

# Podstawy projektowania systemów mikroprocesorowych, część 1

Dlaczego '51 i dlaczego Atmel? Odpowiedź jest prosta: ze względu na wbudowaną pamięć Flash, niską cenę i łatwą dostępność są to mikrokontrolery bardzo popularne, a ze względu na dużą popularność łatwo jest znaleźć tanie, często bezpłatne, programy narzędziowe. Umożliwia to amatorom-elektronikom realizację kompleksowych projektów bez ponoszenia jakichkolwiek kosztów. Co więcej, gwarantowana przez producenta minimalna liczba cykli zapisu przekracza 1000, co w zupełności wystarcza do przygotowania, przetestowania i wprowadzenia poprawek do dowolnie skomplikowanego programu - nie ma więc konieczności kupowania emulatora sprzętowego.

Zapewne niektórzy Czytelnicy stwierdzą, że są lepsze, szybsze, nowocześniejsze konstrukcje (np. rodzina AVR Atmela, PIC-e, czy inne) i będą mieli trochę racji, jednak w większości typowych zastosowań nie jest potrzebna ani zawrotna moc obliczeniowa, ani rozbudowane peryferie wbudowane w mikrokontroler - potrzebny jest tani układ, łatwy do kupienia w sklepie elektronicznym w małym mieście, dobrze opisany i prosty w programowaniu.

Na rynku wydawniczym, w czasopismach i w publikacjach elektronicznych jest dostępnych wiele informacji na temat mikrokontrolerów rodziny '51, ich parametrów, budowy wewnętrznej, działania układów peryferyjnych i zasad programowania. Brakuje pozycji w przystępny sposób przekazującej praktyczne informacje dotyczące projektowania układów elektronicznych wykorzystujących mikrokontrolery. Choć w tym artykule i cyklu następnych skoncentrujemy się na najprostszymi mikrokontrolerach rodziny '51 firmy Atmel (AT89Cx051, AT89C51, AT89C52), to podane tutaj informacje przydadzą się również (poza wykorzystaniem specyficznych cech wymienionych procesorów) przy realizacji projektów, w których stosujemy procesor z rodziny AVR, PIC, układy Motorola czy innych producentów. Pokażemy najprostszą drogę prowadzącą do realizacji konkretnego celu, przedstawimy najczęściej popełniane błędy i sposoby ich unikania, omówimy współpracę mik-

**Rozpoczynamy cykl artykułów, w których zostaną przedstawione najważniejsze zagadnienia związane z projektowaniem systemów mikroprocesorowych. Na modelowy mikrokontroler wybraliśmy atmelowskie wersje 8051, które cieszą się bardzo dużym powodzeniem wśród projektantów w naszym kraju.**

rokontrolera z układami analogowymi, ilustrując wszystko schematami i w razie potrzeby przykładowymi procedurami w assemblerze.

W realiach obecnej elektroniki, zwłaszcza cyfrowej, bez mikroprocesorów ani rusz, dlatego najwyższa pora, aby zapoznać się z niezbyt trudną sztuką praktycznego projektowania urządzeń wykorzystujących mikrokontrolery, która otwiera drogę nowym zastosowaniom niedostępnym lub trudnym w realizacji przy użyciu „zwykłych” układów cyfrowych TTL czy CMOS i pozwala na realizację elastycznych rozwiązań dzięki połączeniu i współpracy układów sprzętowych i programu. Zatem zaczynamy...

