Zephyr Editor and Building with Eclipse Lab

Rev2.1, August 19, 2022

Objectives

In this lab, you will learn

- How to import a Zephyr application into an Eclipse IDE
- How to use Eclipse to edit the Zephyr source files and build the application

Introduction

NXP's MCUXpresso IDE is a free IDE supporting NXP's MCUs, and is based on the Eclipse framework. The previous lab guide Zephyr_MCUXpresso_IDE_Debug_Lab.pdf provided the steps to use MCUXpresso IDE to debug a Zephyr application. Those steps detail how to use the IDE to create a new Debug project, and to leverage all the debugger capabilities offered in MCUXpresso IDE, including the Zephyr Thread Awareness Debug (TAD) feature. Then the Debug Configuration for the IDE was modified to point to the application's zephyr.elf file, to debug the Zephyr application. This Debug project is very useful for debugging the Zephyr application. But the Debug project is not very useful to browse the Zephyr source files, edit source code, or build the Zephyr application.

This lab guide provides additional steps to import a 2nd project into the Eclipse IDE. This project is generated by West, linked to the Zephyr application, and can be used to browse and edit the source files. The application can also be built using Eclipse, instead of command-line. The result when using MCUXpresso IDE is having two projects in the Eclipse workspace for the Zephyr application: one linked to the Zephyr app for browsing and editing source, and another Debug project for debugging that same Zephyr app.

Pre-Requisites

This lab is written for Windows10, but Zephyr and this lab can easily be done in Linux or MacOS. This lab guide was written for Zephyr release v3.0.0.

- This lab assumes a Debug project for debugging was already created in MCUXpresso IDE following Zephyr_MCUXpresso_IDE_Debug_Lab.pdf. This includes first following Zephyr_Installation_Lab.pdf
- Download the latest MCUXpresso IDE, v11.6.0 or later required for some Zephyr OS features
- Terminal Program: MCUXpresso IDE integrates a terminal, or another program like PuTTY or Tera Term can be used
- Follow Zephyr_hello_world_Sample_Lab.pdf to prepare the hardware, verify environment variables, connect the terminal to the board

Hardware Requirements

- MIMXRT1060-EVKB
- Micro-USB cable (may need two cables, see Zephyr_hello_world_Sample_Lab.pdf)

Running the lab

Prepare Command Line

Before using the West commands, a command window should be opened, python virtual environment activated, and environment variables set. These steps only need to be done once after a command window is opened. If already done in a previous lab, skip to the next section. See Zephyr_hello_world_Sample_Lab.pdf for more details.

1. Open a normal command prompt in Windows
2. Navigate to the zephyr directory:
   cd %userprofile%\zephyrproject\zephyr

3. Activate the python virtual environment. After activating, the prompt should now include (.venv), see screenshot below:
   ..\.venv\Scripts\activate.bat

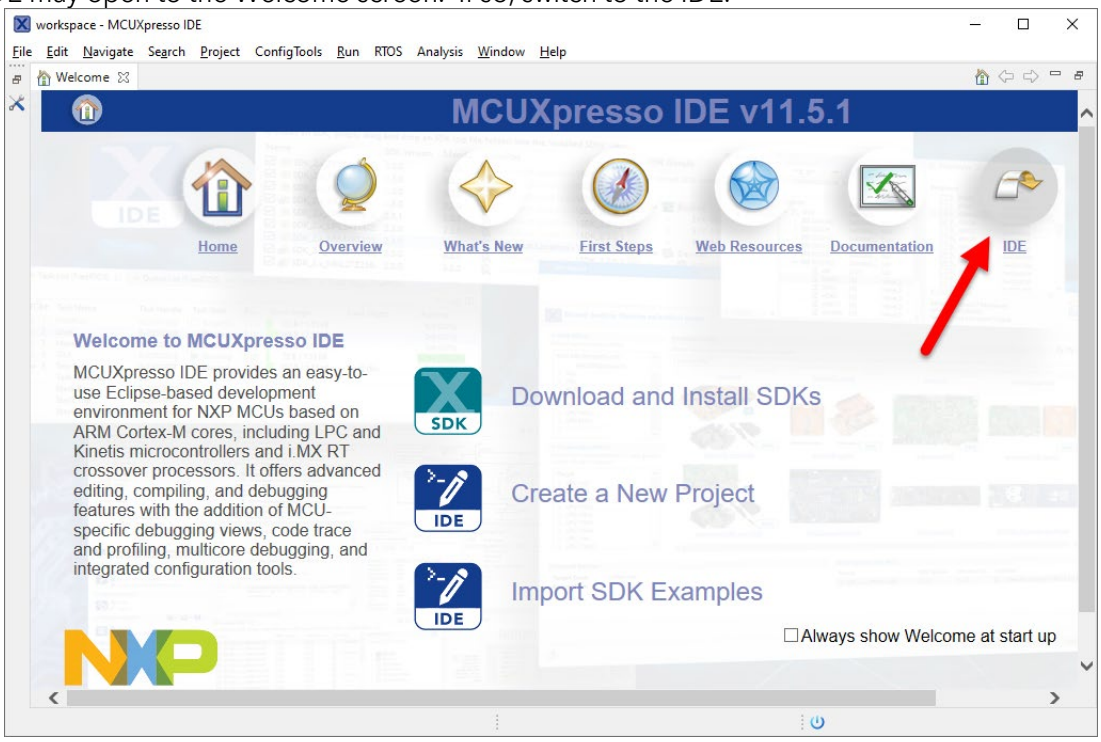
4. Set the environmental variables. Note, when using a command file like this to set the variables, the command file needs to be executed once anytime a new command window is opened:
   zephyr-env.cmd

5. If working on an NXP lab computer, you can restore the Zephyr source files that may have been modified in earlier labs using this Git command:
   git reset --hard

