

USB AVR BOX II

DEBUGGER & ISP PROGRAMMER

REV. 1.2

1. Wstęp	2
2. Opis wyprowadzeń	3
3. Programator ISP	5
3.1 Opis złącza ISP	5
3.2 Zmiana firmware	7
3.3 Lista obsługiwanych procesorów	9
3.4 Konfiguracja oprogramowania	12
4. Debugger JTAG ICE	18
4.1 Opis złącza JTAG	19
4.2 Zmiana firmware	20
4.3 Opcje debuggera	21
5. Instalacja sterowników	22
6. Konfiguracja USB Serial Port	25
8. Tryb interfejsu USB\leftrightarrowRS232	26

AVR BOX II to dwuprocessorowe urządzenie łączące funkcjonalność debuggera i programatora ISP mikrokontrolerów z rodziny AVR.

Sercem debuggera jest układ ATmega16. Debugger jest zgodny z atmelowskim AVR JTAG ICE dzięki czemu bezproblemowo współpracuje z AVR Studio. Przy pomocy interfejsu możemy przejąć kontrolę nad rdzeniem procesora oraz jego peryferiami, wymusić pracę krokową, ustawiać pułapki sprzętowe oraz programowe, podglądać i modyfikować stany rejestrów, zawartość pamięci itp.

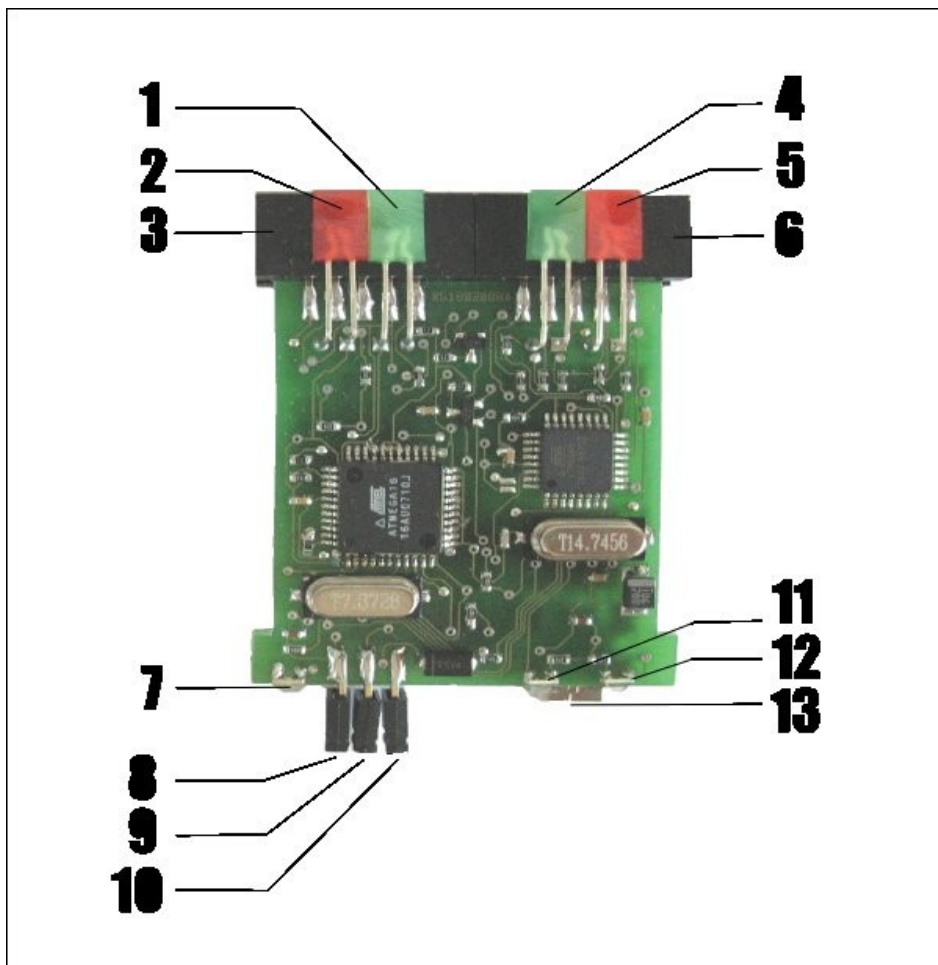
Programator oparty jest o mikrokontroler ATmega8. Aktualne wersje firmware-u pozwalają na obsługę programatora w trybach STK500v2, AVR910 oraz RS232 TTL (sygnały Rx i Tx). Obsługa dwóch protokołów daje możliwość współpracy urządzenia z wieloma programami na różnych systemach operacyjnych.

Dzięki zastosowaniu bootloader-ów w sekcji programatora i debuggera urządzenie można łatwo aktualizować wgrywając nowe wersje firmware-u.

Komunikacja z komputerem odbywa się za pomocą magistrali USB z której pobierane jest również zasilanie debuggera/programatora. Zasilanie może być wyprowadzone do układu badanego. Obsługa magistrali USB w AVR BOX-ie zrealizowana jest za pomocą układu firmy FTDI. Sygnały debuggera i programatora wyprowadzone są 10-cio pinowymi gniazdami BH zgodnymi ze standardami.

W celu zwiększenia odporności na zakłócenia, zabezpieczenia sygnałów oraz poszerzenia zakresu tolerowanych napięć (1,8V-6V) w obu sekcjach urządzenia zastosowano drivery HC245. Nowością jest wyprowadzenie dodatkowego autonomicznego sygnału zegarowego. Sygnał może być przydatny do taktowania programowanego mikrokontrolera w przypadku „zablokowania” niepoprawnie ustawionymi fuse-bitami (mikrokontroler ustawiony na taktowanie z zewnętrznego źródła).

2 OPIS WYPROWADZEŃ



rys.1

- 1 - dioda JTAG LED sygnalizująca stan pracy debuggera
- 2 - dioda JTAG ON; świecenie tej diody oznacza przełączenie urządzenia do trybu debuggera (złącze 3 rys.1)
- 3 - złącze debuggera JTAG ICE
- 4 - dioda ISP LED sygnalizująca stan pracy programatora
- 5 - dioda ISP ON; świecąca dioda oznacza przełączenie urządzenia w tryb programatora ISP (złącze 6 rys.1)
- 6 - złącze programatora ISP
- 7 - dioda POWER LED sygnalizująca zasilanie głównej sekcji AVR BOX-a
- 8 - zworka MODE, służy do przełączania trybu pracy urządzenia; zworka założona uaktywnia tryb JTAG-a (złącze 3 rys.1) natomiast zworka zdjęta przełącza urządzenie w tryb programatora ISP (złącze 6 rys.1)
- 9 - zworka BOOT; założenie zworki i reset urządzenia (wył/wł usb lub zmiana trybu zworką MODE) uruchamia bootloader; sygnalizowane jest to miganiem diody ISP LED lub JTAG LED w zależności od trybu pracy

10 - zworka POWER; zworka założona - zasilanie interfejsu z USB komputera; w tym przypadku układ badany możemy zasilać z pinów VCC programatora/debuggera; wysokość tego napięcia zależy od komputera i wg standardu USB może się zawierać w przedziale 4,3V – 5,25V; zworka zdjęta – zasilanie obwodów wejściowych (HC245) AVR BOX-a z podłączonego mikroprocesora.

UWAGA ! JEŚLI UKŁAD MIKROPROCESOROWY POSIADA WŁASNE ZASILANIE ZWORKA POWER POWINNA BYĆ ZDJĘTA A NAPIĘCIE ZASILANIA PODŁĄCZONEGO MIKROPROCESORA DOPROWADZONE DO PINU VCC.

11 - dioda TX; dioda sygnalizuje transmisję w kierunku komputer -> programator
12 - dioda RX; dioda sygnalizuje transmisję w kierunku programator -> komputer
13 - złącze USB mini B do podłączenia z komputerem PC

3 PROGRAMATOR ISP

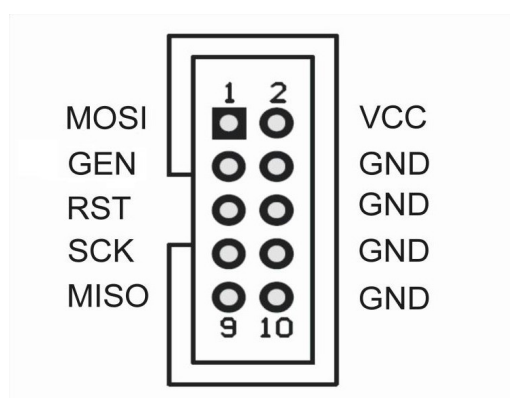
Aby urządzenie działało w trybie programatora ISP zworka MODE (8 rys.1) powinna być zdjęta. Przełączenie AVR BOX-a do trybu programatora sygnalizowane jest świeceniem diody ISP ON (5 rys.1). Programowany układ podłączamy do złącza ISP (6 rys.1) przy pomocy załączonej 10-cio żyłowej taśmy zakończonych standardowymi wtykami IDC. W przypadku korzystania z innego przewodu należy pamiętać iż długość nie powinna być większa niż 40cm. System mikroprocesorowy powinien posiadać złącze ISP z układem wyprowadzeń identycznym ze złączem ISP w AVR BOX. Złącze w AVR BOX jest kompatybilne ze standardem 10-cio wyprowadzeniowym firmy ATMEL (rys.2).

