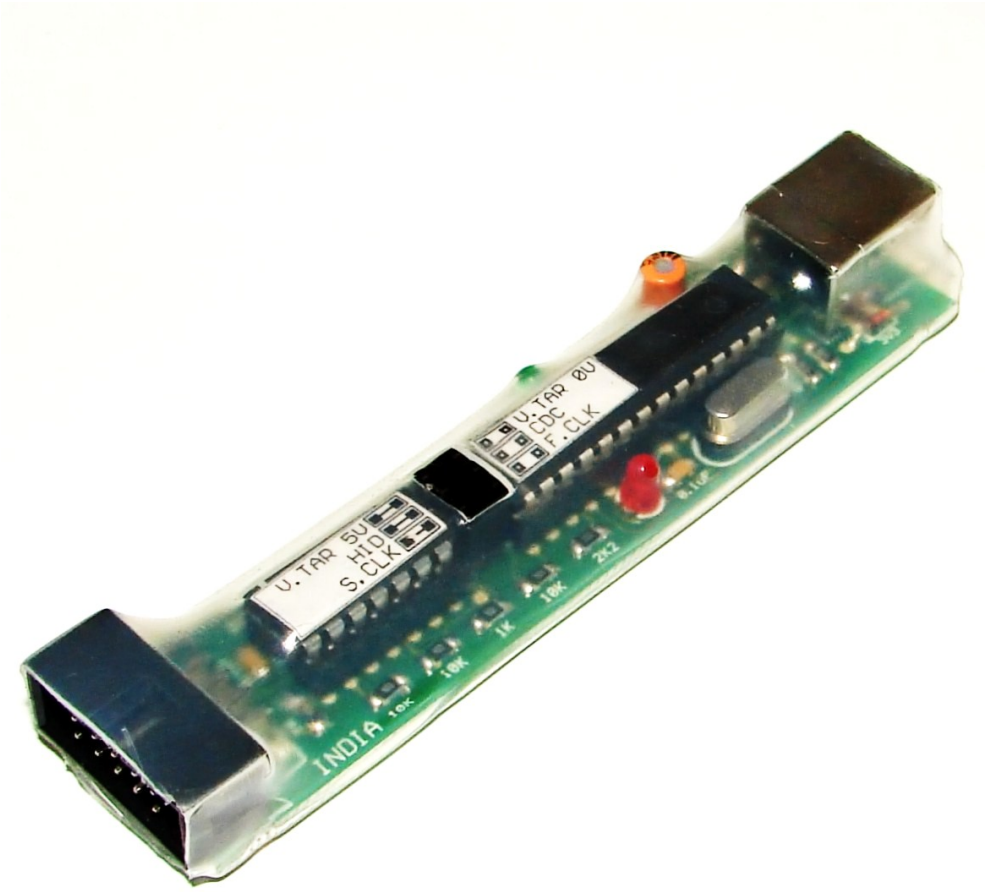
NEX AVR USB ISP STK500V2



1.1 Introduction

NEX AVR USB ISP STK500V2 is a high speed USB powered STK500V2 compatible In-System USB programmer for AVR family of microcontrollers. It can be used with AVR Studio on Win XP platforms. For Windows7 it can be used in HID mode with Avrdude command prompt as programming interface. Its adjustable clock speed allows programming of microcontrollers with lower clock speeds. The programmer is powered directly from a USB port which eliminates need for an external power supply. The programmer can also power the target board from a USB port with limited supply current of up to 100mA.

Note: The USB port of PC provides 5V DC. For 3.3V microcontrollers, please use appropriate voltage regulators.

The compatibility with different window platform is given in below table.

Compatibility Chart

Operating System	AVR Studio (CDC)	Avrdude (HID)	GUI
Windows XP	YES	YES	YES
Windows Vista	X	YES	YES
Windows 7	X	YES	YES

Table 1: - STK500v2 Compatibility Chart

Note: If mode is HID, insert HID/CDC jumper (J2) and if mode is CDC, remove HID/CDC jumper (J2).

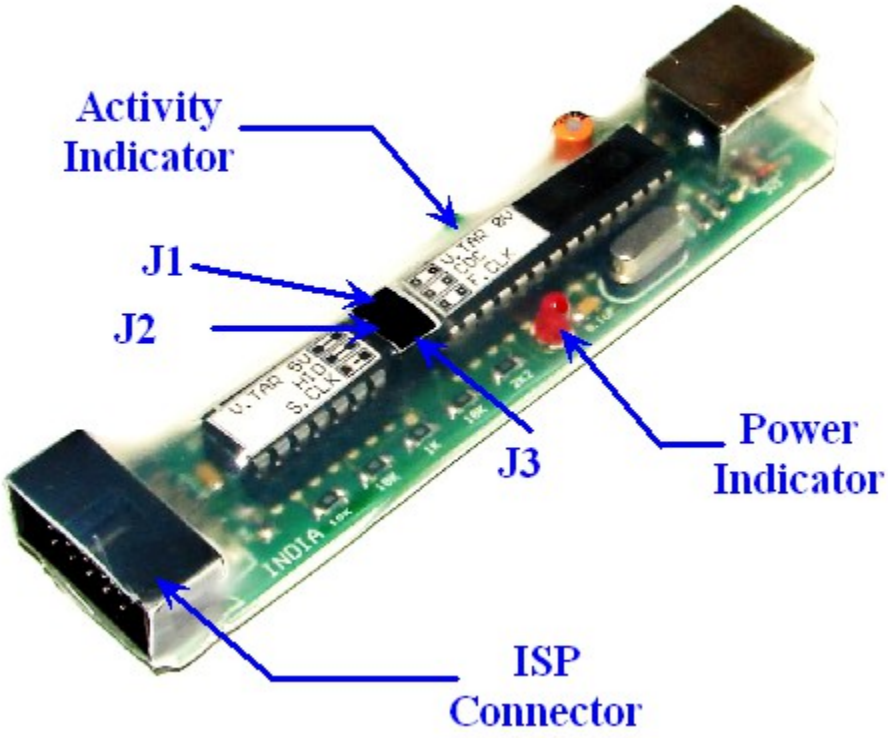
Features

- Low cost USB compatible (No legacy RS232 required)
- Compatible with STK500V2
- Can be used with AVR Studio as STK500 programmer (only WinXP)
- Can be used with AVRdude on Win7/XP/Vista
- Jumper adjustable programming clock speeds for low clock speed microcontrollers. Low speeds from 32 KHz to 1MHz are supported.
- Programs almost all AVR microcontrollers (Refer Table below)
- Jumper selectable HID/CDC mode.
- USB powered
- Jumper selectable 5V power supply for target boards
- Standard 10 pin (5x2) programming connector
- Power and programming activity indicator LEDs
- No external power supply required

Supported AVR Microcontroller

AT90	tinyAVR	megaAVR	megaAVR contd..	xmegaAVR
AT90S1200	ATtiny2313	ATmega8515	ATmega3250P	ATxmega128A1
AT90S2313	ATtiny2313A	ATmega8535	ATmega3290	ATxmega128A3
AT90S2323	ATtiny11	ATmega48(P)	ATmega3290P	ATxmega128D3
AT90S2343	ATtiny12	ATmega8	ATmega32C1	ATxmega16A4
AT90S4414	ATtiny13	ATmega8(A)	ATmega32HVB	ATxmega16D4
AT90S4434	ATtiny13(A)	ATmega88(P)	ATmega32M1	ATxmega192A3
AT90S4433	ATtiny15	ATmega88(PA)	ATmega32U2	ATxmega192D3
AT90S8515	ATtiny167	ATmega8HVA	ATmega32U4	ATxmega256A3
AT90S8535	ATtiny22	ATmega8U2	ATmega32U6	ATxmega256A3B
AT90CAN32	ATtiny24	ATmega16	ATmega103	ATxmega256D3
AT90CAN64	ATtiny24(A)	ATmega16A	ATmega128	ATxmega32A4
AT90CAN128	ATtiny25	ATmega16HVA	ATmega1280	ATxmega32D4
AT90PWM2	ATtiny26	ATmega16HVB	ATmega1281	ATxmega64A1
AT90PWM216	ATtiny261	ATmega16M1	ATmega1284P	ATxmega64A3
AT90PWM2B	ATtiny28	ATmega16U2	ATmega128A	ATxmega64D3
AT90PWM3	ATtiny44	ATmega16U4	ATmega128RFA1	
AT90PWM316	ATtiny45	ATmega161	ATmega2560	
AT90PWM3B	ATtiny48	ATmega163	ATmega2561	
AT90PWM81	ATtiny461	ATmega164(P)	ATmega64	
AT90USB646	ATtiny84	ATmega164(PA)	ATmega644	
AT90USB647	ATtiny85	ATmega162	ATmega644P	
AT90USB1286	ATtiny88	ATmega168(P)	ATmega644PA	
AT90USB1287	ATtiny861	ATmega168(PA)	ATmega645	
AT90USB82	ATtiny861(A)	ATmega165	ATmega649	
		ATmega165P	ATmega640	
		ATmega169	ATmega6450	
		ATmega169P	ATmega6490	
		ATmega1650		
		ATmega1690		
		ATmega32		
		ATmega32A		
		ATmega323		
		ATmega324P		
		ATmega324PA		
		ATmega328P		
		ATmega325		
		ATmega325P		
		ATmega329		
		ATmega329P		
		ATmega3250		

NEX AVR USB ISP STK500V2 Overview



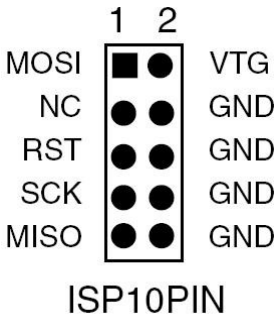
Jumper Description

J1: If inserted, provides 5V at VTG (pin no.2) of ISP connector. If removed 0V at VTG (pin no.2) of ISP connector. **In default mode, this jumper is not inserted.**

J2: If inserted, enables UBS HID mode. If removed enables USB CDC mode. **In default mode, this jumper is not inserted.**

J3: If inserted, enables slow clock speed (for 32 KHz to 1MHz speed microcontrollers). If removed enables normal clock speed. **In default mode, this jumper is not inserted.**

ISP Connector Pin Details

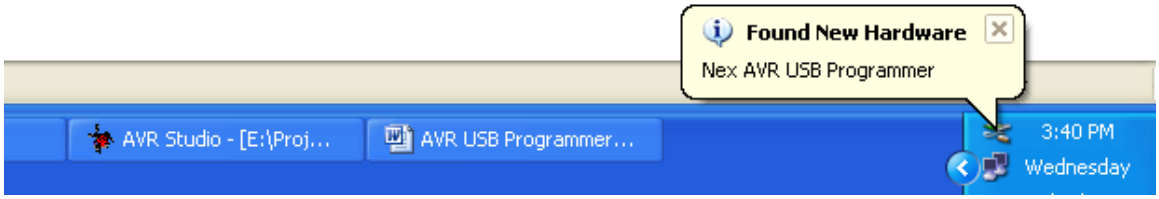


1.1.1 Driver Installation

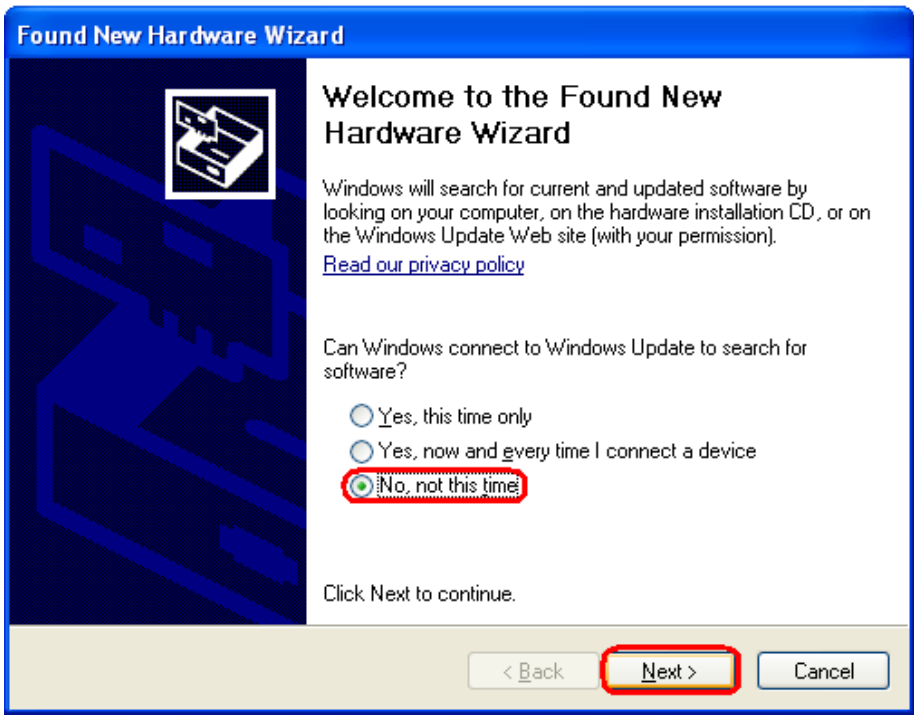
Case 1: Installing drivers for STK500 CDC Mode

Download software package from Nex Robotics website and unzip the contents on your local drive. This zip file contains documentation, driver files and Avrdude software.

