

Miroslaw Łazoryszczak

LABORATORIUM nr 4

Temat: Transmisja szeregową 8051

1. ARCHITEKTURA MCS-51 (CD.)

Uzupełnieniem układów wchodzących w skład mikrokontrolera 8051 jest układ transmisji szeregową. Jego rola jest trudna do przecenienia z kilku względów. Przede wszystkim zapewnia komunikację pomiędzy mikrokontrolerem, a światem zewnętrznym, co umożliwia współdziałanie z innymi systemami. Ponadto komunikacja ta wykorzystuje standardowy mechanizm transmisji szeregową, której parametry mogą być konfigurowane w zależności od potrzeb np. zgodnie ze standardem RS-232C, który jest obsługiwany przez znaczną liczbę urządzeń.

Układ transmisji szeregową wykorzystuje na zewnątrz dwa wyprowadzenia portu P3, a ściślej linie P3.0 oraz P3.1. W przypadku aktywności układu transmisji linie te noszą odpowiednio nazwy RxD (ang. *received data*) oraz TxD (ang. *transmitted data*). Konfiguracja układu transmisji szeregową w mikrokontrolerze 8051 możliwa jest za pomocą rejestru SCON. Znaczenie poszczególnych bitów przedstawia tabela 1.

Tabela. 1: Rejestr SCON konfiguracji układu transmisji szeregową.

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
SM0	SCON.7	Wybór trybu pracy					
SM1	SCON.6	Wybór trybu pracy					
SM2	SCON.5	Uaktywnienie trybu wieloprocesorowego					
REN	SCON.4	Aktywacja/blokowanie odbioru. Ustawiany/kasowany programowo					
TB8	SCON.3	Dziewiąty bit danych nadawany w trybie 2 lub 3					
RB8	SCON.2	Dziewiąty bit danych odbierany w trybie 2 lub 3. W przypadku gdy SM2=0 oznacza odebrany bit stopu.					
TI	SCON.1	Znacznik przerwania nadajnika. Ustawiany automatycznie po wysłaniu ostatniego bitu danych ¹ . Powinien zostać skasowany programowo.					
RI	SCON.0	Znacznik przerwania odbiornika. Ustawiany automatycznie po odebraniu ostatniego bitu danych ² . Powinien zostać skasowany programowo.					

Układ transmisji szeregową może pracować w czterech trybach. Poszczególne tryby pracy zostały wymienione w tabeli 2. Podany tam też został sposób ustawienia bitów SM0, SM1 z rejestru SCON.

Tabela. 2: Znaczenie bitów SM1, SM0 w rejestrze SCON.

SM0	SM1	Tryb	Opis	Prędkość transmisji
0	0	0	Synchroniczny rejestr przesuwany	$f_{XTAL}/12$
0	1	1	Transmisja asynchroniczna ośmiobitowa	Zmienna
1	0	2	Transmisja asynchroniczna dziewięciobitowa	$f_{XTAL}/64$ lub $f_{XTAL}/32$
1	1	3	Transmisja asynchroniczna dziewięciobitowa	Zmienna

Tryb nr 0 oznacza, że układ transmisji szeregową pracuje w sposób synchroniczny jednokierunkowo (ang. *half-duplex*). W trybie tym dane w trakcie transmisji mogą być albo tylko odbierane, albo tylko wysyłane. Sama transmisja odbywa się po linii RxD (P3.0), natomiast linią RxD (P3.1) przesyłany jest sygnał zegarowy o częstotliwości $f_{XTAL}/12$. Tryby nr 1, 2, 3 o tryby transmisji asynchronicznej dwukierunkowej (ang. *full-duplex*).

¹ W ujęciu uproszczonym. Dokładny moment ustawienia znacznika jest opisany w dokumentacji mikrokontrolera [2].

² Jak wyżej.

