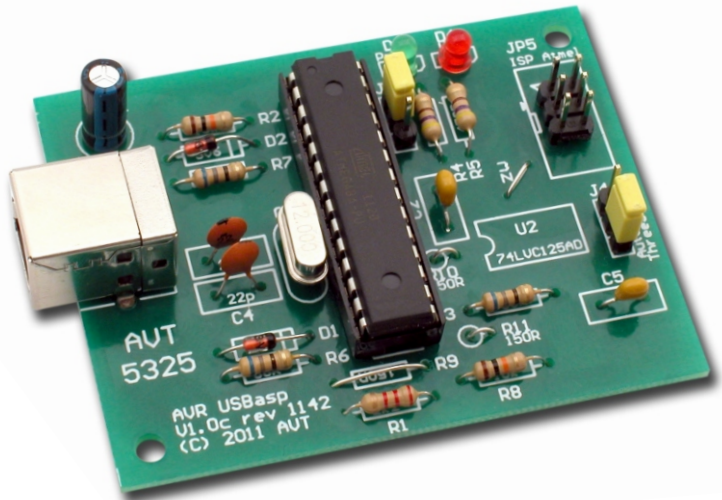


AVT 5325

UsbAsp Programator mikrokontrolerów AVR

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.




Właściwości

- zasilanie programatora z portu USB.
- napięcie zasilające programowanego mikrokontrolera: 3...5 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.

Do pobrania

 instrukcja pdf: <http://serwis.avt.pl/manuals/AVT5325.pdf>

 sterowniki i oprogramowanie (generator, strona internetowa): <http://serwis.avt.pl/files/AVT5325.zip>

Opis układu

Programator jest wzorowany na projekcie opracowanym przez <http://www.fischl.de/usbasp/>. 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.

Transmisja 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. Płytką drukowaną programatora została przystosowana do obudowy typu Z-70.

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. 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

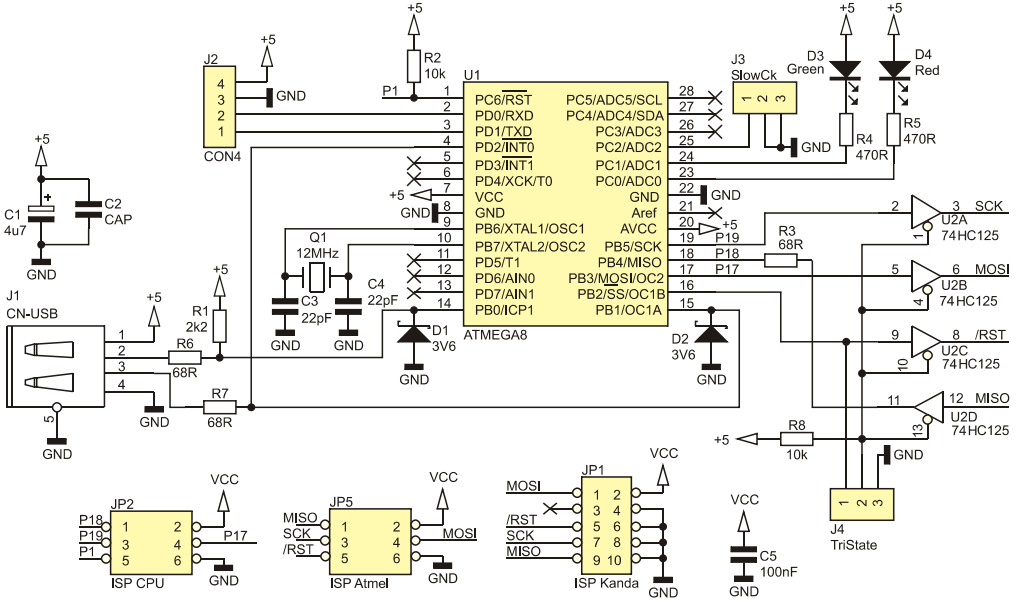
magistrali USB (brak diod spowoduje pojawianie się błędów SYNC). Napięcia zasilania jest filtrowane przez C1, C2.

Diody LED sygnalizują stan programatora:

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

Programator jest odseparowany od programowanego procesora buforem U2 typu 74HC125. Bufor jest zasilany z systemu 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. Rezystor R3 zabezpiecza U1 w wypadku, gdy U2 jest zasilany napięciem 6 V.

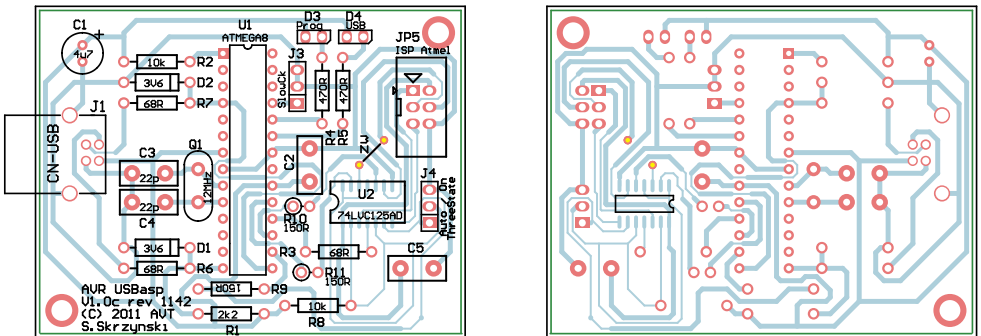
Zworkę J4 można ustawić w jednej z dwóch pozycji. Gdy J4 (ThreeState) ma zwarte wyprowadzenia 2-3 (pozycja ON) bufor jest cały czas przyłączony do programowanego mikrokontrolera. Gdy zwarte są wyprowadzenia 1-2 (pozycja AUTO), wówczas bufor jest przyłączany tylko na czas operacji na mikrokontrolerze (programowanie, weryfikacja, odczyt). Zworka J3 umożliwiła 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.



