Laird™
CONNECTIVITY

Quick Start Guide

451-00004

BL654 USB Dongle - Nordic/Zephyr

Revision History

Version	Date	Notes	Contributors	Approver
1.0	10 July 2019	Initial Release	Jamie Mccrae	Jonathan Kaye

TABLE OF CONTENTS

1 INTRODUCTION

1.1 Scope

The purpose of this document is to give an overview for programming application code to the Laird Connectivity 451-00004 – BL654 based USB dongle supporting Nordic / Zephyr based application deployment.

Note: Laird Connectivity also makes a variant of this product which supports *smart*BASIC development (Laird part # 451-00003). The variants have *different hardware* and are dedicated to their specific development environments.

1.2 Background

The Laird Connectivity 451-00004 USB dongle is a self-contained Bluetooth product which contains a Bluetooth v5 radio and Cortex M4 processor using Nordic's nRF52840 silicon. It comes pre-programmed with the Nordic open bootloader to allow for flashing applications to the dongle. The USB pins are directly connected to the nRF52840, unlike the 451-00003 on which the USB pins are connected to an FTDI USB to UART transceiver. This allows for the use of custom USB profiles. It comes in a sealed plastic case and is production-ready. Please note that this dongle does **not** run *smart*BASIC, use the 451-00003 dongle for *smart*BASIC.

2 HARDWARE DESCRIPTION

The Laird Connectivity BL654 USB dongle contains an nRF52840 Bluetooth SoC and the following supporting components:

- 32-kHz crystal (20 ppm accuracy)
- Blue LED connected to P0.13 (active high)
- USB data lines connected directly to nRF52840
- GPIO voltage set to 3.3v by default bootloader
- Readback and CPU debug protection enabled by default bootloader

The bootloader supports updating the following:

- The Nordic soft-device (optional)
- User application
- Bootloader (requires a signed update)

The bootloader on the dongle can be updated to the Nordic secure bootloader which forces signed application updates and prevents loading of unauthorised code to the product. Laird will generate subsequent documentation to detail how to enable this functionality.

3 PROGRAMMING SETUP

This section describes how to program the dongle. There is a simple interactive GUI-based programming system and a more complex CLI programming system. The CLI system is required to generate DFU packages which can be use with the secure bootloader. If the bootloader remains unchanged with the open bootloader, then signing firmware upgrades is not required.

3.1 GUI Setup

Nordic's nRF connect application (available for Windows/Linux/Mac) can be used to interact with the module via the use of an interactive graphical user interface. This software can be downloaded from the Nordic nRF Connect for Desktop page.

To set up the GUI, follow these steps:

1. Install and open the application.
2. Click the Add/Remove apps tab.

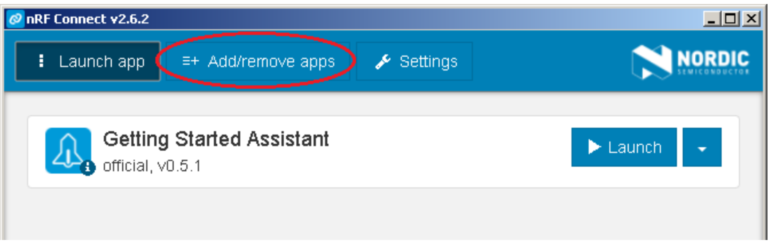
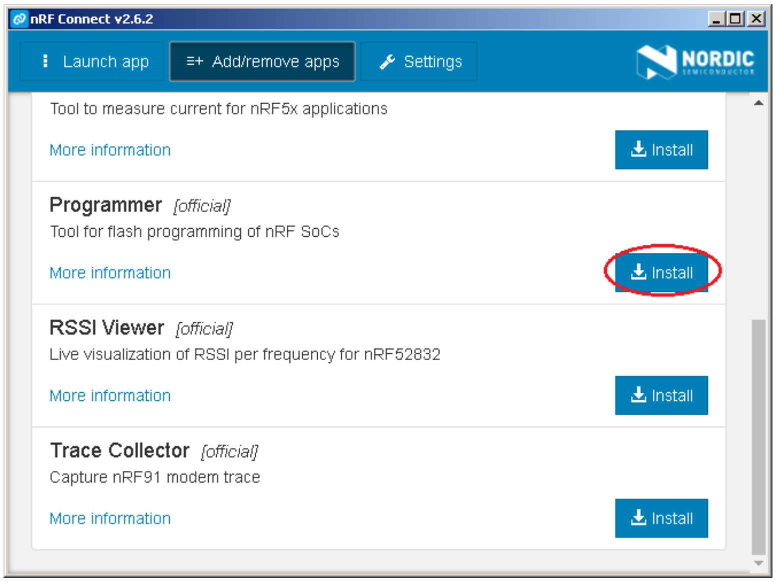


