PDW MARTHEL S.C. ul. Sosnowa 24-5, Bielany Wrocławskie
55-040 Kobierzyce, POLAND
tel. +48 71 311 07 11,12 fax. +48 71 311 07 13

WSPÓŁPRACA UKŁADÓW ISD33XXX/4002/4003 Z MIKROKONTROLERAMI WINBOND.

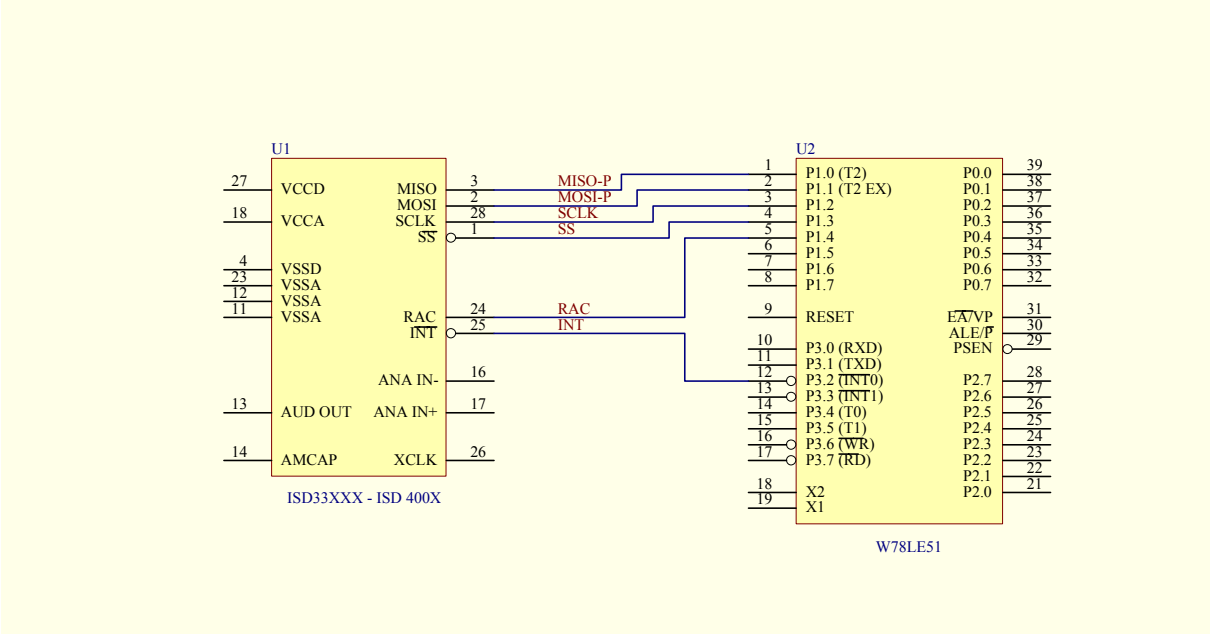
PDW MARTHEL S.C. ul. Sosnowa 24-5, Bielany Wrocławskie
55-040 Kobierzyce, POLAND
tel. +48 71 311 07 11,12 fax. +48 71 311 07 13

1. Wstęp.

Niniejsza nota aplikacyjna została stworzona w celu ułatwienia naszym klientom projektowania i szybkiego uruchamiania systemów mikroprocesorowych z wykorzystaniem układów ISD opartych na procesorach Winbond z serii 80C51. Przedstawiony tu kod programu zawiera podstawowe procedury wymiany danych pomiędzy układem ISD a mikrokontrolerem poprzez interfejs SPI oraz kontrolowania trybu pracy układu ISD. Każdy użytkownik może swobodnie wykorzystywać i modyfikować do własnych potrzeb opisane poniżej fragmenty programu bądź jego całość.
Opisany program, ze względu na dwubajtowe słowo wpisywane do ISD, może być stosowany do następujących układów:

- **ISD 33060 - 33240**
- **ISD 4002 – 4003**.

2. Połączenia elektryczne pomiędzy mikrokontrolerem i układem ISD.



Rys. 1. Schemat połączeń pomiędzy układem ISD i mikrokontrolerem.

