

UnISProg - uniwersalny programator ISP

AVT-560

Praktycznie każdy producent mikrokontrolerów z pamięcią Flash i współczesnych układów programowalnych opracował do ich programowania własny, wyspecjalizowany programator.

Wskutek tego zastosowanie w jednym projekcie mikrokontrolera AVR, układu CPLD firmy Altera i ispGAL-a firmy Lattice zmusza projektanta do korzystania z trzech (sic!) różnych programatorów.

Tak nie musi być, o czym zamierzam przekonać Czytelników tego artykułu.

Rekomendacje: opisany w artykule programator jest narzędziem niezbędnym w pracowni wszystkich elektroników korzystających z układów PLD pochodzących od różnych producentów. „Przy okazji“ można go także wykorzystać do programowania ISP mikrokontrolerów AVR.



Sytuację, gdy do zaprogramowania ISP trzech układów potrzebne były trzy różne programatory, można było uznać za naturalną na początku wdrażania interfejsów ISP (10 lat temu), lecz nie teraz. Jest to tym bardziej absurdalne, że dzięki powszechnemu stosowaniu uniwersalnych języków opisu sprzętu (HDL) do realizacji projektów w układach PLD, korzystanie z układów pochodzących od różnych producentów (choćby z powodu cen) jest rzeczą zupełnie naturalną.

Problem został częściowo rozwiązany przez producenta programatorów Uprog (www.rk-system.com.pl), które wyposażono w interfejs JTAG, za pomocą którego można zaprogramować zdecydowaną większość układów PLD. Nie zawsze jednak zakup programatora o tak dużych możliwościach (Uprog to rodzina programatorów uniwersalnych) jest uzasadniony i możliwy, w związku z czym w naszym laboratorium powstał uniwersalny programator, za pomocą którego można programować w systemie układy CPLD/FPGA

firm: Altera, Atmel, Lattice i Xilinx oraz mikrokontrolery AVR.

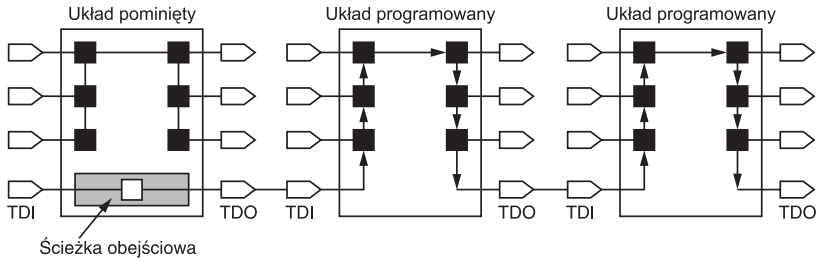
ISP vs JTAG vs SPI

Na początek wyjaśnienie pojęć, które często są mylone, a których rozróżnienie jest istotne.

AVR-y, jako pierwsze z grupy mikrokontrolerów popularnych wyposażono w możliwość programowania w systemie. Atmel zastosował do tego celu interfejs SPI

Możliwości programatora UnISProg:

- ✗ emuluje programatory ISP: ByteBlaster MV, DLC-5, Lattice Cable, ispDOWNLOAD i STK200,
- ✗ programuje układy CPLD firm: Altera (MAX3000, MAX7000S/A/B, MAX9000), Atmel (AT1500/1500S/ASL), Lattice (ispMACH, ispGAL22LV10, ispGAL22V10AV/AB) oraz Xilinx (XC9500/9500XL, CoolRunner),
- ✗ programuje układy FPGA firm: Altera (zasilane napięciem od 3,3 do 5 V), Lattice (ORCA) oraz Xilinx (zasilane napięciem od 3,3 do 5 V),
- ✗ programuje układy ispGDx i ispPAC firmy Lattice,
- ✗ może być stosowany do programowania układów wyposażonych w interfejs ispLattice (CPLD starszych generacji, jak na przykład ispLSI2000, ispGAL22V10),
- ✗ programator jest zasilany z urządzenia z programowanym układem (5 V),
- ✗ współpraca z PC odbywa się poprzez interfejs Centronics.



