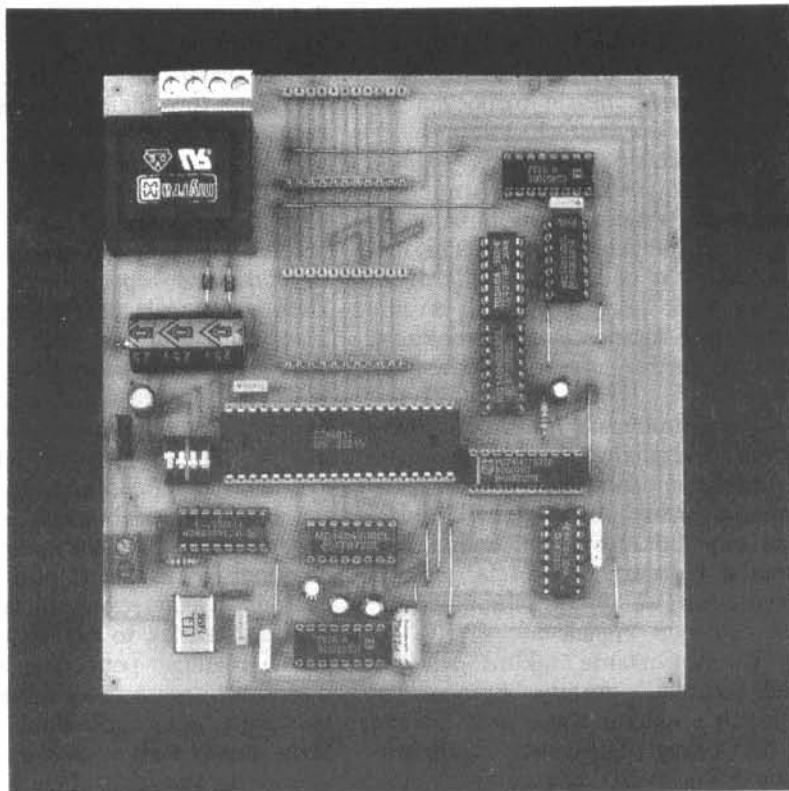


Emulator sprzęgu równoległego na porcie szeregowym

Wykonanie równoległego wyjścia z mikrokomputera przy użyciu portu jego drukarki nie jest w zasadzie problemem, ale wykonanie wejścia równoległego jest już sprawą bardziej złożoną. Istnieją oczywiście różne rozwiązania, wymagają one jednak albo użycia dwukierunkowego sprzęgu drukarki albo modyfikacji istniejącego sprzęgu, co z kolei pociąga za sobą pewne ryzyko. Poza tym, rozwiązania takie są oczywiście niewykonalne dla mikrokomputerów, które w ogóle nie posiadają równoległego 8-bitowego sprzęgu drukarki, zwanego również złączem „Centronics“.



Proponujemy więc wykonanie układu prostego i taniego, dającego się przyłączyć za pośrednictwem portu szeregowego RS232 do każdego mikrokomputera, i to nie tylko kompatybilnego z PC. Układ ten pozwala na wygodne i bezpieczne uzyskanie od 1 do 64 połączeń równoległych, bez żadnego złożonego oprogramowania. Możliwości tego układu są zachęcające, mimo że nie zastosowano w nim ani mikrosterownika, ani jakiegokolwiek zaprogramowanego podzespołu, każdy więc może z łatwością go wykonać. Przy jego konstrukcji posłużono się bowiem, co prawda w wymyślny sposób, jedynie zwykłymi układami logicznymi.

Asynchroniczne połączenia szeregowo

Przed przystąpieniem do omówienia zasady działania układu,

warto pokrótce opisać asynchroniczną transmisję szeregową oraz służące do niej sprzęgi, zwane UART.

W przeciwieństwie do połączeń równoległych, w których każdy bit przesyłany jest osobnym przewodem, do połączeń szeregowych używa się jednego przewodu. Bity informacji są nim przesyłane kolejno, jeden po drugim, oczywiście w regularnym rytmie.

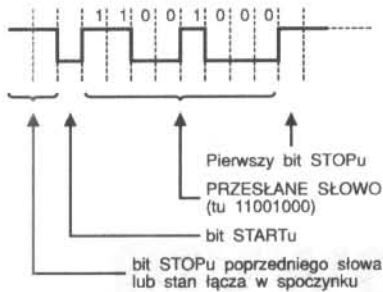
Urządzenia znajdujące się po obu końcach połączenia muszą się rozumieć, trzeba więc zachować pewne warunki. I tak, trzeba aby urządzenia wysyłające i odbierające:

- stosowały ten sam sposób kodowania danych,
- w jednakowy sposób traktowały ich kolejność (na przykład starsze bity albo na początku, albo na końcu),
- pracowały z tą samą szybkością,

- były ze sobą odpowiednio zsynchronizowane, aby urządzenie odbierające wiedziało, w którym momencie rozpoczyna się nadawanie.

Łatwo spełnić dwa pierwsze warunki, ponieważ wystarczy tylko uzgodnić zasady nadawania i odbioru. Dwa pozostałe są nieco bardziej złożone i właśnie dlatego, zależnie od sposobu ich realizacji, transmisje dzielą się na asynchroniczne i synchroniczne.