Interfejs SPI wymaga doprowadzenia do układu ISD czterech linii:

- **MISO** *("Master in Slave out")* - dane przesyłane z układu ISD pracującego jako Slave do mikrokontrolera pracującego jako Master,
- **MOSI** *("Master out Slave in")* dane przesyłane z mikrokontrolera do układu ISD,
- **SCLK** *("Serial Clock")* sygnał taktujący port SPI wytwarzany przez mikrokontroler,
- **SS** *("Slave Select")* sygnał aktywujący port SPI układu ISD.

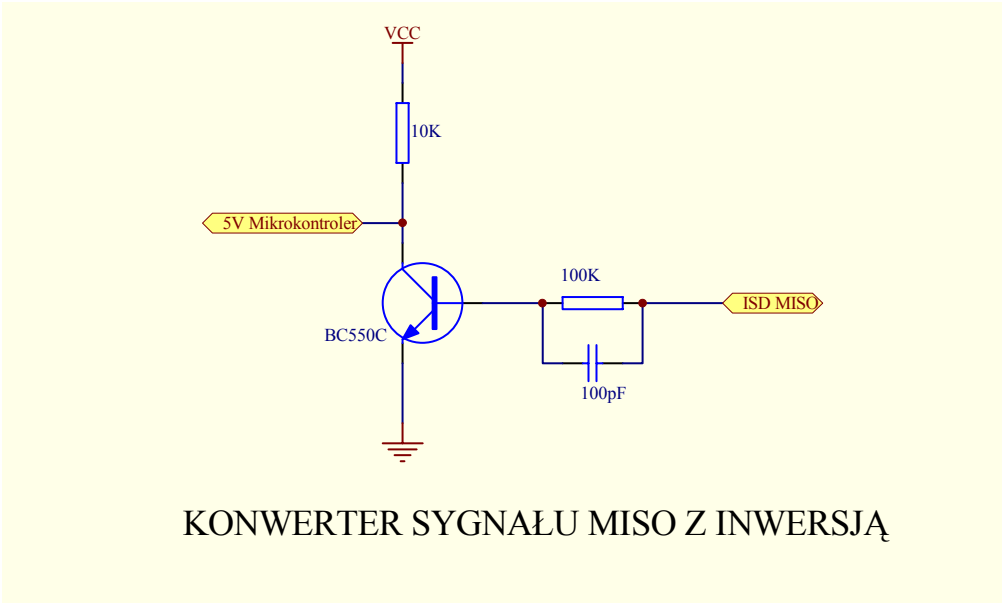
Połączenia dodatkowe układu ISD:

- **RAC** ***("Row Address Clock")*** sygnał reprezentujący jeden wiersz pamięci ISD. Przejście do poziomu niskiego sygnalizuje zbliżanie się końca odtwarzanego lub zapisywanego wiersza. Wyjście typu otwarty dren.
- **INT** ***("Interrupt")*** Sygnał przyjmuje poziom niski po napotkaniu przez układ ISD znacznika końca komunikatu (**EOM -** ***"End of Message"***) lub po przekroczeniu dostępnego zakresu pamięci ISD (**OVF -** ***"Overflow"***). Wyjście typu otwarty dren.

PDW MARTHEL S.C. ul. Sosnowa 24-5, Bielany Wrocławskie
55-040 Kobierzyce, POLAND
tel. +48 71 311 07 11,12 fax. +48 71 311 07 13

Sygnał INT może być skasowany następną sekwencją odczytu portu SPI. Status flag wywołujących przerwanie może być odczytany instrukcją ***"RINT"***.

W powyższym przykładzie mikrokontroler oraz układ ISD pracują z takim samym poziomem napięcia zasilającego tj. **2,7 – 3,3 V**. W przypadku, gdy mikrokontroler zasilany jest napięciem **5 V** należy zastosować odpowiedni konwerter napięć na linii **MISO** opisany w ***ChipCorder* Data Book**.



Rys. 2. Konwerter poziomu napięć pomiędzy mikrokontrolerem a układem ISD.

