AT89S Standalone Programmer Version 2 Instructions

The following instructions are written for programmer version 1.7. You can obtain a FREE upgrade if your programmer's version is below version 1.7. Please email for information. This programmer will program the ATMEL AT89S devices, in circuit without the use of a computer.*

The devices that are supported are:

AT89S2051
AT89S4051
AT89LP2052
AT89LP4052
AT89S51
AT89S52
AT89S53
AT89S8252
AT89S8253

****(A computer is needed for configuration and to load the hex file)**

Set Up Terminal Program on Your PC

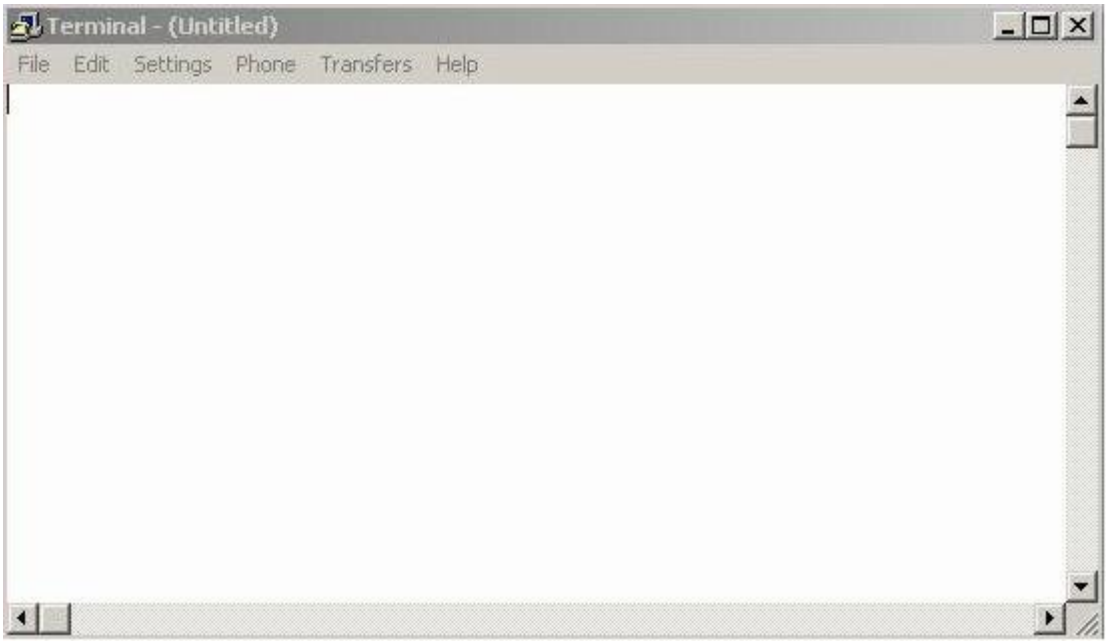
A hex file must be loaded into the programmer's memory for storage. This can be done using any terminal program. For this example, we are using the terminal program downloaded from here.

This program works well in Windows XP and Vista. There is a shortcut in the zip file called Test96.trm. It can be copied to your desktop, and is already set up with the line delay as mentioned above. In this case it is 2/10 sec delay per line.

If using Hyperterminal, Set up communication for 9600 baud, no handshaking, 8, N, 1. Line delay 100mS.

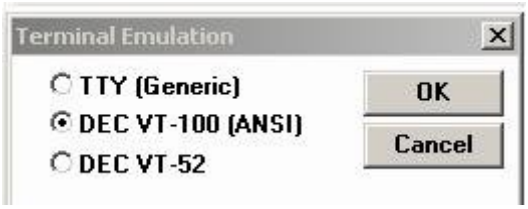
Very important is to slow down the incoming data to line at a time with a line delay of 100mS minimum. This can be done with hyperterminal. If you do not slow down the terminal, data will be missed.