W przypadku synchronicznej transmisji szeregowej nadawnik wysyła do odbiornika sygnał zegarowy. Sygnał ten może być przesyłany samodzielnie albo razem z danymi, zależnie od przyjętej metody. Można upewnić się w ten sposób, że nadawnik i odbiornik nie tylko będą działały z tą samą szybkością, ale także będą pracowały synchronicznie (stąd nazwa nadana



Rys. 1. Przesyłanie jednego znaku łączem szeregowym

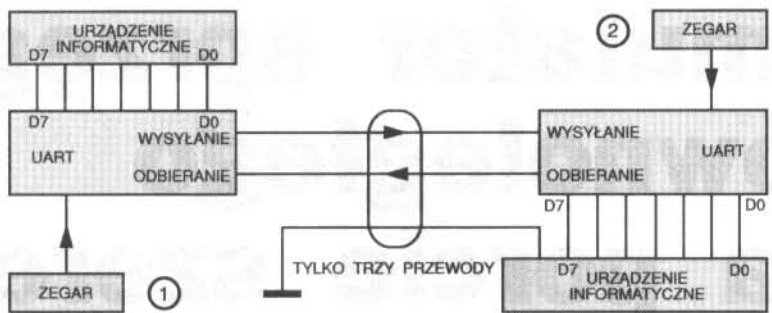
temu rodzajowi transmisji) i że w tym samym czasie rozpoczną obróbkę przekazywanych informacji.

Konieczność przesyłania sygnału zegarowego stwarza wiele trudności, zwłaszcza w łączności na duże odległości, na przykład przy użyciu modemów. W celu ich uniknięcia pomysłowi ludzie wymyślili zasady szeregowej łączności asynchronicznej, obecnie bardzo szeroko rozpowszechnionej.

Asynchroniczna transmisja szeregową uwzględnia omówione już warunki sposobu kodowania i kolejności przesyłania danych. Urządzenia uczestniczące w połączeniu pracują z tą samą szybkością, ale ich zegary nie są już zsynchronizowane zewnętrznym sygnałem.

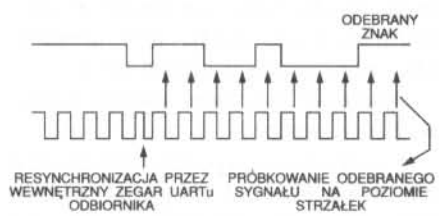
Do sterowania takimi właśnie połączeniami zostały opracowane specjalne układy sprzęgające, zwane UART (ang. Universal Asynchronous Receiver-Transmitter - uniwersalny asynchroniczny nadajnik-odbiorca). Układy te są interfejsami, zawierającymi oprócz rejestrów przesuwanych wszystkie układy potrzebne do generacji poszczególnych sygnałów asynchronicznej transmisji szeregowej. Jak sama nazwa wskazuje, wszystkie UART są podwójne, czyli zawierają i nadajnik i odbiorca. Do transmisji dwukierunkowej wystarczy więc jeden UART.

Cała sztuka asynchronicznej transmisji szeregowej polega na formie wysyłanych sygnałów, które pozwalają resynchronizować odbiorcę każdym odebrany znak. Widać to na rys. 1, przedstawiającym asynchroniczną transmisję bajtu 11001000. W czasie spoczynku linia przesyłająca jest w stanie wysokim. Transmisja zaczyna się przejściem linii w stan 0 na czas jednego cyklu zegarowego, co tworzy bit startu. Bity przesyłanego



Rys. 2. Schemat blokowy szeregowego łącza synchronicznego

słowa, czyli bity „użyteczne“, są wysyłane po bicie startowym, po czym linia przechodzi znowu w stan wysoki na czas jednego czy dwóch cykli zegarowych, tworząc bit lub bity stopu. Mówiąc inaczej, każdy przesyłany znak jest zawarty pomiędzy dwoma specjalnymi bitami, które wyznaczają jego początek i koniec. Bity te są automatycznie wstawiane i usuwane przez układy UARTu. Zegary, nadawczy i odbiorczy, są wbudowane bezpośrednio w UARTy (jak to widać na rys. 2). UART odbierający synchronizuje się odebrany sygnałem przy pomocy bitu startu i rozpoczyna analizę każdego bitu danych pośrodku teoretycznego czasu jego trwania, tak jak to pokazuje rys. 3. Jeżeli różnica pomiędzy częstotliwościami zegara nadawczego i odbiorczego jest niewielka, wówczas przesunięcie fazy w czasie transmisji słowa nie jest zbyt duże i połączenie przebiega pomyślnie. Trzeba wyjaśnić, że częstotliwości zegarowe stosowane w UARTach nie są równe częstotliwości transmisji, lecz zazwyczaj 16 krotnie wyższe. Ułatwia to działanie wewnętrznych układów logicznych urządzenia, a zwłaszcza synchronizację fazy próbkowania odebranych bitów pośrodku ich teoretycznego czasu trwania. W licznych systemach asynchronicznej łączności szeregowej wprowadzono poza tym pojęcie parzystości i wysyła się związany z nim dodatkowy bit.



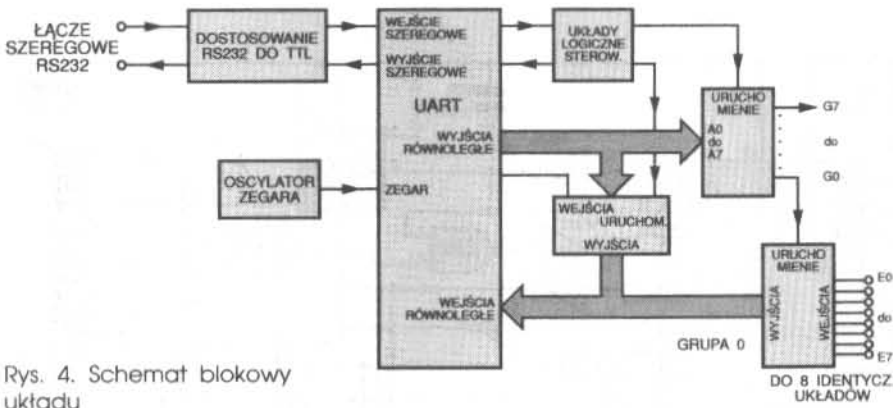
Rys. 3. Sposób synchronizacji i próbkowania danych

Zasada stosowania bitu parzystości jest następująca: zlicza się użyteczne bity niezerowe wysyłanego znaku (to znaczy tylko bity znaku, bez startu i stopu) i zależnie od tego czy jest to ilość parzysta czy nie, bit parzystości otrzymuje wartość 1 lub 0. Po dokonaniu tej czynności przed emisją bit parzystości zostaje wysłany jako część nadawanego znaku. Przy odbiorze wykonywane są takie same operacje i tak obliczony bit jest porównywany z odebrany. Jeżeli się różnią, to w trakcie transmisji niewątpliwie powstał błąd, jeżeli zaś są jednakowe, to wprawdzie nie stanowi to dowodu całkowitej poprawności połączenia (dwa bity mogły równocześnie zmienić wartość na przeciwną), daje to jednak pewną informację o jakości transmisji.