Stosowanie konwertera przedstawionego na rysunku 2 powinno być połączone z negowaniem bitów odbieranych przez mikrokontroler z portu **MISO** (inwersja).
Wszystkie pozostałe porty mogą być bezpośrednio połączone z mikrokontrolerem zasilanym napięciem **+5 V**.

3. Start od podanego adresu.

Rozpoczęcie odtwarzania bądź nagrywania od podanego adresu pamięci ISD rozpoczyna się wpisaniem polecenia **SETPLAY (SETREC)**. Narastające zbocze sygnału **SS** powoduje przepisanie adresu z rejestru MOSI do wewnętrznego licznika wierszy. Odtwarzany/nagrywany jest tylko bieżący zaadresowany wiersz pamięci ISD. Po dojściu do końca bieżącego wiersza układ zatrzymuje się i generuje przerwanie ze znacznikiem OVF. Jeżeli chcemy kontynuować odtwarzanie/nagrywanie poprzez kolejne wiersze pamięci ISD, należy przed końcem odtwarzania/nagrywania bieżącego wiersza wpisać do układu polecenie **PLAY (RECORD)**.
Przykład fragmentu kodu programu wywołującego operację PLAY od podanego adresu poprzez kolejne wiersze pamięci ISD:

```
1.      MOV   A,#SETPLAY   ;polecenie "PLAY" od podanego adresu
2.      LCALL POLEC        ;złożenie polecenia
3.      LCALL SPI          ;wysłanie polecenia wraz z adresem do ISD
4.      MOV   A,#PLAY      ;kontynuacja "PLAY" until eom or ovf
5.      LCALL POLEC        ;złożenie danych dla ISD
6.      LCALL SPI          ;zapis/odczyt ISD
```

PDW MARTHEL S.C. ul. Sosnowa 24-5, Bielany Wrocławskie
55-040 Kobierzyce, POLAND
tel. +48 71 311 07 11,12 fax. +48 71 311 07 13

- Wiersz 1: do akumulatora wpisywany jest kod polecenia **"SETPLAY"** zdefiniowany jako **11100000B**.
- Wiersz 2: wywołana procedura **"POLEC"** korzystając z zawartości akumulatora składa ostateczną postać dwóch bajtów wysyłanych do ISD. Młodszy bajt zawiera młodszą część adresu pamięci ISD. Starszy bajt zawiera sumę logiczną starszej części adresu i kodu polecenia ISD z akumulatora. Zawartość podprogramu **"POLEC"** znajduje się na końcu niniejszego opracowania.
- Wiersz 3: procedura komunikacyjna **"SPI"** poprzez symulację na portach mikrokontrolera interfejsu SPI dokonuje wymiany danych pomiędzy mikrokontrolerem a układem ISD. Bajty wpisywane do ISD: MOSI (H) i MOSI+1 (L). Bajty odbierane z ISD: MISO (H) i MISO+1 (L). Treść procedury dostępna jest w przytoczonym przykładzie na końcu opracowania.

  Uwaga ! Teraz układ ISD znajduje się w trybie odtwarzania jednego wiersza. Jeżeli chcemy kontynuować odtwarzanie poprzez kolejne wiersze pamięci musimy przed końcem bieżącego wiersza wpisać polecenie **"PLAY".**

- Wiersz 4: analogicznie do wiersza nr 1: kod polecenia **"PLAY" 11110000B**.
- Wiersz 5 i 6: analogicznie do wiersza 2 i 3 złożenie i wysłanie słowa dwubajtowego do ISD.

4. Zatrzymanie/wyłączenie układu.

Przytoczony fragment programu przedstawia sekwencję zatrzymania układu ISD, która składa się z trzech części: zatrzymanie/pauza bieżącej operacji (**"STOP"**), odczytanie adresu po zatrzymaniu układu oraz wyłączenie układu (**"STOPPWRDN"**).

```
1.      MOV   A,#STOP          ;polecenie "STOP"
2.      LCALL POLEC            ;złożenie polecenia
3.      LCALL SPI              ;wysłanie polecenia do ISD
4.      LCALL TIME             ;odczekanie ok.100 ms na zatrzymanie układu
5.      MOV   A,#STOPPWRDN     ;polecenie "STOP" i usypianie
6.      LCALL POLEC            ;złożenie polecenia
7.      LCALL SPI              ;odczytanie adresu po zatrzymaniu i usypianie
```

