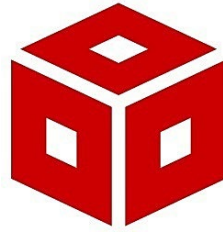


# ECE 3567 Microcontroller Lab

## Lecture #3 – Code Composer Studio



Spring 2025

Dr. Gregg Chapman

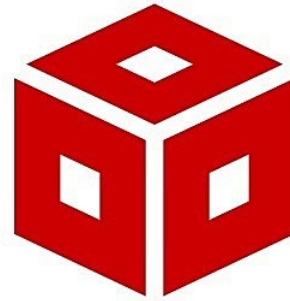
# Project Set-up

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

- You may create more than one project in the same workspace.
- I recommend having a separate workspace for each lab. Code Composer creates the PROJECT folder. You define the WORKSPACE.
- **NEVER** have more than one project in a project folder!!!!

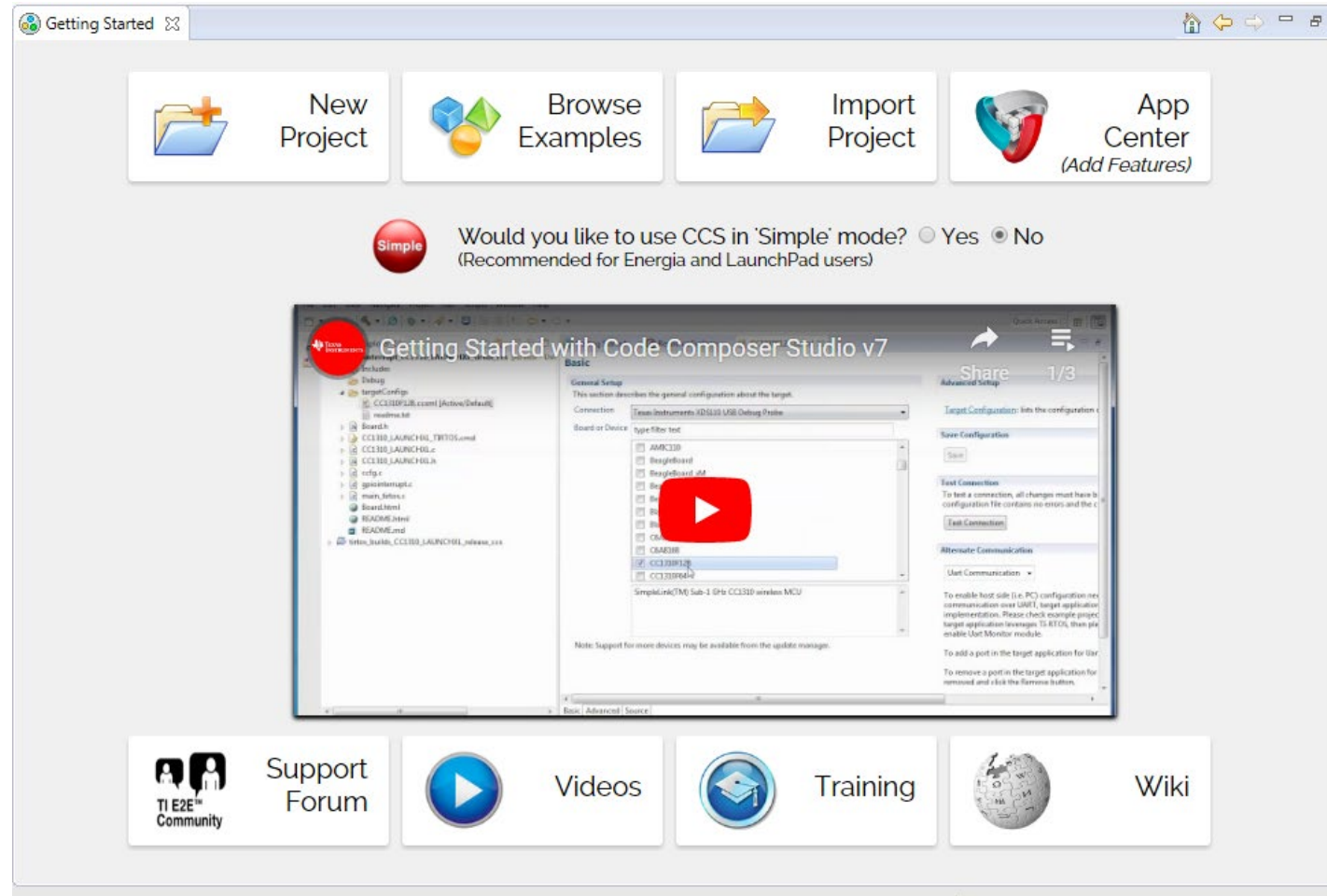
# Project Set-up

Invoke the Code Composer Studio version 10.1.0 software by double clicking on the CCS ICON



# Project Set-up

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



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

# Project Set-up

**New CCS Project**  
Create a new CCS Project.

Target: MSP430FRxxx Family MSP430FR6989

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

MSP430

Project name: Lab 2

☐ Use default location

Location: U:\ECE 3567\Lab2 Workspace Browse...

Compiler version: TI v18.12.2.LTS More...

Tool-chain

Project templates and examples

type filter text

- Empty Project (with main.c)
- Empty Assembly-only Project
- Empty RTSC Project
- Basic Examples
  - Blink The LED
  - Hello World
- MSP430 DriverLib
  - Empty Project with DriverLib Source
- More Examples...

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...

Open [Import Wizard](#) to find local example projects for selected device...