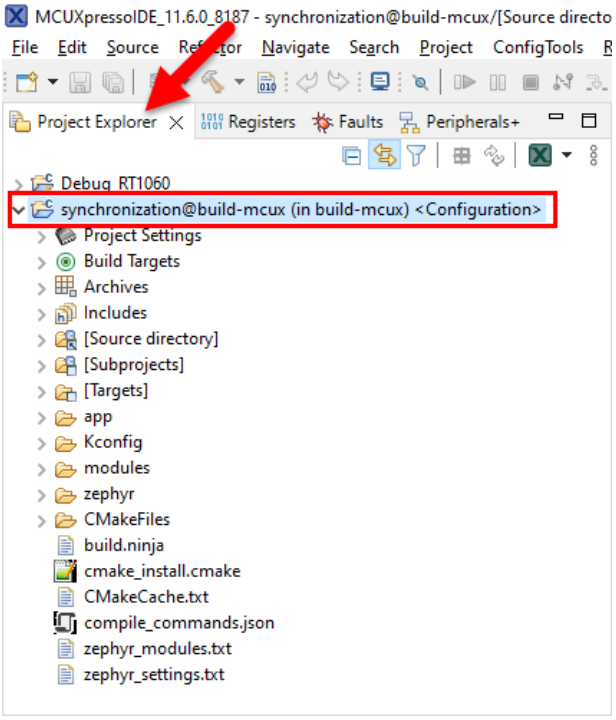
Build application for Eclipse

6. **Open MCUXpresso IDE**. If using an NXP lab computer with multiple versions of MCUXpresso IDE installed, be sure to open the version for this lab **v11.6.0**

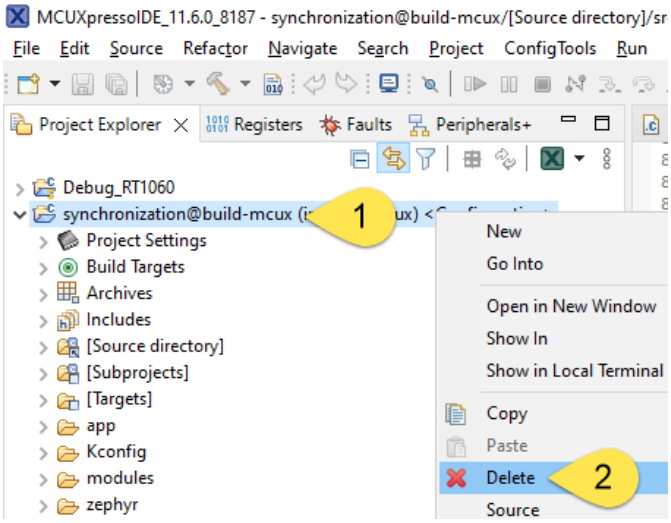
7. The IDE may open to the Welcome screen. If so, switch to the IDE.



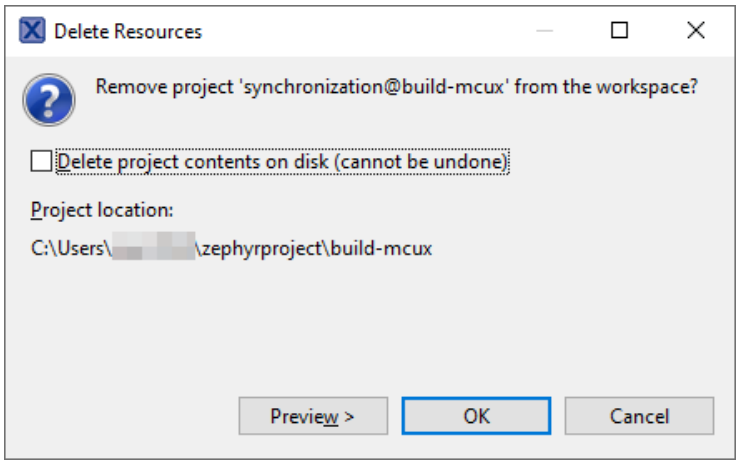
8. For an NXP lab computer that has been used before, the Eclipse project may have already been created and imported in the IDE. Click the **Project Explorer tab** to view the projects in the workspace. If the Synchronization project is listed like below, it should be deleted before the remaining lab steps.



9. To delete an existing project, right-click on the project name, and **select Delete**. Or use the Delete key on the keyboard.



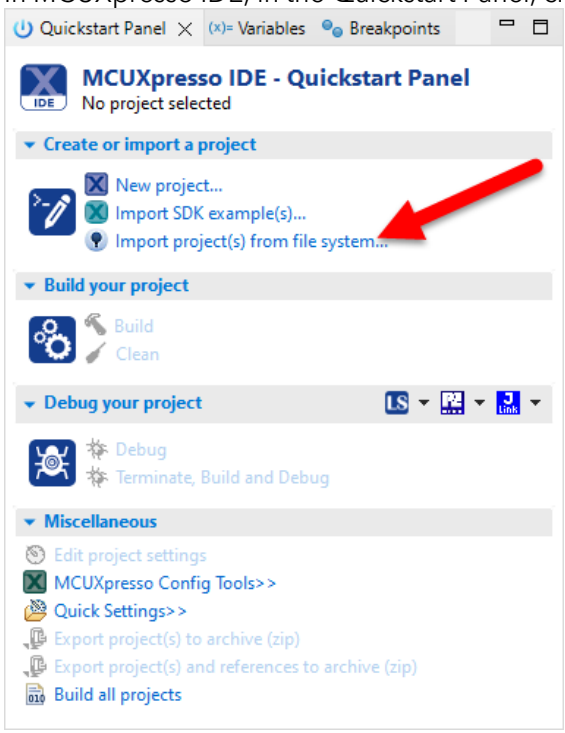
10. You can choose to also delete the project contents on the disk. For this project, the contents are the build-mcux directory that is generated by the `west build` command, including the Zephyr application image. For this lab, this choice is not important as the next step will rebuild this build-mcux folder. Click OK.