Rys. 1. W łańcuchu JTAG można wybrać, które układy będą programowane, a które nie

(Serial Peripheral Interface), który w specyficznych warunkach spełnia rolę interfejsu wejściowego dla wbudowanego w mikrokontrolery programatora pamięci Flash. Dane są wprowadzane i odczytywane szeregowo w sposób synchroniczny.

Nieco inaczej jest w przypadku układów PLD, które w większości wyposażono w interfejs JTAG (zgodny z jedną z norm: IEEE1149 lub IEEE1532) - tak jest w większości układów CPLD - lub synchroniczny interfejs szeregowy służący do konfiguracji układów FPGA. Zarówno w przypadku klasycznego JTAG-a, jak i interfejsów przystosowanych do konfiguracji układów FPGA istnieje możliwość programowania/konfiguracji pojedynczego układu lub wielu układów połączonych w tzw. łańcuch JTAG (rys. 1).

Niezależnie od rodzaju interfejsu zastosowanego do programowania (JTAG/SPI, czasami także RS232), programowanie/konfigurowanie układu zamontowanego w urządzeniu jest nazywane ISP od *In System Programmable*. Programator prezentowany w artykule emuluje interfejsy JTAG oraz SPI w trybach umożliwiających konfigurację najpopularniejszych rodzin układów PLD (FPGA) i mikrokontrolerów AVR w systemie (ISP).

