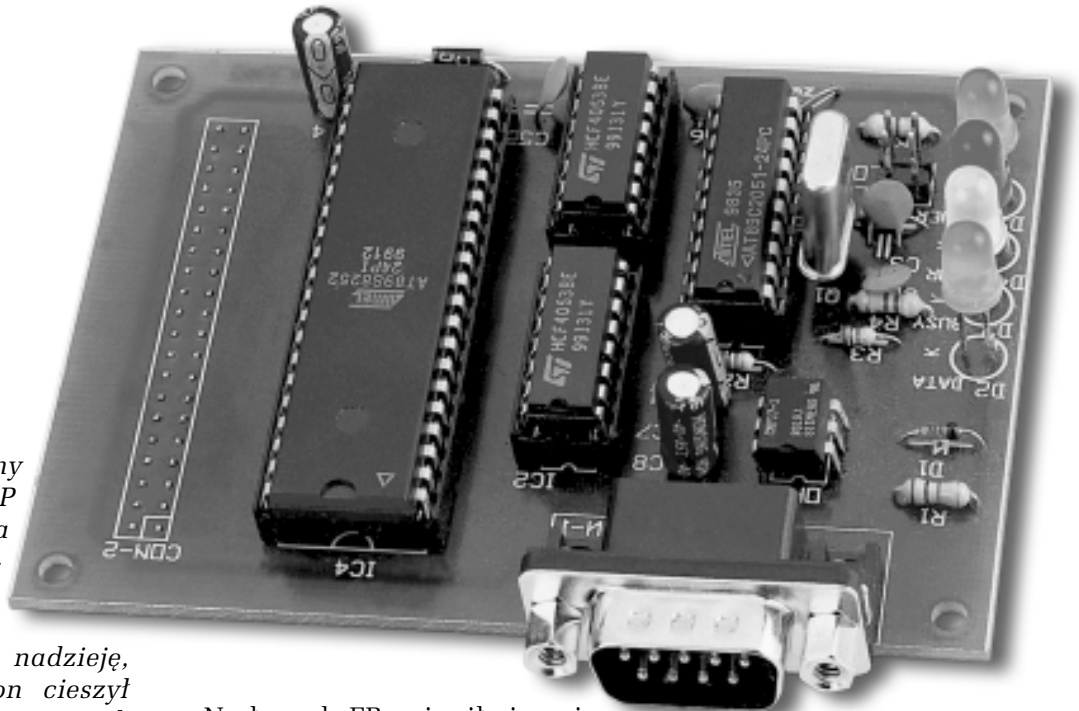


# Emulator 8051 do Amigi

## AVT-995

Przedstawiamy kolejny na łamach EP projekt dla Amigowców - sprzętowy emulator mikrokontrolerów rodziny '51. Mamy nadzieję, że będzie się on cieszył podobnym powodzeniem, jak nasze dotychczasowe amigowe opracowania.



Na łamach EP pojawił się opis programatora-emulatora procesorów AT89Cx051. Dużą zaletą tego urządzenia jest współpraca z każdym typem komputera wyposażonego w port RS232C. W Internecie, na stronie [www.home.mck.pl/~r-mik](http://www.home.mck.pl/~r-mik), są dostępne darmowe programy umożliwiające współpracę Amigi z tym urządzeniem. Niestety, wielkość pamięci programu i liczba dostępnych portów w procesorach serii AT89Cx051 jest często niewystarczająca. Dlatego powstał projekt programatora-emulatora procesorów 8051 z pamięcią programu do 8KB.

Oto jego krótka charakterystyka:

- zasilanie z uruchamianego systemu,
- współpraca z każdym komputerem wyposażonym w port RS232C (także AVT-2250), format transmisji: 4800bd, 8N1,
- izolacja galwaniczna od komputera, dzięki czemu przyłączanie zasilania do programatora może następować bez wyłączenia komputera z sieci,
- przyjmowanie danych w formacie IntelHex,
- emulacja procesorów 8051, 8052, 89S8252,
- programowanie: 89S8252 (także w pracującym urządzeniu przez SPI),
- programowanie bitów blokady 1, 2 i 3,

- programowanie pamięci danych i programu,
- sygnalizacja trybu pracy i błędów.

