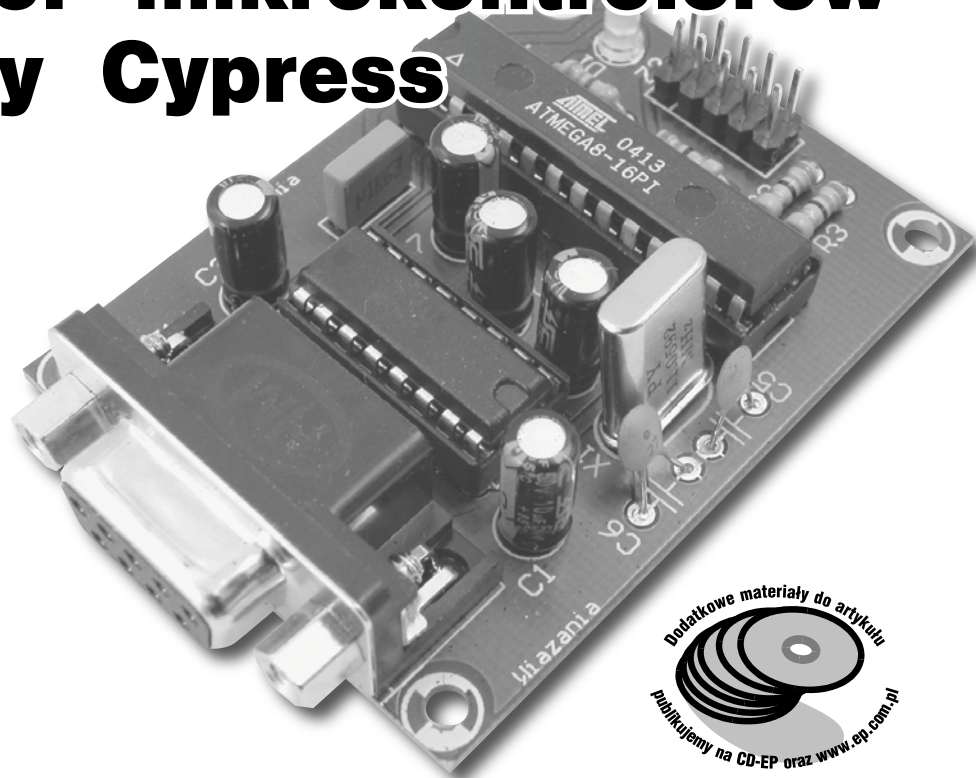


# Programator mikrokontrolerów PSoC firmy Cypress AVT-911

Znajomość popularnych typów mikrokontrolerów, takich jak '51, AVR czy PIC, jest wśród naszych Czytelników powszechna. Natomiast prawdopodobnie niewielu z nich próbowało „opanować” rewelacyjne mikrokontrolery PSoC (Programmable System-on-Chip), których czołowym producentem jest firma Cypress. Do fascynującej zabawy z nimi, prócz zintegrowanego środowiska projektowego niezbędny jest programator.

**Rekomendacje:** odkładanie „na bok” wielu interesujących podzespołów jest często spowodowane brakiem odpowiednich narzędzi ich implementacji. Prezentowany programator przyda się na pewno wszystkim, którzy chcieliby wykorzystać mikrokontrolery PSoC w swoich projektach.



Mikrokontrolery PSoC mogą być, tak jak i AVR programowane w systemie, czyli już po zamontowaniu w urządzeniu docelowym. W przypadku mikrokontrolerów PSoC interfejs przeznaczony do programowania został nazwany ISSP (In-System Serial Programming). Prezentowany programator ma pewne ograniczenia w stosunku do wersji komercyjnej, do strony na której został opisany zamieszczono link w końcowej części artykułu. Podstawowym ograniczeniem programatora jest liczba możliwych do zaprogramowania mikrokontrolerów. Można programować tylko mikrokontrolery w obudowach 20-wyprowadzeniowych.

Mikrokontrolery PSoC mogą być programowane w dwóch trybach: Reset lub PowerOn. Programator w prezentowanej wersji umożliwia programowanie mikrokontrolerów PSoC tylko w trybie Reset. Nie ma możliwości programowania mikrokontrolerów PSoC bez linii Xres, czyli mikrokontrolerów w obudowach 8-wyprowadzeniowych (w wersji komercyjnej możliwe było programowanie w obu trybach). Programator ma dużą szybkość programowania oraz prostą budowę, co wpływa na jego cenę. Komunikacja z programatorem odbywa się poprzez interfejs RS232, który jest w każdym komputerze.

