The example is based on the ble\_app\_beacon project from SDK 15.2.0. It is a “proof of concept” for demonstrating how the MBR can be used to safely copy a FW image from bank 1 to bank 0. Relevant code is enclosed in the DFU\_DEMO #ifdefs found in main.c.

**Prerequisites**

* nrfjprog.exe, jlink.exe, and python 2.7.x. Path to these executables must be added to the env. PATH variable.
* SDK v.15.2.0

**Testing the example**

1. Copy this directory to “nRF5\_SDK\_15.2.0\_9412b96\examples\ble\_peripheral\”
2. Build the keil project and upload the application FW and softdevice. LED\_1 should start blinking to indicate that it is advertising if it is run on a Nordic 52DK
3. Prepare new image for testing activation of new app in BANK\_1: In main.c, change the APP\_VERSION define to a higher version number, then re-compile.
4. Open a cmd window in the “scripts” folder and run the following command that will append a crc32 checksum value and upload the binary image to Bank1:
   1. py -2 create\_dfu\_image.py ../pca10040e/s112/arm5\_no\_packs/\_build/\*hex
   2. Alternatively, if only Python 2.x is in PATH: python create\_dfu\_image.py ../pca10040e/s112/arm5\_no\_packs/\_build/\*hex
5. The main app should be updated at this point. The version number of the app residing in bank 0 can be read out with nrfjprog –memrd 0x1b000.

Notes:

* The linker configurations will only work with Keil uvision (use of “at placement” attribute and linker symbols that are specific to keil)
* I chose to add a CRC32 checksum value at the end of the image, but it’s not a requirement if you have other ways to validate the image in your existing DFU implementation. Same applies to the application info structure.