## Bez prądu ani rusz, czyli zasilanie mikrokontrolera

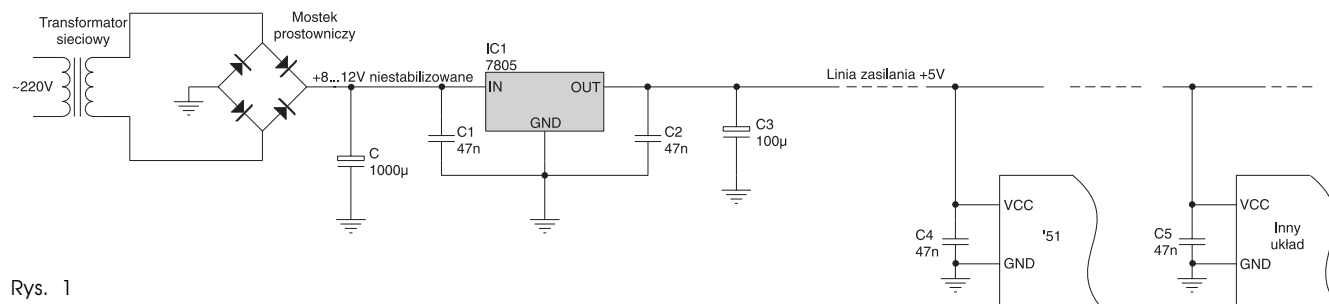
Ważną, lecz niejednokrotnie lekceważoną sprawą jest zasilanie mikrokontrolera. Często właśnie niefortunne rozwiązanie obwodów zasilania jest przyczyną niestabilnej pracy, objawiającej się niekontrolowanym zerowaniem układu lub generacją zakłóceń na szynach zasilania i wpływających na pracę innych podzespołów współpracujących z mikrokontrolerem.

Mikrokontrolery AT89C5x mogą być zasilane napięciem  $5\text{ V} \pm 20\%$ , natomiast układy AT89Cx051 umożliwiają zasilanie ze źródła o napięciu od 2,7 do 6 V. Najbardziej popularnym rozwiązaniem jest zasilanie mikrokontrolera napięciem 5 V i zastosowanie stabilizatora scalonego w układzie jak na rys. 1. Należy pamiętać, że układy stabilizatorów serii 78xx potrzebują do prawidłowej pracy napięcia wejściowego o około 3 V wyższego niż napięcie wyjściowe, co dla układu 7805 daje minimalną wartość napięcia wejściowego około 8 V. Jeżeli napięcie na wejściu będzie niższe, spowoduje to

także spadek napięcia wyjściowego, zmniejszy się tłumienie tętnień zasilania i stabilizator - ogólnie mówiąc - przestanie pełnić swoją funkcję (choć mikrokontroler i układy współpracujące ciągle mogą działać - nie będzie to jednak praca pewna). Kondensatory C1 i C2 powinny być umieszczone jak najbliżej wyprowadzeń stabilizatora - zapobiegają one jego wzbudzeniu. W pobliżu wyprowadzeń zasilania mikrokontrolera, jak i każdego innego układu cyfrowego, muszą być umieszczone kondensatory blokujące zasilanie (C4, C5) zapobiegające zakłócającemu działaniu układów na szynę zasilania i zabezpieczające inne elementy przed zakłóceniami przenoszonymi tą szyną. Pojemność kondensatora C3 nie powinna być zbyt duża (dla systemów pobierających kilkanaście...kilkadziesiąt mA kondensator C3 jest zbyt duży), dużo ważniejsza jest wartość pojemności C przed stabilizatorem, dająca odpowiedni zapas energii redukujący tętnienia sieci.

Ze stabilizatora 7805 można uzyskać prąd o maksymalnym natężeniu 1 A, co jest wartością całkowicie wystarczającą dla większości aplikacji. Ważną sprawą jest moc tracona w stabilizatorze, która przy napięciu zasilającym rzędu 12 V i wyższym (max. 24 V), zmusza nas do stosowania radiatora, czasem o dość pokaźnych rozmiarach.

Pobór prądu przez mikrokontroler jest zależny od napięcia zasilania oraz od częstotliwości taktowania. Producent podaje pobór prądu dla częstotliwości oscylatora 12 MHz, który dla układów 89C5x wynosi 25 mA, a dla układów 89Cx051 wynosi 5,5 mA przy zasilaniu 3 V i 15 mA przy 6 V. Z tego powodu „małe” Atmele są bardzo atrakcyjną propozycją dla aplikacji z zasilaniem bate-



