

Contents

Table of Contents

| | |
|------------------------------------|----------|
| Troubleshooting | 2 |
| Erase Bypass | 3 |
| Mode 3 Advanced Mode | 3 |
| Connection to a Serial Port | 3 |
| Connection to a USB Port | 4 |
| Supported Device List | 5 |
| Using The Programmer | 6 |

Atmega Fuse Repair Programmer User Guide

To use this device, simply insert your avr in the correct socket (see Table 1 below) and connect power. Within one second the programmer will restore the fuses to factory defaults. Either the green LED 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. 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 use a 13.5 - 16 Volt DC power supply with positive center on the plug. It is very important that you measure the output from your power supply using an accurate voltmeter! We are not responsible if you damage the programmer, or anything connected to it. Incorrect voltage or polarity may cause damage to you and/or the programmer!**
- 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)**
- 5. Do you have jumper J1 installed? If the lock bits have been programmed and J1 is installed, the programmer will fail. The device must be erased if the lock bits have been programmed. Please remove jumper J1 and power cycle the programmer, and try again. For more information see "Erase Bypass" below.**

Please Remember: This device is not designed to circumvent LOCK BITS!

Erase Bypass

By installing jumper J1, the erase function will be skipped. This feature allows you to save the code while restoring the fuses. J1 must be installed before the device is powered on. If the lock bits have been programmed, programming will fail if J1 is installed. You must remove J1 in order to restore fuses to default values.

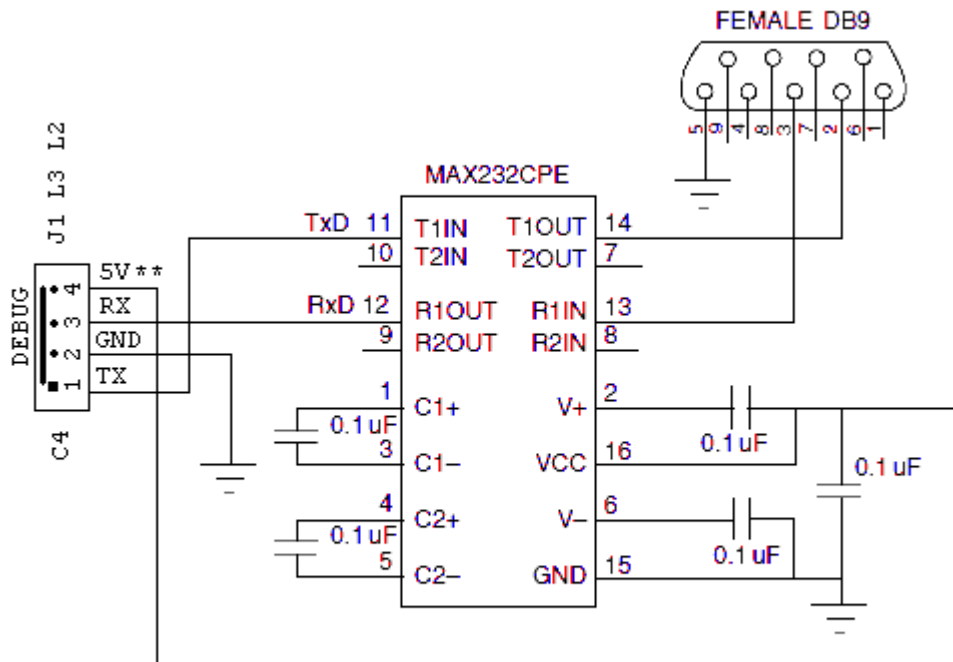
AGAIN, This programmer is not designed to circumvent LOCK BITS!

Mode 3 Advanced Mode

This mode works with a terminal on your PC. It allows reading and writing of the fuse bytes. Also included is a 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!**

****The fuse programmer supplies the MAX232 with 5 Volts.**



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

Fig 1. Connecting the Programmer to the Computer's Serial Port

When connected, the programmer 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 [here](#).