Programator jest sterowany oprogramowaniem PSoC Prog.

## Mikrokontrolery PSoC

Mikrokontrolery PSoC są uniwersalnymi układami znajdującymi zastosowanie w przemyśle, elektronice medycznej, telekomunikacji i w samochodowych systemach sterowania. Tworzą rodzinę 8-bitowych mikrokontrolerów typu „programowalne systemy w jednym układzie scalonym”. Układy te mogą zastąpić tradycyjne mikrokontrolery wzbogacone o układy peryferyjne konfigurowane przez użytkownika, takie jak w pełni konfigurowalne peryferia cyfrowe

### PODSTAWOWE PARAMETRY

- prosta budowa
- duża szybkość programowania
- możliwość programowania mikrokontrolerów PSoC tylko w obudowach 20-wyprowadzeniowych
- komunikacja za pośrednictwem interfejsu RS232
- napięcia zasilania z systemu +5 V
- komunikacja z komputerem z prędkościami 38400, 57600 i 115200 bodów
- funkcja Autoreload

### Tylko PSoC-e

W mikrokontrolerach PSoC możliwe jest wybranie rodzaju i liczby peryferyjnych układów analogowych i cyfrowych oraz sposób wyprowadzenia sygnałów na płytkę drukowaną. Wskutek zmian ustawień rejestru, konfiguracja mikrokontrolera może zmieniać się tak, aby być dostosowana do różnych zastosowań. Wszystkie układy PSoC są dynamicznie rekonfigurowalne, co oznacza możliwość ich przeprogramowania w trakcie pracy systemu. Przykładowo, w dwukierunkowym systemie komunikacyjnym można skonfigurować ustawienia peryferyjne mikrokontrolera PSoC, aby pracował jako układ odbiorczy. Natomiast gdy użytkownik wciśnie przycisk nadawania, to jednostka centralna mikrokontrolera tak przekonfiguruje peryferia, aby służyły do nadawania. Po zwolnieniu przycisku obwody peryferyjne powrócą do trybu odbioru.

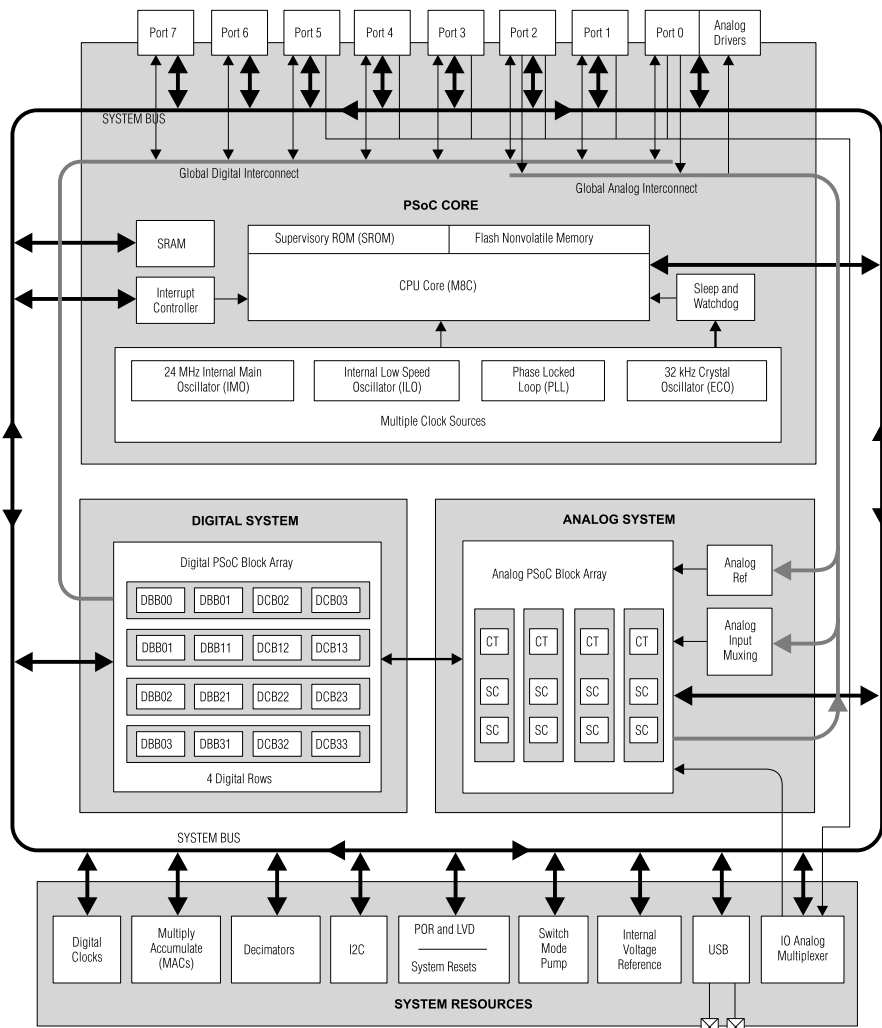
i analogowe, w których są dostępne nawet wzmacniacze operacyjne (rzadkość wśród mikrokontrolerów).