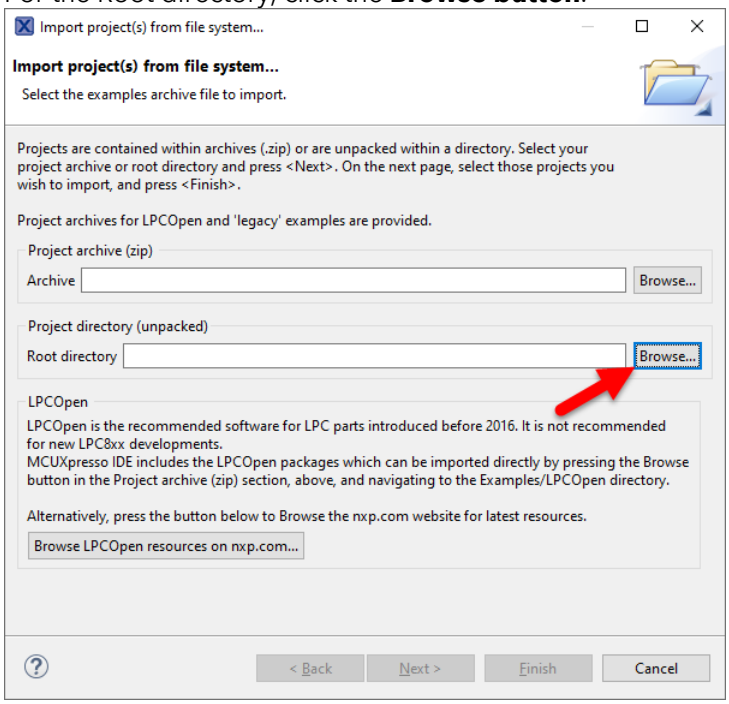
We will rebuild the same sample application used in Zephyr_MCUXpresso_IDE_Debug_Lab.pdf. But this time we will add an additional argument to generate an Eclipse project for the application. The argument to add is: `-G"Eclipse CDT4 - Ninja"`

11. In the command window, build the synchronization sample application using west. NOTE: when pasting this line into the command window, be sure to get both lines below as a single command. You may need to paste the 2nd line separately to the end of the 1st line for the full command:
west build -b mimxrt1060_evk -d ..\build-mcux samples\synchronization --pristine -- -G"Eclipse CDT4 - Ninja" -DCONFIG_DEBUG_THREAD_INFO=y -DCONFIG_INIT_STACKS=y

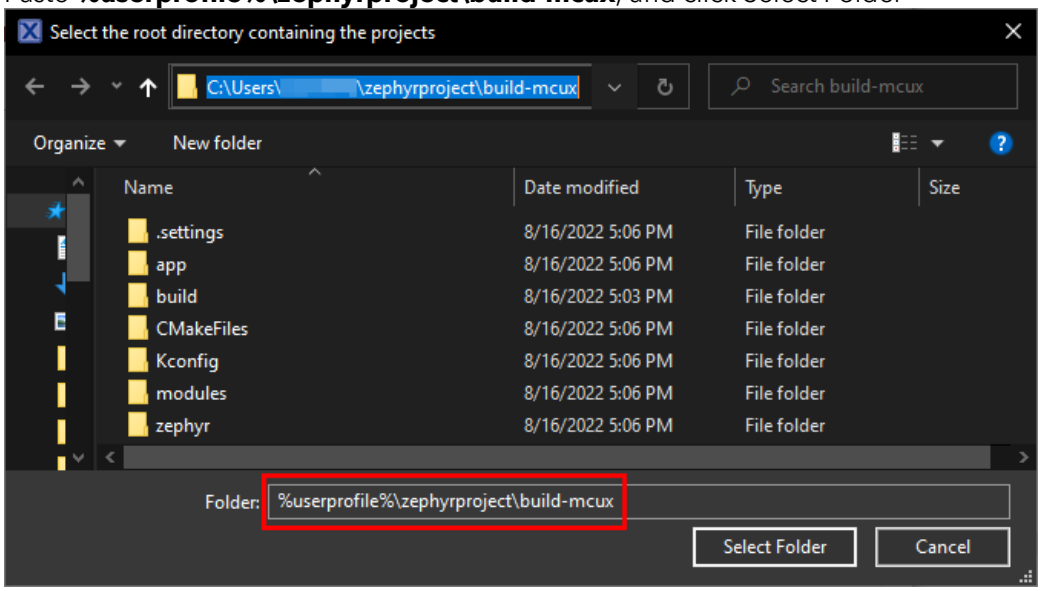
12. In MCUXpresso IDE, in the Quickstart Panel, click **Import project from file system**



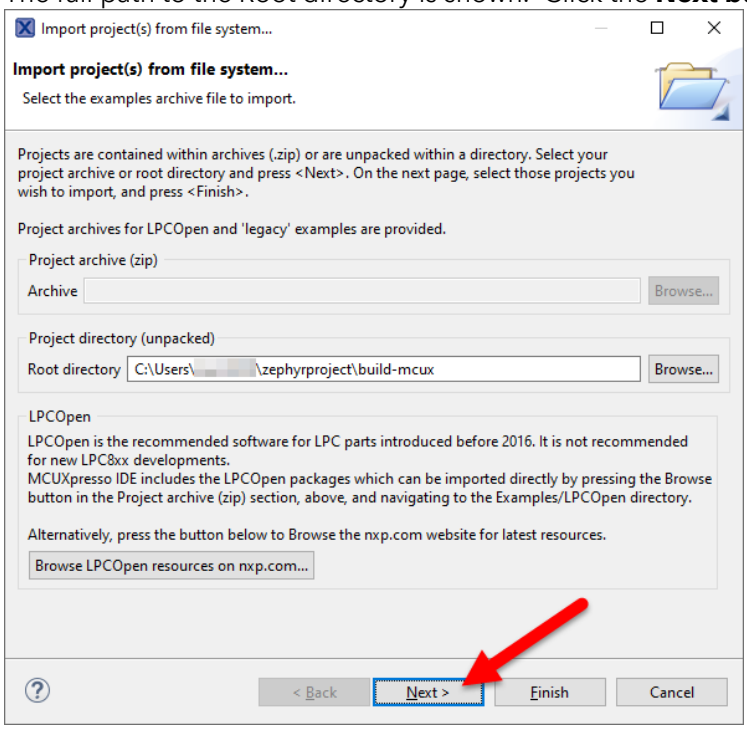
13. For the Root directory, click the **Browse button**.