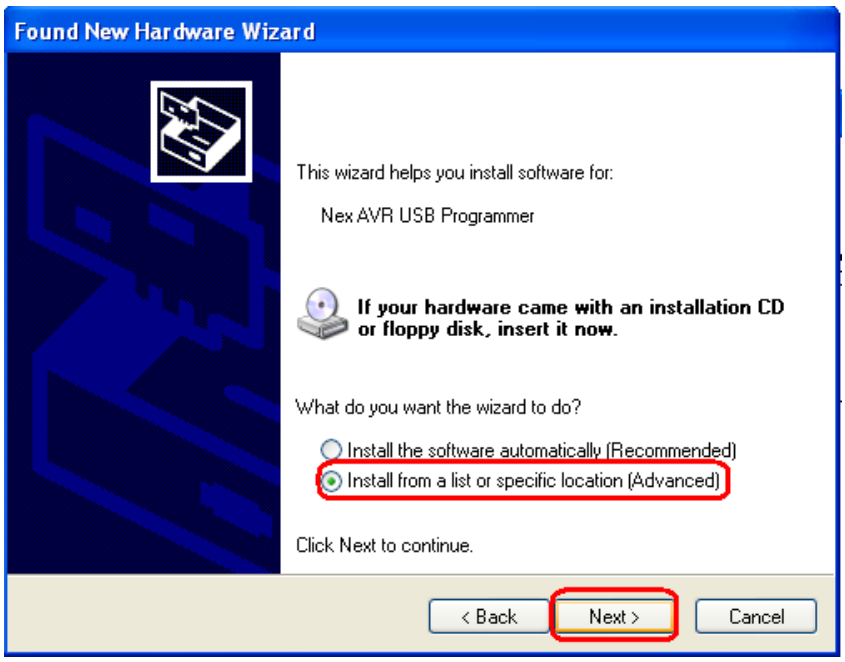
1. If connected, disconnect programmer from PC and remove HID/CDC jumper (J2). Now reconnect programmer to PC and observe the task bar for "Found New Hardware" message.



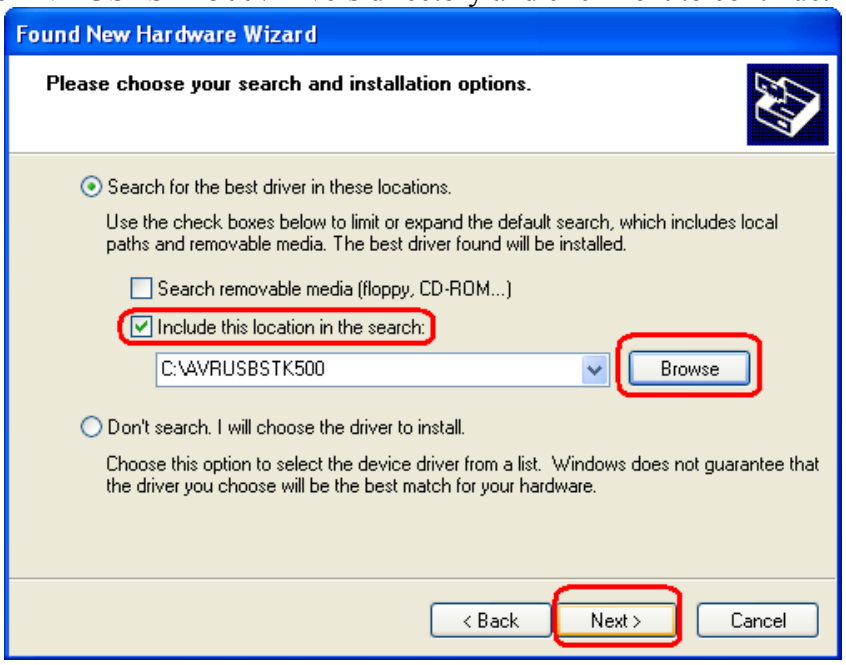
2. After identifying the hardware, the windows driver installation wizard should start. Select "**No, not this time**" and click next to continue.



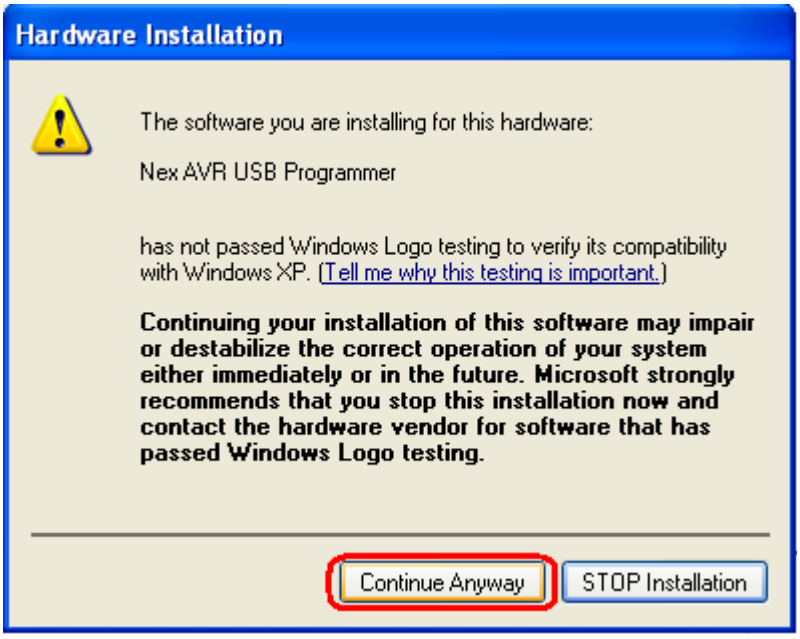
3. Select "**Install from a list of specific location**" and click next to continue.



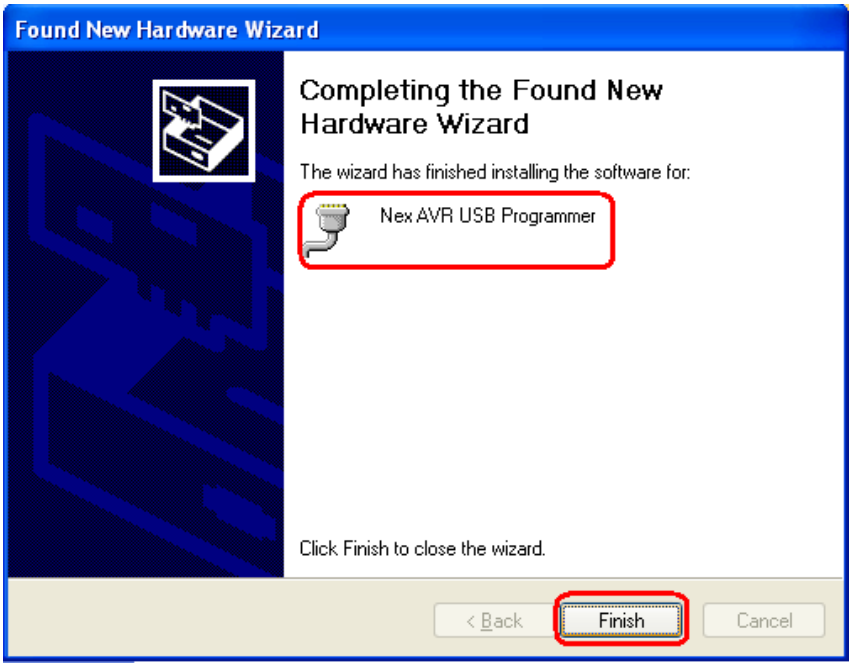
4. Browse to AVRUSBSTK500\Drivers directory and click next to continue.



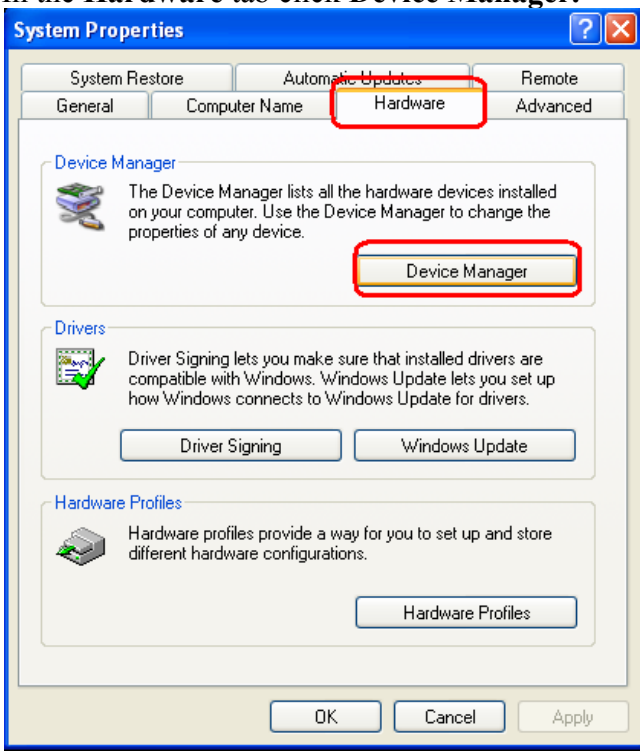
5. In the next window click **Continue Anyway** to proceed.



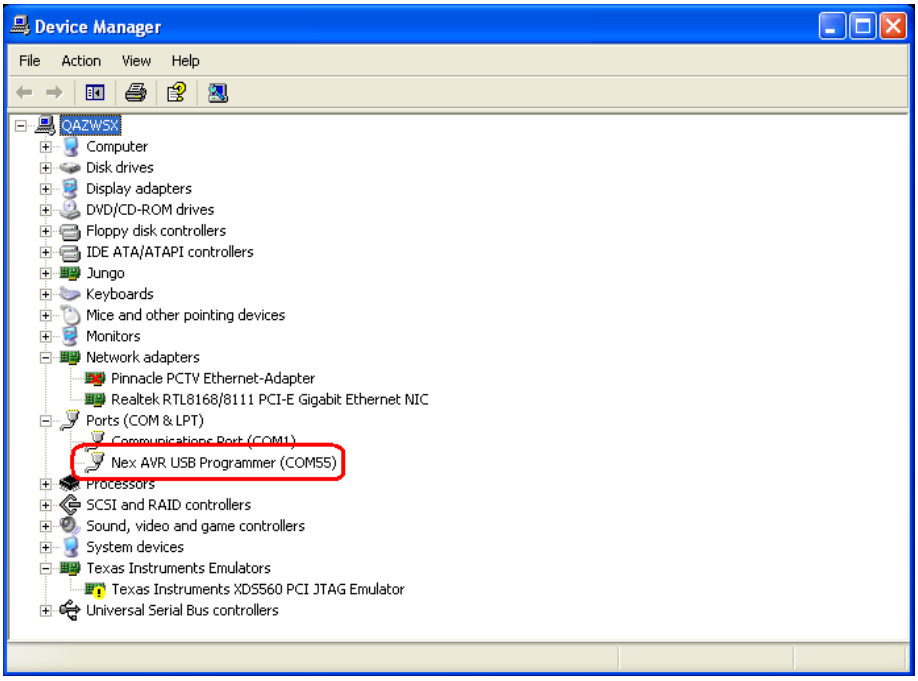
6. After successful installation of drivers following window will appear. Click finish to complete the installation.



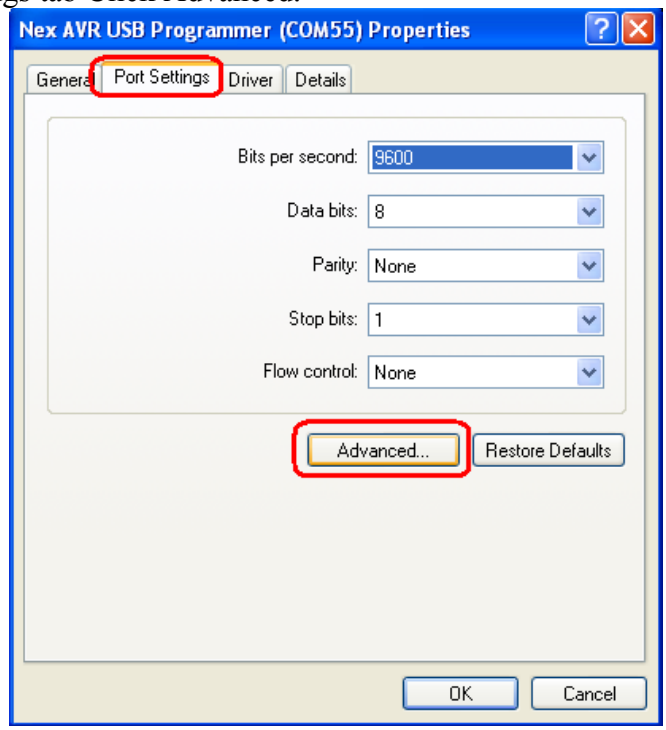
7. To identify the COM port assigned to NEX AVR USB programmer, go to System properties window. In the **Hardware** tab click **Device Manager.**