Finish

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

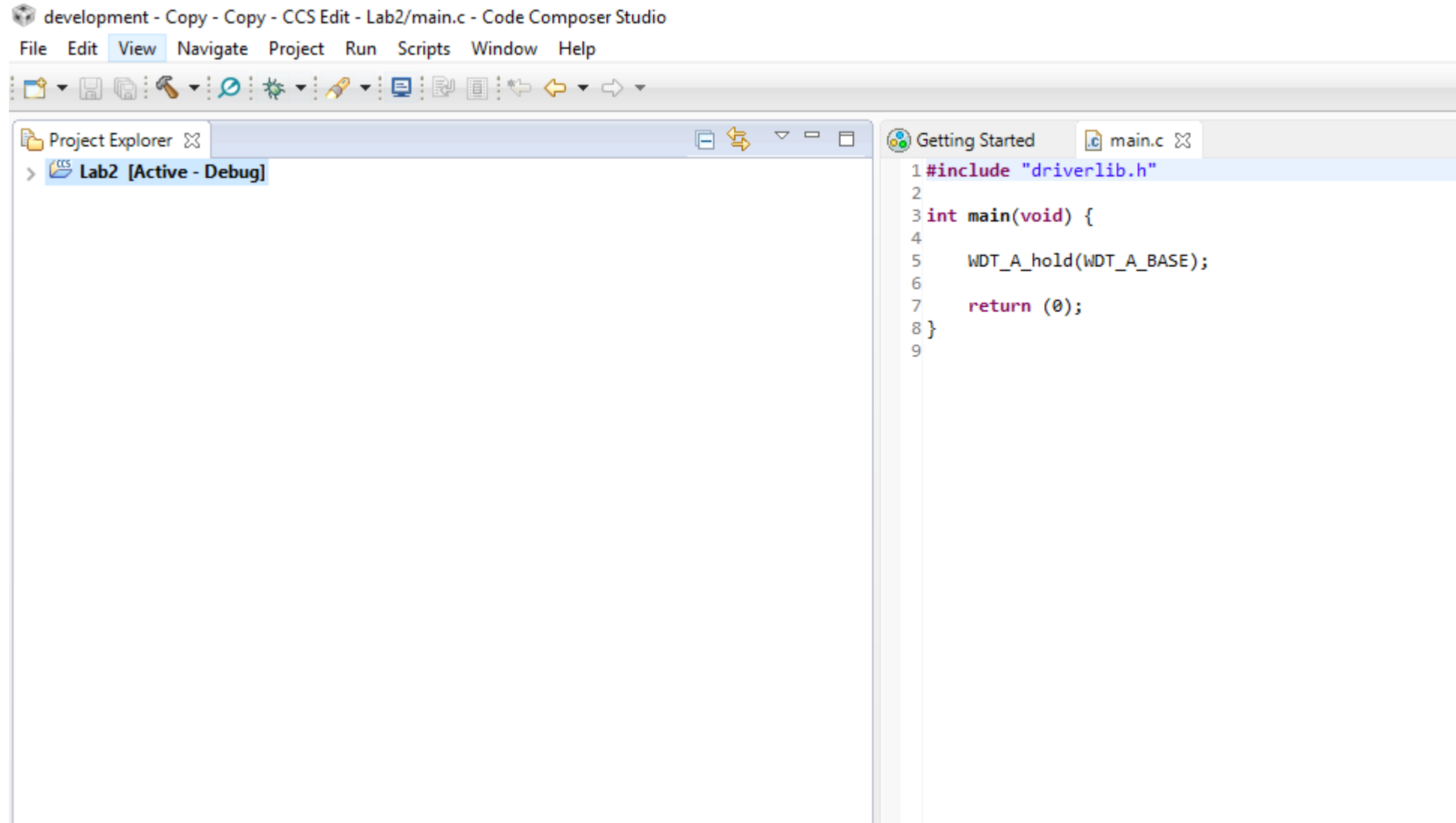
1. Target: **MSP430FRxxx** Family
2. **MSP430FR6989**
3. Connection: Leave at Default
4. Project Name (*e.g. Lab 2*)
5. Uncheck use default location and browse to **YOUR WORKSPACE** (*e.g. U:\ECE3567 OR U:\ECE3567\Lab2 Workspace*)

This depends on how you set it up. CCS will create a project folder with the Project Name under the WORKSPACE you define in LOCATION.

6. Leave Compiler at Default
7. In Project templates and examples, you must **SCROLL DOWN** to **MSP430 DriverLib**, select the down arrow v and **SCROLL DOWN AGAIN**. Highlight:  
**Empty Project with DriverLib Source**
8. Select Finish

# Project Set-up

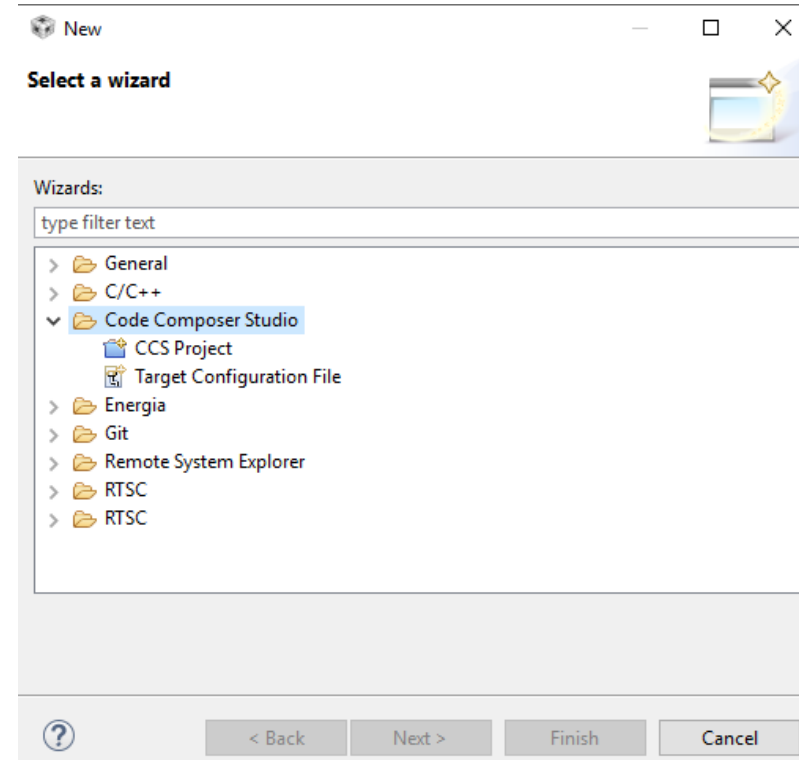
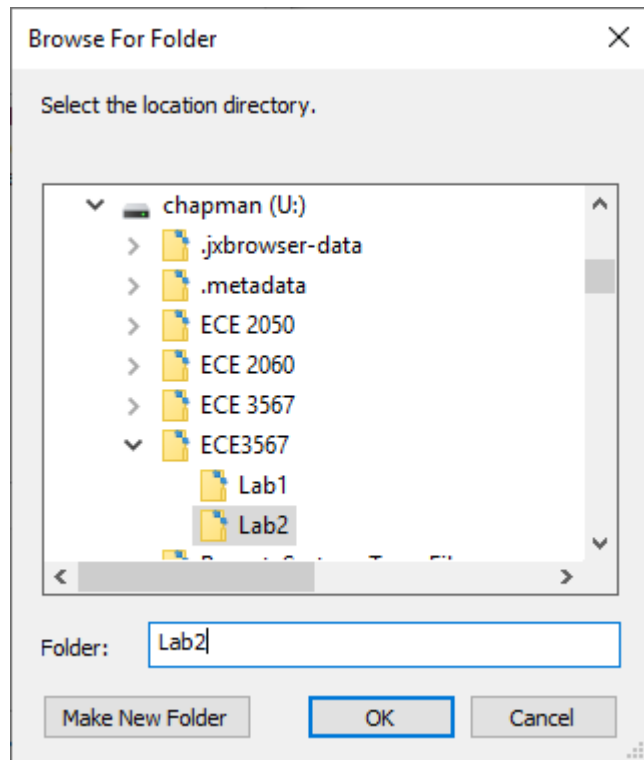
"Voilà"



