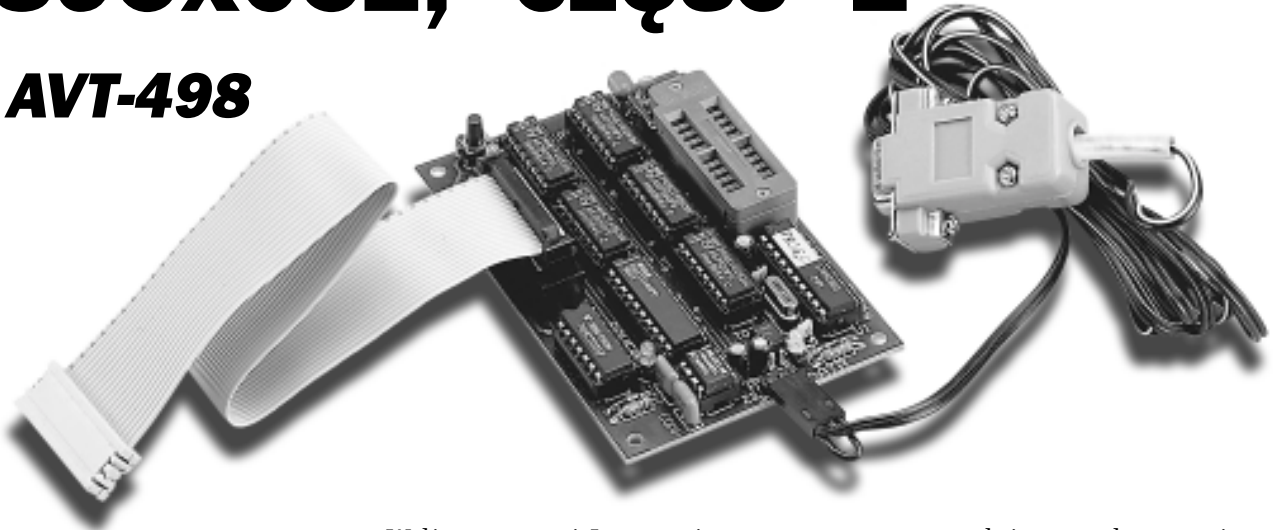


Programator - emulator mikrokontrolerów AT89Cx051, część 1

kit AVT-498



Mikrokontrolery AT89Cx051 produkowane przez firmę Atmel cieszą się wśród konstruktorów zasłużonym powodzeniem. Ich niska cena, zgodność z popularnymi '51, łatwa dostępność, pamięć programu typu Flash oraz stosunkowo wysoka wydajność sprawiają, że konkurenci mają trudny żywot na rynku. Urządzenie prezentowane w artykule pozwala tanio rozwiązać większość problemów narzędziowych, jakie mogą napotkać konstruktorzy korzystający z procesorów x051. Na jednej, niewielkiej płycie otrzymujecie programator oraz emulator - czyli komplet narzędzi, pozwalający bez trudu zbudować dowolne mikroprocesorowe urządzenie.