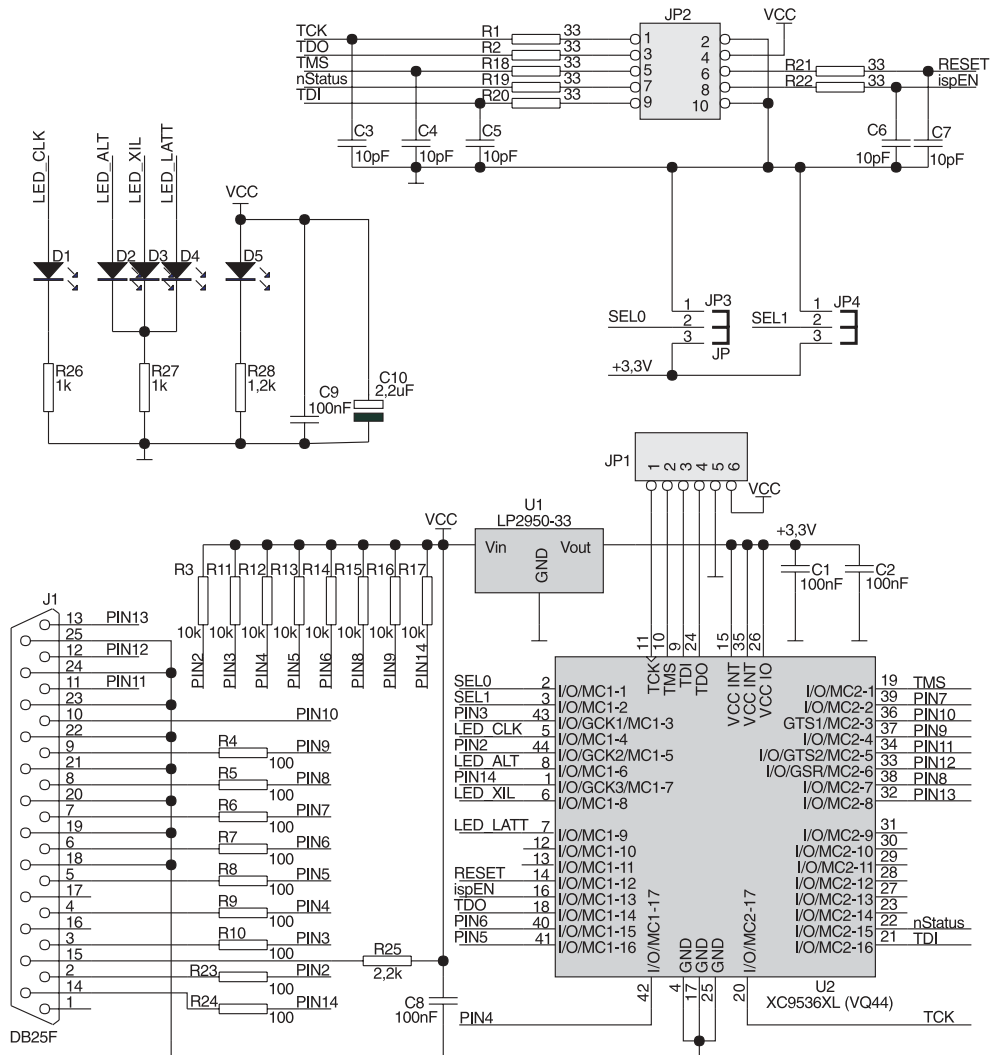
Opis układu

Schemat elektryczny programatora pokazano na rys. 2. Jego budowa jest bardzo prosta dzięki zastosowaniu układu programowalnego firmy Xilinx

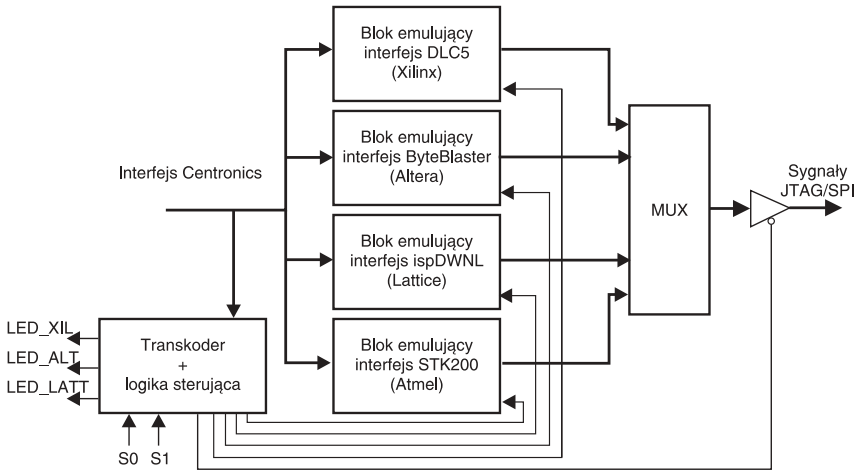
(U2), w którym zintegrowano wszystkie bloki funkcjonalne emulowanych interfejsów. Projekt przygotowano w języku VHDL (źródła i plik wynikowy są dostępne na CD-EP1/2004B), dzięki czemu można go zmodyfikować stosownie do własnych potrzeb, na przykład dodając opisy HDL kolejnych interfejsów ISP. Na rys. 3 pokazano uproszczony schemat blokowy programatora zaimplementowanego w układzie U2.

Co to znaczy ISP?
 Mianem ISP (*In System Programmable*) jest określany sposób programowania pamięci Flash/EEPROM wbudowanej w układy scalone (przed wszystkim mikrokontrolery, PLD i pamięci nieulotne) po ich zamontowaniu w urządzeniu docelowym. Programowanie ISP odbywa się zazwyczaj poprzez interfejs SPI, JTAG lub RS232.

Układ U2 należy do rodziny XC9500XL produkowanej przez firmę Xilinx. Jest on przystosowany do zasilania napięciem 3,3 V, ale wszystkie jego linie wejściowe są przystosowane do współpracy z układami cyfrowymi zasilanymi napięciem 5 V, a poziomy napięć na liniach wyjściowych są zgodne ze standardem TTL.



Rys. 2. Schemat elektryczny programatora



Rys. 3. Uproszczony schemat blokowy projektu zaimplementowanego w układzie U2

Napięcie zasilające jest stabilizowane przez układ U1 (stabilizator LDO - *Low Drop Out*), na wejście którego jest podawane napięcie +5 V z urządzenia, w którym znajduje się programowany układ. Napięcie zasilające na wejściu stabilizatora nie może być niższe niż ok. 3,7 V, ponieważ nie będzie on pracował prawidłowo.

