Zephyr hello_world Sample Lab

Rev1.0, May29, 2022

Objectives

In this lab, you will learn

- → (optional) Flash J-Link firmware onto the on-board OpenSDA circuit
- → Build the hello_world example for the RT1060-EVKB
- → Flash and run application

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.

- Follow Zephyr_Installation_Lab.pdf
- Terminal Program, like PuTTY or Tera Term

Hardware Requirements

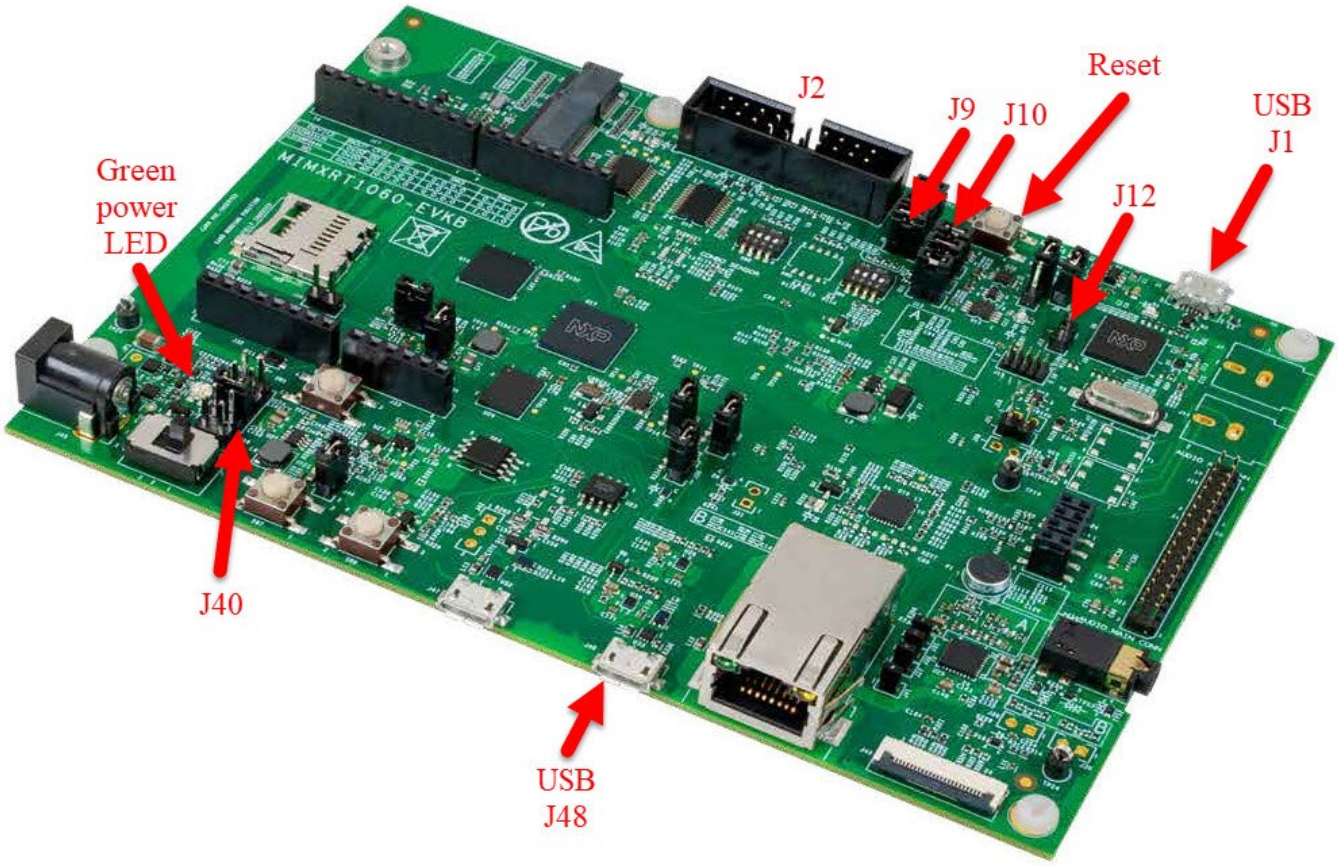
- MIMXRT1060-EVKB
- Micro-USB cable (may need two cables, see below)
- (Optional) External J-Link debug probe with SWD cable

Preparing the Hardware

NXP recommends using Segger Jlink debug probes for Zephyr development, and these labs assume Jlink is used. There are two options using this EVK with Jlink:

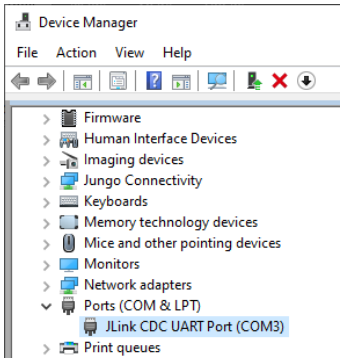
- Use external Jlink debug probe
- Use on-board debugger with Jlink firmware

