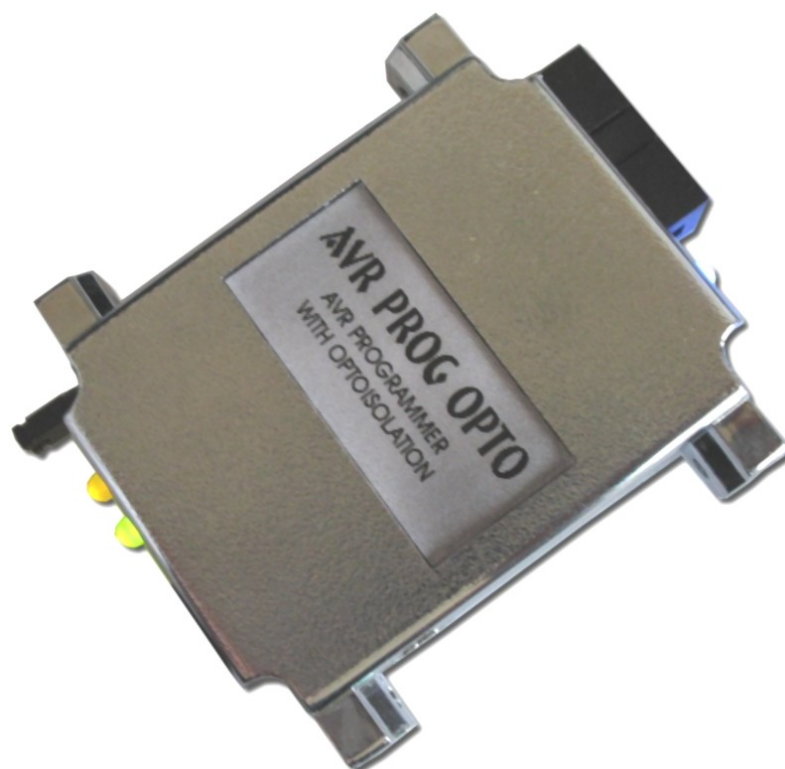


AVR PROG OPTO

Instrukcja obsługi
rev.1.0



Spis treści

1. Wprowadzenie.....	3
2. Opis.....	4
2.1 Diody LED.....	4
2.2 Zworki.....	5
2.3 Złącze ISP.....	6
3. Podłączenie programatora do układu	7
3.1 Pierwsze uruchomienie	7
3.2 Kolejność podłączania.....	9
3.3 Programowanie wielu mikrokontrolerów.....	9
3.4 Kolejność rozłączania.....	9
4. Lista obsługiwanych mikrokontrolerów.....	10
5. Instalacja sterowników.....	14
6. Konfiguracja USB Serial Port – zmiana numeru COM.....	19
7. Konfiguracja USB Serial Port – zmiana czasu opóźnienia.....	21
8. Zmiana firmware programatora.....	22
9. Konfiguracja oprogramowania.....	24
9.1 AVRISP (AVR Studio).....	24
9.2 AVR Prog (AVR Studio).....	26
9.3 AVRDUDE – AVRDUDE GUI.....	27
9.4 AVRDUDE – AVR8 Burn-O-Mat.....	28
9.5 BASCOM-AVR.....	29
9.6 AVR OSP II.....	33
9.7 CodeVision	35
10. Tryb interfejsu USB<->RS232 TTL.....	36
11. Kontakt - wsparcie techniczne	36

1. Wprowadzenie

AVR PROG OPTO jest programatorem ISP mikrokontrolerów z rodziny AVR firmy Atmel. Programator wyposażony jest w optoizolację. Odizolowane są wszystkie sygnały magistrali ISP. Optoizolacja zapewnia ochronę komputera przed przepięciami i zwarciami mogącymi pojawić się podczas pracy z programowanym mikrokontrolerem. Przypadkiem w którym urządzenie znajdzie szczególne zastosowanie będzie praca z układami wysokiego napięcia.

Programator znajdzie również zastosowanie podczas pracy z układami niskonapięciowymi, gdzie niedopuszczalne jest przedostanie się napięcia z USB komputera do "wrażliwego" układu. Urządzenie obsługuje protokoły STK500v2 oraz AVR910. Dzięki zaimplementowaniu obydwu protokołów programator obsługuje praktycznie wszystkie ośmiobitowe mikrokontrolery AVR. Komunikacja z komputerem odbywa się za pomocą złącza USB więc jest to idealne rozwiązanie dla nowych komputerów nie posiadających złącza RS232 czy LPT. Dzięki rozbudowanym sterownikom urządzenie bezproblemowo funkcjonuje w większości systemów operacyjnych (Windows XP, Windows VISTA, Windows 7, Windows 2000, Linux, itp.).

Obsługą USB w programatorze zajmuje się układ FT232RL firmy FTDI.

Rozdzielenie zadań (transmisja USB i obsługa SPI) na dwa osobne układy sprzyja wysokiej stabilności pracy programatora. Urządzenie dostarczone jest z trzema wersjami oprogramowania wewnętrznego które dzięki zastosowaniu bootloadera może być łatwo zmieniane przez użytkownika. W zależności od potrzeby można wgrać firmware obsługujący protokół STK500v2, AVR910 lub RS232 TTL (linia Rx i Tx). Do zmiany firmware nie jest potrzebny dodatkowy programator, wszystko odbywa się z poziomu komputera za pomocą dołączonych aplikacji.

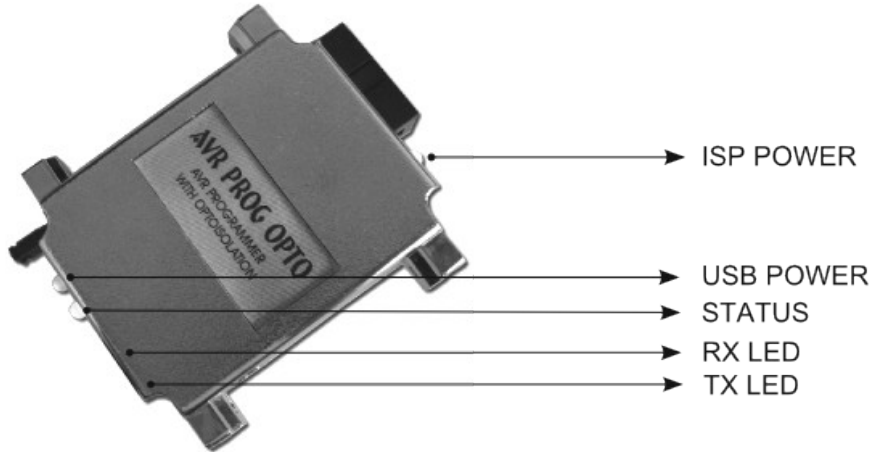
W urządzeniu zastosowano buforowanie sygnałów linii SPI co zwiększyło odporność układu na zakłócenia oraz rozszerzyło zakres tolerowanych napięć zasilania programowanego układu (2,1V – 6V).

Oprócz standardowego zabezpieczenia USB w postaci bezpiecznika plimerowego, w urządzeniu zastosowano szybki klucz MOSFET.

Programator posiada wyprowadzony dodatkowy autonomiczny sygnał zegarowy (GEN - pin nr 3 złącza ISP). Sygnał ten może być przydatny do taktowania programowanego mikrokontrolera w przypadku "zablokowania" niepoprawnie ustawionymi fusebitami (mikrokontroler ustawiony na taktowanie z zewnętrznego źródła).

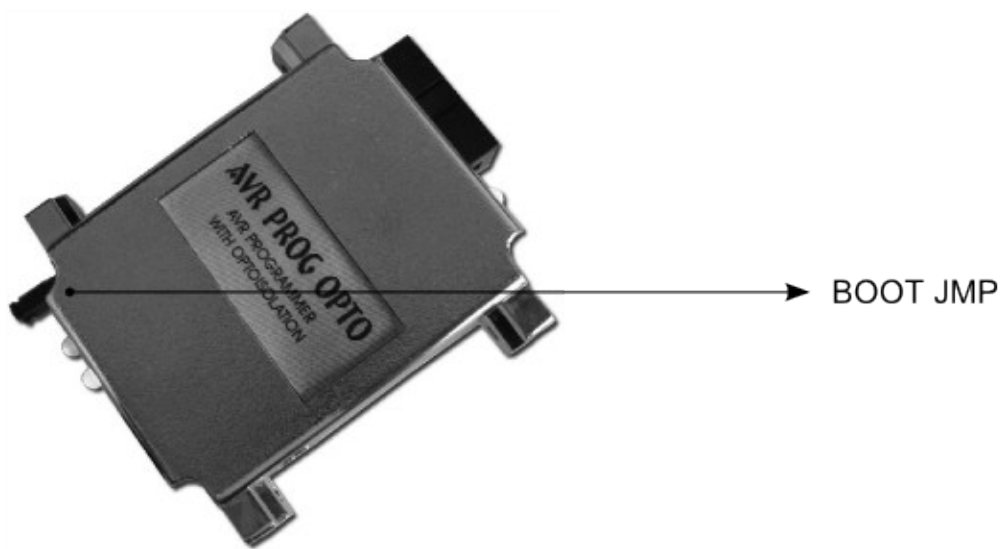
2. Opis

2.1 Diody LED



- ISP POWER** - dioda LED sygnalizująca zasilanie strony „wtórnej” programatora
- USB POWER** - świecenie diody sygnalizuje prawidłowe zainicjowanie połączenia USB w wyniku którego następuje zasilenie części „pierwotnej” programatora.
Do prawidłowego zestawienia połączenia USB niezbędna jest instalacja sterowników dołączonych na płycie CD.
- STATUS** - dwukolorowa dioda LED oznaczająca stan pracy urządzenia. W zależności od wgranego firmware stan pracy wyraża się następująco:
- Tryb STK500 – w tym trybie czerwony kolor oznacza stan spoczynkowy programatora. Dioda zmienia kolor na zielony podczas programowania.
 - Tryb AVR910 – zielony kolor oznacza stan spoczynkowy programatora. Podczas programowania dioda ISP LED zmienia kolor na czerwony.
 - Tryb RS232 – dioda wyłączona.
 - Tryb BOOTLOADER – niezależnie od wgranego firmware, przejście programatora do sekcji bootloadera sygnalizowane jest miganiem diody.
- RX LED** - sygnał odbieranych danych. Dioda sygnalizuje transmisję w kierunku komputer->programator.
- TX LED** - sygnał nadawania. Dioda sygnalizuje transmisję w kierunku programator->komputer.

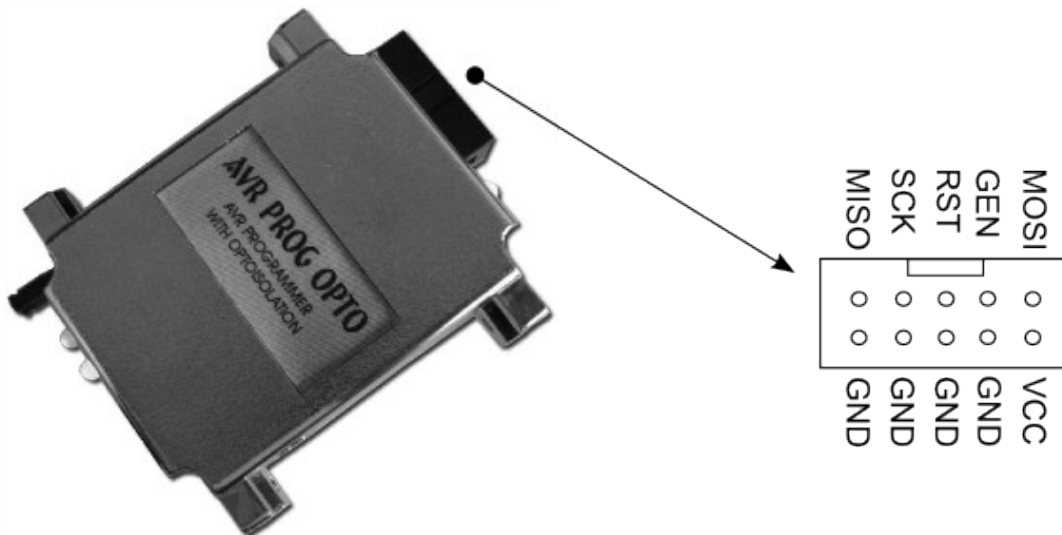
2.2 Zworki



BOOT JMP

- Założona zworka w chwili podłączenia programatora do komputera powoduje uaktywnienie bootloadera. Sygnalizowane jest to miganiem diody STATUS. Podczas normalnej pracy zworka powinna być zdjęta.

