

Dział „Projekty Czytelników” zawiera opisy projektów nadesłanych do redakcji EP przez Czytelników. Redakcja nie bierze odpowiedzialności za prawidłowe działanie opisywanych układów, gdyż nie testujemy ich laboratoryjnie, chociaż sprawdzamy poprawność konstrukcji.

Prosimy o nadsyłanie własnych projektów z modelami (do zwrotu). Do artykułu należy dołączyć podpisane **oświadczenie**, że artykuł jest własnym opracowaniem autora i nie był dotychczas nigdzie publikowany. Honorarium za publikację w tym dziale wynosi 250,- zł (brutto) za 1 stronę w EP. Przesyłanych tekstów nie zwracamy. Redakcja zastrzega sobie prawo do dokonywania skrótów.

# Prosty programator mikrokontrolerów AT89Cx051/AT89C5x

**Projekt 107**



Na rynku dostępnych jest wiele programatorów uniwersalnych (drogich) i specjalizowanych dla danej grupy układów (tańszych). Taki właśnie tani, lecz funkcjonalny programator przedstawiamy w artykule.

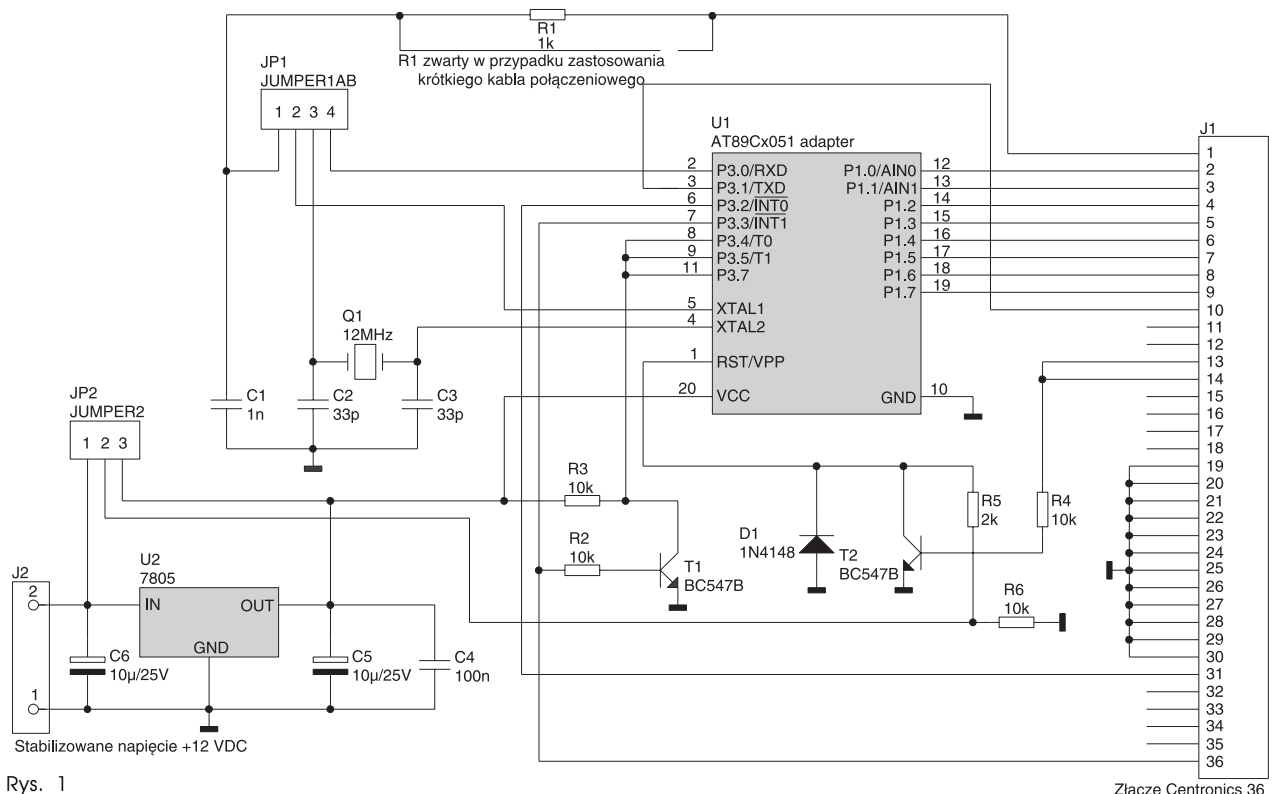
Niewątpliwą zaletą mikrokontrolerów zgodnych programowo z 8051, jest mnogość dostępnego kodu źródłowego. Stwarza to możliwość łatwego wypróbowania swoich sił m.in. w assemblarze. Zapewne wielu elektroników pragnęłoby także dołączyć do grona użytkowników Bascoma 8051, jednak barierą

