delta_dfu_lib

This is a user guider to tell you how to test the delta dfu , it is really ease for you to merge it to your SDK.

A delta firmware upgrade consists of the following steps:

- install essential tools
- pull the new boot code to replace your old code
- prepare test application
- generate patch file and transfer it to the secondary slot
- Restart MCU to perform the update
- Checking the result of the delta update

(1). install essential tools

you should install detools on your PC: enter "**pip install detools**" command in the python environment.

you should install cryptography,intelhex,click,cbor: enter "**pip install -r requirements.txt**" command in the python environment.

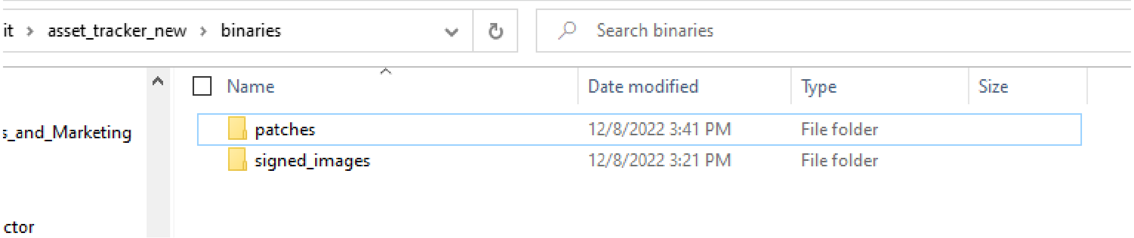
(2). pull the new boot code to replace your old code

you should pull the new boot code from below url and replace the boot folder in your SDK directory (v2.x.x/bootloader/ mcubboot/boot),
remember to change the folder name to boot or copy the folder contents to boot folder.

https://github.com/Noy0908/delta_dfu_lib.git The latest branch is dev

(3). prepare test application

- you can test it on any application samples, but need to create two folders in your project root directory to save images and patch file.
  one is "binaries/signed_images" which used to save source image and target image; the other is "binaries/patches" which used to save patch image.



Name	Date modified	Type	Size
patches	12/8/2022 3:41 PM	File folder	
signed_images	12/8/2022 3:21 PM	File folder	

- copy the scripts folder("boot/zephyr/scripts") from delta_mcuboot folder to the root directory of your application ,
  and double-click the "scripts/patch.exe" file when generating the patch image.
- allocate your flash partition. you can create a "pm_static.yml" file in your project root directory to redefine the flash partition.

  remember that you must define the primary slot and secondary slot, and these two slots support differnet size.
- enable delta dfu. you can create a "child_image/mcuboot.conf" file to set these macros:

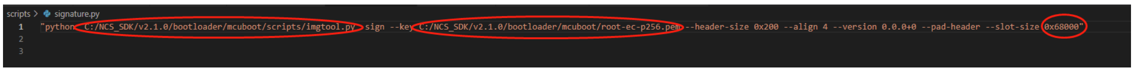
> `CONFIG_BOOT_MAX_IMG_SECTORS=160`
>
> `CONFIG_BOOT_UPGRADE_APP_DELTA=y`

- enable mcuboot in your project config files(prj.conf).

  > `CONFIG_BOOTLOADER_MCUBOOT=y`
  >
  > `CONFIG_IMG_MANAGER=y`
  >
  > `CONFIG_MCUBOOT_IMG_MANAGER=y`

(4). generate patch file and transfer it to the secondary slot

- Edit "scripts/signature.py" file. replace the imgtool.py path and root-ec-p256.pem path in the file with your own path, and modify the primary slot size to your own define.



