Mikroprocesorowy interfejs I²C

kit AVT-480

Opisywane urządzenie powstało z konieczności usprawnienia procesu uruchamiania większej liczby cyfrowych modułów wykorzystujących magistralę I²C (na temat samego standardu I²C, jego zalet oraz dostępnych elementów publikowano ostatnio bardzo wiele - więc nie ma potrzeby do tego wracać).

Założenia projektowe były następujące:

- zasadniczym celem funkcjonalnym jest kontrola zmontowanych modułów wykorzystujących najbardziej popularne układy zgodne ze standardem I²C (PCF 8574, 8593, 8591, SAA 1064, 24C0x itd.);
- ponieważ znane są typy użytych elementów oraz ich adresy sprzętowe, można pominąć funkcje wyszukiwania adresów i automatycznej lokalizacji dołączonych do magistrali modułów;
- w omawianych zastosowaniach wystarczy tryb Master Transmitter/Master Receiver (bez np. podsłuchu magistrali);
- konieczny jest dostęp do możliwości zapamiętania i szybkiego odtworzenia sekwencji komunikacyjnych I²C dla poszczególnych elementów i ich różnych konfiguracji (ze względu na kontrolę serii urządzeń);
- do sprawnego uruchamiania modułów oraz do badania różnych możliwości nowych elementów konieczna jest przyjazna dla użytkownika edycja (najlepiej zgodna z większością publikacji i katalogów - czyli umożliwiająca operacje na bitach);
- z tego samego powodu niezbędna jest kontrola poprawności komunikacji I²C i zgłaszanie ewentualnych błędów.

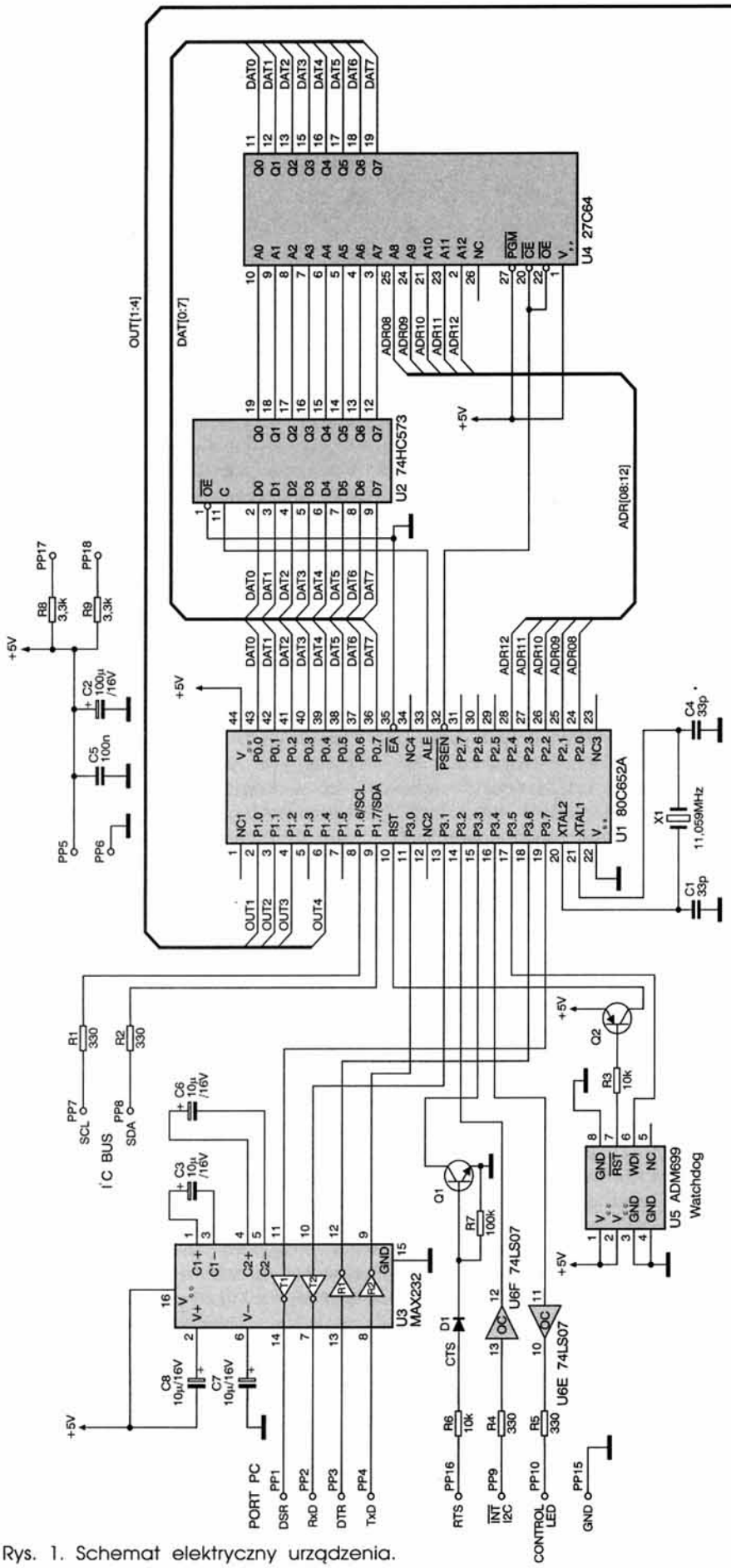
Po uwzględnieniu powyższego, sprawdzeniu zapasów elementowych oraz posiadanych możliwości sprzętowych i wykonawczych przyjąłem następujące rozwiązanie konstrukcyjne:

