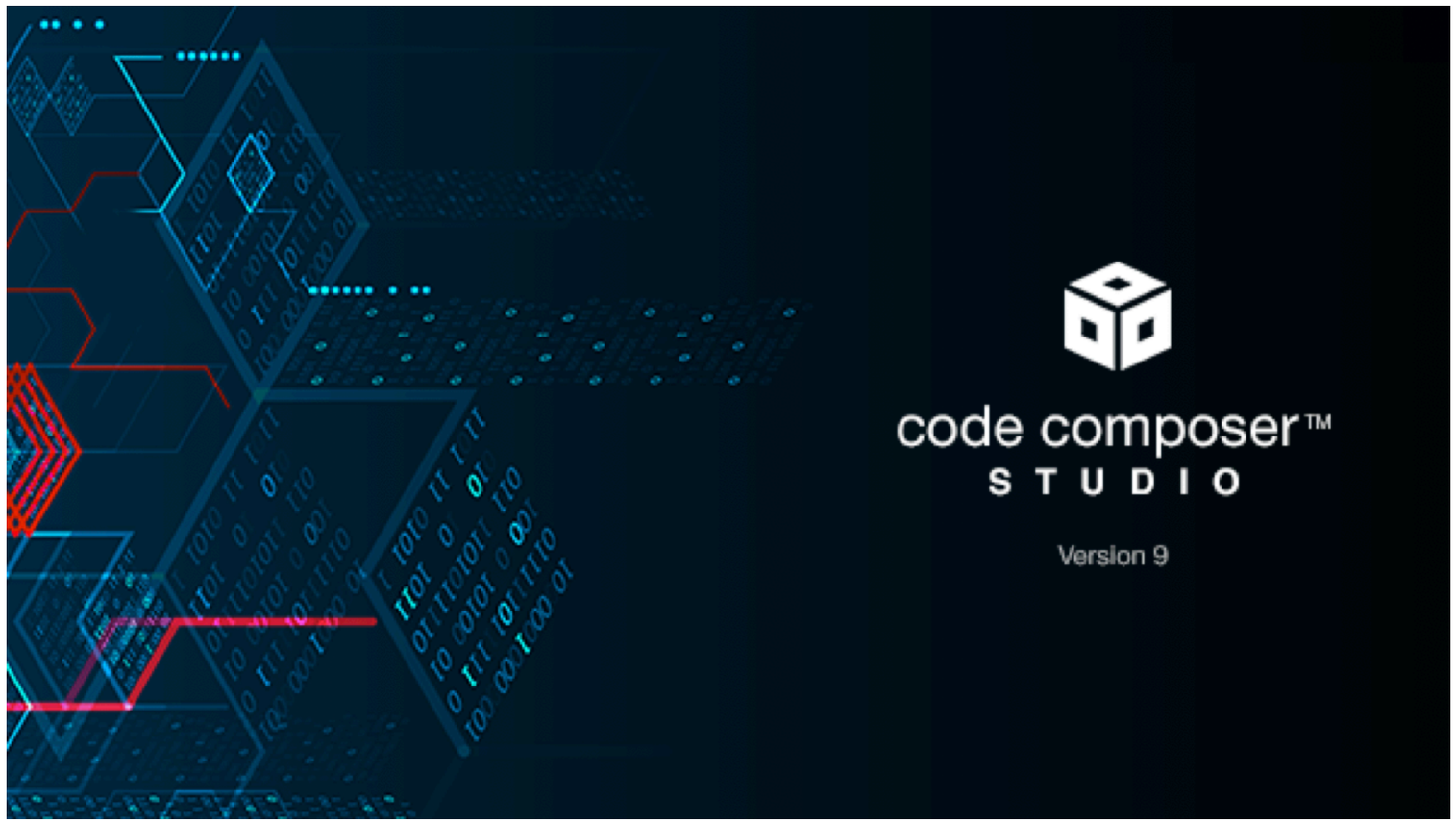
Code Composer Studio Tutorial



Ciaran Moore
ENCE361

Tutorial v. 1.4
CCS v. 9.2

Starting out with CCS

- During your Week-2 lab, you will:
  - Configure a Code Composer Studio (CCS) project using a simple GPIO example.
  - Learn how to compile, link, and debug a CCS project using the TM4C123GH6PM microcontroller.
  - Configure and use the TivaWare application programming interface (API).
  - Understand how general purpose input and output (GPIO) work on TM4C series microcontrollers by examining and modifying C source code.
- This guide is designed to get you started. You may need to refer back to it during later labs.

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Software Configuration

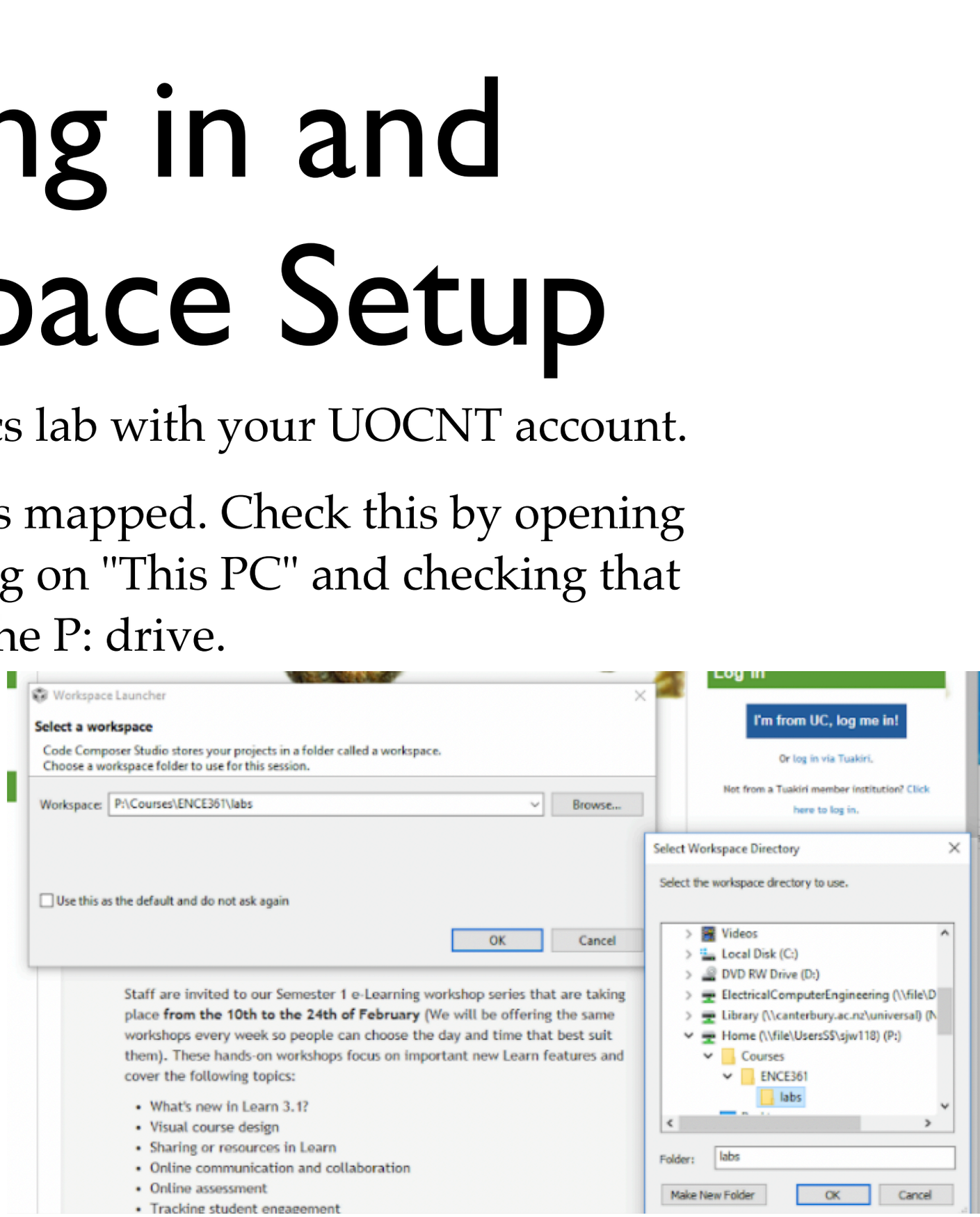
- PCs in the labs on level 2 of the ECE Wing have **CCS v. 9.2.0** installed.
- The API for TM4C series microcontrollers is called TivaWare. It is installed in **`C:\ti\TivaWare_C_Series-2.1.4.178`**.
- Store your source code and project files on your P: drive, but don't use spaces in any of your directory or file names!

