

UsbAsp

Programator mikrokontrolerów AVR

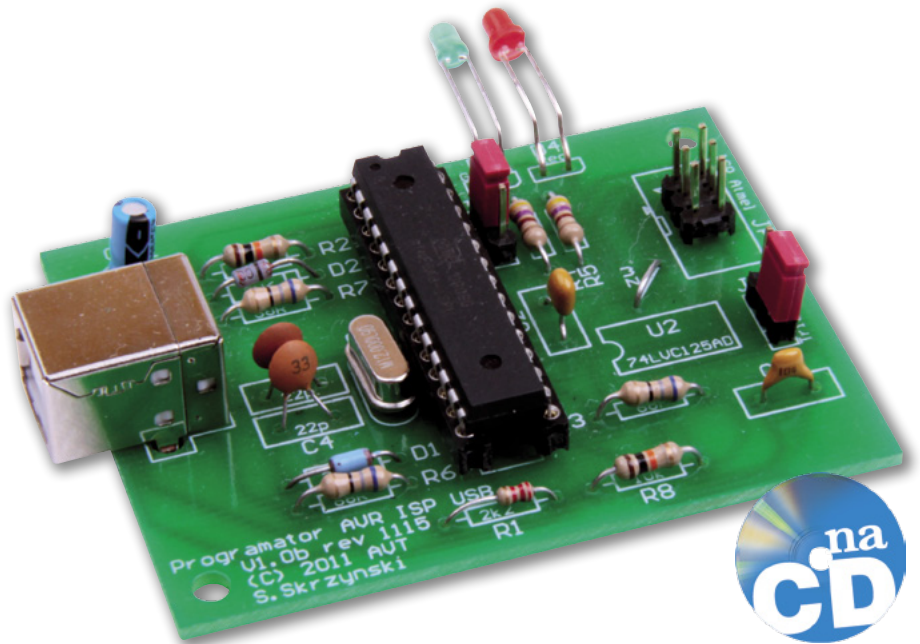
Opisów programatorów ISP dla mikrokontrolerów AVR jest w Internecie bez liku. Ale takich, które są wyposażone w interfejs USB już mniej, a jeśli, to najczęściej są one oparte o konwerter FT232. W artykule opisano programator, który nie wymaga użycia konwertera USB. Programator może być używany pod kontrolą systemów operacyjnych Linux i Windows, ze środowiskami programistycznymi AvrStudio, BascomAVR, CodeVisionAVR i innymi.

Rekomendacje: programator ma interfejs USB, więc doskonale sprawdzi się po zmianie komputera na nowszy, ponieważ zazwyczaj są one wyposażone tylko w ten rodzaj interfejsu.

Opisywany programator jest wzorowany na projekcie opracowanym przez <http://www.fischl.de/usbaspl/>. W porównaniu z oryginałem dodano bufor 74LVC125, dzięki czemu można programować układy zasilane napięciem z zakresu 1,2...6,5 V.

Budowa programatora jest nieskomplikowana. Zawiera on mikrokontroler ATmega8 oraz kilka dodatkowych elementów. Istotną zaletą programatora jest, że ma on interfejs USB, dzięki czemu można go bez problemu używać go na laptopie bez interfejsów LPT czy RS232C. Dostępne jest oprogramowanie interfejsu użytkownika zarówno dla Windows jak i Linux.

Transmisją sygnałów po USB zajmuje się mikrokontroler, nie jest wymagany dodatkowy, stosunkowo drogi konwerter USB. Szybkość programowania oscyluje o okolicach 5 kB/s, w razie potrzeby można ją zmniejszyć. Programator pobiera prąd zasilający mniejszy niż 100 mA, dzięki czemu może być zasilany również z wyjścia pasywnego (bez dodatkowego zasilacza) HUB'a USB. Płytkę drukowaną programatora została przystosowana do obudowy typu Z-70.



Budowa i zasada działania:

Schemat ideowy programatora pokazano na **rysunku 1**. Jego sercem jest mikrokontroler U1 – ATmega8. Emuluje on programowo interfejs USB, dzięki czemu nie jest potrzebny dodatkowy konwerter USB<->RS232 (np. FT232RL), co obniża cenę gotowego urządzenia. Ze względu na to, że mikrokontroler jest zasilany napięciem 5 V z portu USB, zastosowano diody Zenera D1 i D2 dla obniżenia maksymalnego napięcia występującego na magistrali USB (brak diod spowoduje pojawianie się błędów SYNC).

Napięcia zasilania jest filtrowane przez C1, C2.