1. Realizacją sesji komunikacyjnych I²C z przyłączonymi modułami zajmie się mikrokontroler 80C652. Wykorzystałem do tego wewnętrzny sprzętowy interfejs I²C, który znakomicie usprawnia programową obsługę transmisji. Potrzebna długość buforów nadawczego i odbiorczego jest w przewidywanym zastosowaniu na tyle mała, że pozwala zrezygnować z zewnętrznej pamięci danych. Ze względu na uproszczenie sposobu wykonania druku, układ mikrokontrolera będzie zmontowany jako zewnętrzna przystawka na płytce jednowarstwowej.
2. Wizualizację oraz edycję i zapis poszczególnych konfiguracji zapewni sterujący program komputerowy pracujący w środowisku graficznym (Windows 3.x/95).
3. Do połączenia komputera sterującego z mikrokontrolerem będzie wykorzystana transmisja szeregowa o możliwie dużej szybkości.

Opis konstrukcji

Schemat z **rys. 1** przedstawia klasyczny układ mikrokontrolera serii MCS-51 z zewnętrzną pamięcią programu. Mniej znaczące linie adresowe ustawia bufor zatrzaskowy U2 typu '573. Program mieści się w 8kB pamięci EPROM (27C64). Oscylator jest dobrany pod kątem maksymalnej sprawności łącza RS232 - tradycyjne 11,059MHz. Interfejs RS 232C został też zrealizowany tradycyjnie - z układem U3 - MAX 232.

Ponieważ w trakcie budowy układu oprogramowanie było dopiero w sferze idei, na wszelki



Rys. 1. Schemat elektryczny urządzenia.

wypadek dodałem ile się dało dodatkowych linii sterujących:
- do komputera (wejście DSR) - drugi nadajnik w MAX232;
- z komputera (wyjścia RTS i DTR) - drugi odbiornik MAX232 oraz dodatkowe wejście (R6, D1, Q1).

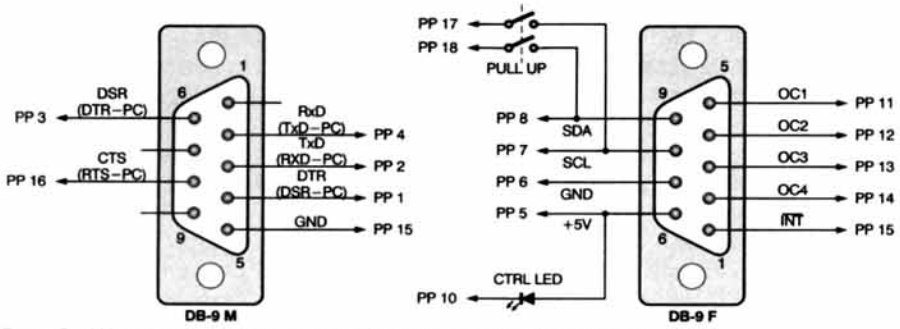
Wprawdzie w wersji 1.0 oprogramowania przystawki te dodatkowe linie nie są wykorzystane, ale mogą się przydać przy ewentualnej rozbudowie urządzenia. Dla wygody użytkowania (eliminacja konieczności stosowania ewentualnego zerowania ręcznego) wbudowałem także układ watchdoga z kostką ADM699 i odwracającym kluczem R3, Q2.