Jumpery JP3 i JP4 służą do ustalenia trybu pracy programatora (fot. 4). Wybór użytkownika jest sygnalizowany za pomocą trzech diod LED (D2, D3 i D4), przy czym w trybie STK200 świecą jednocześnie diody D2 (LED_ALT) i D4 (LED_LATT). Diody D1 służy do monitorowania sygnałów zegarowych każdego z interfejsów, a dioda D5 sygnalizuje dołączenie zasilania do złącza JP2. Złącze to służy do dołączenia programatora do programowanego układu, a wyprowa-

dzenie na nim sygnałów jest zgodne z ByteBlasterem, ByteBlasterem MV oraz interfejsem isp-DOWNLOAD.

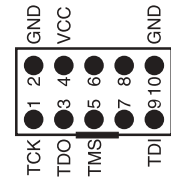
Na rys. 5 pokazano przypisanie sygnałów wyprowadzonym JP2 w trybie pracy JTAG, na rys. 6 w trybie konfiguratora układów FPGA firmy Altera (*Passive Serial*), na rys. 7 w trybie konfiguratora układów FPGA firmy Xilinx, a na rys. 8 w trybie programatora układów z interfejsem isp-Lattice.

Przed rozpoczęciem korzystania z programatora należy zaprogramować układ U2, do czego służy złącze JP1. Jego styki dołączono do wyprowadzeń interfejsu JTAG układu U2. Funkcje styków JP1 opisano na płycie drukowanej, co ułatwia programowanie układu. Do zaprogramowania U2 potrzebny będzie dowolny programator przystosowany do programowania (w systemie) układów XC9536XL. Egzemplarz modelowy programowano za pomocą programatora ZL4PRG (<http://www.btc.pl/index.php?id=zl4prg>) oraz stacjonarnego programatora Uprog 48HS. Aby zminimalizować kłopoty Czytelników decydujących się na zakup kitów, układy dostarczane w zestawach będą przylutowane do płytki i zaprogramowane.

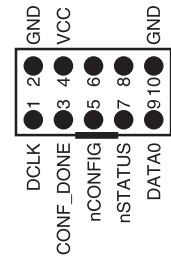
Montaż i uruchomienie

Widok rozmieszczenia elementów na płycie drukowanej programatora pokazano na rys. 9.

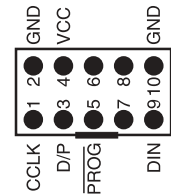
Montaż (w przypadku zakupu samej płytki drukowanej bez zamontowanego układu U2) należy



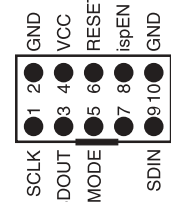
Rys. 5. Przypisanie sygnałów i zasilania wyprowadzonym JP2 w trybie pracy JTAG



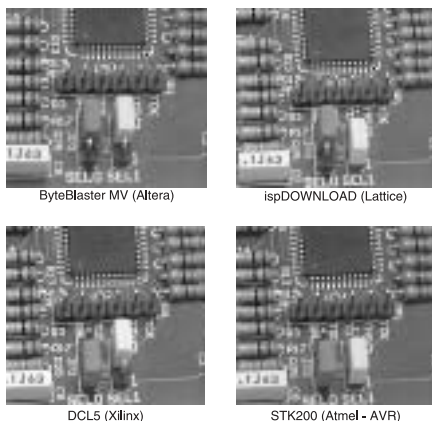
Rys. 6. Przypisanie sygnałów i zasilania wyprowadzonym JP2 konfiguratora układów FPGA firmy Altera (*Passive Serial*)



Rys. 7. Przypisanie sygnałów i zasilania wyprowadzonym JP2 konfiguratora układów FPGA firmy Xilinx



