

Programator USBasp

instrukcja obsługi

Tomasz Ostrowski
Aktualizacja: 2008.08.01

Spis treści

1. Informacje ogólne.....	3
2. Instalacja.....	3
3. Użytkowanie.....	5
3.1 Nakładki graficzne i programy niezależne.....	7
4. Rozwiązywanie problemów.....	8
4.1 Problemy z uruchomieniem USBasp.....	8
4.2 Problemy z programowaniem układu docelowego.....	8
Załączniki.....	10

1. Informacje ogólne

Autorem projektu USBasp jest Thomas Fischl. Kod źródłowy programatora wykorzystuje dostępną na zasadach GPL lub komercyjnych bibliotekę software'owego interfejsu USB firmy Objective Development. Licencja projektu znajduje się na stronie wyżej wymienionej firmy.

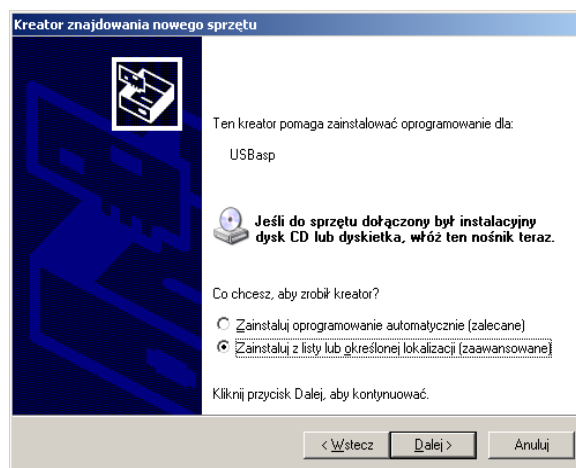
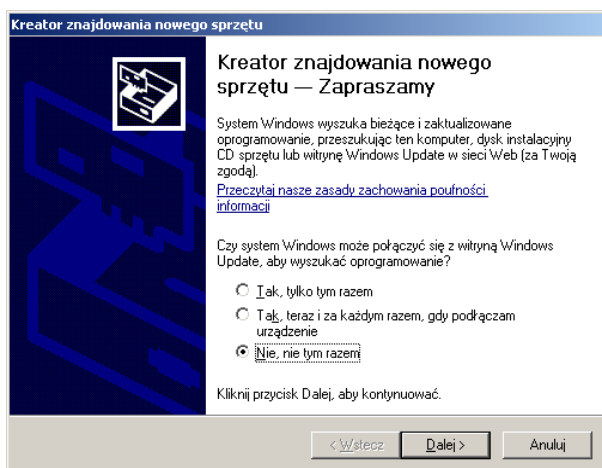
2. Instalacja.

