IoT Embedded Web Server

Hands On!

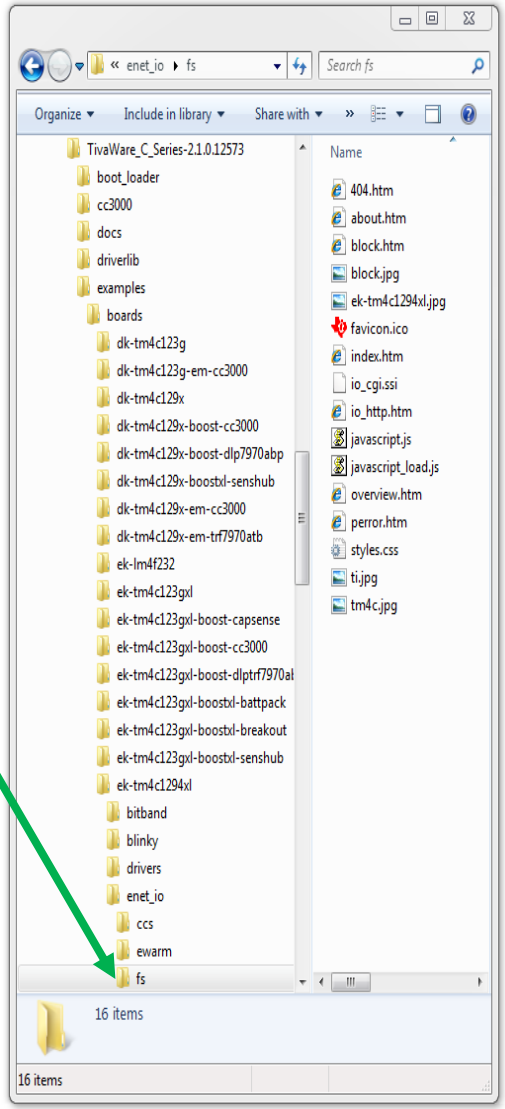
- Time to build an embedded web server!
- In this lab, we will:
  - Import, compile, and debug an embedded web server
  - Control the evaluation board from a web browser
  - Learn how to use the Code Composer Studio (CCS) tools
  - Learn where to go next!

Optional Preliminary Tasks

- The following are strongly recommended if you want to go beyond the out of box experience.
  - Download and Install Code Composer Studio or a supported compiler of your choice http://processors.wiki.ti.com/index.php/Download_CCS
  - Install the complete TivaWare for C Series Package http://www.ti.com/tool/sw-tm4c
  - Install LM Flash Programmer http://www.ti.com/tool/lmflashprogrammer
  - Install a UART Terminal Program such as Tera Term or PuTTY

Hands On: Locating the Webpage Source

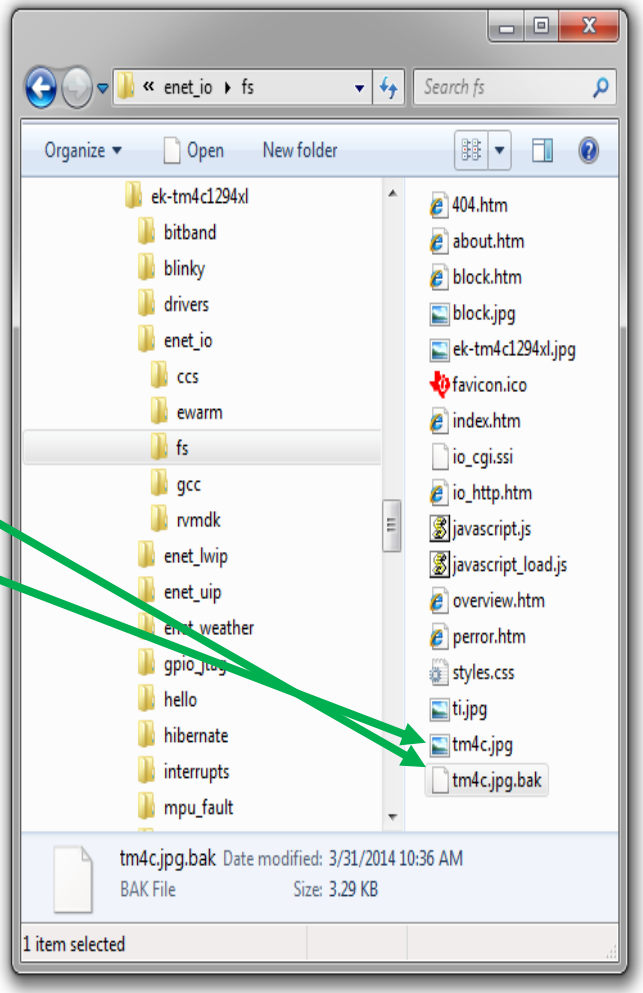
- The enet_io example project includes a sample webpage located within the ./fs folder shown here.
- This folder contains all of the required html, javascript, and image files that will be served directly from DK board.



Hands On : Tweak the Webpage (Optional)

- As a fun twist for the adventurous types, feel free to personalize this lab for either your company or a customer you support.
- Copy or create an image of your choice.
- To maintain formatting and ease of use, please choose an image that is approximately 75x75 pixels (wider is ok)
- Rename original “tm4c.jpg” to “tm4c.jpg.bak”
- Save your image within the ./fs folder, use the file name “tm4c.jpg”
- You are simply replacing the 

image with one of your own.

