

# Biblioteki mikroprocesorowych procedur standardowych



## Procedura obsługi transmisji szeregowej dla mikrokontrolerów MCS-51

*Układy ze sterownikami procesorowymi są z reguły prostymi, autonomicznymi urządzeniami. Do komunikacji z nimi wystarczy kilka przycisków lub nieskomplikowane klawiatury. Zdarza się jednak, że układy te współpracują z komputerami klasy PC. Są sterowane przez takie urządzenie nadrzędne lub wysyłają zebrane dane do PC-ta. W tym przypadku najprościej zastosować do dwustronnej komunikacji port szeregowy.*

Procesory z rodziny '51 wyposażono w sprzętowe i programowe udogodnienia, pozwalające stosunkowo łatwo stworzyć złącze szeregowe pracujące w standardzie RS-232. Co najmniej dwa takie złącza można znaleźć w każdym przeciętnym PC-cie. Jedno zazwyczaj jest zajęte przez mysz, ale drugie może posłużyć do komunikacji ze sterownikiem jednoukładowym.

Standard RS-232 opracowano do współpracy między takimi urządzeniami jak komputer i modem, co w pierwotnej wersji standardu wymagało wielu linii sterujących. Do wymiany danych ze sterownikiem wystarczy jednak najbardziej zubożona postać RS-232.

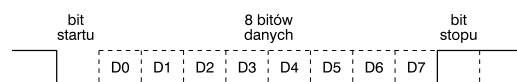
Żeby zapewnić bezbłędne przesyłanie danych, użytkownik musi najpierw ustalić takie parametry transmisji jak: szybkość transmisji, format ramki oraz elektryczne parametry komunikujących się wejść i wyjść. Jak sugeruje sama nazwa, informacje między urządzeniami przesyłane są szeregowo, kolejno bajt za bajtem. Ponieważ interesować nas będzie szeregowa transmisja asynchroniczna, aby przesyłanie danych uwieńczyło zostało sukcesem, należy zarówno nadajnik, jak i odbiornik ustawić w tryb pracy z taką samą szybkością. W przypadku procesorów '51 łatwo można zaprogramować jedną ze standardowych szybkości. Są to: 600, 1200, 2400, 4800, 9600 i 19200 bodów (bitów/s). Złącze szeregowe urządzenia współpracującego, np. PC-ta musi pracować z taką samą szybkością. Oprócz tego oba urządzenia muszą się „umówić” co do formatu przesyłanych danych.

Ponieważ między przesyłanymi bajtami mogą wystąpić dowolnej długości przerwy, to każdy bajt musi być poprzedzony impulsem synchronizującym. Nazywa się on bitem startu i zawsze przyjmuje poziom niski. Po bicie startu są transmitowane kolejno bity danych, począwszy od najmłodszego D0. Standard pozwala ustalić różną liczbę transmitowanych bitów, jednak komunikując się ze sterownikiem '51 najpraktyczniej przyjąć format bajtowy, czyli ustalić liczbę bitów w jednej ramce na 8. Po najstarszym bicie danych wysyłany jest obowiązkowo bit stopu przyjmujący zawsze poziom wysoki. Jeden lub dwa bity stopu oddzielają przesłany bajt od początku transmisji kolejnego. Linia w stanie nieaktywnym, kiedy nie przesyłane są żadne dane, utrzymywana jest zawsze na poziomie wysokim. Format typowej ramki danych pokazano na **rys. 1**.

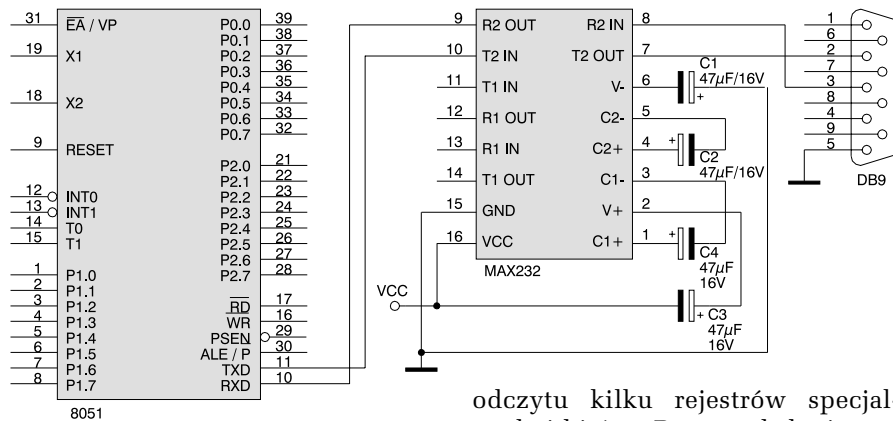
Ponieważ złącze szeregowe przewidziane jest do transmisji danych między dwoma urządzeniami zewnętrznym kablem, w celu zwiększenia odporności na zakłócenia i polepszenia jakości sygnału zdecydowano o przyjęciu innych poziomów napięć niż w standardzie TTL (0/5V).

