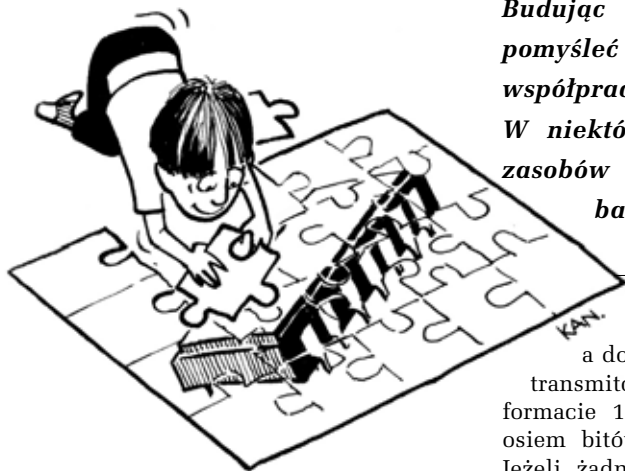


Podstawy projektowania systemów mikroprocesorowych, część 10



Chociaż omawiane modele mikrokontrolerów '51 to jedne z uboższych reprezentantów tej rodziny, to często ich możliwości wystarczają zupełnie do zrealizowania całego urządzenia bez dodatkowych układów peryferyjnych. Szczegółowe opisy funkcjonowania układów peryferyjnych wbudowanych w mikrokontroler znajdzie Czytelnik w specyfikacjach producenta, tutaj przedstawimy pracę tych układów w najczęściej wykorzystywanych przypadkach.

Komunikacja szeregową

W związku z ogromnym rozpowszechnieniem się komputerów klasy PC, budowane obecnie systemy mikroprocesorowe mają zazwyczaj możliwość komunikacji z komputerem nadrzędnym. Czasami zachodzi też konieczność współpracy kilku komunikujących się mikrokontrolerów. W takich wypadkach najczęściej wykorzystuje się transmisję szeregową z wykorzystaniem wbudowanego w większości mikrokontrolerów '51 układu transmisji szeregowej. Wykorzystanie takiego, a nie innego środka komunikacji ma tę zaletę, że realizacja transmisji szeregowej jest obsługiwana sprzętowo - odpada zatem konieczność pisania odpowiednich procedur obsługi, a co za tym idzie nie musimy rezerwować odpowiedniej mocy obliczeniowej procesora na obsłużenie transmisji.

W przypadku mikrokontrolerów rodziny '51 najczęściej wykorzystuje się układ transmisji szeregowej skonfigurowany do pracy w trybie 1. Tak zaprogramowany port szeregowy pracuje asynchronicznie, wykorzystując

Budując urządzenie wykorzystujące mikrokontroler, musimy pomyśleć również o pewnej liczbie elementów współpracujących, które realizują potrzebne nam funkcje. W niektórych sytuacjach wystarczające staje się sięgnięcie do zasobów sprzętowych wykorzystywanego mikrokontrolera, co jest bardzo korzystne zarówno ze względów komplikacji i niezawodności urządzenia, jak i jego ceny.

