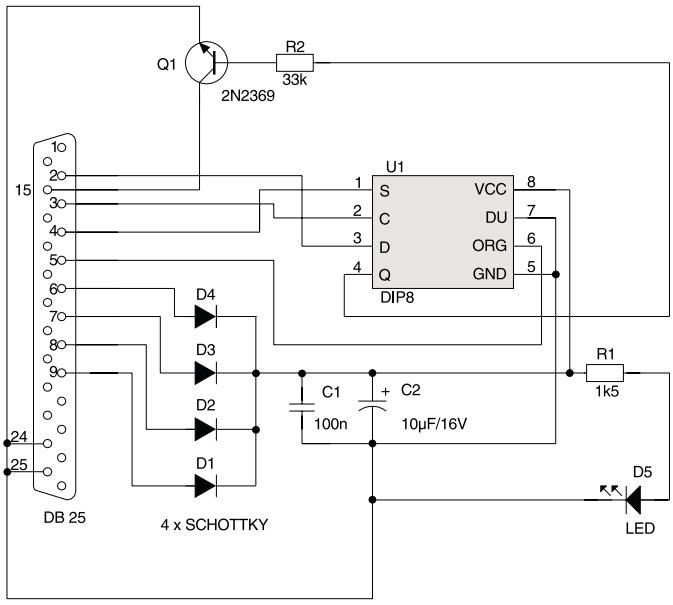
Programator szeregowych pamięci EEPROM 93C46

W artykule prezentujemy konstrukcję niezwykle prostego programatora szeregowych pamięci EEPROM z interfejsem MicroWire. Postać źródłową programu sterującego autor udostępnił na naszej stronie internetowej.

W ubiegłym roku były szeroko opisywane szeregowe pamięci EEPROM. Został też przedstawiony programator układów opartych na magistrali I²C - najbardziej u nas rozpowszechnionych. Prezentowane obecnie urządzenie uzupełnia tę ofertę o możliwość obsługi kostek 93C46 - wykorzystujących do przesyłu danych protokół *MicroWire*. Opisu protokołu nie będę powtarzał - chętni znajdą go w EP7-8/98. Niezbędne sekwencje sygnałów są generowane przez oprogramowanie PC sterujące portem Centronics. Część elektroniczna



Rys. 1.

urządzenia - o schemacie przedstawionym na **rys. 1** - sprowadza się do niewielkiego interfejsu dopasowującego kostkę pamięci do portu.

Nasuwa się pytanie, dlaczego obsługiwany jest tylko jeden typ pamięci? Niniejsze urządzenie powstało mianowicie dla potrzeb konkretnej naprawy serwisowej - właśnie wymiany 93C46. Przy okazji miała to być próba samodzielnego oprogramowania w środowisku Windows 95/98 w celu stwierdzenia, jak system wielowątkowy radzi sobie z generacją przebiegów czasowych na porcie LPT.

Dlatego też nie skorzystałem z gotowych urządzeń ani z dostępnych w internecie DOS-owych programów (np. http://www.hw.cz/constrc/eeprom/ee_prog.html).

Chętni mogą otrzymać kod źródłowy programu (Delphi 3) w celu prowadzenia własnych eksperymentów, np.:
- rozszerzenie pojemności obsługiwanych pamięci,
- udostępnienie w Windows NT poprzez użycie sterownika dostępu do portów (są takie freeware, np. na Delphi Super Page),
- dopisanie innych protokołów (np. dla kostek z interfejsem I^2C).

Działanie programatora

Diody Schottky'ego D1..D4 oraz filtrujące pojemności C1, C2 pozwalają na zasilanie kostki bezpośrednio z portu LPT. Koniecznym warunkiem jest współpraca z portem wyposażonym w bufory nowszej generacji (z napięciem +5 V w stanie wysokim oraz o znacznej obciążalności prądowej).

Zakładam, że komputery pracujące z Windows 95/98 (dla takiego środowiska jest przeznaczony program sterujący) są nowszej generacji i powinny ten wymóg spełniać. Prototyp był sprawdzany z kartą ISA typu Tc-210 oraz z wbudowanym w płytę portem HP Vectra. Na kondensatorze C2 uzyskano ok. 4,8 V, co jest wartością w pełni wystarczającą do zasilania interfejsu.