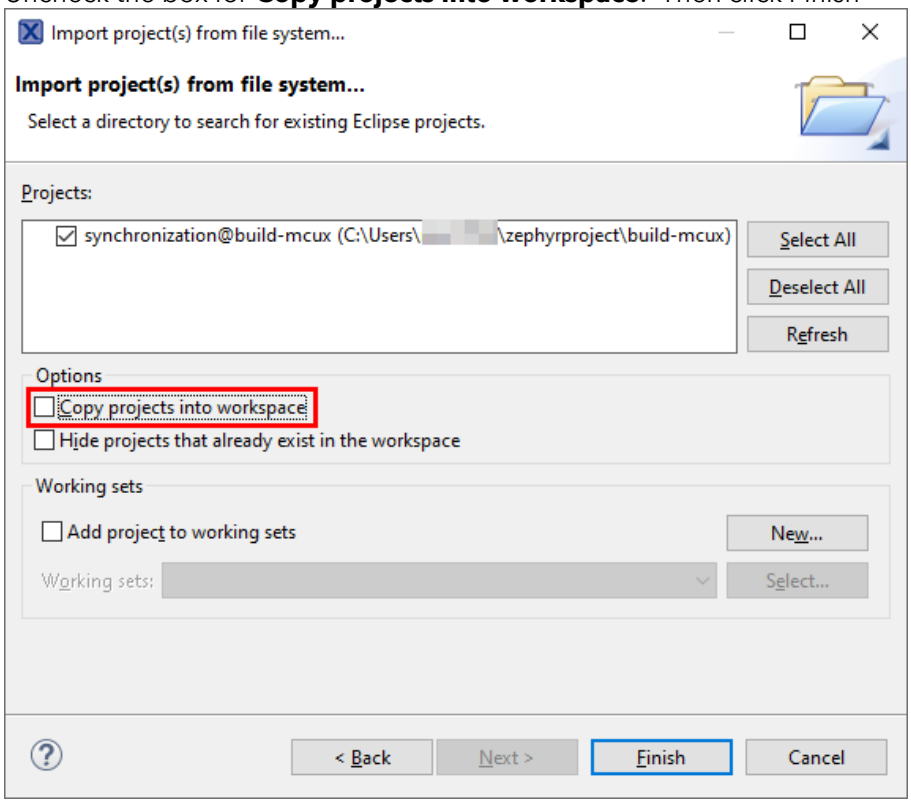
14. Paste **%userprofile%\zephyrproject\build-mcux**, and click Select Folder



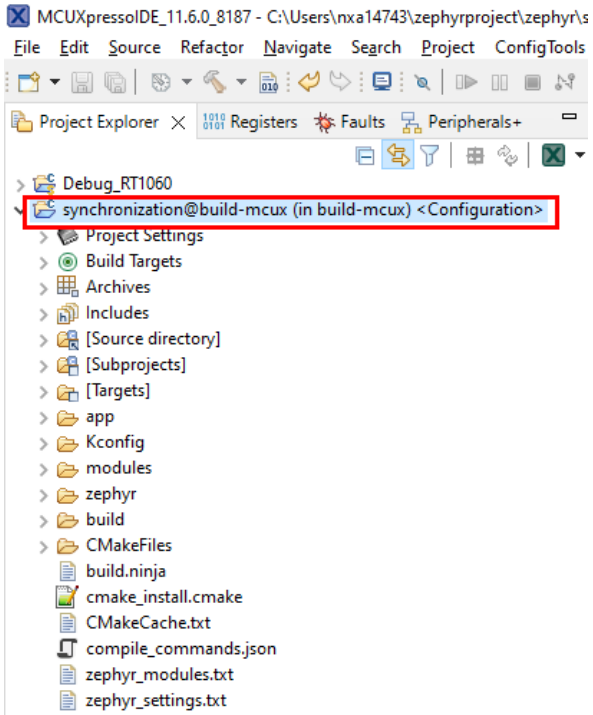
15. The full path to the Root directory is shown. Click the **Next button**.



