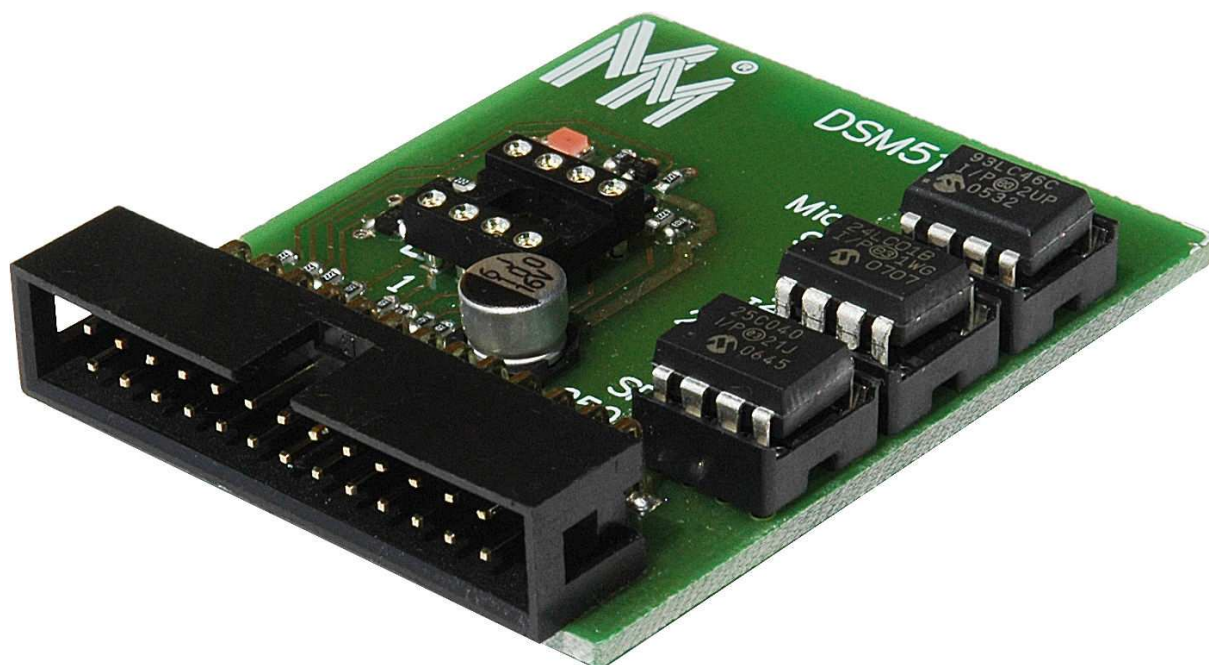


# Programator pamięci EEPROM



**Model M-11**  
do Dydaktycznego Systemu  
Mikroprocesorowego DSM-51

**Instrukcja użytkowania**

Copyright © 2007 by **MicroMade**

All rights reserved

Wszelkie prawa zastrzeżone

# **MicroMade**

**Gałka i Drożdż sp. j.**

**64-920 PIŁA, ul. Wieniawskiego 16**

**Tel./fax: (67) 213.24.14**

**E-mail: [mm@micromade.pl](mailto:mm@micromade.pl)**

**Internet: [www.micromade.pl](http://www.micromade.pl)**

Wszystkie nazwy i znaki towarowe użyte w niniejszej publikacji są własnością odpowiednich firm.

## 1. Przeznaczenie modelu

Model M-11 jest przystawką do Dydaktycznego Systemu Mikroprocesorowego DSM-51 umożliwiającą programowanie szeregowych pamięci EEPROM.

Istnieje wiele różnych pamięci EEPROM, różniących się między sobą interfejsem komunikacyjnym. Wśród nich wyróżniamy trzy najpopularniejsze:

- Microwire
- SPI
- I<sup>2</sup>C

Wszystkie te interfejsy są obsługiwane przez model M-11 - jest to więc w pełni funkcjonalny programator pamięci EEPROM. Wykorzystanie tych możliwości zależy tylko od napisania odpowiedniego oprogramowania.

### ***Uwaga!***

***Model wyposażony jest w 3 dodatkowe podstawki, przeznaczone do przechowywania załączonych pamięci EEPROM. Wyjmowanie układów z podstawek powinno być wykonywane za pomocą cienkiego śrubokrętu, aby nie pogiąć nóżek układów.***

## 2. Budowa i zasada działania

Zaprojektowanie programatora pamięci EEPROM to zupełnie inne zagadnienie niż podłączenie jednego, wybranego EEPROMu do mikrokontrolera. Projektując programator często należy pogodzić sprzeczne wymagania. Na przykład, na tym samym pinie jeden EEPROM wymaga dołączenia masy (minus zasilania), podczas, gdy drugi ma w tym samym miejscu transmisję dwukierunkową. Dlatego model M11 nie jest tylko podłączeniem wejść/wyjść cyfrowych do podstawki.