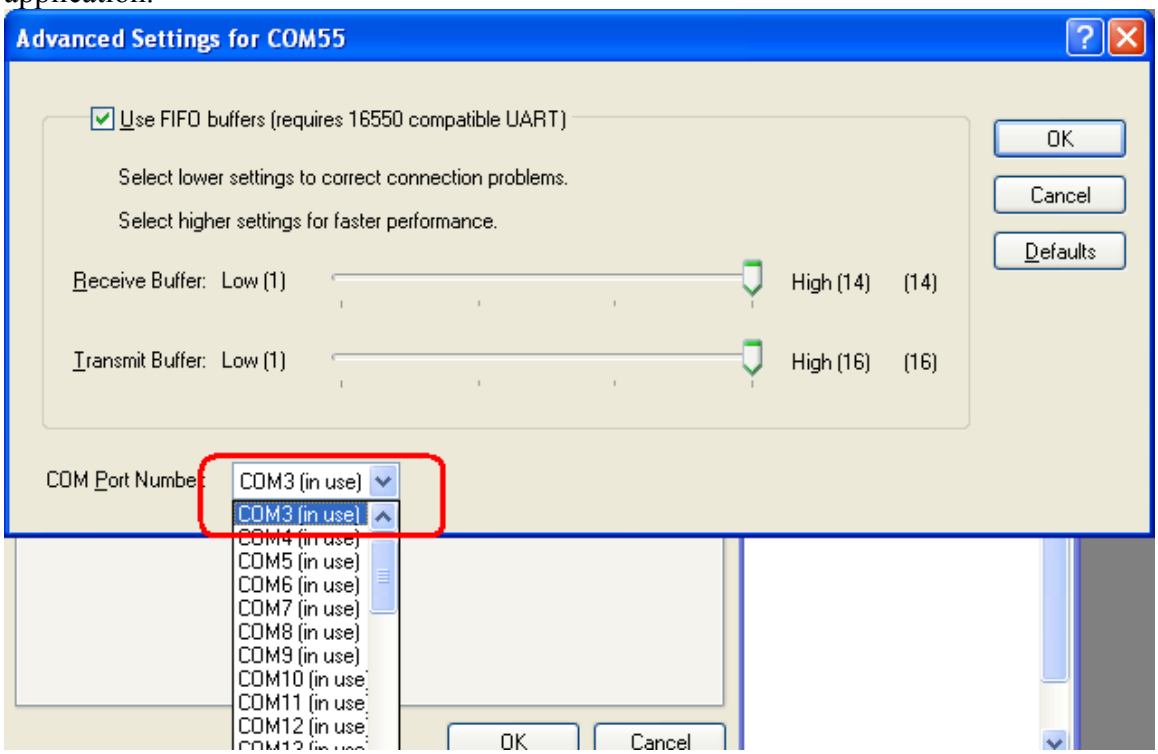
8. In the Device Manager, expand Ports tree and observe that NEX AVR USB Programmer is installed at COM55. The COM port should be less than 4 for it to be recognized by AVR Studio. To change COM port, double click on NEX AVR USB Programmer.



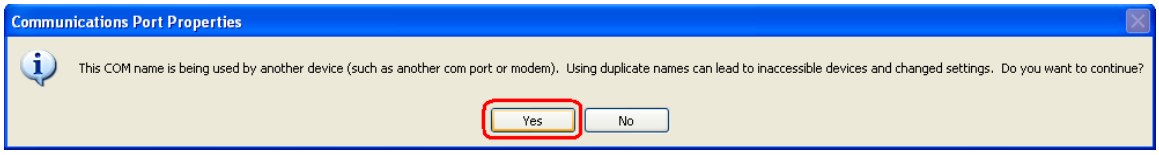
9. In the port settings tab Click Advanced.



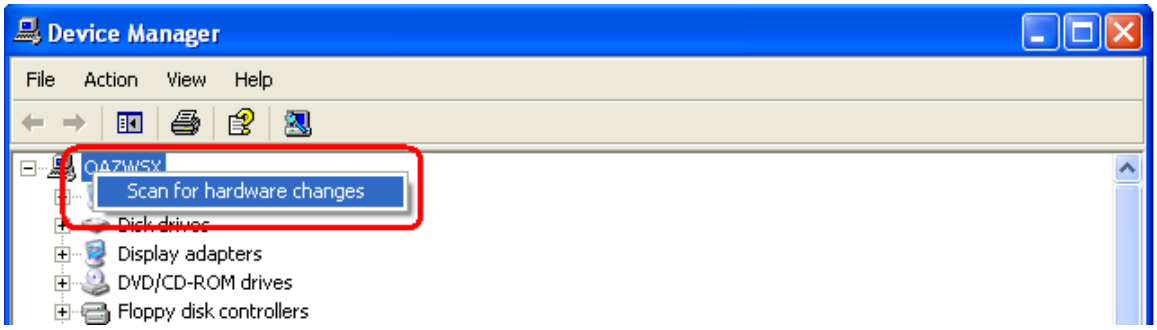
10. In the Advanced Settings window select the appropriate COM port (less than or equal to 4) and click OK. Ensure that the selected COM port is not in use by any other application.



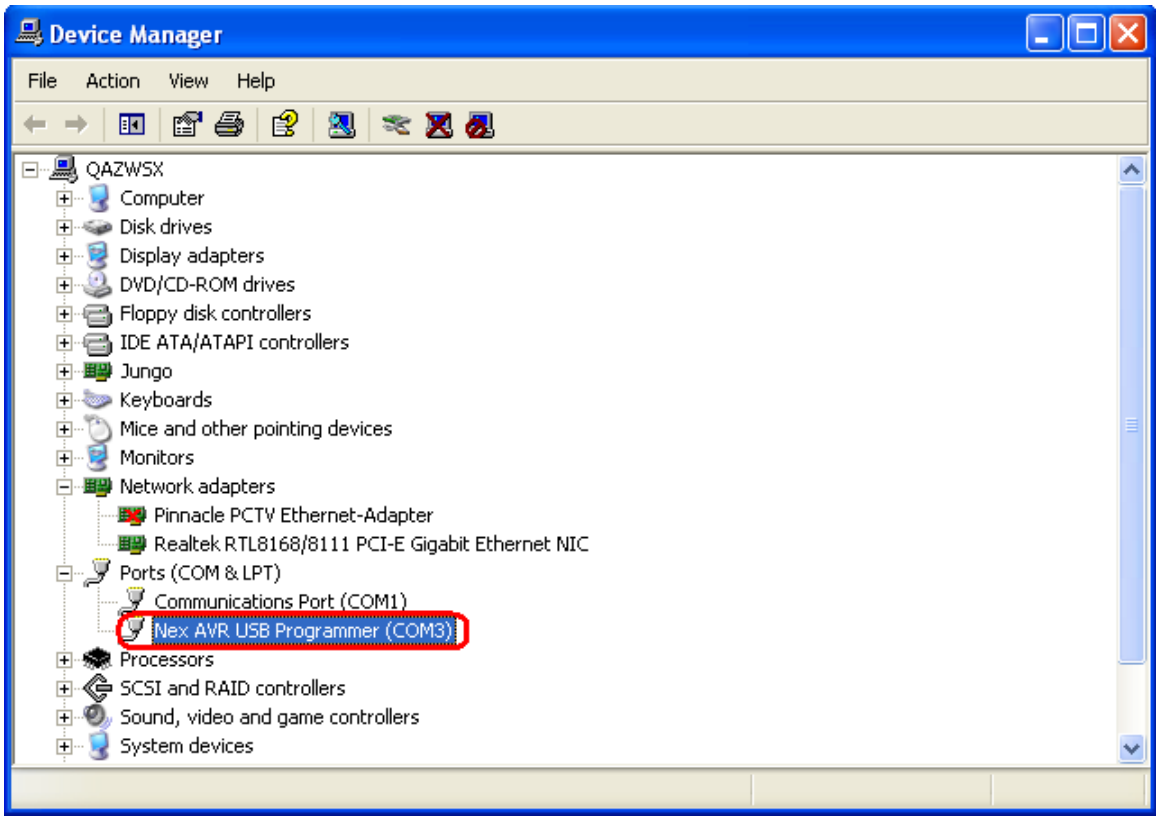
11. In the next window, click Next to continue.



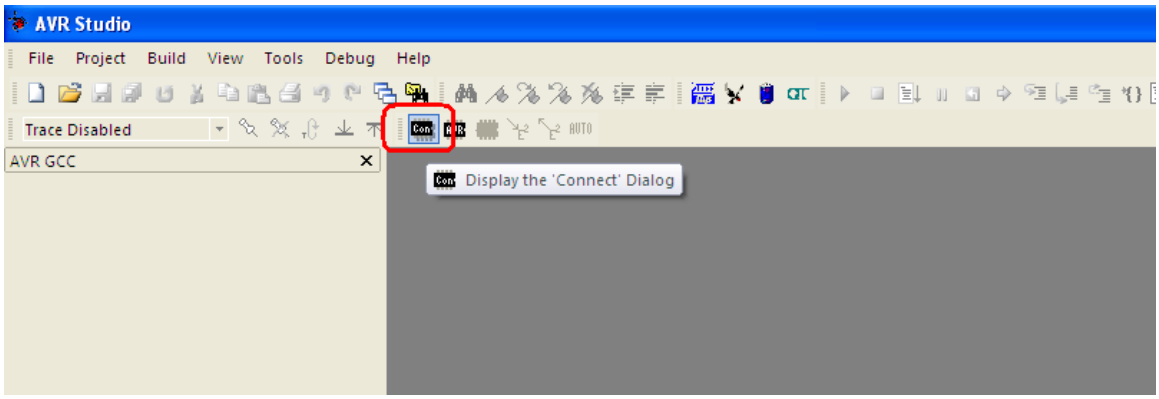
12. Right click anywhere in the device manager and click on “Scan for hardware changes”.



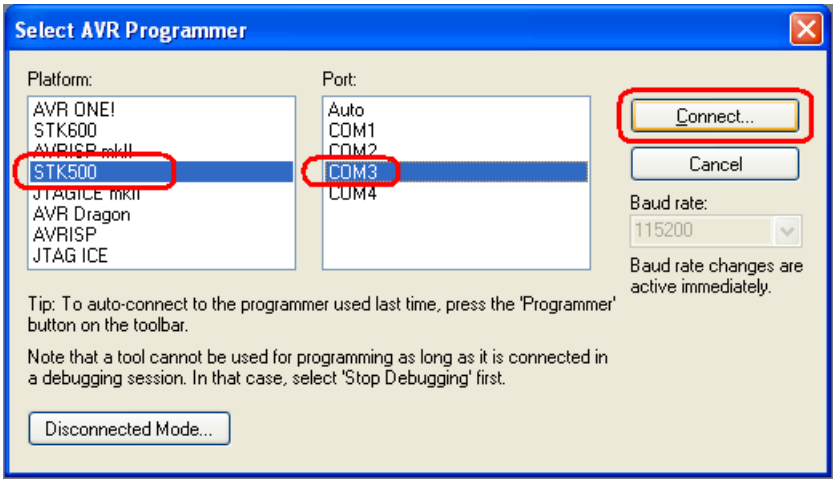
13. After scanning is complete, observe that new COM port is assigned to the programmer.



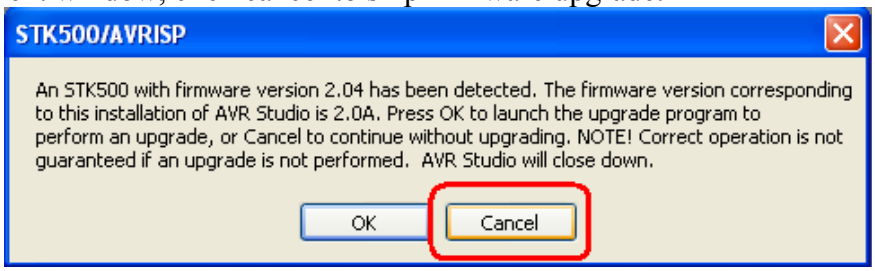
14. Open AVR Studio from Start Menu. Click **Con** button on the tool bar to open Connect Dialog.



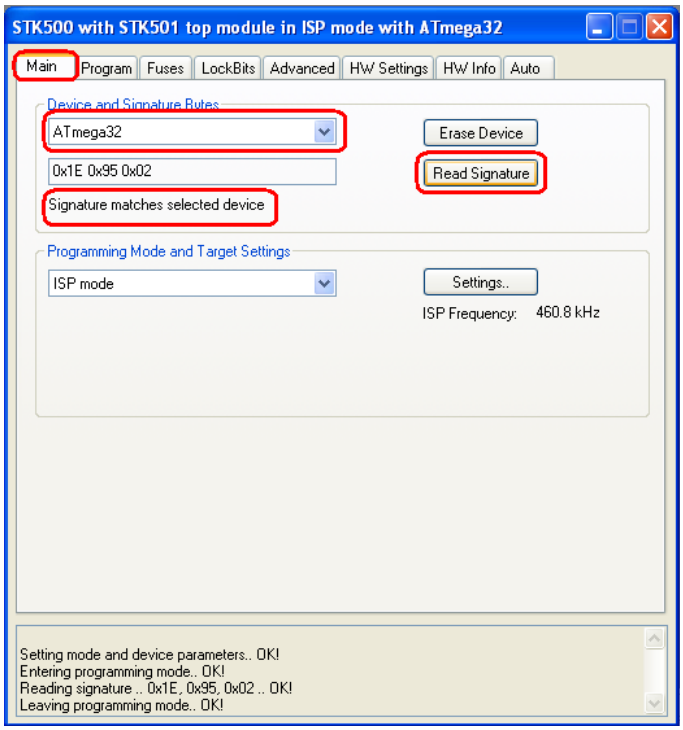
15. In the connect dialog, select platform as STK500, select appropriate port and click connect. If Connect Dialog reappears, then recheck that the COM port is available and try again.



