ECEN 3360

Digital Design Lab #1

Simplicity Exercise

Spring 2020

Objective: Install and become familiarized with the Silicon Labs’ Simplicity Studio development environment

Due: Sunday, January 19th, 2020 at 11:59pm

**I- Installing Simplicity and Importing the example project:**

1. Install Silicon Labs’ Simplicity Studio 4 development environment. You can download the software from the following site:
   1. <http://www.silabs.com/products/mcu/Pages/simplicity-studio.aspx>
2. Connect your Silicon Labs Pearl Gecko to your computer via mini USB cable

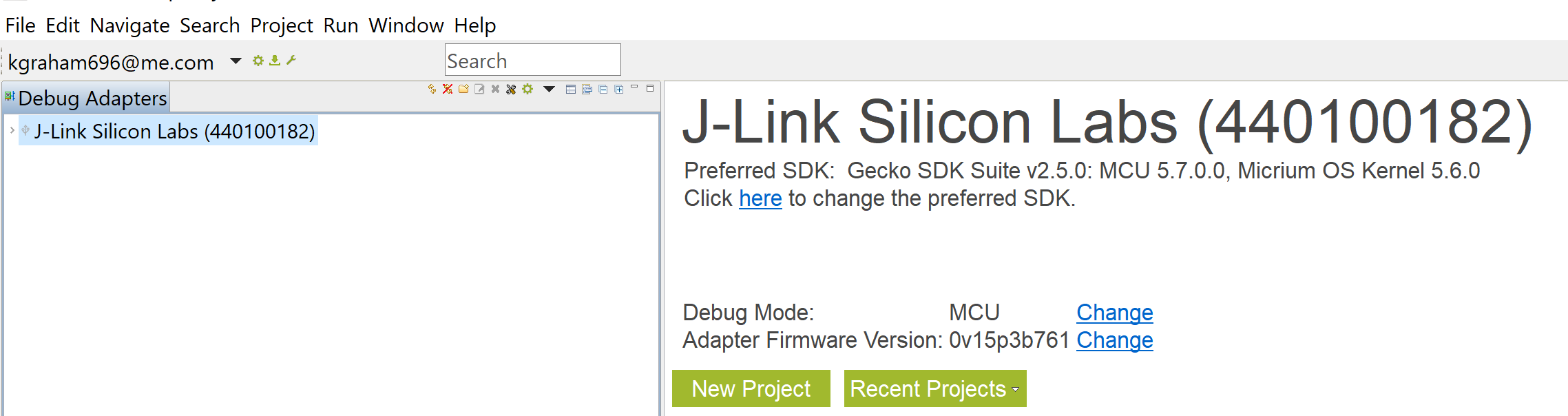


*USB Input*

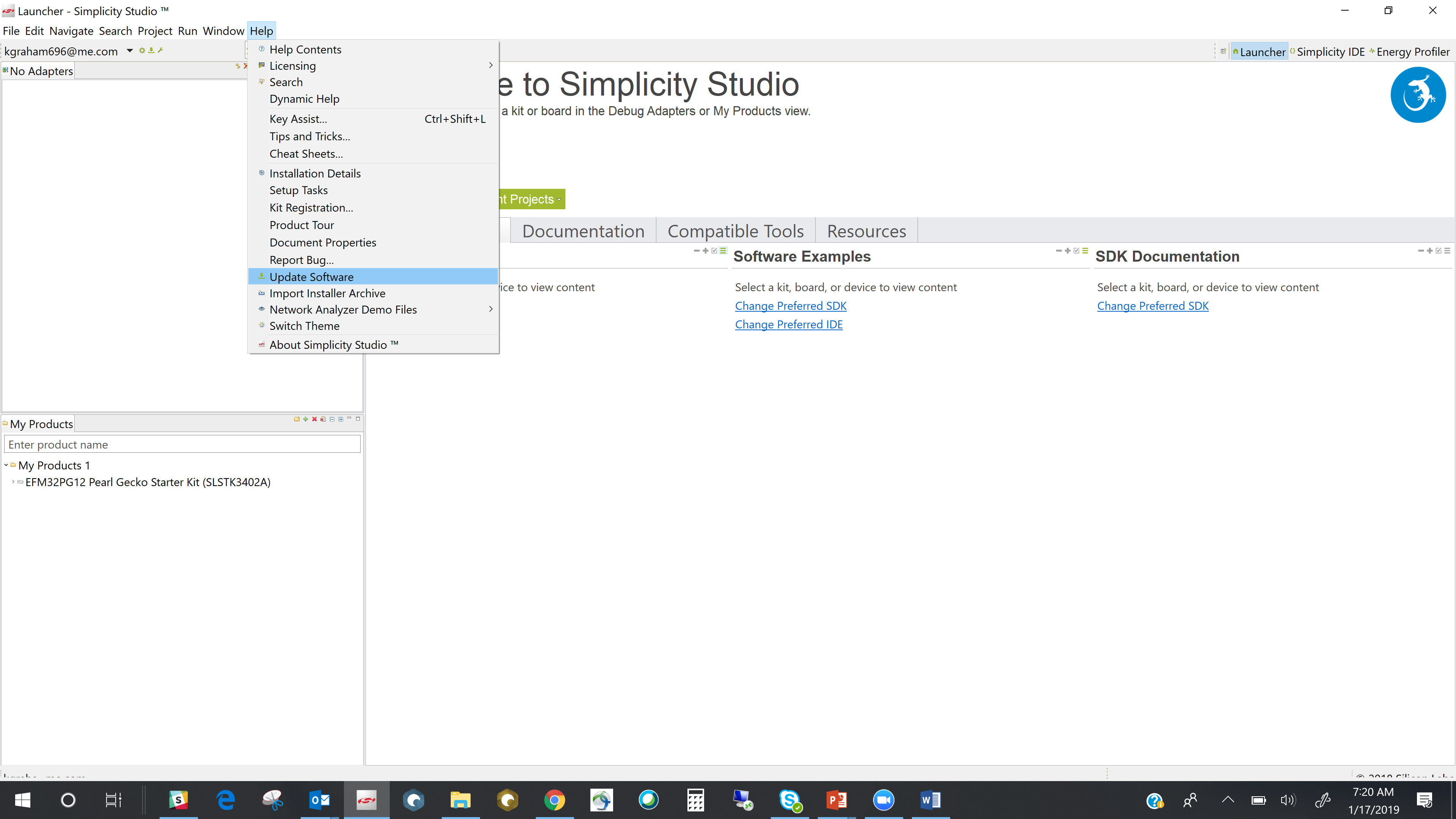
*Power Source Selection*

Silicon Labs Pearl Gecko Board

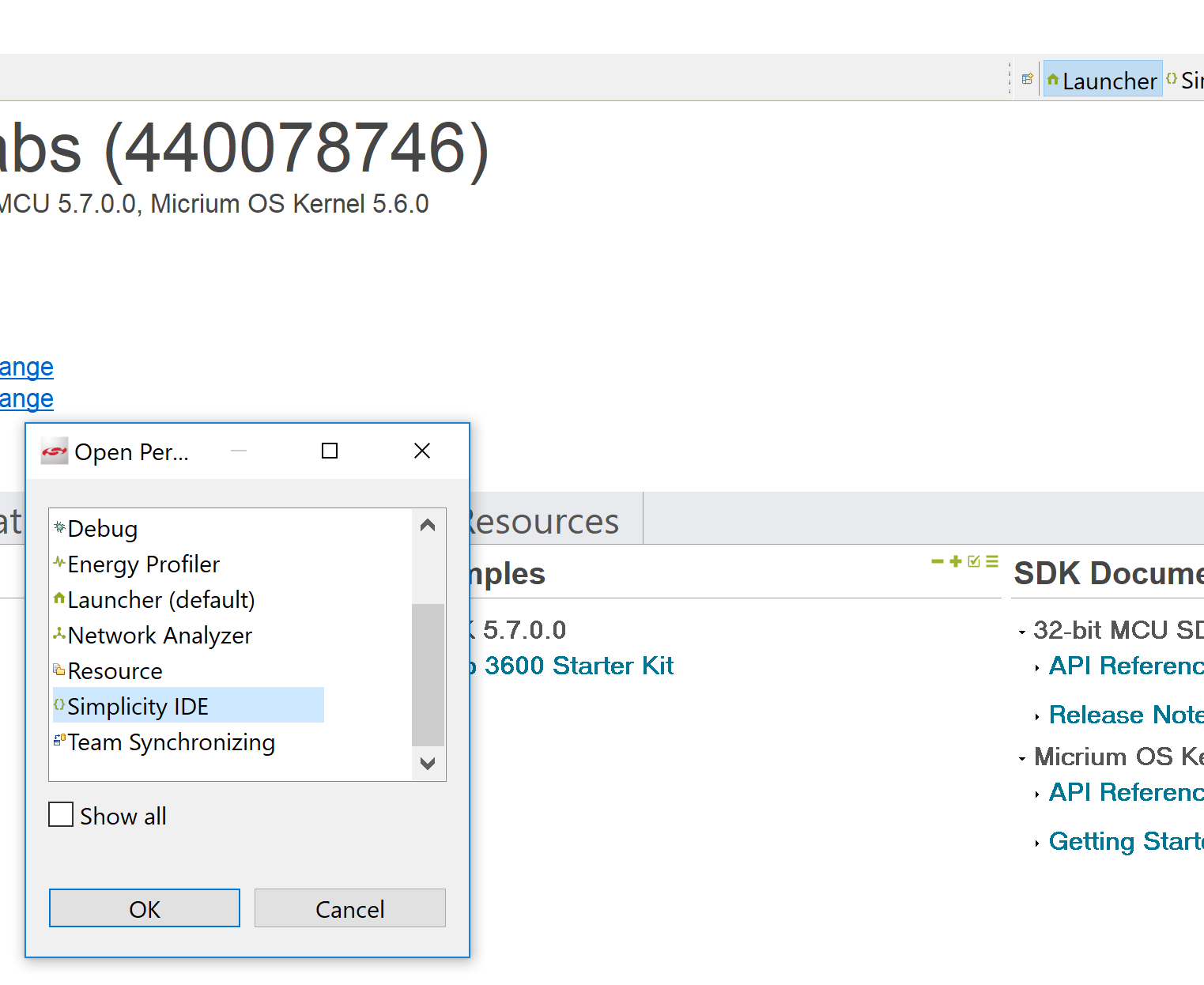
1. Ensure that the Power Source Select is to the DBG/AEM position
2. YOU WILL NEED YOUR Pearl Gecko dev kit plugged in for the next two steps to ensure that all the proper software was installed into Simplicity Studio.
   1. Make sure the J-Link or adapter firmware is up to date or installed. It may require a two-step process of download and then click install. You can determine and execute the download by being in the launcher perspective and clicking on the J-Link symbol to the left and check the middle of the screen whether the adapter is installed and to the latest revision. If not, download and install. The picture below is what it should look like when you complete the operation.



