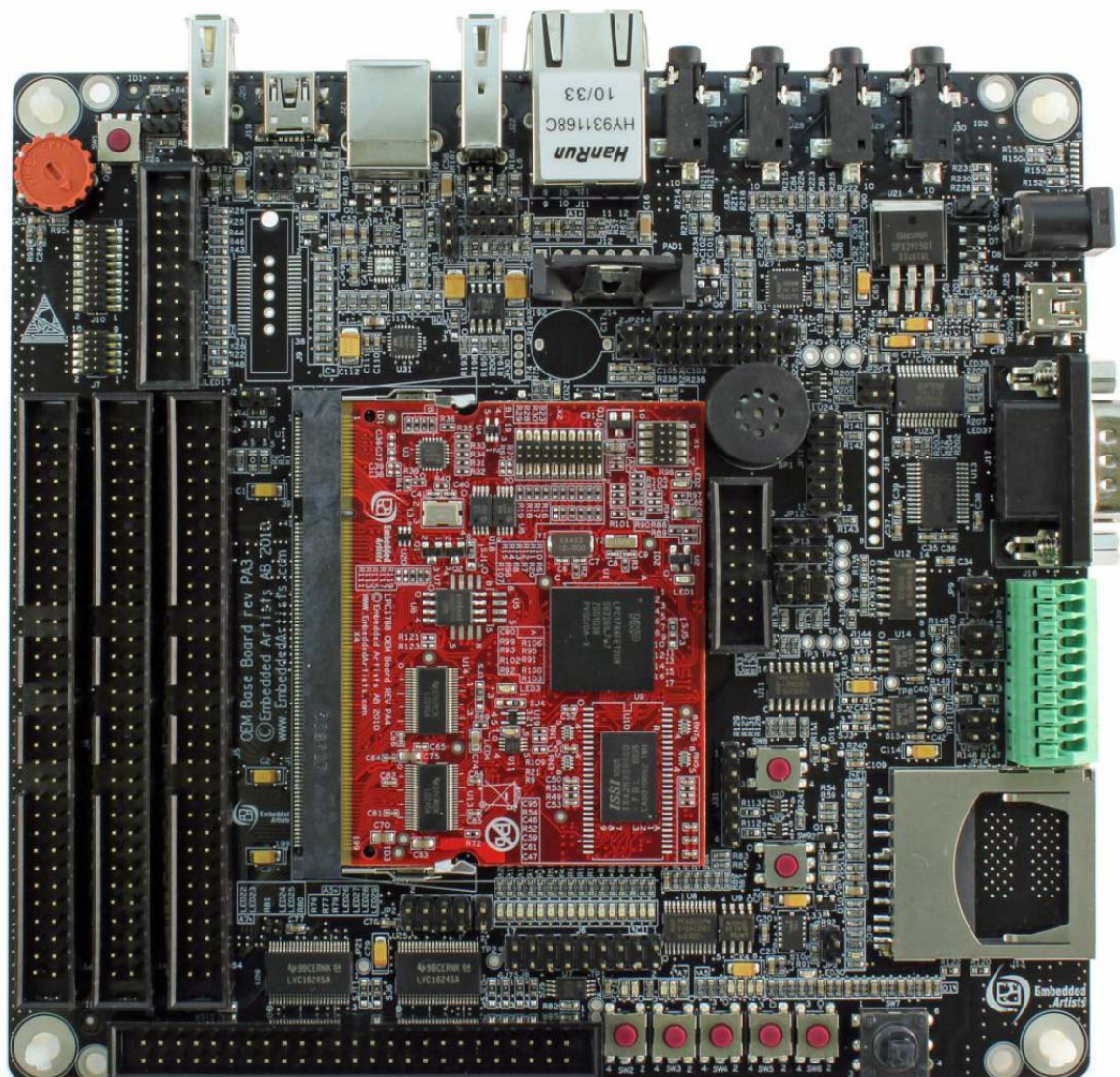


Getting Started with TargetOS on Embedded Artists' LPC1788 Developer's Kit



1 Introduction

This document covers how to get started with Blunk Microsystems' TargetOS embedded operating system on Embedded Artists' LPC1788 Developer's Kit. It covers the following steps:

- Installing the CrossStep IDE, TargetOS Basic, and the ea1788 demo package.
- Setting up a new ea1788 board: cables, jumpers and power.