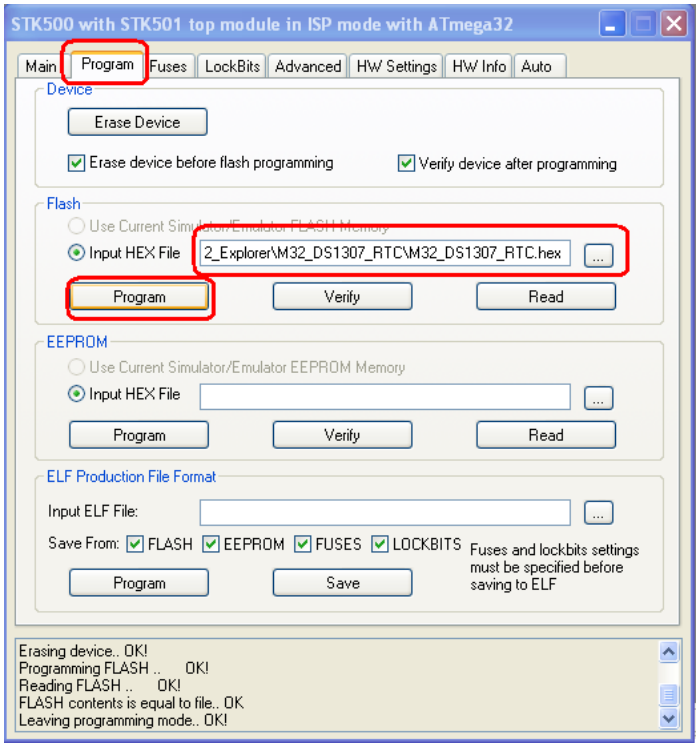
16. In the next window, click cancel to skip firmware upgrade.



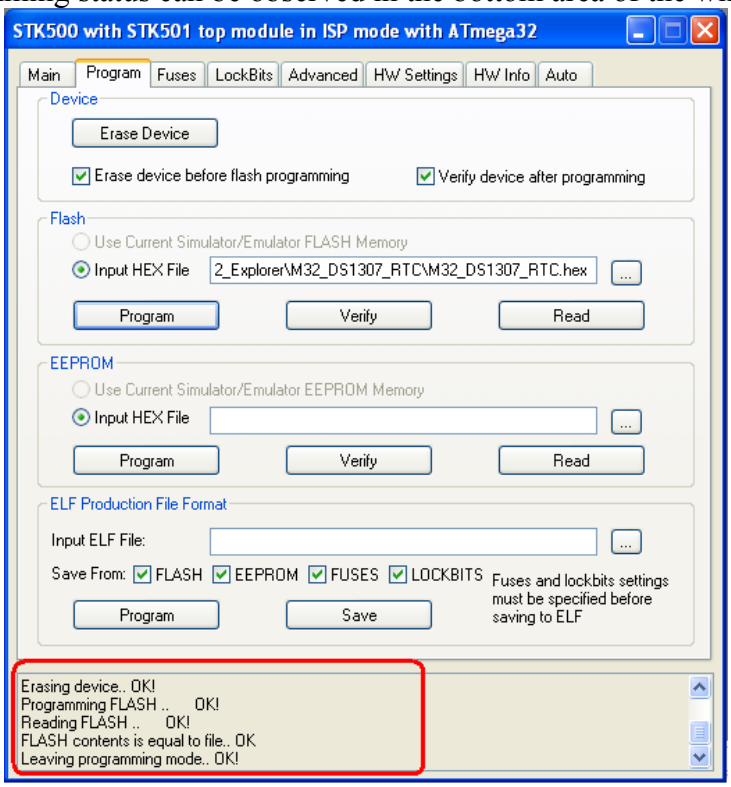
17. After clicking cancel, AVR Studio will open STK500 interface. In the main tab, select the appropriate microcontroller and read its signature. Observe that the signature matches the selected device.



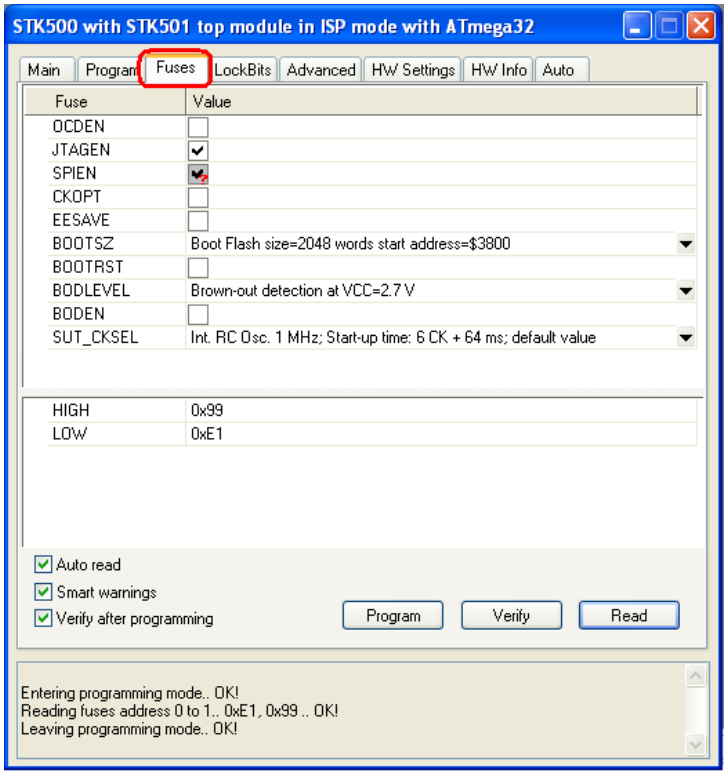
18. In the Program tab, select the appropriate hex file and click program to load the hex file in the microcontroller.



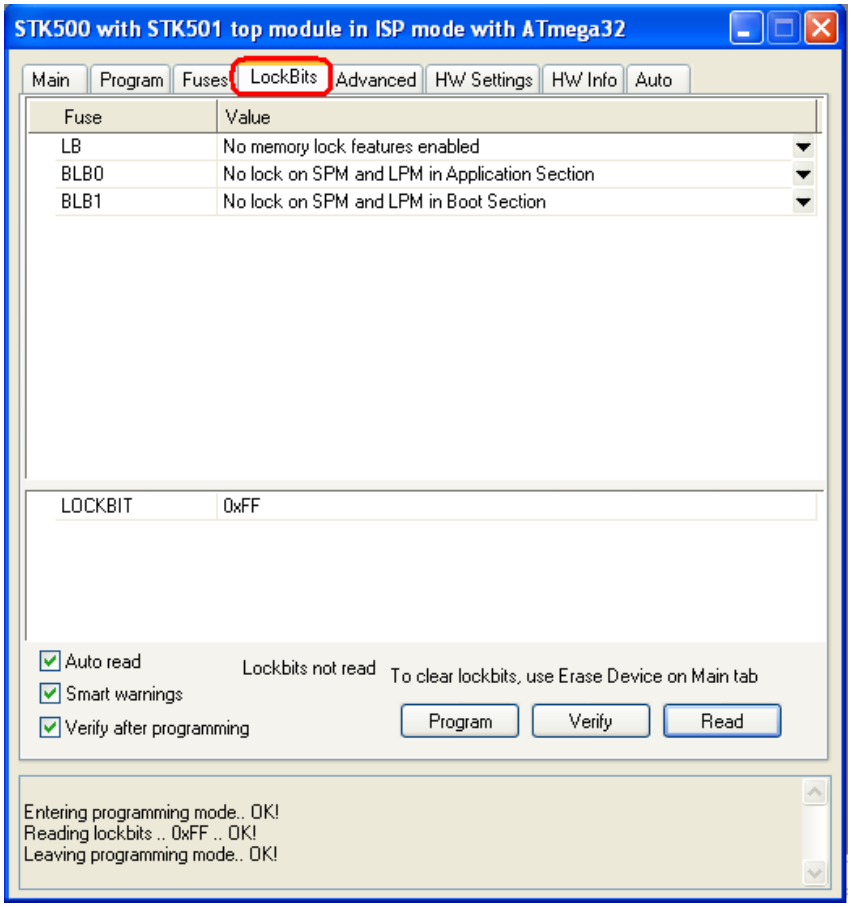
19. The programming status can be observed in the bottom area of the window.



20. The Fuses tab can be used to set the fuse bits of the microcontroller. For appropriate fuse bit values, refer microcontroller datasheet.



21. The Lock Bits tab can be used to set the lock bits of the microcontroller. For appropriate lock bit values, refer microcontroller datasheet.

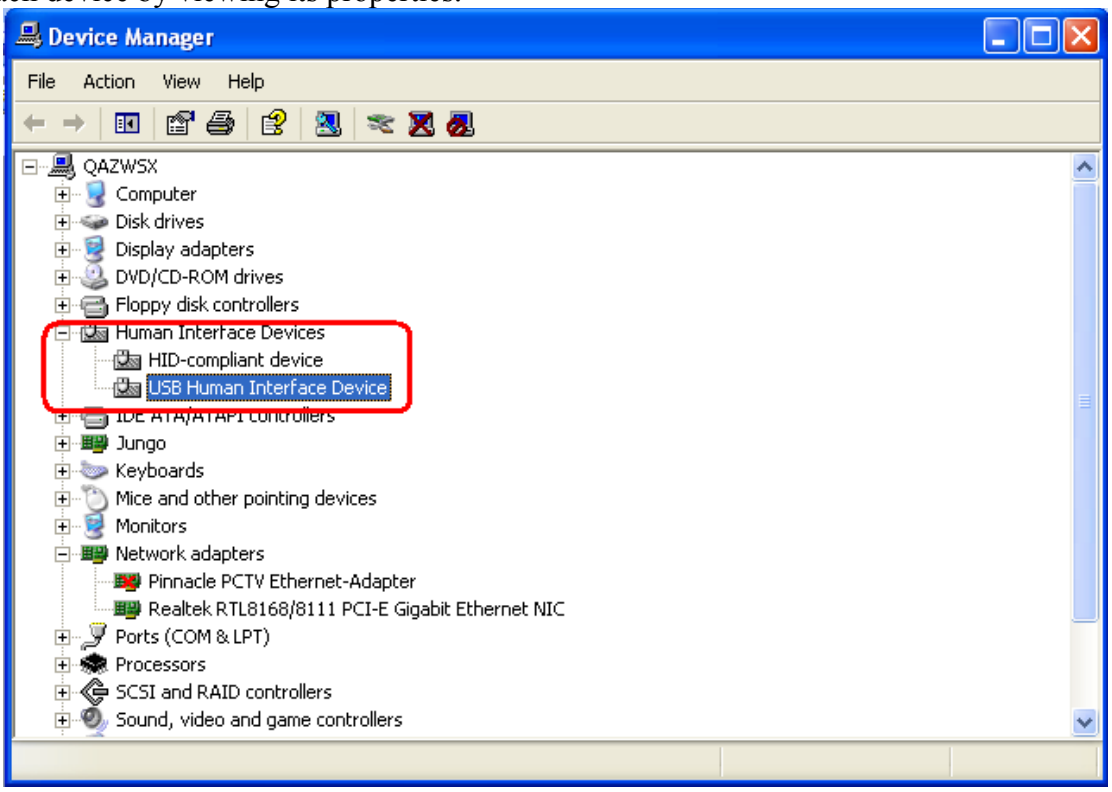


Case 2: Installing drivers for HID mode (Works on all windows operating systems**)**

1. If connected, disconnect programmer from PC and insert HID/CDC jumper. Now reconnect programmer to PC and observe the task bar for "Found New Hardware" message.

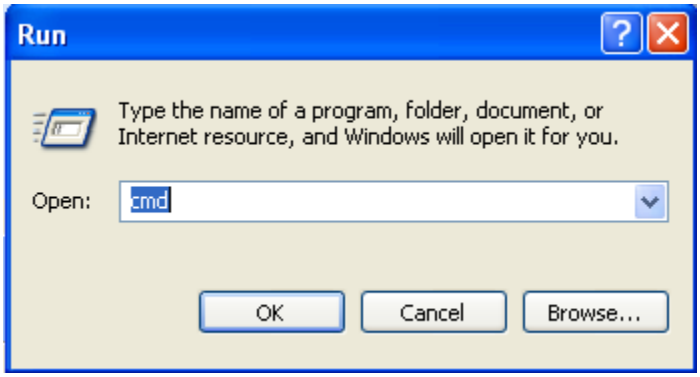
2. HID mode does not require additional drivers. It uses generic windows drivers.

3. Go to Device Manager and observe that new Human Interface Device (HID) is installed. If there are other HID devices connected to PC, you may optionally identify each device by viewing its properties.