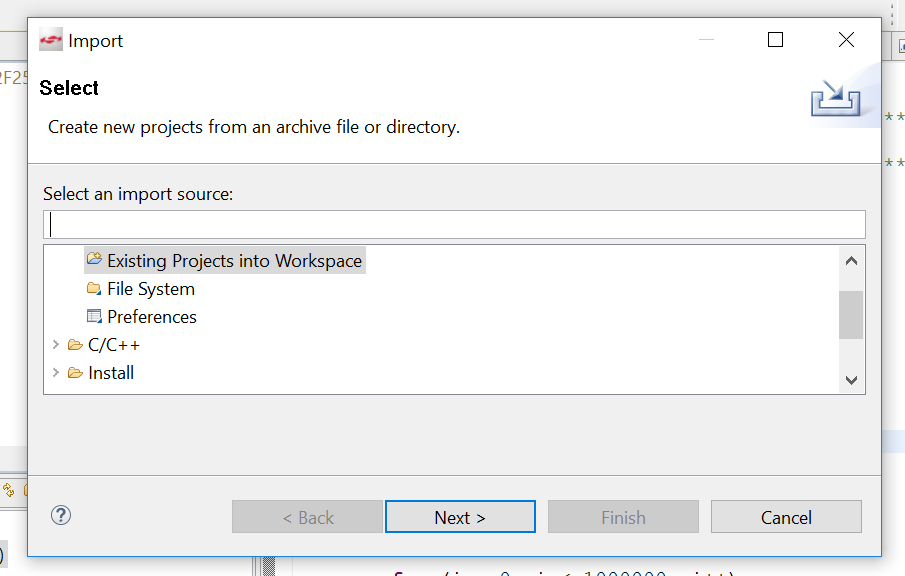
* 1. To ensure that all the software toolchains are downloaded and installed, find “update software” with your board installed and select install by device. Select your kit to move it to the right side when asked to select the device. Once you get to the screen to select the different software packages, scroll down to the bottom and if you use see packages for GNU toolchain v7.2.1, click on it to install.



1. Download the example project from Canvas in the Week 1 folder -> “Pearl\_Blinky.zip”
2. When you open Simplicity, you can open the Simplicity IDE by clicking on the menu just to the left of the Launcher selection on the menu bar.



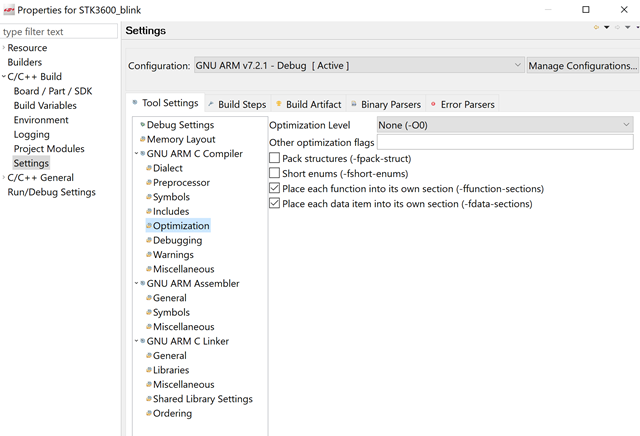
1. Now go into the Simplicity IDE by clicking on Simplicity IDE in the Menu bar
2. Select Import under File and then select “More Import Options” -> “Existing Projects into Workspace” and click on Next



1. Select the “Select Archive File” and browse to select the downloaded archive project file for this lab “Pearl\_Blinky.zip”. Then click on Finish. Now, the demo / example project should be loaded into your workspace.

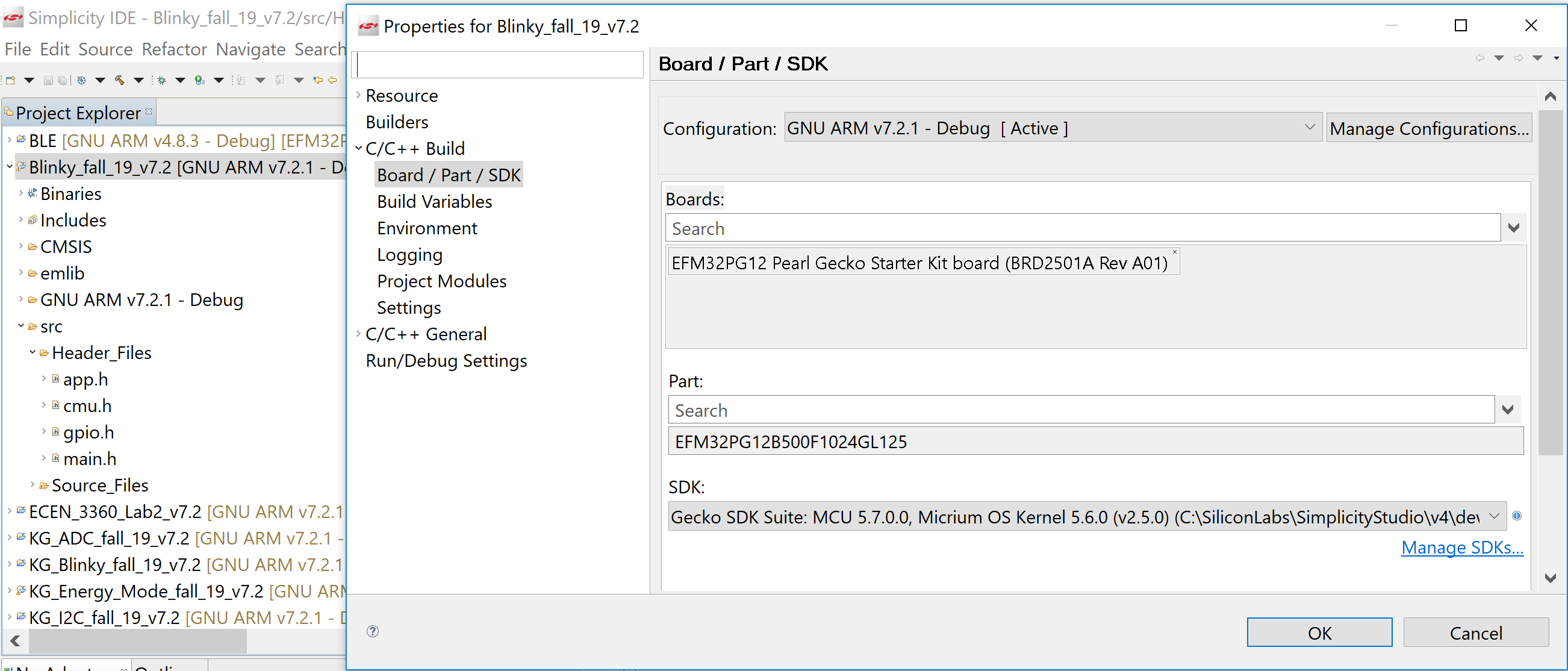
**II- Changing Code Optimization Level:**

1. Right click the project name and scroll all the way down and open Properties
2. Within properties, select Settings within C/C++ Build and set the Optimization Level to “None (-O0)” for Windows and “None” for MACs.
   1. Do not input anything in “Other Debug Flags”



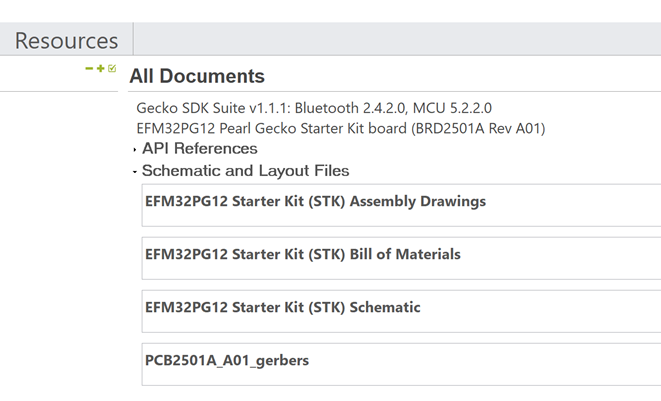
**III- Verify board SDK – MCU 5.7.3.0 or MCU 5.7.0:**