Diody LED sygnalizują stan programatora:

- D3 (czerwona) przyłączenie do USB,
- D4 (zielona) operacje na programowanym procesorze.

Układ U2 powinien być zasilany napięciem zawierającym się w przedziale 1,2...3,6 V, ponieważ w takim zakresie napięcia zasilania producent gwarantuje poprawną pracę układu. W układach produkowanych przez NXP napięcie zasilające do 6,5 V nie spowoduje jego zniszczenia. Maksymalna wartość napięcia zasilającego zależy od producenta układu. Zaleca się więc pracę z programatorem przy zasilaniu programowanego CPU napięciami 1,2...3,6 V. Aby programować układy zasilane napięciem 2...6 V zaleca się wymienić układ U2 na 74HC125.

Programator jest odseparowany od programowanego procesora buforem U2 typu 74LVC125AD. Bufor jest zasilany z systemu

AVT-5325 w ofercie AVT:
AVT-5325A – płytkę drukowaną
AVT-5325B – płytkę drukowaną + elementy

Podstawowe informacje:

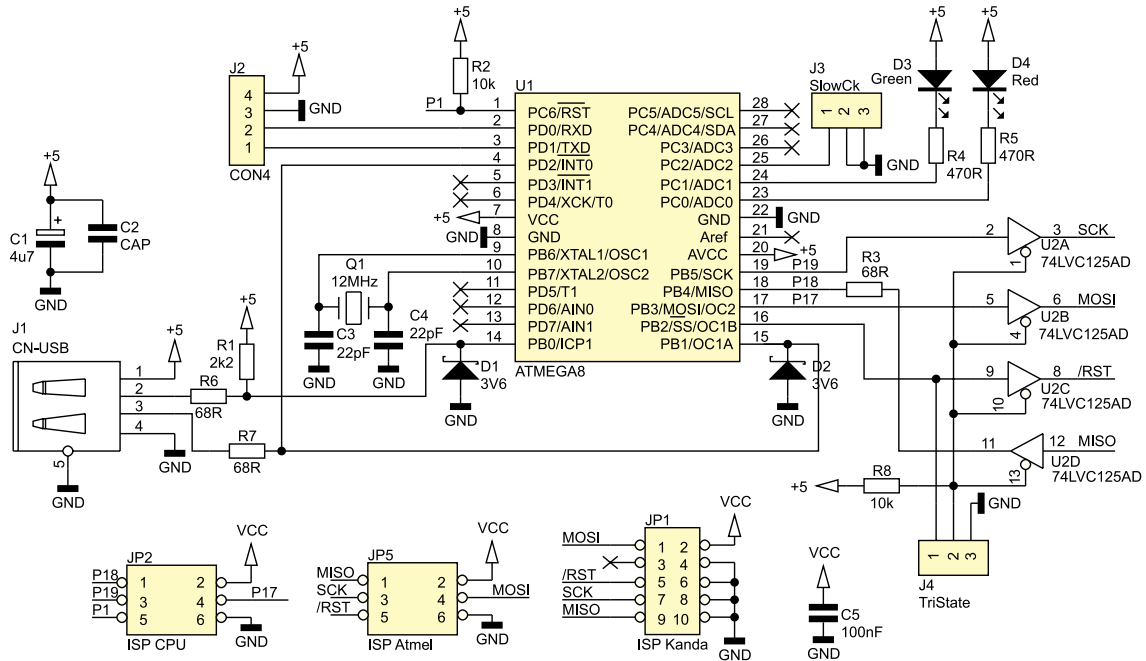
- Zasilanie programatora z portu USB.
- Napięcie zasilające programowanego mikrokontrolera: 1,2...6 V.
- Lista obsługiwanych mikrokontrolerów AVR: ATmega6450, ATmega3250, ATmega645, ATmega325, ATmega2561, ATmega2560, ATmega1281, ATmega1280, ATmega640, ATmega168, ATmega88, ATmega48, ATmega8535, ATmega8515, ATmega8, ATmega161, ATmega32, ATmega6490, ATmega649, ATmega3290P, ATmega3290, ATmega329P, ATmega329, ATmega169, ATmega163, ATmega162, ATmega644P, ATmega644, ATmega324P, ATmega164P, ATmega16, ATmega128, ATmega64, ATmega103, ATtiny15, ATtiny13, ATtiny12, ATtiny11, ATtiny2313, ATtiny84, ATtiny44, ATtiny24, ATtiny85, ATtiny45, ATtiny25, ATtiny861, ATtiny461, ATtiny261, ATtiny26.
- Współpraca ze środowiskami programistycznymi AvrStudio, BascomAVR, CodeVisionAVR itp.
- Praca pod kontrolą systematów: Windows (XP/Vista), Linux, MacOS X, FreeBSD.