Figure 1: nRF Connect GUI

3. Scroll down and click the Programmer Install button.



Tool to measure current for nRF5x applications

Tool for flash programming of nRF SoCs

Live visualization of RSSI per frequency for nRF52832

Capture nRF91 modem trace

Figure 2: Add/remove apps Install button

4. Once installed, return to the Launch app tab and click the Programmer Launch button.

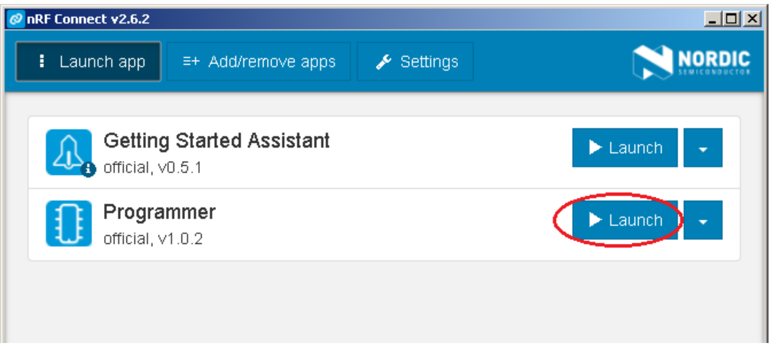


Figure 3: Launch app tab Programmer Launch button

5. Plug the USB dongle into the computer and check that the LED is getting brighter and darker and that it is enumerating as a USB device. If it fails to do this, use a thin object to press the button on the dongle through the hole in the case (whilst keeping the USB dongle plugged into the computer) to restart in bootloader mode.

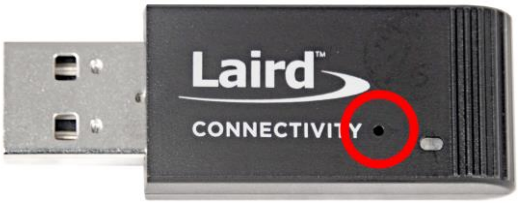


Figure 4: Dongle reset button

6. From the Select device drop-down menu, select the applicable USB dongle.

Note: If there is more than one device, disconnect the other device(s) to get and select the ID of the correct dongle.

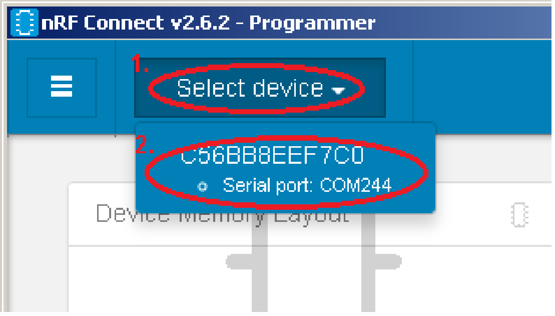
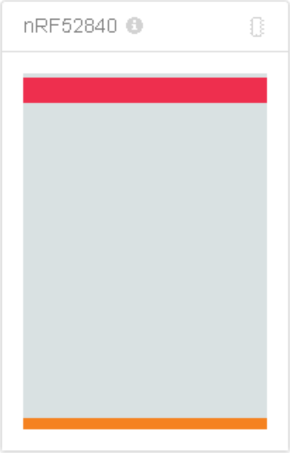


Figure 5: Select device drop-down menu

Once connected, the memory map updates to show which parts of the memory are used by what sections. Hover the mouse over the nRF52840 text to view further details.



