

ARM-y w praktyce

Programowanie ISP mikrokontrolerów

LPC2000 i AT91SAM7S

Mikrokontrolery z rdzeniem ARM weszły już wśród konstruktorów w modę, ale dynamiczny rozwój popularnych aplikacji jest utrudniony przez nie zawsze dobrze przygotowaną (przez producentów) dokumentację.

O co mi chodzi? Otóż o rzecz zupełnie podstawową: programowanie pamięci mikrokontrolerów.

Największą obecnie popularnością wśród mikrokontrolerów z rdzeniem ARM7TDMI cieszą się dwie rodziny:

- produkowane przez firmę Philips układy LPC2000,
- produkowane przez Atmela układy AT91SAM7S.

Ponadto wielu innych producentów oferuje mikroprocesory i mikrokontrolery z rdzeniami z rodziny ARM7, ale z różnych przyczyn (chodzi przede wszystkim o łatwą dostępność pojedynczych egzemplarzy lub niewielkich partii mikrokontrolerów, a także ceny narzędzi) ich przyjęcie na rynku popularnym jest nieco słabsze.

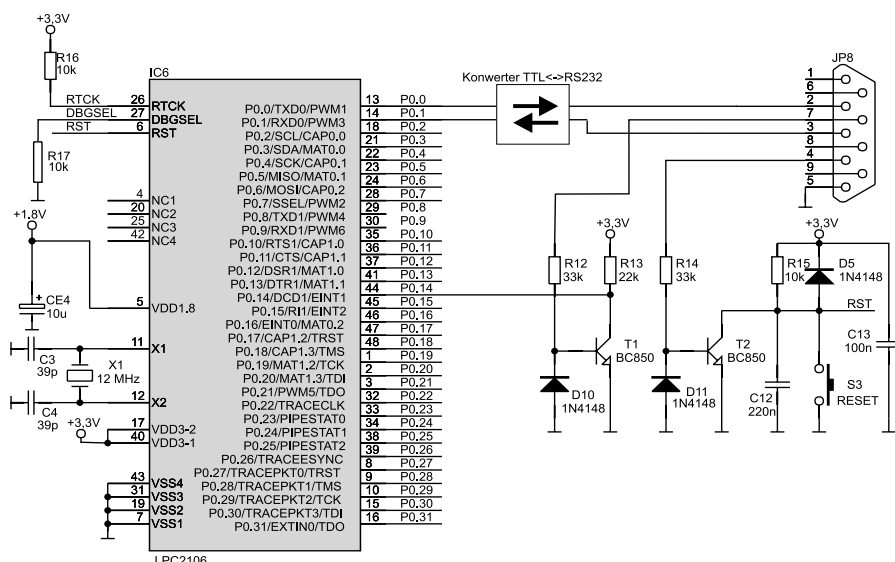
Pomimo dość krótkiej (najstarsze mikrokontrolery z rodziny LPC2000 mają około dwóch lat) obecności na rynku tanich mikrokontrolerów (czyli zintegrowanych w jednej strukturze: rdzenia, pamięci programu i danych oraz różnego rodzaju bloki peryferyjne, interfejsy itp.) z rdzeniem ARM7, ich producenci bardzo szybko wprowadzają mniejsze i większe modyfikacje, które wynikają m.in. z potrzeb



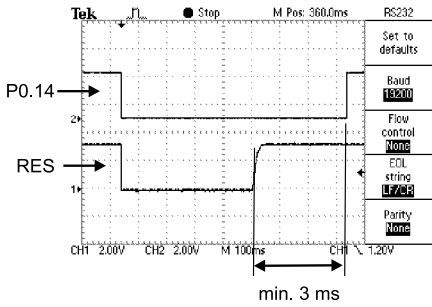
zgłaszanych przez użytkowników. Jednym z najważniejszych (finansowo) problemów, na jaki napotykali użytkownicy pierwszych wersji mikrokontrolerów ARM było ich zorientowanie na programowanie pamięci programu za pomocą interfejsu JTAG. Jest to bez wątpienia dobry sposób zwłaszcza, że JTAG jest wykorzystywany także do konfiguracji sprzętowego debuggera wbudowanego w rdzeń mikrokontrolera. Kolejnymi niebagatelnymi zaletami korzystania z JTAG-a jest duża szybkość programowania i prosta budowa podstawowego interfejsu, który pośredniczy pomiędzy mikrokontrolerem i portem Centronics