W razie potrzeby można zasilić interfejs całkiem oddzielnie (w modelu prezentowanego urządzenia zastosowano dodatkowy stabilizator 78L05 zasilany z gniazda jack - na schemacie nie zostało to zaznaczone). Dioda LED jest zasilana przez rezystor R1 (o dużej - ze względu na oszczędność prądu - wartości rezystancji) i wskazuje włączenie zasilania. Jest to bardzo użyteczne przy podłączaniu programatora i konfigurowaniu portu.

Jako linie wyjściowe MicroWire wykorzystałem linie danych portu. Nie jest to przypadkowe - w porcie używanym do uruchomienia wydajność prądowa linii danych była znacznie wyższa niż linii sterujących (np. Auto Feed itd.), co pozwoliło na uzyskanie wyższych częstotliwości powtarzania dla sygnału wyjściowego. Podczas testów na końcu kabla przedłużającego (ok. 2 m) udało się uzyskać ok. 250 kHz przy ładnym przebiegu prostokątnym (w programie źródłowym możemy zauważyć, że sekwencje ustawiania stanu linii są - z tego właśnie względu - celowo spowolnione).

Jako wejście danych służy jedna z linii statusu (Error - pin15). Sygnał na nią jest podany za pośrednictwem tranzystora impulsowego (Q1 - 2N2369). Tak się dzieje w celu eliminacji wstecznego przepływu prądu z linii przez kostkę wkładaną w podstawkę przy wyłączonym zasilaniu. Z powyższego wynika, że wystarcza standardowy tryb pracy portu - taki też należy ustawić.

Montaż - ze względu na prostotę urządzenia - jest dowolny. Do testów wykorzystywałem prowizorycznego „pająka" złożonego bezpośrednio na wtyku DB-25M. Prezentowany model jest dużo staranniej wykonany, ale też opiera się na przestrzennym (i „klejowym") montażu w obudowach z tworzywa sztucznego. Diody i tranzystor w obudowie wtyku, a kondensatory i dodatkowy stabilizator z gniazdem jack oraz podstawka - w małym pudełku z tworzywa.

Jako podstawkę pamięci zastosowałem zwykłą podstawkę DIL8 przyklejoną na zewnątrz obudowy. Jest to bardzo tanie rozwiązanie, a w razie zużycia można ją łatwo wymienić. Ponadto, przy zaledwie 8 nóżkach wkładanie kostek nie jest zbyt uciążliwe.

Podłączenie i uruchomienie

Najpierw należy wybrać port, do którego podłączymy programator. Najlepiej, żeby mógł on zapewnić (jak opisano powyżej) zasilanie programatora. Program sterujący może pracować z adresem portu równoległego 378h lub 278h. Odpowiada to zazwyczaj portom LPT1 i LPT2 (LPT1 jest dostępny praktycznie zawsze).

WYKAZ ELEMENTÓW

Rezystory
R1: 1,5kΩ
R2: 33 kΩ
Kondensatory
C1: 100nF
C2: 10µF/16V
Półprzewodniki
Q1: tranzystor impulsowy npn (2N2369 lub podobny)
D1..D4: diody Schottky'ego (BAT 85)
D5: LED żółty 3 mm
Różne
wtyk DB 25 M + obudowa
podstawka DIL 8
akcesoria (obudowa, kabel 6-żyłowy w ekranie - GND przez ekran, przepusty kablowe)

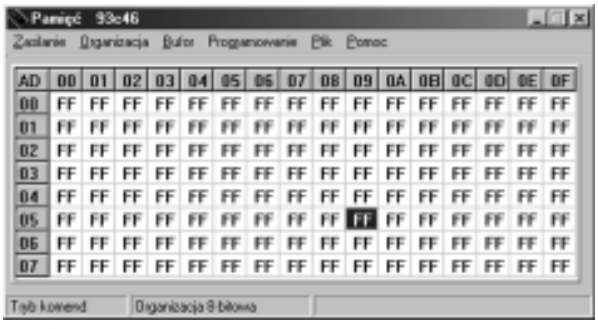
Oprogramowanie sterujące jest dostępne w Internecie pod adresem: www.ep.com.pl/ftp/tools.htm.

Warto jednakże zauważyć, że dysponowanie tylko jednym portem LPT (który przeważnie jest zajęty drukarką i to zazwyczaj w trybie ECP lub EPP) - w nowszych komputerach na ogół wmontowanym w płytę - stawia pod dużym znakiem zapytania sensowność jego używania dla celów warsztatu elektronicznego.

