# *Getting started with Kinetis Design Studio*

## *CC2511*

This guide explains how to load code onto your development board.

### *Setting up your code repository*

We will use GitHub for code management.

There is a link on LearnJCU to set up a repository for your use in CC2511. Follow this link and get your repository created.

The next step is to "clone" the repositoy, meaning make a local copy on your computer so that you can start to work with it.

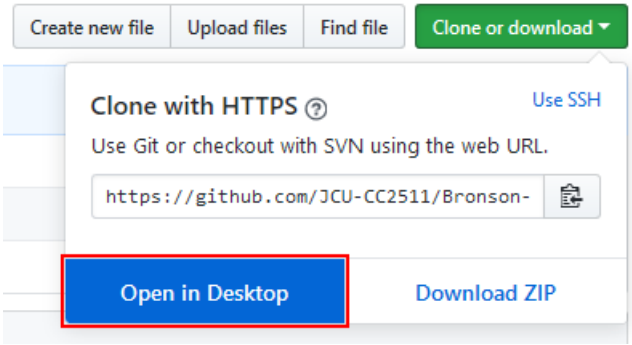
#### *Cloning the repo with GitHub Desktop (preferred)*

1. Install the native GitHub client: `https://desktop.github.com/`

   Hint: you can install this client even on a JCU computer, because it will place the files inside your personal home directory.

2. Open the repository in a web browser.

3. Using the green **Clone or Download** button, click the "Open in Desktop" button.

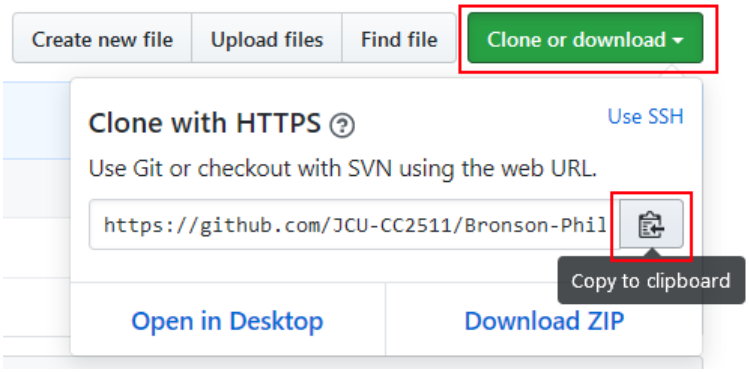



4. Browse to the directory where you wish to place the repository, e.g. where you store your CC2511 work.

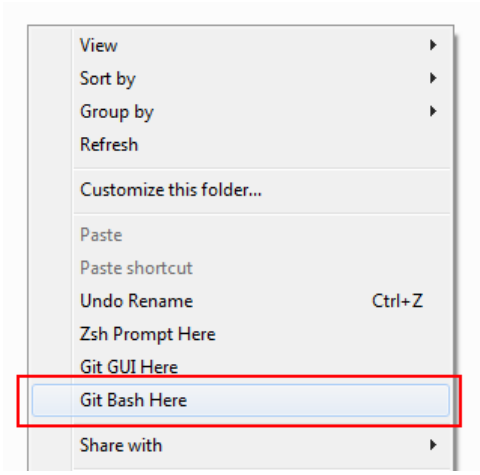
#### *Alternative method: Using the command line*

Instead of GitHub Desktop, you may use the command line Git client. Here is an example.

1. Using the green **Clone or Download** button, copy the **HTTPS** clone URL to the clipboard.




2. Navigate to a location where your files should be stored. On a JCU computer, this must be a portable USB drive. Right click and choose **Git Bash Here.**




*Important note*—The JCU computers will completely erase any files that you create after you log out. It is **strongly recommended that you place your repository on a portable USB drive** so that they do not get erased when you leave the computer.

