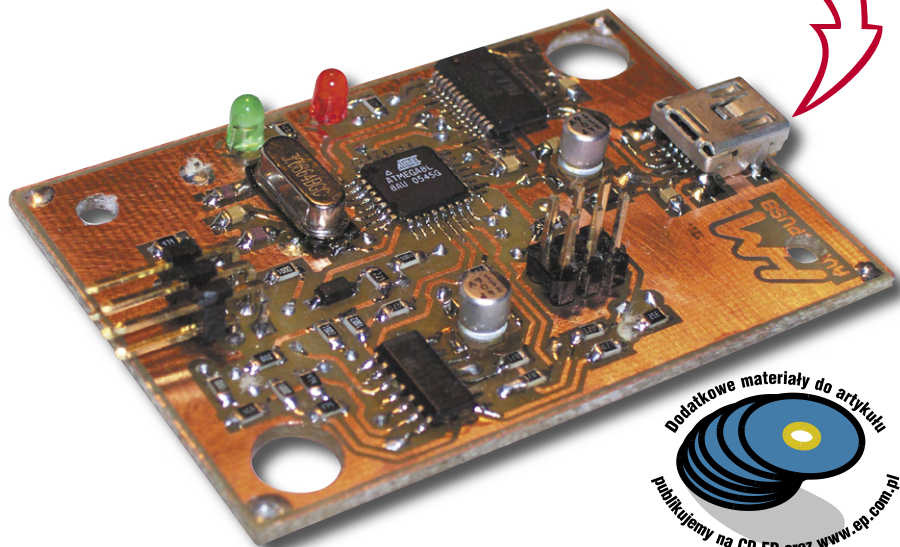


Programator AVRISP z interfejsem USB

AVT-988

PROJEKT
Z OKŁADKI



Dodatkowe materiały do artykułu
publikujemy na CD-EP oraz www.ep.com.pl

Rynek mikrokontrolerów w Polsce przez długi czas zmonopolizowany był przez różne odmiany procesorów rodziny '51, produkowanych przez wielu producentów. Niektóre typy tego mikrokontrolera przeżyły swoją drugą młodość między innymi dzięki wprowadzeniu modeli z pamięcią Flash oraz pracujących ze zredukowaną liczbą cykli zegara przypadających na jeden cykl maszynowy. Nie bez znaczenia był również fakt, że tajniki budowy i programowania tego mikrokontrolera przewijały się i nadal się przewijają przez większość programów nauczania szkół i uczelni.

Rekomendacje:

przyrząd niezbędny do pracy dla każdego elektronika wykorzystującego mikrokontrolery AVR i środowisko programistyczne AVR Studio 4.12.