2.3 Złącze ISP



- MOSI (1)** - linia wyjściowa. Dane nadawane w kierunku programator → mikrokontroler.
- VCC (2)** - wejście napięcia zasilającego. Do pinu VCC niezbędne jest doprowadzenie napięcia zasilania z programowanego mikrokontrolera. Wartość napięcia powinna zawierać się w przedziale od 2,1V do 6V. Napięcie to służy do zasilenia strony izolowanej programatora – transoptorów i bufora wyjściowego.
- GEN (3)** - dodatkowy, niezależny sygnał zegarowy. Może być wykorzystywany podczas przeprogramowywania mikrokontrolerów z ustawionym fuse-bitem na taktowanie zewnętrzne (SUT_CKSEL → Ext. Clock)
- GND (4)** - masa. Sygnał masy występuje również na pinach 6,8 i 10.
- RST (5)** - reset
- SCK (7)** - sygnał zegara
- MISO (9)** - linia wejściowa. Dane odbierane przez programator

Wszystkie sygnały na złączu ISP są izolowane od pozostałej części programatora za pomocą transoptorów. Linie VCC i GND służą do zasilenia strony izolowanej programatora i powinny być połączone z VCC i GND programowanego mikrokontrolera. Wartość napięcia doprowadzonego do VCC powinna zawierać się w przedziale od 2,1V do 6V.

3. Podłączenie programatora do układu

3.1 Pierwsze uruchomienie

W pierwszej kolejności należy zainstalować sterowniki do programatora dołączone na płycie CD. Proces instalacji sterowników opisany jest w punkcie „Instalacja sterowników” tej instrukcji.

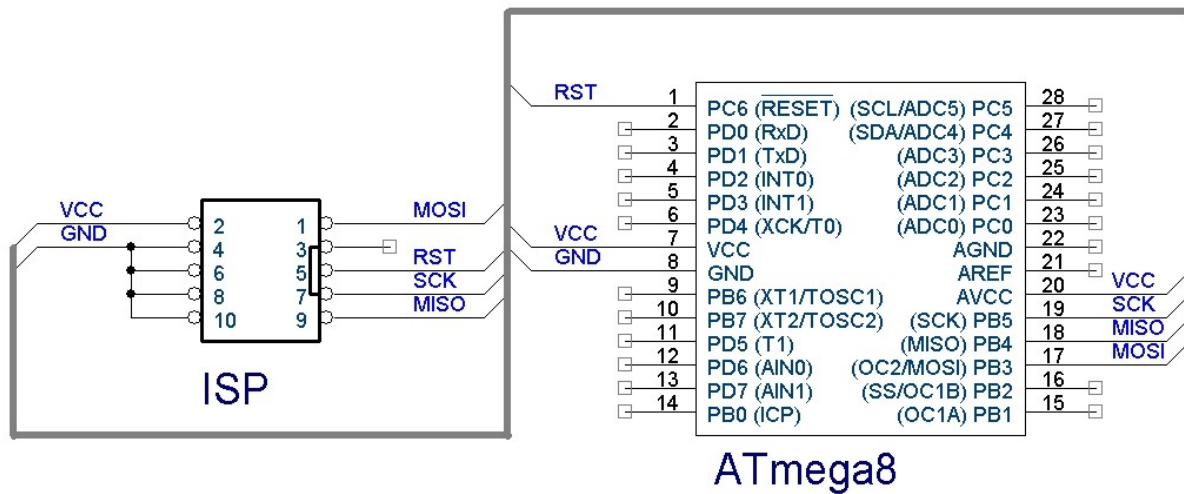
W następnej kolejności należy zestawić połączenie między programatorem a programowanym układem. Na tym etapie programowany mikrokontroler nie powinien być zasilony.

Sygnaly z programatora należy połączyć z odpowiednimi liniami interfejsu ISP programowanego układu. Rozkład portów odpowiedzialnych za programowanie w systemie (Serial Programming Pin Mapping) należy wcześniej sprawdzić w dokumentacji do posiadanego mikrokontrolera. W niektórych mikroprocesorach magistrała ISP pokrywa się ze sprzętową magistralą SPI (np. Atmega8), w innych magistrała do programowania ulokowana jest pod „regularnymi” portami (np. Atmega 128). Podłączając mikrokontroler do programatora należy więc zwrócić uwagę że nie zawsze ISP jest tożsame z SPI.

Łączymy odpowiednio:

Złącze ISP programatora	Magistrała ISP mikrokontrolera
MOSI	MOSI
MISO	MISO
SCK	SCK
RST	RESET
VCC	VCC
GND	GND

Linie VCC i GND powinny być połączone z VCC i GND programowanego mikrokontrolera. Wartość napięcia doprowadzonego do VCC powinna zawierać się w przedziale od 2,1V do 6V.



Połączenie programatora z mikrokontrolerem na przykładzie Atmega 8

Po zestawieniu połączenia między programatorem a mikrokontrolerem należy podłączyć zasilanie programowanego układu. W programatorze powinna zaświecić się niebieska dioda ISP POWER sygnalizująca obecność zasilania na złączu ISP a tym samym zasilanie części „wtórnej” programatora.

Następnym etapem jest podłączenie programatora do komputera.

W programatorze powinna zaświecić się dioda USB POWER świadcząca o prawidłowym zainicjowaniu połączenia USB oraz zasilaniu części „pierwotnej” programatora. Dioda ta nie będzie się świecić jeśli nie zostały zainstalowane sterowniki lub proces instalacji sterowników nie został ukończony.

W zależności od wersji firmware* powinna zaświecić się również dioda STATUS.

*Programator dostarczony jest z wgranym firmware STK500v2 więc dioda STATUS w fazie spoczynkowej powinna świecić się na czerwono.

3.2 Kolejność podłączania.

1. AVR PROG OPTO i programowany mikrokontroler odłączone od zasilania.
2. Połączenie AVR PROG OPTO z docelowym mikrokontrolerem.
3. Zasilenie programowanego układu.
4. Podłączenie AVR PROG OPTO do komputera.

3.3 Programowanie wielu mikrokontrolerów

1. AVR PROG OPTO i programowany mikrokontroler odłączone od zasilania.
2. Połączenie AVR PROG OPTO z mikrokontrolerem.
3. Zasilenie programowanego układu.
4. Podłączenie AVR PROG OPTO do komputera.
5. Programowanie mikrokontrolera.
6. Odłączenie tasiemki łączącej programator i mikrokontroler.
7. Odłączenie zasilania od programowanego mikrokontrolera.
8. Podłączenie zasilania do kolejnego mikrokontrolera.
9. Połączenie programatora z tym mikrokontrolerem.
10. Powtórzenie czynności od punktu 5.

3.4 Kolejność rozłączania

1. Odłączyć programator od programowanego układu
2. Odłączyć programator od komputera.
3. Wyłączyć zasilenie programowanego układu.

4. Lista obsługiwanych mikrokontrolerów

Protokół STK500v2 jest niezależny od sprzętu. Algorytmy obsługi poszczególnych mikrokontrolerów zaszyte są w oprogramowaniu sterującym na komputerze. Co za tym idzie, lista obsługiwanych przez programator układów jest tożsama z „pulą” mikrokontrolerów obsługiwanych przez daną aplikację.

Podstawowa lista wygląda następująco:

MIKROKONTROLER	TRYB PRACY (FIRMWARE)	
	STK500V2	AVR910
AT90CAN128	X	X
AT90CAN32	X	
AT90CAN64	X	
AT90PWM2	X	X
AT90PWM216	X	
AT90PWM2B	X	
AT90PWM3	X	X
AT90PWM316	X	
AT90PWM3B	X	
AT90S1200	X	X
AT90S2313	X	X
AT90S2323	X	X
AR90S2333		X
AT90S2343	X	X
AT90S4414	X	X
AT90S4433	X	X
AT90S4434	X	X
AT90S8515	X	
AT90S8515A		X
AT90S8534		X
AT90S8535	X	X
AT90S8544		X
AT90USB1286	X	

AT90USB1287	X	
AT90USB162	X	
AT90USB646	X	
AT90USB647	X	
AT90USB82	X	
ATmega103	X	X
ATmega128	X	X
ATmega1280	X	X
ATmega1281	X	X
ATmega16	X	X
ATmega161	X	X
ATmega162	X	X
ATmega163	X	X
ATmega164		X
ATmega164P	X	
ATmega165	X	
ATmega165P	X	
ATmega168	X	X
ATmega168P	X	
ATmega169	X	X
ATmega169P	X	
ATmega16HVA	X	
ATmega2560	X	
ATmega2561	X	
ATmega32	X	X
ATmega323	X	
ATmega324		X
ATmega324P	X	
ATmega325	X	
ATmega3250	X	
ATmega3250P	X	
ATmega325P	X	
ATmega328P	X	

ATmega329	X	X
ATmega3290	X	X
ATmega3290P	X	
ATmega329P	X	
ATmega406	X	
ATmega48	X	X
ATmega48P	X	
ATmega64	X	X
ATmega640	X	X
ATmega644	X	
ATmega644P	X	
ATmega645	X	
ATmega6450	X	
ATmega649	X	X
ATmega6490	X	X
ATmega8	X	X
ATmega8515	X	X
ATmega8535	X	X
ATmega88	X	X
ATmega88P	X	
ATtiny10		X
ATtiny11	X	
ATtiny12	X	X
ATtiny13	X	X
ATtiny15	X	X
ATtiny22	X	
ATtiny2313	X	X
ATtiny24	X	X
ATtiny25	X	X
ATtiny26	X	X
ATtiny261	X	
ATtiny28	X	
ATtiny43U	X	

ATtiny44	X	X
ATtiny45	X	X
ATtiny461	X	
ATtiny48	X	
ATtiny84	X	X
ATtiny85	X	X
ATtiny861	X	

5. Instalacja sterowników

Dokument ten zawiera opis procesu instalacji sterowników w systemie Windows XP.

Instrukcje instalacji sterowników FTDI dla innych systemów (Windows Vista, MAC, Windows 2000, Windows 98, Linux) znajdują się na dołączonym CD (CD:\sterowniki\installation guide).