Poziomowi niskiemu odpowiada napięcie z przedziału +5 do +12V, a poziomowi wysokiemu -5 do -12V. Powoduje to konieczność użycia układów pośredniczących, które dopasowują sygnały procesora do poziomów złącza RS. Obecnie stosuje się w tym celu specjalnie zaprojektowane układy scalone, np. popularny MAX232 lub jego liczne funkcjonalne odpowiedniki.

Na **rys. 2** przedstawiono typowo-



Rys. 1.



Rys. 2.

we połączenie wyjść portu szeregowego procesora z układem konwertera poziomów i gniazdem RS. Kabel połączeniowy składa się z 3 przewodów (tzw. null modem). Dwa z nich łączy wejście i wyjście sterownika '51 odpowiednio z wyjściem i wejściem komputera, a trzeci przewód jest przewodem masowym. Kabel można wykonać przy pomocy tzw. skrętki, w której przewód sygnałowy skręcony jest z przewodem masy, co znacznie ogranicza negatywny wpływ zakłóceń na jakość przesyłanego sygnału. W zależności od poziomu zewnętrznych zakłóceń, maksymalna długość kabla może wynosić kilka do kilkunastu metrów. Stosując długi kabel należy się liczyć z koniecznością obniżenia szybkości transmisji.

Schemat kabla połączeniowego złącza RS pokazuje rys. 3. Wtyk od strony komputera musi posiadać dwie dodatkowe zwory. Złącza RS mogą być także 25-stykowe. W tym przypadku przyporządkowanie złącz 9- i 25-stykowych jest następujące:

DB9	DB25
2	3
3	2
4	20
5	7
6	6
7	4
8	5

W sterowniku '51 wysyłane dane wraz z pomocniczymi impulsami ramki podawane są na wyjście TXD, natomiast dane odbierane trafiają do wyprowadzenia RXD. Konstruktorzy procesora zadbali o to, aby proces dwustronnej transmisji był maksymalnie zautomatyzowany.

Z punktu widzenia programisty sprowadza się on do zapisu lub

odczytu kilku rejestrów specjalnych i bitów. Proces obsługiwany jest przez przerwanie portu szeregowego, a nad utrzymaniem wszystkich parametrów czasowych czuwa licznik T1. Z tego powodu w czasie trwania transmisji szeregowej licznik ten nie może być w inny sposób wykorzystywany przez program mikrokontrolera. Do dyspozycji pozostaje natomiast licznik T0.

Przed rozpoczęciem transmisji szeregowej programista oprócz zaprogramowania rejestrów TL1 i TH1 licznika oraz wybrania szybkość transmisji, musi jeszcze ustawić odpowiednią wartość w rejestrze SCON (adres w SFR 98H), którego bity określają pozostałe parametry transmisji. Ich znaczenie jest następujące:

SCON(6,7) - numer (binarnie) trybu pracy portu szeregowego; kolejne wartości definiują następujące tryby:

- 0 -transmisja synchroniczna 8-bitowa;
- 1 -transmisja asynchroniczna 8-bitowa o szybkości określonej programowo;
- 2 -transmisja asynchroniczna 9-bitowa o szybkości równej 1/32 lub 1/64 częstotliwości zegara;
- 3 - transmisja asynchroniczna 9-bitowa o szybkości określonej programowo;
- SCON(5) - bit maskujący odbiór kolejnego znaku;
- SCON(4) - uaktywnienie odbiornika portu szeregowego;
- SCON(3) - jest 9 bitem znaku wysyłanego w trybach 2 i 3;
- SCON(2) - jest 9 bitem odebranego znaku;
- SCON(1) - flaga zakończenia transmisji znaku i znacznik zgłoszenia przerwania;
- SCON(0) - flaga odebrania znaku i znacznik zgłoszenia przerwania.

Istnieją jeszcze dwa ważne rejestry związane z transmisją szeregową. Jest to bufor znaku odbieranego (rejestr do odczytu) i bufor znaku transmitowanego (rejestr do zapisu), oba o takiej samej nazwie SBUF (99H). Pomimo takiej samej nazwy i lokalizacji, oba rejestry obsługują dwa niezależne procesy odbioru i transmisji znaku.