PODSTAWOWE PARAMETRY

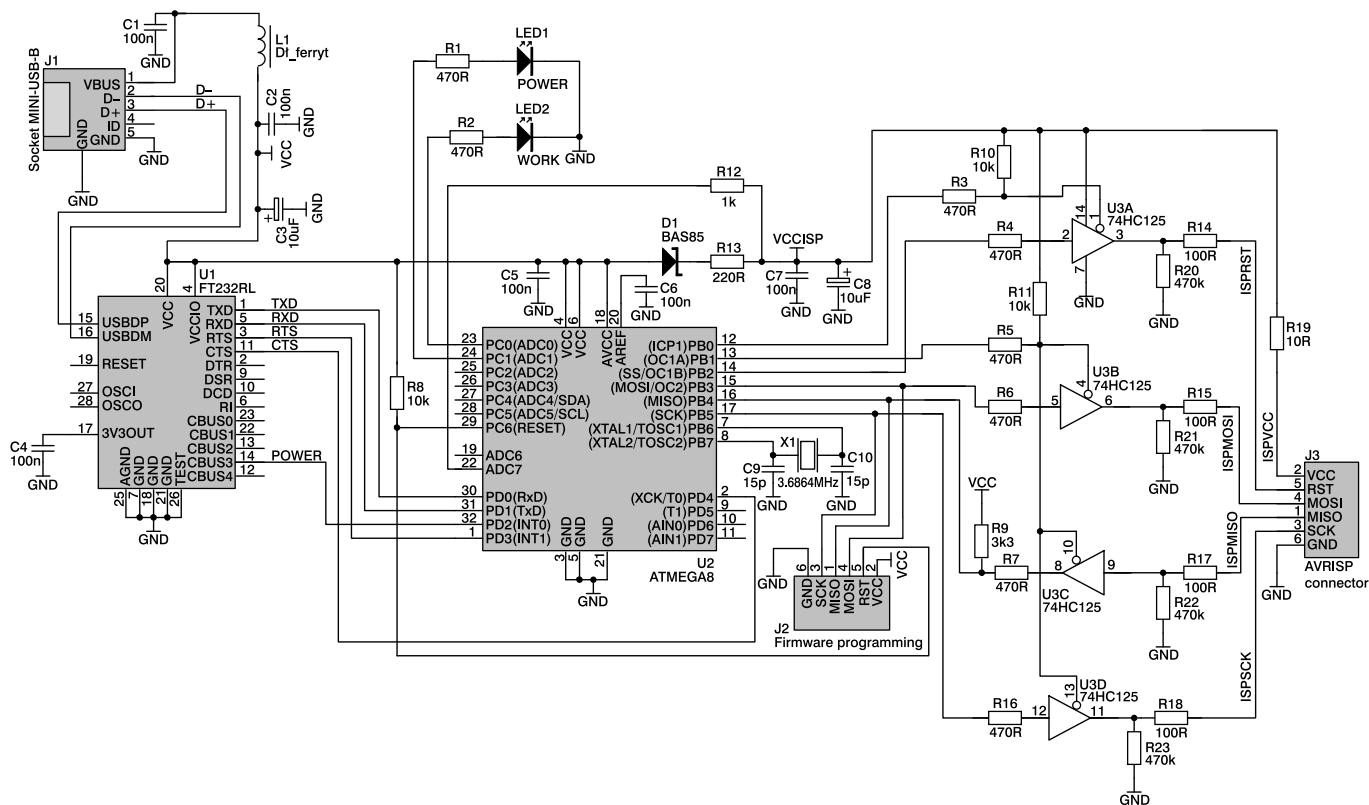
- Płytką o wymiarach 60x41 mm
- Zgodność z protokołem STK500V2
- Bezpośrednia obsługa programatora przez AVR Studio
- Połączenie z komputerem przez interfejs USB, bez konieczności doprowadzania oddzielnego zasilania
- Obsługa trybu uśpienia magistrali USB
- Programowanie mikrokontrolerów AVR w systemie przy wykorzystaniu złącza zgodnego ze standardem Atmel
- Programowanie procesorów nie zamontowanych (z zastosowaniem odpowiednich przystawek)
- Programowanie w szerokim zakresie napięć zasilających systemu docelowego (gwarantowane od 2,7 V do 5 V, poniżej 2 V bez gwarancji)
- Duża szybkość programowania, konfigurowalna w zależności od częstotliwości taktowania procesora docelowego

Dojście do głosu innych rodzin mikrokontrolerów było możliwe głównie dzięki znacznemu spadkowi ich cen, a także dzięki szybko postępującemu rozwojowi sieci dystrybutorów, oferujących niemal dowolnie wymarzony procesor wraz ze wsparciem technicznym. Obecnie najbardziej popularną rodziną mikrokontrolerów stosowanych przez polskich konstruktorów wydają się być AVR-y firmy Atmel. Ceny tych 8-bitowych kości spadły znacznie, kiedy na rynku pojawiły się różnej maści procesory z rdzeniem ARM7TDMI, które oferowały dużo większą wydajność za porównywalną cenę. Jednak mimo umacniającej się pozycji ARM-ów, które zapewne na długi czas przejmą prowadzenie po wiekowej '51, popyt na AVR-y nie maleje. Dzieje się tak dlatego, że moc 32-bitowych ARM-ów nie zawsze jest potrzebna, a dodatkowo firma Atmel wypuszcza nowe modele oraz aktualizuje narzędzia, wychodząc naprzeciw potrzebom konstruktorów.

Jednym z takich kroków, chociaż nieco spóźnionym, było „zauważenie” przez firmę Atmel ogromnej popularności kompilatora GCC wykorzystywanego do programowania AVR-ów w języku C. W listopadzie 2005 roku ukazało się na stronie producenta środowisko AVR Studio oznaczone numerem 4.12, najpierw jako beta, a następnie jako wersja oficjalna. Najnowszą wersję tego darmowego narzędzia można

pobrać ze strony www.atmel.com. Jego najważniejszą nową cechą, oprócz standardowo wprowadzanych usprawnień i poprawek błędów, była integracja z kompilatorem GCC. Chociaż kompilator należy nadal zainstalować osobno, a poprzednie wersje AVR Studio w lepszy lub gorszy sposób również umożliwiały wykorzystanie swoich możliwości dla programów pisanych w C, to dla wielu, zwłaszcza początkujących użytkowników, możliwość wyboru języka programowania (C czy assembler), a także uwolnienie od zagłębiania się w edycję pliku makefile i konfiguracji kompilatora GCC jest cechą wielce pożądaną. No a czego potrzeba jeszcze konstruktorowi oprócz niezłego środowiska i dużej możliwej do wykorzystania rodziny mikrokontrolerów? Potrzeba oczywiście programatora.

