

# Objectives

---

In this lab, you will learn:

- How to import an example from the Zephyr Repo in the VS Code workspace
- How to build, clean, debug, and run the example
- How to connect the Serial Monitor for UART console
- How to navigate and edit files in the Explorer view

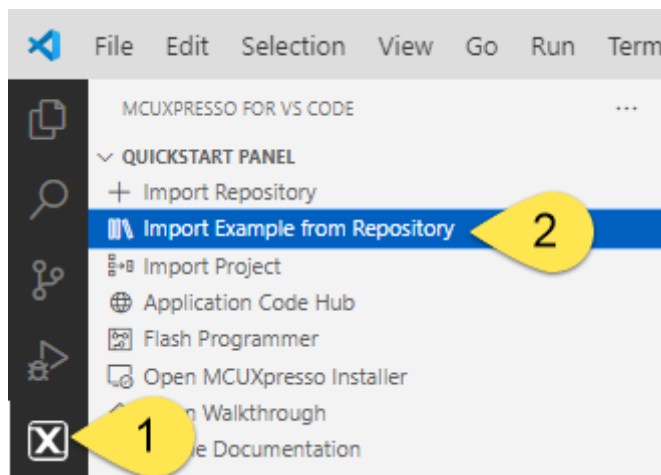
## Hello\_World Lab

---

After successfully installing Zephyr and setting up the VS Code extension for MCUXpresso, let's go through the detailed steps to run the Zephyr *hello\_world* application on the FRDM-MCXN947.

### 1. Import *hello\_world* sample

1. To import the *hello\_world* application from the Zephyr Repository, click **Import example from Repository** in the Quickstart Panel.



2. Select the following board settings to import the example:

- For selecting the board, type **n947** to find the board target **frdm\_mcxn947/mcxn947/cpu0**. Make sure that the board ends with **cpu0**.
- For selecting the example, type **hello** to find the application **zephyr/samples/hello\_world**
- For application type, select **Repository Application**.
- Click **Create** and the example should be added to the Projects view.

Import Example from Repository X

## Import Example from Repository

Repository: C:\Users\... \zephyrproject (Zephyr Repository) | v

Zephyr SDK: zephyr-sdk-0.16.5 (C:\Users\... \zephyr-sdk-0.16.5) | v

Board: NXP FRDM MCXN947 (CPU0) (frdm\_mcxn947/mcxn947/cpu0) | v

FRDM-MCXN947 are compact and scalable development boards for rapid prototyping of MCX N94 and N54 MCUs. [...]

Please refer to [README](#) file for more details.

Template: zephyr/samples/hello\_world (hello world) | v

A simple sample that can be used with any `supported board <boards>` and prints "Hello World" to the console. [...]

Tags: introduction  
Please refer to [README](#) file for more details.

App type: Repository application | v

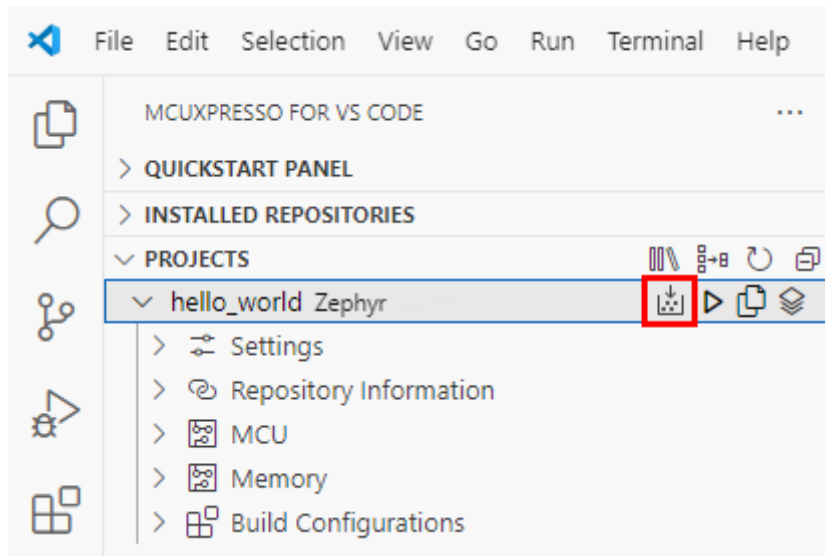
☐ Open readme file after project is imported

