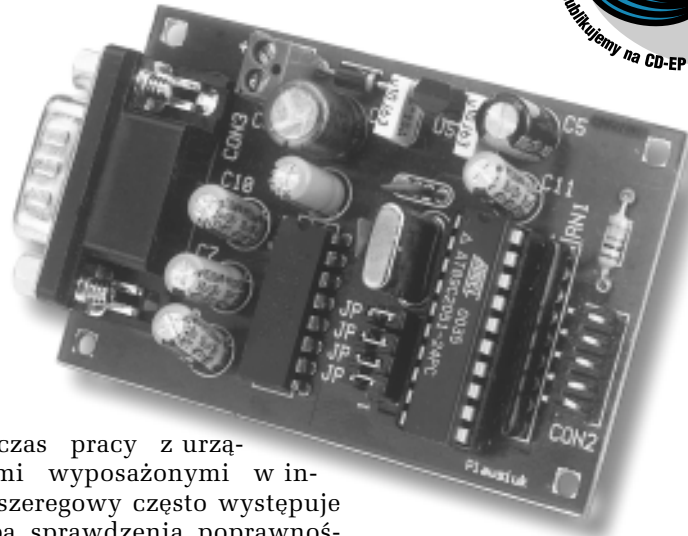


Sprzętowy terminal RS232

AVT-5073



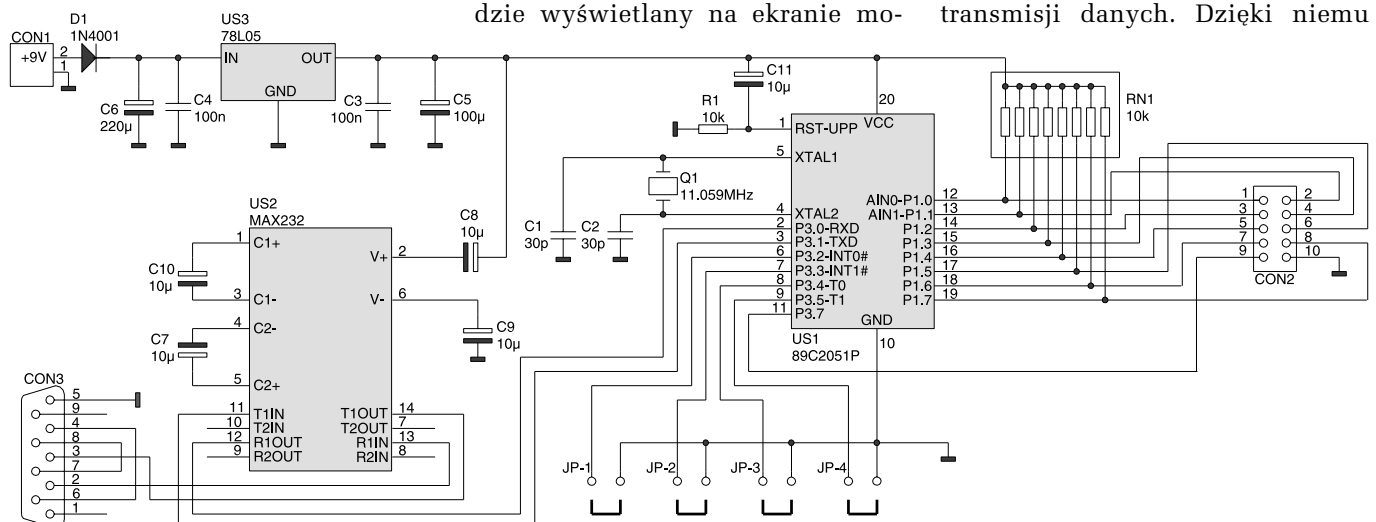
Interfejs komunikacji szeregowej RS232 jest, ze względu na łatwość implementacji i wystarczającą w większości zastosowań szybkość transmisji, stosowany w wielu urządzeniach komunikujących się z otoczeniem. Układ opisany w artykule jest prostym, dwukierunkowym terminalem RS232 o nietypowych możliwościach.