16. Uncheck the box for **Copy projects into workspace**. Then click Finish



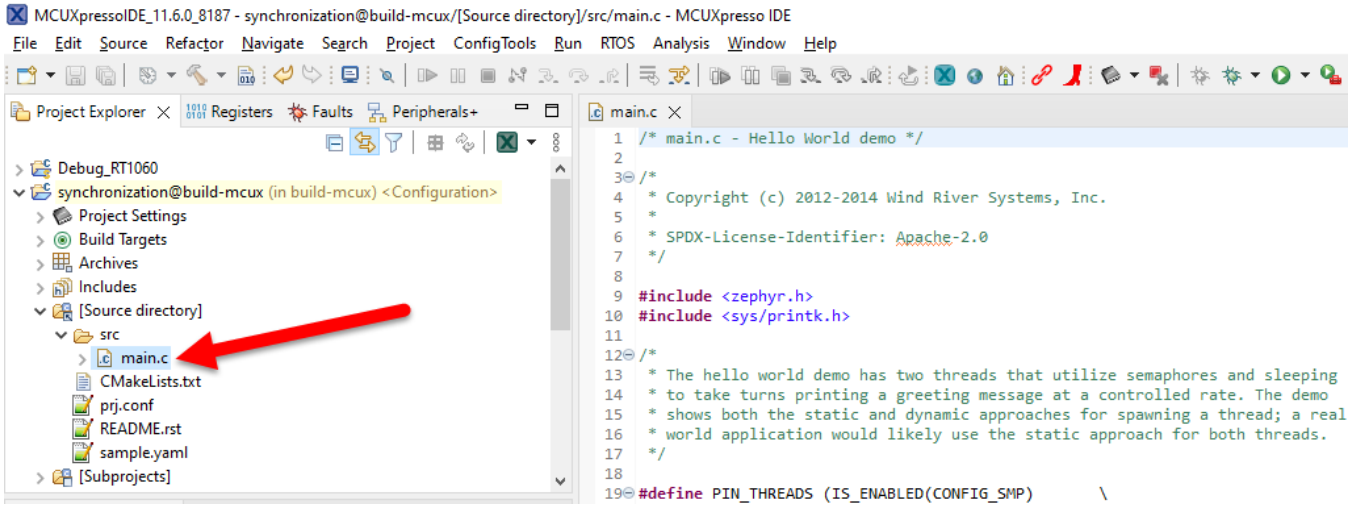
The imported project will be included in the Project Explorer View:



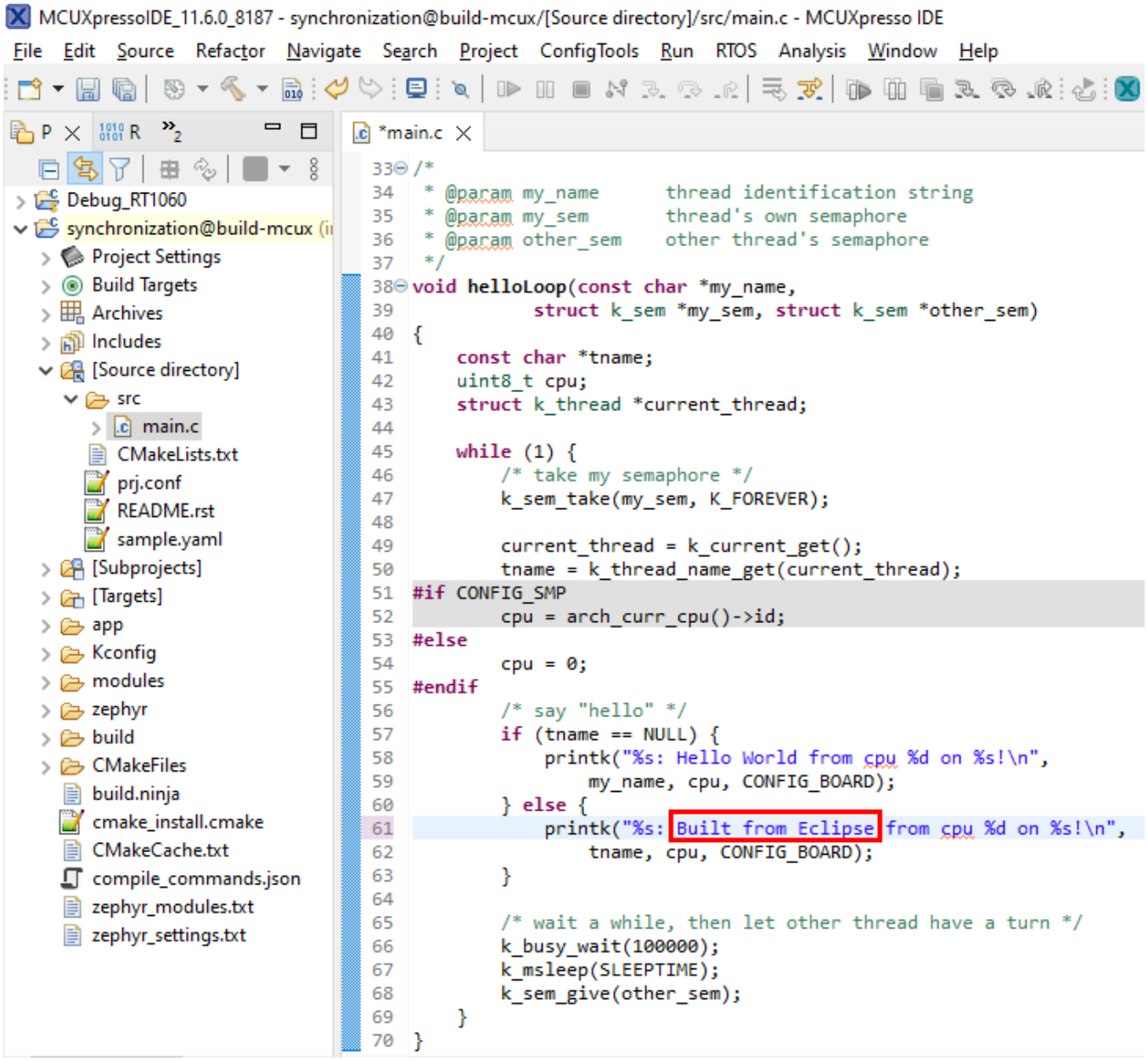
Use Eclipse to edit and rebuild the Zephyr application

This Eclipse project has links to all the source files used to build the Zephyr application. You can now use Eclipse to browse these files, search the code, edit, and rebuild the application. In this section, we will make a simple change to the source code, and rebuild the app.