- Wiersz 1: do akumulatora wpisywany jest kod polecenia **"STOP"** zdefiniowany jako **00110000B**.
- Wiersz 2 i 3: analogicznie do poprzedniego punktu.
- Wiersz 4 : wywołanie procedury "TIME" powoduje wygenerowanie opóźnienia w programie podczas oczekiwania na zatrzymanie układu ISD.
- Wiersz 5, 6 i 7: wysłanie polecenia **"STOP POWER DOWN"** połączone jest z odczytaniem adresu pamięci ISD po zatrzymaniu układu.

PDW MARTHEL S.C. ul. Sosnowa 24-5, Bielany Wrocławskie
55-040 Kobierzyce, POLAND
tel. +48 71 311 07 11,12 fax. +48 71 311 07 13

5. Zmiana adresu odtwarzania/nagrywania w trybie RUN.

W niektórych aplikacjach istnieje potrzeba skoku do innego (nie następnego) adresu pamięci ISD podczas trwania odtwarzania lub nagrywania komunikatu. Kolejność wykonywanych kroków jest następująca:

a) przed końcem odtwarzania/nagrywania bieżącego wiersza pamięci ISD wpisujemy instrukcję **"SETPLAY"** lub **"SETREC"** z adresem wiersza w którym chcemy dalej kontynuować odtwarzanie/nagrywanie. Układ ISD nie wykona natychmiastowego skoku lecz będzie do końca odtwarzał/nagrywał wiersz bieżący. Mikrokontroler w tym czasie odczytuje stan na wyjściu **RAC**. Pojawienie się poziomu niskiego sygnalizuje zbliżanie się końca bieżącego wiersza. Powrót do stanu wysokiego sygnału **RAC** oznacza koniec bieżącego wiersza i wykonanie skoku pod nowy adres w pamięci ISD.

b) po przejściu do odtwarzania/nagrywania nowego wiersza, jeżeli chcemy by układ odtwarzał/nagrywał kolejne wiersze pamięci ISD, to przed końcem nowego wiersza wpisujemy polecenie **"PLAY"** lub **"REC"**. Jeżeli chcemy wykonać kolejny skok w pamięci ISD to powtarzamy czynności z punktu a.

4. Przykładowy program.

W poniższym programie wykorzystane zostały przykłady opisane powyżej z wyjątkiem zmiany adresu w trakcie odtwarzania/nagrywania, która nie była tutaj wykorzystywana. Program, w pętli głównej **"PETLA:",** sprawdza zawartość zmiennej **ROZKAZ** i wykonuje odpowiednie sekwencje poleceń na układzie ISD. Podprogram **SPI** służący do wymiany danych pomiędzy mikrokontrolerem może być wykorzystany tylko w układzie bez odwracającego konwertera sygnału MISO (ISD i mikrokontroler z takim samym napięciem zasilającym). Na końcu przedstawiona została procedura **SPI_K** dla układu z odwracającym konwerterem sygnału MISO (rys.2).

```
;deklaracje stałych:

POWERUP     EQU   00100000B      ;kody poleceń dla ISD
SETPLAY     EQU   11100000B
PLAY        EQU   11110000B
SETREC      EQU   10100000B
REC         EQU   10110000B
SETMC       EQU   11101000B
MC          EQU   11111000B
STOP        EQU   00110000B
STOPPWRDN   EQU   00010000B
RINT        EQU   00110000B

;polecenia dla pętli głównej programu sprawdzane w zmiennej ROZKAZ

ISDPLAY     EQU   01    ;odtwarzanie od podanego adresu
ISDRECORD   EQU   02    ;nagrywanie od podanego adresu
ISDSTOP     EQU   03    ;zatrzymanie układu
ISDMC       EQU   04    ;przewijanie od podanego adresu
```

