2
SCC = 83H = programowanie

Interfejsy kamer

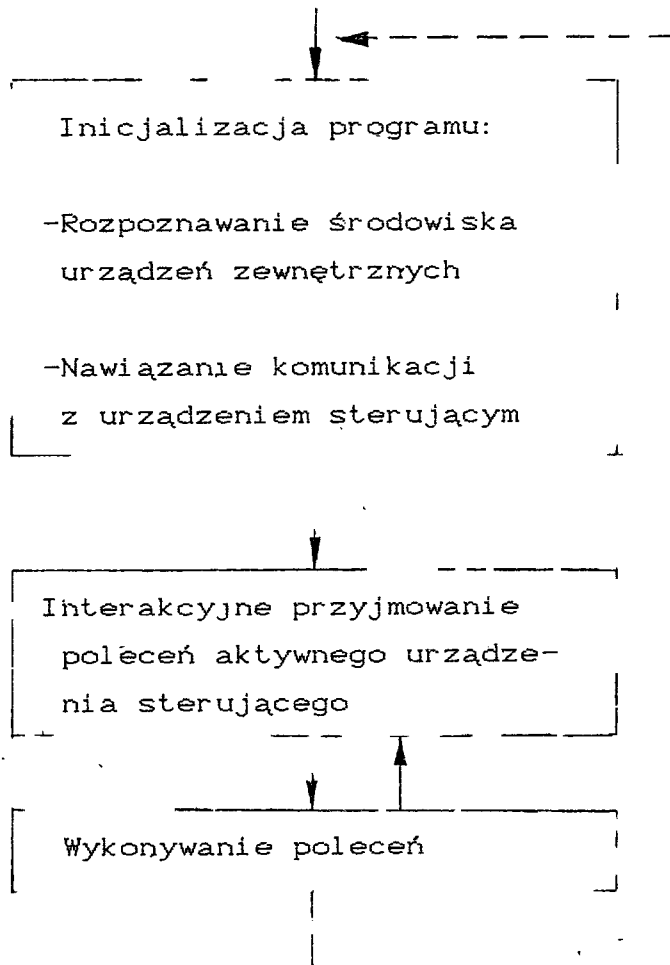
Oprogramowanie układu pomiarowego dopuszcza użycie w konfiguracji do ośmiu wizyjnych torów pomiarowych. Aktualnie zostały zdefiniowane adresy dla trzech interfejsów RIK2 obsługujących kamery typu KL256 lub KL1024.

Adresy interfejsów kamer:

Tor pomiarowy			Funkcja
1	2	3	
0C0H	0D0H	0E0H	START
0C1H	0D1H	0E1H	młodszy bajt danych
0C2H	0D2H	0E2H	starszy bajt danych
0C3H	0D3H	0E3H	STOP

4. Skrócony opis działania programu

Uproszczony algorytm programu przedstawia się następująco



Na bieg programu układu pomiarowego wpływa szereg czynników do których należą:

- rozpoznane środowisko urządzeń zewnętrznych,
- ilość i rodzaj dołączonych urządzeń sterujących (pulpit, komputer) oraz typ interfejsu do którego jest dołączony komputer pracujący jako urządzenie sterujące,
- tryb pracy (ZDALNIE/LOKALNIE) dynamicznie zmieniający się w przypadku dołączenia zewnętrznego urządzenia sterującego,
- bieżący stan programu wynikający z komunikacji z urządzeniem sterującym i wykonywaniem poleceń
- przerwania.

Program odpowiednio modyfikuje stan rejestrów wskaźnikowych oraz korzysta z nich przy sterowaniu swym biegiem.

Rejestratorami wskaźnikowymi są:

MPMS opis stanu układu

RDMPO - wskaźniki odczytywania poleceń

WRMPO - wskaźniki wysyłania poleceń

RKOLEJ - wskaźniki kolejki zadań

INSTNT - wskaźniki poleceń do natychmiastowego wykonania

RQMASK - maska obsługi przerwań

WERINT - wersja obsługi przerwań

Znaczenie bitów tych rejestrów jest opisane w plikach I2,I3,I4 (opis wszystkich zmiennych programu w pliku STALE.YY).

Główne procedury programu są inicjowane przerwaniami :

- pakietów RIK2 obsługi torów kamerowych,
- pakietu RIR interfejsów równoległych,
- pakietu RIS interfejsu szeregowego.

10

Sposób przyjmowania przerw (również priorytety) jest uzależniony od rejestrów wskaźnikowych WERINT, ROMASK i INSTNT, które w dalszej kolejności sterują wykonywaniem podprogramów obsługi przerw.

4.1. Inicjalizacja programu

Inicjalizacja programu jest wykonywana po włączeniu napięć zasilających oraz na żądanie aktywnego urządzenia sterującego.

4.1.1. Rozpoznawanie środowiska urządzeń zewnętrznych

Urządzeniami zewnętrznymi są dla programu urządzenia dołączone do interfejsów pakietów RIR i RIS: pulpit, drukarka, monitor ekranowy i komputer, które pozwalają na stworzenie dowolnej konfiguracji urządzeń zewnętrznych z przedstawionych w tabelicy:

Ad

RX	RP	SC	Numer konfiguracji
P	O	O	1
P	D	O	2
P	O	D	3
P	D	D	4
P/O	K	O	5/9
P/O	O	K	6/10
P/O	K	D	7/11
P/O	D	K	8/12

RX - wewnętrzny interfejs równoległy

RP - zewnętrzny interfejs równoległy

SC - zewnętrzny interfejs szczegółowy V-24

P - pulpit urządzenia

D - drukarka (monitor ekranowy)

K - komputer

O - nie dołączone żadne urządzenie

Podczas inicjalizacji program rozpoznaje w której konfiguracji urządzeń będzie pracował.

4.1.2. Nawiązanie komunikacji z urządzeniem sterującym

Po rozpoznaniu środowisk urządzeń zewnętrznych program oczekuje na nawiązanie komunikacji z urządzeniem sterującym, którym może być pulpit urządzenia albo komputer.

W przypadku stwierdzenia dołączenia komputera (tryb ZDALNIE) program przechodzi do oczekiwania na polecenia od komputera, w przeciwnym przypadku (tryb LOKALNIE) od pulpitu.

10
101

W trybie ZDALNIE aktywnym urządzeniem sterującym układu pomiarowego jest komputer, w trybie LOKALNIE pulpit urządzenia.

4.1.3. Przyjmowanie poleceń

Współpraca programu urządzenia pomiarowego z urządzeniem sterującym polega na wzajemnym przekazywaniu komunikatów, poleceń i danych. W przyjętym protokole komunikacji każda przesyłka rozpoczyna się od kodu FFH (hasło "UWAGA"), po którym następuje kod komunikatu albo polecenia i jego argumenty.

Lista komunikatów i poleceń z pulpitem

NAZWA	Kod komunikatu/polecenia (HEXD)			
	Tryb lokalnie		Tryb zdalnie	
	Odbiór	Nadawanie	Odbiór	Nadawanie
GOTPO	80	-	80	-
NGPO	81	-	80	-
RQRST	88	8c	-	-
KONFK	90	94	-	94
KONFP	91	95	-	95
STRK	98	-	-	-
STRP	99	-	-	-
TROFF	A0	-	-	-
TRONA	A1	-	-	-
TRONB	A2	-	-	-
RQSK	A8	-	A8	-
SK	-	AC	-	AC
RQKPL	B0	-	-	-
RQNK	B8-BF	-	-	-
KOMOK	C0	C4	C0	C4
KOMBL	C1	C5	C1	C5
LOK	D0	D4	-	D4
ZDAL	D1	D5	-	D5
UWAGA	FF	FF	FF	FF
GOTPM	-	84	-	84
NGPM	-	85	-	85
TRBL	-	CC	-	-

Komunikaty i polecenia pulpitu;

GOTPO - komunikat zgłoszenia gotowości pulpitu

NGPO - komunikat zgłoszenia niegotowości pulpitu

RQRST - polecenie programowego restartu

KONFK - komunikat zakończenia przekazywania danych konfi-

167

konfiguracyjnych

- KONFP - komunikat rozpoczynający przekazywanie danych konfiguracyjnych
- STRR - polecenie przesłania kodu zestrojenia
- STRK - komunikat zakończenia przesyłania kodów zestrojenia
- TROFF - polecenie zakończenia przesyłania wyników pomiarów do urządzenia zewnętrznego (drukarki)
- TRONA - polecenie przesyłania wyników pomiarów do urządzenia zewnętrznego dołączonego do interfejsu SC (szeregowego)
- TRONB - polecenie przesyłania wyników pomiarów do urządzenia zewnętrznego dołączonego do interfejsu RP (równoległego)
- RQSK - polecenie przesłania informacji o sprawności torów pomiarowych
- RQKPL - polecenie przesłania wyników pomiarów z wszystkich torów pomiarowych
- RQNK - polecenie przesłania wyników pomiarów z toru pomiarowego NK (NK=0-7)
- KOMOK - komunikat potwierdzający poprawność komunikacji
- KOMBL - komunikat stwierdzający błąd komunikacji
- ZDAL - polecenie przełączenia programu w tryb ZDALNIE
- LOK - polecenie przełączenia programu w tryb LOKALNE
- UWAG A - kod rozpoczynający sekwencję polecenia lub komunikatu

Komunikaty i polecenia układu pomiarowego:

- GOTPM - komunikat zgłoszenia gotowości
- NGPM - komunikat zgłoszenia niegotowości
- TRBL - komunikat zgłaszający błąd w transmisji wyników pomiarów do urządzenia zewnętrznego
- ZDAL - komunikat zgłaszający pracę w trybie ZDALNIE

LOK - komunikat zgłaszający pracę w trybie LOKALNIE
KOMOK - komunikat potwierdzający poprawność komunikacji
KOMBL - komunikat stwierdzający błąd komunikacji
SK - komunikat przekazujący Informację o sprawności torów pomiarowych

KONFP - komunikat przekazyjący dane konfiguracyjne
KONFK - komunikat zamykający przekazywanie danych konfiguracyjnych

Opisane tutaj oprogramowanie użytkowe umożliwia współpracę urządzenia z monitorem ekranowym, z komputerem, drukarką oraz z przewidywanym w przyszłości dla wizyjnego urządzenia pomiarowego mikroprocesorowym pulpitem sterującym.


```
0001 0000+C  
0002 ;*****;  
0003 ; s.548 ; System do pomiarow wizyjnych ;  
0004 ; ver.1 ; Obsluga operatorska z pulpitu urzadzenia albo przez komputer PC ;  
0005 ; ; dolaczony do interfejsu szeregowego / rownoleglego (8255) ;  
0006 ;*****;  
0007 0000+C  
0008 0000+C  
0038 endif  
0039 includ 'macro.vv'  
0040 ;macro.vv  
0041 ;Makroinstrukcje  
0042 M movA macro Re  
0043 M mov R1,# Re  
0044 M mov A,@R1  
0045 endm  
0046 0000+C  
0047 M movA0 macro Re  
0048 M mov R0,# Re  
0049 M mov A,@R0  
0050 endm  
0051 0000+C  
0052 M movAE macro Ad  
0053 M mov R1,# Ad  
0054 M movX A,@R1  
0055 endm  
0056 0000+C  
0057 M movE macro Ad  
0058 M mov R1,# Ad  
0059 M movX @R1,A  
0060 endm  
0061 0000+C  
0062 M movR macro Re  
0063 M mov R1,# Re  
0064 M mov @R1,A  
0065 endm  
0066 0000+C  
0067 M movR0 macro Re  
0068 M mov R0,# Re  
0069 M mov @R0,A  
0070 endm  
0071 0000+C  
0072 ;movRD macro Re,#Da  
0073 ; mov R1,# Re  
0074 ; mov @R1,#Da  
0075 ; endm  
0076 0000+C  
0077 M decR macro Re  
0078 M mov R1,# Re  
0079 M xch A,@R1  
0080 M dec A  
0081 M xch A,@R1  
0082 endm  
0083 0000+C  
0084 M incR macro Re  
0085 M mov R1,# Re  
0086 M inc @R1  
0087 endm  
0088 0000+C  
0089 ;BITSR macro BITNOS, SETRES
```

```
0090 ; mov A,#(2*BITNOS+SETRES)
0091 ; movX ER1,A
0092 ; endm
0093 0000+C
0094 M OUTp1 macro VAL
0095 M mov A,# VAL
0096 M outl P1,A
0097 M endm
0098 0000+C
0099 M OUTp2 macro VAL
0100 M mov A,# VAL
0101 M outl P2,A
0102 M endm
0103 0000+C
0104 M addAR macro Re
0105 M mov R1,#Re
0106 M add A,ER1
0107 M endm
0108 M
```

Input: S:548

Module: ".MAIN."

```

0110      endif
0111      includ 'STALE.VV'
0112      ;stale.vv
0113      ;maski dla wariantowej obsługi przerwan INT
0114 E      00FD      MASKA1 = 1111101B ;dla stanów LDS, ZDAS, ZNAS
0115 E      00BE      MASKA2 = 1011110B ;transmisja RIR
0116 E      00FE      MASKA3 = 1111110B ;dla stanów LOR, ZDAR, ZNAR
0117 E      00BD      MASKA4 = 1011110B ;transmisja RIS
0118 E      0000      MASKA5 = 0
0119 E      0000      MASKA6 = 0
0120 E      0000      MASKA7 = 0
0121 E      0000      MASKA8 = 0
0122      0000+C
0123      ;adresy urzadzen RJC
0124 E      0020      RESA = 32 ;RESOUT adr
0125 E      0020      SWA = 32 ;SW/RJC adr
0126 E      0024      RQA = 36 ;RQR adr
0127      0000+C
0128      ;TCB = 40 ;8253/RJC base addr
0129 E      0028      TC0 = 40 ; = CNT0
0130 E      0029      TC1 = 41 ;+1 = CNT1
0131 E      002A      TC2 = 42 ;+2 = CNT2
0132 E      002B      TCC = 43 ;+3 = Control
0133      0000+C
0134      ;Dane programujace TC (w procedurze PRG53: CW zawsze dla cnt# 0)
0135 E      0036      TC0CW = 0164 + 40 + 3*2 + 0 ;CW: cnt#0 + Both + Mode + BCD, Mode# 3
0136 E      0034      TC1CW = 0164 + 40 + 2*2 + 0 ;CW: cnt#0 + Both + Mode + BCD, Mode# 2
0137 E      0034      TC2CW = 0164 + 40 + 2*2 + 0 ;CW: cnt#0 + Both + Mode + BCD, Mode# 2
0138      0000+C
0139      0000+C      ;Wspolczynniki podzialu TC0, TC1, TC2
0140      0000+C      ;cnt# 0 dzieli OCNT0 = 80us przez 100, co daje okres OCNT0 = 80us
0141 E      0064      TC0LoD = 100
0142 E      0000      TC0HiD = 0
0143      0000+C      ;cnt# 1 dzieli OCNT0 = 80us przez 25, co daje okres OCNT1 = 2ms
0144 E      0019      TC1LoD = 25
0145 E      0000      TC1HiD = 0
0146      0000+C      ;cnt# 2 dzieli OCNT0 = 80us przez 25, co daje okres OCNT2 = 2ms
0147 E      0019      TC2LoD = 25
0148 E      0000      TC2HiD = 0
0149      0000+C
0150      ;adresy pakietow interfejsow szeregowych RIS #0 i #1
0151      ;SCB = 80H ;8253/RIS #0 base addr
0152 E      0080      SC0 = 80H ; = CNT0
0153 E      0081      SC1 = 81H ; +1 = CNT1
0154 E      0082      SC2 = 82H ; +2 = CNT2
0155 E      0083      SCC = 83H ; +3 = Control
0156      0000+C
0157      ;SIB = 84H ;8251/RIS #0 base addr
0158 E      0084      SID = 84H ; (86H) data
0159 E      0085      SIC = 85H ; (87H) control/status
0160      0000+C
0161      ;adresy pakietow interfejsow rownoleglych:
0162      ; RIR #0
0163 E      0090      RPA = 90H ;PA
0164 E      0091      RPB = 91H ;PB
0165 E      0092      RPC = 92H ;PC
0166 E      0093      RRP = 93H ;Control
0167      0000+C
0168 E      0098      RXA = 98H ;PA
0169      0000+C      ;port wyjsciowy komunikacji z PO

```

```

0170 E 0099 RXB = 99H ;PB*
0171 0000+C ;port wejsciowy komunikacji z PD
0172 E 009A RYC = 9AH ;PC
0173 E 009B RXP = 9BH ;Control
0174 0000+C
0175 ;-----
0176 ;State do programowania 8255
0177 ;Mode# 0
0178 E 009B M0IALL = 10011011B ;PA-I, PB-I, PC-I
0179 0000+C
0180 ;Mode# 1
0181 E 00A6 M10IO = 10100110B ;PA-0, PB-I, PC(4,5)-0
0182 E 00AE M10II = 10101110B ; PC(4,5)-I
0183 0000+C
0184 E 00B4 M1100 = 10110100B ;PA-I, PB-0, PC(6,7)-0
0185 E 00BC M110I = 10111100B ; PC(6,7)-I
0186 0000+C
0187 E 00A4 M1000 = 10100000B or 10000100B ;PA-0 or PB-0, PC(4,5)-0
0188 E 00AC M100I = 10101000B or 10000100B ; PC(4,5)-I
0189 0000+C
0190 E 00B6 M1110 = 10110000B or 10000110B ;PA-I or PB-I, PC(6,7)-0
0191 E 00BE M111I = 10111000B or 10000110B ; PC(6,7)-I
0192 0000+C
0193 ;INTEA - I eq.to PC4
0194 ;INTEA - 0 eq.to PC6
0195 ;INTEB - I/O eq.to PC2
0196 0000+C
0197 ;Mode 1 status word format
0198 ;Bit 7 6 5 4 3 2 1 0
0199 ;Input conf. I/O I/O IBFA INTEA INTRA ; INTEB IBFB INTRB
0200 ;Output conf. -OBFA INTEA I/O I/O INTRA ; INTEB -OBFB INTRB
0201 ; !<----- PORT A ----->!<--- PORT B ---->!
0202 0000+C
0203 ;-----

