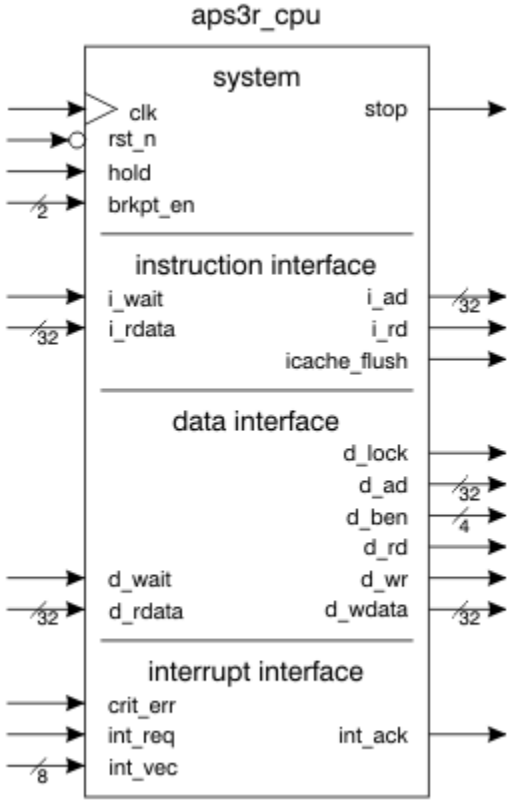
Getting Started with TargetOS on Cortus's APS3R Simulator



1 Introduction

This document covers how to get started with Blunk Microsystems' TargetOS embedded operating system on the APS3R Instruction Set Simulator (ISS) built into Cortus's GDB port. It covers the following steps:

- Installing the CrossStep IDE, TargetOS Basic RTOS, and sim_aps3r demo package.
- Installing the Cortus GNU toolchain.
- Building Blunk's kernel benchmark sample application and running it on the simulator.
- Connecting to the simulated UART channel using a Telnet client.

2 IDE/RTOS installation

This topic is covered lightly here because a CrossStep wizard guides you through this during its installation. To start, download CrossStep from www.crossstep.com and run its installer. Pick "Cortus Instruction Set Simulator" when asked to choose a target. When asked to pick an initial sample application, pick 'kbench' to be in sync with the explanations below.

The installation wizard also installs the Cortus GNU toolchain, if not already installed on your PC. You will have to click on a dialog box to confirm that you accept Cortus's licensing terms.

