

# MCUXpresso for Visual Studio Code

## Zephyr FRDM-MCXM947 Lab Pre-requisite

**May 8, 2024**

### Objectives

These are the pre-requisites to install before the hands-on lab. This guide is written for Windows10. But Linux or macOS can also be used following similar steps. The pre-reqs include installing:

- Visual Studio Code
- MCUXpresso Extension for VS Code
- Dependencies for Zephyr development, including the Zephyr SDK
- Zephyr Repository in VS Code
- MCUXpresso SDK Repository in VS Code

### Software Versions

The steps in these lab guides should also work with later versions of these tools. But this lab guide was written with the following versions:

- MCUXpresso extension for VS Code v1.7.50
- MCUXpresso Installer v1.2
- Zephyr post v3.6.0 release (use main branch)
- Zephyr SDK v0.16.5
- LinkServer v1.5.30
- Embedded Tools extension v0.7.0

# Installation Lab

## 1. Install VS Code and the MCUXpresso extension

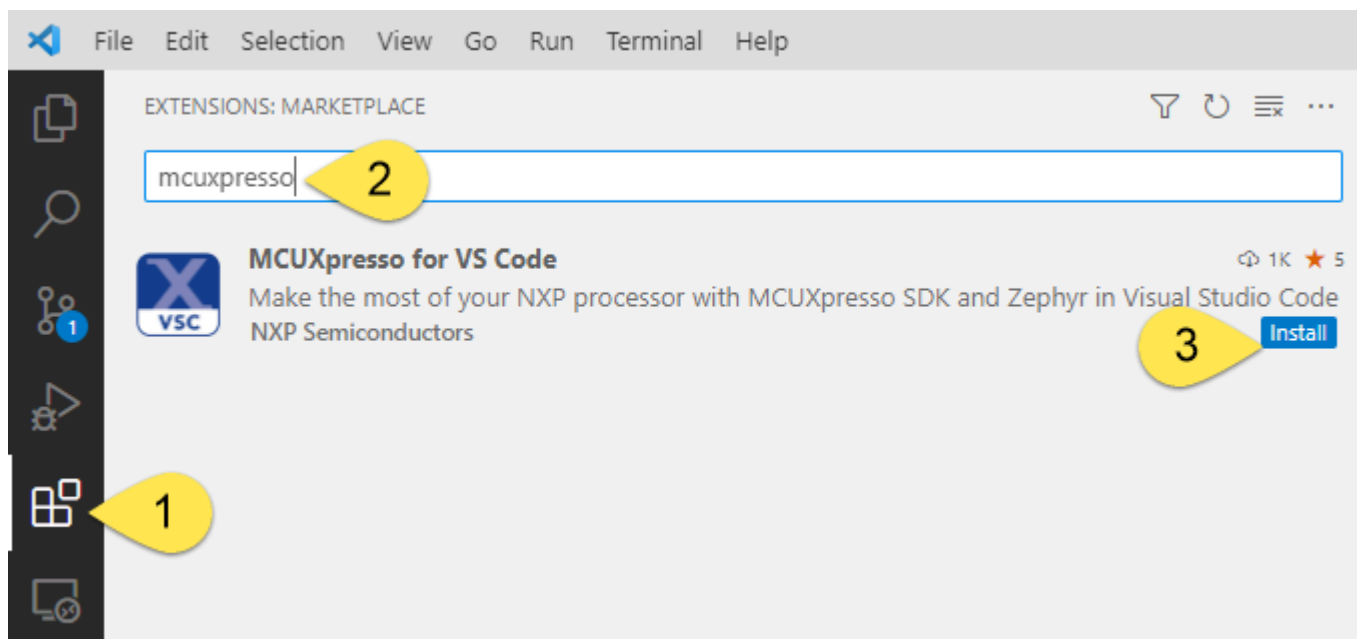
1. To install Visual Studio Code (VS Code), download the installer here:

<https://code.visualstudio.com/download>

2. Open VS Code, and install the MCUXpresso extension.

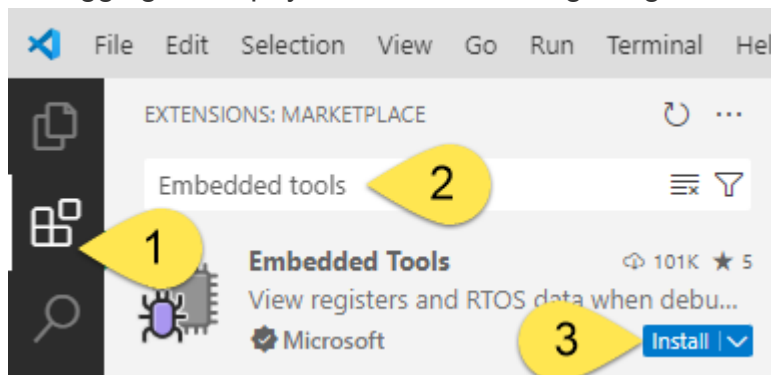
Click the Extensions view (Ctrl+Shift+X) in VS Code. Click the search field at the top of the Extensions view, and enter `mcuxpresso`.