Porównywana może być parzystość albo nieparzystość. Parzystość 1 oznacza, że ilość bitów użytecznych o wartości 1 w przesłanym znaku jest parzysta, w przeciwnym wypadku parzystość przyjmuje wartość 0. W przypadku nieparzystości jest oczywiście odwrotnie.

Bity parzystości na żądanie są generowane automatycznie przez UART przed wysłaniem znaku i automatycznie sprawdzane przez UART po jego odebraniu. Błąd parzystości jest sygnalizowany zmianą stanu jednego z wyprowadzeń sprzęgu z 1 na 0.

Obecność i wykorzystanie bitów parzystości nie jest bezwzględnie konieczna, można nawet uruchomić łączność z inną parzystością przy nadawaniu i inną przy odbiorze. Odbiorca wykaze wtedy permanentny błąd parzystości, jeżeli jednak program jest dobrze napisany, nie zablokuje się i pomimo tego odbierze wszystkie znaki poprawnie. Jest nawet możliwe wysyłanie zna-



Rys. 4. Schemat blokowy układu

ków parzystości do urządzeń niezdolnych do ich wykorzystania. Oczywiście, należy w miarę możliwości programować te same parzystości przy nadawaniu i odbiorze, aby zachować pełną użyteczność tego bitu.

Wszystko jest zupełnie jasne, jeżeli obsługuje się port szeregowy mikrokomputera przy pomocy właściwego programu i odpowiedniego urządzenia peryferyjnego. Istotnie, program sterownika portu szeregowego lub urządzenia peryferyjnego sam zajmuje się dialogiem z UARTem, znajdującym się na płycie sprzętu mikrokomputera. Jeśli pominąć błędy transmisji, jego działanie jest bardzo proste.

Schemat blokowy sprzętu

Szczegółowy schemat blokowy opisywanego układu jest przedstawiony na rys. 4. Sygnały z asynchronicznego łącza szeregowego z mikrokomputera są doprowadzone najpierw do interfejsu RS232 - TTL. Jak wszyscy z pewnością wiedzą, ogromna większość połączeń tego typu dokonywana jest zgodnie z normą RS232, konieczna jest zatem konwersja poziomów. Trzeba tu przypomnieć, że łączność jest dwukierunkowa i układ nie zadowolą się wysyłaniem danych do komputera, lecz będzie z nim prowadził dialog.

Następnie dane, już w standardzie TTL, są przekazywane do UARTu, ponieważ on właśnie jest właściwym odbiornikiem. Otrzymuje on oczywiście także sygnał zegarowy o przelączanej częstotliwości, co umożliwi pracę z dobraną według potrzeby szybkością transmisji.

Przed dalszym omawianiem schematu blokowego warto przeanalizować, w jaki sposób przebiega faza

odbioru danych. Pozwoli to łatwiej zrozumieć zasadę działania związanych z UARTem układów logicznych. Maksymalna możliwa ilość 64 wejść jest podzielona na 8 grup po 8, ponumerowanych od 0 do 7. Aby odczytać stan jednej z grup, mikrokomputer wysyła łączem szeregowym do układu numer wybranej grupy. Układ odpowiada wtedy dwoma kolejnymi słowami. Pierwsze słowo oznacza stany ośmiu linii wybranej grupy, zaś drugie jest słowem kontrolnym jakości połączenia. Zawiera ono odwrotność numeru wybranej grupy i bit parzystości UARTu. Odebranie poprawnego słowa kontrolnego pozwala upewnić się o poprawności połączenia. Jest to bardzo użyteczne, zwłaszcza przy pracy w warunkach silnych zakłóceń (na przykład w przemyśle, w pobliżu pracujących maszyn). Dane równoległe wychodzące z UARTu przenoszą numer grupy, która ma zostać odebrana. Zostają one przesłane do jednego z ośmiu dekodatorów, który z kolei uruchamia jeden z ośmiu układów danych.

Informacje z wyjść tak uruchomionego układu danych są przenoszone na wejścia równoległe UAR-

Tu w celu wysłania ich do mikrokomputera. Równoległe z tym dane z wyjść równoległych UARTu, wraz z oznaczeniem błędu parzystości, zostają ponownie wprowadzone na wejścia równoległe za pomocą układu uruchomionego przez system sterowania logicznego.

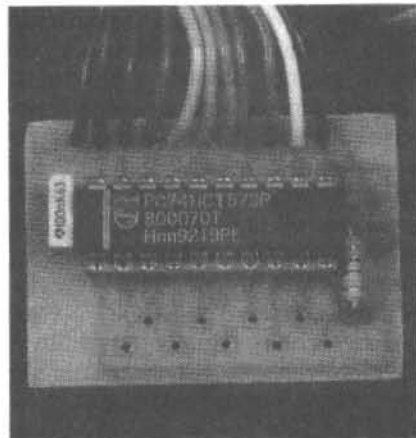
Ten system sterowania realizuje następujące funkcje. Po odebraniu jednego słowa danych uruchamia układ wejściowy wybranej grupy i wydaje UARTowi polecenie wysłania danych które otrzymał na swoich wejściach. Następnie unieruchamia ten układ, uruchamia zaś układ wysyłania odebranych danych wraz z parzystością i ponownie poleca UARTowi wysłanie danych. W ten sposób, jak już wspomniano, są wysyłane do mikrokomputera dwa kolejne słowa.