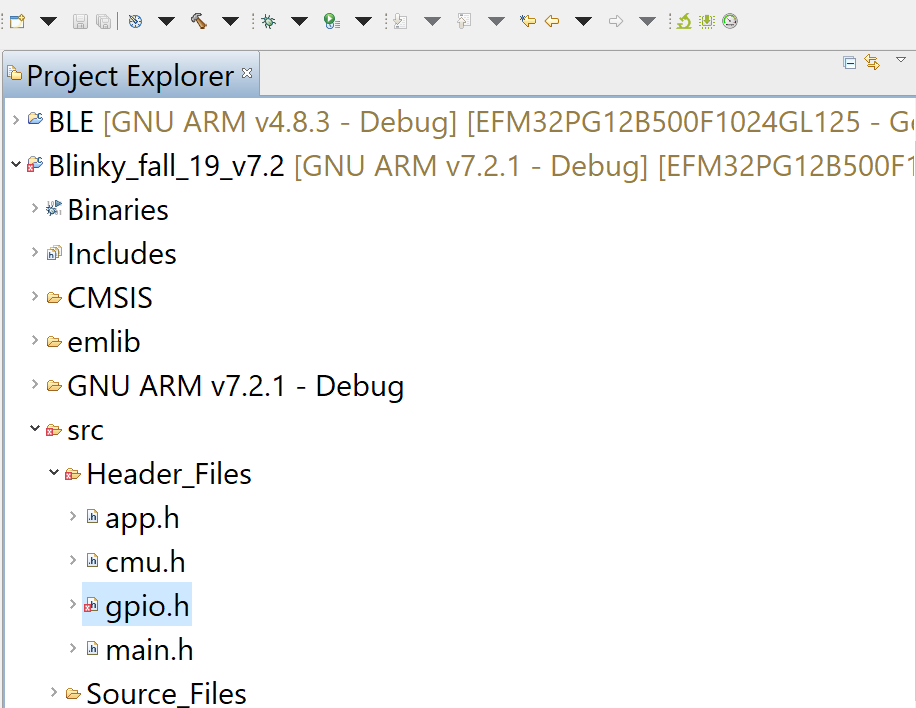
1. As you did previously to check and possibly change the optimization setting, right click your project and then click on Board / Part / SDK.
2. Verify that the settings “snap shot” below are similar to your configuration. The SDK could be MCU 5.7.3 instead of 5.7.0.



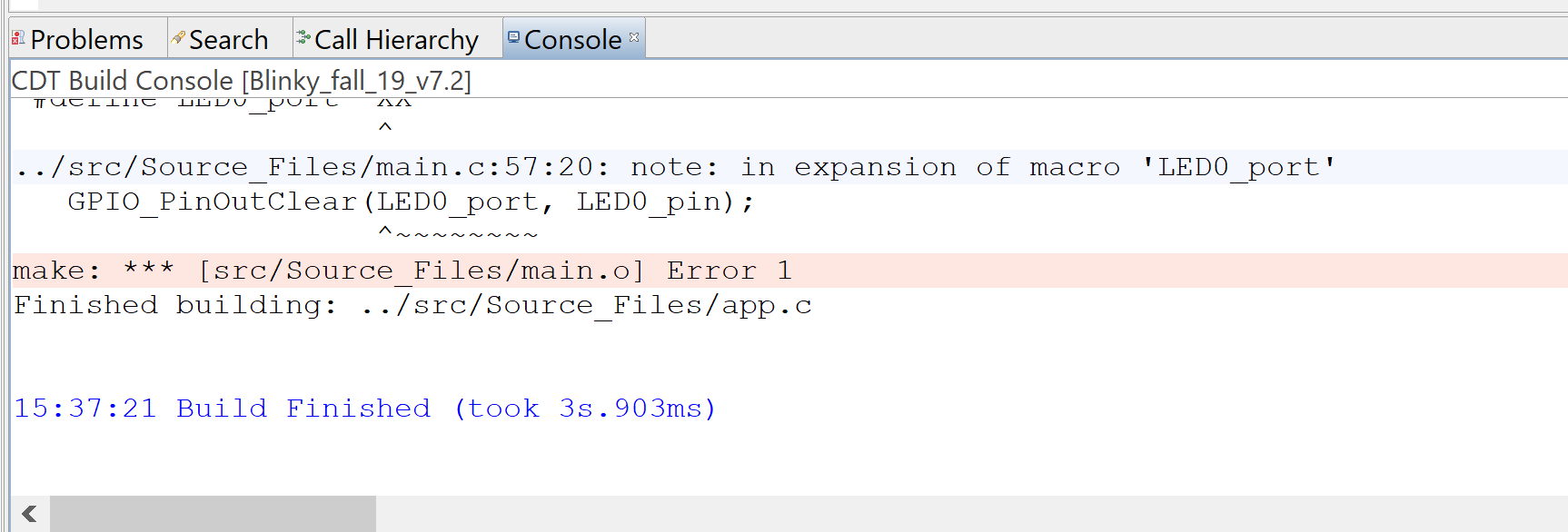
1. If your system is not configured as above, search these parameters and select the item that matches the above.
2. The most likely item you may not find here is the SDK. If you cannot find MCU 5.7.x by clicking on Manage SDKs, you will need to update your Simplicity to this SDK.
   1. To learn how to locate MCU 5.7.x to download, it is the same process as the instruction to possibly fix your build issue by downloading a different GNU toolchain described further in this document. Instead of locating the GNU tools, scroll down to find the SDK files. You may need to click on view all previous versions as well.
   2. If you have any difficulty, please contact the instructing staff immediately.

**IV- Finding and setting the LED Pins:**

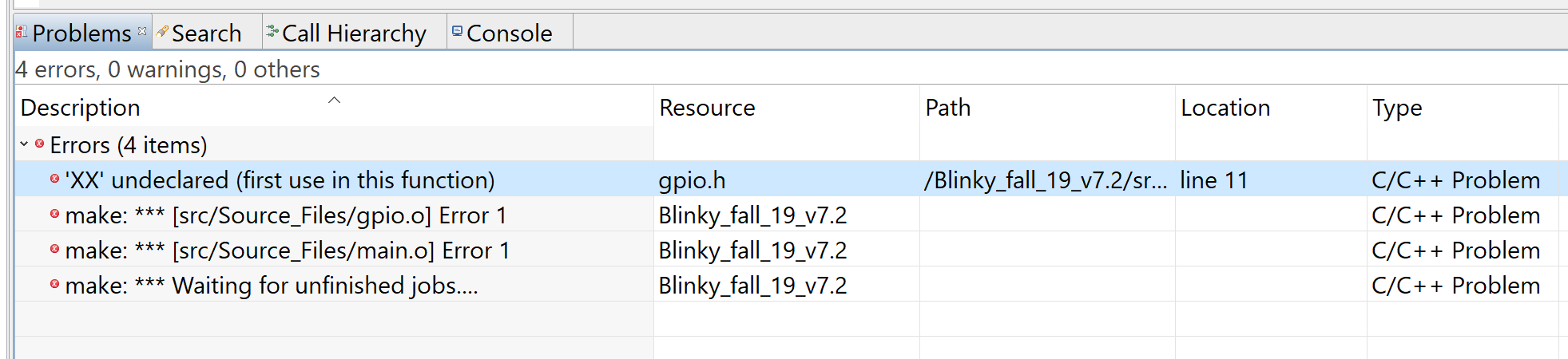
1. Go to the dev kit’s schematic and find which pins of the Pearl Gecko are connected to the LED0 and LED1 of the dev kit. You can find the schematic in Simplicity from the Launcher perspective, under the Documentation Tab.
2. DEBUG HINT:
   1. Now try to build your project by first selecting a file in your project and then clicking the build icon, the hammer, in the top menu bar