List. 1 przedstawia proste procedury assemblerowe ułatwiające korzystanie ze złącza szeregowego. Opisane powyżej główne cechy transmisji i sprzętu oraz obszernie komentarze dodane do listingu powinny pomóc w szybkim zrozumieniu działania tych procedur oraz ich dostosowaniu do swoich potrzeb i ulepszeniu. Przedstawione procedury umożliwiają pracę w trybie 8-bitowej transmisji asynchronicznej z szybkością ustaloną programowo. Mogą być jednak łatwo zaadaptowane do innych trybów pracy. Przedstawiony został także szkielet programu głównego wykorzystującego procedury portu szeregowego.

Najpierw następuje deklaracja kilku bajtów i bitów używanych przez procedury. Ich adresy w pamięci RAM sterownika są oczywiście dowolne. Dane do wysłania są przechowywane w obszarze pamięci RAM sterownika noszącej nazwę 'bufor'. Wielkość tego obszaru jest zależna od wielkości bloku danych do wysłania. Znak odebrany z wejścia RXD umieszczony będzie w 1-bajtowym buforze 'rs\_buf\_in', do natychmiastowego wykorzystania przez program główny lub przepisania w inny obszar pamięci RAM pełniący rolę bufora danych wejściowych.

Najpierw program główny powinien zainicjować parametry transmisji wykorzystując w tym celu procedurę *Rs\_ini*. Szybkość transmisji określa się przez odpowiednie ustawienie bitu PCON.7 i wpisanie wartości do rejestrów licznika T1. Z konstrukcji procesorów '51 wynika, że dla uzyskania maksymalnej zgodności z parametrami czasowymi poszczególnych szybkości transmisji, należałoby stosować w generatorze zegarowym kwarc o częstotliwości 11,059MHz. Przy takiej częstotliwości zegara i wpisaniu do rejestrów TL1 i TH1 wartości podanych na wydruku listingu, bez problemu uzyskuje

się jedną z wybranych, standardowych prędkości transmisji.

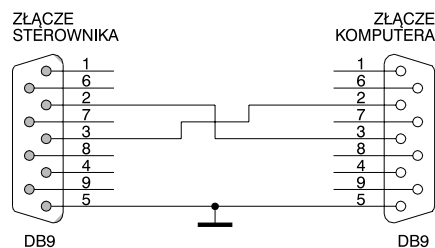
Nie oznacza to jednak, że dla kwarcu o innej częstotliwości nie jest to także możliwe. Wymaga to jednak eksperymentów z dobraniem odpowiednich wartości w rejestrach licznika. I tak, dla kwarcu 12MHz i prędkości 1200 bodów należy ustawić bit PCON.7, a do rejestrów licznika wpisać wartość CCH. Procedura inicjacji kończy się ustawieniem flag zezwoleń na przerwanie portu szeregowego i globalnego zezwolenia na przerwanie.

Sygnalem dla programu głównego po poprawnym skompletowaniu odbieranego znaku jest ustawienie flagi *rs\_flag*. Oznacza to, że w rejestrze *rs\_buf\_in* znajduje się odebrany znak i program główny może go teraz odczytać i wykorzystać. Wcześniej należy jeszcze skasować flagę *rs\_flag*.

W przykładzie program główny oczekuje w pętli na odbiór kolejnych znaków. Taka pętla nie jest konieczna i program może wykonywać inne zadania. Ważne tylko, aby co pewien czas, zależnie od przyjętej szybkości transmisji, sprawdzał stan flagi *rs\_flag*. W przeciwnym wypadku może dojść do nadpisania kolejnego odebranego znaku na poprzednim, zanim został on odczytany przez program główny.

Procedura inicjacji transmisji danych z bufora '*bufor*' zaczyna się w przykładzie od etykiety *rs\_trans*. Na początku należy wpisać do odpowiednich rejestrów liczbę znaków do wysłania i ustawić wskaźnik na początek bufora. Transmisja zostanie rozpoczęta po wpisaniu do rejestru wyjściowego SBUF pierwszego znaku do wysłania. Kolejne znaki zostaną wysłane złączeniem szeregowym bez udziału programu głównego. Proces ten przejmie procedura obsługi przerwania. Po wysłaniu wszystkich znaków ustawiona zostanie flaga *rs\_end\_flag*.

**Ryszard Szymaniak, AVT**



Rys. 3.