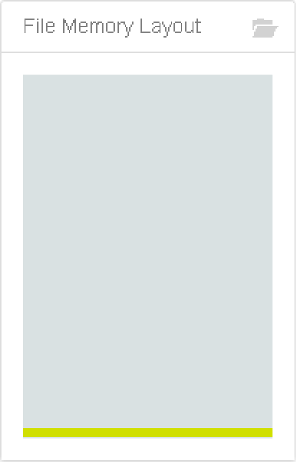
The following is the colour code for different parts of the memory space:

- 🟥 Bootloader
- 🟨 Application
- 🟦 Soft-device
- 🟧 MBR

Figure 6: nRF52840 memory space

To program an application, your build tool must generate a hex file. This hex file only includes the application code or application and soft-device code (if using the Nordic SDK); it must not include a bootloader or an MBR update.

7. Drag the hex file onto the window or click **Add HEX file** > **Browse** and select the hex file. The file memory layout updates to show the sections in the file.



8. Click **Write** to begin the programming procedure.

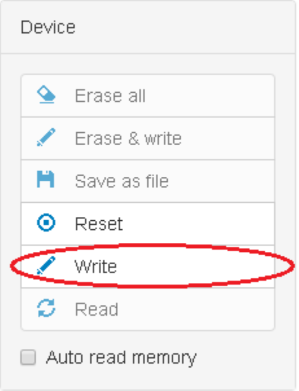


Figure 7: Click Write

Once programming is complete, the module automatically restarts and begins execution of the code.

Note: To return to bootloader mode, press the button through the hole of the dongle at any time.

3.2 CLI Setup

Nordic's command line utility *nrfutil* (available for Windows/Linux/Mac) can be used to interact with the module and to create firmware upgrade files from a terminal or command prompt. For windows, a build is available from the nrfutil Github Releases page; for Linux/Mac, follow the instructions listed on the front page of the nrfutil Github Repository.

To set up the CLI, follow these steps:

1. Use the applicable link to install the Nordic command line utility (*nrfutil*) and add it to the path.
2. Locate the folder with the applicable output hex file from which to create a firmware upgrade.
3. Open a command prompt window or terminal in the directory with the hex file.
4. Use the following command to create an update zip file using a single hex file that does not use the soft-device:

```
nrfutil pkg generate --hw-version 52 --application app.hex --application-version 1 --sd-req 0x00 out.zip
```

If the soft-device is used, use the following command:

```
nrfutil pkg generate --hw-version 52 --application app.hex --application-version 1 --softdevice softdevice.hex --sd-id 0xB6 --sd-req 0xB6 out.zip
```

Notes: Soft-device hex files include the Nordic MBR and soft-device.

The Nordic MBR is located at 0x0 – 0x1000 and cannot be updated.
The soft-device starts at 0x1000 and is variable in length.
The 6.x series soft-device end at address 0x26000.
The MBR must be removed from the hex files before an update file is generated.

The soft-device ID is dependent upon the version of the soft-device you are using. If your application does not use the soft-device, set it to 0 or 0x00. If it does use the soft-device, you must use the correct IDs which can be found on the nrfutil github page.

The following are the current soft-device versions and corresponding IDs:

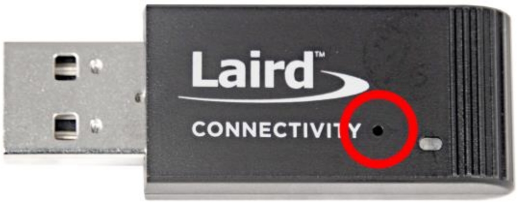
- 6.0.0 – 0xA9
- 6.1.0 – 0xAE
- 6.1.1 – 0xB6

This creates a new DFU package at **out.zip** which can be programming using the nRF connect GUI or nrfutil command line tool. Replace the *app.hex* with the hex file of the application you want programmed (and *softdevice.hex* with the soft-device file minus the MBR, if used).

This can then be program using the following:

```
nrfutil dfu usb_serial -pkg out.zip -p COMx
```

Where **COMx** is the serial port of the USB dongle (which will be a **/dev/ttyX** path on Linux/Mac), ensure that the dongle is in bootloader mode before running this command (the LED should be going from bright to dim when in bootloader mode). If not, enter bootloader mode using a thin object to press the button on the dongle through the hole in the case (whilst keeping the USB dongle plugged into the computer).



Once the module is programmed, it will begin execution of the code.