Instytut Teleinformatyki

Wydział Fizyki, Matematyki i Informatyki
Politechnika Krakowska

Mikroprocesory i mikrokontrolery

„Obsługa portu szeregowego”

laboratorium: 05

autor: mgr inż. Michal Lankosz
dr hab. Zbisław Tabor, prof. PK

Kraków, 2014

Spis treści

1. Wiadomości wstępne

Pierwsza część niniejszej instrukcji zawiera podstawowe wiadomości teoretyczne dotyczące omawianego tematu. Poznanie tych wiadomości umożliwi prawidłowe zrealizowanie praktycznej części laboratorium.

1.1. Niezbędne wiadomości

- Pamięć programu, wewnętrzna pamięć RAM.mapa pamięci, rejestry, tryby adresowania: http://www.8052.com/tut8051.
- Lista instrukcji asemblera 8051: http://www.keil.com/support/man/docs/is51/is51_instructions.htm
- Obsługa liczników
- Obsługa przerwań
- Obsługa portu szeregowego

1.2. Układ UART

Układ UART jest przeznaczony do prowadzenia transmisji szeregowej. Do transmisji szeregowej wykorzystywane są dwie linie, przypisane do konkretnych linii portu P3:
linia nadawcza – P3.1
linia odbiorcza – P3.0

Przed użyciem portu szeregowego należy go najpierw skonfigurować poprzez odpowiednie ustawienie bitów rejestru SCON. Znaczenie poszczególnych bitów rejestru SCON podano w **Tabeli 1**.

Tabela 1 Bity rejestru SCON

Bit	Nazwa	Adres	Funkcja
7	SM0	9Fh	Tryb pracy portu - bit 0
6	SM1	9Eh	Tryb pracy portu - bit 1
5	SM2	9Dh	Maskowanie odbioru znaku; zmieniany programowo; gdy SM2=1 ignorowane są odebrane znaki, w których w trybie 2 i 3 dziewiąty bit danych jest zerem, a w trybie 1 nie został wykryty bit stopu; w trybie 0 powinno być SM2=0

4	REN	9Ch	Uaktywnienie odbioru znaków, gdy REN=0 nie są odbierane znaki
3	TB8	9Bh	Dziewiąty bit nadawanego znaku w trybie 2 i 3, ustawiany programowo
2	RB8	9Ah	Dziewiąty bit odebranego znaku w trybie 2 i 3
1	TI	99h	Flaga wysłania znaku; ustawiana sprzętowo po zakończeniu wysyłania znaku; zerowana tylko programowo; jest sygnałem zgłoszenia przerwania
0	RI	98h	Flaga odebrania znaku; ustawiana sprzętowo po odebraniu znaku; zerowana tylko programowo; jest sygnałem zgłoszenia przerwania

Tryby pracy portu szeregowego są opisane w **Tablicy 2**.

Tablica 2 Tryby pracy portu szeregowego

SM0	SM1	Tryb	Znaczenie
0	0	0	Transmisja synchroniczna; znaki 8-bitowe; taktowanie oscylatorem z częstotliwością FXTAL/12 (FXTAL – częstotliwość oscylatora).
0	1	1	Transmisja asynchroniczna; znaki 8-bitowe; szybkość określona programowo.
1	0	2	Transmisja asynchroniczna; znaki 9-bitowe; taktowanie oscylatorem z częstotliwością FXTAL/64 lub FXTAL/32.
1	1	3	Transmisja asynchroniczna; znaki 9-bitowe; szybkość określona programowo.

W przypadku trybów 1, 2 i 3 częstotliwość jest podwajana po ustawieniu 7 bitu rejestru PCON (SCON).

Dane kierowane na port składają się z bitu startu, ośmiu bitów znaku i bitu stopu. Pierwszy bit danej jest bitem najmniej znaczącym. Znak do wysłania jest pobierany z rejestru SBUF, podobnie jak znak przysłany.

Po skonfigurowaniu portu szeregowego należy - tylko w przypadku 1 i 3 trybu - skonfigurować szybkość transmisji. Szybkość transmisji w trybie 0 i 2 jest zadana przez częstotliwość rezonatora FXTAL i w trybie 0 jest równa FXTAL/12. Jeśli częstotliwość rezonatora jest równa 11.059Mhz i ustawiono tryb 0, to szybkość transmisji będzie zawsze równa 921,583 bodów (bitów na sekundę).

W trybie 1 i 3 szybkość transmisji jest zależna od częstotliwości przepełniania timera 1 - im większa częstotliwość przepełniania tym większa szybkość transmisji. Najprostszym sposobem ustawienia zadanej szybkość transmisji szeregowej w trybie 1 jest skonfigurowanie timera 1 do pracy w trybie 2 (tryb 8-bitowy, auto-reload, bit T1M1 rejestru TMOD ustawiony, bit T1M0 wyczyszczony). W takim przypadku do rejestru TH1 wprowadzamy wartość, która wymusi przepełnienie z odpowiednią częstotliwością.

