

Programator - emulator mikrokontrolerów AT89Cx051, część 2

kit AVT-498



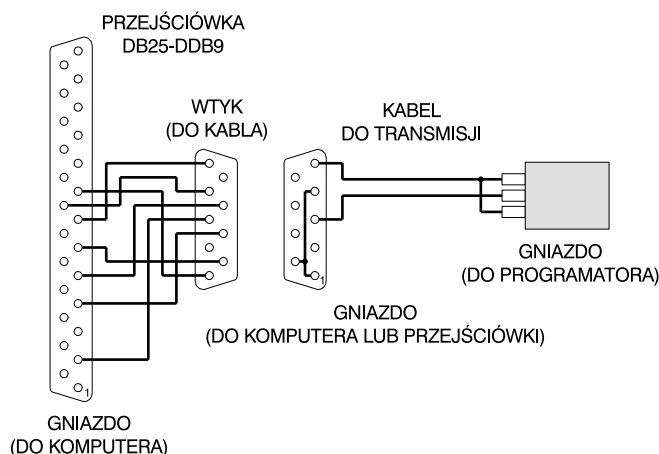
Oprogramowanie

Program dla procesora sterującego napisano w języku C (IAR Systems). W stanie spoczynkowym procesor oczekuje na bajty programu. Transmisja z komputera odbywa się z szybkością 4800 bodów. Słowo musi zawierać 8 bitów danych, bit parzystości (even) i dwa bity stopu. Ponieważ sterownik programatora nie ma możliwości zgromadzenia całego programu w pamięci buforowej, zastosowano inne rozwiązanie. Odebrane bajty są zapisywane w 50-elementowym buforze kołowym FIFO. Stąd, niemal na bieżąco, dane są przepisywane do pamięci FLASH. Transmisja programu rozpoczyna się od bajtu 02H. Jest to pierwszy bajt

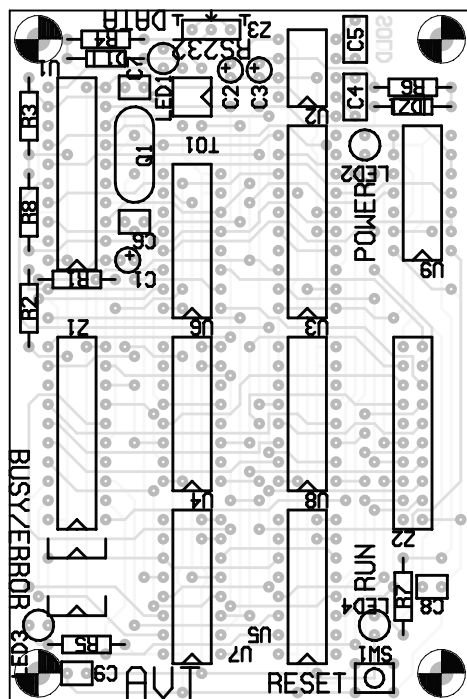
Kończymy opis konstrukcji programatora-emulatora procesorów AT89Cx051.

W tej części artykułu prezentujemy sposób podłączenia urządzenia do komputera oraz zalecenia do montażu i uruchomienia.

rozkazu LJMP, od którego musimy rozpocząć uruchamiany program. Poprawne odebranie takiego bajtu powoduje wykonanie sekwencji początkowej (Power-up Sequence). Sygnał INH przyjmuje wysoki, a PROG, XTAL1/U i ABC niski stan logiczny. Po 10ms jest ustawiane wyjście P3.2/U i kasowane wyjście INH. Kolejną czynnością jest wyczyszczenie całej pamięci FLASH. Wyjście P3.457/U jest kasowane, co powoduje następujące ustawienie wejść programowanego procesora: P3.3=H, P3.4=L, P3.5=L, P3.7=L. Sygnał PROG przyjmuje wysoki stan logiczny, co w połączeniu z zerem logicznym na wyjściu ABC powoduje podanie napięcia programującego +12V na wejście RST procesora. Wówczas na wejściu P3.2 pojawia się ujemny impuls o czasie trwania 10ms. Po tym czasie napięcie programujące jest wyłączane (PROG=0), a po dalszych 2ms procesor z wykasowaną pamięcią jest gotowy do programowania. Warto zwrócić uwagę, że przez cały ten czas (około 25ms) dane przychodzące z komputera są gromadzone w buforze FIFO. Po wyczyszczeniu pamięci procesora rozpoczyna się zapisywanie bajtów nowego programu. Wyjście P3.457 jest ustawiane, co powoduje pojawienie się na wejściach P3.4, P3.5, P3.7



Rys. 4. Sposób wykonania kabli.



