

# Flash Programmer User's Guide

## Macraigor Systems LLC

This application lets you program your Flash EEPROM devices via the On-Chip Debug connector (BDM or JTAG) on your target system using a Macraigor hardware device. Download the demo version which will allow you to test the ERASE, FILL, BLANK CHECK, CHECKSUM, UPLOAD flash content, and VIEW MEMORY functions of the software with your flash devices. Purchasing this license will turn the DEMO into a full working application that will allow you to program and verify your flash device using Motorola(.s19), Intel(.hex), and ELF(.elf) format files in addition to the functionality provided in the demo. The demo version will provide you with a COMPUTER ID which you MUST have when you purchase a license to use with a Raven or Wiggler. Licensing for use with usb2Sprite, usb2Demon, and mpDemon is done outside of the Flash Programmer.

The Flash Programmer is supported under Windows 98/ME, NT, 2000, 2003, XP, and Vista32 operating systems, as well as under Red Hat Linux versions 7.2, 9.0, Fedora Core 2 - 6(2.6.20). The software may work with other operating systems and versions but has not been verified against such and is not guaranteed to work. Only our mpDemon, Usb2Demon/Sprite and UsbWiggler are supported under Linux. On Vista-32, only the mpDemon and the Usb2Demon/Sprite and UsbWiggler are supported.

## Quick Start:

Attach the interface (Wiggler, usb2Demon, whatever) to the target board and power up the board. Got to **Configuration -> Communications** and make sure you have the correct choices.

Go to **File -> Open ocd file ...** and find the file for your target board. If there is not one for your specific board, you may have to make one. You should start with the closest you can find.

If there correct OCD file exists, choose it.

You should be all set to go. Try hitting the Flash ID button and see if you get valid values. If so, you can try erasing, blank check, and program!

If you are not able to communicate with the target, check your connections, the communications dialog, and power. If you still cannot communicate, leave the application and, if you are using a JTAG based target (i.e.: not BDM) try to run the JTAG Scan Chain Analyzer and see what it says. It should see your target board.

First a quick overview of the menu options.

The **File** menu item contains two choices:

- Open .ocd file ...**
- Save .ocd file ...**

An OCD file is a configuration file that describes the target board. Macraigor Systems supplies many .ocd file. A good number of these are for factory evaluation boards. If a .ocd file does not exist for your board, you will need to generate one. The best way to do that is to start with a .ocd file that is closest to your board, minimally the same CPU.

The **Configuration** menu item contains four items:

- Communications**
- Prefix Hex Numbers with ...**
- Default Program File Type ...**
- Target Specific**

The **Communications** option will bring up a dialog box for you to enter information about how you are connected to the target. If you are having problems connecting, it is always a good idea to select a lower Debug Port Clock Rate to see if that solves the problem.

**Prefix Hex Numbers with ...** is an option that allows you to choose "0x" or "\$" as the preferred prefix for when you specify a number has a hex string.

**Default Program File Type ...** this option asks you to specify what format your object code is in (i.e.: Motorola S records, Intel Hex records or ELF format).

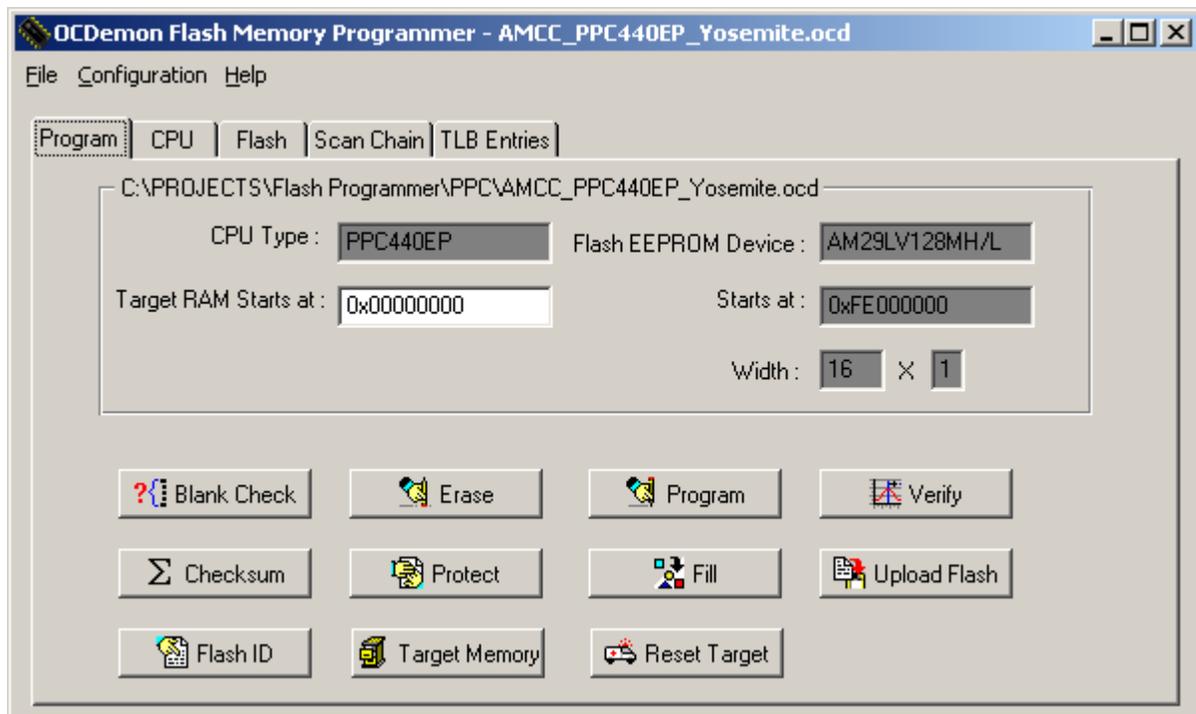
**Target Specific ...** Various CPUs have very CPU specific settings and options available and this is where you will find them. It is important to look at this menu item anytime you start to work with a new target board to see what is available.