Dodatkowe materiały na CD/FTP:
<ftp://ep.com.pl>, user: 17692, pass: 4yv87ftn

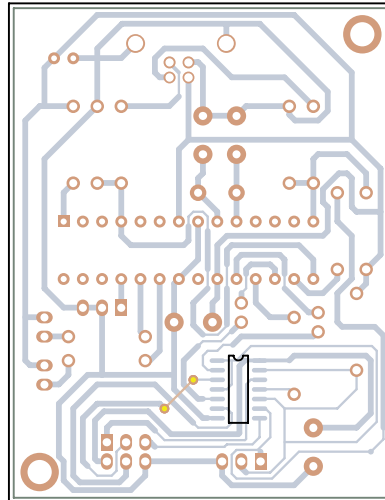
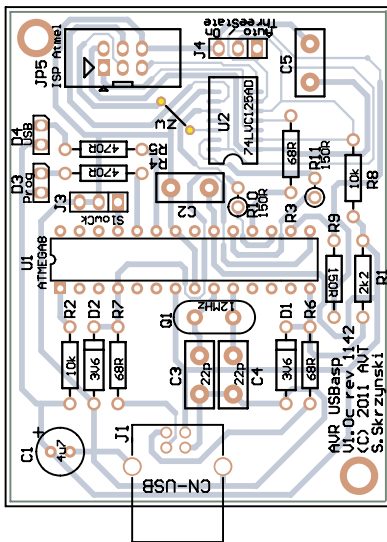
- wzory płytek PCB
- karty katalogowe i noty aplikacyjne elementów oznaczonych w Wykazie elementów kolorem czerwonym

Projekty pokrewne na CD/FTP:
(wymienione artykuły są w całości dostępne na CD)

- AVT-5279 Programator PIC (EP 2/2011)
- AVT-5172 Uniwersalny programator mikrokontrolerów AVR (EP 2/2009)
- AVT-5153 Uniwersalny programator JTAG/ISP (EP 10/2008)
- AVT-5125 Programator USB AVR (STK500) (EP 2/2008)
- AVT-1462 Uniwersalny adapter dla programatorów AVR-ISP (EP 2/2008)
- AVT-2855 Ulepszony programator STK200 (EdW 2/2008)
- AVT-988 Programator AVRISP z interfejsem USB (STK500) (EP 7/2007)



Rysunek 1. Schemat ideowy programatora AVR z interfejsem USB



Rysunek 2. Schemat montażowy programatora AVR z interfejsem USB

Wykaz elementów

Rezystory: (0,125 W)
 R1: 2,2 kΩ
 R2, R8: 10 kΩ
 R3, R6, R7: 68 Ω
 R4, R5: 470 Ω

Kondensatory:
 C1: 4,7 μF/25 V
 C2: 100 nF
 C3, C4: 22 pF
 C5: 100 nF

Półprzewodniki:
 D1, D2: dioda Zenera 3,6 V
 D3: dioda LED 3 mm zielona
 D4: dioda LED 3 mm czerwona
 U2: 74LVC125AD (SO-16)
 U1: ATmega8A-PU (DIP-28)

Inne:
 Q1: rezonator kwarcowy 12 MHz
 J1: gniazdo USB, kątowne do druku
 JP5: gniazdo IDC6 lub goldpin 2×3
 J3, J4: goldpin 1×3 + zworka

programowanego za pośrednictwem złącza JP5. Napięcie zasilające bufor jest filtrowane przez C5. Charakterystyczną cechą U2 jest to, że napięcia na jego wejściach mogą osiągać wartość 5,5 V, choć sam układ może być zasilany napięciem dużo niższym. Dzięki temu nie ma problemu z programowaniem mikrokontrolerów zasilanych napięciem 3,3 V i niższymi. Gdyby zamiast układu LVC zastosować 74HC125, wtedy to rezystory R9...11 zabezpiecza U2 przed zniszczeniem, gdyby był on zasilany napięciem niższym niż 4,4 V. Rezystor R3 zabezpiecza U1 w wypadku, gdy U2 jest zasilany napięciem 6 V.

Zworkę J4 można ustawić w jednej z dwóch pozycji. Gdy J4 (*ThreeState*) ma zwarte wyprowadzenia 2-3 (pozycja *ON*) bufor jest cały czas przyłączony do programowanego mikrokontrolera. Gdy zwarte są wyprowadzenia 1-2 (pozycja *AUTO*), wówczas bufor jest przyłączany tylko na czas operacji na mikro-

kontrolerze (programowanie, weryfikacja, odczyt). Zworka J3 umożliwia współpracę z procesorami AVR taktowanymi sygnałem zegarowym o częstotliwości niższej niż 1,5 MHz. Zwarcie wyprowadzeń 1-2 zmniejsza zegar częstotliwość taktowania interfejsu SPI, zwarcie 2-3 umożliwia pracę z pełną prędkością.