Istnieją dwa warianty zasilania programatora, ustawiane za pomocą zworki POWER (10 rys.1). Przy zworze założonej programator jest zasilany z USB komputera. W tym przypadku układ mikroprocesorowy możemy również zasilać z USB – pin VCC programatora (rys.2). Przy zdjętej zworze POWER zasilanie układów buforujących debuggera jest pobierane z programowanego mikroprocesora a więc powinno być doprowadzone do pinu VCC (rys.2)

**UWAGA ! JEŚLI UKŁAD BADANY POSIADA WŁASNE ZASILANIE
ZWORKA W PROGRAMATORZE POWINNA BYĆ ZDJĘTA A NAPIĘCIE
ZASILANIA PODŁĄCZONEGO UKŁADU DOPROWADZONE DO PINU
VCC ZŁĄCZA ISP W AVR BOX**

Od strony komputera AVR BOX podłączamy do wolnego portu USB. Gotowość programatora sygnalizuje świecąca się dioda ISP LED (4 rys.1). Jeśli jest to pierwsze podłączenie do komputera system operacyjny wykryje urządzenie i poprosi o sterowniki (proces instalacji sterowników został opisany w innej części dokumentu).

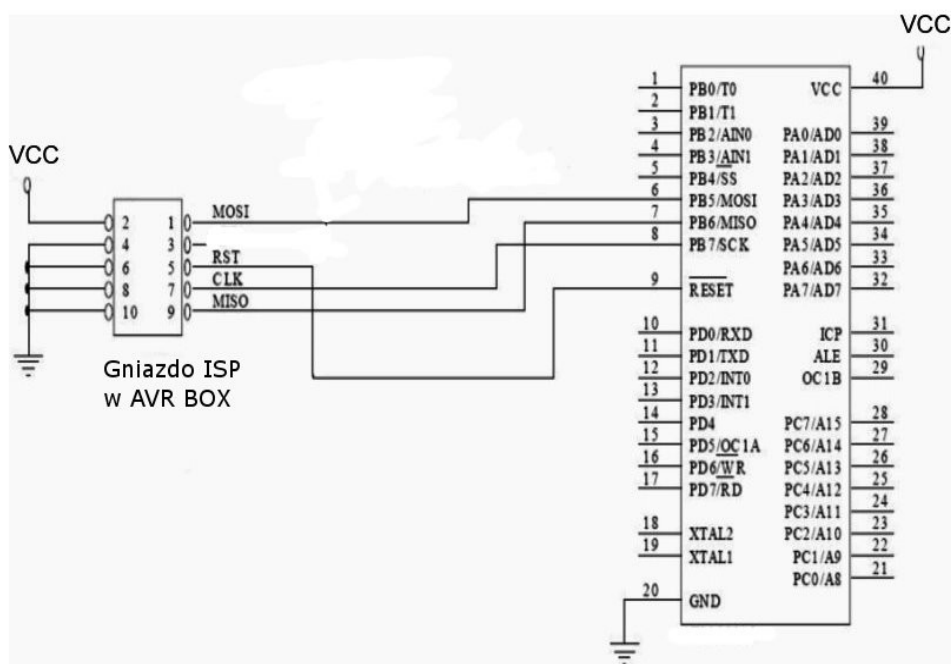
3.1 OPIS ZŁĄCZA ISP



rys.2 Opis złącza ISP w AVR BOX.

MOSI – linia danych, połączyć z linią MOSI mikrokontrolera
RST - podłączyć do linii RESET mikrokontrolera
SCK - sygnał zegarowy; podłączyć do linii SCK mikrokontrolera
MISO – linia danych, podłączyć do linii MISO mikrokontrolera
GND – masa; podłączyć z masą układu programowanego
VCC – jeśli zworka POWER (10 rys.1) jest zdjęta to do tego pinu musi być doprowadzone napięcie zasilania z programowanego mikrokontrolera (1,8V – 6V), jeśli zworka jest założona wtedy z pinu VCC możemy zasilać programowany mikrokontroler (ok. 5V max. 400 mA).
GEN - sygnał zegarowy. Sygnał ten może być wykorzystywany podczas przeprogramowywania procesorów z ustawionymi fuse-bitami na taktowanie zewnętrzne.
 W takim przypadku sygnał GEN powinien być podłączony do wejścia EXTERNAL CLOCK SIGNAL mikrokontrolera - przeważnie jest to wejście XTAL1 mikrokontrolerów ATmega.
 Podczas standardowego programowania sygnał GEN nie musi być podłączony.

Gotowość do programowania oznajmia świecąca się dioda ISP LED (4 rys.1).
 Podczas operacji programowania dioda ta gaśnie.
 Linie magistrali ISP łączymy z odpowiadającymi im liniami magistrali SPI mikrokontrolera, jak pokazano na poniższym rysunku:



rys.3 Przykładowe podłączenie programatora do programowanego procesora.

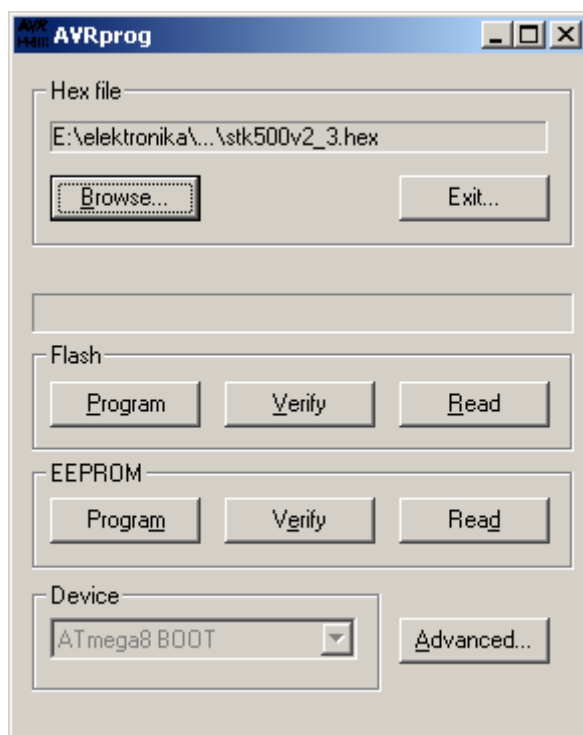
3.2 ZMIANA FIRMWARE

W zależności od wgranego firmware programator może się znajdować w jednym z trzech trybów – STK500v2, AVR910 lub RS232 TTL (tylko sygnały RX i TX zgodne z logiką TTL). Wsady znajdują się na dołączonym CD w katalogu *'firmware\programmer'*.

Standardowo programator dostarczony jest z wgranym wsadem STK500v2.

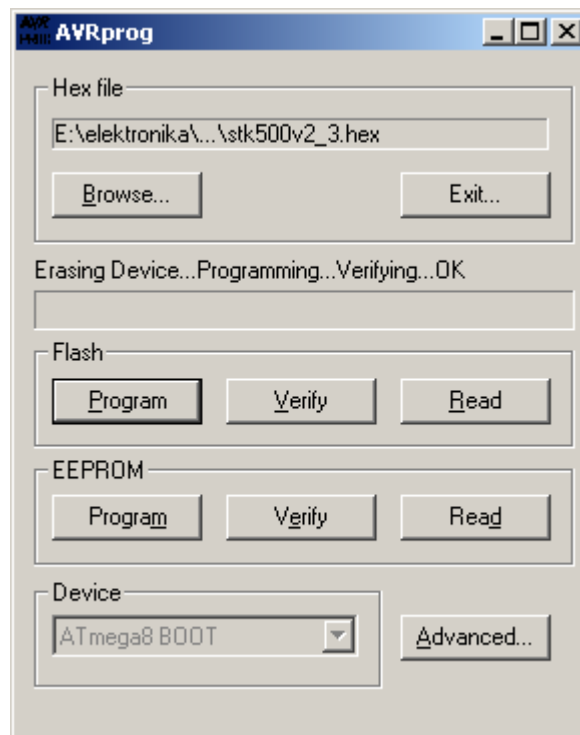
Aby zmienić firmware należy uruchomić bootloader i wgrać wsad przy pomocy programu AvrProg. Oto instrukcja postępowania:

- odłączyć (jeśli był podłączony) programator od USB komputera
- zdjąć zworkę MODE (8 rys.1) - interfejs w trybie ISP
- założyć zworkę BOOT (9 rys.1)
- podłączyć programator do USB komputera
- programator powinien uruchomić się w trybie bootloadera o czym świadczy miganie na zielono diody ISP LED (4 rys.1)
- uruchomić program AvrProg ('*CD:\tools\AvrProg.exe*'). Po uruchomieniu programu powinna przestać migać dioda ISP LED (2 rys.1). Jeśli zamiast programu pojawia się okno z informacją *'No supported board found !'* należy sprawdzić na którym porcie COM zainstalowany jest programator. Dla AvrProg-a wymagane jest umiejscowienie na COM1, COM2 lub COM3 (Zmiana konfiguracji opisana jest w punkcie *Konfiguracja USB Serial Port*). Po uruchomieniu programu pojawi się okno:



rys.4

- klikając przycisk 'Browse...' należy wskazać odpowiedni wsad czyli plik, np. stk500v2_10.hex ('CD:\firmware\programmer\stk500v2_10.hex') lub avr910_2.hex ('CD:\firmware\programmer\avr910_2.hex')
- w sekcji 'Flash' programu nacisnąć przycisk 'Program'
- o pomyślnym przebiegu operacji poinformuje nas komunikat 'Erasing Device...Programming...Verifying..OK' nad wskaźnikiem postępu :



rys.5

- po skończonej operacji upgrade'u zamykamy program (krzyżyk w górnym prawym rogu), zdejmujemy zworkę BOOT (9 rys.1) oraz odłączamy programator od USB komputera
- po podłączeniu do USB, programator jest gotowy do pracy.

3.3 LISTA OBSŁUGIWANYCH PROCESORÓW

MIKROKONTROLER	TRYB PRACY (FIRMWARE)	
	STK500V2	AVR910
AT89S8252		X
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	

rys.6

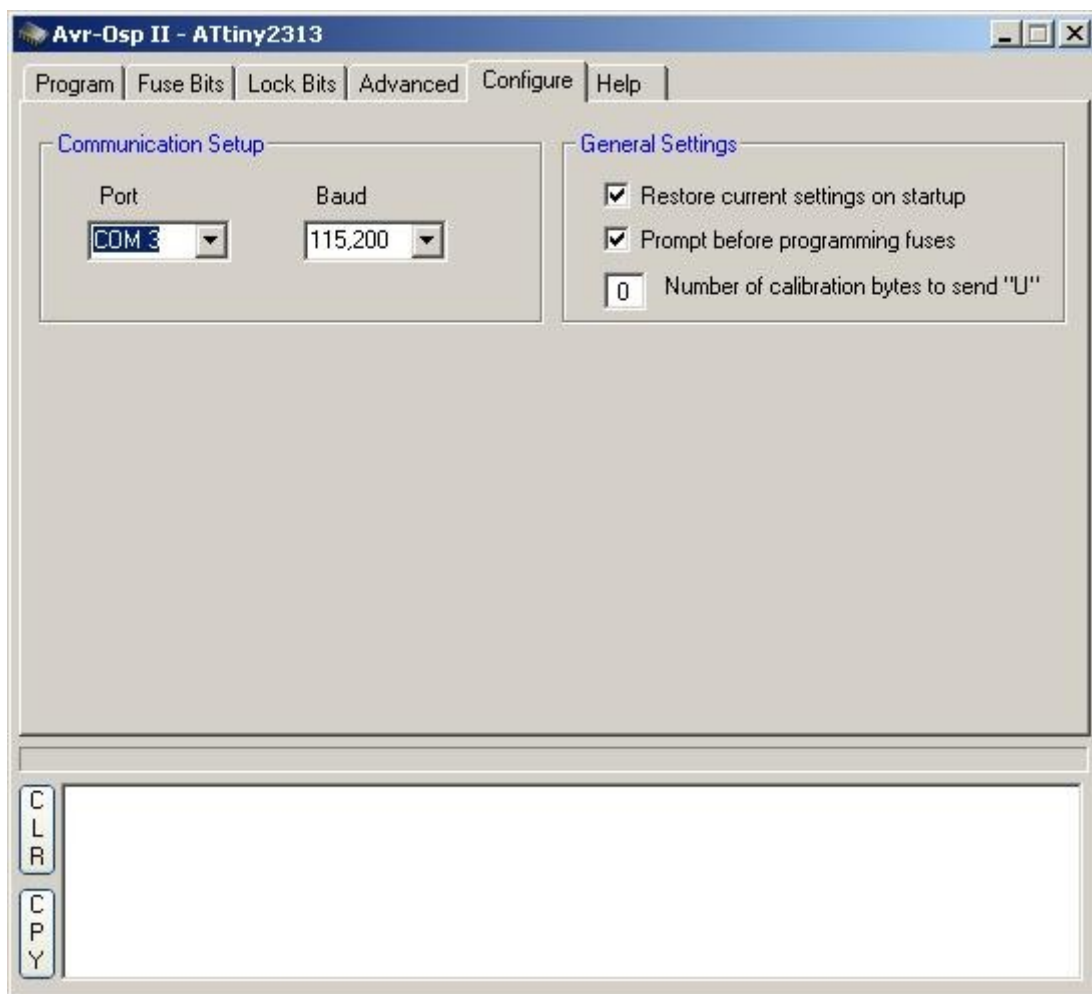
Oczywiście wersje niskonapięciowe ('L') tych procesorów też są obsługiwane.

3.4 KONFIGURACJA OPROGRAMOWANIA

- **AVR-OSP II
(PROGRAMATOR Z WSADEM AVR910)**

W zakładce „Configure” w „Communication Setup” należy ustawić numer portu na którym zainstalowany jest programator oraz prędkość transmisji (aby sprawdzić gdzie zainstalował się nasz programator otwieramy *Menadżer urządzeń* i, rozwijamy gałąź *Porty(COM i LPT)*. Interesuje nas pozycja ‘*USB Serial Port*’ z numerem *COM* w nawiasie).

Następnie należy ustawić prędkość „*Baud*” na 115200 (tak jak na rys.7).

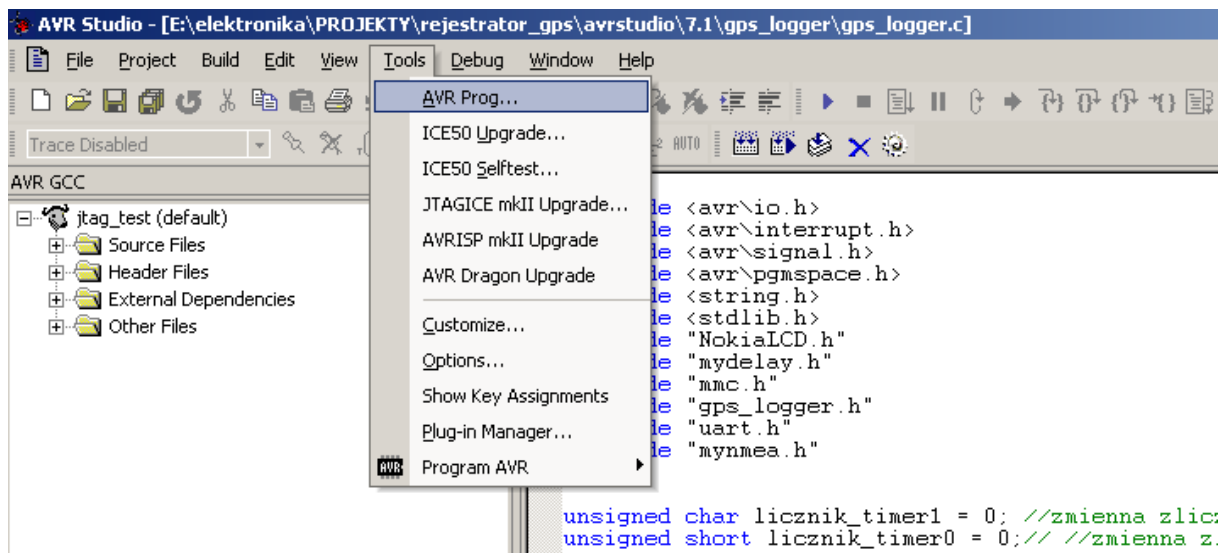


rys.7

- **AVRprog (AVR Studio)**
(PROGRAMATOR Z WSADEM AVR910)

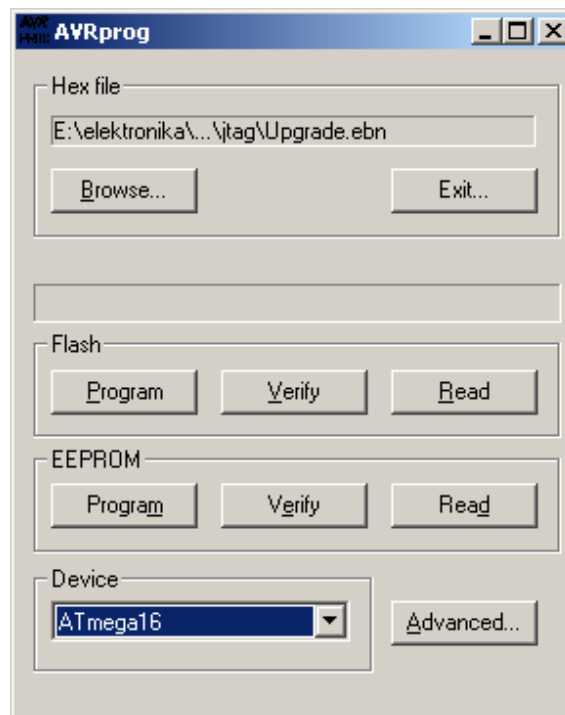
AVRprog może być wywoływany z poziomu środowiska lub jako samodzielna aplikacja.