Środowisko AVR Studio oferuje możliwość obsługi przede wszystkim programatorów „firmowych”, takich jak STK500, AVRISP, AVRISP mk II oraz interfejsów JTAG ICE, także w wersji mk II. Nic jednak nie stoi na przeszkodzie, aby wykonać swój programator „podszywający się” pod jeden z powyższych. Nie jest to zadanie specjalnie trudne, zwłaszcza, że firma Atmel udostępniła protokół komunikacyjny pomiędzy PC a programatorem i oczywiście dosyć dokładnie opisuje sygnały niezbędne do prawidłowego zaprogramowania procesora.



Rys. 1. Schemat elektryczny programatora

W przypadku programatora AVRISP w rzeczywistości mamy do czynienia z protokołem opisywanym jako STK500, gdyż w tym zestawie uruchomieniowym został zintegrowany protoplasta programatora AVRISP. Od wersji 4.11 AVR Studio, Atmel wprowadził nowy standard komunikacyjny pomiędzy PC a programatorem (STK500V2), który nie jest kompatybilny z rozwiązaniem poprzednim. Został on jednak znacznie unowocześniony i posiada cechy niezależniące programator od programowanego procesora, co pozwala na łatwe dostosowanie do nowo powstałych komponentów wyłącznie poprzez uaktualnienie oprogramowania na PC. Cały protokół jest dosyć dokładnie opisany w nocie aplikacyjnej AVR068. Do opisu producent dołącza również plik *command.h* zawierający definicje stałych wykorzystywanych przez protokół, przydatnych przy tworzeniu własnego oprogramowania. Prezentowany programator został opracowany na podstawie powyższych materiałów.

Budowa programatora

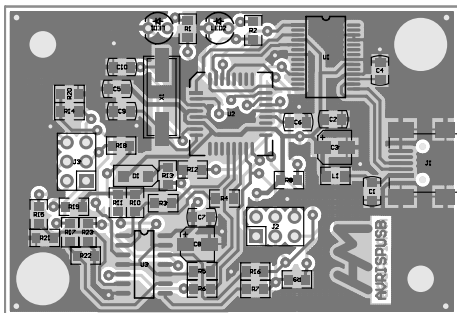
Schemat elektryczny programatora przedstawiono na rys. 1. Programator składa się z trzech głównych bloków:

układu FT232RL stanowiącego interfejs łączący z portem USB komputera PC, mikrokontrolera ATmega8 realizującego wszystkie funkcje logiczne programatora oraz bufora wyjściowego 74HC125. Całość jest zasilana napięciem +5 V pochodzącym z linii zasilającej interfejsu USB poprzez filtrujący obwód LC. Cewkę L1 stanowi tutaj przeciwzakłóceńowy dławik SMD o indukcyjności około 1 μ H, o budowie podobnej do perełki ferrytowej nałożonej na przewód.

Oryginalny programator AVRISP posiada interfejs RS-232 wykorzystywany do komunikacji z PC. W opracowywanej konstrukcji założono jednak współpracę programatora z dowolnym komputerem (czyli także nowoczesnym laptopem bez portu RS) oraz niekłopotliwe zasilanie programatora, zdecydowano się więc na wykorzystanie interfejsu USB. Układ FT232RL (U1) stanowi tutaj pomost pomiędzy złączem USB PC a portem szeregowym mikrokontrolera i został na wybrany nieprzypadkowo z wielu podzespołów innych producentów pełniących podobną funkcję. Jest to stosunkowo nowy podzespół należący do rodziny konwerterów RS-232/USB produkowanej przez firmę FTDI o ciekawych cechach upraszczających konstrukcję opisywanego urządzenia. Największą

jego zaletą jest brak konieczności dołączenia jakichkolwiek zewnętrznych komponentów, poza jednym kondensatorem odsprężającym. Dodatkowo próby przeprowadzane z tym układem oraz innymi konwerterami RS/USB wykazały, że pracuje on bardzo stabilnie w przedstawionym zastosowaniu i w odróżnieniu do innych podzespołów nie ma tendencji do zawieszania sterownika (a wraz z nim całej aplikacji AVR Studio). Jak każdy konwerter tego typu, po zainstalowaniu sterownika widziany jest z poziomu systemu operacyjnego PC jako kolejny port COM, co oczywiście jest niezbędne dla możliwości korzystania z aplikacji pisanych z myślą o obsłudze standardowego interfejsu RS-232.