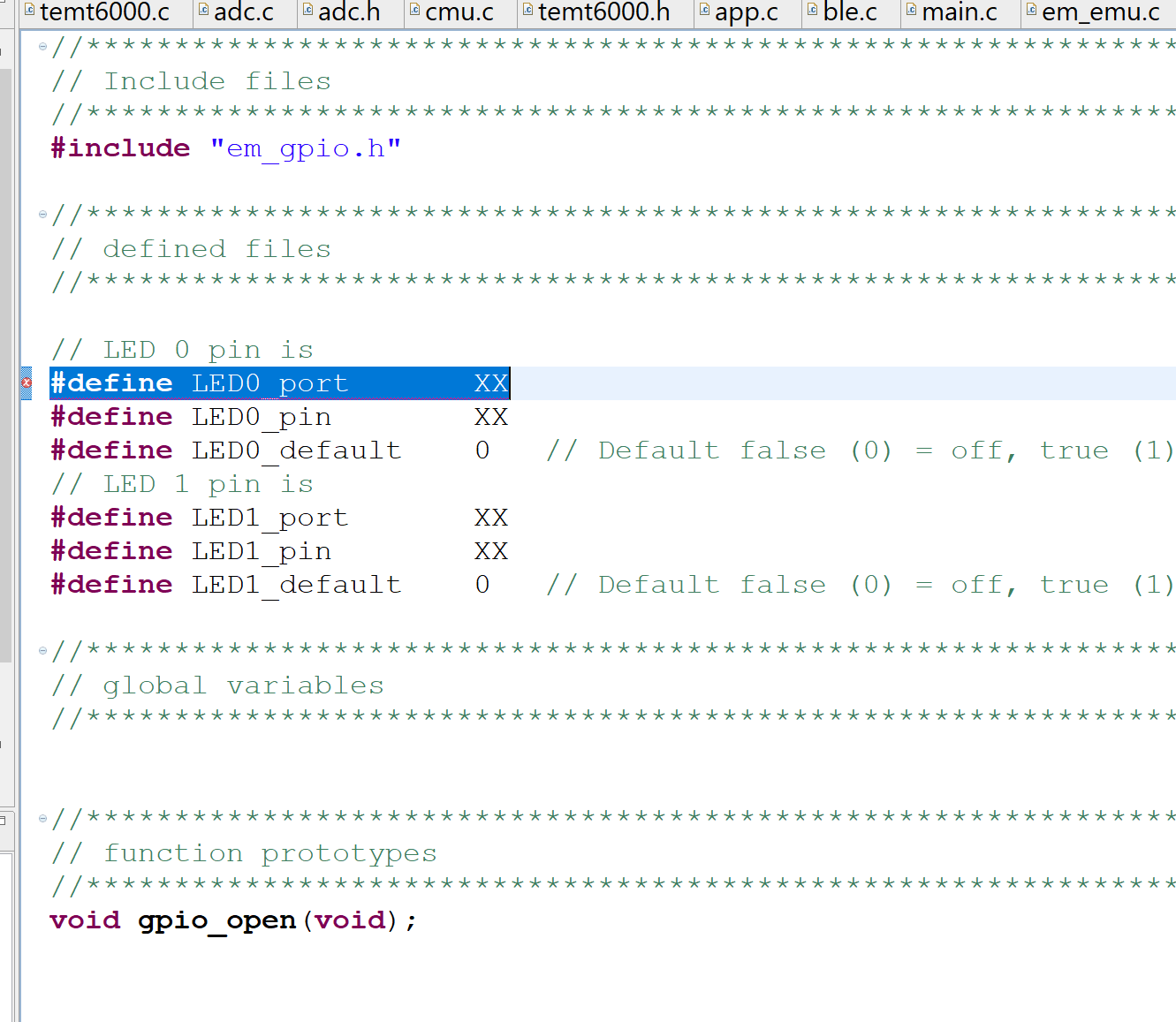
* 1. The build will fail, and you should see something similar to the below screen shot in the console window:



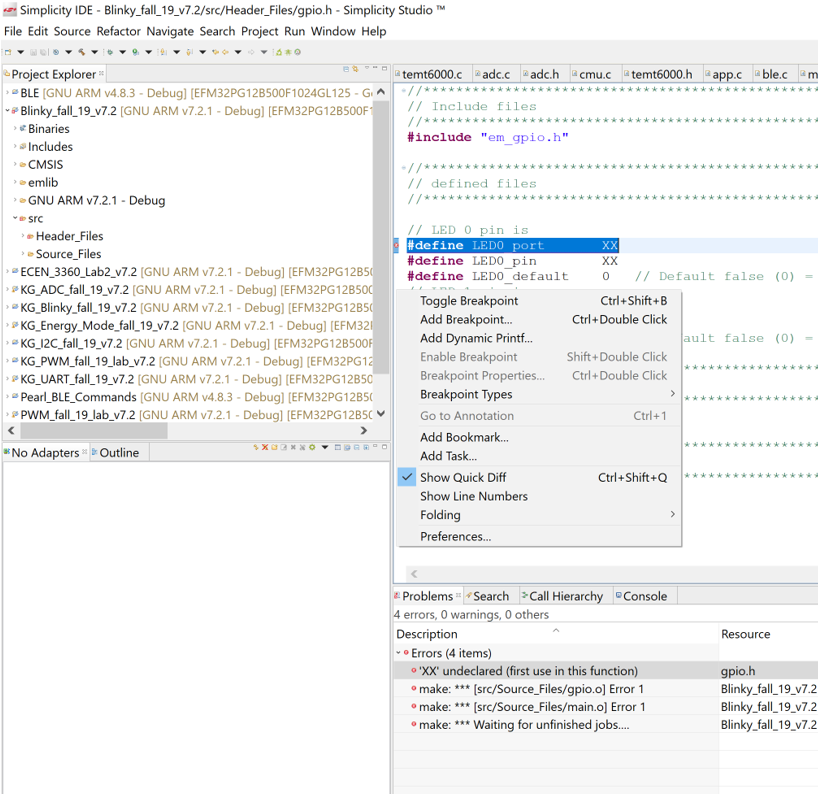
* 1. To get to the compile error quickly, click on the Problem Tab to get to a list of all the make / build errors
     1. You many need to expand “Errors” to see the list



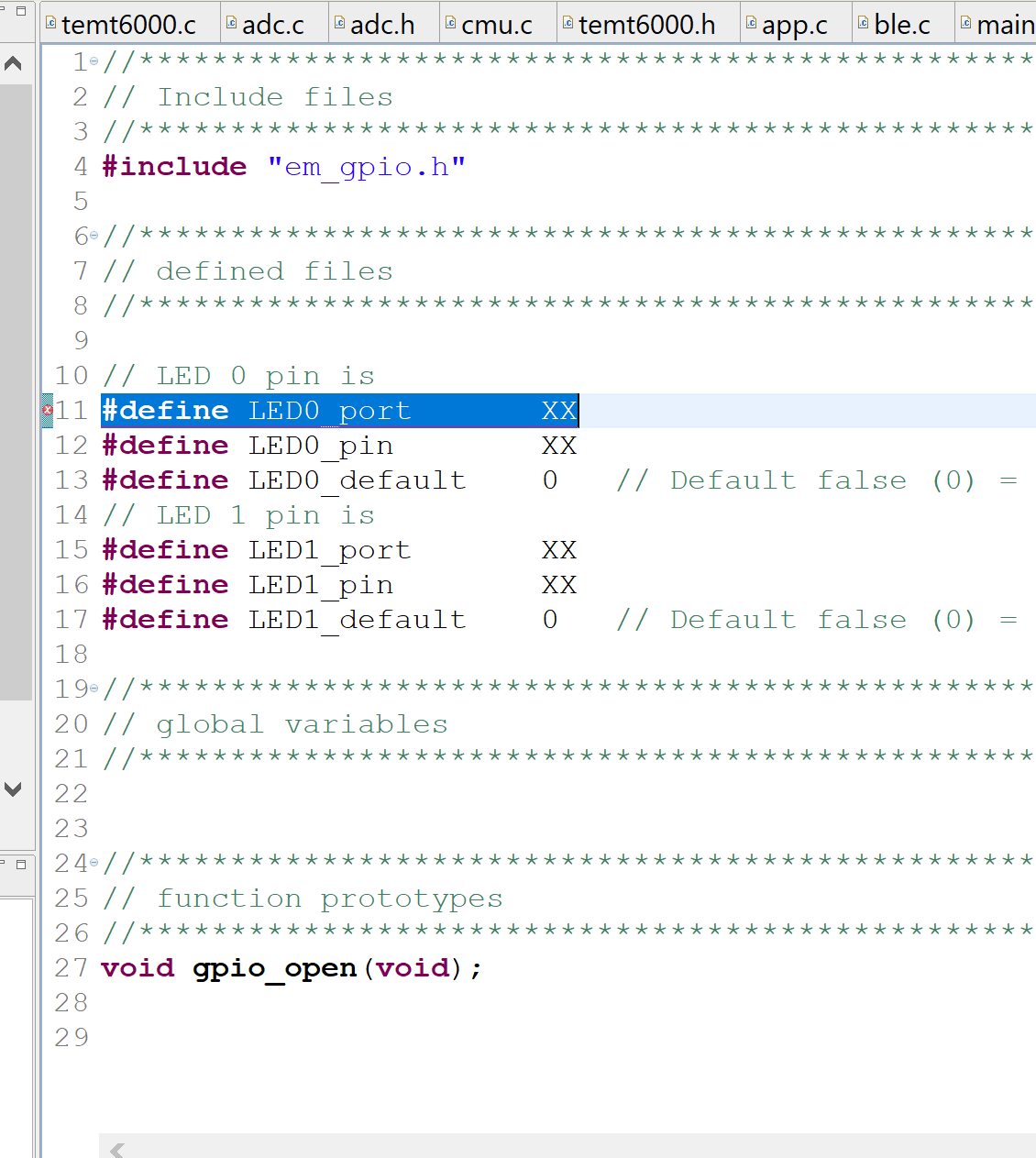
* 1. For all the non “make” errors, you can double-click the error and it will open the “Resource” file and take you to the error



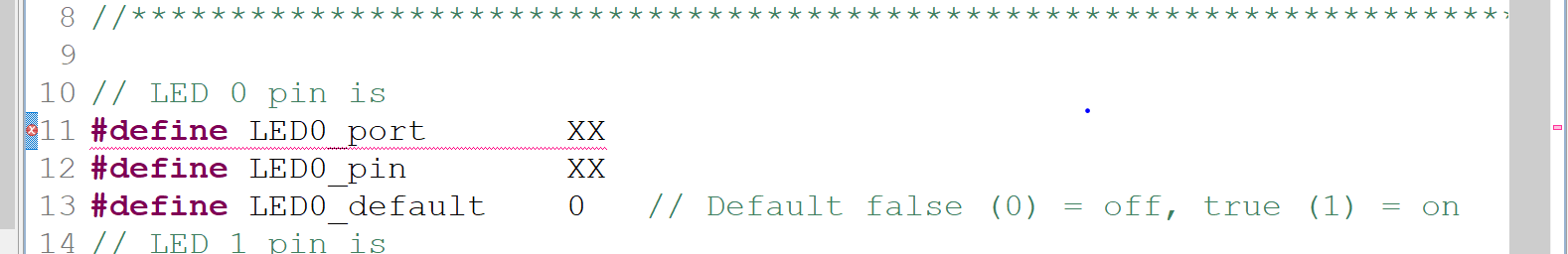
* 1. Sometimes it is helpful to know the line numbers in a file. To view line numbers in a file, right click the left edge/bar of the .c or .h file to open the menu. Select “Show Line Numbers.”