Listing 1.

```

;*****
;* Procedura portu szeregowego RS
;* dla procesorów '51 taktowanych zegarem 11,059MHz
;*****

org 0
using 0

rs_buf_in equ 30h ;bufor wejściowy RS
rs_count equ 31h ;licznik transmitowanych bajtów
rs_buf_point equ 32h ;wskaźnik do aktualnej pozycji w buforze
                    ;danych do transmisji
bufor equ 33h ;bufor zawierający dane do transmisji portem
                    ;szeregowym
rs_end_flag bit 00h ;flaga sygnalizująca zakończenie transmisji
                    ;wszystkich bajtów
rs_flag bit 01h ;flaga sygnalizująca odebranie kolejnego
                    ;bajtu

jmp Start
org 023h ;wektor przerwania portu szeregowego
jmp Rs_int ;skok do procedury obsługi przerwania portu szeregowego

Start: ;przykładowy program wykorzystujący procedury odbioru i
        ;transmisji danych portem szeregowym
call Rs_ini ;skok do procedury inicjującej parametry transmisji portu
        ;szeregowego
        ;fragment programu użytkowego realizujący odbiór danych z
        ;portu szeregowego

clr rs_flag
rs_rec:
jnb rs_flag,$ ;pętla oczekiwania na odbiór bajtu z portu szeregowego
clr rs_flag
mov a,rs_buf_in ;odczytanie z bufora wejściowego odebranego bajtu
                ;do wykorzystania w programie użytkowym,
                ;który powinien zaczynać się w tym miejscu
jmp rs_rec ;skok do pętli oczekiwania na odbiór kolejnego
                ;bajtu

rs_trans: ;fragment programu użytkowego realizujący
        ;transmisję danych portem szeregowym
mov rs_count,#3 ;ustawienie licznika ilości bajtów do wysłania (tu
                ;np.3) w przypadku wysłania 1 bajtu rs_count=0
mov rs_buf_point,#bufor ;ustawienie wskaźnika na początek bufora
                ;zawierającego dane
mov sbuf,bufor ;inicjacja transmisji przez wpisanie do rejestru
                ;wyjściowego pierwszego bajtu z bufora
inc rs_buf_point ;adres kolejnego bajtu do wysłania
clr rs_end_flag ;zerowanie flagi zakończenia transmisji wszystkich
                ;bajtów z bufora
jnb rs_end_flag,$ ;oczekiwanie na zakończenie transmisji wszystkich
                ;znaków
ret

;*****
;* Procedura inicjacji parametrów transmisji portu szeregowego*
;*****
Rs_ini:
mov scon,#01010000b ;ustawienie trybu 1 i inicjacja odbiornika
anl pcon,#7fh ;pcon.7=0 dla 600 - 9600 bodów, pcon.7=1 dla 19200
                ;bodów
orl tmod,#20h ;licznik T1 pracuje w trybie 2 (8 bitowy z
                ;automatycznym ładowaniem)
mov th1,#0e8h ;wartości początkowe dla poszczególnych prędkości
                ;transmisji:
mov t1l,#0e8h ;D0h-600, E8h-1200, F4h-2400, FAh-4800, FDh-9600,
                ;FDh-19200

clr tfl
clr rs_flag
setb tr1
setb es ;zezwoleń na przerwanie transmisji
                ;szeregowej portu RS
setb ea ;globalne zezwolenie na przerwanie
ret

;*****
;* Procedura przerwania portu szeregowego *
;*****
Rs_int:
push psw ;zachowanie zawartości rejestrów używanych przez procedurę
        ;przerwania
push acc
push ar0
jnb scon.1,rs_t ;przerwanie wywołane zostało przez zakończenie transmisji
                ;poprzedniego bajtu
clr scon.0 ;zerowanie flagi odbioru kolejnego bajtu
mov rs_buf_in,sbuf ;odebrany bajt do bufora
setb rs_flag ;ustawienie flagi sygnalizującej odebranie kolejnego
                ;bajtu
jmp rs_t1

rs_t: clr scon.1 ;zerowanie flagi sygnalizującej zakończenie
                ;transmisji bajtu
mov a,rs_count ;sprawdzenie czy wysłano już wszystkie bajty
jnz rs_t2
setb rs_end_flag ;ustawienie flagi sygnalizującej zakończenie
                ;transmisji wszystkich znaków

jmp rs_t1
rs_t2: mov r0,rs_buf_point
mov a,@r0 ;pobranie kolejnego bajtu z bufora do wysłania
                ;portem szeregowym
inc rs_buf_point ;ustawienie wskaźnika na kolejny bajt bufora
dec rs_count ;zmniejszenie o 1 licznika wysłanych bajtów
mov sbuf,a ;bajt do rejestru wyjściowego portu szeregowego
rs_t1:
pop psw ;odtworzenie pierwotnych wartości rejestrów używanych przez
        ;procedurę przerwania
pop ar0
pop acc
reti

end

```