

Atmega Fuse Fix Instructions

To use this device, simply insert your avr in the correct socket (see Table 1 below) and connect power. Within one second either the green or the yellow led will illuminate. A Green LED indicates success, A Yellow LED indicates failure. Failure can be caused by a number of reasons:

1. You do not have the correct power supply. This is often the problem when the programming fails. Please use a 13.5 - 18 VDC power supply with positive centre on the plug. When in doubt, measure the output of your adapter with a voltmeter. (minimum 13.5 volts output)
2. You are not inserting the chip in the correct socket. Please see Table 1 below.
3. The device signature cannot be read. Is the device damaged? Is it inserted correctly?
4. The device is not supported. Please check the device list below. (Table 1) or enter terminal mode and disable signature verification. (read more below)

If the device does not reprogram your avr and there are no led's illuminated, perhaps you have a damaged power supply. Once power is connected, the red led will illuminate to indicate power has been applied. If it does not, you are not using the correct power supply.

PLEASE BE AWARE that in order to reprogram the fuses (if the lock bits have been programmed) the programmer must erase your avr device. If jumper J1 is installed before the programmer is powered up, the erase operation will be bypassed, preserving the memory while restoring the fuses to factory defaults.

Please Remember: If the LOCK bits have been programmed in your device, the fuse restore operation will fail if jumper J1 is installed.

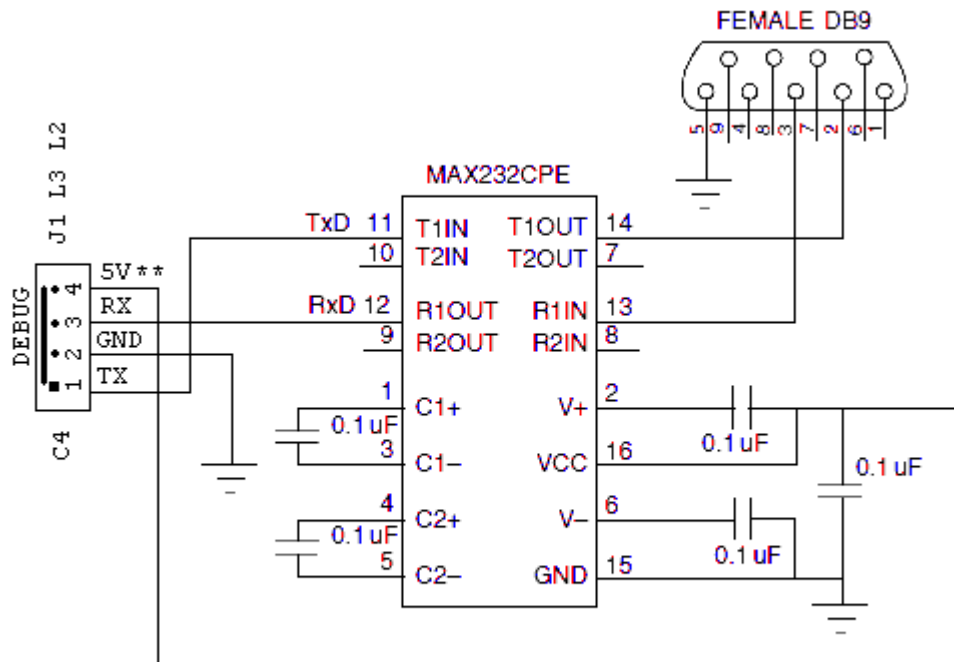
New in Version 3, "advanced mode" or "terminal mode". This mode works with a terminal on your PC. It allows reading and writing of the fuse bytes. Also included is a new feature that disables signature verification, allowing the user to program fuses on avr devices that are not in the device list (Table 1).

WARNING: We are not responsible for damage caused by programming non standard fuse values.

KNOW WHAT YOU ARE DOING!

Connecting the Programmer to the Terminal for Debugging

If you are continuously seeing errors (Yellow LED), you can connect the programmer to a terminal to view the results. We recommend using hyperterminal. Some terminal programs will not display the text correctly particularly the clear screen commands, therefore the text will not be formatted correctly.



**The Fuse Programmer supplies the Max232 with 5V.

Fig 2. Connecting the Programmer to the Computer's Serial Port

When connected, the circuit will display fuse information or error messages to a terminal at 9600 baud. Hyperterminal settings: **Baud Rate: 9600, Data Bits: 8, Stop Bits: 1, Parity: None, Flow Control: None, Terminal Emulation: VT-100 (ANSI).**

A low Cost Serial to TTL Converter can be found on our [website's main page](#).