3. Clone the repository by running **git clone**.

```
$ git clone https://github.com/JCU-CC2511/Bronson-Philippa-Labs.git
```

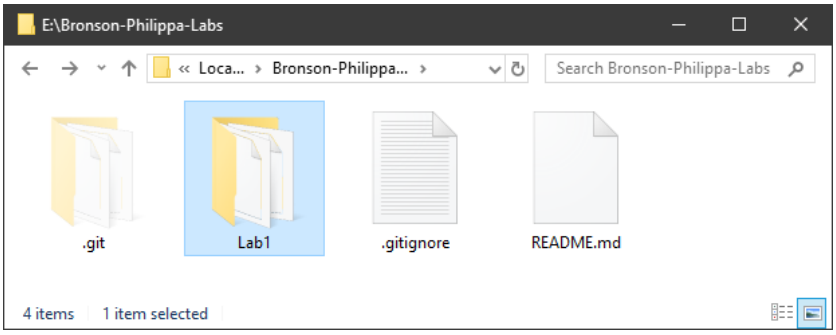
Hint: to paste, right click on the system icon in the top left of the window (older versions) or right click on the window itself (newer versions).

4. You should see a new folder containing your repository.

*Programming the board*

1. **Download the starter code from LearnJCU.** There will be a different file dependent upon the type of board that you have, e.g. K20 or K22 microcontroller.

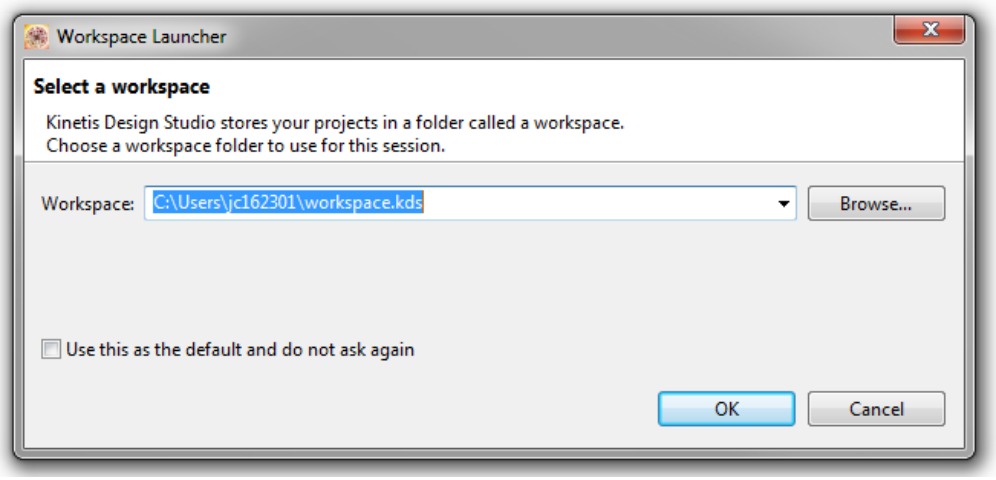
2. Unzip the starter code and place the "Lab1" folder in the Git repository that you previously cloned.



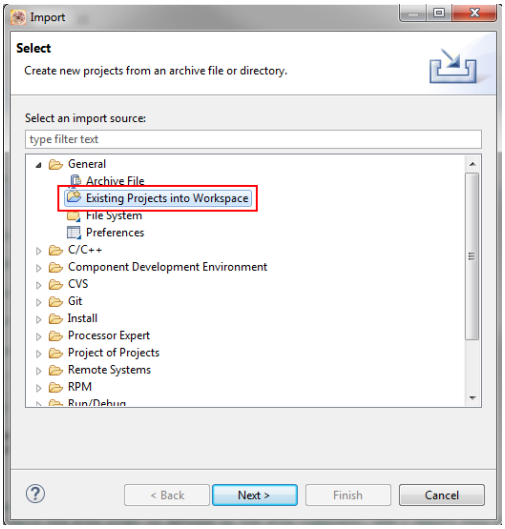

3. Start up the program "**Kinetis Design Studio**".

4. Kinetis Design Studio will prompt for the location of the "**workspace**". This will hold the settings such as which projects you have open. It can be any directory.