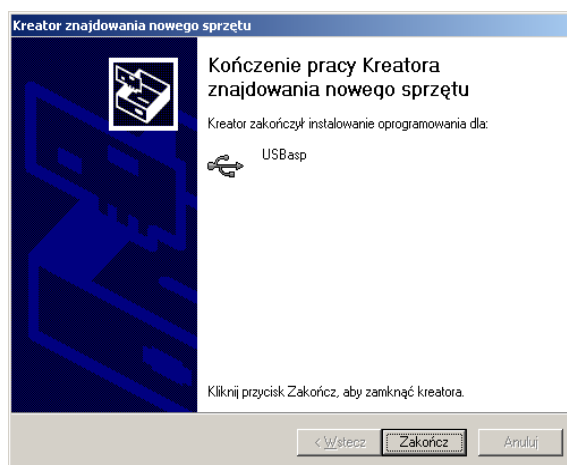
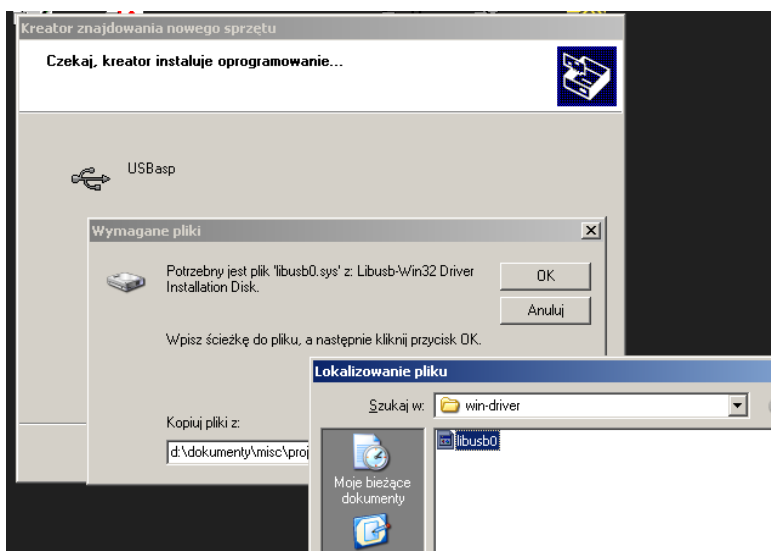
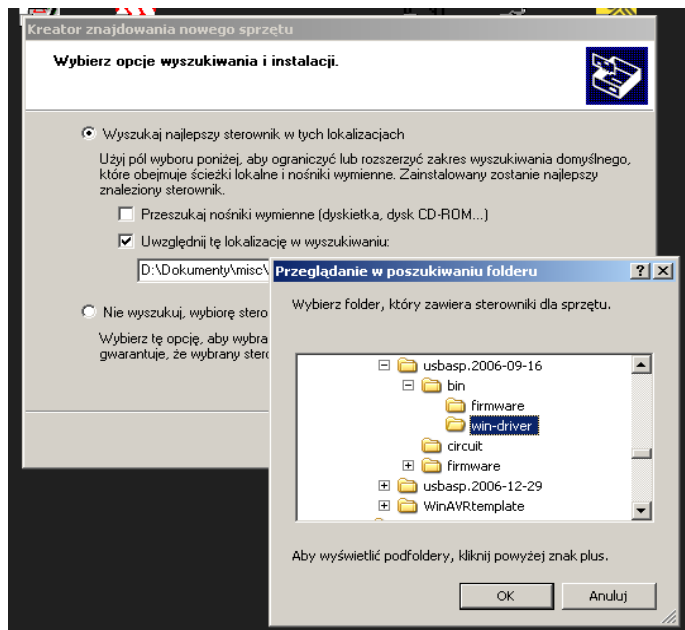
Do podłączenia programatora z PC wykorzystywany jest przewód USB A-B (często spotykany przy drukarkach). Zalecane jest wykorzystanie przewodu o długości 1.8m lub mniejszej i nie korzystanie z gniazd na przednim panelu komputera połączonych z płytą główną dodatkowym przewodem.

Połączenie z układem programowanym zapewnia taśma zakończona z obu stron wtykami IDC-10. Przy podłączaniu programatora do układu programowanego istotny jest sposób dołączenia wtyków: skrajny przewód taśmy o wyróżniającym się kolorze powinien zostać dołączony z obu stron do pinu nr 1 gniazda. Strona po której znajduje się pin 1 gniazda wyróżniona jest czarną kropką oraz cyfrą 1 po stronie druku.

Rozkład pinów złącza goldpinowego na płycie programatora odpowiada standardowi STK-200. Możliwe jest wykonanie dodatkowego przewodu o innym połączeniu pinów lub innym standardzie gniazda ISP który potraktować można jako przejściówkę.

Przy pierwszym podłączeniu programatora do komputera wykryty zostanie on jako nowe urządzenie. Konieczne jest wskazanie ścieżki do sterownika (katalog win-driver).





Po zainstalowaniu sterownika programator powinien być widoczny w menadżerze urządzeń jako

urządzenie o nazwie USBasp. Podłączenie go do innego portu USB niż poprzednio wywoła ponownie okno instalacji sterownika (jak zresztą przy każdym innym urządzeniu USB).