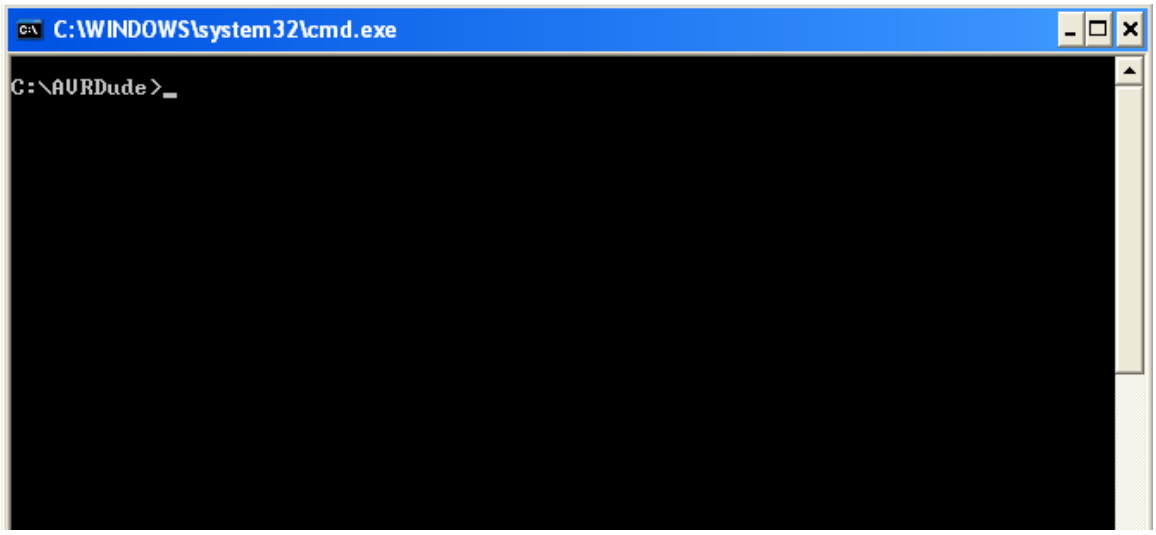


4. Before proceeding ensure that you have AVRDude.exe and AVRDude.conf on your PC. These files are available in the zip file that was downloaded earlier from Nex Robotics website.

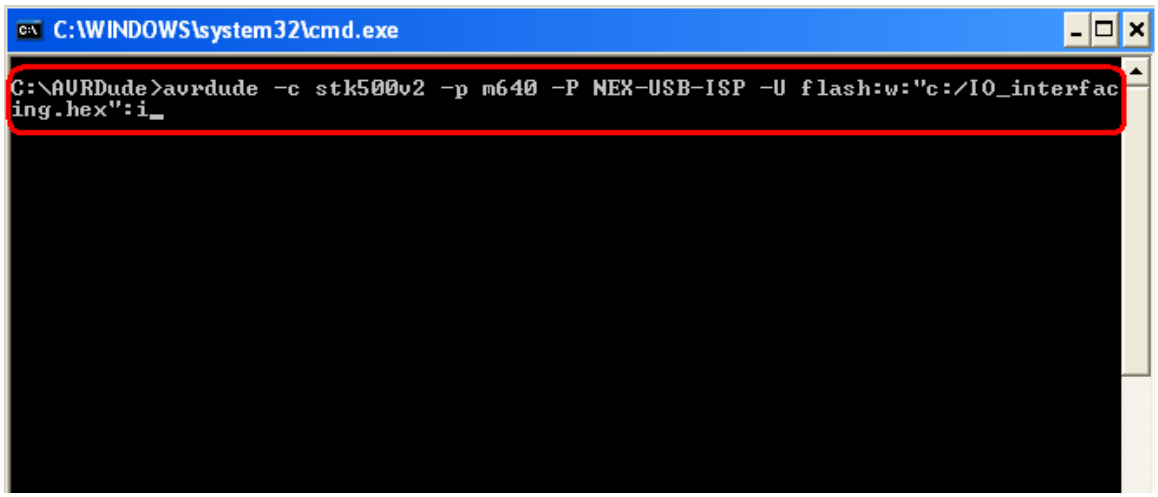
5. Go to Start Menu>Run and type “cmd” to open command prompt.



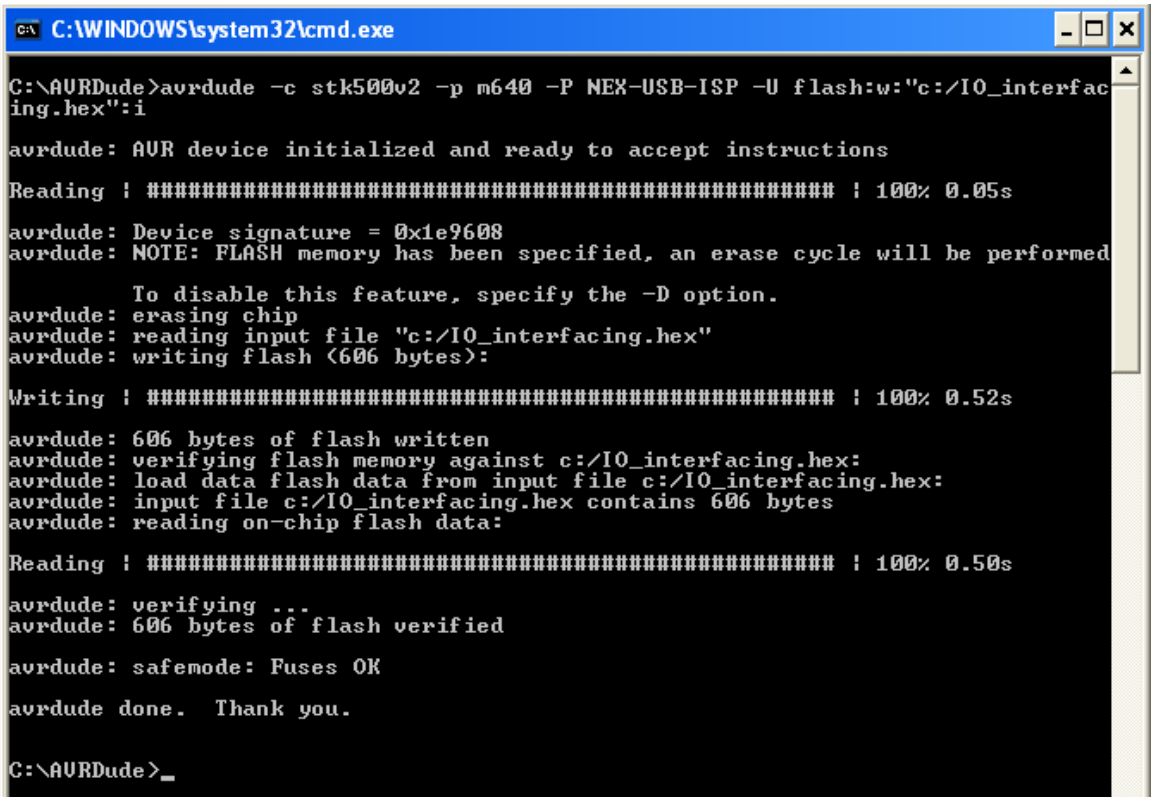
6. On the command prompt, type the path of the folder that contains avrdude.exe and avrdude.conf files. For e.g. refer fig. below.



7. On the command line type the command as shown in the fig. below. Here ***-p m640*** refers to the microcontroller part number. The last section after –U in quotes specifies the location of hex file. In the command line edit the part number and hex file location as required and connect the programmer to the target board using 10 pin FRC cable provided with the programmer and turn ON the target board.



8. Press enter. You should see the programming status in the command prompt window. If there is any error, recheck ISP connection and command line parameters.



Example Commands:

Eg 1. Transfer a file called example1.hex to a Mega128 device.

```
avrdude -c stk500v2 -p m128 -P NEX-USB-ISP -U flash:w:example1.hex
```

Eg 2. Transfer a file called "example1.hex" present on Desktop to a Mega128 device and change hfuse to 0xfe and lfuse to 0xdc.

```
avrdude -c stk500v2 -p m128 -P NEX-USB-ISP -U flash:w:example1.hex -U hfuse:w:0xfe:m lfuse:w:0xdc:m
```

Eg 3. Modify the contents of hfuse, lfuse and efuse of Atmega640 microcontroller

```
avrdude -c stk500v2 -p m640 -P NEX-USB-ISP -U efuse:w:0xf7:m -U hfuse:w:0xd7:m -U lfuse:w:0xff:m
```

Eg 4. View Avrdude's version number and other details.

```
avrdude -v
```

Eg 5. Read the contents of the FLASH memory and store them in a file called test1.hex

```
avrdude -c stk500v2 -p m128 -P NEX-USB-ISP -U flash:r:"c:\test1.hex":i
```

Eg 6. Read the contents of the EEPROM memory and store them in a file called test1.eep

```
avrdude -c stk500v2 -p m128 -P NEX-USB-ISP -U eeprom:r:"c:\test1.eep":i
```

Eg 7. Read the contents of HFUSE and LFUSE and store them in files hfuse.hex and lfuse.hex

```
avrdude -c stk500v2 -p m128 -P NEX-USB-ISP -U hfuse:r:"c:\hfuse.hex":i -U lfuse:r:"c:\lfuse.hex":i
```

Eg 8. Read the contents of HFUSE and LFUSE and EFUSE and store them in files hfuse.txt and lfuse.txt and efuse.txt

```
avrdude -c stk500v2 -p m640 -P NEX-USB-ISP -U efuse:r:"c:\efuse.txt":h -U hfuse:r:"c:\hfuse.txt":h -U lfuse:r:"c:\lfuse.txt":h
```

Eg 9. Read device signature bytes and store them in a file called sig.hex

```
avrdude -c stk500v2 -p m128 -P NEX-USB-ISP -U signature:r:"c:\signature.hex":i
```

Note: - The signature bytes are fixed for a specific AVR device

Eg 10. Read device lock bytes and store them in a file called lock.hex

```
avrdude -c stk500v2 -p m128 -P NEX-USB-ISP -U lock:r:"c:\lock.hex":i
```

Eg 11.Read device RC oscillator calibration bytes and store them in a file called calibration.hex

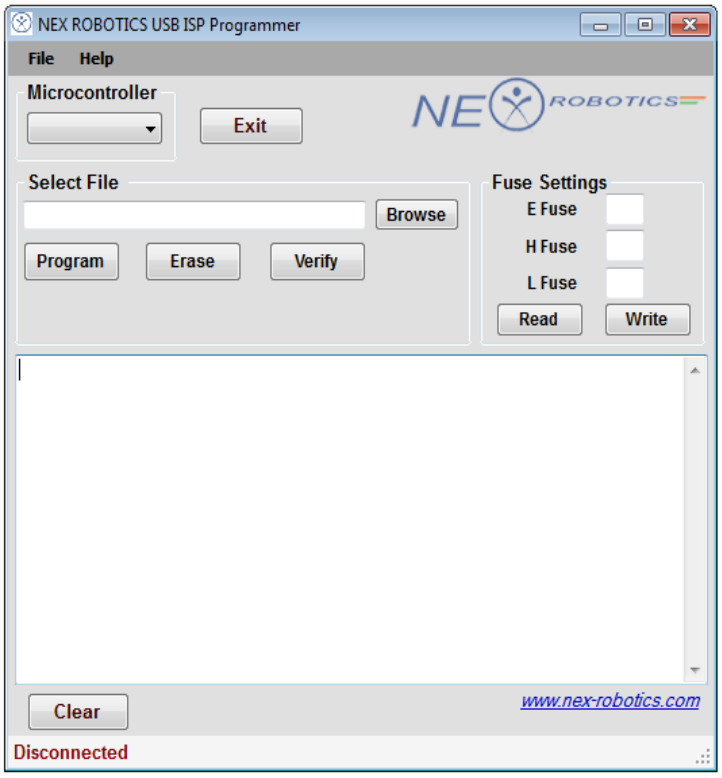
```
avrdude -c stk500v2 -p m128 -P NEX-USB-ISP -U calibration:r:"c:\calibration.hex":i
```

Eg 12. Erase the chip

```
avrdude -c stk500v2 -p m128 -P NEX-USB-ISP -e
```

1.2 STK500v2 GUI

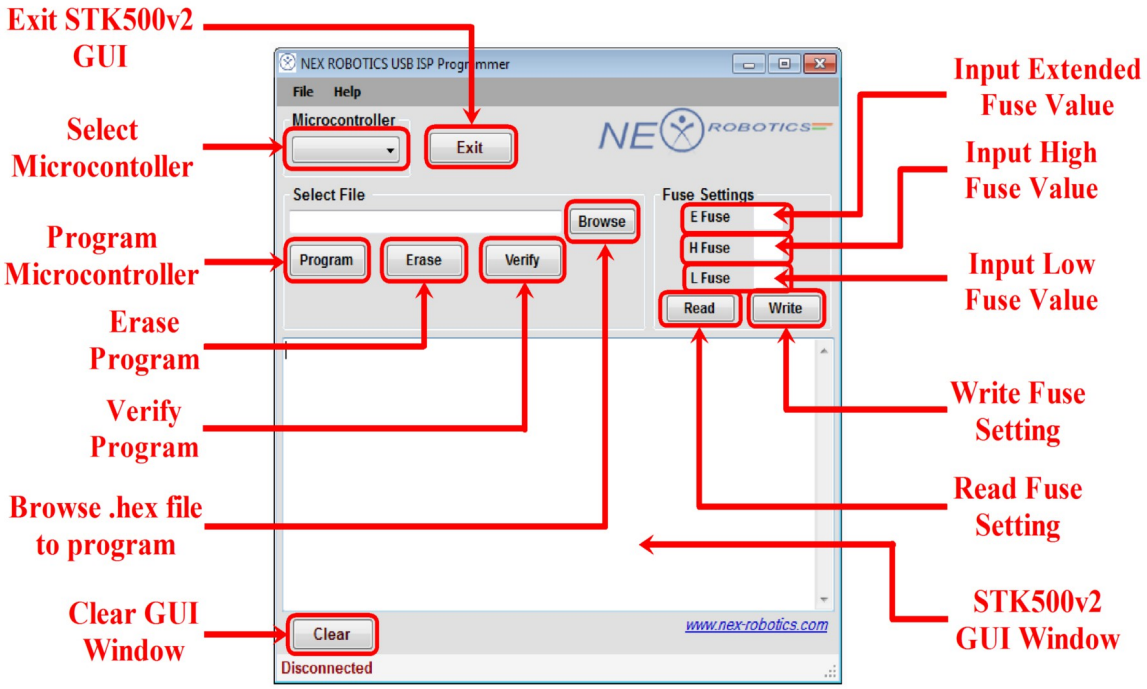
STK500V2 is a high speed USB powered STK500V2 compatible In-System USB programmer for AVR family of microcontrollers. STK500v2 can be used with STK500v2 GUI, provided by NEX-Robotics, to program files on microcontrollers. STK500v2 GUI can be downloaded from STK500v2 product page available on NEX-Robotics website. STK500v2 has to be configured in HID mode to work with STK500v2 GUI.
The below figure shows the STK500v2 GUI.