Create

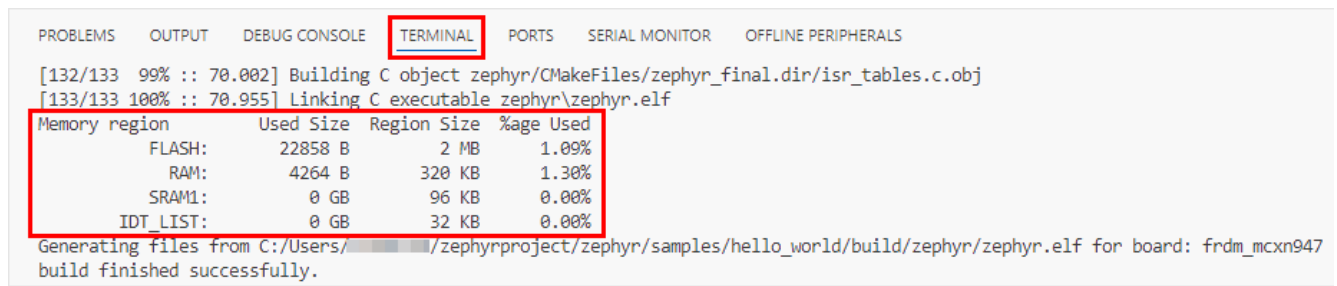
## 2. Build the application

We need to build *hello\_world* project to generate the application image. After the code builds without any errors, the application can be run on the board.

3. Build the project by clicking the **Build Selected** icon.



After the build, the Terminal view displays the memory usage (or compiler errors if any).