If you do not have a serial port on your PC, the programmer can be connected to one of our low cost USB to TTL adapters found on our [website's main page](#). See Fig 3 for connection details.



Fig 3. Connecting the Programmer to the Computer's USB Port

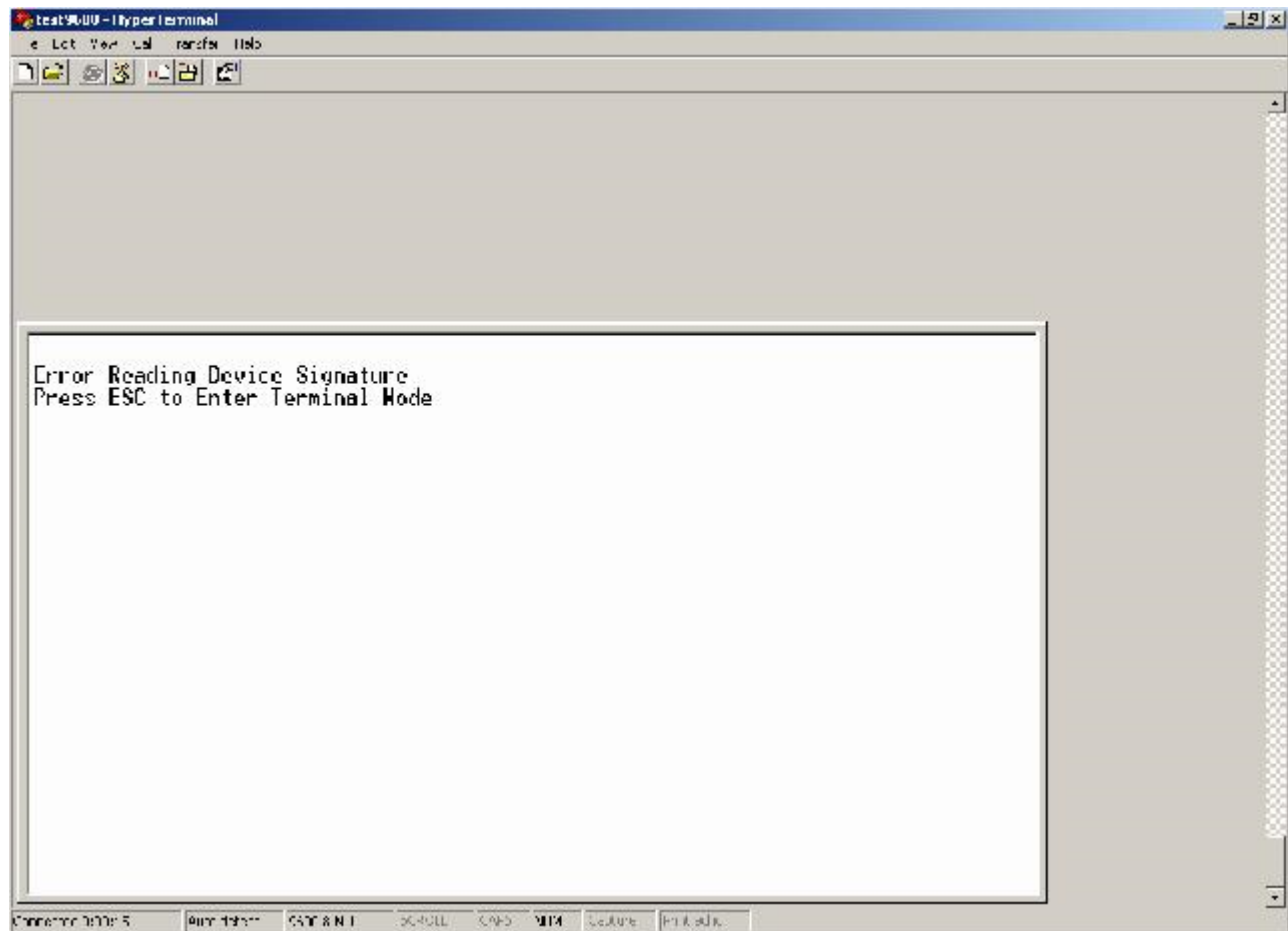
Supported Devices for Atmega Fuse Repair Programmer
****** (All SMD pins must be wired directly to the socket)**