Układy rodziny PSoC składają się z 8-bitowego mikrokontrolera o częstotliwości taktowania 24 MHz, pamięci Flash o pojemności 4...64 kB, pamięci SRAM, zegara RTC oraz z programowalnych macierzy analogowych i cyfrowych, w których można realizować różne obwody peryferyjne. Dostępnych jest ponad 100 różnych układów peryferyjnych. Na rys. 1 przedstawiono schemat blokowy jednego z mikrokontrolerów PSoC. Cypress oferuje również zestaw narzędzi programowych do obsługi mikrokontrolerów PSoC. Zawiera on: kompilator C (za opłatą), assembler, linker, debugger, edytor peryferii oraz emulator wewnętrzny. Edytor peryferii umożliwia konfigurowanie mikrokontrolera PSoC metodą „drag and drop” przez przenoszenie peryferiów oraz modułów funkcjonalnych z biblioteki modułów użytkownika. Wybrane moduły są odwzorowywane w dostępnych blokach konfiguracyjnych PSoC. Dlatego też wykorzystanie we własnych aplikacjach rozbudowanych mikrokontrolerów PSoC nie jest więc trudne.

Wymienione narzędzia Cypress udostępnia bezpłatnie w zintegrowanym środowisku projektowym *PSoC Designer*. *PSoC Designer* zawiera zestaw prekonfigurowanych modułów peryferyjnych, zarówno analogowych jak i cyfrowych, co znacznie skraca czas tworzenia projektu. Zadaniem projektanta jest jedynie wybór odpowiednich modułów wykorzystywanych w realizowanym projekcie, określenie połączeń pomiędzy nimi, wybór parametrów z rozwijanego menu i kliknięcie na przycisku „Generate Application”. Wówczas następuje skompilowanie opisu projektu i wygenerowanie pliku z kodem programującym dla mikrokontrolera PSoC wraz z interfejsami API, tablicami wektorów i pełną dokumentacją.

**Opis działania układu**

Na rys. 2 przedstawiono schemat elektryczny programatora PSoC. Składa się on tylko z kilkunastu łatwo dostępnych elementów. Procesem programowania steruje mikrokontroler AVR ATmega8 lub ATmega48. Program sterujący można rozbudować tak, aby było możliwe także programowanie mi-