If you do not have a serial port on your PC, the programmer can be connected to one of our low cost [USBtoTTL](#) adapters found on our website [here](#). See Table 2 below for connection details.

| <u>Atmega Fuse Repair Programmer</u> <u>Debug Ports</u> | <u>USBtoTTL Converter</u> |
|--|----------------------------------|
| 1. RX | TXD |
| 2. TX | RXD |
| 3. - | GND |
| 4. + | 5V |

Table 2. Connecting the Programmer to the Computer's USB Port using the [USBtoTTL](#).

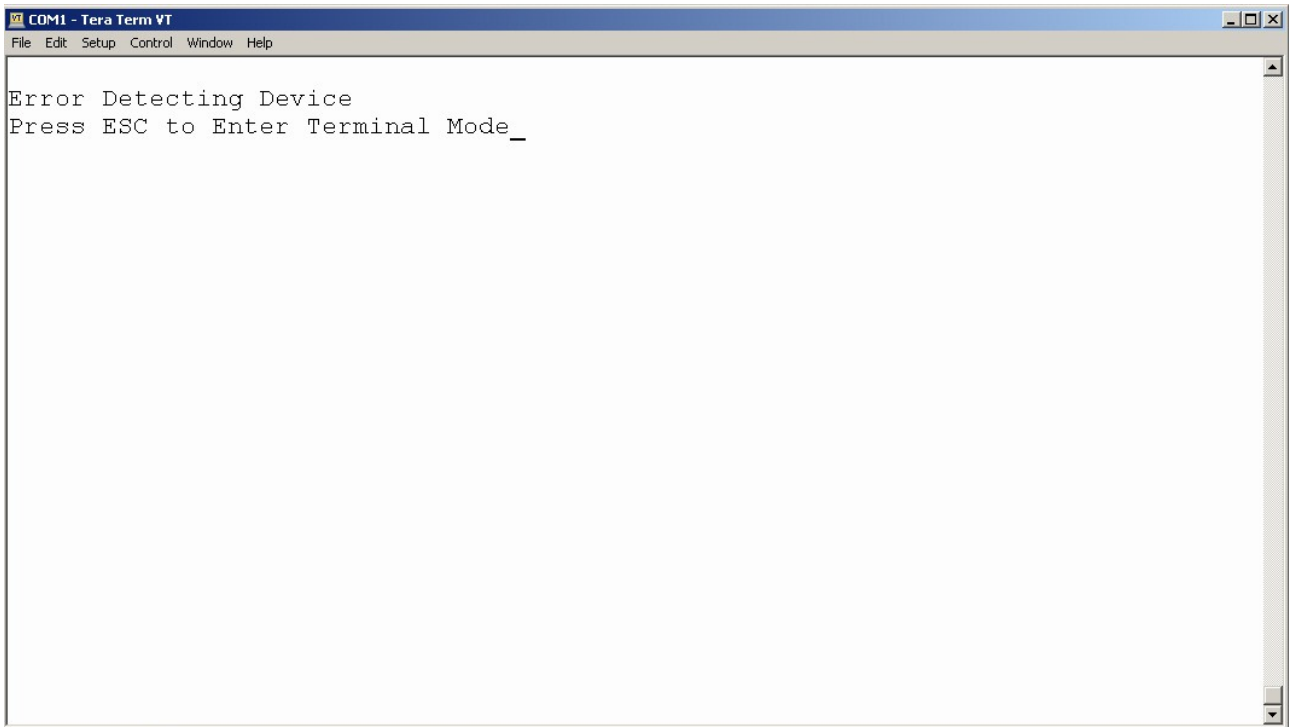
**Supported Devices for Atmega Fuse Repair Programmer:
 **** (All SMD pins must be wired directly to the socket)**