Rys. 1

List. 1.

```

ORG 0000H
LJMP POCZATEK

;(procedury obsługi przerw)

POCZATEK:
CLR P1.3      ;włączenie podtrzymania zasilania

MOV A,P1     ;odczyt stanu klawiszy
ANL A,#0F0h  ;wyzerowanie niepotrzebnych bitów
SWAP A      ;zamiana części bajtu
MOV R7,A    ;otrzymujemy w rejestrze R7 stan klawiszy:
            ;bit 0 - stan klawisza P1 (0-wciśnięty)
            ;bit 1 - stan klawisza P2 (0-wciśnięty)
            ;bit 2 - stan klawisza P3 (0-wciśnięty)
            ;bit 3 - stan klawisza P4 (0-wciśnięty)

;(inicjalizacja liczników, zerowanie zmiennych, itp.)

;(właściwa część programu)

SETB P1.3   ;wyłączenie podtrzymania po wykonaniu programu
SJMP $      ;pozostań w pętli (ewentualne oczekiwanie na puszczenie
            ;długo trzymanego klawisza, mimo wyłączenia tranzystorów
            ;dopiero po jego puszczeniu wyłącz się zasilanie)
    
```

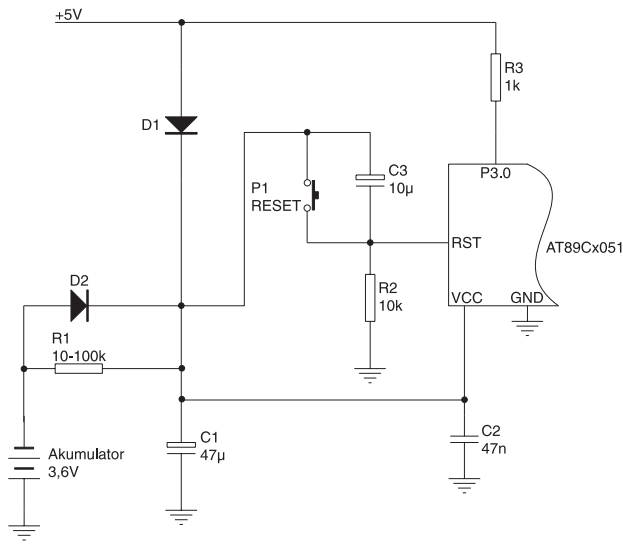
ryjnym lub wymagających baterijnego zasilania awaryjnego. Na rys. 2 przedstawiono jeden z wielu możliwych wariantów rezerwowego zasilania mikrokontrolera. Jako źródło rezerwowe zastosowano akumulator 3,6 V. Dobrze w tej roli sprawdzają się akumulatory NiCd 3,6 V/60 mAh, stosowane dawniej w płytach głównych komputerów PC. Można również zastosować akumulatory wykorzystywane w słuchawkach telefonów bezprzewodowych, posiadają one większą pojemność, lecz są droższe. Dioda D2 powinna być diodą Schottky'ego, ze względu na niższy niż w standardowych diodach spadek napięcia na złączu, D1 może być zwykłą diodą krzemową - stanowi ona zabezpieczenie przed zasilaniem całego urządzenia ze źródła rezerwowego. Rezystor R1 dobieramy w celu uzyskania pożądanego prądu ładowania akumulatora - należy przy tym pamiętać, że akumulator będzie doładowywany przez cały czas pracy zasilania głównego, więc należy ograniczyć prąd ładowania do wartości dla niego bezpiecznej. Dla rezystora R1

o wartości 47 kΩ maksymalny prąd ładowania przy rozładowanym akumulatorze osiąga 0,05 mA, co daje nam ponad miesiąc czasu potrzebny do pełnego naładowania. Taka wartość prądu nie zagraża nawet w pełni naładowanemu akumulatorowi. Jeżeli przewiduje się częstsze przerwy w zasilaniu (ze względu np. na przenoszenie urządzenia), warto zastosować większe prądy ładowania, niezbędne do uzupełniania zużytej energii. Wówczas dioda D1 powinna być diodą krzemową, co przy zasilaniu 5 V ograniczy maksymalne napięcie na akumulatorze do bezpiecznej wartości około 4,4 V bez konieczności ograniczania prądu ładowania, gdy akumulator zostanie w pełni naładowany.

