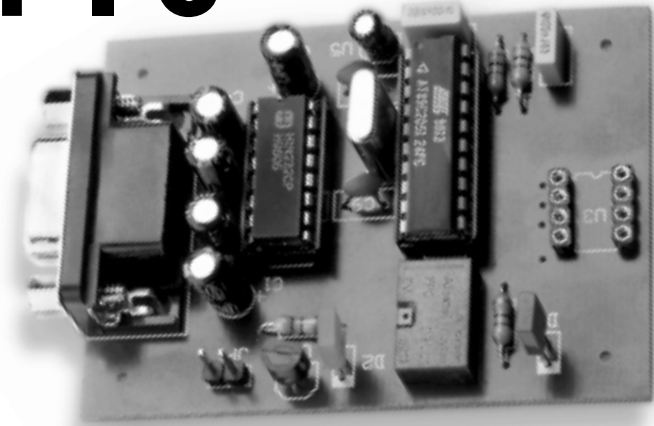


# Programator pamięci EEPROM z interfejsem szeregowym I<sup>2</sup>C

## kit AVT-466



*Czytelnicy, którzy przeczytali tekst o sposobach programowania pamięci EEPROM, zamieszczony w sierpniowym numerze EP, mogą traktować ten projekt jako uzupełnienie tamtego artykułu.*

*Zamieszczone w nim informacje o sposobie transmisji danych i pracy układów EEPROM z szeregowym interfejsem I<sup>2</sup>C wykorzystano podczas projektowania programatora i przygotowywania jego oprogramowania.*

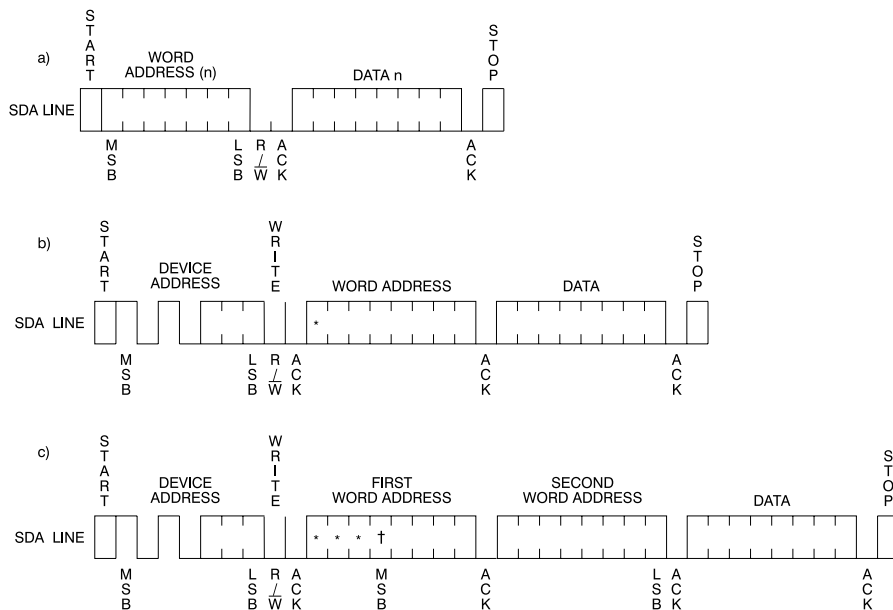
Pamięci szeregowo EEPROM są zazwyczaj wykorzystywane jako pamięci nieulotne, przechowujące niewielkie ilości danych, np. parametry konfiguracyjne w wielu urządzeniach powszechnego użytku takich jak odbiorniki telewizyjne, magnetowidy czy zestawy muzyczne. EEPROM-y szeregowo można także spotkać w przemysłowych sterownikach, kserokopiarkach, termostatach itp. Są często i chętnie stosowane ze względu na niewielki koszt i dość prosty sposób zapisu i odczytu informacji poprzez powszechnie stosowany interfejs I<sup>2</sup>C.