| Socket #1 | Socket #2 | Socket #3 | Socket #4 | Socket #5 |
|------------------|------------------|------------------|------------------|------------------|
| ATmega16 | ATtiny2313 | ATmega8515 | ATmega8 | ATtiny26 |
| ATmega16L | ATtiny2313A | ATmega8515L | ATmega8A | ATtiny26L |
| ATmega32 | ATtiny2313V | Atmega128** | ATmega8L | ATtiny261 |
| ATmega32L | ATtiny4313 | ** | ATmega48 | ATtiny261A |
| ATmega161 | ****The | ATmega162 | ATmega48A | ATtiny261V |
| ATmega161L | following | ATmega162V | ATmega48V | ATtiny461 |
| ATmega164 | devices are | | ATmega48P | ATtiny461A |
| ATmega164A | supported in | | ATmega48PA | ATtiny461V |
| ATmega164P | auto mode, | | ATmega48PV | ATtiny861 |
| ATmega164V | however | | ATmega88 | ATtiny861A |
| ATmega164PV | they must | | ATmega88A | ATtiny861V |
| ATmega164PA | be wired to any | | ATmega88P | |
| ATmega164PA | socket. | | ATmega88PA | |
| ATmega324 | ATtiny43U | | ATmega88V | |
| ATmega324A | ATtiny87 | | ATmega88PV | |
| ATmega324P | ATtiny88 | | ATmega164 | |
| ATmega324PA | ATtiny167 | | ATmega164A | |
| ATmega324V | | | ATmega164P | |
| ATmega644 | | | ATmega164PA | |
| ATmega644A | | | ATmega168 | |
| ATmega644P | | | ATmega168A | |
| ATmega644V | | | ATmega168V | |
| ATmega644PA | | | ATmega168P | |
| ATmega644PV | | | ATmega168PA | |
| ATmega8535 | | | ATmega168PV | |
| ATmega8535L | | | ATmega328P | |
| Atmega1284 | | | ATmega328PV | |
| ATmega1284P | | | | |

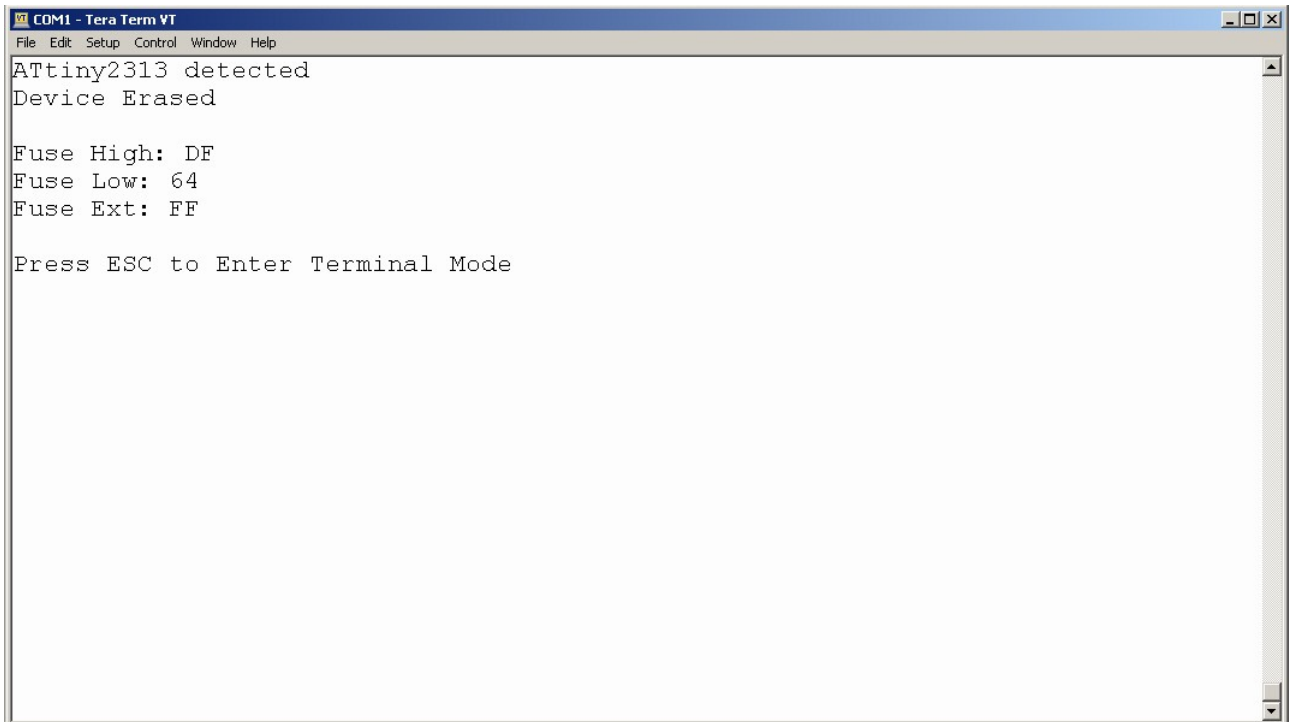
Table 1.