Układ FT232RL komunikuje się z mikrokontrolerem U2 poprzez linie RxD, TxD, natomiast linie sterujące RTS i CTS zostały dołączone na zapas w celu ewentualnego obsłużenia innego zastosowania programatora (obecna wersja oprogramowania ich nie wykorzystuje). Dodatkowo, układ U1 za pośrednictwem sygnału na linii CBUS3 informuje mikrokontroler o stanie pracy magistrali USB, co jest wykorzystywane przez program do wprowadzenia programatora w stan uśpienia. Dołączony do U1 kondensator C4 filtruje wewnętrzne napięcie



Rys. 2. Schemat montażowy

zasilania 3,3 V. Programator został wyposażony w standardowe gniazdo MINI-USB-B, umożliwiające dołączenie go do komputera PC.

Głównym podzespołem realizującym wszystkie funkcje programatora jest mikrokontroler U2 – ATmega8. Oprogramowanie dla mikrokontrolera przygotowano w oparciu o opis protokołu komunikacyjnego STK500V2 w języku C z wykorzystaniem kompilatora GCC. Mikrokontroler U2 pracuje w standardowej aplikacji z zewnętrznym rezonatorem kwarcowym wyznaczającym częstotliwość jego pracy. Częstotliwość kwarcu X1 została wybrana z punktu widzenia prędkości transmisji poprzez złącze szeregowo. Starano się nie wybierać wartości zbyt dużej ze względu na rosnący wraz z częstotliwością pobór mocy i poziom generowanych zakłóceń.

Do mikrokontrolera zostały dołączone dwie diody świecące: LED1 o kolorze zielonym informującym o włączeniu zasilania i gotowości programatora do pracy oraz LED2 (czerwona) informująca o realizacji przez programator operacji na dołączonym układzie. Do jednego z wejść analogowych (ADC7) zostało doprowadzone napięcie zasilania programowanego systemu.

Programowanie układu docelowego odbywa się za pośrednictwem linii portu PB0 pracujących częściowo w sposób programowy, a częściowo wykorzystując wbudowany w mikrokontroler interfejs SPI. Interfejs sprzętowy wykorzystywany jest przy dużych prędkościach programowania, natomiast przy prędkościach małych transmisja jest realizowana na drodze programowej. Do interfejsu SPI dołączone jest również standardowe złącze ISP umożliwiające zaprogramowanie programatora podczas jego budowy. Do wykonania tego zadania niezbędny jest niestety dodatkowy programator procesorów AVR (pojawia się tutaj odwieczny problem jajka i kury). Operację tę wykonuje się jed-

nak jednorazowo i nie jest ona zbyt uciążliwa nawet dla osób nie posiadających innego programatora, gdyż najprostszy z nich można zlutować z kilku rezystorów. W Internecie można znaleźć opisy programatorów nie posiadających tej wady, wykorzystujących tryb Bit Bang układu FT232RL, ale wymagają one wprowadzania dodatkowych połączeń na czas programowania. Wiąże się to także z koniecznością otwarcia obudowy, a w dodatku samo programowanie za pośrednictwem FT232RL trwa kilkanaście minut.

Programowany układ jest dołączany do mikrokontrolera sterującego za pośrednictwem buforów zrealizowanych na układzie 74HC125. Bufory te, oprócz dołączania i odłączania linii ISP, zapewniają dopasowania poziomów napięć. Układ U3 jest zawsze zasilany napięciem równym napięciu zasilania programowanego systemu, a w przypadku braku tego zasilania, energia niezbędna do pracy jest dostarczana od strony programatora za pośrednictwem diody D1 i rezystora R13. Rezystory włączone w linie sygnałowe zabezpieczają bufor przed przeciążeniem i wymuszają stan ustalony w przypadku przełączenia wyjść w stan wysokiej impedancji. Złącze J3 służy do dołączenia przewodu taśmowego połączonego z takim samym złączem w programowanym systemie (zgodnie ze standardem Atmela).