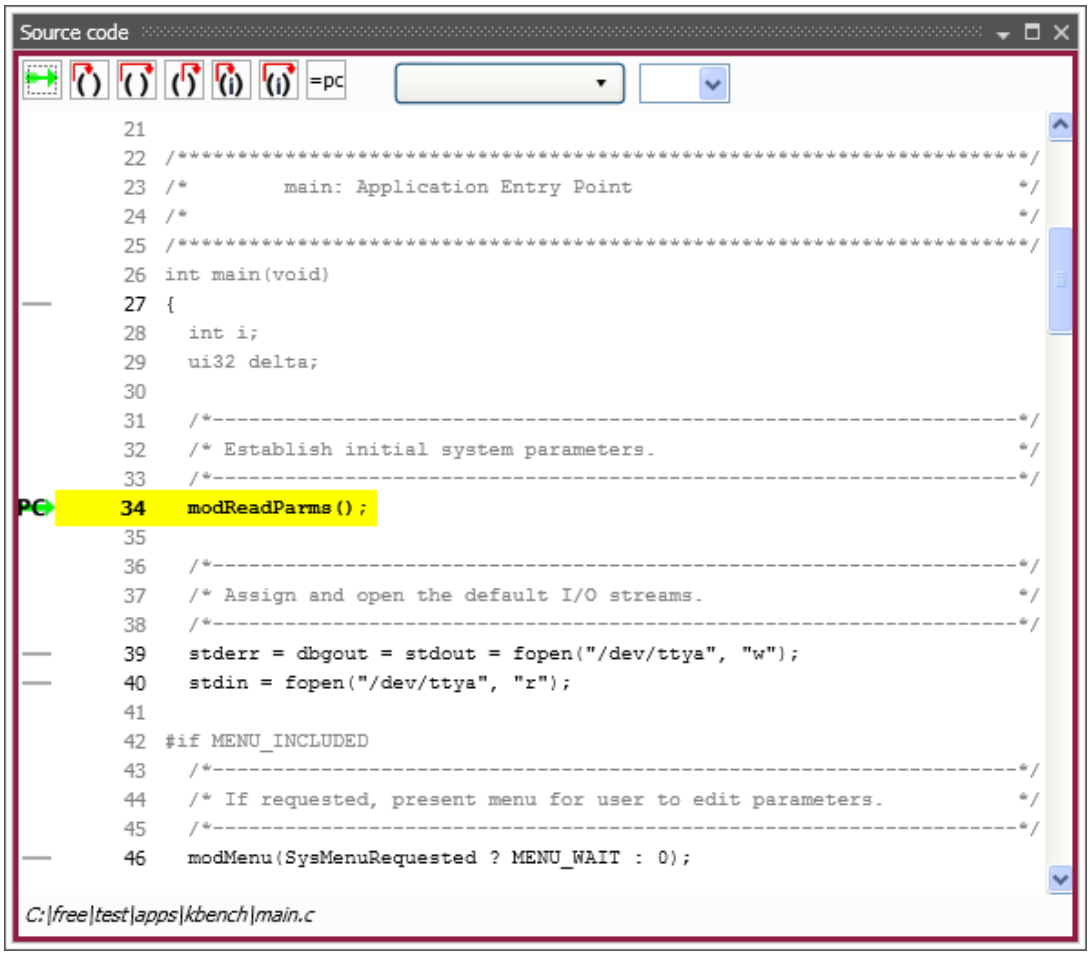
3 Running kbench

Now you are ready to build and run the kbench sample application, which 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:

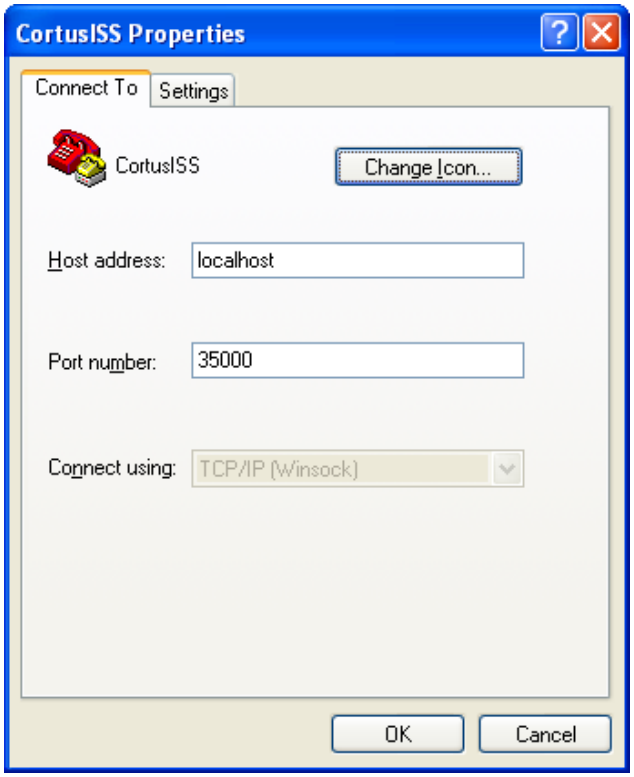
```
apps\kbench\kbench.gbp
```

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, "dbg.elf successfully built" 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, 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.



The Cortus ISS includes a virtual UART connected to a Telnet server. The channel is available after the debugger is launched, so you can use it now. The port number is 35000 and the IP address is the loopback address (127.0.0.1). The setup for using HyperTerminal is shown below:



Since the channel is closed each time the debugger is closed or restarted, it is helpful to use a Telnet client that is easy to restart.

Now that the debugger is running, you can use its step-in, step-over, step-out, etc. commands and breakpoint facility to peruse the kbench sample application. The user interface is fairly intuitive. After you run past the modMenu() line shown in the picture of the debugger's source window above, you will see the following output in the Telnet window:

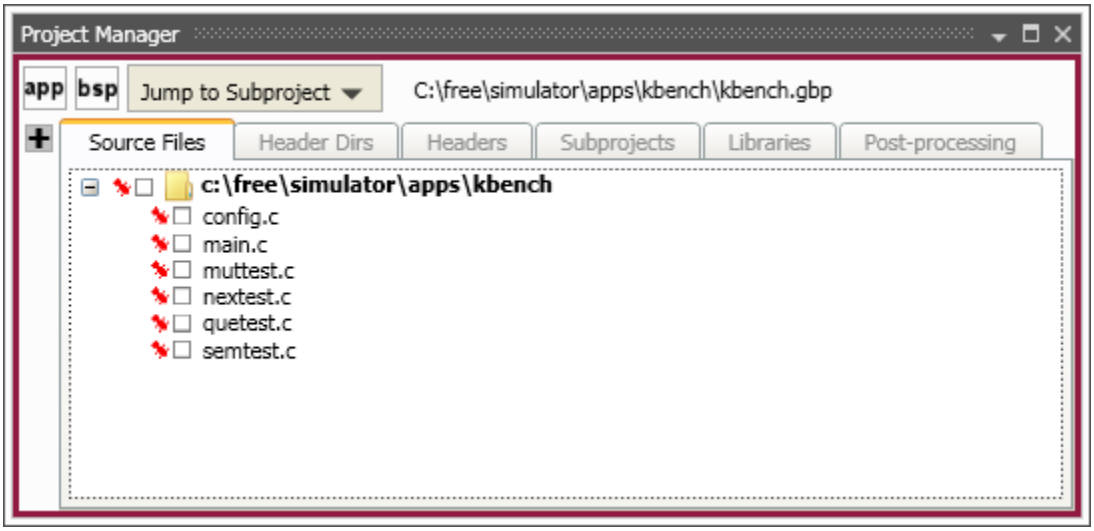
```
TargetOS, Release 2016.0. Built at 7/18/2016 1:19:09 PM.
Copyright 2015, Blunk Microsystems. All rights reserved.
-------------------------------------------------------
Starting Benchmark
SRAM data size = 1.5MB
SRAM text size = 2MB
Hostname: cortus_iss
IPv4 Default gateway: 192.168.1.1
Starting HTTP server for IPv4
Starting Telnet server for IPv4
-------------------------------------------------------
Press 'M' in 3 seconds to modify these settings:
```

Running the application's main loop once generates the following output:

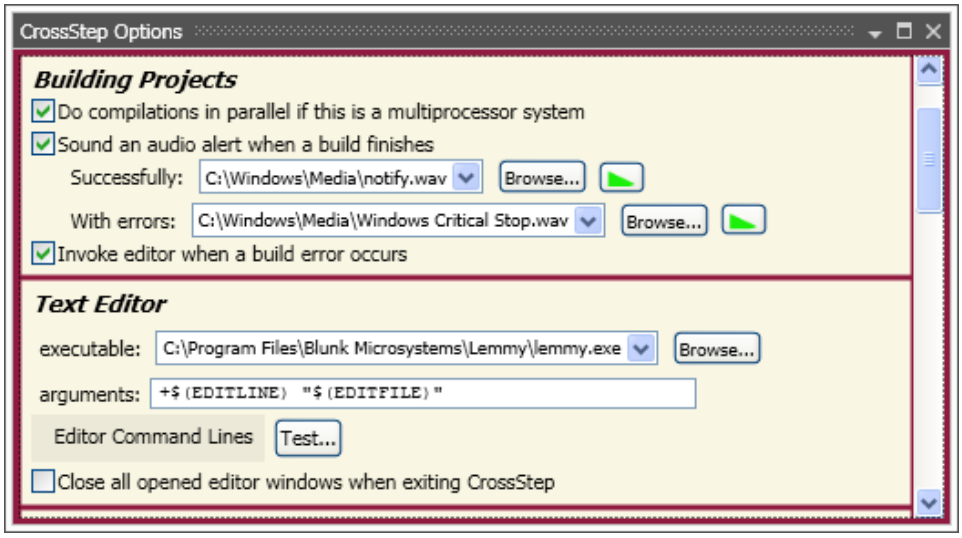
```
Semaphore Test
semPend() with waiting token = 39.0 usec
```

```
semPost() with no task waiting = 38.3 usec
semPost() with task made ready, no context switch = 75.8 usec
semPost() with context switch = 96.8 usec
semPend() with context switch = 135.8 usec
```

Clicking the **Source Files** tab in CrossStep's Project Manager window displays the application's source files. Double-click any source file line to launch an editor on the selected file. 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. The macros $(EDITLINE) and $(EDITFILE) are replaced with a 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. Specify the command line format expected by your editor on the "arguments" line using these macros. Clicking "Editor Command Lines" shows example argument strings for some 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