

# IoT Embedded Web Server

# Hands On!

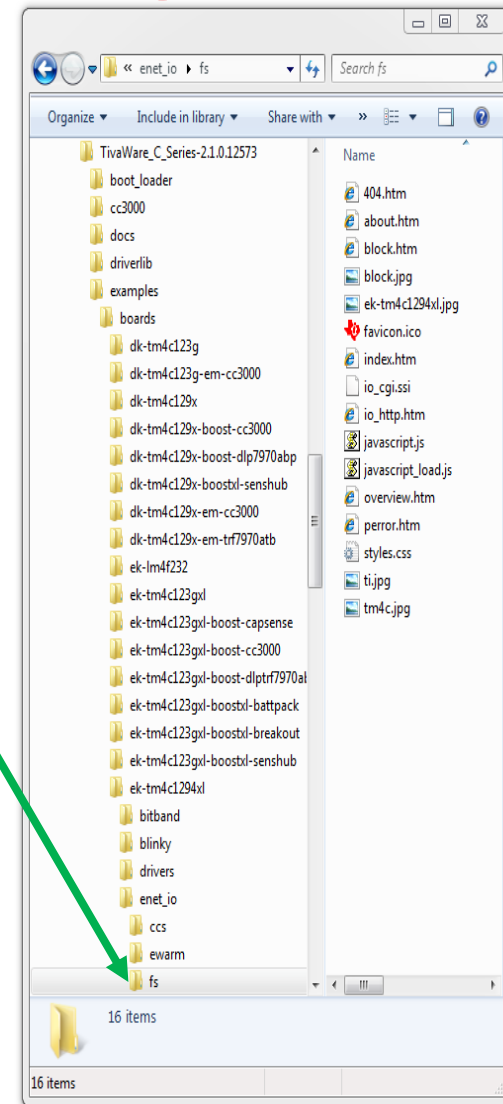
- Time to build an embedded web server!
- In this lab, we will:
  - Import, compile, and debug an embedded web server
  - Control the evaluation board from a web browser
  - Learn how to use the Code Composer Studio (CCS) tools
  - Learn where to go next!

# 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](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

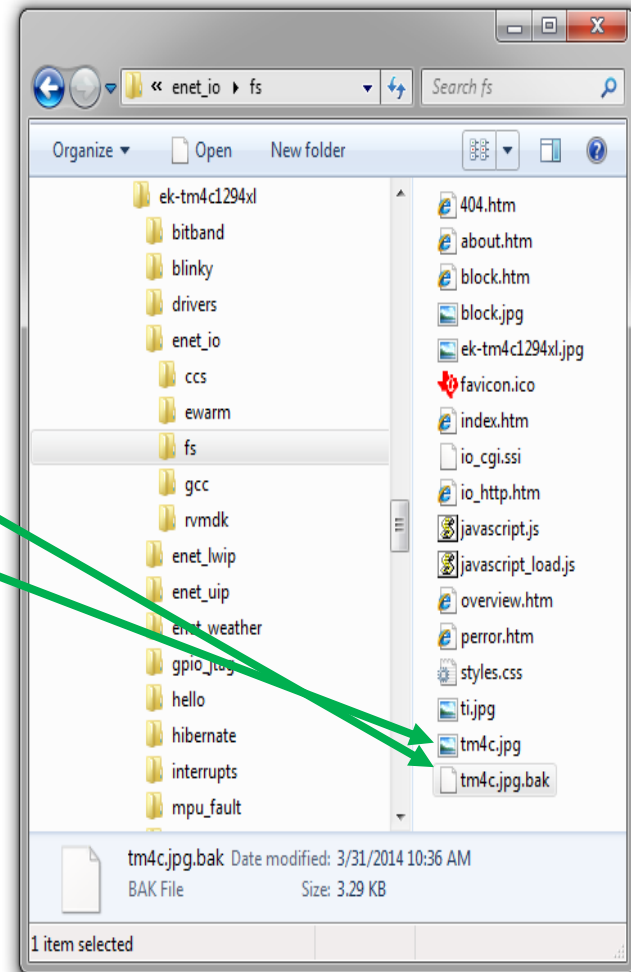
# Hands On: Locating the Webpage Source

- 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.