Model M-11 jest sterowany przez system DSM-51 za pośrednictwem złącza wejść/wyjść cyfrowych. Wyjścia te są dwukierunkowe, ale przestawiane mogą być tylko całymi portami. Programator wymaga natomiast indywidualnej zmiany kierunku poszczególnych pinów. Dlatego wybrano następujące rozwiązanie.

Jako port wyjściowy użyto port C, ponieważ pozwala on na indywidualne sterowanie poszczególnymi pinami. Do sterowania pinów, które dla pewnych pamięci są również wyjściami, zastosowano tzw. otwarte kolektory. Ustawienie na danym pinie stanu 1 (podawanego faktycznie przez opornik podwieszający) pozwala na jego sterowanie przez programowany EEPROM.

W celu odczytu tych pinów dołączono je dodatkowo do portu A, ustawionego jako wejściowy.

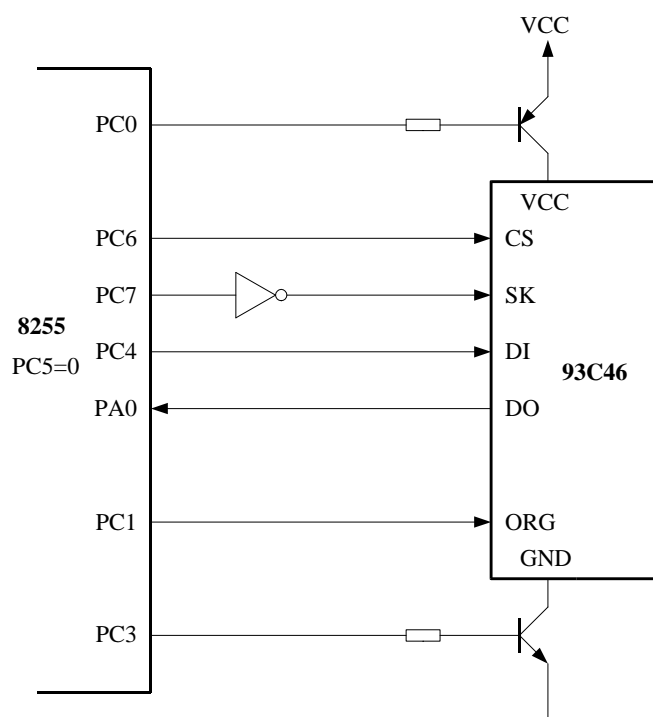
Konsekwencją sterowania pinu poprzez jeden tranzystor w układzie otwartego kolektora jest zanegowanie stanu z wyjścia portu. Trzeba o tym pamiętać pisząc oprogramowanie.

## M-11 Programator pamięci EEPROM

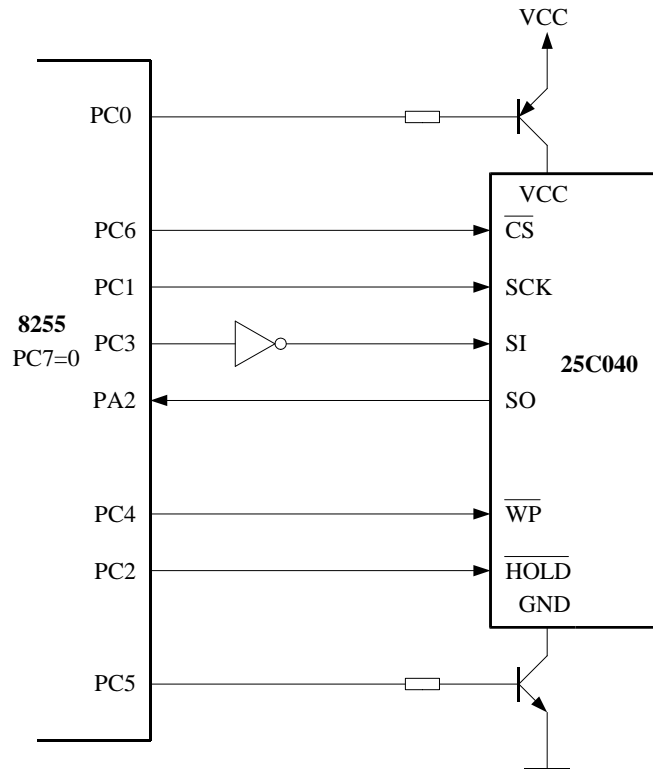
Dołączenie masy (ujemnego bieguna zasilania układu) jest wykonane w ten sam sposób, ponieważ dla różnych pamięci EEPROM masa musi być podłączona do różnych pinów.