Using the Programmer

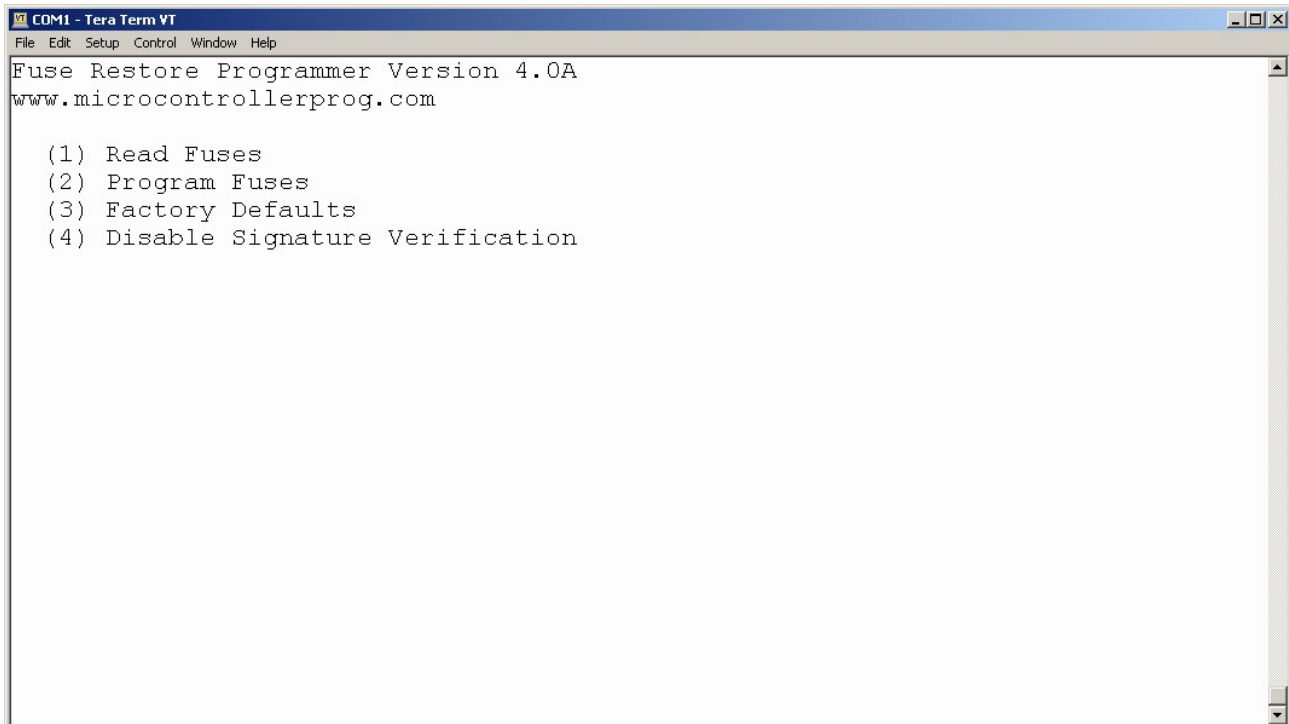
If the programmer is connected to the terminal, and it is powered up with no devices inserted, you will see a screen like this:



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 device has been corrupted? Pressing "1" will show you the fuses. Then 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 (Table 1 below), you must select "4" (Disable Signature Verification) before pressing "2", Program Fuses.