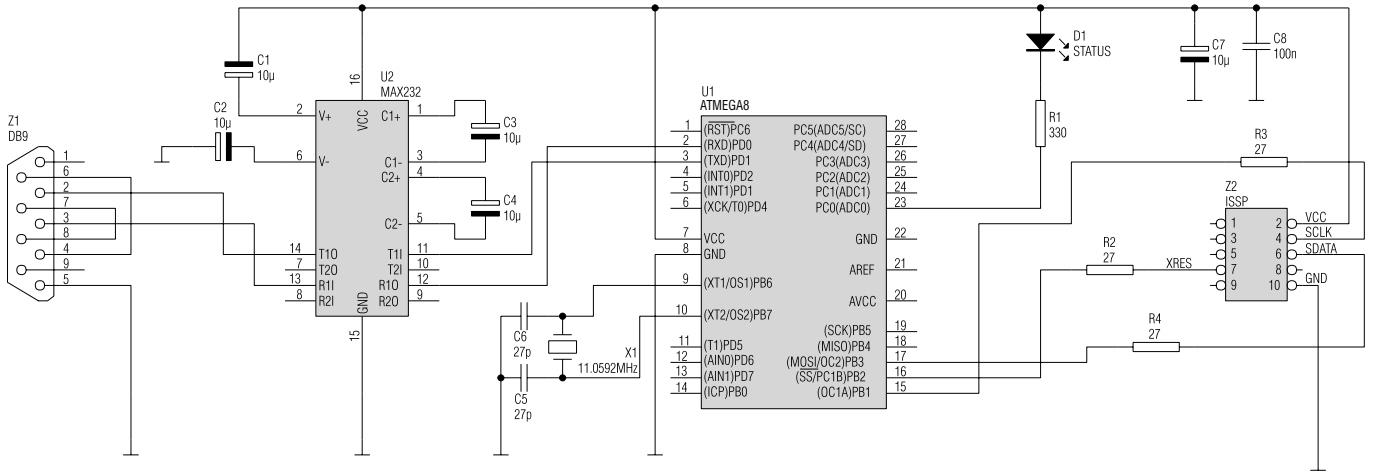


Rys. 1. Schemat blokowy mikrokontrolera PSoC

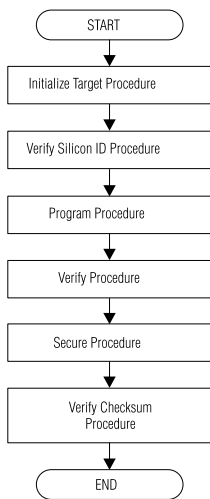
crokontrolerów AVR. Programator z komputerem komunikuje się za pośrednictwem interfejsu RS232, dlatego też wymagany jest konwerter poziomów napięć TTL do standardu RS232. W tym celu zastosowano układ MAX232. Kondensatory C1...C4 są wymagane do prawidłowej pracy przetwornicy zawartej w układzie MAX232. Częstotliwość taktowania mikrokontrolera U1 wynosi 11,0592 MHz i jest wyznaczona przez rezonator kwarcowy X1. Dioda LED D1 sygnalizują pracę programatora. Rezystor R1 ogranicza prąd płynący przez diodę LED. Linie interfejsu programującego ISSP są zabezpieczone rezystorami R2...R4. Kondensatory C7, C8 filtrują napięcie zasilające programator, które jest pobierane z programowanego systemu. Na złącze Z2 zostały wyprowadzone linie (jest ich 5) interfejsu programującego ISSP. Dwie z nich są liniami zasilającymi, Xres jest linią zerowania, SDATA to linia danych wejściowych i wyjścio-

wych, natomiast SCLK to linia sygnału zegarowego. W przypadku programowania w trybie Reset są wykorzystywane wszystkie linie interfejsu ISSP, natomiast w trybie PowerOn linia Xres nie jest wykorzystywana.

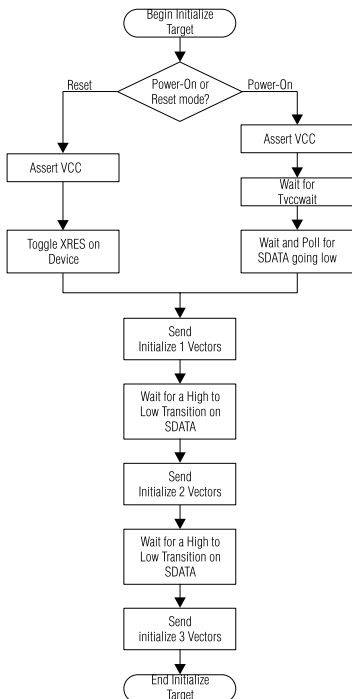
Na rys. 3 przedstawiono algorytm programowania mikrokontrolera PSoC. W pierwszej kolejności następuje inicjalizacja programowania układu docelowego (wejście w tryb programowania), a następnie odczytywany jest ID programowanego układu, czyli przypisany mu fabryczny numer identyfikacyjny. W dalszej kolejności są realizowane procedury programowania, weryfikacji, zabezpieczenia wczytanego do mikrokontrolera programu, weryfikacji sumy kontrolnej i zakończenia programowania. Oczywiście etapy weryfikacji oraz sprawdzenia sumy kontrolnej mogą być pominięte, ale wówczas wzrośnie ryzyko tego, że układ mikrokontrolera będzie błędnie zaprogramowany.