Montaż i uruchomienie

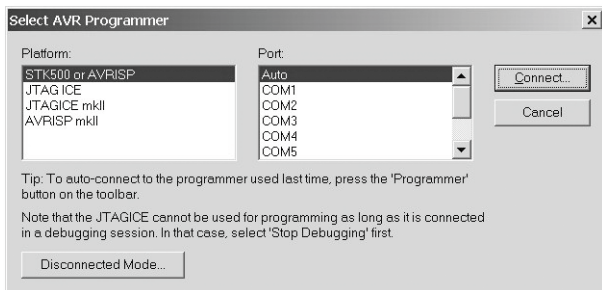
Dla programatora przewidziano dwustronną płytkę drukowaną, jednostronnie obłożoną elementami SMD i kilkoma podzespołami przewlekany. Schemat montażowy przedstawiono na rys. 2. Montaż należy przeprowadzić z precyzją i starannością należną elementom SMD o gęstym rasterze. Płytkę drukowaną została zaprojektowana z myślą o umieszczeniu jej w półprzezroczystej obudowie Z-24Ap firmy Kradex. Jako złącze J3 najlepiej wykorzystać kątową listwę goldpinów, co po przygotowaniu otworu w ścianie obudowy umożliwi swobodne dołączanie i odłączanie przewodu ISP.

Przed pierwszym podłączeniem zasilania należy sprawdzić poprawność lutowania i usunąć ewentualne zwarcia między nóżkami układów scalonych, które ze względu na mały raster, zwłaszcza dla U1, mogą być trudne do zauważenia. Zmontowanego programatora nie należy od razu podłączać do komputera ze względu na niebez-

pieczeństwo uszkodzenia PC w przypadku błędu w montażu. Uruchomienie należy przeprowadzić podłączając jako zasilanie zasilacz stabilizowany 5 V z ograniczeniem prądowym nie większym niż 100 mA. Uruchomienie sprowadza się w zasadzie do sprawdzenia, czy napięcie zasilania układu U1, U2 i U3 są prawidłowe. Napięcia zasilania dla U1 i U2 powinny wynosić 5 V, natomiast układ U3 będzie zasilany napięciem o wartości obniżonej przez diodę D1 i R13. Oczywiście aby urządzenie mogło działać, należy zaprogramować mikrokontroler U2. Skompilowany program zamieszczono w pliku *avriscusb.hex* na stronie internetowej Elektroniki Praktycznej. Nie należy zapomnieć o takim ustawieniu fusebitów, aby procesor był taktowany dołączonym kwarcem (np. CKSEL=1111, SUT=11). Korzystne jest również włączenie watchdoga (jego odświeżanie jest realizowane przez program), a także detektora zaniku napięcia zasilania na poziomie 4 V.

Przed podłączeniem programatora do PC, należy pobrać ze strony www.ftdichip.com sterowniki do układu FT232RL. Proces instalacji sterownika nie różni się od opisu podanego przez producenta tego elementu. Po podłączeniu i wykryciu przez system Windows nowego urządzenia, należy wskazać pobrany wcześniej sterownik. Po poprawnym przeprowadzeniu instalacji w Menedżerze Urządzeń powinien pojawić się nowy port COM, którego numer warto zapamiętać.

Kolejnym krokiem jest sprawdzenie współpracy z AVR Studio. W tym celu należy uruchomić to środowisko i wybrać z menu polecenie Tools/Program AVR/Connect... Pojawi się wówczas okno przedstawione na rys. 3. Należy wybrać programator „STK500 or AVRISP” oraz port COM, który został przydzielony przy instalacji sterownika. Można również zaznaczyć opcję „Auto”, która automatycznie przeskanuje wszystkie porty COM w poszukiwaniu programatora. Aby zatwierdzić ustawienia należy wcisnąć przycisk „Connect...”. Jeżeli programator działa poprawnie i zostanie wykryty przez AVR Studio, naszym oczom ukaże się okno przedstawione na rys. 4. W polu „Revision” pojawi się wersja oprogramowania programatora. Jest ona wpisana w oprogramowaniu na stałe, zgodnie z wersją oryginalnego oprogramowania Atmela (odpowiadająca najnowszej wersji AVR Studio) i nie należy próbować wykonywać upgrade'u