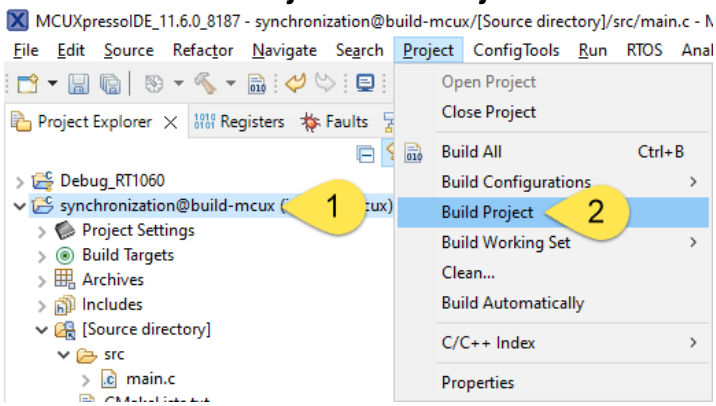
17. Open the main source file for the synchronization sample app. Expand the folders, and browse to the file **[Source directory]->src->main.c**. Then double-click main.c to open in the editor.



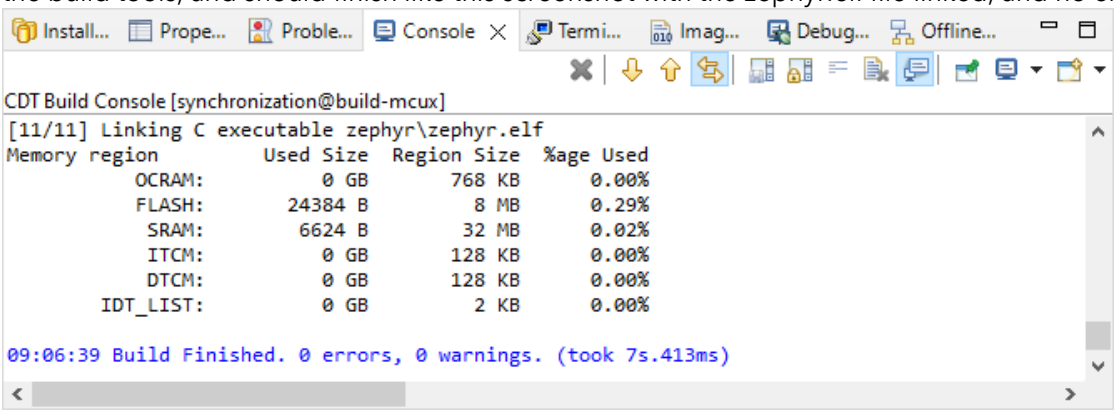
18. On line 61 in main.c, modify the print message to update the Zephyr sample application. In this screenshot, the string is update to “Built from Eclipse”



19. To rebuild the Zephyr application:
    a. First click the project to build in the Project Explorer view
    b. Then use the menu **Project->Build Project**



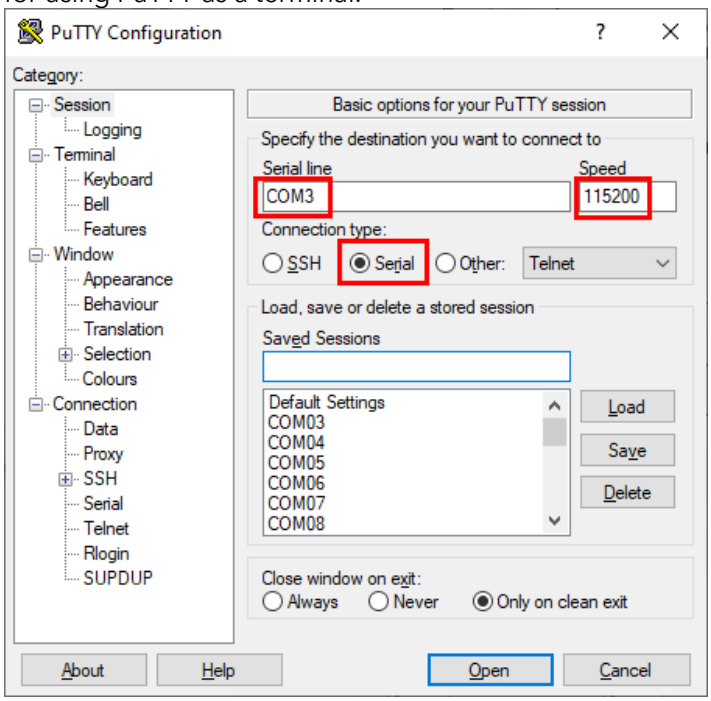
Eclipse will invoke the build tools to rebuild the application. The Console View will show the output of the build tools, and should finish like this screenshot with the zephyr.elf file linked, and no errors.



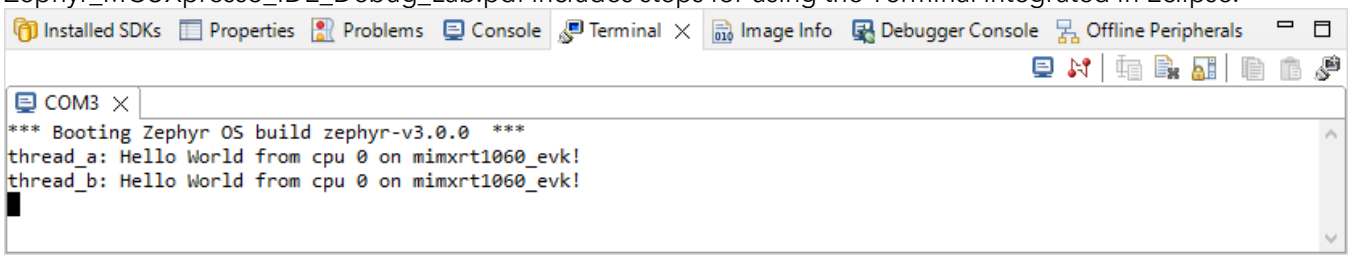
Connect a Terminal to the board

20. Ensure the board is powered, and the debug probe is connected. This lab uses the on-board debugger of MIMXRT1060-EVKB with USB connector J1 using Jlink firmware, plus powered through USB J48. For more details on preparing the hardware, see Zephyr_hello_world_Sample_Lab.pdf.