W przypadku transmisji asynchronicznej sygnał zegarowy nie jest transmitowany, zatem zarówno nadajnik jak i odbiornik powinny zostać skonfigurowane do pracy z taką samą prędkością. W trybach nr 1 oraz 3 prędkość transmisji ustalana jest przez układ czasowo-licznikowy. Szczegóły konfiguracji układu transmisji szeregową w tych trybach opisane są w materiale źródłowym [2, 4, 5].

Do celów realizacji niniejszego ćwiczenia wykorzystany zostanie tryb nr 2, który zostanie opisany nieco dokładniej. W trybie nr 2 możliwa jest transmisja dziewięciu bitów danych. Ośiem podstawowych bitów danych jest transmitowanych przy pomocy rejestru SBUF. Funkcje tego rejestru różnią się w zależności od kierunku przepływu danych. Przy zapisie do rejestru SBUF pełni on funkcję bufora nadajnika, natomiast przy odczycie z rejestru SBUF pełni funkcję bufora odbiornika. Należy o tym pamiętać w trakcie pracy z układem transmisji szeregową. Nie należy zwłaszcza np. wykonywać próby odczytu rejestru SBUF bezpośrednio po zapisaniu do niego wartości w celu sprawdzenia poprawności zapisu. Błędny jest również wielokrotny odczyt danych z rejestru SBUF np. za pomocą instrukcji znajdujących się obok siebie, ponieważ odczyty mogą być nieprawidłowe. Wynika to z faktu, że transmisja szeregową odbywa się ze znacznie mniejszą prędkością, aniżeli cykl rozkazowy mikrokontrolera 8051. Zatem w przypadku kolejnych zapisów danych do rejestru SBUF pierwotna wartość może nie zostać jeszcze wysłana, a już zostanie nadpisana przez nową wartość, co może zniekształcić transmitowane. Analogiczna sytuacja może wystąpić przy kolejnych odczytach danych z rejestru SBUF. W celu zapewnienia poprawności danych nadawanych lub odbieranych należy obserwować stan bitów RI oraz TI. Bity te są ustawiana automatycznie przez układ transmisji szeregową odpowiednio po wysłaniu danych linią TxD lub po skompletowaniu wszystkich bitów danych w rejestrze odbiornika. Stanowią one sygnał, że można ponownie zapisać kolejną daną do bufora nadajnika (SBUF) lub odczytać kolejną daną z bufora odbiornika. Przy czym po wykonaniu tych operacji należy programowo wyzerować odpowiednio bity TI lub RI. W trybie nr 2 transmisja przebiega z ustaloną prędkością równą $f_{XTAL}/64$ lub $f_{XTAL}/32$. Wybór odpowiedniej prędkości możliwy jest za pomocą najstarszego bitu (SMOD) w rejestrze PCON. Niestety rejestr PCON nie jest adresowany bitowo, zatem należy wpisać odpowiednią wartość nie naruszając jednocześnie zawartości pozostałych bitów. Wartość bitu SMOD równa 1 określa prędkość transmisji na $f_{XTAL}/32$, zaś wartość SMOD=0 wyznacza prędkość transmisji równą $f_{XTAL}/64$. W celu ustawienia bądź skasowania zawartości bitu należy wykonać operację sumy bitowej lub iloczynu bitowego oryginalnej zawartości rejestru PCON z odpowiednią wartością maski. Ustawienie bitu SMOD (przypisanie wartości 1) można zatem uzyskać w wyniku operacji:

$$PCON = PCON | 0x80,$$

zaś skasowanie (przypisanie wartości 0) za pomocą operacji:

$$PCON = PCON \& 0x7F.$$

Układ transmisji szeregową posiada możliwość zgłoszenia przerw, przy czym dla całego układu przewidziane jest tylko jedno przerwanie. Zadaniem ew. procedury obsługi przerwania jest identyfikacja właściwego źródła przerwania. Może nim być zarówno skompletowanie danej w odbiorniku sygnalizowane przez ustawienie bitu RI jak też opróżnienie bufora nadajnika poprzez ustawienie bitu TI. Każde z ww. zdarzeń jest sterowane sprzętowo i zachodzi automatycznie. Swego rodzaju potwierdzeniem obsługi potencjalnego przerwania bądź też obsługi zdarzenia bez użycia przerwań jest programowe skasowanie (wyzerowanie) odpowiedniego bitu RI lub TI. Obsługa układu transmisji szeregową możliwa jest zatem zarówno za pomocą przerwań jak też z wykorzystaniem mechanizmu cyklicznego sprawdzania stanu bitu TI przy wysyłaniu oraz bitu RI przy odbiorze danych. Zezwolenie na otrzymywanie przerwań możliwe jest za pomocą ustawienia bitu ES w rejestrze IE.