Aplikacją współpracującą ze starszą wersją firmware programatora jest patchowany przez Thomasa Fischla program `avrdude`. Nowsza wersja (*usbasp.2006-12-29.tar.gz*) obsługiwana jest przez „zwykły” `avrdude` w wersji **5.3.1**.

Wraz z `avrdude` dostarczany jest sterownik `giveio.sys` umożliwiający programowi dostęp do portu LPT w systemach Win2000/XP. Nie jest on wykorzystywany przez `usbasp`, ale jeżeli wykorzystywany będzie obok niego programator STK200/300 czy `bsd` warto zainstalować `giveio.sys` poprzez uruchomienie skryptu `install_giveio.bat`.

Najprostszym sposobem korzystania z niego jest wykorzystanie środowiska WinAVR wraz z Programmer's Notepadem. Instalacja polega na skopiowaniu plików z katalogu `bin` do katalogu `WinAVR/bin` i nadpisaniu plików poprzednio tam istniejących.

! Nowa wersja `avrdude` (5.3.1) jest niekompatybilna ze starszym firmware programatora i odwrotnie, użyj wersji zgodnych ze sobą. Niezgodność wersji objawia się komunikatem:
`avrdude: error: could not find USB device vendor=0x3eb product=0xc7b4`
lub podobnym, pomimo tego, że urządzenie jest widoczne w menedżerze urządzeń i zainstalowany został sterownik.

3. Użytkowanie.

Przeznaczenie zworek programatora:

- zworka 1 – aktualizacja firmware programatora,
- zworka 2 – zasilanie układu programowanego przez USBasp (lub ewentualne zasilanie USBasp przez programator przy wgrywaniu firmware do niego),
- zworka 3 – zmniejszenie częstotliwości pracy ISP, przydatne gdy programowany mikrokontroler taktowany jest z częstotliwością poniżej 1MHz.

Aby korzystać z programatora USBasp z poziomu Programmers Notepad'a lub podobnego IDE należy w pliku `makefile` projektu zadeklarować użycie tego programatora:

- znajdź i zakomentuj (dodaj znak `#` na początku) wiersz rozpoczynający się od

```
AVRDUDE_PROGRAMMER,
```

- dopisz wiersz:

```
AVRDUDE_PROGRAMMER = usbasp
```

- opcjonalnie: znajdź i zakomentuj wiersz

```
AVRDUDE_PORT = lpt1
```

dodaj:

```
AVRDUDE_PORT = usb
```

(wybór nie ma w rzeczywistości znaczenia, starsze wersje `avrdude` nie pozwalały jednak na użycie parametru `usb` a wymagały wprowadzenie w to miejsce jakiegokolwiek parametru)

- w zależności od potrzeby dodaj uzupełnij cykl programowania o programowanie pamięci `eeprom` i bajtów `fusebitów/lockbitów`:

```

AVRDUDE_WRITE_EEPROM = -U eeprom:w:eeeprom.hex
AVRDUDE_WRITE_LOCK = -U lock:w:0x3C:m
AVRDUDE_WRITE_HFUSE = -U hfuse:w:0xC9:m
AVRDUDE_WRITE_LFUSE = -U lfuse:w:0x9F:m

```

Jeżeli któraś z tych opcji nie jest potrzebna, wskazane jest zakomentowanie związanej z nią wiersza.

Przykładowy zestaw fusebitów dla mikrokontrolera Atmega8 pracującego z zewnętrznym kwarcem 12MHz:

```

# Fuse high byte:
# 0xc9 = 1 1 0 0   1 0 0 1 <-- BOTRST (boot reset vector at 0x0000)
#         ^ ^ ^ ^   ^ ^ ^----- BOOTSZ0
#         | | | |   | +----- BOOTSZ1
#         | | | |   + ----- EESAVE (don't preserve EEPROM over chip erase)
#         | | | +----- CKOPT (full output swing)
#         | | +----- SPIEN (allow serial programming)
#         | +----- WDTON (WDT not always on)
#         +----- RSTDISBL (reset pin is enabled)
# Fuse low byte:
# 0x9f = 1 0 0 1   1 1 1 1
#         ^ ^ \ /   \---+---/
#         | | |       +----- CKSEL 3..0 (external >8M crystal)
#         | | +----- SUT 1..0 (crystal osc, BOD enabled)
#         | +----- BODEN (BrownOut Detector enabled)
#         +----- BODLEVEL (2.7V)