```

```
0204 0000+C  
0205 ;adresy pakietow RIK2 (wykonanie 08)  
0206 ;(adresy: C0H...CFH, D0H...DFH, E0H...EFH, F0H...FFH)  
0207 E 0003 LTK = 3 ;liczba torow kamerowych  
0208 E 00C0 KX20 = 0C0H ;RIK2 #1 - start  
0209 ;KX21 = 0C1H ; dane Lo  
0210 ;KX22 = 0C2H ; dane Hi, b7= flaga danych  
0211 ;KX23 = 0C3H ; - STOP, b7= flaga alarmu  
0212 0000+C  
0213 E 00D0 KY20 = 0D0H ;RIK2 #2 - start  
0214 ;KY21 = 0D1H ; dane Lo  
0215 ;KY22 = 0D2H ; dane Hi, b7= flaga danych  
0216 ;KY23 = 0D3H ; -STOP; b7= flaga alarmu  
0217 0000+C  
0218 E 00E0 KZ20 = 0E0H ;RIK2 #3 - start  
0219 ;KZ21 = 0E1H ; dane Lo  
0220 ;KZ22 = 0E2H ; dane Hi, b7= flaga danych  
0221 ;KZ23 = 0E3H ; -STOP; b7= flaga alarmu  
0222  
-----  
0223 0000+C
```

```

0224 0000+C
0225 ;rejstry procesora
0226 E 007F RERR = 127 ;rejestr błędów
0227 E 007E RKOLEJ = 126 ;zadania do wykonania
0228 E 007D XRAMPG = 125 ;strona XRAM dla przesyłek
0229 E 007C PNTDAT = 124 ;wskaznik bajtu XRAM na stronie
0230 E 007B PNTARG = 123 ;wskaznik argumentu
0231 E 007A RQMASK = 122 ;maska przerwan dla wersji MERINT
0232 E 0079 RQTMP = 121 ;(RQA), modyfikowany podczas obsługi przerwan
0233 E 0078 MERINT = 120 ;wersja obsługi przerwan INT
0234 E 0077 ACCSTO = 119 ;akumulator na czas obsługi przerwan
0235 ; = 118 ; ...
0236 ; = 117 ; ...
0237 E 0074 PNTIO = 116 ;adres IO/XRAM dla przesyłek
0238 E 0073 SUMA = 115 ;sum kontrolna
0239 E 0072 ARG01 = 114 ;argument do wysłania
0240 E 0071 ARG02 = 113 ;argument do wysłania
0241 E 0071 NK = 113 ;numer kanału wg poleceń RSTRP, RRQNK
0242 E 0070 KODD = 112 ;kod polecenia MPM
0243 E 006F LARGO = 111 ;licznik argumentów / danych
0244 E 006E K4 = 110 ;b(3-0):= KODI(6-3)
0245 E 006D A3 = 109 ;b(2-0):= KODI(2-0)
0246 E 006C KODI = 108 ;odebrany kod polecenia
0247 E 006B PRNS = 107 ;stan drukarki
0248 E 006A RDMPO = 106 ;wskazniki odbierania poleceń
0249 E 0069 WRMPO = 105 ;wskazniki wysyłania poleceń
0250 E 0068 MPMS = 104 ;opis stanu MPM
0251 E 0067 KS = 103 ;stan torów kamerowych
0252 E 0066 INSTNT = 102 ;lista poleceń do natychmiastowego wykonania
0253 E 0065 DIVPNT = 101 ;wskaznik danych programujących TC dla kamer
0254 E 0064 CIBTMP = 100 ;bajt sterujący dla B251
0255 E 0063 P2TMP = 99 ;P2 na czas dostępu do IO / XRAM
0256 E 0062 P1TMP = 98 ;przechowywany stan PI
0257 E 0061 KONTMP = 97 ;KON przechowywany podczas wysyłania danych
0258 E 0060 MSCTMP = 96 ;
0259 0000+C
0260 ; = 95 ;
0261 ; = 94 ;
0262 ; = 93 ;
0263 ; = 92 ;
0264 ; = 91 ;
0265 0000+C
0266 ; = 90 ;BUFORD + 8, koniec bufora transmisji
0267 ; = 89 ; ...
0268 ; = 88 ; ...
0269 ; = 87 ; ...
0270 ; = 86 ; ...
0271 ; = 85 ; ...
0272 ; = 84 ; ...
0273 ; = 83 ; ...
0274 E 0052 BUFORD = 82 ;początek bufora transmisji
0275 0000+C
0276 ;Dane konfiguracyjne dla torów kamerowych A - H:
0277 ;KPx0 = pierwsze przejście, KPxi = drugie przejście
0278 ;KPx0 = KPA0 + 2 * (kanal#)
0279 ;KPx1 = KPA0 + 2 * (kanal#) + 1
0280 ;KPH1 = 81 ;
0281 ;KPH0 = 80 ;
0282 ;KPG1 = 79 ;
0283 ;KPG0 = 78 ;

```

0284		;KPF1	=	77	;
0285		;KPF0	=	76	;
0286		;KPE1	=	75	;
0287		;KPE0	=	74	;
0288		;KPD1	=	73	;
0289		;KPD0	=	72	;
0290		;KPC1	=	71	;
0291		;KPC0	=	70	;
0292		;KPB1	=	69	;
0293		;KPB0	=	68	;
0294		;KPA1	=	67	;
0295 E	0042	KPA0	=	66	;
0296 E	0041	KT	=	65	;wskazniki typow kamer: 0 = KL256, 1 = KL1024
0297 E	0040	KON	=	64	;wskazniki wlaczonych torow kamerowych
0298	0000+C				;bity 7 - 0 odpowiadaja torom H - A

```

0299 0000+C
0300 0000+C
0301 ;rejestry procesora (cd)
0302 0000+C
0303 ;wartosci przejsc odczytane z torow kamerowych A - H
0304 ;Kn1Hi = KA1Hi + 4 * (kanal#) + 3
0305 ;Kn1Lo = KA1Lo + 4 * (kanal#) + 2
0306 ;Kn0Hi = KA0Hi + 4 * (kanal#) + 1
0307 ;Kn0Lo = KA0Lo + 4 * (kanal#)
0308 ;KH1Hi = 63
0309 ;KH1Lo = 62
0310 ;KH0Hi = 61
0311 ;KH0Lo = 60
0312 ;KG1Hi = 59
0313 ;KG1Lo = 58
0314 ;KG0Hi = 57
0315 ;KG0Lo = 56
0316 ;KF1Hi = 55
0317 ;KF1Lo = 54
0318 ;KF0Hi = 53
0319 ;KF0Lo = 52
0320 ;KE1Hi = 51
0321 ;KE1Lo = 50
0322 ;KE0Hi = 49
0323 ;KE0Lo = 48
0324 ;KD1Hi = 47
0325 ;KD1Lo = 46
0326 ;KD0Hi = 45
0327 ;KD0Lo = 44
0328 ;KC1Hi = 43
0329 ;KC1Lo = 42
0330 ;KC0Hi = 41
0331 ;KC0Lo = 40
0332 ;KB1Hi = 39
0333 ;KB1Lo = 38
0334 ;KB0Hi = 37
0335 ;KB0Lo = 36
0336 ;KA1Hi = 35
0337 ;KA1Lo = 34
0338 ;KA0Hi = 33
0339 E 0020 KA0Lo = 32
0340 ;
0341 ;RB1
0342 ; = 31 ;R7
0343 ; = 30 ;R6
0344 ; = 29 ;R5
0345 ; = 28 ;R4
0346 ; = 27 ;R3
0347 ; = 26 ;R2
0348 ; = 25 ;R1
0349 ; = 24 ;R0
0350 ;
0351 ;RB0
0352 ; = 7 ;R7
0353 ; = 6 ;R6
0354 ; = 5 ;R5
0355 ; = 4 ;R4
0356 ; = 3 ;R3
0357 ; = 2 ;R2
0358 ; = 1 ;R1

```


0359

; = 0 ;R0

0360

```

0361 0000+C
0362 ;Kody polecen MPD, MPM: b7= 1, b6-3= K4, b2-0= A3
0363 ;NK=0XXX -- numer toru kamerowego
0364 ;TK= 0001 dla KL256 - typ kamery
0365 ; 0010 dla KL1024
0366 ;P0, P1= XXXX - numer przejścia
0367 ;NP = 000X / 0010 - numer parametru
0368 ;B0, B1 - argumenty polecenia
0369 0000+C ;KODI: K4: A3: B0: B1:
0370 E 0080 RGOTPO = 80H ; 0 0
0371 E 0081 RRGPO = 81H ; 0 1
0372 E 0088 RRQRST = 88H ; 1 0
0373 E 0090 RKONFK = 90H ; 2 0
0374 E 0091 RKONFP = 91H ; 2 1 NK,TK P0,P1
0375 E 0098 RSTRK = 98H ; 3 0
0376 E 0099 RSTRP = 99H ; 3 1 NK,NP B0/RSTRK
0377 E 00A0 RTROFF = 0A0H ; 4 0
0378 E 00A1 RTRONA = 0A1H ; 4 1
0379 E 00A2 RTRONB = 0A2H ; 4 2
0380 E 00AB RRQSK = 0ABH ; 5 0
0381 E 00B0 RRQKPL = 0B0H ; 6 0
0382 ;RRQNK = 0BBH + NK ; 7 NK
0383 E 00C0 RKOMOK = 0C0H ; 8 0
0384 E 00C1 RKOMBL = 0C1H ; 8 1
0385 E 00D0 RLOK = 0D0H ; 10 0
0386 E 00D1 RZDAL = 0D1H ; 10 1
0387 E 00FF UWAGA = 0FFH ; 15 7
0388 0000+C ;KODO:
0389 E 0084 TGOTPM = 84H ; 0 4
0390 E 0085 TNGPM = 85H ; 0 5
0391 E 008C TRQRST = 8CH ; 1 4
0392 E 0094 TKONFK = 94H ; 2 4
0393 E 0095 TKONFP = 95H ; 2 5
0394 E 00AC TSK = 0ACH ; 5 4
0395 E 00C4 TKOMOK = 0C4H ; 8 4
0396 E 00C5 TKOMBL = 0C5H ; 8 5
0397 E 00CC TTRBL = 0CCH ; 9 4
0398 E 00D4 TLOK = 0D4H ; 10 4
0399 E 00D5 TZDAL = 0D5H ; 10 5
0400 E 00FF UWAGA = 0FFH ; 15 7
0401 0000+C
0402 title 'VV ver.0 - MB0'

```


Input: S.548

Module: ".MAIN."

```

0474 0038+C 72 DB+C          jb3   low 3$          ;konfigurowanie
0475 003A+C 52 E3+C          jb2   low 2$          ;podac stan kanalow
0476 003C+C 32 E8+C          jb1   low 1$          ;podac kod zestrojenia
0477 003E+C 12 ED+C          jb0   low 0$          ;transmisja
0478 0040+C                ;nie ma zadnych zadan, powrot do petli przerwan
0479 0040+C 04 C9+C          jmp   00$
0480 0042+C                7$   ;transmisja do komputera
0481 0042+C                outP1 33              ;nie realizowane
0484 0045+C E4 55+C          jmp   CLOOP
0485 0047+C                6$   ;komunikacja z komputerem
0486 0047+C                outP1 34              ;nie realizowane
0489 004A+C E4 55+C          jmp   CLOOP
0490 004C+C                5$   ;do wykonania polecenie natychmiastowe:
0491 004C+C                movA  INSTNT
0494 004F+C C6 D1+C          jz    low 57$
0495 0051+C 37                cpl   A
0496 0052+C F2 61+C          jb7   low 56$
0497 0054+C                ;INSTNT(7)= 1
0498 0054+C 04 61+C          jmp   56$
0499 0056+C 00                nop
0500 0057+C 00                nop
0501 0058+C                movA  INSTNT
0504 0058+C 53 7F          anl   A,#01111111B   ;INSTNT(7):= 0
0505 005D+C A1                mov   @R1,A
0506 005E+C C6 C1+C          jz    low 58$
0507 0060+C 37                cpl   A
0508 0061+C D2 70+C          56$  jb6   low 55$
0509 0063+C                ;INSTNT(6)= 1
0510 0063+C 04 70+C          jmp   55$
0511 0065+C 00                nop
0512 0066+C 00                nop
0513 0067+C                movA  INSTNT
0516 006A+C 53 BF          anl   A,#10111111B   ;INSTNT(6):= 0
0517 006C+C A1                mov   @R1,A
0518 006D+C C6 C1+C          jz    low 58$
0519 006F+C 37                cpl   A
0520 0070+C B2 81+C          55$  jb5   low 54$
0521 0072+C                ;INSTNT(5)= 1, do wykonania IRD
0522 0072+C F5                set   MB1
0523 0073+C 74 00+D          call  IRD
0524 0075+C E5                set   MB0
0525                if um = 1
0526 0076+C D4 00+C          call  DISPRR          ;wyswietlenie rejestrow roboczych
0527                ;P1 MS WF RF RK IF PF KF KT WI K4 A3 KI SU
0528                ;
0529                else
0533 0078+C                movA  INSTNT
0536 007B+C 53 DF          anl   A,#11011111B   ;INSTNT(5):= 0
0537 007D+C A1                mov   @R1,A
0538 007E+C C6 C1+C          jz    low 58$
0539 0080+C 37                cpl   A
0540 0081+C 92 8F+C          54$  jb4   low 53$
0541 0083+C                ;INSTNT(4)= 1, do wykonania INR
0542 0083+C F5                set   MB1
0543 0084+C 54 00+D          call  INR
0544 0086+C E5                set   MB0
0545                if um = 1
0546 0087+C D4 00+C          call  DISPRR          ;wyswietlenie rejestrow roboczych
0547                ;P1 MS WF RF RK IF PF KF KT WI K4 A3 KI SU
0548                ;
0549                else

```

```

0553 0089+C          movA  INSTNT
0556 008C+C          ;INSTNT(4)=0;np;jesli odebrany wyslany bajt
0557 008C+C  C6 C1+C    jz    low 58$
0558 008E+C  37        cpl   A
0559 008F+C  72 9E+C    53$    jb3   low 52$
0560 0091+C          ;INSTNT(3)=1
0561 0091+C  04 9E+C    jmp   52$
0562 0093+C  00        nop
0563 0094+C  00        nop
0564 0095+C          movA  INSTNT
0567 0098+C  53 F7      anl   A,# 11110111B ;INSTNT(3)=0
0568 009A+C  A1        mov   @R1,A
0569 009B+C  C6 C1+C    jz    low 58$
0570 009D+C  37        cpl   A
0571 009E+C  52 A2+C    52$    jb2   low 51$
0572 00A0+C          ;INSTNT(2)=1, odczytac dane z torow kamerowych - wskaznik dla VRKNF
0573 00A0+C  04 A2+C    jmp   51$
0574          ;
0575          ;
0576          ;
0577          ;
0578          ;
0579          ;
0580          ;
0581 00A2+C  32 B2+C    51$    jb1   low 50$
0582 00A4+C          ;INSTNT(1)=1, przygotowac dane do wydruku
0583 00A4+C  37        cpl   A
0584 00A5+C  52 CF+C    jb2   low 81$ ;INSTNT(2)=1, nie ma wynikow
0585 00A7+C  54 D6+C    call  BCDALL
0586 00A9+C          movA  INSTNT
0589 00AC+C  53 FD      anl   A,# 11111011B ;INSTNT(1)=0
0590 00AE+C  A1        mov   @R1,A
0591 00AF+C  C6 C1+C    jz    low 58$
0592 00B1+C  37        cpl   A
0593 00B2+C  12 B9+C    50$    jb0   low 82$
0594 00B4+C          ;INSTNT(0)=1, przeslac dane do PO
0595 00B4+C  37        cpl   A
0596 00B5+C  52 CF+C    jb2   low 81$ ;INSTNT(2)=1, nie ma wynikow
0597 00B7+C  B4 64+C    call  SMDATA
0598 00B9+C  82$    movA  INSTNT
0601 00BC+C  53 FE      anl   A,# 11111110B ;INSTNT(0)=0
0602 00BE+C  A1        mov   @R1,A
0603 00BF+C  96 CF+C    83$    jnz   low 81$
0604          ;
0605 00C1+C  58$    movA  RKOLEJ
0608 00C4+C  53 DF      anl   A,# 11011111B ;RKOLEJ(5)=0
0609 00C6+C  A1        mov   @R1,A
0610 00C7+C  96 36+C    jnz   low 59$
0611 00C9+C  80$    movA  RDMP0
0614 00CC+C  53 7F      anl   A,# 01111111B ;RDMP0(7)=0;czytac
0615 00CE+C  A1        mov   @R1,A
0616 00CF+C  04 22+C    81$    jmp   ENINT
0617          ;
0618 00D1+C  57$    outP1 35
0621 00D4+C  E4 55+C    jmp   CLOOP
0622 00D6+C  4$    ;komunikaty od MP0
0623 00D6+C          outP1 36
0626 00D9+C  E4 55+C    jmp   CLOOP
0627 00DB+C  3$    ;konfigurowanie
0628 00DB+C  B9 6A      mov   R1,# RDMP0
0629 00DD+C  23 3F      mov   A,# 00111111B

```

```
0630 00DF+C 51          ani    A,@R1
0631 00E0+C A1          mov    @R1,A
0632 00E1+C 04 22+C     jmp    ENINT
0633 00E3+C           2$    ;podaj stan torow pomiarowych
0634 00E3+C           outP1 38
0637 00E6+C E4 55+C     jmp    CLOOP
0638 00E8+C           1$    ;podaj kod zestrojenia
0639 00E8+C           outP1 39
0642 00EB+C E4 55+C     jmp    CLOOP
0643 00ED+C           0$    ;transmisja danych
0644 00ED+C           outP1 40
0647 00F0+C E4 55+C     jmp    CLOOP
0648 00F2+C           90$   ;blad w wykonywaniu polecen natychmiastowych
0649 00F2+C           outP1 41
0652 00F5+C E4 55+C     jmp    CLOOP
0653                                     ;-----
```