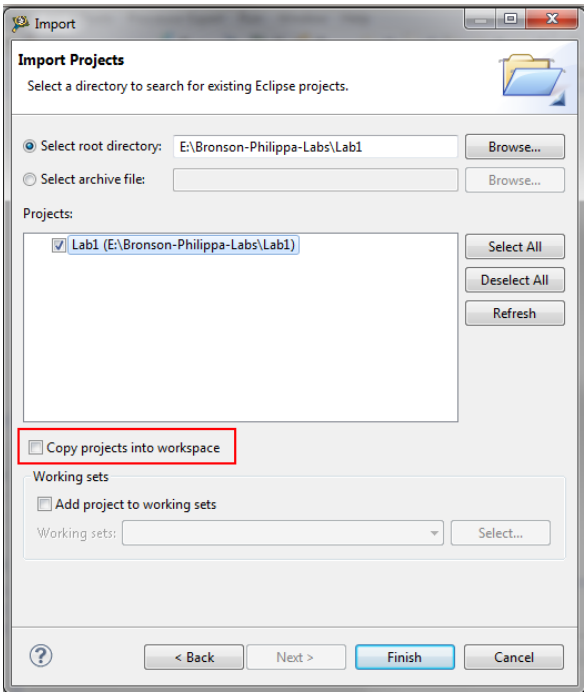
*Bring your own laptop*—Install Kinetis Design Studio from the URL provided on LearnJCU; otherwise download it from the NXP website.



5. Select **File -> Import** and choose "**Existing Projects into Workspace**".

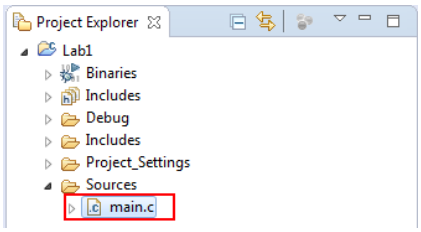



6. Choose the Lab1 directory. **Deselect "Copy projects into workspace"** because you want to leave the source code in the location that is managed by Git.

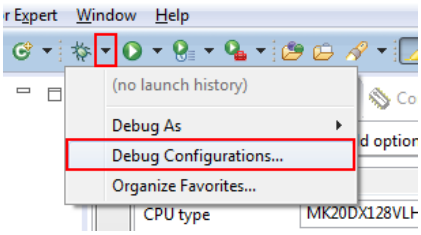



7. Click Finish.

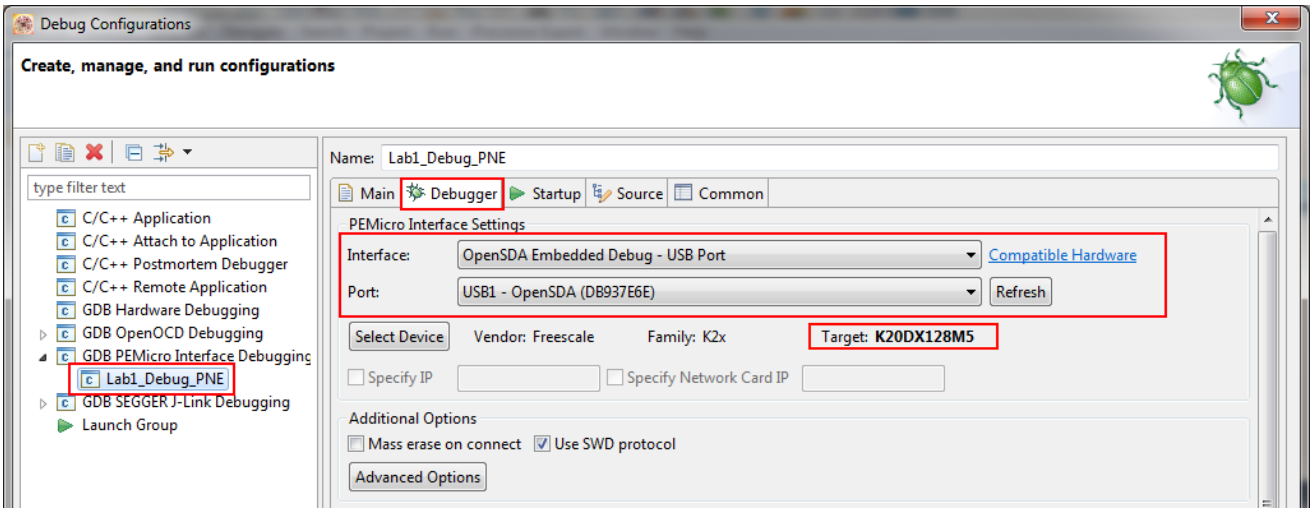
8. Under **Project Explorer,** open **Sources -> main.c** and browse the source code.