<u>Socket #1</u>	<u>Socket #2</u>	<u>Socket #3</u>	<u>Socket #4</u>	<u>Socket #5</u>
ATmega16	ATtiny2313	ATmega8515	ATmega8	ATtiny26
ATmega16L	ATtiny2313V	ATmega8515L	ATmega8L	ATtiny26L
ATmega32		ATmega128****	ATmega48	ATtiny261
ATmega32L		ATmega162	ATmega48V	ATtiny261V
ATmega161		ATmega162V	ATmega48P	ATtiny461
ATmega161L			ATmega48PV	ATtiny461V
ATmega164			ATmega88	ATtiny861
ATmega164V			ATmega88V	ATtiny861V
ATmega324			ATmega88P	
ATmega324V			ATmega88PV	
ATmega644			ATmega168	
ATmega644V			ATmega168V	
ATmega8535			ATmega168P	
ATmega8535L			ATmega168PV	
			ATmega328P	
			ATmega328PV	

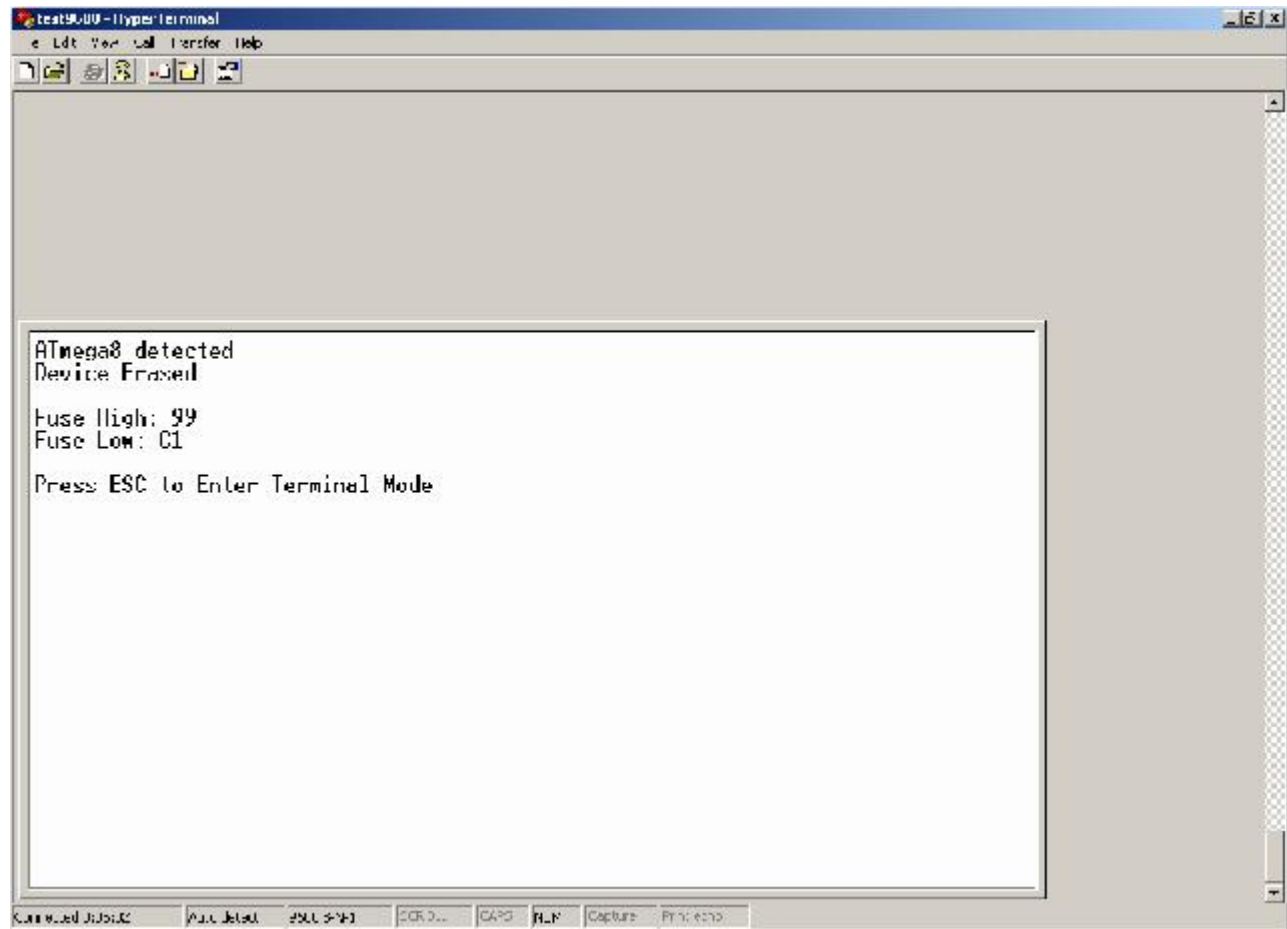
Table 1.