### Opis działania

Jak widać na rys. 1, budowa emulatora nie jest złożona. Wszystko to za sprawą procesora z magistralą SPI typu AT89S8252, który do programowania wymaga trzech linii oraz linii zerującej. Dzięki temu do przełączania trybu pracy programowanie/emulacja wystarczą dwa multiplexery analogowe.

Zasilanie programatora jest pobierane z uruchamianego systemu (za pośrednictwem kabla złącza emulacyjnego). Kondensatory C4..C8 filtrują napięcie linii zasilania. Opcjonalnie można podłączyć zewnętrzny zasilacz stabilizowany 5V. Sygnały danych z portu RS są konwertowane do poziomów TTL przez transoptor OPT1. Rezystor R1 ogranicza prąd diody D2 (sygnalizującej transmisję po RS) i diody w strukturze transoptora. D1 zabezpiecza diody LED przed zbyt dużym napięciem wstecznym. Transzystor optotriaka steruje wejście RxD procesora IC1. Rezystor R2 „podciąga” to wejście od +5V, dzięki czemu zbrocza są bardziej strome. W tym przypadku wewnętrzny rezystor na wejściu portu ma zbyt dużą rezys-

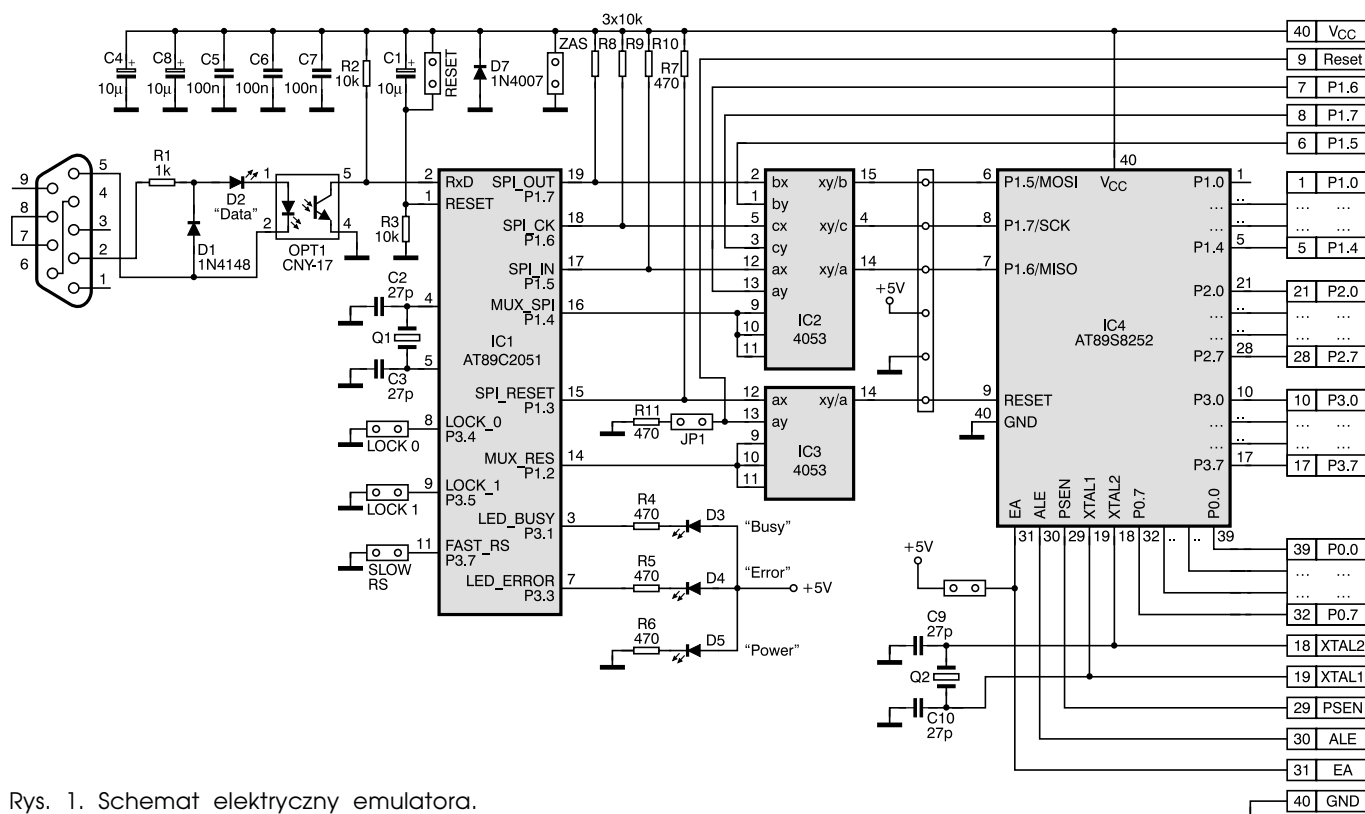
### Właściwości emulatora

#### Zalety:

- Pełna emulacja 8051/8052/AT89S8252.
- Możliwość wykorzystania jako programator procesorów uruchamianych w systemie przez SPI.
- Możliwość wykorzystania jako programator AT89S8252.
- Weryfikacja zapisanych danych.
- Współpraca z każdym typem komputera (dzięki RS, formatowi IntelHex i uniwersalnej procedurze wykrywania znaku końca linii).
- Sygnalizacja kilku rodzajów błędów, co ułatwia uruchamianie.
- Programowanie pamięci danych w AT89S8252.
- Izolacja galwaniczna od uruchamianego systemu.
- Nie wymaga dodatkowego zasilacza.
- Możliwość programowania pamięci „na raty” (nie jest wymagane czyszczenie pamięci procesora przed zapisem).

#### Wady:

- Pobiera dodatkowo 50mA z uruchamianego systemu.
- Brak możliwości symulowania procesorów z kwarcem o częstotliwości mniejszej niż 4MHz.
- Brak możliwości odczytu kodu programu z procesora.
- Stosunkowo mała prędkość transmisji (4800bd) - transmisja 1kB trwa 6s (8kB - 45s)



Rys. 1. Schemat elektryczny emulatora.