2. ŚRODOWISKO PRACY – ELEMENTY DODATKOWE

W niniejszym ćwiczeniu niezbędne będzie wykorzystanie procedury obsługi przerwania od układu transmisji szeregową. Z uwagi na fakt, iż w mikrokontrolerze adresy procedur obsługi przerwań są ustalone przez projektantów, należy zwrócić uwagę na uwzględnienie właściwego numeru przerwania. Na wydruku nr 1 przedstawiono składnię przykładowej procedury obsługi przerwania od układu transmisji szeregową. Wewnątrz procedury obsługi przerwania znajduje się fragment kodu, który rozstrzyga o właściwym źródle przerwania (gotowość nadajnika do przyjęcia kolejnej danej czyli opróżnienie bufora nadajnika sygnalizowane bitem TI oraz skompletowanie danej w rejestrze odbiornika sygnalizowane stanem bitu RI). W obu przypadkach po obsłudze przerwania należy skasować odpowiedni bit RT lub TI, będący bezpośrednim źródłem obsługiwanego przerwania.

```

01 void nazwa_procedury (void) __interrupt (4) __using (0) {
02     if {RI==1} {
03         // Tutaj należy umieścić obsługę odbioru
04         RI = 0;
05     }
06     else if (TI==1) {
07         // Tutaj należy umieścić obsługę nadawania
08         TI = 0;
09     }
10 }

```

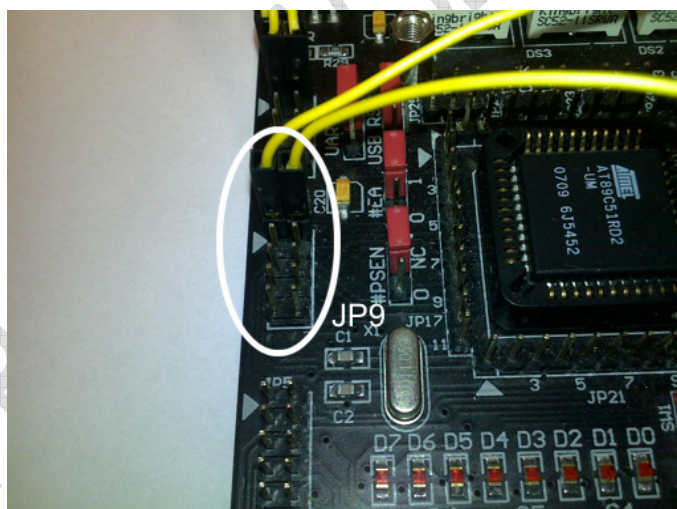
Wydruk 1: Procedura obsługi przerwania układu transmisji szeregowej

W zestawie uruchomieniowym ZL2MCS51 linie transmisji szeregowej RxD oraz TxD wyprowadzone są na złączu JP9 (wyprowadzenia nr 1 i 2). Linie te są też doprowadzone do układu scalonego U3 (MAX232), są więc wykorzystywane również do programowania układu oraz do ew. komunikacji między mikrokontrolerem a innymi urządzeniami. Ponadto wszystkie linie portu P3 są doprowadzone równolegle do zestawu przycisków S0 ÷ S7 włączając linie P3.0 (RxD) oraz P3.1 (TxD).

**UWAGA:**

Z powodu zastosowanego sposobu połączeń NIE jest możliwe wykorzystywanie przycisków S0, S1 jednocześnie z obsługą linii RxD oraz TxD układu transmisji szeregowej. Wciśnięcie jednego z ww. przycisków może zniekształcić transmitowane lub odbierane dane.