Podczas pracy z urządzeniami wyposażonymi w interfejs szeregowy często występuje potrzeba sprawdzenia poprawności transmisji. Prezentowany w artykule układ umożliwia odbieranie oraz wysyłanie danych przez interfejs RS232. Odbierane dane są kierowane do portu procesora, który może na przykład wysterować diody świecące w celu porównania danych wysłanych przez komputer lub procesor z tymi, które faktycznie pojawiają się na szeregowej linii danych. Do wyjść mikrokontrolera można również dołączyć urządzenia sterujące odbiornikami dużej mocy, umożliwiając sterowanie za pomocą komputera różnymi urządzeniami domowymi. W trybie nadajnika dane pojawiające się na liniach portu procesora są wysyłane do komputera szeregowo. Do portu tego można dołączyć klawiaturę lub dowolny czujnik, a stan linii będzie wyświetlany na ekranie mo-

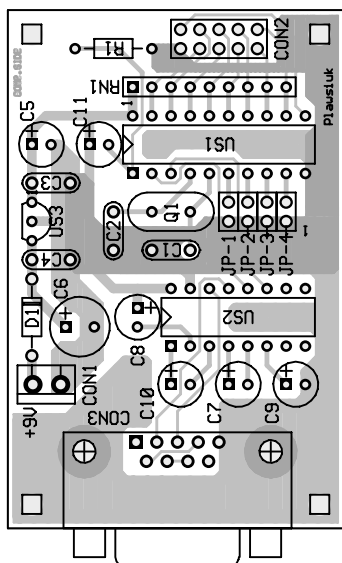
nitora. Do sterowania przepływem danych można użyć dowolnego programu terminalowego, na przykład Hyper Terminal, w który jest wyposażony każdy system Windows. Prędkość transmisji może być zmieniana w zakresie od 300 do 19200 bd. Transmisja odbywa się w sposób asynchroniczny, z jednym bitem startu i jednym bitem stopu, bez bitu parzystości.

Budowa i działanie

Na rys. 1 przedstawiono schemat elektryczny sprzętowego terminala. „Sercem” układu jest mikrokontroler firmy Atmel AT89C2051, który zawiera w swojej strukturze 2 kB pamięci programu typu Flash oraz - bardzo istotny dla działania urządzenia - sprzętowy interfejs szeregowy transmisji danych. Dzięki niemu



Rys. 1. Schemat elektryczny terminala z portem szeregowym



Rys. 2. Rozmieszczenie elementów na płytce drukowanej terminala

procedura obsługi portu szeregowego jest bardzo uproszczona, gdyż procesor nie musi programowo odbierać i wysyłać danych bit po bicie. Jest to wykonywane całkowicie sprzętowo. Wbudowany w mikrokontroler nadajnik-odbiornik samodzielnie przetwarza dane i po odebraniu lub wysłaniu ich informuje procesor, że cały bajt został odebrany lub wysłany. W czasie wysyłania lub odbierania danych z portu szeregowego procesor może zajmować się innymi procesami.

Sygnał zegarowy potrzebny do pracy procesora jest wytwarzany przez oscylator z zewnętrznym rezonatorem kwarcowym o częstotliwości 11,059MHz. Taka częstotliwość pracy procesora umożliwia uzyskanie standardowych prędkości transmisji. Zerowanie procesora po włączeniu zasilania zapewnia obwód różniczkujący złożony z rezystora R1 i kondensatora C11. Zestaw rezystorów RN1 zapewnia zewnętrzne podciąganie wyprowadzeń portu P1 do poziomu wysokiego. Jest to konieczne, gdyż w wewnętrzne rezystory podciągające (*pull-up*) wyposażone są tylko wyprowadzenia P1.2...P1.7, natomiast wyprowadzenia P1.0 i P1.1 są typu otwarty kolektor. Dla ujednoczenia warunków pracy portu, rezystory zastosowano dla wszystkich jego wyprowadzeń.