Jedno wejście przerwań zewnętrznych procesora jest podłączone poprzez układ buforujący R4 i 1/6 '07 do linii przerwań magistrali I²C (niektóre kostki, np. 8574 potrafią zgłosić przerwanie) - jednak w obecnej wersji oprogramowanie nie obsługuje tych przerwań. Drugi wzmacniacz (U6E) układu 74LS07 wykorzystałem (z rezystorem R5) do obsługi zewnętrznej diody kontrolnej LED (pozwala na bieżąco kontrolować, czy przystawka pracuje prawidłowo).

Aby nie marnować pozostałych czterech wzmacniaczy są one wykorzystane jako drivery sterowanych programowo wyjść typu otwarty kolektor (uniwersalne wyjścia cyfrowe) - jest to dodatkowa funkcja przystawki i nie ma nic wspólnego z obsługą magistrali I²C.

Rezystory R1 i R2 zapewniają tłumienie zakłóceń w.cz., natomiast R8 i R9 realizują funkcję „pull-up", czyli zasilanie linii SDA i SCL. Może być ono - w zależności od potrzeb - włączone lub wyłączone zewnętrznym przełącznikiem.

Kondensatory C5 i C2 filtrują zasilanie przystawki. Założyłem zasilanie z warsztatowego źródła +5V. Wynikło to z chęci uproszczenia podłączeń przy testowaniu modułów właśnie tak zasilanych. Nic nie stoi jednak na przeszkodzie, aby wbudować własny stabilizator i kontrolę biegunowości. W prototypie zabezpieczenie przed odwrot-



Rys. 2. Wyprowadzenia sygnałów na złącza interfejsu.

nym podłączeniem i przed zbyt wysokim napięciem zmontowałem w obudowie wtyku DB-9, użytego do łączenia z zasilaczem i magistralą I²C - składa się ono z włączonej szeregowo diody Schottky'ego oraz diody Zenera 5V1. Nie wpływa to w żadnym stopniu na pracę przystawki, ale chroni przed efektami własnych pomyłek.

Na schemacie zwraca uwagę duża liczba punktów lutowniczych PP. Wynika to z przyjętych założeń konstrukcyjnych:
- montaż w uniwersalnej obudowie plastykowej typu Z-VII-A;
- kontrolny LED oraz przełącznik „pull-up" wklejone w pokrywę obudowy (Poxipol);
- jako gniazda RS 232 oraz magistrala/zasilanie/wyjścia OC: DB-9 (męskie dla RS i żeńskie dla magistrali, żeby uniknąć pomyłek) wmontowane w pokrywę obudowy;
- podłączenia do płytki przy użyciu przewodów taśmowych lutowanych do punktów lutowniczych (bez żadnych złącz).

Taki sposób połączeń jest grzechem głównym przy produkcji seryjnej. Sprawdza się za to doskonale przy wykonywaniu pojedynczych urządzeń amatorskich i radykalnie upraszcza zaprojektowanie druku (zwłaszcza przy druku jednowarstwowym!). Schemat okablowania przedstawia **rys. 2**.

Pokrywę przystawki wyposażyłem w opisy gniazd, co znakomicie poprawia walory użytkowe w trakcie wykonywania podłączeń modułów. Technologia wykonania etykiety: wydruk zafoliowany i przyklejony przy pomocy taśmy dwustronnej.

Z powyższego opisu wynika, że część sprzętowa nie jest krytyczna. Praktycznie można wykorzystać większość uniwersalnych płyt (np. minimoduł AVT z 80C652 w wersji DIL) odpowiednio zmieniając montaż mechaniczny. Cała funkcjonalność przystawki wynika z wbudowanego oprogramowania.

Opis oprogramowania

Oprogramowanie mikrokontrolera realizuje następujące funkcje:
- kasowanie watchdoga;
- migotanie diody kontrolnej;
- komunikacja z komputerem nadrzędnym poprzez łącze RS 232;
- komunikacja z testowanym elementem poprzez łącze I²C.