Montaż i uruchomienie:

Schemat montażowy programatora umieszczono na **rysunku 2**. Montaż rozpoczynamy od wlutowania układu U2 od strony miedzi. Później montujemy elementy od najmniejszego do największego. Pod U1 warto zastosować podstawkę precyzyjną. U1 należy zaprogramować przy użyciu zewnętrznego programatora plikiem *usbasp.atmega8.2011-05-28.hex*. Ustawienie bitów konfiguracyjnych (*fuse bitów*) jest następujące („1” oznacza bit niezaprogramowany, tak jak w kartach katalogowych):

Fuse high byte: *Oxc9* (*szesnastkowo*)
RSTDISBL = 1
WDTON = 1
SPIEN = 0
CKOPT = 0

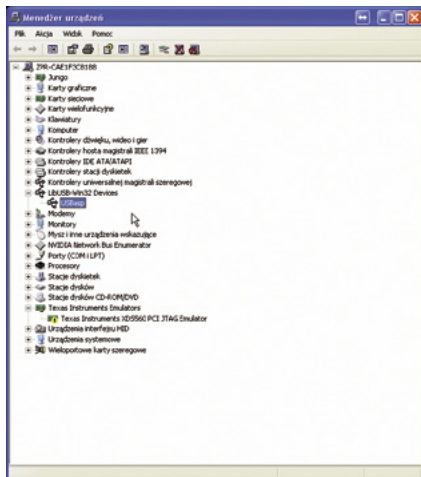
REKLAMA

WWW.STM32.EU

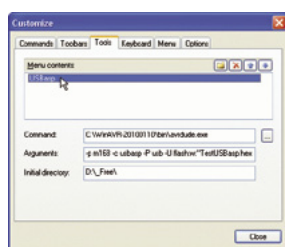
U nas znajdziesz wszystko na temat...

STM32 F4



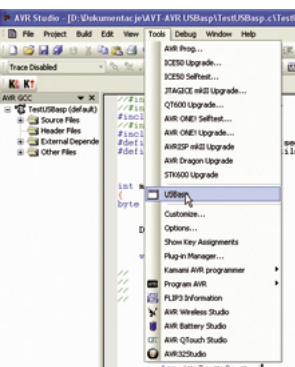


Rysunek 3. Okno menedżera urządzeń z zainstalowanym programatorem

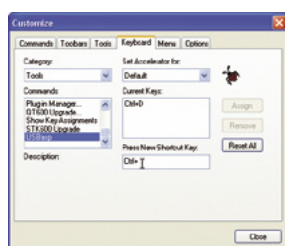


Rysunek 4. Wpisanie nazwy programatora w AVR Studio

```
EESAVE = 1
BOOTSZ1 = 0
BOOTSZ0 = 0
BOOTRST = 1
Fuse low byte: 0x9f (szesnastkowo)
DODLEVEL = 1
BODEN = 0
SUT 1..0 = 01
CKSEL 3..0 = 1111
```



Rysunek 5. Wybór programatora w AVR Studio



Rysunek 6. Przypisanie klawisza skrótu w AVR Studio

Po umieszczeniu zaprogramowanego procesora w podstawce U1 i sprawdzeniu funkcjonowania programatora, płytkę można umieścić w obudowie Z-70.

Instalacja sterowników

Aby programator działał prawidłowo, w systemie Windows należy zainstalować sterowniki USB. Po podłączeniu USBasp do komputera system wykryje programator i uruchomi kreatora znajdowania nowego sprzętu. Wskazujemy sterownik (plik „usbasp.inf”) znajdujący się w pakiecie oprogramowania dostępnym na CD-EP12/2011 i serwerze ftp w katalogu *usbasp.2011-05-28/bin/win-driver/libusb_1.2.4.0*. Na pytanie o plik *usblib0.sys* wskazujemy folder *x86*. Po instalacji sterowników w menedżerze urządzeń pojawi się USBasp (rysunek 3). Teraz można zainstalować program *avrdude*.

W Linuksie instalacja jest łatwiejsza. Wystarczy zainstalowanie programu *avrdude* z repozytorium użytkowanego systemu Linux.