Rys. 1 Schemat ideowy programatora AVR z interfejsem USB

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.



Rys. 2 Schemat montażowy programatora AVR z interfejsem USB

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 tutaj: <http://serwis.avt.pl/files/AVT5325.zip> 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

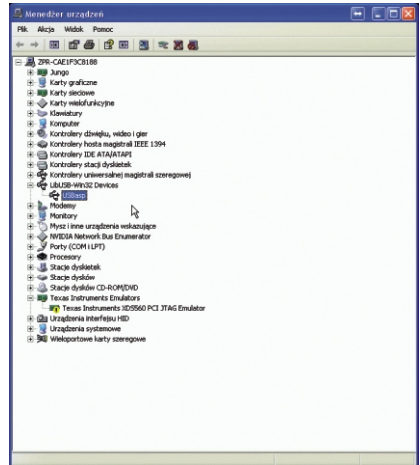
Przy korzystaniu z oprogramowania avrdude programator obsługuje się za pomocą linii komend. Poniżej kilka standardowych komend:

- 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`,

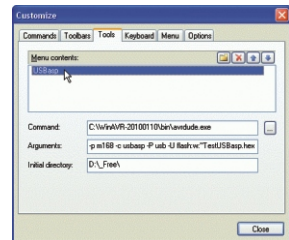
Pozostałe rozkazy możemy poznać wpisując w linii komend *avrdude* bez parametrów (wyświetli się pomoc). Obsługa programatora z linii komend nie jest zbyt wygodna. W systemie Windows obsługę ułatwia program *AVR Burn-O-Mat*. Oczywiście, jego użytkowanie jest opcjonalne. Po zainstalowaniu *AVR Burn-O-Mat* 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 *BascomAVR*.

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*)
- 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 skrót (**rysunek 6**).



Rys. 3 Okno menedżera urządzeń z zainstalowanym programatorem

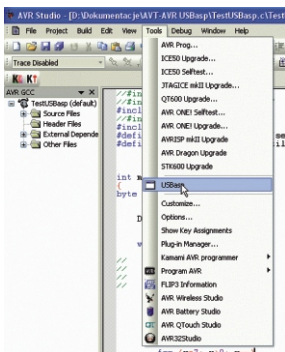


Rys. 4

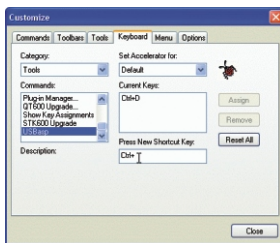
W Bascom AVR:

- 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**).

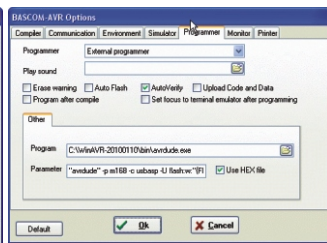
Przykładowe typy procesorów po parametrze „-p”: m8 (ATmega8), m32 (ATmega32), m128 (ATmega128), m168 (ATmega168), t85 (ATtiny85), t2313 (ATtiny2313), 8515 (AT90S8515), 1200 (AT90S1200).



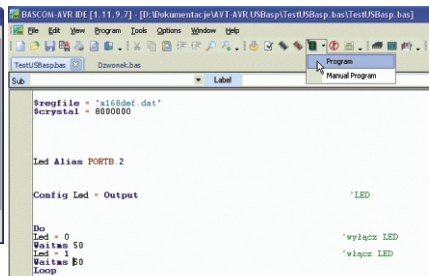
Rys. 5



Rys. 6



Rys. 7



Rys. 8

Wykaz elementów

Rezystory: (0,125 W)

R1:2,2 kΩ
 R2, R8:10 kΩ
 R3, R6, R7:68 Ω
 R4, R5:470 Ω
 R9, R10, R11ZWORA

Kondensatory:

C1:4,7 uF/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
 U2:74HC125 (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



AVT Korporacja sp. z o.o.

ul. Leszczynowa 11
 03-197 Warszawa
 tel.: 22 257 84 50
 fax: 22 257 84 55
 www.sklep.avt.pl



Dział pomocy technicznej:
 tel.: 22 257 84 58
 serwis@avt.pl



Produktu nie wolno wyrzucać do zwykłych pojemników na odpady. Obowiązkiem użytkownika jest przekazanie zużytego sprzętu do wyznaczonego punktu zbiórki w celu recyklingu odpadów powstających ze sprzętu elektrycznego i elektronicznego.

AVT Korporacja zastrzega sobie prawo do wprowadzania zmian bez uprzedniego powiadomienia.

Montaż i podłączenie urządzenia niezgodny z instrukcją, samowolna zmiana części składowych oraz jakiegokolwiek przeróbki konstrukcyjne mogą spowodować uszkodzenie urządzenia oraz narazić na szkodę osoby z niego korzystające. W takim przypadku producent i jego autoryzowani przedstawiciele nie ponosi odpowiedzialności za jakiegokolwiek szkody powstałe bezpośrednio lub pośrednio w wyniku użycia lub nieprawidłowego działania produktu.