Input: S.548

Module: ".MAIN."

```

0654 @0F7+C
0655 ;pg.1 MB: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
0656 ;=====
0657 @0F7+C org 100H ; MBANK #0
0658 ;-----
0659 @100+C RUNIK2 ;start torow kamerowych wg. danych konfiguracyjnych
0660 @100+C
0661 @100+C F4 17+C call CSETID ;selIO
0662 @102+C BF FE mov R7,# 254 ;maska do sprawdzania konfiguracji
0663 @104+C BE 00 mov R6,# 0 ;numer kanalu
0664 @106+C movA KDN
0667 @109+C C6 0F+C jz low 6$
0668 if wpo = 1
0723 @10B+C
0724 if wpo = 2
0725 ;wpo = 2, wersja dla konfiguracji dowolnej
0726 @10B+C 24 12+C jmp 1$
0727 @10D+C F6 12+C 4$ jC low 1$
0728 @10F+C ;C = 0, sprawdzono 8 kanalow
0729 @10F+C F4 10+C 6$ call CRSTP2 ;restore P2
0730 @111+C B3 ret
0731 @112+C
0732 @112+C 1$ ;wyznaczenie wlaczonego kanalu
0733 @112+C movA KDN
0736 @115+C 4F orl A,R7
0737 @116+C 37 cpl A
0738 @117+C 96 22+C jnz low 30$ ;ten kanal (R6) nie nalezy do konfiguracji
0739 @119+C 2$ ;ten kanal (R6) nalezy do konfiguracji
0740 @119+C FE mov A,R6
0741 @11A+C 6B add A,R3
0742 @11B+C E3 movP3 A,@A
0743 @11C+C C6 0F+C jz low 6$ ;adres poza lista TARIK2
0744 @11E+C AB mov R0,A ;adres START wlaczonego kanalu
0745 @11F+C 23 01 mov B,# 1
0746 @121+C 90 movX @R0,A ;START
0747 @122+C
0748 @122+C FF 30$ mov A,R7
0749 @123+C 97 clr C
0750 @124+C A7 cpl C
0751 @125+C F7 rlc A
0752 @126+C AF mov R7,A ;maska
0753 @127+C 1E inc R6 ;dla nastepnego kanalu
0754 @128+C 24 0D+C jmp 4$
0755 @12A+C
0756 endif
0757 @12A+C
0758 ;-----
0759 @12A+C RDIK2 ;odczyt przejsc 0 i 1 wg danych konfiguracyjnych
0760 @12A+C
0761 @12A+C F4 17+C call CSETID ;selIO
0762 @12C+C BF FE mov R7,# 254 ;maska do sprawdzania konfiguracji
0763 @12E+C BE 00 mov R6,# 0 ;numer kanalu
0764 if wpo = 1
0833 @130+C
0834 if wpo = 2
0835 ;wpo = 2, wersja dla konfiguracji dowolnej
0836 @130+C 24 37+C jmp 1$
0837 @132+C F6 37+C 4$ jC low 1$
0838 @134+C ;C = 0, sprawdzono 8 kanalow
0839 @134+C F4 10+C 6$ call CRSTP2 ;restore P2

```

```

0840 0136+C 83          ret
0841 0137+C
0842 0137+C          1$   ;wyznaczenie wlaczonego kanalu
0843 0137+C          mov   KON
0846 013A+C 4F          orl   A,R7
0847 013B+C 37          cpl  A
0848 013C+C C6 46+C    jZ   low 2$
0849 013E+C          ;ten kanal (R6) nie nalezy do konfiguracji
0850 013E+C
0851 013E+C FF          30$   mov   A,R7
0852 013F+C 97          clr  C
0853 0140+C A7          cpl  C
0854 0141+C F7          rlc  A
0855 0142+C AF          mov  R7,A          ;maska
0856 0143+C 1E          inc  R6          ;dla nastepnego kanalu
0857 0144+C 24 32+C    jmp  4$
0858
0859 0146+C
0860 0146+C          2$   ;ten kanal (R6) nalezy do konfiguracji
0861 0146+C FE          mov  A,R6
0862 0147+C E7          rl   A
0863 0148+C 03 42      add  A,# KPA0
0864 014A+C A8          mov  R0,A
0865 014B+C F0          mov  A,@R0
0866 014C+C AD          mov  R5,A          ;przejscie #(KPN0)
0867 014D+C 18          inc  R0
0868 014E+C F0          mov  A,@R0
0869 014F+C AC          mov  R4,A          ;przejscie #(KPN1)
0870 0150+C          ;wyznaczenie adresu odczytu danych Lo przejscia kanalu #(R6)
0871 0150+C FE          mov  A,R6
0872 0151+C 00          nop
0873 0152+C 6B          add  A,R3          ;w R3 przekazany do procedury
0874 0153+C          ;wskaznik do tablicy TARIK2
0875 0153+C E3          movP3 A,@A
0876 0154+C C6 34+C    jz   low 6$          ;adres poza lista TARIK2
0877 0156+C A9          mov  R1,A          ;adres RIK2 odczytu danych to
0878          endif
0879 0157+C          ;wyznaczenie adresu bazowego rejestrow do skladowania danych kanalu #(R6)
0880 0157+C FE          mov  A,R6
0881 0158+C E7          rl   A
0882 0159+C E7          rl   A
0883 015A+C 03 20      add  A,# KA0Lo
0884 015C+C A8          mov  R0,A          ;R0:= Kn0Lo, adres dla danych Lo
0885 015D+C          ;przejscia #0 kanalu #(R6)
0886 015D+C
0887 015D+C          ;poszukowanie danych przejscia 0
0888 015D+C 34 9A+C    10$   call 50$
0889 015F+C F2 76+C    jb7  low 15$          ;sa dalsze dane do odczytania
0890 0161+C FD          mov  A,R5          ;nie ma dalszych danych do odczytania
0891 0162+C C6 7D+C    jZ   low 20$          ;to sa dane przejscia 0
0892 0164+C          ;to nie jest przejscie 0 i nie ma dalszych danych
0893 0164+C          ;sygnalizacja bledu NIE MA PRZEJSC 0 i 1
0894 0164+C 23 FF      11$   mov  A,# 255
0895 0166+C A0          mov  @R0,A
0896 0167+C 18          inc  R0
0897 0168+C 23 FF      mov  A,# 255
0898 016A+C A0          mov  @R0,A
0899 016B+C 18          inc  R0
0900 016C+C 23 FF      12$   mov  A,# 255
0901 016E+C A0          mov  @R0,A
0902 016F+C 18          inc  R0

```


Input: S.548

Module: ".MAIN."

```

0903 0170+C 23 FF          mov    A,# 255
0904 0172+C A0          mov    @R0,A
0905 0173+C 18          inc    R0
0906 0174+C 24 3E+C     jmp    30$
0907 0176+C
0908 0176+C          15$    ;sa dalsze dane do odczytania
0909 0176+C FD          mov    A,R5
0910 0177+C C6 7D+C     jZ     low 20$          ;to sa dane przejścia 0
0911 0179+C CD          dec    R5          ;to jeszcze nie jest przejście 0
0912 017A+C CC          dec    R4
0913 017B+C 24 5D+C     jmp    10$
0914 017D+C
0915 017D+C          20$    ;poszukiwanie danych przejścia 1
0916 017D+C          ;przepisać dane przejścia 0 do danych przejścia 1
0917 017D+C F0          mov    A,@R0
0918 017E+C AA          mov    R2,A
0919 017F+C 18          inc    R0
0920 0180+C F0          mov    A,@R0
0921 0181+C 2A          xch   A,R2
0922 0182+C 18          inc    R0
0923 0183+C A0          mov    @R0,A
0924 0184+C 2A          xch   A,R2
0925 0185+C 18          inc    R0
0926 0186+C A0          mov    @R0,A
0927 0187+C CB          dec    R0
0928 0188+C 00          nop
0929 0189+C 24 94+C     jmp    25$
0930 018B+C
0931 018B+C 34 9A+C     21$    call   50$
0932 018D+C F2 94+C     jb7   low 25$          ;sa dalsze dane do odczytania
0933 018F+C FC          mov    A,R4          ;nie ma dalszych danych
0934 0190+C C6 3E+C     jZ     low 30$          ;to sa dane przejścia 1
0935 0192+C          ;to nie jest przejście 1 i nie ma dalszych danych
0936 0192+C          ;sygnalizacja błędu NIE MA PRZEJŚCIA 1
0937 0192+C 24 6C+C     jmp    12$
0938 0194+C
0939 0194+C          25$    ;sa dalsze dane do odczytania
0940 0194+C FC          mov    A,R4
0941 0195+C C6 3E+C     jZ     low 30$          ;to sa dane przejścia 1
0942 0197+C CC          dec    R4          ;to jeszcze nie jest przejście 1
0943 0198+C 24 8B+C     jmp    21$
0944
;.....
0945 019A+C          ;odczyt danych Lo, Hi według adresu (R1) i zapamiętanie pod adresem wg (R0)
0946          if um = 0
0958          if um = 1
0959 019A+C          50$
0960 019A+C 28          xch   A,R0
0961 019B+C 39          outL  P1,A          ;wyswietlenie adresu dla danych na P1
0962 019C+C 28          xch   A,R0
0963 019D+C 29          xch   A,R1
0964 019E+C 39          outL  P1,A          ;wyswietlenie adresu RIK2 na P1
0965 019F+C 29          xch   A,R1
0966 01A0+C 81          movX  A,@R1
0967 01A1+C 39          outL  P1,A          ;wyswietlenie odczytanych danych na P1
0968 01A2+C A0          mov    @R0,A          ;dane Lo
0969 01A3+C 19          inc    R1
0970 01A4+C 18          inc    R0
0971 01A5+C 81          movX  A,@R1
0972 01A6+C 39          outL  P1,A          ;wyswietlenie odczytanych danych na P1
0973 01A7+C 53 83       andl  A,# 10000011B   ;acc(6-2) = 0
0974 01A9+C A0          mov    @R0,A          ;dane Hi

```

```
0975  #1AA+C  CY          dec  K1  
0976  #1AB+C  C8        <dec> R0  
0977  #1AC+C  B5        ret  
0978                                     endif  
0979                                     ;-----
```



```

1040 023A+C ;SACC = R6
1041 023A+C ;TTS = R5
1042 023A+C ;FDJ = R4
1043 023A+C ;DTS = R3
1044 023A+C ;DDJ = R2
1045 023A+C 27 ;clr A
1046 023B+C AC ;mov R4,A ;clr FDJ
1047 023C+C AD ;mov R5,A ;clr TTS
1048 023D+C AA ;mov R2,A ;clr DDJ
1049 023E+C AB ;mov R3,A ;clr DTS
1050 023F+C F1 ;mov A,@R1 ;get BCD
1051 0240+C F2 72+C ;jb7 low 1% ;dodac:100 do D
1052 0242+C ;BLo < 120
1053 0242+C 53 7F 2% ;anl A,#7FH
1054 0244+C 57 ;da A
1055 0245+C AA ;mov R2,A ;DDJ:=BCD(BLo(5-0))
1056 0246+C A7 ;cpl C
1057 0247+C F6 58+C ;jc low 12%
1058 0249+C ;dodac:100 do D
1059 0249+C BD 01 ;mov R5,#01H
1060 024B+C BC 00 ;mov R4,#00H ;T:=0100 BCD
1061 024D+C 54 7B+C ;call DADDT ;D:=D+T
1062 024F+C ;starszy bajt
1063 024F+C FE 28% ;mov A,R6
1064 0250+C F2 58+C ;jb7 low 12% ;BLo(7)=0; BLo < 120
1065 0252+C BD 01 ;mov R5,#01H ;BLo(7)=1; BLo >= 120
1066 0254+C BC 28 ;mov R4,#28H ;T:=0128 BCD
1067 0256+C 54 7B+C ;call DADDT ;D:=D+T
1068 0258+C 19 12% ;inc R1
1069 0259+C F1 ;mov A,@R1 ;get BHi
1070 025A+C 53 03 ;anl A,#3
1071 025C+C C6 71+C ;jz low 99% ;BHi <= 3
1072 025E+C 37 ;cpl A
1073 025F+C AE ;mov R6,A
1074 0260+C 32 68+C ;jb1 low 56% ;BHi(1)=0; B <= 512
1075 0262+C BD 05 ;mov R5,#05H ;BHi(1)=1; B >= 512
1076 0264+C BC 12 ;mov R4,#12H ;T:=0512 BCD
1077 0266+C 54 7B+C ;call DADDT ;D:=D+T
1078 0268+C FE 56% ;mov A,R6
1079 0269+C D2 71+C ;jb6 low 99% ;BHi(0)=0
1080 026B+C 0D 02 ;mov R5,#02H ;BHi(0)=1
1081 026D+C BC 56 ;mov R4,#56H ;T:=0256 BCD
1082 026F+C 54 7B+C ;call DADDT ;D:=D+T
1083 0271+C 03 99% ;ret
1084 0272+C
1085 0272+C BD 01 1% ;mov R5,#01H
1086 0274+C BC 28 ;mov R4,#28H ;T:=0128
1087 0276+C 54 7B+C ;call DADDT ;D:=D+T
1088 0278+C F1 ;mov A,@R1
1089 0279+C 44 42+C ;jmp 2%
1090
-----
1091 027B+C
1092 027B+C DADDT ;sumuje dwie czterocyfrowe liczby BCD (D:= D + T)
1093 027B+C ;TTS = R5
1094 027B+C ;TDJ = R4
1095 027B+C ;DTS = R3
1096 027B+C ;DDJ = R2
1097 027B+C 2A ;xch A,R2
1098 027C+C 6C ;add A,R4 ;acc:= DDJ + TDJ
1099 027D+C 57 ;da A
1100 027E+C 2A ;xch A,R2 ;DDJ:= DDJ + TDJ, C:= 0/1

```

```

1101 027F+C 2B          xch  A,R3
1102 0280+C 7D          addc A,R5          ;acc:= UFS + ITS+C
1103 0281+C 57          da   A
1104 0282+C 2B          xch  A,R3
1105 0283+C 83          ret
1106 -----
1107 0284+C          ;zamienia cztery cyfry BCD na znaki ASCII
1108 0284+C          BCD4AS ;Liczba BCD umieszczona w R5 (Hi) i R4 (Lo) zostaje zamieniona na znaki
1109 0284+C          ;ASCII w R2 (Lo) ... R5 (Hi)
1110 E 0002          -AOUTL = 2
1111 E 0004          -DINL  = 4
1112 0284+C B8 02          mov  R0,# AOUTL      ;Lo-bajt ASCII
1113          ;BCD4AS+2
1114 0286+C B9 04          mov  R1,# DINL      ;Lo-bajt BCD
1115          ;BCD4AS+4
1116 0288+C 54 8D+C       call 5$
1117 028A+C 19           inc  R1
1118 028B+C 44 8D+C       jmp 5$
1119 028D+C
1120 028D+C 54 93+C       5$ call 10$
1121 028F+C F1           mov  A,@R1
1122 0290+C 47           swap A
1123 0291+C 44 94+C       jmp 12$
1124 0293+C F1           10$ mov  A,@R1
1125 0294+C 47           12$ swap A
1126 0295+C 53 0F        anl  A,# 0FH
1127 0297+C 43 30        15$ orl  A,# 48
1128 0299+C A0           16$ mov  @R0,A
1129 029A+C 18           inc  R0
1130 029B+C 83          ret
1131 -----
1132 029C+C          BCD4AX ;Zamienia cztery cyfry BCD na znaki ASCII umieszczone w XRAM od adresu
1133 029C+C          ;wskazywanego przez R0 w porzadku od najbardziej znaczącej cyfry.
1134 029C+C          ;Za znakami ASCII umieszczana jest spacja i bajt NUL.
1135 029C+C          ;Cyfry BCD sa umieszczone w dwu bajtach, (R1) wskazuje tysiacę-setki
1136 029C+C          ;(R1)-1 dziesiątki-jednostki.
1137 029C+C          ;Standardowe wejście z BCD10A/B nie wymaga ustawienia K1
1138 E 0003          DINH  = 3
1139 029C+C B9 03          mov  R1,# DINH      ;Hi byte BCD
1140          ;BCD4AX+2
1141 029E+C 54 AA+C       call 5$
1142 02A0+C C9           dec  R1
1143 02A1+C 54 AA+C       call 5$
1144 02A3+C 23 20        mov  A,# 32          ;spacja
1145 02A5+C 54 B6+C       call 16$
1146 02A7+C 27           clr  A               ;NULL
1147 02A8+C 44 B6+C       jmp 16$
1148 02AA+C 54 B0+C       5$ call 10$
1149 02AC+C F1           mov  A,@R1
1150 02AD+C 47           swap A
1151 02AE+C 44 B1+C       jmp 12$
1152 02B0+C F1           10$ mov  A,@R1
1153 02B1+C 47           12$ swap A
1154 02B2+C 53 0F        anl  A,# 0FH
1155 02B4+C 43 30        15$ orl  A,# 48
1156 02B6+C 90           16$ movx @R0,A
1157 02B7+C 18           inc  R0
1158 02B8+C 83          ret
1159 -----
1160 02B9+C          BCD4AR ;Zamienia cztery cyfry BCD na znaki ASCII umieszczone w PRAM od adresu
1161 02B9+C          ;wskazywanego przez R0 w porzadku od najbardziej znaczącej cyfry.
    
```

37

```

1162 02B9+C      ;Za znakami ASCII umieszczana jest spacja i bajt NUL
1163 02B9+C      ;Cyfry BCD sa umieszczone w dwu bajtach, (R1) wskazuje tysiace-setki
1164 02B9+C      ;(R1)-i dziesiatki-jednostki.
1165 02B9+C      ;Standardowe wejście z BCD10A/B nie wymaga ustawienia R1
1166 E          0003      DINH = 3
1167 02B9+C      B9 03      mov R1,# DINH      ;Hi byte BCD
1168             ;BCD4AS+2
1169 02BB+C      54 C7+C    call 5$
1170 02BD+C      C9        dec R1
1171 02BE+C      54 C7+C    call 5$
1172 02C0+C      23 20      mov A,# 32      ;spacja
1173 02C2+C      54 D3+C    call 16$
1174 02C4+C      27        clr A           ;NULL
1175 02C5+C      44 D3+C    jmp 16$
1176 02C7+C      54 CD+C    5$ call 10$
1177 02C9+C      F1        mov A,R1
1178 02CA+C      47        swap A
1179 02CB+C      44 CE+C    jmp 12$
1180 02CD+C      F1        10$ mov A,R1
1181 02CE+C      47        12$ swap A
1182 02EF+C      53 0F      anl A,# 0FH
1183 02D1+C      43 30      15$ orl A,# 48
1184 02D3+C      A0        16$ mov @R0,A
1185 02D4+C      18        inc R0
1186 02D5+C      83        ret
1187             ;-----
1188             ;-----
1189 02D6+C      BCDALL ;tworzy w XRAM lancuch ASCII do wydruku danych
1190 02D6+C      ;
1191             ;-----