```

w pliku makefile powinna znaleźć się linia:

```

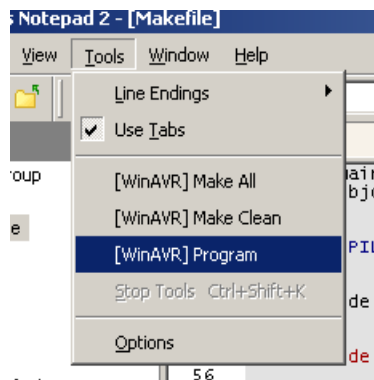
program: $(TARGET).hex $(TARGET).eep
(lub np. program: $(TARGET).hex eeprom.eep)

$(AVRDUDE) $(AVRDUDE_FLAGS) $(AVRDUDE_WRITE_FLASH) $(AVRDUDE_WRITE_EEPROM)
$(AVRDUDE_WRITE_HFUSE) $(AVRDUDE_WRITE_LFUSE) $(AVRDUDE_WRITE_LOCK)

```

! nieostrożność przy ustalaniu wartości fusebitów może zakończyć się zablokowaniem mikrokontrolera, niemożliwym do usunięcia bez wysokonapięciowego programatora równoległego; konieczne jest dokładne zapoznanie się z kartą katalogową

Po przeprowadzeniu powyższych operacji możliwe jest programowanie mikrokontrolera poprzez opcję Tools/Program Programmer's Notepad.



Alternatywne sposoby korzystania z avrdude jest użycie linii komend lub jednej z wielu nakładek graficznych.

3.1 Nakładki graficzne i programy niezależne

Lista nie jest kompletna.

Nazwa	Opis
AVR8 Burn-O-Mat	GPL, java, cross-platform http://www.brischalle.de/avr8_burn-o-mat_avrdude_gui/avr8_burn_o_mat_avrdude_gui_en.html
avrdude-GUI	BSD, C# http://yuki-lab.jp/hw/avrdude-GUI/index.html
Idnaf.AVRdude.FE	GPL, C# http://fandigunawan.wordpress.com/2007/12/28/idnafavrdudefe-free-avrdude-front-end/
gnome-avrdude	linux, GPL http://sourceforge.net/projects/gnome-avrdude/
avrdude-gui	GPL; win32, wxDevC++ = moim zdaniem ślepa uliczka http://sourceforge.net/projects/avrdude-gui/ http://tomeko.net/misc.php#USBasp
eXtreme Burner-AVR	wxWidgets, win32, obsługuje USBasp w sposób bezpośredni, nie korzystając z avrdude http://extremeelectronics.co.in/avr-tutorials/gui-software-for-usbasp-based-usb-avr-programmers/

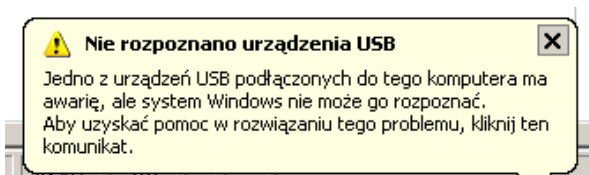
AVR fuse Calculator	delphi, win32, plik wykonywalny zawiera w sobie spakowane avrdude http://www.elektroda.pl/rtvforum/topic1354542.html
SinaProg	LabView, win32 http://www.microstar.ir/
AVRBurner	QT, win32 + linux, język interfejsu - niemiecki http://www.soft-land.de/index.php?page=avrburner

4. Rozwiązywanie problemów.

4.1 Problemy z uruchomieniem USBasp.

Programator można uznać za uruchomiony w momencie gdy po podłączeniu jest on widoczny na liście menedżera urządzeń i opisany jako USBasp. We właściwościach urządzenia (*Szczegóły*) powinny być widoczne numery VID i PID.