Click the **Install** button to install the extension.



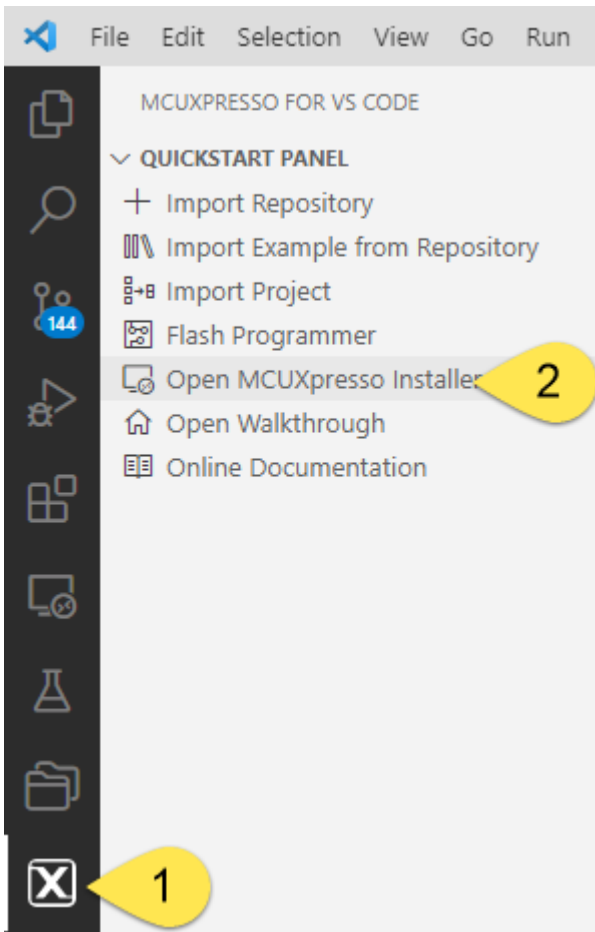
The MCUXpresso extension will be added to the Installed list after successful installation.

3. Click the **Extensions view**, type **embedded tools** in the search field, and then click the **Install** button for the Microsoft Embedded Tools extension. This extension enables thread aware debugging for Zephyr, used in the debug lab guide.