Budując układ wykorzystujący EEPROM-y wygodnie jest mieć pod ręką urządzenie, dzięki któremu można podejrzeć zawartość pamięci lub ją zmodyfikować. Dotyczy to zwłaszcza sytuacji, gdy układ nie działa prawidłowo i chciałoby się wiedzieć czy dane wpisywane do pamięci są właściwe. Programator pozwoli także sprawdzić, czy używana kostka pamięci nie jest uszkodzona oraz powielić jej zawartość w innych egzemplarzach.

Układy pamięci EEPROM z interfejsem I<sup>2</sup>C produkowane w obudowach DIP8 mają, bez względu na typ, standardowy rozkład wyprowadzeń. Dotyczy to zarówno produktów firmy XICOR, ATMEL, jak i SGS-THOMSON. Dodatkowo napięcie zasilające (najczęściej

+5V, chociaż produkowane są wersje zasilane +3V, a nawet +1,8V!) dołącza się do 8 wyprowadzenia, a masę do 4. Linia SDA to nóżka 5, a SCL nóżka 6. Wyprowadzenia 1-3 służą do adresowania i wyboru kostki pamięci lub pozostają nie podłączone. Nóżka 7 podłączona do plusa może blokować zapis lub producent pozostawia ją nie wykorzystaną. Generalnie pamięci różnią się pojemnością i związanych z tym sposobem transmisji danych magistralą I<sup>2</sup>C. Ze względu na sposób programowania układy można podzielić na trzy grupy: o pojemności 1kb (128B) oznaczane jako 24C01, o pojemności 1/2/4/8/16kb (odpowiednio 128, 256, 512, 1024 i 2048B) oznaczane symbolami 24C01A, 24C02-C16 oraz układy o pojemności 32 i 64 kb oznaczane, jak łatwo się domyślić, symbolami 24C32-C64. Programator potrafi zapisywać i odczytywać dane z pamięci wszystkich wymienionych typów.

Różnice w sposobie transmisji danych pomiędzy układami tych trzech grup pokazuje **rys. 1**, przedstawiający schematycznie sposób zapisu do pamięci 1 bajtu danych. Różnice wynikają głównie ze sposobu adresowania, co z kolei wiąże się z pojemnością pamięci. Ponieważ układ 24C01 może zapamiętać tylko 128 bajtów i do magistrali I<sup>2</sup>C może być dołączony



Rys. 1. Formaty słów zapisu danych do pamięci.

tylko jeden taki układ, to jego adresowanie mieści się w jednym bajcie poprzedzającym bajt danych. Najstarszych 7 bitów określa adres komórki, a najmłodszy bit decyduje o tym czy informacja ma być zapisywana do komórki pamięci (bit = 0) czy też odczytywana (bit = 1). Pojemność układów z drugiej grupy może sięgać 2k i do zaadresowania wszystkich komórek są potrzebne dwa bajty, a dokładniej 11 bitów. Młodsze bity adresu zawiera bajt nazwany WORD ADDRESS, a 3 najstarsze bity zostały przeniesione do bajtu o nazwie DEVICE ADDRESS (razem z bitem R/W, określającym tryb pracy - zapis/odczyt).

Duża pojemność EEPROM-ów z trzeciej grupy wymaga dodania jeszcze jednego bajtu adresowego nazwanego FIRST WORD ADDRESS. W tym bajcie 5 najmłodszych bitów pełni rolę najstarszych bitów adresujących komórkę pamięci, a pozostałe bity adresu zawiera bajt SECOND WORD ADDRESS. W bajcie DEVICE ADDRESS pozostawiono bit sterujący trybem pracy i bity adresu układu.

Układ programatora składa się z małego sterownika współpracującego z komputerem PC poprzez łącze szeregowo RS-232. Program w komputerze i operator decydują o sposobie działania programatora, a procesor sterownika jest odpowiedzialny za bezpośrednią obsługę programowanej pamięci EEPROM-

ROM. Schemat sterownika pokazano na rys. 2. Kody poleceń i danych, transmitowane łączem RS, docierają do gniazda P1 i przez układ U4 są przekształcane na sygnały o poziomie TTL i podawane do procesora U1. Program procesora interpretuje polecenia z PC i sterując przekaźnikiem PK1 dołącza zasilanie do programowanego układu EEPROM (osadzonego w podstawce U3), co jest sygnalizowane świeceniem diody D1. Magistralą I<sup>2</sup>C dołączoną do portów P1.7 i P1.6 procesora są wysyłane adresy i dane. W przypadku odczytu, dane przekształcane są na sygnały RS-a i tą drogą wysyłane do komputera. Po zakończeniu sesji odczytu lub zapisu procesor wyłącza przekaźnik i programowany EEPROM może być wyjęty z podstawki, a sterownik czeka na kolejne rozkazy.

