

Programator szeregowych pamięci EEPROM I²C sterowany z pakietu BASCOM

Małe gabarytowo, tanie i łatwe do nabycia pamięci szeregowo EEPROM już dawno zdobyły sobie uznanie projektantów systemów mikroprocesorowych. Stanowią one niezbędne uzupełnienie mikrokontrolerów niewyposażonych w wewnętrzną pamięć nieulotną, takich jak na przykład popularny AT89C2051.

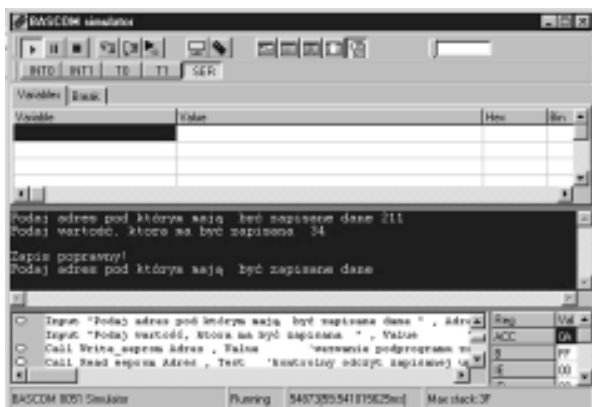
Najczęściej pamięci te służą do przechowywania danych i są programowane i przeprogramowywane w systemie. Jednak niekiedy może zaistnieć potrzeba zaprogramowania takiej pamięci poza systemem i traktowania jej jako dodatkowej pamięci ROM procesora. W pamięci takiej, łatwej do przeprogramowania i wymiany, można przechowywać dane dotyczące sposobu pracy układu, które mogą być zmieniane w zależności od jego aktualnego zastosowania.

Programatory pamięci EEPROM I²C były już wielokrotnie opisywane na łamach Elektroniki Praktycznej, i to, że pozwałam sobie zaprezentować

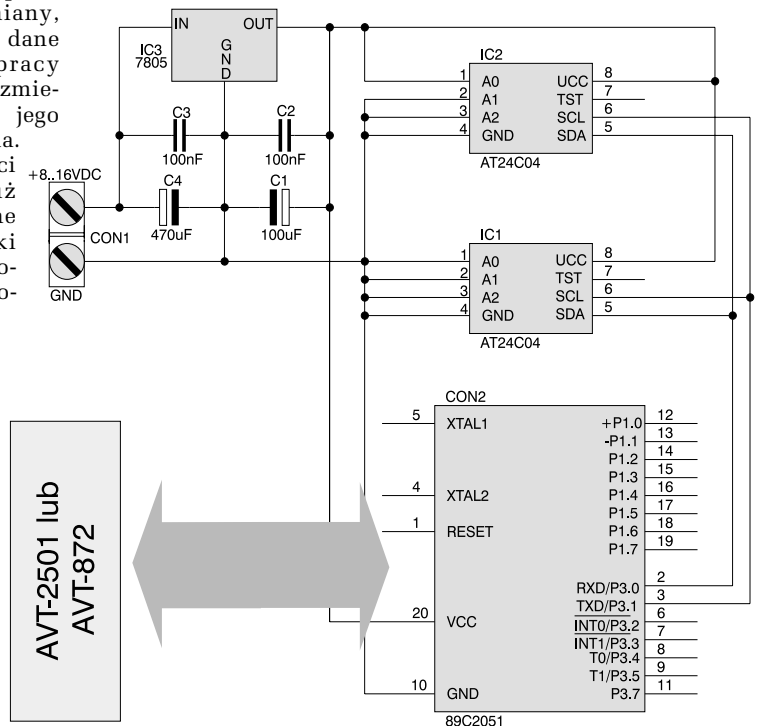
opis kolejnego, usprawnionego jest jedynie jego bardzo nietypowym rozwiązaniem konstrukcyjnym. Programator korzysta bowiem z procesora, z tym że procesora wirtualnego, istniejącego jedynie w pamięci operacyjnej komputera. Nie musimy nawet definiować typu tego

procesora, ale umówmy się, że będzie nim wspomniany już '2051.

O pakietach BASCOM 8051 i AVR zdobywających sobie ogromną popularność wśród elektroników, napisano już w EP wiele pochlebnych słów. Sądzę więc, że większość Czytelników wie,



Rys. 1.