The **Help** menu item has two choices:

- About ...**
- Contents ...**

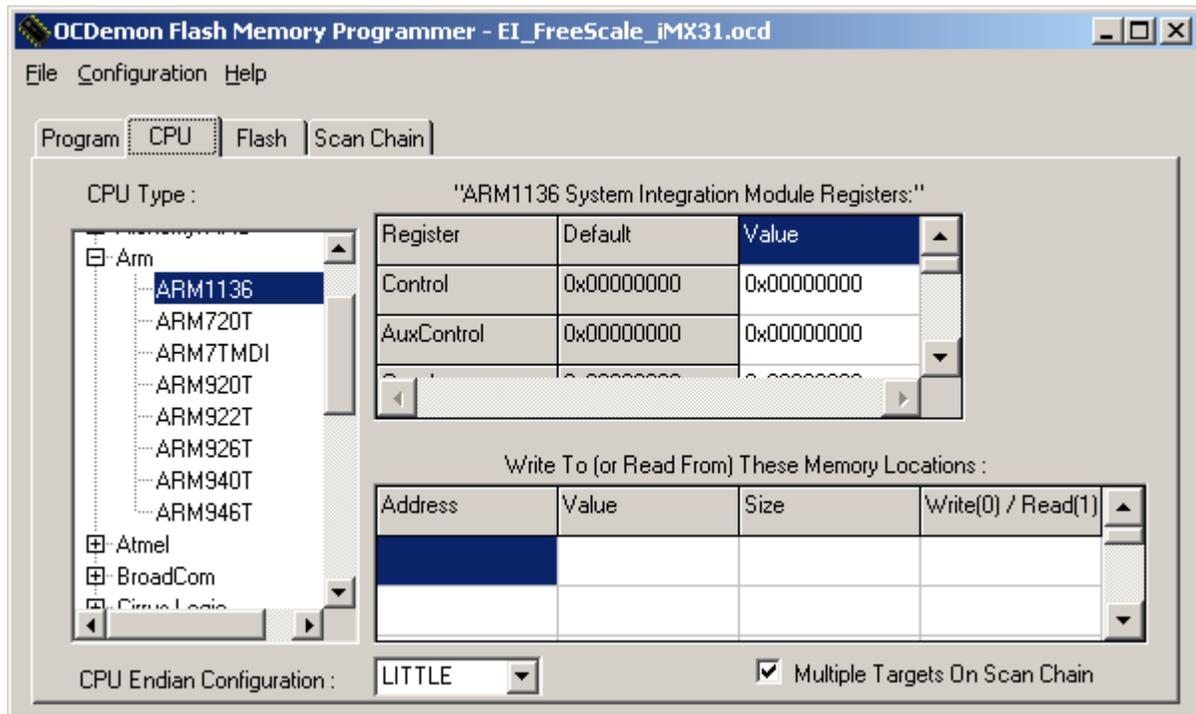
**About** will show the version number of the application.