komputera. Najpopularniejszy (i łatwo dostępny, zwłaszcza w postaci „składowanej”) jest interfejs opracowany przez firmę Macraigor sterowany przez specjalny program OCDemon. O ile zastosowane przez producenta rozwiązania sprzętowe są powszechnie znane, to – niestety – wymiana danych pomiędzy PC i interfejsem JTAG wbudowanym w mikrokontroler owiewa mgła tajemnicy. Opis protokołu wymiany danych z interfejsem umożliwiającym na przykład programowanie pamięci Flash jest niedostępny (oficjalnie) u któregośkolwiek z producentów mikrokontrolerów. Pewnego wylomu dokonał At-

JTAG w ARM
Interfejs JTAG w mikrokontrolerach z rdzeniem ARM7TDMI umożliwia komunikację ze sprzętową jednostką wspomagającą debugowanie programów, pozwala także na programowanie wbudowanej w mikrokontroler pamięci Flash – jej obsługa **wybiega poza standardowe** polecenia ujęte w dokumentacji JTAG. Niestety, kody tych poleceń zostały utajnione przez producentów mikrokontrolerów.



Rys. 1. Podstawowy schemat aplikacji mikrokontrolera LPC2106 z programatorem ISP



Rys. 2. Przebiegi inicjujące tryb ISP w mikrokontrolerach LPC2100

mel, ale tak naprawdę kompletne informacje o sposobie obsługi JTAG-a (poza standardowymi funkcjami BST – Boundary Scan Testing) nie są dostępne. Taka polityka producentów pozwala żądać niektórym firmom za proste oprogramowanie do programowania pamięci Flash via JTAG 500...

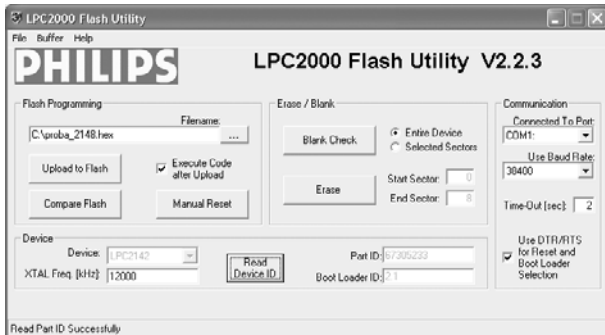
750 USD. Z tego powodu w artykule skupimy się na mniej wygodnych sposobach programowania pamięci Flash w mikrokontrolerach z rdzeniem ARM, ale za to takich, które są dostępne dla każdego.

Jeśli nie JTAG, to co?

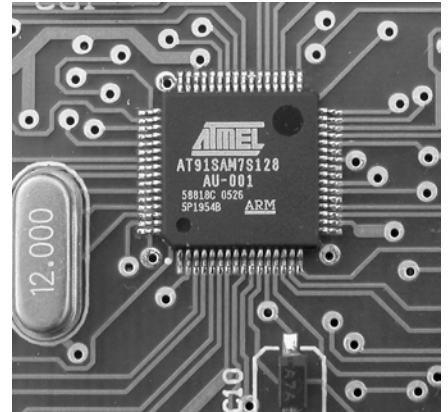
Ponieważ dostęp do Flasha za pomocą JTAG-a jest dość kosztowny (co wynika z wysokich cen prostych programów sterujących, jak wspomniano wcześniej), skupimy się na alternatywnym kanale dostępu do pamięci Flash, który jest udostępniany przez wpisany na stałe do pamięci mikrokontrolerów (opracowany oczywiście przez producentów mikrokontrolerów) program zwany najczęściej (choć nie do końca słusznie) *bootloaderem*.