Extract the files terminal.exe and terminal.hlp to c:/terminal. Run terminal.exe. You will get a screen like this:

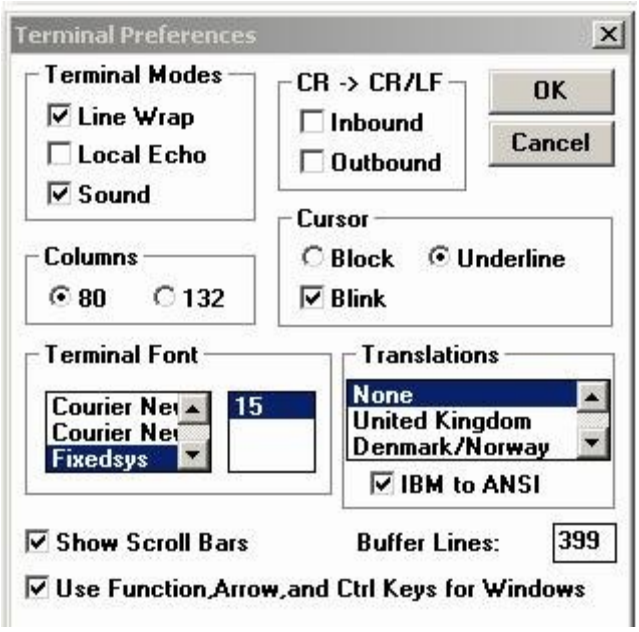


To set up the terminal program, please follow these instructions:

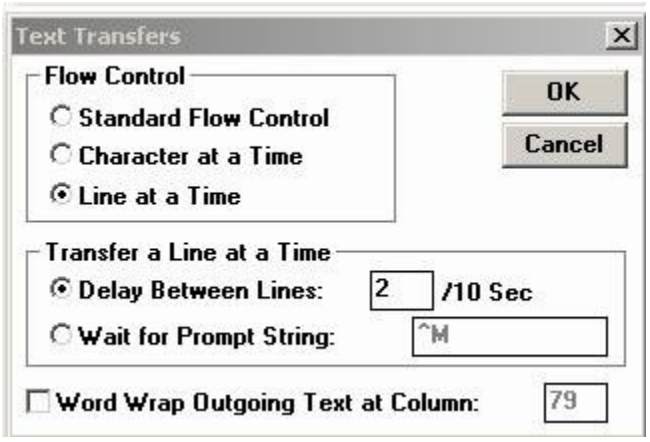
Select settings, Terminal Emulation. The settings should look like this:



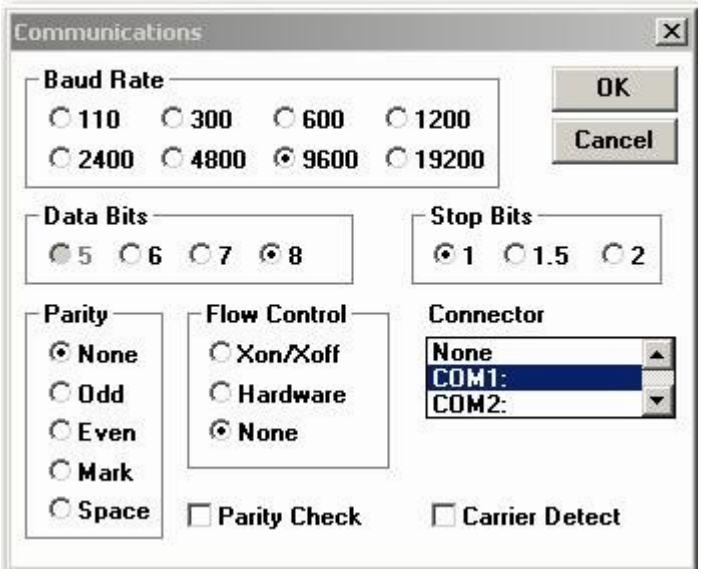
Select OK, then select Settings, Terminal Preferences. The settings should look like this:



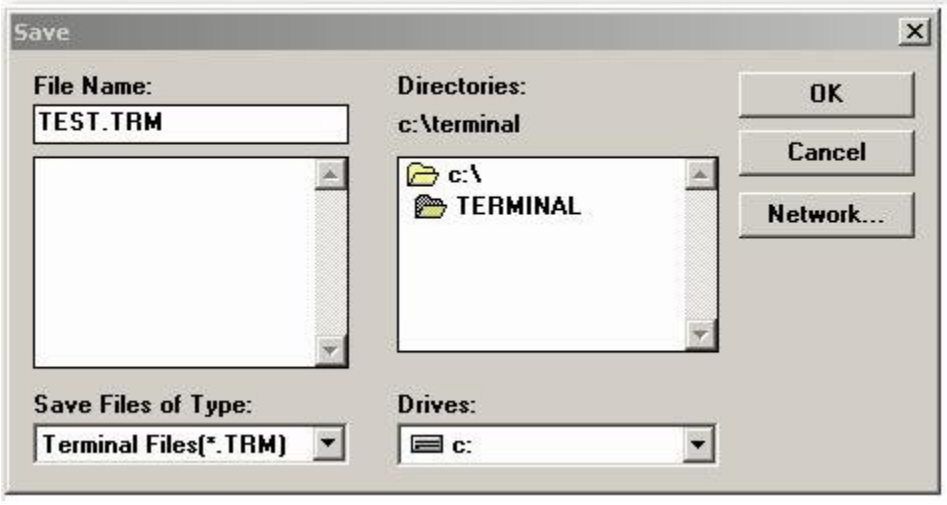
Again Select OK to save your settings. Select Settings, Text Transfers. You should get a window like this:



Ensure that all of your settings match the above. The "Delay between Lines" setting is VERY IMPORTANT. If this is not set correctly the programmer will lose data from the PC resulting in your hex files not programming correctly. When you are finished select OK. NEXT, select settings, Communications. You will see a window like this:



Ensure that all of your settings match the above, and select OK. Finally, select FILE, SAVE to save your settings in a file. (see below)



You can copy the file test.trm to your desktop and use it as a shortcut for next time.

Using your Programmer

The programmer can take power from the target or from the USBtoTTL interface. This can be configured with the jumper J4.

Figure 1.

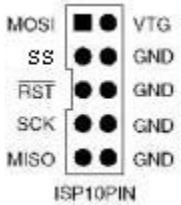
Jumper J4 installed to the right hand side, programmer will take power from USBtoTTL interface.

Figure 2.

Jumper J4 installed to left hand side. Programmer will take power from the target. (Maximum 5VDC)

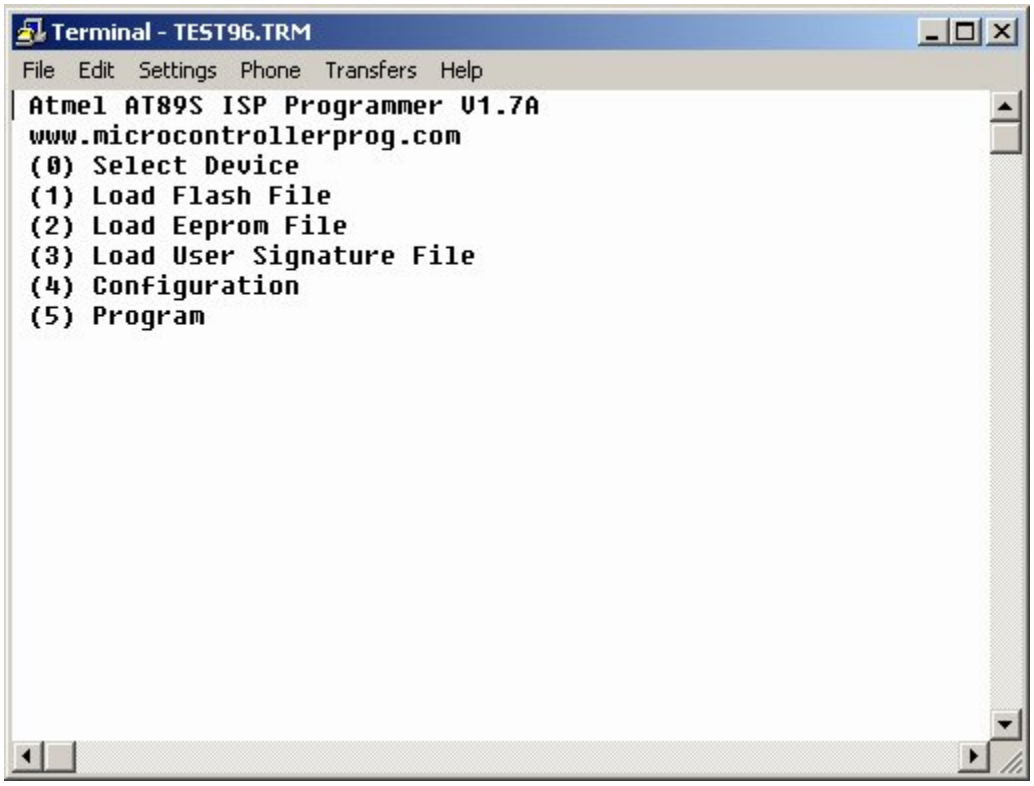
*****Please remember that there is no protection if you are using power from the target! Please be sure that the target voltage does not exceed 5VDC.*****

The programmer uses the ATMEL standard 10 pin ISP connector.