Jak z powyższego opisu wynika, poszczególne piny pełnią różne funkcje, w zależności od typu programowanego układu. W celu ułatwienia pisania oprogramowania, można programator przedstawić w postaci innego schematu blokowego, dla każdego z typów pamięci EEPROM.

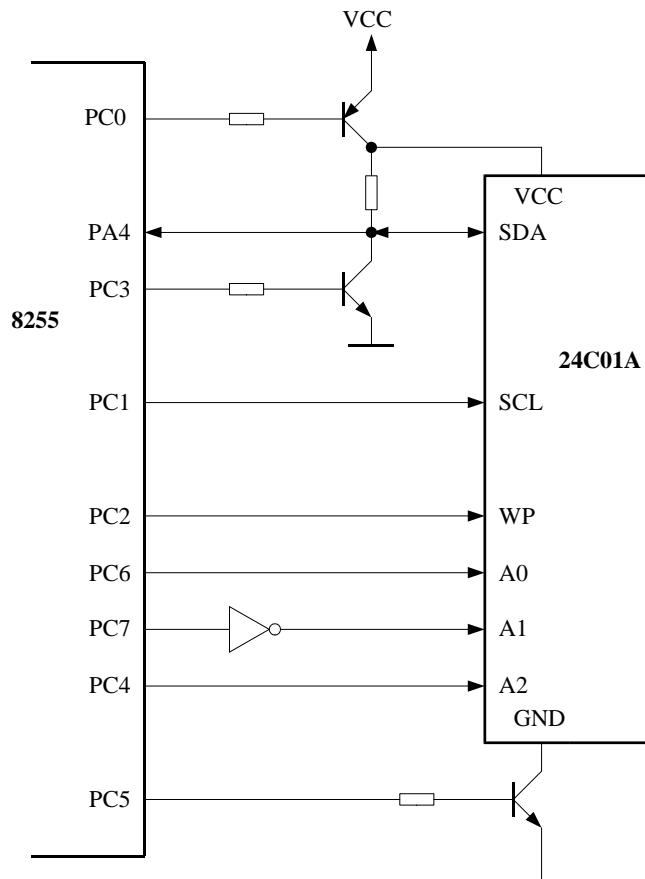
### Programator pamięci microwire



Programator pamięci SPI



Programator pamięci I<sup>2</sup>C



### 3. Oprogramowanie

Wraz z modelem M-11 dostarczane są trzy pamięci EEPROM:

- 93C46 - Microwire, 128 x 8 bitów (64 x 16 bitów)
- 25C040 - SPI, 512 x 8 bitów
- 24C01A - I<sup>2</sup>C, 128 x 8 bitów

Dostępne są również trzy przykładowe programy, które potrafią odczytać i zapisać dane do określonego EEPROMu.

Każdy z tych programów potrafi odczytać 8 pierwszych komórek pamięci EEPROM. Po naciśnięciu klawisza Enter pozwala wybrać adres jednej z nich i wpisać do niej nową wartość. Programy te nie są oczywiście pełnowartościowymi programatorami pamięci EEPROM, ale wykorzystując zawarte w nich procedury odczytu i zapisu, można taki program napisać.

### 4. Interfejsy szeregowe

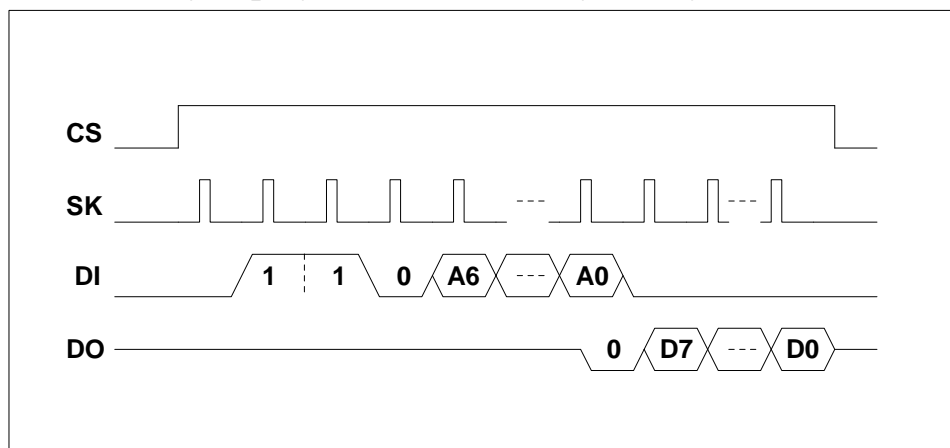
#### Microwire.