- Installing Blunk's 1) First Stage Boot Loader and 2) target monitor (with debug-over-Ethernet support) into the LPC1788's internal Flash memory.
- Building Blunk's kbench kernel benchmark sample application and running it on the board.

2 IDE/RTOS installation

This topic is covered lightly here because a CrossStep wizard guides you through this during its installation. Mainly you just go to www.crossstep.com and download the CrossStep installer. Pick 'ea1788' when asked to select a board. When asked to pick an initial sample application, please pick 'kbench' to be in sync with the explanations below.

3 Board Setup

The ea1788 has a large number of jumpers, but initially only the following need to be checked or changed from the default position:

JP2 – With the Ethernet connection as the top of the board, JP2 is an eight-pin header near the bottom of the board, left of center. Jumper pins 3-4 to connect the NAND Busy output to a microcontroller input.

JP1 – A two-pin header located immediately adjacent to JP2, on its right edge. To enable the external NOR flash, connect pins 1 and 2.

JP15 - Located below USB mini-AB connector J19 in the top left corner of the board. Controls whether USB-A connector J20 supports USB device or host mode. To use Blunk's TargetUSB host mode USB stack, JP15 should not be jumpered.

With the board configured as above, perform the following steps:

- Connect an Ethernet cable between the network your PC is on and the ea1788's RJ45 Ethernet connector J11, located midline at the top of the board.
- Connect a USB cable between your PC and USB mini-B connector J25, located on the right side of the board, near the top right corner. This serves as a USB-UART connection and alternate power-supply.
- Optionally, to use TargetUSB, connect a USB flash drive to USB-A connector J20, located at the top left corner of the board.
- If your USB connection to USB mini-B connector J25 cannot supply enough power, you can connect an external 5v power supply spec'd at 1A via center-positive power jack J24.
- Power on the board to add its USB-UART channel to your PC. If the USB-UART (FTDI's FT232R) is not automatically recognized, download a driver installation guide from the link below. After the driver is installed, you should verify that your PC has a new USB UART channel.

www.ftdichip.com/Support/Documents/InstallGuides.htm

- Open a terminal program on this channel using 115200 baud rate, 1 stop bit, no parity, and 8-bit character length.

Before you can do anything useful with the board, you need to install two files into the LPC1788's internal Flash memory. That is the topic of the next section.