The target must use its own clock. The SS pin connection is only required on the AT89LP devices. It will be ignored on all other devices.

Before powering up the programmer, please ensure that jumper J1 has been removed, or the programmer will automatically boot into Stand Alone programming mode. When the programmer is powered up, you will see a menu appear on the screen. You then have six options. The first thing you MUST DO is select the device you will be programming. The MAIN MENU will look like this:



FROM THE MAIN MENU Select 0. Select Device

When 0 is selected, you will see another menu like this:

The programmer will display the last device that was selected. To select a device, simply enter 1 - 9. When you are done, press ESC on your keyboard, to return to the main menu.

FROM THE MAIN MENU Select 1, Load Flash File

(Load a hex file into the programmer's flash buffer.) The programmer will begin to clear the buffer for the file to be stored. You must wait for the prompt ("Send File Now") before sending the file. On devices with large capacity, this may take a long time. The file will be stored in an eeprom for programming the flash memory. The data will remain in the memory until it is overwritten. The file must be in hex format with a valid checksum at the end of every line and a valid end record at the end of the file. A valid end record looks like this:

:00000001FF

A bad checksum will return an error message. The file will be programmed to your device according to the addresses in the hex file.

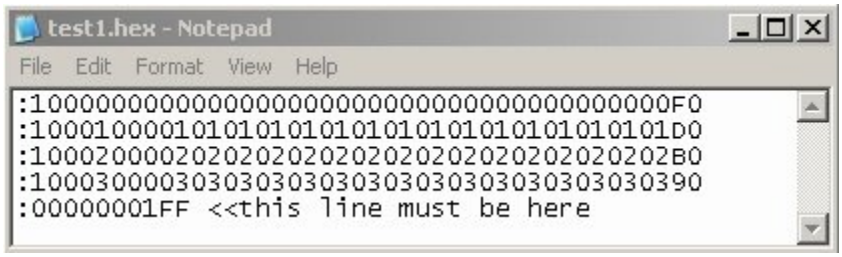
FROM THE MAIN MENU Select 2, Load Eeprom File

(Load a hex file into the programmer's eeprom buffer.) This is the same as loading the flash file except that the file will be used to program the eeprom in devices where an eeprom is present. If your device does not have an eeprom, this can be ignored.

FROM THE MAIN MENU Select 3, Load User Sig File

Same thing again. (Load a hex file into the programmer's User Signature Row Buffer) The file will be stored in an eeprom for programming the User Signature Row in devices that support that feature. The file will remain in the memory until it is overwritten. The maximum filesize is 64 bytes to match the devices supported with this programmer.

The file must be in hex format with an end record. The address values here are not important, however the length of each line must be 0x10 (16 bytes). If it is not, wn error message will be displayed. Here is a sample of the USER Signature File that the programmer will accept:



The addresses in the hex file not matter, they will be ignored. The programmer will load the lines in consecutive order, regardless of the addresses. Each line must be complete, and there must be at least one line, up to 4 lines (depending on the capacity of the user signature row in the device selected) If your device has 32 bytes of user signature and you load in 64 byte file, only the first 32 bytes will be programmed. If the file is larger than 64 bytes, only the first 64 bytes will be loaded and the programmer will wait for the last line (the EOF). The last line tells the programmer when to stop the loading process. If that line is omitted, the programmer will continue to load the file forever, even if the data after the first 64 bytes is ignored! If there is a bad checksum at the end of a line, an error message will be displayed.