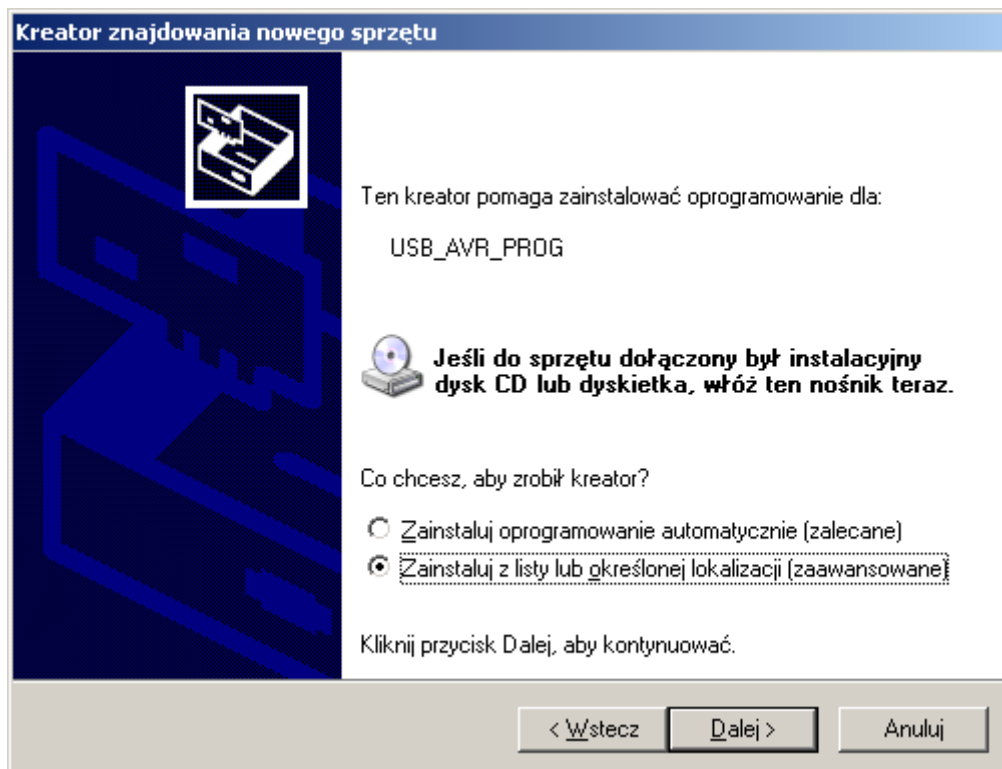
Zawsze aktualne wersje instrukcji instalacji dostępne są również na stronie producenta układu FT232R pod adresem:

<http://www.ftdichip.com/Documents/InstallGuides.htm>

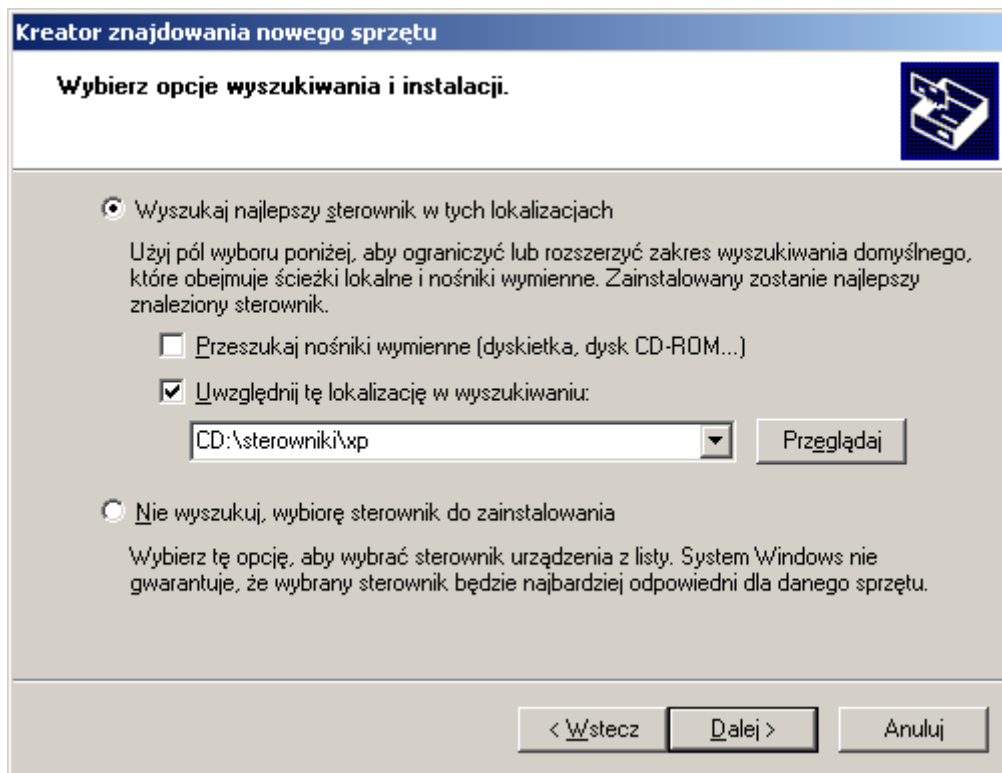
Programator podłączamy do komputera. Po wykryciu urządzenia zostanie uruchomiony kreator znajdowania nowego sprzętu. W kreatorze wybieramy opcję *'Nie, nie tym razem'* i klikamy *'Dalej'*.



W następnej planszy wybieramy opcję 'Zainstaluj z listy lub określonej lokalizacji (zaawansowane)' i przechodzimy dalej.

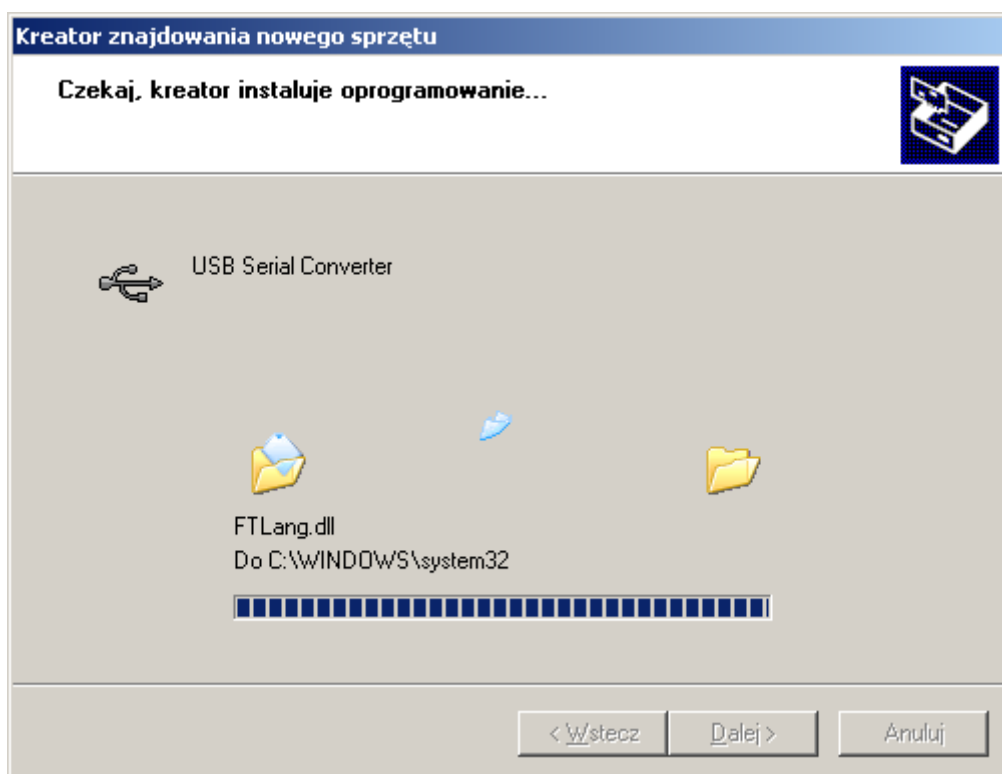


Następnie należy wskazać lokalizację sterowników

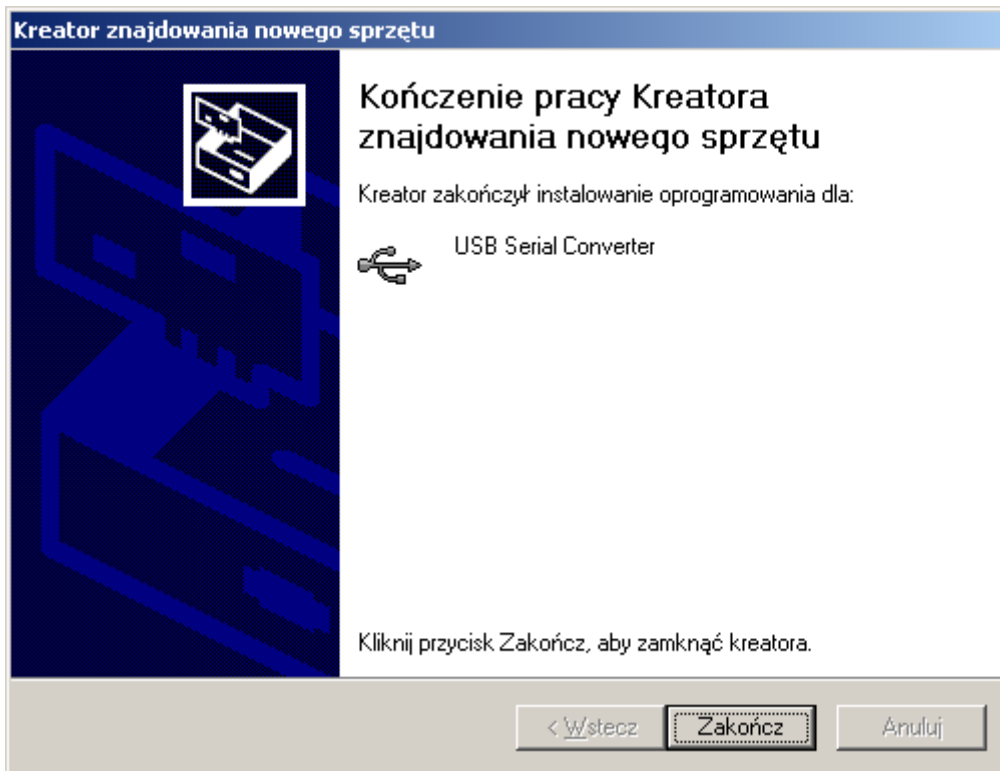


W planszy która się pojawi wybieramy opcję 'Wyszukaj najlepszy sterownik w tych lokalizacjach' oraz zaznaczamy 'Uwzględnij tę lokalizację w wyszukiwaniu'. Wciskamy 'Przełóżaj' i wskazujemy lokalizację z płyty – \sterowniki\xp

System powinien znaleźć i zainstalować sterowniki z podanej lokalizacji.



Po wyświetleniu okna końca pracy kreatora należy wcisnąć przycisk 'Zakończ'.



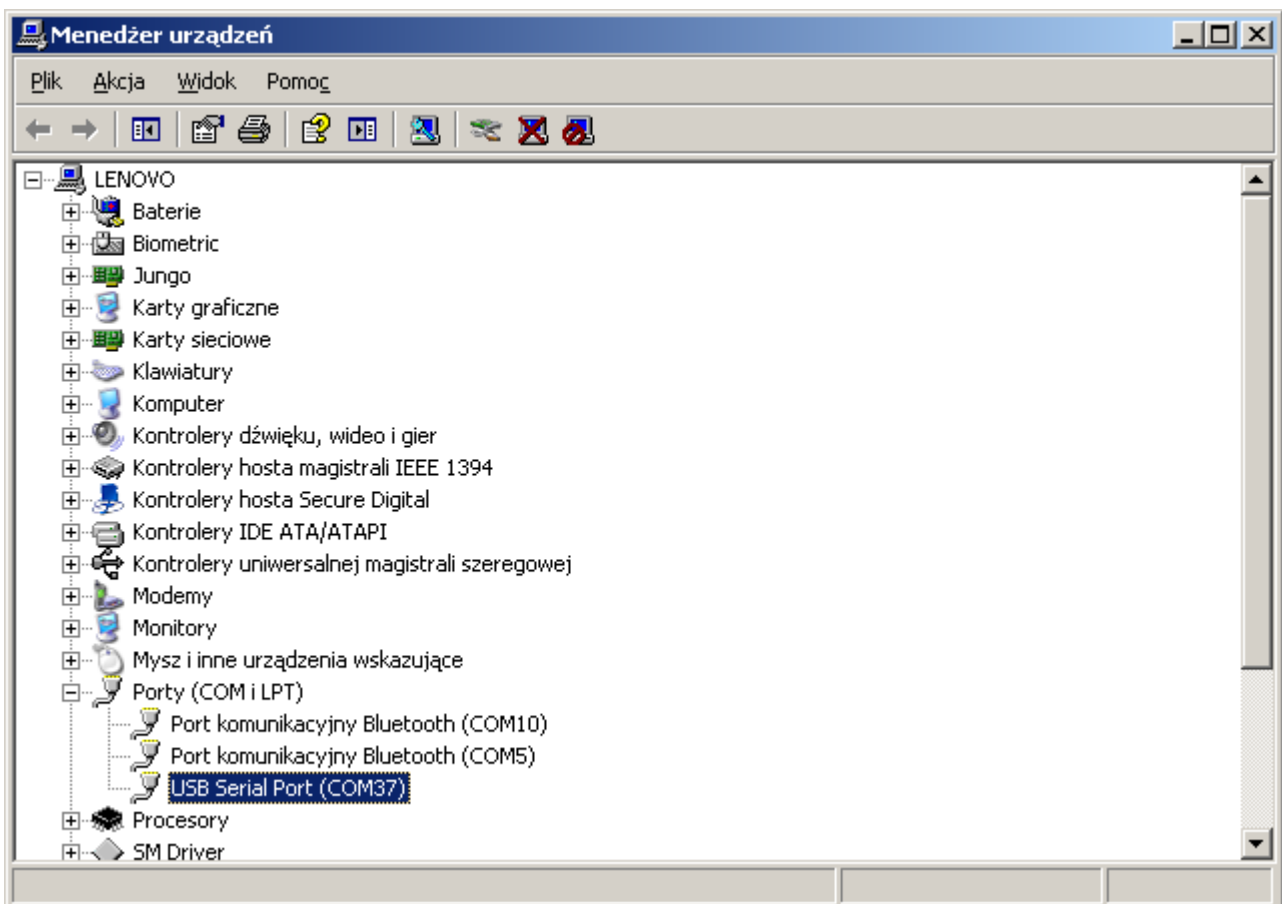
Chwilę potem system wykryje kolejne urządzenie – '*USB Serial Port*'. Proces instalacji tego urządzenia przebiega identycznie jak poprzednio. W kolejnych etapach pracy kreatora zaznaczamy te same opcje co poprzednio, a do sterowników podajemy tą samą lokalizację z płyty.