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Logging in and Workspace Setup

- Login to a PC in the Electronics lab with your UOCNT account.
- Make sure that your P: drive is mapped. Check this by opening a file explorer window, clicking on "This PC" and checking that your user code is mapped to the P: drive.
- Create the directory `P:\Courses\ENCE361\labs`.
  Remember – no spaces!

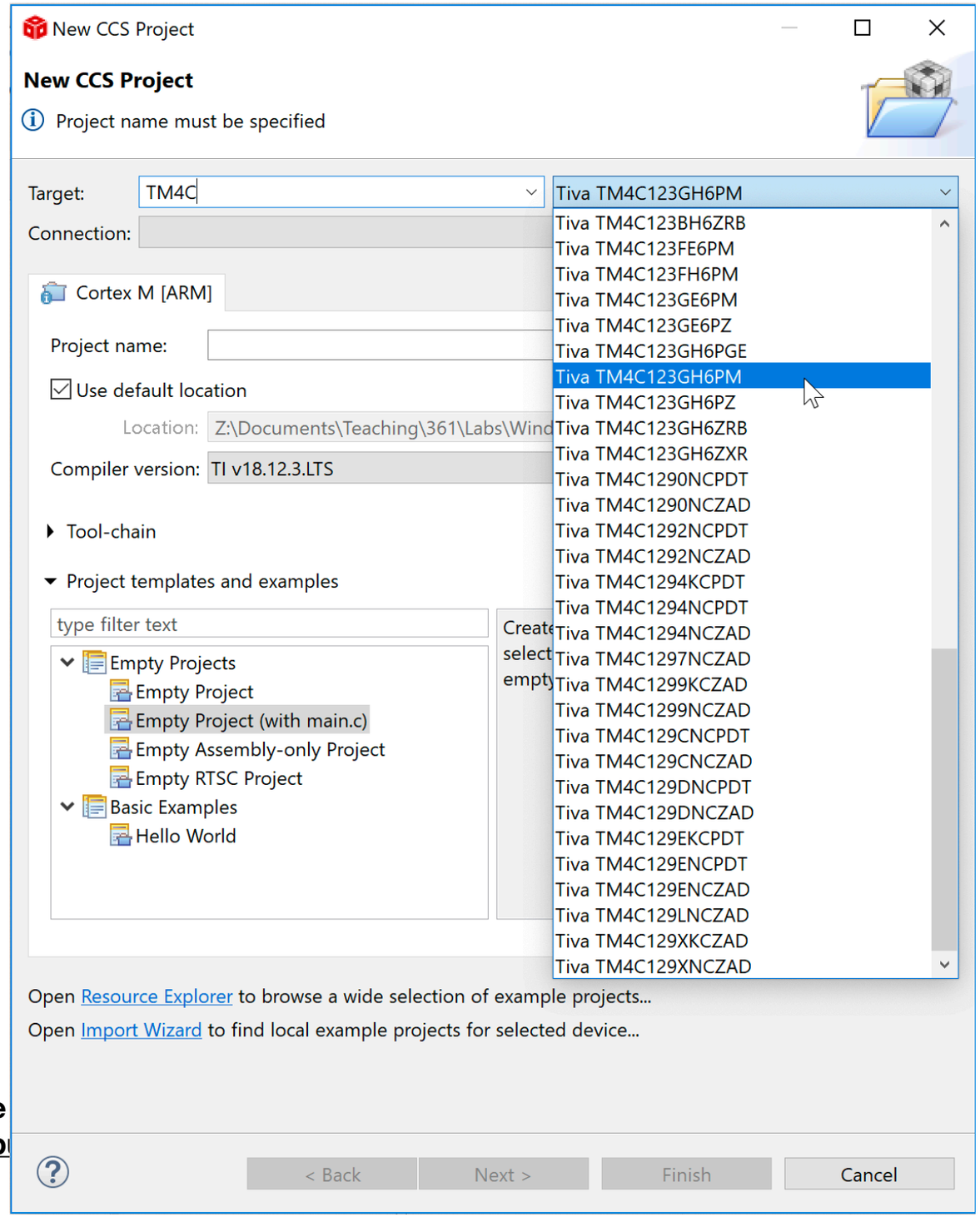
  Open CCS and select `P:\Courses\ENCE361\labs`
  as your workspace,
  then click OK.