# Project Set-up

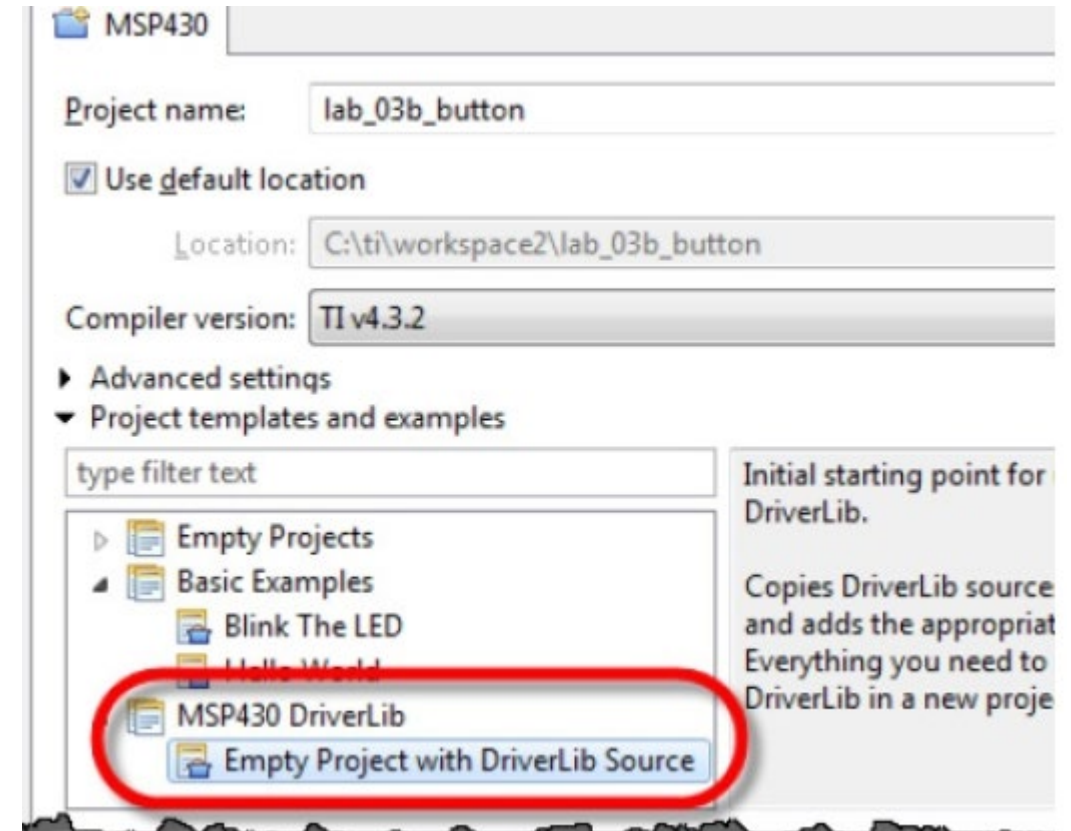
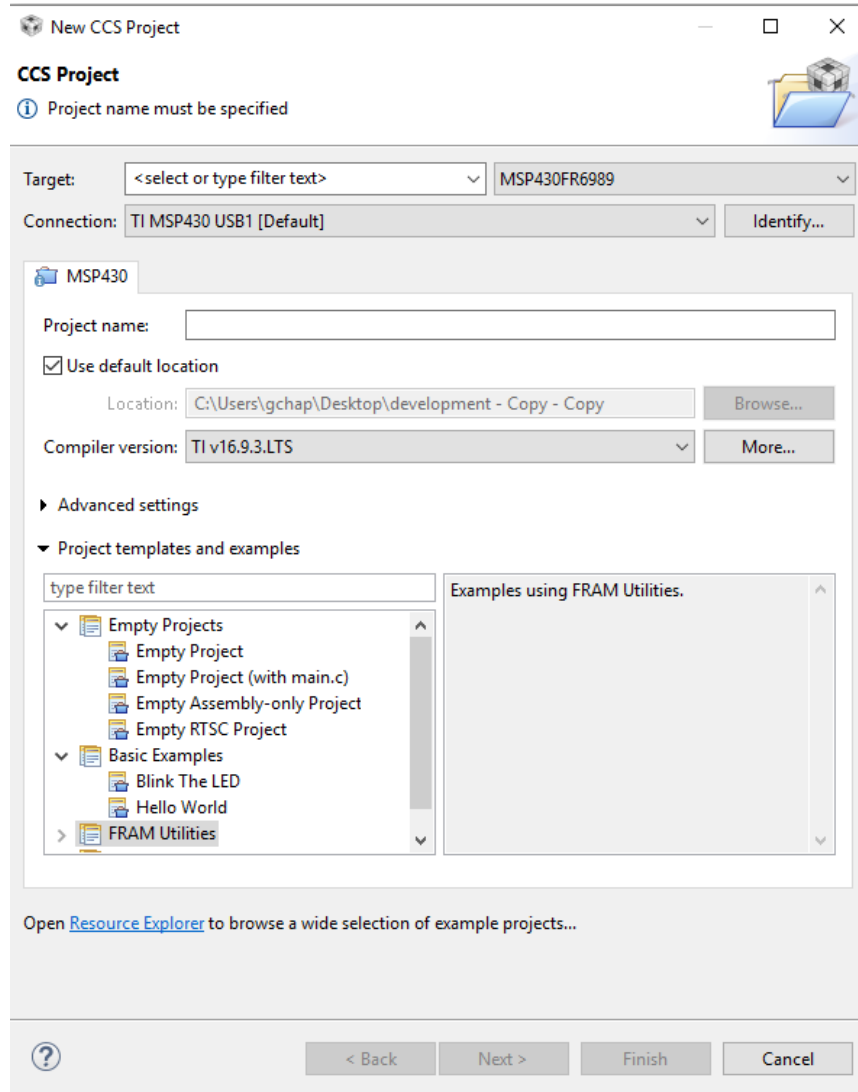
**ALTERNATIVELY:** If you are beginning with another project already open in Project Explorer:

1. Select **File** → **Switch Workspace** and navigate to your new workspace
2. Select **File** → **New** → **Project** → **Code Composer Studio** → **CCS Project**



# Project Set-up

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



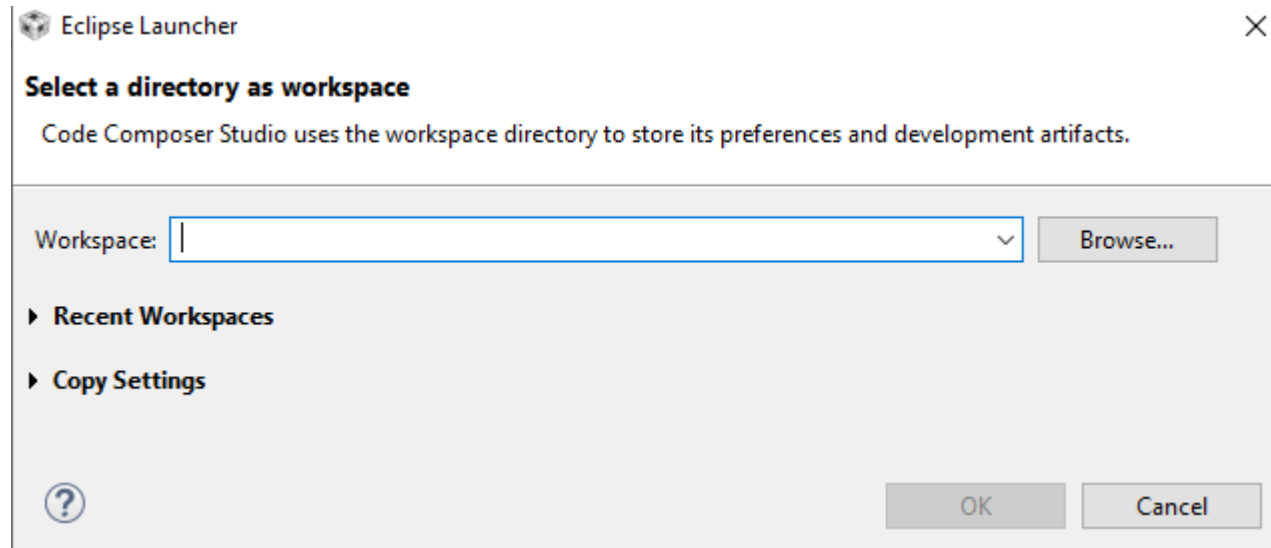
Don't Forget to select this template.



# Project Set-up

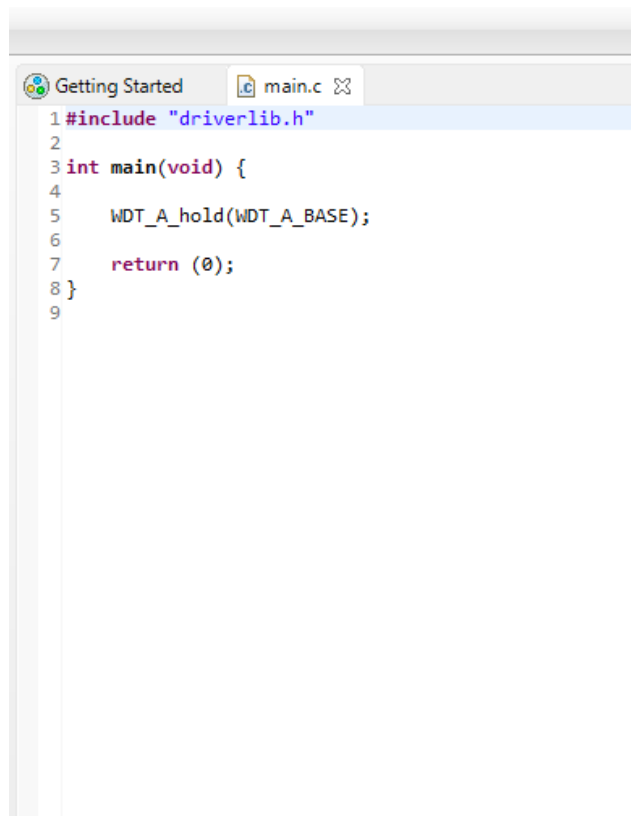
The FIRST time you use the workspace, CCS MAY open the Eclipse Launcher.