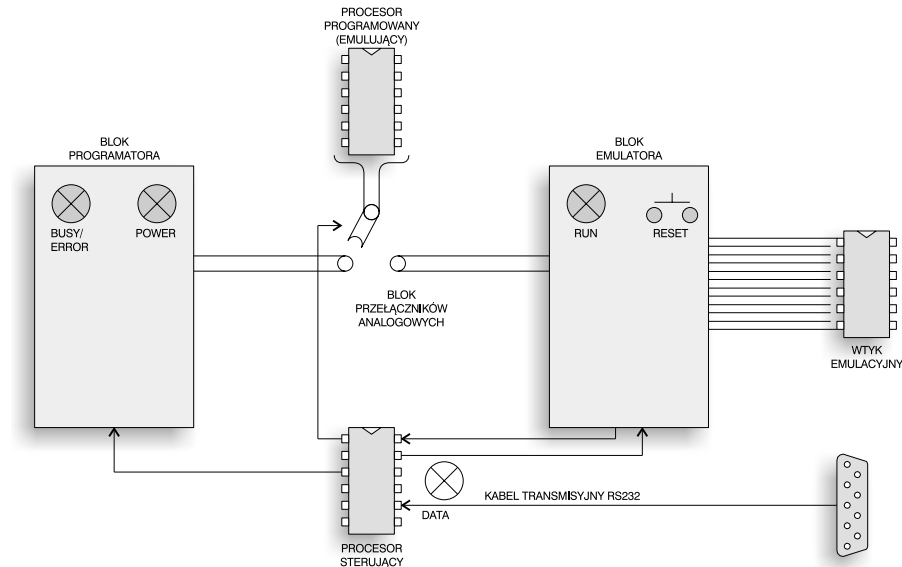
W literaturze i Internecie pojawiło się już wiele opisów programatorów dla mikroprocesorów firmy Atmel z pamięcią FLASH. Trochę mniej znalazłoby się projektów emulatorów. Te ostatnie pracują zazwyczaj z własnym generatorem zegarowym i słabo (lub wcale) emulują wysokoomowe wejścia wewnętrznego komparatora. Często wymagają sprzętowego ustawiania konfiguracji pinów P1.0 (AIN0) i P1.1 (AIN1). Gdy chcemy używać je jako analogowe wejścia komparatora czy wejścia lub wyjścia cyfrowe, to trzeba odpowiednio ustawić zworki.

Pomysł opracowania takiego urządzenia powstał wówczas, gdy niżej podpisany "spalił" kolejno trzy procesory przez odwrotne włożenie ich do tej samej podstawki. Konstrukcja układu nawiązuje do zestawów uruchomieniowych STARTER KIT i łączy w sobie cechy programatora i emulatora. Dlatego będzie z pewnością atrakcyjna dla kolegów stawiających pierwsze kroki w programowaniu mikroprocesorów jednocukładowych. Praktyka pokazała jednak, że w rękach trochę bardziej zaawansowanego elektronika-programisty może stać się potęż-

nym narzędziem, skutecznie wspomagającym uruchamianie nowych układów.

Dla niektórych Czytelników może być ważna informacja, że układ współpracuje poprawnie z KAŻDYM komputerem wyposażonym w łącze szeregowo, jeśli tylko można na nim ustawić odpowiednie parametry transmisji. Po niewielkiej przeróbce obwodu wejściowego, programator może nawet pracować z "nie-dokończonym" portem o poziomach napięcia 0 i +5V. Osobną sprawą jest dostępność programów assemblera dla procesorów rodziny MCS-51 na komputery typu Amiga, Atari czy Commodore. Zapewne w Internecie można znaleźć takie oprogramowanie z opisem i tekstami źródłowymi.

Jak w każdej tego typu realizacji, przyjęte rozwiązanie jest kompromisem między kosztem elementów i funkcjonalnością układu. Wymieniając podstawowe cechy urządzenia rozdzieliłem je na cachy programatora i emulatora, dla każdego z nich wyodrębniając najistotniejsze wady i zalety. Powinno to pomóc w szybkim zorientowaniu się, gdzie przebiega linia wspomnianego kompromisu.



Rys. 1. Schemat blokowy urządzenia.

Może najpierw - aby przybliżyć Czytelnikowi jaka idea przyświecała mi przy opracowywaniu projektu programatora - pokrótce opiszę, w jaki sposób tworzę oprogramowanie dla mikroprocesorów jednocukrowych. Zaczynam od skopiowania programu z jakiegoś starego, podobnego projektu. Aktualizuję deklaracje wejść, wyjść i stałych. Sprawdzam i ewentualnie zmieniam ustawienia rejestrów SFR dla przerwań i timerów. Tworzę od nowa lub poprawiam procedury obsługi przerwań. Na końcu piszę program główny. Uruchamianie zaczynam od symulatora, a gdy program jest prosty, przechodzę od razu do emulatora. Po znalezieniu i usunięciu poważnych błędów, program zaczyna z grubsza realizować to co powinien. Teraz trzeba sprawdzić poszczególne jego fragmenty. W tej fazie bardzo ważna jest umiejętność przewidywania stanów wejściowych, jakie może otrzymać program. Zazwyczaj to właśnie zajmuje najwięcej czasu. Znacznie szybsze jest wprowadzenie poprawek lub uzupełnień. Rzadko, ale jednak zdarza się, że na tym ostatnim etapie nie mogę korzystać z emulatora. Wynika to albo z "pływającego" zasilania w układzie docelowym, albo z ograniczeń emulatora. Kończy się na wielokrotnym wyjmowaniu procesora z podstawki w układzie uruchamianym, programowaniu jego pamięci poprawionym programem i wkładaniu