Z poziomu studia AVRprog uruchamiamy wybierając pierwszą od góry pozycję w zakładce „Tools”.



rys.8

Jeśli programator jest podłączony i prawidłowo skonfigurowany pojawi się główne okno aplikacji AVRprog:

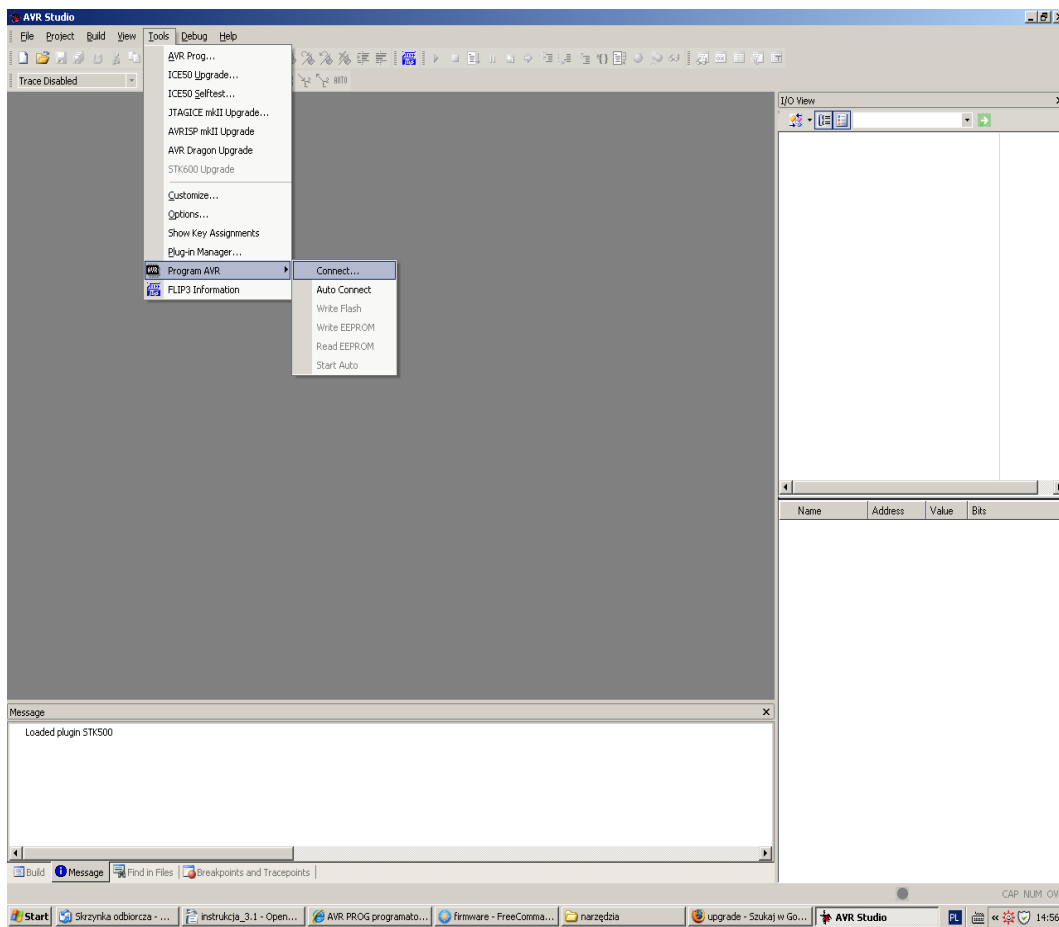


rys.9

Aby zaprogramować pamięć flash procesora należy :

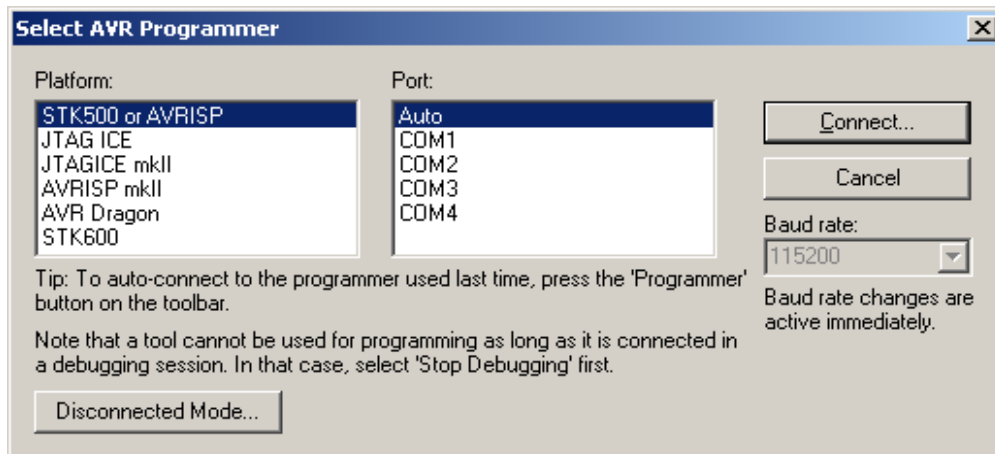
- podać ścieżkę do pliku z programem w formacie „hex” (przycisk „Browse”)
- wybrać odpowiedni procesor (pole „Device”)
- wcisnąć „Program” w polu „Flash”

- ***AVRISP (AVR Studio)***
(PROGRAMATOR Z WSADEM STK500V2)



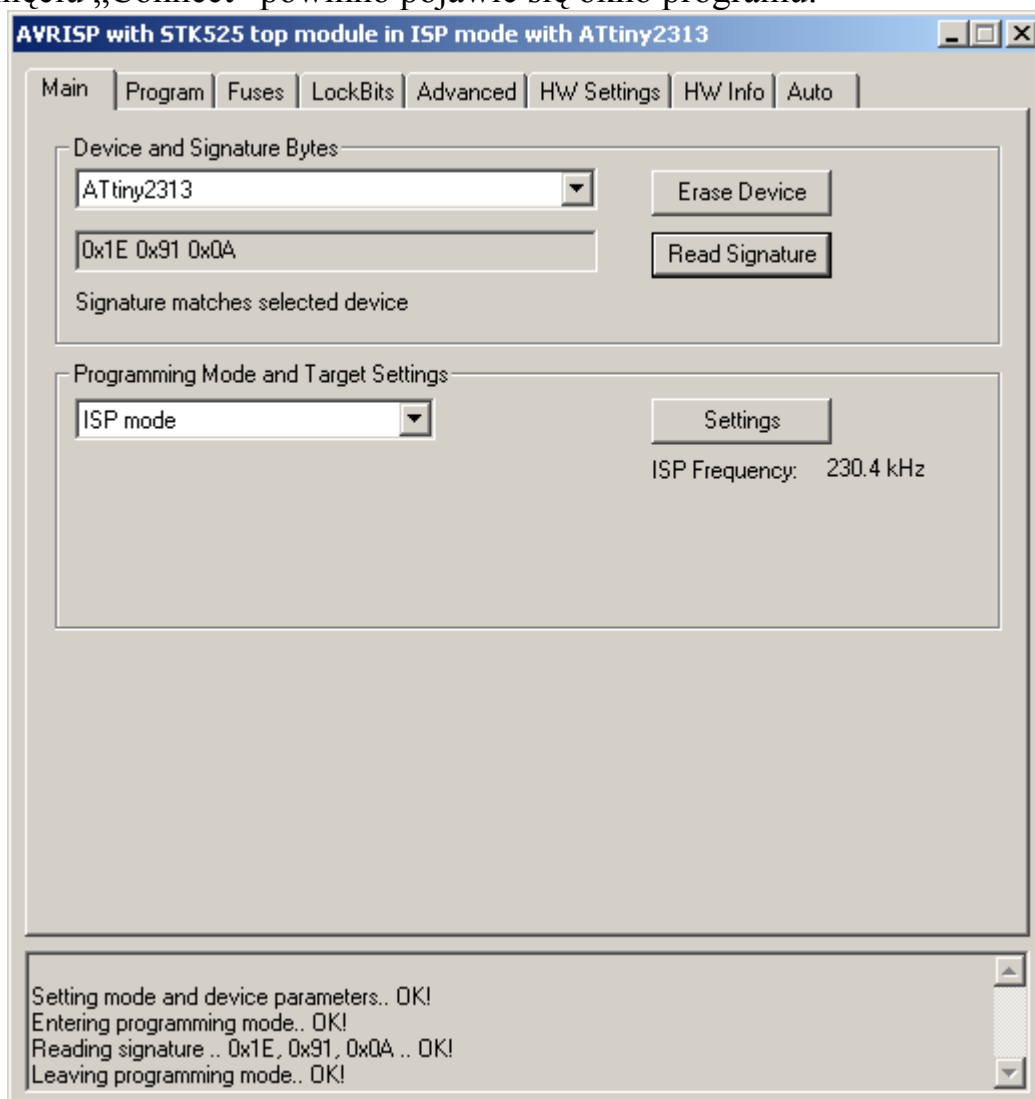
rys.10

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



rys.11

Po kliknięciu „Connect” powinno pojawić się okno programu:

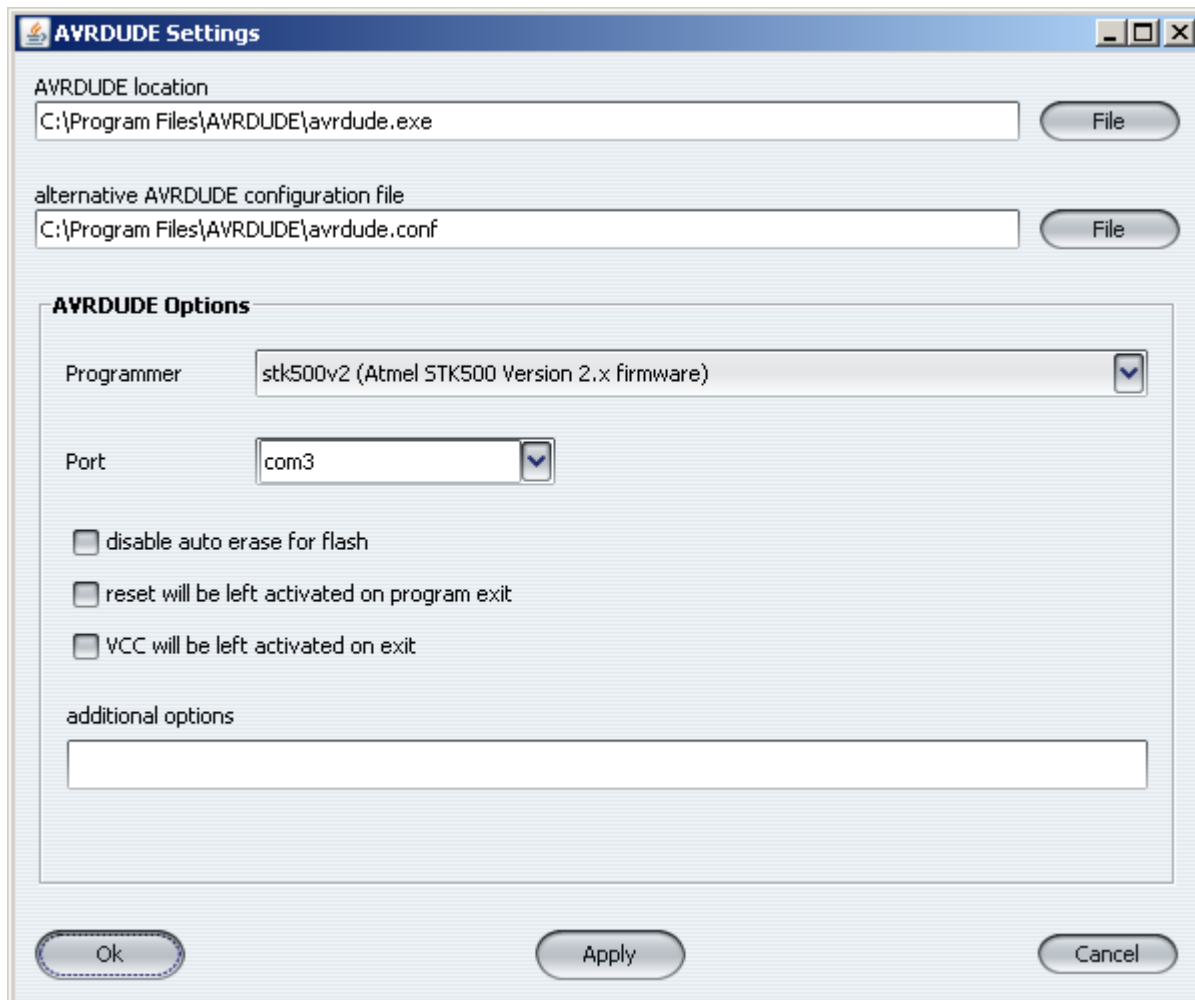


rys.12

W polu „Device and Signature Bytes” należy wybrać mikrokontroler który będziemy programować.

- **AVRDUDE + AVR8 Burn-O-Mat
(PROGRAMATOR Z WSADEM STK500V2)**

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



rys.13

W „AVRDUDE location” podać lokalizację pliku avrdude.exe

W „alternative AVRDUDE configuration file” należy podać lokalizację pliku avrdude.conf.

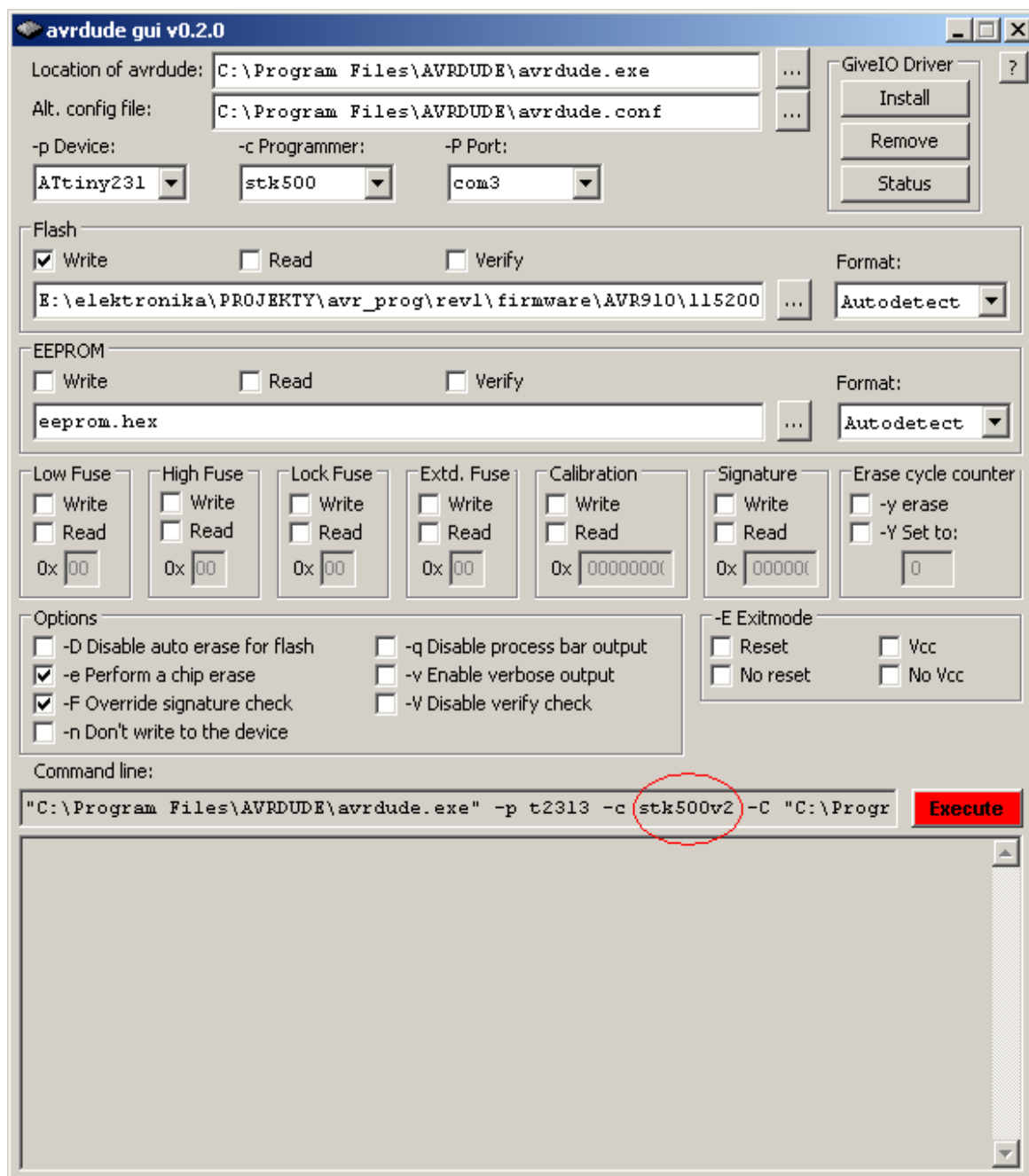
Następnie zamknąć program. Po ponownym uruchomieniu gui jeszcze raz przejść do AVRDUDE Settings i w „Programmer” wybrać „stk500v2...” oraz numer portu com zajmowanego przez programator. Zaakceptować ustawienia i nacisnąć „Ok”.