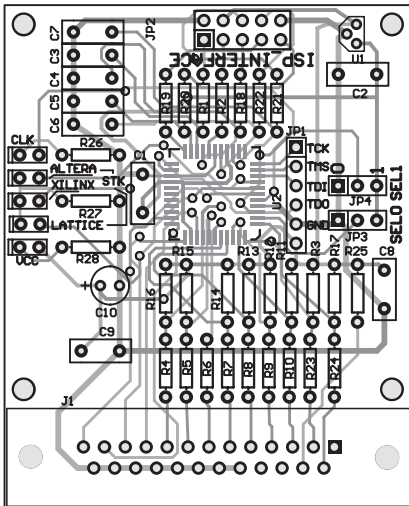
Rys. 8. Przypisanie sygnałów i zasilania wyprowadzonym JP2 w trybie pracy ispLattice



Fot. 4. Tryb pracy programatora ustala się za pomocą jumperów JP3 i JP4

Kłopot z programowaniem

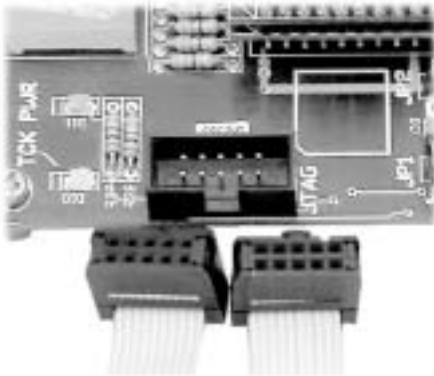
Zastosowany w projekcie układ XC9536XL musi być przed użyciem zaprogramowany, za pomocą programatora ISP. Z tego powodu w kitach układy te będą dostarczane przylutowane do płytki i zaprogramowane.



Rys. 9. Rozmieszczenie elementów na płytce drukowanej

punktach lutowniczych (trzeba sprawdzić położenie wyprowadzenia numer 1!), jedno ze skrajnych wyprowadzeń przygrzewamy grotem lutownicy do cyny znajdującej się na powierzchni punktu. Następnie w ten sam sposób przygrzewamy wyprowadzenie znajdujące się po przeciwnej stronie. Pozostałe wyprowadzenia tak przymocowanego do płytki układu lutujemy, korzystając ze spoiwa o średnicy 0,38 mm lub zbliżonej. Po przyłutowaniu układu U2 z powierzchni płytki warto usunąć resztki topnika za pomocą któregoś z dostępnych środków chemicznych (np. KONTAKT PCC).

Kolejność montażu pozostałych elementów jest obojętna. Nie jest także wymagana szczególna precyzja montażu. W zależności od obudowy można zastosować różne rodzaje złącza



Fot. 10. Kabel połączeniowy powinien być zakończony z dwóch stron wtyczkami ZWS10

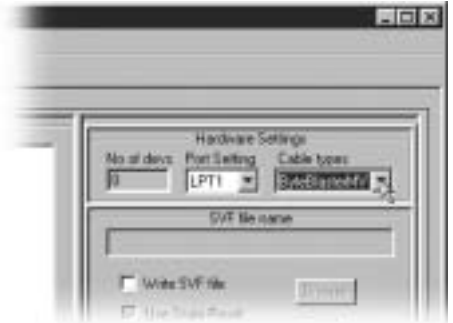
JP2 (z obudową, z zatrzaskiem, „gołe“ szpilki złożone itp.). Nastawniki JP3 i JP4 można zastąpić dowolnymi innymi przełącznikami dwupozycyjnymi.

Eksplatacja

Programator współpracuje z oryginalnym oprogramowaniem udostępnianym bezpłatnie przez firmę produkującą układy PLD oraz dostępnym bezpłatnie na kilku niezależnych portalach internetowych (dotyczy to programowania mikrokontrolerów AVR). Podłączenie programatora do komputera wymaga zastosowania kabla BD25M/DB25M (jego zalecana długość wynosi 1 metr), natomiast połączenie z programowanym układem wymaga zastosowania płaskiego kabla taśmowego zakończonego z dwóch stron wtyczkami ZWS10 (fot. 10). Długość tego kabla także nie powinna przekraczać 1 metra, co wynika z dość wysokiej częstotliwości sygnału zegarowego synchronizującego transfer danych.

