
Hands-on Lab 2: Loading Custom Firmware via DFU Boot

1. Introduction

In addition to installing firmware to the HackRF, you can boot firmware directly from USB. This is useful when you have a non-booting HackRF or you simply want to test out a change to firmware without replacing the firmware on the SPI flash. In this lab, you will compile the “blinky” firmware and boot the HackRF to the new firmware. Finally you will reboot the HackRF and boot back into normal firmware.

2. Building the blinky firmware

We will repeat the process for building firmware. But instead of building the typical `hackrf_usb`, we will build `blinky`. Blinky doesn’t actually do anything other than blink 3 of the LEDs on the HackRF One.

Build firmware using similar process to lab 1.

```
$ cd ~/hackrf/firmware/blinky
$ mkdir build
$ cd build
$ cmake ../ -DBOARD=HACKRF_ONE
$ cd ..
$ make
```

Now we should see a firmware files

```
$ cd ~/hackrf/firmware/blinky/blinky
$ ls

blinky.bin
blinky_dfu.elf
blinky_m0.elf
```

```
blinky.dfu  
blinky.elf
```

In the case of DFU booting the HackRF, we are interested in the `blinky.dfu` file.

3. Boot the HackRF into DFU Boot mode

Now that we have `blinky.dfu`, we need to boot the hackRF into DFU Boot mode. To do this:

1. Unplug the HackRF from the USB cable
2. Press and hold the DFU button
3. Reconnect the USB cable to the HackRF
4. Release the DFU button when the 3v3 LED illuminates

4. Upload the DFU file to the HackRF

The HackRF is now awaiting us to load firmware via USB. To do this, enter the following command:

```
$ dfu-util -device 1fc9:00c -alt 0 -download blinky.dfu
```

If successful, the USB, RX, and TX LEDs should now be blinking on the HackRF.

To return the HackRF to normal operations, just hit the reset button.

Side Note: Some VM Systems will prompt you to pick where you want the HackRF to be connected. If it does, just answer “Linux” or “Virtual Machine”

This should re-enumerate it as a usb device. Verify your HackRF is back to the firmware you built in Lab 1 by verifying with:

```
$ hackrf_info
```