ECE 3567 Microcontrollers Lab

Lecture #3 – Code Composer Studio v 8.1.0

Autumn 2019
Dr. Gregg Chapman

Project Set-up

Before doing ANYTHING, make a folder on the **U: Drive** that will be your Workspace.

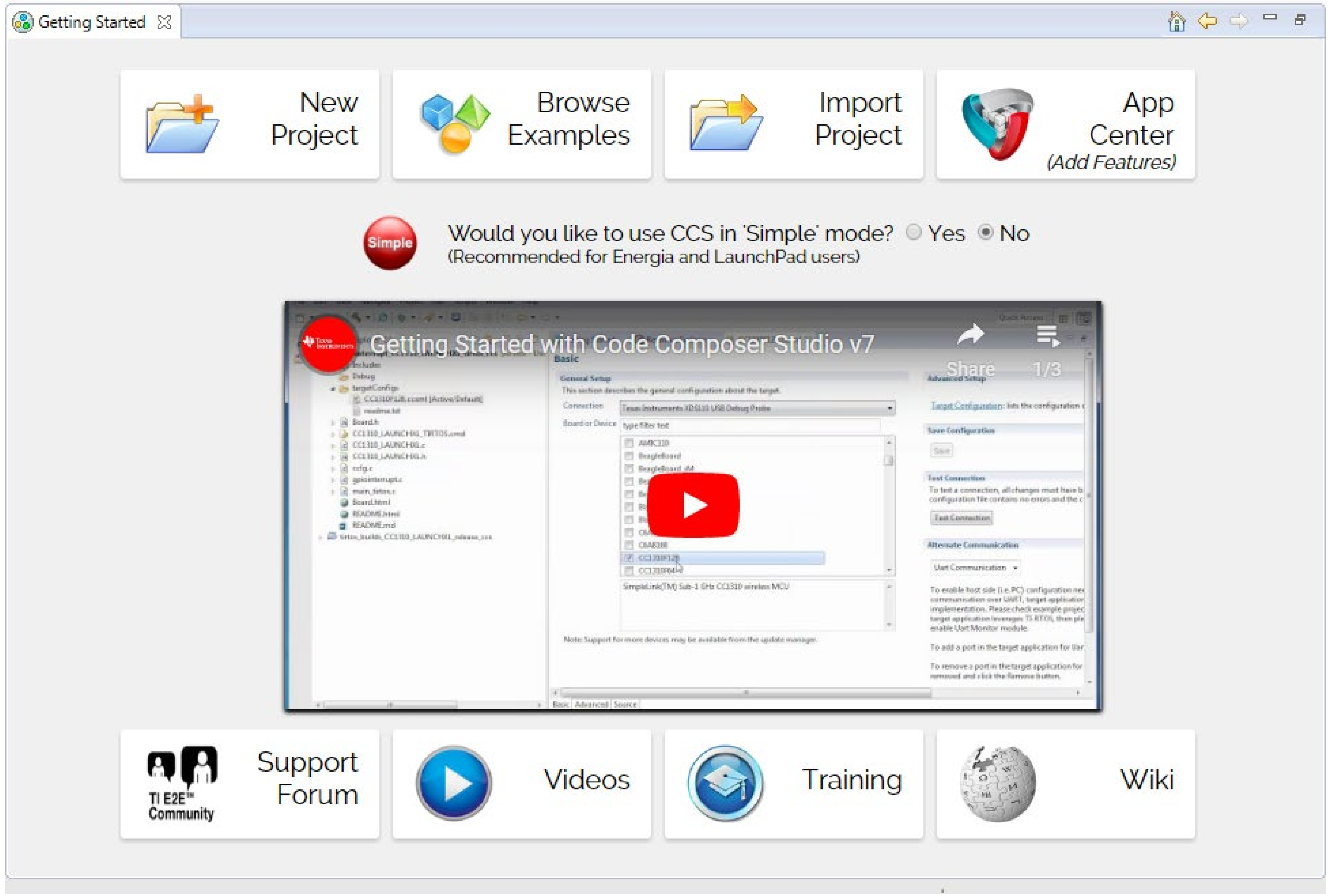
Never place more than one project in the same workspace.

Project Set-up

Invoke the Code Composer Studio version 8.1.0 software by double clicking on the CCS ICON

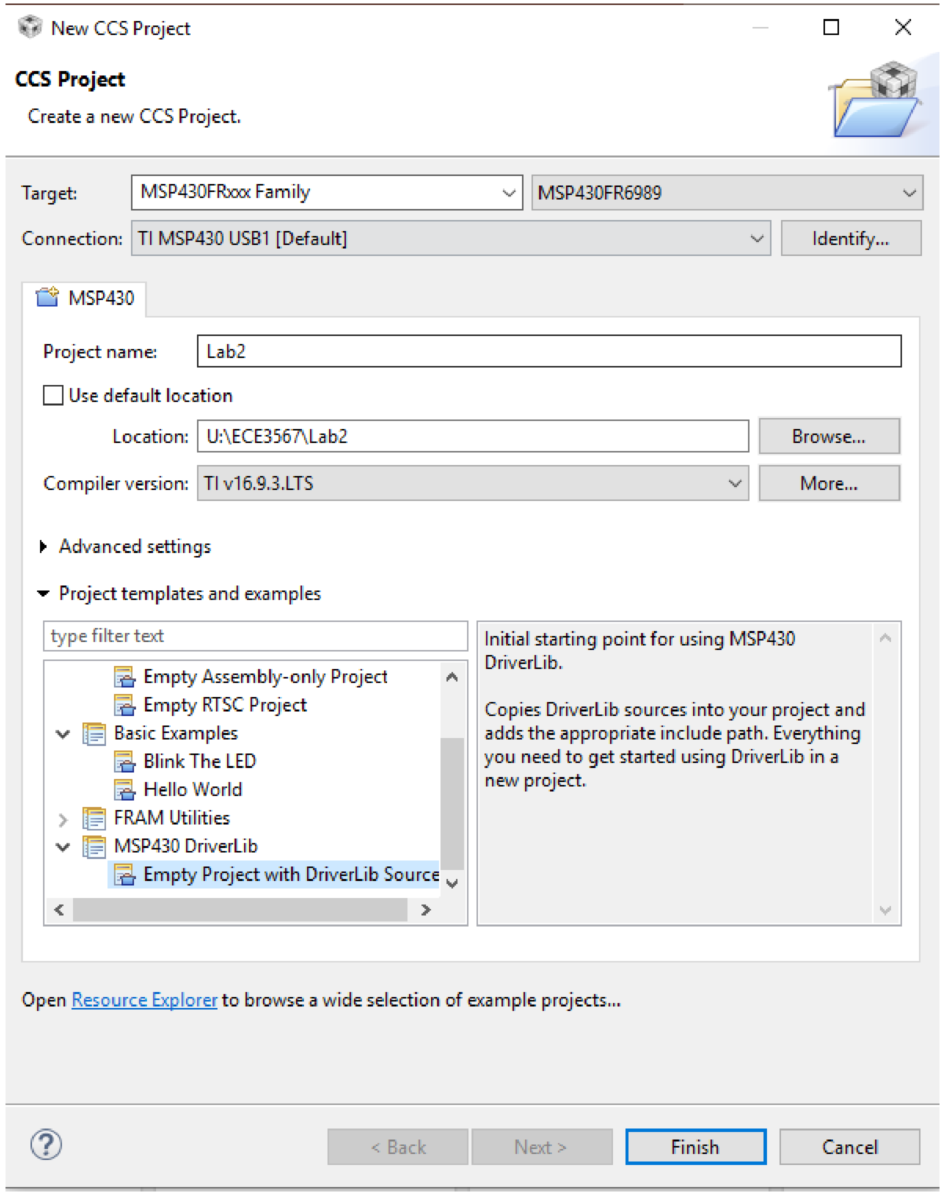
Project Set-up

If you are beginning with the Getting Started screen, select **New Project:**


Getting Started
New Project
Browse Examples
Import Project
App Center (Add Features)
Simple
Would you like to use CCS in 'Simple' mode? Yes [✓ No]
(Recommended for Energia and LaunchPad users)
Getting Started with Code Composer Studio v7
Share
1/3
Support Forum
TI E2E™ Community
Videos
Training
Wiki

If only Project Explorer is open, select **View → Getting Started**

Project Set-up


New CCS Project

CCS Project

Create a new CCS Project.

Target: MSP430FRxxx Family | MSP430FR6989

Connection: TI MSP430 USB1 [Default] | Identify...

MSP430

Project name: Lab2

☐ Use default location

Location: U:\ECE3567\Lab2 | Browse...

Compiler version: TI v16.9.3.LTS | More...

