Over the Air Firmware Update

Introduction:

This document presents the instructions for the OTA (Over-the-air) updates with the Nordic devices related to the co-development project under ER 6837 using DFU (Device Firmware Update) libraries.

Project ID	ER 6837
Customer	DragonFly Energy
Contact	Andrew Warren, Bayaraa Hopkins
Description	Release of OTA for software co-development for Hub and Dongle Firmware
P/N	Related hardware: • 399-203-6121 Rev A (Dongle) • 399-203-6122 Rev A (HUB)

Contents Table

List of Figures

Revision History

Rev	Comments	ECO	Author	Date added
A	The initial release of the document	6837	E. Pedroza/A. Tamez	01/12/22

1. Steps for Over the Air firmware update

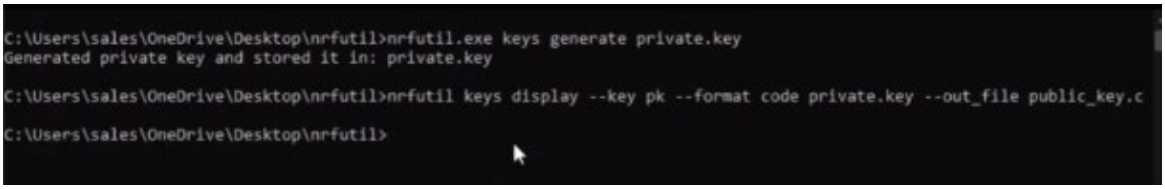
- Use "ble_peripheral" (42Q) or “zigbee”(50Q) projects and compile them in the correct directory to avoid file location errors.

- You must install the following tools to create a firmware update.
  - nRF Command Line Tools
  - nrfutil
  - nRF Connect

- Create a folder called "nrfutil" and place the nrfutil.exe app there.

- Open Command Line Interface and type the following commands inside the nrfutil folder.
  - nrfutil keys generate private.key
  - nrfutil keys display --key pk --format code private.key --out_file public_key.c

- The commands above will create a private and a public key to download the firmware using a secure bootloader. That means the firmware will be updated only if the key matches.



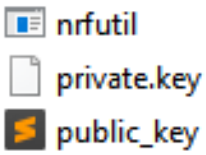


Figure 1 –Command to Generate Public Key

- For 42Q, go to SDK \examples\dfu\secure_bootloader\pca10056e_s112_ble\ses project, and for 50Q, \examples\dfu\secure_bootloader\pca10056_s140_ble\ses and compile it.

- **If you haven’t generated the public key, It should** display a “Debug public key not valid for production…” error.

  NOTE: Generated public_key.c needs to coincide with dfu_public_key.c (see step below).

- Copy the public_key.c file to SDK\examples\dfu and replace it with the dfu_public_key.c file, which is already there, but rename it as the replaced file.

- Build the project again, and it should do it correctly.

- Find the secure_bootloader_ble_s112_pca10056e.hex file already created.

- Find the SoftDevice hex file corresponding to pca10056e version.

- SDK\components\softdevice\s112\hex\s112_nrf52_7.2.0_softdevice.hex

- Using the CMD (Command Line) execute the following.
  - mergehex -m secure_bootloader_directory softdevice_directory -o nrfutil_directory\name_of_bootloader_with_sd.hex

```
$ mergehex -m C:/Users/sales/OneDrive/Desktop/DeviceDownload/nRF5SDK1500a53641a/nRF5_SDK_15.0.0_a53641a/examples/dfu/secure_bootloader/pca10056_ble/ses/Output/Release/Exe/secure_bootloader_ble_s140_pca10056.hex C:/Users/sales/OneDrive/Desktop/DeviceDownload/nRF5SDK1500a53641a/nRF5_SDK_15.0.0_a53641a/components/softdevice/s140/hex/s140_nrf52_6.0.0_softdevice.hex -o C:/Users/sales/OneDrive/Desktop/nrfutil/bootloader_dfu.hex
Parsing input hex files.
Merging files.
Storing merged file.
```

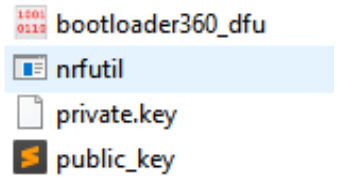


Figure 2 –Merge of bootloader and softdevice

- Download the merged hex file (bootloader with SoftDevice) to the HUB or Dongle board using the command tool as follows.