Rys. 2. Schemat elektryczny programatora



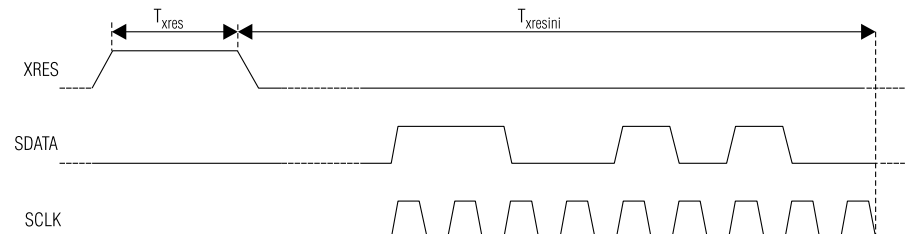
Rys. 3. Algorytm programowania mikrokontrolera PSoC



Rys. 4. Algorytm inicjalizacji programowania

Na rys. 4 przedstawiono etapy inicjalizacji programowania. Jak już wspomniano, są dwa tryby programowania: Reset oraz PowerOn. Prezentowany programator można wprowadzić tylko w tryb programowania Reset. W tym trybie do wejścia w tryb programowania używa się linii Reset. Niektóre mikrokontrolery PSoC (zwłaszcza w obudowach 8-wyprowadzeniowych) nie mają linii Xres. Można je programować tylko w trybie PowerOn, w którym do wejścia w tryb programowania używa się linii zasilającej mikrokontroler. Z tego powodu programator nie ma możliwości programowania małych PSoC-ów.

Na rys. 5 przedstawiono czasowe przebiegi sygnałów inicjalizujących programowanie dla trybu Reset. Po przedstawionej na rysunku sekwencji sygnałów na liniach programujących, do mikrokontrolera należy wysłać tak zwane wektory, które są niczym innym jak komendami. Komenda wejścia w tryb programowania składa się z trzech wektorów, które przedstawiono na rys. 6. Wektor jest ciągiem bitów o wartościach 0 i 1. W pierwszej kolejności należy wysłać wektor *Inicjalize-1*, następnie wektor *Inicjalize-2*, po czym wektor *Inicjalize-3*. Po wysłaniu tych trzech ciągów bitów mikrokontroler jest ustawiony w trybie programowania.



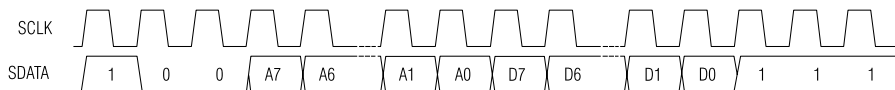
Rys. 5. Przebiegi czasowe sygnałów inicjalizujących programowanie dla trybu Reset

ler jest ustawiony w trybie programowania. Każda komenda (jak np.: wejście w tryb programowania Read, Write, Read ID, Erase) wysyłana do mikrokontrolera jest wektorem stanowiącym ciąg bitów, których liczba zależy od wysyłanej komendy. Po zidentyfikowaniu programowanego mikrokontrolera (poprzez odczyt jego numeru ID), można przejść do procedury programowania.

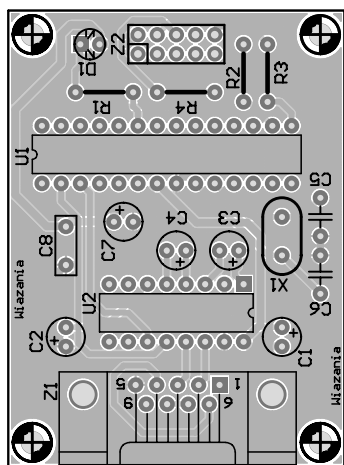
Na rys. 7 przedstawiono schemat blokowy procedury programowania. W mikrokontrolerach PSoC pamięć Flash jest podzielona na 64-bajtowe bloki. Podczas jego programowania w pierwszej kolejności następuje kasowanie zawartości pamięci Flash przez wysłanie wektora odpowiadającego komendzie Erase. W dalszej kolejności są przesyłane dane bloku (64 bajty) pamięci Flash, po czym następuje jego zapisywanie pod wskazany adres poprzez wysłanie odpowiedniego wektora. Kiedy zapisane zostaną wszystkie bloki pamięci Flash mikrokontrolera, procedurę programowania można uznać za zakończoną.

Na rys. 8 przedstawiono przebiegi czasowe sygnałów podczas zapisu 1 bajta danej do mikrokontrolera PSoC. Plik programujący dla mikrokontrolera generowany przez oprogramowanie PSoC Designer jest zapisywany w formacie IntelHEX.



