

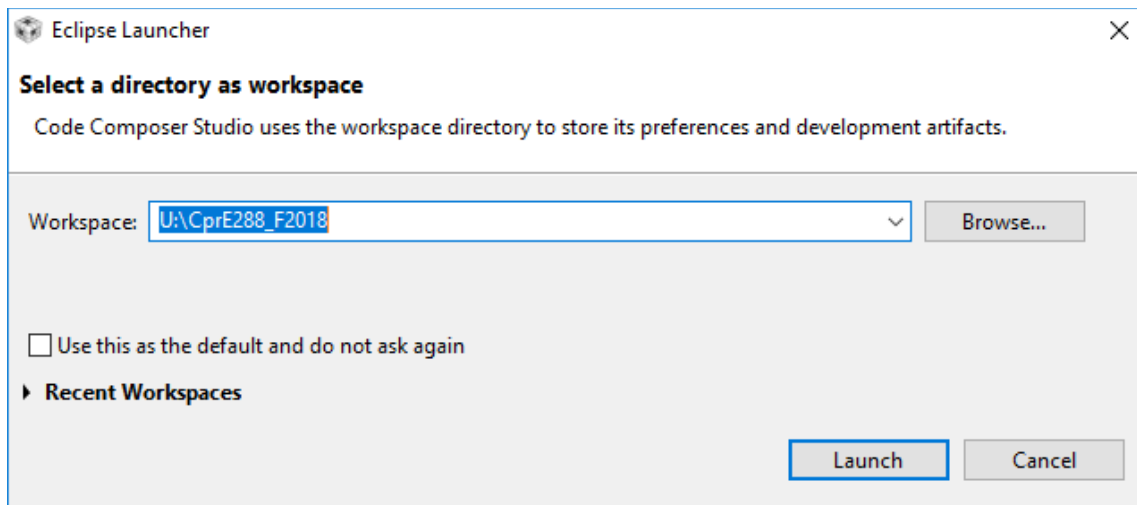
Part I. "Hello, world"

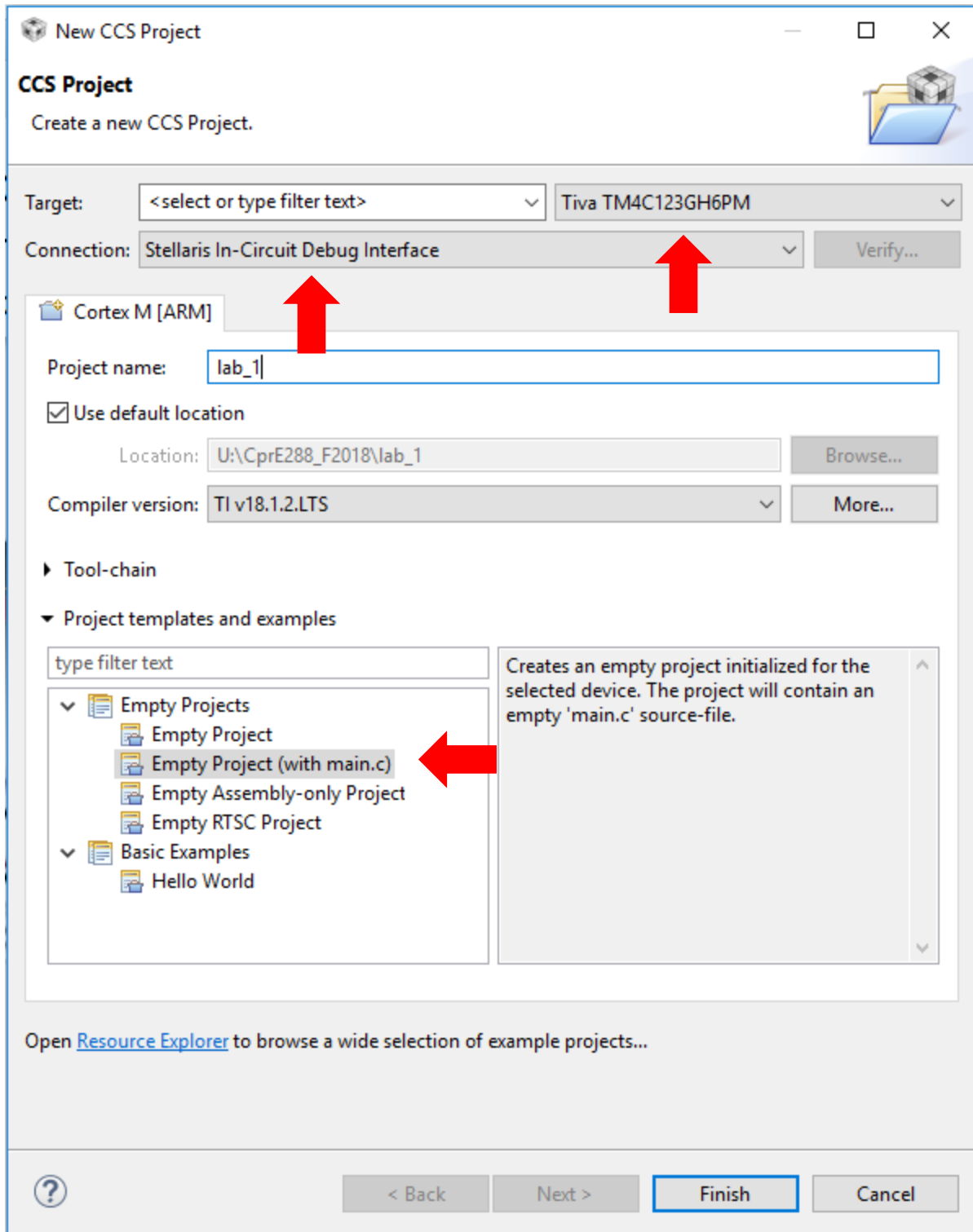
In this part, you will build a program executable from files provided to you.