Advanced settings

Project templates and examples

type filter text

- Empty Assembly-only Project
- Empty RTSC Project
- Basic Examples
  - Blink The LED
  - Hello World
- FRAM Utilities
- MSP430 DriverLib
  - Empty Project with DriverLib Source

Initial starting point for using MSP430 DriverLib.

Copies DriverLib sources into your project and adds the appropriate include path. Everything you need to get started using DriverLib in a new project.

Open Resource Explorer to browse a wide selection of example projects...

< Back | Next > | Finish | Cancel

In the **CCS Project** window, configure the following fields:

1. Target: **MSP430FRxxx** Family
2. **MSP430FR6989**
3. Connection: Leave at Default
4. Project Name (e.g. Lab 2)
5. Uncheck use default location and browse to your workspace (e.g. U:\ECE3567\Lab2)

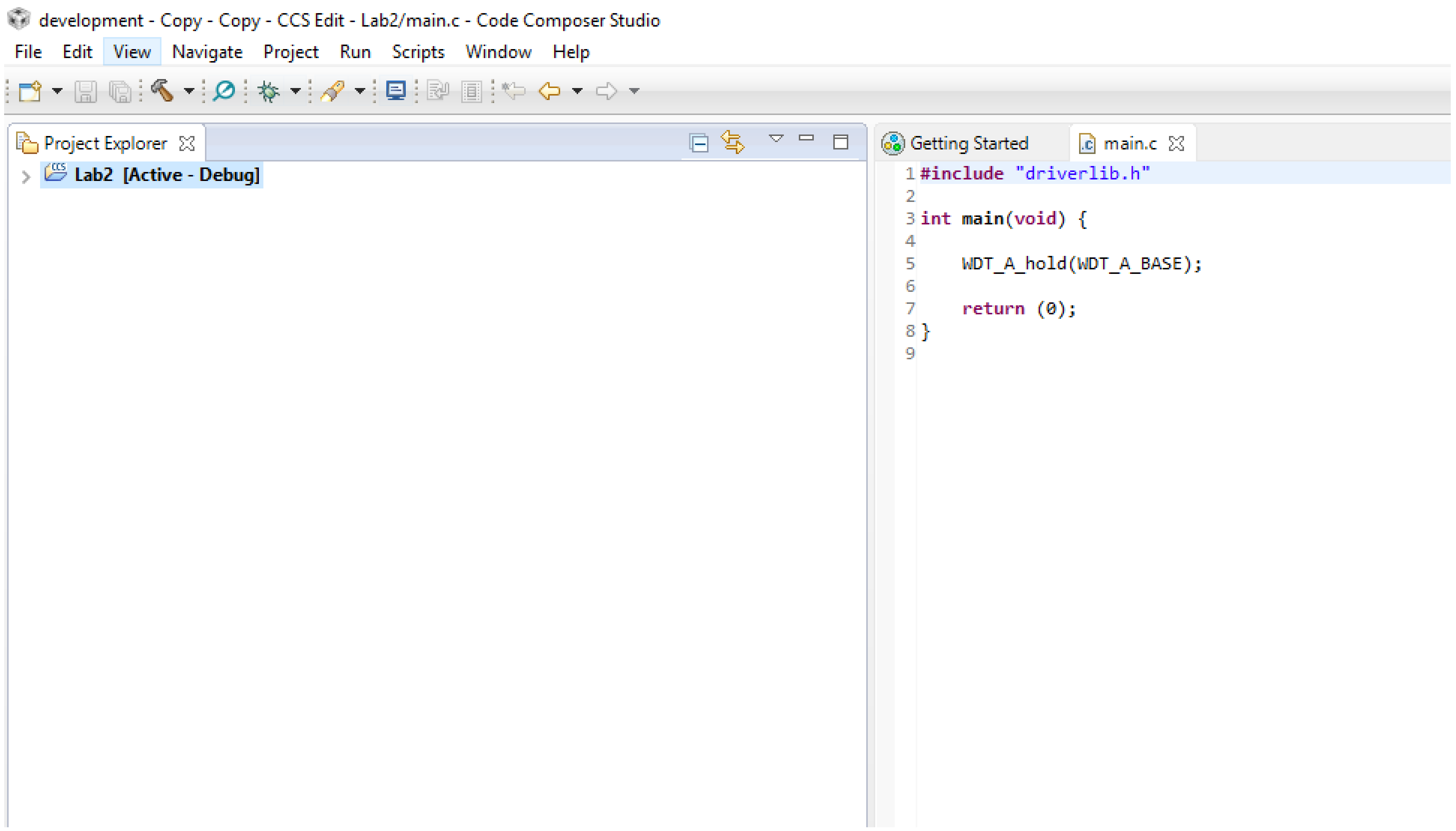
6. Leave Compiler at Default

7. In Project templates and examples, you must ***SCROLL DOWN*** to **MSP430 Driverlib**, select the down arrow ˅ and ***SCROLL DOWN AGAIN.*** Highlight:
   Empty Project with DriverLib Source

8. Select Finish

Project Set-up

"Voilà"


development - Copy - Copy - CCS Edit - Lab2/main.c - Code Composer Studio
File Edit View Navigate Project Run Scripts Window Help
Project Explorer
Lab2 [Active - Debug]
Getting Started
main.c

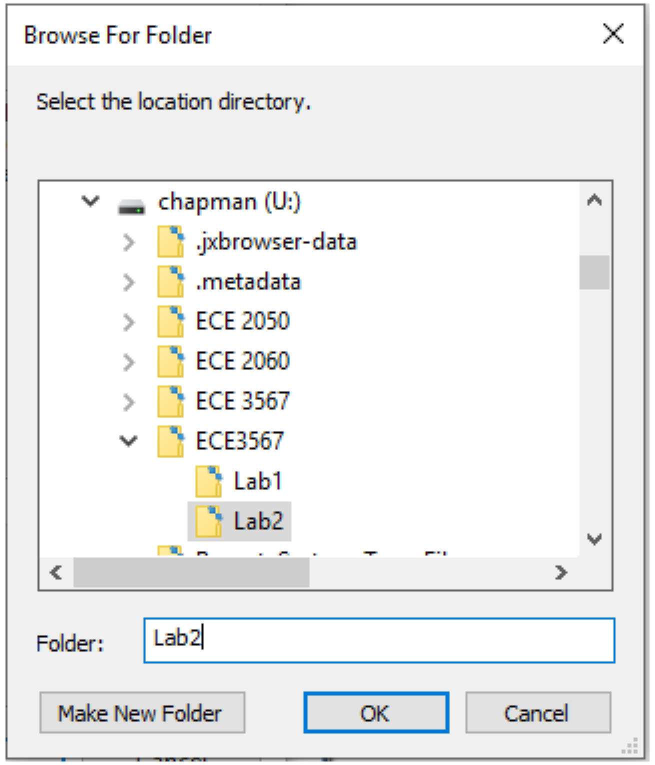
```c
#include "driverlib.h"

int main(void) {

    WDT_A_hold(WDT_A_BASE);

    return (0);
}

```

Project Set-up

ALTERNATIVELY: If you are beginning with another project already open in Project Explorer:

1. Select **File → Switch Workspace** and navigate to your new workspace

2. Select **File → New → Project → Code Composer Studio → CCS Project**

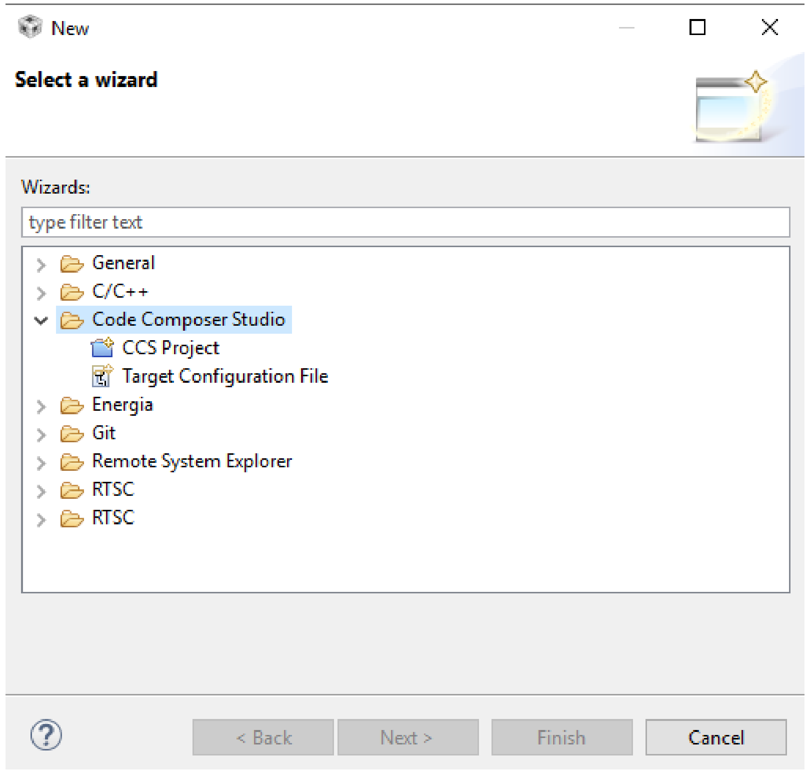


Browse For Folder

Select the location directory.

