Zephyr MCUboot Lab

Rev1.0, June 20, 2021

Pre-Requisites

If this lab is done in sequence with other labs, there a no pre-requisites.

If you have not gone through these steps, please go back and:

1. Download the latest MCUXpresso IDE
2. Download the latest J-Link Software and Documentation Pack (Version v7.22 is recommended. Avoid v7.20b, it causes issues with shell and prevents the example from working properly)
3. Download a Terminal program, like PuTTY or TeraTerm
4. Go through the Zephyr Getting-Started Lab guide to install the latest release version Zephyr (2.6.0) and all its dependencies

Objectives

In this lab, you will learn

- → About MCUboot in Zephyr
- → How to build and flash MCUboot in Zephyr
- → How to use mcumgr-cli tool with MCUboot
- → How to build and update the user application by MCUboot

Hardware

- × MIMXRT1060-EVK
- × Micro-USB cable
- × J-Link Probe (see options)

Lab Overview

MCUboot is an open-source bootloader (not to be confused with NXP's MCUboot with the same name), which Zephyr Project integrates as a module. MCUboot supports robust firmware updates with multiple images, and authenticated boot. MCUmgr is a protocol and tool used to manage updating the firmware images (similar to blhost from NXP). This lab will show how to implement MCUboot with Zephyr, and using the MCUmgr tool to update firmware images with the following steps:

- Install the mcumgr-cli tool
- Build and flash MCUboot. Check if the bootloader is running with no application installed.
- Build SMP_svr sample for MCUboot using shell, flash to board. Check MCUboot booting SMP_svr
- Demonstrate shell commands of SMP_svr, how they relate to MCUboot
- Use mcumgr-cli tool to talk to MCUboot, get details

- Re-build the SMP_svr application again with different version numbers (e.g., an upgrade of the same application)
- Use mcumgr to download the newly generated SMP_svr ver1 binary to RT1060
- Reboot MCUboot, use SMP_svr and mcumgr commands to see how both apps are in flash now, and differences compared to before with single app: SMP_svr ver0 in Slot0, and SMP_svr ver1 in Slot1. MCUboot still boots to SMP_svr ver0.
- Use mcumgr commands to confirm the SMP_svr ver1 image. Show how the new image is copied to Slot0, and after reboot, MCUboot now boots the new version.

Installing MCUmgr tool

MCUmgr provides a command-line tool for managing remote devices. The tool is written in the Go programming language. So first, the Go language must be downloaded and installed from the link below:

https://golang.org/

After the “Go language” is installed, the mcumgr tool can be installed by opening a regular user command prompt terminal and typing the following:

```
go get github.com/apache/mynewt-mcumgr-cli/mcumgr
```

For more information on this mcumgr tool, visit: https://docs.zephyrproject.org/latest/guides/device_mgmt/mcumgr.html

Build and flash MCUboot onto MIMXRT1060-EVK board.

Use these commands in the Windows command window:

```
cd %userprofile%\zephyrproject\

zephyr\zephyr-env.cmd (this command is needed once after opening command window)
```

Build the MCUbootloader project:

```
west build -b mimxrt1060_evk -d build-mcuboot bootloader/mcuboot/boot/zephyr/
```

Connect the MIMXRT1060-EVK vie USB, and Flash the bootloader to memory:

```
west flash -d build-mcuboot
```

Running now prints this to terminal:

```
*** Booting Zephyr OS build zephyr-v2.6.0-124-gdce85ebe4500  ***
I: Starting bootloader
I: Primary image: magic=unset, swap_type=0x1, copy_done=0x3, image_ok=0x3
I: Scratch: magic=unset, swap_type=0x1, copy_done=0x3, image_ok=0x3
```

```
I: Boot source: primary slot
I: Swap type: none
E: Unable to find bootable image
```

Build and Flash “smp_svr” application version 0 (default).