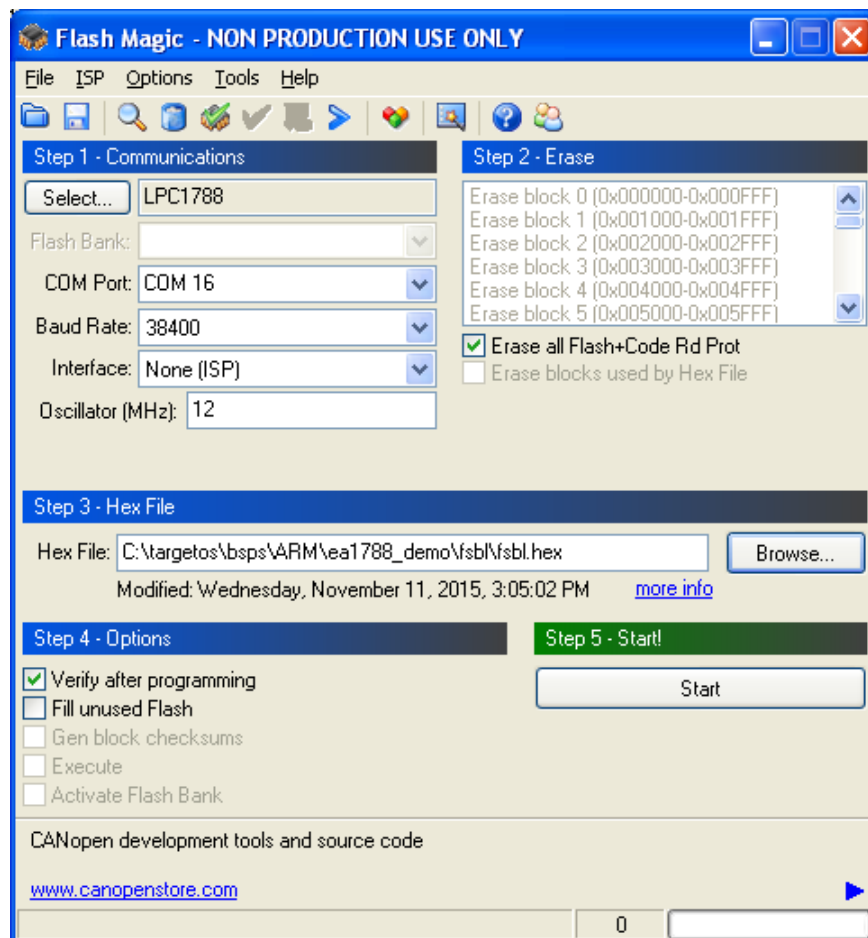
4 Preparing ea1788's Flash Memory

The First Stage Boot Loader (FSBL) is the first post-reset code to run after the LPC1788 on-chip boot loader. Its purpose is to initialize SDRAM and load an application into it. To support this, two files must be installed in the LPC788's internal flash memory: 1) the FSBL image, which is started by the on-chip boot loader and 2) application code.

It is possible to run the FSBL via a JTAG debug interface and have the FSBL copy itself to flash, but it is perhaps easier to install the FSBL using NXP's Flash Magic, which can be downloaded from this site:

www.flashmagictool.com

After installing Flash Magic, start it and set its front panel options to match the picture below:



The COM Port setting should be the appropriate channel for your PC and the path to 'fsbl.hex' depends on where you installed TargetOS Basic.

There is a row of five buttons at the bottom of ea1788. SW6 is the rightmost button, just left of the joystick. To get the LPC1788 into In-System Programming mode, press SW6 while pressing and then releasing the reset switch SW1 (just below SD socket).

Flash Magic uses the USB-UART channel while executing commands, so you need to disconnect the channel from your terminal program while giving Flash Magic commands. With this done, click on Flash Magic's 'Start' icon to program the FSBL into LPC1788 Flash memory.

The bottom of Flash Magic window shows status messages and should eventually say "Finished". You shouldn't have any problems, but if you do, Embedded Artists' LPC1788 Developer's Kit User's Guide and the Flash Magic manual have more information.

With the FSBL installed, reconnect your terminal program and press reset (SW1). You should see the following on your terminal program:

```
Boot application signature check failed
Cold start, init finished, press any key for menu
```

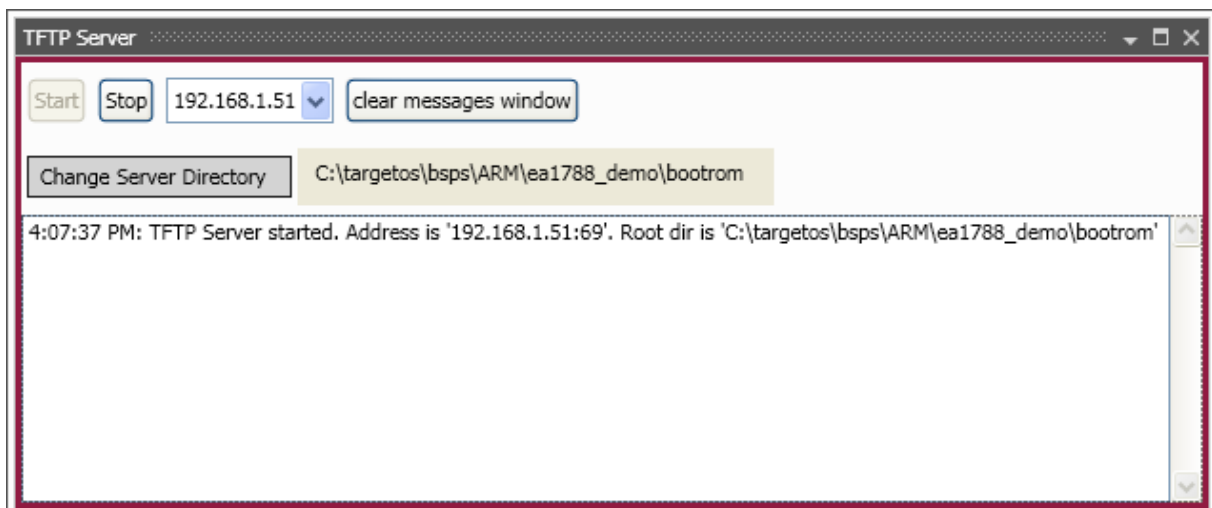
Press any key to get the following menu:

```
'1' - show boot ROM info and memory use <
'2' - edit IP addresses >
'3' - blank check boot app memory v
'4' - start the boot application
'5' - erase the boot application
'6' - install /dev/tftp/app.elf as boot app
'7' - load/start /dev/tftp/dbg.elf
'8' - perform SDRAM reliability test
```