For more details, see: Using J-Link with MIMXRT1060-EVKB



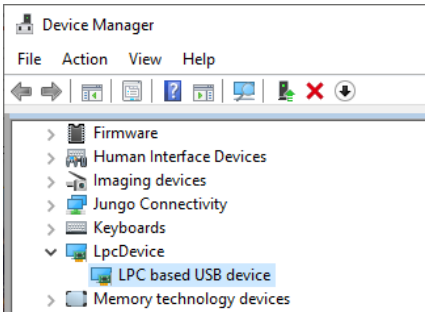
Use on-board debugger with Jlink firmware

The on-board debugger has CMSIS-DAP firmware by default. If the firmware has not already been updated, install the tool LPCScrypt to update the firmware. If the Jlink firmware is already installed, connecting USB to J1 will enumerate as a Jlink COM port like this:



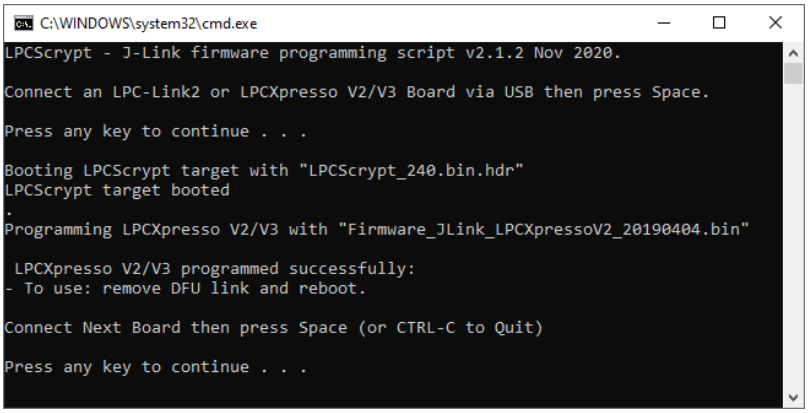
To update the firmware for Jlink, configure the board like this:

- Unplug the board
- Install J12 to force bootloader mode
    - If needed, borrow jumper J9 or J10
- Plug USB cable to J1.
- Red power LED should be lit near J1
    - Device manager should show LpcDevice, Indicates bootloader enumerated



Run LPCScrypt script to update the firmware:

- Run script **C:\nxp\LPCScrypt_2.1.2_57\scripts\program_JLINK.cmd**. Script should show success like image below.
- When done, unplug EVK

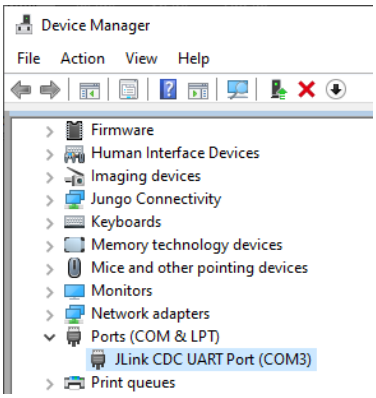


Configure the EVK for use with the on-board Jlink. Note that this Jlink firmware will no longer power the EVK from USB J1. One option used here is to use a second USB cable to power the EVK. Another option is rework the board to force power enabled through J1; refer to the schematics for this rework option. These steps below will avoid reworking the board, and configure the EVK for this lab:

- Unplug the board
- Remove J12 to use the onboard debugger
- Install jumpers J9 and J10 – this connects SWD signals from onboard debug circuit. These jumpers are installed by default.
- Plug USB cable to J1 - provides connection for J-Link debugger and UART/USB bridge.
- Move the jumper on J40 to short pins 3-4 (default shorts pins 5-6) - Powers the EVK from USB J48.
- Connect a 2nd USB cable to J48 - powers the EVK.

With the above steps completed, confirm the board is in this state for the following lab:

  - USB enumeration shows Jlink CDC COM port, like image below
  - green LED next to J40 will be lit when the EVK is properly powered.



Use external Jlink debug probe

For this option, the Jlink probe must be Rev9 or greater to debug ARM Cortex-M7 or M33 cores. Configure the EVK with these settings:

- Unplug the board
- Remove jumpers J9 and J10, to disconnect the SWD signals from onboard debug circuit. These jumpers or installed by default.
- Use default power selection on J40 with pins 5-6 shorted.
- Connect the J-Link probe to J2, 20-pin dual-row 0.1" header.
- Power the EVK with one of the power supply options. Typically USB connector J1 is used to power the board, and provides a UART/USB bridge through the onboard debug circuit.
- Green LED next to J40 will be lit when the EVK is properly powered.

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.