The smp_svr sample implements a server for the MCUmgr tool, and enables the application to download new firmware images, and fetch details from MCUboot. These Kconfig options build the sample to use the shell as interface, and provides an RSA key pair for signing the firmware image.

```
west build -b mimxrt1060_evk -d build-smp-svr-v0
zephyr/samples/subsys/mgmt/mcumgr/smp_svr/ -- -DOVERLAY_CONFIG='overlay-shell.conf'
-DCONFIG_MCUBOOT_SIGNATURE_KEY_FILE=\"bootloader/mcuboot/root-rsa-2048.pem\"
```

Flash the application. Because this application was built for MCUboot, it will be flashed to a flash partition called Slot0. The MCUboot bootloader will still be present in flash as well. MCUboot will boot a valid and authenticated image present in Slot0:

```
west flash -d build-smp-svr-v0
```

After Reset, MCUboot confirms the smp_svr app in Slot0, and boots it. The board prints on RS232 terminal:

```
*** Booting Zephyr OS build zephyr-v2.6.0-124-gdce85ebe4500  ***
I: Starting bootloader
I: Primary image: magic=unset, swap_type=0x1, copy_done=0x3, image_ok=0x3
I: Scratch: magic=unset, swap_type=0x1, copy_done=0x3, image_ok=0x3
I: Boot source: primary slot
I: Swap type: none
I: Bootloader chainload address offset: 0x10000
I: Jumping to the first image slot
*** Booting Zephyr OS build zephyr-v2.6.0-124-gdce85ebe4500  ***

uart:~$
```

Using the MCUmanager tool for connection with the bootloader

The MCUmgr tool will use the same UART to connect to the smp_svr application now executing. Therefore, the terminal program needs to be disconnected from the COM port, or MCUmgr will not be able to connect. Close the terminal program, or disconnect from COM port.

Connect MCUmgr tool using the command window with command below, where dev=COMx needs to match the COM port number for your EVK:

```
mcumgr conn add COMPORT type="serial" connstring="dev=COM17,baud=115200,mtu=512"
```

Check the communicatiomn with the board:

```
mcumgr -c COMPORT image list
```

Command console returns like below. Note the version number for the Slot0 image:

```
Images:
 image=0 slot=0
    version: 0.0.0
    bootable: true
    flags: active confirmed
    hash: 74167b9373435a7a92de33143fbecd888a5b88a23adcc894840e66c2e649023f
Split status: N/A (0)
```

Updating the "smp_svr" application to ver. 1

MCUboot supports version numbers for the firmware images in their image header. We will build another version of smp_svr, this time with a newer version number, using the Kconfig option CONFIG_MCUBOOT_EXTRA_IMGTOOL_ARGS:

```
west build -b mimxrt1060_evk -d build-smp-svr-v1
zephyr/samples/subsys/mgmt/mcumgr/smp_svr/ -- -DOVERLAY_CONFIG='overlay-shell.conf'
-DCONFIG_MCUBOOT_SIGNATURE_KEY_FILE=\"bootloader/mcuboot/root-rsa-2048.pem\" -
DCONFIG_MCUBOOT_EXTRA_IMGTOOL_ARGS=\"--version=1.0.0\"
```

And now use MCUmgr and MCUboot for updating the application:

```
mcumgr -c COMPORT image upload -e build-smp-svr-v1/zephyr/zephyr.signed.bin
```

When the update is sucessfully completed, then check the memory content by listing the images again:

```
mcumgr -c COMPORT image list
```

The command console now returns below. Notice the new version is in Slot1:

```
Images:
 image=0 slot=0
    version: 0.0.0
    bootable: true
    flags: active confirmed
    hash: 74167b9373435a7a92de33143fbecd888a5b88a23adcc894840e66c2e649023f
 image=0 slot=1
    version: 1.0.0
    bootable: true
    flags:
    hash: 869fb41491c3262405fcd34626c1a33800d08498642ced5ac693a53f1b15cddb
Split status: N/A (0)
```

Now we can run a test on the new image, before it is used.

This command should mark a test upgrade, which means that after the next reboot the bootloader will execute the upgrade and jump into the new image. If no other image operations are executed on the newly running image, it will revert back to the image that was previously running on the device on the subsequent reset. When a test is requested, flags will be updated with *pending* to inform that a new image will be run after a reset.