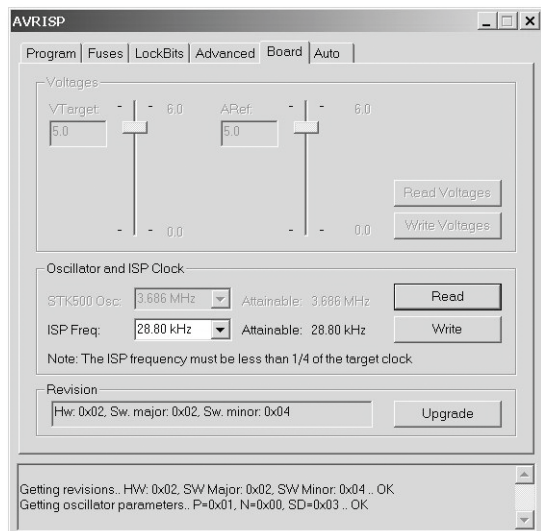


Rys. 3. Okno wyświetlane po prawidłowym skomunikowaniu się programatora z komputerem

przez wciskanie położonego obok przycisku. Prezentowany programator nie jest zgodny z oryginałem pod tym względem i operacja ta nie powiedzie się. Gotowość programatora do działania jest sygnalizowana świeceniem się zielonej diody LED1. Jeżeli uzyskanie połączenia z programatorem nie powiodło się, należy jeszcze raz sprawdzić montaż, poprawność zainstalowania sterownika FTDI i poprawność zaprogramowania mikrokontrolera.

Obsługa programatora

Jeżeli po uruchomieniu programatora pojawiło się okno z rys. 4, oznacza to, że oprogramowanie programatora funkcjonuje prawidłowo i jest on gotowy do pracy. Rozpoczynając od przedstawionej na rys. 4 zakładki, możemy przeprowadzić konfigurację programatora. Jedyнным parametrem przewidzianym do modyfikacji w tym miejscu jest parametr ISP Freq. Oznacza on częstotliwość zegarową interfejsu ISP przy programowaniu układu docelowego. Może być ona ustawiana na wartości: 921,6 kHz, 230,4 kHz, 57,6 kHz, 28,8 kHz, 4 kHz (przy odczycie



Rys. 4. Okno wyświetlane po prawidłowym wykręciu programatora

nastawy: 3,955 kHz) i 1,811 kHz. Aby uzyskać ostatnią częstotliwość, należy wybrać wartość 603 Hz, która niestety nie jest możliwa do uzyskania, ze względu na błąd w AVR Studio (zbyt krótki timeout, zjawisko występuje nawet na oryginalnym programatorze Atme-

la). W praktyce podane częstotliwości umożliwiają programowanie procesorów taktowanych zegarami począwszy od około 8 kHz, więc niemal we wszystkich użytecznych przypadkach. Należy pamiętać, że ustawienie częstotliwości zbyt dużej (większej niż jedna czwarta częstotliwości zegara) uniemożliwi zaprogramowanie procesora. Oczywiście im wyższa jest ustawiona częstotliwość, tym operacja programowania trwa krócej.

Zakładka „Program”, przedstawiona na rys. 5, umożliwia ustawienie głównych opcji programowania. Są to: typ procesora, nazwa pliku źródłowego dla programu i pamięci EEPROM oraz opcje dotyczące programowania i weryfikacji. Możliwe jest również przeprowadzenie procesu kasowania zawartości pamięci mikrokontrolera.

Zakładki „Fuses” (rys. 6) i „LockBits” (rys. 7) umożliwiają odczyt i konfigurację bitów konfiguracyjnych procesora. Trzeba pamiętać, że przy wchodzeniu na te zakładki następuje automatyczna próba odczytu aktualnych ustawień, więc bez do-

łączonego procesora docelowego będzie zgłaszany błąd wejścia w tryb programowania.

Zakładka „Advanced” (rys. 8) pozwala na kontrolę sygnatury dołączonego mikrokontrolera, jak również pozwala odczytać, zmodyfikować i zapisać w pamięci bajty kalibracyjne wewnętrznego oscylatora programowanego mikrokontrolera.