dla nich jest cena narzędzi sprzętowych (układy PCF8574A są stosunkowo drogie). Słuszną alternatywą w takim wypadku jest zbudowanie własnego programatora, którego koszt wykonania będzie ograniczony do minimum. Od takiego programatora wymagana jest tylko jedna cecha - ma kasować i zapisywać pamięć Flash układu. Nie musi on ustawiać żadnych dodatkowych opcji w programowanym układzie, gdyż ma służyć on do celów „laborato-

ryjnych“, a nie komercyjnych (choć można go stosować i tutaj). Okazuje się, że w Internecie opisów takich programatorów jest wiele. Jednak albo ich oprogramowanie pozostawia wiele do życzenia albo mają drobne, acz istotne błędy w schemacie.

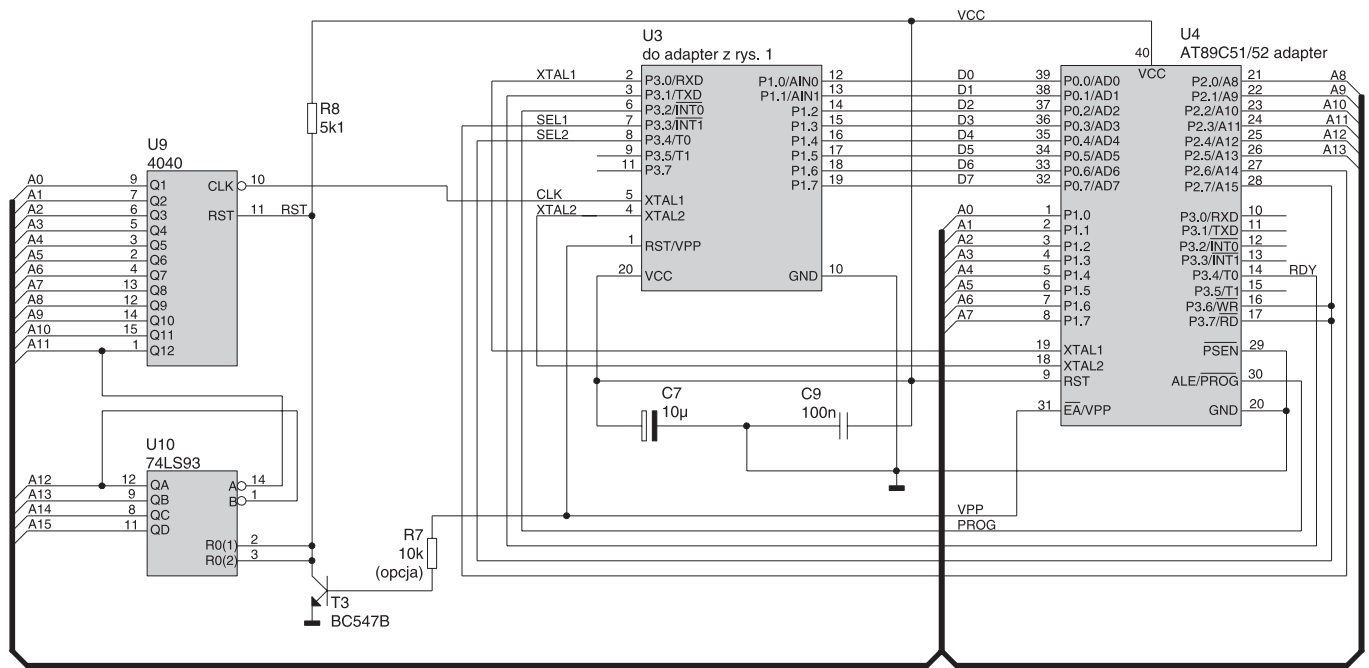
### Budowa i działanie

Postanowiłem więc zaprojektować własny programator, który będzie domyślnie programował układy AT89Cx051, a po zastosowaniu dodatkowej płytki, także AT89C5x. Zakładam, że rozpoczynający przygodę z mikrokontrolerami dysponują zazwyczaj komputerem, więc sterowanie pracą



Rys. 1

Złącze Centronics 36



Rys. 2

programatora zostało przerzucone na PC-ta. Program do jego obsługi został napisany w Turbo Pascalu i jest dostępny na płycie CD-EP2/2003.

Schemat elektryczny programatora pokazano na rys. 1. Programator jest wykrywany automatycznie przez oprogramowania za sprawą połączenia wyprowadzeń 13 i 14 złącza drukarkowego. Owo złącze ma tę zaletę, że nie potrzeba nam dodatkowego kabla do portu równoległego - wystarczy odłączyć go od drukarki. Program sprawdza również, czy programowany układ jest włożony do programatora (linia ACK), a w przypadku braku odpowiedniego napięcia programującego również informuje o błędzie.