Using the Programmer

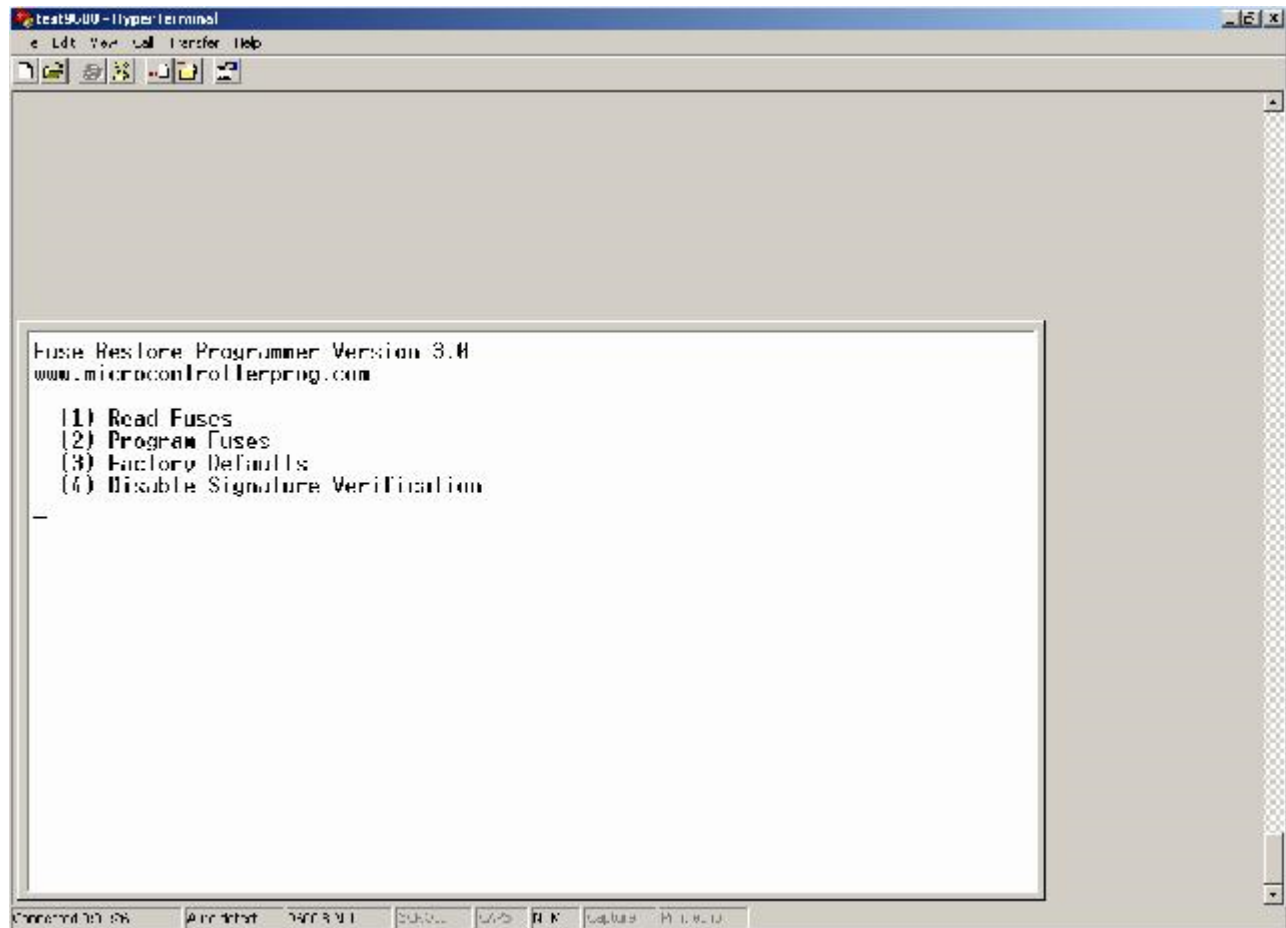
If the programmer is connected to the terminal, and it is powered up with no devices inserted, you will see a screen like this:



If a device was inserted, you will see this message:

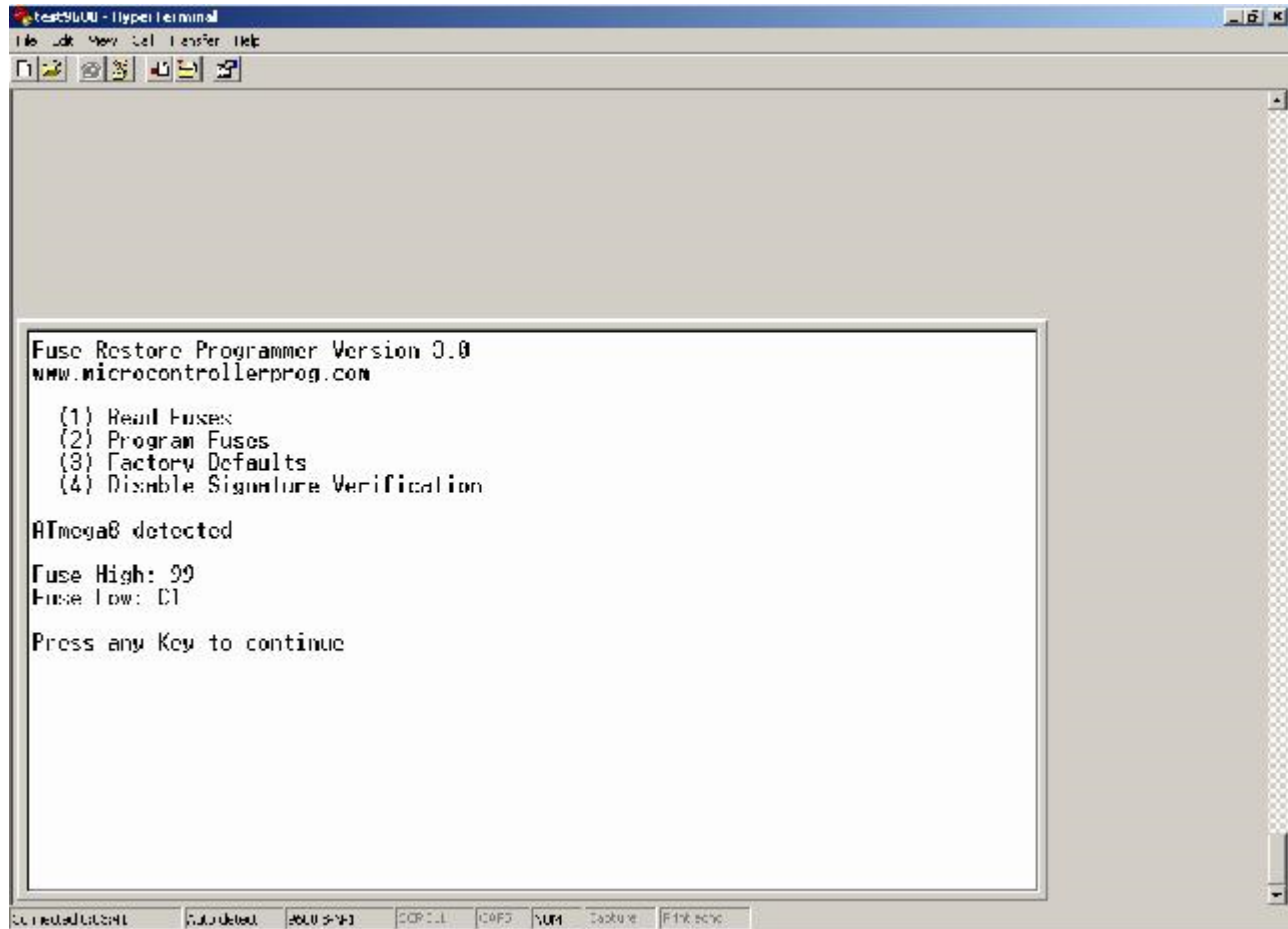


Pressing ESC gets you into MODE3 (Advanced Mode). You will then see a screen like this:



You now have Four Options.

1. **Read Fuses.** Have you ever been curious to know what the fuse values are when your avr has been corrupted by your ISP programmer? Pressing "1" will show you the fuses. See below:



```
Fuse Restore Programmer Version 0.0
www.microcontrollerprog.com

(1) Read Fuses
(2) Program Fuses
(3) Factory Defaults
(4) Disable Signature Verification

ATmega8 detected
Fuse High: 99
Fuse Low: C1

Press any Key to continue
```

Pressing any key will always return you back to the main menu.