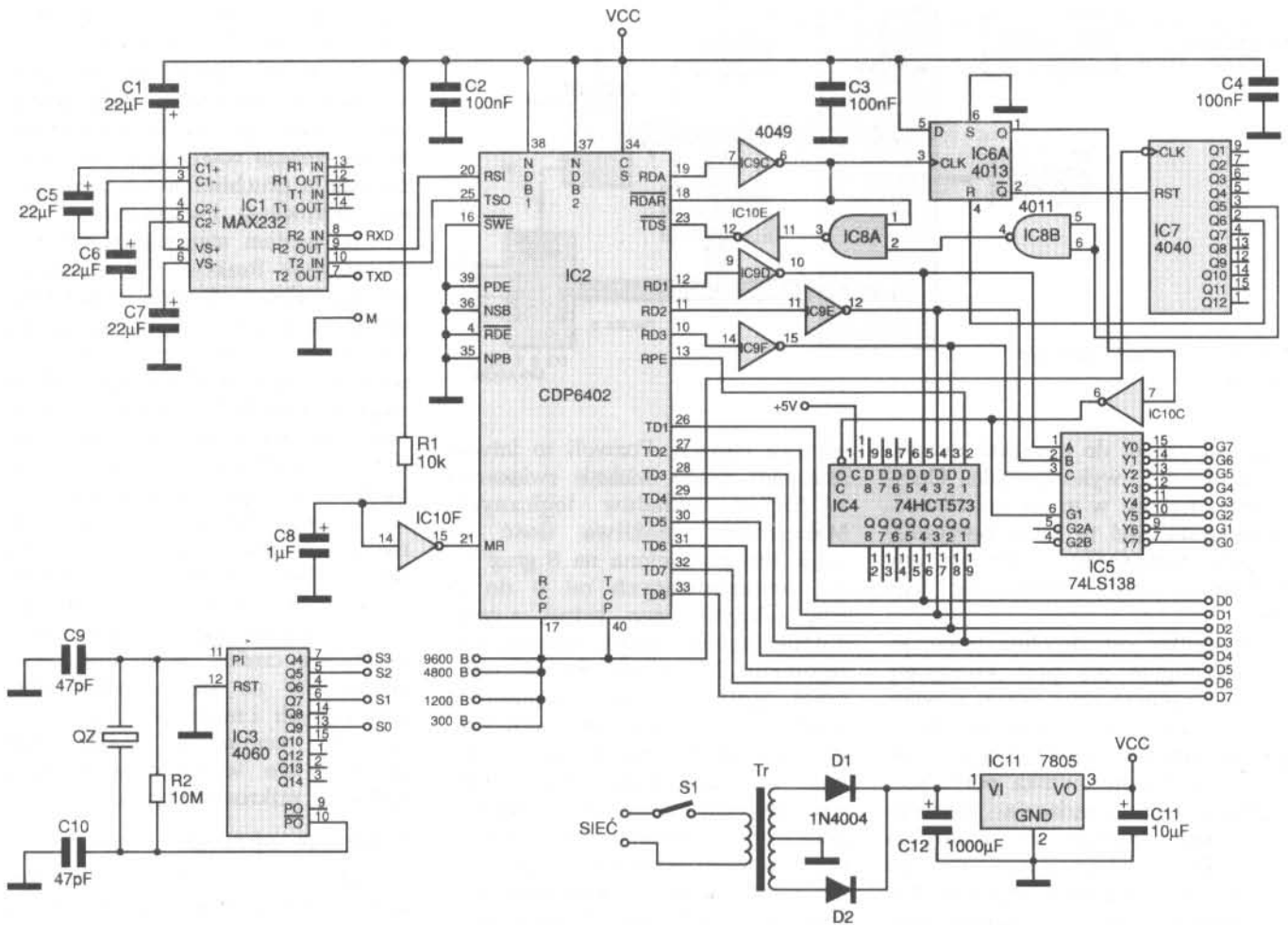
Ten system logiczny jest najdelikatniejszą częścią całego urządzenia; składa się na niego tylko kilka zwyczajnych i niedrogich logicznych układów scalonych CMOS, jest prosty i nie wymaga stosowania żadnego mikrosterownika.

Schemat sprzętu

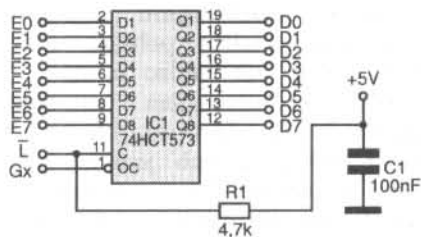
Schemat elektryczny głównej części sprzętu przedstawiony jest na rys. 5, zaś schemat modułu grupy, sprowadzającego się do jednego układu scalonego, na rys. 6. Po uważnym prześledzeniu schematu blokowego, analiza pełnego schematu będzie już bardzo łatwa.

IC1 jest sprzężeniem pomiędzy RS232 i TTL. Jest to klasyczny układ MAX232, lub jeden z jego wielu odpowiedników, pozwalający zasilać całe urządzenie napięciem 5 V, dysponując pomimo tego na wyjściu poziomami właściwymi dla RS232. Warto tu przypomnieć, że MAX232 zawiera w sobie dwa (zasilane napięciem +5V) przetworniki napięcia stałego na stałe, dostarczające napięcie +9V i -9V. Są to przetworniki o przelączanych pojemnościach; w tym wypadku tymi pojemnościami są C1, C5, C6 i C7. Właściwy UART to IC2. Część jego wyprowadzeń sterujących jest połączona z masą, a część z +5V, celem skonfigurowania zgodnego z zastosowaniem. W ten sposób jest on na stałe zaprogramowany do transmisji 8-bitowych danych, 1 bitu stopu i nieparzystości. Powinno to zostać uwzględnione przy konfigurowaniu portu szeregowego mik-





Rys. 5. Pełny schemat elektryczny płytki podstawowej



Rys. 6. Schemat modułu wejściowego

rokomputera, z którym będzie on połączony.