chapman (U:)
.jxbrowser-data
.metadata
ECE 2050
ECE 2060
ECE 3567
ECE3567
Lab1
Lab2

Folder: Lab2



New

Select a wizard

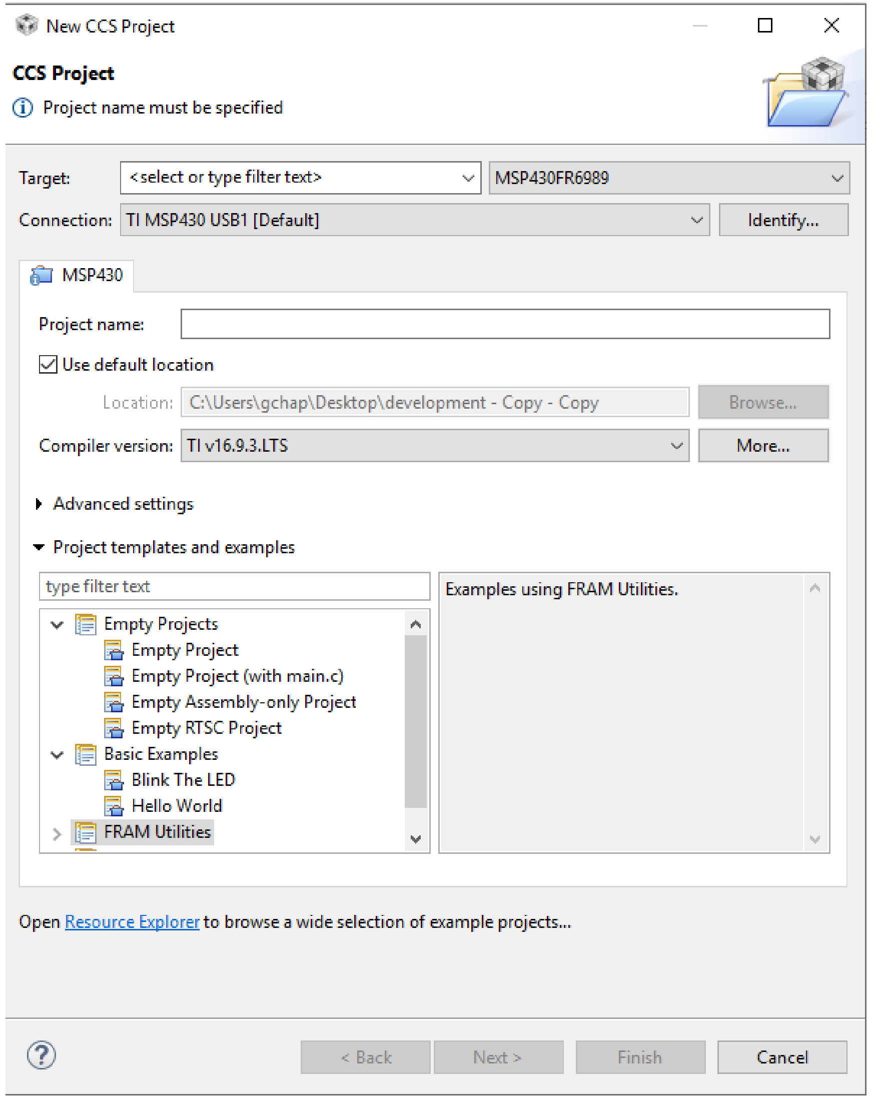
Wizards:

type filter text

General
C/C++
Code Composer Studio
CCS Project
Target Configuration File
Energia
Git
Remote System Explorer
RTSC
RTSC

Project Set-up

3. This should open the **New CSS Project** window. Proceed as previously described.



New CCS Project

CCS Project

Project name must be specified

Target: <select or type filter text> MSP430FR6989

Connection: TI MSP430 USB1 [Default] Identify...

MSP430

Project name:

☑ Use default location

Location: C:\Users\gchap\Desktop\development - Copy - Copy Browse...

Compiler version: TI v16.9.3.LTS More...

Advanced settings

Project templates and examples

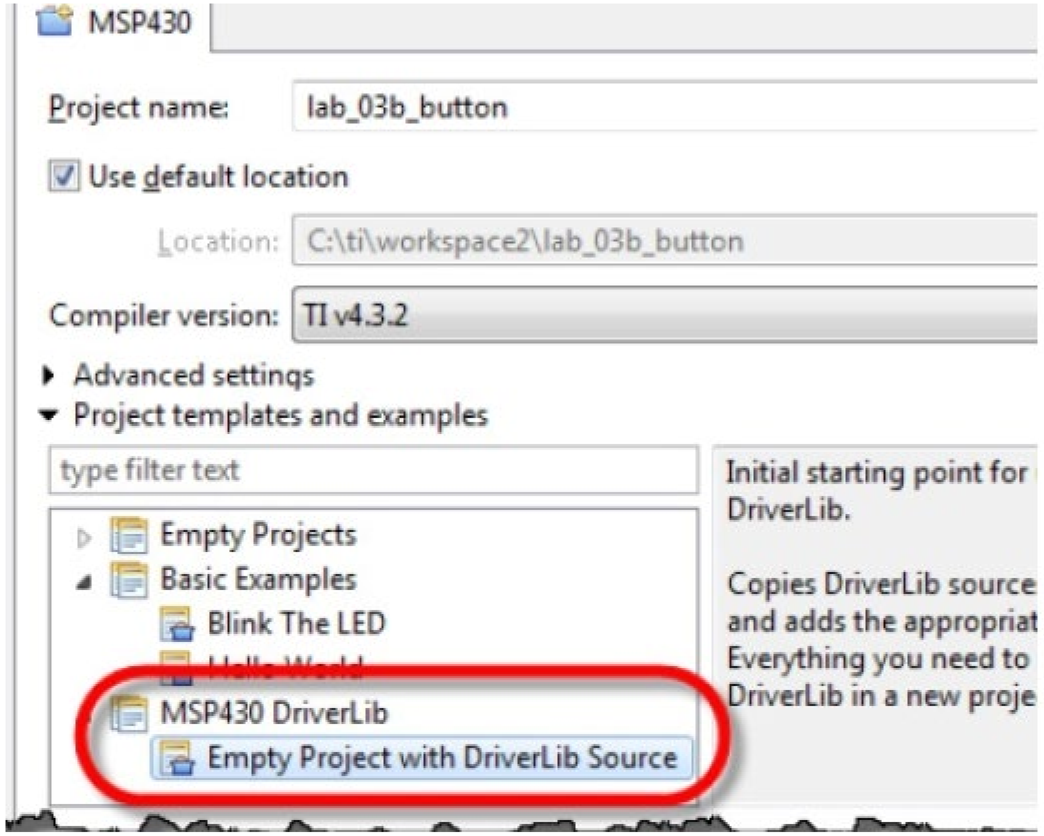
type filter text

- Empty Projects
  - Empty Project
  - Empty Project (with main.c)
  - Empty Assembly-only Project
  - Empty RTSC Project
- Basic Examples
  - Blink The LED
  - Hello World
- FRAM Utilities

Examples using FRAM Utilities.

Open Resource Explorer to browse a wide selection of example projects...

< Back Next > Finish Cancel



MSP430

Project name: lab_03b_button

☑ Use default location

Location: C:\ti\workspace2\lab_03b_button

Compiler version: TI v4.3.2

Advanced settings

Project templates and examples

type filter text

- Empty Projects
- Basic Examples
  - Blink The LED
  - Hello World
- MSP430 DriverLib
  - Empty Project with DriverLib Source

Initial starting point for DriverLib.