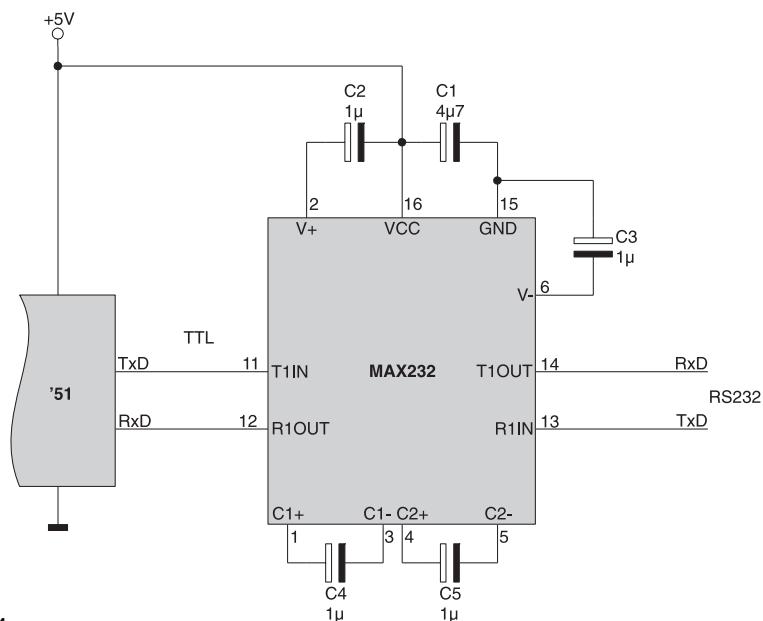
do nadawania linię TxD, a do odbioru linię RxD. Dane transmitowane są w standardowym formacie 10-bitowym (bit startu - 0, osiem bitów danych, bit stopu - 1). Jeżeli żadne dane nie są przesyłane, to obydwie linie pozostają w stanie wysokim. Prędkość transmisji jest ustawiana programowo poprzez odpowiednie zaprogramowanie licznika T1 - należy zatem zwrócić uwagę na częstotliwość taktowania mikrokontrolera, gdyż nie zawsze będzie można uzyskać standardowe prędkości transmisji. W przypadku urządzeń wykorzystujących transmisję szeregową najczęściej wykorzystuje się częstotliwość kwarcu równą 11,0592 MHz pozwalającą na uzyskanie większości standardowych prędkości do 57600 bd. Dla układu licznikowego T1 wykorzystywanego w trybie 2 (automatyczne przeładowywanie) zawartość rejestru TH1 (wartość licznika po przeładowaniu) można wyznaczyć z zależności (dla podanej częstotliwości kwarcu): $TH1=256 \cdot 28800/V$ (dla bitu SMOD=0, V - prędkość transmisji w bodach) lub $TH1=256 \cdot 57600/V$ (dla SMOD=1). Bit SMOD (rejestr PCON) jest bitem konfiguracyjnym układu transmisji szeregowej obsługującym blokadę wewnętrznego dzielnika przez dwa częstotliwości określającej szybkość transmisji. W **tab. 1** przedstawiono wartości przeładowania licznika T1 w zależności od oczekiwanej prędkości transmisji.

Do programowej obsługi układu transmisji szeregowej wykorzystywane są rejestry: SCON - służący do konfiguracji i obsługi transmisji oraz SBUF - będący jednobajtowym buforem nadajnika i odbiornika. Dodatkowo wykorzystywany jest też omówiony wcześniej bit SMOD rejestru PCON. Znaczenie i położenie bitów

sterujących rejestru SCON można znaleźć w specyfikacji 8051. Nas interesować tutaj będą jedynie bity: SM0 i SM1 - decydujące o trybie pracy - dla trybu 1 będą ustawione odpowiednio na 0 i 1. REN - bit włączający (dla REN=1) odbiornik układu transmisji szeregowej. RI - bit ustawiany sprzętowo na 1 po odebraniu bajtu. TI - bit ustawiany sprzętowo na 1 po zakończeniu nadawania bajtu. Programowa realizacja wysłania czy odebrania bajtu z układu transmisji szeregowej jest niezmiernie prosta. Polega ona jedynie na odpowiednim skonfigurowaniu portu oraz sprawdzaniu stanu bitów RI i TI. Przy odbiorze danych należy sprawdzić stan bitu RI i w przypadku pojawienia się stanu 1 należy odczytać odebraną informację z rejestru SBUF oraz wyzerować bit RI. W miarę bezwzględny odczyt odebranego bajtu jest konieczny ze względu na możliwość zamazania jego zawartości przez kolejny odebrany bajt (na odczyt mamy w najgorszym przypadku czas równy czasowi transmisji całego bajtu). W przypadku nadawania wystarczy zapisać daną do transmisji do rejestru SBUF - proces transmisji szeregowej rozpocznie się

Tab. 1

prędkość transmisji w bodach	TH1 dla SMOD = 0	TH1 dla SMOD = 1
120	16	-
150	64	-
300	160	64
600	208	160
1200	232	208
2400	244	232
4800	250	244
9600	253	250
19200	-	253
28800	255	254
57600	-	255