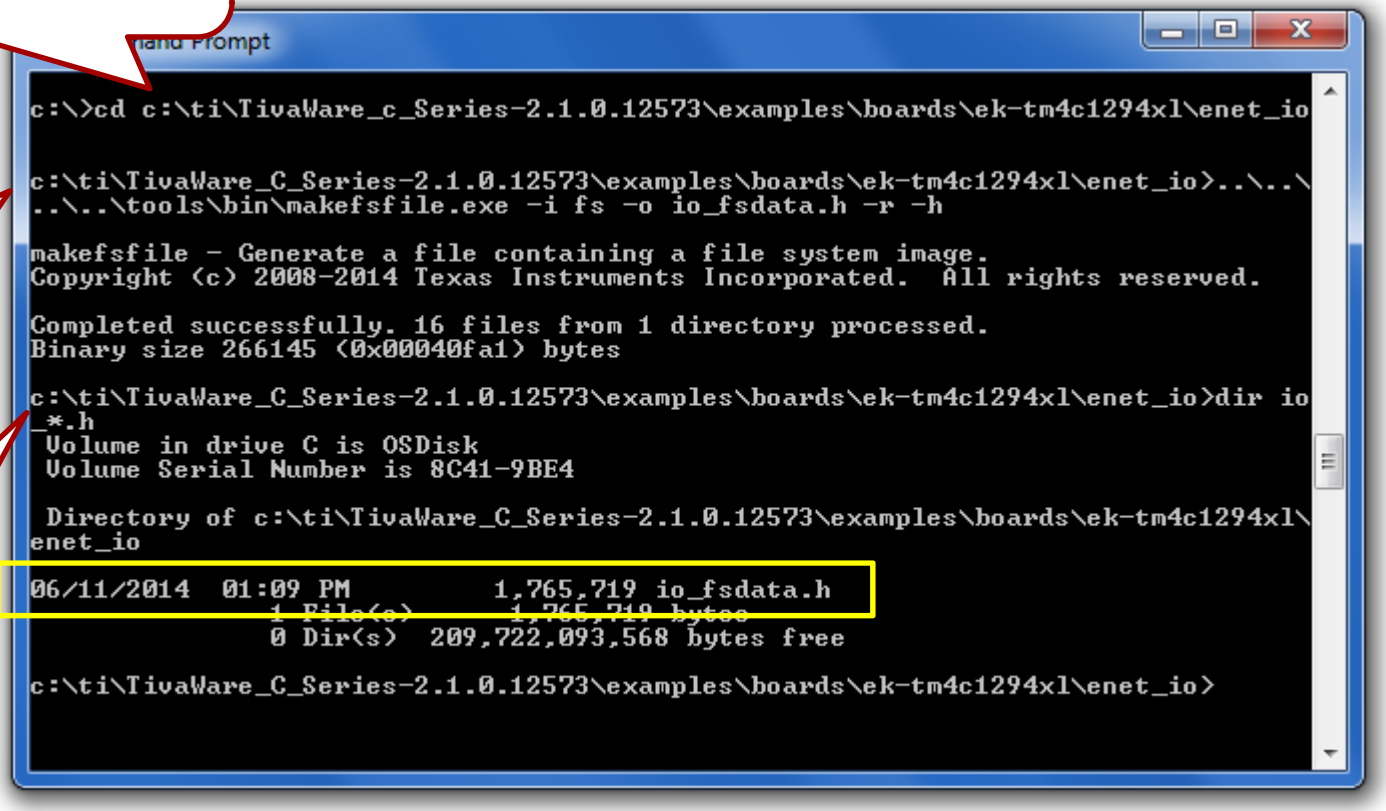


Hands On : Building the Webpage

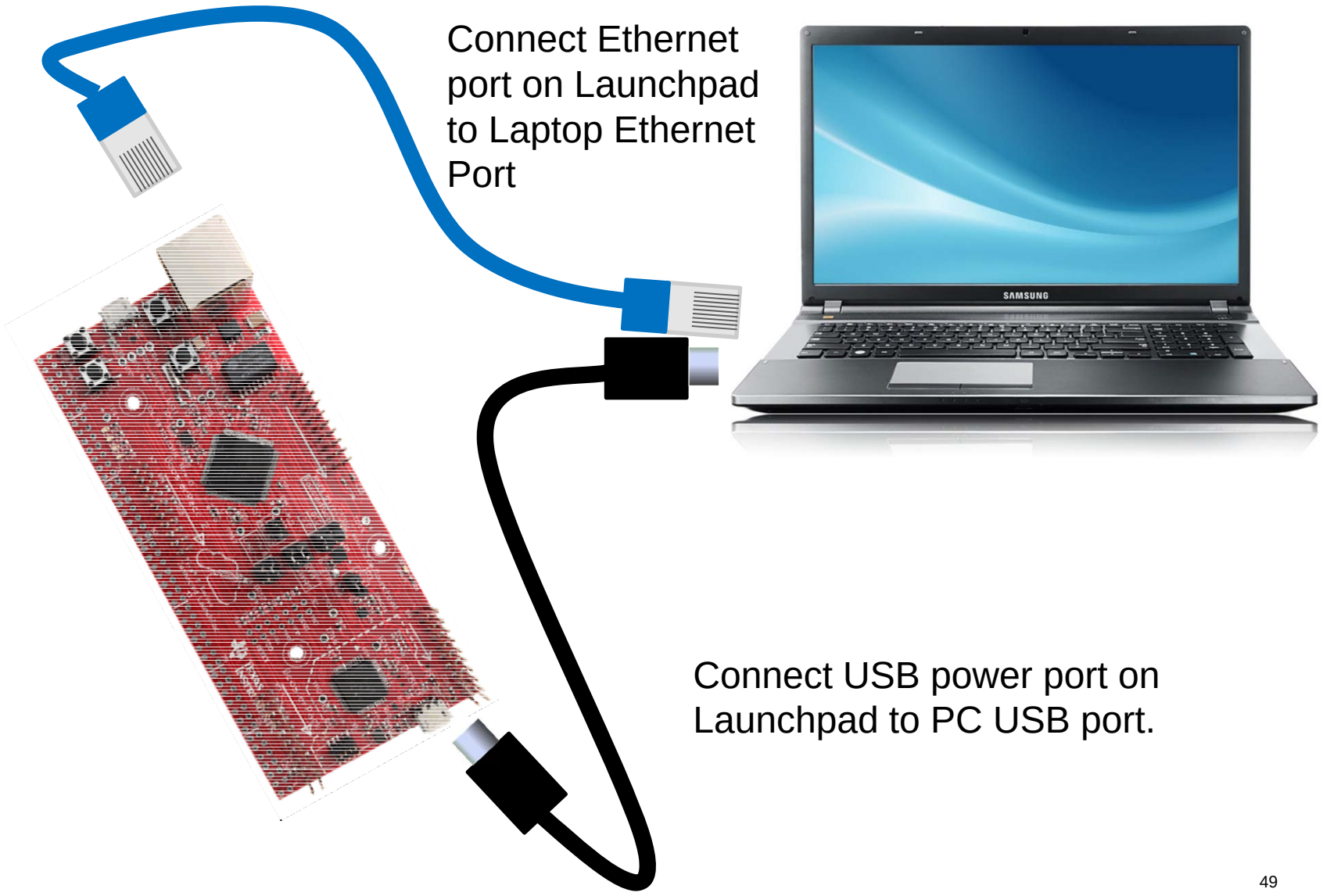
Open a CMD window and change the working directory to the directory shown or the equivalent on your system.

Execute the makefsfile.exe program via the following command.

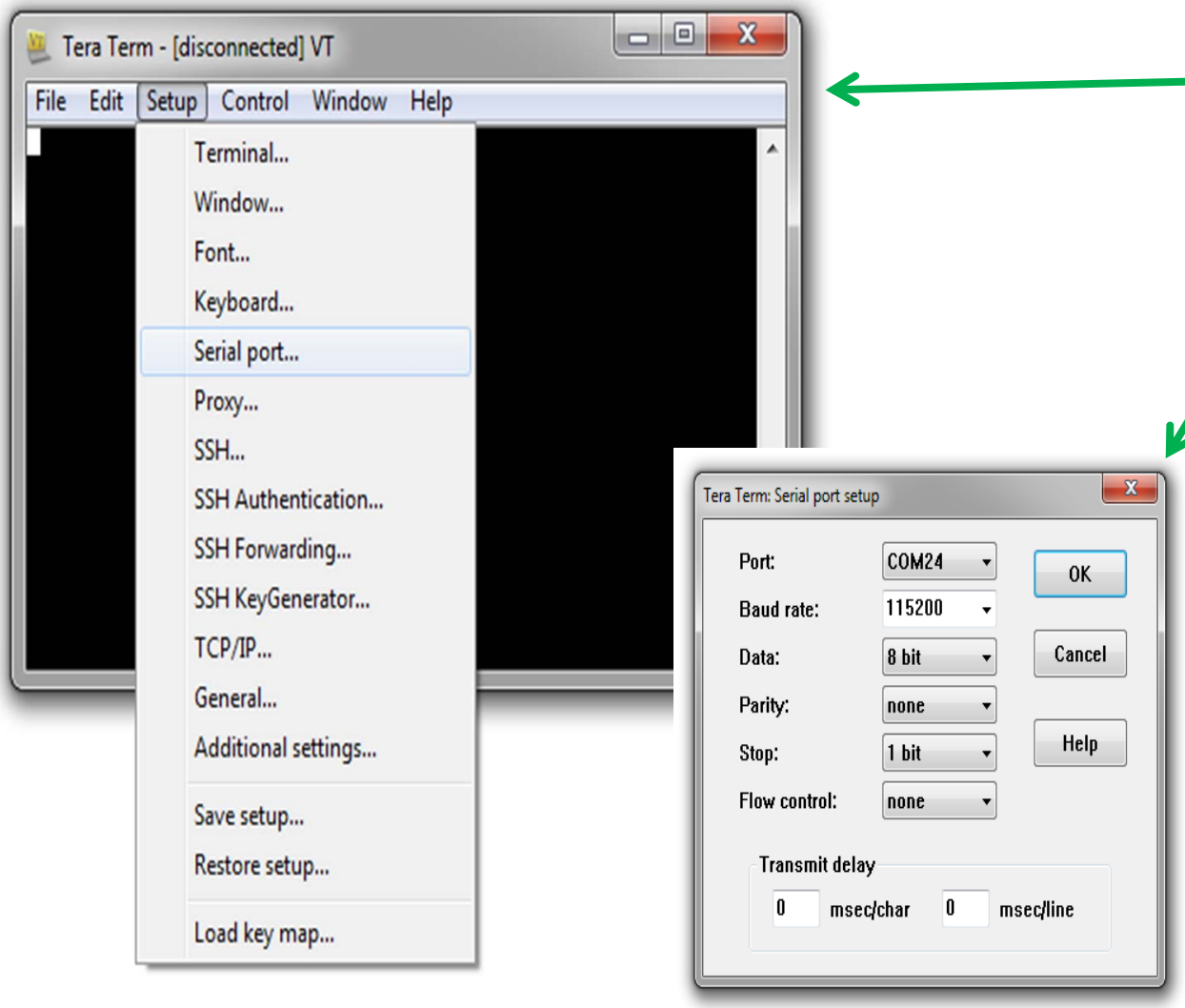
Confirm that the “io_fsdata.h” file has been created.