Rys. 5. Rozmieszczenie elementów na płycie drukowanej.

stanu wysokiego, a na P3.3 stanu niskiego. Wyjście PROG również jest ustawiane, a na port P1 wystawiany jest zapisywany bajt. Krótki, ujemny impuls na wejściu P3.2 zezwala na wpisanie danej. Następnie wyłączane jest napięcie programujące (PROG=0). Ze względu na ograniczoną liczbę wejść, nie jest testowany pin P3.1 (READY/BUSY) programowanego procesora. Przyjęto maksymalny czas oczekiwania na zapisanie bajtu (Byte Write Cycle Time), gwarantowany przez producenta (2ms). Cykl zapisu jest pomijany, jeśli bajt do zapisania ma wartość FFh.

Ostatnią czynnością jest inkrementowanie licznika bajtów dodatnim impulsem na wejściu XTAL1. Cykl ten powtarza się dla każdego odebranego bajtu. Jeśli wystąpi błąd parzystości, programowanie zostaje natychmiast przerwane. Programator przechodzi w stan oczekiwania nie gasząc diody LED3 (BUSY/ERROR). Należy wówczas powtórzyć transmisję. Po sekundzie od poprawnego odebrania ostatniego bajtu następuje ustawienie sygnału ABC i przyłączenie procesora do układu uruchamianego oraz odblokowanie kluczy przez skasowanie sygnału INH. Wynika z tego, że między nadawanymi z komputera bajtami programu nie mogą wystąpić przerwy dłuższe od

jednej sekundy. Przyjęta szybkość transmisji danych z komputera została zdeterminowana czasem programowania jednego bajtu pamięci programu i wynosi 4800 bodów. Do połączenia programatora z komputerem należy wykonać dwużyłowy kabel zakończony z jednej strony wtykiem DB9 lub DB25. Na drugim końcu znajduje się trójpinowe gniazdo do złącz igłowych. Schemat takiego kabla przedstawiono na **rys. 4**.

Do transmisji zbioru z komputera do programatora najprościej napisać plik wsadowy o krótkiej nazwie np. P.BAT. Przy założeniu, że korzystamy z pierwszego portu szeregowego, plik taki powinien zawierać ustawienie parametrów transmisji:

```
@MODE COM1 4800 E|8|2
```

i polecenie przesłania binarnego pliku z programem na port szeregowy:

```
@COPY /B %1 COM1
```

Wywołanie w DOS-ie wygląda następująco:

P | PROGRAM.BIN

Jeśli skorzystamy z programu Norton Commander, możemy wykorzystując komendę Extension File Edit utworzyć w pliku EXT.MNU wiersz o następującej treści:

BIN P.BAT !:!\!.!

Dzięki temu każde „najechnanie“ na plik z rozszerzeniem .BIN i zatwierdzenie ENTER-em spowoduje przesłanie go do programatora.

W systemie WINDOWS⁹⁵ możemy również zastosować dwa rozwiązania. Pierwsze to nakazać systemowi otwieranie plików z rozszerzeniem *.BIN z programu P.BAT. Programowanie uruchamia się wówczas podwójnym kliknięciem na pliku binarnym z uruchamianym programem.

Drugie rozwiązanie polega na utworzeniu skrótu do pliku P.BAT. We Właściwościach skró-

tu na karcie Program należy ustawić „Uruchom: Zminimalizowane” i “Zamknij przy zakończeniu”. W tym przypadku wystarczy „przeciągnąć” plik z programem w postaci binarnej na ikonę skrótu do P.BAT i “upuścić”.