Pracę z programatorem należy rozpocząć od ustalenia za pomocą JP3 i JP4 rodzaju emulowanego interfejsu. W zależności od typu programowanych układów należy zastosować odpowiednie oprogramowanie (udostępniamy je na płycie CD-EP1/2004B):

- układy PLD firmy Altera można programować za pomocą programów ASAP lub Quartus Programmer - programator powinien pracować w trybie ByteBlaster MV,
- układy CPLD firmy Atmel można programować za pomocą programu AtmelISP, przy czym programator powinien pracować w trybie ByteBlaster MV (w menu *Cable Types* należy wybrać



Rys. 11. Podczas programowania układów CPLD firmy Atmel za pomocą programu AtmelISP w menu *Cable Types* należy wybrać *ByteBlaster MV*

ByteBlaster MV, jak na rys. 11),
- układy PLD, ispPAC i ispGDX firmy Lattice można programować za pomocą programu ispVM System - programator należy skonfigurować do pracy w trybie ispDOWNLOAD/Lattice Cable (rys. 12),

- układy PLD firmy Xilinx można programować na dwa sposoby:
 - korzystając z niezależnych programów służących do programowania układów z rodzin XC9500, FPGA i XPLA (programator może pracować w trybie ByteBlaster lub DLC5),
 - korzystając z programu obsługującego programator, wbudowanego w system projektowy WebPack ISE (programator musi być skonfigurowany jako DLC5),
 - mikrokontrolery AVR można programować za pomocą programu IC-PROG, programator musi być skonfigurowany jako STK200.

W kolejnym kroku należy podłączyć programator do urządzenia, w którym znajduje się programo-

Niestandardowa możliwość

Programator UprogHS 48 może pracować jako programator-konfigurator układów wyposażonych w interfejs JTAG. Programator jest przystosowany do obsługi JTAG-a w standardach napięciowych: 2,7/3,3 i 5 V (zrzut ekranu poniżej), a pliki wejściowe muszą być przygotowane w formacie SVF lub JAM.





Rys. 12. Podczas programowania układów PLD, ispPAC i ispGDX firmy Lattice za pomocą programu ispVM System należy wybrać opcję Lattice w oknie selekcji Cable Type

wany układ. Urządzenie powinno mieć włączone zasilanie, a do złącza służącego do dołączenia programatora powinno być dostarczone stabilizowane napięcie +5 V (pobór prądu przez programator nie przekracza 40 mA).

Pierwszy krok do PLD
Programator UnISProg
będzie podstawowym
narzędziem wykorzystywanym
podczas kursu układów
programowalnych, który
rozpoczniemy w EP3/2004.

W zależności od połączenia programowanych układów (pojedynczy, łańcuch JTAG itp.) należy odpowiednio skonfigurować oprogramowanie sterujące, załadować plik/pliki JED/SVF/POF/HEX/SOF/BIN (w zależności od programowanego układu) i rozpocząć programowanie.

Piotr Zbysiński, EP
piotr.zbysinski@ep.com.pl

Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: pcb.ep.com.pl oraz na płycie CD-EP1/2004B w katalogu PCB.

WYKAZ ELEMENTÓW

Rezystory

R1, R2, R18...R22: 22Ω
 R3, R11...R17: 10kΩ
 R4...R10, R23, R24: 100Ω
 R25: 2,2kΩ
 R26, R27: 1kΩ
 R28: 1,2kΩ

Kondensatory

C1, C2, C8, C9: 100nF
 C3...C7: 10pF
 C10: 2,2μF/16V

Półprzewodniki

U1: LP2950-33
 U2: XC9536XL-10VQ44C (zaprogramowany)
 D1, D2, D3, D4, D5: diody LED prostokątne

Różne

J1: DB25F do druku
 JP2: ZWS10-G
 JP1: gold-pin 6x1
 JP3, JP4: 2 gold-piny 3x1 + jumper