Rodzina LPC2000

Zacznijmy od przybliżenia sposobu programowania pamięci Flash w mikrokontrolerach z rodziny LPC2000, na przykładzie dwóch mikrokontrolerów: LPC2106 (w „małej” obudowie VQFP48) oraz LPC2148 (w „dużej” obudowie VQFP64). Przedstawio-



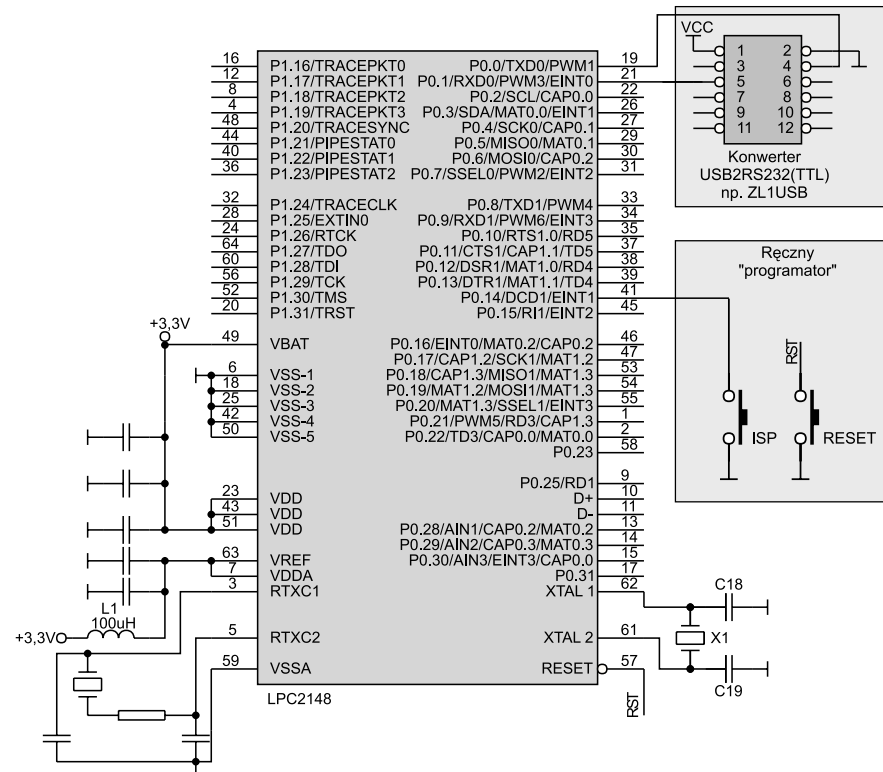
Rys. 3. Widok okna programu LPC2000 Flash Utility



Fot. 5. Wygodne w stosowaniu są nowsze wersje mikrokontrolerów AT91SAM7S128

ny sposób programowania jest charakterystyczny dla wszystkich mikrokontrolerów z tej rodziny wyposażonych we wbudowaną pamięć Flash.