Po pierwsze - przy wszelkich eksperymentach i przełączaniach zawsze istnieje ryzyko uszkodzenia portu.

Po drugie - prostsze aplikacje zazwyczaj posługują się trybem standard lub BiDirectional i należy wciąż pamiętać o BIOS-owym ustawieniu trybu.

Po trzecie - samo przełączanie mechaniczne urządzeń jest mocno uciążliwe o ile nie stosujemy dodatkowych specjalizowanych przełączników. Jeśli więc chcemy wykorzystywać LPT dla własnych, nietypowych celów, bardzo wskazane będzie wyposażenie PC w dodatkową kartę I/O. Może to być całkiem stary typ z giełdy lub komisu (ale za-



Rys. 2.

zwyczaj jego możliwości nie będą wielkie) albo nowa karta specjalizowana, wybrana według potrzeb.

Przykładowe rozszerzenie: karta ISA typu Tc-210 (2xCOM + 1xLPT/BiDir) z kablem przedłużającym ok. 2 m znakomicie ułatwia wszelkie prace uruchomieniowe, a przy okazji mamy do dyspozycji dodatkowe porty szeregowe.

Program obsługujący programator wymaga systemu Windows 95/98 (obecna wersja nie pracuje pod NT). Instalujemy go z dyskietki poprzez uruchomienie programu *setup.exe*, albo z panelu sterowania: dodaj-usuń programy. Następnie sprawdzamy w zasobach systemu pod jakimi adresami mamy ulokowane porty LPT.

Program domyślnie startuje z adresem 278h (LPT2). Aby użyć 378h (LPT1) należy podać parametr startowy 1. Wykonujemy to w następujący sposób:

- w folderze Windows\Start Menu\Programs\93c46 odnajdujemy skrót *Programator.lnk*,
- otwieramy z menu skrótów Właściwości i w linii Target (program docelowy) dopisujemy za ścieżką dostępu (za cudzysłowem) spację i parametr, np: *<„C:\Program Files\Avt\Programator 93c46\M_93.exe“ 1>*.

Sprawdzamy jeszcze przed uruchomieniem tryb pracy portu (BIOS dla portu na płycie lub zworki dla karty ISA) - powinien być ustawiony Standard (Output) lub BiDirectional (Byte).

Teraz możemy uruchomić program z paska Menu Start. Dioda D5 powinna się zaświecić po włączeniu zasilania *(Zasilanie/Wyłącznik)*, co wskaże jednocześnie na prawidłowe skonfigurowanie portu. Skontrolujmy jeszcze napięcie pomiędzy pinami 4 i 8 podstawki - powinno wynosić 4,7..4,8V. Teraz można przystąpić do programowania.

Obsługa programu

Okienko programu uruchomionego pokazano na **rys. 2**. Wszystkie funkcje programatora są dostępne z poziomu menu głównego:

- *Zasilanie/Wyłącznik* - włącza i wyłącza zasilanie pamięci,
- *Organizacja* - przełącza podział pamięci na komórki 8- lub 16-bitowe. Tu można dodać, że pamięci 93C46 w wersji SMD używają tylko trybu 16-bitowego,
- *Bufor/Zerowanie* - ustawia wszystkie komórki ekranowego bufora na ffh (lub ffffh), w zależności od organizacji,
- *Bufor/Wypełnianie* - wypełnia wszystkie komórki wartością komórki aktualnie wybranej,
- *Bufor/Edycja* - przejście do edycji wartości wybranej komórki (komenda osiągalna również poprzez *'e'* z klawiatury),
- Polecenie menu *'Programowanie'* odpowiada operacjom na kostce pamięci (zapis komórki, zapis całości, kasowanie, odczyt z wpisem do bufora, porównanie z buforem). Opcja '*Samoczynne kasowanie*' określa, czy kostka wymaga oddzielnego kasowania komórki przed zapisem. Jest ona domyślnie włączona, co odpowiada nowszym typom kostek pozwalającym na pominięcie cyklu kasowania,
- *Plik* - pozwala na zapis i odczyt bufora z/do pliku *.c46,
- *Pomoc* - uruchamia typowe windowsowe okienko pomocy.

Jerzy Szczesiul, AVT
jerzy.szczesiul@ep.com.pl

Program sterujący pracą programatora oraz jego postać źródłowa są dostępne na stronie internetowej EP, pod adresem:

- *www.ep.com.pl/ftp/tools.html*
- *www.ep.com.pl/ftp/other.html*