Interfejs Microwire jest opisany w podręczniku do DSM51. Interfejs stanowią 4 linie.

- CS - wybór układu
- SK - zegar transmisji
- DI - wejście danych
- DO - wyjście danych

EEPROMy Microwire posiadają jedną cechę nie występującą w innych pamięciach. Istnieją pamięci Microwire o organizacji 8 lub 16 bitowej, a niektóre mogą pracować w obu tych trybach. W takim wypadku, do zmiany organizacji pamięci służy linia ORG. Dostarczany z przystawką EEPROM AT93C46 może pracować w obu trybach. Stan 0 na linii ORG powoduje wybór trybu 8-bitowego, a stan 1 trybu 16-bitowego.

Poniżej przedstawiona jest przykładowa sekwencja odczytu:



## SPI

Interfejs SPI jest bardzo zbliżony do Microwire. Tu również występują cztery podstawowe linie interfejsu, w zasadzie o identycznych funkcjach (choć oznaczane innymi skrótami).

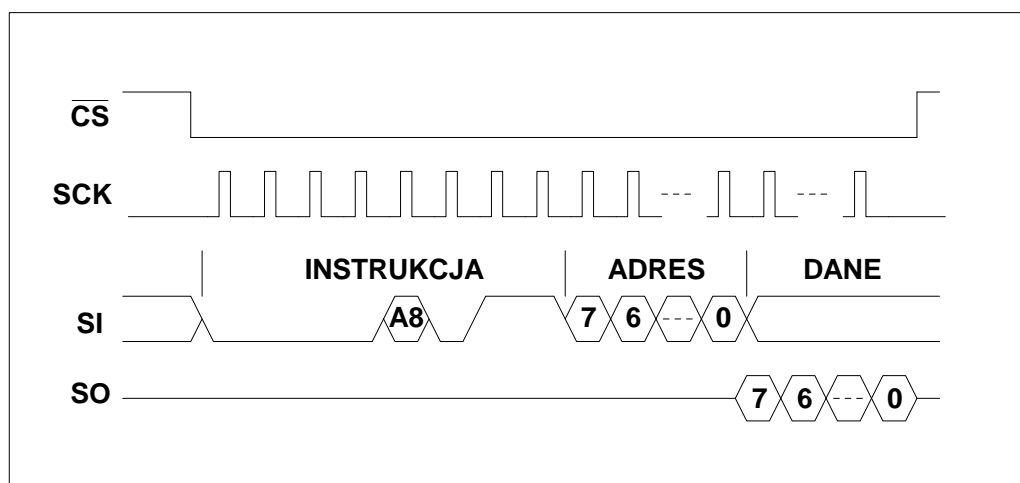
- CS - wybór układu (polaryzacja odwrotna niż w Microwire)
- SCK - zegar transmisji
- SI - wejście danych
- SO - wyjście danych

W porównaniu do interfejsu Microwire ujednolicono tu długość wszystkich rozkazów i danych - zawsze przesyłana jest wielokrotność 8 bitów. W Microwire długość rozkazu zmienia się wraz z rozmiarem pamięci EEPROM.

Stan 0 na linii WP powoduje zabezpieczenie zawartości EEPROMu 25C040 przed modyfikacją. Aby rozkazy zapisu do pamięci działały linia ta musi być w stanie 1.

Linia HOLD pozwala na chwilowe zawieszenie komunikacji z EEPROMem w celu wykonania na szynie SPI innej operacji. W czasie komunikacji z EEPROMem linia ta musi być ustawiona w stan 1.

Przykładowa sekwencja odczytu dla EEPROMu AT25C040 przedstawiona jest na rysunku.