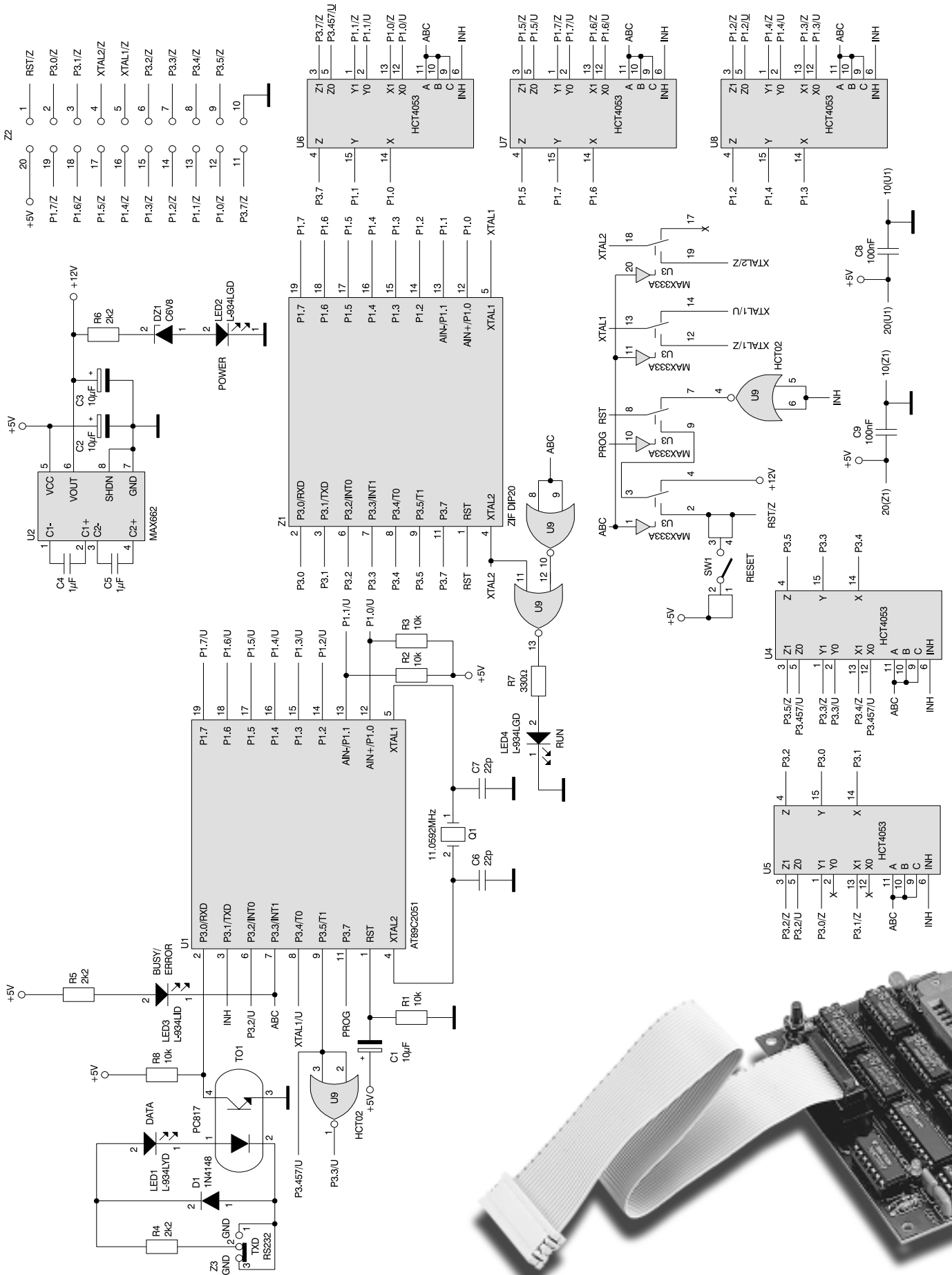
z powrotem do podstawki. Przy każdej z tych czynności można się pomylić i włożyć procesor odwrotnie. Dopracowane firmowe programatory potrafią wykryć i zasygnalizować taką sytuację, dzięki czemu unika się uszkodzenia mikroprocesora. Błąd przy wkładaniu do podstawki w układzie uruchamianym kończy się zazwyczaj znacznie gorzej. Chcąc uniknąć takich sytuacji zacząłem opracowywać dla procesora przełącznik PROGRAMATOR<->UKŁAD_DOCELOWY. Dzięki temu nie byłoby potrzeby wyjmowania procesora z podstawki, a wystarczałoby proste przełączanie. W czasie opracowywania szczegółowej koncepcji układ trochę się rozrósł. Do przełącznika zrealizowanego na scalonych kluczach analogowych dobudowałem programator. Z drugiej strony dodano kilka prostych, ale bardzo użytecznych drobiazgów, takich jak przycisk zerowania i wskaźnik zatrzymania procesora. Schemat blokowy powstałego w ten sposób programatora-emulatora przedstawiono na rys. 1. Procesor wkłada się w podstawkę programatora. Do układu uruchamianego dołącza się przewód taśmowy zakończony wtykiem emulacyjnym. Plik z programem w postaci binarnej, który przychodzi po łączu szeregowym jest wpisywany do pamięci FLASH procesora. Po zaprogramowaniu procesor jest dołączany do uruchamianego układu i rozpoczyna wykonywanie zapisanego programu.