## 2. Install the Zephyr dependencies and debugger tools

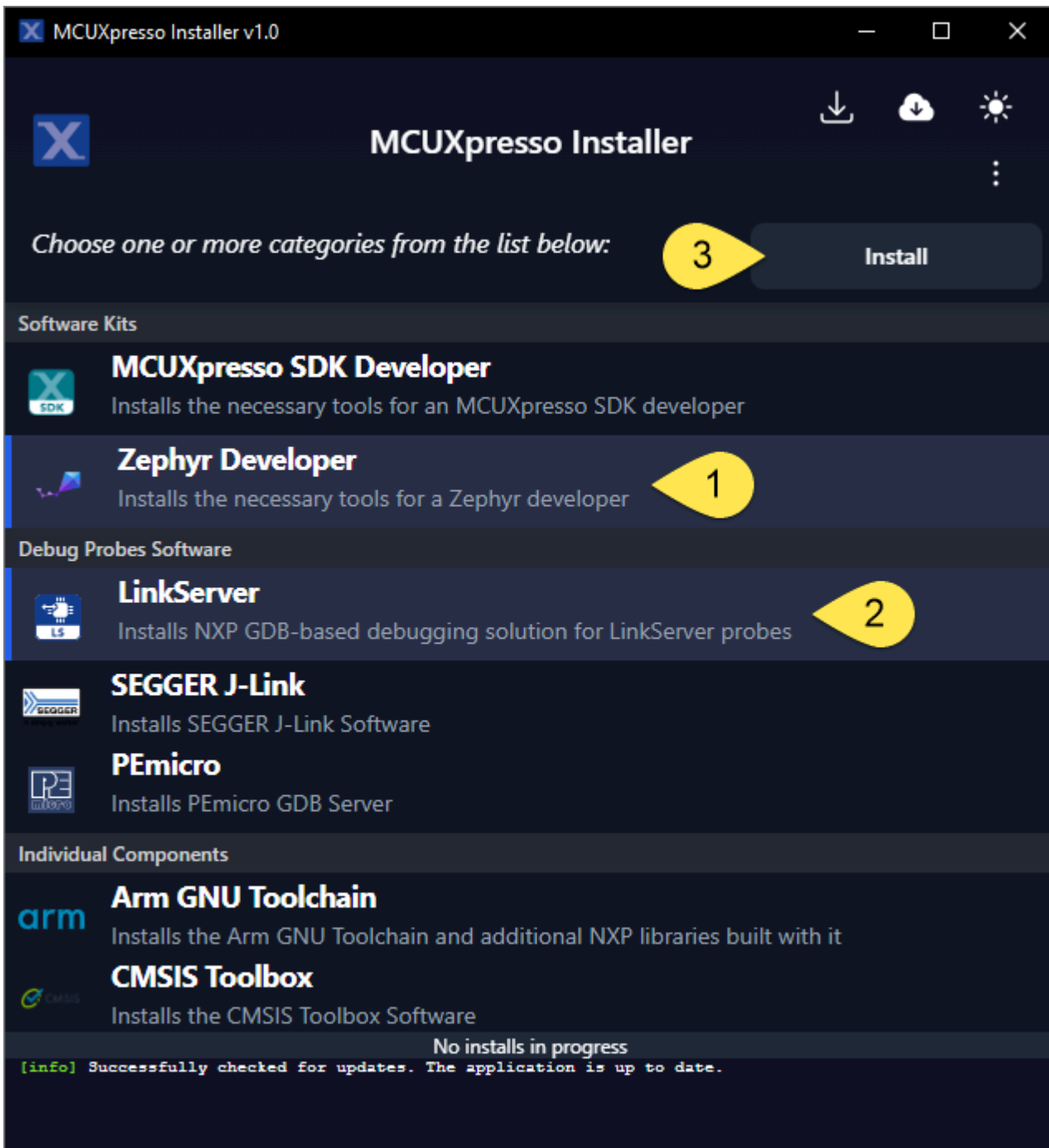
4. Open the MCUXpresso Extension. In the Quickstart Panel, click `Open MCUXpresso Installer`.



5. The Zephyr Developer option is required to support CMake, Python, the Zephyr SDK (includes the GCC build tools) and other Zephyr dependencies. LinkServer is also required to debug with the NXP on-board probes.

**Note** this lab guide assumes a LinkServer debug probe is used, which is default on the evaluation boards. When using other supported debug probes, this Installer can also install Segger JLink and PEmicro tools.

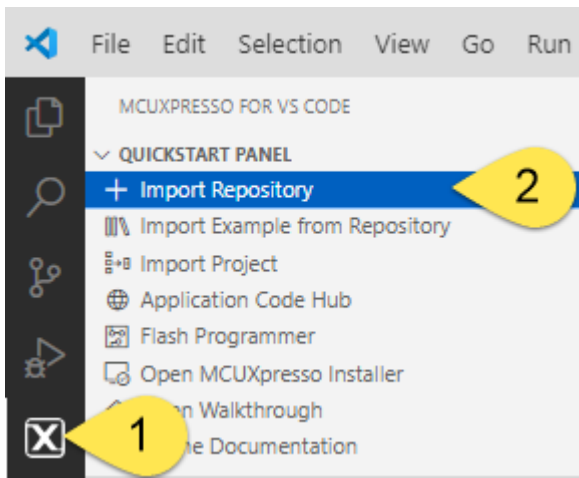
After selecting the install options, click the Install button. The status bar at the bottom displays the status of the installation.



6. Restart VS Code to enable the changes to environment variables.

### 3. Import the software repositories

7. To import the Zephyr repository, open the MCUXpresso view in VS Code and click Import Repository in the Quickstart Panel.



8. Select the folder location `C:\Users\...\zephyrproject` and Zephyr Repository. For revision, change to `main` to checkout the latest code, which supports the FRDM-MCXXN947. When finished, click Import. The repository should be added to the Installed Repository view once the import is successful.

## Import Repository

REMOTE
REMOTE ARCHIVE
LOCAL
LOCAL ARCHIVE

Location:
Browse...

Note: Path doesn't exist. Folder(s) will be created.

Repository:

Revision:

Manifest file:

Import

**Note** This import step takes a long time to clone the Zephyr repo, and dependency repos (currently 55!). VS Code shows a progress bar pop-up with the status of the West tool as it clones all the repos.