PDW MARTHEL S.C. ul. Sosnowa 24-5, Bielany Wrocławskie
55-040 Kobierzyce, POLAND
tel. +48 71 311 07 11,12 fax. +48 71 311 07 13

```
;wykorzystywane porty I/O mikrokontrolera

SS              EQU   P1.3    ;Slave Select
SCLK            EQU   P1.2    ;Clock
MOSI_P          EQU   P1.1    ;Master Out Slave In
MISO_P          EQU   P1.0    ;Master In Slave Out
RAC             EQU   P1.4    ;sygnał RAC

;segment danych:

dseg at 30h

MISO:           DS 2    ;bajty odczytane z ISD H-L
MOSI:           DS 2    ;bajty wpisywane do ISD H-L
ADRESISD:       DS 2    ;aktualny (odczytywany) adres  ISD H-L
ADRES:          DS 2    ;adres zapisywany do ISD H-L
RACBLOK:        DS 1    ;blokuje wielokrotny odczyt RAC w tym samym cyklu RAC
ROZKAZ:         DS 1    ;kod polecenia dla pętli głównej programu
TRYB:           DS 1    ;identyfikacja aktualnego trybu pracy ISD (ostatnio wykonane polecenie)

cseg at 0

LJMP                    MAIN    ;skok do pętli głównej programu

cseg at 03h

LJMP    INT0_SERV       ;obsługa przerwania INT0

MAIN:

;Pętla główna programu
;---------------------------------------------------------------------
PETLA:          MOV     A,ROZKAZ

SPR1:           CJNE    A,#ISDPLAY,SPR2       ;sprawdzenie rozkazu
                MOV     TRYB,A          ;ISD w trybie PLAY
                LCALL   WAKEUP          ;budzenie układu ISD
                MOV     A,#SETPLAY      ;polecenie "PLAY" od podanego adresu
                LCALL   POLEC           ;złożenie polecenia
                LCALL   SPI             ;wysłanie polecenia
                MOV     A,#PLAY         ;kontynuacja "PLAY" until eom or ovf
                LCALL   POLEC           ;złożenie danych dla ISD
                LCALL   SPI             ;zapis/odczyt ISD
                MOV     ROZKAZ,#0       ;kasujemy polecenie już wykonane
                JMP     PETLA

SPR2:           CJNE    A,#ISDRECORD,SPR3
                MOV     TRYB,A          ;ISD w trybie RECORD
                LCALL   WAKEUP
                MOV     A,#SETREC       ;nagrywanie od podanego adresu
                LCALL   POLEC
                LCALL   SPI
                MOV     A,#REC          ;kontynuacja "REC"
                LCALL   POLEC
                LCALL   SPI
                MOV     ROZKAZ,#0
                JMP     PETLA

SPR3:           CJNE    A,#ISDSTOP,SPR4
                MOV     TRYB,A          ;ISD w trybie STOPPWRDN
```

PDW MARTHEL S.C. ul. Sosnowa 24-5, Bielany Wrocławskie
55-040 Kobierzyce, POLAND
tel. +48 71 311 07 11,12 fax. +48 71 311 07 13

```
                MOV   A,#STOP       ;polecenie "STOP"
                LCALL POLEC
                LCALL SPI
                LCALL TIME          ;odczekanie ok.100 ms na zatrzymanie układu
                MOV   A,#STOPPWRDN
                LCALL POLEC         ;odczytanie adresu po zatrzymaniu i usypianie
                LCALL SPI
                LCALL ADRISD        ;przesuwanie odczytanych danych
                MOV   ROZKAZ,#0
                JMP   PETLA         ;powrót do pętli
```

;Przewijanie poprzez kolejne komunikaty nagrane w ISD. Układ ISD zatrzymuje się napotkawszy marker EOM ;lub OVF i generuje przerwanie.

```
SPR4:           CJNE  A,#ISDMC,SPR5
                MOV   A,TRYB        ;sprawdzenie trybu pracy ISD
                CLR   C
                SUBB  A,#ISDMC      ;czy MC drugi raz z kolei ?
                JZ    SPR4_SK       ;jeśli tak to tylko kontynuacja przewijania
                MOV   TRYB,#ISDMC   ;zapamiętanie trybu pracy ISD
                LCALL WAKEUP
                MOV   A,#SETMC      ;włączenie przewijania od podanego adresu
                LCALL POLEC
                LCALL SPI
SPR4_SK:
                MOV   A,#MC         ;kontynuacja przewijania
                LCALL POLEC
                LCALL SPI
                MOV   ROZKAZ,#0
                JMP   PETLA
```