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



# Code Composer Studio – Adding Code

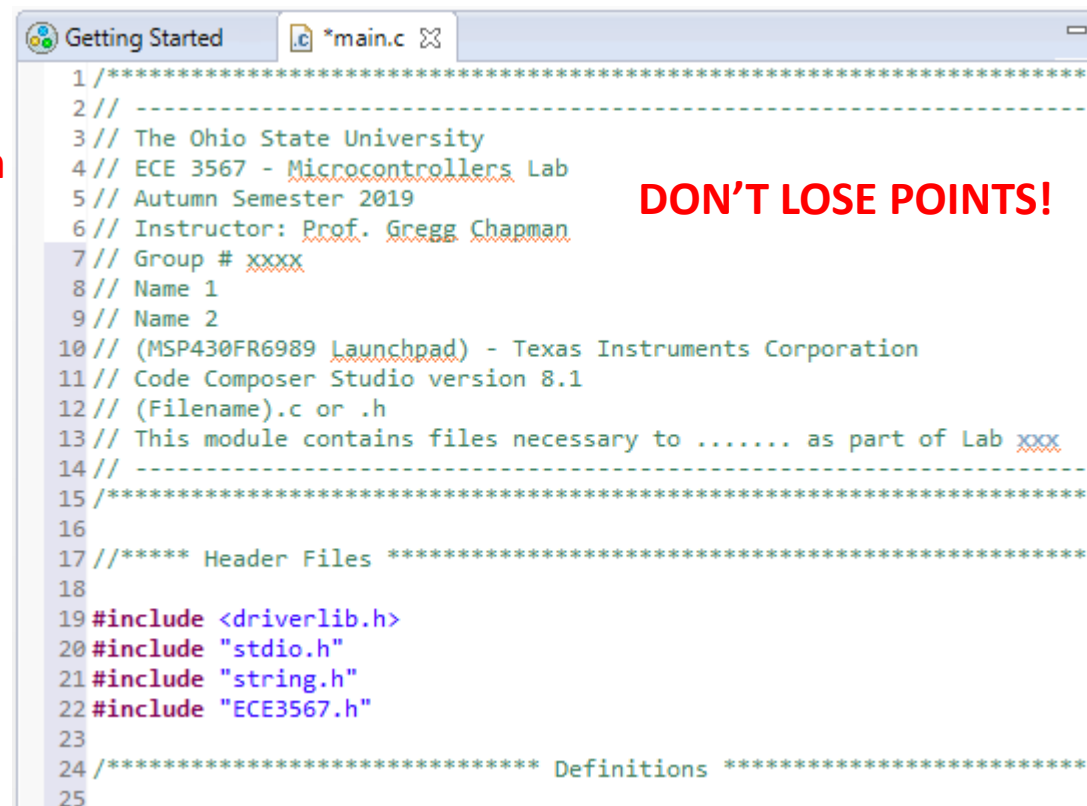
1. Select File → New → Source File
2. Enter Code
3. Save the File
4. Select Project → Rebuild Project
5. You may also COPY files into your project folder then add them to the project with a Right-Click on your **Project Name [Active-Debug]** . Then select **Add Files . . .**



The screenshot shows the Code Composer Studio interface with a single tab titled 'main.c'. The code in the editor is as follows:

```
1 #include "driverlib.h"
2
3 int main(void) {
4     WDT_A_hold(WDT_A_BASE);
5
6     return (0);
7 }
8
9
```

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







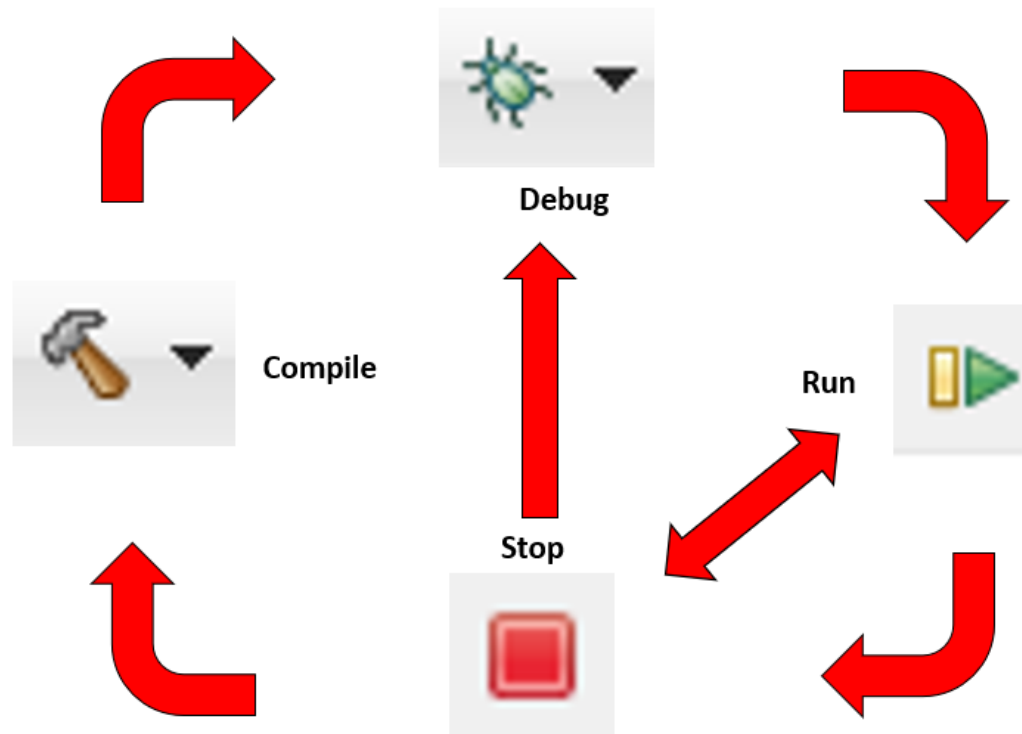
The screenshot shows the Code Composer Studio interface with a tab titled '\*main.c'. The code in the editor is as follows:

```
1 /*****
2 // -----
3 // The Ohio State University
4 // ECE 3567 - Microcontrollers Lab
5 // Autumn Semester 2019
6 // Instructor: Prof. Gregg Chapman
7 // Group # xxxx
8 // Name 1
9 // Name 2
10 // (MSP430FR6989 Launchpad) - Texas Instruments Corporation
11 // Code Composer Studio version 8.1
12 // (Filename).c or .h
13 // This module contains files necessary to ..... as part of Lab xxx
14 // -----
15 /*****
16
17 //***** Header Files *****
18
19 #include <driverlib.h>
20 #include "stdio.h"
21 #include "string.h"
22 #include "ECE3567.h"
23
24 /***** Definitions *****
25
```