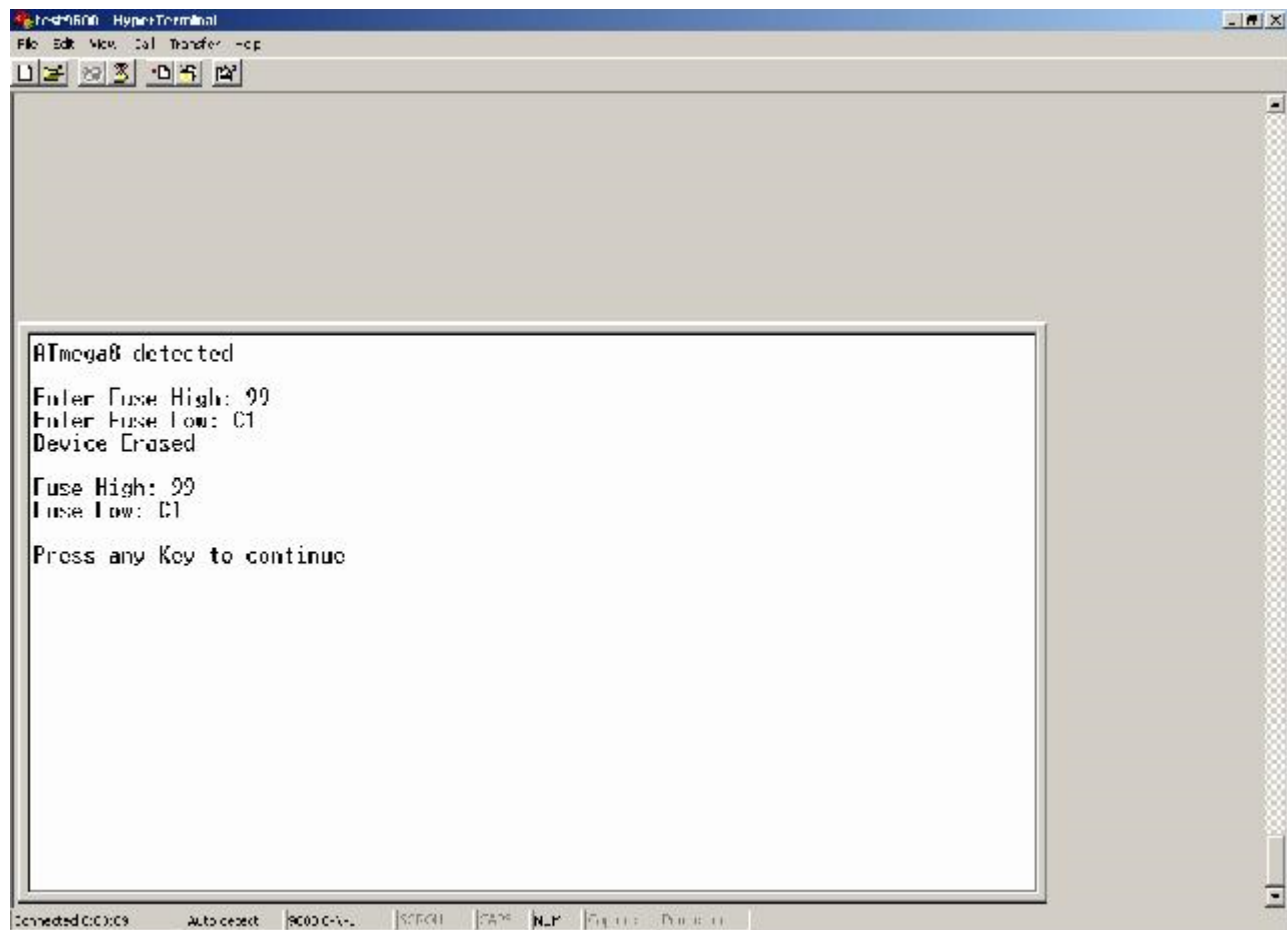
2. Pressing "2" will allow you to program the fuses on any of the devices in the device list. If you would like to program any device NOT in the device list, you must select "4" (Disable Signature Verification) before pressing "2", Program Fuses.

AGAIN, PLEASE Be careful here! NON Standard fuse values may cause unknown results preventing your avr from entering programming mode, even with this programmer!!

BE SURE YOU KNOW WHAT YOU ARE DOING!!

If signature verification is disabled, the programmer does not know any information about the device you are trying to program. (i.e. The default fuse values, whether or not the device has high, low or extended fuse bytes or all three). When programming a device (i.e. a mature avr device such as the 90S series) you will be asked for all three fuse bytes. Even though the device has only one fuse byte, you must enter the same fuse byte for all three fuse bytes.