  - nrfjprog -f nrf52 --ereaseall
  - nrfjprog -f nrf52 --program bootloader360_dfu.hex --sectorerase -r

  NOTE: Make sure pca10056e.h and sdk_config,h files have the right pinout configuration

- Compile the main project, obtain the generated hex file, and copy it to the nrfutil folder.

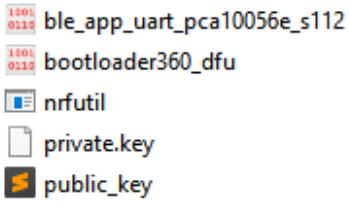


Figure 3 –Files to generate firmware update package

Now, it is necessary to create a ZIP package using the application hex file that will be used to download the firmware through BLE using a phone application.

- Using the command tool and locating the nrfutil folder, execute the following command.

  - nrfutil pkg generate --hw-version 52 --application-version 1 --application application_file.hex --sd-req 0x103 --key-file private.key app_dfu_package.zip

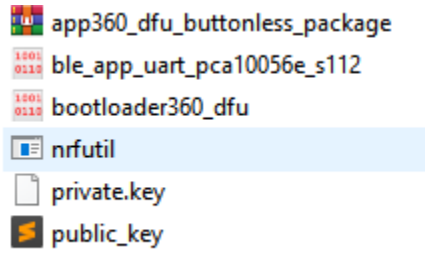


Figure 4 –Files after generated package

Once the zip package is generated, send it through the Cloud to download it to a cellphone.
Nordic implements its applications to download the firmware to the board.

nRF Connect is the official Nordic application to scanning and testing BLE communication.
nRF Connect Device Manager (DFU) is a testing application that uses free libraries to create a custom application based on the DFU Nordic application to download the ZIP-generated packages to update the firmware.

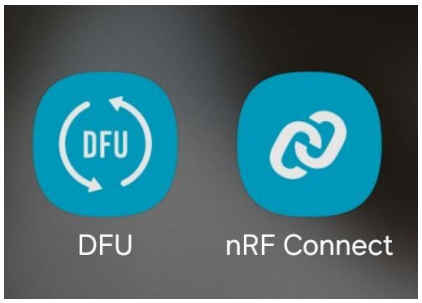


Figure 5 –Nodic Apps for DFU

Here is the GitHub link to start using the libraries for Android (iOS available, too), including documentation for using a custom app (e.g., Dragon Fly app).

https://github.com/NordicSemiconductor/Android-nRF-Connect-Device-Manager

- Once the board has the bootloader installed, open the application (DFU in this example) and select the ZIP file.

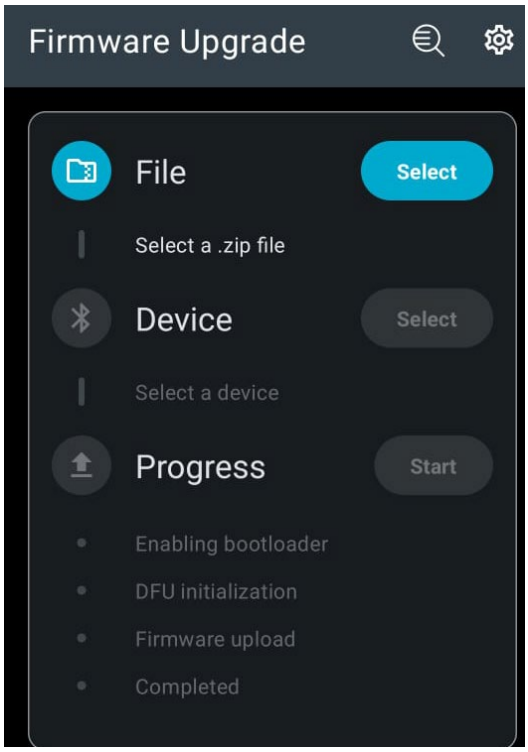


Figure 6 –Nordic DFU main screen

- Select the BLE device called in this example “DfuTarg”.
- In the next step, the application will start to download the firmware, showing the percentage of download.
- When the download is complete, the board will run the firmware update.
- You will notice the BLE device change to the main project device name.
- You can reenter the bootloader, sending a command programmed in the main project (service initialized in *services_init()* function.
- You can use the nRF connect application to connect to the BLE device and send the bootloader entering command.
- If the bootloader is running, a new zip-generated file can be downloaded.