O prawidłowym zainstalowaniu sterowników system poinformuje nas komunikatem '*Twój nowy sprzęt jest zainstalowany i gotowy do użycia*'.

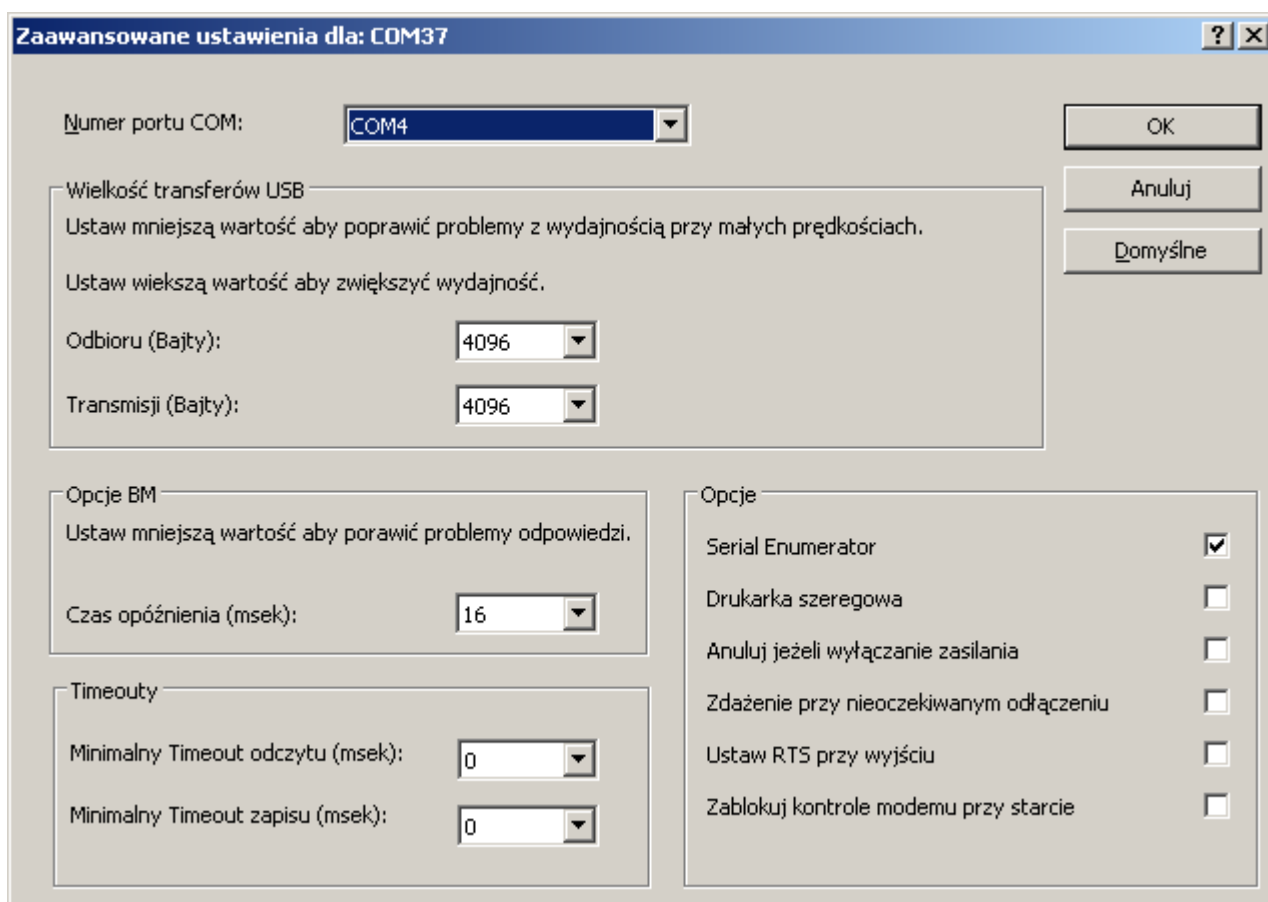
6. Konfiguracja USB Serial Port – zmiana numeru COM

Przy współpracy z niektórymi programami może zaistnieć potrzeba zmiany numeru portu COM który zajmuje programator. Np AVR Studio odpytuje tylko początkowe porty komputera COM1 – COM4. Chcąc zarejestrować programator na którymś z początkowych portów COM należy wykonać następujące czynności.

W pierwszej kolejności należy otworzyć menadżer urządzeń i rozwinąć gałąź 'Porty (COM i LPT)'. Interesuje nas pozycja 'USB Serial Port' z numerem COM-a w nawiasie. Jeśli występujący tu numer jest dosyć wysoki, korzystnie jest zmienić na jeden z pierwszych tj. COM1 – COM4.



Chcąc zmienić numer COM-a otwieramy prawym przyciskiem myszy właściwości USB Serial Port-u. Przechodzimy do zakładki 'Ustawienia Portu' i naciskamy 'Zawansowane'. W polu 'Numer portu COM' wybieramy któryś z początkowych portów, najlepiej wolny. Jeśli obok interesującego nas numeru widnieje napis '(w użyciu)' możemy to zignorować pod warunkiem że fizycznie na tym COM-ie nie znajduje się żadne urządzenie.



Klikamy 'OK' i po kilku sekundach odłączamy interfejs od komputera. Po ponownym podłączeniu konwerter powinien ulokować się na wskazanym numerze portu COM.

7. Konfiguracja USB Serial Port – zmiana czasu opóźnienia

Znaczną poprawę wydajności programatora można uzyskać zmniejszając w konfiguracji portu COM opóźnienia. W tym celu należy otworzyć menadżer urządzeń i rozwinąć gałąź 'Porty(COM i LPT)'. Interesuje nas pozycja 'USB Serial Port'. Otwieramy (prawy przyciskiem myszy) właściwości USB Serial Port-u. Przechodzimy do zakładki 'Ustawienia Portu' i naciskamy 'Zawansowane'.

W sekcji 'Opcje BM' zmniejszamy czas opóźnienia z wartości domyślnej (16ms) na niższą.

Zaawansowane ustawienia dla: COM3

Numer portu COM: COM3

Wielkość transferów USB
Ustaw mniejszą wartość aby poprawić problemy z wydajnością przy małych prędkościach.
Ustaw większą wartość aby zwiększyć wydajność.

Odbioru (Bajty): 4096
Transmisji (Bajty): 4096

Opcje BM
Ustaw mniejszą wartość aby porawić problemy odpowiedzi.

Czas opóźnienia (msek): 1

Timeouty
Minimalny Timeout odczytu (msek): 0
Minimalny Timeout zapisu (msek): 0

Opcje
Serial Enumerator
Drukarka szeregową
Anuluj jeżeli wyłączenie zasilania
Zdążenie przy nieoczekiwanym odłączeniu
Ustaw RTS przy wyjściu
Zablokuj kontrole modemu przy starcie

OK
Anuluj
Domyślne

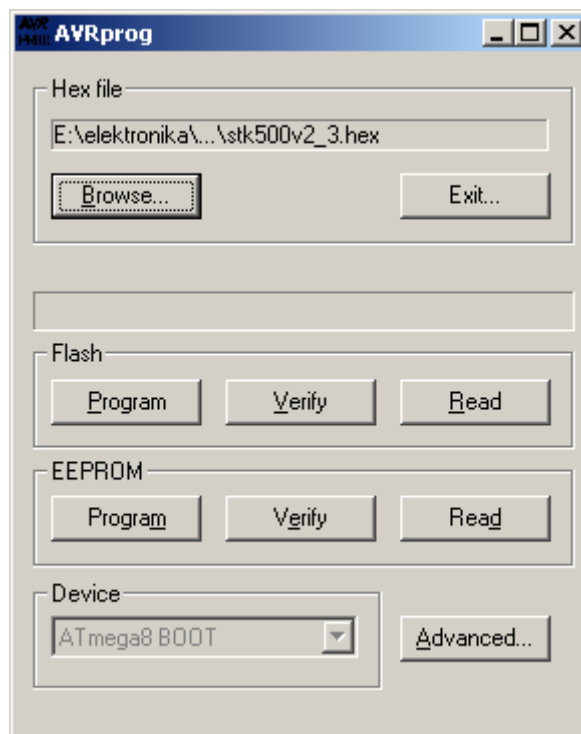
Wartość tą należy dobrać doświadczalnie – testując po każdej zmianie komunikację z programatorem. Teoretycznie, minimalna wartość przy której uda się uzyskać poprawną transmisję może zależeć od konkretnego programu, w praktyce jednak na nowszych komputerach udaje się zejść z opóźnieniem do 1ms na stałe. W przypadku pojawienia się problemów z komunikacją należy wrócić do ustawień domyślnych.

8. Zmiana firmware programatora

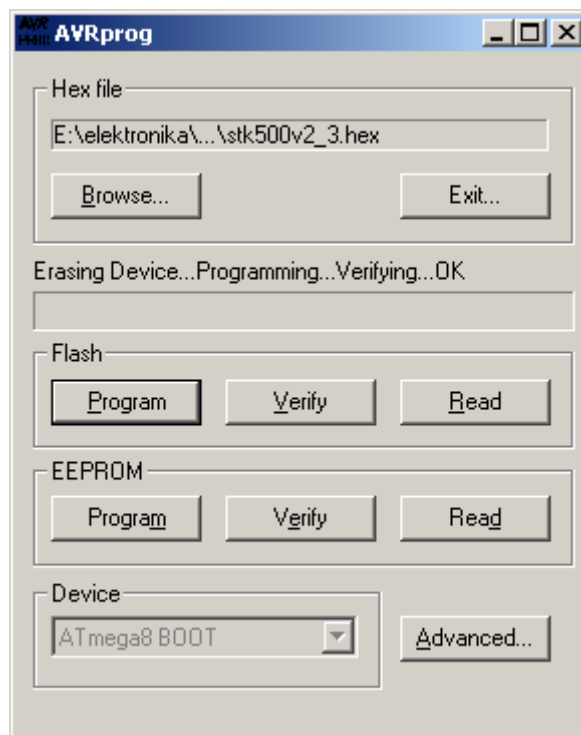
Firmware programatora może być zmieniany programem AVR Prog wchodzącym w skład pakietu AVR Studio.