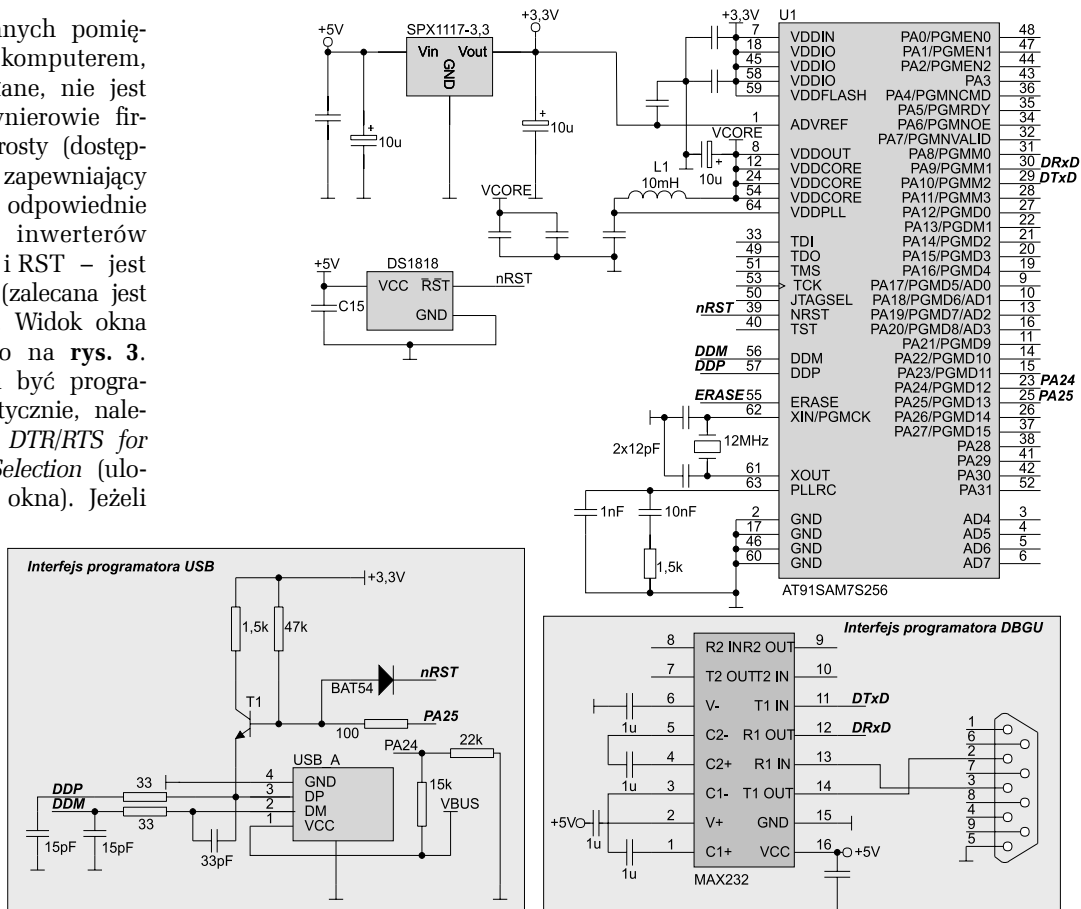
Na rys. 1 pokazano podstawowy schemat aplikacyjny mikrokontrolera LPC210x (szczegółowo opisany w AN10302), do którego dołączono dodatkowe elementy umożliwiające programowanie jego pamięci Flash w systemie (ISP – In System Programming). Dzięki zastosowaniu tych elementów programowanie można całkowicie zautomatyzować, minimalizując jednocześnie zaangażowanie użytkownika. Tranzystor T1, z kolektora którego jest sterowana linia P0.14, jest sterowany sygnałem DTR interfejsu RS232. Z kolei inwerter z tranzystorem T2, sterowany sygnałem RTS z RS232, służy do sterowania linii zerującej mikrokontrolera. Przełączenie mikrokontrolera z rodziny LPC2000 w tryb programowania ISP jest możliwe po podaniu na wejścia P0.14 i RST sekwencji sygnałów jak na rys. 2. Po przełączeniu mikrokontrolera w tryb ISP *bootloader* wykonuje procedurę automatycznego wykrycia prędkości transmisji via RS232 (testowany jest odbiór sekwencji bajtów o wartościach 0xAA oraz 0x55) i przechodzi w tryb oczekiwania na dane, które są przesyłane poprzez UART0. Jakkolwiek poprawną transmisję można uzyskać dla większości częstotliwości taktujących, to zastosowanie rezonatora o częstotliwości zoptymalizowanej pod kątem interfejsu UART (np. 11,0592 lub 14,7456 MHz) pozwala uzyskać większe maksymalne prędkości transmisji (do 115 kb/s), a także większą liczbę prędkości obsługiwanych poprawnie.