**Contents** will bring up the help screens.



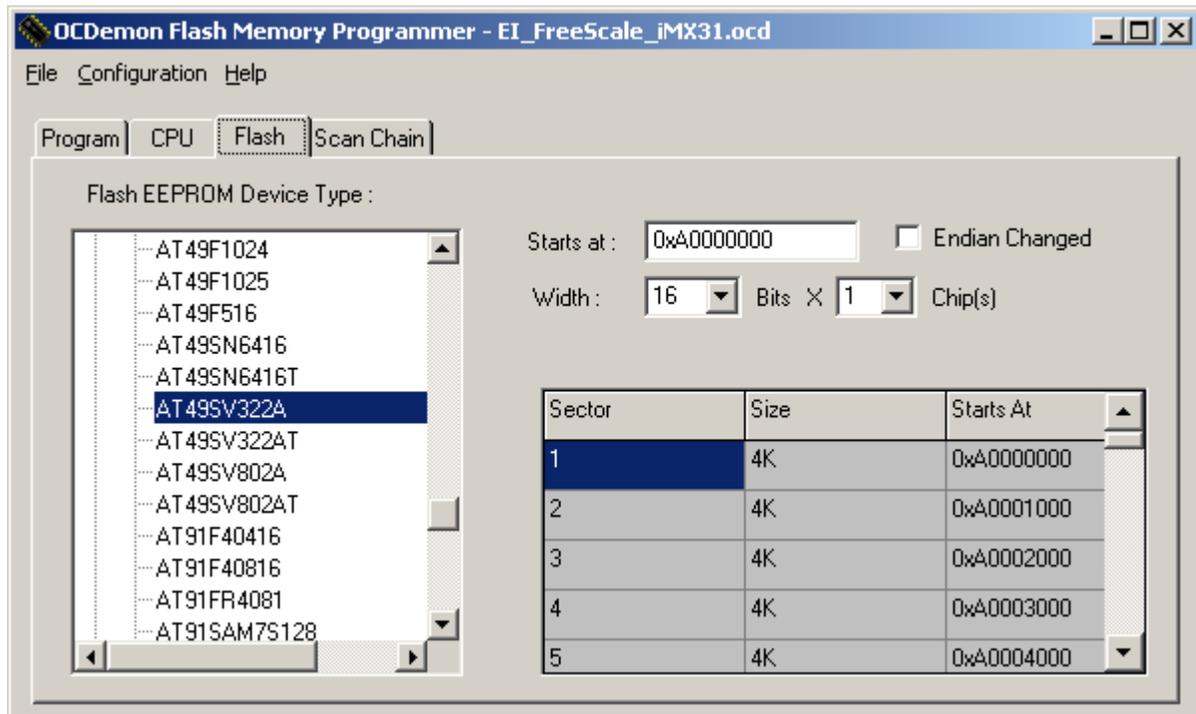
The main window of the Flash Programmer will have UP TO five tabbed pages depending on the target CPU and the configuration of the target board. Additionally, some buttons may be 'grayed out' and not usable, and this depends upon the target CPU, target flash memory devices and whether or not the Flash Programmer is in demo mode.

Most of the values on the **Program** tab are actually specified on other tabbed pages. The one important value to specify here is **Target RAM Starts at**: For the vast majority of target CPUs the Flash Programmer uses RAM for the programming. You must specify an address in RAM that has at least 2K available. Any registers or CPU resources that must be set up for the RAM to be available (chip selects, DRAM parameters, etc.) will be specified on the **CPU** tab.



The Flash Programmer needs to know how to set up the target system upon a hard reset. This information is usually found in the first couple of hundred lines of boot source code for the board. It may include, and not be limited to, how to shut the watchdog timer, setting up chip selects for flash memory and RAM, DRAM or SRAM control parameters, etc.

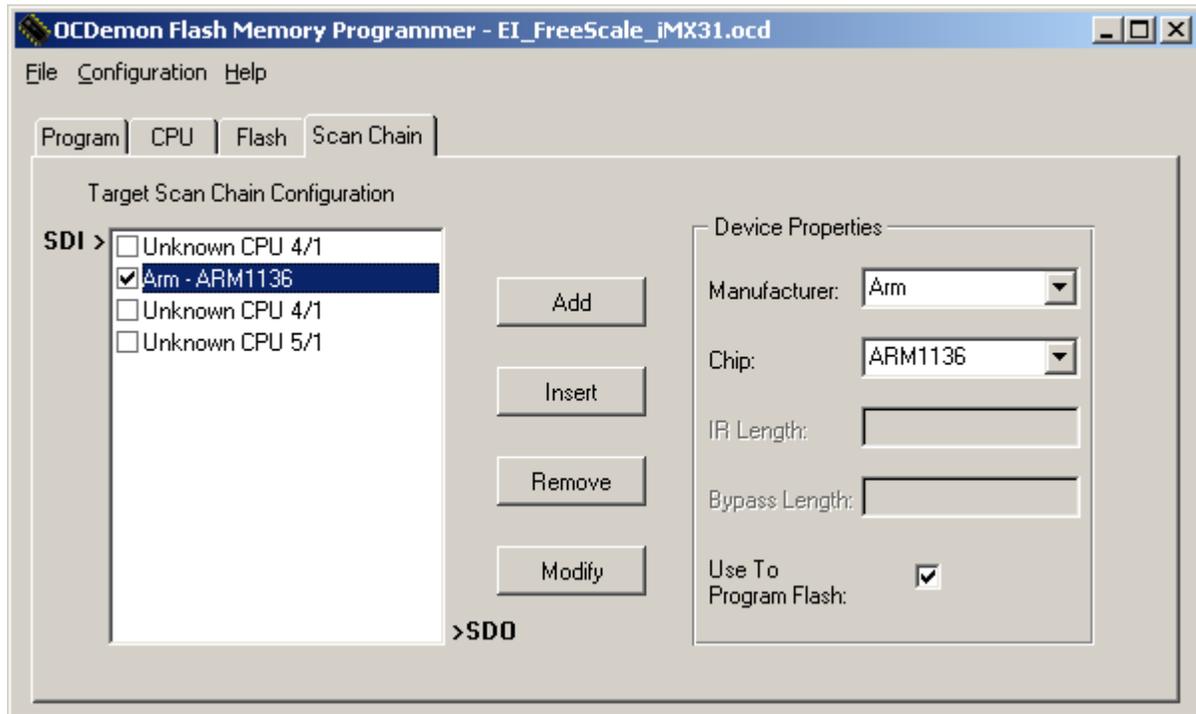
First choose your CPU from the tree structure in the CPU Type window. Second, fill in any new values necessary in the **System Integration Module Register** grid. Third, add any memory locations that need to be read or written in the **Write To (or Read From) These Memory Location:** grid. If your target processor has an option for making it Big or Little Endian, then the **CPU Endian Configuration** drop down will be available. Make sure to set it to the same value as your hardware. Finally, if the CPU is not the only device on the JTAG scan chain, check the **Multiple Targets On Scan Chain** box. Note that if there are multiple devices on the scan chain inside the CPU chip itself, you must check this box, as on the iMX31 for example.