9. The MCUXpresso SDK repository can be imported the same way. This is not required when working with Zephyr. But the debugger lab guide shows how to add the Peripheral Register view to the debugger, and this requires some SVD files located in the MCUXpresso SDK repository. To add this repo, select the folder location:

**C:\Users\...\mcux-sdk**

# Import Repository

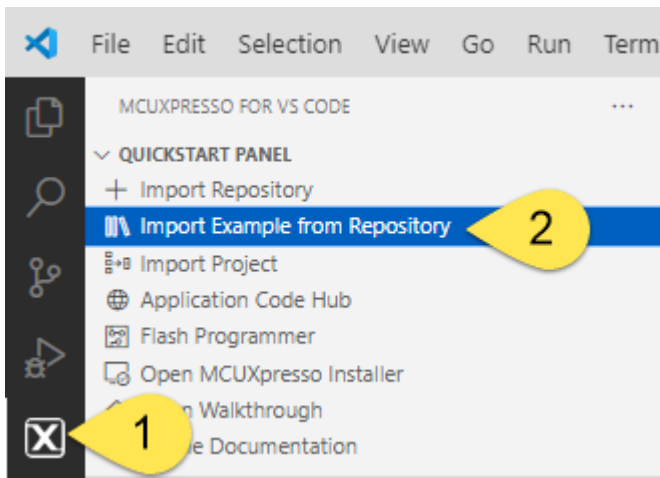
REMOTE	LOCAL ARCHIVE	LOCAL
Location:	<div>C:\Users\...\mcux-sdk</div> <div>Note: Path doesn't exist. Folder(s) will be created.</div>	<div>Browse...</div>
Repository:	MCUXpresso SDK (https://github.com/NXP-mcuxpresso/mcux-sdk)   v	
Revision:	main   v	
Manifest file:	Enter the manifest file name (optional)	
<input type="checkbox"/> Clone all examples (this will take a while)		
<div>Import</div>		

The repositories are successfully added to the Installed Repository view once the import is successful.



## 4. Import and build *hello\_world* sample

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



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

- For selecting the board, type **mcx** 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) ▼

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

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

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) ▼

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

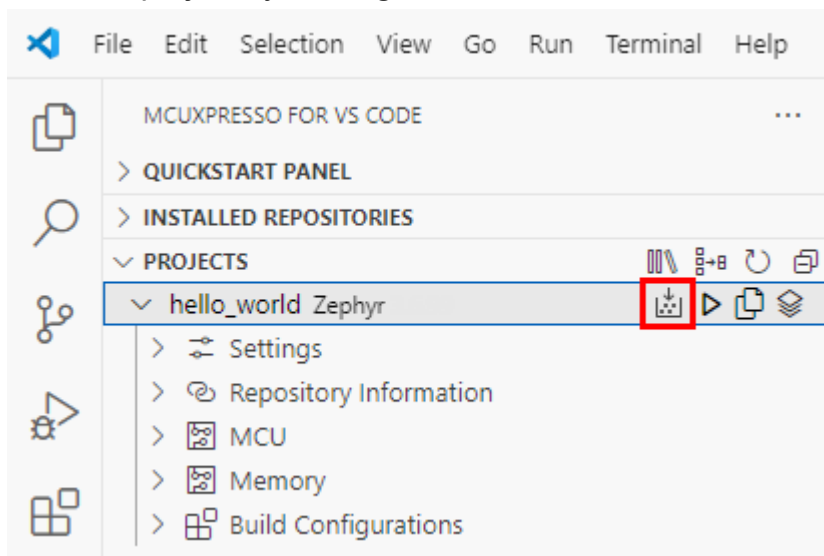
Tags: introduction  
Please refer to [README](#) file for more details.

App type: Repository application ▼

☐ Open readme file after project is imported

Create

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



After the build, the debug console window displays the memory usage (or compiler errors if any).



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SERIAL MONITOR  OFFLINE PERIPHERALS

[132/133 99% :: 70.002] Building C object zephyr/CMakeFiles/zephyr_final.dir/isr_tables.c.obj
[133/133 100% :: 70.955] Linking C executable zephyr\zephyr.elf
Memory region      Used Size  Region Size  %age Used
  FLASH:           22858 B      2 MB          1.09%
    RAM:           4264 B      320 KB          1.30%
   SRAM1:            0 GB       96 KB          0.00%
  IDT_LIST:         0 GB       32 KB          0.00%

Generating files from C:/Users/nxa14743/zephyrproject/zephyr/samples/hello_world/build/zephyr/zephyr.elf for board: frdm_mcxn947
build finished successfully.
```

With a successful build of the *hello\_world* sample, the pre-requisites are complete. The remaining labs require the FRDM-MCXN947 board, and will be completed during the hands-on training.