Źródłem impulsów zegarowych nadawania i odbioru jest IC3 (CMOS 4060) sterowany kwarcem 2,4576MHz. Z wyprowadzeń Q4 do Q9, za pośrednictwem przełącznika, można odbierać częstotliwości odpowiadające znormalizowanym szybkościom transmisji 300, 1200, 4800 i 9600 bodów, co pozwala zaspokoić wszystkie potrzeby.

Odebrane przez UART dane otrzymuje się z wyprowadzeń odbiorczych RD1...RD3. Zawierają one do odczytania numer grupy wejść. Po

inwersji w IC9D, E, F dane te wystawiają układ IC5, który jest przetwornikiem 1 na 8. Uruchomiony za pośrednictwem wejścia G1, przerzuca w stan niski to wyjście spośród Y0...Y7, które zostało wybrane przez odebrane dane. Ta linia z kolei aktywizuje odpowiedni układ wejściowy, przedstawiony na rys. 6. Jest nim 74HCT573, czyli zespół 8 zatrząsków o trójstanowych wyjściach, na które oddziałują się poleceniem przejścia w trzeci stan. W ten sposób do urządzenia można dołączyć równocześnie do ośmiu takich układów, które są uruchamiane na żądanie za pośrednictwem wyjść IC5. Do dyspozycji jest więc 64 wejść, które można czytać 8 razy po 8. Wyjścia tych zatrząsków są przyłączone do wyprowadzeń nadawczych TD1...TD8, przez które wprowadza się dane wysyłane przez UART.

Odwrócone sygnały wyjściowe IC2 kierowane są równocześnie do IC4, takiego samego układu zatrząskowego, który otrzymuje oprócz te-

go z UARTu sygnał RPE, zawierający oznaczenie błędu parzystości. Z chwilą gdy IC4 zostaje uruchomiony za pośrednictwem swojego wejścia OEN, UART otrzymuje na swoje wejścia nadawcze odwrotność numeru wybranej grupy i ewentualną informację o błędzie parzystości. Logiczny system sterowania UARTem składa się z IC6, IC7 i kilku układów odwracających. Wykorzystuje on trzy linie sterowania stanem UARTu:

- RDA, która przejściem w stan wysoki sygnalizuje, że za pośrednictwem łącza szeregowego został odebrany znak,
- RDAR; przez sprowadzenie tej linii do stanu niskiego można wyzerować RDA,
- TDS; sprowadzając tę linię do stanu niskiego można spowodować wysłanie przez UART danych na TD1...TD8.

Nie wchodząc w szczegóły wykresów czasowych, których analizę pozostawia się Czytelnikowi, zasady tego systemu logicznego są nastę-

pujące. Z chwilą otrzymania numeru wybranej grupy RDA przechodzi w stan wysoki i natychmiast zostaje wyzerowany poprzez IC9C i RDAR\ . Wywołuje to wysłanie poprzez IC8A i IC10E bardzo krótkiego impulsu do TDS\ , który poleca UARTowi dokonanie pierwszej emisji.

W tym czasie IC6 i IC7 były w spoczynku. IC7 był blokowany przez wyjście Q\ IC6, podczas gdy wyjście Q było w stanie niskim, skutkiem czego IC5 był utrzymywany w stanie czynnym za pośrednictwem IC10C. Dane z układu wejściowego (grupy wybranej przez RD1...RD3 i IC5) pojawiają się więc na TD1...TD8.

Wyzerowanie RDA wywołuje generację narastającego zbocza na CLK w IC6, co wywołuje zmianę jego stanu i uruchamia licznik IC7.

Wyjście Q układu IC6 przechodzi więc w stan wysoki i blokuje IC5 oraz uruchamia IC4. Linie RD1...RD3 i informacja o błędzie

parzystości są podawane na TD1...TD4, a gdy IC7 zliczy 32 impulsy zegarowe, jego wyjście Q5 przechodzi w stan wysoki i wysyła impuls na TDS\ . Słowo kontrolne zostaje więc wyemitowane. Po następnych 32 impulsach zegarowych wyjście Q6 IC7 przechodzi w stan wysoki, wyzerowuje IC6 i pozostawia system logiczny w oczekiwaniu na następną wymianę sygnałów.

Zasilanie jest klasyczne, z transformatorem małej mocy i stabilizatorem scalonym +5V.

Układy wejściowe, o których wspomniano wyżej i których schemat pokazany jest na rys. 6, dysponują możliwością zapamiętywania danych poprzez swoje wejście L\ . Gdy wyprowadzenie to jest na poziomie wysokim, wymuszonym przez rezystor R1, zatrzaśki w nim zawarte są przeźrocyste i ich wyjścia śledzą wejścia. Gdy L\ jest w stanie niskim, zatrzaśki są zamknięte i stany wejść z czasu przed zmianą stanu L\ na niski

zostają zapamiętane na wyjściach. Od użytkownika zależy, czy te informacje będą wykorzystane.