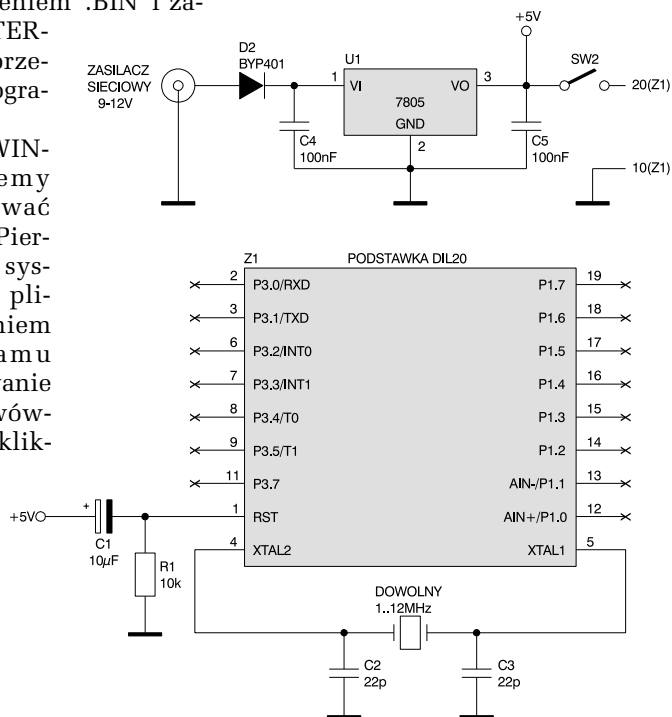
Zapewne w innych komputerach realizacja transmisji będzie równie prosta. W każdym przypadku przesłanie i wpisanie pliku o długości 2048 bajtów zajmuje niecałe 5s.

UWAGA: Programator nie sprawdza, czy liczba przesłanych bajtów wykracza poza pojemność pamięci procesora. Z jednej strony nakłada to na obsługującego programator obowiązek sprawdzenia, czy plik binarny nie jest za duży. Z drugiej strony programator umożliwia programowanie nowych procesorów, jak np. AT89C4051, wyposażonych w 4KB pamięci programu. A nie jest to z pewnością ostatnie słowo firmy Atmel...

Montaż

Uwaga: Przed rozpoczęciem montażu warto wyraźnie zaznaczyć mikroprocesor z zapisanym programem sterującym, aby nie pomylić go z procesorem przeznaczonym do programowania.

Wszystkie podzespoły programatora zamontowano na jednej, dwustronnej płycie drukowanej. Ze względu na znaczną liczbę



Rys. 6. Proponowany układ testowy.

otworów przejściowych, warto skorzystać z gotowej płytki z metalizowanymi otworami. Rozmieszczenie elementów przedstawiono na **rys. 5**. Montujemy kolejno: rezystory, diody, transoptor, podstawki pod układy scalone, kondensatory i złącza. Jako ostatnie wlotowujemy przycisk, rezonator kwarcowy i diody świeące. Pod rezonatorem kwarcowym po stronie elementów przebiegają ścieżki, dlatego wlotowując rezonator należy zastosować przekładkę z materiału izolacyjnego. Jeśli przewidujemy wykorzystanie układu jako zwykłego programatora, możemy zamiast zwykłej podstawki pod programowany procesor zamontować podstawkę typu ZIF (z dźwigienką). Najbardziej popularne podstawki ZIF firmy 3M/Textool mają dość szerokie wyprowadzenia i ich montaż na płytce może być nieco utrudniony. Znacznie tańsze, choć trudniejsze do zdobycia, są podstawki firmy Aries. Przewidując możliwość montażu podstawki Z1 typu ZIF, pozostawiono na płytce drukowanej nieco więcej wolnego miejsca oraz zdublowano opis sitodrukowy wskazujący sposób wkładania programowanego procesora.