Najciekawszą z punktu widzenia wygody użytkownika jest zakładka „Auto”,

WYKAZ ELEMENTÓW

Rezystory

- R1...R7, R16: 470 Ω (SMD 0805)
- R8, R10, R11: 10 kΩ (SMD 0805)
- R9: 3,3 kΩ (SMD 0805)
- R12: 1 kΩ (SMD 0805)
- R13: 220 Ω (SMD 0805)
- R14, R15, R17, R18: 100 Ω (SMD 0805)
- R19: 10 Ω (SMD 0805)
- R20...R23: 470 kΩ (SMD 0805)

Kondensatory

- C1, C2, C4...C7: 100 nF (SMD 0805)
- C3, C8: 10 μF/16V (SMD Package A)
- C9, C10: 15 pF (SMD 0805)

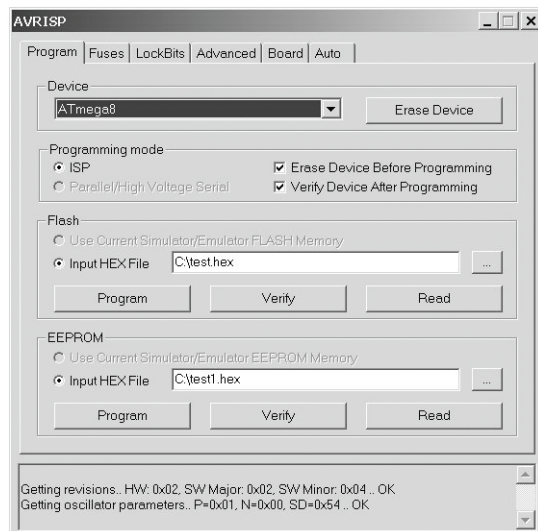
Półprzewodniki

- U1: FT232RL (SSOP-28)
- U2: ATmega8 (TQFP-32) (programowany przy uruchamianiu)
- U3: 74HC125 (SO-14)
- D1: BAS85 (MINIMELF)
- LED1: LED 3 mm zielona
- LED2: LED 3 mm czerwona

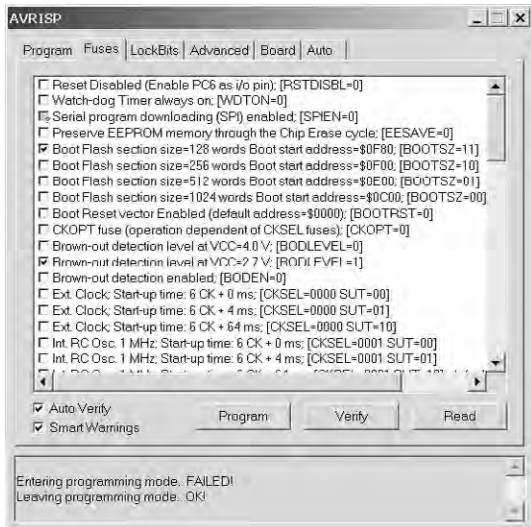
Inne

- X1: kwarc 3,6864 Mhz (HC49S)
- L1: dławik przeciwzakłóceńowy 1 μH (SMD 0805)
- J1: gniazdo MINI-USB-B
- J2: goldpin 2x3, 2,54 mm, prosty
- J3: goldpin 2x3, 2,54 mm, kątowny

przedstawiona na rys. 9. Umożliwia ona wybranie czynności, które będą wykonywane podczas automatycznego programowania z AVR Studio. Aby skorzystać z tej możliwości, należy zaznaczyć pożądane opcje, a następnie zminimalizować okno AVRISP (nie należy go zamykać!). Spowoduje to uaktywnienie na głównym pasku narzędzi ikony opisanej



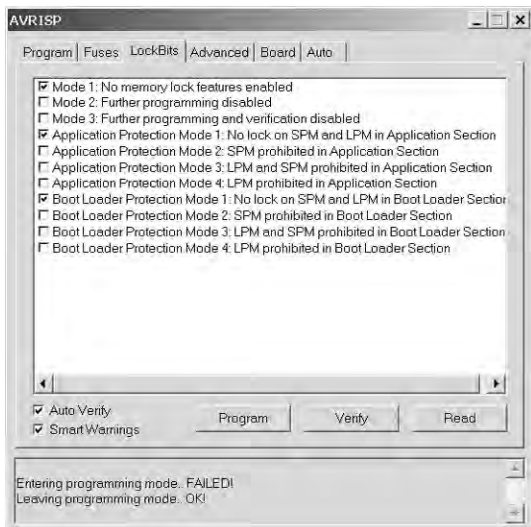
Rys. 5. Okno zakładki „Program” umożliwiające ustawienie głównych opcji programowania



Rys. 6. Zakładka „Fuses” umożliwiająca odczyt i modyfikację bitów konfiguracyjnych procesora

czewonym napisem „Auto”, której wciśnięcie spowoduje automatyczne wykonanie wszystkich zaprogramowanych wcześniej operacji, zgodnie z ustawieniami z pozostałych zakładek aplikacji obsługi programatora.