Opis układu

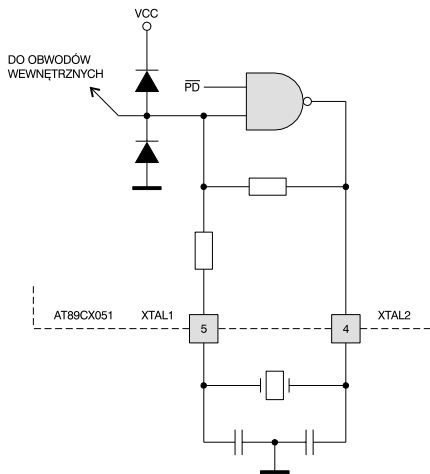
Schemat elektryczny programatora przedstawiono na rys. 2. Przyjęto zasadę, że sygnały z indeksem /U pochodzą z procesora sterującego, zaś indeks /Z oznacza, że linie są połączone z wtykiem emulacyjnym za pośrednictwem złącza Z2. Linie sygnałów bez indeksu są połączone z programowanym procesorem znajdującym się w podstawce Z1.

Układ sterowania programatora zrealizowano na procesorze AT89C2051 (U1) taktowanym sygnałem z oscylatora kwarcowego (rezonatorem kwarcowym Q1 o częstotliwości 11,0592MHz). Jednokierunkową komunikację z komputera do programatora, przy równoczesnym zapewnieniu izolacji galwanicznej, umożliwia transoptor TO1. Dioda LED1 (żółta) włączona szeregowo z diodą świecącą transoptora służy do sygnalizowania transmisji z komputera. Dioda D1 włączona antyrównolegle do diod świecących TO1 i LED1 ogranicza napięcie o odwrotnej polaryzacji, które mogłoby je uszkodzić. Rezystor R4 ogranicza prąd pobierany ze złącza szeregowego komputera. Rezystor R8 zapewnia dodatkowe podciągnięcie pinu RxD procesora do napięcia +5V i przyspiesza powrót tego wejścia do stanu wysokiego. Złącze Z3 to listwa z trzema igłami, z których dwie zewnętrzne są połączone z masą, a środkowa to sygnał TxD z komputera. Dzięki takiemu rozwiązaniu nie trzeba uważać na sposób podłączania gniazda na kablu transmisyjnym do programatora.