tancję, aby skutecznie przeładować pojemności pasożytnicze. Procesor IC1 czeka na otrzymanie znaku „@“ lub „:“: Pierwszy z nich komunikuje, że zostanie wysłany rozkaz, drugi, że plik w formacie IntelHex. Po zdekodowaniu rozkazu czy rekordu pliku IntelHex, procesor IC1 odłącza IC4 od uruchamianego systemu i przyłącza go do procesora IC1. Od tego momentu zaczyna migać dioda D3 („Busy“) sygnalizująca pracę programatora. Multiplexer IC2 odpowiada za przełączanie linii interfejsu SPI, natomiast IC3 za linię zerującą. Po przełączeniu jest wysyłana sekwencja rozpoczęcia programowania przez SPI. Warto wiedzieć, że aby programowanie szeregowe się powiodło, musi być ustawiony odpowiedni bit w procesorze (producent dostarcza procesory w takim stanie, a wartość tego bitu można zmienić jedynie w trybie programowania równoległego), a przy aktywnym stanie linii zerującej (RESET), linia SCK musi znajdować się w stanie niskim. O to wszystko dba IC1 wysyłając odpowiednią sekwencję sterującą multiplexery. Następnie jest wysyłana sekwencja sygnałów inicjalizująca programowanie. Od tego momentu można programować pamięć programu i danych. Należy wspomnieć, że czyszczenie pamięci procesora przed

jego zapisaniem nie jest konieczne, ponieważ przy programowaniu szeregowym każdy bajt przed zapisem jest kasowany. Operacja przygotowania do programowania trwa prawie 20ms. W tym czasie znaki przychodzące po RS są gromadzone w buforze odbiorczym.

Sama procedura programowania jest dosyć skomplikowana. Najpierw sprawdzana jest wartość bajtu w pamięci procesora. Jeśli jest taka sama jak bajtu przeznaczonego do zapisu, następuje wyjście z procedury. Jeśli bajty są różne, wysyłana jest sekwencja zapisująca pamięć, następnie przeprowadzana jest weryfikacja, aż do poprawnego odczytu (wtedy następuje wyjście z procedury) lub przekroczenia czasu 10ms, co oznacza niepowodzenie zapisu. Gdy zapis się nie powiedzie, powyższa sekwencja jest powtarzana jeszcze 3 razy, przy czym dwie ostatnie próby są poprzedzone zerowaniem układu.