;Sprawdzanie sygnału RAC oraz inkrementacja aktualnego adresu pamięci ISD:

```
SPR5:
                JNB   RAC,SPR6      ;zniesienie blokady po zakończeniu impulsu RAC
                MOV   RACBLOK,#0
                JMP   PETLA

SPR6:           MOV   A,TRYB        ;sprawdzenie trybu pracy ISD
                CJNE  A,#ISDMC,POM  ;jeżeli impulsy RAC przewijania, to nie zliczam
                JMP   PETLA
POM:            MOV   A,RACBLOK
                JNZ   W             ;podprogram robimy tylko 1 raz w jednym cyklu RAC
                INC   RACBLOK
                MOV   A,ADRESISD+1  ;inkrementacja adresu ISD
                ADD   A,#1
                MOV   ADRESISD+1,A
                MOV   A,ADRESISD
                ADDC  A,#0          ;dodanie przeniesienia
                MOV   ADRESISD,A

W:              JMP   PETLA

WAKEUP:         MOV   A,#POWERUP    ;budzenie układu ISD
                LCALL POLEC
                LCALL SPI
                LCALL TIME
                RET
```

PDW MARTHEL S.C. ul. Sosnowa 24-5, Bielany Wrocławskie
55-040 Kobierzyce, POLAND
tel. +48 71 311 07 11,12 fax. +48 71 311 07 13

```
;generacja opóźnienia ok. 100 ms

TIME:   MOV   R7,#0
        MOV   R6,#180
SKK:    DJNZ  R7,$          ;256*2*180*1.085 us; Q=11.0592 MHz
        DJNZ  R6,SKK        ;nie licząc reszty
        RET

;obsługa przerwania INT0 :

INT0_SERV:
        PUSH  ACC
        MOV   A,#RINT
        LCALL POLEC
        LCALL SPI           ;odczytanie flagi przerwania
        LCALL ADRISD        ;przesuwanie odczytanego adresu ISD
        MOV   A,MISO+1      ;flagi na najniższych pozycjach
        JNB   ACC.1,OVF     ;skok jeśli przepełnienie pamięci układu
        POP   ACC
        RETI

OVF:    MOV   TRYB,#0               ;skasowanie trybu pracy ISD
        MOV   A,#STOPPWRDN          ;usypianie
        LCALL POLEC
        LCALL SPI
        POP   ACC
        RETI

;przesuwanie adresu ISD w celu pozbycia się bitów EOM i OVF z adresu:
;wykorzystane rejestry: R4, R5, R7, ACC

ADRISD: CLR   C
        MOV   R7,#2
        MOV   R4,MISO
        MOV   R5,MISO+1
PRZES:  MOV   A,R4          ;odzyskanie adresu ISD
        RRC   A
        MOV   R4,A
        MOV   A,R5
        RRC   A
        MOV   R5,A
        DJNZ  R7,PRZES
        MOV   A,R4
        CLR   ACC.7         ;skasowanie bitu przepełnienia
        CLR   ACC.6         ;skasowanie bitu OVF
        MOV   R4,A
        MOV   ADRESISD,R4       ;H
        MOV   ADRESISD+1,R5     ;L
        RET

;komunikacja z ISD:

;ZAWARTOŚĆ POSZCZEGÓLNYCH REJESTRÓW:
;R2 -> MOSI (H)
;R3 -> MOSI+1 (L)
;R4 -> MISO (H)
;R5 -> MISO+1 (L)
;ZAWARTOŚĆ REJESTRÓW R2, R3, R4, R5, R7 ZOSTAJE ZMIENIONA PO WYKONANIU
;PROCEDURY
```