Previous lab guides provided the steps for terminal options. Zephyr_hello_world_Sample_Lab.pdf gives steps for using PuTTY as a terminal.



Zephyr_MCUXpresso_IDE_Debug_Lab.pdf includes steps for using the Terminal integrated in Eclipse.

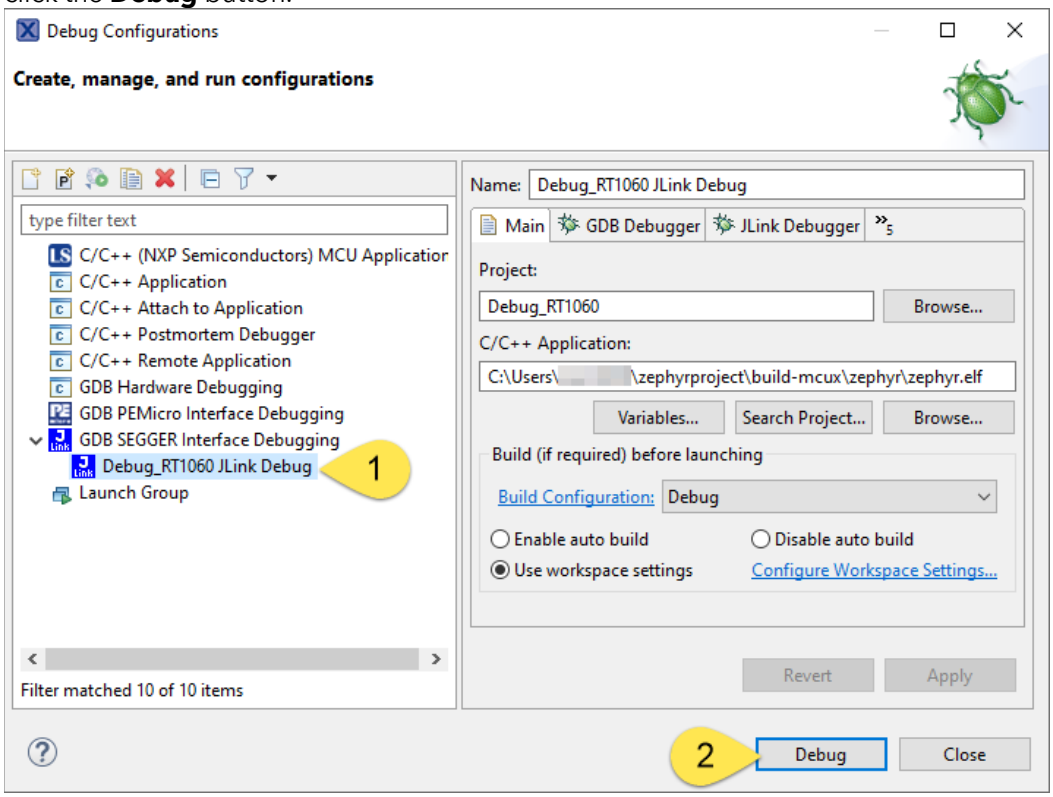


Program the flash and debug with MCUXpresso IDE

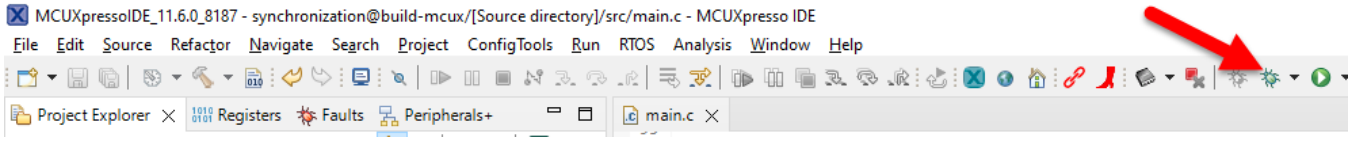
For this section, we will use the Debug Configuration in the Debug project, already created using the steps in Zephyr_MCUXpresso_IDE_Debug_Lab.pdf.

21. In MCUXpresso IDE, use the menu **Run->Debug Configurations**…

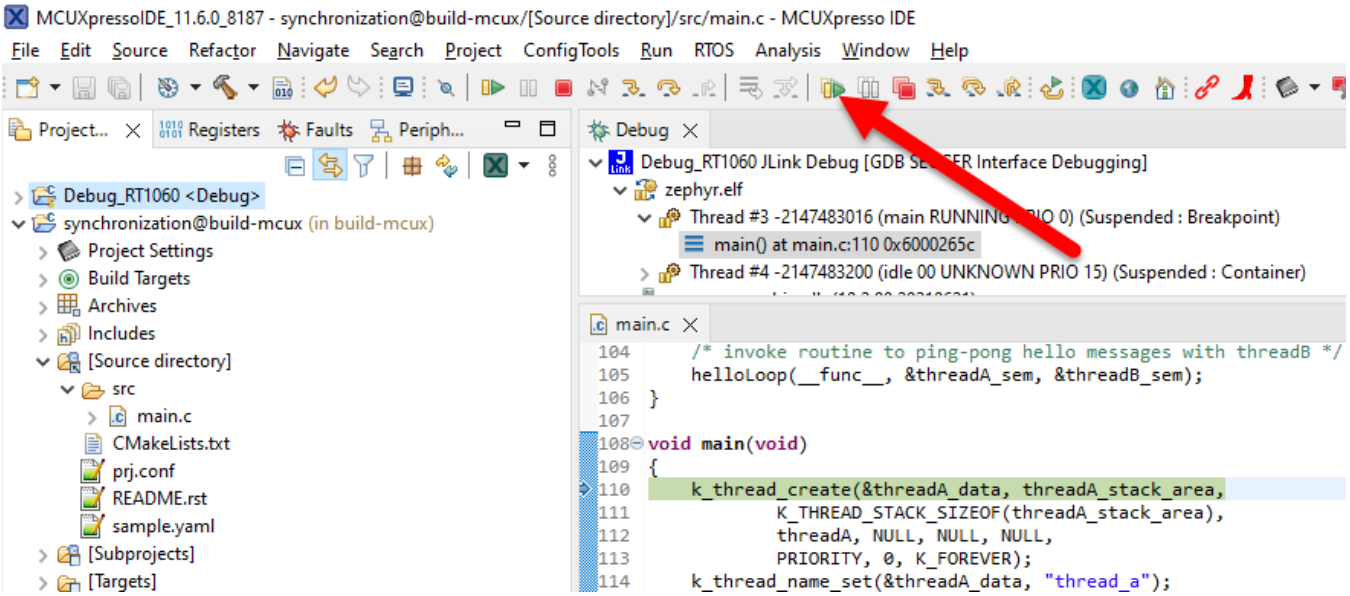
22. Select the Debug Configuration for the **Debug_RT1060** project created in the previous lab, and then click the **Debug** button.