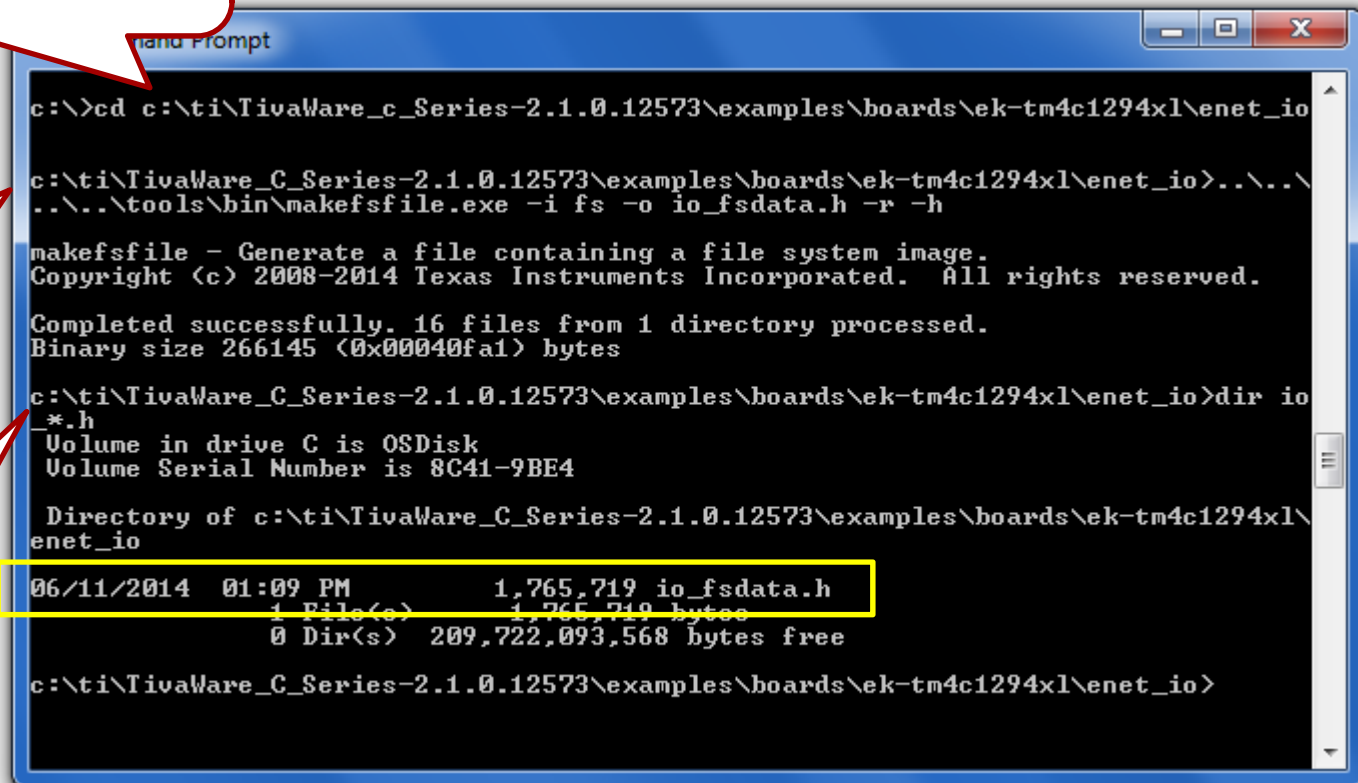


# Hands On : Building the Webpage

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.



```
c:\>cd c:\ti\TivaWare_C_Series-2.1.0.12573\examples\boards\ek-tm4c1294xl\enet_io

c:\ti\TivaWare_C_Series-2.1.0.12573\examples\boards\ek-tm4c1294xl\enet_io>..\..\tools\bin\makefsfile.exe -i fs -o io_fsdata.h -r -h

makefsfile - Generate a file containing a file system image.
Copyright (c) 2008-2014 Texas Instruments Incorporated. All rights reserved.

Completed successfully. 16 files from 1 directory processed.
Binary size 266145 (0x00040fa1) bytes

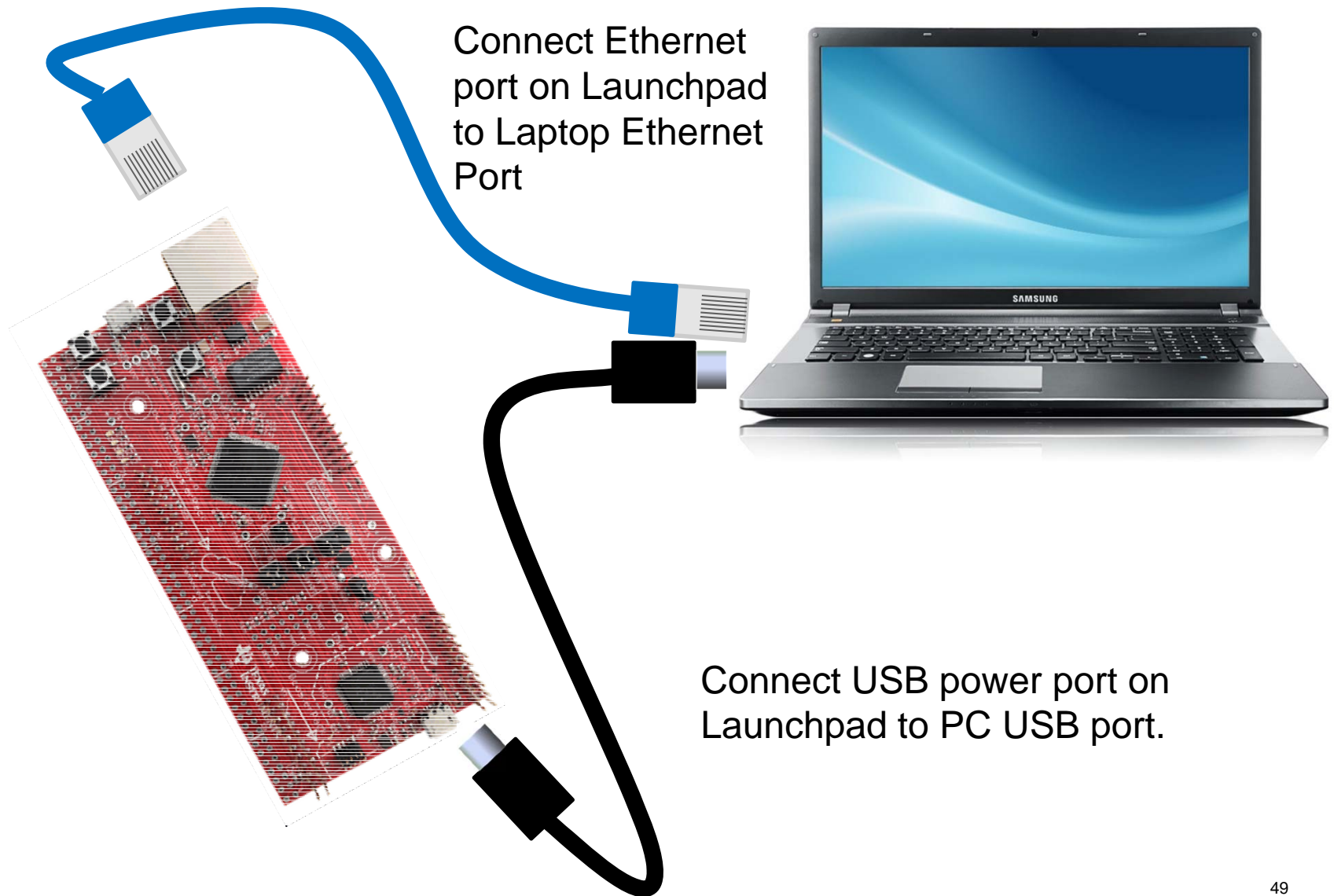
c:\ti\TivaWare_C_Series-2.1.0.12573\examples\boards\ek-tm4c1294xl\enet_io>dir io_*.h
Volume in drive C is OSDisk
Volume Serial Number is 8C41-9BE4

Directory of c:\ti\TivaWare_C_Series-2.1.0.12573\examples\boards\ek-tm4c1294xl\enet_io

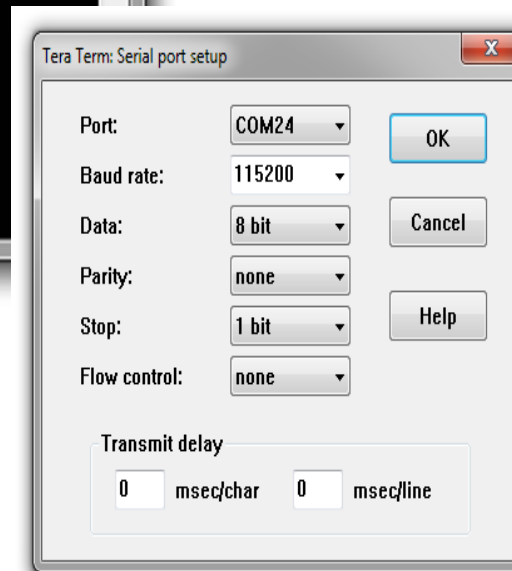
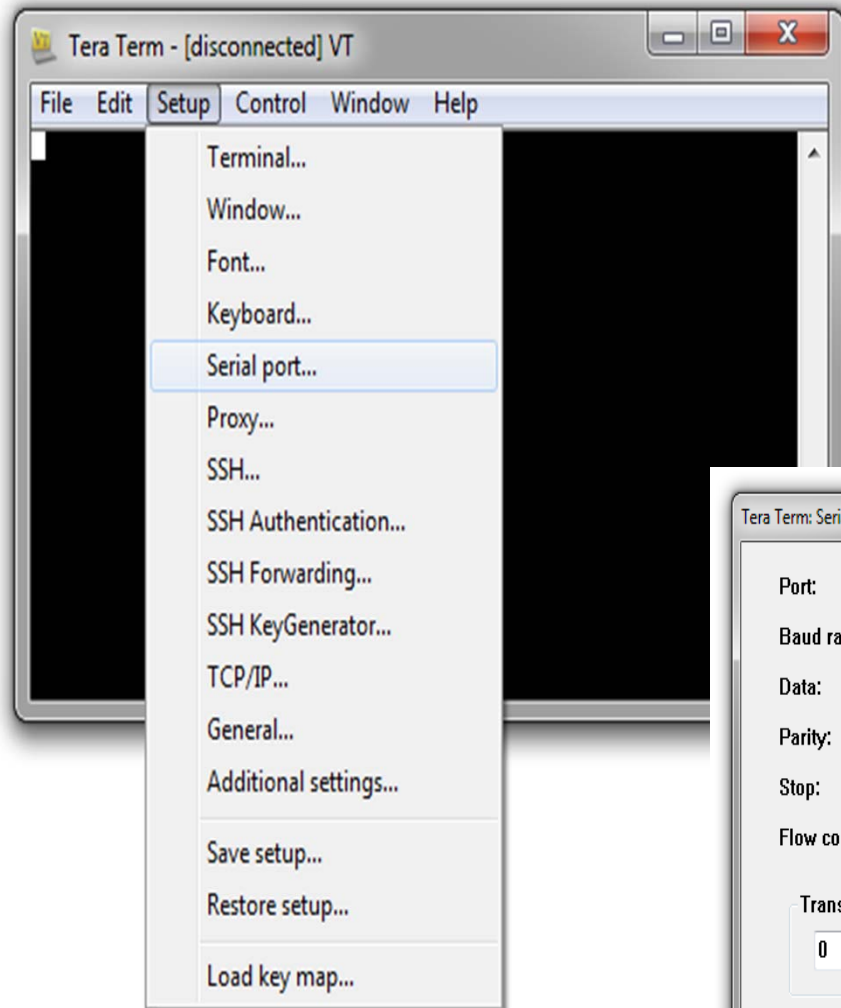
06/11/2014  01:09 PM                1,765,719 io_fsdata.h
               1 File(s)                1,765,719 bytes
               0 Dir(s)  209,722,093,568 bytes free

c:\ti\TivaWare_C_Series-2.1.0.12573\examples\boards\ek-tm4c1294xl\enet_io>
```