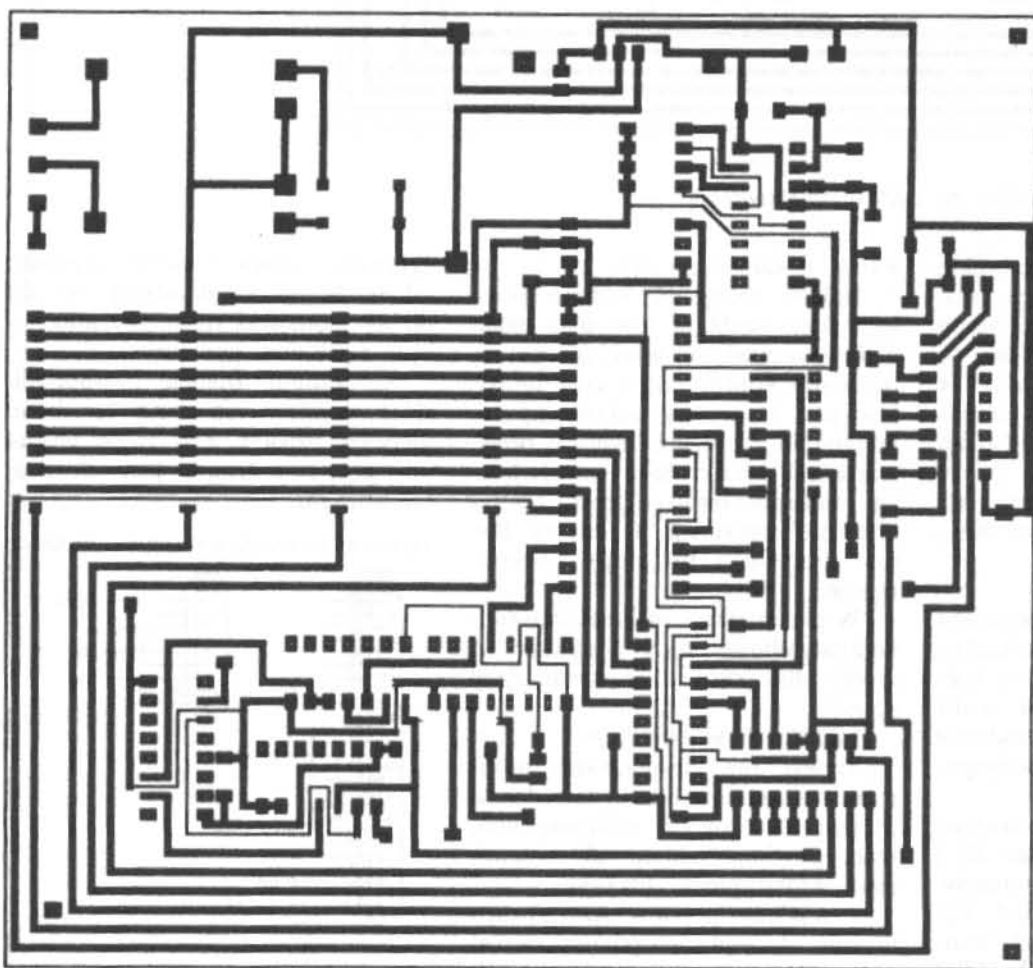
Wykonanie

Układ zaprojektowano jako modułowy, aby każdy mógł go łatwo przystosować do swoich celów. Często przydaje się 8 lub 16 wejść równoległych, ale nie wszyscy potrzebują 64. Podstawowa płytkawiera wszystkie elementy z rys. 5, na której można pomieścić do 4 modułów 8 wejściowych. Jeżeli potrzebnych będzie 64 wejść, przewidziano złącze do przyłączenia dalszych 4 modułów zewnętrznych.

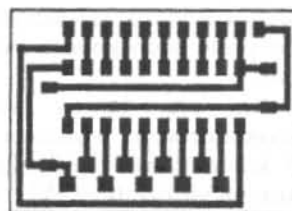
Mozaikę ścieżek płyty podstawowej pokazuje rys. 7, a modułu wejściowego rys. 8. Pomimo pewnej złożoności połączeń wybrano druk jednostronny, jako łatwiejszy do amatorskiego wykonania niż dwustronny z metalizowanymi otworami. Z tego powodu trzeba wykonać kilka zwor po stronie elementów.

Rozmieszczenie podzespołów na płytkach przedstawiają rys. 9 i 10. Montaż elementów nie powinien sprawić żadnych trudności. Układy scalone można instalować na podstawkach albo bezpośrednio w płytce. Warto jednak polecić użycie podstawek dla 4 lub 8 układów wejściowych oraz CI1 na karcie podstawowej.

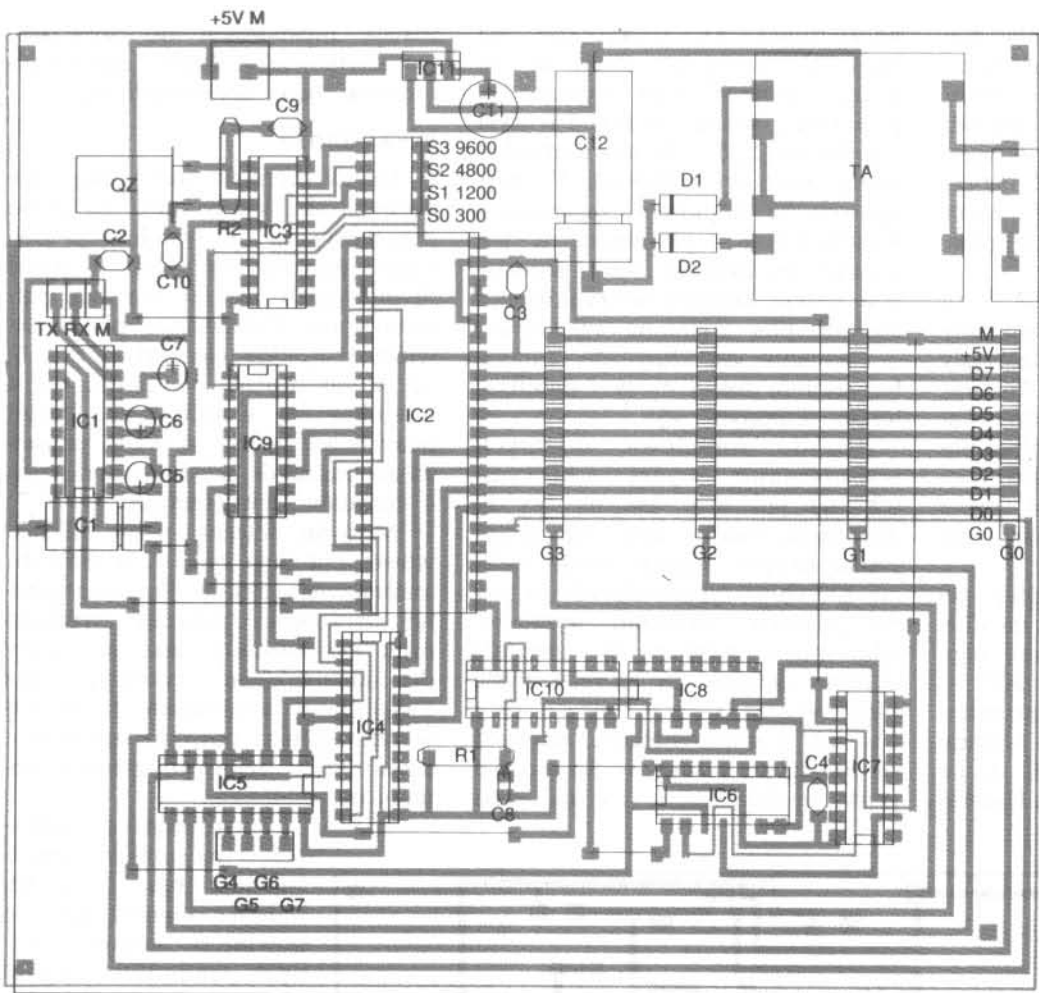
Układy wejściowe można połączyć z płytką podstawową w różny sposób, zależnie od potrzeby. W prototypie użyto złącza grzebieniowego, widocznego na fotografii, można też zastosować krótkie połączenia sztywnym przewodem lub dłuższe, choć nie więcej