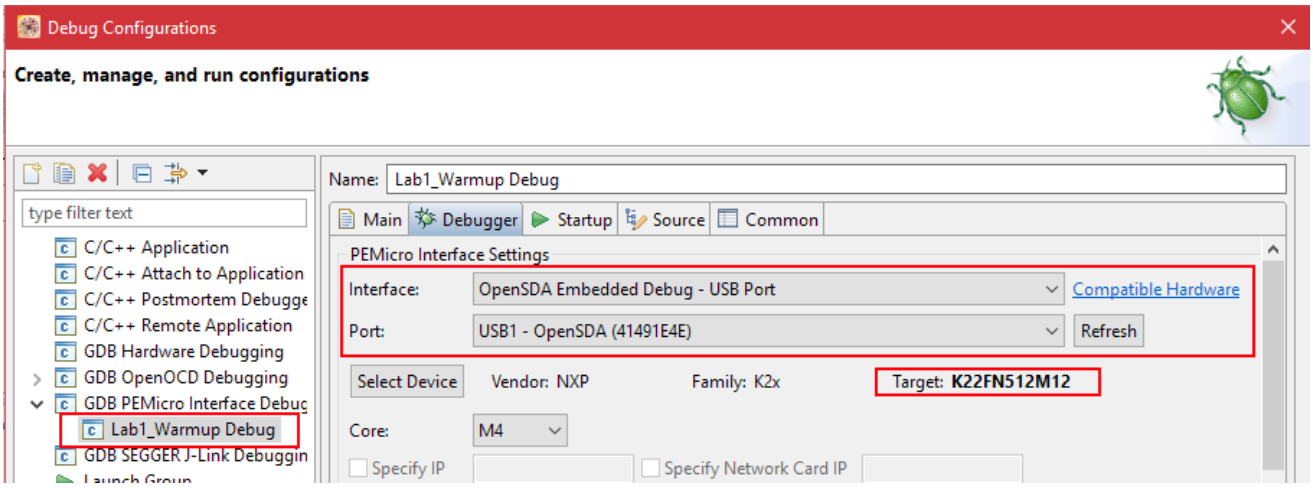

9. Set up the debug interface, which is used to program the board. Open the **Debug Configurations**, then set up **OpenSDA** underneath **PEMicro Interface Debugging**.




- Important note: the debug target depends upon that board that you have.
- If you have a red K20D50M dev board, your target processor is a **K20DX128M5** (below).



- If you have an orange K22F dev board, your target processor is a **K22FN512M12** (below).