For example, you want to restore the SPIEN fuse on a AT90S8515 and disable the FSTRT fuse at the same time, enter hex value DF (not case sensitive) for all three values (high fuse, low fuse, and extended Fuse bytes). The avr will ignore the extra fuse bytes. After programming, the programmer will read back three fuse bytes. This mode does not work for restoring factory defaults since the programmer must know what the default values are based on the device's signature. This is what the screen will look like when programming fuses without disabling the signature verification:



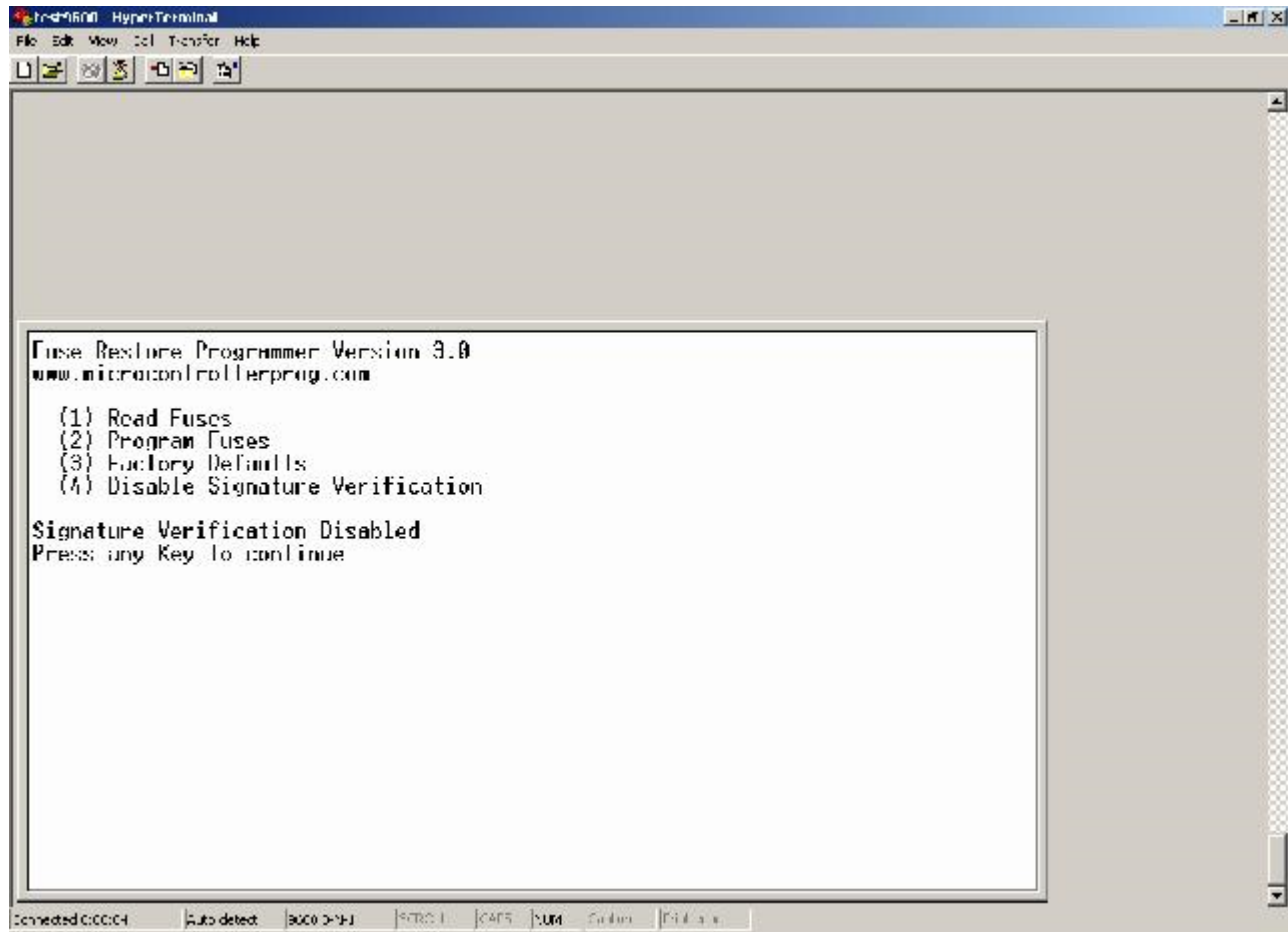
```
ATmega8 detected
Fuser Fuse High: 99
Fuser Fuse Low: C1
Device Erased

Fuse High: 99
Fuse Low: C1

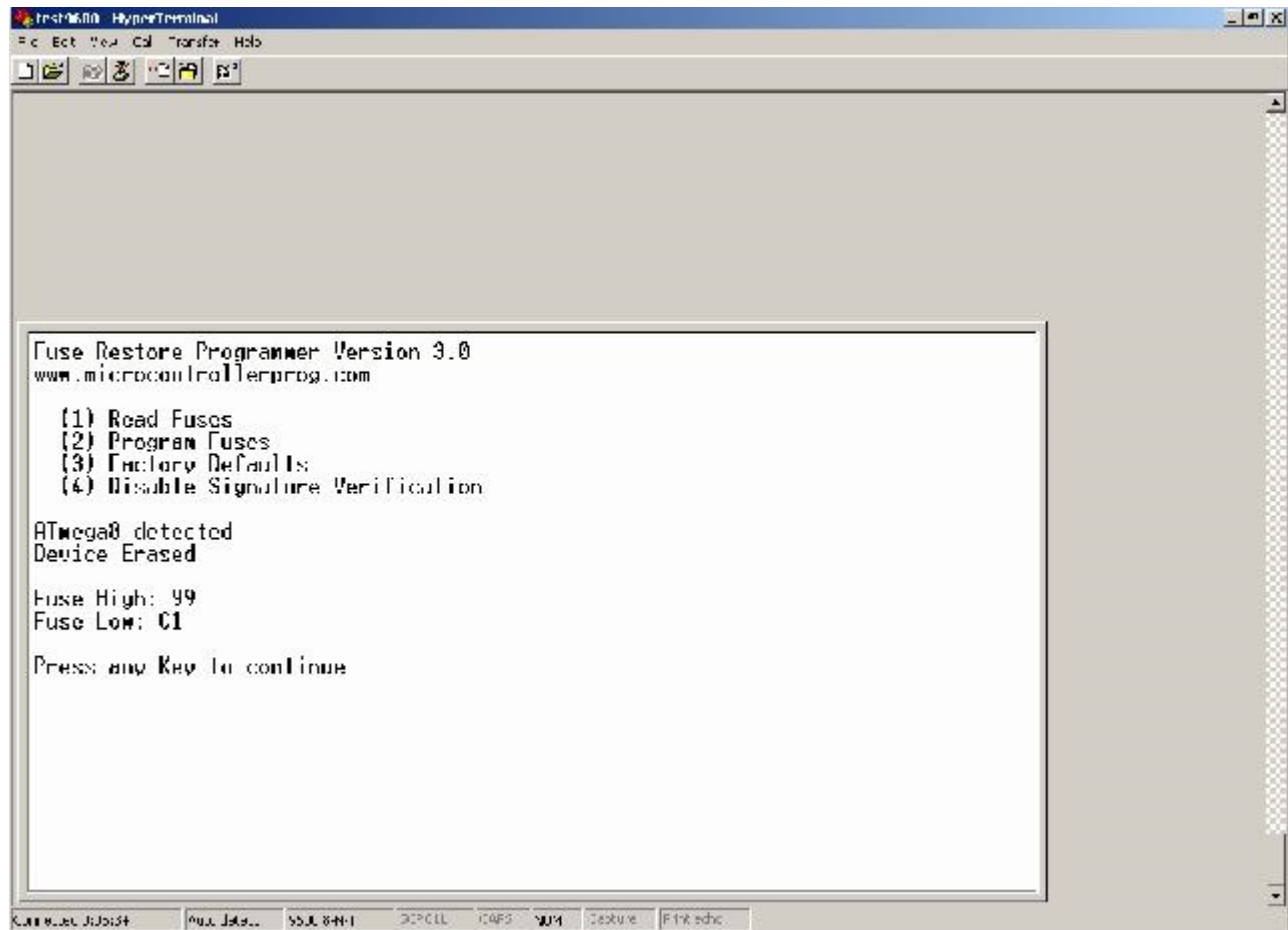
Press any Key to continue
```

The screenshot shows a HyperTerminal window with a menu bar (File, Edit, View, Cal, Transfer, Help) and a toolbar. The main text area displays the output of a device detection process. At the bottom, a status bar shows connection details: Connected (COM9), Auto reset, 9600 Baud, 8N1, 5V, and a baud rate of 9600.

The only way to Enable Signature Verification (if you disabled it) is to restart the programmer. (unplug power and reconnect). Signature verification is always enabled by default. When pressing "4" (Disable Signature Verification), you will see a screen like this:



Pressing "3" will restore the fuses to factory defaults, same as in standalone mode and will not be successful if signature verification has been disabled.



Factory Defaults selection "3". You can exit Advanced mode by powering off the programmer.

Any questions? Email us at: microcontrollerprog@yahoo.com