Ciaran Moore
ciaran.moore@canterbury.ac.nz

Creating a Project

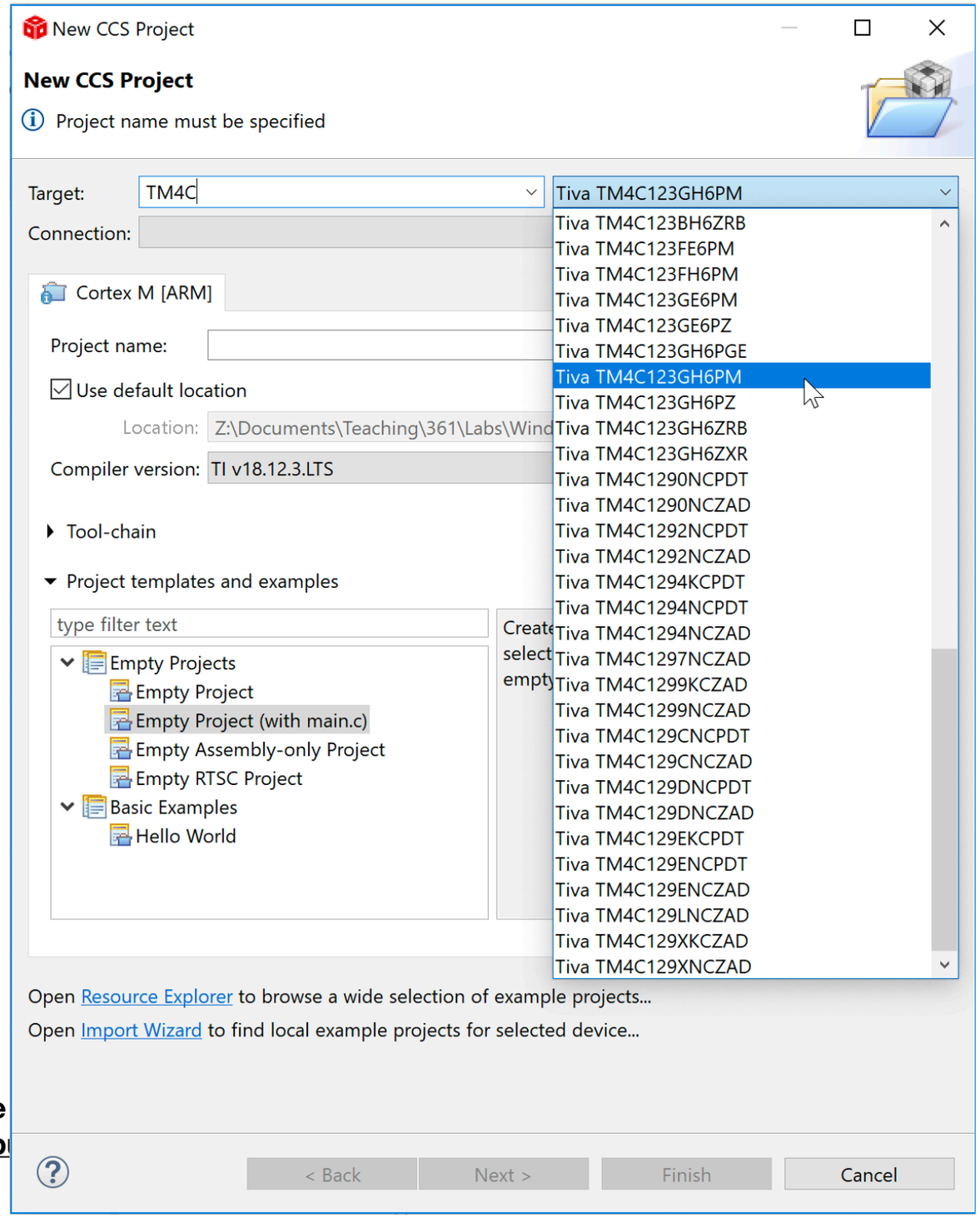
1. Create a new project:
   select File>New>
   CCS project.
2. Specify the microcontroller: start typing "TM4C123GH6PM" into the Target: search box and choose the corresponding entry from the adjacent drop down list.
   Be careful to choose the right part number: the "H6PM" suffix is essential.



Ciaran Moore
ciaran.moore@canterb

Creating a Project

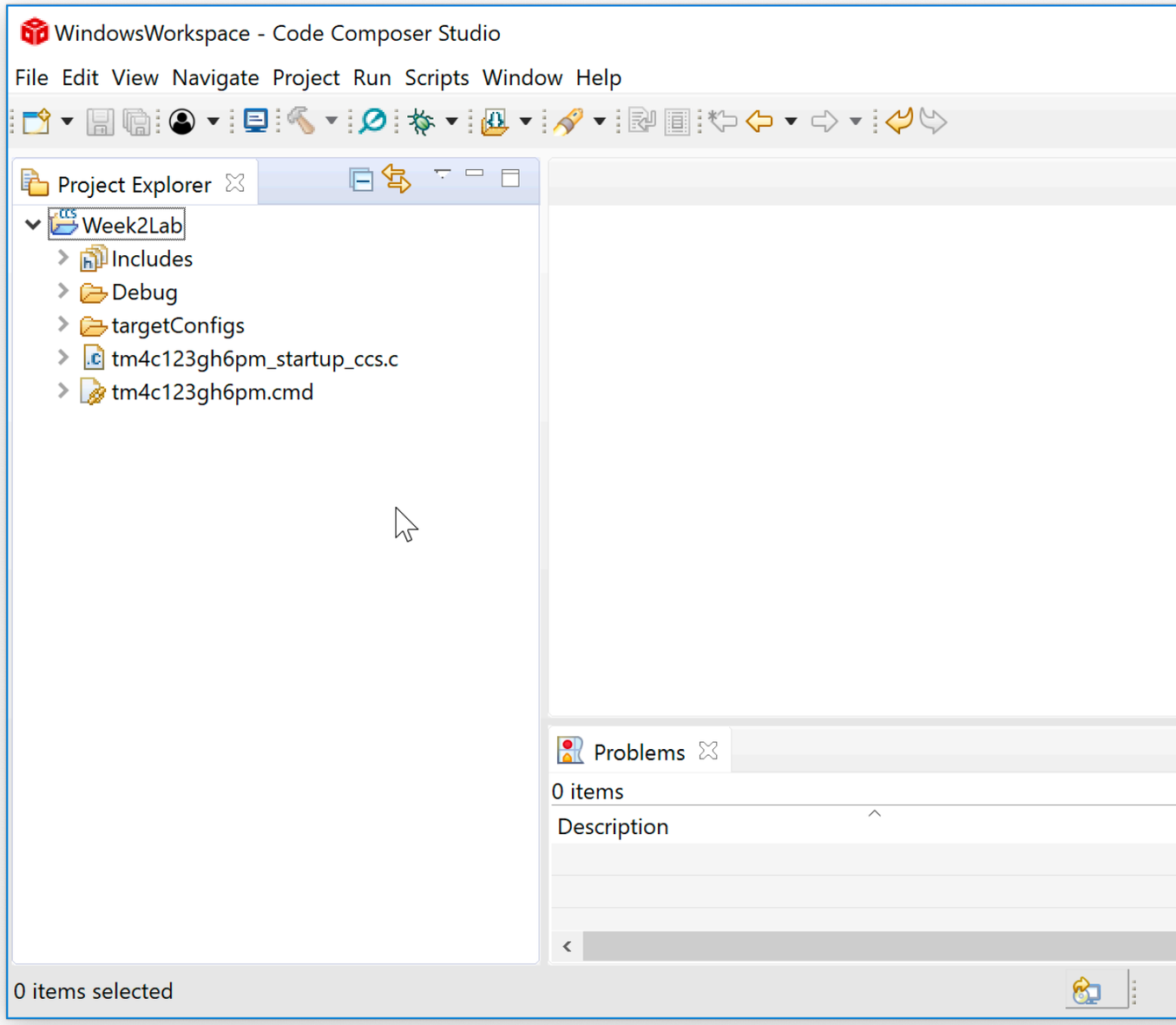
3. Select "Stellaris In-Circuit Debug Interface" from the Connection: drop down list.
4. Name your project: Week2Lab or similar.
5. Finally, select "Empty Project" from the list of Project templates.
6. Click Finish.