Programowanie procesorów AVR z wykorzystaniem opisanego programatora jest również możliwe z wykorzystaniem dostarczonej wraz ze środowiskiem AVR Studio aplikacji STK500.EXE. Jest to program tekstowy, wywoływany z linii poleceń, dający możliwość programowania procesorów przez jego wywołanie z innych aplikacji, np. dowolnych środowisk programistycznych. Listę dostępnych poleceń można wyświetlić poprzez wywołanie STK500.EXE -h.



Rys. 7. Zakładka „LockBits” umożliwiająca odczyt i modyfikację bitów konfiguracyjnych procesora

Przeprowadzanie operacji programowania polega na dołączeniu interfejsu ISP programatora z bliźniaczym złączem w systemie docelowym i odpowiednim skonfigurowaniu oprogramowania, w zależności od potrzeb. Podczas programowania świeci się czerwona dioda LED2, nie należy wówczas odłączać programatora od programowanego układu. Jeżeli układ docelowy nie chce się programować (zgłaszany jest błąd wejścia w tryb

programowania), należy przede wszystkim sprawdzić poprawność połączeń pomiędzy programatorem a układem programowanym. Problem może również leżeć po stronie ustawionej zbyt wysokiej prędkości interfejsu ISP, niedostosowanej do prędkości zegarowej mikrokontrolera. W tym miejscu należy pamiętać, że fabrycznie nowe mikrokontrolery są standardowo skonfigurowane na takowanie stosunkowo wolnym oscylatorem wewnętrznym, którego częstotliwość (najczęściej 1 MHz) jest za mała do obsługi maksymalnych prędkości programowania.

Błąd programowania może również wystąpić przy braku zasilania systemu docelowego. Jeżeli napięcie zasilające bufor U3 spadnie poniżej wartości około 2,5 V, wówczas

zielona dioda LED będzie migać. Nie powoduje to jednak blokady pracy programatora, lecz sygnalizuje jedynie, że ewentualny błąd programowania może być związany ze zbyt niskim napięciem zasilania systemu docelowego. Praktyka pokazała, że programator poprawnie programuje mikrokontrolery już przy napięciu około 1,4 V (testowano na ATtiny2313).

Przy pomocy opisywanego programatora można również programować procesory

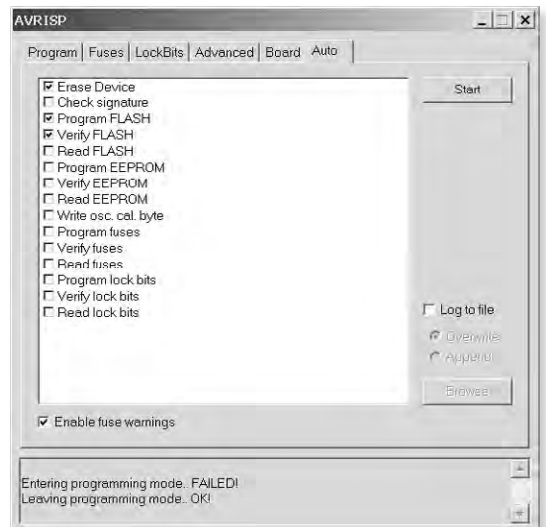


Rys. 8. Zakładka „Advanced” umożliwiająca sprawdzanie sygnatur mikrokontrolera

nie wlotowane do układu. W tym celu należy przygotować odpowiednie podstawki z przyłączem dla interfejsu ISP. Nie jest konieczne zewnętrzne zasilanie programowanego w ten sposób procesora, gdyż linia VCC, normalnie pobierająca energię z programowanego systemu, może oddać niewielką moc wystarczającą do zasilania układu podczas programowania.

Opisywany programator z pewnością przyspieszy przygotowywanie aplikacji za pomocą środowiska AVR Studio oraz ułatwi życie osobom wykorzystującym komputery pozbawione portów komunikacyjnych innych niż USB.

Paweł Hadam, EP
pawel.hadam@ep.com.pl



Rys. 9. Zakładka „Advanced” służąca do sprawdzania sygnatur dołączonego mikrokontrolera oraz modyfikowania bajtów kalibracyjnych wewnętrznego oscylatora mikrokontrolera