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

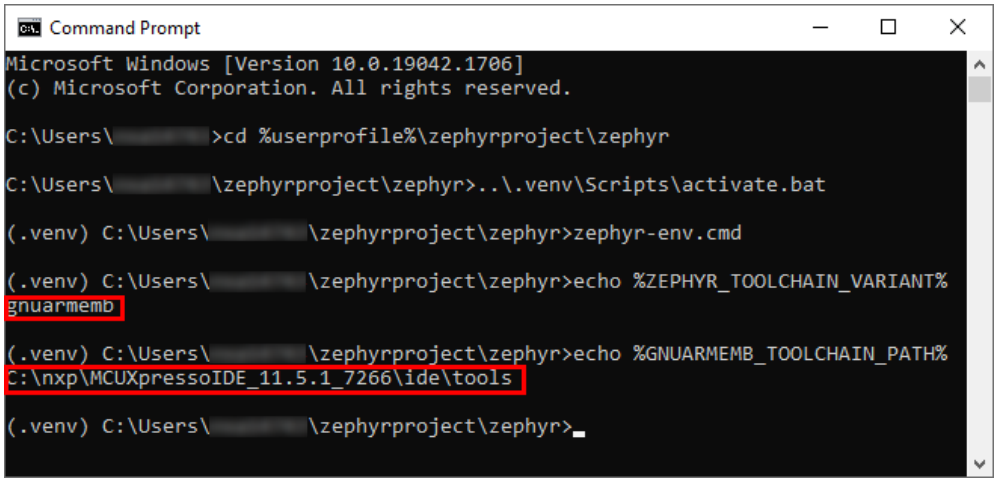
4. Set the environmental variables. Note, when using a command file like this to set the variables, the command file needs to be executed once anytime a new command window is opened:
   zephyr-env.cmd

5. Ensure the environmental variables are being set correctly:
   echo %ZEPHYR_TOOLCHAIN_VARIANT%

   ➔ You should see gnuarmemb

   echo %GNUARMEMB_TOOLCHAIN_PATH%

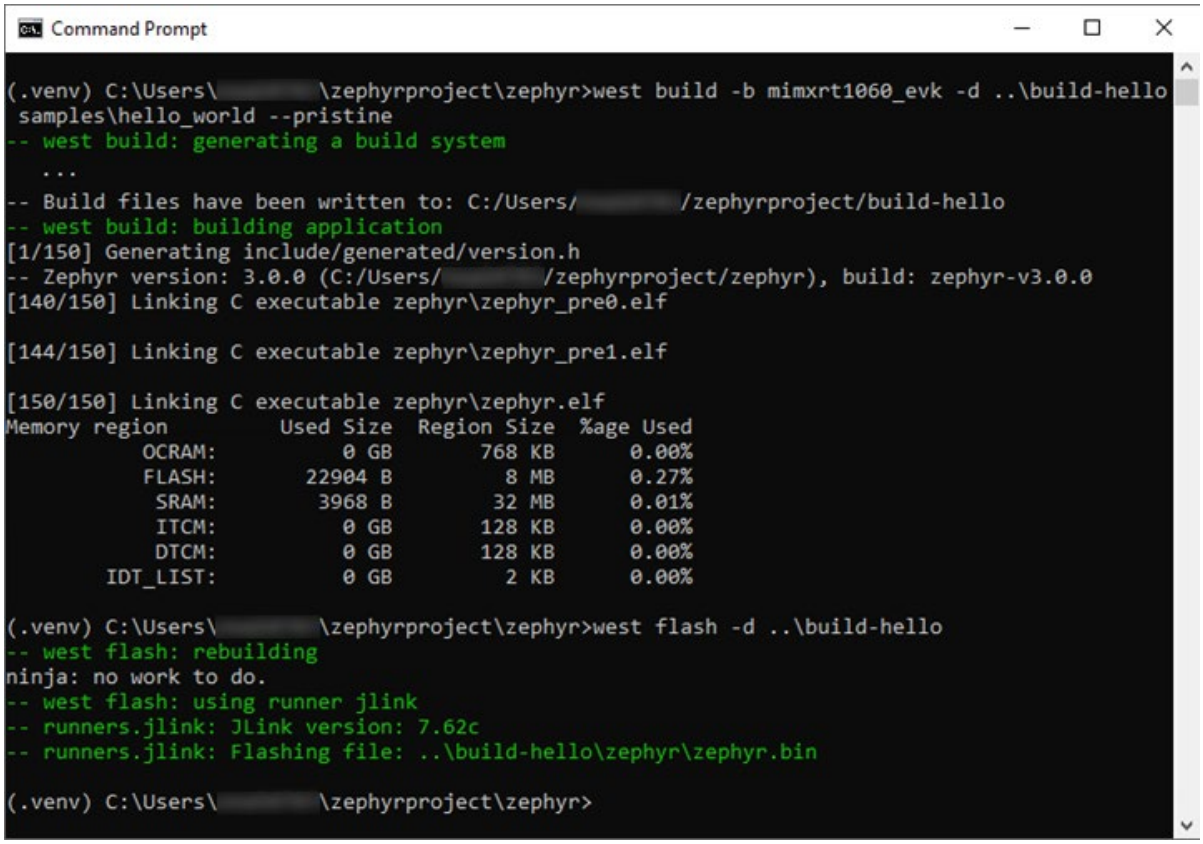
   ➔ You should see the MCUXpresso IDE path.