Rys. 34

automatycznie. Po zakończeniu transmisji zostanie ustawiony bit TI, którego sprawdzanie pozwala na stwierdzenie gotowości układu do transmisji kolejnej danej. Po programowym wyzerowaniu TI można do SBUF wpisać kolejną daną do przesłania (zerowanie TI nie jest konieczne do rozpoczęcia kolejnej transmisji, umożliwi nam jednak określenie chwili jej zakończenia). W mikrokontrolerach '51 ustawienie bitów RI lub TI może również być źródłem przerwania, jednak brak jest sprzętowej możliwości rozstrzygnięcia, który bit wywołał przerwanie (trzeba to sprawdzić programowo na początku procedury obsługi przerwania, również programowo trzeba wyzerować dany bit przed opuszczeniem procedury obsługi przerwania). Na list. 13 pokazano prosty program przeznaczony do odbioru dwubajtowych paczek danych.

Każdy z dwóch odebranych bajtów zostaje zapamiętany w odrębnej komórce pamięci. Do rozstrzygnięcia, który bajt jest odbierany, wykorzystywana jest zmienna bitowa BYL1 ustawiana w chwili odebrania pierwszego bajtu i zerowana po odebraniu drugiego. Program ten mógłby zostać napisany tak, aby bezpośrednio po odebraniu pierwszego bajtu oczekiwać na nadejście kolejnego, jednak należy pamiętać o tym, że prędkość transmisji zawsze będzie dużo niższa od prędkości wykonywania programu, więc takie bezczynne oczekiwanie jest bezcelowe i marnuje moc obliczeniową mikrokontrolera.

Drugi przedstawiony program (list. 14) jest kompletnym oprogramowa-

niem przeznaczonym dla prostego układu testera portu RS-232. Realizuje on bardzo prosty algorytm polegający na odsyłaniu odebranego bajtu i pozwalający w ten sposób sprawdzić działanie komunikacji pomiędzy urządzeniami. Po włączeniu zasilania następuje wysłanie przez układ transmisji szeregowej tekstu „OK”. Transmisja odbywa się z prędkością 57600 (dla kwarcu 11,0592 MHz).

Przy wykorzystywaniu portu szeregowego do połączenia z komputerem PC należy pamiętać, że standard RS-232 wykorzystuje inne po-

ziomy napięcie niż stosowany przez mikrokontrolery standard TTL. W celu połączenia konieczne staje się wykorzystanie dodatkowego układu zapewniającego konwersję. Można do tego wykorzystać różne metody, począwszy od tranzystorów i bramek CMOS, jednak obecnie najczęściej wykorzystuje się scalony konwerter poziomów MAX232 (lub zamiennik), który ma tę ogromną zaletę, że zasilany standardowym napięciem +5V sam tworzy się o wytworzenie napięcia dodatniego i ujemnego zgodnego ze standardem RS-232 oraz jest stosunkowo tani i łatwo dostępny. Schemat takiego konwertera pokazano na rys. 34. Niekiedy do uzyskania komunikacji mikrokontrolera z komputerem PC należy zadbać o połączenie ze sobą od strony złącza RS-232 PC-ta wyprowadzeń RTS-CTS oraz DTR-DSR-DCD (rys. 35), informujących komputer o gotowości mikrokontrolera do odbioru/nadawania danych (układ transmisji szeregowej '51 nie jest wyposażony w funkcję sprzętowego blokowania transmisji - transmisja może być kontrolowana jedynie na drodze programowej). Oczywiście należy również zadbać o ustawienie identycznej prędkości transmisji, gdyż w przeciwnym wypadku prawidłowy przesył danych nie będzie możliwy.