Create a project and build

To start a new project, you'll need to follow these steps: In Code Composer Studio, select File->New->CCS Project, Select the following settings for your project.

Note: Make sure to use your **U drive (U:)**. Do not use the C drive (C:\) as this will get erased!!





You may use "helloworld" as the project name and then the default initial file is "helloworld.c". Other project names can be used, but then you need to keep in mind that the initial file name will not be "helloworld.c".

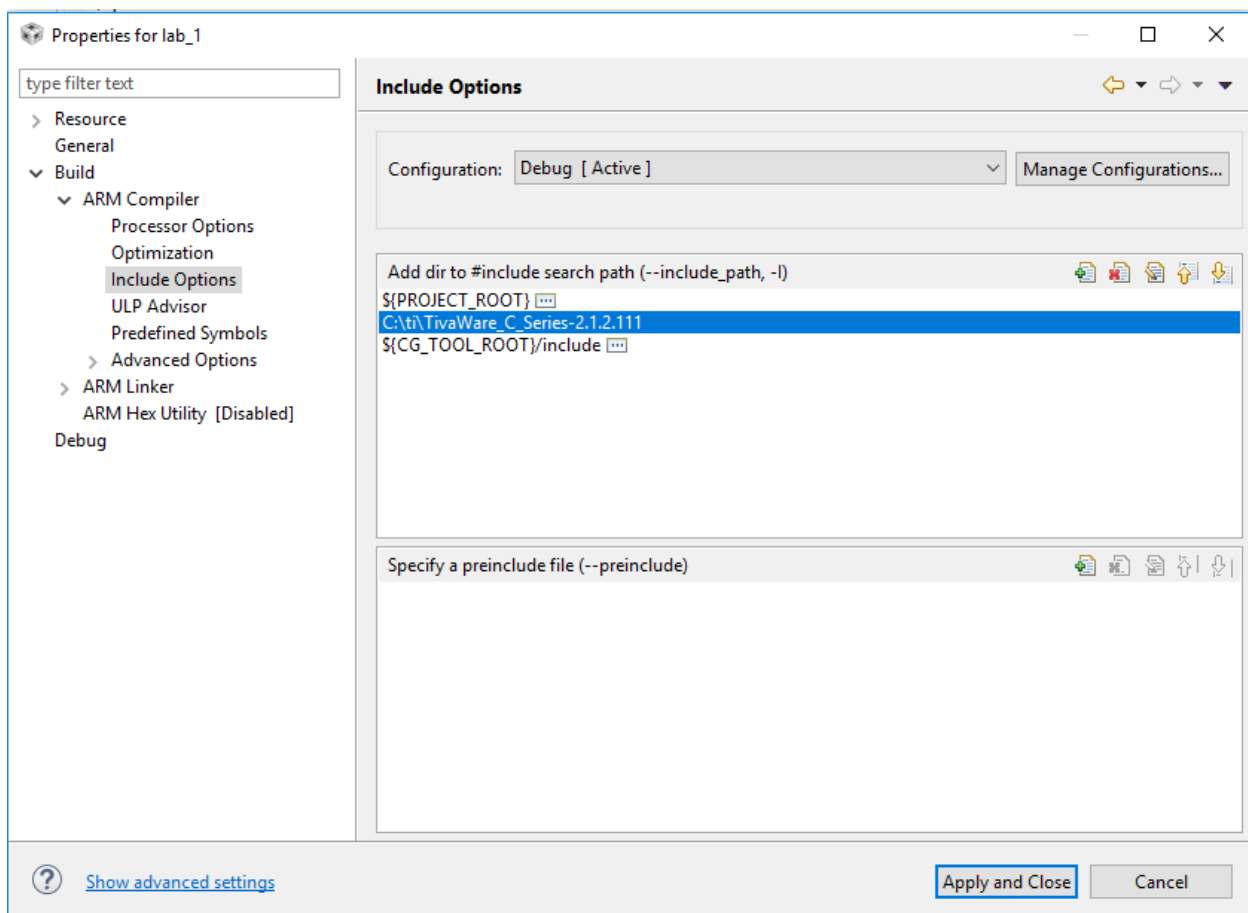
Click finish to create your project

Right click on your project in the Project Explorer, and select 'Properties'

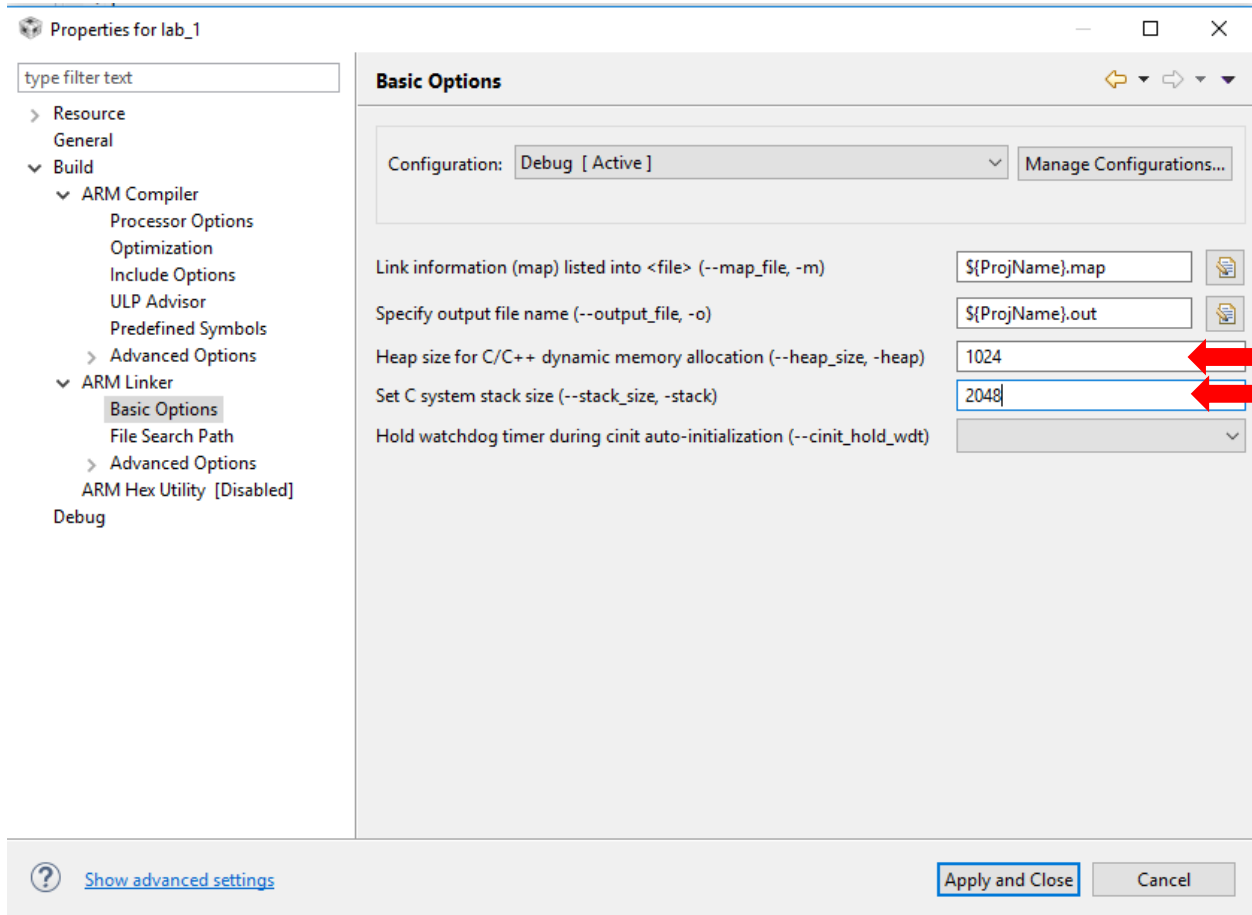
Caveat: Pay attention to details when entering information. Actual windows may differ slightly from the pictures below. Enter information in the correct boxes based on the name of the box.