Build and flash your project

6. Build the hello_world sample application using west:
   west build -b mimxrt1060_evk -d ..\build-hello samples\hello_world --pristine

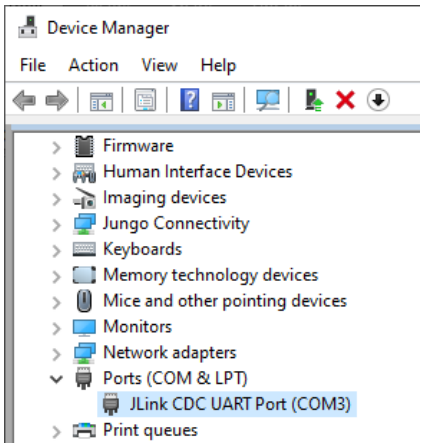
7. Now, connect the mimxrt1060 board (remember to use 2 USB connectors if using on-board J-Link Firmware, on J48 for power and J1 for debugger), and flash the example:
   west flash -d ..\build-hello



Connect Terminal to EVK

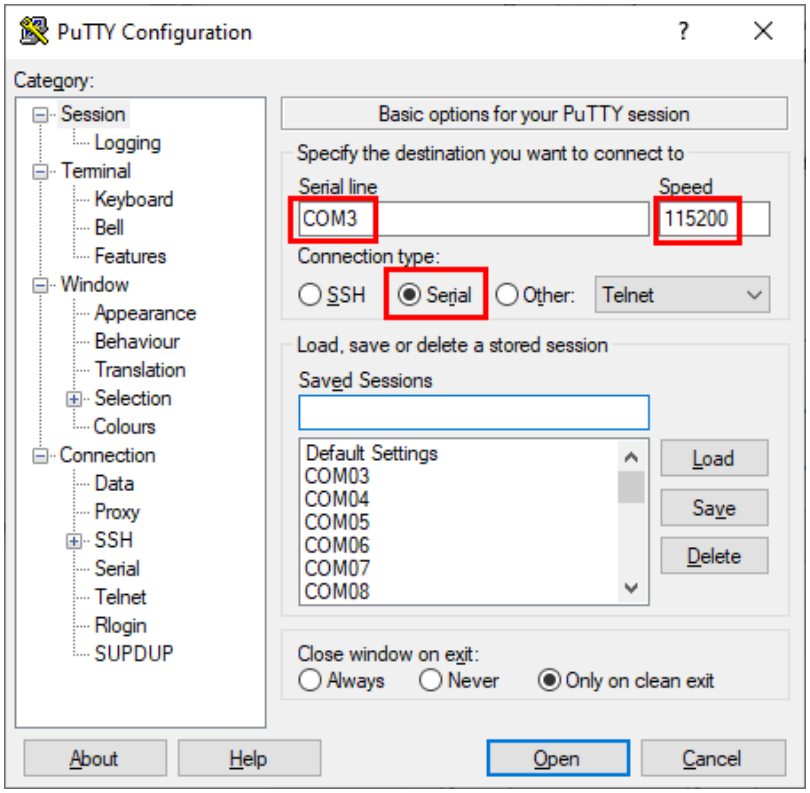
The on-board debugger provides a USB/UART bridge to interface to the Console/Shell of the Zephyr application.

8. Open Windows Device Manager to confirm the COM port that is enumerated for the EVK. In this guide, the port is COM3

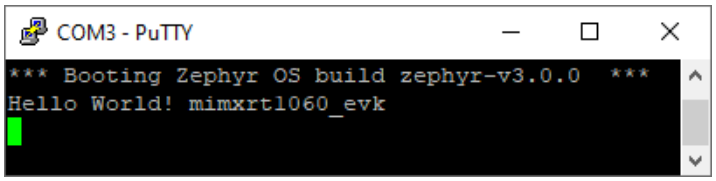


9. Open a serial terminal like PuTTY.

- Select the Serial Connection type
- Enter your COM port and Baud rate to 115200:



10. Reset the board (SW1), and you will see the hello_world application print to the terminal



This completes the lab.

Revision History

Rev	Date	Details
1.0	5/29/2022	Initial Version