The compatibility with different window platform is given in Table 1.

Note: - To use STK500v2 GUI, use STK500v2 in HID mode.

1.2.1 STK500v2 GUI Description

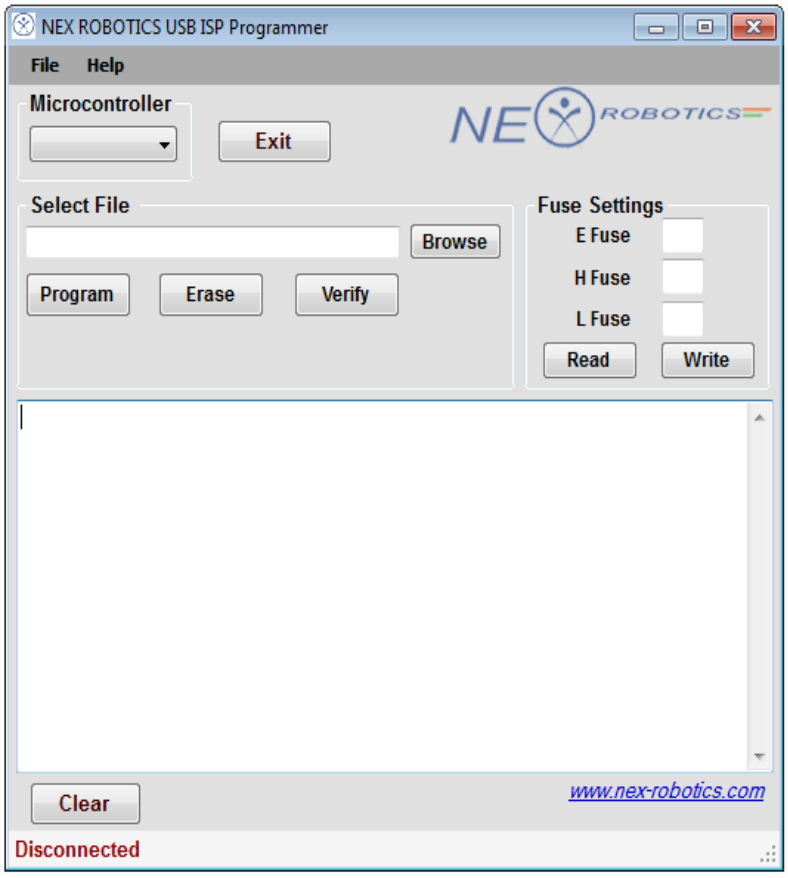


Microcontroller: - Select micro controller from the list of microcontrollers present in the GUI to write file on them.
Exit: - Exit STK500v2 GUI.
Browse: - Browse the path of the file that you want to write on the microcontroller.
Program: - Program/Write selected file on microcontroller.
Erase: - Erase the file that is currently written on the microcontroller.
Verify: - Verify the currently loaded file on the microcontroller.
Clear: - Clear STK500v2 GUI window.
E Fuse: - Input proper extended fuse value from Table 2 or Table 3 to write the microcontrollers fuse setting.
H Fuse: - Input proper High fuse value from Table 2 or Table 3 to write the microcontrollers fuse setting.
L Fuse: - Input proper Low fuse value from Table 2 or Table 3 to write the microcontrollers fuse setting.
Read: - Read the microcontrollers current fuse setting.
Write: - Write microcontrollers fuse setting.

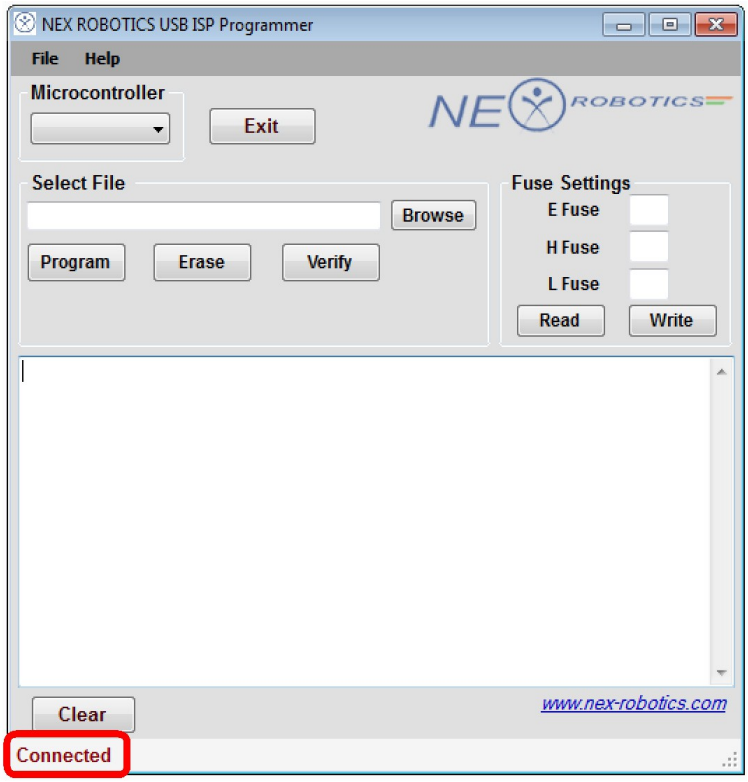
1.2.2 Programming Microcontroller via STK500v2 GUI

STK500v2 programmer can be used to program different microcontrollers like ATmega2560, ATmega640, ATmega128, ATmega32, ATmega16 and ATmega8. STK500v2 is provided with GUI to load/write hex file different microcontrollers with ease. STK500v2 has to be connected in HID mode to work with STK500v2 GUI.
Follow the following procedure to load HEX file on AVR microcontrollers via STK500v2 GUI.

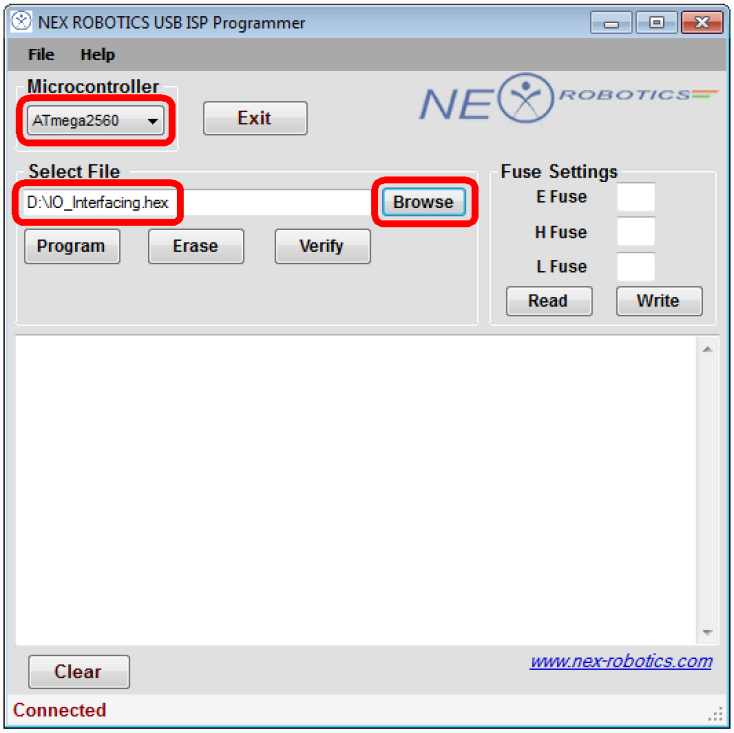
1. Open Stk500v2 GUI. The following window shown below will pop up.



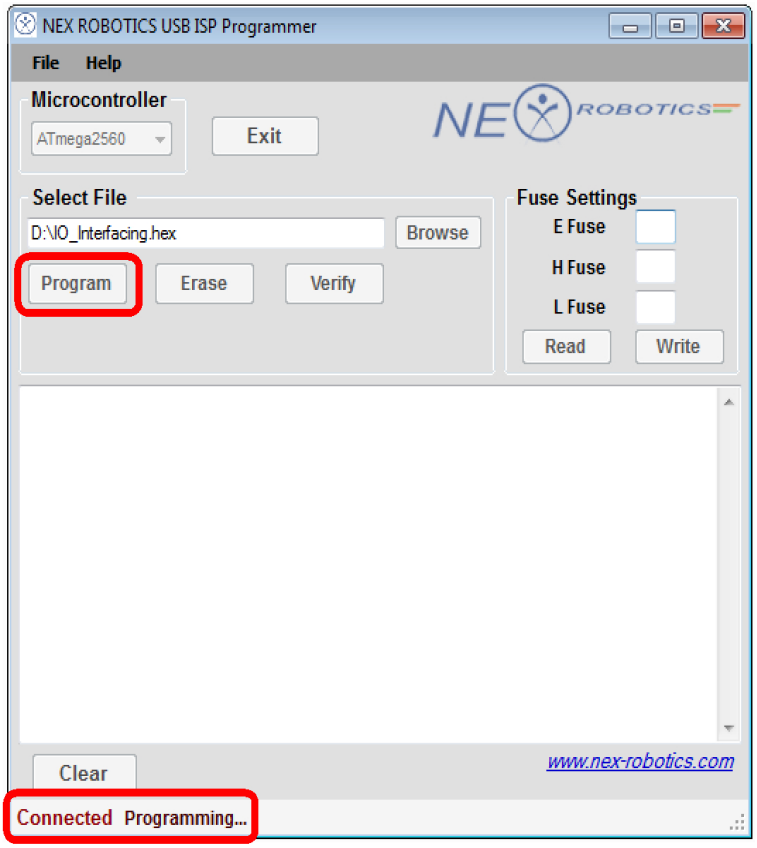
2. Connect STK500v2. In the bottom left of the GUI we can observe that the STK500v2 connection status has changed from “Disconnected” to “Connected” as highlighted in the below figure.



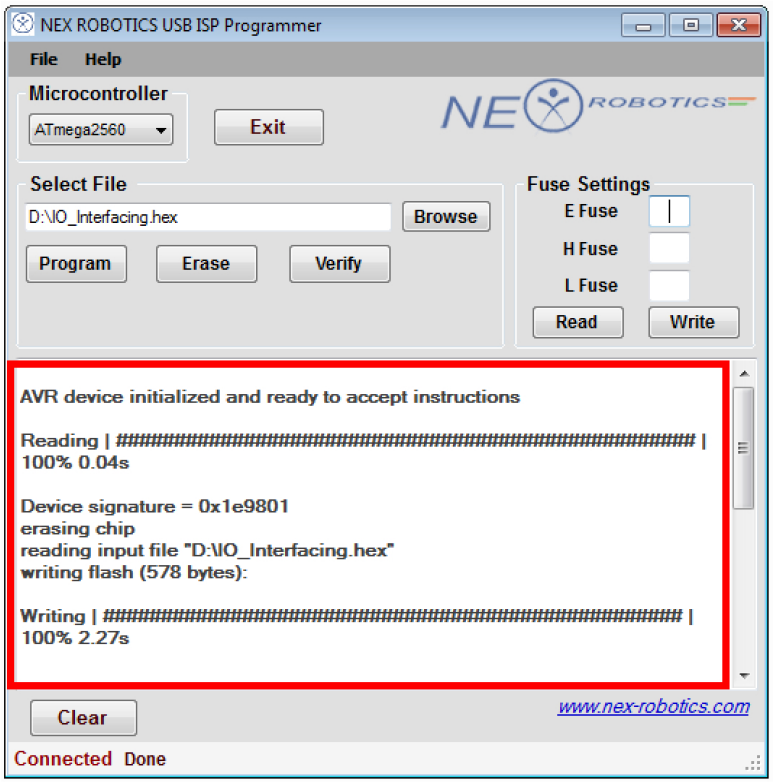
3. Select proper microcontroller and browse to the file that needs to be loaded on the microcontroller as shown in the figure below. In the figure given below we have selected Atmega2560 as microcontroller and IO_Interfacing.hex file to load on Atmega2560 microcontroller.