Go to Build->ARM Compiler->Include Options, and add "C:\ti\TivaWare_C_Series-**2.1.2.111**"

Note: Do NOT copy/paste, as the version number **2.1.2.111** has most likely been updated. Instead, navigate through the directory structure to find the correct location. If this is not done correctly you will get **compile errors** later.

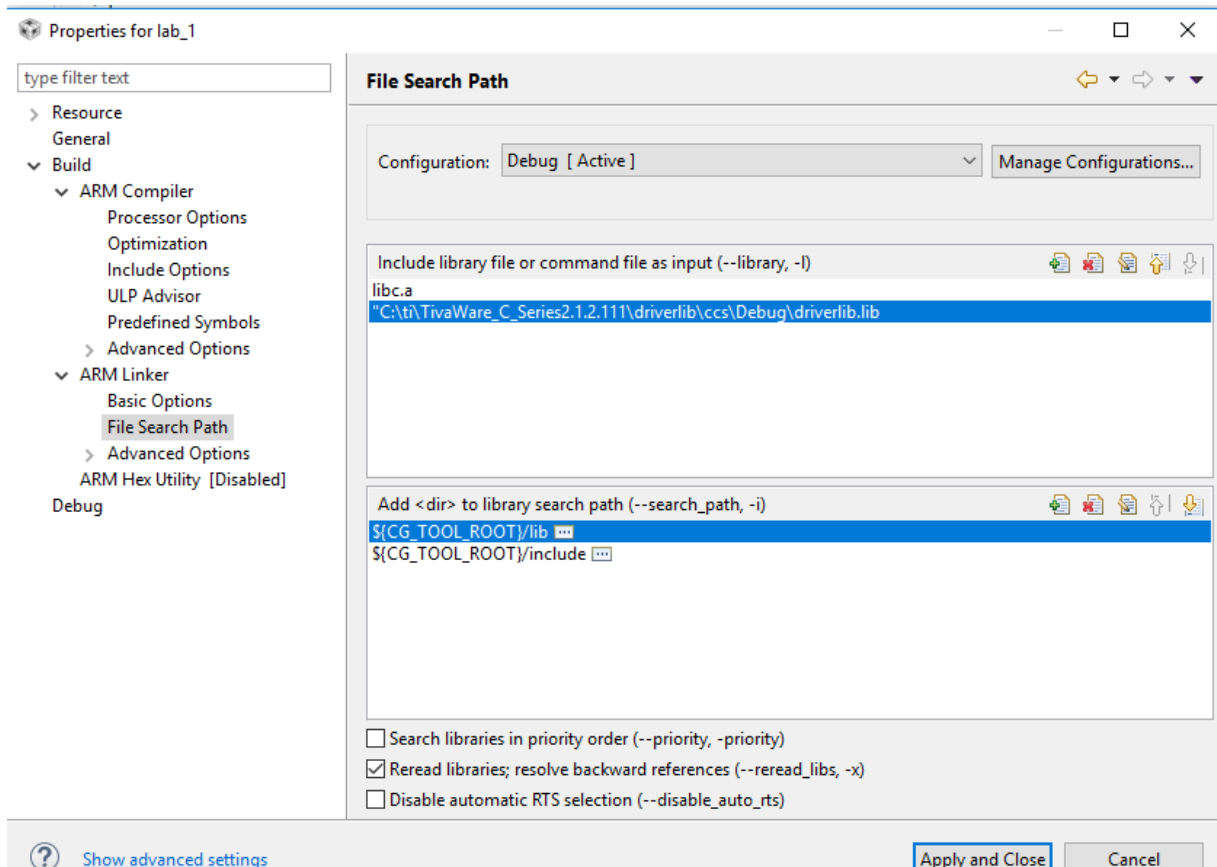


Go to Build->ARM Linker->Basic Options and modify the stack and heap sizes as shown:



Navigate to **File Search Path** and add "C:\ti\TivaWare_C_Series-**2.1.2.111**\driverlib\ccs\Debug\driverlib.lib" and any other differences from the top box.