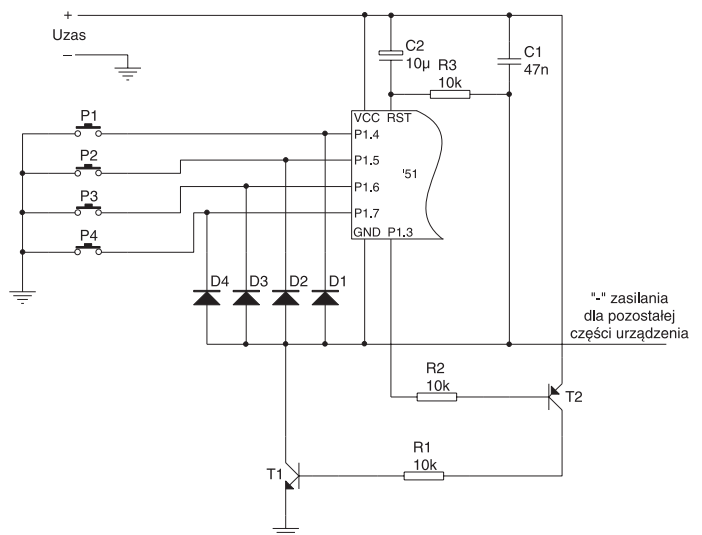
Zastosowany kondensator C2 blokuje zasilanie, natomiast C1 jest odpowiedzialny za stabilną pracę w momentach wyłączania, a zwłaszcza włączania zasilania głównego. Kondensator C1 powinien mieć pojemność kilka razy większą niż kondensator obwodu zerowania mikrokontrolera (C3, R2). Spowodowane jest to tym, że przy braku C1 po pojawie-

niu się napięcia głównego nastąpi doładowanie kondensatora C3 od napięcia zapewnianego przez obwód rezerwowy do wartości 4,4...4,7 V (w zależności od typu D1). Przy niskim napięciu akumulatora rezerwowego istnieje możliwość, że przepływ prądu przez obwód C3 i R2 (oraz wewnętrzny rezystor mikroprocesora) wywoła spadek napięcia na R2, który może wyzerować mikrokontroler. Obecność C1 oraz rezystancja dynamiczna diody D1 ograniczają prędkość narastania napięcia, a co za tym idzie wartość prądu doładowującego C2, zatem napięcie na R2 nie osiągnie wartości grożącej wyzerowaniem mikrokontrolera. Opisany układ nie jest zalecany do pracy z mikrokontrolerami AT89C5x, ponieważ w przypadku tych układów należałoby zastosować akumulator o napięciu 4,8 V - pojemność akumulatora 3,6 V nigdy nie będzie całkowicie wykorzystana, gdyż „duże” Atmele nie działają przy tak niskich napięciach jak „małe”.

W praktyce często wykorzystywane są urządzenia, które zasilane powinny być tylko w chwili zaistnienia odpowiedniego zdarzenia zewnętrznego - najczęściej wciśnięcia przycisku (np. nadajniki zdalnego sterowania, itp.), gdy przez pozostały czas układ pozostaje bezczynny, co przy zasilaniu baterijnym jest czystym marnotrawstwem. Na rys. 3 pokazano fragment schematu przykładowego urządzenia, którego zasilanie jest włączane tylko na czas obsługi wciśnięcia przycisku. Wciśnięcie któregośkolwiek z przycisków powoduje dołączenie ujemnego bieguna zasilania do masy urządzenia. Po rozpoczęciu pracy przez mikrokontroler, pierwszą instrukcją jest wyzerowanie linii portu P1.3, co powoduje włączenie tranzystora T2 i podanie napięcia na bazę T1. Następuje włączenie T1 i podtrzymanie zasilania mimo zwolnienia naciśniętego przycisku. Po wykonaniu programu odpowiedzialnego za obsługę naciśnięcia konkretnego przycisku mikroprocesor ustawia linię P1.3, powodując zatkanie T2 i T1, i odłączenie zasilania. Obwód zasilania mikrokontrolera