Układ US2 jest konwerterem dostosowującym napięcia standardu RS232 (-15V...+15V) do napięć

w standardzie TTL (0...5V), akceptowanych przez mikrokontroler. Układ MAX232 ma wbudowany podwajacz napięcia (do uzyskania napięcia na wyjściach około +10V) oraz układ generujący ujemne napięcie o wartości równej około -10V. Komputery akceptują napięcia wejściowe dla dodatniego w przedziale +5V...+15V oraz ujemnego -5V...-15V, więc układ US2 spełnia wymogi napięć standardu RS232 pod względem napięciowym.

Do zasilania całego układu wystarczy tylko napięcie 5V, które jest otrzymywane na wyjściu stabilizatora US3.

Dane odbierane przez terminal z portem szeregowym są przekazywane do złącza CON2. Jeśli układ pracuje jako nadajnik szeregowy, dane są pobierane z portu P1. Na złączu CON2, oprócz wyprowadzeń portu P1, jest dostępne również jedno wyprowadzenie portu P3. Wyprowadzenie to jest wykorzystywane jedynie wtedy, gdy dane są pobierane z portu P1 i mają być wysłane szeregowo.

Po ustawieniu odpowiednich stanów na złączu CON2 należy podać ujemny impuls na wyprowadzenie P3.7 portu P3. Spowoduje on wysłanie danych ze złącza CON2 szeregowo do komputera. Na tym wejściu wyzwalania transmisji są programowo eliminowane drgania zestyków, co umożliwi wyzwalanie za pomocą klawisza.

Prezentowany układ umożliwia transmisję przy jednej z siedmiu standardowych prędkości. Do wyboru prędkości służą zworki JP1...JP3. Wykorzystanie tych zwoerek i ustawienia żądanej prędkości transmisji zestawiono w **tab. 1** (1 - zwarcie, 0 - rozwarcie). Ponieważ za pomocą trzech zwoerek można wybrać osiem stanów, dlatego kombinacji pozycji siódmej i ósmej odpowiada taka sama prędkość.

Do przełączania układu między nadawaniem i odbiorem służy zworka JP4 - jej zwarcie wprowadza układ w tryb odbioru danych z portu szeregowego i przesłanie na złącze CON2. Rozwarcie tej zworki umożliwia wysłanie danych pobranych ze złącza CON2

przez port szeregowy. Wszelkie zmiany prędkości, czy trybu pracy należy wykonywać przy wyłączonym zasilaniu, gdyż ustawienie wszystkich zwoerek jest odczytywane tylko raz, przy starcie procesora. Wszelkie zmiany ustawienia zwoerek w czasie pracy procesora będą uwzględnione dopiero po jego wyzerowaniu, czyli wyłączeniu i ponownym włączeniu zasilania.

Dzięki sprzętowemu sterownikowi transmisji szeregowej wbudowanemu w mikrokontroler, procedura nadawania i odbioru danych sprowadza się jedynie do odczytu lub zapisu danych do odpowiedniego rejestru. Ponieważ praca portu szeregowego jest asynchroniczna, w zależności od wybranej prędkości poszczególne bity są wysyłane z różną prędkością. Do odliczania czasu pomiędzy kolejnymi bitami wysłanego lub odbieranego bajtu danych najwygodniej jest zastosować *Timer*. W mikrokontrolerze zastosowanym w terminalu do dyspozycji są dwa *Timery*, jednakże do obsługi transmisji jest przypisany *Timer1*. Składa się on z dwóch rejestrów (TH1 - starszy bajt i TL1 - młodszy bajt). W zależności od konfiguracji *Timera*, rejestry można połączyć ze sobą, dzięki temu otrzymamy licznik dwubajtowy, o maksymalnej zliczanej wartości 65536. Do współpracy ze sterownikiem transmisji szeregowy wykorzystano tryb pracy tego *Timera*, nazywany *trybem 2*. W tym trybie *Timer1* może odliczyć maksymalnie 256 cykli. Zliczanie odbywa się w rejestrze TH1, a w rejestrze TL1 przechowywana jest początkowa wartość, od której licznik ma rozpocząć zliczanie. Po uzyskaniu maksymalnej wartości rejestru