PDW MARTHEL S.C. ul. Sosnowa 24-5, Bielany Wrocławskie
55-040 Kobierzyce, POLAND
tel. +48 71 311 07 11,12 fax. +48 71 311 07 13

```
SPI:        MOV   R2,MOSI       ;dane do wysłania
            MOV   R3,MOSI+1

            CLR   SCLK
            MOV   R7,#8
            CLR   SS            ;aktywacja układu
BAJT_L:
            CLR   SCLK
            MOV   A,R5          ;MISO+1 (L)
            MOV   C,MISO_P
            RRC   A
            MOV   R5,A          ;MISO+1 (L)
            MOV   A,R3          ;MOSI+1 (L)
            RRC   A
            MOV   MOSI_P,C
            MOV   R3,A          ;MOSI+1 (L)
            SETB  SCLK
            DJNZ  R7,BAJT_L
            MOV   R7,#8
BAJT_H:
            CLR   SCLK
            MOV   A,R4          ;MISO (H)
            MOV   C,MISO_P
            RRC   A
            MOV   R4,A          ;MISO (H)
            MOV   A,R2          ;MOSI (H)
            RRC   A
            MOV   MOSI_P,C
            MOV   R2,A          ;MOSI (H)
            SETB  SCLK
            DJNZ  R7,BAJT_H
            SETB  SS            ;koniec transmisji

            MOV   MISO,R4       ;dane odebrane
            MOV   MISO+1,R5
            RET

;przygotowanie danych dla ISD:

;W AKUMULATORZE KOD POLECENIA
;W ADRES I ADRES+1 ADRES ISD DO ZAPISU
;WYNIK UMIESZCZONY W MOSI (HIGH)  MOSI+1 (LOW)

POLEC:      MOV   MOSI+1,ADRES+1    ;młodszy bajt polecenia
            ORL   A,ADRES           ;dodanie kodu polecenia do starszego bajtu adresu ISD
            MOV   MOSI,A
            MOV   ADRESISD,ADRES
            MOV   ADRESISD+1,ADRES+1    ;zapamiętuje aktualny adres ISD
            RET


;komunikacja z ISD (inwersja sygnału MISO):

;ZAWARTOŚĆ POSZCZEGÓLNYCH REJESTRÓW:
;R2 -> MOSI (H)
;R3 -> MOSI+1 (L)
;R4 -> MISO (H)
;R5 -> MISO+1 (L)
;ZAWARTOŚĆ REJESTRÓW R2, R3, R4, R5, R7 ZOSTAJE ZMIENIONA PO WYKONANIU
;PROCEDURY
```

PDW MARTHEL S.C. ul. Sosnowa 24-5, Bielany Wrocławskie
55-040 Kobierzyce, POLAND
tel. +48 71 311 07 11,12 fax. +48 71 311 07 13

```
SPI_K:      MOV   R2,MOSI       ;dane do wysłania
            MOV   R3,MOSI+1

            CLR   SCLK
            MOV   R7,#8
            CLR   SS            ;aktywacja układu
BAJT_L:
            CLR   SCLK
            MOV   A,R5          ;MISO+1 (L)
            MOV   C,MISO_P
            CPL   C             ;korekcja inwersji
            RRC   A
            MOV   R5,A          ;MISO+1 (L)
            MOV   A,R3          ;MOSI+1 (L)
            RRC   A
            MOV   MOSI_P,C
            MOV   R3,A          ;MOSI+1 (L)
            SETB  SCLK
            DJNZ  R7,BAJT_L
            MOV   R7,#8
BAJT_H:
            CLR   SCLK
            MOV   A,R4          ;MISO (H)
            MOV   C,MISO_P
            CPL   C             ;korekcja inwersji
            RRC   A
            MOV   R4,A          ;MISO (H)
            MOV   A,R2          ;MOSI (H)
            RRC   A
            MOV   MOSI_P,C
            MOV   R2,A          ;MOSI (H)
            SETB  SCLK
            DJNZ  R7,BAJT_H
            SETB  SS            ;koniec transmisji

            MOV   MISO,R4       ;dane odebrane
            MOV   MISO+1,R5
            RET
```