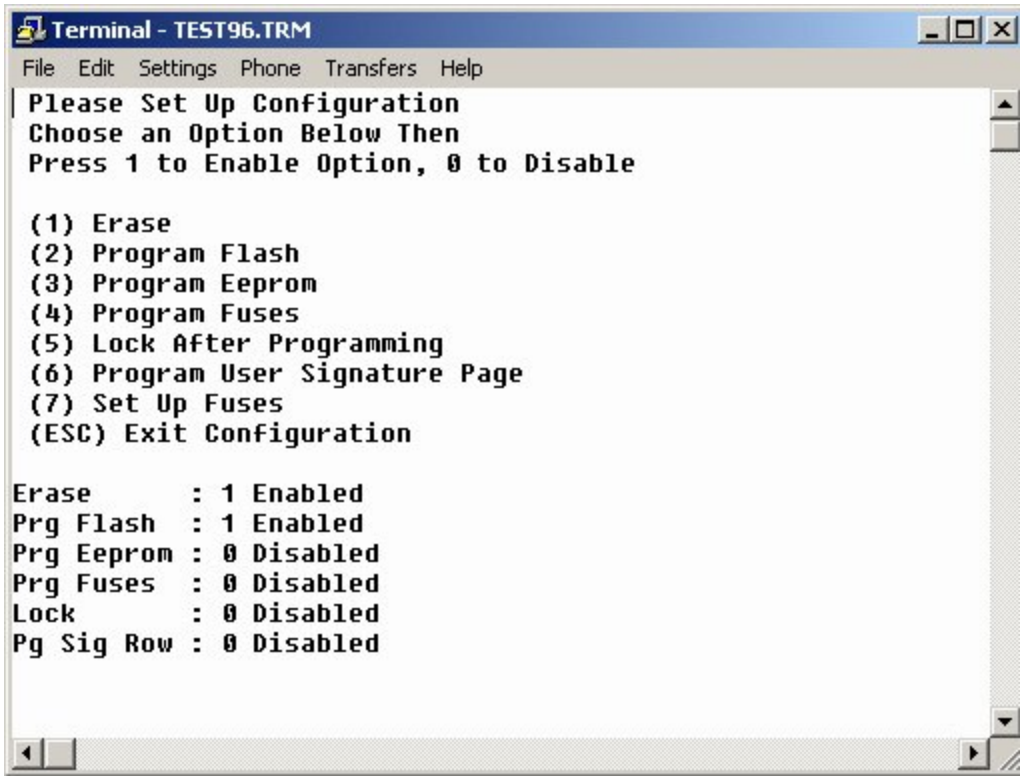
***** IMPORTANT NOTE:*****

On devices that contain a User Signature Row, the USER signature fuse must be

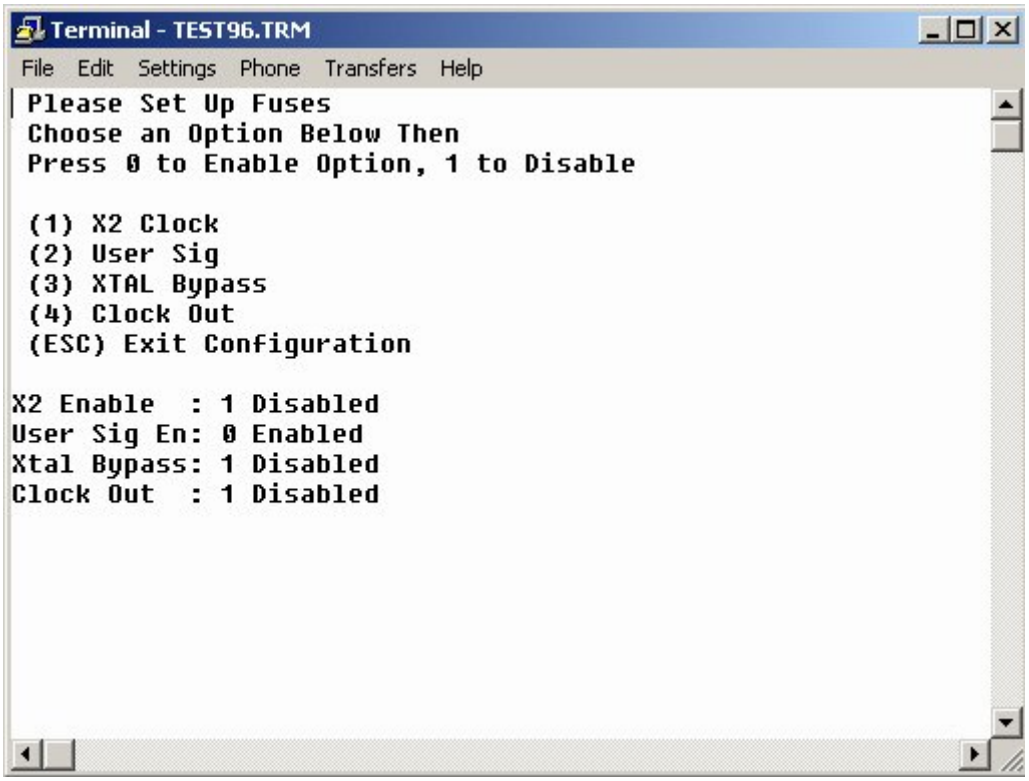
programmed in order for the programming to be successful. The programmer will program the fuses (if enabled) before programming the User Row.