W celu wymiany wsadu należy wykonać następujące czynności:

- odłączyć (jeśli był podłączony) programator od komputera
- założyć zworkę BOOT JMP.
- podłączyć programator do komputera
- programator zostanie uruchomiony w trybie bootloadera – dioda STATUS powinna migać na zielono
- uruchomić program AVR Prog ('CD:\narzędzia\AvrProg.exe'). Po uruchomieniu programu powinna przestać migać dioda ISP LED. Jeśli zamiast programu pojawia się okno z informacją *'No supported board found !'* należy sprawdzić na którym porcie COM zainstalowany jest programator. Aplikacja AVR Prog wymaga umiejscowienia programatora na COM1, COM2 lub COM3 (zmiana konfiguracji opisana jest w punkcie *Konfiguracja USB Serial Port*). Po uruchomieniu programu powinno pojawić się okno:



- klikając przycisk 'Browse...' należy wskazać odpowiedni wsad czyli np. plik stk500v2_1.hex (CD:\firmware\stk500v2_1.hex) lub avr910_1.hex (CD:\firmware\avr910_1.hex) lub rs232_1.hex (CD:\firmware/rs232_1.hex)
- w sekcji 'Flash' programu nacisnąć przycisk 'Program'
- o pomyślnym przebiegu operacji flashownia poinformuje nas komunikat 'Erasing Device...Programming...Verifying..OK' nad wskaźnikiem postępu.



- po skończonej operacji upgrade'u zamykamy program (krzyżyk w górnym prawym rogu), zdejmujemy zworkę BOOT JMP oraz odłączamy programator od USB komputera.
- po podłączeniu do USB, programator jest gotowy do pracy.

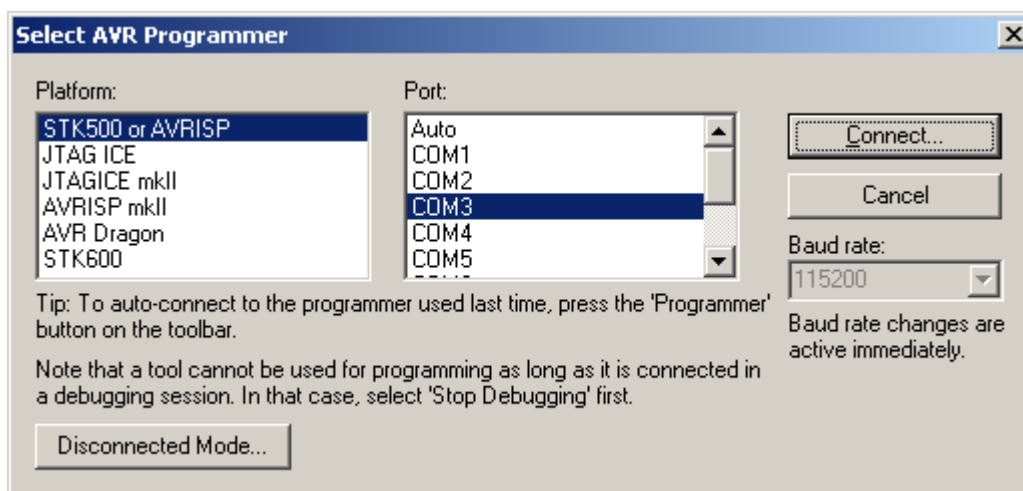
9. Konfiguracja oprogramowania

9.1 AVRISP (AVR Studio)

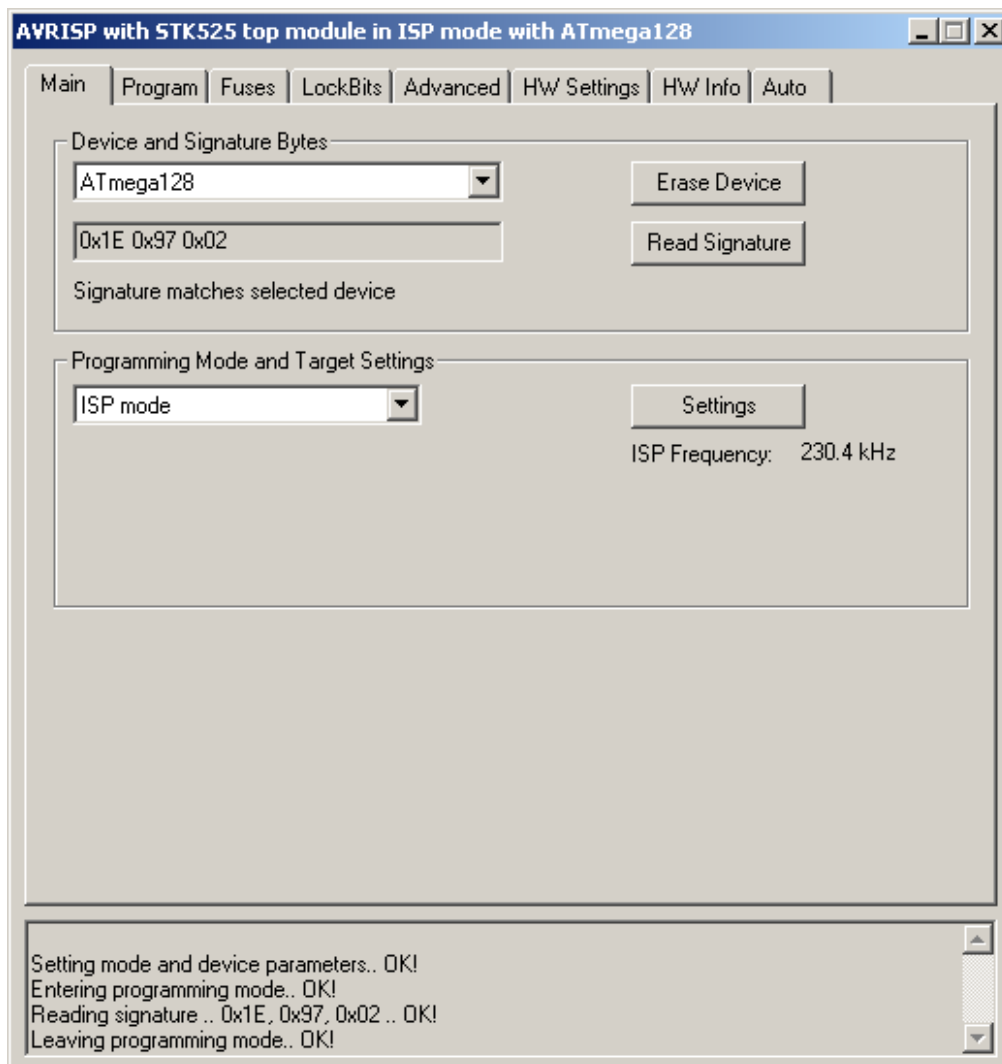
Do programatora należy wgrać firmware STK500v2.

W celu połączenia z programatorem wybieramy AVR Studio -> Tools -> Program AVR -> Connect.

W oknie *Select AVR Programmer* jako *Platform* wybieramy pierwszą pozycję czyli *STK500 or AVRISP* oraz numer portu na którym zainstalowany jest programator.



Po kliknięciu *Connect* powinno pojawić się główne okno programu.



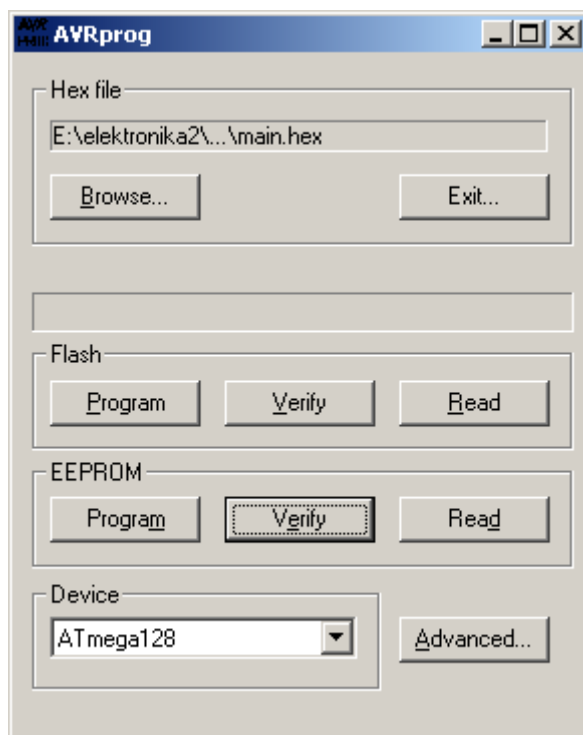
Uwaga ! Na chwilę obecną AVR Studio nie posiada autodetekcji typu podłączonego do programatora mikrokontrolera, dlatego bardzo ważne jest, żeby przed przystąpieniem do jakichkolwiek operacji związanych z programowaniem, w polu *Device and Signature Bytes* zakładki *Main* ręcznie wybrać z rozwijanej listy nazwę docelowego mikroprocesora.

Wciskając przycisk *Read Signature* należy sprawdzić czy prawidłowo jest rozpoznawana sygnatura – *Signature matches selected device*. Jeśli odczytana sygnatura nie pasuje do typu wybranego mikrokontrolera – *WARNING:Signature does not match selected device!*, należy upewnić się czy prawidłowo został wybrany typ mikrokontrolera. W następnej kolejności należy zwrócić uwagę na właściwe zasilanie układu oraz zweryfikować poprawność połączeń między programatorem a mikrokontrolerem.

9.2 AVR Prog (AVR Studio)

Do programatora należy wgrać firmware AVR910.

Program AVR Prog może być wywoływany z poziomu środowiska lub jako samodzielna aplikacja. Z poziomu AVR Studio, AVRprog uruchamiamy wybierając pierwszą od góry pozycję w zakładce *Tools* (AVR Studio -> Tools -> AVR Prog...). Aplikacja przed uruchomieniem odpytuje porty COM1 do COM4 komputera w poszukiwaniu programatora. Programator musi być zatem podłączony i prawidłowo skonfigurowany czyli zainstalowany na którymś z początkowych portów COM. Zmiana numeru COM opisana jest w rozdziale *Konfiguracja USB Serial Port – zmiana numeru COM*.



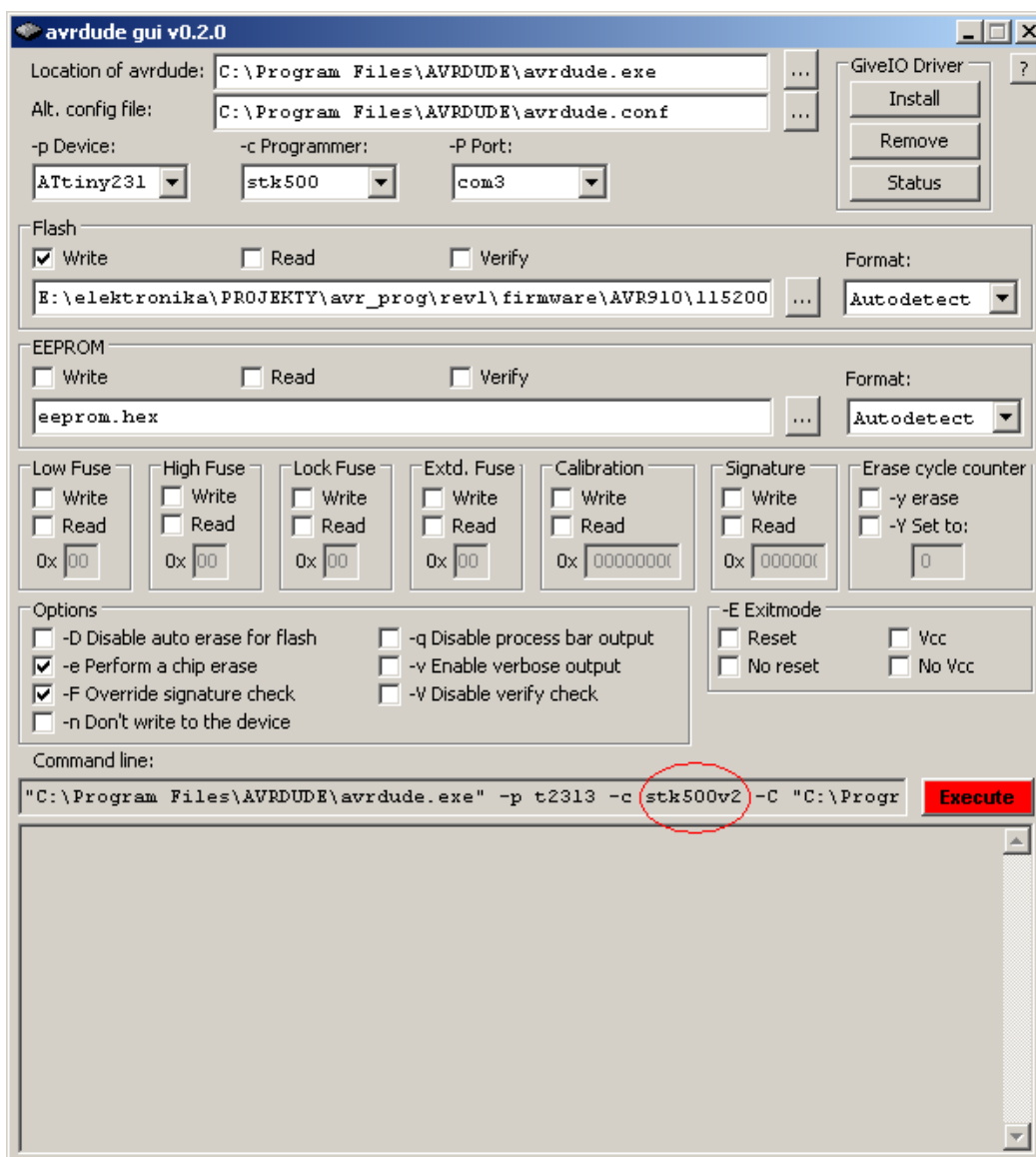
Bardzo ważne jest, żeby przed przystąpieniem do jakichkolwiek operacji związanych z programowaniem, w polu *Device* wybrać z rozwijanej listy nazwę podłączonego mikrokontrolera.