Układy U4, U5, U6, U7 i U8 (HCT4053) to zespół przełączników analogowych, dzięki którym jest możliwe przełączanie wejść/wyjść programowanego procesora umieszczonego w podstawce Z1. Do przełączania sygnałów zegarowych XTAL1 i XTAL2 oraz napięcia programującego konieczne było zastosowanie przełączników (U3) o lepszych parametrach. Wybór padł na układ MAX333A firmy Maxim. Jest to układ zawierający cztery analogowe klucze przełączne o niskiej rezystancji w stanie włączenia. Ponadto układ zasilany tak jak w naszym przypadku ze źródła o napięciu +12V może być, bez żadnych konwer-



Rys. 2. Schemat elektryczny programatora.



Rys. 3. Sposób sterowania generatora.

terów poziomu, sterowany sygnałami o poziomach TTL. Do sterowania przełącznikami służą trzy sygnały ABC, INH i PROG. W czasie programowania sygnał ABC przyjmuje niski stan logiczny. Powoduje to przełączenie procesora umieszczonego w podstawce Z1 na programator i zapalenie czerwonej diody LED3 oznaczającej programowanie. Port P1 (Z1) jest dołączony do portu P1 procesora sterującego (U1). Na liniach tego portu pojawiają się bajty odebrane z portu szeregowego przeznaczone do wpisania do pamięci programu. Rezystory R2 i R3 podciągają wyjścia P1.0 i P1.1 do +5V. Do wyprowadzeń sterujących P3.4, P3.5 i P3.7 jest dołączany sygnał P3.457/U. Na pin P3.3 jest podany sygnał P3.457/U zanegowany w bramce (1, 2, 3) układu U9. Pozwala to wybrać jeden z dwóch trybów programowania:

- *Chip Erase* - kasowanie pamięci programu: P3.457/U = L;
- *Write Code Data* - zapis do pamięci programu: P3.457/U = H.

Do programowania pamięci typu FLASH konieczne jest podanie napięcia +12V na wejście RST (Z1). Służą temu dwa przełączniki układu U3. Jeden sterowany sygnałem PROG z wyjścia P3.7 (U1) i drugi sterowany sygnałem ABC. Na wyprowadzeniu P3.2 pojawiają się w czasie programowania impulsy zezwalające - sygnał P3.2/U. Po zaprogramowaniu każdego bajtu inkrementowany jest wewnętrzny licznik adresu - Internal Address Coun-

ter. Służy do tego krótki impuls podany na wyprowadzenie XTAL1 (Z1), pochodzący z wyjścia P3.4 (U1). Sygnał INH powoduje w stanie wysokim zablokowanie kluczy analogowych i po zanegowaniu w bramce (4,5,6) U9 - podanie niskiego stanu logicznego na wejście RST programowanego procesora, co jest konieczne podczas sekwencji rozpoczynającej programowanie.

Napięcie zasilające +5V jest czerpane wprost z układu uruchamianego. Pobór prądu w stanie spoczynku z pustą podstawką Z1 wynosił w układzie modelowym 16mA. W czasie programowania prąd wzrastał do 26mA, a w trybie emulacji w układzie z zegarem 12MHz wynosił około 33mA. Napięcie programujące +12V jest wytwarzane w układzie MAX662, którego odpowiednikiem jest LTC1262 firmy Linear Technology. Są to przetwornice kondensatorowe (bezindukcyjne), zaprojektowane specjalnie do wykorzystywania przy programowaniu pamięci FLASH. Zielona dioda LED2 jest dołączona do napięcia +12V przez szeregowo połączoną diodę Zenera o napięciu 6,8V i rezystor R6 ograniczający prąd. Dzięki temu LED1 świecąc wskazuje nie tylko obecność napięcia zasilającego, ale również poprawną pracę układu przetwornicy. Napięciem +12V są zasilane przełączniki analogowe z układu U3. Jest to konieczne, gdyż klucze te służą m.in. do przyłączenia napięcia programującego.