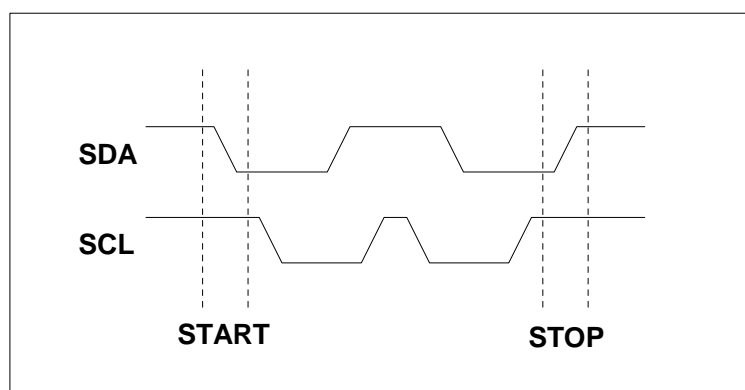
## I<sup>2</sup>C

Interfejs I<sup>2</sup>C jest już zdecydowanie inny. Z 4 linii pozostawiono tylko dwie:

- SCL - zegar transmisji
- SDA - dwukierunkowa linia danych.

## M-11 Programator pamięci EEPROM

Początek i koniec transmisji, czyli zbocza linii CS, zostały zastąpione odpowiednią sekwencją zmian na linii SDA w czasie, gdy linia SCL jest w stanie wysokim.



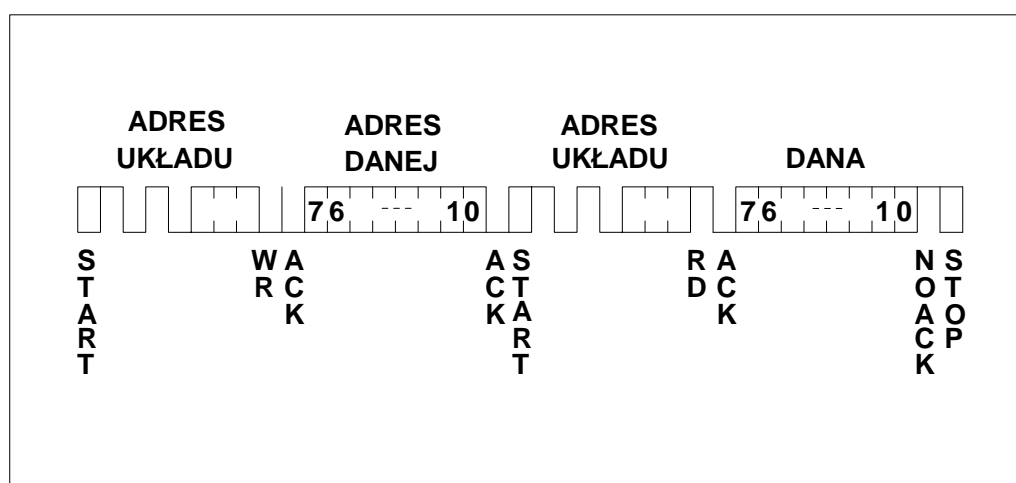
Poza tymi sekwencjami, zmiana na linii danych może następować tylko wtedy, kiedy na linii SCL jest stan niski.

Do interfejsu I<sup>2</sup>C można dołączyć wiele układów, gdyż w transmisji jest przesyłany również adres urządzenia. Każde przesłane 8 bitów jest zawsze potwierdzone na dziewiątym bicie przez układ docelowy.

Trzy najmłodsze bity adresu EEPROMu 24C01A są określone przez stany podane na jego linii A2,A1,A0. Pozwala to na podłączenie do 8 układów 24C01A do jednej szyny I<sup>2</sup>C i niezależne komunikowanie się z każdym z nich. W interfejsach MicroWire i SPI takie rozwiązanie wymagałoby wytworzenia i rozprowadzenia niezależnych linii CS dla każdego EEPROMu.

Stan 1 na linii WP powoduje zabezpieczenie zawartości EEPROMu przed modyfikacją.

Przykład sekwencji odczytu EEPROMu AT24C01A przedstawia rysunek.



Poszczególne interfejsy zostały przedstawione dość pobieżnie. Dokładne dane dostarczonych EEPROMów zawarte są w ich kartach katalogowych, umieszczonych w katalogu DSM-51\Modele\M11.



Załączone, przykładowe programy wykorzystują najprostszy sposób obsługi EEPROMów. Odczyt i zapis wykonywane są zawsze po jednym bajcie. Nie jest to metoda zbyt efektywna. EEPROMy często pozwalają na odczyt kolejnych komórek pamięci bez podawania kolejnych adresów (sequential read). Natomiast zapis odbywa się przeważnie większymi blokami danych, co znacznie przyspiesza programowanie (czas programowania 1 bajtu i jednego bloku jest przeważnie jednakowy). Chcąc jednak wykorzystywać te możliwości, trzeba zawsze dokładnie przejrzeć dane katalogowe konkretnego EEPROMu. Zawsze mogą istnieć mniejsze lub większe różnice pomiędzy EEPROMami różnych firm.