9.3 AVRDUDE – AVRDUDE GUI

Program AVRDUDE może współpracować zarówno z firmware STK500 jak i AVR910. Programowanie w trybie STK500 przebiega jednak szybciej więc w instrukcji skupię się na tym wariantcie – programator z firmware STK500.

Na początek należy skopiować katalogi AVRDUDE oraz AVRDUDE_GUI na dysk twardy komputera. Następnie uruchamiamy program *avrdude-gui.exe*. W linii *Location of avrdude* wskazujemy lokalizację pliku „avrdude.exe”, natomiast niżej, w linii *Alt. Config file* podajemy ścieżkę do pliku konfiguracyjnego „avrdude.conf”.

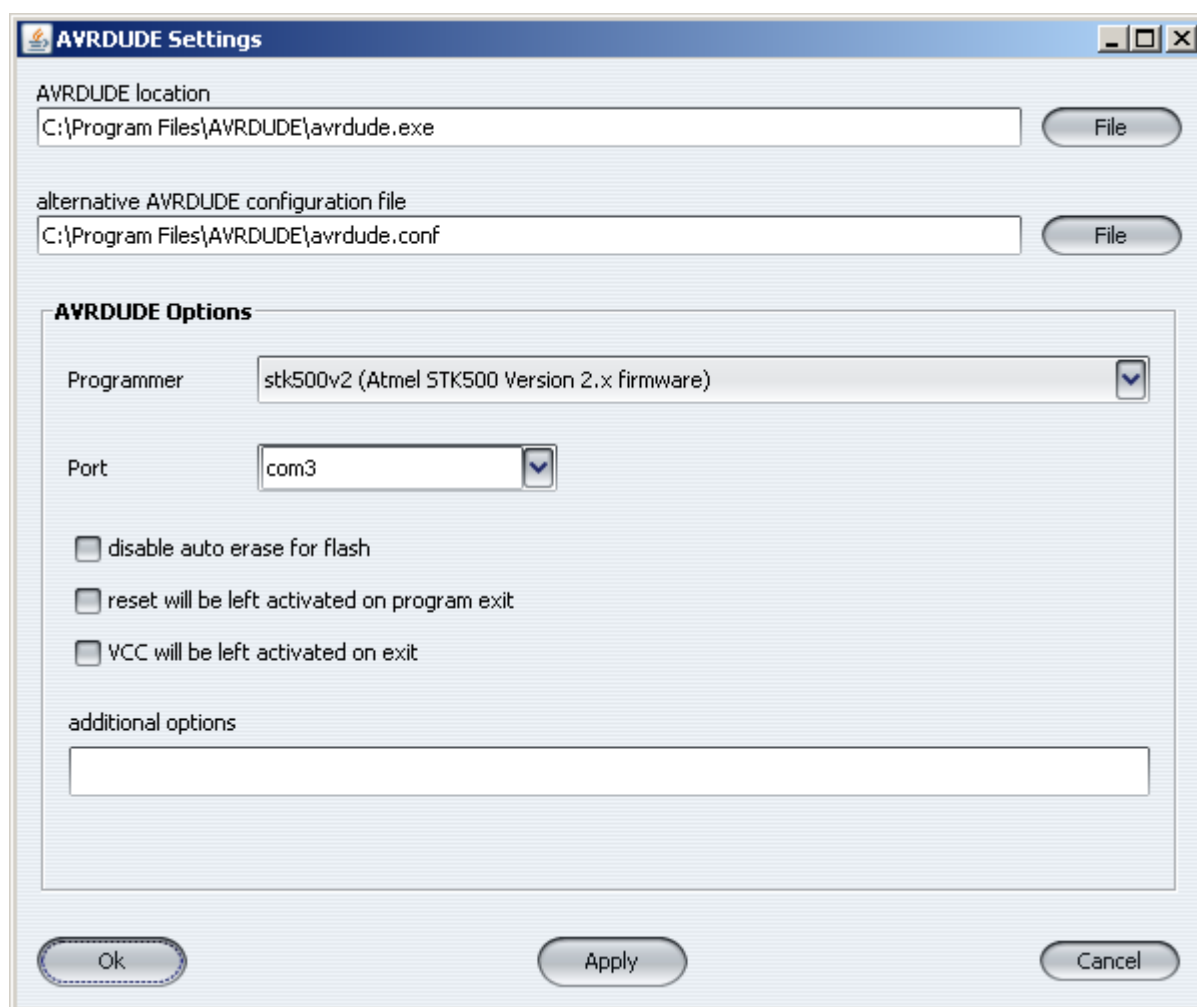
Wybieramy rodzaj programowanego mikrokontrolera (*-p Device*), numer portu COM (*-P Port*) na którym jest programator a jako „*-c Programmer*” wybieramy „stk500”. Uwaga ! Przed finalnym naciśnięciem czerwonego przycisku „Execute” w „*Command line*”, do „stk500” dostawiamy „v2”- tak jak poniżej.



9.4 AVRDUDE – AVR8 Burn-O-Mat

Programator z wsadem STK500.

Aby uruchomić program AVRDUDE z GUI AVR8 Burn-O-Mat należy w pierwszej kolejności zainstalować środowisko JAVA (CD\narzędzia\jre-6u4-windows-i586-p). Następnie należy przekopiować na dysk twardy katalog AVRDUDE (CD\narzędzia\AVRDUDE) oraz katalog AVR8_Burn-O-Mat (CD\narzędzia\AVRDUDE_GUI\AVR8_Burn-O-Mat). Z katalogu AVR8_Burn-O-Mat uruchomić plik wsadowy *start.bat*. Z menu programu wybrać Settings->AVRDUDE



W '*AVRDUDE location*' podać lokalizację pliku *avrdude.exe*

W '*alternative AVRDUDE configuration file*' należy podać lokalizację pliku *avrdude.conf*.

Następnie zamknąć program. Po ponownym uruchomieniu gui jeszcze raz przejść do AVRDUDE Settings i w „Programmer” wybrać '*stk500v2...*' oraz numer portu *com* zajmowanego przez programator. Zaakceptować ustawienia i nacisnąć „Ok”.

W głównym oknie programu należy wybrać rodzaj programowanego mikrokontrolera.

9.5 BASCOM-AVR

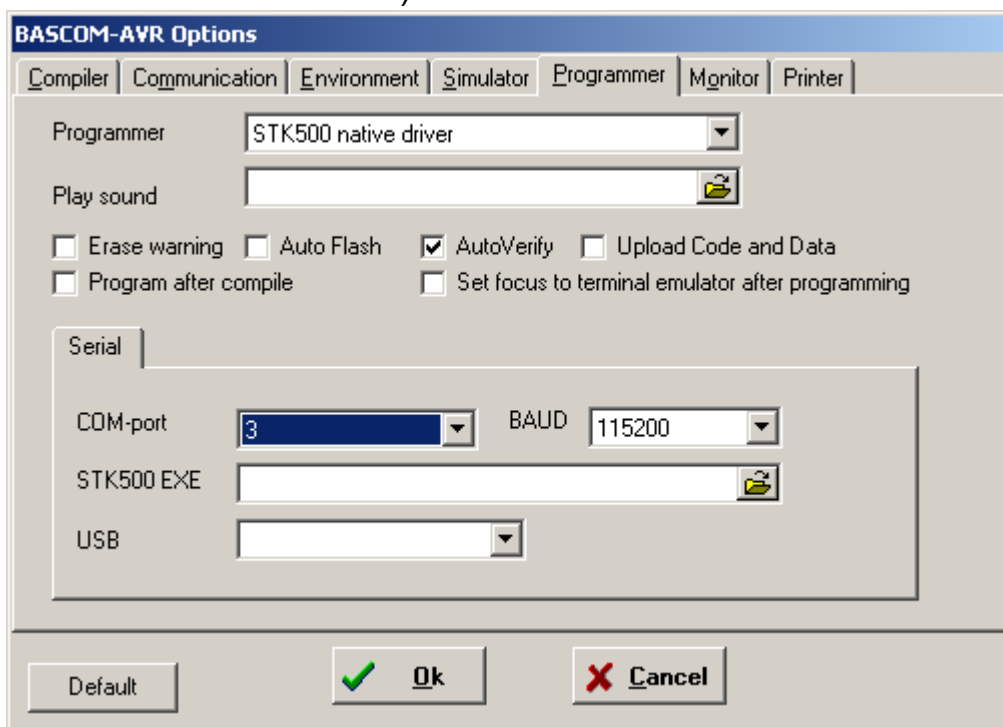
Jest kilka możliwości współpracy AVR PROG USB ze środowiskiem Bascom, poniżej kilka z nich. We wszystkich opisanych tu wariantach do programatora powinien być załadowany firmware STK500.

STK500 native driver

Począwszy od wersji 1.11.9.3 w środowisku Bascom zaimplementowana została bezpośrednia obsługa programatorów zgodnych z stk500.

Jest to ukłon w stronę użytkowników przyzwyczajonych do korzystania z programatora STK200 - obsługa programowanego mikrokontrolera w tym trybie wygląda wówczas bardzo podobnie jak w STK200.

W oknie konfiguracji programatora (*BASCOM-AVR->Options->Programmer*) jako *Programmer* należy wybrać *STK500 native driver*. Następnie należy wprowadzić numer portu COM na którym zainstalowany jest programator (pole *COM-port*) oraz ustawić prędkość *BAUD* na 115200 (Sprawdzenie oraz zmiana numeru portu COM na którym znajduje się programator opisane jest w rozdziale *Konfiguracja USB Serial Port – zmiana numeru COM*)



Jedynym ale bardzo poważnym mankamentem tego trybu pracy jest mała prędkość programowania. Być może w przyszłych wersjach BASCOM-AVR zostanie to poprawione.

Wydajność programatora w tym trybie można nieznacznie poprawić zmniejszając czas opóźnienia portu COM. Zmiana ta opisana jest w rozdziale *Konfiguracja USB Serial Port – zmiana czasu opóźnienia*.

Poniżej znajdują się przykładowe zrzuty ekranu z pracy w tym trybie.