To test a new upgrade image the "*test*" command is used. The test command requires the image hash shown in the previous step. Be aware, your hash value will be different. Copy the hash value from previous step, and use in this command:

```
mcumgr -c COMPORT image test
869fb41491c3262405fcd34626c1a33800d08498642ced5ac693a53f1b15cddb
```

The console now returns:

```
Images:
 image=0 slot=0
    version: 0.0.0
    bootable: true
    flags: active confirmed
    hash: 74167b9373435a7a92de33143fbecd888a5b88a23adcc894840e66c2e649023f
 image=0 slot=1
    version: 1.0.0
    bootable: true
    flags: pending
    hash: 869fb41491c3262405fcd34626c1a33800d08498642ced5ac693a53f1b15cddb
Split status: N/A (0)
```

After a reset and rebooting (wait 10 seconds after the reset), the output from the listing will be as below, meaning that the new version of the application has been booted.

Use command:

```
mcumgr -c COMPORT image list
```

Console returns:

```
Images:
 image=0 slot=0
    version: 1.0.0
    bootable: true
    flags: active
    hash: 869fb41491c3262405fcd34626c1a33800d08498642ced5ac693a53f1b15cddb
 image=0 slot=1
    version: 0.0.0
    bootable: true
    flags:
    hash: 74167b9373435a7a92de33143fbecd888a5b88a23adcc894840e66c2e649023f
Split status: N/A (0)
```

The new application booted just for test, because it has not been confirmed yet.

It means that after the subsequent Reset, it will revert back to the previous image (version 0), and the new version1 image will be placed back in Slot1.

After the Reset the list command returns:

```
mcumgr -c COMPORT image list


Images:
 image=0 slot=0
    version: 0.0.0
    bootable: true
    flags: active confirmed
    hash: 74167b9373435a7a92de33143fbecd888a5b88a23adcc894840e66c2e649023f
 image=0 slot=1
    version: 1.0.0
    bootable: true
    flags:
    hash: 869fb41491c3262405fcd34626c1a33800d08498642ced5ac693a53f1b15cddb
Split status: N/A (0)
```

When the new version has been booting properly and everything was fine, then we can use it permanently. It means that we need to *confirm* the new version. Again, you need to replace the hash in this command with your hash value:

```
 mcumgr -c COMPORT image confirm
869fb41491c3262405fcd34626c1a33800d08498642ced5ac693a53f1b15cddb


Images:
 image=0 slot=0
    version: 0.0.0
    bootable: true
    flags: active confirmed
    hash: 74167b9373435a7a92de33143fbecd888a5b88a23adcc894840e66c2e649023f
 image=0 slot=1
    version: 1.0.0
    bootable: true
    flags: pending permanent
    hash: 869fb41491c3262405fcd34626c1a33800d08498642ced5ac693a53f1b15cddb
Split status: N/A (0)
```

And after the consequent Resets, the new version will be permanently used for booting.

We can again check, using the list command:

```
mcumgr -c COMPORT image list

Images:
 image=0 slot=0
    version: 1.0.0
    bootable: true
    flags: active confirmed
    hash: 869fb41491c3262405fcd34626c1a33800d08498642ced5ac693a53f1b15cddb
 image=0 slot=1
    version: 0.0.0
    bootable: true
    flags:
    hash: 74167b9373435a7a92de33143fbecd888a5b88a23adcc894840e66c2e649023f
Split status: N/A (0)
```

NOTE: The previous version can be manually reverted back using the “confirm” command with the appropriate hash value.

NOTE: When applications are built, they can be set to automatically confirm the image using the Kconfig -DCONFIG_MCUBOOT_GENERATE_CONFIRMED_IMAGE=y. This will skip the manual steps we just did of testing and confirming the new app. Instead, MCUboot will copy the new image to Slot0 and boot to it.

Additional resources

- MCUmgr - detailed explanation of all the commands is available at: https://docs.zephyrproject.org/latest/guides/device_mgmt/mcumgr.html
- Zephyr Device Firmware Upgrade using MCUboot
- SMP Server Sample readme

Revision History

Rev	Date	Details
1.0	6/20/2021	Initial Version