```



```

034B+C 00
1249 034C+C ;adresy odczytu danych Lo (poczatek @ TARIK2 + LTK + 1)
1250 034C+E C1 D1 E1 00 db KX20+1, KY20+1, KZ20+1, 0, 0, 0, 0, 0, 0
0350+C 00 00 00 00
0354+C 00
1251 0355+C ;adresy STARTU (poczatek @ TARIK2 + 2*LTK + 1)
1252 0355+C C0 D0 E0 00 db KX20, KY20, KZ20, 0, 0, 0, 0, 0, 0
0359+C 00 00 00 00
035D+C 00
1253 ;-----
1254 035E+C SYMKD ;symulacja danych odczytanych z kamer
1255 ; mov A,# 12H
1256 ; mov R2,A
1257 ; movR KA0Lo
1258 ; add A,# 7
1259 ; inc R1
1260 ; mov @R1,A
1261 ; djnz R2,low 1$
1262 035E+C
1263 035E+C B9 20 mov R1,# KA0Lo
1264 0360+C BA 0C mov R2,# 12
1265 0362+C BB 6C+C mov R3,# low 10$
1266 0364+C FB 1$ mov A,R3
1267 0365+C A3 movP A,@A
1268 0366+C A1 mov @R1,A
1269 0367+C 19 inc R1
1270 0368+C 1B inc R3
1271 0369+C EA 64+C djnz R2,low 1$
1272 036B+C B3 ret
1273 036C+C 12 00 20 01 10$ DB 12H, 0, 20H, 1, 2FH, 2, 3CH, 3, 4CH, 0, 5BH, 1
0370+C 2F 02 3C 03
0374+C 4C 00 58 01
1274 ;-----

```




```

1345 044D+C F4 00+C          call  CGETRO
1346 044F+C 37              cpl   A
1347 0450+C 5C              anl   A,R4          ;maska @R4
1348 0451+C              movR  R0TMP         ;store
1351 0454+C          TSTSV2 ;identyfikacja i obsluga przerwan wg priorytetu dla wariantu #2
1352 0454+C F2 D4+C          jb7   low ACLOSV   ;SR0 7: ACLO, highest priority
1353 0456+C 72 68+C          jb3   low 6$      ; 3: RIK
1354 0458+C 32 6B+C          jb1   low 5$      ; 1: RIR
1355 045A+C 12 6E+C          jb0   low 4$      ; 0: RIS
1356 045C+C D2 71+C          jb6   low 3$      ; 6: RIX
1357 045E+C 52 D9+C          jb2   low RIOSVC  ; 2: RIO
1358 0460+C 92 DE+C          jb4   low INTERR  ; 4:
1359 0462+C B2 DE+C          jb5   low INTERR  ; 5: , lowest priority
1360 0464+C          ;obsluzone wszystkie przerwania
1361 0464+C          movA  ACCSTO       ;restore acc
1364 0467+C 93              retr
1365 0468+C          ;wektory obslugi przerwan
1366 0468+C 6$ ; jmp   !!!          ;RIK
1367 0468+C          ds   3
1368 046B+C 5$ ; jmp   ...          ;RIR
1369 046B+C          ds   3
1370 046E+C 4$ ; jmp   !!!          ;RIS
1371 046E+C          ds   3
1372 0471+C 3$ ; jmp   !!!          ;RIX
1373 0471+C          ds   3
1374          ;-----
1375 0474+C          RETSV2 ;powrot z obslugi przerwan wg wariantu #2, musi byc wyzerowany bit
1376 0474+C          ;R0TMP obsluzonego przerwania
1377 0474+C          movA  R0TMP       ;restore R0R
1380 0477+C          ;sel  MB0
1381 0477+C 84 29+C          jmp   TSTSV1
1382          ;-----
1383 0479+C          JRQSV3 ;wariant #3 obslugi przerwan w stanach LOR, ZDAR, ZNAR
1384 0479+C          ;jmp  RQSV3
1385 0479+C F4 00+C          call  CGETRO
1386 047B+C 37              cpl   A
1387 047C+C 5C              anl   A,R4          ;maska @R4
1388 047D+C          movR  R0TMP         ;store
1391 0480+C          TSTSV3 ;identyfikacja i obsluga przerwan wg priorytetu dla wariantu #3
1392 0480+C F2 D4+C          jb7   low ACLOSV   ;SR0 7: ACLO, highest priority
1393 0482+C 72 94+C          jb3   low 6$      ; 3: RIK
1394 0484+C 32 97+C          jb1   low 5$      ; 1: RIR
1395 0486+C 12 9A+C          jb0   low 4$      ; 0: RIS
1396 0488+C D2 9D+C          jb6   low 3$      ; 6: RIX
1397 048A+C 52 D9+C          jb2   low RIOSVC  ; 2: RIO
1398 048C+C 92 DE+C          jb4   low INTERR  ; 4:
1399 048E+C B2 DE+C          jb5   low INTERR  ; 5: , lowest priority
1400 0490+C          ;obsluzone wszystkie przerwania
1401 0490+C          movA  ACCSTO       ;restore acc
1404 0493+C 93              retr
1405 0494+C          ;wektory obslugi przerwan
1406 0494+C 6$ ; jmp   !!!          ;RIK
1407 0494+C          ds   3
1408 0497+C 5$ ; jmp   ...          ;RIR
1409 0497+C          ds   3
1410 049A+C 4$ ; jmp   !!!          ;RIS
1411 049A+C          ds   3
1412 049D+C 3$ ; jmp   !!!          ;RIX
1413 049D+C          ds   3
1414          ;-----
1415 04A0+C          RETSV3 ;powrot z obslugi przerwan wg wariantu #3, musi byc wyzerowany bit

```

```

1416 04A0+C          ;RQTMP obsluzonego przerwania
1417 04A0+C          movA  RQTMP          ;restore RQR
1420 04A3+C          ;sel  MB0
1421 04A3+C  84 80+C  jmp    TSTSV3
1422
-----
1423 04A5+C          JRQSV4 ;wariant #4 obslugi przerwan w stanie transmisja RIS
1424 04A5+C          ;jmp  RQSV4
1425 04A5+C  F4 00+C  call  CGETRQ
1426 04A7+C  37      cpl   A
1427 04A8+C  5C      anl   A,R4          ;maska 0R4
1428 04A9+C          movR  RQTMP          ;store
1431 04AC+C          TSTSV4 ;identyfikacja i obsluga przerwan wg priorytetu dla wariantu #4
1432 04AC+C  F2 D4+C  jb7   low ACLOSV    ;SRQ 7: ACLO, highest priority
1433 04AE+C  72 C0+C  jb3   low 6$      ;      3: RIK
1434 04B0+C  32 C3+C  jb1   low 5$      ;      1: RIR
1435 04B2+C  12 C6+C  jb0   low 4$      ;      0: RIS
1436 04B4+C  D2 C9+C  jb6   low 3$      ;      6: RIX
1437 04B6+C  52 D9+C  jb2   low RIOSVC   ;      2: RIO
1438 04B8+C  92 DE+C  jb4   low INTERR   ;      4:
1439 04BA+C  B2 DE+C  jb5   low INTERR   ;      5: , lowest priority
1440 04BC+C          ;obsluzone wszystkie przerwania
1441 04BC+C          movA  ACCSTO          ;restore acc
1444 04BF+C  93      retr
1445 04C0+C          ;wektory obslugi przerwan
1446 04C0+C  6$     ; jmp  !!!          ;RIK
1447 04C0+C          ds    3
1448 04C3+C  5$     ; jmp  ...          ;RIR
1449 04C3+C          ds    3
1450 04C6+C  4$     ; jmp  !!!          ;RIS
1451 04C6+C          ds    3
1452 04C9+C  3$     ; jmp  !!!          ;RIX
1453 04C9+C          ds    3
1454
-----
1455 04CC+C          RETSV4 ;powrot z obslugi przerwan wg wariantu #4, musi byc wyzerowany bit
1456 04CC+C          ;RQTMP obsluzonego przerwania
1457 04CC+C          movA  RQTMP          ;restore RQR
1460 04CF+C          ;sel  MB0
1461 04CF+C  84 AC+C  jmp    TSTSV4
1462
-----
1463 04D1+C          JRQSV5 ;wariant #5 obslugi przerwan
1464 04D1+C          JRQSV6 ;wariant #6 obslugi przerwan
1465 04D1+C          JRQSV7 ;wariant #7 obslugi przerwan
1466 04D1+C          JRQSV8 ;wariant #8 obslugi przerwan
1467 04D1+C          ; jmp  !!!          ;do sygnalizacji bledu
1468 04D1+C          ds    3
1469
-----
1470 ;JRQSVn ;wariant #n obslugi przerwan
1471 ; jmp  RQSVn
1472 ; call CGETRQ
1473 ; cpl  A
1474 ; anl  A,R4          ;maska 0R4
1475 ; movR RQTMP          ;store
1476 ;TSTSVn ;identyfikacja i obsluga przerwan wg priorytetu dla wariantu #n
1477 ; jb7  low 7$      ;SRQ 7, highest priority :ACLO
1478 ; jb6  low 6$      ;      6: RIK
1479 ; jb5  low 5$      ;      5: RIR
1480 ; jb4  low 4$      ;      4: RIS
1481 ; jb3  low 3$      ;      3: RIX
1482 ; jb2  low 2$      ;      2: RIO
1483 ; jb1  low 1$      ;      1:
1484 ; jb0  low 0$      ;      0, lowest priority :

```

```

1485 ;# ;obsluzone wszystkie przerwania
1486 ;# movA ACCSTD ;restore acc
1487 ;# retr
1488 ;# ;wektory obslugi przerwan
1489 ;#7$ ; jmp ... ;ACLO
1490 ;# ds 3
1491 ;#6$ ; jmp ... ;RIK
1492 ;# ds 3
1493 ;#5$ ; jmp ... ;RIR
1494 ;# ds 3
1495 ;#4$ ; jmp ... ;RIS
1496 ;# ds 3
1497 ;#3$ ; jmp ... ;RIX
1498 ;# ds 3
1499 ;#2$ ; jmp ... ;RIO
1500 ;# ds 3
1501 ;#1$ ; jmp ... ;
1502 ;# ds 3
1503 ;#0$ ; jmp ... ;
1504 ;# ds 3
1505 ;-----
1506 ;#RETSVn ;powrot z obslugi przerwan wg wariantu #n, musi byc wyzerowany bit
1507 ;# ;RQTMP obsluzonego przerwania
1508 ;# movA RQTMP ;restore RQR
1509 ;# ;sel NB0
1510 ;# jmp TSTSV1
1511 ;-----
1512 04D4+C ACLOSV ;nieobslugiwane przerwanie SR07 (ACLO)
1513 04D4+C outP1 29 ;ERROR# 29
1516 04D7+C E4 55+C jmp CLOOP
1517 ;-----
1518 04D9+C RIOSVC ;nieobslugiwane przerwanie SR02 (RIO)
1519 04D9+C outP1 30 ;ERROR# 30
1522 04DC+C E4 55+C jmp CLOOP
1523 ;-----
1524 04DE+C INTERR ;nieobslugiwane przerwania SR04,5
1525 04DE+C outP1 31 ;ERROR# 31
1528 04E1+C E4 55+C jmp CLOOP
1529 ;-----
1530 04E3+C INTCNT ;Obsluga przerwania CNT
1531 ;-----
1532 04E3+C IIRV1 ;obsluga przerwania RIR, v1
1533 04E3+C ;UWAGA: w obecne wersji pakietu RIR nie sa rozrozniane przerwania
1534 04E3+C ;B255 /1 i /2. Przerwanio jest obslugiwana tylko komunikacja z PO.
1535 04E3+C outP1 27
1538 04E6+C E4 55+C jmp CLOOP
1539 ;-----
1540 04E8+C IISV1 ;obsluga przerwania RIS, v1
1541 04E8+C ;przerwanie jest ignorowane
1542 04E8+C 53 FE anl A,# 11111110B ;RQTMP(0)= 0
1543 04EA+C A1 mov @R1,A
1544 04EB+C 84 4B+C jmp RETSV1
1545 ;-----
1546 04ED+C RIKSV1 ;obsluga przerwan RIK w wariancie #1
1547 ;-----
1548 04ED+C RIGSV1 ;obsluga przerwan RIS w wariancie #1
1549 04ED+C ;przerwania zglosza wylacznie odbiornik RIS
1550 04ED+C ;***** OBSLUGA NIE ZREALIZOWANA *****
1551 04ED+C outP1 32 ;ERROR# 32
1554 04F0+C ; nop
1555 04F0+C E4 55+C jmp CLOOP

```

HH

1556

;

Input: S.548

Module: ".MAIN."

```

1627 0549+C          ;C= 0, bufor nadajnika nie odczytany
1628 0549+C          ;moze COS zrobic ?
1629 0549+C          ;
1630 0549+C  F4 4B+C  15$  call  CRxRdy          ;kontrola got. odb.komunikacji
1631 054B+C  F6 59+C          jc    low 20$
1632 054D+C          ;C= 0, sa dane do obczytania
1633 054D+C          ;umozliwic przejście do IRD:
1634 054D+C          movA  INSTNT
1637 0550+C  43 20          orl  A,# 00100000B          ;INSTNT(5)= 1
1638 0552+C  A1          mov  @R1,A
1639 0553+C          movA  RKOLEJ
1642 0556+C  43 20          orl  A,# 00100000B          ;RKOLEJ(5)= 1
1643 0558+C  A1          mov  @R1,A
1644 0559+C  84 4B+C  20$  jmp   RETSV1
1645
-----
1646 055B+C          RIXSV1 ;obsługa przerwan RIX w wariacie #1
1647 055B+C          ;przerwania zgłasza wyłącznie odbiornik (XPB)
1648 055B+C  53 F7          anl  A,#11110111B          ;RQTMP(3)=0
1649 055D+C  A1          mov  @R1,A          ;store RQTMP
1650 055E+C          ;wywołanie procedury odbioru danych dla interpretera polecen
1651 055E+C  F5          sel  NB1
1652 055F+C  74 00+D        call IRD
1653 0561+C  E5          sel  NB0
1654 0562+C  84 4B+C        jmp   RETSV1          ;powrót z obsługi RIX
1655
-----
1656 0564+C          SMDATA ;wykonanie polecen: RQKPL, RQNK, RQSK
1657 0564+C
1658
if um = 1
1659 0564+C          ; call  SYMKD
1660 0564+C  00          nop
1661 0565+C  00          nop
1662 0566+C  D4 40+C        call  DSPRKP
1663 0568+C  D4 50+C        call  DSPRKD
1664
else
1670 056A+C
1671 056A+C  27          clr  A
1672 056B+C          movR  SUMA
1675 056E+C  B8 69          mov  R0,# WRNPO
1676 0570+C  B0 8A          mov  @R0,# 10001010B     ;do wysłania dane
1677 0572+C          movA  KODI
1680 0575+C  AA          mov  R2,A
1681 0576+C  D3 B0          xrl  A,# 10110000B
1682 0578+C  C6 88+C        jz   low 20$          ;KODI = RQKPL
1683 057A+C  FA          mov  A,R2
1684 057B+C  53 F8          anl  A,# 11110000B
1685 057D+C  D3 B8          xrl  A,# 10111000B
1686 057F+C  C6 A9+C        jz   low 50$          ;KODI = RQKN
1687 0581+C  FA          mov  A,R2
1688 0582+C  D3 AB          xrl  A,# 10101000B
1689 0584+C  96 84+C        10$  jnz  low 10$          ;error
1690 0586+C  C4 D6+C        jmp   SNDSK          ;KODI = RQSK
1691 0588+C
-----
1692
20$ ;wysłać komplet wyników
1693 0588+C          mov  R7,# 254
1694 058B+C  BF FE          mov  R6,# 0
1695 058A+C  BE 00          movA  KON
1696 058C+C          jz   low 80$
1699 058F+C  C6 DB+C
1700
if wpo = 1
1748 0591+C
1749
if wpo = 2

```

47

```

1750 ;mpo = 2, wersja dla konfiguracji dowolnej
1751 0591+C A4 97+C jmp 1$
1752 0593+C F6 97+C 4$ jc low 1$
1753 0595+C ;C = 0, sprawdzono 8 kanalow
1754 0595+C A4 8B+C 6$ jmp 52$ ;wszystkie tory
1755 0597+C
1756 0597+C 1$ ;wyznaczenie wlaczonego kanalu
1757 0597+C movA KON
1760 059A+C 4F orl A,R7
1761 059B+C 37 cpl A
1762 059C+C 96 A1+C jnz low 30$ ;ten kanal (R6) nie nalezy do konfiguracji
1763 059E+C 2$ ;ten kanal (R6) nalezy do konfiguracji
1764 059E+C FE mov A,R6
1765 059F+C D4 BA+C call SNDKN ;wyslac dane kanalu #(A)
1766 05A1+C
1767 05A1+C FF 30$ mov A,R7
1768 05A2+C 97 clr C
1769 05A3+C A7 cpl C
1770 05A4+C F7 rlc A
1771 05A5+C AF mov R7,A ;maska
1772 05A6+C 1E inc R6 ;dla nastepnego kanalu
1773 05A7+C A4 93+C jmp 4$
1774 endif
1775 05A9+C
1776 ;.....
1777 05A9+C 50$ ;wyslac wyniki okreslone przez RQKN
1778 05A9+C movA A3 ;argument polecenia = KN
1781 05AC+C 03 10+C add A,# low TABOCT
1782 05AE+C E3 movP3 A,EA
1783 05AF+C B8 40 mov R0,# KON
1784 05B1+C 50 anl A,R0
1785 05B2+C 96 B8+C jnz low 51$
1786 05B4+C ;blad, ten tor nieaktywny
1787 05B4+C B8 69 mov R0,# WRMPO
1788 05B6+C A4 D8+C jmp 00$ ;wyslac 'UWAGA','TKONBL'
1789 05B8+C
1790 05B8+C 51$
1791 05B8+C F1 mov A,R1 ;acc:= A3
1792 05B9+C D4 BA+C call SNDKN
1793 05BB+C B8 69 52$ mov R0,# WRMPO
1794 05BD+C B0 84 mov R0,# 10000100B ;do wyslania SUMA
1795 05BF+C F5 sel NB1
1796 05C0+C 54 00+D call IWR
1797 05C2+C 74 C1+D 53$ call VRxRdy
1798 05C4+C F6 C2+C jc low 53$
1799 05C6+C B8 6A mov R0,# RDMPO
1800 05C8+C B0 10 mov R0,# 00010000B ;RDMPO:= 10H, czeka na potwierdzenie SUMY
1801 05CA+C 74 00+D call IRD
1802 05CC+C E5 sel NB0
1803 05CD+C movA RDMPO ;i powrot do petli
1806 05D0+C B1 E0 mov R1,# 11100000B ;RDMPO(7-5):= 111
1807 05D2+C A1 mov R1,A
1808 05D3+C 27 clr A
1809 05D4+C movR WRMPO ;WRMPO:= 0
1812 05D7+C B3 ret
1813 ;.....
1814 05D8+C ;blad, wszystkie tory nieaktywne
1815 05D8+C B0 85 80$ mov R0,# 10000101B ;do wyslania 'UWAGA','<KODO>'
1816 05DA+C B8 70 mov R0,# KODO
1817 05DC+C B0 C5 mov R0,# TKONBL
1818 05DE+C F5 sel NB1