Zakładając, że bit PCON.7 jest wyczyszczony, wartość, którą należy wpisać do rejestru TH1, aby otrzymać zadaną szybkość transmisji obliczamy ze wzoru:

TH1 = 256 - ((FXTAL / 384) / Szybkość)

Jeśli PCON.7 jest ustawiony TH1 wyliczamy ze wzoru:

TH1 = 256 - ((FXTAL / 192) / Szybkość)

Aby na przykład otrzymać szybkość transmisji równą 19200 bodów przy częstotliwości rezonatora równej 11,059MHz należy:
1. Ustawić tryb pracy portu równy 1 lub 3.
2. ustawić tryb pracy timera 1 równy 2.
3. Ustawić TH1 na 253, co odpowiada szybkości 19,200 bodów.
4. Ustawić PCON.7 (SMOD).

Typowe szybkości transmisji w bodach to: 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, 115200.

Wysłanie jednego bajtu polega na wpisaniu wartości do rejestru SBUF. Transmisja szeregowa zostaje zainicjowana automatycznie. Zakończenie transmisji jest sygnalizowane przez układ ustawieniem flagi TI (bit w rejestrze SCON). Odczytywanie znaku jest inicjowane programowo przez wyzerowanie w rejestrze SCON bitu flagi RI, pod warunkiem ustawienia REN=1. W chwili skompletowania całego bajtu w rejestrze przesuwnym jest on przepisywany do SBUF, a flaga RI zostaje ustawiona. Ustawienie flagi RI lub TI jest sygnałem zgłoszenia przerwania z portu szeregowego. Przyjęcie tego przerwania (gdy nie jest ono zamaskowane) nie powoduje automatycznego wyzerowania flag, a zatem ich stan może być w programie obsługi odczytany w celu identyfikacji przyczyny przerwania.

Flagi RI i TI powinny być programowo zerowane odpowiednio po odebraniu i wysłaniu danej.

1.3. Typowa konfiguracja portu szeregowego

Zaprogramowanie układu UART do pracy nie jest trudne. Należy tylko skonfigurować sam układ portu poprzez wpisanie odpowiedniej wartości do rejestru sterującego – SCON oraz licznika timera 1 – TMOD, TCON i TH1.

Listing 1 Przykładowy fragment programu odbierający znak

```
JNB RI,$        ;sprawdzenie flagi odbioru
MOV A,SBUF      ;czytanie z uarta
CLR RI          ;zerowanie flagi odbioru
```

Program posiada pętlę oczekującą na nadejście danej z portu szeregowego. Taka pętla

nie jest jednak praktyczna ze względu na to, że procesor nie może wykonywać innych zadań (oprócz obsługi przerwań).

Listing 2 Przykład wysłania znaku

```
JNB TI,$          ;czekanie na opróżnienie bufora nadajnika
CLR TI            ;wyzerowanie flagi wysłania
MOV SBUF,A        ;zapis do uarta
```

W przypadku wysyłania danych przed wpisaniem wartości do SBUF należy mieć pewność, że poprzednia dana została wysłana (testowanie flagi TI). Zastosowanie powyższej sekwencji rozkazów spełnia to założenie. Podobnie jak na **Listingu 1**, program spędza jałowo czas w pętli w oczekiwaniu na wysłanie danych.

Zdarzenia portu szeregowego mogą być źródłem przerwania. Jeśli przerwanie z portu szeregowego zostało odblokowane (poprzez ustawienie bitów EA i ES rejestru IE), ustawienie się którejkolwiek flagi RI lub TI powoduje podjęcie obsługi przerwania. Z tym portem jest związany adres 23h, od którego zaczyna się procedura obsługi. Zatem szkielet programu jest podobny jak dla obsługi innych przerwań:

Listing 3 Obsługa przerwań portu szeregowego

```
CSEG AT 0

LJMP start          ; po resecie wykonujemy skok do początku programu

CSEG AT 23h         ; pod adresem 23h z pamięci programu umieszczamy
LJMP serial_IT      ; instrukcję skoku do procedury obsługi przerwania

CSEG AT 30h
start:              ; program główny zaczyna się od adresu 30h
                    ;(za tablicą wektorów przerwań)
MOV SCON,...        ; Konfiguracja układu portu szeregowego
MOV TMOD,...        ; Konfiguracja Timera 1
MOV TH1,...         ; Konfiguracja szybkości transmisji szeregowej

SETB TR1            ; uruchomienie licznika T1
SETB ES             ; zezwolenie na przerwanie z portu szeregowego
SETB EA             ; globalne odblokowanie przerwań

loop:            ; główna pętla programu


LJMP loop



serial_IT:           ; procedura obsługi przerwania
PUSH PSW             ; zapamiętanie ważnych rejestrów na stosie
PUSH ACC
JBC TI,ser_tx        ; sprawdzenie powodu przerwania
CLR RI               ; odbiór
MOV ...,SBUF
.
```