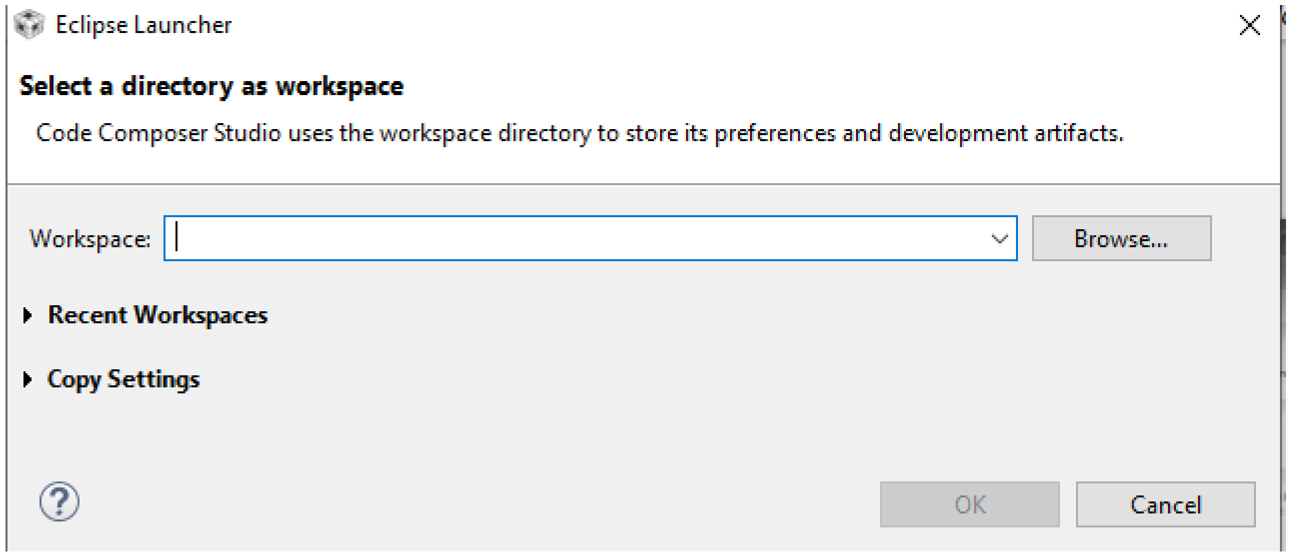
Copies DriverLib source and adds the appropriat Everything you need to DriverLib in a new proje

Don't Forget to select this template.

Project Set-up

The FIRST time you use the workspace, CCS MAY open the Eclipse Launcher.

Browse to the workspace that you have created, then select OK



Eclipse Launcher

Select a directory as workspace

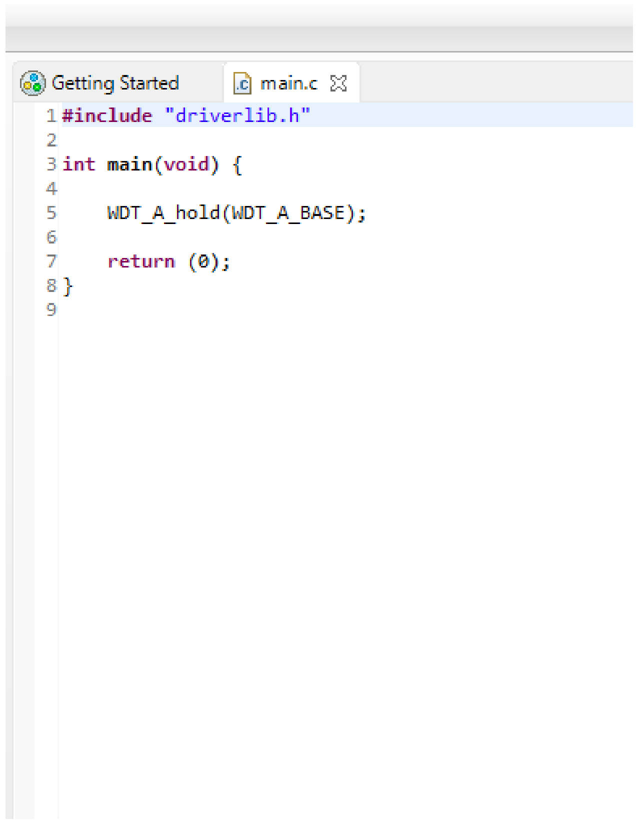
Code Composer Studio uses the workspace directory to store its preferences and development artifacts.

Workspace:

Code Composer Studio – Adding Code

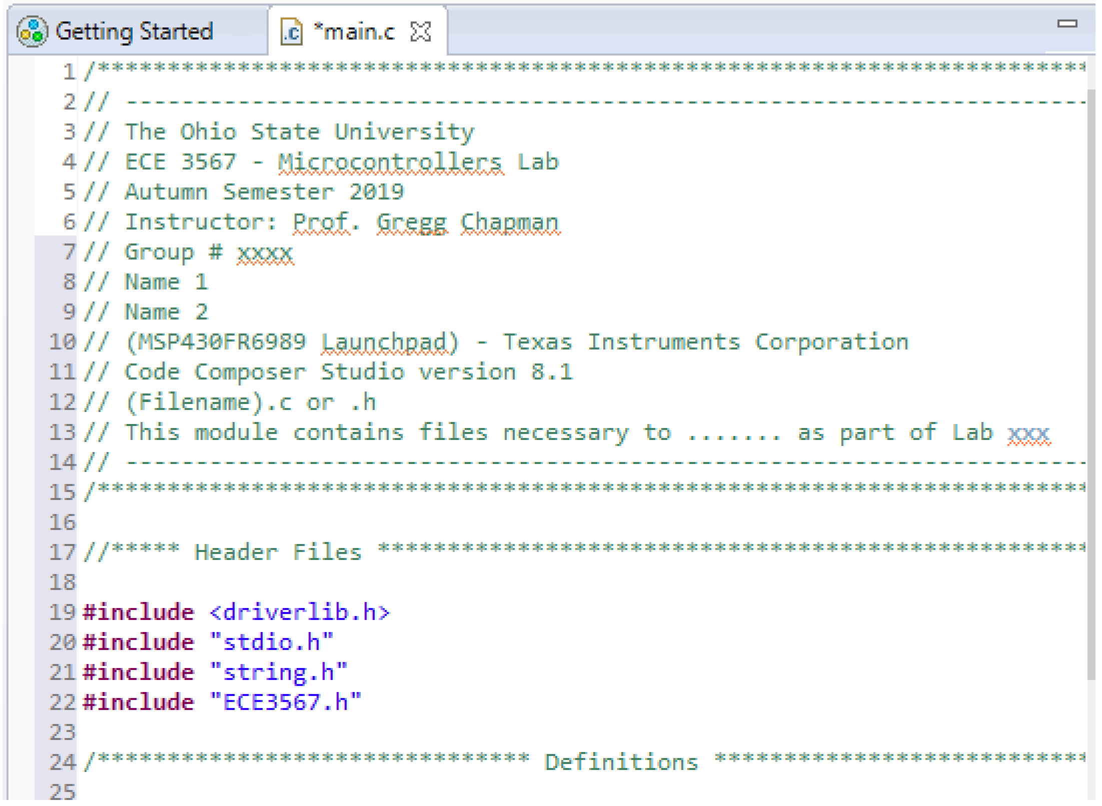
1. Select File → New → Source File
2. Enter Code
3. Save the File
4. Select Project → Rebuild Project

5. You may also COPY files into your project folder then add them to the project with a Right-Click on your **Project Name [Active-Debug] .** Then select **Add Files . . .**



```c
#include "driverlib.h"

int main(void) {

    WDT_A_hold(WDT_A_BASE);

    return (0);
}