Praca programu jest zorganizowana w następujący sposób :
- zegar systemowy (oraz timeouty transmisji szeregowych) działa w oparciu o timer T0;
- odbiornik linii RS 232 pozostaje „na nasłuchu" w oczekiwaniu na komendę z komputera sterującego - hosta;
- przerwanie portu szeregowego 0 (UART) obsługuje załadowanie i sprawdzenie kompletności komendy, co potwierdza ustawieniem flagi, a także realizuje na polecenie pętli głównej wysłanie odpowiedzi;
- przerwanie portu szeregowego 1 (I²C) realizuje sesję komunikacyjną.

Po inicjalizacji zmiennych oraz zasobów mikrokontrolera program wchodzi w pętlę główną, składającą się z szeregu procedur, których działanie zależy od stanu flag ustawianych wewnątrz przerwań (coś w rodzaju programowania zdarzeniowego). Dzięki temu pętla ma prostą budowę i program jest prostszy w uruchamianiu (można rozwijać oddzielnie poszczególne procedury, łatwo sprawdzić, która procedura wprowadza błąd itd.).

```
;----------------
;Inicjalizacja
;----------------
;Pętla
LOOP -|CALL WATCHDOG ; KASOWANIE WATCHDOGA
      CALL UART_SERV  ; OBSLUGA KOMEND
                      ; ODBIORNIKA
      CALL I²C_SERV   ; OBSLUGA STARTU
MAGISTRALI I²C
      CALL I²C_ANS    ; OBSLUGA ODPOWIEDZI
                      ; MAGISTRALI I²C
      CALL CLK_TEST   ; OBSLUGA LED
      JMP LOOP
```

Obsługa watchdoga sprawdza poprawność konfiguracji mikrokontrolera i jeśli wszystko jest w porządku - zmienia stan linii WDI kostki ADM 699, co powoduje kasowanie wewnętrznego timera kostki:

```
WATCHDOG -|MOV A,TH1
          CJNE A,#T1_LOAD,E_WT
          MOV A,#T0_MODE
          ORL A,#T1_MODE
          CJNE A,TMOD,E_WT
          MOV A,SCON
          ANL A,#0F0H
          CJNE A,#RS0_MODE,E_WT
          MOV A,PCON
          ANL A,#80H
          CJNE A,#RS_RATE,E_WT
          CPL WATCH
E_WT -|   RET
```

Obsługa komend odbieranych z hosta działa w momencie, gdy stwierdza ustawienie flagi kompletacji nowej komendy. Flagą tą zawiaduje obsługa przerwania odbiornika, która ładuje przychodzącą komendę do bufora i sprawdza liczbę znaków. Niezbędną synchronizację początku komendy w razie wystąpienia błędów transmisji zapewnia timeout odbiornika (jeśli znaki zostaną zgubione odbiornik jest zerowany po upływie czasu timeoutu).

Format komendy jest następujący:

```
; OBSLUGA KOMEND ODBIORNIKA UART
; ODBIERANY JEST BLOK 16 BAJTOW -|OSTATNI
; BAJT TO SUMA KONTROLNA
; 15.BAJT OKRESLA RODZAJ KOMENDY
; |X|X|X|E|W4 W3 W2 W1
;|
;|E|-|USTAWIENIE OZNACZA ZE REALIZOWAC
;    TYLKO WYJSCIA
;|W1-W4 ZADANY STAN WYJSC OUT1 -|OUT4
;  (USTAWIONE >> STAN NISKI NA WYJSCIU)
;
; 12 BAJTOW JEST PRZEZNACZONYCH NA KOLEJNO
; WYSYLANE BAJTY I²C
; 13.BAJT OKRESLA LICZBE BAJTOW DO WYSLANIA
; 14.BAJT OKRESLA LICZBY: BAJTOW DO ODEBRANIA
; (MLODSZY POLBAJT)
; POZYCJE PONOWNEGO STARTU (STARSZY POLBAJT)
```

Procedura najpierw kasuje flagę kompletacji komendy (wykonanie jednokrotne), sprawdza zgodność sumy kontrolnej, sprawdza bit E. Jeśli E ustawiony to tylko przepisuje Wx do odpowiednich wyjść, jeśli nie, to także przepisuje zadane parametry sesji I²C do odpowiednich zmiennych (liczba bajtów do wysłania, położenie ponownego startu, zawartość bajtów wysyłanych, liczba bajtów danych do odebrania z testowanego elementu) i ustawia flagę rozpoczęcia sesji I²C.