# 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.





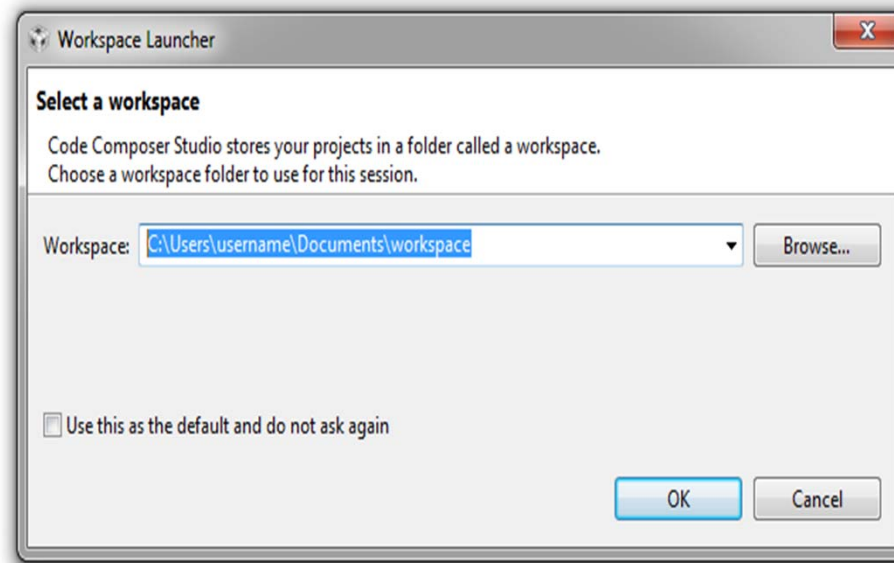
# Hands On: Using Code Composer Studio v5.5

- 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](http://www.ti.com/ccs)
- TI E2E Support Community for CCS at:  
[http://e2e.ti.com/support/development\\_tools/code\\_composer\\_studio/default.aspx](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:

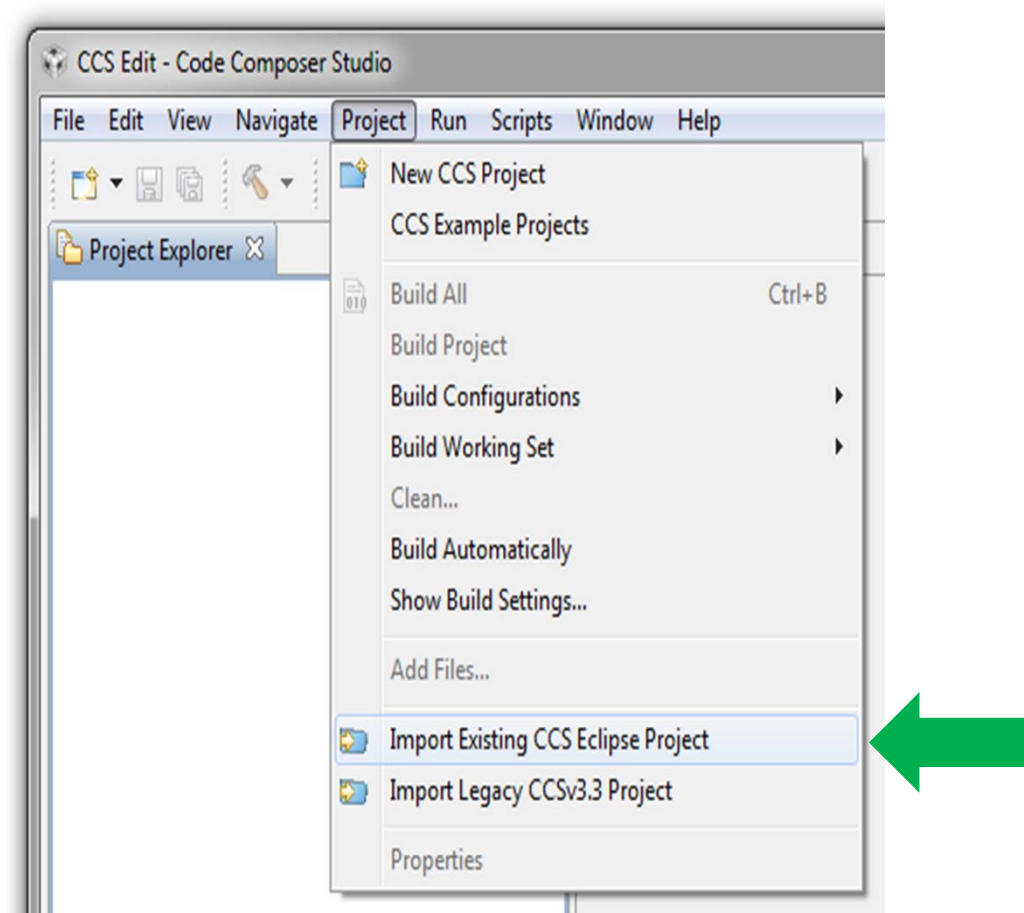
`C:\Users\<username>\Documents\workspace`