Rys. 4. Schemat aplikacyjny mikrokontrolera LPC2148 z możliwością programowania ISP poprzez zewnętrzny interfejs USB

Protokół wymiany danych pomiędzy mikrokontrolerem i komputerem, z którego są one przesyłane, nie jest skomplikowany, ale inżynierowie firmy Philips opracowali prosty (dostępny bezpłatnie!) program zapewniający jego pełną obsługę oraz odpowiednie sterowanie (via RS232) inwerterów sterujących linie P0.14 i RST – jest to *LPC2000 Flash Utility* (zalecana jest wersja 2.2.3 lub nowsza). Widok okna tego programu pokazano na rys. 3. Jeżeli mikrokontroler ma być programowany w pełni automatycznie, należy zaznaczyć opcję *Use DTR/RTS for Reset and Boot Loader Selection* (ulożona w prawej części okna). Jeżeli (co wygodne w praktyce) po zaprogramowaniu pamięci mikrokontroler ma rozpocząć pracę, należy zaznaczyć opcję *Execute Code after Upload* (umieszczoną w lewej części okna).

W przypadku, gdy w otoczeniu mikrokontrolera nie zastosowano elementów pomocniczych, jak to pokazano na rys. 1, można przełączyć go w tryb



Rys. 6. Schemat aplikacyjny mikrokontrolera AT91SAM7S256 z interfejsem DBGU i USB

programowania ręcznie. Wystarczy bowiem wymusić logiczne „0” na linii P0.14 i wyzerować mikrokontroler (wymuszenie na P0.14 po zerowaniu musi być utrzymane przez co najmniej 3 ms, później nie ma znaczenia). Od tego momentu, aż do kolejnego zerowania, pracą mikrokontrolera zawiaduje *bootloader*.

Interesującą możliwością programowania pamięci Flash mikrokontrolerów z rodziny LPC2000 jest zastosowanie interfejsu USB2RS232 (na przykład interfejsy ZL2USB, ZL1USB lub zbliżony do niego *FT232BM-module*) w torze UART0, jak pokazano na rys. 4. Sposób programowania mikrokontrolera nie ulega zmianie

w stosunku do przedstawionego opisu, zmienia się jedynie interfejs komunikacyjny: zamiast standardowego RS232 wykorzystywany jest interfejs USB emulujący połączenie RS232.

Rodzina AT91SAM7S

Inaczej rozwiązała sposób programowania mikrokontrolerów AT91SAM7S firma Atmel. Pierwotnie mikrokontrolery z tej rodziny mogły być programowane wyłącznie za pomocą interfejsu JTAG, ale problemy przedstawione na wstępie artykułu najwyraźniej zachęciły producenta do dołożenia większych starań o drobnych klientów, którzy nie chcą lub

nie mogą inwestować w kosztowne narzędzia. Programowanie ISP jest więc możliwe wyłącznie w przypadku najnowszych mikrokontrolerów, wyposażonych w *bootloader* SAM-BA (*SAM Boot Assistant*), które oznaczono:

- mikrokontrolery AT91SAM7S32/64 - rev. E (product ID: 58814E),
- mikrokontrolery AT91SAM7S128/256 - rev. C (product ID: 58818C - fot. 5).

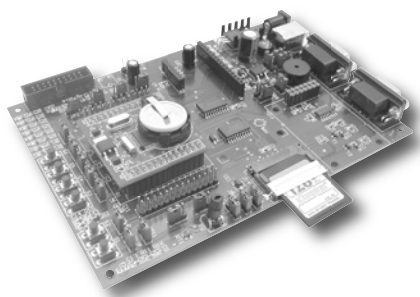
Wymienione mikrokontrolery mają wbudowaną w sterowniku wewnętrznej pamięci Flash dodatkową pamięć nieulotną (poza dostępnym obszarem adresowym), w której znajdują się procedury obsługi FFPI (*Fast Flash Programming Interface*) oraz SAM-BA. Procedury SAM-BA umożliwiają m.in. ładowanie danych z pamięci Flash (co wymaga zastosowania specjalnego programu na PC - SAM-Prog) oraz odtwarzanie *bootloadera* (czyli SAM-BA), który jest kopiowany do pierwszych dwóch sektorów pamięci Flash (jest to tzw. *Boot Recovery*). Oprogramowanie SAM-Prog jest udostępniane bezpłatnie.