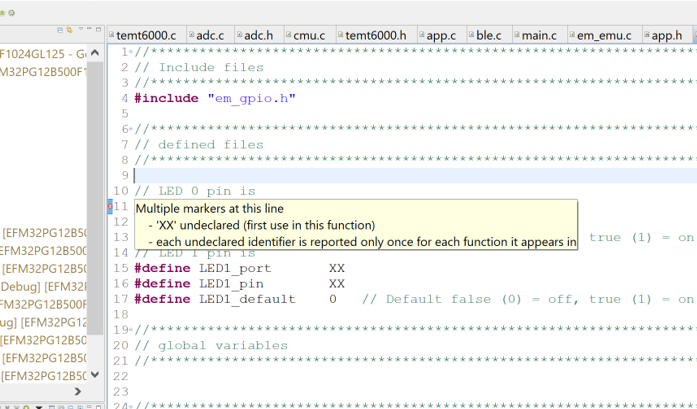
* 1. After selecting / clicking on “Show Line Numbers,” your view should now look like the below screen shot



* 1. Here is a third way of locating a compile error. On the right side of the .c or .h file, if you see a “red dot,” there is an error on the line of code. Scroll up or down until the error appears in the file window



* 1. To learn more about the error, you can hoover the cursor over the red circle with a white X to the left of the line of code to obtain a helpful popup window



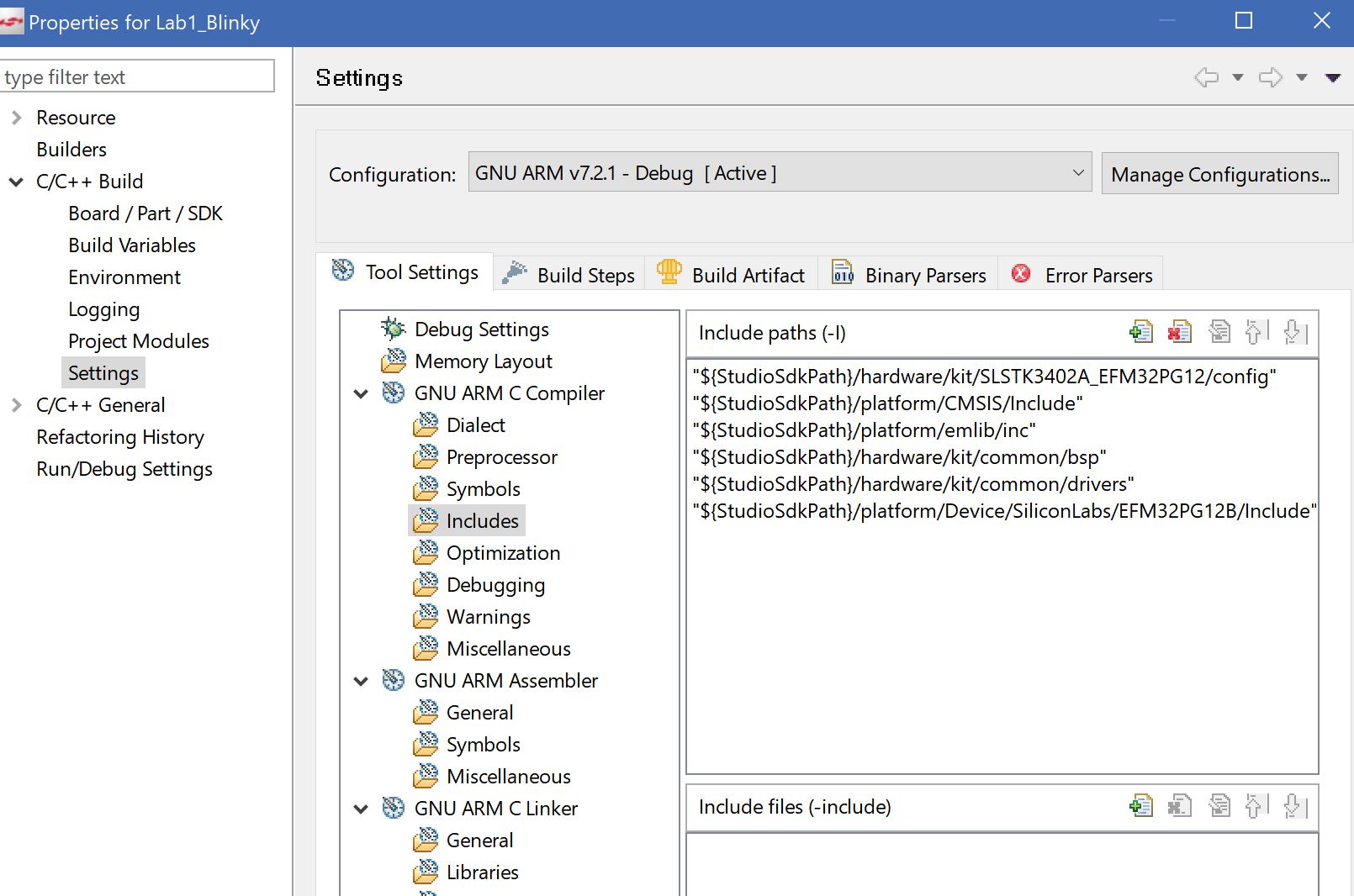
1. Once you find the Port and Pin of the LEDs, expand the Pearl\_Blinky project and then open the /src/Header\_Files folder. Next, open the gpio.h file and complete the following #define statements by tracing the connection from the LEDs to the Pearl Gecko. **Ports are defined using the enumerations of gpioPortA, gpioPortB, etc.. Pins should be unsigned integers.**

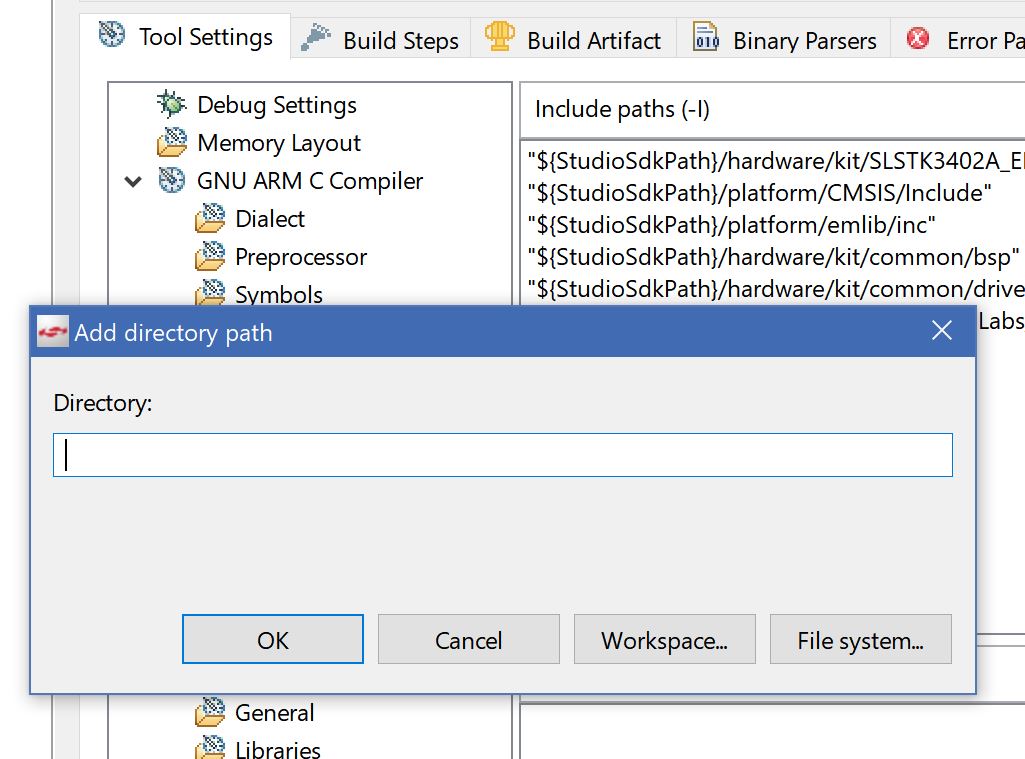
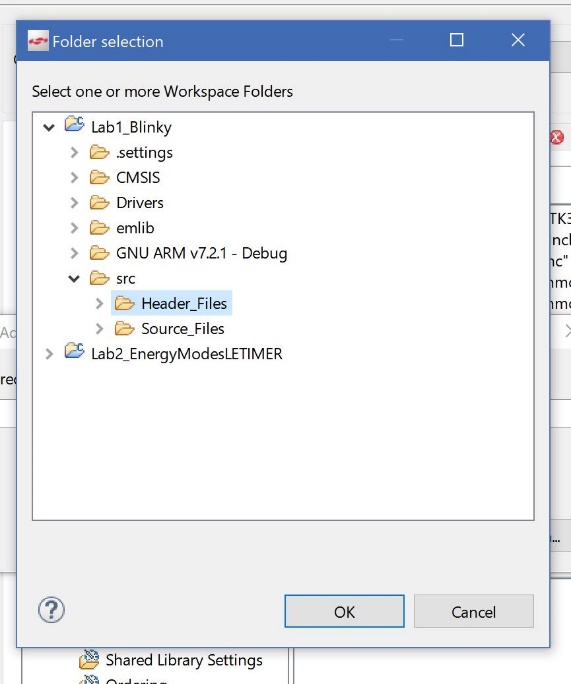
* //LED 0 pin is
* #define LED0\_Port XX
* #define LED0\_Pin XX
* #define LED0\_Default 0 //Default false (0) = off, true (1) = on
* //LED 1 pin is
* #define LED1\_Port XX
* #define LED1\_Pin XX
* #define LED1\_Default 0 // Default false (0) = off, true (1) = on