Tranzystor T1, sterowany sygnałem DSL ustala, czy układ będzie kasowany czy programowany. Tranzystor T2, sterowany sygnałem ALF włącza napięcie programujące. Dioda D1 zabezpiecza programowany układ przed uszkodzeniem w przypadku, gdy podłączymy odwrotnie spolaryzowane napięcie zasilania. Wówczas co najwyżej wzrośnie temperatura stabilizatora U2.

Rezystor R1 wraz z kondensatorem C1 stanowi układ

całkujący dla sygnału „zegarowego”. Jest on niezbędny gdy stosujemy długi przewód połączeniowy. W praktyce okazało się jednak, że R1 w większości przypadków powinien być zwarty.

Zworka JP1 pozwala wybrać źródło sygnału zegarowego. Kiedy programujemy powinniśmy wybrać pozycję 1-2, zaś jeśli chcemy „na szybko” sprawdzić poprawność naszego prostego programu (np. generatora) po zaprogramowaniu układu, należy wybrać pozycję 2-3. Kiedy chcemy programować układ AT89C51 za pomocą omówionego dalej adaptera dla „dużych” ’51 (drugiej płytki) powinniśmy ustawić dwie zworki na pozycjach 1-2 oraz 3-4.

Zworka JP2 pozwala wybrać napięcie programujące dla „większych” ’51, a także sterować sygnałem zerującym mikrokontrolera. W pozycji 1-2 ustalamy napięcie 12 V, zaś 2-3 5 V. Gdy zdejmemy zworkę (ustawimy ją prostopadłe) napięcie wyniesie 0. Procedurę programowania omówię dalej. Dane 8-bitowe są przesyłane do układu bezpośrednio z portu (D0...D7) podobnie jak sygnał programujący (INI).

W adapterze dla układów AT89C5x (schemat elektryczny na rys. 2) wewnętrzny licznik adresu układów AT89x051 został zastąpiony przez liczniki U9 i U10, co pozwala adresować pamięć o 16-bitowym adresie (można go w razie potrzeby wykorzystać do programowania innych układów, np. pamięci EP-ROM). Układy te są zerowane poprzez tranzystor T3 w czasie, gdy nie ma żadnego napięcia programującego. Najlepszym rozwiązaniem jest umieszczenie w miejscu T3 tranzystora MOSFET (z kanałem typu N) i usunięcie rezystora R7. Wtedy układ zerujący nie będzie obniżał napięcia programującego.