# Hands On: CCS Import existing Project

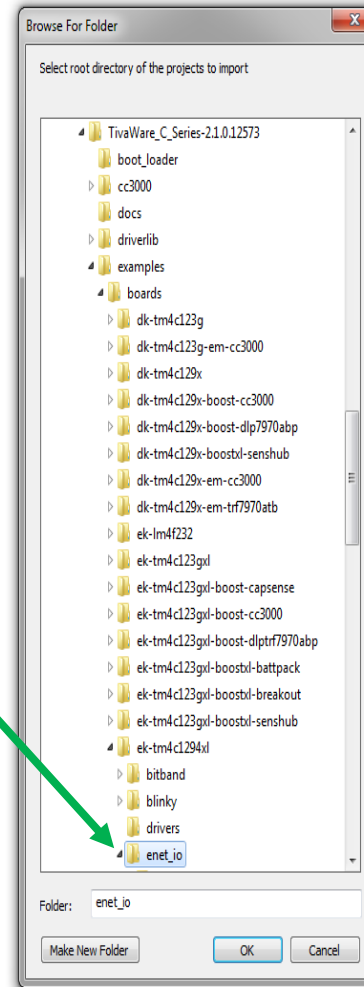
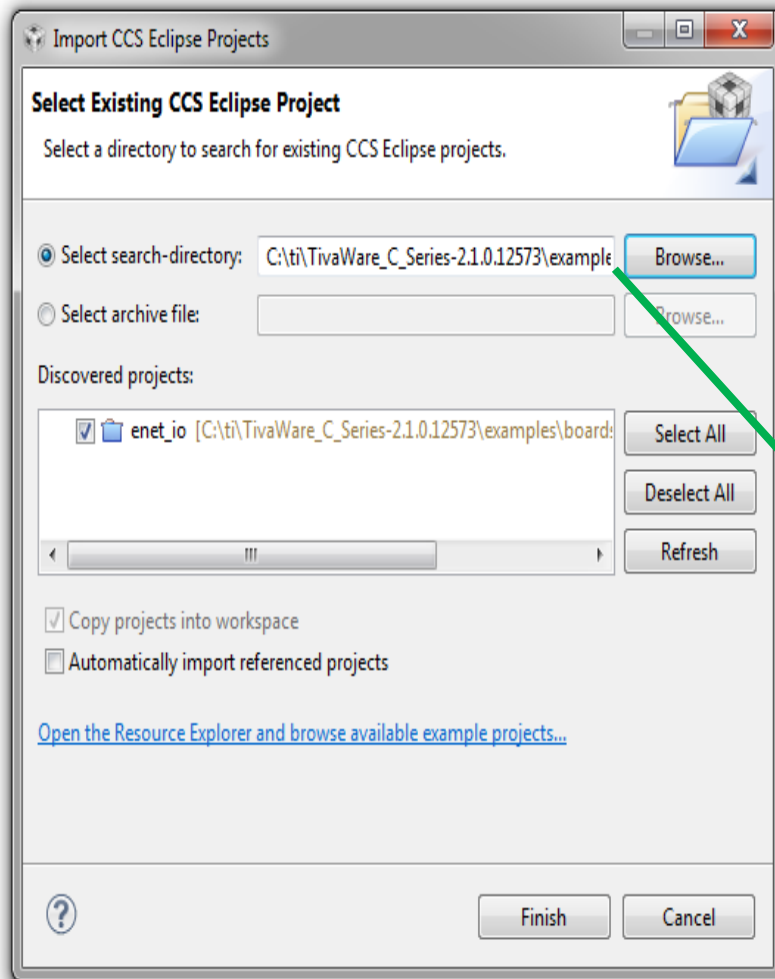
Import enet\_io from TivaWare, the path will be similar to:

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.