Uruchomienie

Uruchomienie jak zawsze rozpoczynamy od sprawdzenia poprawności montażu. Należy zwrócić baczną uwagę, czy przy lutowaniu nie powstały zwarcia między polami lutowniczymi. Do uruchamiania będą niezbędne oba kable: emulacyjny i transmisyjny, tak więc teraz jest ostatni dzwonek, aby je wykonać. W pierwszym etapie wkładamy w podstawkę tylko układ przetwornicy U2. Po włączeniu napięcia zasilającego +5V z zewnętrznego zasilacza sprawdzamy, czy świeci dioda LED2 (POWER) i ewentualnie mierzymy napięcie na wyprowadzeniu 6(U2) lub na kondensatorze C3. Powinno ono wynosić od 11,4 do 12,6V. Upewniamy się, czy jest wlotowany transoptor TO1 i podłączamy programator do komputera za pomocą przygotowanego wcześniej kabla transmisyjnego. W stanie spoczynku dioda LED1 (DATA) powinna pozostać zgaszona. W czasie nadawania dowolnego pliku dioda zacznie nierównomiernie świecić. Jest

to prawidłowy objaw, bowiem LED1 pokazuje bezpośrednio stan na linii TxD łącza szeregowego, który podczas trwania transmisji się zmienia. Przy przesyłaniu krótkiego programu to świecenie diody może wyglądać jak krótkie, ledwie zauważalne błysnięcie. Jeśli wszystko przebiegło poprawnie, wyłączamy napięcie zasilające, wkładamy w podstawki pozostałe układy scalone, pozostawiając na razie pustą podstawkę Z1. W takim stanie przeprowadzamy ponownie transmisję dowolnego pliku. Pierwszy nadany bajt (musi to być liczba 02H) powinien zapalić diodę LED3 (BUSY/ERROR). Sekundę po nadaniu ostatniego bajtu dioda powinna zgasnąć. Jeśli tak się nie stało, to znaczy, że wystąpił błąd transmisji i musimy ją powtórzyć. Następnie w podstawkę Z1 wkładamy procesor przeznaczony do zaprogramowania. Przykładowy program testujący TEST.ASM może mieć postać:

```
0000  ORG          0
0000  02 00 03    LJMP
0003
0003  43 87 02    ORL PCON,#02
```

Początkowy skok jest wymagany przez odbiornik szeregowy, bowiem jak wspomniano wcześniej, pierwszy odebrany bajt musi mieć wartość 02H. Dla sprawdzenia poprawności działania programatora niezbędny jest układ, który zapewni zasilanie, taktowanie i poprawny sygnał zerowania. Schemat takiego minimalnego układu przedstawiono na **rys. 6**. Wyłącznik SW2 służy do wyłączenia napięcia zasilającego przed wyciągnięciem programowanego procesora z podstawki. Oczywiście programator można również sprawdzić w dowolnym innym układzie, który spełnia trzy wymienione wcześniej warunki. W podstawkę takiego układu wkładamy wtyk emulacyjny. Po załączeniu zasilania programujemy procesor programem testującym w postaci binarnej TEST.BIN. Chwilę po zakończeniu programowania powinny zgasnąć diody BUSY/ERROR i RUN. Każdorazowe naciśnięcie przycisku RESET musi spowodować zapalenie diody RUN. Dzieje się tak dlatego, że zegar procesora startuje już przy aktywnym stanie na wejściu

RST. Natychmiast po zwolnieniu przycisku dioda zgaśnie, gdyż działanie programu testowego polega na wprowadzeniu procesora w stan Power Down. W tym stanie zegar procesora jest zatrzymany i dioda LED4 (RUN) gaśnie.

Jeśli wszystko przebiegło pomyślnie, to właśnie staliśmy się posiadaczami bardzo przydatnego urządzenia, które zaoszczędzi nam masę czasu zwykle traczonego na wyciąganie procesora, programowanie i ponowne wkładanie go z powrotem do podstawki. Nie mówiąc już o korzyściach, jakie daje możliwość skupienia się na uruchamianym programie bez rozpraszania uwagi na innych prozaicznych czynnościach. Aby jednak programator stał się w pełni przyrządem warsztatowym, należałoby go jeszcze - jak to określił jeden z uczestników internetowej Listy Dyskusyjnej EP - „ubrać w jakieś szatki”.

Tomasz Gumny, AVT