```


1819	05DF+C	54 00+D	call	IMR
1820	05E1+C	E5	sel	MB0
1821	05E2+C	83	ret	
1822				

3

49


```

1915 0654+C BA 0C      mov    R2,# 12
1916 0656+C B9 20      mov    R1,# KA0Lo
1917 0658+C F1        1$     mov    A,@R1
1918 0659+C 39        outL   P1,A
1919 065A+C 19        inc    R1
1920 065B+C EA 58+C   djnz   R2,low 1$
1921 065D+C F0        mov    A,@R0
1922 065E+C 39        outL   P1,A
1923 065F+C 83        ret
1924
-----
1925                ;przepisują ze str. # 3 ROM wg. (R1)=(ROMPNT) do IO/XRAM adresowanego przez (R0):
R3toX3  mov    A,R1                ;trzy bajty
1926 0660+C F9        movP3  A,@A
1927 0661+C E3        movX   @R0,A
1928 0662+C 90        inc    R1
1929 0663+C 19        R3toX2 mov    A,R1                ;dwa bajty
1930 0664+C F9        movP3  A,@A
1931 0665+C E3        movX   @R0,A
1932 0666+C 90        inc    R1
1933 0667+C 19        R3toX1 mov    A,R1                ;jeden bajt
1934 0668+C F9        movP3  A,@A
1935 0669+C E3        movX   @R0,A
1936 066A+C 90        inc    R1
1937 066B+C 19        ret
1938 066C+C 83
1939
-----
1940 066D+C          PRG53 ;Procedura programowania układu 8253
1941 066D+C          ;Dane wejściowe: PNTIO - adres Control układu, R7 - numer licznika,
1942 066D+C          ;R6 - wskaźnik adresu danych (pg# 3) do programowania licznika.
1943 066D+C          ;Dane w kolejności: CW, LoDiv, HiDiv.
1944 066D+C          ;Procedura zwraca R6 = R6 + 3, zmienia A, R0, R2.
1945 066D+C F4 17+C  call   CSETIO                ;setIO
1946 066F+C FE        mov    A,R6                ;pnt to data
1947 0670+C E3        movP3  A,@A                ;get CW
1948 0671+C AC        mov    R4,A                ;R4:= CW
1949 0672+C FF        mov    A,R7                ;cnt#
1950 0673+C 77        rr     A
1951 0674+C 77        rr     A
1952 0675+C 4C        orl   A,R4                ;CW of cnt# R7
1953 0676+C D4 87+C  call   30$                 ;snd CW
1954 0678+C C8        dec    R0                 ;cnt# 2
1955 0679+C C8        dec    R0                 ;cnt# 1
1956 067A+C C8        dec    R0                 ;cnt# 0
1957 067B+C FB        mov    A,R0
1958 067C+C 6F        add    A,R7
1959 067D+C A8        mov    R0,A                ;adr. cnt# R7
1960 067E+C D4 85+C  call   20$                 ;snd LoDiv
1961 0680+C D4 85+C  call   20$                 ;snd HiDiv
1962 0682+C F4 10+C  call   CRSTP2              ;restore P2
1963 0684+C 83        ret
1964 0685+C FE        20$   mov    A,R6
1965 0686+C E3        movP3  A,@A
1966 0687+C 90        30$   movX   @R0,A
1967 0688+C 1E        inc    R6
1968 0689+C 83        ret
1969
-----
1970 068A+C          INI51 ;Procedura inicjowania 8251
1971 068A+C F4 17+C  call   CSETIO                ;setIO
1972 068C+C 27        clr    A
1973 068D+C 90        movX   @R0,A                ;MIB: synch mode
1974 068E+C 90        movX   @R0,A                ;1-st synch chr
1975 068F+C 90        movX   @R0,A                ;2-nd synch chr

```

```

1976 0690+C 23 40      mov     A,#64          ;CIB=40H, IR
1977 0692+C 90        movX   @R0,A          ;snd
1978 0693+C          movAE  SWA           ;rd SW
1981 0696+C 53 03     anl    A,#3          ;b7-2:= 0
1982 0698+C 03 AD+C   add    A,#low 30$
1983 069A+C A3        movP   A,@A          ;
1984 069B+C 6F        add    A,R7
1985 069C+C A9        mov    R1,A          ;data addr.in PR651D (pg# 3)
1986 069D+C D4 64+C   call  -R3toX2       ;snd MIB, CIB (enable rd)
1987 069F+C          movR   CIBTMP        ;store CIB
1990 06A2+C 00        nop
1991 06A3+C 00        nop
1992 06A4+C C8        dec    R0
1993 06A5+C 80        movX   A,@R0
1994 06A6+C 80        movX   A,@R0
1995 06A7+C 1B        inc    R0
1996 06A8+C D4 B1+C   call  CLRXE         ;dis. rd
1997 06AA+C F4 10+C   call  CRSTP2        ;restore P2
1998 06AC+C 83        ret
1999 06AD+C 00 02 04 06 30$ db    #,2,4,6
2000          ;-----
2001 06B1+C          CLRXE ;
2002 06B1+C B9 64      mov    R1,# CIBTMP
2003 06B3+C 21        xch   A,@R1         ;acc - CIB
2004 06B4+C 53 EB     anl   A,# 11101011B ;clr RXE, ER
2005 06B6+C 18        inc   R0
2006 06B7+C 90        movX  @R0,A         ;snd
2007 06B8+C 21        xch   A,@R1         ;CIB - acc
2008 06B9+C 83        ret
2009          ;-----
2010 06BA+C          SNDKN ;wyslac wyniki z kamery #(A)
2011 06BA+C E7        rl    A
2012 06BB+C E7        rl    A
2013 06BC+C 03 20     add   A,# KA0Lo
2014 06BE+C B8 7B     mov   R0,# PNTARG
2015 06C0+C A0        mov   @R0,A         ;PNTARG wskazuje skad dane
2016 06C1+C B8 69     mov   R0,# WRNPO
2017 06C3+C B0 8A     mov   @R0,# 10001010B ;wyslac dane
2018 06C5+C B8 6F     mov   R0,# LARGO
2019 06C7+C B0 04     mov   @R0,# 4       ;cztery bajty
2020 06C9+C FA        mov   A,R2
2021 06CA+C          movR  KONTMP        ;store R2
2022 06CD+C F5        sel   MB1
2023 06CE+C 54 00+D   call  IWR
2024 06D0+C E5        sel   MB0
2025 06D1+C          movA  KONTMP        ;restore R2
2026 06D4+C AA        mov   R2,A
2027 06D5+C 83        ret
2032          ;-----
2033 06D6+C          SNDK ;wyslac do PO stan torow kamerowych
2034 06D6+C C4 D6+C   1$   jmp   1$
2035 06D8+C          ;etykieta PRO_FORMA
2036          ;-----

```

52


```

2105 ;-----
2106 073B+C CPUTXM ;acc pisze do XRAM na strone XRAMPG, bajt adresowany przez PNTDAT
2107 073B+C 2A xch A,R2 ;store acc
2108 073C+C F4 41+C call CSETXM ;selXRAM pg# XRAMPG
2109 073E+C 2A xch A,R2 ;restore acc
2110 073F+C E4 2C+C jmp CPUTX
2111 ;-----
2112 0741+C CSETXM ;ustawia port P2 na dostep do XRAM pg# XRAMPG
2113 0741+C F5 sel MB1
2114 0742+C 34 41+D call SETXM
2115 0744+C E5 sel MB0
2116 0745+C 83 ret
2117 ; in A,P2
2118 ; movR P2TMP ;store P2
2119 ; anl A,# 11110000B ;selXRAM pg# 0
2120 ; mov R0,# XRAMPG
2121 ; orl A,@R0 ;pg# XRAMPG
2122 ; outl P2,A
2123 ; movA PNTDAT
2124 ; mov R0,A
2125 ; ret
2126 ;-----
2127 0746+C CTxRdy ;kontrola gotowosci nadajnika (komunikacja)
2128 0746+C F5 sel MB1
2129 0747+C 94 C1+D call VTxRdy
2130 0749+C E5 sel MB0
2131 074A+C 83 ret
2132 ; movA MPMS
2133 ; jbl low 50$
2134 ; ;komunikacja lokalnie
2135 ; call CSIEXA ;ustaw XINTEA:= 1
2136 ; call C6TPOS ;odczytaj slowo stanu PD
2137 ; clr C
2138 ; jb3 low 10$ ;INTRA= 1
2139 ; ;XPC3 = INTRA = 0, bufor nadajnika nie odczytany
2140 ; cpl C
2141 ;10$ call CRIEXA ;ustaw XINTEA:= 0
2142 ; ret ;INTRA= 1, TxRdy -> C:= 0
2143 ; ;INTRA= 0, TxNRdy -> C:= 1
2144 ;50$ ;komunikacja zdalnie
2145 ; outP1 43
2146 ; jmp CLOOP
2147 ;-----
2148 074B+C CRxRdy ;kontrola gotowosci odbiornika (komunikacja)
2149 074B+C F5 sel MB1
2150 074C+C 74 C1+D call VRxRdy
2151 074E+C E5 sel MB0
2152 074F+C 83 ret
2153 ; movA MPMS
2154 ; jbl low 50$
2155 ; ;komunikacja lokalnie
2156 ; call C6TPOS ;odczytaj slowo stanu PD
2157 ; clr C
2158 ; jbl low 10$ ;IBFB= 1
2159 ; ;XPC1 = IBFB = 0, bufor odbiornika pusty
2160 ; cpl C
2161 ;10$ ret ;IBFB = 1, RxRdy, -> C:= 0
2162 ; ;IBFB = 0, RxNRdy, -> C:= 1
2163 ;50$ ;komunikacja zdalnie
2164 ; outP1 44
2165 ; jmp CLOOP

```

```

2166 ;-----
2167 0750+C CPTTXD ;wysyla acc do nadajnika komunikacji
2168 0750+C F5 sel MB1
2169 0751+C 34 D8+D call PUTTXD
2170 0753+C E5 sel MB0
2171 0754+C 83 ret
2172 ; xch A,R2
2173 ; movA MPMS
2174 ; jbl low 50$
2175 ; ;komunikacja lokalnie
2176 ; xch A,R2
2177 ; jmp CPTPOD ;pisz acc do PO
2178 ;50$ ;komunikacja zdalnie
2179 ; xch A,R2
2180 ; outP1 45
2181 ; jmp CLOOP
2182 ;-----
2183 0755+C CSIEXA ;ustawia INTEA:=1 w RIX/RIX
2184 ; mov R0, #PNTIO
2185 ; movA MPMS
2186 ; jbl low 50$
2187 ; ;komunikacja lokalnie (PO)
2188 ; mov @R0, #RXP
2189 ; mov A, # 13
2190 ; jmp CPUTIO
2191 ;50$ ;komunikacja zdalnie (RIR)
2192 ; mov @R0, #RRP
2193 ; mov A, # 13
2194 ; jmp CPUTIO
2195 ;-----
2196 0755+C CRIEXA ;ustawia INTEA:=0 w RIX/RIX
2197 ; mov R0, #PNTIO
2198 ; movA MPMS
2199 ; jbl low 50$
2200 ; ;komunikacja lokalnie (PO)
2201 ; mov @R0, #RXP
2202 ; mov A, # 12
2203 ; jmp CPUTIO
2204 ;50$ ;komunikacja zdalnie (RIR)
2205 ; mov @R0, #RRP
2206 ; mov A, # 12
2207 ; jmp CPUTIO
2208 ;-----
2209 0755+C CLOOP; sygnalizacja bledow w MB0
2210 if um = 1
2211 0755+C D4 00+C call DISPRR ;wyswietlenie rejestrow roboczych
2212 ;P1 MS WF RF RK IF PF KF KT WI K4 A3 KI SU
2213 ;
2214 else
2218 0757+C E4 57+C 1$ jmp 1$
2219 ;-----
2220 0759+C SIK2 ;Procedura STOP dla torow kamerowych RIK2.
2221 0759+C ;Wejscie: R2 - wskaznik tablicy adresow X3H pakietow RIK2, adresy
2222 0759+C ;na pg# 3 zakonczone bajtem 0.
2223 0759+C F4 17+C call CSETIO ;selIO
2224 075B+C FB 1$ mov A,R3 ;wskaznik tablicy adresow X3H (pg# 3)
2225 075C+C E3 movP3 A,@A
2226 075D+C C6 64+C jz low 10$ ;wszystkie tory kamerowe
2227 075F+C A9 mov R1,A
2228 0760+C 81 movX A,@R1 ;odczyt X3H, STOP RIK2
2229 0761+C 1B inc R3

```

```
2230 0762+C E4 5B+C      jmp 1$
2231 0764+C F4 10+C      10$ call CRSTP2      ;restore P2
2232 0766+C B3          ret
2233
-----
2234 0767+C      RMtoXL ;(R2) bajtow ze str.# 3 ROM od adresu (R1)
2235 0767+C      ;przepisuje do IO/XRAM wskazanego przez (R0)
2236 0767+C F9          1$   mov  A,R1
2237 0768+C E3          movP3 A,@A
2238 0769+C 90          movX @R0,A
2239 076A+C 19          inc  R1
2240 076B+C EA 67+C     djnz R2,low 1$
2241 076D+C B3          ret
2242
-----
2243 076E+C      RMtoXN ;ze str.# 3 ROM, od adresu (R1) do NUL przepisuje do IO/XRAM
2244 076E+C      ;wskazanego przez (R0).
2245 076E+C F9          1$   mov  A,R1
2246 076F+C E3          movP3 A,@A
2247 0770+C C6 70+C     jZ   low $
2248 0772+C 90          movX @R0,A
2249 0773+C 19          inc  R1
2250 0774+C E4 6E+C     jmp 1$
2251 0776+C B3          2$   ret
2252
-----
```




```

2323 0039+D 14 BF+D          call  DPR653
2324 003B+D
2325                          ;Inicjowanie RIK2
2326 003B+D E5              sel   MB0
2327 003C+D BA 43+C        mov   R2,# low TARIK2
2328 003E+D F4 59+C        call  SIK2                ;STOP RIK2
2329 0040+D BA 43+C        mov   R2,# (( low TARIK2 ) + 2 # LTK + 2)
2330 0042+D 34 00+C        call  RUNIK2
2331 0044+D F5              sel   MB1
2332 0045+D
2333                          ;Programowanie B253/RIS
2334                          ;INISTC inicjowanie zegara RIS
2335 0045+D 23 20          mov   A,# SWA
2336 0047+D                movR  PNTIO
2339 004A+D 34 0D+D        call  GETIO                ;odczyt przelacznikow RJC
2340 004C+D 00            nop
2341 004D+D 53 03          anl   A,# 3                ;SW(1,0) - szybkość transmisji RIS
2342 004F+D 03 22+C        add   A,# low RISTRO       ;adres tablicy ofsetow dla danych
2343 0051+D E3              movP3 A,EA                 ;ofset dla tablicy danych programujacych
2344 0052+D 03 26+C        add   A,# low TRDAT        ;
2345 0054+D AE              mov   R6,A                 ;wskaznik adresu danych programujacych
2346 0055+D 23 83          mov   A,# SCC              ;adr. Control licznika
2347 0057+D                movR  PNTIO
2350 005A+D BF 02          mov   R7,# 2                ;cnt# 2
2351 005C+D 14 BF+D        call  DPR653                ;procedura programujaca licznik
2352 005E+D
2353                          ;Programowanie B251/RIS
2354 005E+D 23 85          mov   A,# SIC
2355 0060+D                movR  PNTIO
2358 0063+D BF 3B+C        mov   R7,# low PR651D
2359 0065+D E5              sel   MB0
2360 0066+D 04 BA+C        call  INI51
2361 0068+D F5              sel   MB1
2362 0069+D
2363                          ;Inicjowanie RIR
2364 0069+D 23 9B          mov   A,# M0IALL           ;Mode 0; PA, PB, PC jako wejsciowe
2365 006B+D                movE  RRP
2368 006E+D B8 92          mov   R0,# RPC
2369 0070+D B0              movX  A,ER0                ;get RPC
2370 0071+D B2 75+D        jb5   low LOS              ;RPC25=1
2371 0073+D 04 BA+D        jmp   ZNAR                  ;ZNAR - do RIR jest dolaczony komputer
2372 0075+D                LOS ;do RIR nie jest dolaczony komputer
2373 0075+D B8 6B          mov   R0,# MPMS
2374 0077+D B0 04          mov   ER0,# 00000100B      ;MPMS(2)= 1
2375 0079+D                movAE SWA                  ;odczyt przelacznikow RJC
2378 007C+D B2 B5+D        jb5   low 20$              ;parallel PRN
2379 007E+D                ;Centronics PRN
2380 007E+D 23 A6          mov   A,# M1QIO           ;Mode 1; PA-0, PB-1, PC(5,4)-0
2381 0080+D                movE  RRP
2384                          ;
2385 0083+D 23 0B          mov   A,# 8                ;clr PRN, RPC24:=1
2386 0085+D 91              movX  ER1,A                ; (2#BITNOS + SETRES)
2387 0086+D 00            nop
2388 0087+D 00            nop
2389                          ;
2390 0088+D 23 09          mov   A,# 9                ;RPC24:=0
2391 008A+D 91              movX  ER1,A                ; (2#BITNOS + SETRES)
2392 008B+D 14 A9+D        call  5$
2393 008D+D FA              mov   A,R2
2394 008E+D C6 A0+D        jz    low 2$
2395 0090+D 23 0A          mov   A,# 10               ;LF