Rys. 7. Mozaika ścieżek płytki podstawowej



Rys. 8. Mozaika ścieżek płytki modułu wejściowego



Rys. 9. Rozmieszczenie elementów na płycie podstawowej

niż kilkucentymetrowe, miękkie przewody taśmowe. Na płycie przewidziane są połączenia dla grup o numerach od 0 do 3, jak to widać na rys. 9. Dla grup od 4 do 7 przewidziane są zaciski umieszczone koło CI5. G7 na Y0 i G0 na Y7 w IC5 nie powinny być zaskoczeniem; wynika to z odwracania sygnałów RD1 do RD3 przez IC9.

Próby i użytkowanie

Po starannej kontroli montażu i sprawdzeniu działania stabilizatora 5V, można przejść do bardziej konkretnych testów. W tym celu wystarczy tylko przyłączyć urządzenie do portu szeregowego mikrokomputera.

Na rys. 11 są przedstawione znormalizowane połączenia złącz 25 i 9 kontaktowych mikrokomputerów kompatybilnych z PC. Układ nie wysyła sygnałów kontrolnych transmisji RS232 (RTS, DCD, CTS i t.d.), trzeba więc „oszukiwać” mikrokomputer, wykonując na złączach

zwory pokazane na rys. 11.

Należy następnie wybrać przełącznikiem szybkość transmisji (9600 bodów, oczywiście, jeżeli nie jest za duża dla komputera) i skonfigurować port szeregowy mikrokomputera zgodnie z formatem danych przetwarzanych w układzie, uwzględniając zaprogramowaną przez S0...S3 szybkość transmisji, 8 bitowy format danych, 1 bit stopu i nieparzystość.

W przypadku mikrokomputera kompatybilnego z PC można to zrobić dla portu 2 w następujący sposób:

MODE COM2:9600,o,8,1.

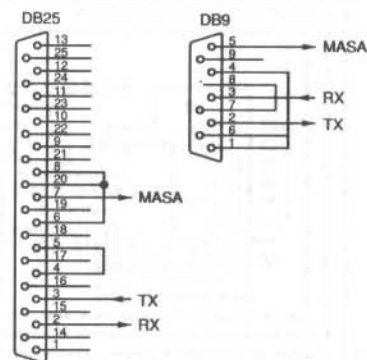
Trzeba też dysponować minimum oprogramowaniem do zarządzania portem szeregowym mikrokomputera, które powinno realizować następujące bardzo proste funkcje:

- wysłanie z portu szeregowego (a więc do wykonanego urządzenia) numeru grupy wejściowej do odczytania. Numer ten jest zakodowany w trzech młodszych bitach wy-

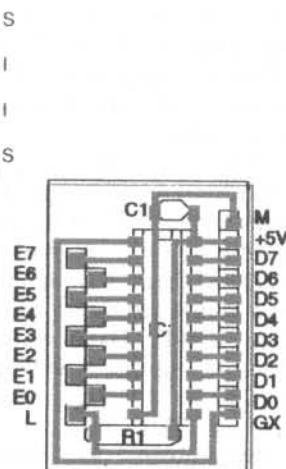
syłanego słowa i może wynosić od 0 do 7, a dokładniej od X0 do X7, ponieważ tylko trzy młodsze bity są brane pod uwagę,

- odbiór dwóch następnych słów w odpowiedzi na te dane: pierwsze oznacza stan wejść wybranej grupy a drugie jest słowem kontrolnym.