W głównym oknie programu wybrać rodzaj programowanego mikrokontrolera.

- **AVRDUDE + AVRDUDE-GUI
(PROGRAMATOR Z WSADEM STK500V2)**

Wskazujemy lokalizację plików „avrdude.exe” oraz „avrdude.conf”.

Wybieramy rodzaj programowanego mikrokontrolera, numer portu com na którym jest programator a jako „-cProgrammer” wybieramy „stk500”. Uwaga ! Przed finalnym naciśnięciem „Execute” w „Command line” do „stk500” dostawiamy „v2” - tak jak na rysunku:



rys.14

4 DEBUGGER JTAG ICE

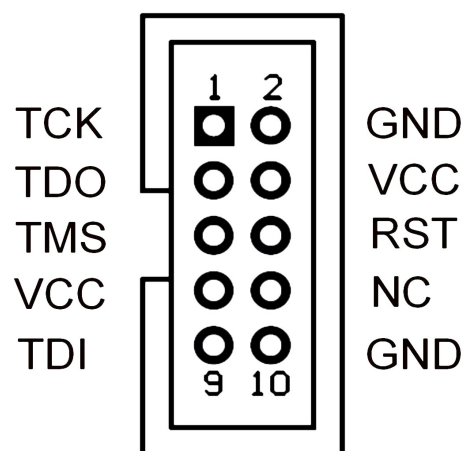
Aby urządzenie działało w trybie debuggera JTAG ICE zworka MODE (8 rys.1) powinna być założona. Przełączenie AVR BOX-a do trybu debuggera sygnalizowane jest świeceniem diody JTAG ON (2 rys.1). Programowany układ podłączamy do złącza JTAG (3 rys.1) przy pomocy załączonej 10-cio żyłowej taśmy zakończonej standardowymi wtykami IDC. W przypadku korzystania z innego przewodu należy pamiętać iż długość nie powinna być większa niż 40cm. System mikroprocesorowy powinien posiadać złącze jtag z układem wyprowadzeń identycznym ze złączem JTAG w AVR BOX.

Istnieją dwa warianty zasilania debuggera ustawiane za pomocą zworki POWER (10 rys.1) Przy zworce założonej debugger jest zasilany z USB komputera. W tym przypadku układ mikroprocesorowy możemy również zasilać z USB – piny VCC emulatora (rys.15). Przy zdjętej zworce POWER zasilanie układów buforujących debuggera jest pobierane z badanego mikroprocesora a więc powinno być doprowadzone do pinu VCC (rys.15)

UWAGA ! JEŚLI UKŁAD BADANY POSIADA WŁASNE ZASILANIE ZWORKA W PROGRAMATORZE POWINNA BYĆ ZDJĘTA A NAPIĘCIE ZASILANIA BADANEGO MIKROPROCESORA DOPROWADZONE DO PINU VCC W USB ICE

Od strony komputera emulator podłączamy do wolnego portu USB. Gotowość do pracy w trybie debuggera sygnalizuje świecąca się dioda JTAG LED (1 rys.1). Jeśli jest to pierwsze podłączenie do komputera system operacyjny wykryje urządzenie i poprosi o sterowniki (proces instalacji sterowników został opisany w dalszej części dokumentu). Jeśli sterowniki są zainstalowane można uruchomić AVR Studio. Środowisko wyświetli okno dialogowe w celu wybrania projektu. Należy wskazać projekt który chcemy debugować a następnie przejść dalej naciskając 'NEXT'. W kolejnym oknie jako 'Debug platform' wybieramy 'JTAG ICE' a jako 'Device' zastosowany w układzie badanym mikrokontroler. Ustawienia zatwierdzamy klikając 'NEXT'. System jest już gotowy do pracy.

4.1 OPIS ZŁĄCZA JTAG



rys.15 Opis wyprowadzeń złącza JTAG w AVR BOX

TCK – sygnał zegarowy
TDO – wyjście danych
TMS – sygnał przełączający
TDI – wejście danych
RST – sygnał reset
VCC – zasilanie układu docelowego
GND – masa układu

4.2 ZMIANA FIRMWARE

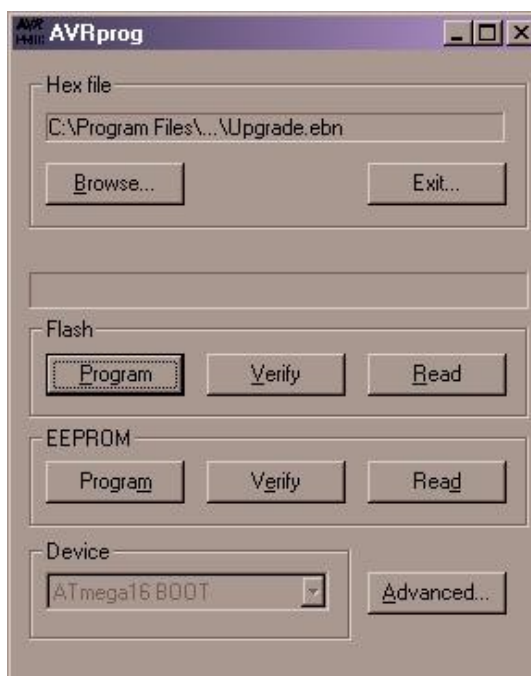
JTAG ICE pozwala na aktualizację firmware-u jeśli zainstalowana wersja AVR Studio jest niezgodna z debuggerem. Niekompatybilność wersji sygnalizowana jest po wybraniu sesji debugowania lub programowania wyświetleniem okna :



rys.16

Po wyświetleniu się takiego komunikatu interfejs JTAG ICE trzeba przełączyć w tryb bootloadera. Robimy to poprzez zresetowanie jtag-a (wył/wł usb lub przełączenie trybu zworką MODE) przy założonej zworce BOOT (9 rys.1) Przejście do sekcji bootloadera, jtag sygnalizuje miganiem diody JTAG LED (1 rys.1)

Teraz możemy wcisnąć przycisk OK (rys.16). Programator powinien zostać zdetektowany co skutkować będzie uruchomieniem okna programu AVRProg (rys.17).



rys.17

Program automatycznie wskazuje ścieżkę do aktualnego pliku z oprogramowaniem, tak więc nasza działalność ogranicza się do wciśnięcia przycisku 'Program' w sekcji 'Flash'. Jeśli z jakiejś przyczyny pole 'Hex file' jest puste lub zawiera ścieżkę do innego pliku niż 'Upgrade.ebn' należy ręcznie wskazać miejsce pliku. Plik 'Upgrade.ebn' jest umieszczony w miejscu instalacji środowiska AVR Studio, zazwyczaj jest to 'C:\Program Files\Atmel\AVR Tools\JTAGICE\Upgrade.ebn'. Po skończonej aktualizacji należy zamknąć aplikację AVRprog, zdjąć zworkę BOOT oraz zrestartować interfejs AVR BOX (odłączenie zasilania lub zmiana trybu zworką MODE).

4.3 OPCJE DEBUGGERA

Po naciśnięciu Alt+O lub wybraniu 'JTAG ICE Options' z menu 'Debug' mamy możliwość ręcznej konfiguracji parametrów pracy debugera. Dla zapewnienia optymalnej komunikacji pomiędzy interfejsem i układem badanym częstotliwość wpisana w polu 'JTAG port frequency' (zakładka 'Connection') powinna być równa lub mniejsza $\frac{1}{4}$ częstotliwości zegara procesora w badanym układzie. Aby zwiększyć prędkość komunikacji pomiędzy komputerem a interfejsem JTAG można zwiększyć częstotliwość portu COM z domyślnej wartości 19200 na 115200 kb/s. Wpłynie to korzystnie na komfort pracy z urządzeniem.