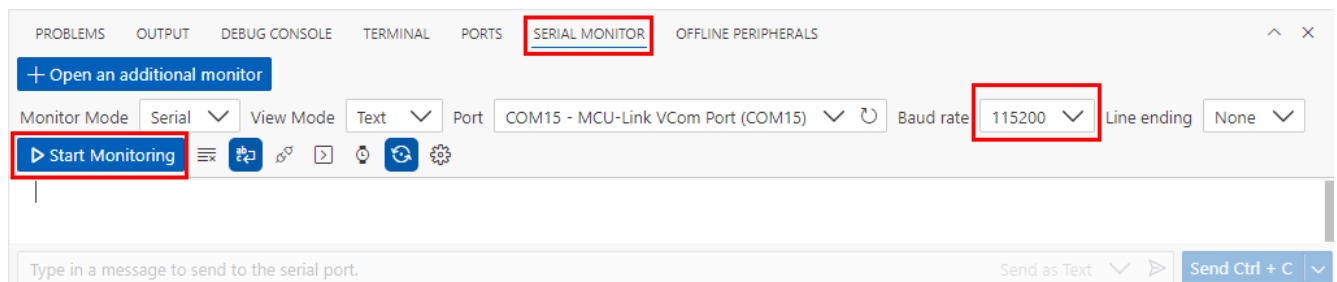


### 3. Connect Serial Monitor to the board

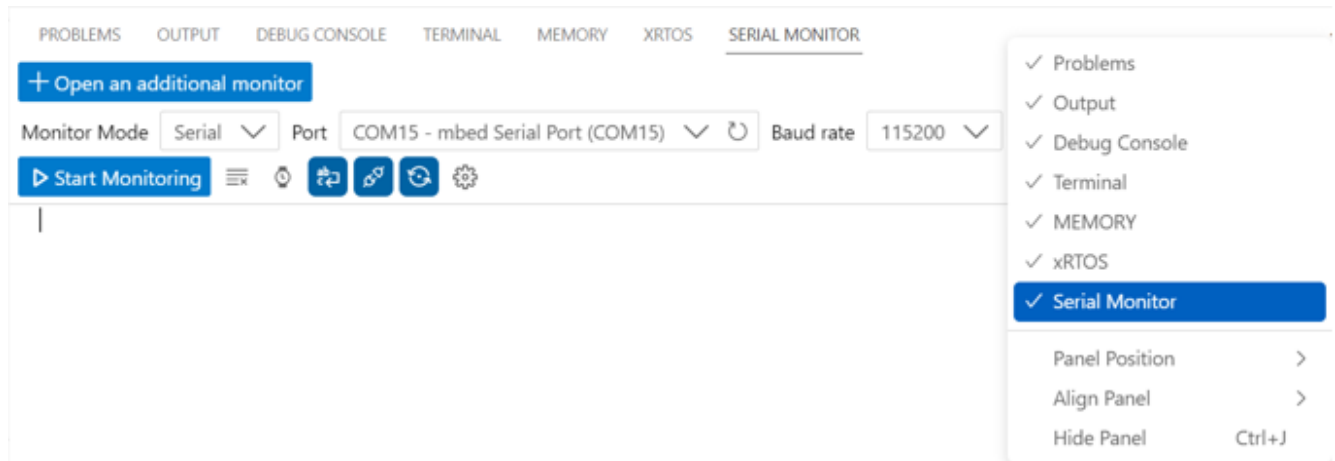
To use the Serial Monitor integrated into VS Code, connect the USB Type-C cable to J17 to power the board. The onboard debugger provides a USB-UART bridge to interface with the Serial monitor.

4. Connect the Serial monitor to the board's **COM port** and select the **Baud rate**. The screenshots show the board enumerates as COM15, but the COM port number may be different for you. To capture the data sent over UART, click **Start Monitoring**.

It can be disconnected by clicking **Stop Monitoring** after debug.