Układy dodatkowe

Na płytce drukowanej programatora znalazło się miejsce na przycisk RESET, włączony standardowo między wejście RST (Z1) i napięcie +5V. Przy uruchamianiu oprogramowania bardzo przydatna jest możliwość ręcznego wyzerowania procesora, a nie zawsze taki przycisk znajduje się w uruchamianym układzie. Po zaprogramowaniu pamięci programu wejście RST, podobnie jak pozostałe wyprowadzenia procesora, jest przełączane na kabel emulacyjny (Z2). Jeśli z uruchamianego modułu przyjdzie na to wejście niski stan logiczny, procesor rozpocznie wykonywanie

wpisanego programu. Przyciskiem RESET można wymusić realizację programu od początku. Z przycisku ręcznego zerowania należy ostrożnie korzystać, jeśli uruchamiany system jest wyposażony

WYKAZ ELEMENTÓW

Rezystory

R1, R2, R3, R8: 10kΩ/0,25W
R4, R5, R6: 2,2kΩ/0,25W
R7: 330Ω/0,25W

Kondensatory

C1, C2, C3: 10μF/16V
C4, C5: 1μF/63V MKT
C6, C7: 22pF
C8, C9: 100nF/63V

Półprzewodniki

D1: 1N4148
DZ1: BZX83C6V8
LED1: L-934LYD (Kingbright) lub inna żółta, niskoprądowa
LED2, LED4: L-934LGD (Kingbright) lub inna zielona, niskoprądowa
LED3: L-934LID (Kingbright) lub inna czerwona, niskoprądowa
TO1: PC817
U1: AT89C2051 (Atmel) zaprogramowany
U2: MAX662 (Maxim), LTC1262 (Linear Technology), ST662A (STMicroelectronics)
U3: MAX333A (Maxim)
U4, U5, U6, U7, U8: HCT4053
U9: HCT02
Z1: AT89Cx051 (Atmel) typ procesora zależny od aplikacji

Różne

Gniazdo 2x10 na kabel taśmowy
Kabel taśmowy 20-żyłowy 20cm
*Kabel połączeniowy do komputera: gniazdo DB9 do połączeń lutowanych z obudową
Gniazdo do złącz igłowych 3x1 do połączeń lutowanych
*Kabel 2-żyłowy 2m
Kabel emulacyjny: wtyk emulacyjny na kabel taśmowy DIP20
Podstawki pod układy scalone: DIL20 - 2 szt., DIL8 - 1szt., DIL16 - 5 szt., DIL14 - 1 szt.
SW1: mikroswitch do druku H=10mm
Q1: rezonator kwarcowy 11,0592MHz
Z1: DIL20 precyzyjna lub ZIF (3M, Textool lub Aries)
Z2: goldpin 2x10
Z3: goldpin kątowy 1x3
*Elementy oznaczone * nie wchodzi w skład kitu.*

w układy nadzoru napięcia i watch-dog. Nie wszystkie tego typu układy zezwalają na zwieranie swojego wyjścia do napięcia zasilającego.

Do programatora dobudowano prosty, ale bardzo użyteczny wskaźnik zatrzymania procesora. Wskaźnik ten tworzą dwie bramki NOR z układu U9 i dioda LED4 zielona z rezystorem szeregowym R7. W czasie normalnej pracy procesora na wyjściu XTAL2 występuje przebieg sinusoidalny o amplitudzie bliskiej 5V i częstotliwości zależnej od dołączonego kwarcu. Przebieg ten po przejściu przez bramkę NOR (11,12,13) U9 (ukształtowany do przebiegu prostokątnego) zasilą diodę LED4, która świecąc sygnalizuje wykonywanie programu. Jeśli procesor natopka rozkaz ustawiający bit PD w rejestrze PCON, przejdzie w stan Power Down. W stanie tym wewnętrzny generator jest zatrzymany, a na wyjściu XTAL2, zgodnie ze schematem na rys. 3, występuje wysoki stan logiczny, który powoduje wygaszenie diody. Bramka (8,9,10) U9 blokuje wskaźnik na czas programowania pamięci FLASH.