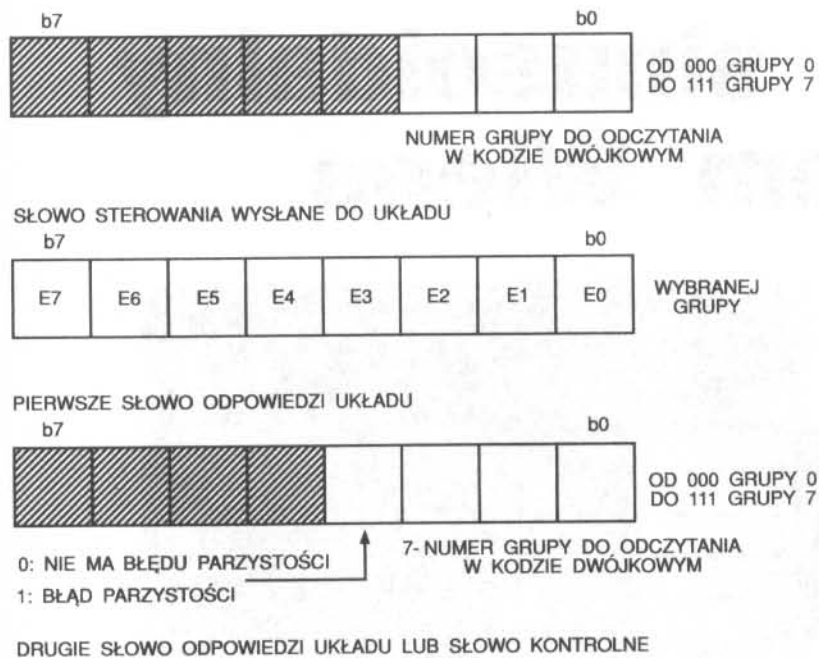
ZWORY, KTÓRE NALEŻY WYKONAĆ NA GNIAZDKACH



Rys. 11. Połączenia złączy 9- i 25-kontaktowych, używanych w łączach szeregowych RS232



Rys. 10. Rozmieszczenie elementów na płycie modułu wejściowego



Rys. 12. Sposób kodowania słów wysyłanych i odbieranych przez układ

Słowo stanu wejść jest zakodowane na wejściu (oznaczonym na rys. 6 i 10 E7 w starszym bicie i E0 w młodszych bitach).

W słowie kontrolnym trzy młodsze bity zawierają uzupełnienie do 7 numeru wybranej grupy (7 dla grupy 0, 6 dla grupy 1 i tak dalej). Czwarty bit przyjmuje wartość 0, gdy przez UART nie został wykryty żaden błąd parzystości i 1 - w przypadku przeciwnym.

Powyższe warunki są przedstawione graficznie na rys. 12. Do sterowania urządzeniem można oczywiście użyć dowolnego języka programowania. Działanie programu jest bardzo proste i sprowadza się do wysłania słowa do portu szeregowego i do zdekodowania dwóch słów otrzymanych w odpowiedzi. Załączony listing (rys. 13) przedstawia przykładowy program napisany w Microsoft QuickBasic, który pozwala łatwo przetestować zmontowany układ za pomocą mikrokomputera zgodnego z PC, pracującego pod kontrolą systemu operacyjnego MSDOS. W programiku tym zastosowano jedynie najprostsze instrukcje, może więc on zostać z łatwością przetłumaczony na dowolny inny język dla dowolnego mikrokomputera.

Przewidziana w nim została możliwość pracy zarówno przez port 1 jak i port 2 oraz wybór grup jedynie od 0 do 3, czyli tych, które są zainstalowane na karcie.

Wystarczy jednak zmodyfikować testy, które następują po odpowiednich pytaniach, aby poszerzyć ten wybór. Trzeba też zwrócić uwagę, że wyjście znaku z portu szeregowego odbywa się w mało elegancki sposób, przez OUT prosto do rejestru wyjściowego sprzęgu szeregowego PC. Natomiast odebranie odpowiedzi i ich odczyt następuje za pomocą OPEN COM i INPUT\$. Trzeba też pamiętać, że zależnie od kolejności uruchomienia programu i włączenia napięcia w urządzeniu, pierwsza odpowiedź może być nieprawidłowa. Po następnej wymianie sygnałów wszystko już będzie w porządku.

ERP

```

REM Program testujący sprzęg równoległy portu szeregowego
REM Copyright C. Broussas i ERP
N% = 0
CLS
port: INPUT "Do jakiego portu szeregowego układ jest przyłączony? (1 lub 2) ", P%
IF P% = 1 THEN GOTO port1
IF P% = 2 THEN GOTO port2
GOTO port
port1: OPEN "COM1" FOR INPUT AS #1
ADPORT = &H3F8
GOTO question
port2: OPEN "COM2" FOR INPUT AS #2
ADPORT = &H2F8
question: INPUT "Która grupa ma zostać odczytana? (0, 1, 2, 3, albo 4 aby wyjść): ", N%
IF N% < 0 OR N% > 4 THEN GOTO question
IF N% = 4 THEN END
OUT ADPORT, N%
R1$ = INPUT$(1, 1)
R2$ = INPUT$(2, 1)
CONTROL0 = ASC(R2$)
CONTROL1 = CONTROL0 AND &HF
CONTROL2 = 7 - CONTROL1
IF CONTROL2 > 3 THEN GOTO error
PRINT "Dana "; HEX$(ASC(R1$)); " w grupie "; CONTROL2
GOTO question
error: PRINT "Błąd połączenia - program zakończony"
END
    
```

Rys. 13. Listing przykładowego programu testującego układ

WYKAZ ELEMENTÓW

Płyta główna

Rezystory 0,25W 5%

R1: 10kΩ

R2: 10MΩ

Kondensatory

C1, C5, C6, C7: 22μF/15V, elektrolityczny

C2, C3, C4: 0,1μF, mylarowy

C8: 1μF/63V, elektrolityczny

C9, C10: 47pF, ceramiczny

C11: 10μF/15V, elektrolityczny

C12: 1000μF/25V, elektrolityczny

Półprzewodniki

IC1: MAX232, ICL232 lub odpowiednik

IC2: AY5-1013, COM8502,

COM8017, CDP6402 lub odpowiednik

IC3: 4060

IC4: 74HCT573

IC5: 74LS138

IC6: 4013

IC7: 4040

IC8: 4011

IC9, IC10: 4049

IC11: 7805, stabilizator +5V, 1A

D1, D2: 1N4004

Różne

QZ: kwarc 2,4576MHz

TA: transformator 220V/2x9V, 2,5A,

S0 do S3: przetłacznik DIL

Moduł 8-wejściowy

Rezystory

R1: 4,7kΩ

Kondensatory

C1: 0,1μF, mylarowy

Półprzewodniki

IC1: 74HCT573