**Montaż i uruchomienie**

Podstawowa płytka programatora została zaprojektowana jako jednowarstwowa. Jej wzór zamieszczamy wraz z oprogramowaniem na CD-EP3/2003.

Montaż programatora należy rozpocząć od ośmiu zwork. Najlepiej od razu również zerwać także wyprowadzenia rezystora R1. Następnie wlotowujemy kolejno podstawki pod układ scalony, stabilizator, rezonator kwarcowy. Obudowę tego ostatniego, w celu ograniczenia liczby harmonicznych, warto połączyć z masą, co jest możliwe, gdy zamontujemy go na leżąco. Jeśli wykonaliśmy te czynności, zabieramy się do montażu elementów dyskretnych. Na koń-

cu wlotowujemy goldpiny (do jumperów - JP2 kątowe - i zasilania) oraz złącze drukarkowe, od którego należy wcześniej odkręcić plastikową część lub zwikszyć w niej odległość nóżek pomiędzy rzędami.

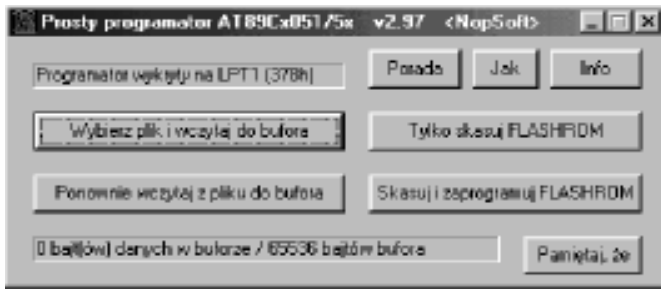
Uruchomienie rozpoczyna się od podłączenia napięcia zasilania (12 V, stabilizowane) i sprawdzenia poprawności napięć w podstawie do programowania układu. Następnie możemy zabrać się do pierwszego programowania.

Druga płytka - adaptera dla AT89C5x - pomimo wszelkich prób minimalizacji, została zaprojektowana jako dwustronna. Wybrałem kompromis w liczbie przelotek dla tych osób, które będą wykonywały ją jako dwuwarstwową oraz dla osób, które wykonają ją jako jednostronną, a resztę połączeń wykonają przewodami (np. kynarem ścieżki sygnałowe i jakimś grubszym ścieżki zasilania). Montaż tej płytki rozpoczyna się od podstawek układów, poza U3. Następnie wlotowujemy elementy dyskretnie. Teraz zabieramy się za U3 i wlotowujemy od spodu płytki dwa rzędy 10-nóżkowych goldpinów (dla druku jednostronnego wpuszczamy je w płytkę tylko w niewielkim stopniu).

Po tej operacji osoby wykonujące płytkę jednostronną powinny zabrać się za połączenia przewodowe (z górnej warstwy druku dwustronnego). Goldpiny nie będą nam

**W celu programowania AT89C55 należy dodać dwie ścieżki dla linii A14 oraz A15. W tym celu należy połączyć:**

- 8 wyprowadzenie 74LS93 (U10) z 10 wyprowadzeniem AT89C5x (U4),
- 11 wyprowadzenie 74LS93 (U10) z 11 wyprowadzeniem AT89C5x (U4).



Rys. 3

nietety pasowały do podstawki precyzyjnej w płytce bazowej. Musimy wykonać prostą przejściówkę - czyli użyć dwóch jednorzędowych podstawek precyzyjnych. One będą już nam pasowały. Ewentualnie, zamiast tego całego „zamieszania” wokół U3, możemy użyć kabla (taśmy) z wtykiem emulacyjnym.