```

```

2396 0092+D          movE   RPA
2399 0095+D   14 A9+D  call   5$
2400 0097+D   FA      mov    A,R2
2401 0098+D   C6 A1+D  jz     low 3$
2402 009A+D          movA   PRNS
2405 009D+D   43 42   orl    A,# 01000010B ;PRNS(6,1):= 11, Rdy Centronics PRN
2406 009F+D   A1      mov    @R1,A
2407 00A0+D          2$ ;Centronics PRN NRdy
2408 00A0+D          10$ ;cd_po_LOS
2409 00A0+D   83      ret     ;return from INIALL
2410
2411 00A1+D          ;Ustaw PRNS
2412 00A1+D          3$   movA   PRNS
2415 00A4+D   43 40   orl    A,# 01000000B ;PRNS(6):= 1, Centronics PRN
2416 00A6+D   A1      mov    @R1,A
2417 00A7+D   04 A0+D  jmp    10$
2418
2419 00A9+D          5$   ;sprawdzenie gotowosci PRN (Centronics)
2420 00A9+D   BA FF   mov    R2,# 255
2421 00AB+D   B8 91   mov    R0,# RPB
2422 00AD+D   80      6$   movX   A,@R0
2423 00AE+D   00      nop
2424 00AF+D   F2 B2+D  jb7    low 7$ ;cpl A ??
2425 00B1+D   83      ret     ;RPB27=1, PRN Busy
2426 00B2+D   EA AD+D  7$   djnz  R2,low 6$ ;PRN Ready
2427 00B4+D   83      ret     ;PRN NReady
2428
2429 00B5+D          20$  ;Parallel PRN
2430 00B5+D          ds     3 ;programowanie-wyjsc RIR; testowanie-PRN
2431 00B8+D   04 A0+D  jmp    10$
2432
2433 00BA+D
2434 00BA+D          ZNAR ;Do RIR jest dolaczony komputer
2435 00BA+D          outP1 42
2438 00BD+D   E4 62+D  jmp    LOOPER
2439
2440 00BF+D
2441
2442 00BF+D          DPR653 ;wyzwanie procedury PRG53 w MB0
2443 00BF+D   E5      sel    MB0
2444 00C0+D   D4 6D+C  call   PRG53
2445 00C2+D   F5      sel    MB1
2446 00C3+D   83      ret
2447
2448 00C4+D          DDSPRR ;wyzwanie DISPRR w MB0, tj. wystawienie rejestrow na P1
2449 00C4+D   E5      sel    MB0
2450          if um = 1
2451 00C5+D   D4 00+C  call   DISPRR ;wyswietlenie rejestrow roboczych
2452          ;P1 MS WF RF RK IF PF KF KT WI K4 A3 KI SU
2453          ;
2454          else
2458 00C7+D   F5      sel    MB1
2459 00C8+D   83      ret
2460

```



```

2529 ~ 0139+D 24 0F+D          jmp     GETX
2530                                     ;
2531 013B+D          PUTXM ;acc pisze do XRAM na stronie XRAM6, bajt adresowany przez PNTDAT
2532 013B+D 2A          xch    A,R2          ;store acc
2533 013C+D 34 41+D    call   SETXM          ;selXRAM pg# XRAM6
2534 013E+D 2A          xch    A,R2          ;restore acc
2535 013F+D 24 2C+D    jmp     PUTX
2536                                     ;
2537 0141+D          SETXM ;ustawia port P2 na dostep do XRAM pg# XRAM6
2538 0141+D 0A          in     A,P2
2539 0142+D          movR   P2TMP          ;store P2
2542 0145+D 53 F0      anl    A,# 11110000B  ;selXRAM pg# 0
2543 0147+D 8B 7D      mov    R0,# XRAM6
2544 0149+D 40          orl    A,R0          ;pg# XRAM6
2545 014A+D 3A          outl  P2,A
2546 014B+D          movA   PNTDAT
2549 014E+D A8          mov    R0,A
2550 014F+D 83          ret
2551                                     ;
2552          ;weryfikacja legalnosci polecen MPO, zwrócony acc=0 oznacza LEGALNOSC
2553 0150+D          VLGOTP ;weryfikacja warunku MPMS=xxx1xxx0 dla : GOTPO
2554 0150+D          movA   MPMS
2557 0153+D 53 11      anl    A,# 00010001B
2558 0155+D D3 10      xrl    A,# 00010000B
2559 0157+D 83          ret
2560                                     ;
2561 0158+D          VLNQPO ;weryfikacja warunku MPMS=XXX11XX0 dla : NGOTPO, ZDAL
2562 0158+D          movA   MPMS
2565 015B+D 53 19      anl    A,# 00011001B
2566 015D+D D3 18      xrl    A,# 00011000B
2567 015F+D 83          ret
2568                                     ;
2569 0160+D          VLRSTP ;weryfikacja warunku MPMS=XXX11X00 dla : SOFTRST, KONF.POCZ, LOK
2570 0160+D          movA   MPMS
2573 0163+D 53 1A      anl    A,# 00011010B
2574 0165+D D3 18      xrl    A,# 00011000B
2575 0167+D 83          ret
2576                                     ;
2577 0168+D          VLSTRP ;weryfikacja warunku MPMS=X1X11X00 dla : RSTRP, TR.X
2578 0168+D          movA   MPMS
2581 016B+D 53 5B      anl    A,# 01011011B
2582 016D+D D3 5B      xrl    A,# 01011000B
2583 016F+D 83          ret
2584                                     ;
2585 0170+D          VLRQP  ;weryfikacja warunku MPMS=X1X11X00 dla : RQKS, RQDKPL, RQDNK
2586 0170+D          movA   MPMS
2589 0173+D 53 59      anl    A,# 01011001B
2590 0175+D D3 58      xrl    A,# 01011000B
2591 0177+D 83          ret
2592                                     ;
2593 0178+D          VERNK  ;czy tor kamery #NK włączony ?
2594 0178+D AA          mov    R2,A
2595 0179+D 1A          inc    R2          ;R2:=A3 + 1
2596          ;          mov    R0,# WDAKR          ;wskaznik roboczych danych konfigur.
2597          ;          dec    R0
2598          ;          dec    R0
2599          ;          mov    A,R0          ;get KON # R
2600 017A+D          movA   KON
2603 017D+D 67          10$   rrc    A
2604 017E+D EA 7D+D    djnz  R2,low 10$
2605 0180+D 83          ret          ;C:= 0 - wylaczony, C:= 1 - włączony

```

61

```

2606 ;-----
2607 0181+D ;Ustawic: RDMPD do odczytu 'UNAGA',
2608 0181+D ; WRMPD do wyslania potwierdzenia kodu: ECHO
2609 0181+D SECHD1 ;wer.1 - polecenie wykonane (GOTPD / NGOTPD)
2610 0181+D B9 6A mov R1,# RDMPD
2611 0183+D B1 E0 mov @R1,# 11100000B ;RDMPD:= 111----B
2612 0185+D ;do odbioru 'UNAGA', dane nie spodziewane
2613 0185+D 34 AF+D CECHD1 call VWRMPD
2614 0187+D B9 69 mov R1,# WRMPD
2615 0189+D B1 88 mov @R1,# 10001000B ;WRMPD:= 10--1000B, do wyslania ECHO
2616 018B+D movA RKOLEJ
2619 018E+D 43 20 orl A,# 00100000B ;RKOLEJ(5):= 1, polecenie natychmiastowe
2620 0190+D A1 mov @R1,A
2621 0191+D movA INSTNT
2624 0194+D 43 10 orl A,# 00010000B ;INSTNT(4):= 1, wywolac IWR
2625 0196+D A1 mov @R1,A
2626 0197+D 83 ret ;jmp IWR
2627 ;-----
2628 0198+D SECHD3 ;wer.3 bledny kod
2629 0198+D B9 6A mov R1,# RDMPD
2630 019A+D B1 E0 mov @R1,# 11100000B ;RDMPD:= 111----B
2631 019C+D ;do odbioru 'UNAGA', dane nie spodziewane
2632 019C+D 34 AF+D call VWRMPD
2633 019E+D B9 69 mov R1,# WRMPD
2634 01A0+D B1 89 mov @R1,# 10001001B ;10--1001B ;do wyslania TKOMBL
2635 01A2+D movA RKOLEJ
2638 01A5+D 43 20 orl A,# 00100000B ;RKOLEJ(5):= 1, polecenie natychmiastowe
2639 01A7+D A1 mov @R1,A
2640 01A8+D movA INSTNT
2643 01AB+D 43 10 orl A,# 00010000B ;INSTNT(4):= 1, wywolac IWR
2644 01AD+D A1 mov @R1,A
2645 01AE+D 83 ret ;jmp IWR
2646 ;-----
2647 01AF+D VWRMPD ;sprawdzenie WRMPD czy sa oczekiwane stany: 00, 04, 07, 08, 09
2648 01AF+D movA WRMPD
2651 01B2+D C6 D3+D jz low 10$ ;eq.
2652 01B4+D F1 mov A,@R1
2653 01B5+D D3 04 xrl A,# 000001000B ;4
2654 01B7+D C6 D3+D jz low 10$ ;eq.
2655 01B9+D F1 mov A,@R1
2656 01BA+D D3 07 xrl A,# 00000111B ;7
2657 01BC+D C6 D3+D jz low 10$ ;eq.
2658 01BE+D F1 mov A,@R1
2659 01BF+D D3 08 xrl A,# 00001000B ;8
2660 01C1+D C6 D3+D jz low 10$ ;eq.
2661 01C3+D F1 mov A,@R1
2662 01C4+D D3 09 xrl A,# 00001001B ;9
2663 01C6+D C6 D3+D jz low 10$ ;eq.
2664 01C8+D B9 7F mov R1,# RERR
2665 01CA+D B1 80 mov @R1,# 80H ;konflikt w WRMPD
2666 01CC+D F4 44+D 11$ call ERROR
2667 01CE+D B9 7F 12$ mov R1,# RERR
2668 01D0+D B1 00 mov @R1,# 0
2669 01D2+D 83 ret ;ignore err. / NO-err.
2670 01D3+D FA 10$ mov A,R2 ;w R2 jest przekazywany wskaznik bledow
2671 01D4+D ;stwierdzonych przez interpretator do
2672 01D4+D ;wykorzystania w procedurze ERROR,
2673 01D4+D ;R2 = 0'ok!
2674 01D4+D C6 CE+D jz low 12$
2675 01D6+D 24 CC+D jmp 11$
2676 ;-----

```

62

```
2677 01D8+D          PUTTXD ;wysyla acc do nadajnika komunikacji
2678 01D8+D  2A      xch  A,R2
2679 01D9+D      movA  MPMS
2682 01DC+D  32 E1+D  jbl   low 50%
2683 01DE+D          ;komunikacja lokalnie
2684 01DE+D  2A      xch  A,R2
2685 01DF+D  24 22+D  jmp   -PUTPOD          ;pisz acc do PO
2686 01E1+D      50%  ;komunikacja zdalnie
2687 01E1+D  2A      xch  A,R2
2688 01E2+D          outP1 45
2691 01E5+D  E4 62+D  jmp   -LOOPER
2692
```

```

2766 0244+D 44 6B+D          jmp 3$
2767                                ;.....
2768 0246+D          14$  ;odebrane: SUMA / KOD b.a. / potw.KOMOK / potw.ECHO / potw.KOM:BL
2769 0246+D          ;/ ostatni B. danych
2770 0246+D          ;4 / 6 / 7 / 8 / 9 / B
2771 0246+D          movA  WRMPO
2774 0249+D 53 30          anl  A,# 00110000B
2775 024B+D A1          mov  @R1,A
2776 024C+D          movA  INSTNT
2779 024F+D 53 EF          anl  A,# 11101111B          ;INSTNT(4)= 0
2780 0251+D A1          mov  @R1,A
2781 0252+D B3          ret
2782                                ;BLAD, w tych stanach nie moze byc danych-do wyslania ???
2783                                ;.....
2784 0253+D          20$  ;A/B, odebrany bajt danych
2785 0253+D          movA  LARGO
2788 0256+D 07          dec  A
2789 0257+D A1          mov  @R1,A          ;LARGO - 1
2790 0258+D D3 01          xrl  A,# 1
2791 025A+D C6 2E+D          jz   low 10$          ;20$ ;do wyslania ostatni bajt
2792 025C+D          ;do wyslania nie ostatni bajt
2793 025C+D 00          nop
2794 025D+D 00          nop
2795 025E+D          movA  WRMPO
2798 0261+D 53 BF          anl  A,# 10111111B          ;WRMPO(6)= 0
2799 0263+D A1          mov  @R1,A
2800 0264+D 44 6B+D          jmp 3$
2801 0266+D          28$  ;do wyslania ostatni bajt
2802 0266+D 00          nop ;mov  A,@R1
2803 0267+D 00          nop ;dec  A
2804 0268+D 00          nop ;mov  @R1,A          ;LARGO - 1
2805 0269+D 00          nop ;jmp  10$
2806 026A+D 00          nop ;
2807                                ;-----
2808 026B+D          3$   ;jakies dane do wyslania, TX gotowy; WRMPO(7,6)=10
2809 026B+D          movA  WRMPO
2812 026E+D 53 0F          anl  A,#-15
2813 0270+D 03 73+D          add  A,# low 29$
2814 0272+D B3          jmpP @A
2815 0273+D          29$  ;tablica adesow pp. wyslania danych
2816 0273+D B3 92 9B 9B          db   low 30$,low 31$,low 32$,low 32$ ;WRMPO(3-0)=0-3
2817 0277+D AF B3 92 B4          db   low 34$,low 30$,low 31$,low 37$ ;WRMPO(3-0)=4-7
2818 027B+D B8 C7 CB CB          db   low 38$,low 39$,low 40$,low 40$ ;WRMPO(3-0)=8-B
2819 027F+D D9 D9 D9 D9          db   low 99$,low 99$,low 99$,low 99$ ;WRMPO(3-0)=C-F
2820                                ;.....
2821 0283+D          30$  ;do wyslania 'UNAGA'
2822 0283+D          ;0 / 5
2823 0283+D 23 FF          mov  A,# 255
2824 0285+D 34 D8+D          50$  call  PUTTXD          ;wyslac zawartosc acc
2825 0287+D          movA  WRMPO
2828 028A+D 43 40          orl  A,# 01000000B          ;WRMPO(6)=1
2829 028C+D A1          mov  @R1,A
2830                                ; jmp  !!!          ;koniec INR
2831 028D+D 97          clr  C
2832 028E+D A7          cpl  C
2833 028F+D 44 00+D          jmp  INR
2834 0291+D B3          ret
2835                                ;.....
2836 0292+D          31$  ;1 / 6, do wyslania KODO
2837 0292+D 27          clr  A
2838 0293+D          movR  SUMA          ;SUMA:=0

```

```

2841 0296+D          movA  KODD
2844 0299+D  44 BB+D  jmp   51$
2845                                     ;.....
2846 029B+D          32$   ;2 / 3, do wyslania argumenty polecenia
2847 029B+D          movA  LARGO
2850 029E+D  B8 7B   mov   R0,# PNTARG
2851 02A0+D  60      add   A,@R0
2852 02A1+D  A8      mov   R0,A          ;pnt-to-ARG#
2853 02A2+D  F0      mov   A,@R0        ;get ARG#
2854 02A3+D  B9 73   mov   R1,# SUMA    ;pnt to SUMA
2855 02A5+D  61      add   A,@R1
2856 02A6+D  A1      mov   @R1,A
2857 02A7+D  F0      mov   A,@R0
2858 02A8+D  00      nop
2859 02A9+D  00      nop
2860 02AA+D  00      nop
2861 02AB+D  00      nop
2862 02AC+D  00      nop
2863 02AD+D  44 85+D jmp   50$          ;do TX wyslac acc
2864                                     ;.....
2865 02AF+D          34$   ;4, do wyslania SUMA
2866 02AF+D          movA  SUMA
2869 02B2+D  44 BB+D  jmp   51$          ;do TX wyslac acc
2870                                     ;.....
2871 02B4+D          37$   ;7, do wyslania potw. KOMOK
2872 02B4+D  23 C4   mov   A,# TKOMOK
2873 02B6+D  44 BB+D jmp   51$
2874                                     ;.....
2875 02B8+D          38$   ;8, do wyslania potw. ECHO(KODI)
2876 02B8+D          movA  KODI
2879 02BB+D  2A      51$   xch   A,R2
2880 02BC+D          movA  RDMPO
2883          ;     anl   A,# 01111111B    ;RDMPO(7)= 0, dane oczekiwane
2884 02BF+D  00      nop
2885 02C0+D  00      nop
2886 02C1+D  43 E0   orl   A,# 11100000B    ;RDMPO(7-5)= 111, oczek. 'uwaga'
2887 02C3+D  A1      mov   @R1,A
2888 02C4+D  2A      xch   A,R2
2889 02C5+D  44 85+D jmp   50$          ;do TX wyslac acc
2890                                     ;.....
2891 02C7+D          39$   ;9, do wyslania potw. KOM.BL
2892 02C7+D  23 C5   mov   A,# TKOMBL
2893 02C9+D  44 BB+D jmp   51$          ;do TX wyslac acc
2894                                     ;.....
2895 02CB+D          40$   ;A / B, do wyslania dane
2896 02CB+D          movA  PNTARG
2899 02CE+D  AB      mov   R0,A
2900 02CF+D  17      inc   A
2901 02D0+D  A1      mov   @R1,A
2902 02D1+D  F0      mov   A,@R0        ;bajt do wyslania
2903 02D2+D  B9 73   mov   R1,# SUMA
2904 02D4+D  61      add   A,@R1
2905 02D5+D  A1      mov   @R1,A        ;suma kontrolna
2906 02D6+D  F0      mov   A,@R0        ;bajt do wyslania
2907 02D7+D  44 85+D jmp   50$          ;do TX wyslac acc
2908                                     ;-----
2909          ;     ;sygnalizacja bledow na wyjsciui P1
2910 02D9+D  E4 5F+D  99$   jmp   ERR23        ; mov   A,# 23
2911 02DB+D          ;     ; outl  P1,A
2912 02DB+D  E4 62+D  98$   jmp   LDOOPER      ; jmp   98$
2913 02DD+D  E4 66+D  97$   jmp   ERR24        ; mov   A,# 24