FROM THE MAIN MENU Select 4. Configuration

You MUST set up configuration in order for the programmer to work! When "4" is pressed, you will see the configuration menu which will look like this:



Select 1, then press either 1 or 0 to enable/disable that option. Do the same for #2, and #3 up to #7. The configuration settings will remain in memory. They can only be overwritten next time you enter the configuration menu. The only way to exit this menu is to press #8. If #7 is selected, ("Set Up Fuses"), a menu will be shown like this one:



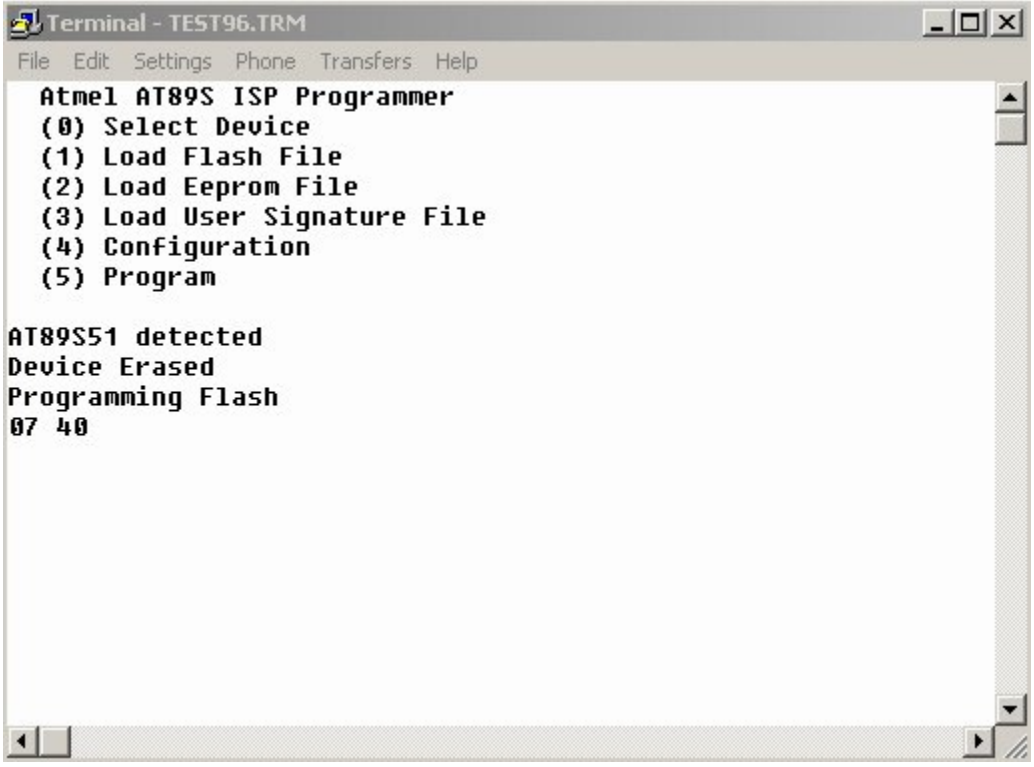
Same thing again, Select 1, then press either 0 or 1 to enable/disable that option. Do the same for #2, and #3 up to #5. The fuse configuration settings will remain in memory. They can only be overwritten next time you enter the fuse configuration menu. The only way to exit this menu is to press #6, you will then be returned back to the configuration menu, where you can press ESC to get back to the main menu. Any of the fuse settings are dependant upon the device selected. (i.e. If the device selected is an AT89S51 that does not contain any fuses, the fuse settings will be ignored.)

FROM THE MAIN MENU Select 5, Program.