The FSBL stays out of view as much as possible. If an application file is installed in flash, the FSBL silently starts it at each reset. But currently there is no application it can load and start. To install Blunk's bootrom application, which runs the TargetMon debug-over-Ethernet target monitor, start a TFTP server on your PC, serving the TargetOS subdirectory shown below:

```
bsps\arm\ea1788_demo\bootrom\app.elf
```

Using CrossStep's TFTP server (accessed via the Tools menu), this looks like the following:



In the terminal window for the FSBL menu, enter '2' to edit IP addresses and then '1' again to set the IP address of the TFTP server. Usually this would be the IP address of your PC.

Press '6' to install the bootrom 'app.elf' into the LPC1788's internal Flash memory. You should see something close to the following output:

```
Waiting for network interface to come up
ETH: link=up speed=100M duplex=full
ETH IP addr = 192.168.1.110
Using TFTP server at 192.168.1.51
Loading ELF file "app.elf"
.....
Clearing flash memory
.....
Programming boot image to flash
.....
Finished!
```

Now if you press the reset button, your terminal shows the bootrom application start, which should look similar to this:

```
TargetOS, Release 2015.1. Built at 11/11/2015 3:28:08 PM.
Copyright 2015, Blunk Microsystems. All rights reserved.
-----
Starting TargetMon
SDRAM size = 32MB
Tue Jan  6 01:25:24 1970
Hostname: ea1788
IPv4 Default gateway: 192.168.0.1
ETH IP addr/mask: DHCP
ETH MAC address: 00:00:00:17:88:00
Using: 0x00010020 - 0xA0100000 (cold boot)
-----
Press 'M' in 3 seconds to modify these settings: .Starting..
ETH: link=up speed=100M duplex=full
ETH: DHCP IPv4 address 192.168.1.110
```

You can return to the FSBL menu by holding the board's joystick in any position other than the neutral untouched position while pressing and releasing the reset button. This skips the application start. Do this and then enter '1' to view a summary of FSBL status, as below:

```
Part number: LPC1788
Boot code version: 8.1
Device serial #: 0x0B0B0C17 0x535624B5 0x4F03B4D0 0xF5000003
ETH IP addr/mask: DHCP
ETH MAC address: 00:00:00:17:88:00
Boot application free heap = 23280
```

Finally, to prepare for the next section, press the reset button again and leave the bootrom application running.