Note: Do NOT copy/paste, as the version number **2.1.2.111** has most likely been updated. Instead, navigate through the directory structure to find the correct location. If this is not done correctly you will get **compile errors** later.



Next, **delete the default template file named main.c**. Delete it by right-clicking on "main.c" in the Project Explorer window, then select delete.

Now add files into the project. Simply add the lab files to the project folder and they will appear in your Project Explorer.

Build the binary image by selecting Build->Build Project or you can build all with the shortcut ctrl+B. You should see this message at the end: "Build succeeded."

Load the Binary Image on to the Tiva™ TM4C123GH6PM's processor

- Connect the robot with the USB cable
- To load your program, select the "Run" file menu and click "Debug". After the project builds hit the "resume" button (F8). The LCD should now say "Hello World."

Check list: Show your TA that you successfully create a project, and run it on the robot.

Part 2. System Sketch

As you are training on the Cybot platform, it is important that you gain an understanding of the hardware for which you will be deploying your Embedded System application software. In this lab you will be using the Cybot LCD screen to display information to a user.

For this part of the training lab you will use the following documents to sketch a diagram that shows how the LCD screen of the Cybot platform connects to the Cybot Microcontroller.

Documents:

- a) TM4C123GH6PM Datasheet
- b) Cybot Baseboard
- c) Cybot LCD Board
- d) LCD Controller Datasheet

1) Your diagram should show the package pin number and datasheet names of the LCD screen, and how they connect to the Microcontroller pins (giving the pin number and detailed PORT names used on the Microcontroller).

2) Is the LCD screen we are using connected using a 4-bit, or 8-bit Data Bus (DB) wiring interface?

Check list: Show your TA your System Sketch, and explain it to them using the documentation that you referenced.

Part 3. Debugger

Debug mode allows us to execute code and observe changes line by line. It also has useful features like the Variables, Expressions, and Registers tabs that show you the current state of different aspects of our code and the microcontroller. It spares us countless hours and print statements when trying to debug code.

In helloworld.c replace "**lcd_printf**" with "**lcd_puts**". You will be stepping through and explaining to your TA how "lcd_puts" works. **Optional:** Can you see the connection between the software and the Timing Diagram given in the "LCD Controller Datasheet"?

Click the "Run" file menu and then click "Debug". Press F6 to step over each statement until you reach the "lcd_puts" function. Press F5 to step into this function.

Check list: Explain how to debug to the TA, and how lcd_puts works.

Part 4. Rotating Banner

In this section, you will write a program to show a rotating banner on the LCD screen. Every 300 milliseconds, the text "Microcontrollers are lots of fun!" should move one position to the left. **This banner should repeat indefinitely**. Edit your helloworld project, as it provides a good starting point.

Here are the requirements of the output:

- Only the first line of the LCD screen should be used. The LCD screen has 4 lines. The second line should be blank all the time.
- At the beginning, the first letter 'M' appears on the right side of Line 1. The following characters appear one by one every 0.3 second.
- After the 20th letter appears, new characters appear one by one every 0.3 second and push the old letters off the LCD screen.
- After the last character '!' appears on the right, the message continues shifting to the left. Every 300 milliseconds the leftmost letter is pushed out until the screen is completely cleared.
- After the screen is cleared for 0.3 second, repeat the whole procedure again. The banner should repeat forever until the board is turned off.
- You need to write your program so that the message can be changed quickly. The TA will ask you to display a new message during the demo. You should only need to make a minor change to your program and compile.

There are three functions that have been written for you that will help:

- `timer_waitMillis(uint32_t millis)`: Delay the program execution for 'millis' milliseconds.
- `lcd_init()`: Initializes the IO ports to communicate with the LCD controller; clears the screen.
- `lcd_printf(const char *fmt, ...)`: Clears the screen and displays text. See documentation on [printf](http://www.cplusplus.com/reference/cstdio/printf/) (<http://www.cplusplus.com/reference/cstdio/printf/>)

Check list: Demo your program to your TA.