**DON'T LOSE POINTS!**

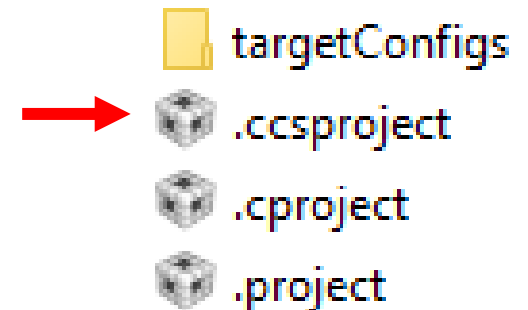
# Code Composer Studio – Running the Project Code

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)

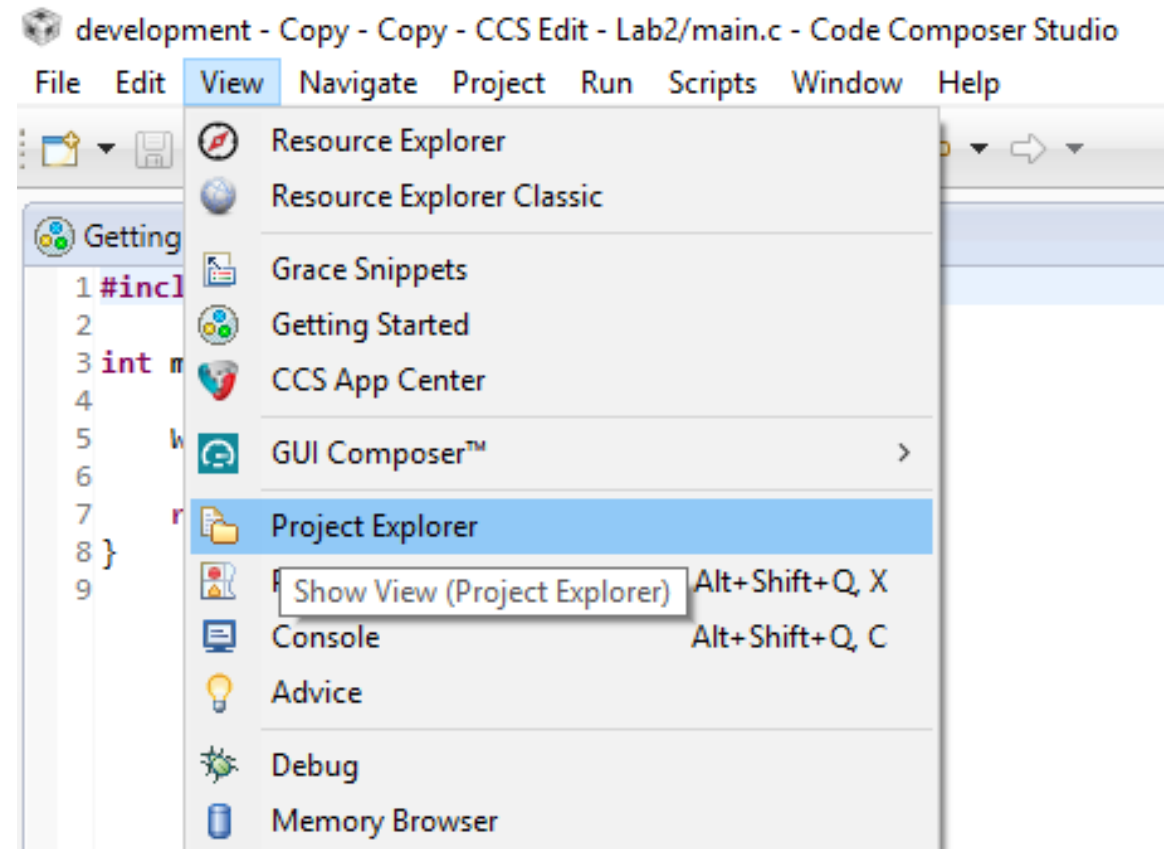
- This doesn't always work
- Make sure that the Workspace is set to your project location. To change it, select **File → Switch Workspace**, and navigate to the project location
- Double-click the **.ccsproject** ICON in the project folder
- If it doesn't work, try the Manual Method