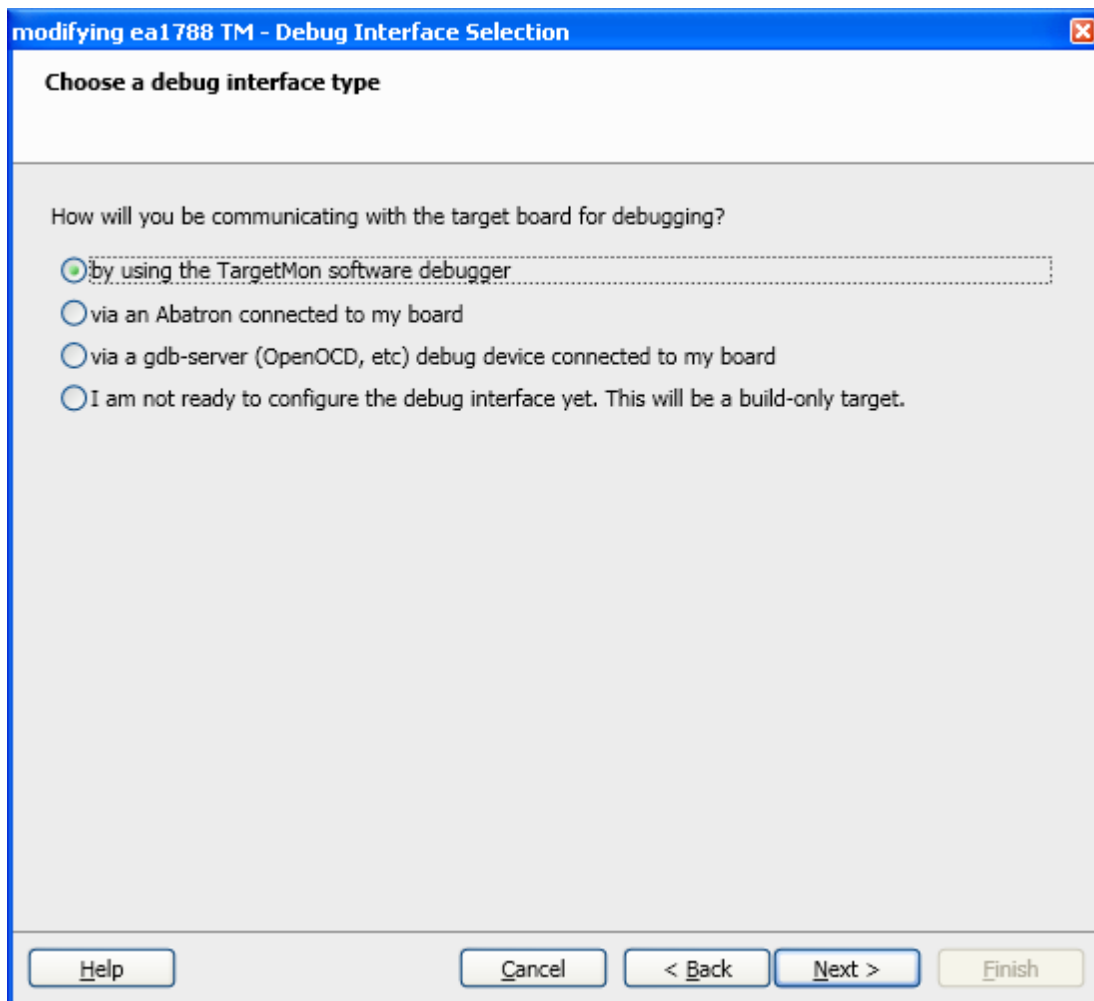
5 Running kbench

Now you are ready to build and run the kbench sample application. kbench measures TargetOS kernel performance. If you specified the kbench application during the CrossStep installation, it will already be loaded. Otherwise, use the Project/Open dialog to browse to and select this file:

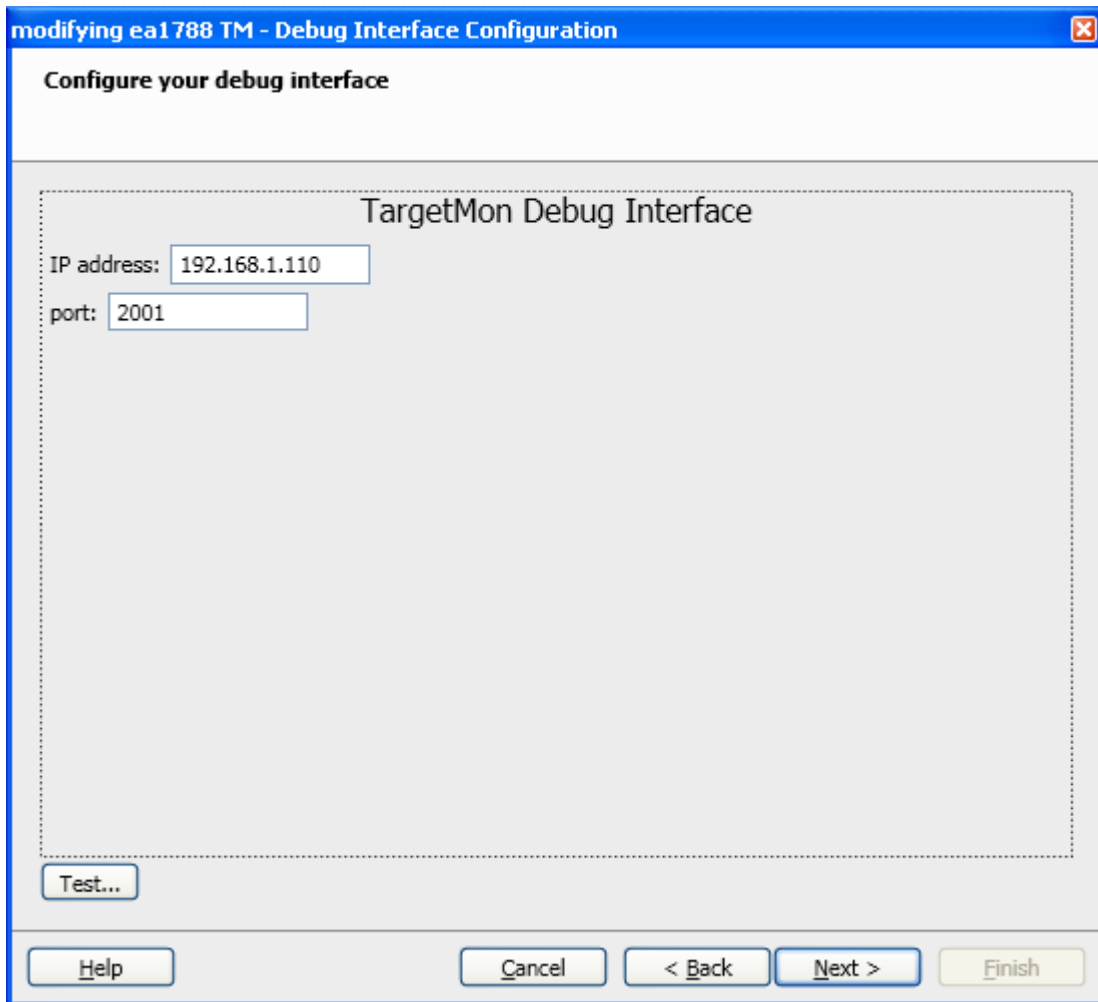
```
apps\kbench\kbench.gbp
```

To use a TargetMon debug connection, you need to modify your CrossStep target to use the IP address of your ea1788. In CrossStep's Options menu, select 'Manage Targets' to bring up the Targets dialog. Select the entry for the target name you created during installation, then right-click on it and select "modify.." in the menu that comes up.



Select "ea1788 – Embedded Artists LPC1788 Dev Kit (demo BSP)" as your target board, accept the GNU toolchain settings, and choose 'TargetMon' on the next the panel, as shown below.