```

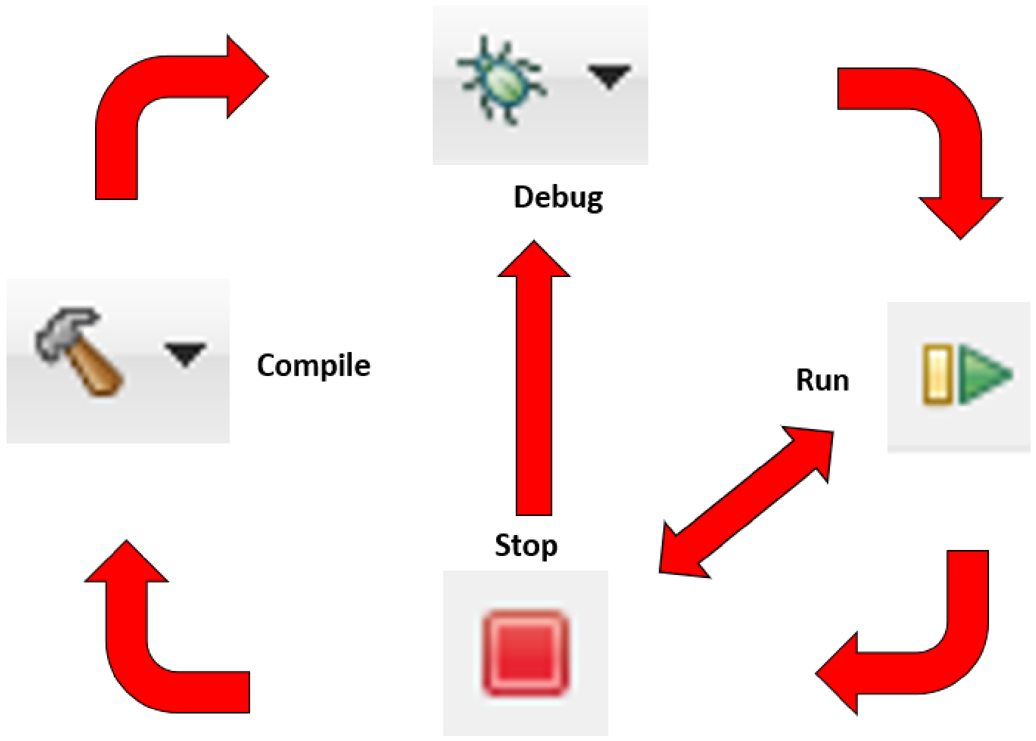
NOTE: It's better to copy in the standard file header and edit it than starting with the default main.c



```c
/*********************************************************************
// -------------------------------------------------------------------
// The Ohio State University
// ECE 3567 - Microcontrollers Lab
// Autumn Semester 2019
// Instructor: Prof. Gregg Chapman
// Group # xxxx
// Name 1
// Name 2
// (MSP430FR6989 Launchpad) - Texas Instruments Corporation
// Code Composer Studio version 8.1
// (Filename).c or .h
// This module contains files necessary to ....... as part of Lab xxx
// -------------------------------------------------------------------
/*********************************************************************

//***** Header Files *************************************************

#include <driverlib.h>
#include "stdio.h"
#include "string.h"
#include "ECE3567.h"

/******************************** Definitions ****************************

```

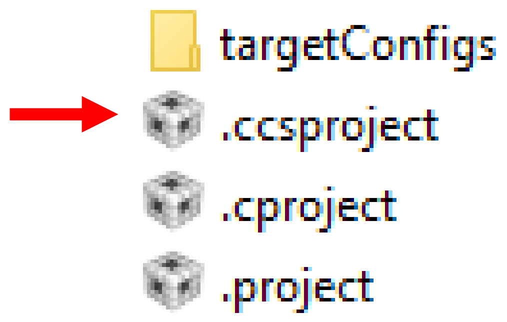
Code Composer Studio – Running the Project Code

1. At this point it is essential to connect the hardware
2. Make sure that the Project is selected as [Active – Debug]
3. You can check to see if the code compiles by selecting the hammer ICON
4. Select the Debug ICON (NOTE: This will also recompile the project)
5. Once the GREEN ARROW comes up you can run the code
6. Halt execution with the RED SQUARE



Code Composer Studio – To Open an Existing Project (Automatic Method)

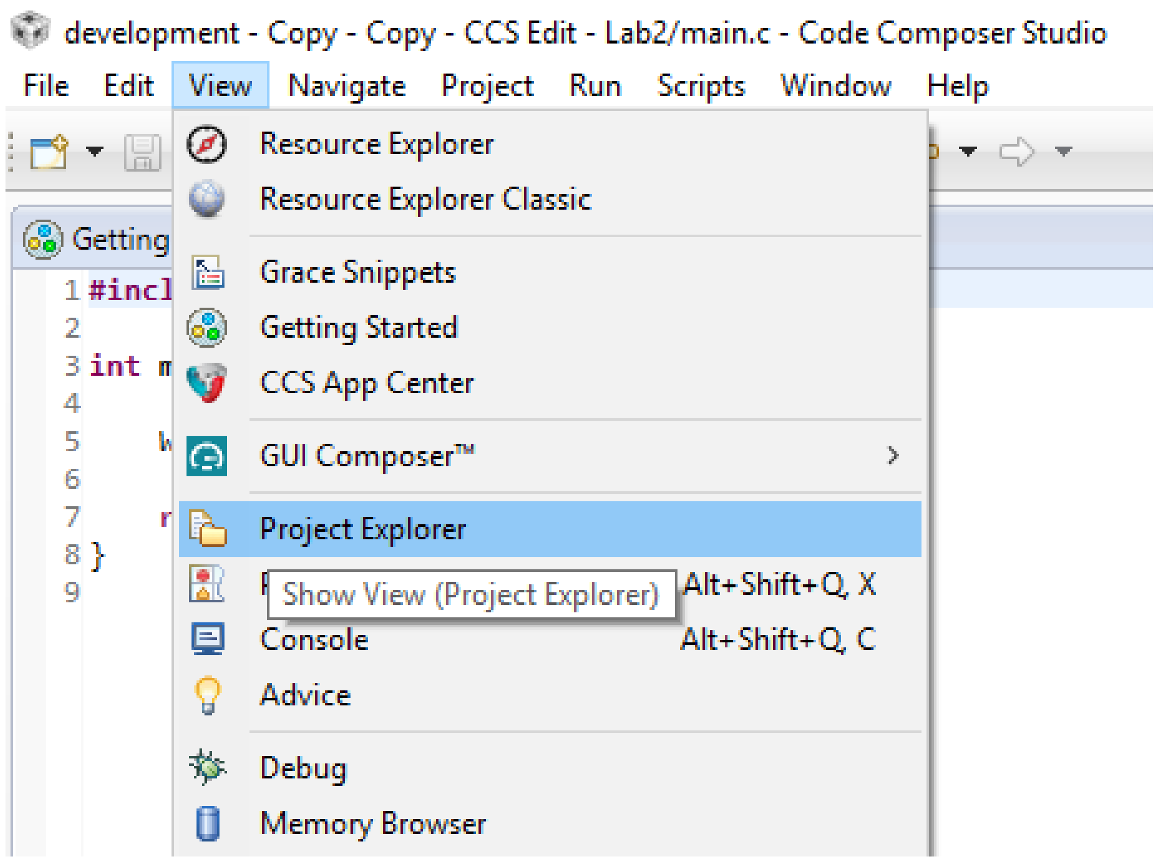
- This doesn’t always work
- Make sure that the Workspace is set to your project location. To change it, select **File → Switch Workspace**, and navigate to the project location
- Double-click the **.ccsproject** ICON in the project folder
- If it doesn’t work, try the Manual Method



Code Composer Studio – To Open an Existing Project (Manual Method)

If your project does not appear

- Select **View → Project Explorer**

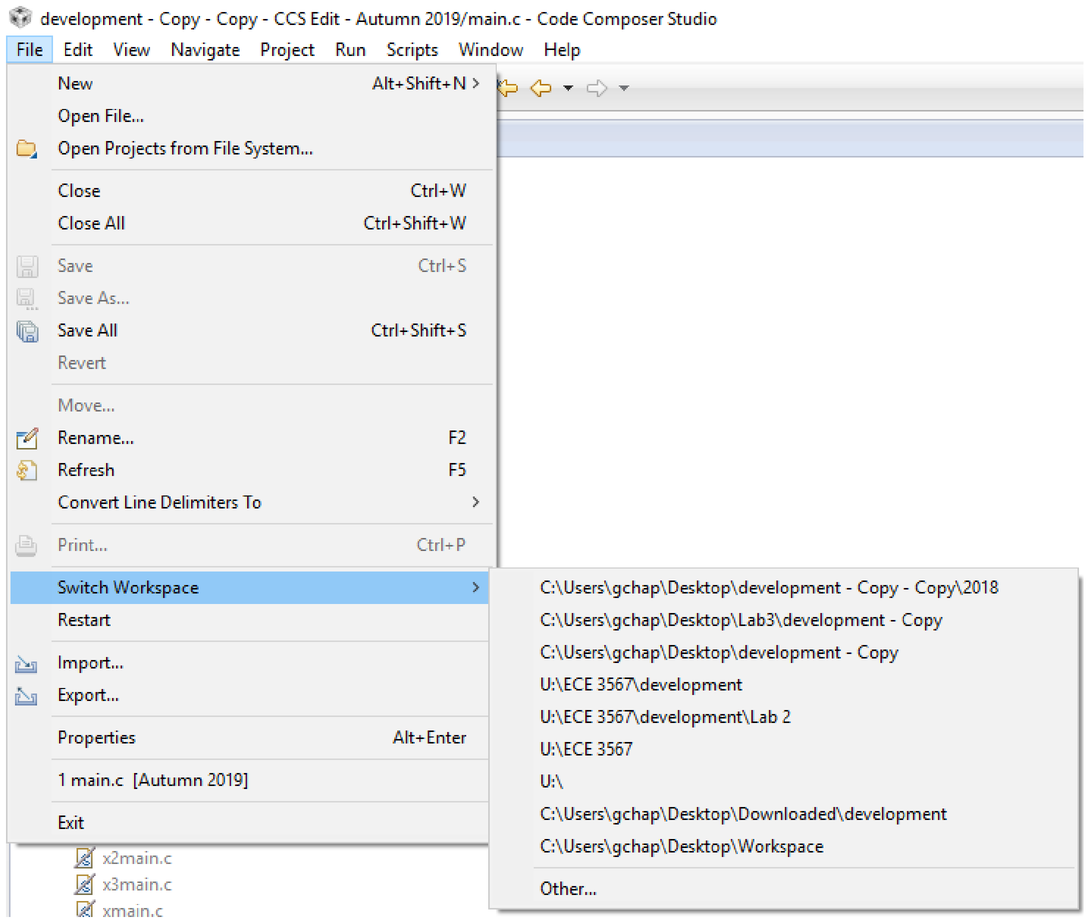

development - Copy - Copy - CCS Edit - Lab2/main.c - Code Composer Studio

```
#incl

int m

    W

    r
}
```