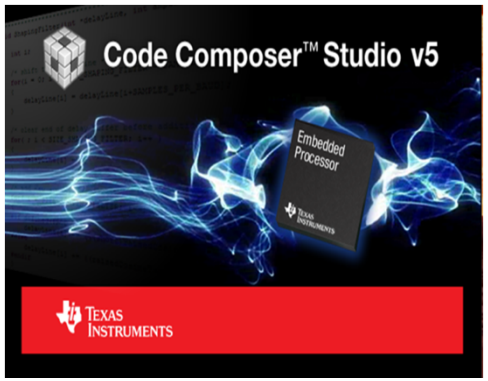
Hands On: Connect the Launchpad



Hands On : Open the terminal app



- Launch a terminal application such as Tera Term.
- Setup the terminal program as shown in the configuration window at the right including the correct COM port associated with your Connected Launch Pad.



Hands On: Using Code Composer Studio v5.5

- The CCS IDE is a suite of tools for embedded applications. It includes C/C++ compilers for each of TI's device families, source code editor, project build environment, debugger, profiler, simulators and many other features.
- CCS licenses:
  - Evaluation – free limited licenses that can be used to evaluate TI tools and devices.
  - Node locked - a license that is tied to a specific computer.
  - Floating - licenses that can be shared across multiple computers.
  - Code Size Limited - free 16KB code size limited license for MSP430.
  - Bundle/Dev Kit - free license for use with EVMs and development boards (Tiva, MSP430 and Cortex R4)
  - University - contact the TI University Program for details.

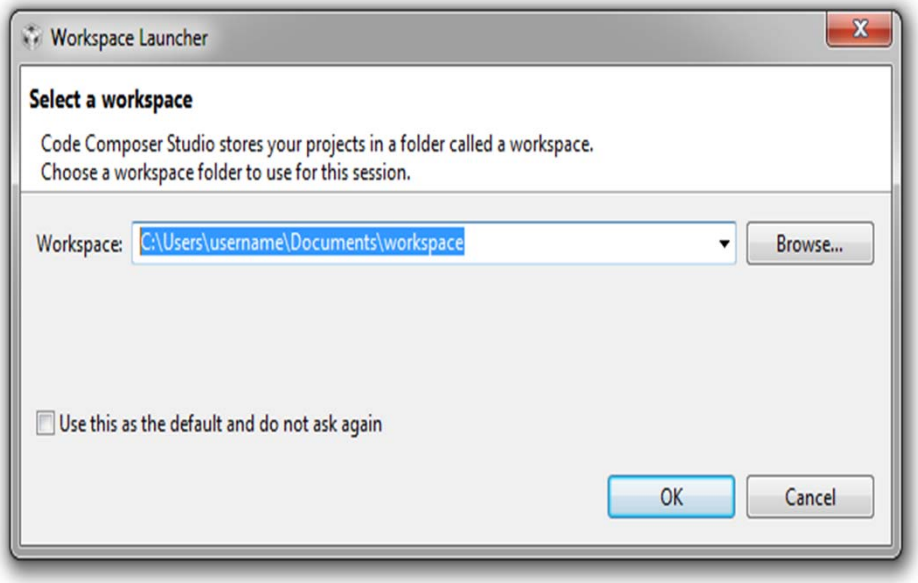
- Learn more about CCS at: www.ti.com/ccs

- TI E2E Support Community for CCS at: http://e2e.ti.com/support/development_tools/code_composer_studio/default.aspx

Hands On: Open CCS

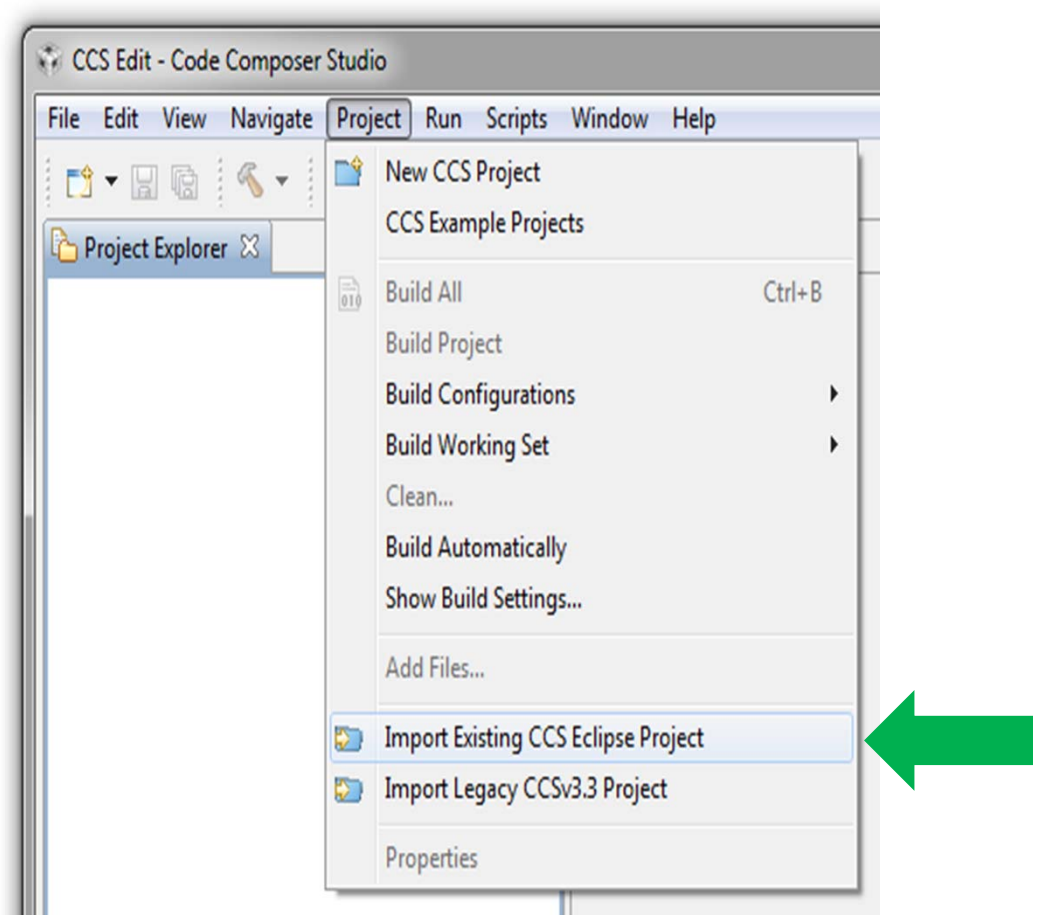
- When CCS starts, it will ask you to select a workspace. Use the default location it has pre-selected and click OK. This path should be similar to:

```
C:\Users\<username>\Documents\workspace
```