W odróżnieniu od mikrokontroler-

Zestawy uruchomieniowe ARM

W ofercie handlowej AVT są dostępne zestawy uruchomieniowe dla mikrokontrolerów AVR, w tym ZL9ARM (dla mikrokontrolerów LPC213x oraz LPC214x montowanych na modułach *dipARM*) oraz ZL11ARM (dla mikrokontrolerów AT91SAM7S64...256 montowanych na modułach *dipARM*).

Uwaga! Zestawy ZL9ARM i ZL11ARM zamówione w Dziale Handlowym AVT do 31 stycznia 2006 będą wyposażone bezpłatnie w interfejsy JTAG ZL14PRG.
Zamówienia: www.sklep.avt.pl



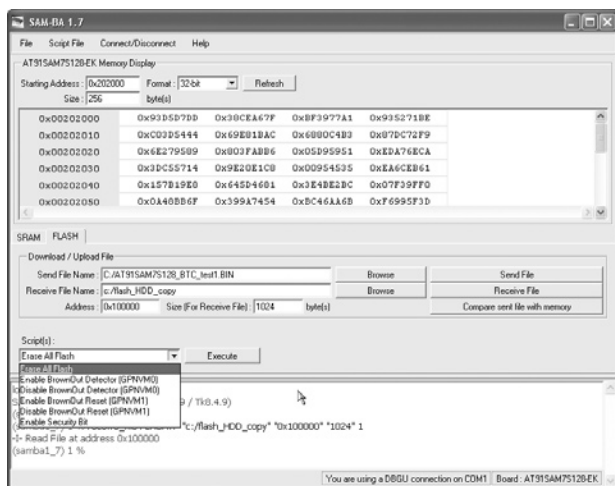
rów z rodziny LPC2000, mikrokontrolery AT91SAM7S do programowania wykorzystują specjalny interfejs szeregowy nazywany DBGU (*DeBuG Unit*) lub wbudowany interfejs USB. Na rys. 6 pokazano podstawowy schemat aplikacyjny mikrokontrolera AT91SAM7S128 z dwoma interfejsami umożliwiającymi programowanie ISP pamięci Flash.

W przypadku korzystania z DBGU częstotliwości rezonansowe zastosowanych kwarców mogą wynosić 3... 20 MHz, ponieważ program SAM-BA wyosażono w autodetekcję prędkości transmisji. W przypadku korzystania z USB częstotliwość rezonansowa kwarcu musi być równa 18,432 MHz, ponadto na potrzeby interfejsu (poza liniami D+ i D-) zajęte muszą być dwie linie I/O (jedna do kontroli pojawienia się napięcia V_{BUS} na złączu USB, druga do włączania rezystora pull-up USB). Należy pamiętać, że sterownik dla interfejsu USB (o nazwie *atm6124*) jest instalowany automatycznie w systemie Windows podczas instalacji programu sterującego SAM-Prog, dzięki czemu po podłączeniu (obudowanego w peryferia jak pokazano na rys. 6) mikrokontrolera do USB sterownik jest aktywowany automatycznie.