Ciaran Moore
ciaran.moore@canterb

CCS Edit View

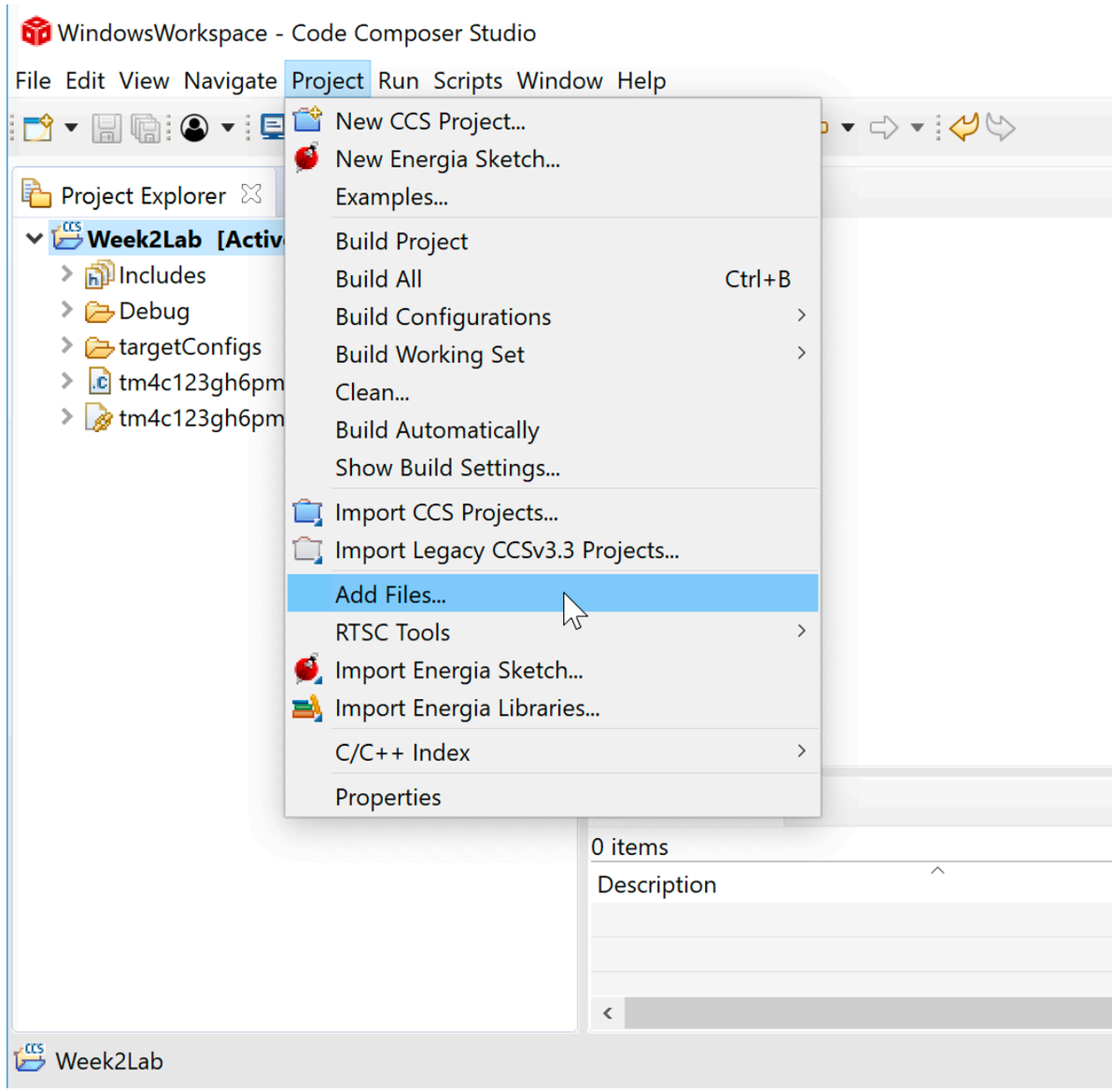
- CCS should now show you the Edit view: note the Project Explorer window on the left, showing the files associated with your project.



Ciaran Moore
ciaran.moore@canterbury.ac.nz

Adding files to your project

1. Download the source file for the lab from Learn: `Project and Laboratories|Laboratory source code|Week-2`.
2. For Week 2 there is only a single file, `week2_blink.c`, but there will be more for later labs.
3. Copy the source file into your `P:\courses\labs\Week2Lab` folder.
4. Go back to CCS and look at the Project Explorer: it should now list week2_blink.c along with the other files in your project.
5. NB: You can also select Project>Add Files… in CCS to achieve the same result.



Ciaran Moore
ciaran.moore@canterbury.ac.nz

Setting up CCS

- Before we can test our code, we need to tell CCS about the TivaWare API:

1. We need to #include the API so that the code will compile, and

2. We need to add the API to the file search path so that our project can be linked successfully.

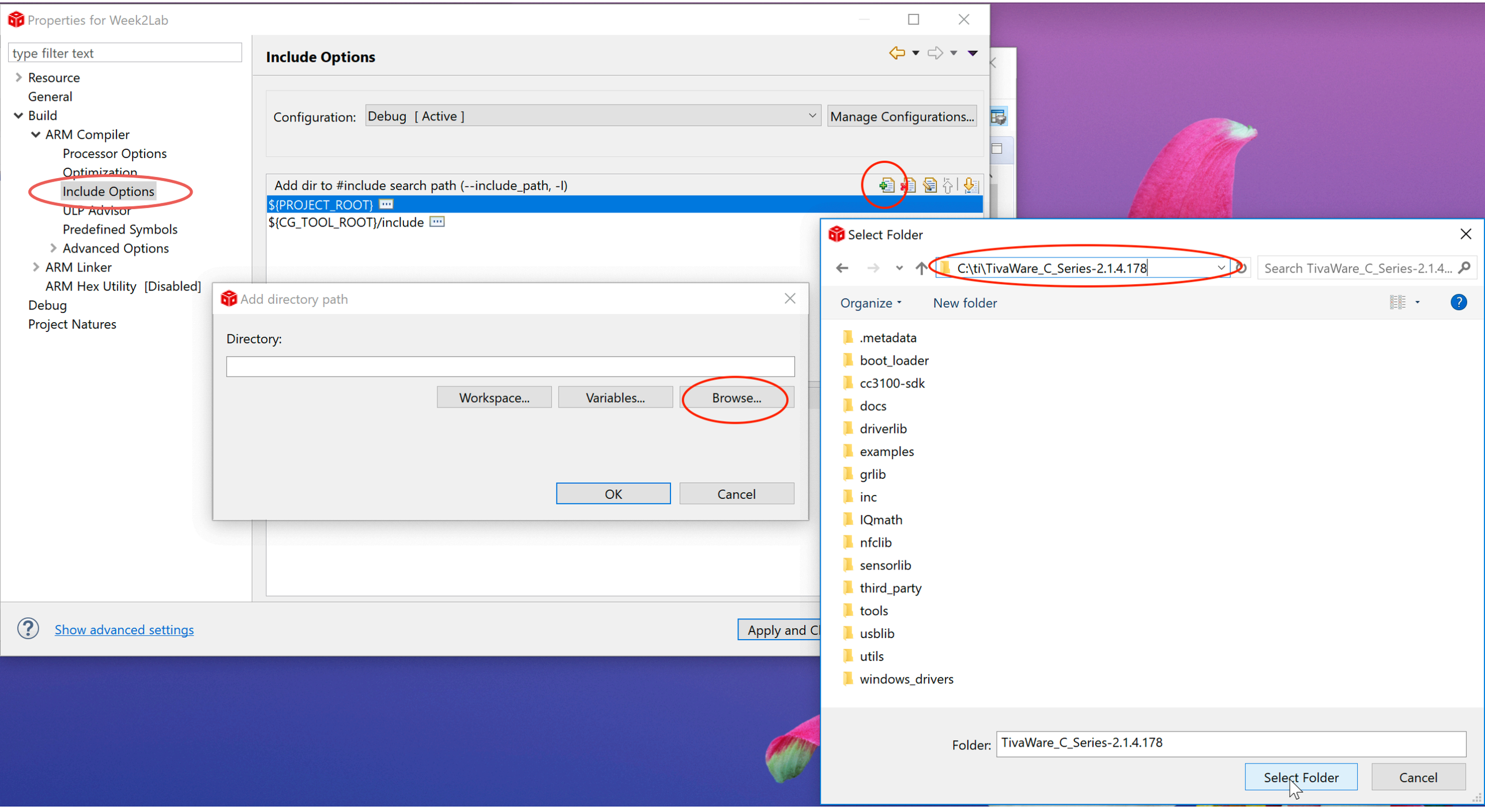
Ciaran Moore
ciaran.moore@canterbury.ac.nz

Setting up CCS - #include TivaWare API

1. Click Project>Properties
2. Select Build>ARM Compiler>Include Options in the left pane.
3. Click the "Add directory path" button (green plus icon).
4. Then click "Browse…" & navigate to `C:\ti\TivaWare_C_Series-2.1.4.178`.
5. Click "Select Folder" & "OK".

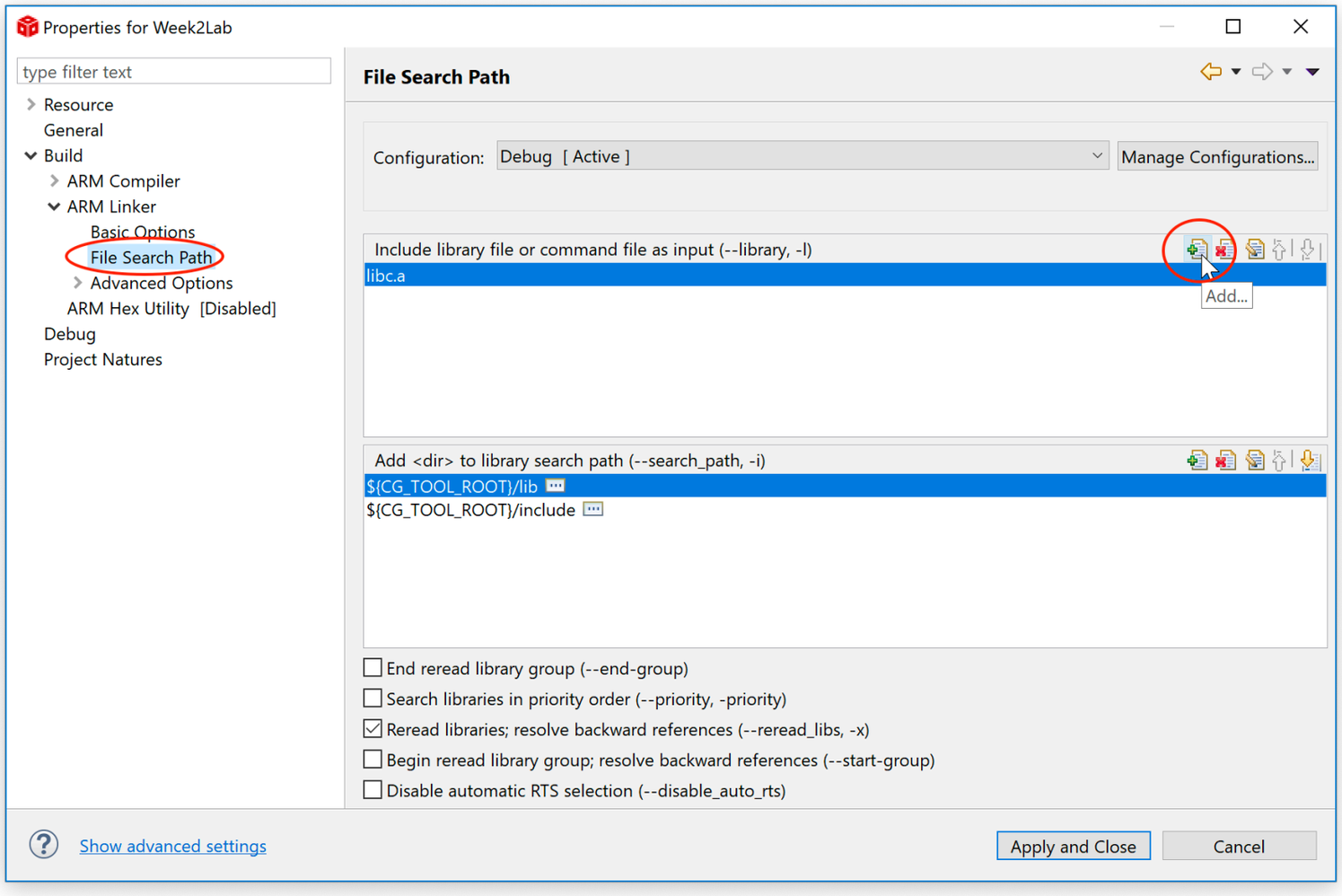
Ciaran Moore
ciaran.moore@canterbury.ac.nz

Setting up CCS - #include TivaWare API



Setting up CCS - add API to file search path

1. Click Project>Properties
2. Select Build>ARM Linker>File Search Path in the left pane.
3. Click the "Add file path" button (green plus icon).
4. Then click "Browse…" & navigate to `C:\ti\TivaWare_C_Series-2.1.4.178\driverlib\ccs\Debug\driverlib.lib`
5. Click "Open" & "OK".