```

```
2914 02DF+D ; outl P1,A  
2915 02DF+D ; jmp 98$  
2916 ;-----  
2917 ;-----
```



```

2993 034B+D 64 4D+D 17% jmp 19% ;grupa 7 - zadane dane z kanalu NK
2994 ;-----
2995 034D+D 19% ;bledna wartosc KODI(3-0)
2996 034D+D outPl 4 ;ERROR #4
2999 0350+D E4 62+D jmp LOOPER ;97%
3000 ;-----
3001 0352+D 25% ;K4 = 15, SPODZIEWANY 'KOD'
3002 0352+D 74 D2+D call GETRxD
3003 0354+D movR KODI ;store
3006 0357+D F2 5E+D jb7 low 8%
3007 0359+D ;to nie jest KOD !
3008 0359+D ;
3009 0359+D outPl 1
3012 035C+D E4 62+D jmp LOOPER ;97%
3013 ;-----
3014 035E+D 53 07 8% anl A,# 7
3015 0360+D 19 inc R1
3016 0361+D A1 mov @R1,A ;store A3
3017 0362+D C9 dec R1
3018 0363+D F1 mov A,@R1 ;restore KOD
3019 0364+D 53 7B anl A,# 01111000B
3020 0366+D E7 rl A
3021 0367+D 47 swap A
3022 0368+D 19 inc R1
3023 0369+D 19 inc R1
3024 036A+D A1 mov @R1,A ;store K4
3025 036B+D 03 6E+D add A,# low 9%
3026 036D+D B3 jmpP @A ;wg odebranego KODu
3027 036E+D 9% ;tabela skokow wg odebranego KODu
3028 036E+D 7E 80 82 84 db low 80%,low 81%,low 82%,low 83% ;KODI(3-0)=0-3
3029 0372+D 86 88 8A 8C db low 84%,low 85%,low 86%,low 87% ;KODI(3-0)=4-7
3030 0376+D 8E A7 92 A7 db low 88%,low 99%,low 90%,low 99% ;KODI(3-0)=8-B
3031 037A+D A7 A7 A7 B6 db low 99%,low 99%,low 99%,low 97% ;KODI(3-0)=C-F
3032 037E+D ;skok do etykiety 99% dla kodow niewykonywanych (blednych)
3033 037E+D ;tj.: 9, B, C, D, E, F
3034 ;Wektory odebranego kodu
3035 037E+D 84 00+D 80% jmp KODI0
3036 0380+D 84 29+D 81% jmp KODI1
3037 0382+D 84 39+D 82% jmp KODI2
3038 0384+D 84 88+D 83% jmp KODI3
3039 0386+D C4 00+D 84% jmp KODI4
3040 0388+D C4 4F+D 85% jmp KODI5
3041 038A+D C4 5E+D 86% jmp KODI6
3042 038C+D C4 7B+D 87% jmp KODI7
3043 038E+D C4 8C+D 88% jmp KODI8
3044 0390+D 89% ds 2 ;jmp KODI9
3045 0392+D C4 8B+D 90% jmp KODIA
3046 0394+D 91% ds 2 ;jmp KODIB
3047 0396+D 92% ds 2 ;jmp KODIC
3048 0398+D 93% ds 2 ;jmp KODID
3049 039A+D 94% ds 2 ;jmp KODIE
3050 039C+D 95% ;KODIF
3051 039C+D movA KODI ;restore get_byte
3054 039F+D 37 cpl A
3055 03A0+D 96 A7+D jnz low 99%
3056 03A2+D ;sekwencja 'UWAGA', 'UWAGA',
3057 03A2+D B9 6A mov R1,# RDNPD *
3058 03A4+D B1 20 mov @R1,# 00100000B
3059 03A6+D B3 ret
3060 03A7+D
3061 03A7+D 99% ;bledna wartosc KODI

```

Input: S.548

Module: ".MAIN."

```

3062 03A7+D BA 03          mov    R2,# 3
3063 03A9+D 24 9B+D        jmp    SECH03
3064 03AB+D
3065 03AB+D                60$   ;odczyt potwierdzenia SUMY
3066 03AB+D 74 D2+D        call  GETRxD
3067 03AD+D                movR  KODI
3070 03B0+D 53 07        anl  A,# 7
3071 03B2+D 19          inc  R1
3072 03B3+D A1          mov  @R1,A          ;store A3
3073 03B4+D C4 8C+D        jmp  KODI8
3074
-----
3075 03B6+D                97$
3076 03B6+D  RETRY ;ponowienie oczekiwania na 'uwaga'
3077 03B6+D 27          clr  A
3078 03B7+D                movR  RKOLEJ          ;RKOLEJ:= 0
3081 03BA+D                movR  INSTNT          ;INSTNT:= 0
3084 03BD+D AA          mov  R2,A
3085 03BE+D 00          nop
3086 03BF+D 00          nop
3087 03C0+D B3          ret
3088
-----
3089 03C1+D  VRXRdy ;kontrola gotowosci odbiornika (komunikacja)
3090 03C1+D                movA  MPMS
3093 03C4+D 32 CD+D        jbi  low 50$
3094 03C6+D                ;komunikacja lokalnie
3095 03C6+D 34 04+D        call  GETPOS          ;odczytaj slowo stanu PO
3096 03C8+D 97          clr  C
3097 03C9+D 32 CC+D        jbi  low 10$          ;IBFB= 1
3098 03CB+D                ;XPC1 = IBFB = 0, bufor odbiornika pusty
3099 03CB+D A7          cpl  C
3100 03CC+D B3          10$   ret          ;IBFB = 1, RxRdy, -> C:= 0
3101 03CD+D                ;IBFB = 0, RxNRdy, -> C:= 1
3102 03CD+D                50$   ;komunikacja zdalnie
3103 03CD+D                outP1 44
3106 03D0+D E4 55+C        jmp  CLOOP
3107
-----
3108 03D2+D  GETRXD ;czyta dane odbiornika komunikacji
3109 03D2+D                movA  MPMS
3112 03D5+D 32 D9+D        jbi  low 50$
3113 03D7+D                ;komunikacja lokalnie
3114 03D7+D 24 0B+D        jmp  GETPOD          ;czyta dane z PO
3115 03D9+D                50$   ;komunikacja zdalnie
3116 03D9+D 64 D9+D        jmp  50$
3117 03DB+D 00          nop
3118
-----

```



```

3191 044F+D A1          mov    @R1,A
3192 0450+D BA 00       mov    R2,# 0
3193 0452+D 24 81+D     jmp    SECH01
3194                                     ;-----
3195 0454+D 54$      ;KONF.POCZ
3196 0454+D 34 60+D     call   VLRSTP          ;sprawdzenie legalnosci
3197 0456+D C6 5A+D     jZ     low 1$
3198 0458+D E4 6F+D     jmp    NLEG           ;polecenie nielegalne
3199 045A+D 1$
3200 045A+D          movA   RKOLEJ
3203 045D+D 53 17       anl    A,# 00010111B
3204 045F+D C6 65+D     jz     low 73$
3205 0461+D          ;kolejka zadan nie jest pusta
3206 0461+D BA 0A       mov    R2,# 10        ;ERROR# 10 (0AH)
3207 0463+D 24 98+D     jmp    SECH03         ;zostanie wyslane potwierdzenie kodu =
3208 0465+D                                     ;komunikacja bledna, ktore oznacza
3209 0465+D                                     ;niedokonczenie albo niewylaczenie
3210 0465+D                                     ;zleconych do wykonania polecen.
3211 0465+D                                     ;Byloby wlasciwe uzycie innego potwier
3212 0465+D                                     ;-dzenia kodu (sygnalizacja bledu MPO)
3213 0465+D                                     ;
3214 0465+D 43 0B       73$      ori    A,# 00001000B   ;RKOLEJ(3)= 1, KONfiguracja
3215 0467+D A1          mov    @R1,A
3216 0468+D          movA   MPMS
3219 046B+D 53 BF       anl    A,# 10111111B   ;MPMS(6) := 0, konfiguracja nie okreslona
3220 046D+D A1          mov    @R1,A
3221 046E+D 23 A3       mov    A,# 10100011B   ;101__11B
3222 0470+D          movR   RDMPO
3225 0473+D 23 F2       mov    A,# 11110010B   ;b7 = 1 nie odebrane ECHO
3226 0475+D                                     ;b6 = 1 nie potwierdzone argumenty
3227 0475+D                                     ;b5 = 1 nie odebrane potwierdzenie SUMY
3228 0475+D                                     ;b4 = 1 pocz.
3229 0475+D                                     ;b3-0 = 0010 - KONF.
3230 0475+D          movR   K4
3233 0478+D BA 00       mov    R2,# 0
3234 047A+D 27          clr    A
3235 047B+D          movR   SUMA
3238 047E+D 24 85+D     jmp    CECH01         ;powrot do petli, wyslac ECHO
3239 0480+D 34 AF+D     call   VWRMPO         ;kontrola stanu WRMPO
3240 0482+D 23 8B       mov    A,# 10001000B   ;10--1000B ;do wyslania ECHO
3241 0484+D          movR   WRMPO
3244 0487+D 83          ret                    ;jmp    IWR
3245                                     ;-----
3246 0488+D          KODI3 ;K4=3      RSTRK / RSTRP
3247 0488+D          movA   A3          ;restore A3
3250 048B+D C6 93+D     jz     low 55$
3251 048D+D D3 01       xrl   A,# 1
3252 048F+D C6 97+D     jz     low 56$
3253 0491+D E4 6B+D     jmp    BLKOD         ;bledny arg. kodu
3254 0493+D 55$      ;RSTRK
3255 0493+D BA 09       mov    R2,# 9          ;ERROR# 9
3256 0495+D 24 98+D     jmp    SECH03
3257 0497+D          ;UWAGA: Odebranie sekwencji 'UWAGA' 'RSTRK' jest bledem.
3258                                     ;-----
3259 0497+D          56$      ;RSTRP
3260 0497+D 34 68+D     call   VLSTRP         ;sprawdzenie legalnosci
3261 0499+D C6 9D+D     jZ     low 1$
3262 049B+D E4 6F+D     jmp    NLEG           ;polecenie nielegalne
3263 049D+D 1$
3264 049D+D          movR   RKOLEJ
3267 04A0+D 53 3F       anl    A,# 00111111B

```



72


```

3268 04A2+D C6 AB+D          jz      low 72%
3269 04A4+D          ;kolejka zadan nie jest pusta
3270 04A4+D BA 00          mov     R2,# 8          ;ERROR# 8
3271 04A6+D 24 9B+D       jmp     SECH03         ;uwaga jak dla KONF.POCZ
3272 04A8+D
3273 04AB+D 43 02      72%   orl     A,# 00000010B   ;RKOLEJ(1):= 1
3274 04AA+D A1          mov     @R1,A
3275 04AB+D 23 A0       mov     A,# 10100000B   ;101----B
3276 04AD+D          movR    RDMPO
3279 04B0+D 23 F3       mov     A,# 11110011B   ;b7 = 1 nie odebrane ECHO potw. kodu
3280 04B2+D          ;      0   odebrane ECHO potw. kodu
3281 04B2+D          ;b6 = 1 spodziewane ZADANIE / RSTRK
3282 04B2+D          ;      0   odebrane
3283 04B2+D          ;b5 = 1 nie odebrany kod zestrojenia
3284 04B2+D          ;b4 = 1 pocz.
3285 04B2+D          ;b3-0 = 0011 - STR
3286 04B2+D          movR    KODI
3289 04B5+D BA 00       mov     R2,# 0
3290 04B7+D 24 B5+D     jmp     CECH01
3291 04B9+D 34 AF+D     call    VWRMPO
3292 04BB+D 23 80       mov     A,# 10001000B   ;WRMPO:= 10--1000B, do wyslania ECHO
3293 04BD+D          movR    WRMPO
3296 04C0+D 83         ret     ;jmp    INR
3297 04C1+D          ;UWAGA: W przyjetej konwencji nie ma mozliwosci zasygnalizowania
3298 04C1+D          ;      odebrania w ZADANIU bladnego numeru kanalu i numeru parametru.
3299          ;-----
3300 04C1+D          VTXRdy ;kontrola gotowosci nadajnika (komunikacja)
3301 04C1+D          movA    MPMS
3304 04C4+D 32 D1+D     jbi     low 50%
3305 04C6+D          ;komunikacja lokalnie
3306 04C6+D 94 D6+D     call    SIEXA          ;ustaw XINTEA:= 1
3307 04C8+D 34 04+D     call    GETPOS         ;odczytaj slowo stanu PO
3308 04CA+D 97         clr     C
3309 04CB+D 72 CE+D     jb3     low 10%       ;INTRA= 1
3310 04CD+D          ;XPC3 = INTRA = 0, bufor nadajnika nie odczytany
3311 04CD+D A7         cpl     C
3312 04CE+D 94 E9+D     10%    call    RIEXA          ;ustaw XINTEA:= 0
3313 04D0+D 83         ret     ;C = 0 - TxRdy, C = 1 - TxNRdy
3314 04D1+D          50%    ;komunikacja zdalnie
3315 04D1+D          outP1  43
3318 04D4+D E4 55+C     jmp     CLOOP
3319          ;-----
3320 04D6+D          SIEXA ;ustawia INTEA:=1 w RIX/RIX
3321 04D6+D B8 74       mov     R0,#PNTIO
3322 04D8+D          movA    MPMS
3325 04DB+D 32 E3+D     jbi     low 50%
3326 04DD+D          ;komunikacja lokalnie (PO)
3327 04DD+D B0 9B       mov     @R0,#RXP
3328 04DF+D 23 0D       mov     A,# 13
3329 04E1+D 24 2B+D     jmp     PUTIO
3330 04E3+D          50%    ;komunikacja zdalnie (RIR)
3331 04E3+D B0 93       mov     @R0,#RRP
3332 04E5+D 23 0D       mov     A,# 13
3333 04E7+D 24 2B+D     jmp     PUTIO
3334          ;-----
3335 04E9+D          RIEXA ;ustawia INTEA:=0 w RIX/RIX
3336 04E9+D B8 74       mov     R0,#PNTIO
3337 04EB+D          movA    MPMS
3340 04EE+D 32 F6+D     jbi     low 50%
3341 04F0+D          ;komunikacja lokalnie (PO)
3342 04F0+D B0 9B       mov     @R0,#RXP

```

```
3343 04F2+D 23 0C          mov  A, # 12 ***
3344 04F4+D 24 28+D        jmp  PUTIO
3345 04F6+D          50$ ;komunikacja zdalnie (RIR)
3346 04F6+D B0 93          mov  @R0, #RRP
3347 04F8+D 23 0C          mov  A, # 12
3348 04FA+D 24 28+D        jmp  PUTIO
3349                                     ;-----
```

B

74


```

3439 0557+D E5          sel    MB0
3440 0558+D D4 40+C     call   DSPRKP          ;wyswietlenie rejestrow KP
3441 055A+D F5          sel    MB1
3442                                else
3448 055B+D
3449 055B+D              movA   RKOLEJ
3452 055E+D 53 F7      anl    A,# 11110111B   ;RKOLEJ(3)= 0
3453 0560+D A1          mov    @R1,A
3454 0561+D 27          clr    A
3455 0562+D              movR   SUMA
3458 0565+D              ; movRD  RDNPD, E0H
3459 0565+D B9 6A      mov    R1,# RDNPD
3460 0567+D B1 E0      mov    @R1,# 11100000B
3461 0569+D              movA   INSTNT
3464 056C+D 43 10      orl    A,# 00010000B   ;INSTNT(4)= 1
3465 056E+D A4 D1+D    jmp    48$
3466                                ;.....
3467 0570+D              38$   ;przejscia (B1)
3468 0570+D 74 D2+D    call   GETRxD
3469 0572+D AA          mov    R2,A           ;store B1
3470 0573+D              movA   NK
3473 0576+D E7          rl     A
3474 0577+D 03 42      add    A,# KON + 2    ;pnt.to KONF.DATA, KP0A0
3475 0579+D 00          nop
3476 057A+D AB          mov    R0,A
3477 057B+D FA          mov    A,R2           ;restore B1
3478 057C+D 47          swap   A
3479 057D+D 53 0F      anl    A,# 15         ;przejscie# 0
3480 057F+D A0          mov    @R0,A
3481 0580+D 10          inc    R0
3482 0581+D FA          mov    A,R2           ;restore B1
3483 0582+D 53 0F      anl    A,# 15         ;przejscie# 1
3484 0584+D A0          mov    @R0,A
3485 0585+D A4 BE+D    jmp    45$
3486                                ;.....
3487 0587+D              39$   ;B0 (NK,TK)
3488 0587+D 74 D2+D    call   GETRxD
3489 0589+D F2 B0+D    jb7   low 44$         ;error
3490 058B+D AA          mov    R2,A           ;store B0
3491 058C+D 47          swap   A
3492 058D+D 53 07      anl    A,# 7         ;NK
3493 058F+D              movR   NK
3496 0592+D 03 10+C    add    A,# low TABOCT
3497 0594+D E3          movP3  A,@A
3498 0595+D AB          mov    R3,A           ;binary NK
3499 0596+D B8 40      mov    R0,# KON       ;pnt to KON
3500 0598+D 40          orl    A,@R0          ;wlaczenie kanalu
3501 0599+D A0          mov    @R0,A
3502 059A+D 10          inc    R0             ;pnt to KT
3503 059B+D FA          mov    A,R2
3504 059C+D 53 0F      anl    A,# 10001111B
3505 059E+D D3 01      xrl   A,# 00000010B
3506 05A0+D C6 BA+D    jz    low 42$         ;TK = KL256
3507 05A2+D FA          mov    A,R2
3508 05A3+D 53 0F      anl    A,# 10001111B
3509 05A5+D D3 02      xrl   A,# 00000010B
3510 05A7+D C6 B5+D    jz    low 43$         ;TK = KL1024
3511 05A9+D FA          mov    A,R2
3512 05AA+D 53 0F      anl    A,# 10001111B
3513 05AC+D D3 00      xrl   A,# 00000000B
3514 05AE+D C6 D9+D    jz    low 46$         ;TK = 0, wylaczyc kanal