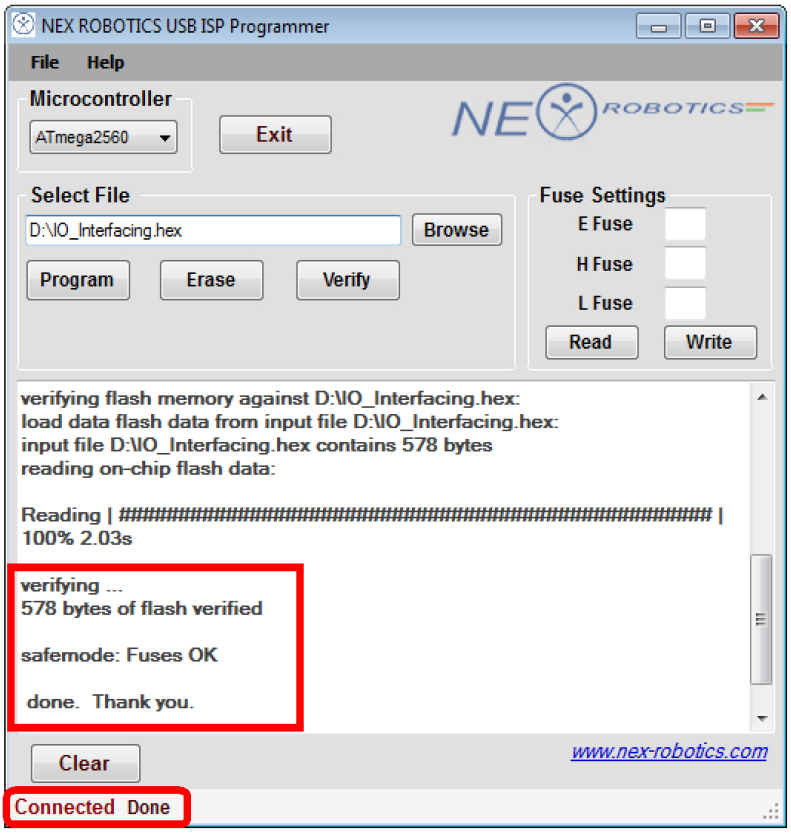


4. Click on the “Program” button to write file on microcontroller. As the programmer starts programming, on the bottom left corner of the GUI, a “Programming” text will start flashing. This flashing text indicates that the programmer is programming the microcontroller.

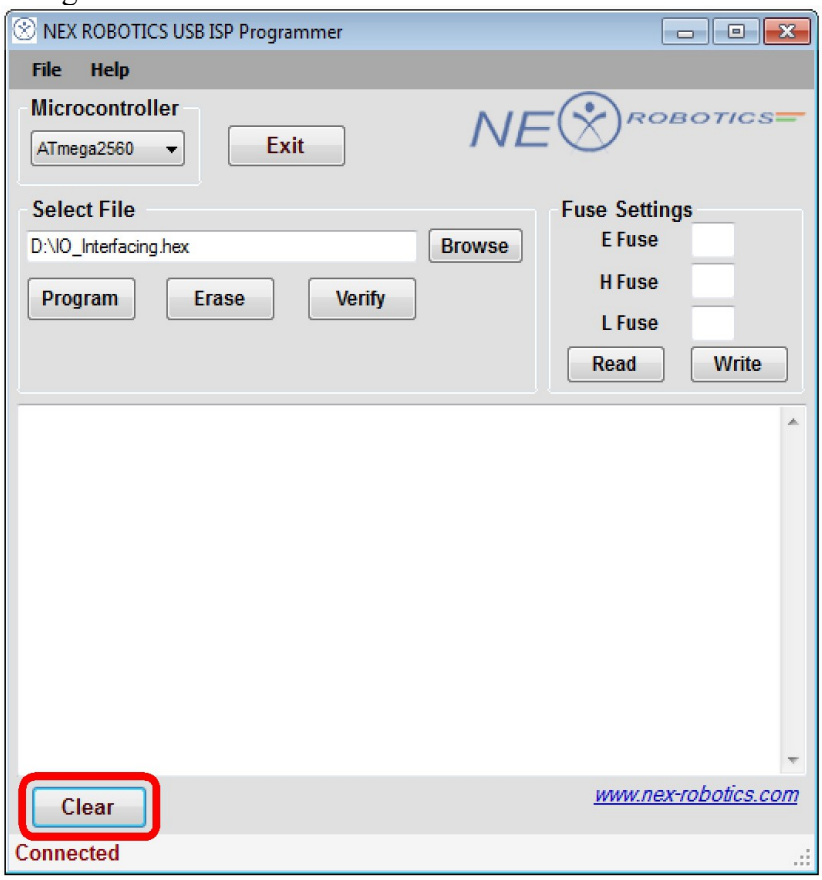


5. Programming is completed when a “Done” text appears at the bottom left corner of the GUI. Also after programming is completed we can check the device information on the GUI window like Device Signature, number of bytes written on flash memory, Fuse setting Status, etc as shown in the figures below.


NEX ROBOTICS USB ISP Programmer
File Help
Microcontroller
ATmega2560
Exit
Select File
D:\IO_Interfacing.hex
Browse
Program
Erase
Verify
Fuse Settings
E Fuse
H Fuse
L Fuse
Read
Write
AVR device initialized and ready to accept instructions
Reading | ## |
100% 0.04s
Device signature = 0x1e9801
erasing chip
reading input file "D:\IO_Interfacing.hex"
writing flash (578 bytes):
Writing | ## |
100% 2.27s
Clear
www.nex-robotics.com
Connected Done


NEX ROBOTICS USB ISP Programmer
File Help
Microcontroller
ATmega2560
Exit
Select File
D:\IO_Interfacing.hex
Browse
Program
Erase
Verify
Fuse Settings
E Fuse
H Fuse
L Fuse
Read
Write
verifying flash memory against D:\IO_Interfacing.hex:
load data flash data from input file D:\IO_Interfacing.hex:
input file D:\IO_Interfacing.hex contains 578 bytes
reading on-chip flash data:
Reading | ## |
100% 2.03s
verifying ...
578 bytes of flash verified
safemode: Fuses OK
done. Thank you.
Clear
www.nex-robotics.com
Connected Done

6. Clear the screen after programming is done to avoid confusion while loading a new file as shown in the figure below.



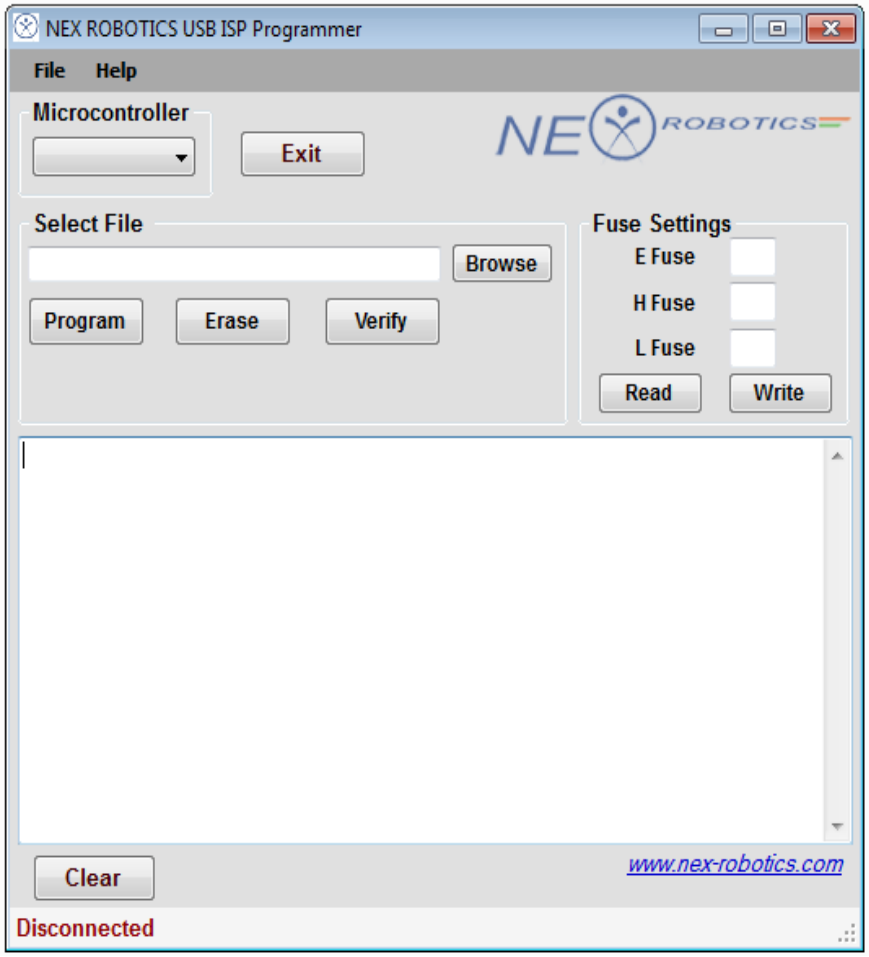
1.2.3 Loading bootloader file via STK500v2 GUI

Bootloader file provided by NEX-Robotics can be loaded on AVR microcontroller via STK500v2.

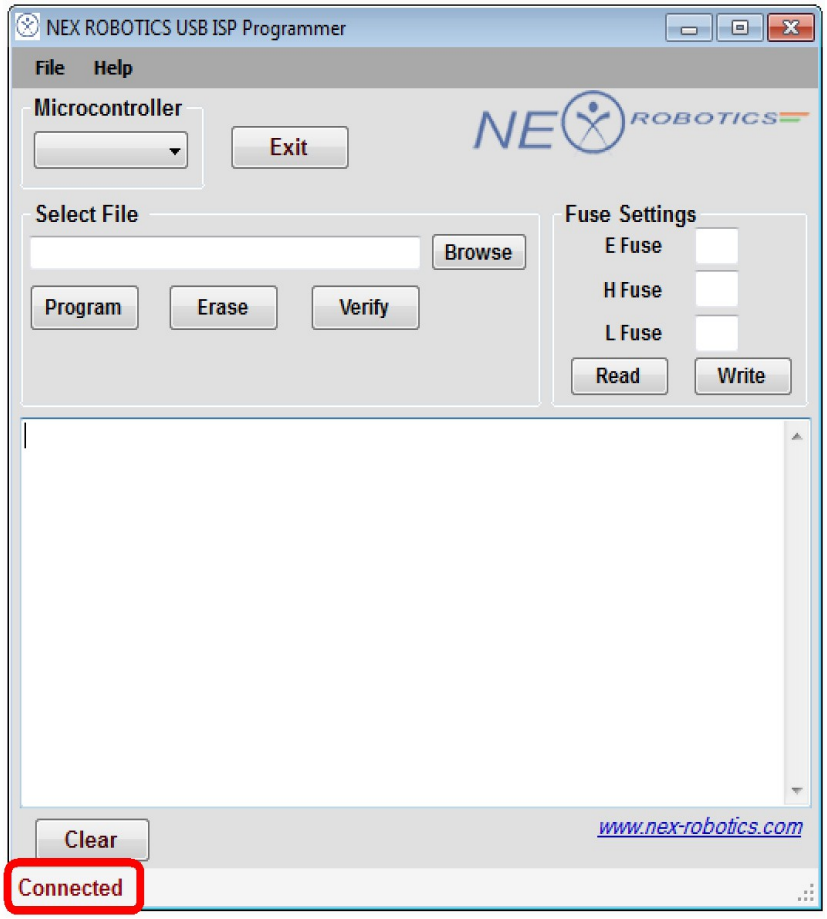
Bootloader file may take approximately 15 – 20 minutes to load via STK500v2 Programmer. To use bootloading we need to first write proper Fuse setting before loading bootloader file.

Follow the following procedure to load bootloader file.

1. Open Stk500v2 GUI. The following window shown below will pop up.

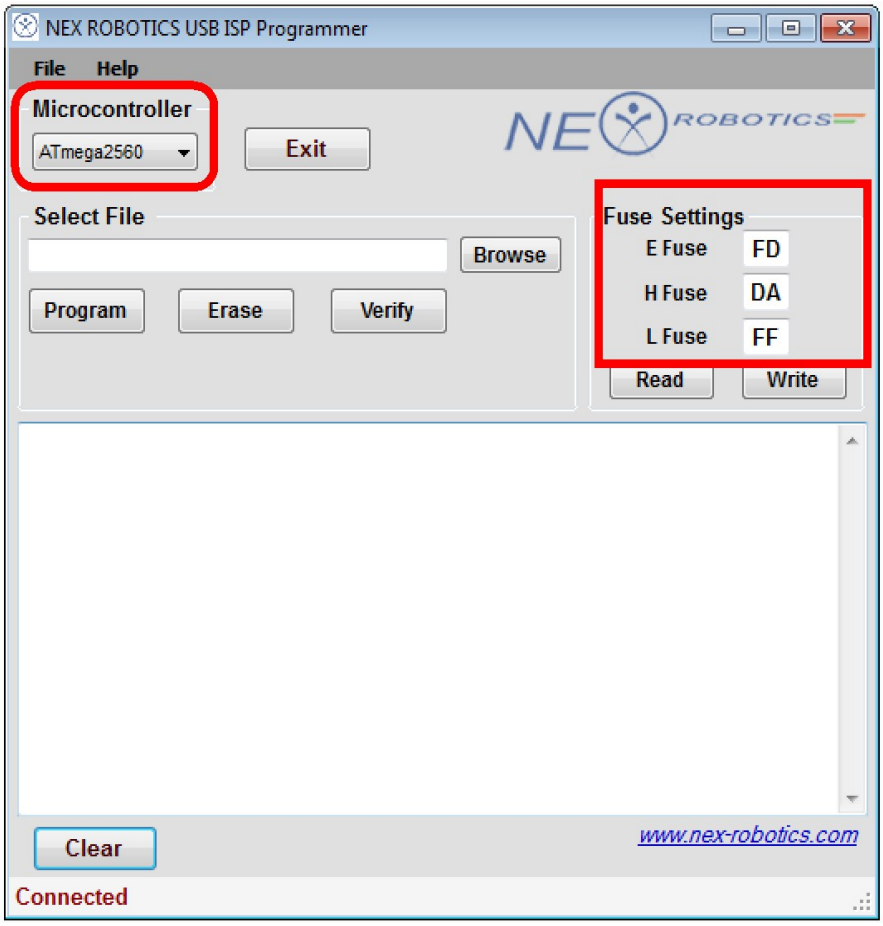


2. Connect STK500v2. In the bottom left of the GUI we can observe that the STK500v2 connection status has changed from “Disconnected” to “Connected” as highlighted in the below figure.