Sterownik reaguje na 4 rozkazy przesyłane łączem RS. Każdy rozkaz składa się z kodu ASCII dużej litery oraz bajtów parametrów, danych i cyfry kontrolnej, pozwalającej stwierdzić czy transmisja przebiega bez zakłóceń. Poniżej podano zestawienie rozkazów i ich strukturę (litery w nawiasach oznaczają poszczególne parametry).

**P(abc)** - ustawienie parametrów pracy sterownika. Bajt 'a' określa szybkość transmisji (0 - 1200 bodów, 1 - 4800 bodów), bajt 'b' wybiera typ programowanego

EEPROM-u (0 - 24C01, 1 - 24C01A, 24C02/C16, 2 - 24C32/C64), bajt 'c' jest bajtem kontrolnym.

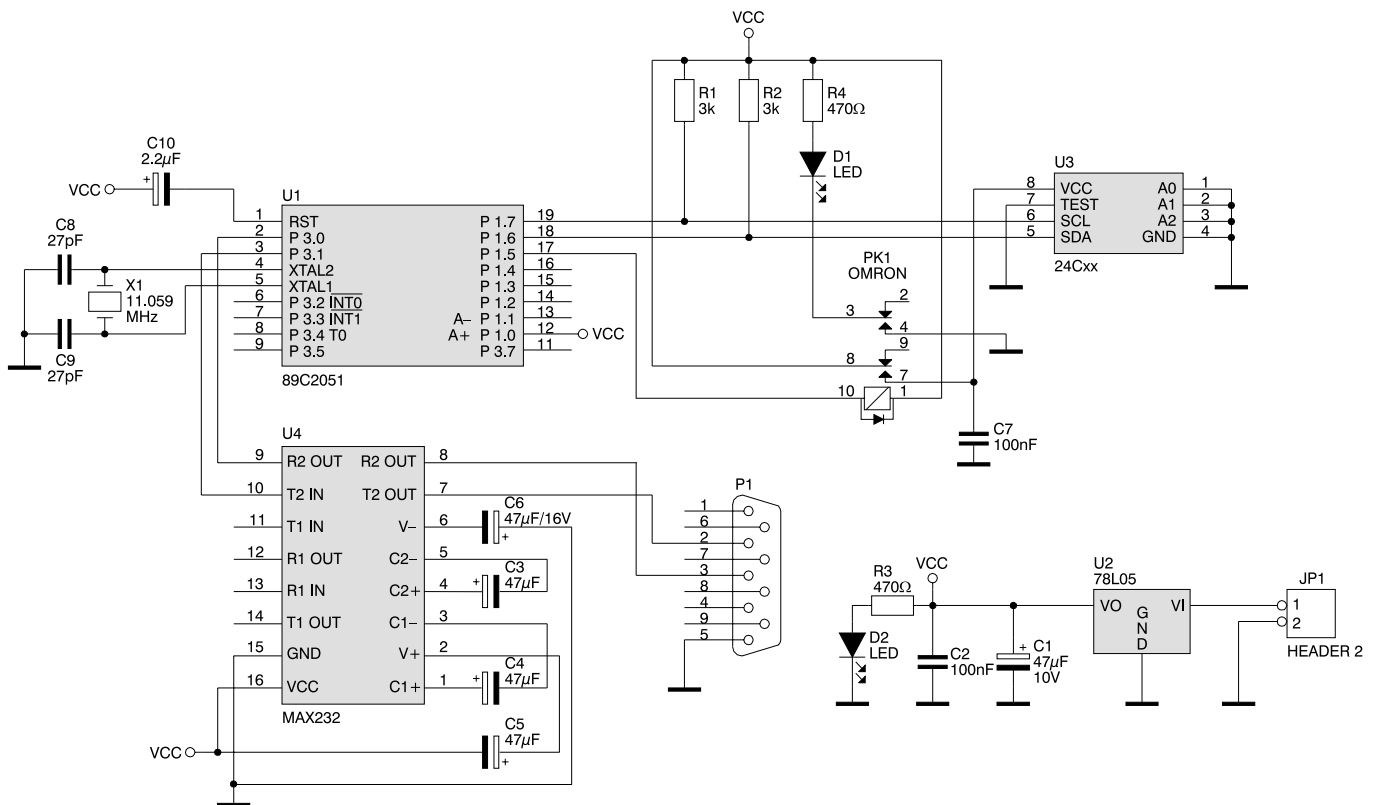
**S(abc)** - ustawienie adresu EEPROM-u do zapisu lub odczytu. Bajt 'a', to starsza część adresu, bajt 'b' młodsza część, a 'c' to bajt kontrolny.