Tab. 1. Wykorzystanie zwoerek do ustawienia prędkości transmisji

L.p.	JP3	JP2	JP1	Prędkość transmisji [bd]
1	0	0	0	300
2	0	0	1	600
3	0	1	0	1200
4	0	1	1	2400
5	1	0	0	4800
6	1	0	1	9600
7	1	1	0	19200
8	1	1	1	19200

List. 1. Procedura inicjalizacji i obsługi portu szeregowego

```

***** Ustawienie TIMERA *****
;TIMER 1
T1_G EQU 0 ;GATE
T1_C EQU 0 ;COUNTER/~-TIMER
T1_M EQU 2 ;MODE (0..3)
TIM1 EQU T1_M+T1_C*4+T1_G*8
TMOD_SET EQU TIM0+TIM1*16 ;
ustawienie trybu 2 dla timer1
TR_M EQU 1 ;tryb transmisji (1..3)
TR_R EQU 1 ;zezwozenie na odbior
SCON_SET EQU TR_M*64+TR_R*16 ;ustawienie
;trans. asynchronicznej

MOV TH1,#TH1_SET ;wpisanie wartości
;początkowej Timera
MOV TL1,#TL1_SET ;wg. wzoru
;TH1=256-(57600/V)

START:
MOV SCON,#SCON_SET ;konfiguracja portu
;szeregowego jako
ORL PCON,#80H ;asynchroniczny
MOV TMOD,#TMOD_SET ;konfiguracja Timer1
;do pracy w trybie 2
SETB TR1;start Timera 1
SETB TI
MOV P1,#255 ;port1=FF
MOV P3,#255 ;port3=FF
JNB TRYB,LOOP_1 ;jeżeli tryb=1,to nadajnik
;Praca jako nadajnik
LOOP:
JNB TI,S ;czy można nadać znak
JB DIODA,LOOP ;jeżeli P3.7=0 nadać znak
;z Portu1
CLR TI ;zeruj znacznik nadawania
LOOP_2:
MOV A,P1 ;kopiuj port1 do ACC
MOV SBUF,A ;wyślij znak
LOOP_3:
JNB DIODA,LOOP_3 ;czekaj na 1 na P3.7
MOV A,#10 ;czekaj 10ms
CALL WAIT_10
SJMP LOOP ;kolejny znak

;Praca jako odbiornik
LOOP_1:
SETB DIODA ;ustaw P3.7
JNB RI,S ;czekaj na znak z portu
;szeregowego
CLR RI ;jeżeli jest, to zeruj
;znacznik
MOV A,SBUF ;przepisz znak do ACC
MOV P1,A ;przepisz znak do portu P1
SJMP LOOP_1 ;następny znak

```

TH1 następuje automatyczne przepisanie jego wartości początkowej z rejestru TL1, dzięki czemu nie trzeba za każdym razem wpisywać jego wartości początkowej. Na list. 1 przedstawiono procedurę obsługi portu szeregowego.

Na początku programu ustalamy tryb pracy *Timera* (współczynnik podziału częstotliwości sygnału zegarowego). Wyliczona z zależności: $T1_M+T1_C*4+T1_G*8$ wartość jest przechowywana w *TIM1*. Następnie ustalamy tryb pracy sterownika transmisji szeregowej (w przykładzie: praca asynchroniczna oraz zezwolenie na odbiór) i wstępnie zapisujemy wyliczoną z zależności $TR_M*64+TR_R*16$ wartość w *SCON_SET*. Początkowa wartość wpisana do rejestru *Timera*, dla rezonatora kwarcowego o częstotliwości 11,059MHz, jest obliczana ze wzoru $TH1=256-(57600/V)$, gdzie V oznacza prędkość transmisji. Dla przykładu, dla prędkości 9600 wzór ma postać $TH1=256-(57600/9600)$, czyli $TH1=250$. Wartość tę wpisujemy

zarówno do rejestru TH1, jak również do TL1. Wpis ten wykonujemy jednorazowo, na początku programu.