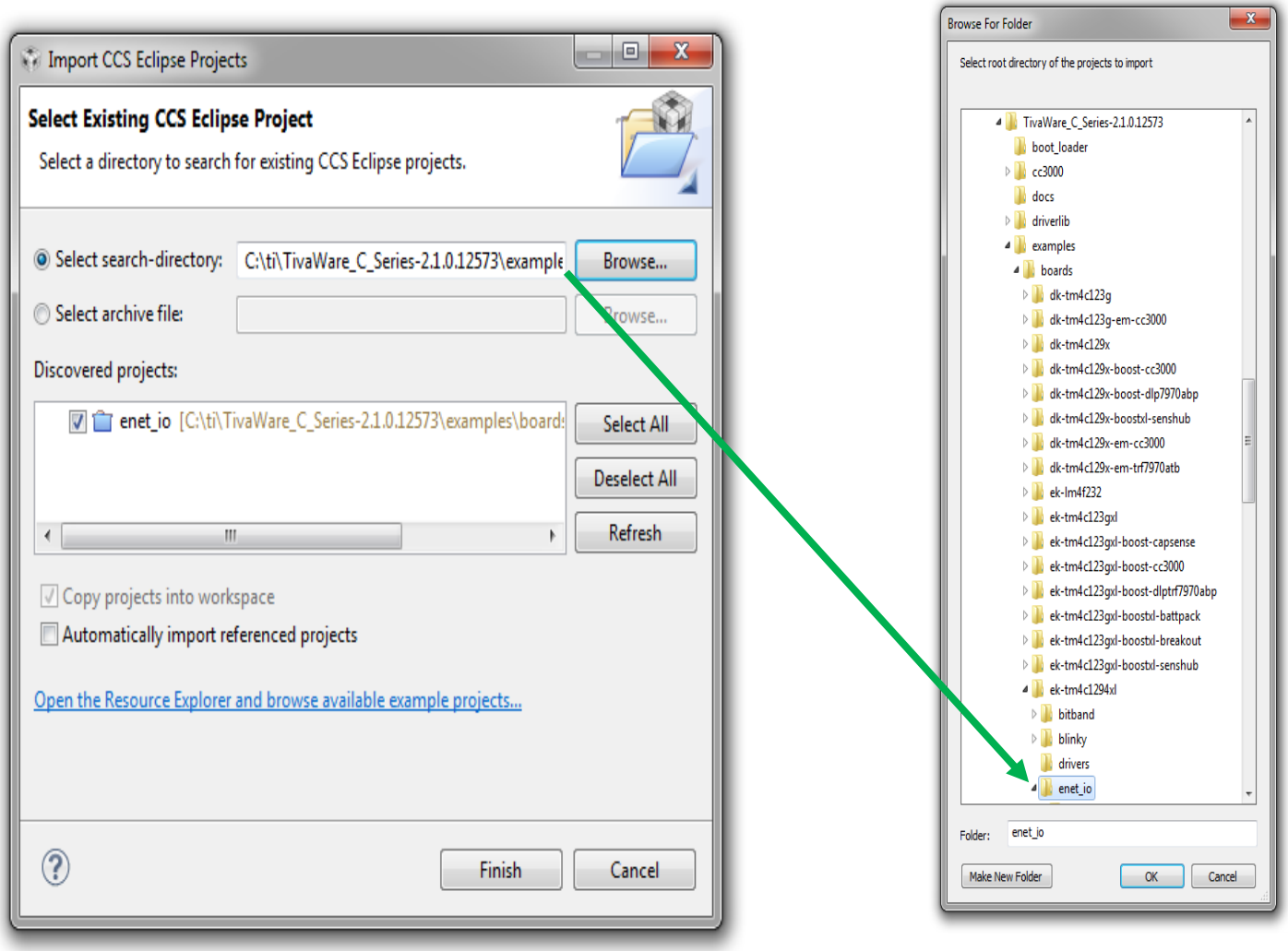
Hands On: CCS Import existing Project

Import enet_io from TivaWare, the path will be similar to:
C:\ti\TivaWare_C_Series-2.1.0.12573\examples\boards\ek-tm4c1294xl\enet_io



Hands On: CCS Import existing Project

Click browse button and find enet_io folder



Once enet_io project folder is selected click finish.