Programista umieszczając w programie rozkaz:

```
ORL PCON,#02H ;w assemblerze
lub
```

```
PCON |= 0x02 /* w języku C!*/
```

może sprawdzić, czy procesor wykonując program doszedł do tego miejsca. Dodatkową zaletą jest, że procesor po takim zatrzymaniu zachowuje stan portów

wyjściowych. Wadą jest to, że aby ponownie uruchomić procesor, należy go wyzerować.

Ograniczenia

Bardzo oszczędna konstrukcja programatora narzuciła kilka ograniczeń. Przede wszystkim możliwy jest wybór tylko dwóch trybów programowania Chip Erase i Write Code Data. Zatem nie ma możliwości programowania bitów zabezpieczających LB1 i LB2 (Write Lock Bit 1/2), odczytania bajtu sygnatury (Read Signature Byte) czy nawet odczytania zapisanych bajtów programu (Read Code Data) w celu ich weryfikacji.

Zasilanie programatora z uruchamianego układu narzuca konieczność zaprojektowania zasilacza z niewielkim zapasem mocy. W rzadkich przypadkach, gdy to nie jest możliwe, należy zastosować dodatkową przejściówkę, która umożliwi zasilanie programatora z zewnętrznego zasilacza. Przejściówkę taką można zbudować z podstawki DIL20 nasadzonej na wtyk emulacyjny. Odcinek niebieskiego przewodu delikatnie dolutowujemy do pinu nr 10 podstawki. Drugi koniec przewodu podłączamy do masy zasilacza. Do pinu nr 20 dolutowujemy kabel czerwony i resztę tego pinu obcinamy. Czerwony przewód dołączamy do wyjścia +5V w zasilaczu. Tak przygotowany „wtyk“ wkładamy w podstawkę układu uruchamianego zwracając uwagę, aby skró-

Podstawowe cechy programatora

Zalety:

- ✓ komunikuje się z komputerem łączem szeregowym RS232C;
- ✓ zapewnia izolację galwaniczną między komputerem i uruchamianym układem;
- ✓ nie wymaga specjalnego oprogramowania - do obsługi wystarczają programy systemowe komputera;
- ✓ na czas programowania odłącza procesor od uruchamianego układu, po zaprogramowaniu przyłącza go z powrotem.

Wady:

- ✓ zapisywany program musi być w postaci binarnej;
- ✓ wymaga, aby pierwszy bajt programu miał wartość 02H;
- ✓ nie sprawdza, czy zapisywany program zmieści się w pamięci;
- ✓ nie weryfikuje poprawności zapisanego programu;
- ✓ nie ma możliwości programowania bitów zabezpieczających;
- ✓ jest zasilany z uruchamianego układu - pobiera około 16mA w stanie spoczynku i 26mA w czasie programowania.

Podstawowe właściwości emulatora

Zalety:

- ✓ pracuje z rezonatorem kwarcowym układu uruchamianego;
- ✓ w pełni emuluje wszystkie piny procesorów AT89Cx051 łącznie z analogowymi wejściami komparatora;
- ✓ jest wyposażony w przycisk RESET zerujący procesor;
- ✓ posiada optyczny wskaźnik pracy/zatrzymania procesora.

Wady:

- ✓ jest zasilany z uruchamianego układu - pobiera około 33mA przy 12MHz w trybie emulacji.

cony pin 20 „wtyku“ nie zetknął się z odpowiadającym mu stykiem w podstawce układu uruchamianego.

Tomasz Gumny, AVT