Ustawienie tej flagi powoduje reakcję procedury startowej I²C: sprawdza ona czy magistrala jest wolna, zeruje interfejs I²C i wysyła na magistralę sygnał START.

Zarazem uruchamia odliczanie timeoutu sesji I²C - program będzie umiał wycofać się z sesji nawet jeśli się ona zawiesi (czasem bowiem zdarza się to bez zgłoszenia błędu magistrali, np. jeśli oczekujemy na dane od elementu, który nie umie ich wysłać). Dalej całą sesję realizują przerwania (wysłanie podanych w komendzie bajtów, ewentualne wstawienie ponownego startu i odebranie zadanej ilości bajtów). Wynikiem sesji jest ustawienie odpowiednich flag: błędu magistrali, prawidłowego zakończenia sesji, odebrania danych z elementu, ewentualnie przekroczenia timeoutu sesji. Flagi te są sprawdzane w procedurze obsługi odpowiedzi I²C.

Obsługa ta wpisuje odpowiednie bity do słowa statusu, ładuje bufor nadajnika UART, wylicza i wpisuje sumę kontrolną oraz inicjuje wysyłanie odpowiedzi do komputera sterującego (hosta). Resztą (wysłaniem całego bufora) zajmuje się obsługa przerwania nadajnika RS 232.

Format odpowiedzi jest następujący:

```
;WYSYLANY BLOK 10 BAJTOW -1OSTATNI TO SUMA
KONTROLNA
;9.BAJT OKRESLA STATUS URZADZENIA
;1X1X1X1T1D1E1I1W
;1W1-1USTAWIONO WYJSCIA
;1I1-1WYSTARTOWANO I2C
;1E1-1WYSTAPIL BLAD MAGISTRALI I2C
;1D1-1SA DANE DO ODCZYTANIA
;1T1-1TIMEOUT TRANSMISJI I2C -1NP. PRZY
BEZOWOCNYM OCZEKIWANIU NA DANE
```

Maksymalne długości buforów wynikły z potrzeb najczęściej stosowanych układów I²C. Nic jednak nie stoi na przeszkodzie aby je zwiększyć (w pamięci wewnętrznej mikrokontrolera pozostaje dużo wolnego miejsca) w razie konieczności.

Obsługa programowa zorientowanego bajtowo interfejsu I²C jest zrealizowana - zgodnie z sugestiami katalogowymi - z wykorzystaniem słowa statusu portu I²C jako wektora skoku do procedur przypisanych poszczególnym zdarzeniom na magistrali.

Obejmuje - jak zaznaczyłem w założeniach - tryb Master Transmitter i Master Receiver.

Opis tego interfejsu i sposobu jego używania można znaleźć w [1], a także m.in. w EP 5/96, w artykule Ryszarda Szymaniaka „Płytka bazowa mikrokontrolera 80C552" (procesor 80C652 ma identyczny interfejs).

Obsługa diody kontrolnej sprawdza ustawienie (w przerwaniu T0) flagi uniwersalnego sekundnika - jeśli jest flaga, kasuje ją i przełącza wyjście LED.

Program został napisany w asemblerze w postaci oddzielnych modułów dla pętli głównej, przerwań i procedur magistrali I²C. Można rzecz jasna użyć jednego modułu i nie korzystać z linkera.

Opis programu sterującego

Program sterujący został napisany przy pomocy Delphi 1.0 dla środowiska 16-bitowego (Windows 3.x). Działa również doskonale w środowisku 32-bitowym (Win95), chociaż bez jego udogodnień (jak np. długie nazwy plików). Ekran projektowałem tak, aby można było bez kłopotów z przewijaniem używać rozdzielczości VGA 640x480 (**rys. 3**).