The maximum filesize that will be programmed is determined by the device selected. (i.e. if a 12K file has been loaded, and the device selected is a 2K device, ONLY 2K will be programmed)

IT IS EXTREMELY IMPORTANT TO SELECT THE CORRECT DEVICE TO PREVENT ERRORS!

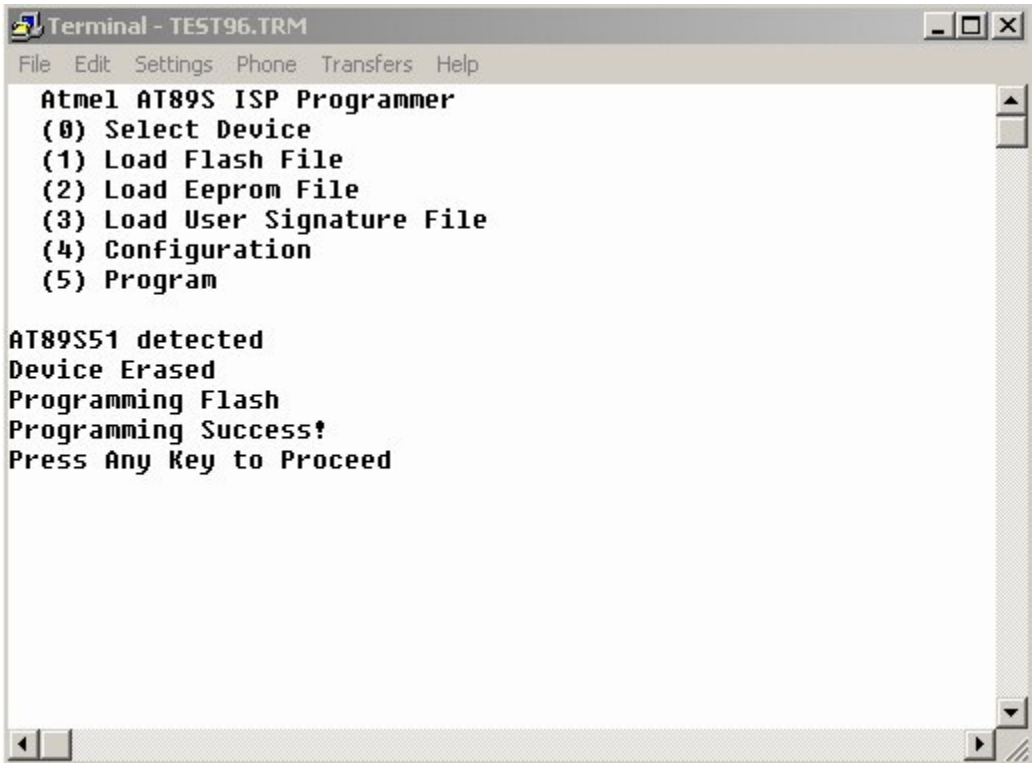
The target can be programmed while the programmer is connected to the PC if you would like to see the results in the terminal, or you can just view the LED's when in stand alone mode. If the programmer is connected to the PC during programming, you should see a screen like this:



The programmer will detect the target device and being to program based on the configuration settings. The programming address will be displayed while programming.

*****If the terminal program you are using does not support CR (carriage return) or LF (line feed) chars, then the display will not appear correctly. This is why we use the terminal.exe program from the above link.*****

Once programming has completed, the green led will be illuminated if there are no errors. You will then see a screen like this:



The results will also show on the led's.

The LED's represent:

Red (L3) - ERROR, or programming failed.

Yellow (L2) - Busy Programming or receiving DATA.

Green (L1) - Success Programming or receiving DATA

Press any key to return to the main menu. After loading the hex file and configuration, the unit can run as a stand alone programmer. When programmer is powered down, install jumper J1 (see Figure 3) and connect programmer to target device.

Figure 3.

With Jumper J1 installed, the programmer will operate as a standalone unit. When power is applied to programmer, it will begin to program the target based on the settings you have set in the configuration menu.

Figure 4.

Jumper J1 must NOT be installed for loading and configuration.

The programmer automatically verifies every byte it programs. You can choose the option to lock the target for extra security.

If you experience any errors while programming simply retry the operation. Success will be dependent on the efficiency of your target's power supply with respect to noise.

For support or any questions, please email: microcontrollerprog@yahoo.com