Atmel STK500 protocol compatible Programmer

File Buffer Chip

Chip: **ATmega128** Flash size: **128 KB** EEPROM size: **4 KB**

Flash	EEPROM	Lock and Fuse bits
	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	
00000	65 57 20 64 75 4A 20 6C 39 32 31 20 3A 38 37 32	eW dW I921 :872
00010	35 3A 20 34 6F 4C 64 61 64 65 66 20 6C 69 20 65	5: 4oLdedef li e
00020	3A 45 45 5C 45 4C 54 4B 4F 52 49 4E 41 4B 5C 32	:EE\ELTKORINAK\2
00030	52 50 4A 4F 4B 45 59 54 55 5C 42 53 4F 54 53 52	RPJQKEYTU\BSOTSR
00040	33 32 5C 32 43 50 5C 42 5C 32 56 41 5F 52 4F 42	32\2CP\B\2VA_ROB
00050	2E 58 43 54 0D 48 57 0A 64 65 4A 20 6C 75 32 20	.XCT.HW.deJ lu2
00060	20 39 38 31 32 3A 3A 38 30 30 53 20 76 61 64 65	9812::800S vade
00070	61 20 20 73 69 66 65 6C 45 20 5C 3A 4C 45 4B 45	a sifeI E \:LEKE
00080	52 54 4E 4F 4B 49 32 41 50 5C 4F 52 45 4A 54 4B	RTNOKI2AP\OREJTK
00090	5C 59 53 55 54 42 52 4F 32 53 32 33 50 5C 42 43	\YSUTBRD2S23P\BC
000A0	32 5C 55 5C 42 53 4F 54 53 52 33 32 2E 32 41 4D	2\U\BSOTSR32.2AM
000B0	0D 58 57 0A 64 65 4A 20 6C 75 32 20 20 39 38 31	.XW.deJ lu2 981
000C0	32 3A 3A 38 30 30 53 20 76 61 64 65 61 20 20 73	2::800S vadea s
000D0	69 66 65 6C 42 20 43 41 55 4B 31 50 4D 2E 58 41	ifeI B CAUK1PM.XA
000E0	0A 0D 65 57 20 64 75 4A 20 6C 39 32 31 20 3A 38	..eW dW I921 :8
000F0	38 32 30 3A 20 30 61 53 65 76 20 64 73 61 66 20	820: 0aSev dsaf

layout.bin FLASH : 42 KB EEPROM : 4 KB

Atmel STK500 protocol compatible Programmer

File Buffer Chip

Chip: **ATmega128** Flash size: **128 KB** EEPROM size: **4 KB**

Flash EEPROM Lock and Fuse bits

Chip	
Name	MEGA128
Calibration 0	AC
Lockbits	FF
Lockbit 54	11:No restrictions for SPM or LPM accessing the boot loader section
Lockbit 32	11:No restrictions for SPM or LPM accessing the application section
Lockbit 10	11:No memory lock features enabled
Fusebits	FD
Fusebit 7	1:Brown-out detection level at VCC=2.7 V
Fusebit 6	1:Brown-out detection disabled
Fusebit 98DCBA	111101:Ext. Crystal/Resonator Medium Freq.
Fusebits High	89
Fusebit High E	1:Disable OCD
Fusebit High F	0:Enable JTAG
Fusebit High G	0:Enable serial downloading
Fusebit High H	0:osc 0
Fusebit High I	1:EEPROM memory is erased when erasing chip

Refresh

Write LB

Write FS

Write F6H

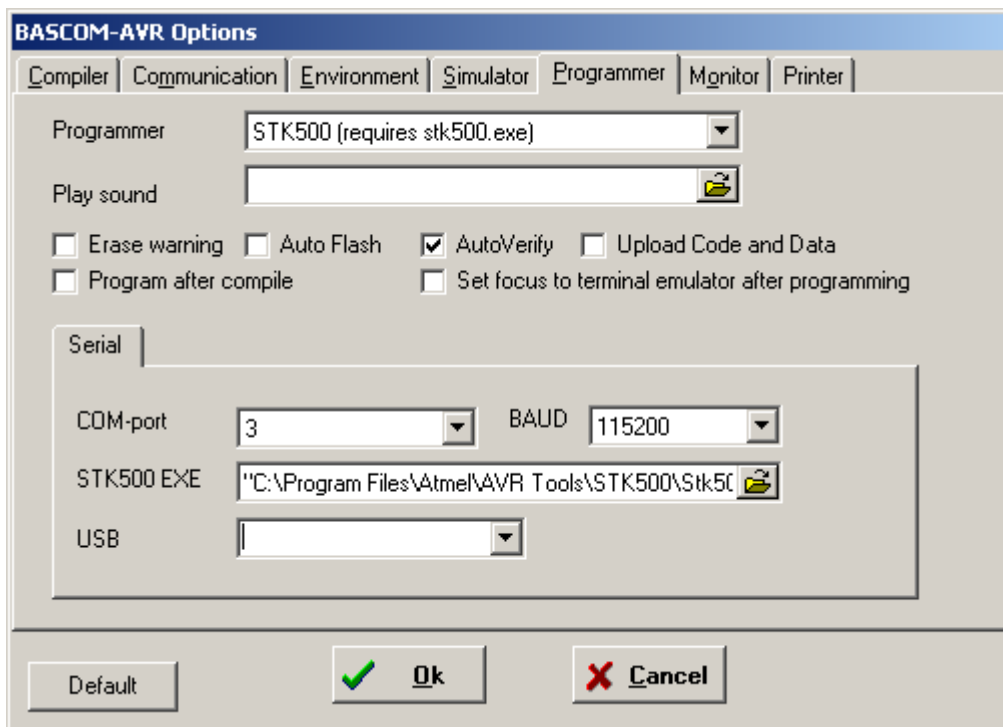
Write F6E

Write PRG

layout.bin FLASH : 42 KB EEPROM : 4 KB

STK500

Aby pracować w tym trybie na komputerze musi być zainstalowane środowisko AVR Studio (bascom do programowania będzie korzystał z bibliotek tego środowiska).



W oknie konfiguracji programatora (*BASCOM-AVR->Options->Programmer*) jako *Programmer* należy wybrać *STK500 (requires stk500.exe)*. Następnie należy wprowadzić numer portu COM na którym zainstalowany jest programator (pole *COM-port*) oraz ustawić prędkość *BAUD* na 115200 (Sprawdzenie oraz zmiana numeru portu COM na którym znajduje się programator opisane jest w rozdziale *Konfiguracja USB Serial Port – zmiana numeru COM*). W trybie tym należy również uzupełnić ścieżkę do pliku *stk500.exe* (pole *STK500 EXE*). Zazwyczaj plik ten znajduje się w lokalizacji "C:\Program Files\Atmel\AVR Tools\STK500".

External programmer

W trybie tym mamy możliwość wskazania zewnętrznej aplikacji którą BASCOM będzie wywoływał podczas programowania. W przykładzie posłużę się programem AVRDUDE. Programator powinien mieć wgrany wsad STK500.

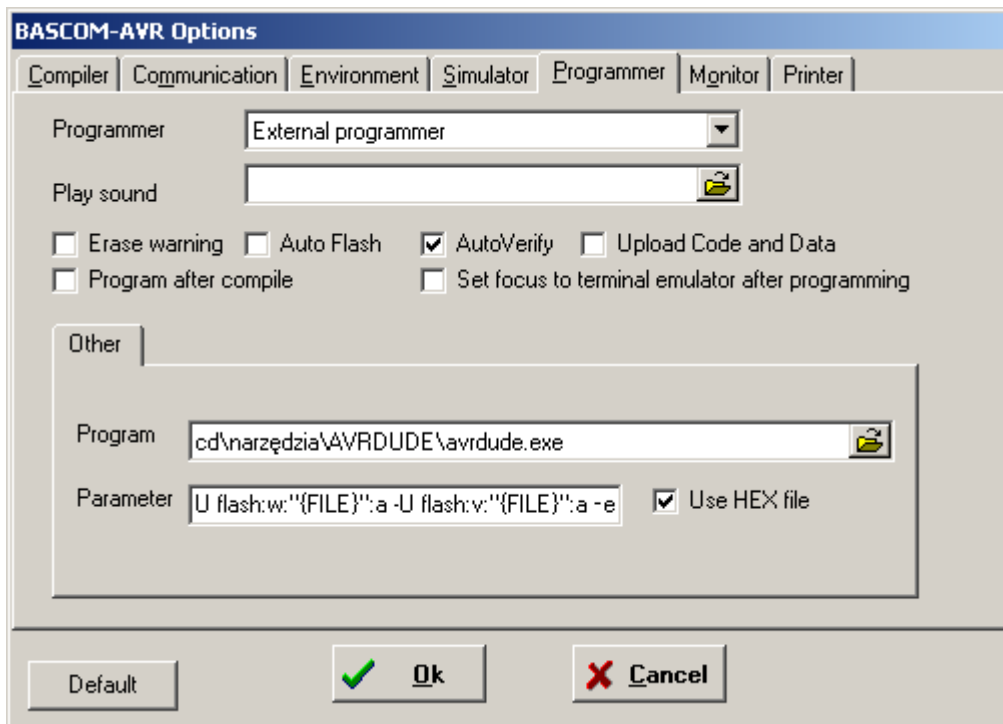
Ważne jest by korzystać z AVRDUDE dostarczonego z programatorem ze względu na specyficzne ustawienia pliku konfiguracyjnego. Tak więc katalog AVRDUDE z płyty CD należy skopiować na dysk twardy.

W polu *'Program'* podajemy ścieżkę do programu avrdude.exe, natomiast w *'Parameter'* wpisujemy (można wkleić):

```
"avrdude" -p m128 -c stk500v2 -P com3 -U flash:w:"{FILE}":a -U flash:v:"{FILE}":a -e
```

Oczywiście zamiast m128 musi być wpis odpowiedni dla programowanego procesora np.dla Atmega 8 będzie m8 , dla ATmega16 będzie m16 , itd.

W miejscu com3 wpisujemy numer portu na którym zainstalowany jest programator. Ponadto zaznaczamy *'Use HEX file'*



Po takiej konfiguracji programator jest gotowy do pracy.

Tradycyjnie, w celu zaprogramowania wybieramy *'Send to chip'* lub naciskamy F4.

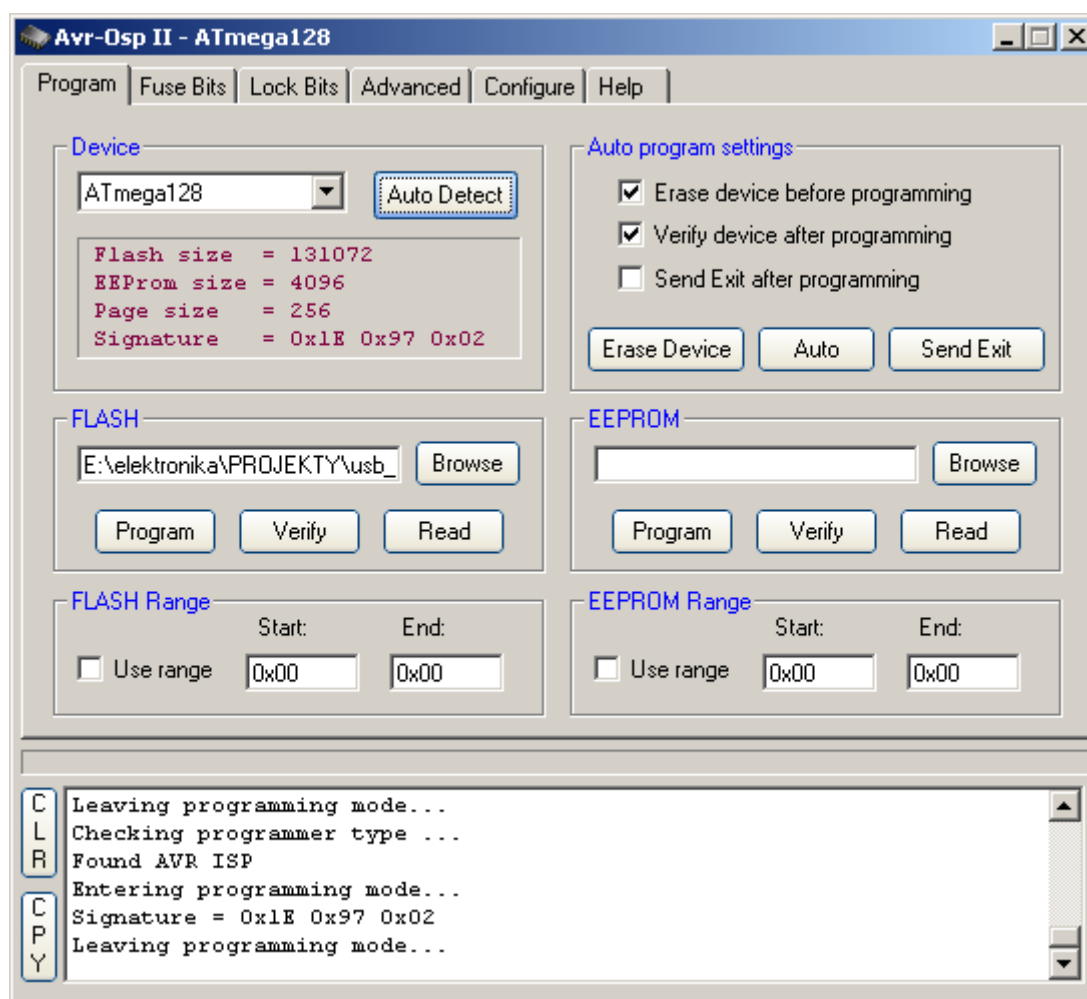
9.6 AVR OSP II

Programator z wsadem AVR910.