The Flash tab is where you specify the target flash device(s). Using the tree on the left, choose the flash chip that is on your target board.

Next, specify the starting address of the flash memory in the **Starts at:** box. You then specify the physical arrangement of the flash chip(s) in the **Width** and **X Chip(s)** boxes. The example above shows a single flash memory chip that is 16 bits wide.

The **Endian Changed** box is to be checked if the flash memory chip is attached to the address bus in such a fashion that the flash memories native endian is modified. In other words, if you have a 16 bit flash device and address bit A0 of the flash chip is attached to the lower order address line of the processor, then DO NOT check this box. If, for instance, flash memory chip address line A0 is attached to CPU address line A8 and flash memory chip address line A8 is attached to CPU address line A0, then the native endian of the flash chip is being changed and you would check the **Endian Changed** box.

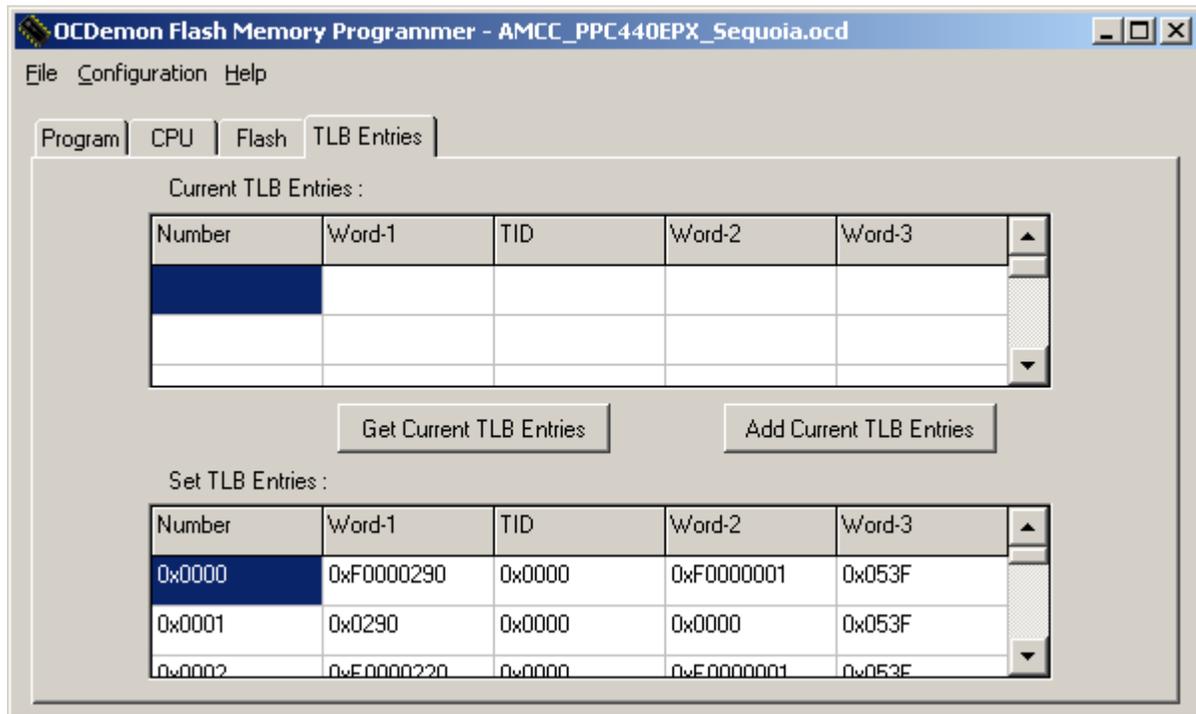


If the **Multiple Targets On Scan Chain** box on the **CPU** tab is checked, this tabbed page will be available. This is where you must describe the JTAG scan chain that exists on the target board.