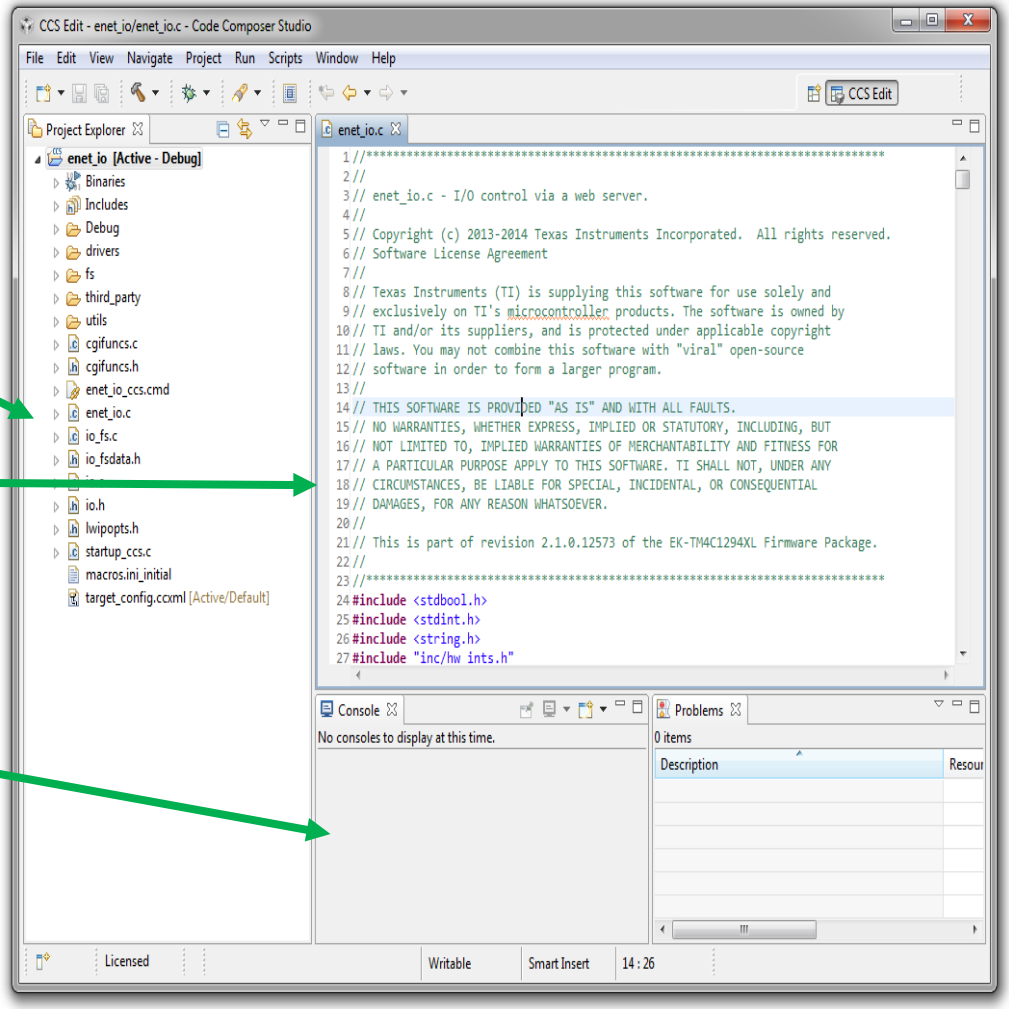
Hands On: CCS IDE C/C++ Perspective

- The CCS IDE is a customized version of the open source Eclipse IDE

Workspace/source browser

Source editor

Console, error window



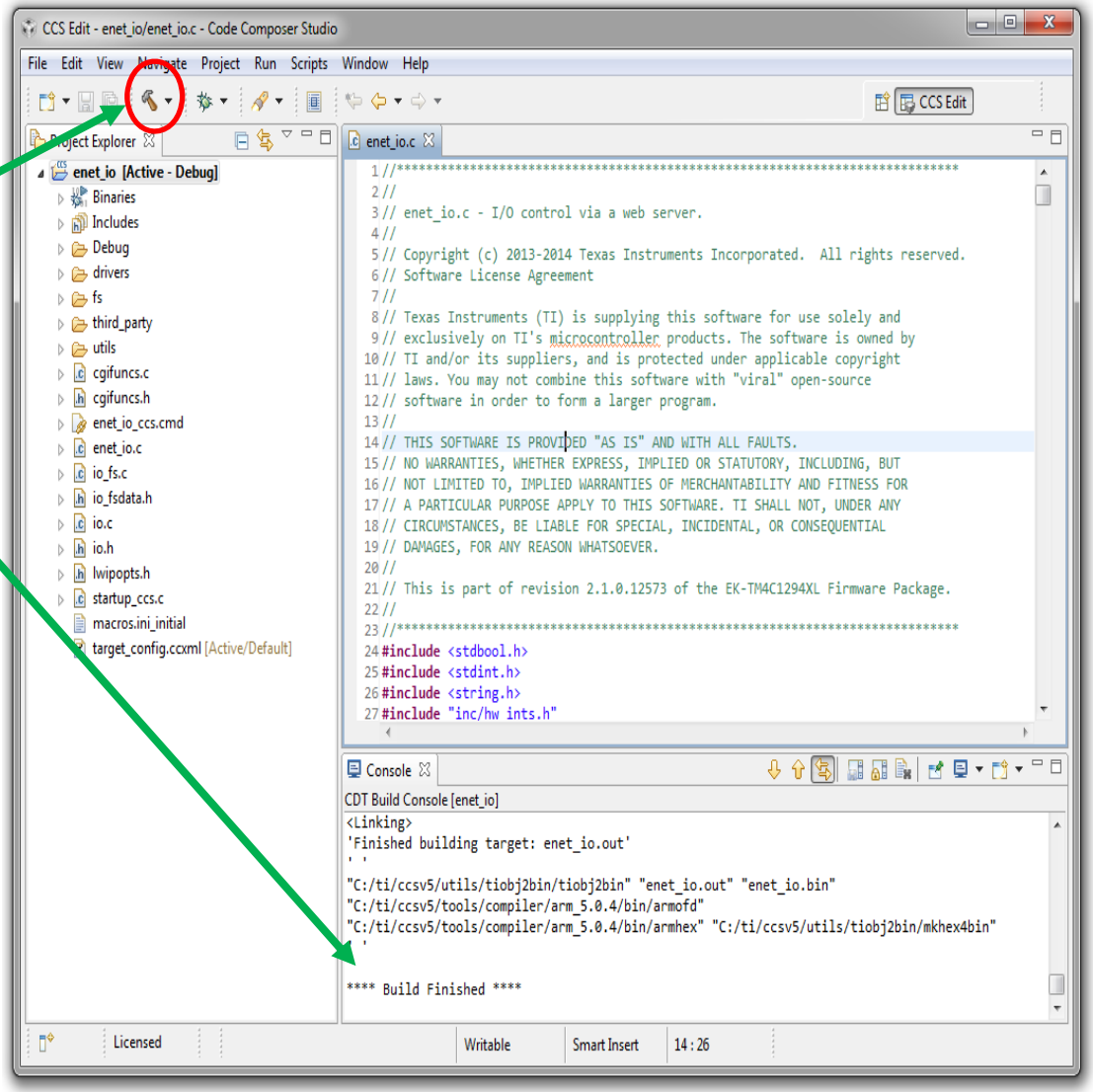
Hands On: Issue patch in demo code

- An issue has been identified that impacts the timeout period while obtaining an IP Address.
- The highlighted code (lines 534 – 546) below in Enet_io.c should be commented out before building to prevent an extended timeout period.

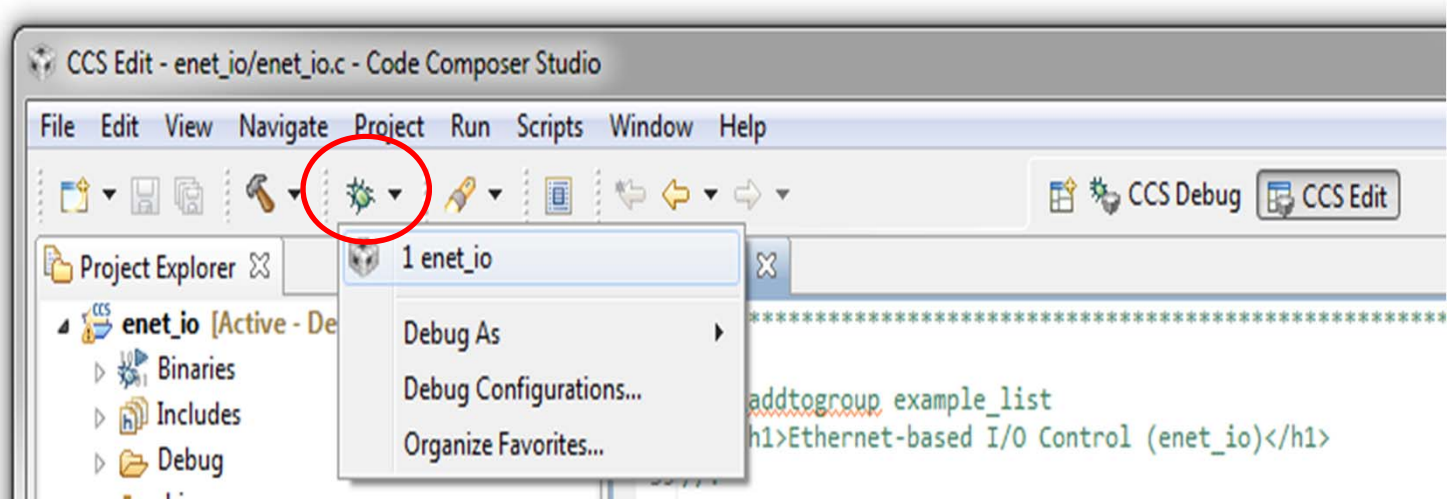
```
528     if((ui32NewIPAddress == 0) || (ui32NewIPAddress == 0xffffffff))
529     {
530         //
531         // Loop through the LED animation.
532         //
533
534         for(ui32Idx = 1; ui32Idx < 17; ui32Idx++)
535         {
536
537             //
538             // Toggle the GPIO
539             //
540             MAP_GPIOPinWrite(GPIO_PORTN_BASE, GPIO_PIN_1,
541                              (MAP_GPIOPinRead(GPIO_PORTN_BASE, GPIO_PIN_1) ^
542                               GPIO_PIN_1));
543
544             SysCtlDelay(g_ui32SysClock/(ui32Idx << 1));
545
546         }
547     }
548 }
549
```

Hands On: Building the enet_io Project

- The enet_io and supporting projects build when icon is clicked.
- Wait until it finishes.

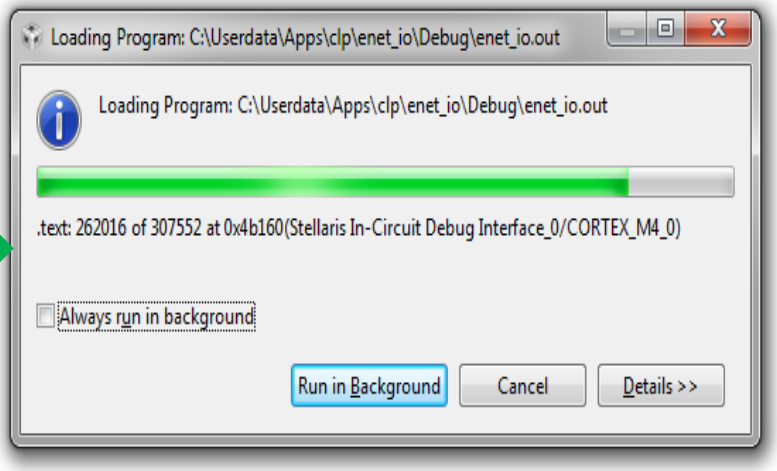


Hands On: Program and Debug Project



Note: To prevent conflicting network settings, please disable your WiFi or other active TCP/IP connections while debugging directly connected to the Tiva C 1294x EK.

- Make sure EK-TM4C1294XL evaluation board is connected to your host PC
- Click the  icon, and the application will be programmed into the internal flash and the debugger will be started.