Uruchomienie drugiej płytki polega na prawidłowym wpięciu U3 do podstawki U1, a następnie zmierzeniu w podstawkach napięć zasilających (uwaga na licznik U10, który ma napięcie zasilania na nóżkach 5 i 10). Jeśli są poprawne, to możemy umieścić w nich układy scalone. Jeśli nie, należy dokładnie przeanalizować połączenia. Teraz wystarczy uruchomić program i włożyć w podstawkę układ do zaprogramowania.

### Obsługa programatora

Programowanie układu nie jest zbyt skomplikowane. Najpierw przedstawiamy procedurę programowania mikrokontrolerów AT89C51:

1. Podłączamy do programatora wymagane zasilanie oraz przewód od drukarki.
2. Ustawiamy jumper JP1 na pozycji 1-2 (*clock*), a JP2 2-3 (5 V).
3. Uruchamiamy program *My\_x051* z nazwą pliku binarnego, którego zawartość chcemy umieścić w programowanym układzie, jako parametr.
4. Wkładamy układ do podstawki.
5. Przesławiamy jumper JP2 na pozycję 1-2 (12 V).
6. Wciskamy ENTER.
7. Po zakończeniu programu przesławiamy jumper JP2 z powrotem na pozycję 2-3 (5 V).
8. Odłączamy zasilanie (niekoniecznie) i (starannie) wyjmujemy zaprogramowany układ.

Płytkę bazową możemy wykorzystywać także jako prostą płytkę uruchomieniową:

1. Podłączamy do progra-

matora wymagane zasilanie, ale bez przewodu od drukarki.

2. Ustawiamy jumper JP1 na pozycji 2-3 (*xtal*), a JP2 2-3 (5 V - *reset on*).

3. Wkładamy układ AT89C51 do podstawki.

4. Zdejmujemy JP2 (*reset off*) - od tej pory nasz program jest wykonywany.

5. Program restartujemy zakładając ponownie JP2 na pozycję 2-3 (5 V).

6. Odłączamy zasilanie i wyjmujemy zaprogramowany układ.

Programowanie mikrokontrolerów AT89C5x przeprowadza się następująco:

1. Ustawiamy dwie zworki JP1 na pozycjach 1-2 (*clock*) i 3-4 (*xtal*), a JP2 na pozycję 2-3 (5V).

2. Wpinamy (poprawnie!) adapter AT89C5x do płytki bazowej.

3. Podłączamy do programatora wymagane zasilanie oraz przewód od drukarki.

4. Uruchamiamy program *My\_x051* z nazwą pliku binarnego, którego zawartość chcemy umieścić w programowanym układzie, jako parametr.

5. Wkładamy programowany układ do podstawki.

6. Jeśli mamy wersję wysokonapięciową układu (12 V zamiast 5 V), przestawiamy jumper JP2 na pozycję 1-2 (12 V).

7. Wykonujemy punkty 6, 7 i 8 dla AT89C51.

Oprogramowanie sterujące programatora będzie wciąż rozwijane tak, aby nie wymagało szybkiego CPU (w tej chwili czas mierzony jest bardzo dokładnie, zajmując przy tym znaczny czas procesora). Dostępna jest także wersja przystosowana do pracy „pod opieką” systemem Windows (rys. 3). Nowe wersje pojawiają się na stronie autora: <http://nopsoft.republika.pl>.

### Uwagi końcowe

Większość kompilatorów generuje jedynie pliki *hex*. Aby móc je wykorzystać w na-

szym programatorze, musimy je przekonwertować do pliku *bin*. Możemy to uczynić z łatwością stosując np. program *hex2bin* (zamieszczamy go na CD-EP2/2003) i podając jako parametr nazwę pliku *hex*.