Okno edytora komend zawiera kilka paneli i przycisków:

- lista bajtów do wysłania (12) - podwójne kliknięcie na bajcie powoduje jego przepisanie do panelu edycji;
- skojarzona z nią lista komentarzy (12) - podwójne kliknięcie na komentarzu powoduje otworzenie pola edycyjnego komentarza (powrót z wpisaniem - ENTER);
- panel bitowej edycji bajtu - poszczególne bity ustawiamy przy pomocy 8 checkboxów, edycja bitowa jest najbardziej zgodna z większością materiałów katalogowych i opisowych dotyczących elementów I²C;
- przycisk *Wpis* - przepisuje edytowany bajt z panelu edycji do podświetlonego pola w liście bajtów wysyłanych;
- nastawnik *ND* określa liczbę bajtów do wysłania w sesji;
- nastawnik *NS* określa, po którym bajcie wysłać "Następny Start" (jest to zaznaczane przy numerach bajtów wysyłanych);
- nastawnik *Odb.* określa, ile bajtów danych odebrać z testowanego elementu;
- lista bajtów odebranych (8) - jest wypełniana w przypadku odebrania danych;
- pole nastawy wyjść OC (jest to funkcja dodatkowa, nie związana z komunikacją I²C), zaznaczenie checkboxa ustawia poziom niski na odpowiednim wyjściu, zaznaczenie E ogranicza wykonanie komendy do ustawienia wyjść pomijając sesję I²C;
- przycisk *Nowy* - resetuje wszystkie pola edytora;
- przycisk *Do Pliku* - zapisuje ustawioną konfigurację do pliku *.I2C w bieżącym katalogu;
- przycisk *Z Pliku* - odtwarza w edytorze konfigurację z pliku;
- pole opisu elementu - zawiera skrócony opis konfiguracji - okienko edycji otwieramy podwójnym kliknięciem (powrót z wpisaniem - ENTER);
- linia statusu - opisuje rezultat wykonania operacji;
- przycisk TRX - wysyła do przystawki komendę z aktualnie ustawioną konfiguracją.

Program nie wymaga specjalnej instalacji, nic nie zmienia w katalogu System ani w plikach konfiguracyjnych - wystarczy go skopiować do wybranego katalogu. Uruchamiamy z parametrem określającym numer wykorzystywanego portu szeregowego np. <I²C 1> (Com1). Bez parametru program domyślnie próbuje użyć

Rys. 3. Widok okna działającego programu.

portu Com2. Jeśli portu brak lub jest zajęty, program wprawdzie się uruchamia, ale zgłasza problem z otwarciem portu - komunikacja z przystawką będzie niemożliwa.

Używanie jest intuicyjne i bardzo łatwe, dlatego nie wyposażałem edytora w pomoc On Line. Istnieje wprawdzie plik pomocy (dla Win 95), ale ma on raczej charakter materiału informacyjnego - niniejszy opis powinien być w zupełności wystarczający, zwłaszcza po kilku eksperymentach.

Szybkość transmisji PC - przystawka ustaliłem na 57600 baud - tyle można „wycisnąć" z zastosowanego mikrokontrolera. Na DX4 133MHz/Windows 3.1 łączność działa bez problemu, natomiast nie sprawdzałem jak jest na wolniejszych maszynach.

W obecnej wersji urządzenia zmiana szybkości wymaga niestety ponownego skompilowania obu programów. Ponieważ najbardziej wrażliwy jest odbiór, to sytuacja gdy element realizuje polecenia dotyczące wyjść, a program zgłasza błąd komunikacji - na ogół będzie świadczyć o zbyt małej prędkości komputera.

Montaż i uruchomienie

Przystawkę montujemy zgodnie z przyjętymi regułami, tzn. najpierw weryfikacja płytki, potem zworki, podstawki, elementy bierne i przewody - na końcu półprzewodniki. Rozmieszczenie elementów na płytce drukowanej przedstawiono na **rys. 4**.

Ponieważ zasilamy układ bezpośrednio z +5V należy uważać przy jego podłączaniu. Przed włożeniem procesora i pamięci możemy sprawdzić działanie kostki watchdoga - podanie na wejście WDI niezmiennego poziomu niskiego lub wysokiego powoduje generowanie co 1 sekundę ok. 200 ms impulsu zerującego (na wejściu RST ma to być impuls wysoki). Uwaga - WDI „pływające" nie wytwarza sygnału zerującego.