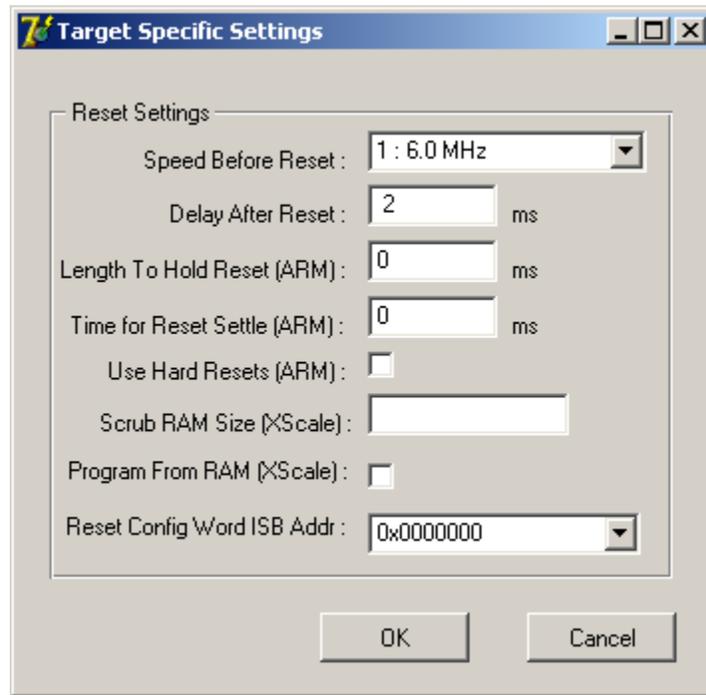
The order of devices must be the same as they are on your board, the first device in this list is the first device on the scan chain, that is the first device to get data that comes out of the JTAG header to the target board, the chip that is directly connected to TDI on the JTAG header. The last device in the list is the device that directly sends data to the JTAG header via TDO.

If the device is not a known device (not available in the **Manufacturer** and **Chip** dropdowns in the **Device Properties** area), then choose "Unknown" and specify the **IR Length** of the JTAG state machine for that chip and the **Bypass Length** (almost always '1').

Make sure to choose the device that drives the flash memory you want to program and check off the box **Use To Program Flash** (as in the above example for the iMX31).



If the processor chosen has a TLB the TLB Entries tab will be available. Any entries that are necessary for the CPU to access the Flash Memory and RAM must be entered here.



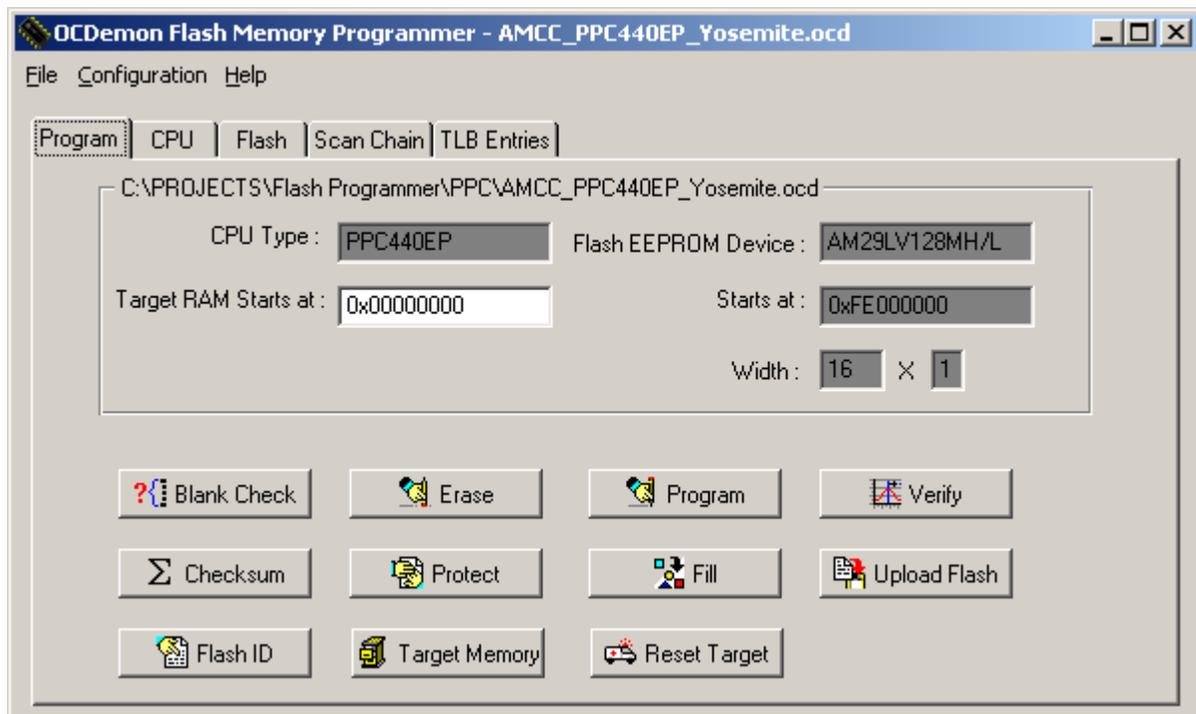
Under the main menu, **Configuration -> Target Specific** brings up this dialog. Only some of the options will be available, depending on the target CPU.