Rys. 2.

```

List. 1.
'##### KONFIGURACJA SPRZĘTOWA #####
$sim 'praca w symulacji
Config Sda = P3.5 'konfiguracja magistrali I2C
Config Scl = P3.7 'konfiguracja magistrali I2C

'##### DEKLARACJE ZMIENNYCH #####

Declare Sub Read_eeprom(adres As Byte, Test As Byte)
'deklaracja podprogramu odczytu pojedynczej
'komórki pamięci EEPROM
Declare Sub Write_eeprom(adres As Byte, Value As Byte)
'deklaracja podprogramu zapisu do pojedynczej
'komórki pamięci EEPROM
Dim Adres As Byte, Value As Byte
'deklaracja zmiennej określającej adres w pamięci
'i zapisywaną wartość
Dim Test As Byte
'deklaracja zmiennej pomocniczej

'##### GŁÓWNA PĘTLA PROGRAMOWA #####
Do
Input "Podaj adres pod którym mają być zapisane dane: ", Adres
'zapytanie o adres, pod który ma być zapisana
'informacja
Input "Podaj wartość, która ma być zapisana: ", Value
'zapytanie o wartość, jaka ma być umieszczona
'w podanej komórce pamięci
Call Write_eeprom Adres, Value
'wezwanie podprogramu zapisania wskazanej komórki
'pamięci podaną wartością
Call Read_eeprom Adres, Test
'kontrolny odczyt zapisanej uprzednio wartości

If Value = Test Then
'jeżeli wynik porównania wartości podanej
'z odczytaną jest pomyślny, to:
Print "Zapis poprawny!"
'wyświetl na ekranie terminala komunikat
'o poprawności zapisu
End If

Loop

'##### PODPROGRAMY ODCZYTU I ZAPISU DANYCH #####

Sub Read_eeprom(adres As Byte, Value As Byte)
I2Cstart
I2Cwbyte 160
I2Cwbyte Adres
I2Cstart
I2Cwbyte 161
I2Crbyte Test, 9
I2Cstop
End Sub

Sub Write_eeprom(adres As Byte, Value As Byte)
I2Cstart
I2Cwbyte 160
I2Cwbyte Adres
I2Cwbyte Value
I2Cstop
Waitms 10
End Sub
    
```

że jednym z najsilniejszych narzędzi zawartych w tym oprogramowaniu są emulatory sprzętowe umożliwiające przetestowanie opracowane-

go programu w środowisku, dla którego został przeznaczony. Jedną z zalet języka MCS-BASIC jest wyjątkowo łatwa obsługa magistrali I²C,

czyli także wykonywania wszystkich operacji związanych z programowaniem pamięci EEPROM z taką magistralą. Jakże stąd płyną wnioski? Ano takie, że minimalnym nakładem środków możemy zbudować programator takich pamięci, składający się zaledwie z dwóch ośmiopinowych podstawek i jednej podstawki 20-pinowej, przeznaczonej do włożenia w nią wtyku emulacyjnego. Oprogramowanie sterujące pracą programatora mamy w zasadzie gotowe: jest nim program BASCOM 8051 i napisane samodzielnie banalnie proste programiki przeznaczone dla wirtualnego procesora, których zadaniem będzie zapisywanie danych w pamięciach, odczytywanie ich, kopiowanie całych pamięci i wszystko to, co tylko przyjdzie Wam do głowy (rys. 1).

Prezentowany programator ma, poza prostotą budowy, jeszcze jedną zaletę: pozwala na skupienie kilku narzędzi niezbędnych konstruktorowi wokół jednego środowiska programowego.

Opis działania

Na rys. 2 pokazano schemat elektryczny programatora, a właściwie przystawki do emulatora sprzętowego współpracującego z pakietem BASCOM. Schemat ten pozostawimy bez komentarza, skupiając się na omówieniu metod programowania układu. Zakładam, że Czytelnik zapoznał się już z pakietem BASCOM i zna podstawowe zasady jego obsługi i pisania programów w języku MCS-BASIC.

Aby wykonać dowolne operacje związane z programowaniem lub odczytywaniem pamięci EEPROM, należy po prostu napisać program, który po skompilowaniu zostanie uruchomiony w emulacji sprzętowej. Sądzę, że przykład programu przedstawiony na list. 1 może być pomocny przy pisaniu własnych, bardziej rozbudowanych programów. Dysponujemy dwoma

WYKAZ ELEMENTÓW

Kondensatory

C1, C4: 100uF/16V

C2, C3: 100nF

Półprzewodniki

IC3: 7805

Różne

CON1: ARK2 (3,5mm)

CON2: podstawka precyzyjna DIL20

IC1, IC2: podstawka precyzyjna DIL8

Płytką drukowaną wraz z kompletem elementów jest dostępna w AVT - oznaczenie AVT-1299.

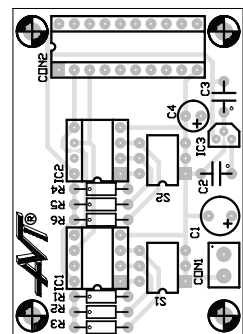
Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: <http://www.ep.com.pl/?pdf/marzec01.htm> oraz na płycie CD-EP03/2001 w katalogu PCB.

podstawkami przeznaczonymi dla pamięci EEPROM i w związku z tym napisanie programu kopiującego zawartość jednej pamięci do drugiej nie powinno nikomu zająć więcej czasu, niż kilka minut. Przypominam tylko, że pamięć w podstawce IC1 posiada adres 160 dla zapisu i 161 dla odczytu, a pamięć umieszczona w podstawce IC2 analogicznie 162 i 163 (adresy podane dziesiętnie).

Montaż i uruchomienie

Na rys. 3 zostało pokazane rozmieszczenie elementów na płytce obwodu drukowanego wykonanego na laminacie jednostronnym.

Zbigniew Raabe, AVT
zbigniew.raabe@ep.com.pl



Rys. 3.