Takie skomplikowanie procedury miało na celu wyeliminowanie ewentualnych błędów, jakie mogłyby się pojawić podczas transmisji po SPI. W początkowym stadium uruchamiania emulatora posługiwano się prostszymi procedurami, co powodowało częste pojawianie się błędów „Błąd zapisu bajtu do procesora“. Po poprawnym zapro-

gramowaniu całego kodu badany jest stan jumperów LOCK0 i LOCK1. Zależnie od ich stanu ustawiane są bity zabezpieczające pamięć procesora przed odczytem. I tak odpowiednio:

zworki	ustawione bity
LOCK0 LOCK1	żaden
rozarty rozarty	1
zarty rozarty	1 i 2
rozarty zarty	1, 2 i 3
zarty zarty	1, 2 i 3

Należy wspomnieć, że stan jumperów jest badany przed każdą operacją programowania. Dzięki temu można go zmieniać podczas pracy urządzenia.

Przypomnę znaczenie tak ustawionych bitów:

LED BUSY	LED ERROR	Stan urządzenia
zgaszona	zgaszona	Tryb emulacji
miga	zgaszona	Zajętość emulatora
zgaszona	miga	Przekroczony adres
zgaszona	świeci	Przekroczono czas oczekiwania na transmisję RS
świeci	miga	Błąd pliku IntelHex
miga	miga	Błąd zapisu bajtu do procesora
świeci	świeci	Przepełniony bufor odbiorczy RS
miga naprzemiennie	miga naprzemiennie	Przepełnienie stosu



tora. W tym trybie podstawka pod IC4 musi być pusta. C9, C10 i Q2 nie muszą być zamontowane.

Urządzenie można wykorzystać także jako programator AT89S8252. Sens takiego trybu pracy jest wątpliwy, ale ułatwił testowanie programatora. W tym trybie w miejsce IC4 umieszczamy procesor do zaprogramowania (wtedy R11, C9, C10, Q2 muszą być zamontowane, a JP1 zwarty). Aby zaprogramować procesor należy:

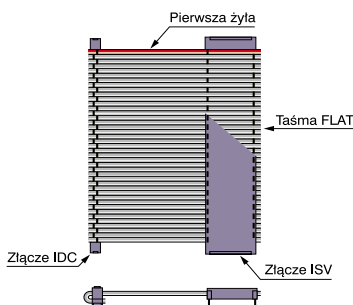
1. umieścić procesor do zaprogramowania w IC4,
2. podłączyć zasilanie +5V do programatora,
3. wysłać RS-em plik IntelHex,
4. odłączyć zasilanie,
5. wyjąć zaprogramowany procesor z podstawki.

W tab. 2 przedstawiono wszystkie tryby pracy programatora.

### Montaż i uruchomienie

Schemat montażowy przedstawiono na rys. 2. Montaż tradycyjnie rozpoczynamy od zwor. W dalszej kolejności montujemy rezystory, podstawki, kondensatory, złącze emulacyjne, listwy do jumperów, półprzewodniki, na końcu gniazdo RS.

Kabel emulacyjny wykonujemy zgodnie z rys. 3. Taśma nie powinna być zbyt długa (max. 15..20cm). Podłączamy zasilanie, sprawdzamy napięcia zasilania układów scalonych. Podłączamy urządzenie do komputera standardowym kablem RS ze skrzyżowanymi liniami TxD i RxD. Uruchamiamy program terminalowy i po ustawieniu przedstawionych wcześniej parametrów transmisji naciskamy dowolne klawisze. Dioda D2 „Data“ powinna migotać. Teraz można umieścić układy w podstawkach. Programator można przetestować bez umieszczania sondy w uruchamianym systemie. W tym celu należy:



Rys. 3. Sposób wykonania kabla emulacyjnego.