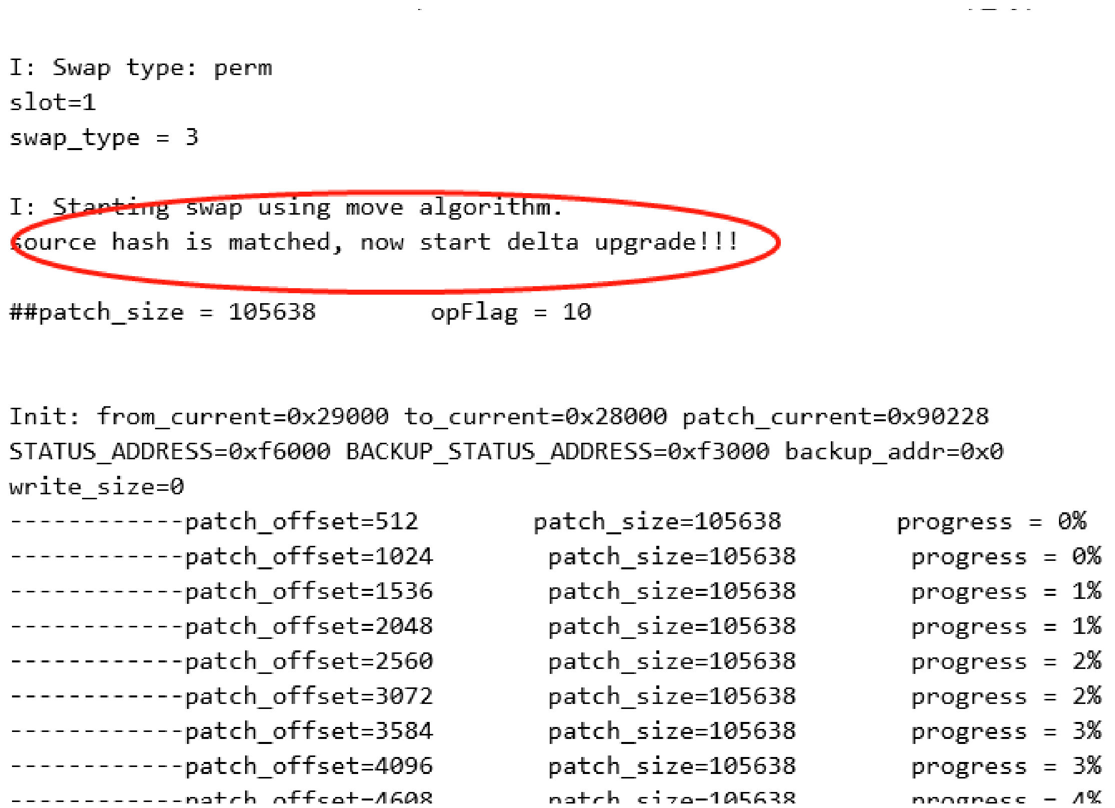
scripts > signature.py

```
"python C:/NCS_SDK/v2.1.0/bootloader/mcuboot/scripts/imgtool.py sign --key C:/NCS_SDK/v2.1.0/bootloader/mcuboot/root-ec-p256.pem --header-size 0x200 --align 4 --version 0.0.0+0 --pad-header --slot-size 0x68000"
```

- Generate the patch file. The patch file is generated by comparing the difference between the source image and the target image. The source image is a binary file which converted from the "app_signed.hex" file that compiled from the source project, we can directly use the J-Flash tool to convert "app_signed.hex" to a binary file by saving it to a binary file named "source_xxx.bin"; or convert it with command line : **arm-none-eabi-objcopy --input-target=ihex --output-target=binary --gap-fill=0xff app_signed.hex source_xxx.bin**. Modify the source project and compile again, then get the target file: app_update.bin, rename it to target_xxx.bin, such as target_2.0.0.bin. Copy the source_ xxx.bin and target_xxx. bin to binaries/signed_images folder, then execute scripts/patch.exe, and the differential file will be automatically generated in the directory binaries/patches. Please use the differential file signed_ patch.bin as patch file.
- Transmit the patch image to secondary slot. Now we supports multiple OTA methods, such as **4G/WiFi/Bluetooth/NFC**, etc. It can also be delivered through wired methods, such as **UART, USB or SPI bus**.