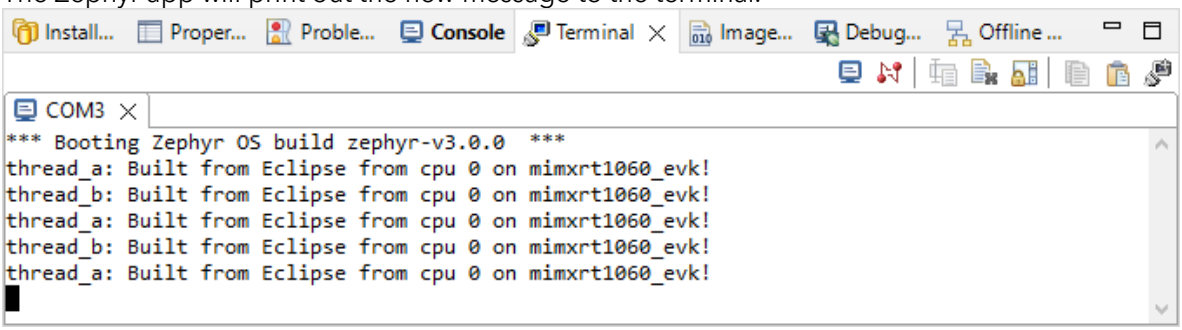
A shortcut for next time you debug: Eclipse will remember the last Debug Configuration used. Once you have launched this Debug_RT1060 configuration, next time you debug can skip the Debug Configurations menu, and just click the green bug icon on the toolbar.



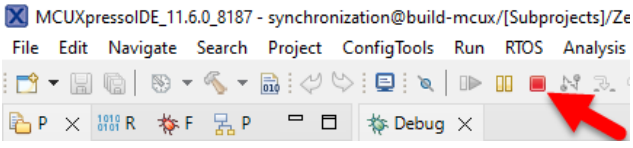
23. The IDE will connect the debugger, program the flash, start the Zephyr application, run to main() and halt. Click the **Resume button**.



The Zephyr app will print out the new message to the terminal.



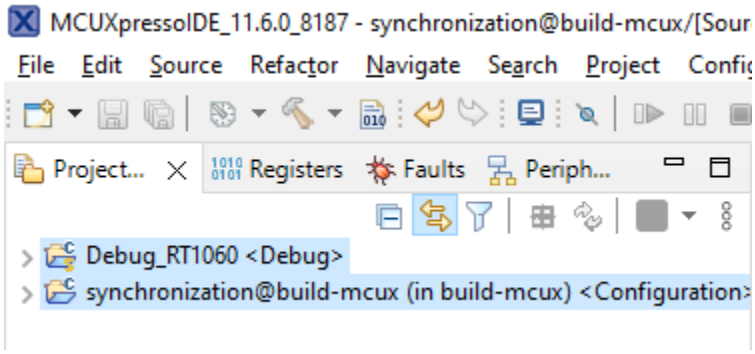
24. When done debugging, click the **Terminate** button.



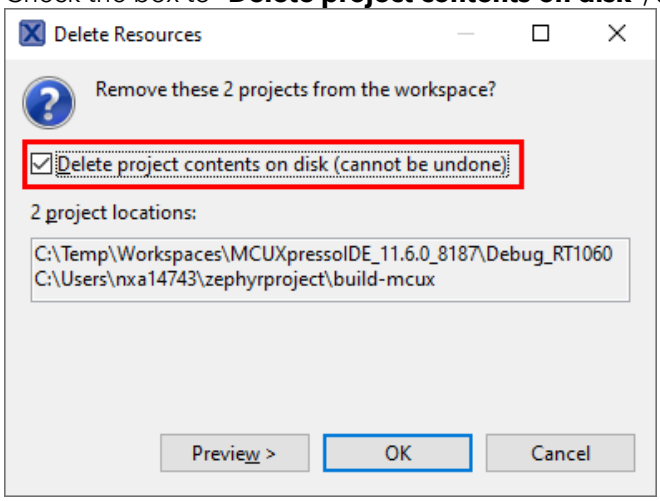
25. Disconnect the Terminal

Deleting the Eclipse projects

26. If using an NXP lab computer, be kind to the next user, and delete the projects you created. In the Project Explorer tab, **select both the Debug_RT1060 and synchronization projects**. Then press the **Delete key** on the keyboard.



27. Check the box to "**Delete project contents on disk**", and click OK.



This completes this lab.

Revision History

Rev	Date	Details
1.0	5/29/2021	Initial Version
2.0	8/18/2022	Updated to use this IDE as Editor project, and separate Debug project for debugging
2.1	8/19/2022	Fixed minor typos and formatting