Hands On: Program and Debug Project

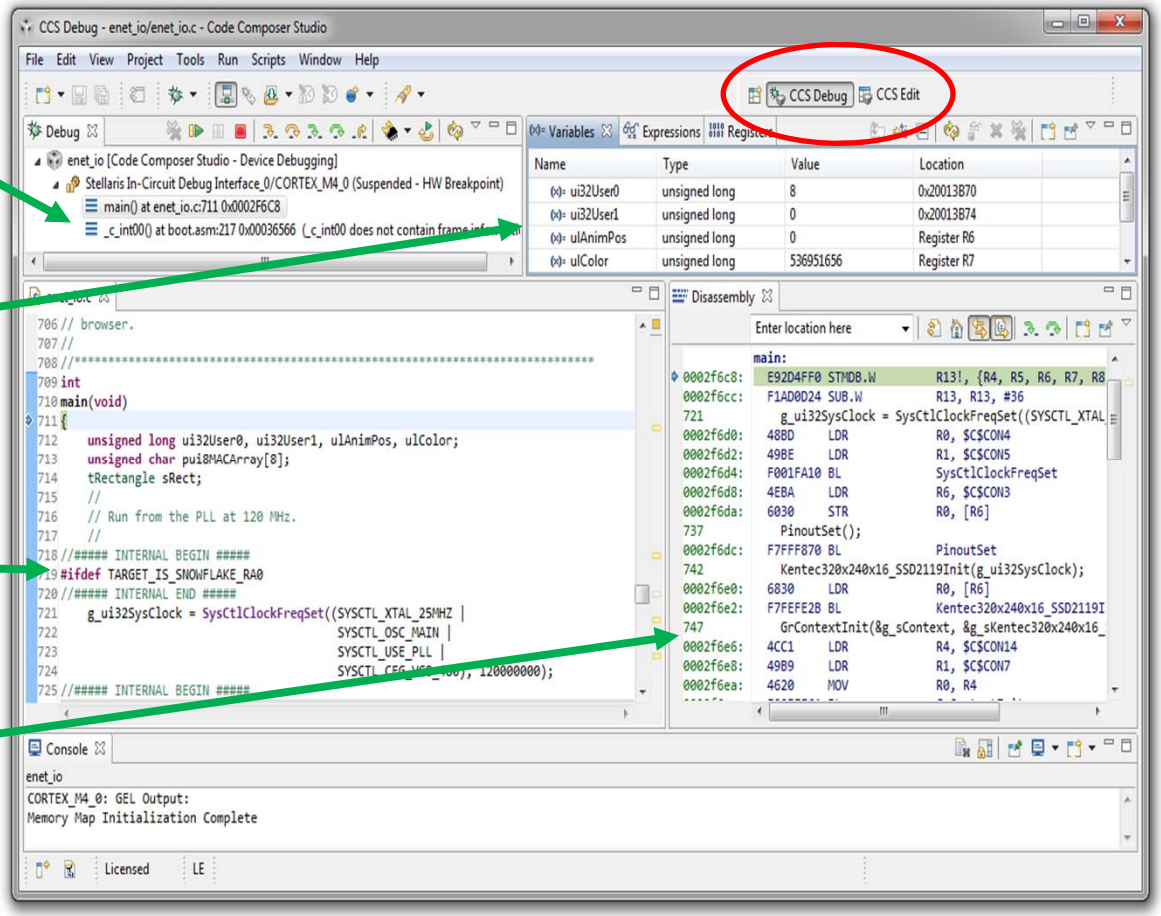
- The CCS debugger will automatically connect to your evaluation board, program the flash and run to the beginning of the main() function.

Debug thread view window

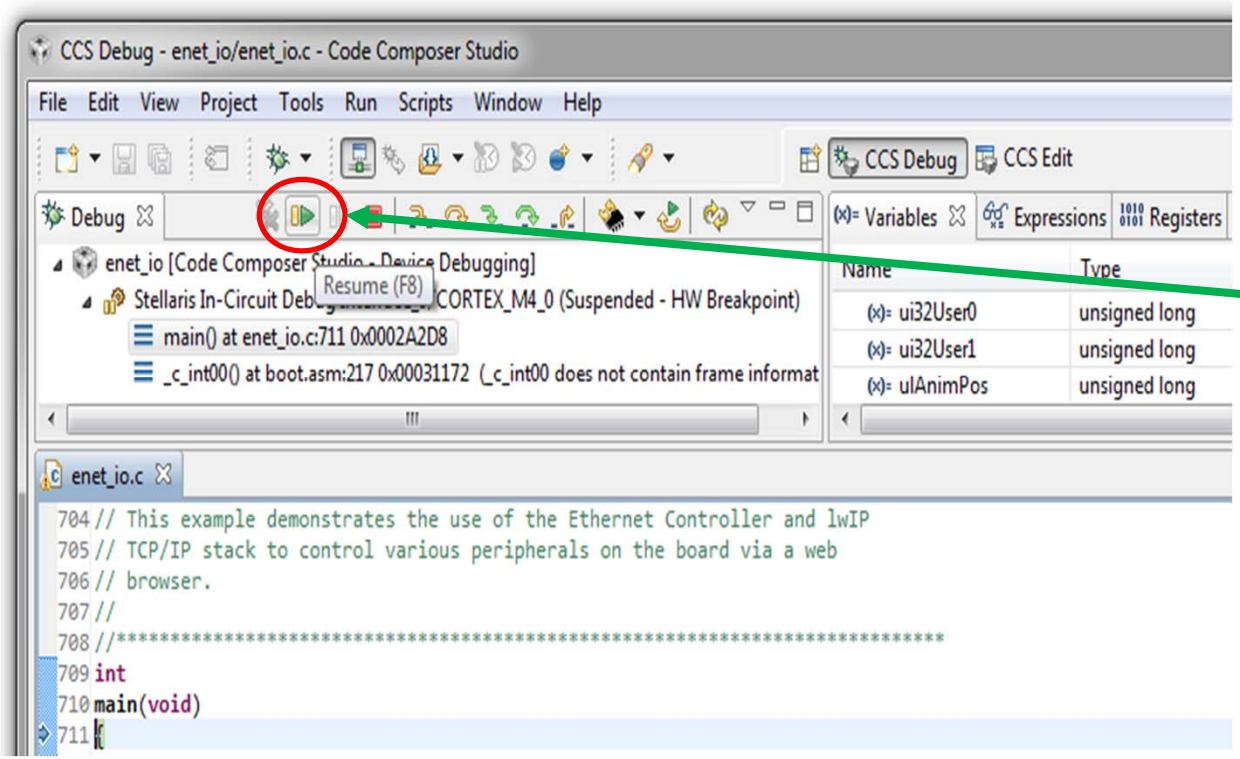
Core register, variable, breakpoint, peripheral viewers

Source/PC view

Disassembly and Memory windows

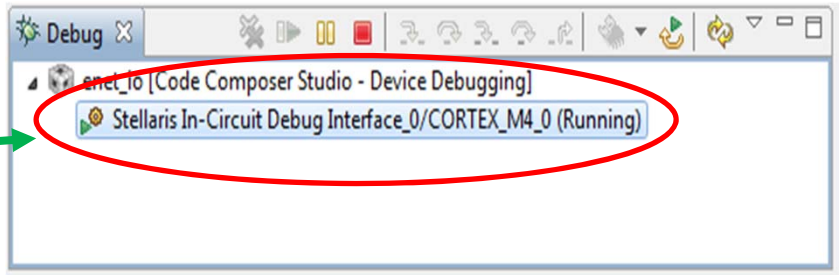


Hands On: Program and Debug Project



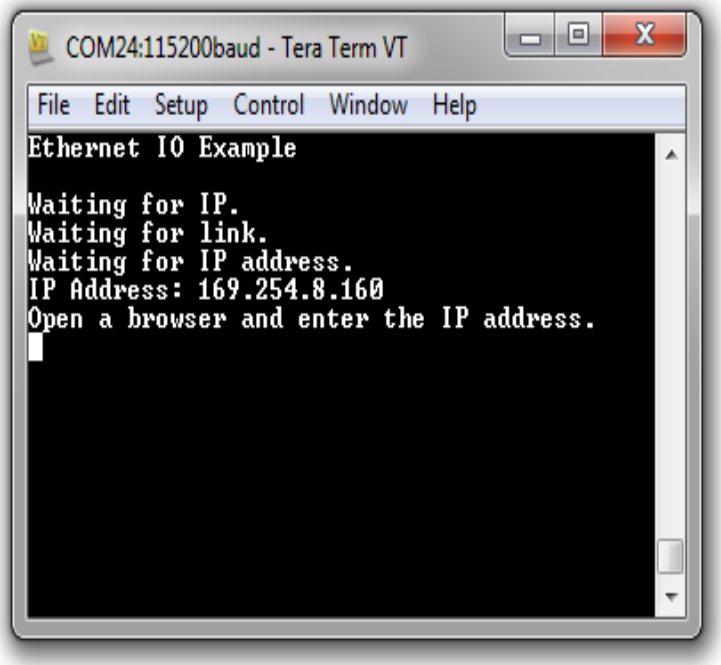
- Once you have acquainted yourself with the IDE windows, click the  icon, and the application will begin executing on the Tiva C-series core.