Komunikat:



pojawi się nawet gdy programator pozbawiony zostanie mikrokontrolera, nie informuje o jakimkolwiek stopniu poprawności działania.

W przypadku wystąpienia opisanego błędu oprócz sprawdzenia elektrycznego programatora i wgrania/zweryfikowania firmware należy:

- upewnić się czy poprawnie zaprogramowane są fusebity – **źródłem taktowania powinien być kwarc 12 MHz**, najlepiej skorzystać z dostarczonego projektu dla WinAVR i opcji *make all/program*,
- upewnić się, czy zdjęta została zworka zakładana do wgrywania firmware do programatora,
- w przypadku niektórych laptopowych portów USB konieczne może być dodanie dwóch diod Zenera 3V6 zgodnie z nowszą wersją schematu programatora.

4.2 Problemy z programowaniem układu docelowego.

W przypadku gdy wystąpi błąd programowania, poprzedzony odczytem przez avrdude zerowej sygnatury (*avrdude: Device signature = 0x000000*) winą może być niskie taktowanie lub duże opóźnienie startowe mikrokontrolera programowanego – należy założyć w programatorze **zworkę nr 3** aby zmniejszyć szybkość SPI przy programowaniu. Dotyczyć to może m.in. fabrycznie nowych mikrokontrolerów (dla Atmega8 ustawione fusebity SUT dają duże opóźnienie startowe a źródłem taktowania jest generator 1MHz).

Począwszy od wersji avrdude 5.8 i firmware 2009-02-28 częstotliwość taktowania SPI podczas programowa ograniczyć można także w sposób programowy wywołując avrdude z dodatkową

opcja `-B spi_delay`, gdzie `spi_delay` jest żądanym okresem dla SPI wyrażonym w mikrosekundach. USBasp ustawi częstotliwość możliwie bliską zadanej z zakresu od 500 Hz do 1.5 MHz. Ustalona przez program częstotliwość jest wyświetlana przy rozpoczynaniu programowania i powinna być dla prawidłowego działania co najmniej 4 razy mniejsza niż częstotliwość taktowania mikrokontrolera.

W przypadku zasilania układu docelowego z programatora mogą pojawić się problemy, jeżeli pobór prądu zestawu programator + układ programowany przekroczy 100 mA (takie zapotrzebowanie na prąd deklaruje programator podczas enumeracji USB). Wiele (zdecydowana większość?) kontrolek portów USB komputerów PC nie uwzględnia (lub ze względów oszczędnościowych pozbawiona jest odpowiednich elementów wykonawczych) limitu prądu z enumeracji - w takiej sytuacji bezpiecznie pobierać można z programatora prąd bliski 500 mA.

Mniej lub bardziej sporadyczne błędy przy programowaniu lub odczycie pamięci mogą oznaczać konieczność dodania diod Zenera 3V6 przy złączu USB zgodnie z nowszą wersją schematu. Na komputerach z którymi używałem USBasp (ASRock K7S8X - SIS 746FX, Dell D600 – i855PM, typowa konfiguracja Centrino 1) nie były one potrzebne. Jeżeli mają zostać zamontowane, warto zwrócić uwagę na moc strat – znaleźć można informacje, że diody o mocy strat 500 mW i więcej mogą mieć zbyt dużą pojemność pasożytniczą. Alternatywą jest użycie zamiast każdej z diod dwóch połączonych szeregowo czerwonych diod LED.

Źródło:

<http://www.avrfreaks.net/index.php?name=PNphpBB2&file=printview&t=62894&start=0>

<http://forums.obdev.at/viewtopic.php?f=8&t=2026>

Załączniki

USBasp_scheme.pdf – schemat elektryczny
USBasp_assembly.pdf – schemat montażowy
przykładowy plik makefile