

Toolchain Installation

GCC compiler:

You need the right GCC compiler, which is `gcc-arm-none-eabi` .

For MacOS Users:

Usage of the [Homebrew package manager](#) for MacOS is recommended. The installation of Homebrew is quick and easy: [installation instructions](#).

After installing Homebrew, copy these commands to your shell:

```
brew update
brew install git bash-completion gcc-arm-none-eabi
```

JLink driver:

You also need the right `JLinkexe` for flash MindTag board.

For both Ubuntu and MacOS users:

Download the J-Link software from the Segger website and install it according to their instructions.

[Download pages](#)

nRF Tools:

You need some nrftools in the process of building and flash the Dronetag board, such as nrfutil. Nrfutil is a Python package and command-line utility that supports Device Firmware Updates (DFU) and cryptographic functionality, you can get detail from [here](#)

Installing from PyPI:

```
pip install nrfutil
```

Installing from sources:

```
pip install -U setuptools
pip install pyinstaller
git clone https://github.com/NordicSemiconductor/pc-nrfutil.git
cd pc-nrfutil
pip install -r requirements.txt
python setup.py install
pyinstaller nrfutil.spec
```

[More detail about nrfutil install](#)

nRF5 SDK

The SDK14.2 and softdevice5.0.0 is necessary for you when you build the bootloader、app and flash them to the board. You can get the SDK14.2 and softdevice5.0.0 from [here](#)

Building the project

1、Generate a private key

```
nrfutil keys generate private.pem
```

2、Build the Dronetag bootloader

```
git clone https://github.com/airmind/DroneTag-bootloader.git
cd DroneTag-bootloader
```

after these, you need modified the variable *SDKROOT* point to your real path of SDK in Makefile at *DroneTag-bootloader/DroneTagbootloader_ble/pca10040/armgcc/*

then

```
make
```

3、Build the BLE app Firmware

```
git clone https://github.com/airmind/DroneTag.git
cd DroneTag/DroneTag/pca10040/armgcc/
make
```

4、Gnererate a bootloader settings

You can the follow commands to generate a bootlader with application version 1, bootloader version 1 and bootloader setting version 1. You also need replace the [xxxName] with real filename.

```
nrfutil settings generate --family NRF52 --application [appName.hex] --application
-version 1 --bootloader-version 1 --bl-settings-version 1 [settingsName.hex]
```

5、Generate a app Firmware package

You can use the follow commands to generate a package with application version 1 and use the private key key.pem.

```
nrfutil pkg generate --hw-version 52 --sd-req 0x80 --application-version 1 --appli
cation [appName.hex] --key-file private.pem [app_dfu_package.zip]
```

Flash the DroneTag board

1、Flash the Softdevice

Connect the JLink cable to the MindTag like this:

After connect the JLink correctly, use these command in your shell, and the command with

`J-Link>` begining is run in `JLinkExe` .

```
cd /Application/SEGGER/JLink
./JLinkExe
- J-Link> device nrf52832
- J-Link> if swd
- J-Link> speed 4000
- J-Link> connect
- J-Link> r
- J-Link> h
- J-Link> loadfile [softdeviceName.hex]
```

2、Flash the bootloader

```
- J-Link> r
- J-Link> h
- J-Link> loadfile [bootloaderName.hex]
```

3、Flash the ble app

```
- J-Link> r
- J-Link> h
- J-Link> loadfile [appName.hex]
```

4、Flash the bootloader settings

```
- J-Link> r
- J-Link> h
- J-Link> loadfile [bootloader-settings.hex]
- J-Link> r
- J-Link> g
```

you can see the application running, then

```
- J-Link> q
```

for quit the JLinkExe.

Update the DroneTag board with BLE

To do this, you need a ios device such a iphone or ipad, and install nrfconnect on it, transfer the firmware zip package to you ios device through itunes.

As the DroneTag is work correctly, open you nrfconnect and choose the dronetag device to build the blue tooth connection, then choose the service of DFU, select the package you transfer in, and tap start to begin the BLE DFU update.