- You'll notice that the process status within the "Debug" window will change to "(Running)"



Hands On: IP Address Assignment

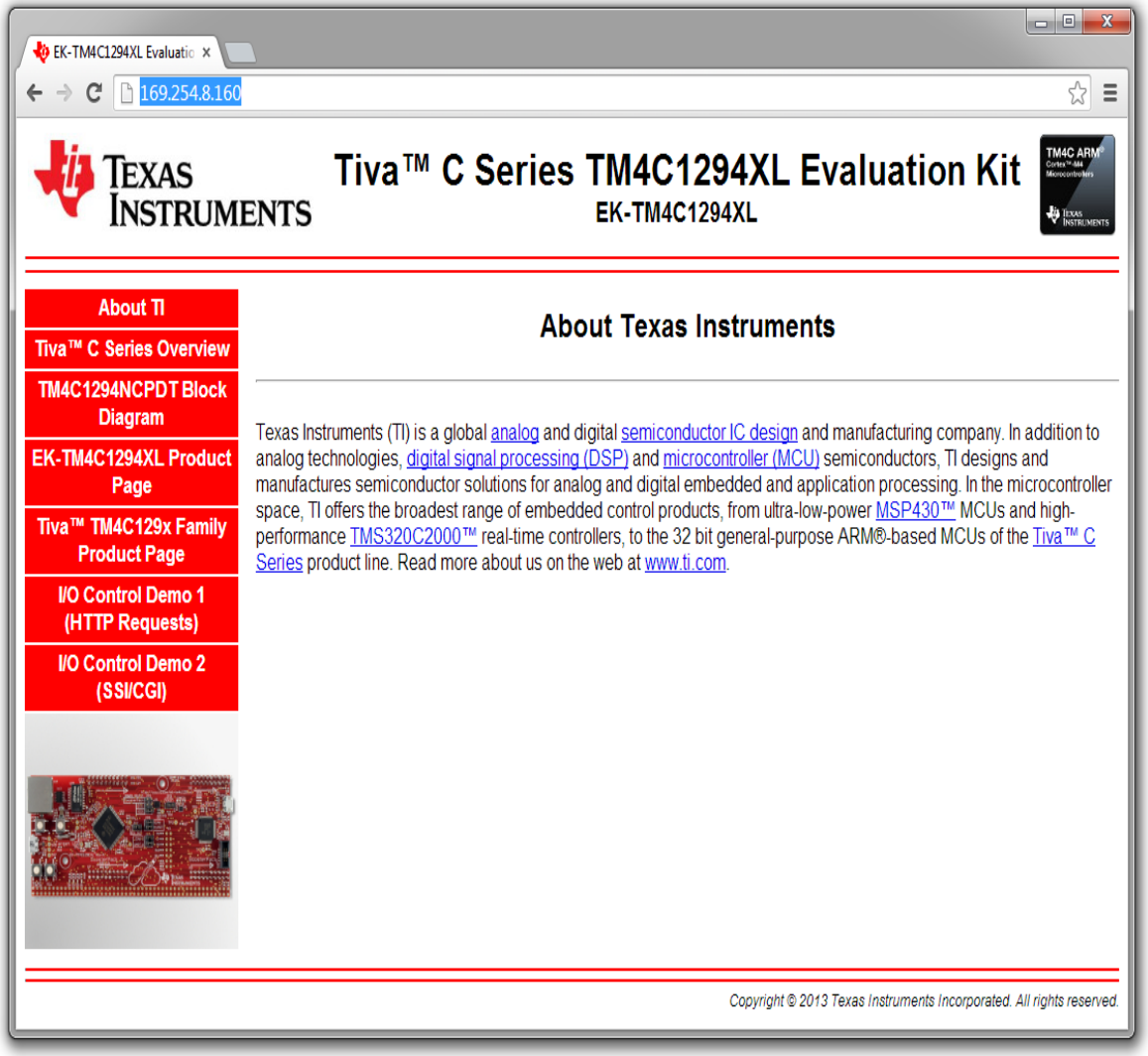
- Our example application first attempts to use DHCP to obtain an IP address.
- With a DHCP server available, an IP address is typically assigned to the board in a few seconds.
- If DHCP times out (as it will here since we do not have a DHCP server available), a link-local IP address (169.254.xx.yy) is chosen using the AUTOIP (RFC3297) protocol.
- DHCP timeout defaults to 1 minute so it may take some time for the board and laptop to pick an IP addresses.
- The lwIP TCP/IP stack can be configured for DHCP, AUTOIP or static IP address use.



Hands On: Embedded Control Web Page

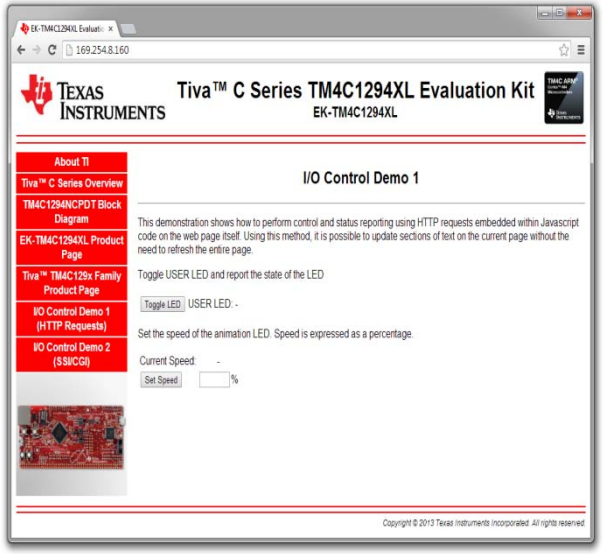
HTML code and images are stored within the application binary in FLASH memory as a file system image (io_fs.c)

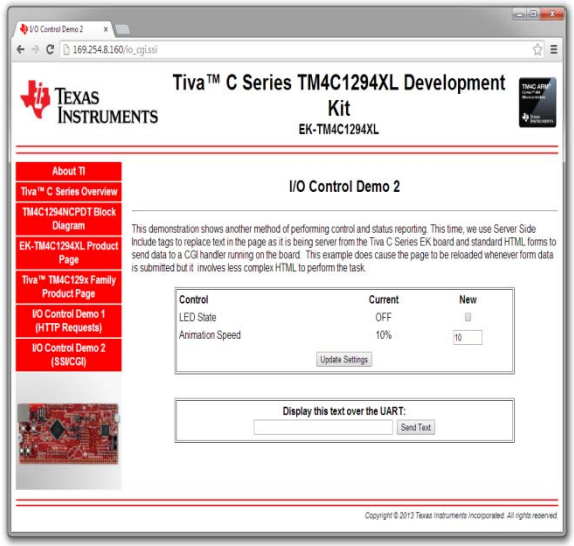
HINT: You may need to disable firewall software installed on your PC