Code Composer Studio – To Open an Existing Project

If you still don't see anything in Project Explorer

- Select **File → Switch Workspace**, and navigate to the project location

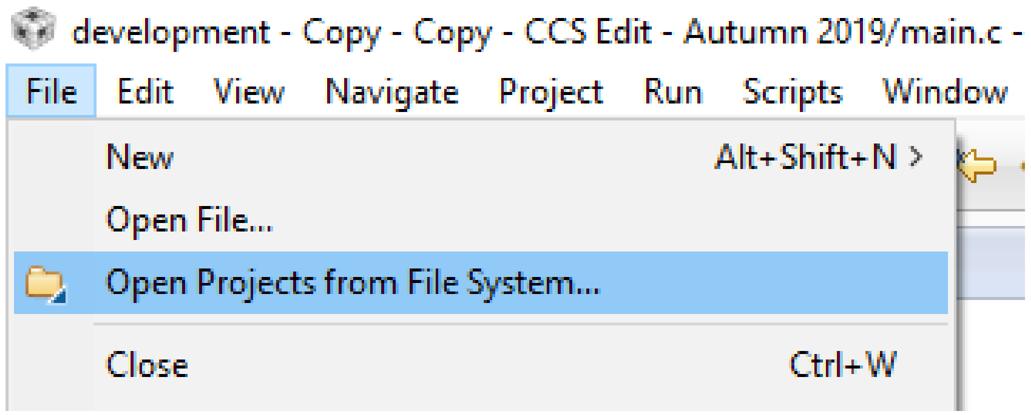


development - Copy - Copy - CCS Edit - Autumn 2019/main.c - Code Composer Studio
File
Edit
View
Navigate
Project
Run
Scripts
Window
Help
New Alt+Shift+N
Open File...
Open Projects from File System...
Close Ctrl+W
Close All Ctrl+Shift+W
Save Ctrl+S
Save As...
Save All Ctrl+Shift+S
Revert
Move...
Rename... F2
Refresh F5
Convert Line Delimiters To
Print... Ctrl+P
Switch Workspace
Restart
Import...
Export...
Properties Alt+Enter
1 main.c [Autumn 2019]
Exit
C:\Users\gchap\Desktop\development - Copy - Copy\2018
C:\Users\gchap\Desktop\Lab3\development - Copy
C:\Users\gchap\Desktop\development - Copy
U:\ECE 3567\development
U:\ECE 3567\development\Lab 2
U:\ECE 3567
U:\
C:\Users\gchap\Desktop\Downloaded\development
C:\Users\gchap\Desktop\Workspace
Other...
x2main.c
x3main.c
xmain.c

Code Composer Studio – To Open an Existing Project

If you STILL don’t see your project in Project Explorer, re-open the file:

- **File → Open Project Files from File System . . .**


development - Copy - Copy - CCS Edit - Autumn 2019/main.c -
File
Edit
View
Navigate
Project
Run
Scripts
Window
New Alt+Shift+N
Open File...
Open Projects from File System...
Close Ctrl+W

Transferring a Project to a New Location

If you move the location of your CCS project files , you must do **two things** before your project will function normally:

1. Change the Workspace

2. Update the path for the Include Options paths under the Complier settings

Change Workspace

1. Select **File → Switch Workspace → Other**

2. Enter the new workspace. Always select the file ABOVE your Project file that CCS created.

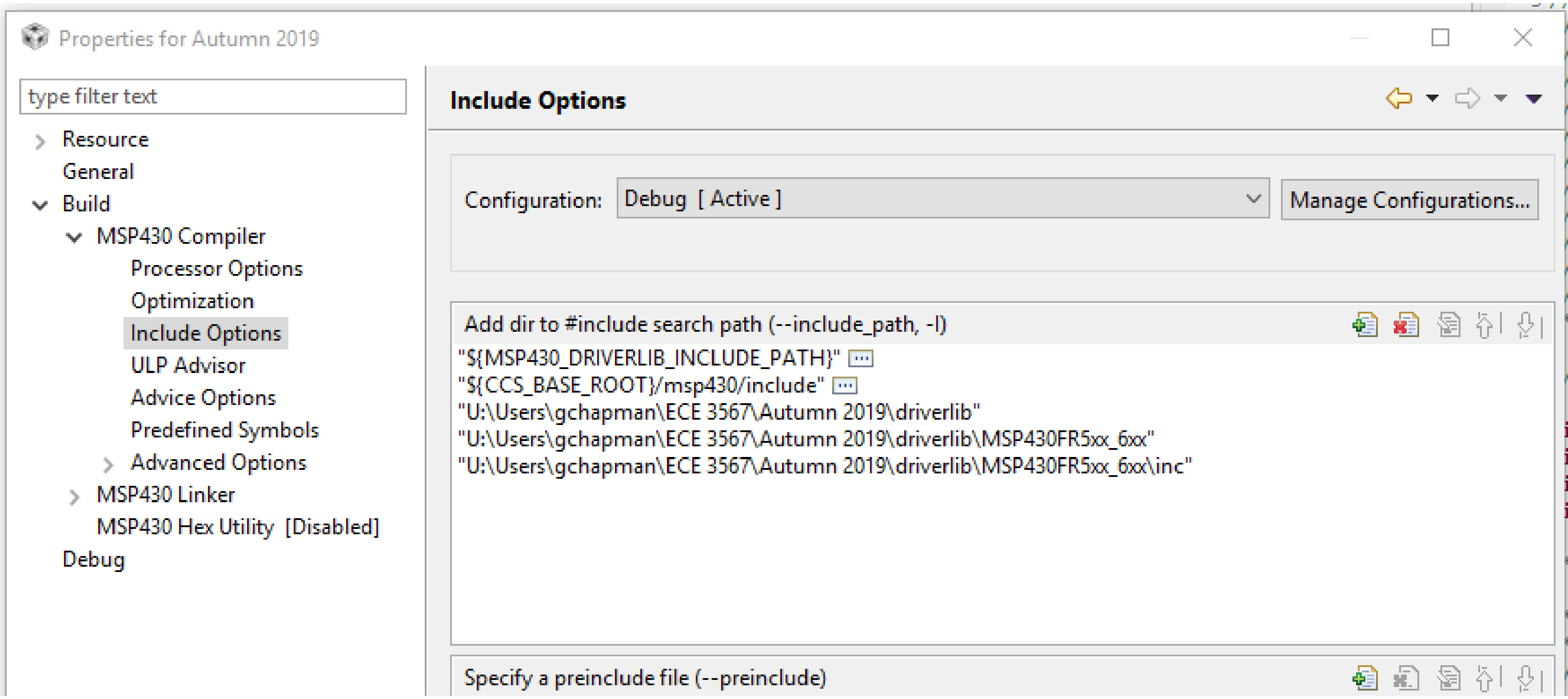
   When you select OK, Code Composer will restart

Change Include Options

1. Right Click on your **Project Name [Active-Debug] .** Select **Properties** (all the way at the bottom of menu)
2. Go to **Include Options** under **MSP430 Compiler**
3. You will need to add the proper path to all three of the following :

   "U:\<your path>\<your project name.\driverlib"
   "U:\<your path>\<your project name.\driverlib\MSP430FR5xx_6xx"
   "U:\<your path>\<your project name.\driverlib\MSP430FR5xx_6xx\inc"
   NOTE: DON'T copy my path format, this is on my computer at home.