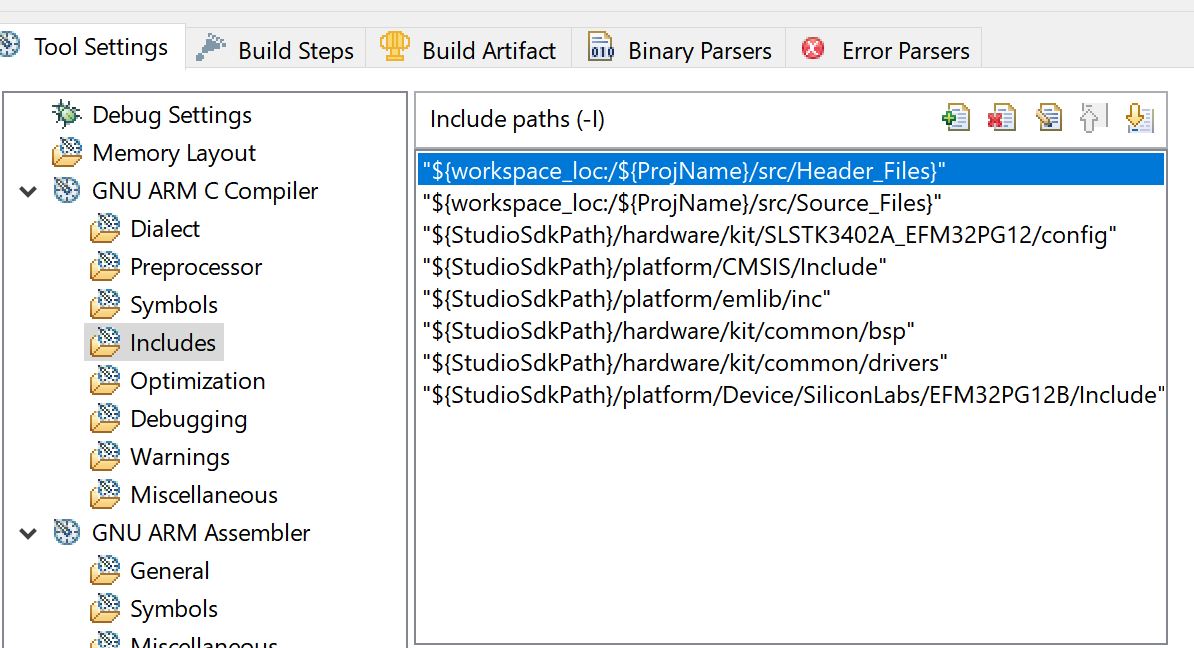
**NOTE 1:**

**If you have header file inclusion errors, follow these steps:**

1. Right click on your project and click on Properties
2. Expand C/C++ Build and click on Settings
3. Under GNU ARM C Compiler, click on Includes and then Add:



1. You should see this dialog box, click on Workspace:
2. Expand your project, and src. You should see Header\_Files and Source Files
3. Click on Header\_Files and then OK
4. Click on OK again to close the dialog box
5. Repeat the same steps to add the Source\_Files folder
6. Your screen should look like this:



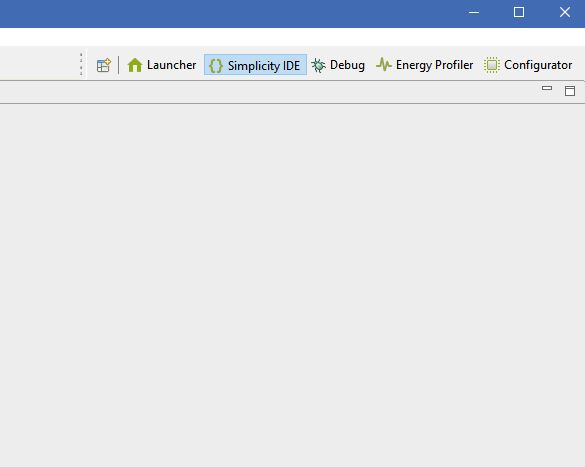
1. Next, under GNU ARM Assembler click on General
2. Repeat the same steps ( 3 – 8 )
3. Click on Apply then OK
4. Your project should now build without include errors

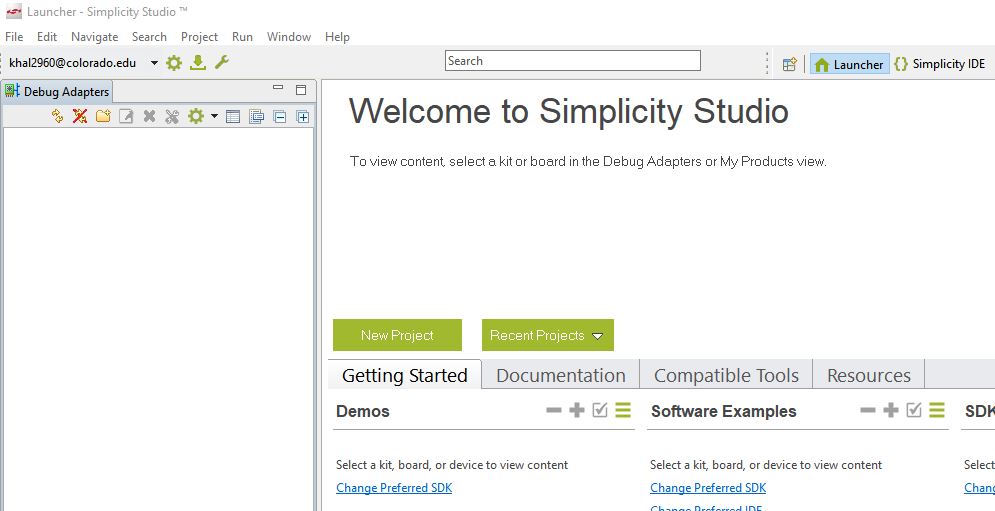
**NOTE 2:**

**If you get a “MAKE” error or similar, you might be missing a toolchain:**

To install the missing toolchain (such as GNU ARM v7.2.1) follow these steps

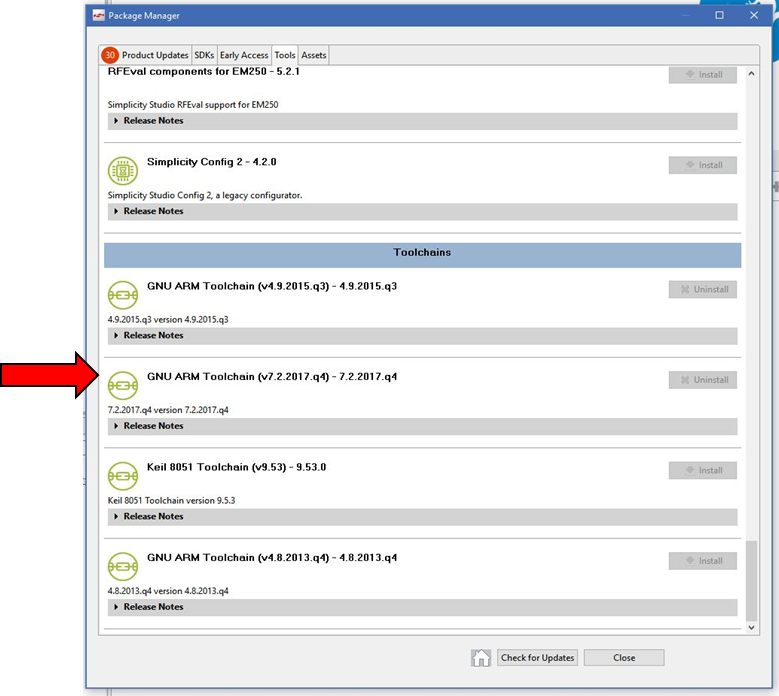
1. Click on the launcher view if you are not already on it:





At the top right, you should see an icon like the one shown below “Update Software”:

1. Once the Package Manger window appears, click on the “Tools” tab and scroll towards the end. You should see a toolchains section



1. Find GNU ARM Toolchain v7.2 and simply click on Install if it is not already installed:
2. That’s it! Your project should now build without errors.