**W(add...dc)** - zapis danych do EEPROM-u. Bajt 'a' jest liczbą przesyłanych bajtów do zapisu plus bajt kontrolny. Litery 'd' oznaczają bajty danych, a litera 'c' cyfrę kontrolną.

**R(ac)** - odczyt danych z EEPROM-u. Litera 'a' oznacza liczbę bajtów do odczytu, począwszy od adresu ustalonego wcześniej rozkazem „S”, a 'c' to cyfra kontrolna.

Układ sterownika, po odebraniu rozkazu i jego pomyślnym wykonaniu, wysyła do komputera jako potwierdzenie znak „A” (skrót angielskiego słowa „acknowledgement”). Jeżeli rozkaz z jakichś przyczyn nie mógł być wykonany, wysyłany jest znak „N” (brak potwierdzenia). Wyjątek stanowi odpowiedź po rozkazie odczytu danych z EEPROM-u. Sterownik wysyła wtedy znak „A”, bajty 'ddd' w liczbie podanej w rozkazie odczytu, a na końcu bajt sumy kontrolnej. Jeżeli odczyt się nie powiódł, sterownik odpowiada jedynie znakiem „N”.

Wyjaśnienia może wymagać wprowadzenie i sposób tworzenia bajtu kontrolnego. Ponieważ sterownik i komputer porozumiewają się ze sobą przy pomocy szeregowego łącza RS, a kabel połączeniowy może mieć długość nawet kilku metrów, w czasie transmisji mogą zdarzyć się zakłócenia, które zniekształcą przesyłane dane lub rozkazy. Może to doprowadzić np. do zafałszowania zapisywanych w EEPROM-ie danych lub nawet zawieszenia pracy programatora. W celu lepszej kontroli poprawności transmisji, urządzenie nadawcze oblicza sumę kontrolną wszystkich transmitowanych bajtów i dołącza ją na końcu wysyланego rozkazu. Odbiornik z drugiej strony łączy RS przeprowadza podobną operację. Oblicza sumę kontrolną odbieranych bajtów (ale oczywiście bez bajtu kontrolnego), a następnie porównuje z bajtem sumy obliczonej



Rys. 2. Schemat elektryczny układu.

przez urządzenie nadające. Jeżeli sumy są identyczne, można mieć dużą dozę pewności, że do przesyłanych danych nie zakradły się żadne błędy.

Do kontroli transmisji przy pomocy sumy kontrolnej można wykorzystać różne sposoby jej obliczania. W programatorze został wykorzystany algorytm jaki firma DALLAS stosuje podczas transmisji danych między układami *iButton* magistralą 1-przewodową. Algorytm ten pozwala wychwycić nawet wielokrotne, złośliwe błędy transmisji, które są niewykrywalne metodą zwykłego sumowania. Stwierdzenie błędów transmisji może spowodować jej powtórzenie lub powiadomienie o tym fakcie operatora.

Jak wcześniej wspomniano, aby pamięć EEPROM mogła być odczytana lub zapisana, sterownik programatora musi być nadzorowany przez program w komputerze PC.