4. Use the ICON to add path
5. Delete old paths to same folders using ICON
6. When finished, click OK



Configuration: Debug [Active]

Add dir to #include search path (--include_path, -I)

```
"${MSP430_DRIVERLIB_INCLUDE_PATH}"
"${CCS_BASE_ROOT}/msp430/include"
"U:\Users\gchapman\ECE 3567\Autumn 2019\driverlib"
"U:\Users\gchapman\ECE 3567\Autumn 2019\driverlib\MSP430FR5xx_6xx"
"U:\Users\gchapman\ECE 3567\Autumn 2019\driverlib\MSP430FR5xx_6xx\inc"
```

Specify a preinclude file (--preinclude)

Code Composer Studio

Common Error Messages

ERROR: Unable to Launch. The selection cannot be launched, and there are no recent launches

Solution: Switch Workspace : File => Switch Workspace => Other
Always select the folder that is ONE LEVEL ABOVE your project folder as the workspace.

ERROR: Cannot open source file “driverlib.h”

Solution: Change the paths to the following folders -

1. Right click the project set as **[Active-Debug]** and open Properties
2. Under MSP430 Compiler options, select Include Options
3. Delete old paths to driverlib

Add **ALL THREE** new paths:

…/ driverlib
…/ driverlib/MSP430FR5xx_6xx
…/ driverlib/MSP430FR5xx_6xx/inc

Code Composer Studio - Watch Window

Note that, unlike the TI C2000 series there is not a dynamic debugger for the TI MPS430 LaunchPads. Expressions and Variables are only updated after a Breakpoint is reached, and NOT available of the Debugger is halted.

1. Select View → Expressions

2. Right click to add registers or variable names.

3. Set breakpoints

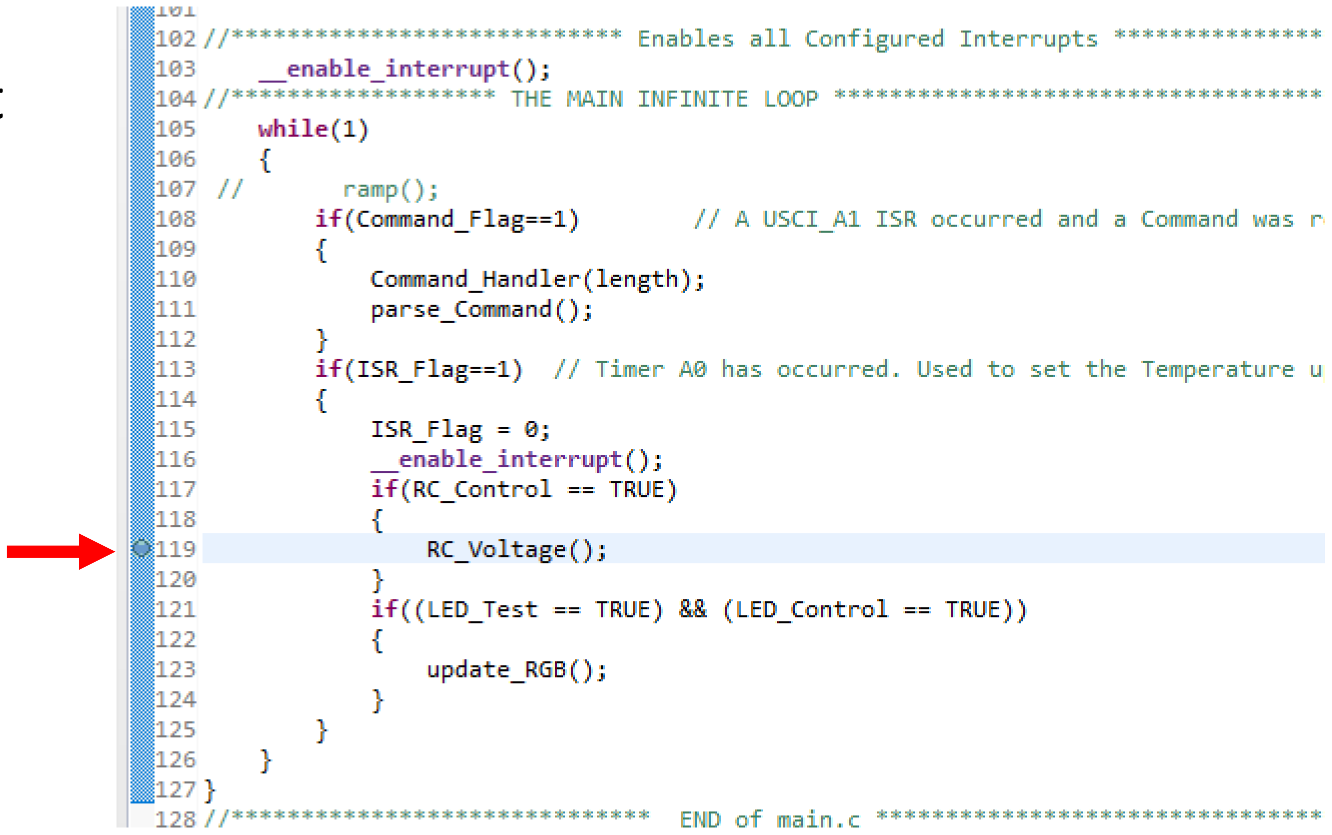
4. Run the Debugger

Troubleshooting - Breakpoints

1. Double click the line number to set a breakpoint

2. You must re-compile to incorporate the breakpoint

3. Click twice more to clear a breakpoint

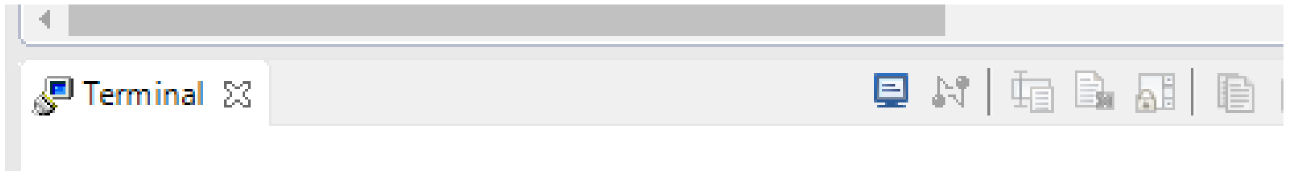


```
101
102 //*************************** Enables all Configured Interrupts ***************
103     __enable_interrupt();
104 //****************** THE MAIN INFINITE LOOP ***********************************
105     while(1)
106     {
107  //         ramp();
108         if(Command_Flag==1)         // A USCI_A1 ISR occurred and a Command was r
109         {
110             Command_Handler(length);
111             parse_Command();
112         }
113         if(ISR_Flag==1)  // Timer A0 has occurred. Used to set the Temperature u
114         {
115             ISR_Flag = 0;
116             __enable_interrupt();
117             if(RC_Control == TRUE)
118             {
119                 RC_Voltage();
120             }
121             if((LED_Test == TRUE) && (LED_Control == TRUE))
122             {
123                 update_RGB();
124             }
125         }
126     }
127 }
128 //****************************  END of main.c *********************************
```

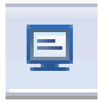
Serial Communications via UARTs

Setting up the Terminal Window

Code Composer Studio – The Terminal Window



1. View → Terminal

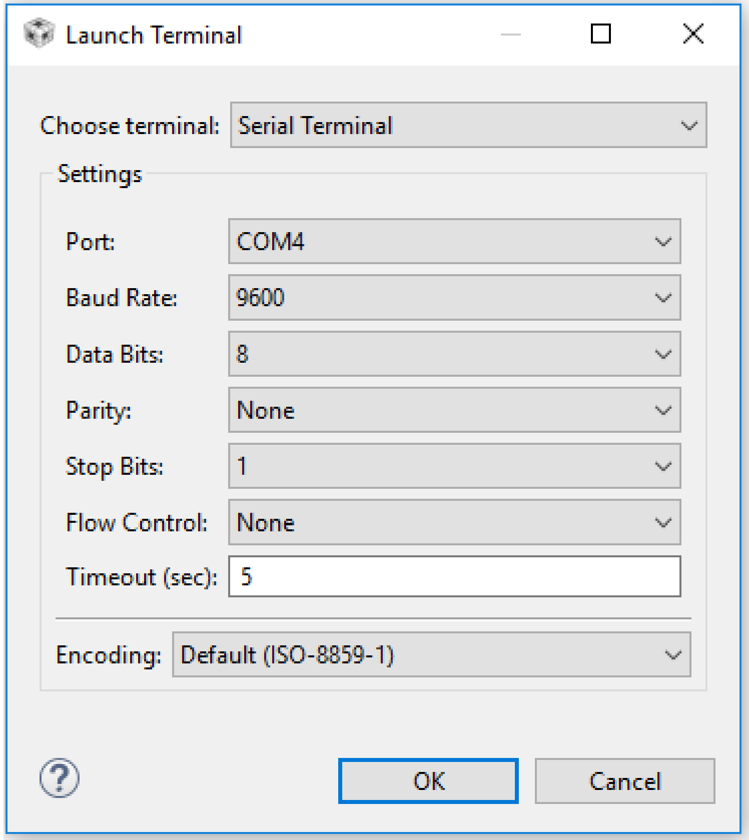
2. Click on the  ICON to launch the terminal. The Launch Terminal window should open:

   Choose the Serial Terminal to switch to the proper settings

3. The pulldown menu for Port should list the available COM ports. If these end up not working, you will have to resort to the process on the following slide to discover the COM ports

5. If not already the default settings, select: Baud Rate – ***9600***, Data Bits – ***8***, Parity – ***None***, Stop Bits – ***1***, Flow Control – ***None***, Timeout – ***5 secs***, Encoding - ***Default (ISO-8859-1)***

6. Select OK



Choose terminal: Serial Terminal

Settings

Port:	COM4
Baud Rate:	9600
Data Bits:	8
Parity:	None
Stop Bits:	1
Flow Control:	None
Timeout (sec):	5

Encoding: Default (ISO-8859-1)

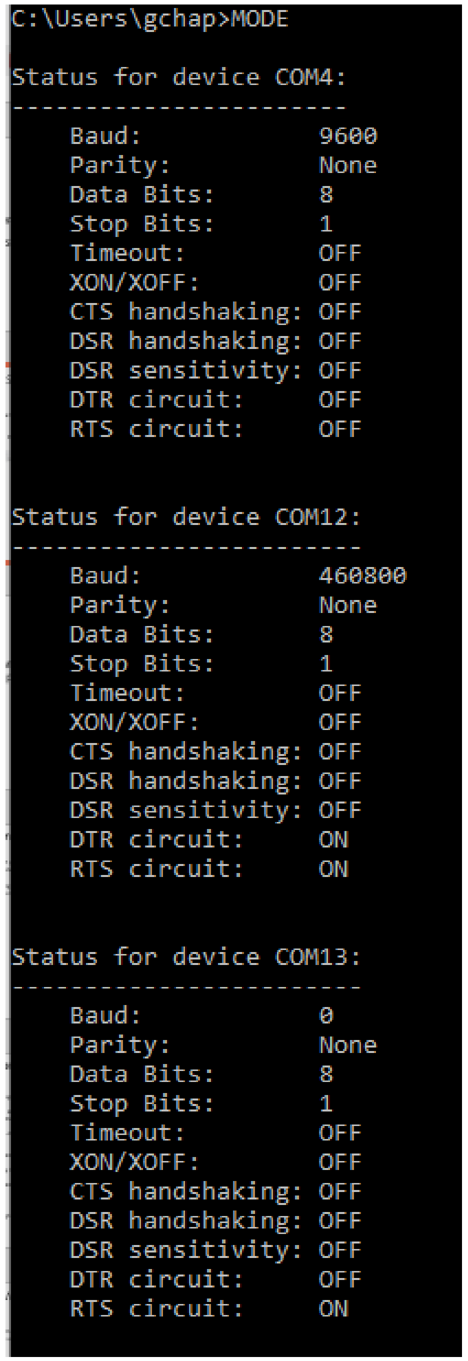
Finding an Available COM Port

- Each lab computer uses numerous COM ports. Before you activate the Terminal in CCS, you must identify an available COM port. To do this outside of the CCS Terminal:

1. type CMD in the search window to bring up the Command Line Prompt
2. type MODE and write down the communication port numbers that are available (e.g. COM4, COM12, COM13)
3. Close the Command window

The list of available COM ports should update in the CCS Terminal menus. This procedure should only be necessary if you suspect CCS is not updating the list.

NOTE: COM port numbers are **DYNAMIC**. When you unplug your LaunchPad and plug it back in, the COM port numbers WILL be different. Check the Terminal list of COM ports, or run the CMD/MODE again to document the NEW list of available COM ports.



```
C:\Users\gchap>MODE

Status for device COM4:
-----------------------
    Baud:            9600
    Parity:          None
    Data Bits:       8
    Stop Bits:       1
    Timeout:         OFF
    XON/XOFF:        OFF
    CTS handshaking: OFF
    DSR handshaking: OFF
    DSR sensitivity: OFF
    DTR circuit:     OFF
    RTS circuit:     OFF


Status for device COM12:
------------------------
    Baud:            460800
    Parity:          None
    Data Bits:       8
    Stop Bits:       1
    Timeout:         OFF
    XON/XOFF:        OFF
    CTS handshaking: OFF
    DSR handshaking: OFF
    DSR sensitivity: OFF
    DTR circuit:     ON
    RTS circuit:     ON


Status for device COM13:
------------------------
    Baud:            0
    Parity:          None
    Data Bits:       8
    Stop Bits:       1
    Timeout:         OFF
    XON/XOFF:        OFF
    CTS handshaking: OFF
    DSR handshaking: OFF
    DSR sensitivity: OFF
    DTR circuit:     OFF
    RTS circuit:     ON
```

Communication

THIS IS PART OF A LATER LAB. It is introduced now for completeness in presenting Code Composer Studio. Once the Command Structure has been added to your code:

- When you run the ECE 3567 Project program, You should receive the following message from the ECE 3567 software:

  ECE 3567 Microcontroller Lab
  Please enter a Command Code:

- You can test the transmit with any valid two-character command, which must be capitalized. If they are not programmed, you will receive:

  UNKNOWN COMMAND
  Please enter the next Command Code:

You will add this functionality to the code later in the semester

Introduction to Lab 1
September 17th

Lab 1 – Blinking LEDs

Goals:

1) Learn to set-up a new Project in Code Composer Studio

2) Lear to code, compile and run a project in Code Composer Studio

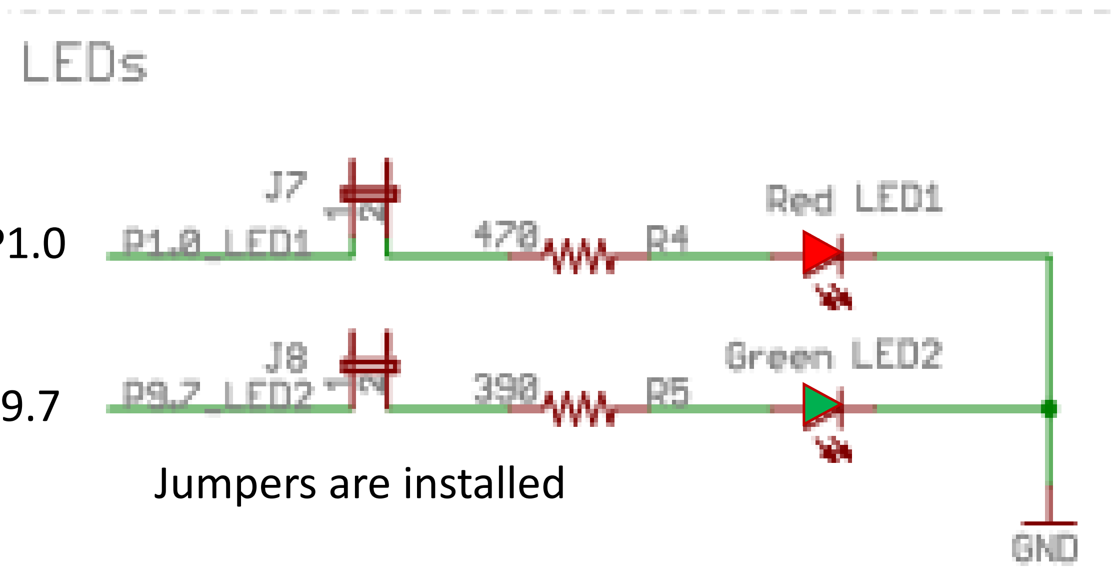
3) Gain experience with the Bitwise C operators to control external hardware.

4) Learn how to re-configure your project when the code is moved to a different path.

Lab 1 – Blinking LEDs

- Write a program that will illuminate the red and green LEDs on the MSP430FR6989 Launchpad board.
- Which LED is illuminated should alternate at approximately one second intervals.
- For this lab, we will not use timers or interrupts. Write delay code in the main loop and manipulate the proper output pins to the LEDs using generalized I/O ports 1 and 9 (see schematic).
- We will not use TI macros to control the I/O ports in this lab.
- Once your project is working. You will be asked to change the location of your code and make the necessary changes to get the code running again
- You will NOT be prepared to write the Lab 1 code or pass the quiz at the end of the lab session (this is a single class lab) if you do not watch and understand the Lab 1 Video.

Lab 1 - Schematic



P1.0

P9.7

Jumpers are installed