TERN INC. 1724 Picasso Avenue, Suite A, Davis CA 95616-0547 USA
Tel: 530-758-0180 Fax: 530-758-0181
Email: sales@tern.com http://www.tern.com

Flash Programming Guide

for the optional on-board 16-bit Flash(Am29F400) on the IE-M, IE-P, and ID

Overview

On the TERN i386-Engine-P (IE-P), i386-Engine-M (IM), and i386-Drive (ID), a optional 16-bit Flash can be installed for non-volatile storage of completed applications. In past documentation, this was referred to as 'Step Three' of the TERN development process.

This Guide explains the process of compiling a completed application into a .HEX file, and then download this .HEX file into the on-board Flash for automatic power-up execution. The flash chip (U15) is a surface mounted, 16-bit 256KW blank flash (Am29F400BT).

Before you start following this Guide, you should already be able to compile and download your application into the battery-backed SRAM from within the debugger. You should also be able to successfully run your application from the battery backed SRAM in stand-alone mode (Step Two of the development process).

Minimum Requirements

TERN Paradigm C++ Development Kit (DV-P Kit)
i386-Engine-P, i386-Engine-M, i386-Drive with the 16-bit Flash (U15, Am29F400).
A debug ROM (IE16_115, or IE8_115) should be installed in the 32-pin DIP socket.

Memory Mapping

Memory for the 16-bit Flash configuration is shown in figure 1. The Debug ROM is located at the top of the memory map and is the first block to execute after power-on/reset.

Flash memory is mapped starting at address 0x80000.

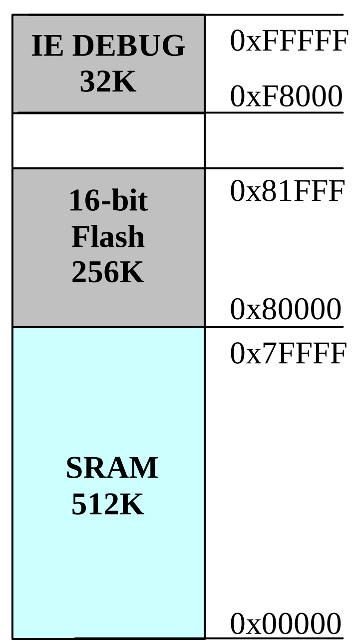

Figure 1: Memory mapping configuration

Generating a HEX File

For this procedure, refer to the sample project *\tern\386\rom\flash_ie16.ide* as guide for the correct final configuration.

1) For your target application (*led_iep* in the sample), you must first change the configuration file from the one used during debugging. This allows you to generate a .HEX file as output, as well as relocating the file to the appropriate

TERN INC. 1724 Picasso Avenue, Suite A, Davis CA 95616-0547 USA
Tel: 530-758-0180 Fax: 530-758-0181
Email: sales@tern.com http://www.tern.com

memory addresses for Flash.

Change the configuration node from 386.cfg to actf386.cfg (you can right-click on the *.cfg* node, and then choose 'Edit Node Attributes') to point to a copy of: *\tern\386\config\actf386.cfg.*

2) In this new configuration file, make sure the correct options are selected for your board. Double-check the BOARD type, as well as the Flash size (should be 512).

3) Right-click on the *.axe* node (*led_iep.axe*) and select 'Target Expert'. Change the 'Target Connection' option to: '**No Target/ROM**'.

4) Right-click on the *.axe* node again, and choose 'Build Target'. A .HEX file named after your target will be created in your working (or output) directory (*led_iep.hex*, for this sample).

Downloading a HEX file into the 16-bit Flash

*NOTE: Be sure that the 'Step 2' address is setup correctly to 0x08000. If you are not sure, run **step2.c** in the debugger for your controller. A **step2** target is made available for you in* flash_ie16.ide*; just download and run it.*

The downloading process requires an intermediate loading program, ***l_f16.c***, to prepare the 16-bit Flash, and to receive the final HEX file. This file is located in `C:\TERN\386\ROM\`.

Download the *l_f16.axe* application into your controller using the debugger. After the debugger has downloaded the program, terminate the debug session (*Debug->Terminate Debug Session*) immediately without running; *l_f16* tries to use the serial port 0, and will crash your debugger.