```

76

```

3515 05B0+D
3516 05B0+D          ;blad danych
3517 05B0+D          44$   outP1  17          ;ERROR# 17
3520 05B3+D  E4 62+D   jmp    LOOPER
3521
;-----
3522 05B5+D  FB          43$   mov    A,R3
3523 05B6+D  40          ori    A,@R0          ;TK = KL1024, KT(NK):= 1
3524 05B7+D  A0          mov    @R0,A
3525 05B8+D  A4 BE+D    jmp    45$
3526
;-----
3527 05BA+D  FB          42$   mov    A,R3
3528 05BB+D  37          cpl   A
3529 05BC+D  50          anl   A,@R0          ;TK = KL256, KT(NK):= 0
3530 05BD+D  A0          mov    @R0,A
3531 05BE+D          45$   decR  RDMPO
3536 05C3+D          movA  RKOLEJ
3539 05C6+D  53 DF      anl   A,# 11011111B   ;RKOLEJ(5):= 0
3540 05C8+D  A1          mov    @R1,A
3541 05C9+D          movA  SUMA          ;
3544 05CC+D  6A          add   A,R2
3545 05CD+D  A1          mov    @R1,A
3546
;-----
3547 05CE+D          47$   movA  INSTNT
3550 05D1+D  53 DF      48$   anl   A,# 11011111B   ;INSTNT(5):= 0
3551 05D3+D  A1          mov    @R1,A
3552 05D4+D
3553          if um = 1
3554 05D4+D  14 C4+D    call  DDSRR          ;wyswietlenie rejestrow roboczych
3555          ;P1 NS WF RF RK IF PF KF KT WI K4 A3 KI SU
3556          ;
3557          else
3561 05D6+D
3562 05D6+D  E5          sel   MB0
3563 05D7+D  04 22+C    jmp   ENINT
3564
;-----
3565 05D9+D          46$   ;wylaczenie kanalu
3566 05D9+D  FB          mov    A,R3
3567 05DA+D  37          cpl   A
3568 05DB+D  C8          dec   R0          ;pnt to KON
3569 05DC+D  50          anl   A,@R0          ;wylaczenie kanalu
3570 05DD+D  A0          mov    @R0,A
3571 05DE+D  18          inc   R0          ;pnt to KT
3572 05DF+D  A4 BE+D    jmp    45$
3573
;-----

```

8


```

3650 065A+D E4 6F+D          jmp     NLEG          ;polecenie nielegalne
3651 065C+D          1$
3652 065C+D C4 6B+D          jmp     PDW          ;polecenie do wykonania
3653
-----
3654 065E+D          KODI6 ;K4=6      RQDKPL  !!! identyczne z 85$ ? !!!
3655 065E+D          movA   A3          ;restore A3
3658 0661+D C6 65+D          jz     low 62$
3659 0663+D E4 6B+D          jmp     BLKOD        ;bledny arg. kodu
3660 0665+D          62$ ;RQDKPL
3661 0665+D 34 70+D          call  VLRQP        ;sprawdzenie legalnosci
3662 0667+D C6 6B+D          jZ    low 1$
3663 0669+D E4 6F+D          jmp     NLEG          ;polecenie nielegalne
3664 066B+D          1$
3665 066B+D          PDW   ;Polecenie do wykonania
3666 066B+D          movR  RKOLEJ
3669 066E+D 43 20          orl   A,# 00100000B
3670 0670+D A1          mov   @R1,A
3671 0671+D          movA  INSTNT
3674 0674+D 43 05          orl   A,# 00000101B ;INSTNT(2,0)= 11
3675 0676+D A1          mov   @R1,A
3676 0677+D BA 00          3$   mov   R2,# 0
3677 0679+D 24 81+D          jmp   SECHO1
3678
-----
3679 067B+D          KODI7 ;K4=7      RQDNK
3680 067B+D          movA  A3          ;restore A3
3683 067E+D 34 7B+D          call  VERNK        ;czy tor kamery #NK włączony ?
3684 0680+D
3685 0680+D F6 84+D          jc    low 61$      ;zadany tor kamerowy - włączony
3686 0682+D E4 6B+D          jmp   BLKOD        ;C = KON(A3) = 0, - nie włączony
3687 0684+D          61$ ;RQDNK
3688 0684+D 34 70+D          call  VLRQP        ;sprawdzenie legalnosci
3689 0686+D C6 8A+D          jZ    low 1$
3690 0688+D E4 6F+D          jmp   NLEG          ;polecenie nielegalne
3691 068A+D          1$
3692 068A+D C4 6B+D          jmp   PDW          ;polecenie do wykonania
3693
-----
3694 068C+D          KODI8 ;K4=8      KOM.POPR / KOM.BL
3695 068C+D          movA  A3          ;restore A3
3698 068F+D C6 97+D          jz    low 64$
3699 0691+D D3 01          xrl  A,# 1
3700 0693+D C6 AC+D          jz    low 65$
3701 0695+D E4 6B+D          jmp   BLKOD        ;bledny arg. kodu
3702 0697+D          64$ ;KOM.POPR (KODI = C0H)
3703 0697+D          movA  WRMPO
3706 069A+D D3 00          xrl  A,# 0 ;?? 00000100B ;00--0100B ;oczek. potw. SUMY
3707 069C+D C6 AB+D          jz    low 68$      ;eq.
3708 069E+D          ;potwierdzenie SUMY nie oczekiwane
3709 069E+D BA 05          mov  R2,# 5      ;ERROR# 5
3710 06A0+D F4 44+D          71$ call  ERROR
3711 06A2+D 23 E0          69$ mov  A,# 11100000B ;111----B ;do odebrania 'UWAGA'
3712 06A4+D          movR  RDMPO      ;dane NIE oczekiwane
3715 06A7+D B3          ret   ;jmp   INR
3716 06A8+D B1 00          68$ mov  @R1,# 0      ;WRMPO:= 0
3717 06AA+D C4 A2+D          jmp   69$
3718
-----
3719 06AC+D          65$ ;KOM.BL (KODI = C1H)
3720 06AC+D          movA  WRMPO
3723 06AF+D D3 00          xrl  A,# 0 ;?? 00000100B ;00--0100B ;oczek. potw. SUMY
3724 06B1+D C6 B7+D          jz    low 70$
3725 06B3+D          ;potwierdzenie SUMY nie oczekiwane
3726 06B3+D BA 06          mov  R2,# 6      ;ERROR# 6

```



```

3727 06B5+D C4 A0+D      jmp 71$
3728 06B7+D BA 07      70$  mov R2,# 7      ;ERROR# 7
3729 06B9+D C4 A0+D      jmp 71$
3730
;-----
3731 06BB+D      KODIA ;K4=10 (A) - ZDALNIE / LOKALNIE
3732 06BB+D      movA A3      ;restore A3
3735 06BE+D C6 C6+D      jz low 66$
3736 06C0+D D3 01      xrl A,# 1
3737 06C2+D C6 CF+D      jz low 67$
3738 06C4+D E4 6B+D      jmp BLKOD      ;bledny arg. kodu
3739 06C6+D      66$ ;LOKALNIE      !!! Polecenie nie realizowane !!!
3740 06C6+D 34 6B+D      call VLSTRP    ;sprawdzenie legalnosci
3741 06C8+D C6 CC+D      jZ low 1$
3742 06CA+D E4 6F+D      jmp NLEG      ;polecenie nielegalne
3743 06CC+D      1$
3744 06CC+D 00      nop
3745 06CD+D E4 6B+D      jmp BLKOD
3746 06CF+D      67$ ;ZDALNIE      !!! Polecenie nie realizowane !!!
3747 06CF+D 34 5B+D      call VLNGPO    ;sprawdzenie legalnosci
3748 06D1+D C6 D5+D      jZ low 2$
3749 06D3+D E4 6F+D      jmp NLEG      ;polecenie nielegalne
3750 06D5+D      2$
3751 06D5+D 00      nop
3752 06D6+D E4 6B+D      jmp BLKOD      ;jmp SECHO1
3753
;-----

```



```
3832 0752+D 37          cpl    A
3833 0753+D F2 5A+D      jb7   low 1$          ;SW(7)=0
3834 0755+D D2 5C+D      jb6   low 2$          ;SW(6)=0
3835 0757+D              ;SW(7,6)=11
3836 0757+D F5          sel    MB1
3837 0758+D E4 5B+D      0$    jmp   0$
3838 075A+D              1$    ;SW(7,6)=0x
3839 075A+D 04 00        jmp   0                ;programowy RESTART
3840 075C+D              2$    ;SW(7,6)=10
3841 075C+D FA          mov    A,R2
3842 075D+D 3A          outl  P2,A            ;restore P2
3843 075E+D B3          3$    ret                    ;ignore error
3844
3845 ;-----
3845 ;Sygnalizacja bledow na wyjsci P1
3846 ;-----
3847 075F+D              ERR23
3848 075F+D              outP1 23
3851 0762+D              LOOPER
3852 0762+D
3853 if um = 1
3854 0762+D D4 00+C      call  DISPRR          ;wyswietlenie rejestow roboczych
3855 ;P1 NS WF RF RK IF PF KF KT WI K4 A3 KI SU
3856 ;
3857 else
3861 0764+D
3862 0764+D E4 64+D      1$    jmp   1$
3863 ;-----
3864 0766+D              ERR24
3865 0766+D              outP1 24
3868 0769+D E4 62+D      jmp   LOOPER
3869 ;-----
3870 076B+D              BLKOD ;bledna wartosc KODI
3871 076B+D BA 03        mov    R2,# 3
3872 076D+D 24 98+D      jmp   SECH03
3873 ;-----
3874 076F+D              NLEG ;Nielegalne polecenia MPO
3875 076F+D              outP1 16
3878 0772+D E4 62+D      jmp   LOOPER
3879 ;-----
```

M O D U L E .MAIN.:

CSEG 0777 bytes

DSEG 0774 bytes

3880 END

*** I D E N T I F I E R S ***

A3 = 006D (R) ACCSTO= 0077 (R) ACLOS= 04D4+C AOUTL = 0002 (R)
ARG01 = 0072 (R) ARG02 = 0071 (R) BCD10A= 0200+C BCD10B= 023A+C
BCD4AR= 02B9+C BCD4AS= 0204+C BCD4AX= 029C+C BCDALL= 02D6+C
BLKOD = 076B+D BUFORD= 0052 (R) CECH01= 0105+D CGETI = 070A+C
CGETIO= 070D+C CGETRQ= 0700+C CGETX = 070F+C CGETXM= 0737+C
C6TPOD= 0708+C C6TPOS= 0704+C C6TXML= 072F+C CIBTMP= 0064 (R)
CLOOP = 0755+C CLRXE = 06B1+C CNTINT= 0007+C CPTPOD= 0722+C
CPTTXD= 0750+C CPUTIO= 0728+C CPUTX = 072C+C CPUTXM= 073B+C
CRIEXA= 0755+C CRSTP2= 0710+C CRXRDY= 074B+C CSETIO= 0717+C
CSETXM= 0741+C CSIEXA= 0755+C CTXRDY= 0746+C DADDT = 027B+C
DATPTC= 0332+C DDSPRR= 00C4+D DINH = 0003 (R) DINL = 0004 (R)
DISPRR= 0600+C DIVPNT= 0065 (R) DPR653= 00BF+D DSPRKD= 0650+C
DSPRKP= 0640+C ENINT = 0022+C ERR23 = 075F+D ERR24 = 0766+D
ERROR = 0744+D EXTINT= 0003+C GETI = 010A+D GETIO = 010D+D
GETPOD= 0108+D GETPOS= 0104+D GETRQ = 0100+D GETRXD= 03D2+D
GETX = 010F+D GETXM = 0137+D GETXML= 012F+D HEXTAB= 0300+C
IIB = 0405+C IIK2V1= 052B+C IIRV1 = 04E3+C IISV1 = 04E8+C
INI51 = 060A+C INIALL= 0000+D INSTNT= 0066 (R) INTCNT= 04E3+C
INTERR= 04DE+C INTEXT= 0400+C IPOV1 = 0534+C IRD = 0300+D
IWR = 0200+D JRQSV1= 0422+C JRQSV2= 044D+C JRQSV3= 0479+C
JRQSV4= 04A5+C JRQSV5= 04D1+C JRQSV6= 04D1+C JRQSV7= 04D1+C
JRQSV8= 04D1+C K4 = 006E (R) K4.2 = 0500+D K4.3 = 0700+D
KA0LO = 0020 (R) KODI = 006C (R) KODI0 = 0400+D KODI1 = 0429+D
KODI2 = 0439+D KODI3 = 0408+D KODI4 = 0600+D KODI5 = 064F+D
KODI6 = 065E+D KODI7 = 067B+D KODI8 = 068C+D KODIA = 06BB+D
KODD = 0070 (R) KON = 0040 (R) KONTMP= 0061 (R) KPA0 = 0042 (R)
KS = 0067 (R) KT = 0041 (R) KX20 = 00C0 (R) KY20 = 00D0 (R)
KZ20 = 00E0 (R) LARGD = 006F (R) LOOPER= 0762+D LOS = 0075+D
LTK = 0003 (R) M = 0001 (R) M0IAL= 009B (R) M1111 = 00BE (R)
M1110 = 00B6 (R) M1101 = 00BC (R) M1100 = 00B4 (R) M1011 = 00AE (R)
M1010 = 00A6 (R) M1001 = 00AC (R) M1000 = 00A4 (R) MASKA1= 00FD (R)
MASKA2= 00BE (R) MASKA3= 00FE (R) MASKA4= 00BD (R) MASKA5= 0000 (R)
MASKA6= 0000 (R) MASKA7= 0000 (R) MASKA8= 0000 (R) MPMS = 0068 (R)
MSCTMP= 0060 (R) NK = 0071 (R) NLEG = 076F+D P1TMP = 0062 (R)
P2TMP = 0063 (R) PDW = 066B+D PNTARG= 007B (R) PNTDAT= 007C (R)
PNTIO = 0074 (R) PR651D= 033B+C PR653 = 066D+C PRNS = 006B (R)
PUTIO = 0120+D PUTPOD= 0122+D PUTTXD= 01D8+D PUTX = 012C+D
PUTXM = 013B+D R = 0000 (R) R3TOX1= 0668+C R3TOX2= 0664+C
R3TOX3= 0660+C RDIK2 = 012A+C RDMPD = 006A (R) REINIC= 000C+C
RERR = 007F (R) RESA = 0020 (R) RESET = 0000+C RESTP2= 0110+D
RETRY = 03B6+D RETSV1= 0448+C RETSV2= 0474+C RETSV3= 04A0+C
RETSV4= 04CC+C RGOTPO= 0000 (R) RIEXA = 04E9+D RIKSV1= 04ED+C
RIOSVC= 04D9+C RISSV1= 04ED+C RISTRO= 0322+C RIXSV1= 055B+C
RKOLEJ= 007E (R) RKOMBL= 00C1 (R) RKOMOK= 00C0 (R) RKONFK= 0090 (R)
RKONFP= 0091 (R) RLOK = 00D0 (R) RMT0XL= 0767+C RMT0XN= 076E+C
RNGPO = 0081 (R) RPA = 0090 (R) RPB = 0091 (R) RPC = 0092 (R)
RQA = 0024 (R) RQMASK= 007A (R) RQTMP = 0079 (R) RRP = 0093 (R)
RRQKPL= 00B0 (R) RRQRST= 00B8 (R) RRQSK = 00A8 (R) RSTRK = 0098 (R)
RSTRP = 0099 (R) RTROFF= 00A0 (R) RTRONA= 00A1 (R) RTRONB= 00A2 (R)
RUN1 = 0012+C RUNIK2= 0100+C RXA = 0098 (R) RXB = 0099 (R)
RXC = 009A (R) RXP = 009B (R) RZDAL = 00D1 (R) S = 0001 (R)
SC0 = 0000 (R) SCI = 00B1 (R) SC2 = 00B2 (R) SCC = 0083 (R)
SECH01= 0181+D SECH03= 0198+D SETIOA= 0117+D SETXM = 0141+D
SIC = 0085 (R) SID = 00B4 (R) SIEXA = 04D6+D SIK2 = 0759+C
SMDATA= 0564+C SNDKN = 06BA+C SNDSK = 06D6+C SUMA = 0073 (R)

SWA = 0020 (R)	SYMKD = 035E+C	TABOCT= 0310+C	TABRDJ= 0318+C
TARIK2= 0343+C	TC0 = 0028 (R)	TC0CW = 0036 (R)	TC0HID= 0000 (R)
TC0LOD= 0064 (R)	TC1 = 0029 (R)	TC1CW = 0034 (R)	TC1HID= 0000 (R)
TC1LOD= 0019 (R)	TC2 = 002A (R)	TC2CW = 0034 (R)	TC2HID= 0000 (R)
TC2LOD= 0019 (R)	TCC = 002B (R)	TGOTPM= 0084 (R)	TKOMBL= 00C5 (R)
TKOMOK= 00C4 (R)	TKONFK= 0094 (R)	TKONFP= 0095 (R)	TLOK = 00D4 (R)
TNGPM = 0085 (R)	TRDAT = 0326+C	TRORST= 008C (R)	TSK = 00AC (R)
TSTSV1= 0429+C	TSTSV2= 0454+C	TSTSV3= 0480+C	TSTSV4= 04AC+C
TTRBL = 00CC (R)	TZDAL = 00D5 (R)	UM = 0001 (R)	UWAGA = 00FF (R)
VERNK = 017B+D	VKOLEJ= 002B+C	VLGOTP= 0150+D	VLNGPO= 0158+D
VLRQP = 0170+D	VLRSTP= 0160+D	VLSTRP= 0168+D	VRKONF= 0500+C
VRXRDY= 03C1+D	VTXRDY= 04C1+D	VWRMPO= 01AF+D	WERINT= 0078 (R)
WPD = 0002 (R)	WRMPO = 0069 (R)	XRAMP6= 007D (R)	ZNAR = 00BA+D