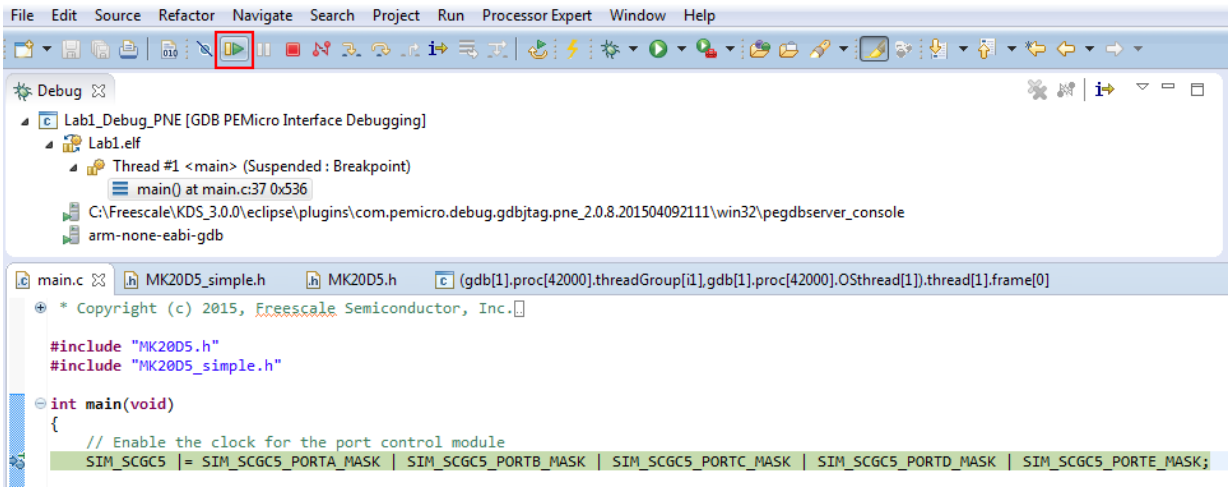



1. Choose the menu item **Run -> Debug** or press the **Debug** button on the toolbar.

   In the **Console** view, you should see the **compiler** running and then the **flash programmer** writing your program to the board.

   *Note*: There is a "Run" command and a correspondingly inviting toolbar icon, but it doesn't work for our development boards. The "Run" command is intended to execute code on the local PC, and not on the connected development board. In this subject, always use the "Debug" command.

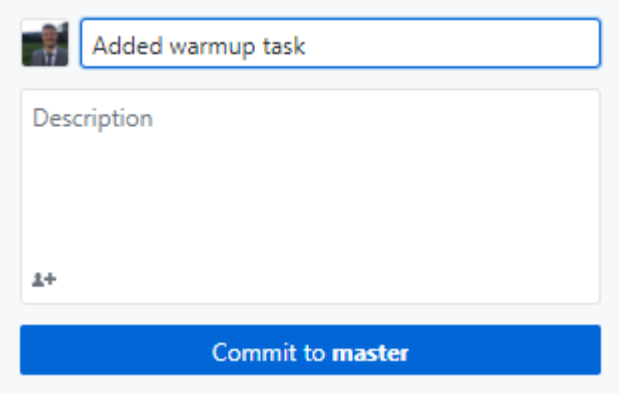
2. Press the "Resume" button to let the program run. You can also step line by line, inspect variables, and generally use the debugger.



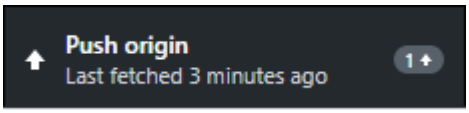

- You will be rewarded with a satisfying blinking light.

### *Committing your code to GitHub*

If you are using GitHub Desktop, open the "Changes" tab. You will see a list of all the files that you have created. In the bottom left of the window, type a description of the work that you have done and press the "Commit to master" button.




Then use the "Push origin" button at the top to upload your work to Github.com.



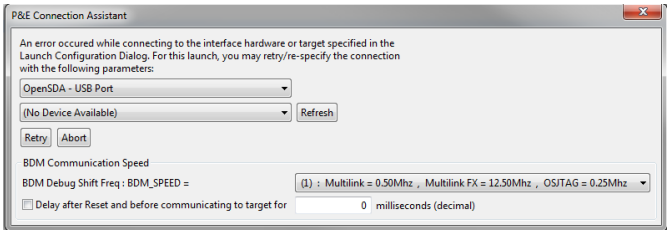

### *Next steps*

- Can you change the speed at which the LED flashes?
- Can you change the duty cycle, i.e. the ratio of the time on to the time off?

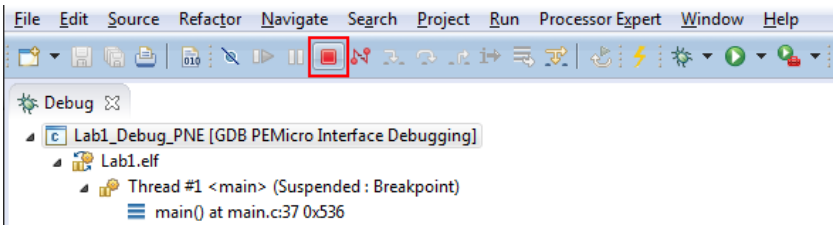
## *Troubleshooting*

### *Debugging fails to connect to the board*

- The debugger in Kinetis Design Studio is a separate program called “gdb”. Only one instance of gdb can communicate with the board at a time. If you try to start a second instance, it will be unable to connect:




- The solution is to always terminate the debug session when you are done by clicking the red **Terminate** button on the toolbar.




- If you have multiple debug sessions running, you can see them in the Debug view.