# RecoilGun Firmware Upgrade Guide

|  |  |  |  |
| --- | --- | --- | --- |
| ***Version 1.0*** | ***Matteo Scordino*** | ***7 Mar 2016*** | ***Initial Version*** |

Table of Contents

[RecoilGun Firmware Upgrade Guide 1](#_gjdgxs)

[Introduction 2](#_30j0zll)

[Getting started 3](#_1fob9te)

[DFU mode 3](#_3znysh7)

[Entering DFU Mode 3](#_2et92p0)

[Implementing the DFU protocol 3](#_tyjcwt)

[What if something goes wrong 3](#_3dy6vkm)

[Configuration parameters 5](#_4d34og8)

[Weapon Definition 5](#_2s8eyo1)

[ID 5](#_17dp8vu)

[Trigger Mode 5](#_3rdcrjn)

[Rate of Fire 6](#_26in1rg)

[PowerIR1 (long-range) 6](#_lnxbz9)

[PowerIR2 (short-range) 6](#_35nkun2)

[PowerLED1 (muzzle) 6](#_1ksv4uv)

[PowerLED2 (power LED) 6](#_44sinio)

[PowerMotor 6](#_2jxsxqh)

[FlashLED1 (muzzle) 6](#_z337ya)

[FlashLED2 7](#_3j2qqm3)

[Flash Parameter 1 7](#_1y810tw)

[Flash Parameter2 7](#_4i7ojhp)

[Global parameters 8](#_2xcytpi)

[Shot Config 8](#_1ci93xb)

[Trigger Mode Override 8](#_3whwml4)

# Introduction

The RecoilGun firmware system has been designed to allow firmware upgrades. This happens over-the-air, using the DFU protocol by Nordic Semiconductors.

# Getting started

## DFU mode

RecoilGun does not allow firmware upgrades during normal operation of the game. To upgrade the firmware, RecoilGun must be set to DFU mode. In DFU mode, the normal firmware is not running: rather, a bootloader firmware implementing the DFU BLE service executes and allows firmware upgrades.

## Entering DFU Mode

RecoilGun enters DFU mode in 2 cases: intentionally (when a specific command is received) or unintentionally (as a fallback measure if the firmware is found to be corrupted or missing altogether)

To intentionally enter DFU mode, the application needs to be connected to RecoilGun and send Command 0x0100 (Reboot to bootloader). Refer to the DFU Protocol, revision 2.1 or later for the format of the Command characteristic in the Recoil BLE service.

After receiving the 0x0100 command, RecoilGun will:

1. Disconnect the BLE radio
2. Reboot
3. Start the bootloader
4. Turn ON the Muzzle LED
5. Turn OFF the Power LED
6. Wait for a connection and subsequent DFU commands

## Implementing the DFU protocol

The DFU protocol is well documented by Nordic Semiconductors.

A few useful links are:

<https://devzone.nordicsemi.com/blogs/683/a-quick-presentation-on-how-nrf51-dfu-works/>

<https://devzone.nordicsemi.com/tutorials/9/setting-up-device-firmware-updatedfu-on-the-nrf51-/>

<https://devzone.nordicsemi.com/blogs/685/common-faq-on-dfu/>

Especially useful are the reference implementations for iOS, Android and Windows, which are all available on the app stores (nRFToolbox for iOS, nRF Connect for Android, Master Control Panel for Windows

Source is available on github at https://github.com/NordicSemiconductor

## What if something goes wrong

The bootloader applies a sequence of checks to the received image before overwriting the previous one, including:

1. The image fits in flash
2. The image metadata is correct and specifies the correct hardware, softdevice and bootloader revisions
3. The image is signed with Hotgen’s private key and the signature is valid
4. The CRC matches the received data

If any of these checks fails, the bootloader will restart the old app and discard the received one.

In case the old app was corrupted, the device should never become unrecoverable anyway: the bootloader will keep refusing to boot the corrupted old app and try to wait for a new image download.