**V- Building and Profiling:**

1. Now, click on build to build the project and under the Run pull down menu, select “Profile.”
   1. Simplicity IDE should begin to compile the project, and then download and flash the code onto the Pearl Gecko dev kit
   2. After flashing the microcontroller, the code should begin to run, and the LEDs on the dev kit should begin to flash
   3. ***You should always power cycle your board after programming to reset the Energy profiler by switching the power switch from AEM/DBG to BAT and back***
2. In the Energy Profiler, click the pause button towards the top center.
3. Reset the measurements by clicking on their respective resets. See next figure.
   1. Click the play button towards the top center to restart the Energy Profiler measurements.

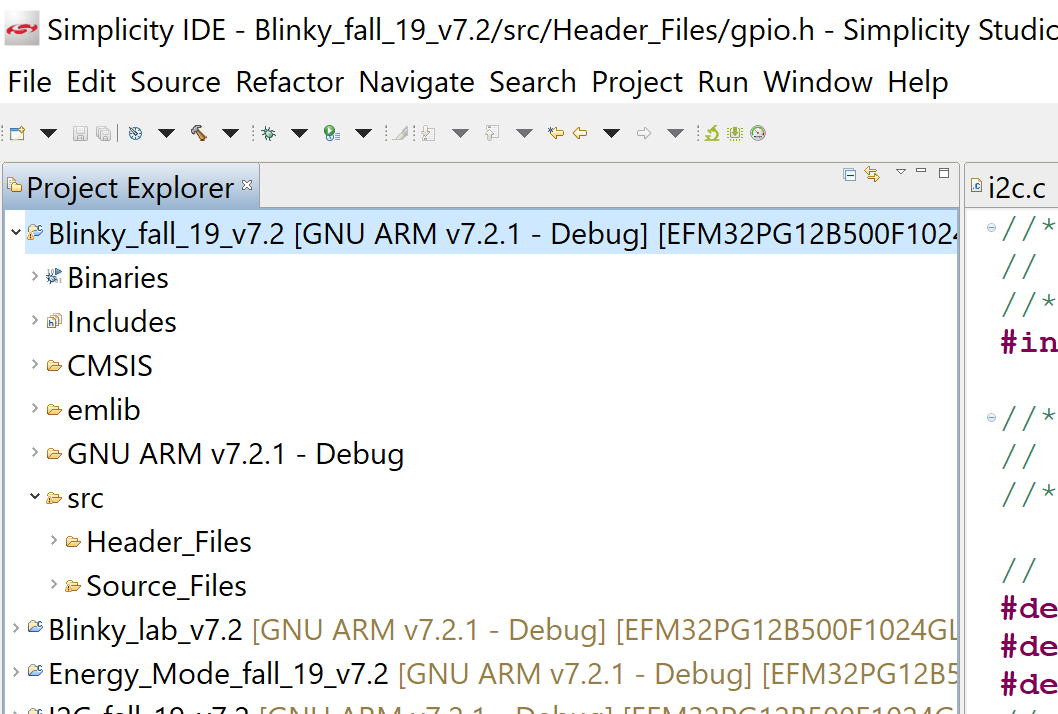


Resets

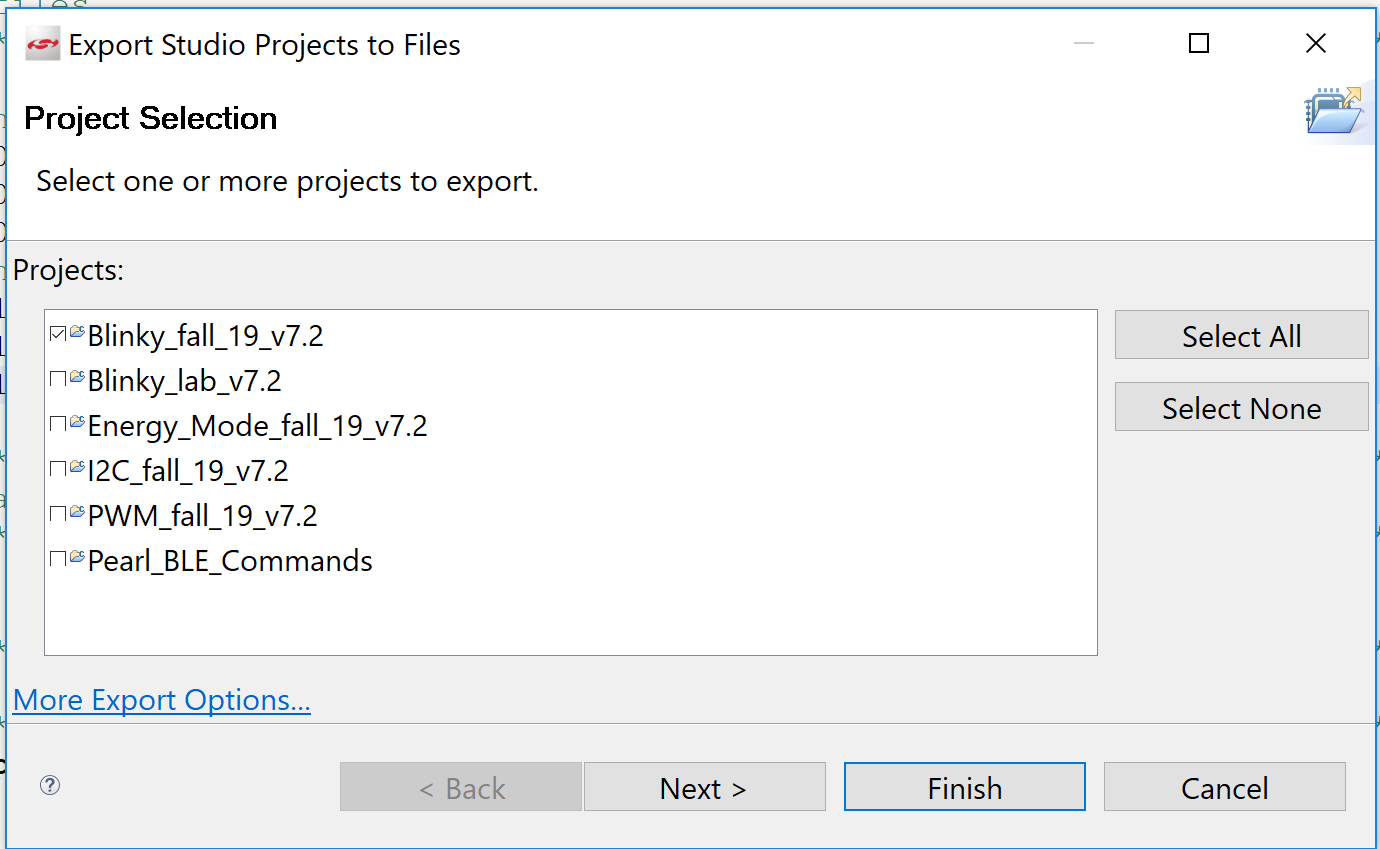
[Grab your reader’s attention with a great quote from the document or use this space to emphasize a key point. To place this text box anywhere on the page, just drag it.]Reset

[Grab your reader’s attention with a great quote from the document or use this space to emphasize a key point. To place this text box anywhere on the page, just drag it.]