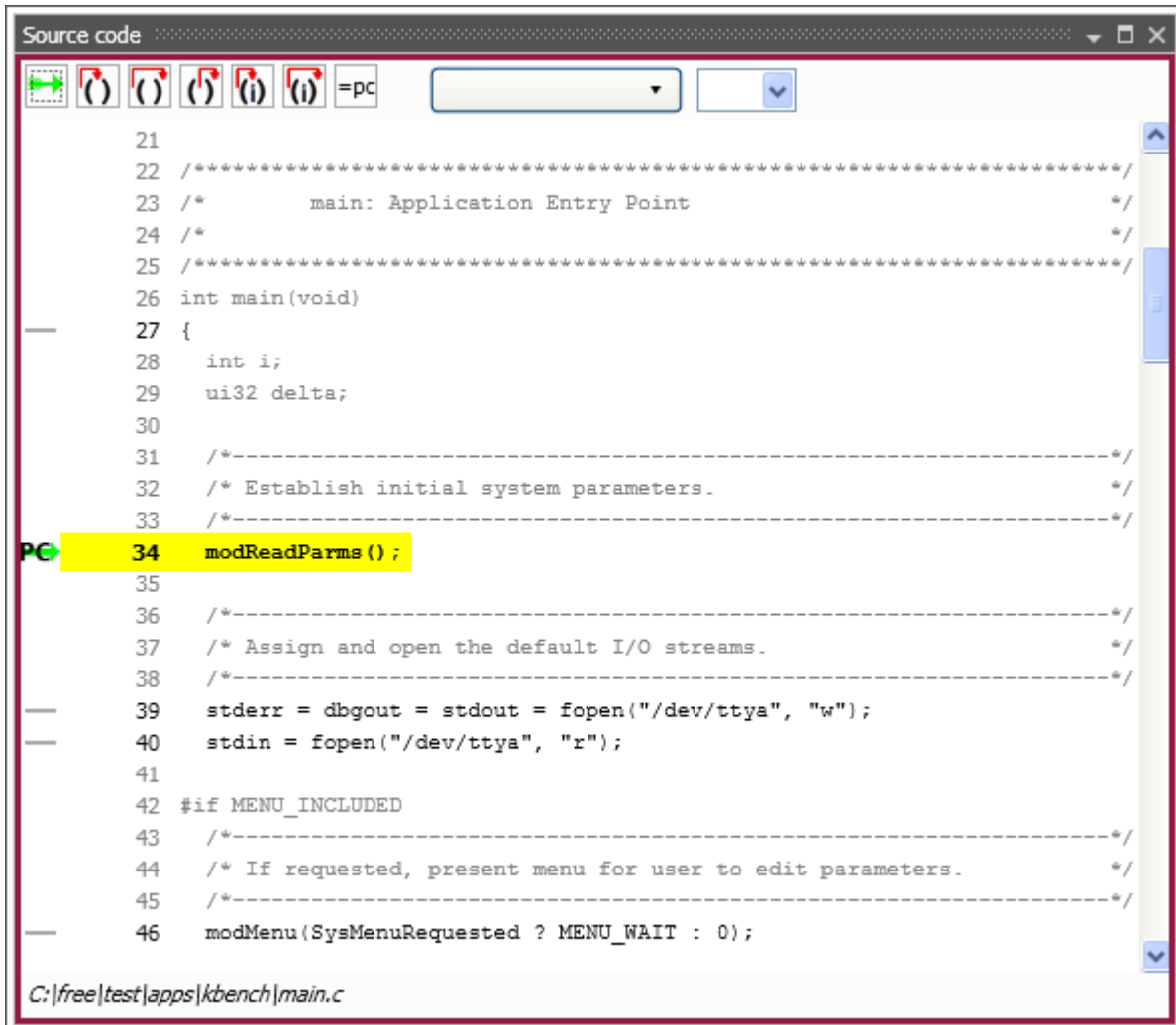
Enter your ea1788's IP address in the next panel, getting the address from the terminal window that shows the bootrom application's start-up. The IP address is on the last line. Leave the port number as 2001.



You can optionally test the TargetMon debug connection by clicking the “Test” button. Click past the following two dialogs and then click “Finish” to close the target modification dialog.

Build the sample application by clicking the button with the hammer icon  in CrossStep’s upper-left corner. Or you can use the keyboard shortcut control-B. When the build finishes, a “successfully built” message appears in the Output window. The Stop button  next to the Build button becomes active whenever a build is in progress. You can click that icon if you need to stop a build for any reason.

After the build completes, or instead of clicking the build button if you want to give a combined “build and launch debugger” command, either click on the “debug” button  or use keyboard shortcut control-D. This ensures the build is up-to-date, launches the source code debugger, downloads your application and (optionally) runs to its main() function.