Start a terminal program (either H*yperterminal*, or *Tools->RTLOAD* within the Paradigm environment), and configure it for 19200 baud operation. Place the red "Step 2 jumper" on the board (refer to your controller manual if you're not sure where this is), and then reset the controller with the STEP2 jumper installed.

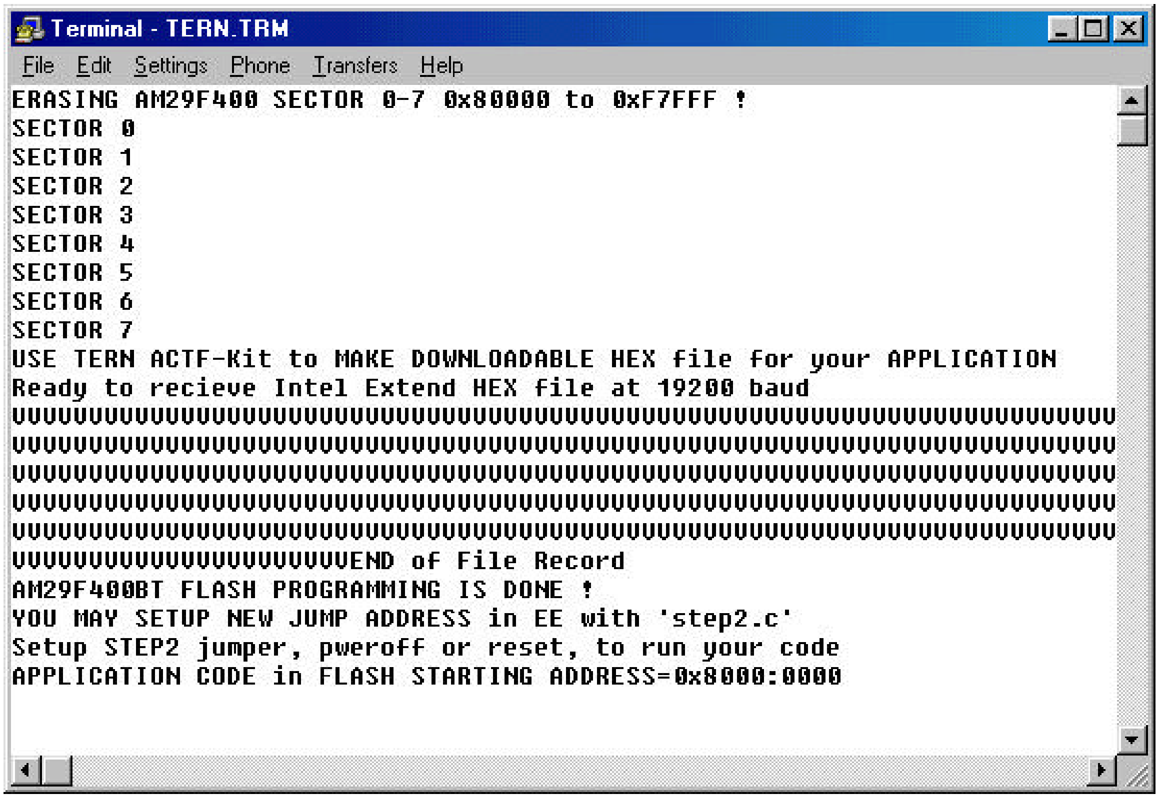

Figure 2: Sample session with *l_f16*

l_f16 erases the onboard Flash, and prepares it for receiving your downloaded application. After you see the message:

'**`Ready to receive Intel Extend HEX file at 19200 baud`**'

… the board is ready to receive the .HEX file you generated in the previous step.

Send the .HEX file over as a text file (within *Hyperterminal*, choose *Transfers->Send Text File*). You will see a series of **V** fill up your screen as the file is received and written to the onboard Flash. Upon completion, the step 2 jump address in the EEPROM will be rewritten to point at **0x80000**, the starting address of your application.

Now, each time you power up the controller with the step 2 jumper in place (and the DEBUG ROM in the socket), your application resident at 0x80000 will automatically begin executing. To start the debug kernel instead (in order to debug a new application), just remove the step 2 jumper. Remember, the jump address is now at 0x80000, and you will need to run ***step2.c*** again before running an application out of the battery-backed SRAM.