W procedurze *Start* następuje uruchomienie portu szeregowego oraz start *Timera1*. W zależności od stanu zworki *TRYB* (JP-4) program przechodzi do procedury odbioru lub nadawania znaków. Jeśli zworka jest rozwarta, to wykonywany jest podprogram nadawania (pętla *LOOP*). W pętli tej sprawdzany jest stan bitu *TI*. Bit ten informuje, czy sterownik portu szeregowego skończył nadawanie poprzedniego znaku. Jeśli $TI=1$, to sprawdzany jest stan portu P3.7 (warunek odczytu portu P1). Jeśli obydwa warunki są spełnione następuje odczyt stanów portu P1 i przepisanie tych stanów do rejestru *SBUF*. Po wpisaniu danych do tego rejestru następuje automatyczne jego wysłanie portem szeregowym. Następnie program oczekuje na stan wysoki na wyprowadzeniu portu P3.7, i powraca do początku pętli *LOOP*.

Procedura wysyłania danych przez port szeregowy sprowadza się do sprawdzenia czy został już wysłany poprzedni znak oraz wpisania odpowiedniej wartości do rejestru nadawczego *SBUF*, a cała procedura wysyłania poszczególnych bitów jest wykonywana poza programem. Podobnie jest dla odbioru transmisji szeregowej. Zapewnia go procedura zawarta w pętli *LOOP1*. Jej zadaniem jest sprawdzanie stanu flagi *TI*. Jeśli $TI=1$, oznacza to, że w rejestrze *SBUF* jest odebrany szeregowo bajt danych i następuje jego przepisanie do portu P1. Rejestr *SBUF* pełni podwójną rolę - w czasie nadawania jest buforem nadawczym, a w czasie odbioru buforem odbiorczym.

Montaż i uruchomienie

Montaż według schematu montażowego pokazanego na rys. 2 wykonujemy rozpoczynając od elementów o najmniejszych gabarytach (rezystory, diody, podstawki pod układy scalone), a następnie montujemy kondensatory i złącza CON1...CON3. Po zmontowaniu ze sprawnych elementów układ jest gotowy do pracy. Dla pewności przed włożeniem układów

WYKAZ ELEMENTÓW

Rezystory

R1: 10kΩ

RN1: R-pack - 10kΩ

Kondensatory

C1, C2: 33pF

C3, C4: 100nF

C5: 100μF/16V

C6: 220μF/16V

C7...C11: 10μF/16V

Półprzewodniki

D1: 1N4004

US1: AT89C2051 zaprogramowany

US2: MAX232

US3: 78L05

Różne

CON1: ARK2(3,5mm)

CON2: goldpin 5*2

CON3: DB9M do druku

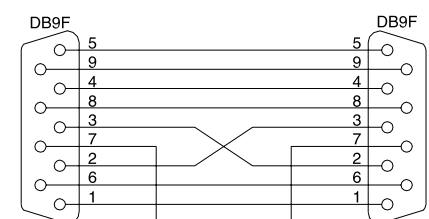
JP1...JP4: goldpin 1*2

Q1: kwarc 11,059MHz

scalonych można sprawdzić, czy napięcie zasilające układy wynosi 5V (uprzednio do złącza CON1 należy dołączyć napięcie 9V). Połączenie płytki z komputerem wykonujemy kablem zakończonym z obydwu stron złączami DB9-żeńskimi. Kable takie są wykorzystywane do połączenia dwóch komputerów. Ważne jest, aby posiadał przeplot pomiędzy wyprowadzeniami „2” i „3”. Schemat połączeń umożliwiający samodzielne wykonanie takiego kabla jest przedstawiony na rys. 3. Po wykonaniu powyższych czynności układ jest gotowy do współpracy z dowolnym komputerem.

Krzysztof Pławiuk, AVT
krzysztof.plawiuk@ep.com.pl

Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: <http://www.ep.com.pl/?pdf/lipiec02.htm> oraz na płycie CD-EP07/2002B w katalogu PCB.



Rys. 3. Sposób wykonania kabla