Po skompletowaniu elementów i włączeniu zasilania powinno nastąpić migotanie kontrolnej diody LED z częstotliwością 1 Hz (jeśli nie ma tego efektu, to pozostaje wykorzystać swój arsenał uruchomieniowy dla systemów mikroprocesorowych: sprawdzić oscylator, obecność sygnałów na liniach adresów i danych, zapuścić proste testy z symulatora EPROM itd.). Przystawkę łączymy z komputerem kablem. W naszym przypadku może to być kabel skrajnie uproszczony - 3 przewody ze skrzyżowanymi RxD i TxD (*Null Modem*). Najpierw zaznaczmy checkbox E (tylko wyjścia OC) i wyślijmy komendę TRX. Jeśli linia statusu zgłosi „Polecenie wykonane" to w porządku - możemy posprawdzać działanie wyjść OC (nawet zwykłym omomierzem - ale uwaga na biegunowość). Jeśli natomiast po krótkim komunikacie oczekiwania otrzymamy „Brak odpowiedzi urządzenia", to znaczy, że jest problem z portem RS 232 i musimy wszystko od początku posprawdzać, począwszy od numeru używanego przez program portu szeregowego a skończywszy na kostce MAX232 i obecności sygnałów na pinach mikrokontrolera.

Do dalszego uruchamiania należy przygotować (najlepiej wcześniej sprawdzony) układ z prostym elementem I²C. Proponuję PCF 8574 z LED-ami włączonymi przez rezystory (ok. 470Ω). Edytujemy dwubajtową komendę zapalenia diody, sprawdzamy podciągnięcie linii magistrali, gasimy checkbox E i wysyłamy przyciskiem TRX. Status „Polecenie wykonane" i zapalona dioda świadczą o sukcesie. (Uwaga zwłaszcza na „pull up" - jego brak potrafi urządzenie skutecznie „ogłupić"). Jeśli teraz dodamy trzeci bajt z adresem do odczytu i ustawimy NEXT START po drugim bajcie oraz zadamy odbiór jednego bajtu, to ponowienie komendy powinno spowodować odczyt rejestru 8574 w polu bajtów odebranych (wartość taka sama jak wysyłana) z komentarzem linii statusu: "Polecenie wykonane. Odebrano dane z magistrali I²C." Na zakończenie zmieńmy adres elementu na niewłaściwy - status powinien zgłosić „Polecenie wykonane. Wykryto błąd magistrali I²C."

Jeśli wszystkie etapy uruchamiania zakończą się powodzeniem, możemy poskładać obudowę, przykręcić gniazda, przykleić etykietę i nóżki. Wzbogaciliśmy się o następny przyrząd warsztatowy.

Jerzy Szczesiul,
jerzbial@polbox.com

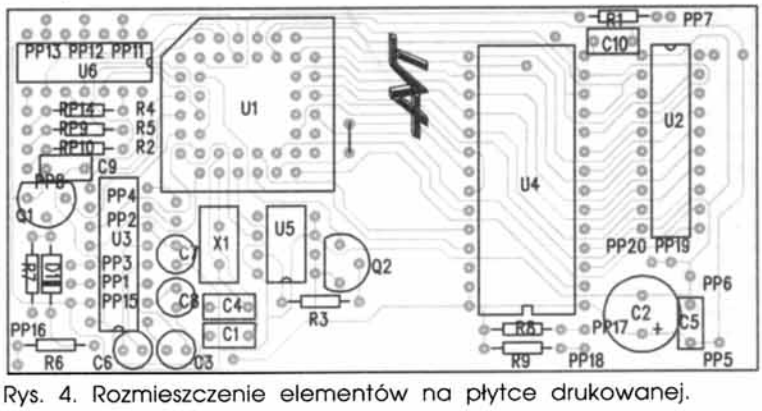
WYKAZ ELEMENTÓW

Rezystory
(1/8W)
R1, R2, R4, R5: 330Ω
R3, R6: 10kΩ
R7: 100kΩ
R8, R9: 3,3kΩ

Kondensatory
C1, C4: 33pF ceramiczne
C2: 100µF/16V elektrolityczny
C3, C6, C7, C8: 10µF/16V
C5: 100nF/63V

Półprzewodniki
U1: 80C652 w obudowie PLCC44
U2: 74HC573
U3: MAX232
U4: 27C64 (zaprogramowana)
U5: ADM699
U6: 74LS07
D1: 1N4148
Q1: BC337
Q2: BC327

Różne
obudowa plastykowa Z-VII-A,
kwarc 11,059MHz
LED ϕ3mm
podstawka PLCC44,
podstawka DIL28-600 precyzyjna,
gniazdo DB9M
gniazdo DB9F
przełącznik miniaturowy - 2 pary styków.



Rys. 4. Rozmieszczenie elementów na płytce drukowanej.