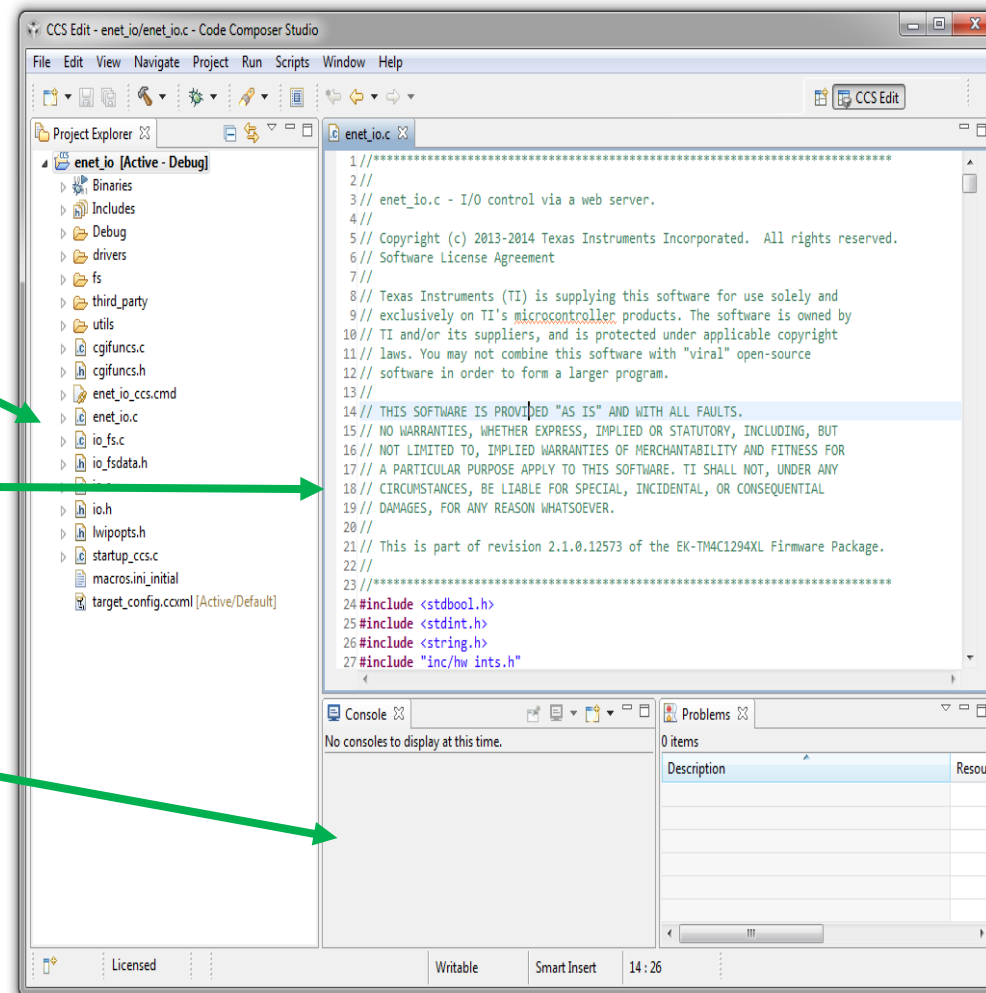
# Hands On: CCS IDE C/C++ Perspective

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

Workspace/source browser

Source editor

Console, error window



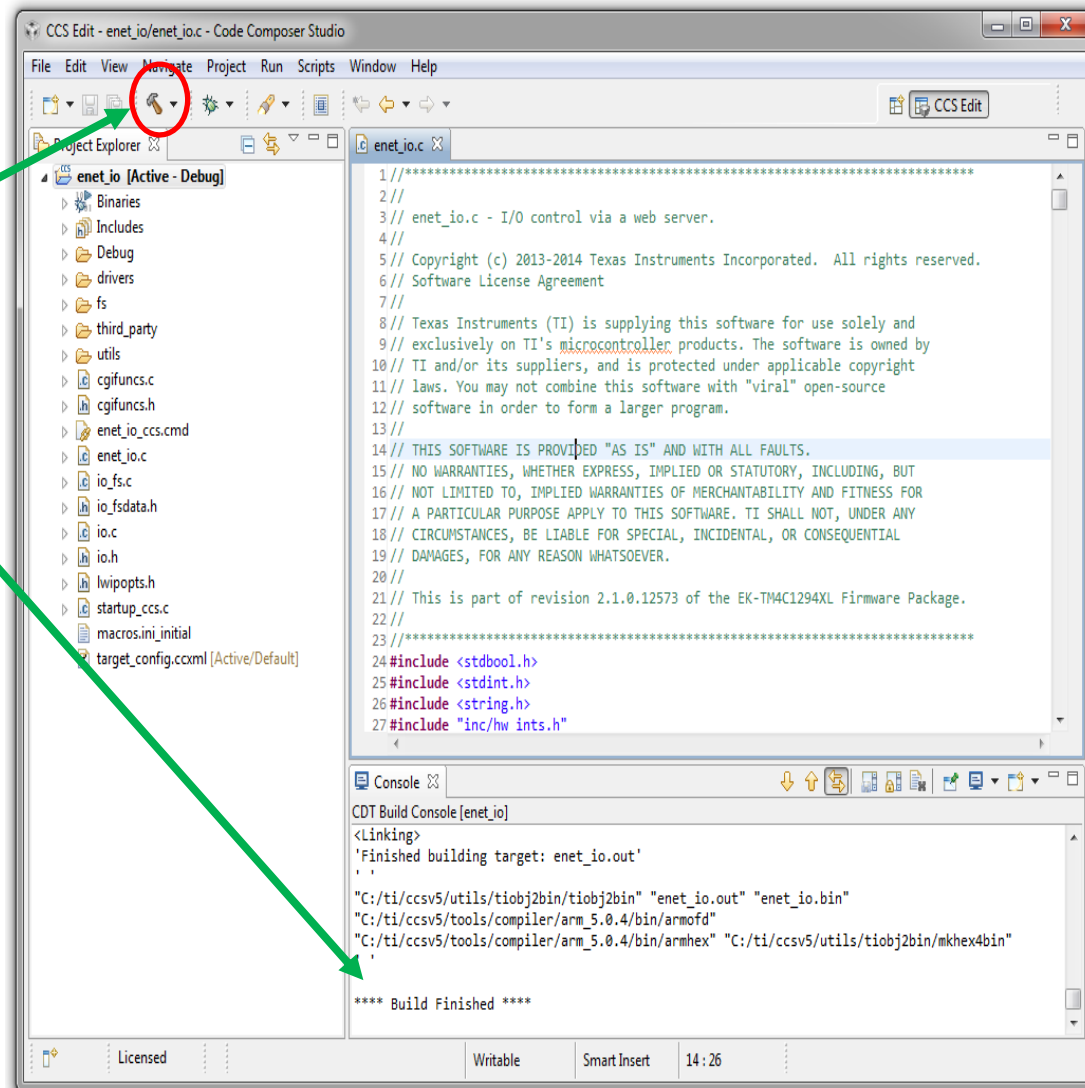
# Hands On: Issue patch in demo code

- 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.