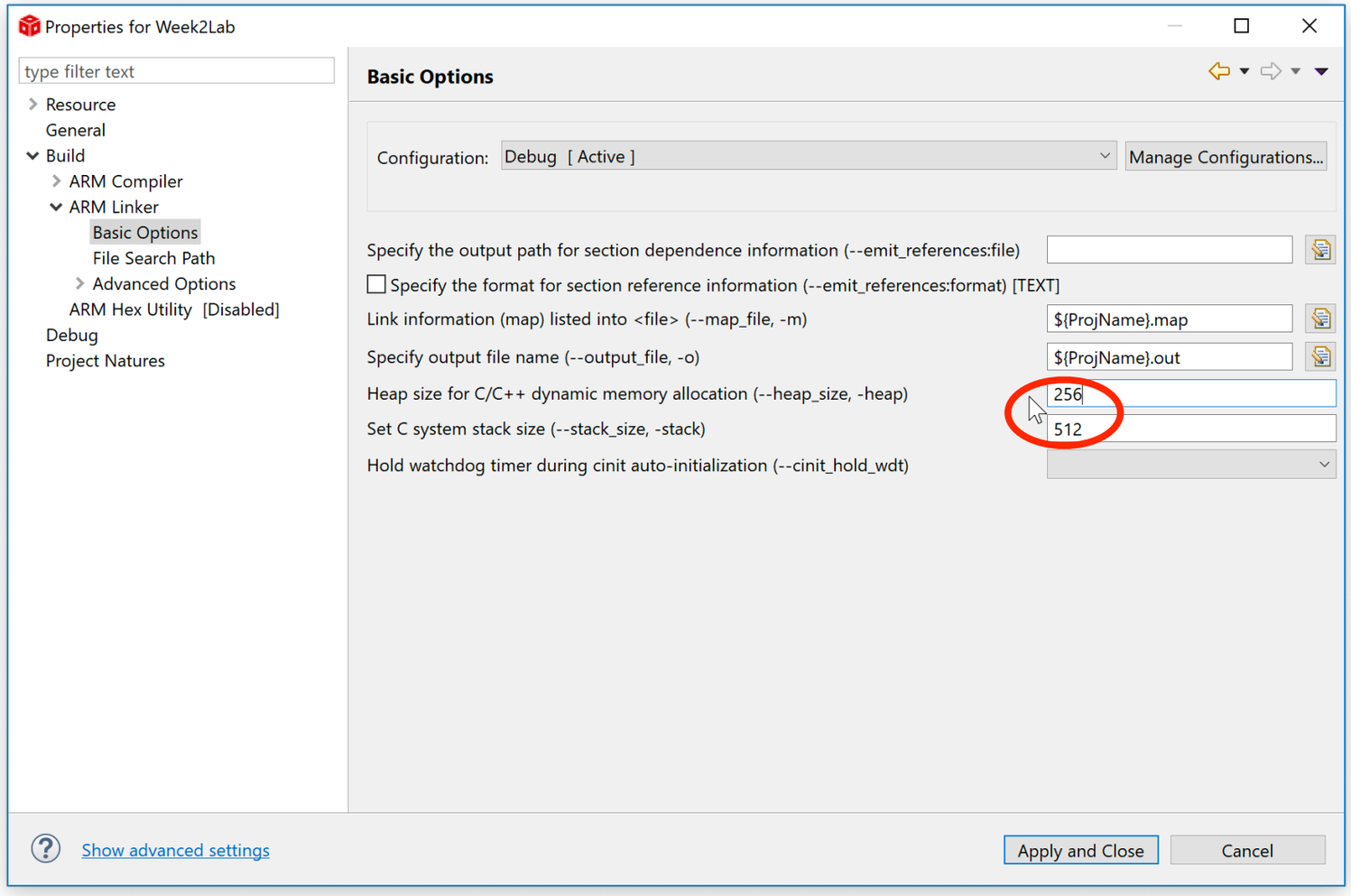


Ciaran Moore
ciaran.moore@canterbury.ac.nz

Setting up CCS – Heap and Stack Size

- For some projects, we may need to adjust the **heap size** and/or **stack size**.

1. Click Project>Properties
2. Select Build>ARM Linker>Basic Options.
3. Set Heap size in bytes as required.
4. Set stack size in bytes as required.
5. Click OK.



Ciaran Moore
ciaran.moore@canterbury.ac.nz

Testing Your Code

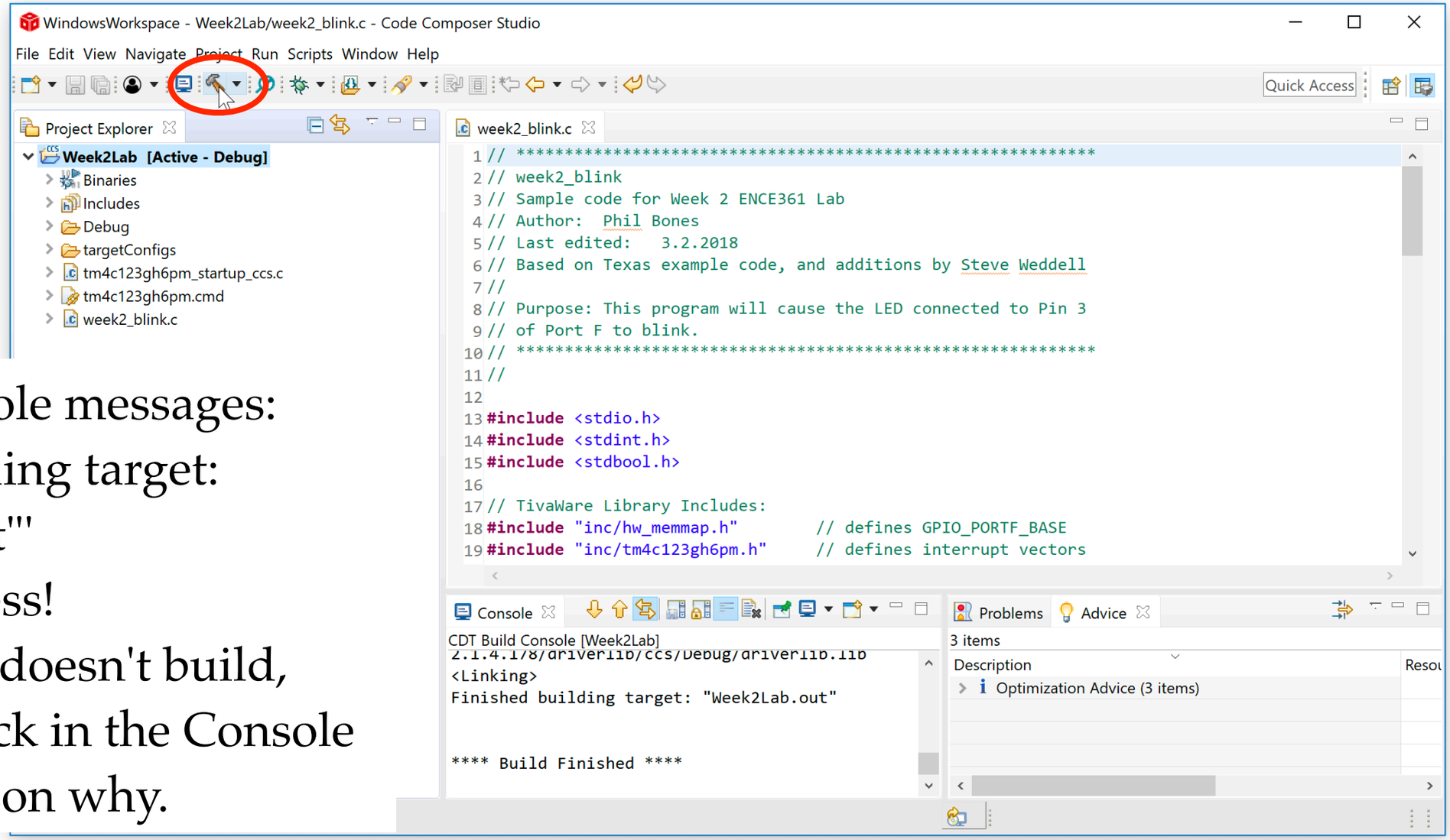
- There are two steps to testing your code:

1. Compile & Link your code: this creates a .out file that can be loaded onto your microcontroller.
2. Starting a Debug session, which writes the .out file to the microcontroller and pauses execution at the start of your program.

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Compiling and Linking

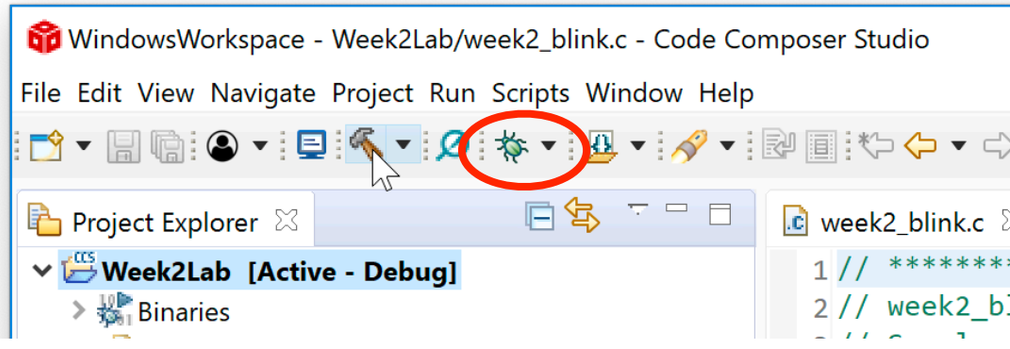
- To *build* (compile and link) your project, select Project>Build Project.
- Alternatively, click the hammer icon.



- Note the Console messages:
  'Finished building target: "Week2Lab.out"'
  indicates success!
- If your project doesn't build, use the feedback in the Console to find the reason why.

Debugging

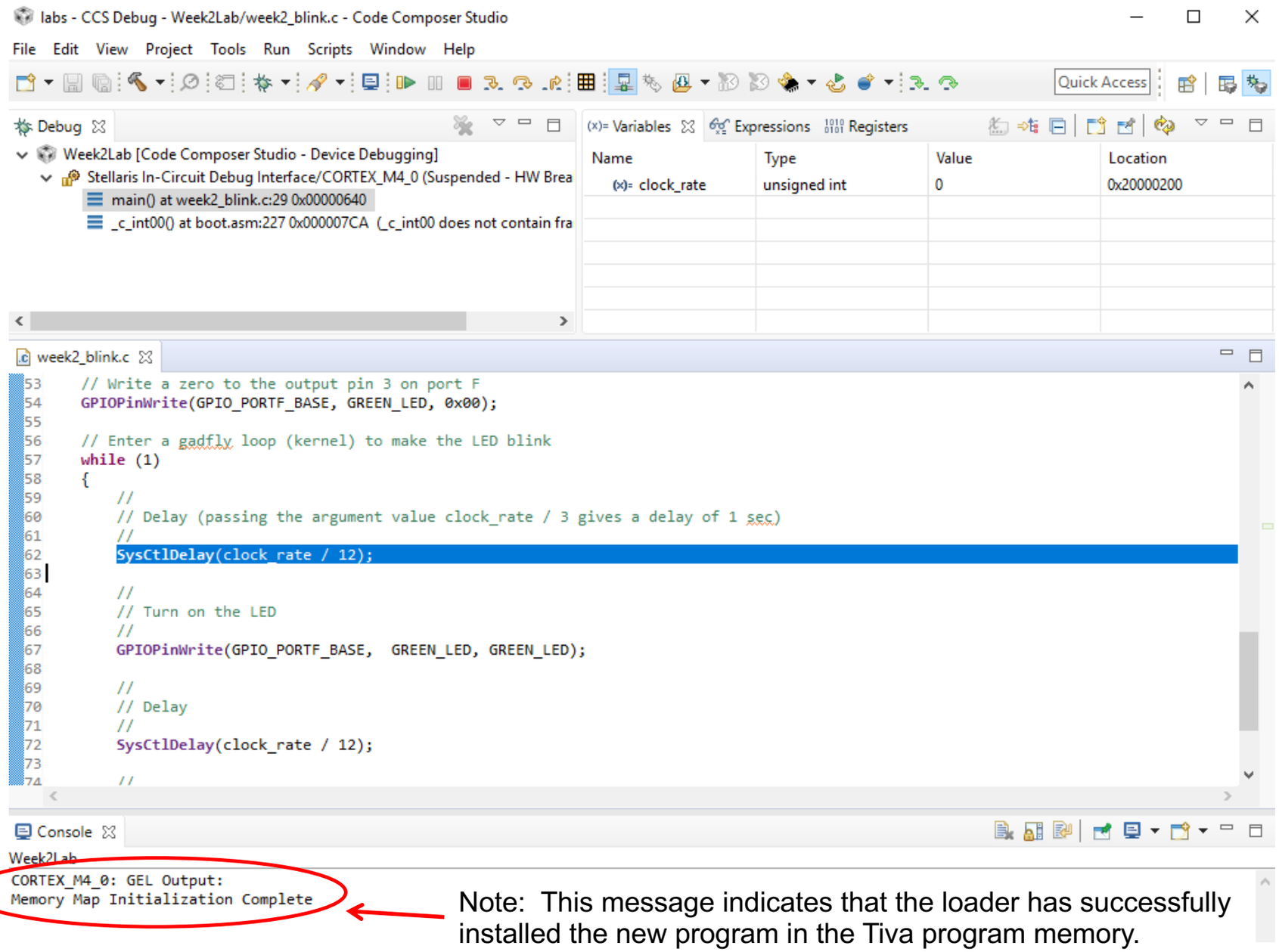
1. After unwrapping your Tiva board and checking the contents, use the USB cord to connect your board to your bench PC.
   - The device driver should find the board automatically. If not, consult a TA.
2. Select Run>Debug. Alternatively, click the bug icon.
   - If you get asked for a "New Target Configuration", choose "Yes", "Stellaris In-Circuit Debugger" and "Tiva TM4C123GH6PM". The file **NewTargetConfiguration.ccxml** will appear in your project directory.
   - Alternatively, you may get an error when you click Debug: "CORTEX_M4_0: Error initialising flash programming…" Go to Project Properties>General and make sure "Manage the project's target-configuration automatically" is checked and select "Stellaris In-Circuit Debug Interface" from the Connection: drop down list.