Na użytek poniższego zadania wykorzystana zostanie transmisja szeregową w ramach jednego zestawu ZL2MCS51. Dane będą więc nadawane na linii TxD. Dzięki połączeniu za pomocą przewodu wyprowadzenia TxD mikrokontrolera z wyprowadzeniem RxD (odpowiadające zwarcie wyprowadzeń nr 1 i 2 złącza JP9, rys. 1) możliwe będzie nadawanie i odbiór danych w tym samym układzie.



Rys. 1. Połączenie do testowania transmisji szeregowej.

Mikrokontroler będzie zatem wysyłał dane sam do siebie. Z powodu jednak powyższych ograniczeń pamięć należy o usunięciu połączenia przewodowego pomiędzy wyprowadzeniami 1 i 2 złącza JP9 na czas programowania układu.

PROCEDURA PROGRAMOWANIA I TESTOWANIA APLIKACJI:

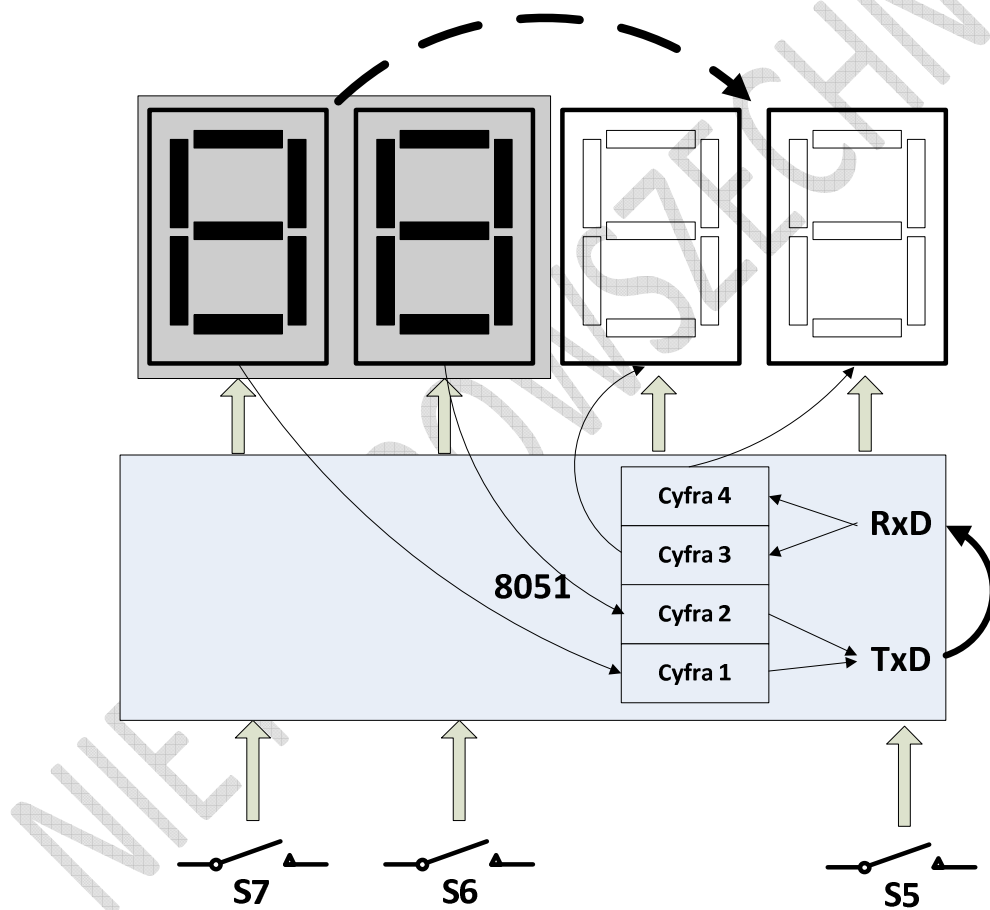
1. Załaduj program do układu z pomocą programatora Flip wg standardowej procedury.
2. Wykonaj RESET mikrokontrolera uruchamiając jednocześnie program
3. Połącz wyprowadzenia 1 i 2 w złączu JP9.
4. Przetestuj program.
5. Usuń połączenie z punktu 3.
6. W celu ponownego zaprogramowania mikrokontrolera powtórz czynności od kroku 1.