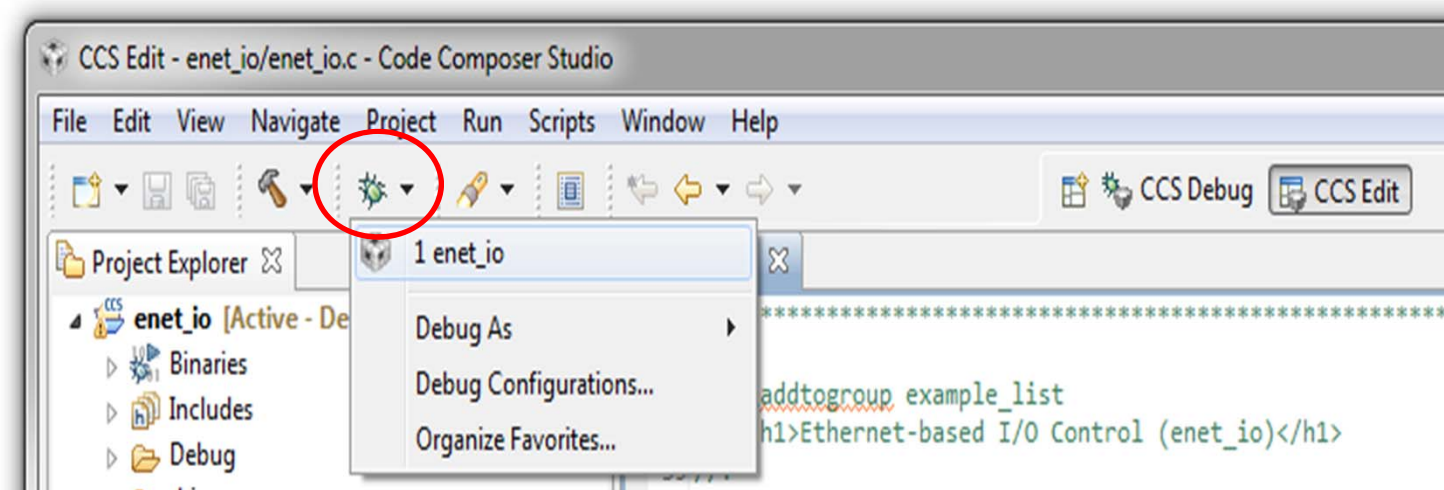
```
528     if((ui32NewIPAddress == 0) || (ui32NewIPAddress == 0xffffffff))
529     {
530         //
531         // Loop through the LED animation.
532         //
533         for(ui32Idx = 1; ui32Idx < 17; ui32Idx++)
534         {
535             //
536             // Toggle the GPIO
537            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

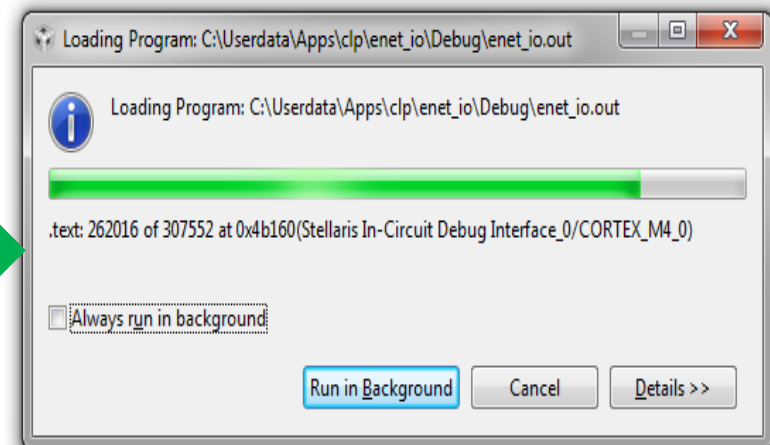


# Hands On: Program and Debug Project



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

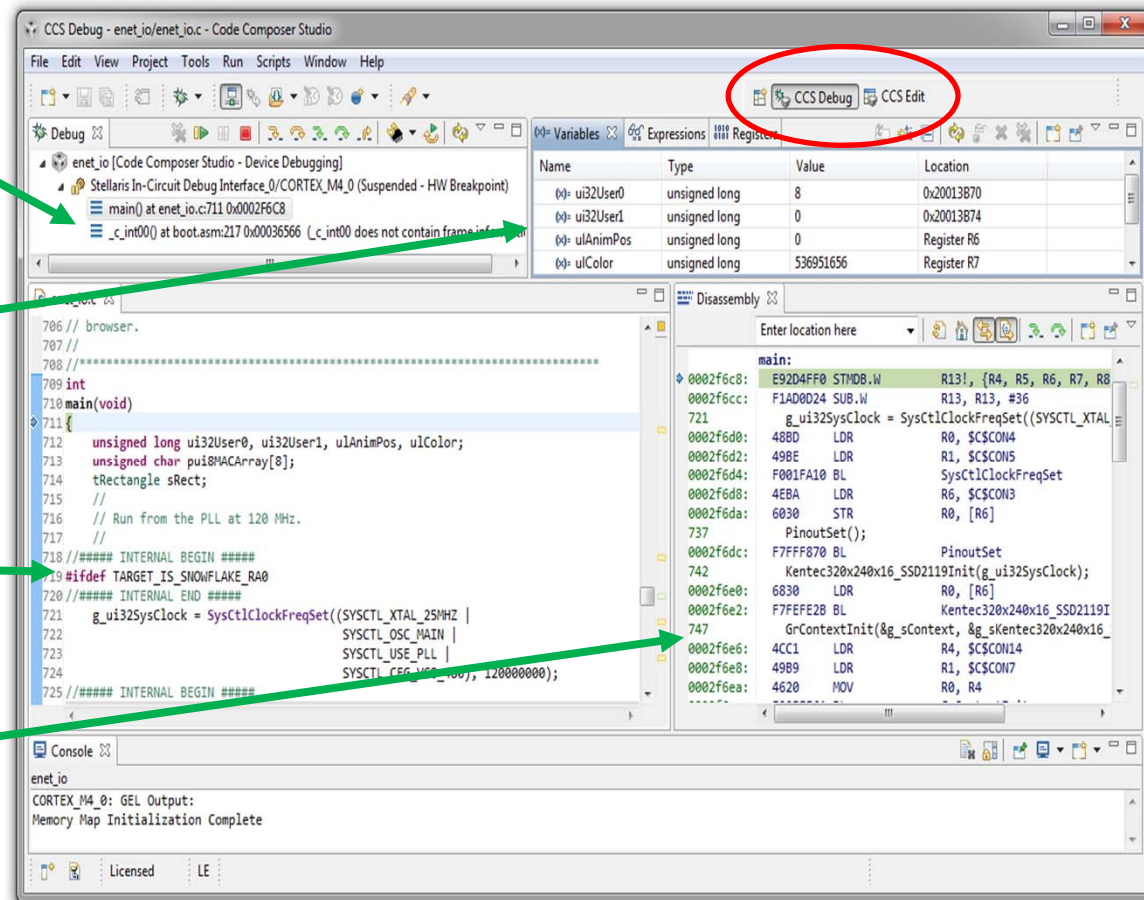