If the target CPU comes out of reset at a slow speed, you must set the JTAG speed in the **Speed Before Reset** selection. Setting on the **CPU** tab page will then increase the speed, usually by setting a PLL or something, so that the Flash Programmer will be more efficient. The **Delay After Reset** is how long the Flash Programmer should wait, in milliseconds, after removing the reset signal and before doing any actions specified on the **CPU** tab page.

Some ARM implementations, both on chip and on the target board, are a bit more complicated when it comes to reset. **Time To Hold Reset (ARM)** is how long, in milliseconds, to hold the reset line active. **Time for Reset Settle (ARM)** is how long after releasing reset before sending any JTAG commands to the CPU. Additionally, some implementations do not have the ARM CPU hardware reset line attached to the JTAG header. The Flash Programmer defaults to a software reset. If you want to force hardware resets to occur, check the **Use Hard Resets (ARM)** box.

Improved flash programming times will be seen with the XScale family of processors if you use an external RAM buffer. You can enable this by checking the **Program From RAM (XScale)** box. Additionally, some XScale boards need the external RAM scrubbed, please specify, in Bytes, the size of this RAM in the **Scrub RAM Size (XScale)**.

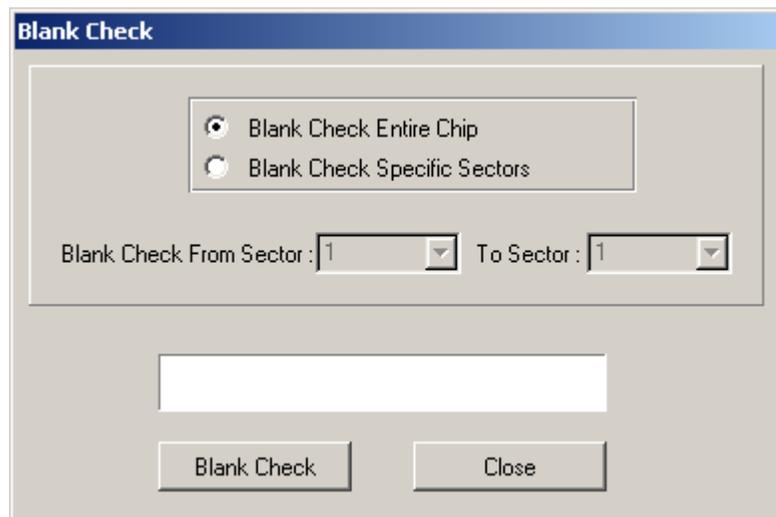
The **Reset Config Word ISB Addr**: is the Internal Space Base defined for the Freescale 82xx family of processors.



Back to the Program tab ...

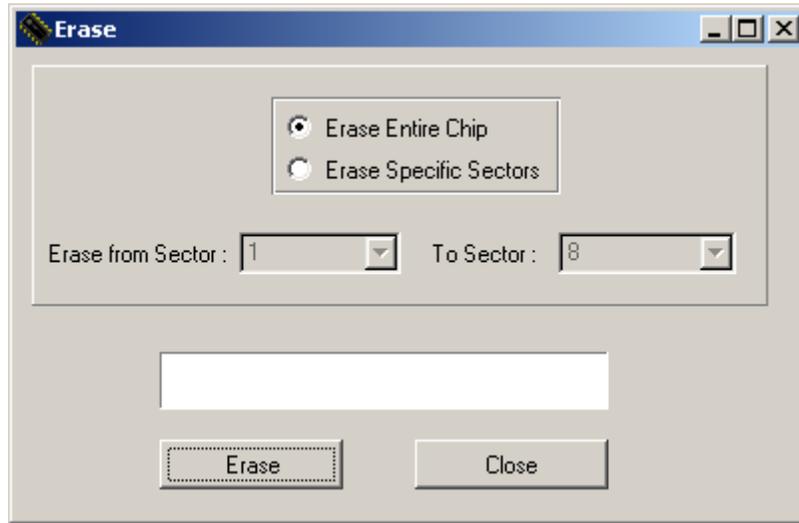
The button names are self-explanatory. Most will bring up a dialog box for you to supply some additional information.