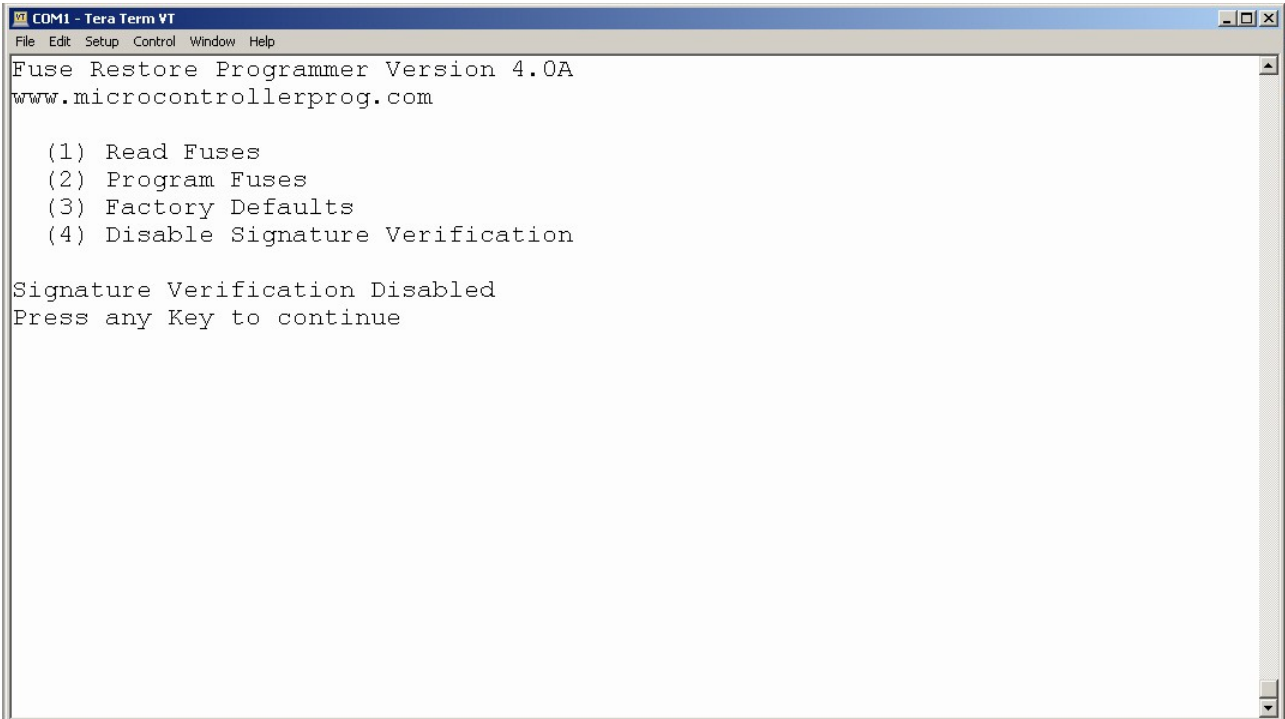
**AGAIN, PLEASE Be careful here! NON Standard fuse values may cause unknown results preventing your device from entering programming mode, even with this programmer!!
BE SURE YOU KNOW WHAT YOU ARE DOING!!**

If signature verification is disabled (Selection "4") , the programmer does not know any information about the device it is trying to program. (i.e. The default fuse values, or 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 AT90S2313 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 device will ignore the extra fuse bytes. After programming, the fuse repair programmer will read back three fuse bytes. This mode does not work for restoring factory defaults since the information is unknown as to what the default values are based on the device's signature.

The only way to Enable Signature Verification (if you disabled it) is to restart the programmer. (unplug power and reconnect) or select Factory Defaults by pressing "3". Signature verification is always enabled by default. When pressing "4" (Disable Signature Verification), you will see a screen like this:



3. Factory Defaults selection "3".

Pressing "3" will restore the fuses to factory defaults, same as in standalone mode. If signature verification was disabled, it will be re-enabled by selecting factory defaults.

You can exit Advanced mode by powering off the programmer.

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