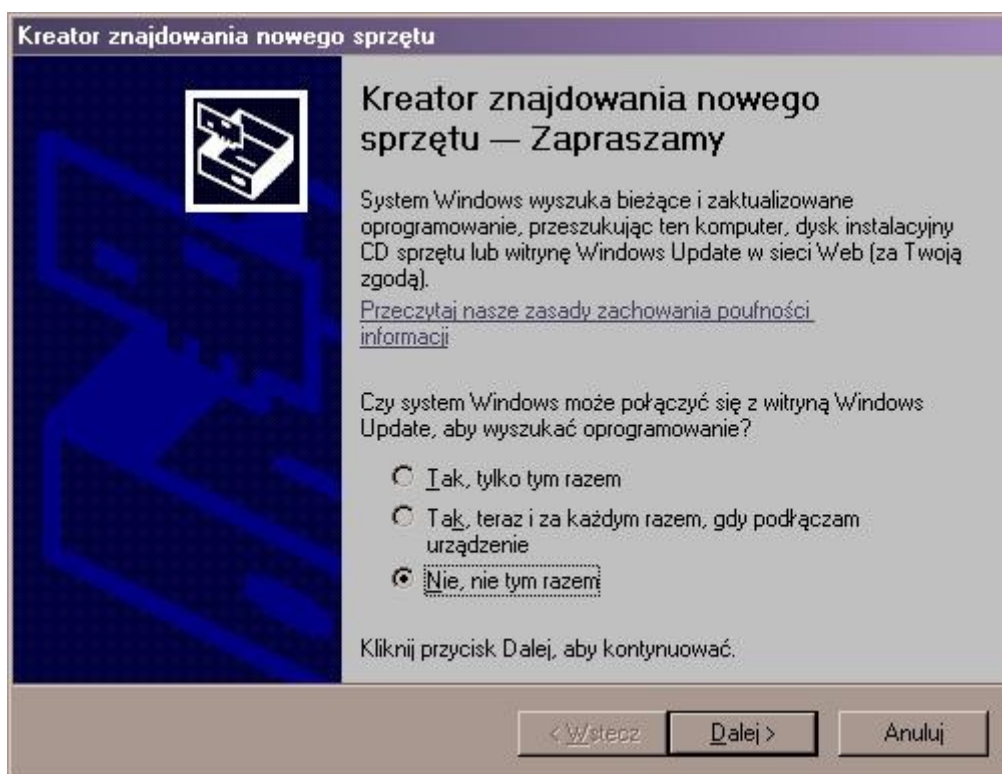
5. INSTALACJA STEROWNIKÓW

Opis procesu instalacji sterowników do FT232 w systemie Windows XP.
Urządzenia pierwszy raz podłączone do komputera zostanie wykryte jako AVR BOX (rys.18)



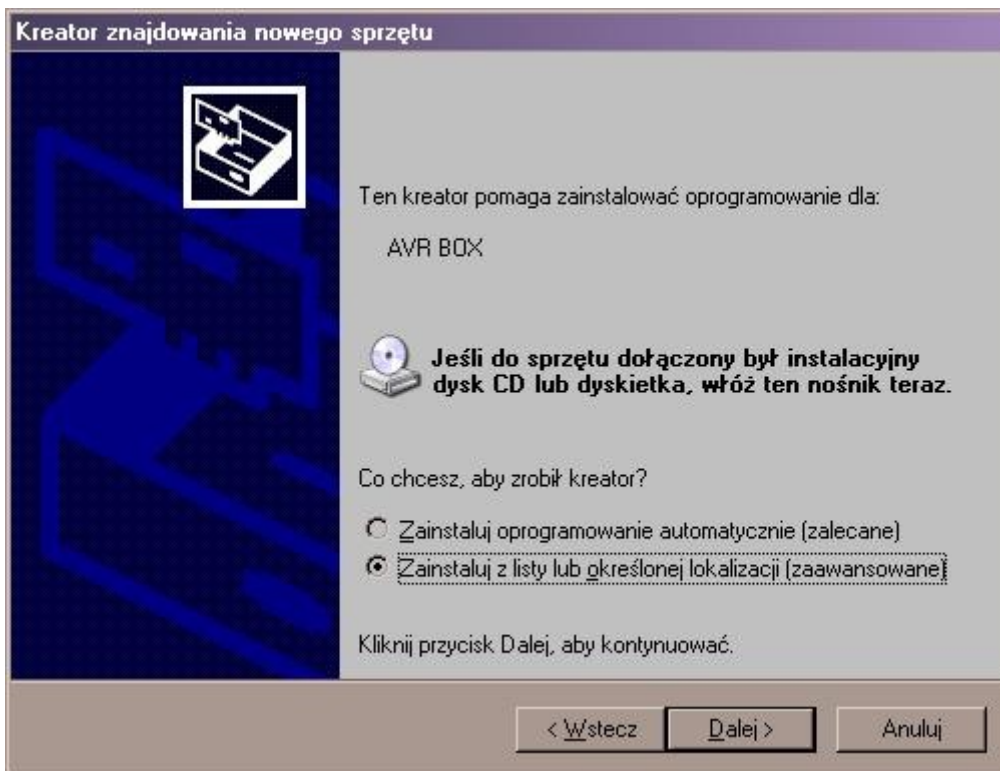
Rys.18

W kreatorze znajdowania nowego sprzętu wybieramy opcję ‘Nie, nie tym razem’ i klikamy ‘Dalej’ (rys.19)



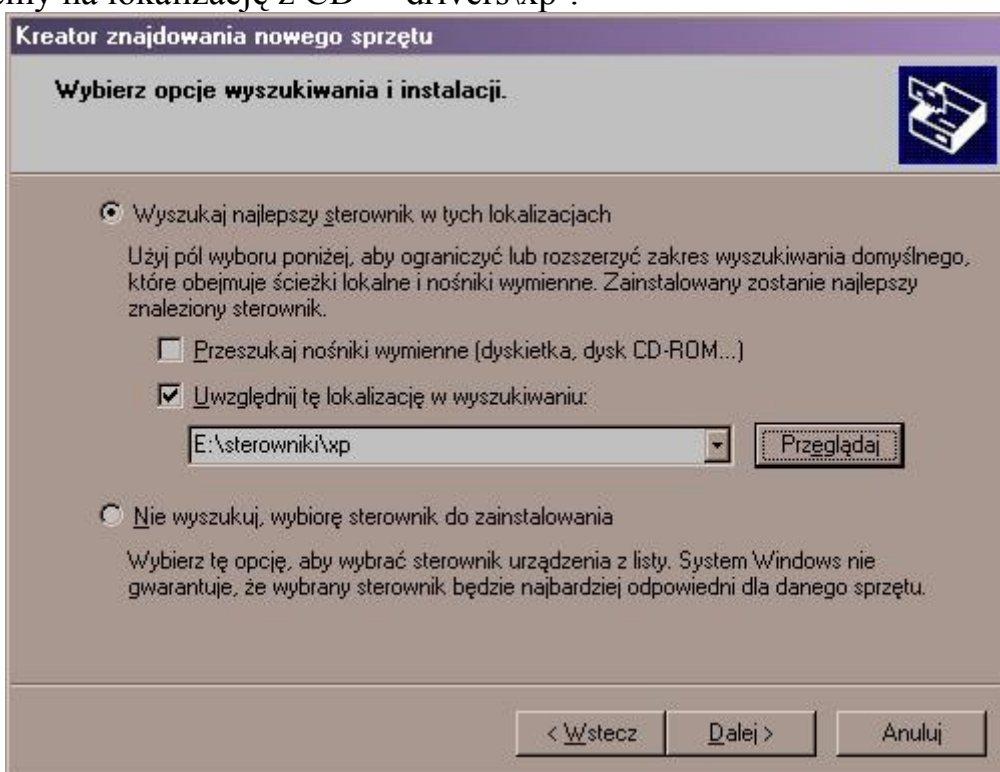
Rys.19

W następnej planszy (rys. 20) wybieramy opcję ‘Zainstaluj z listy lub określonej lokalizacji(zaawansowane)’ i wciskamy ‘Dalej’



Rys.20

Teraz musimy wskazać lokalizację plików. W kolejnej planszy która się pojawi (rys.21) wybieramy opcję 'Wyszukaj najlepszy sterownik w tych lokalizacjach' oraz 'Uwzględnij tę lokalizację w wyszukiwaniu.'. Tu wciskamy 'Przełącz' i wskazujemy na lokalizację z CD – 'drivers\xp'.



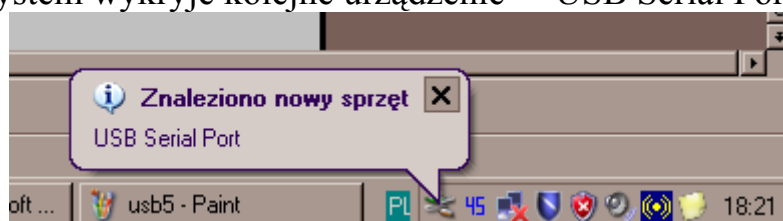
Rys.21

System powinien znaleźć i zainstalować sterowniki z podanej lokalizacji. Po wyświetleniu okna końca pracy kreatora (rys.22) należy nacisnąć ‘Zakończ’.