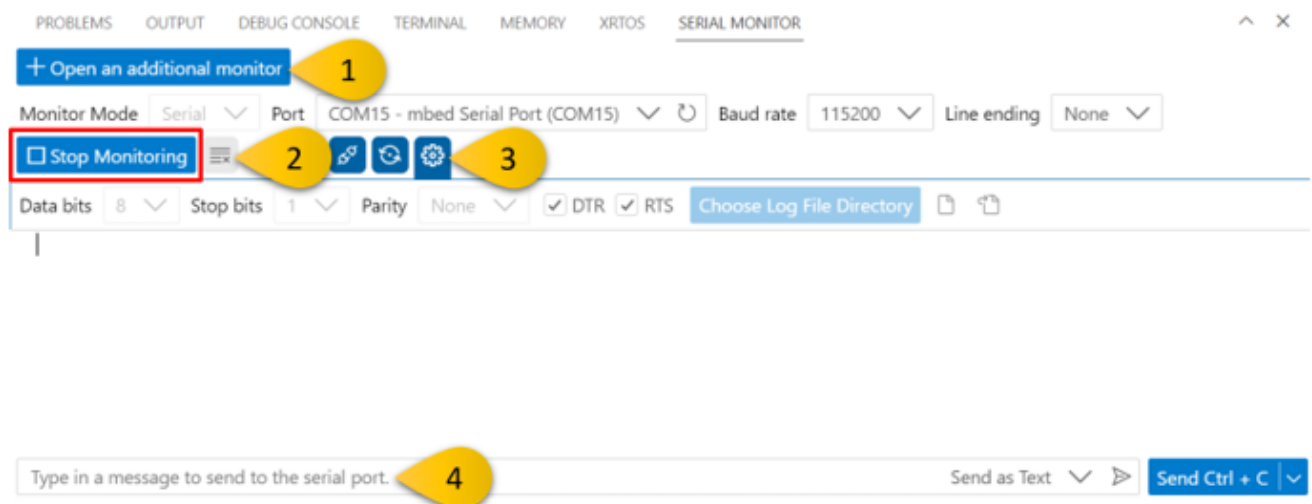


5. If the serial monitor view is disabled, right-click on the panel and select the Serial Monitor option.



6. Optional settings for the Serial Monitor:

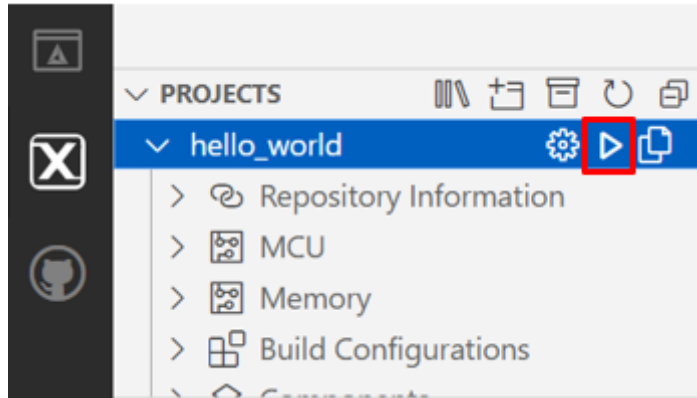
- 6.1. To add a monitor for another COM port select **Open an additional monitor**.
- 6.2. Click the icon to clear the output terminal.
- 6.3. Click the **gear** icon for additional settings (data bits, stop bits, parity, DTR, RTS).
- 6.4. To send a message to the serial terminal, type the command in the text box.



## 4. Debug the application

This section uses the GDB debugger to connect to the MCU, and program the flash.

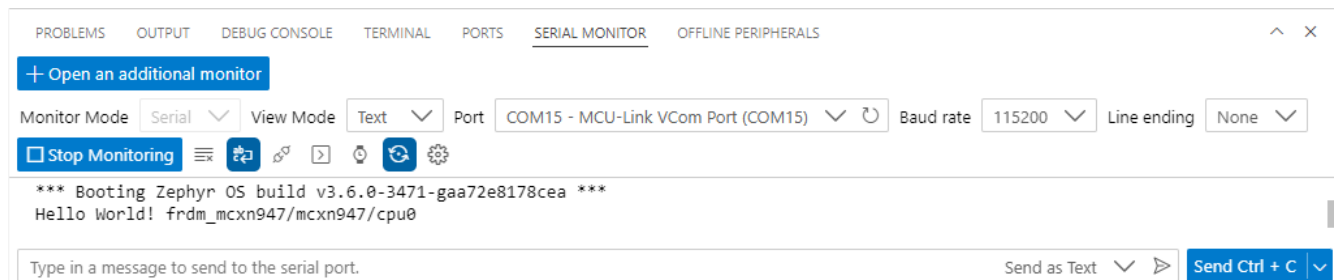
7. Click the play icon to **Debug** the application:



8. The execution will pause, and a debugger toolbar pops up at the top of VS Code. To continue execution click **Continue** on the debug options.



9. In the Serial Monitor view, the application prints the boot banner during the startup code and halts at `main()`. The boot banner also includes the version of Zephyr used in the build. In `main()`, the application prints `hello world`.



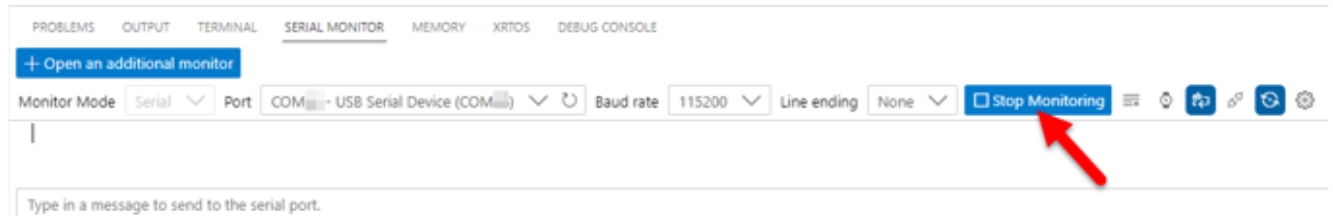
## 5. Clean up after lab

This lab is completed. But the following steps will clean up the VS Code workspace:

10. Click **Stop** to end the debug session.



11. Close all Editor tabs. Right-Click on a tab, and select **Close All**.
12. If this is the last lab and you are done using the board, you should disconnect the Serial Monitor. Find the Serial Monitor view, and click **Stop Monitoring**.



---

Lab completed. Return to the MCXN947 Zephyr Labs Overview [[Training Zephyr Getting Started MCXN947]]