### Blank Check



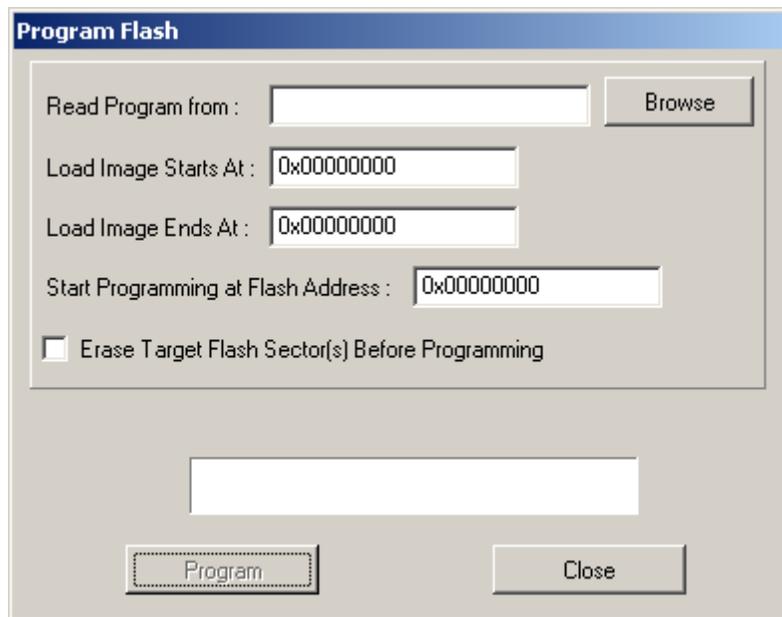
You can specify either the entire chip or just specified sectors to be checked.

## Erase



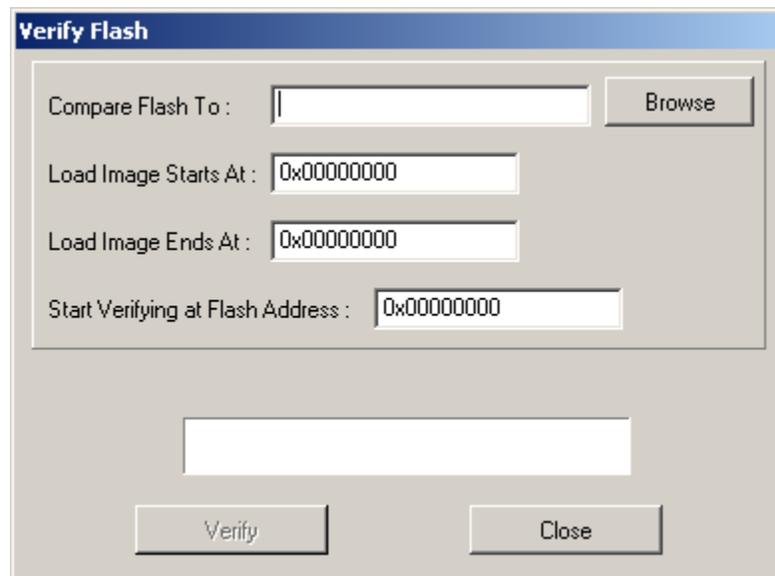
You can specify if you want the entire chip erased or just specific sectors.

## Program



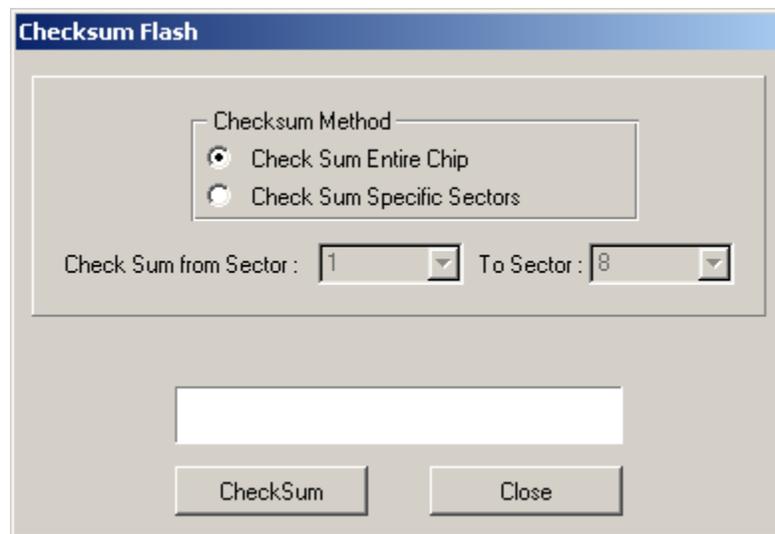
**Read Program from:** is where you specify the object code file to be programmed into the flash memory. You can use the Browse button to find the file. Note that the Flash Programmer requires files be in Intel Hex, Motorola S-record or ELF format. The Flash Programmer will show you where it believes the image starts and ends. You can specify where to **Start Programming at Flash Address:** which will then offset the image to the appropriate memory space.

## Verify



**Compare Flash To:** is where you specify the object code file to be compared to the flash memory. You can use the Browse button to find the file. Note that the Flash Programmer requires files be in Intel Hex, Motorola S-record or ELF format. The Flash Programmer will show you where it believes the image starts and ends. You can specify where to **Start Verifying at Flash Address:** which will then offset the image to the appropriate memory space.