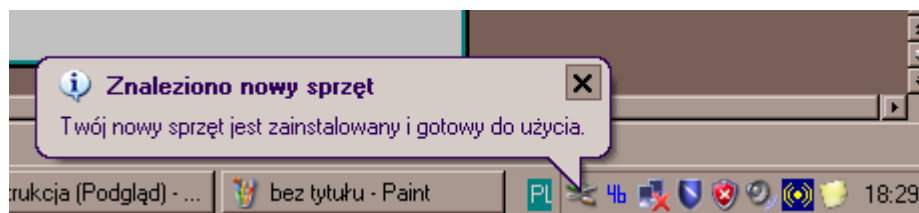
Rys.22

Chwilę potem system wykryje kolejne urządzenie – ‘USB Serial Port’ (rys.23)



Rys.23

Proces instalacji dla tego urządzenia przebiega identycznie jak poprzednio. W kolejnych etapach w kreatorze zaznaczamy te same opcje co poprzednio a do sterowników podajemy tą samą ścieżkę z dołączonej płyty CD tj. ‘\drivers\xp’. O prawidłowym zainstalowaniu sterowników poinformuje nas system komunikatem ‘Twój nowy sprzęt jest zainstalowany i gotowy do użycia’ (rys.24)



Rys.24

Uwaga! Aby AVR BOX był detektowany przez oprogramowanie na komputerze, USB Serial Port musi być zarejestrowany na którymś z wolnych portów od COM1 do COM4 - tylko na tych COM-ach wysyłana jest sekwencja rozpoznawcza przez AVR PROG czy AVR Studio.

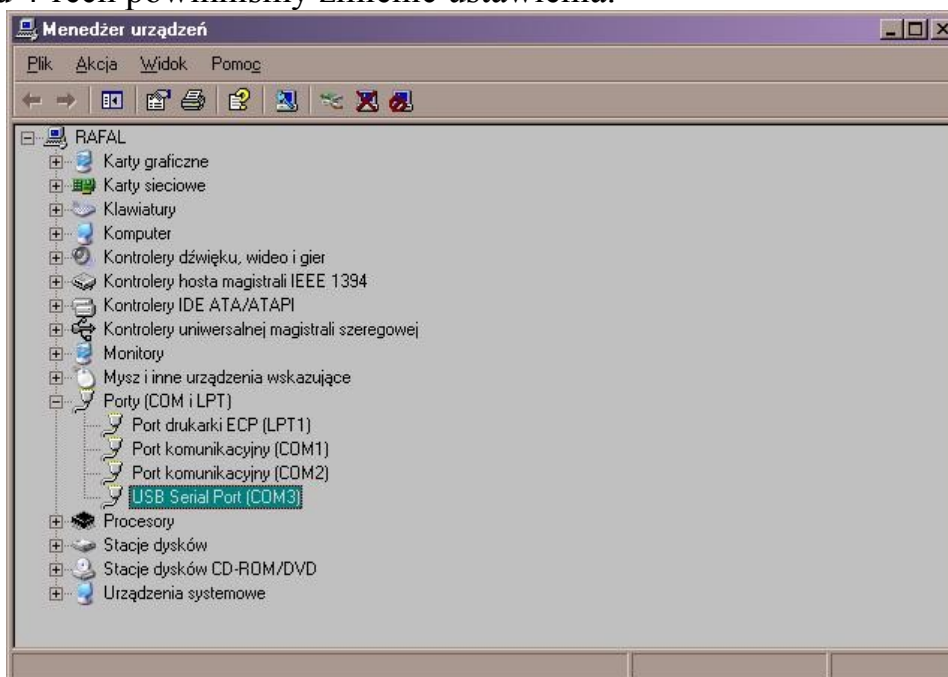
Instrukcje instalacji sterownika FTDI (w wersji angielskojęzycznej) dla innych systemów (MAC, 2000, 98, Linux) dostępne są pod adresem:

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

6. KONFIGURACJA USB SERIAL PORT

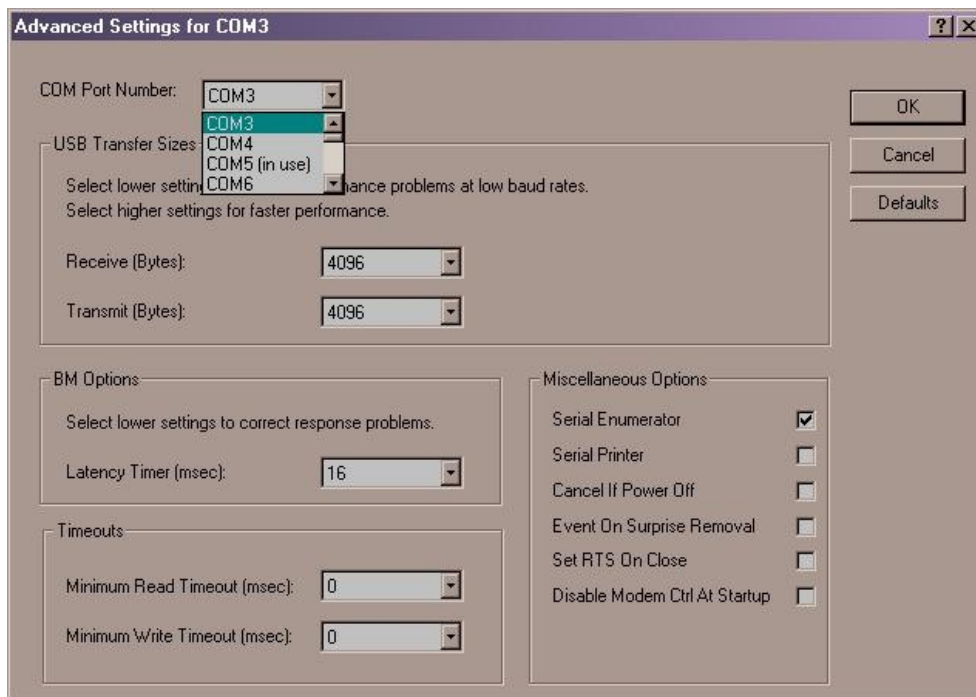
Ponieważ niektóre aplikacje wymagają podłączenia programatora na początkowym COM-ie, dobrym zwyczajem jest *'umieszczenie'* AVR BOX-a na COM1, COM2, COM3 lub COM4 (np. AVR Studio szuka urządzeń tylko na portach 1 do 4).

Aby sprawdzić na którym COM-ie zainstalowany został nasz AVR BOX otwieramy 'Menedżer urządzeń' (rys.25), rozwijamy gałąź 'Porty(COM i LPT)'. Interesuje nas pozycja 'USB Serial Port' z numerem COM-a w nawiasie. Jeśli numer ten jest większy od 4-rech powinniśmy zmienić ustawienia.



Rys.25

Chcąc zmienić numer COM-a otwieramy prawym przyciskiem myszy właściwości USB Serial Port-u. Przechodzimy do zakładki 'Port Settings' i naciskamy 'Advanced'. W polu 'COM Port Number' wybieramy któryś z wolnych portów (rys.26).



rys.26

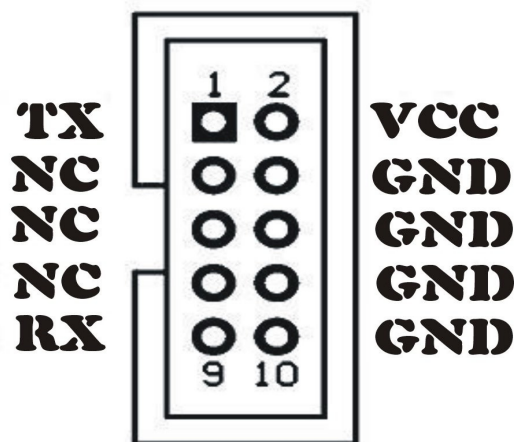
Klikamy ‘OK.’ . Po kliknięciu opcji ‘Skanuj w poszukiwaniu zmian sprzętu’ lub restarcie ‘Menadżera urządzeń’ w nawiasie obok ‘USB Serial Port’ (gałąź ‘Porty(COM i LPT)’) powinien się pojawić ustawiony przez nas numer portu.

8. TRYB INTERFEJSU USB<->RS232

Po wgraniu firmware **rs232** programator może pełnić rolę mini przejściówki USB<->RS232 TTL. Funkcję sygnałów RX i TX przejmują wówczas piny MISO i MOSI.

Pin nr 1 (MOSI) to TX natomiast pin nr 9 (MISO) to RX.

Sygnały te są zgodne z logiką TTL czyli 0-5 V



rys.27