3. ZADANIA

Napisz program wykorzystujący układ transmisji szeregową oparty na programie z ćwiczenia nr 3. Założenia funkcjonalne są następujące:

1. W programie użyte zostaną wszystkie wyświetlacze siedmiosegmentowe oraz trzy przyciski (oprócz S0 i S1).
2. Dwa przyciski powinny służyć do zwiększenia stanu odpowiadających im dwóch wyświetlaczy, np. z lewej strony płytki uruchomieniowej analogicznie jak w ćwiczeniu nr 3
3. Wciśnięcie trzeciego przycisku wyzwała transmisję szeregową zawartości obu wyświetlaczy. Do transmisji danych wykorzystaj układ transmisji szeregową skonfigurowany do pracy w trybie nr 2
4. Odebrane liczby powinny zostać wyświetlone na pozostałych wyświetlaczach (rys. 2)
5. Usunięcie przewodu łączącego wyprowadzenia nr 1 i 2 złącza JP9 w przerwie między transmisją i próbą transmisji powinny skutkować brakiem możliwości uaktualnienia wyświetlaczy „odbiorczych”, zaś ponowne wykonanie połączenia przywraca układ do normalnej pracy z poprawnym odbiorem i prezentacją danych.



Rys. 2. Zasada działania programu

WSKAZÓWKI:

1. Odbiór danych zaimplementuj w procedurze obsługi przerwania układu transmisji szeregową.
2. Nadawanie danych wykonaj w pętli głównej programu.
3. W celu minimalizacji przesyłanych danych oraz uproszczenia programu umieść obie nadawane cyfry w jednym bajcie danych, natomiast po jego odebraniu ponownie wyodrębnij dwie cyfry do oddzielnych zmiennych, które wykorzystywane są do bezpośredniej prezentacji na wyświetlaczu. Pozwoli to uniknąć ew. niejednoznaczności w określaniu kolejności bajtów w przypadku transmisji każdej z cyfr w oddzielnej ramce.



```
01 void serial_init(void) {
02     SM0 = 1;        // Ustawienie trybu pracy nr 2
03     SM1 = 0;
04     TI = 1;         // Ustawienie bitu TI pozwalające
05                     // na rozpoczęcie nadawania w pętli głównej
06     RI = 0;
07     ES = 1;         // Dopuszczenie przerw od układu
08                     // transmisji szeregowej
09 }
10 void main(void) {
11     EA = 0;          // Zablokowanie przerw
12     serial_init();   // Przygotowanie układu transmisji
13     EA = 1;          // Odblokowanie przerw
14     REN = 1;         // Zezwolenie na odbiór danych
15     while(1) {
16     }
17 }
```

Wydruk 2: Procedura konfiguracji układu transmisji szeregowej wraz z fragmentem pętli głównej

4. LITERATURA

- [1] BTC Korporacja: Dokumentacja techniczna zestawu uruchomieniowego ZL2MCS51, <http://www.btc.pl/pdf/zl2mcs51.pdf> (dostęp: luty 2011).
- [2] Intel, MCS51 Microcontroller Family User's Manual, 1994 (dokumentacja dostępna pod różnymi adresami w sieci Internet).
- [3] Systemy wbudowane - Laboratorium nr 1, 2, 3, instrukcje do ćwiczeń.
- [4] Majewski J.: Programowanie mikrokontrolerów 8051 w języku C. Pierwsze kroki, BTC, Warszawa, 2005.
- [5] Stępień C. (red.): Mikroprocesory firmy Intel, PWN, Warszawa 1992.