Program ten pełni rolę interfejsu między operatorem i sterownikiem. Wykorzystuje cztery rozkazy, na które reaguje sterownik. Program można oczywiście stworzyć samemu korzystając z podanego wcześniej opisu struktury rozkazów. Na potrzeby prototypu

została napisana w języku C (dla kompilatora BORLAND) jego wersja robocza. Program w trybie znakowym działa w DOS-ie. Brak mu efektywnej szaty graficznej i nie jest zbyt „inteligentny“, czyli nie potrafi przewidzieć i zareagować na niektóre pomyłki operatora, ale z jego pomocą można w pełnym zakresie zaprogramować i odczytać wszystkie trzy typy EEPROM-ów z interfejsem I<sup>2</sup>C, a także ustawić parametry pracy sterownika.

Naciskanie klawisza [c] klawiatury spowoduje, że do transmisji będzie wykorzystywany COM 1 lub 2 w komputerze. Wybór zależy od tego, które wyjście RS jest w danej chwili do wykorzystania.

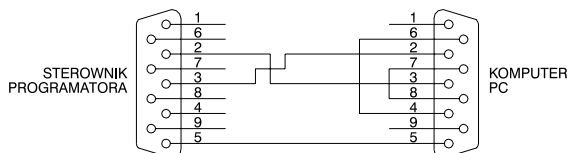
Klawisz [b] zmienia szybkość transmisji między komputerem a sterownikiem. Klawisz [e] wybiera typ EEPROM-u jaki ma być programowany. Wybór każdej z opisanych do tej pory funkcji spowoduje jednoczesną zmianę w linii statusu na ekranie.

Naciśnięcie klawisza [r] uruchamia procedurę odczytu danych z EEPROM-u. Na dole ekranu wyświetlona zostanie linia z pytaniem o adres pierwszej i ostatniej komórki do odczytu. Adresy należy podać w kodzie heksadecymalnym, np. dla adresu dziesięt-

nego '26' należy wypisać liczbę '1A' i nacisnąć klawisz ENTER. Jeżeli odczyt przebiegł bezbłędnie, program zażąda podania ścieżki i nazwy pliku, w którym odczytane dane zostaną zapisane. Zapisywane dane można obejrzeć przy pomocy jakiegoś edytora narzędziowego, ponieważ sam program jest pozbawiony możliwości edycyjnych.

Klawisz [w] inicjuje procedurę zapisu danych do EEPROM-u. Sposób postępowania jest podobny jak podczas odczytu. Najpierw program pyta o ścieżkę dostępu do pliku (binarnego) zawierającego dane do zapisu. W przypadku odnalezienia takiego pliku należy określić adresy obszaru pamięci (podając adresy pierwszej i ostatniej komórki pamięci), do którego dane z pliku będą zapisywane. Po zapisie zakończonym sukcesem pojawi się odpowiedni komunikat. Klawisz [q] służy do opuszczenia programu.

Ze względu na swoją prostotę program nie został zabezpieczony przed wszystkimi pomyłkami operatora. Dotyczy to w szczególności wyboru COM-u, szybkości transmisji i określenia zakresu programowania pamięci. Jeżeli użytkownik ustawi szybkość transmisji na



Rys. 3. Sposób połączenia programatora z komputerem PC.

4800 baudów, a następnie sterownik zostanie wyłączony i ponownie włączony, to program w komputerze PC nie będzie mógł „dogadać się” ze sterownikiem, ponieważ będą pracowały z inną szybkością transmisji. Podobnie jest z ustawieniem adresów komórek do odczytu lub zapisu. Program podpowiada wartości maksymalne, jeśli jednak do pamięci o pojemności np. 256 bajtów użytkownik będzie chciał wpisać 300 to w czasie procedury zapisu program zasygnalizuje błąd i zakończy działanie. W przypadku wyboru COM-u, to wskazanie portu, do którego podłączona jest np. myszka, spowoduje zmianę jego parametrów. Komputer przestanie widzieć myszkę i przywrócenie poprzedniego stanu wiąże się z ponownym uruchomieniem sterownika myszy lub zresetowaniem komputera.