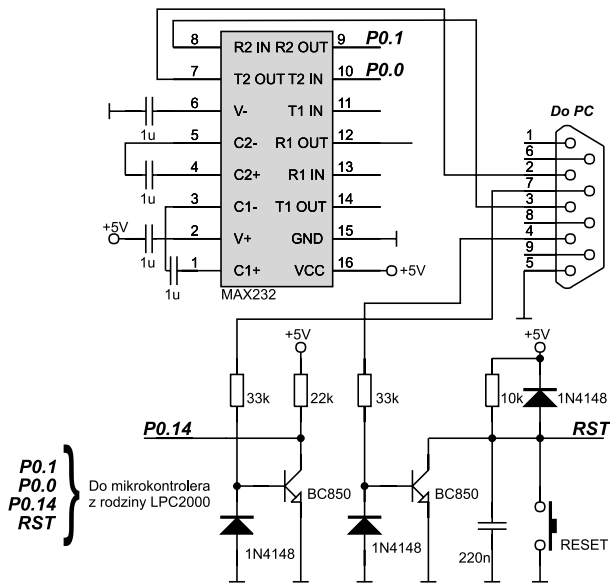
Mikrokontrolery AT91SAM7S przed pierwszym użyciem muszą zostać zainicjalizowane, co wymaga ze strony użytkownika (przy odłączonym zasilaniu) dołączenia wejścia TST i wejść

PGMEN0...2 (można je zostawić także niepodłączone, mają bowiem wewnętrzne pull-upy) do linii zasilającej mikrokontroler (+3,3 V), włączenia zasilania i oczekania co najmniej 10 sekund, po którym to czasie zasilanie mikrokontrolera należy odłączyć. Następnie odłączamy wejście TST od linii +3,3 V – mikrokontroler jest gotowy do współpracy z programem SAM-Prog (rys. 7). Niestety producent nie przewidział żadnej sygnalizacji poprawnego przebiegu kopiowania pamięci Flash, więc w niektórych przypadkach procedurę trzeba będzie (być może, autorowi się to zdarzało) powtórzyć. Należy pamiętać, że odtworzenie SAM-BA w pamięci mikrokontrolera jest konieczne za każdym razem, gdy chcemy zaprogramować ją ISP – wymaga to każdorazowo powtórzenia opisanej procedury. Kolejną ważną rzeczą, jaką należy zapamiętać, jest to, że przed ponownym zainicjowaniem SAM-BA pamięć Flash mikrokontrolera należy wykasować. W tym celu na wejście ERASE mikrokontrolera należy podać (przy włączonym zasilaniu) napięcie +3,3 V na czas nie krótszy niż 15 ms.

Rozwiązanie zastosowane przez Atmelą nie jest najwygodniejsze, ale pomimo niedociągnięć, sprawdza się w praktyce. Podczas projektowania urządzenia wyposażonego w mikrokontrolery AT91SAM7S warto pamiętać o zastosowaniu zworek lub innych rozwiązań umożliwiających jego wygodne programowanie ISP.



Rys. 7. Widok okna programu SAM-Prog



Rys. 8. Schemat elektryczny programatora ISP dla mikrokontrolerów LPC2100

Podsumowanie

Prezentowane w artykule rozwiązania sprzętowe – ze względu na wygodę – powinny być stosowane w systemach docelowych, oczywiście w przypadku, gdy nie wchodzi w grę korzystanie z JTAG-a. W przypadku budowania aplikacji na bazie mikrokontrolera z rodziny LPC2000 można pokusić się o wykonanie prostego programatora, którego schemat pokazano na rys. 8.

Mam nadzieję, że wskazówki zawarte w artykule rozwiewają najpoważniejsze wątpliwości Czytelników. Ewentualne pytania na temat mikrokontrolerów ARM można kierować na adres arm@ep.com.pl, będziemy starali się na nie na bieżąco odpowiadać. **Piotr Zbysiński, EP**
piotr.zbysinski@ep.com.pl

W jednym z najbliższych numerów EP zajmiemy się przedstawieniem sposobów programowania ISP mikrokontrolerów z rdzeniem ARM7TDMI z rodziny STR730.

Dodatkowe informacje:

- program do ISP firmy Philips: http://www.semiconductors.philips.com/files/markets/microcontrollers/philips_flash_utility.zip,
- programy narzędziowe dla mikrokontrolerów AT91: www.at91.com.