Ciaran Moore
ciaran.moore@canterbury.ac.nz

Debugging

- Eventually, the display will change to show the Debugger screen:

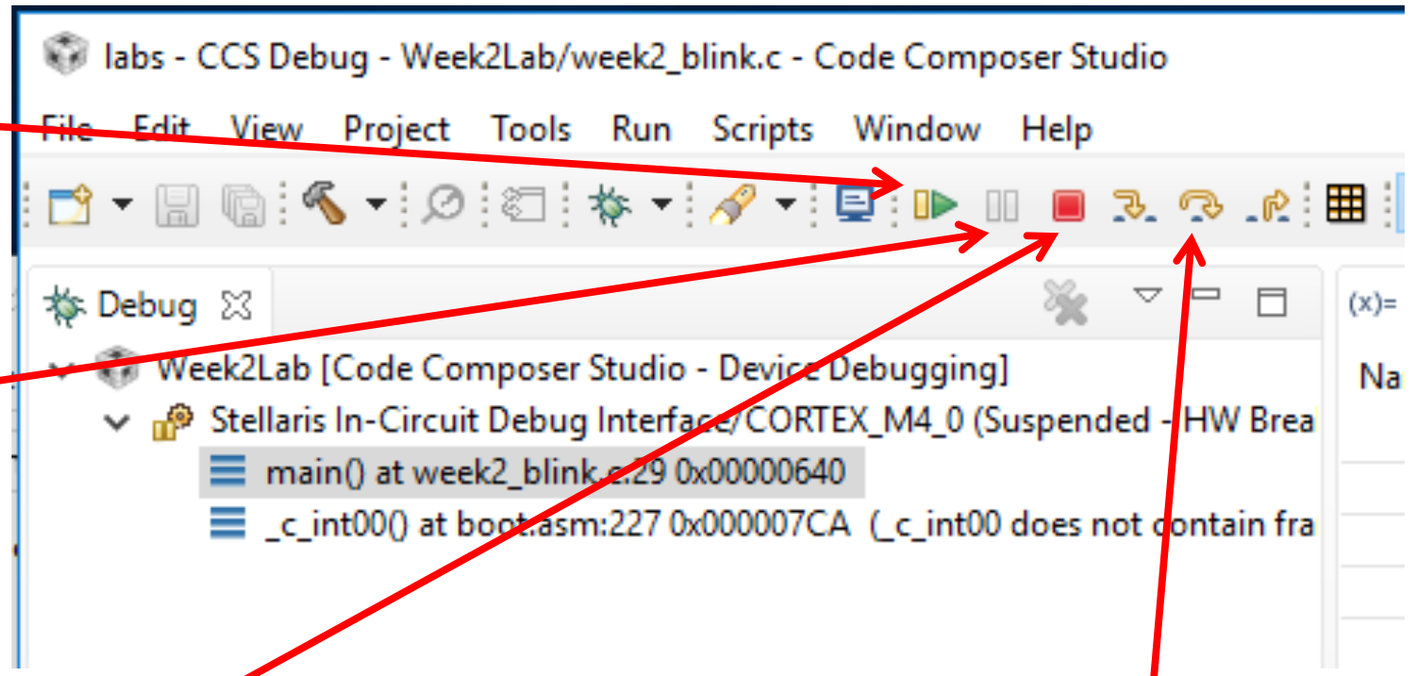


Note: This message indicates that the loader has successfully installed the new program in the Tiva program memory.

Debugging

Also see:
http://processors.wiki.ti.com/index.php/GSG:CCSv5.0_Debugging_projects

- To run your program that you have already loaded into the Tiva memory, click-on the green arrow…



- To stop the program, click on the two yellow parallel lines…

- Close the debugger by clicking on the red square… or go to the top right utility and switch between debug and edit mode (much faster!).

In the Help screen for CCS, select “stepping” in the index for more information.

- To single-step (“step over”), click on the right yellow arrow…

Debugging

1. With your program paused, click on the **week2_blink.c** file and find the TivaWare function GPIOPinWrite().
2. Set a breakpoint on the GPIOPinWrite() function by clicking on the left margin of the line where the function is defined. A dark blue diamond shape will appear.
3. Resume your programme by pushing F8 or clicking on the green "resume" arrow.
   - Does your program stop?
   - What is the status of the green LED?
   - What happens when you press F6 or the "step over" button?

Ciaran Moore
ciaran.moore@canterbury.ac.nz

Frequent Errors

Symptom: Compile error – **"#1965: cannot open source file "inc/hw_memmap.h"**.
Cause: **TivaWare_C_Series-2.1.4.178** directory path not included.
Solution: Project Properties > ARM Compiler > Include Options > Add Directory to Path… C:\ti\TivaWare_C_Series-2.1.4.178.

Symptom: Compile error – **"#10234-D null: unresolved symbols remain"** also **"unresolved symbol ADCIntClear, first referenced in ./ADCdemo1.obj"**
Cause: driverlib.lib not included as input.
Solution: Project Properties > ARM Linker > File Search Path > C:\ti\TivaWare_C_Series-2.1.4.178\driverlib\ccs\Debug\driverlib.lib.

Symptom: Can compile but can't load code onto Tiva: "Load program Error."
Cause: No ICD defined.
Solution: Project Properties > General > Connection > Stellaris In-Circuit Debug Interface.

Symptom: Can compile but can't load code onto Tiva: "Error connecting to the target"
Cause: Tiva not plugged in correctly.
Solution: Check USB connection, check the PWR SELECT switch next to USB connector on Tiva board is to the right at "DEBUG", not "DEVICE".

Ciaran Moore
ciaran.moore@canterbury.ac.nz