1. Zewrzeć jumper JP1 i EA.
2. Zamontować C9, C10, Q2.
3. Przyłutować dwie diody LED z rezystorami szeregowymi 470Ω (jedną do pinu P1.0, drugą do P1.1). Na płytce jest miejsce na LED-y i rezystory oznaczone RTEST1, RTEST2, DTEST1 i DTEST2.
4. Podłączyć zasilanie 5V.
5. Wysłać do Programatora program „Test8252.hex“. W czasie transmisji dioda D3 („Busy“) migocze i da się też zauważyć modulację jasności D2 „data“.
5. Gdy LED-y zgasną, diody podłączone do P1.0 i P1.1 powinny migotać naprzemiennie.

Takie zachowanie świadczy o poprawnej pracy programatora. Należy pamiętać, aby przy normalnej pracy emulatora nie były zamontowane elementy testujące. Pin LOCK1 spełnia dwie funkcje. Procesor odczytując go decyduje, jakie bity zabezpieczające ustawić, natomiast w czasie programowania występują tam ujemne impulsy o czasie trwania równym czasowi programowania jednego bajtu. Pin ten może być pomocny przy określaniu przyczyny wystąpienia błędu przepełnienia bufora RS lub błędu zapisu do procesora. Typowy czas programowania bajtu jest równy 3,8ms. Jeśli czas zwiększy się do 5ms, nastąpi szybkie zapełnienie bufora RS. W miejsce C9, C10 i Q2 warto zamontować odcinki listwy z gniazdkami (tulipanowej), dzięki temu elementy te można umieszczać tam w zależności od potrzeb.

### Obsługa

Teraz co nieco o obsłudze i pisaniu programów. Aby zaprogramować procesor, wystarczy wysłać RS-em plik IntelHex. Można to zrobić systemową komendą *Copy*. Aby wyzerować uruchamiany system, należy wysłać RS-em tekst: @emu8051 @r. Aby wyczyścić pamięć programowanego procesora, należy wysłać RS-em tekst: @emu8051 @c.

Autor, chcąc ułatwić życie użytkownikom emulatora, stworzył odpowiednie skrypty (pliki wsadowe), które uruchamiają kompilację, wysyłają sygnał zerujący do uruchamianego systemu lub czyszczą pamięć procesora. Ikony plików: „@reset“ i „@clear“ można wyciągnąć na blat Workbencha.

### WYKAZ ELEMENTÓW:

#### Rezystory

R1: 1kΩ  
R2, R3, R8..R10: 10kΩ  
R4, R5, R7, R11: 470Ω  
R6: 510Ω

#### Kondensatory

C1, C4, C8: 10μF  
C2, C3: 27pF  
C5..C7: 100nF

#### Półprzewodniki

D1: 1N4148  
D2..D5: LED  
D7: 1N4007  
IC1: AT89C2051  
IC2, IC3: 4053  
IC4: 89S8252  
OPT1: CNY-17

#### Różne

Q1: 12,000MHz  
CON-1: DB9PIN-M  
CON-2: IDC40

*Elementy opcjonalne (patrz tekst)*

C9, C10: 27pF

Q2: min 4,000MHz

*Elementy opcjonalne do testowania urządzenia (patrz tekst)*

DTEST1, DTEST2: LED  
RTEST1, RTEST2: 470Ω

Po kliknięciu na nich, pamięć procesora zostanie wyczyszczona lub wysłany będzie sygnał zerowania do uruchamianego systemu. Oczywiście nic nie stoi na przeszkodzie, aby skrypty „podpiąć“ pod klawisze funkcyjne. W pliku „Compiler.DokPL“ opisano dokładnie sposób instalacji assemblera 8051. Należy jeszcze dodać, że na płycie CD-EP12/2000B znajdują się skrypty i *include* dla kitów AVT-2250 i AVT-498 (programator-emulator AT89Cx051).

**Sławomir Skrzyński, AVT**

*Najnowsze wersje oprogramowania dla Amigi i PC będą dostępne na stronie [www.ep.com.pl](http://www.ep.com.pl) w dziale download i pod adresem: <http://home.mck.pl/~r-mik>.*

*Uwagi, propozycje, zauważone błędy proszę przysyłać na adres: [skrzyński@zt.wloclawek.tpsa.pl](mailto:skrzyński@zt.wloclawek.tpsa.pl), w polu treść listu umieścić: emulator 8051.*

*Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: <http://www.ep.com.pl/pcb.html> oraz na płycie CD-EP12/2000 w katalogu PCB.*