Pliki tego formatu generuje także Bascom 8051, ale należy je najpierw od niego „wykraść”. Na początku wybieramy z menu *Options>Compiler>Output* i zaznaczamy pola *Binary File* i *HEX File* (w razie potrzeby *Old Intel HEX File*). Następnie przechodzimy do *Options>Programmer>Other*. W polu *Programmer* wybieramy *External Programmer*, odznaczamy opcje *Auto Flash* i *Auto Verify*, a w *Program* wskazujemy program edycyjny *notepad* i zaznaczamy *Use HEX File*. W okno *Parameter* wpisujemy {file}, a w *Mode* wybieramy *Go to application directory and run*. Jeśli potrzebujemy danych w formacie *bin* to odznaczamy opcję *Use HEX File*. Po skompilowaniu programu wybieramy *Run Programmer* i możemy z poziomu notatnika zapisać plik *bin*, a następnie przejść do DOS-u i zaprogramować nasz układ.

DOS-em nie musimy się jednak w Bascomie przejmować. Gdy nie chcemy stosować powyższej, dość długiej metody, wystarczy wybrać jako programator urządzenie o nazwie *BlowIT Programmer* (dostępny na pewno w wersji Bascom 8051 2.0.2.0 i 2.0.6.0) - prezentowany programator jest z nim kompatybilny. Następnie wybieramy odpowiedni port LPT (*LPT-address*) w zakładce *Parallel* (programator nie zostanie wykryty automatycznie), ustawiamy *Port Delay* np. na 5 i odznaczamy pozostałe pola. Od tej pory możemy programować z poziomu Bascom-a układy do 4 kB pamięci Flash. Pozostaje jeszcze dodać, że trzeba prawidłowo wybrać ustawienie zworek: JP1 w pozycji 1-2 (dla AT89C5x dodatkowo w 3-4), JP2 dla AT89C51 w pozycji 1-2, a dla AT89C5x w zależności od wersji (5/12 V).

Przypominam o możliwości sprawdzenia działania prostego programu w programatorze (po odłączeniu kabla LPT). Należy tylko ustawić odpowiednio zworki. Jako przykładowy program testowy proponuję (Bascom):

### WYKAZ ELEMENTÓW

#### Rezystory

- R1: 1kΩ lub zwora (patrz opis)
- R2...R4, R6: 10kΩ
- R5: 2kΩ
- R7: 10kΩ lub zwora (patrz opis)
- R8: 5,1kΩ

#### Kondensatory

- C1: 1nF
- C2, C3: 20...40pF
- C4, C9: 100nF
- C5, C6, C7: 10μF/10V

#### Półprzewodniki

- U2: 7805
- U9: CMOS 4040
- U10: 74LS93
- D1: 1N4148
- T1, T2: BC547B
- T3: BC547B lub MOSFET-N BS170, BS109 (patrz opis)

#### Różne

- U1: podstawka precyzyjna DIP20
- U3: dwa goldpin-y oraz dwie podstawki precyzyjne jednorzędowe
- U4: podstawka precyzyjna DIP40
- U9: podstawka DIL16
- U10: podstawka DIL14
- Q1: rezonator kwarcowy 4...12MHz
- JP1: goldpin z dwoma jumperami
- JP2: goldpin z jumperem
- J1: 36-nóżkowe złącze drukarkowe (żeńskie)
- J2: goldpin

```
Do
cpl p1.7
Wait 1
cpl p1.6
Loop
a dla assemblera:
loop1:
  CPL P1.7
  MOV A, #5
;zależnie od częstotli-
;wości rezonatora
;kwarcowego
loop2:
  PUSH ACC
  CLR A
loop3:
  MOV B, #0
  DJNZ B, $
  DJNZ ACC, loop3
  POP ACC
  DJNZ ACC, loop2
  CPL P1.6
  SJMP loop1
```

Pozostaje już tylko uruchomić i sprawdzić woltomierzem lub sondą logiczną czy zmieniają się stany wyprowadzeń 18 i 19 układu AT89C51.

**Janusz Użycki**  
nopsoft@skrzyinka.pl