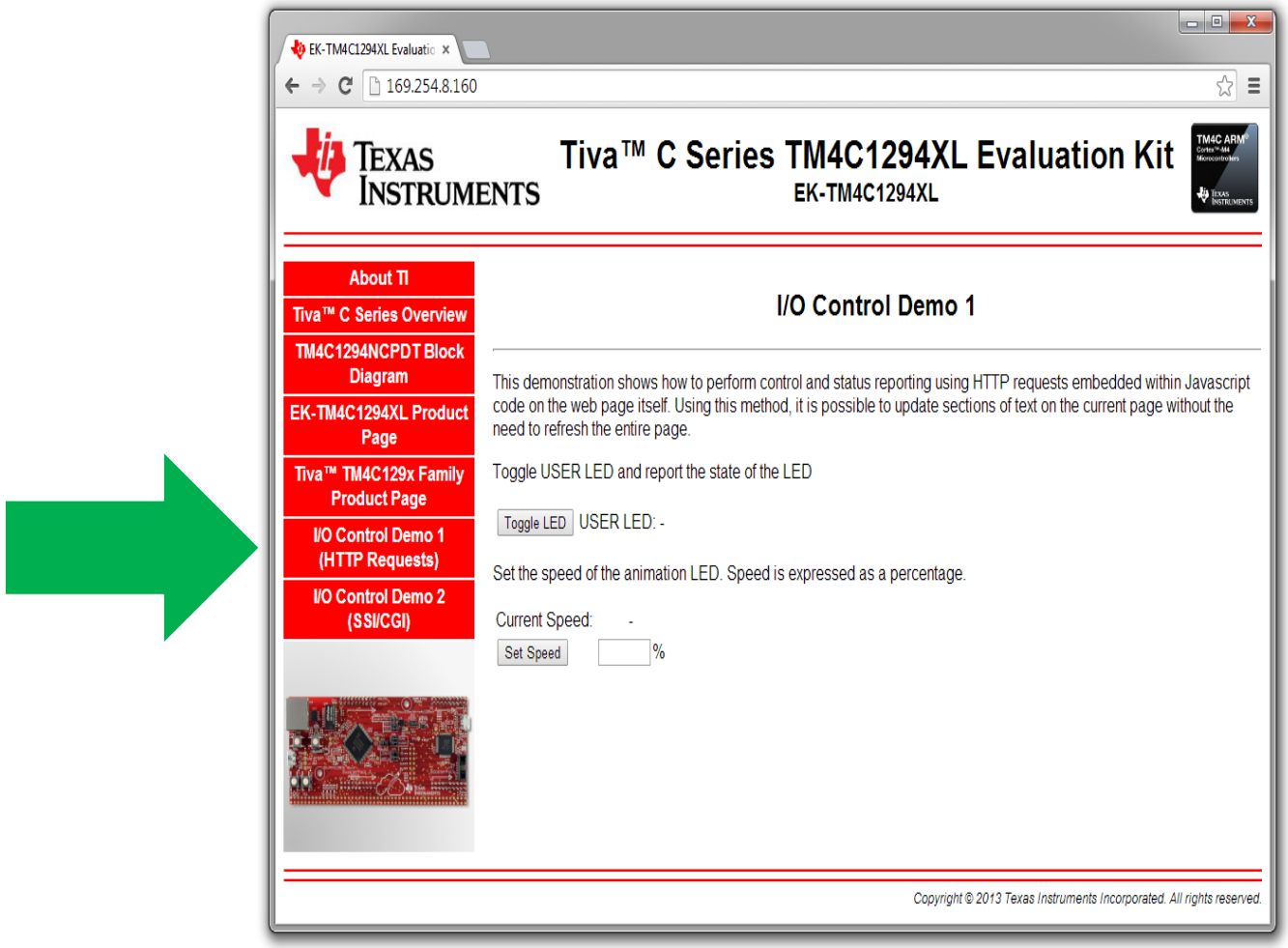
Hands On: I/O Control Demonstrations

- The enet_io application illustrates two methods of controlling board operations from the web browser:
  - I/O Control Demo 1 shows direct HTTP requests generated via JavaScript code in the web page (io_http.html).
  - I/O Control Demo 2 shows the use of Server Side Includes (SSI) and Common Gateway Interface (CGI) to perform the same operations (io_cgi.ssi)





Hands On: I/O Control Demo 1



Hands On: Demo 1– HTTP Requests

- JavaScript in the web page generates HTTP GET requests for specific filenames.

- lwIP web server passes these filenames to the file system (`io_fs.c`) which recognizes them as “special” and performs whichever task the filename is intended to trigger.
  - Toggle LED D1 state (`/cgi-bin/toggle_led`)
  - Set the blink rate of LED D2 (`/cgi-bin/set_speed?percent=`<speed_percent>)

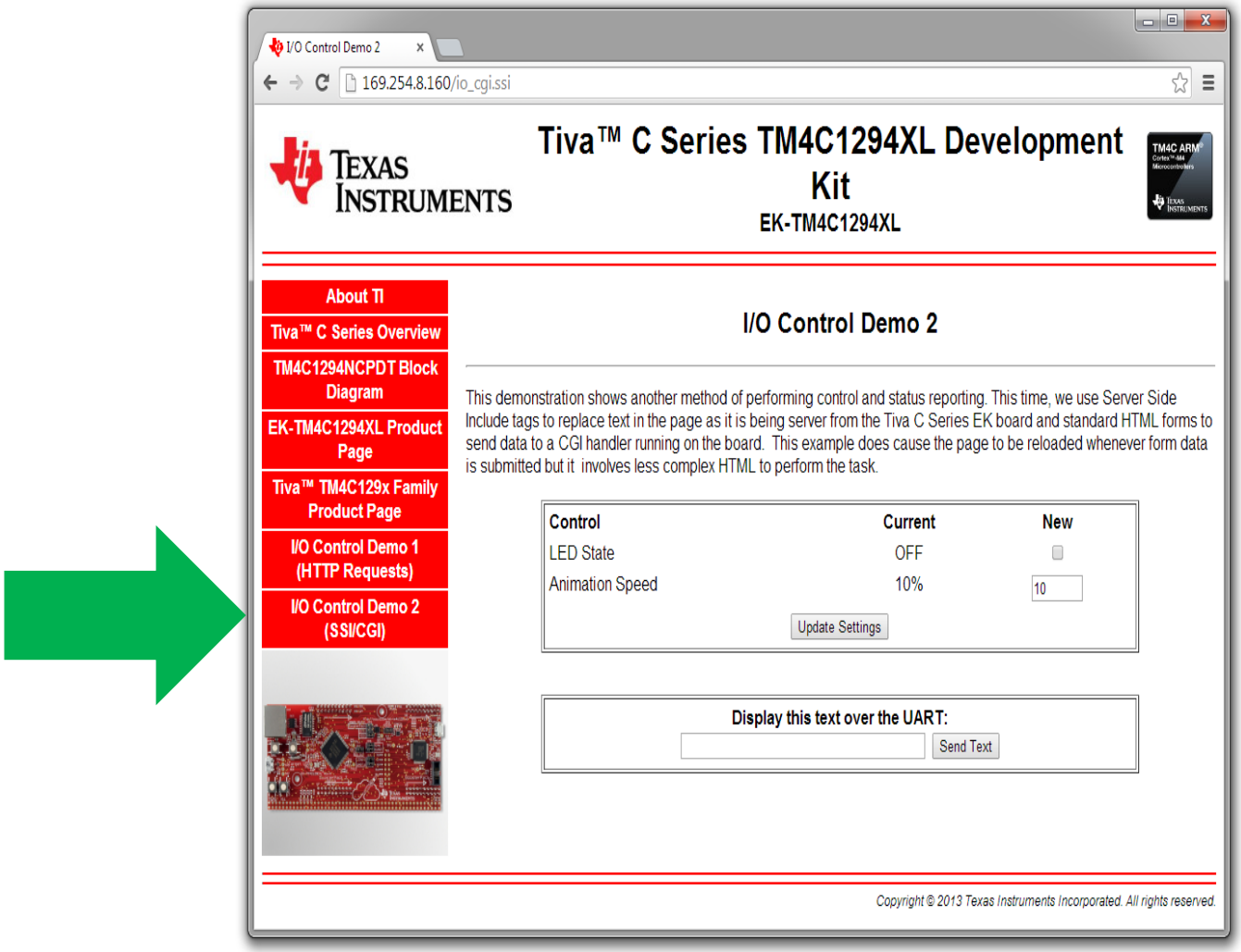
- JavaScript reads the response from the web server and uses `<div>` tags to insert the relevant text into the displayed page.

Hands On: Demo 1– HTTP Requests

- **Advantages**:
  - Updates can be made without having to reload the whole web page.
  - Quicker user feedback.
  - Lower network traffic.

- **Disadvantages**:
  - Client browser must support JavaScript.
  - HTML files are more complex to develop and harder to understand.

Hands On: Demo 2 – SSI/CGI



Hands On: Demo 2 – SSI/CGI

- HTML pages include “Server Side Include” tags indicating values to be inserted in the page data as it is served to the browser.
- The application registers SSI and CGI handlers with the HTTP server during initialization.
- The HTTP server calls the SSI handler when a tag from the registered list is detected and handler returns the text to insert after the tag.
- The HTTP server calls the registered CGI handler if a URL matching the registered CGI name is requested.
- The HTML contains standard forms to gather user input.

Hands On: Demo 2 – SSI/CGI

- **Advantages**
  - Client browser does not need to support JavaScript.
  - HTML is extremely simple and uses only standard forms and some “comment-like” SSI tags (<!--#<tag>-->).
  - Offloads work to the common HTTP server module (URL checking, parameter parsing).
  - File system driver is independent of the application data that it is managing.

- **Disadvantages**
  - Page reload each time a form is submitted.

Review: Embedding your own web site

- To reiterate, HTML and images are stored within the application binary as a file system image.
- This data is collapsed into a single header file called `io_fsdata.h`.
- The application web site can be found in the `fs` directory imported alongside the enet_io application.
- After changing content in the fs directory, rebuild the file system image using command:

  ```
  makefsfile -i fs -o io_fsdata.h -h -r
  ```

  executed from the enet_io directory under your workspace (typically): `C:\ti\TivaWare_C_Series-2.1.0.12573\examples\boards\ek-tm4c1294xl\enet_io\`
- The makefsfile.exe utility can be found in: `C:\ti\TivaWare_C_Series-2.1.0.12573\tools\bin\makefsfile.exe`