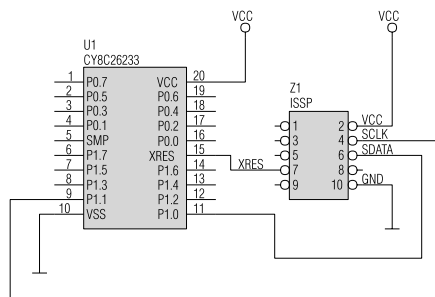


Rys. 8. Przebiegi czasowe sygnałów podczas zapisu 1 bajtu danej do mikrokontrolera PSoC



Rys. 9. Schemat montażowy programatora mikrokontrolerów PSoC

lecz wystarczy wybrać to polecenie, aby wybrany wcześniej plik został załadowany do bufora. W menu Device znajdują się polecenia związane z zerowaniem, identyfikowaniem, czyszczeniem, zapisem, odczytem, weryfikacją, zabezpieczeniem i sprawdzeniem sumy kontrolnej pamięci mikrokontrolera. W menu Options dostępna jest opcja Serial Port, w której można ustawić numer portu oraz prędkość transmisji. W menu Options można także włączyć/wyłączyć weryfikację, a także włączyć/wyłączyć funkcję Auto Reload. Włączenie funkcji Auto Reload powoduje, że po każdym przyciśnięciu przycisku programującego Program, zawartość wcześniej wybranego pliku z kodem programującym jest ładowana do bufora. Po każdym skompilowaniu opisu projektu nie trzeba ręcznie uaktualniać pliku w buforze. W menu Help znajdują się informacje o wersji programu oraz odczytane informacje z programatora o je-



Rys. 10. Schemat dołączenia programatora do mikrokontrolera PSoC

go wersji software'u i hardware'u. Po podłączeniu programatora do systemu z mikrokontrolerem PSoC oraz do komputera i wybraniu prawidłowego numeru portu, programator powinien od razu poprawnie pracować. Sprawdzenie komunikacji programatora z komputerem można dokonać wybierając z menu Help opcję Firmware Version. Powinno się ukazać okienko z odczytaną z programatora wersją oprogramowania. Dioda LED programatora powinna podczas programowania migać z częstotliwością zależną od realizowanej funkcji. Przycisk Identyfikacji w głównym oknie programu służy do identyfikacji programowanego układu. Identyfikacja jest przeprowadzana także po naciśnięciu przycisku programowania. Programator w przedstawionej wersji może programować mikrokontrolery PSoC tylko w trybie Reset, więc okno wyboru trybu pracy jest niedostępne. Mikrokontrolery PSoC posiadają kilka trybów zabezpieczenia wczytanego kodu programującego. Rodzaj zabezpieczenia może być zapisany w pliku z programem lub wybrany w oknie programu. Wybranie opcji Protection from HEX spowoduje, że rodzaj zabezpieczenia będzie odczytywany z pliku programu. Opcja Unprotected powoduje że program nie będzie zabezpieczony. Opcja Faktory Upgrade blokuje tylko odczyt pliku programującego z mikrokontrolera. Opcja Fidel Upgrade blokuje odczyt i zapis programu mikrokontrolera, ale jest możliwy zapis danych do pamięci Flash przez program zapisany do mikrokontrolera. Opcja Full Protection blokuje dodatkowo możliwość zapisu danych do pamięci Flash mikrokontrolera przez załadowany do niego program. Przycisk Reset w głównym

**WYKAZ ELEMENTÓW**

**Rezystory**

- R1: 330 Ω
- R2...R4: 27 Ω

**Kondensatory**

- C1...C4, C7: 10 μF/16 V
- C5, C6: 27 pF
- C8: 100 nF

**Półprzewodniki**

- U1: ATmega48 lub ATmega8
- U2: MAX232
- D1: LED 3 mm zielona
- X1: Kwarc 11,0592 MHz

**Inne**

- Z1: Złącze DB9F kątowe do druku
- Z2: Goldpin 2x10

oknie programu służy jedynie do zerowania mikrokontrolera, co czasami może być przydatne.

**Podsumowanie**

Przedstawiony programator jest jednym z podstawowych narzędzi bez którego nie można rozpocząć pracy z mikrokontrolerami PSoC. W jednym z następnych numerów EP zostanie opisany zestaw uruchomieniowy przeznaczony dla PSoC-ów, więc już teraz należy podjąć decyzję o zbudowaniu programatora. Do zabawy z mikrokontrolerami PSoC zachęca także darmowe oprogramowanie narzędziowe PSoC Designer, które wykonuje wiele pracy za projektanta. Dzięki temu przygotowywanie aplikacji na mikrokontroler PSoC nie jest trudne i zajmuje mało czasu. W najbliższym czasie na łamach EP będzie publikowany kurs programowania mikrokontrolerów PSoC.

**Marcin Wiązania, EP**  
marcin.wiazania@ep.com.pl



Rys. 11. Okno programu PSoC Prog