Jeżeli transmisja szeregowa będzie wykorzystywana do komunikacji pomiędzy dwoma mikrokontrolerami

List. 13

```

;Konfiguracja
MOV TMOD,#020H ;licznik T1 w trybie 2
MOV PCON,#080H ;ustawienie bitu SMOD=1 w rej. PCON
MOV TL1,#250
MOV TH1,#250 ;prędkość transmisji 9600

CLR SM0 ;ustawienie
SETB SM1 ;trybu pracy 1

CLR RI ;wyzerowanie flagi

SETB TR1 ;włączenie licznika T1

SETB REN ;włączenie odbiornika

CLR BYL1 ;wyzerowanie flagi odbioru bajtu 1

;Podprogram obsługi odbioru
ODBIERZ:
JB RI,JEST1 ;sprawdzenie czy nadeszła dana
RET ;i powrót do programu głównego jeśli nie
JEST1:
CLR RI ;wyzerowanie flagi
JB BYL1,JEST2 ;sprawdzenie czy wcześniej odebrano pierwszy bajt,
;jeśli tak to skok do odbioru bajtu 2

SETB BYL1 ;ustawienie flagi odbioru pierwszego bajtu
MOV PIERWSZY,SBUF ;zapamiętanie zawartości bufora w komórce pamięci
;zadeklarowanej jako PIERWSZY
RET ;powrót do programu głównego
JEST2:
CLR BYL1 ;wyzerowanie flagi (następny będzie bajt pierwszy)
MOV DRUGI,SBUF ;zapamiętanie drugiego odebranego bajtu

RET ;powrót do programu głównego

```

rami, wówczas sposób ich połączenia zależy od odległości łączącego je przewodu. Jeżeli ograniczymy się do połączenia nie dłuższego niż 1...2m (zwłaszcza przy niskich prędkościach, nieprzekraczających 9600 bd), możemy wtedy pozostać przy standardzie TTL i bezpośrednio połączyć pomiędzy sobą wyprowadzenia RxD i TxD mikrokontrolerów (RxD jednego z TxD drugiego) oraz połączyć masy obu urządzeń. Dodatkowo należy zastosować rezystory podciągające te linie do Vcc o rezystancji 1...2kΩ ze względu na konieczność szybszego przeładowania stosunkowo dużej pojemności przewodu połączeniowego. Jeżeli odległość będzie większa, to nic nie stoi na przeszkodzie, aby zastosować dwa identyczne układy konwertera omówionego wyżej. Dla dużo większych odległości (powyżej 20 m) może stać się konieczne zastosowanie innego standardu - może to być np. RS-485 lub inny wykorzystujący magistralę symetryczną.

Paweł Hadam, EP
pawel.hadam@ep.com.pl

List. 14

```

;;Program testujący port szeregowy
;;Konfiguracja
MOV TMOD,#020H          ;licznik T1 w trybie 2

MOV PCON,#080H          ;ustawienie bitu SMOD=1 w rej. PCON
MOV TL1,#255             ;ustawienie licznika T1
MOV TH1,#255             ;prędkość transmisji 57600

CLR SM0                  ;ustawienie
SETB SM1                 ;trybu pracy 1

CLR RI                   ;wyzerowanie flagi (nic nie odebrano)
SETB TI                 ;ustawienie flagi (gotowy do nadawania)
SETB TR1                 ;włączenie licznika T1

SETB REN                 ;włączenie odbiornika

;;Transmisja "OK"

MOV SBUF,#79             ;wysłanie kodu ASCII znaku "O"
CLR TI                   ;wyzerowanie flagi
JNB TI,$                 ;oczekiwanie na zakończenie transmisji
                           ;poprzedniego bajtu
MOV SBUF,#75             ;wysłanie kodu ASCII znaku "K"
CLR TI                   ;wyzerowanie flagi

;;Pętla główna
PETLA:
JNB RI,$                 ;sprawdzanie czy odebrano bajt, jeśli nie, to czekaj

CLR RI                   ;wyzerowanie flagi

MOV R7,SBUF              ;zapamiętanie danej w rejestrze
JNB TI,$                 ;czekaj, jeśli jeszcze nadaje

MOV SBUF,R7              ;nadanie zapamiętanej danej
CLR TI                   ;wyzerowanie flagi

SJMP PETLA               ;skok na początek

```