Obsługa

Przy korzystaniu z oprogramowania *avrdude* programator obsługuje się za pomocą linii komend. Poniżej kilka standardowych komend:

- Tylko odczyt sygnatury: (m8 to typ mikrokontrolera): *avrdude -p m8 -c usbasp*,
- Odczyt programu z pamięci mikrokontrolera do pliku hex: *avrdude -p m8 -c usbasp -U flash:r:program.hex*,
- Zapis programu z pliku hex do pamięci mikrokontrolera: *avrdude -p m8 -c usbasp -U flash:w:program.hex*,

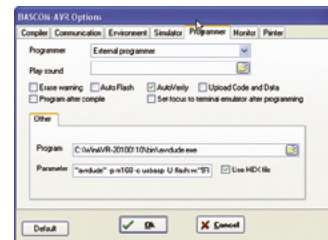
Pozostałe rozkazy możemy poznać wpisując w linii komend *avrdude* bez parametrów (wyświetli się pomoc).

Obsługa programatora z linii komend nie jest zbyt wygodna. W systemie Windows obsługę ułatwia program *AVR Burn-O-Mat*. Oczywiście, jego użytkowanie jest opcjonalne. Po zainstalowaniu *AVR Burn-O-Mat* w ustawieniach (menu *Settings*) wybieramy port USB. Obsługa programu jest intuicyjna i nie wymaga szerszego omawiania.

Dzięki temu, że program obsługuje się z linii komend, bez problemu może on współpracować z różnymi środowiskami programistycznymi takimi jak *AvrStudio*, *BascomAVR*, *CodeVisionAVR*, *Keil* itp. Dla przykładu omówiono konfigurację popularnych środowisk programistycznych *AVR Studio* i *Bascom AVR*.

W *AVR Studio*:

- Z menu *Tools* wybieramy *Customize...*
- Otworzy się okno *Command*, w którym wybieramy zakładkę *Tools*.
- Wskazujemy ikonkę *New* lub wciskamy klawisz *Insert*.
- Wpisujemy nazwę programatora, np. *USBasp* (rysunek 4).
- W oknie *Command* wskazujemy ścieżkę do *avrdude* (np.: *C:\WinAVR-20100110\bin\avrdude.exe*).



Rysunek 7. Wybór programatora zewnętrznego w Bascom AVR

- W oknie *arguments* podajemy parametry: *-p m168 -c usbasp -P usb -U flash:w:"plik.hex":a -U flash:v:"plik.hex":a*, gdzie m168 to typ mikrokontrolera.
- W *Initial directory* wpisujemy ścieżkę dostępu do pliku (trzeba pamiętać o końcowym znaku „\”)
- Zmiany zatwierdzamy przyciskiem *Close*.
- W celu zaprogramowania mikrokontrolera wybieramy w menu *Tools* nazwę naszego programatora (w przykładzie jest to *USBasp* (rysunek 5)).

Do funkcji programowania można przypisać klawisz skrótu (rysunek 6).

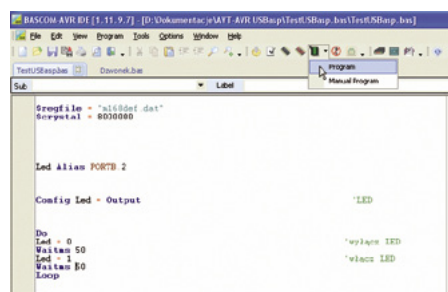
- W *Bascom AVR*:
- Z menu wybieramy *Options* → *Programmer*.
- Z listy rozwijanej wybieramy *External programmer* (rysunek 7).
- W zakładce *Other* (w dolnej części okna) podajemy ścieżkę do *avrdude*.
- W oknie parametrów wpisujemy „*avrdude*” *-p m168 -c usbasp -U flash:w:"{FILE}" :a -U flash:v:"{FILE}" :a* oraz zaznaczamy opcję *Use HEX file*, gdzie m168 to typ mikrokontrolera.
- Programowanie wywołujemy klawiszem *F4* lub ikonką *Program chip* (rysunek 8).

Przykładowe typy procesorów po parametrze „-p”: m8 (ATmega8), m32 (ATmega32), m128 (ATmega128), m168 (ATmega168), t85 (ATtiny85), t2313 (ATtiny2313), 8515 (AT90S8515), 1200 (AT90S1200).

Sławomir Skrzyński, EP

Bibliografia:

- http://mirley.firlej.org/programator_na_usb_usbasp
- <http://www.elektroda.pl/rtvforum/topic1784303.html>
- http://avrside.fr/pl/pol/ind_pol.html



Rysunek 8. Uruchomienie programowania w Bascom AVR