- The CCS debugger will automatically connect to your evaluation board, program the flash and run to the beginning of the main() function.

Debug thread view window

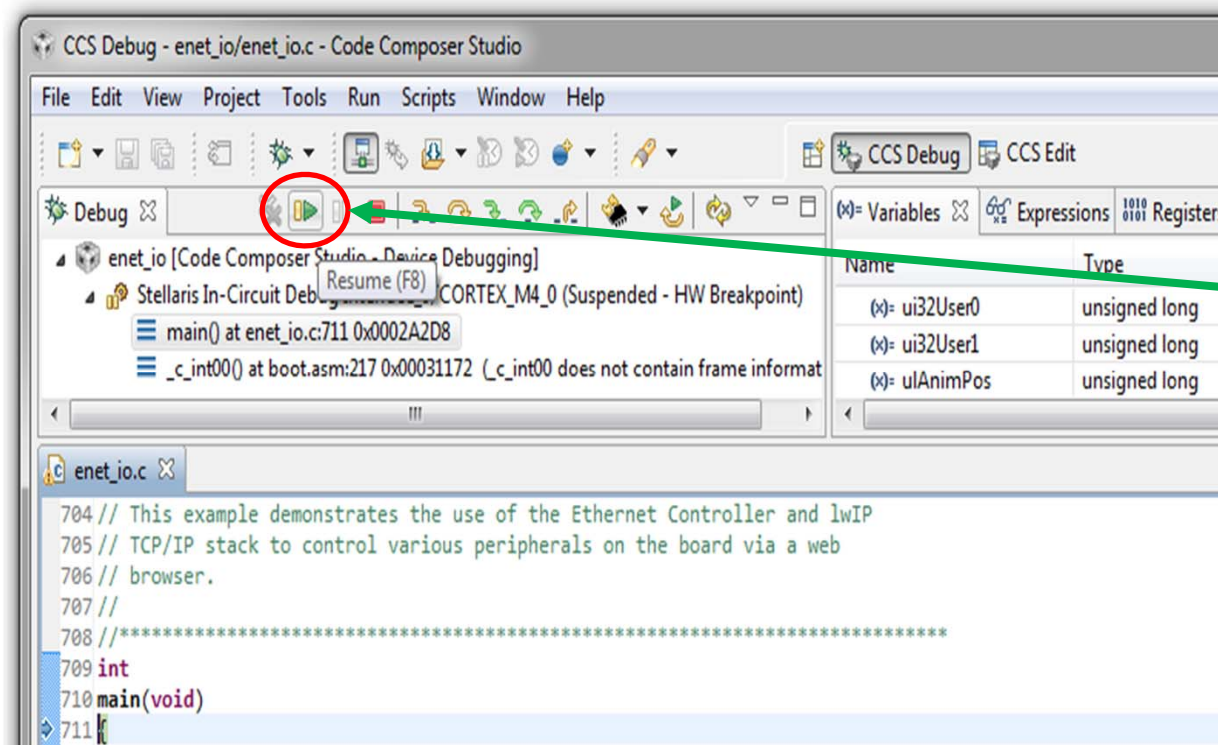
Core register, variable, breakpoint, peripheral viewers


Source/PC view

Disassembly and Memory windows

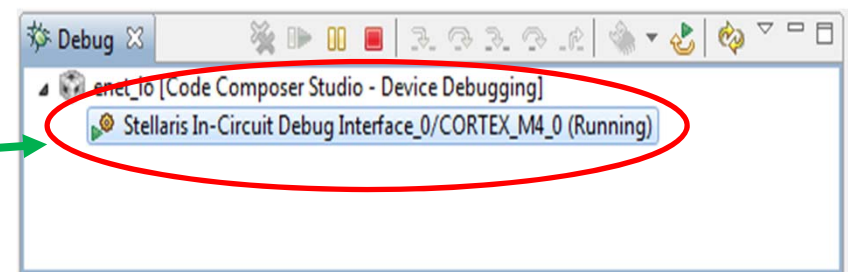


# Hands On: Program and Debug Project



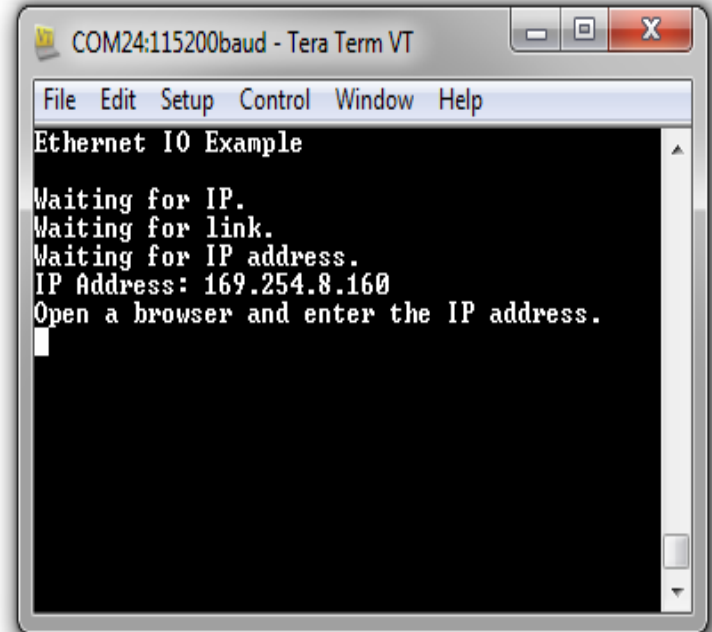
- Once you have acquainted yourself with the IDE windows, click the  icon, and the application will begin executing on the Tiva C-series core.

- 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.

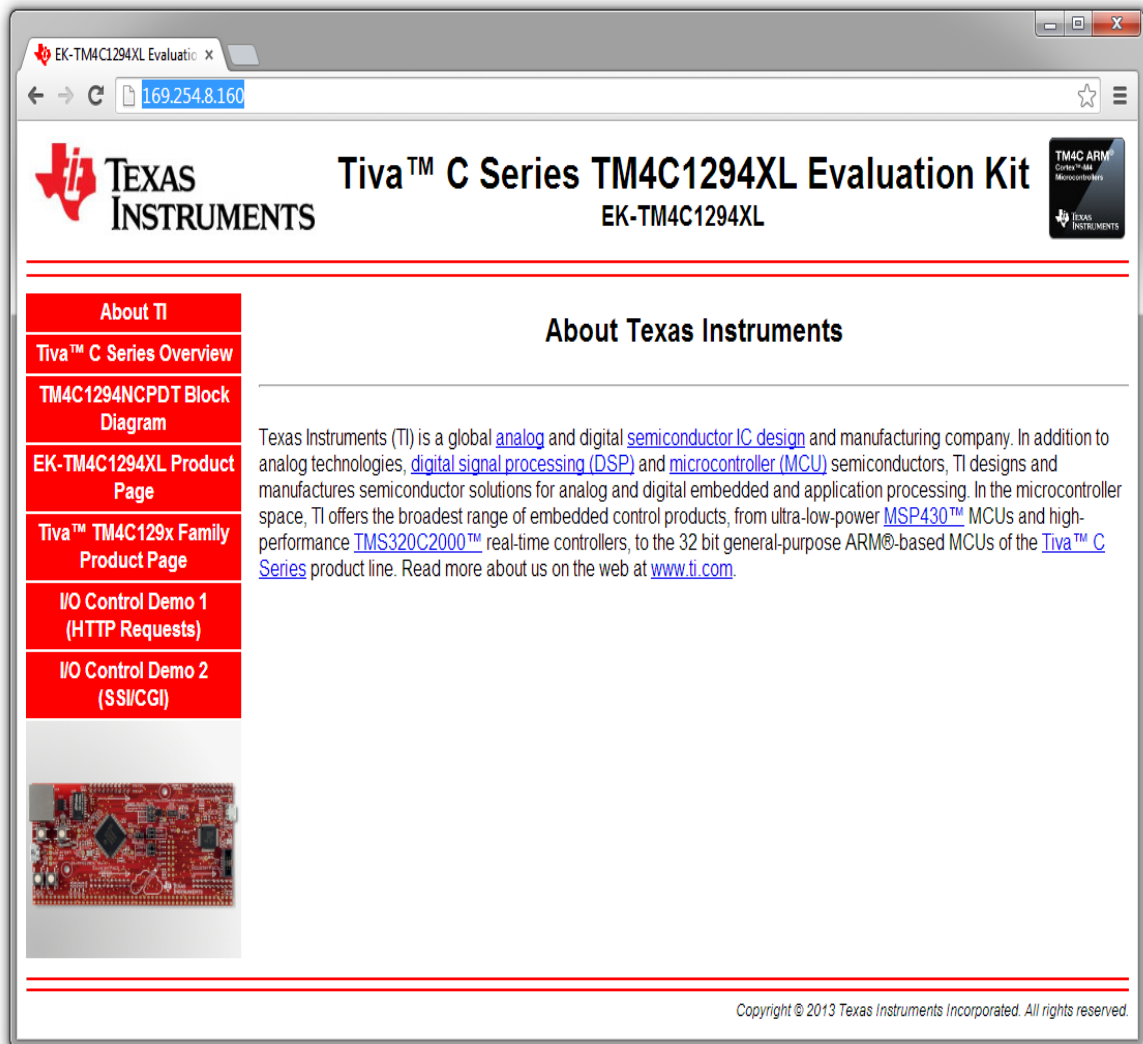