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

If your project does not appear

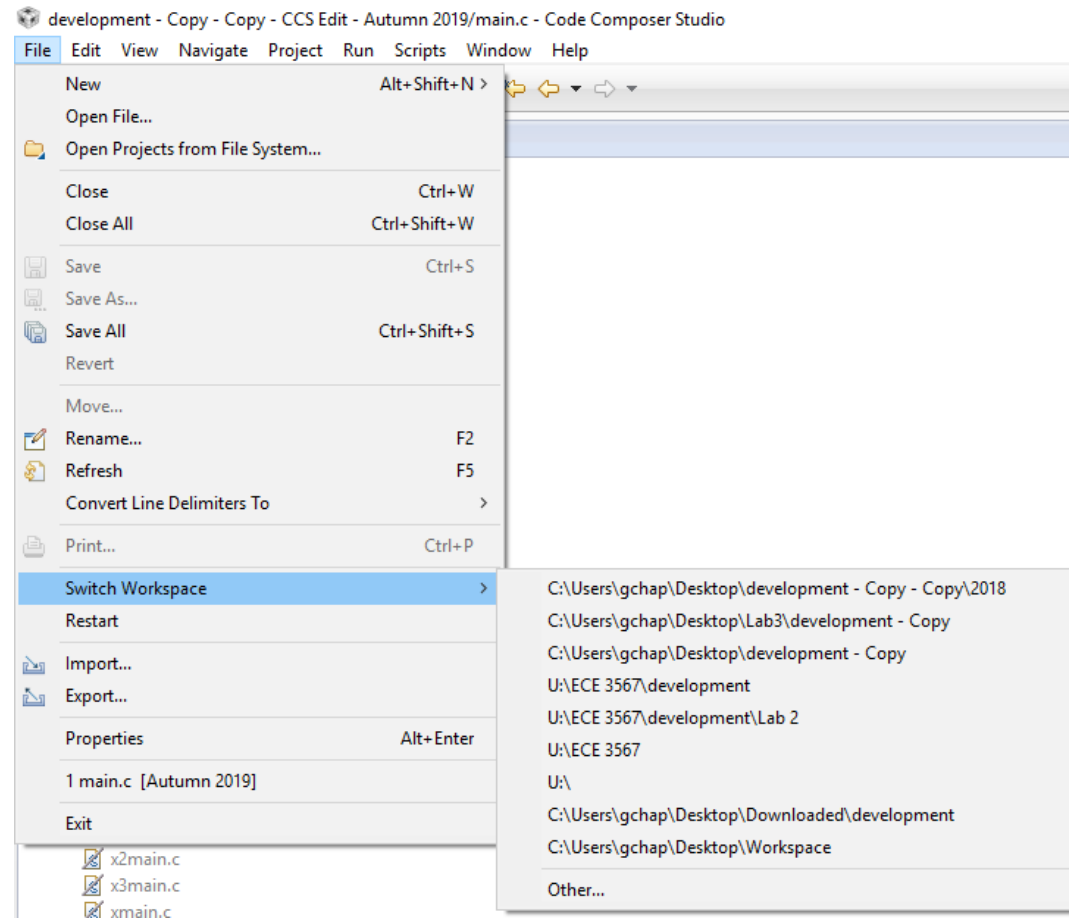
- Select **View** → **Project Explorer**



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

If you still don't see anything in Project Explorer

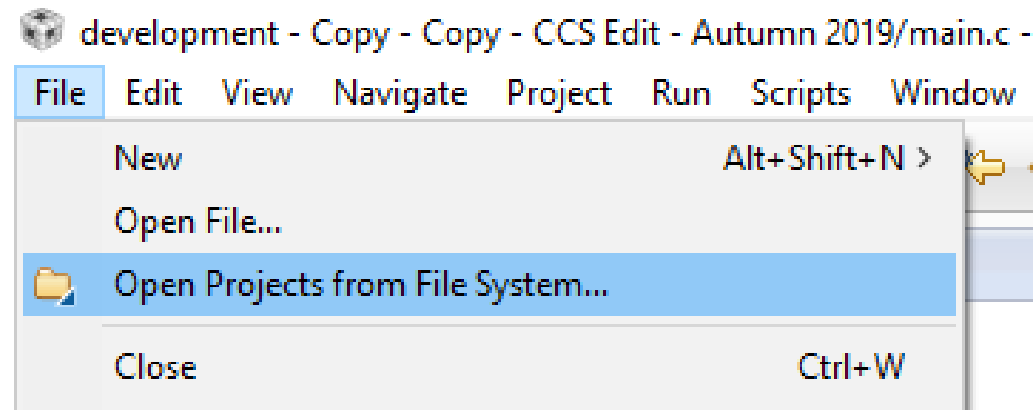
- Select **File** → **Switch Workspace**, and navigate to the project location



# 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 . . .**



# 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 Compiler settings if they are not correct.



# Change Workspace

1. Select **File → Switch Workspace → Other**
2. Enter the new workspace. **Always select the file ABOVE your Project file that CCS created. This is usually your current Workspace folder.**

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”

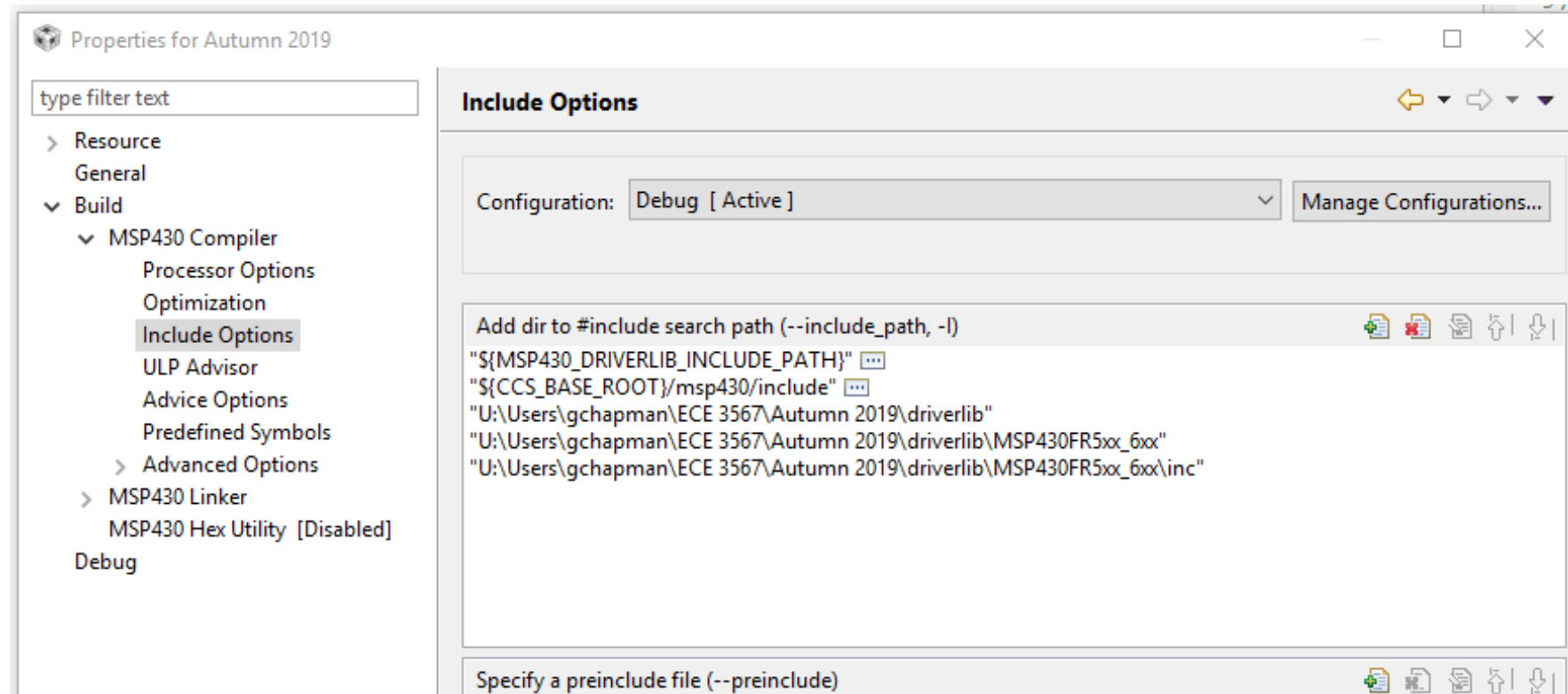
“U:\<your path>\<your project name.\driverlib\MSP430FR5xx\_6xx\inc”