Rys. 2



Rys. 3

i pozostałych układów nie powinien posiadać filtrujących kondensatorów elektrolitycznych o pojemnościach powyżej 10  $\mu\text{F}$ , gdyż wydłużą one czas włączenia i mogą spowodować trudności z wyłączeniem (układ może się „zatrzasnąć“ ze względu na niestabilną pracę mikrokontrolera przy powolnym opadaniu napięcia zasilania). Pewność działania układu zwiększy dodanie dodatkowego obciążenia w postaci rezystora (np. 330  $\Omega$ ) podłączonego między szynami Vcc i GND mikrokontrolera - spowoduje to jednak wzrost poboru prądu podczas działania urządzenia. Wadą przedstawionego rozwiązania jest konieczność przytrzymania przycisku tak długo, jak trwa czas zerowania mikroprocesora, plus czas potrzebny na wykonanie rozkazów odpowiedzialnych za włączenie podtrzymania i odczyt wyprowadzeń poszczególnych klawiszy (określenie, który klawisz został naciśnięty) - dlatego te części programu powinny być jednymi z pierwszych rozkazów. Można to zrealizować w sposób pokazany na **list. 1**. Niewątpliwą zaletą tej metody jest praktycznie zerowy pobór prądu, gdy układ jest nieaktywny.

Zmniejszanie poboru mocy można dokonać na drodze zmniejszenia częstotliwości taktowania, która niemal linowo wpływa na pobór prądu, duży zysk

energii osiągniemy także, obniżając napięcie zasilania. Jest to jednak możliwe tylko w układach AT89C51, ze względu na szeroki zakres napięć zasilających - układy AT89C5x przestają poprawnie pracować już przy napięciu zasilania rzędu 3,5 V. Niektóre egzemplarze układów AT89C51 pracują poprawnie już od napięcia rzędu 1,8...1,9 V, co z powodzeniem pozwala na zasilanie ich z dwóch akumulatorów Ni-Cd (2,4 V) lub dwóch zwykłych ogniów cynkowych czy alkalicznych (3 V).

Jeszcze większe obniżenie napięcia zasilania możliwe jest po wprowadzeniu mikrokontrolera w tryb *Power Down* (przez ustawienie bitu PD rejestru PCON), co pozwala na obniżenie napięcia zasilania zarówno „dużych“, jak i „małych“ Atmeli do wartości 2 V (gwarantowane przez producenta), a praktycznie nawet do 1,2...1,3 V. Niestety procesor w takim stanie nie pracuje, podtrzymywana jest jedynie zawartość wewnętrznej pamięci danych i rejestrów SFR. Powrót do normalnej pracy możliwy jest tylko przez generację sygnału RESET (po uprzednim przywróceniu pełnego napięcia zasilania). Wejście w stan *Power Down* powinno nastąpić zatem przed zanikiem napięcia zasilania. Realizacja układu wykrywania zaniku głównego napięcia zasilania jest bardzo pro-

sta - wystarczy dołączenie któregoś z wyprowadzeń dowolnego portu mikrokontrolera do linii zasilania głównego przez rezystor rzędu 1 do 10 k $\Omega$  (na rys. 2 - rezystor R3). W chwili zaniku napięcia głównego na linii zasilania pojawi się potencjał bliski masy (ze względu na dołączone inne odbiorniki), co spowoduje wprowadzenie linii portu w stan niski - wykrycie stanu niskiego oznacza brak zasilania i konieczność podjęcia określonych działań, np. wejście w stan *Power Down*. Wyjście z tego stanu jest możliwe wyłącznie po naciśnięciu przycisku P1 (rys. 2.). Możliwe jest również wejście w tryb pracy *Idle* (uśpienie), który również zatrzymuje pracę mikrokontrolera, działają jednak przerwanie i generator sygnału zegarowego. Wyjście z trybu *Idle* następuje w momencie nadejścia zgłoszenia dowolnego przerwania (lub sygnału RESET) - umożliwia to realizację wielu zadań, np. realizację zegara czasu rzeczywistego pobierającego z zasilania rezerwowego prąd dużo mniejszy niż przy normalnej pracy. Możliwy jest też powrót do normalnej pracy przy dołączeniu do linii przerwań zewnętrznych obwodu wykrywającego powrót głównego napięcia zasilania (generującego na wyjściu stan 0 po pojawieniu się napięcia).

**Paweł Hadam**