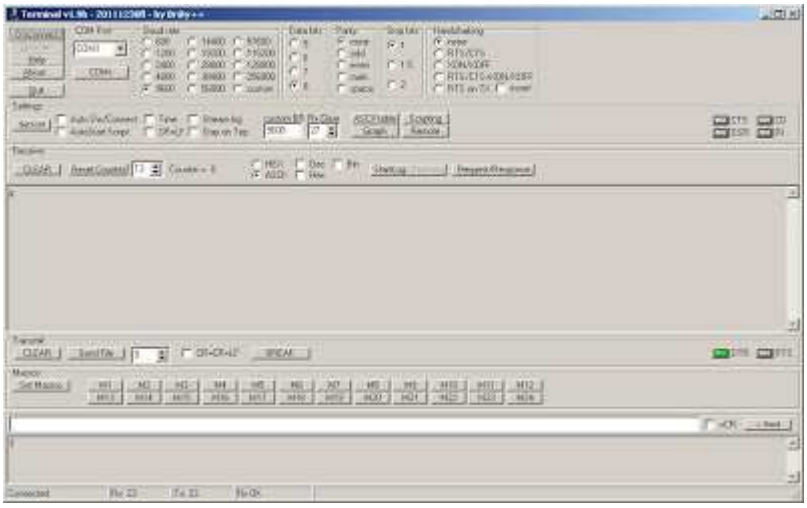
```
.
SJMP ser_end
ser_tx:              ; wysyłanie
;
MOV SBUF,...
;
ser_end:

POP ACC
POP PSW
RETI

END
```

1.4. Komunikacja z PC

Mikrokontroler 8051 może poprzez układ szeregowy komunikować się również z komputerem PC. Do komunikacji z zestawem ZL2MCS51 potrzebny będzie na PC terminal portu szeregowego, dostępny na https://sites.google.com/site/terminalbpp/ (**Rys.1**) Ponieważ ten sam port jest wykorzystywany do programowania zestawu ZL2MCS51 jak i transmisji szeregowej, programator Flip może blokować dostęp do portu. Należy zatem odpowiednio skonfigurować program Flip lub wyłączać go na czas pracy aplikacji. Rozpoczęcie pracy z terminalem polega na wybraniu właściwej prędkości transmisji i numeru portu komunikacyjnego. Ze względu na specyfikę zestawu laboratoryjnego należy również włączyć sygnał DTR tak, aby kontrolka w aplikacji świeciła na zielono. Następnie należy wcisnąć klawisz Connect. Każdy wpisany do okna nadawczego znak zostaje natychmiast wysyłany, natomiast dane nadchodzące są wyświetlane w oknie odbiorczym.



Rys. 1 Okno terminala

Literatura:

[1] http://www.8052.com/tut8051
[2] http://www.keil.com/support/man/docs/is51/is51_instructions.htm

2. Przebieg laboratorium

Druga część instrukcji zawiera zadania do praktycznej realizacji, które demonstrują zastosowanie technik z omawianego zagadnienia.

2.1. Zadanie 1. Na ocenę 3.0 (dst)

Należy przeanalizować pracę mikrokontrolera poprzez wykonanie symulacji krokowej kodu przedstawionego na **Listingu 4**, a następnie uruchomić program na płytce ZL2MCS51.

Listing 4

```
CSEG AT 0

AJMP reset

CSEG AT 30h

reset:
MOV SCON,#50h     ; uart w trybie 1 (8 bit), REN=1
MOV TMOD,#20h     ; licznik 1 w trybie 2
MOV TH1,#0FDh     ; 9600 Bds at 11.0592MHz
SETB TR1          ; uruchomienie licznika
CLR TI            ; wyzerowanie flagi wysłania

loop:

JNB RI,$          ; sprawdzenie flagi odbioru
MOV A,SBUF        ; czytanie z uarta
CLR RI            ; zerowanie flagi odbioru

CLR P2.0

INC A
MOV SBUF,A        ; zapis do uarta
JNB TI,$          ; czekanie na opróżnienie bufora nadajnika
CLR TI            ; wyzerowanie flagi wysłania

SETB P2.0

AJMP loop
END
```

Proszę uruchomić w symulatorze pracę krokową programu. Należy skorzystać z okna Peripherals->Serial do wprowadzania i odczytywania znaków, a także podglądać stany

flag RI i TI w oknie Serial Channel. Proszę sprawdzić działanie programu w rzeczywistym układzie.

2.2. Zadanie 2. Na ocenę 4.0 (db)

Program jest swego rodzaju koderem informacji, gdyż zamienia każdy znak na znak o kodzie o jeden większym. Proszę rozszerzyć program o „dekoder” zakodowanej informacji. Funkcja ma być przełączana za pomocą przycisków S6 i S7, a tryb pracy sygnalizowany odpowiednio diodami D6 i D7.

2.3. Zadanie 3. Na ocenę 5.0 (bdb)

Wykorzystując system przerwań proszę napisać program, który przychodzące dane będzie wpisywał do cyklicznego bufora o długości 16 bajtów umiejscowionego w wewnętrznej pamięci RAM.