**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



# 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.  
REBUILD PROJECT.

**ERROR:** Cannot open source file “driverlib.h” (*You may not have selected ‘Empty Project with Driverlib Source’ when you set up the project!*)

**Solution:** MANUALLY 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 (ECE 2050), there is not a dynamic debugger for the TI MSP430 LaunchPads. Expressions and Variables are only updated after a Breakpoint is reached, and NOT available if 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

Note: The debugger cannot always set a breakpoint on a selected line. If you get a compiler error, select another line of code.



```
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

(for Lab 5 – Serial Communications)

# Code Composer Studio – The Terminal Window

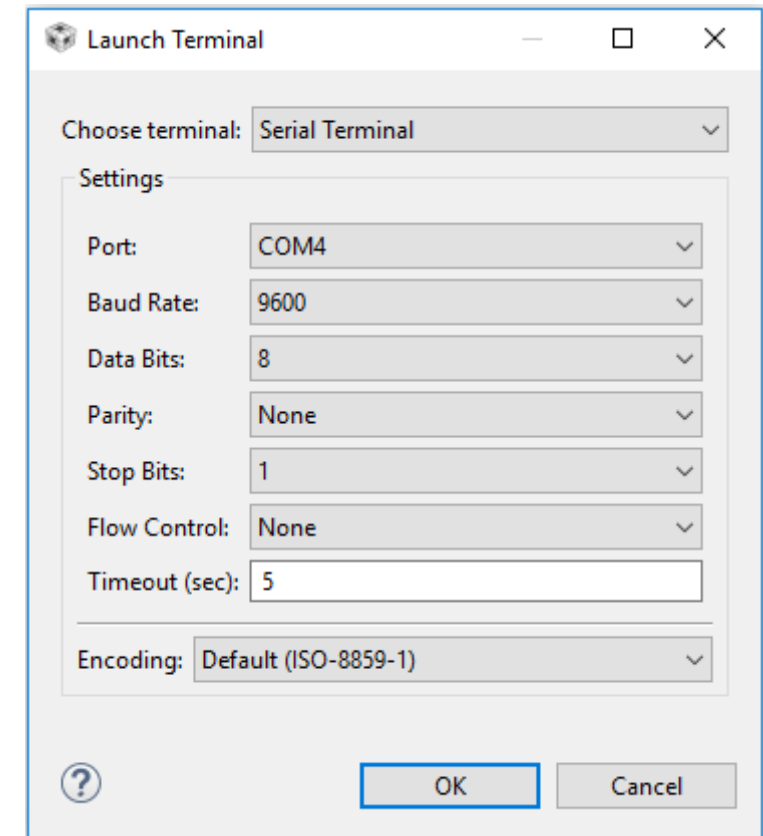
1. View → Terminal



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

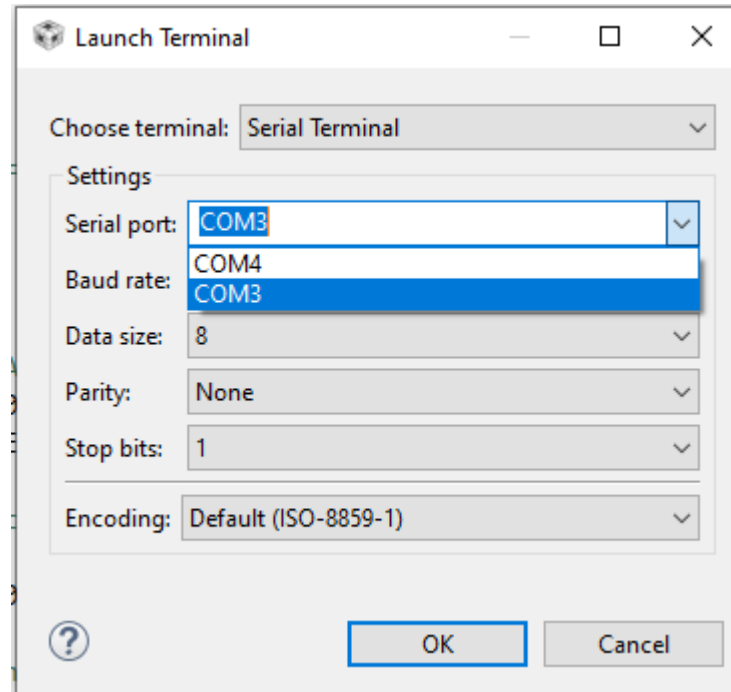
Choose the Serial Terminal to switch to the proper settings

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



# Finding an Available COM Port

- The list of available COM ports should update in the CCS Terminal menus.





# Finding an Available COM Port

- If the list of COM ports is not available in CCS, you can look at available COM ports in the Command window. This procedure should only be necessary if you suspect CCS is not updating the list.
- Each lab computer uses numerous COM ports. To identify:
  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). The first listed is usually already in use.
  3. Close the Command window

**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

- When you run the ECE 3567 Project program in Lab 5, 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 in Lab 5

# Introduction to Lab 1

# Lab 1 – Blinking LEDs

## Goals:

- 1) Learn to set-up a new Project in Code Composer Studio
- 2) Learn 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 if you do not watch and understand the Lab 1 Video.

# Lab 1 - Schematic