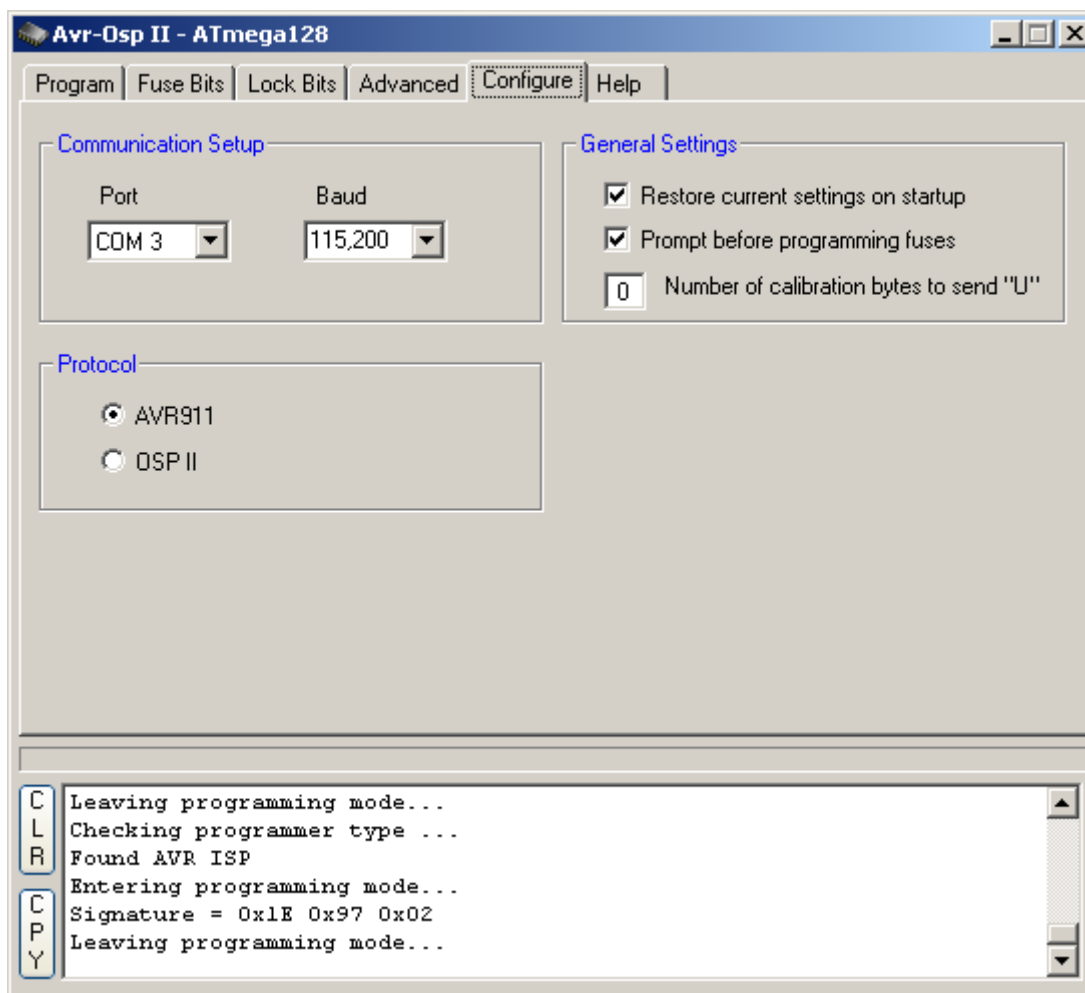
Bardzo przyjemny i intuicyjny interfejs oraz rozbudowana lista obsługiwanych mikrokontrolerów sprawiają iż warto od czasu do czasu sięgnąć do programu AVR OSP II. Aplikacja obsługuje programatory z implementacją protokołu AVR910.

Jedynym mankamentem aplikacji jest trochę powolne (w porównaniu do np. AVR Studio) programowanie. Jeśli jednak plik wsadowy jest niewielkich rozmiarów różnica może być mało odczuwalna a korzyści płynących z przejrzystej obsługi zmiany fusebitów nikomu nie trzeba tłumaczyć.

Przed uruchomieniem, folder z aplikacją należy skopiować na dysk twardy komputera (CD\narzędzia\AvrOspII). Po kliknięciu na plik AvrOspII.exe otwarte zostanie główne okno aplikacji.



Przed przystąpieniem do programowania należy przejść do zakładki „Configure” i w polu „Communication Setup” ustawić numer portu na którym zainstalowany jest programator oraz prędkość transmisji „Baud” na 115200.



Warto również w ustawieniach portu COM zmniejszyć czas opóźnienia – poprawi to znacznie prędkość programowania. Zmiana ta opisana jest w rozdziale *Konfiguracja USB Serial Port – zmiana czasu opóźnienia*.

Poprawność zestawionego połączenia można sprawdzić wciskając w zakładce "Program" przycisk "Auto Detect". W polu "Device" powinna pojawić się nazwa podłączonego do programatora mikrokontrolera.

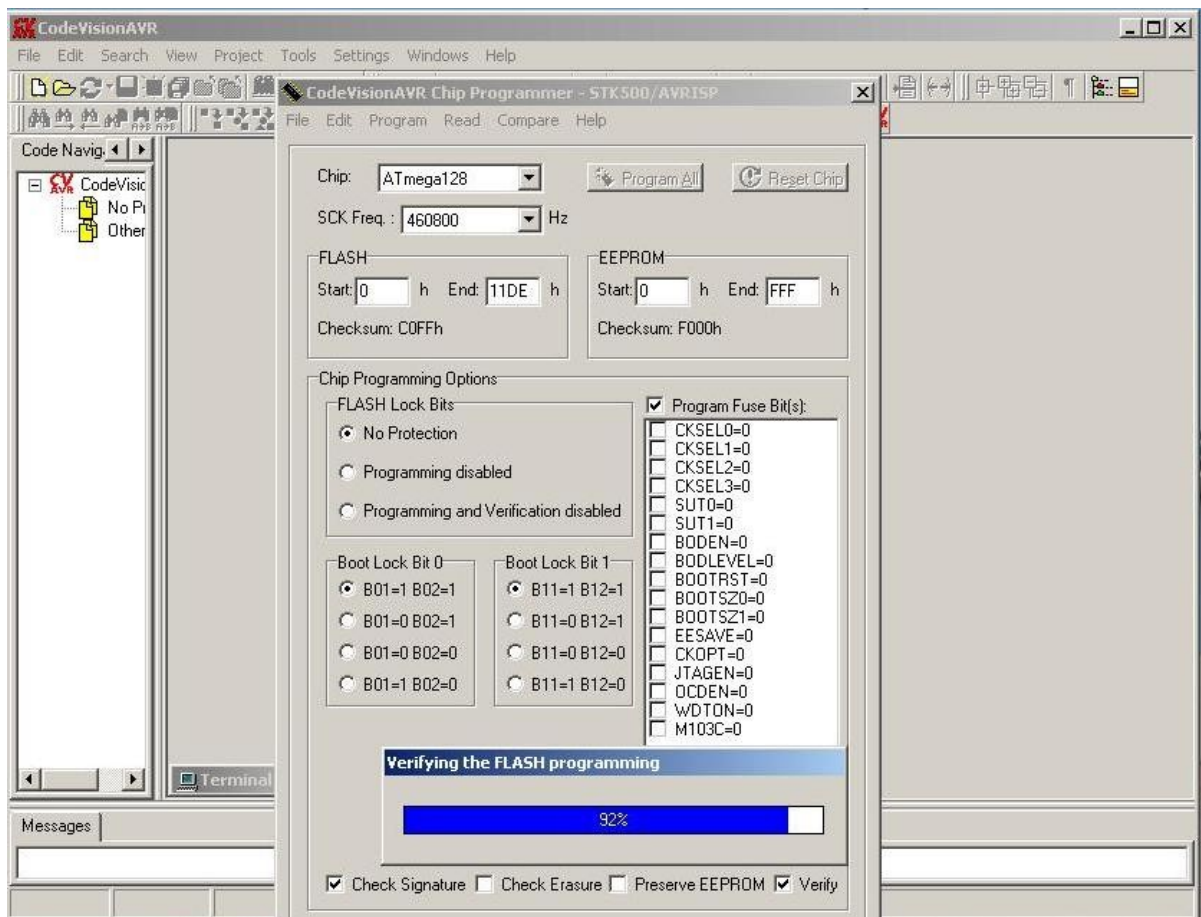
9.7 CodeVisionAVR

Do programatora należy wgrać firmware STK500.

W *Programmer Settings* w polu *AVR Chip Programmer Type* wybieramy opcję *Atmel STK500/AVRISP*. W polu *Communication Port* należy wskazać numer portu COM na którym zainstalowany jest programator.

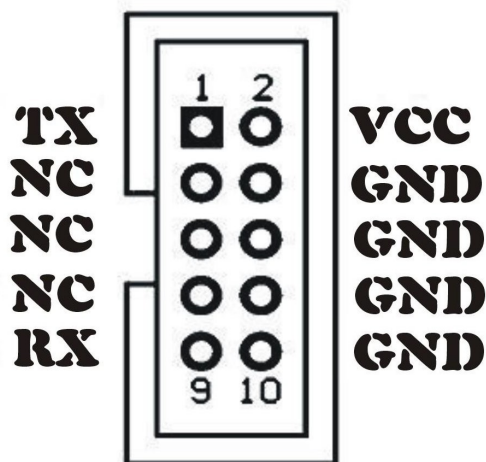


Po tych ustawieniach aplikacja jest przygotowana do współpracy z programatorem.



10. Tryb interfejsu USB<>RS232 TTL

Po wgraniu firmware RS232 (*rs232_1.hex*) programator może pełnić rolę przejściówki USB<>RS232 TTL z optoizolacją. Funkcję sygnałów RX TTL i TX TTL pełnią wtedy piny MISO i MOSI. Pin nr 1 (MOSI) to TX natomiast pin nr 9 (MISO) to RX.



Do prawidłowego działania interfejsu niezbędne jest zasilenie części izolowanej programatora. Napięcie powinno być doprowadzone do pinów VCC i GND. Wartość napięcia powinna zawierać się w przedziale od 2,1V do 6V.

11. Kontakt - wsparcie techniczne

W celu uzyskania dodatkowej pomocy technicznej skontaktuj się z nami serwis@sibit.pl

Windows is a registered trademark of Microsoft Corporation.
 ATMEL, AVR and AVR Studio are the registered trademarks of Atmel.
 BASCOM-AVR is trademark of MCS Electronics.
 CodeVisionAVR is trademark of HP InfoTech.

All other brand names or product names are the property of their respective holders.