W przypadku korzystania z dołączonego do kitu programu, użytkownik może zmienić jego wersję źródłową dostosowując ją do swoich potrzeb, a następnie skompilować do postaci pliku wykonywalnego .exe lub .com. W przypadku korzystania z innego kompilatora niż BORLAND może pojawić się konieczność drobnych zmian w składni niektórych linii. Oprócz opisywanych niedogodności, program nie powinien ni-

czego zmienić w funkcjonowaniu komputera, na którym jest uruchamiany. Jednak musimy się zastrzec, że każdy użytkownik programu czyni to na własną odpowiedzialność.

Ze względu na małe wymiary i przeznaczenie układu sterownika został on zaprojektowany z myślą o stosowaniu go bez obudowy. Podstawka dla programowanych EEPROM-ów, najlepiej precyzyjna, zapewniająca pewny kontakt nawet po wielokrotnym wyjmowaniu i wkładaniu układów, jest montowana bezpośrednio na płycie drukowanej. Podobnie jest montowana dioda LED D2 sygnalizująca włączenie napięcia zasilającego i D1, której świecenie informuje, że w tym czasie EEPROM nie powinien być wyjmowany. Standardowe 9-stykowe gniazdo szufladowe DB9 „żeńskie”, wlotowywane w płytke drukowaną służy do dołączenia kabla RS.

Na rys. 3 pokazano połączenia przewodu z wtykami od strony sterownika i komputera. Ważne, aby wtyk od strony komputera zwierzał ze sobą wyprowadzenia 4-6 i 7-8. Wynika to ze specyfiki pracy portów szeregowych w komputerze. Były one projektowane głównie z myślą o współpracy z peryferiami typu modem, które przez ustawienie odpowiednich poziomów logicznych na określonych wyprowadzeniach gniazda informowały komputer o swojej obecności i gotowości do pracy.

Po zmontowaniu układu na płycie, której widok przedstawio-

### WYKAZ ELEMENTÓW

#### Rezystory

R1, R2: 3kΩ  
R3, R4: 470Ω

#### Kondensatory

C1: 47μF/10V  
C2, C7: 100nF  
C3, C4, C5, C6: 47μF/16V  
C8, C9: 27pF  
C10: 2,2μF

#### Półprzewodniki

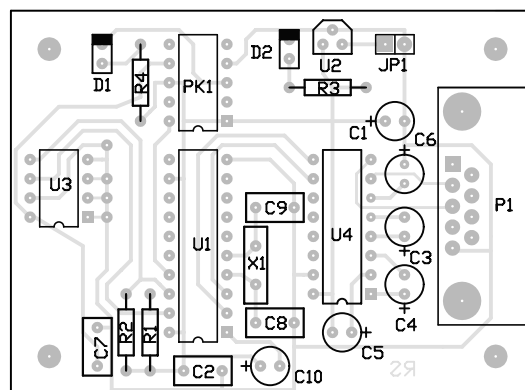
D2, D1: LED  
U1: 89C2051 (zaprogramowany)  
U2: 78L05  
U4: MAX232

#### Różne

PK1: OMRON 5V  
P1: złącze DB9 „żeńskie” do druku  
X1: 11,059MHz  
podstawka precyzyjna DIP8  
dyskietka z programem obsługi programatora

no na wkładce wewnątrz numeru, i dołączeniu zasilania, trzeba sprawdzić czy na wyjściu stabilizatora U2 napięcie ma wartość +5V. Następnie należy połączyć sterownik z komputerem przy pomocy kabla RS-a. Cały układ działa sprawnie jeżeli po zmianie parametru np. szybkości transmisji lub wyborze innego typu EEPROM-a, na ekranie komputera pojawi się komunikat o sukcesie przeprowadzonej operacji. Pobór prądu przez sterownik zależy od programowanego EEPROM-a i wielkości napięcia niestabilizowanego podanego na wejście JP1. Zasilacz powinien jednak mieć wydajność minimum 100mA przy napięciu 8..12V.

**Ryszard Szymaniak, AVT**



Rys. 4. Rozmieszczenie elementów na płycie drukowanej.