(5). Restart MCU to perform the update

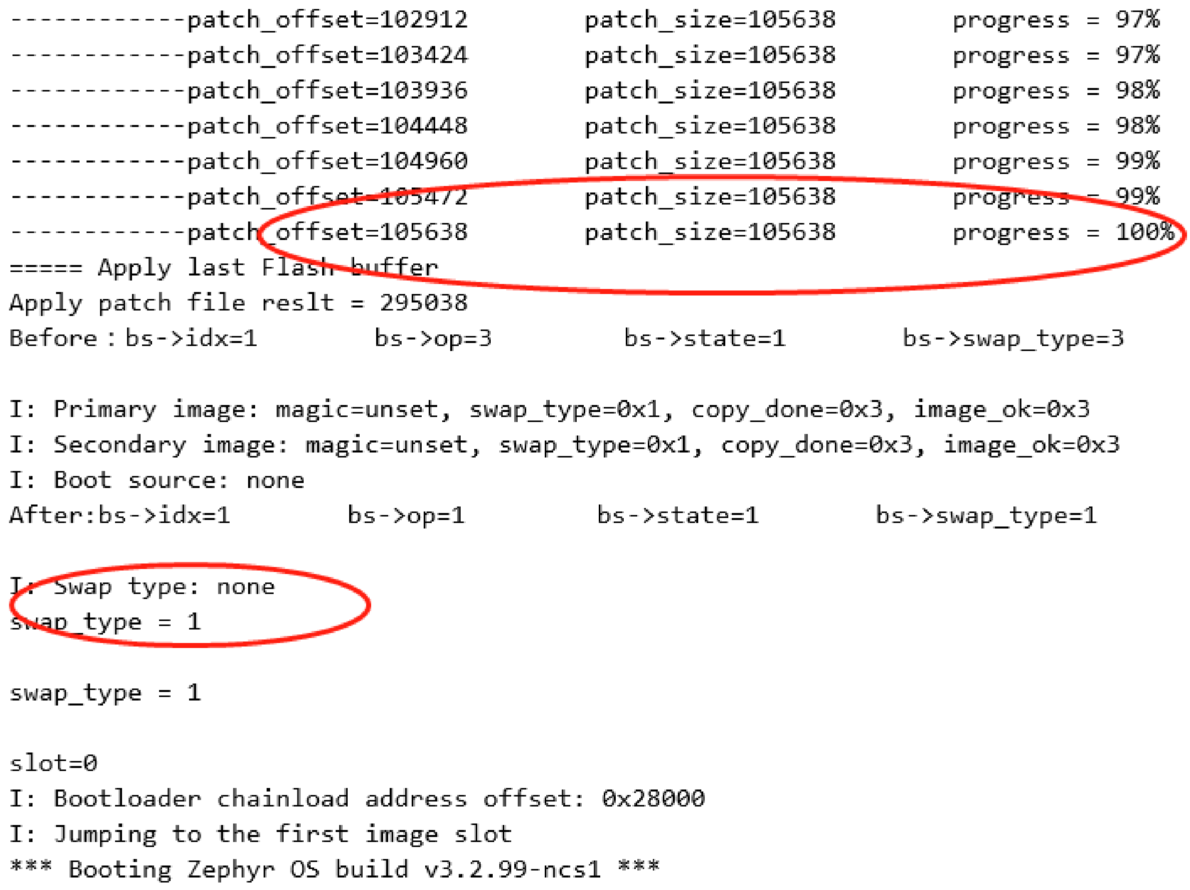
After the patch image is saved to flash, you should call the function `(boot_request_upgrade(BOOT_UPGRADE_PERMANENT)` and then restart MCU. After the MCU restarts, it will check whether the differential image in the flash area is valid, verify the signature and hash data. Differential upgrading will only be performed when all these information are matched. While applying patch.bin, the old image (source. bin) and patch image combines to generate a new image and save it in the flash primary slot area, then MCU jump to the application entry, the whole process supports power down protection.



```
I: Swap type: perm
slot=1
swap_type = 3

I: Starting swap using move algorithm.
Source hash is matched, now start delta upgrade!!!

##patch_size = 105638         opFlag = 10


Init: from_current=0x29000 to_current=0x28000 patch_current=0x90228
STATUS_ADDRESS=0xf6000 BACKUP_STATUS_ADDRESS=0xf3000 backup_addr=0x0
write_size=0
------------patch_offset=512        patch_size=105638        progress = 0%
------------patch_offset=1024        patch_size=105638        progress = 0%
------------patch_offset=1536        patch_size=105638        progress = 1%
------------patch_offset=2048        patch_size=105638        progress = 1%
------------patch_offset=2560        patch_size=105638        progress = 2%
------------patch_offset=3072        patch_size=105638        progress = 2%
------------patch_offset=3584        patch_size=105638        progress = 3%
------------patch_offset=4096        patch_size=105638        progress = 3%
------------patch_offset=4608        patch_size=105638        progress = 4%
```

(6). Checking the result of the update

you can check the output log to see if the upgrade is successful and the MCU run the target image.



```
------------patch_offset=102912        patch_size=105638        progress = 97%
------------patch_offset=103424        patch_size=105638        progress = 97%
------------patch_offset=103936        patch_size=105638        progress = 98%
------------patch_offset=104448        patch_size=105638        progress = 98%
------------patch_offset=104960        patch_size=105638        progress = 99%
------------patch_offset=105472        patch_size=105638        progress = 99%
------------patch_offset=105638        patch_size=105638        progress = 100%
===== Apply last Flash buffer
Apply patch file reslt = 295038
Before : bs->idx=1        bs->op=3        bs->state=1        bs->swap_type=3

I: Primary image: magic=unset, swap_type=0x1, copy_done=0x3, image_ok=0x3
I: Secondary image: magic=unset, swap_type=0x1, copy_done=0x3, image_ok=0x3
I: Boot source: none
After:bs->idx=1        bs->op=1        bs->state=1        bs->swap_type=1

I: Swap type: none
swap_type = 1

swap_type = 1

slot=0
I: Bootloader chainload address offset: 0x28000
I: Jumping to the first image slot
*** Booting Zephyr OS build v3.2.99-ncs1 ***
```