## Checksum



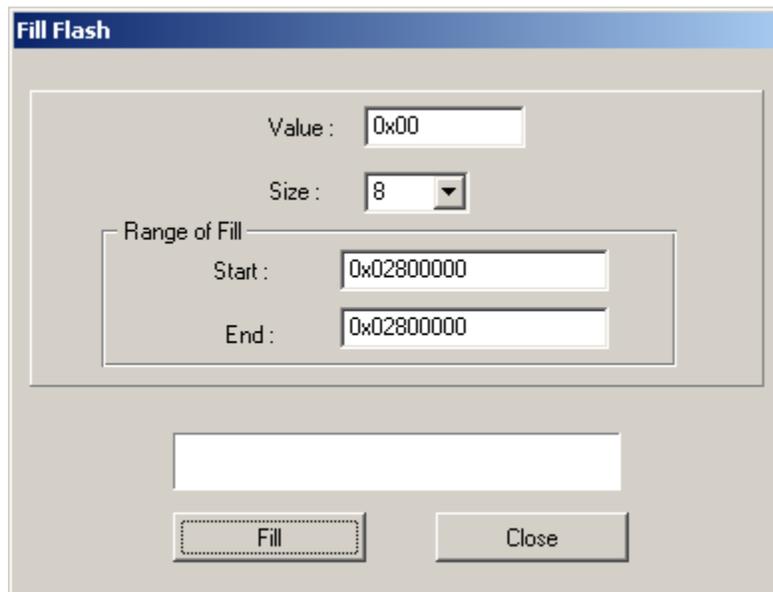
You can specify if you want the entire chip checksummed or just specific sectors.

## Protect



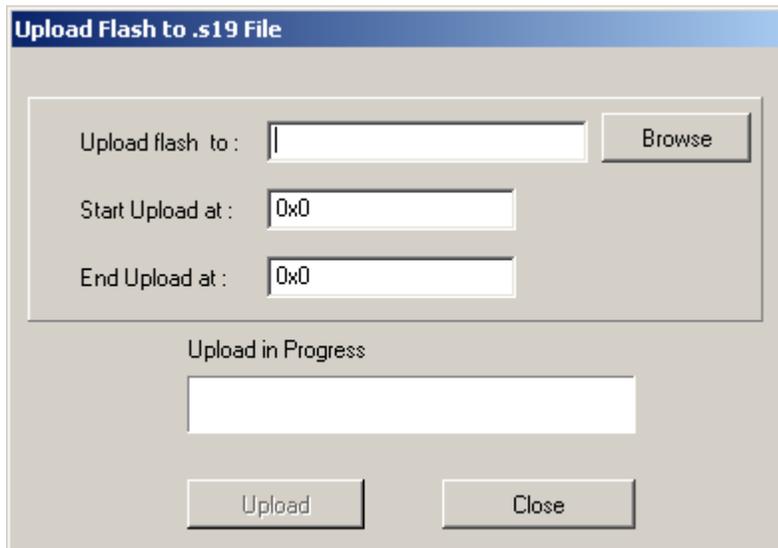
You can see and optionally (depending on flash chip(s) and your hardware) protect the entire chip or various sectors.

## Fill



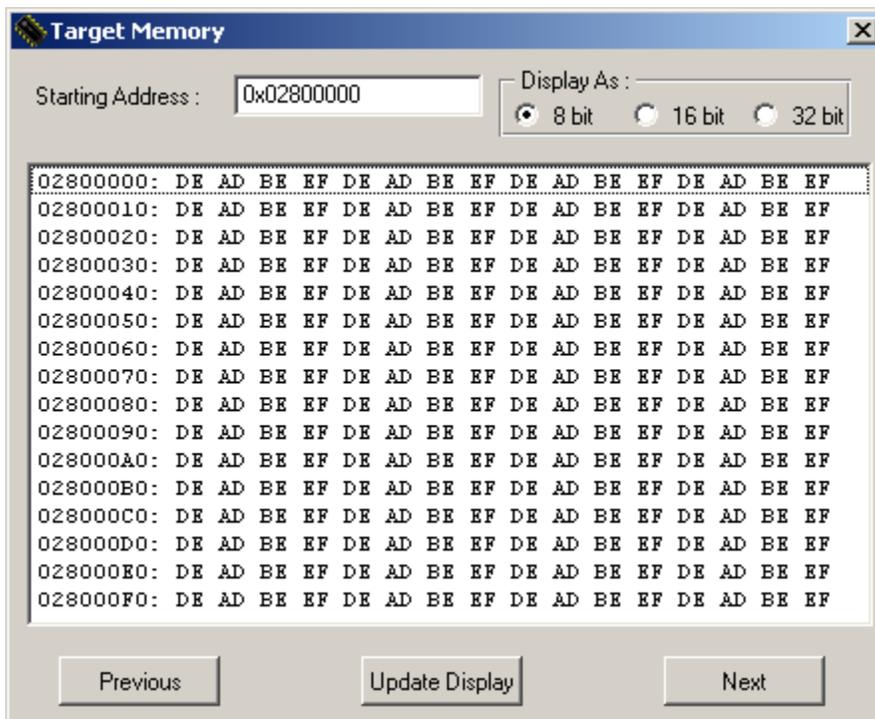
You can specify a **Value** and its **Size** in bits to use to fill a range of addresses.

## Upload Flash



You may specify a filename, start address and end address for the program to read and convert into a Motorola S-record formatted file.

## Target Memory



## Reset Target

Simply resets the target CPU. No dialog is associated with this button.