```
COM24:115200baud - Tera Term VT
File Edit Setup Control Window Help
Ethernet I/O Example
Waiting for IP.
Waiting for link.
Waiting for IP address.
IP Address: 169.254.8.160
Open a browser and enter the IP address.
█
```

# Hands On: Embedded Control Web Page

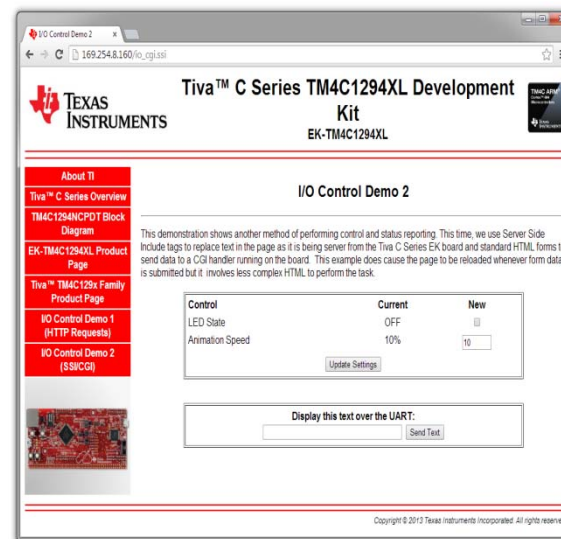
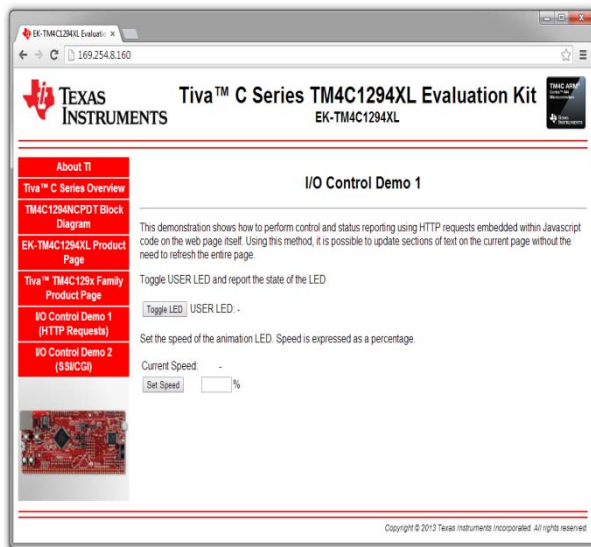
HTML code and images are stored within the application binary in FLASH memory as a file system image (io\_fs.c)

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

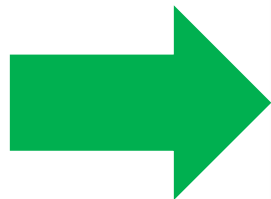


# Hands On: I/O Control Demonstrations

- 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



The screenshot shows a web browser window with the address bar displaying '169.254.8.160'. The page title is 'Tiva™ C Series TM4C1294XL Evaluation Kit EK-TM4C1294XL'. The Texas Instruments logo is in the top left. A sidebar on the left contains a list of links: 'About TI', 'Tiva™ C Series Overview', 'TM4C1294NCPDT Block Diagram', 'EK-TM4C1294XL Product Page', 'Tiva™ TM4C129x Family Product Page', 'I/O Control Demo 1 (HTTP Requests)', and 'I/O Control Demo 2 (SSI/CGI)'. The main content area is titled 'I/O Control Demo 1'. It contains a paragraph explaining that the demonstration shows how to perform control and status reporting using HTTP requests embedded within Javascript code on the web page itself. Below this, there is a section for 'Toggle USER LED and report the state of the LED' with a 'Toggle LED' button and a 'USER LED: -' display. Another section for 'Set the speed of the animation LED. Speed is expressed as a percentage.' includes a 'Current Speed: -' display and a 'Set Speed' button followed by a text input field and a '%' symbol. At the bottom of the page, there is a copyright notice: 'Copyright © 2013 Texas Instruments Incorporated. All rights reserved.'

# 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:**

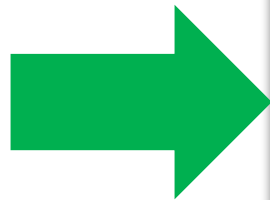
- Updates can be made without having to reload the whole web page.
- Quicker user feedback.
- Lower network traffic.

- **Disadvantages:**

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




# Hands On: Demo 2 – SSI/CGI




I/O Control Demo 2 x

169.254.8.160/io\_cgi.ssi




## Tiva™ C Series TM4C1294XL Development Kit

EK-TM4C1294XL



---

[About TI](#)[Tiva™ C Series Overview](#)[TM4C1294NCPDT Block Diagram](#)[EK-TM4C1294XL Product Page](#)[Tiva™ TM4C129x Family Product Page](#)[I/O Control Demo 1 \(HTTP Requests\)](#)[I/O Control Demo 2 \(SSI/CGI\)](#)

### I/O Control Demo 2

This demonstration shows another method of performing control and status reporting. This time, we use Server Side Include tags to replace text in the page as it is being served from the Tiva C Series EK board and standard HTML forms to send data to a CGI handler running on the board. This example does cause the page to be reloaded whenever form data is submitted but it involves less complex HTML to perform the task.

| Control         | Current | New                             |
|-----------------|---------|---------------------------------|
| LED State       | OFF     | <input type="checkbox"/>        |
| Animation Speed | 10%     | <input type="text" value="10"/> |

Display this text over the UART:

Copyright © 2013 Texas Instruments Incorporated. All rights reserved.

# 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
```