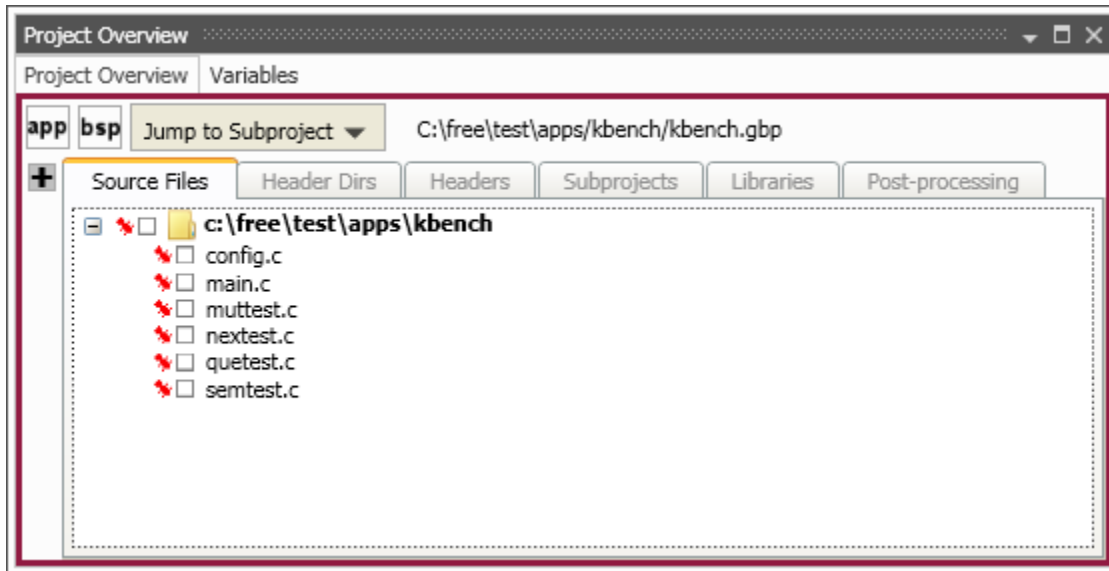


```
Source code
21
22 /*****
23 /*      main: Application Entry Point      */
24 /*                                          */
25 /*****
26 int main(void)
27 {
28     int i;
29     ui32 delta;
30
31 /*-----*/
32 /* Establish initial system parameters. */
33 /*-----*/
PC 34 modReadParms ();
35
36 /*-----*/
37 /* Assign and open the default I/O streams. */
38 /*-----*/
39 stderr = dbgout = stdout = fopen("/dev/ttya", "w");
40 stdin = fopen("/dev/ttya", "r");
41
42 #if MENU_INCLUDED
43 /*-----*/
44 /* If requested, present menu for user to edit parameters. */
45 /*-----*/
46 modMenu(SysMenuRequested ? MENU_WAIT : 0);

C:\free\test\apps\kbench\main.c
```

You can use the debugger's step-in, step-over, step-out, etc. commands and breakpoint facility to peruse the sample application. The user interface is fairly intuitive.

Clicking the **Source Files** tab in the Project Overview window displays the application's source files. Double-click on any source file to launch an editor on the selected file. Include files listed in the **Headers** tab can be opened similarly.





In the Options menu, select 'General Options' to specify which editor CrossStep uses to open source files:



Enter the path to your preferred editor on the "executable" line. \$(EDITLINE) and \$(EDITFILE) on the "arguments" line are replaced with the desired line number and file name, respectively,

when CrossStep launches your editor to - for example - open a source file at the line of a compile error. You must use these macros in the expected format for your editor. Click on “Editor Command Lines” to select among prefilled argument strings for several popular editors.

When finished with the debugger, click on the “End debug session” button  to close it. If you make changes to the code and save them, you can use the “debug” button  to rebuild the project, download it, and launch the debugger again.

6 Sample Applications

You can experiment with other sample applications in the demo package. The following sample applications are provided, which demonstrate various TargetOS middleware components. Each application contains a “readme.txt” with more details on the application.

bsearch - Loops over all installed file system volumes, measuring the number of binary searches performed in a set time interval.

clients – Cycles through this list of network clients: TCP echo, TCP discard, TCP chargen, UDP echo, and UDP chargen. The server IP address is specified via the TargetOS boot menu.

file_test - Loops over all installed TargetOS file systems, performing file system regression tests that primarily focus on the POSIX API.

graphics – Demonstrates TargetGraphics, Blunk’s embedded graphics drawing library. Only supported on targets with a frame buffer based display.

kbench - Measures kernel operations, including context switches in various conditions. The worst case value after multiple measurements is printed to stdout.

server - Runs the following servers: TCP echo, TCP discard, TCP chargen, UDP echo, and UDP discard. The server IP address is specified via the TargetOS boot menu.

shell - Launches the TargetOS command line interface. This is a shell that can be extended with user commands and accessed via UART connections and Telnet.

ui_demo – Demonstrates TargetOS’s support for touch panels, button, and joy sticks. Only supported on targets with a frame buffer based display and UI elements.

webserv - Demonstrates TargetWeb, Blunk’s embedded web server, both sending data to the target using browser forms and getting data from the target using custom pages that contain target data. Includes sample HTML pages.

Blunk Microsystems, LLC
Tel: 408/323-1758
Web: www.blunkmicro.com
Email: support@blunkmicro.com