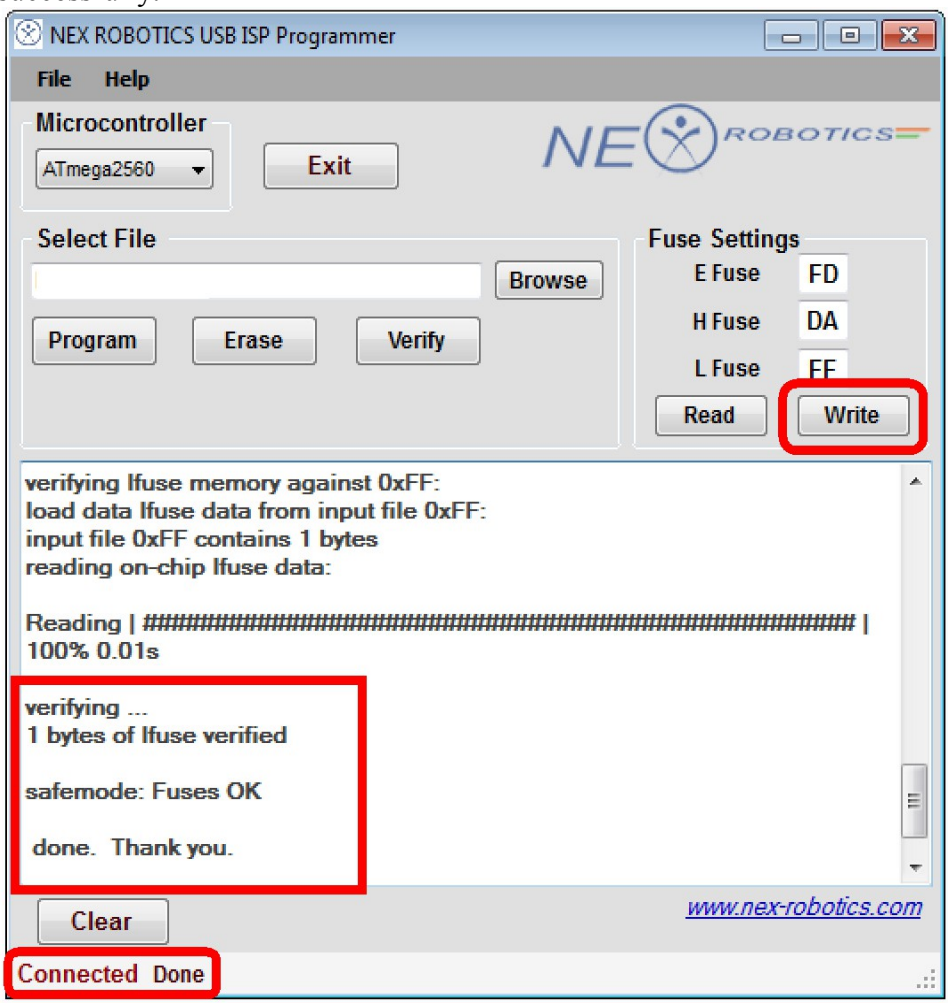


Note: - Correct Fuse settings are done in the microcontroller before shipping out our product. Do not change them unless you have very clear understanding of the fuse settings. Below Figure shows the Fuse setting for ATmega2560 Microcontroller. This information is only given for the reference.

3. Select proper microcontroller. Refer to Table 3 in section 2.5.2 for Extended, High and Low fuse values. Iput proper fuse values in "Fuse Setting" to enable BOOT RST of the microcontroller as shown in the figure given below. In the figure below we have choosen Atmega2560 as microcontroller and E Fuse as "FD", H Fuse as "DB" and L Fuse as "FF". This paticular fuse setting enables microcontrollers BOOT RESET functionality.

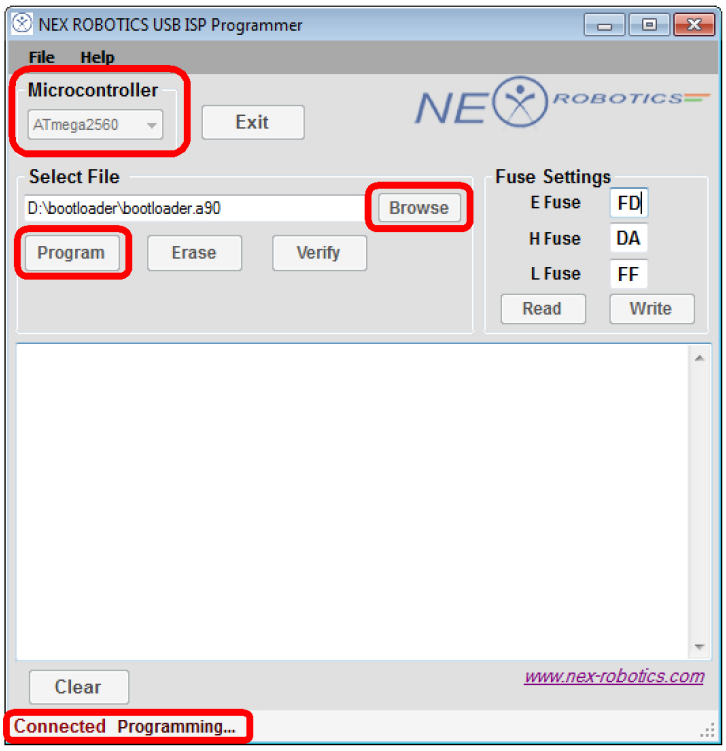


4. Click on “Write” to write these fuse setting on the microcontroller as shown in the figure below. If the fuse setting are properly written on the microcontroller, a “Done” text will be seen at the bottom left corner of the GUI also in the GUI window a Fuse OK status will also be seen along with a “done” text indicating that fuse setting has been written successfully.

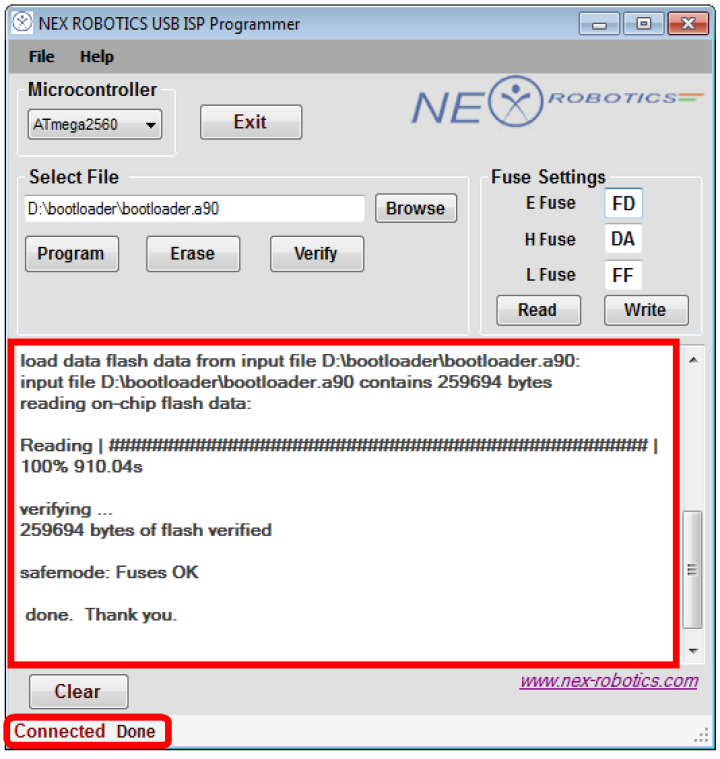


5. We can check the E Fuse, H Fuse and L Fuse details in the GUI window.

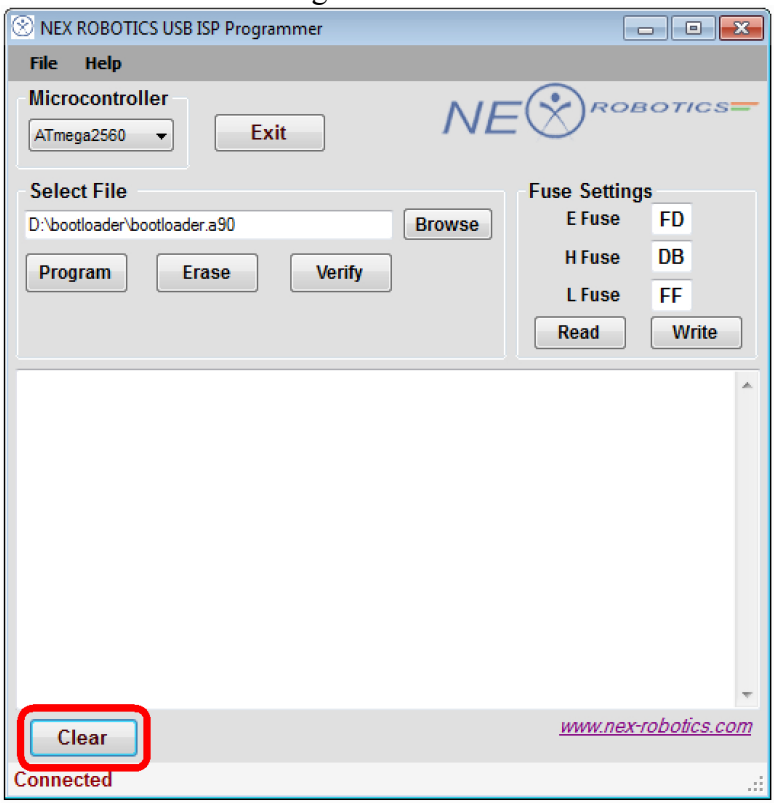
6. After writing and verifying the Fuse setting, clear the GUI window and select the bootloader file. Click on “Program” button to start writing the file on the microcontroller. It will Take approximately 15 – to 20 minutes to completely load the bootloader file. Do not terminate the GUI during this time. “Programming...” text will be constantly flash during this time at the bottom left corner of the GUI window.



7. After 15 – 20 minutes when the programming is done, a “Done” text will appear at the bottom left of the GUI and in the GUI window the Fuse OK status and Programing Done status will appear. All the details about the File and writing process can be seen on the GUI window.



8. Clear the GUI window before loading another file



1.2.4 Fuse setting

Use the following fuse values from the given tables to set the fuse setting in GUI.

1.2.4.1 Fuse setting to enable BOOT RESET on AVR Microcontroller

Use Fuse value from Table 2 to enable BOOT RST function of microcontroller

Development Board/Robotic Platform	Microcontroller name according to AVRDude	Microcontroller name in Stk500v2 GUI	Boot Size	Extended Fuse (E Fuse)	High Fuse (H Fuse)	Low Fuse (L Fuse)
Fire Bird V	m2560	ATmega2560	2048	FD	DA	FF
ATMEGA 2560/ ATMEGA 2560 MINI	m2560	ATmega2560	2048	FD	DA	FF
ATMEGA 640/ ATMEGA 640 MINI	m640	ATmega640	2048	FD	DA	FF

ATMEGA 128	m128	ATmega128	2048	FF	DA	FF
ATMEGA 128 MINI	m128	ATmega128	1024	FF	DC	FF
ATMEGA 32	m32	ATmega32	1024	----	DA	FF
ATMEGA 16	m16	ATmega16	1024	----	D8	FF
ATMEGA 8	m8	ATmega8	1024	----	D8	FF

Table 2: - Fuse setting for enabling BOOT RST fuse while writing microcontroller's Fuse

1.2.4.2 Fuse setting to disable BOOT RESET on AVR Microcontroller

Use Fuse values from Table 3 to write fuse setting without enabling BOOT RST finction of the listed microcontrollers.

Development Board/Robotic Platform	Microcontroller name according to AVRDude	Microcontroller name in Stk500v2 GUI	Boot Size	Extended Fuse (E Fuse)	High Fuse (H Fuse)	Low Fuse (L Fuse)
Fire Bird V	m2560	ATmega2560	2048	FD	DB	FF
ATMEGA 2560	m2560	ATmega2560	2048	FD	DB	FF
ATMEGA 640	m640	ATmega640	2048	FD	DB	FF
ATMEGA 128	m128	ATmega128	2048	FF	DB	FF
ATMEGA 128	m128	ATmega128	1024	FF	DD	FF
ATMEGA 32	m32	ATmega32	1024	----	DB	FF
ATMEGA 16	m16	ATmega16	1024	----	D9	FF
ATMEGA 8	m8	ATmega8	1024	----	D9	FF

Table 3: - Fuse setting for microcontroller's without enabling BOOT RST while writing microcontroller Fuse