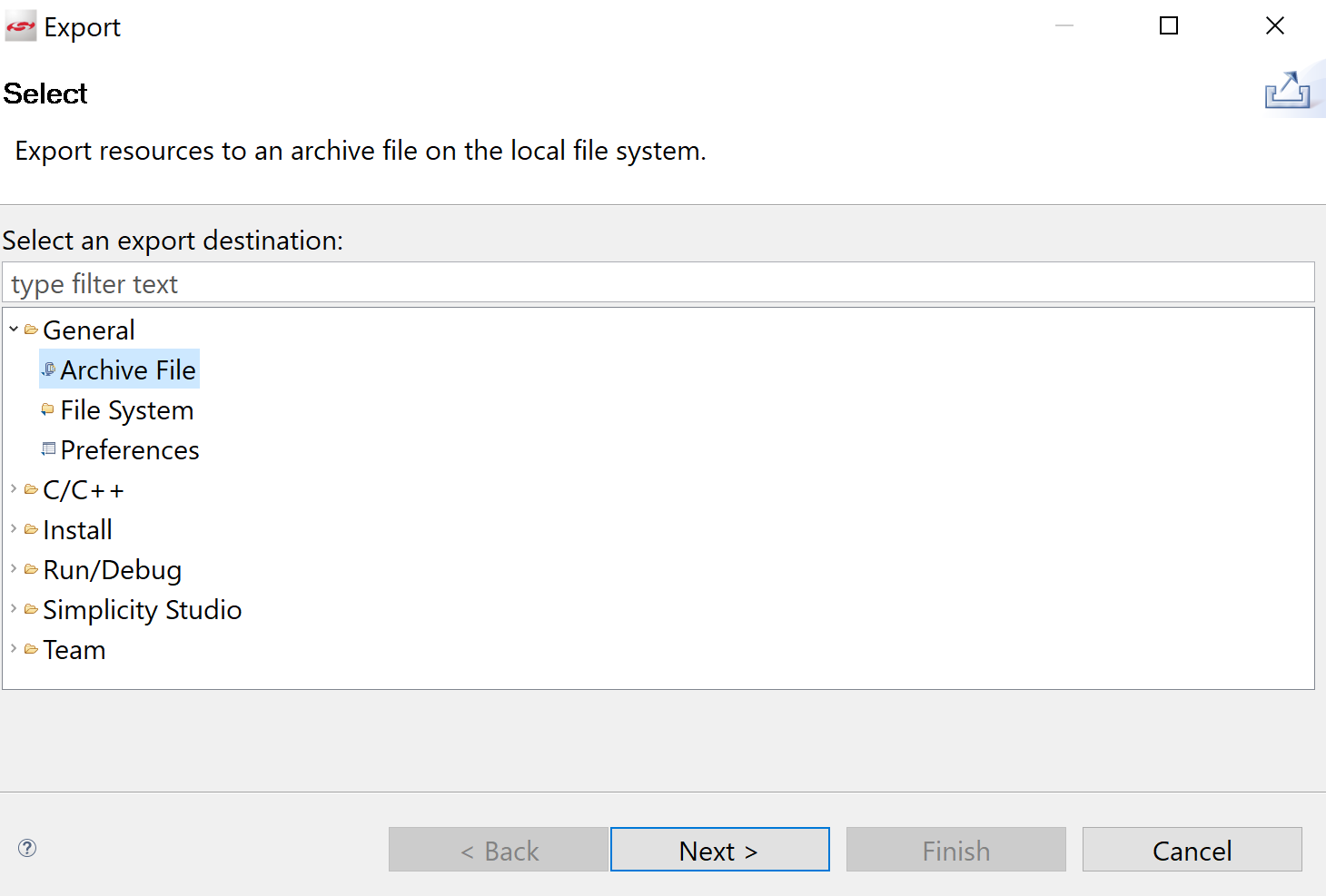
1. Click on the lower left of the vertical access to change the scale from logarithmic to linear. Linear is indicated by a straight line at a 45 degree angle while logarithmic is indicated by an arc. Use the linear mode for all measurements in this course. WE WILL BE USING THE LINEAR MODE FOR ALL MEASUREMENTS IN THIS COURSE UNLESS OTHERWISE SPECIFIED!
2. To get accurate current readings, after each code download, you must move the Power Select Switch on the Pearl Geck dev kit completely to the left and then back to the right.
   1. Failure to perform this power reset can affect your Lab worksheet scores
3. You can zoom in and out the vertical axis, current measurement, via the + and – buttons located on the left side approximately ¼ down from the top. The circular arrow will reset the scale to the default current / division setting.
4. You can zoom in and out the horizontal axis, time, via the + and – buttons located at the bottom right corner. The circular arrow will reset the scale to the default time / division setting.
5. Pausing the Energy Profiler, with zooming in while using the linear scale, take the average current measurement to determine the following:
   1. Total current drawn when both LEDs are OFF
   2. Total current drawn when one LED is ON
   3. Total current drawn when both LEDs are ON
6. How much current is one LED consuming?
7. In gpio.c, comment out these lines of code:
   1. GPIO\_DriveStrengthSet(LED0\_Port, gpioDriveStrengthStrongAlternateStrong);
   2. GPIO\_DriveStrengthSet(LED1\_Port, gpioDriveStrengthStrongAlternateStrong);
8. In gpio.c, uncomment these lines of code:
   1. GPIO\_DriveStrengthSet(LED0\_Port, gpioDriveStrengthWeakAlternateWeak);
   2. GPIO\_DriveStrengthSet(LED1\_Port, gpioDriveStrengthWeakAlternateWeak);
9. Rebuild/compile and run the updated program. ***Remember to always switch the power connector from AEM/DBG to BAT and back to reset the Energy Profiler***
10. Pausing the Energy Profiler, with zooming in with the linear scale, take the average current measurement to determine the following:
    1. Total current drawn when both LEDs are OFF
    2. Total current drawn when one LED is ON
    3. Total current drawn when two LEDs are ON
11. How much current is one LED consuming?
12. Now go back into the Simplicity IDE, and comment out the following line of code in the main.c routine
    1. GPIO\_PinOutSet(LED1\_port, LED1\_pin); => //GPIO\_PinOutSet(LED1\_port, LED1\_pin);
13. Now click on Run for Simplicity Studio to compile, flash, and start the updated program.
14. To complete the assignment, you must export your project and upload it into the Lab 1 Assignment in the course Canvas website.
    1. Export the project by right clicking your project in the IDE perspective and then selecting Export from the main menu File pulldown



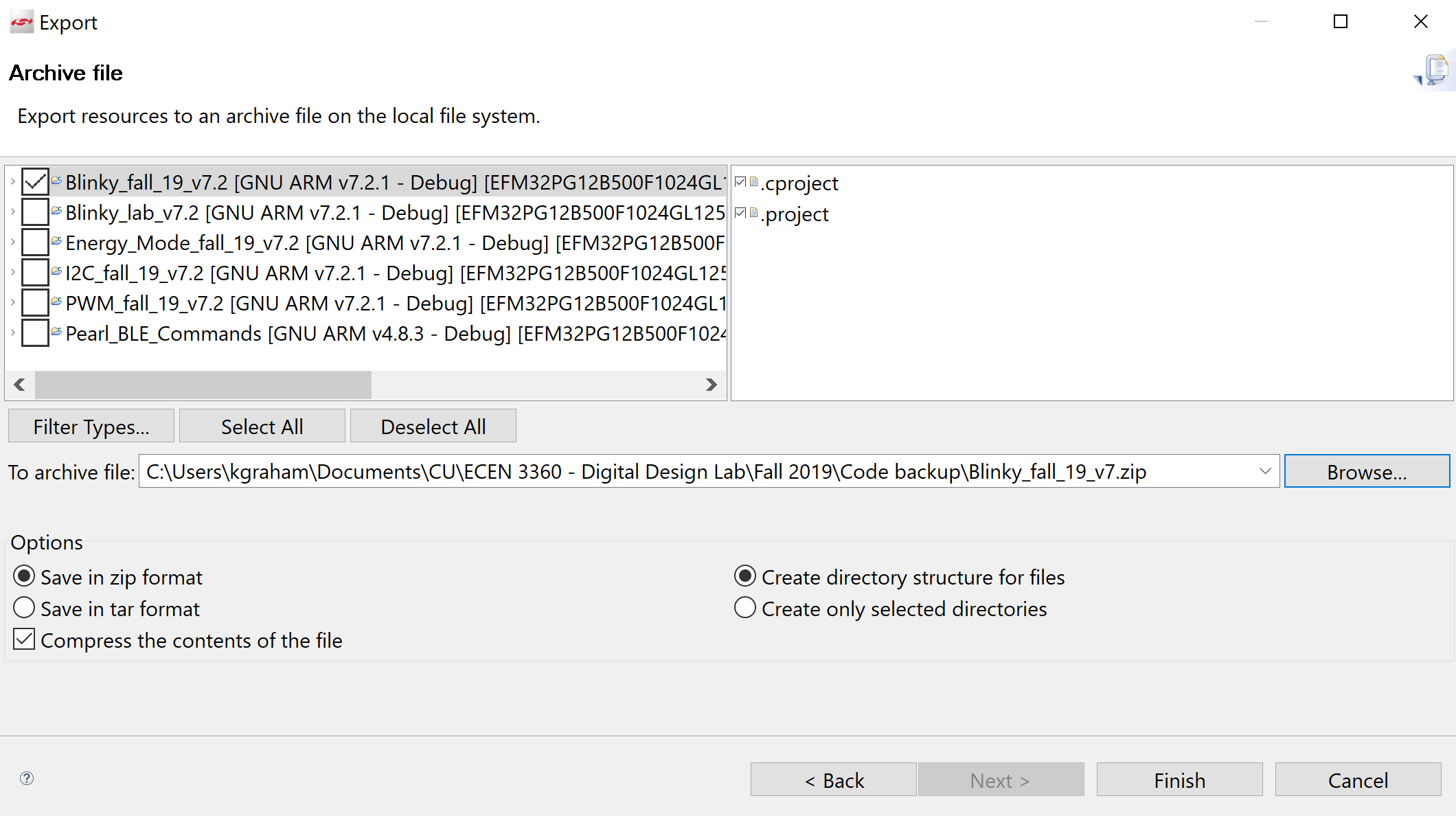
* 1. In the Export popup window, insure your project is select and then click on More Export Options.



* 1. In the More Export Options, you may need to expand the window to see all the options. Select Archive File and then click on Next.



* 1. In the next Export popup window, make sure that only your project is selected and that you have selected a folder and name for the exported .zip project. Once ready, click Finish.



* 1. Now locate your .zip exported project and upload it into the Lab 1 assignment located in this course Canvas course website.

Reference link: Hardware Abstraction Level

* <http://devtools.silabs.com/dl/documentation/doxygen/5.7/>
* Select the EFM32 Pearl Gecko 12

Questions:

Complete the Lab 1 worksheet for this Lab. The Lab 1 worksheet can be found in the Canvas Course Quiz section.

Point breakdown:

Lab 1

* Program execution (exported project) 10 pts

Late Penalty deduction:

* Exported program
  + Due day + 1 day max score is 8 pts
  + 1 day late to 3 days late max score is 6 pts
  + After 3 days late max score is 0 pts