Adding a Bluetooth Low Energy (BLE) Data Link and Over-the-Air (OTA) Firmware Update Utility to the STM32L4 and Teensy 3.X Families of Microcontroller Development Boards

By Greg Tomasch

Background

Tlera Corporation (https://www.tindie.com/stores/TleraCorp/) offers a family STM32L4-based microcontroller (MCU) development boards suitable for developing a wide range of IoT devices. These MCU's are compact, powerful and have very rich I/O capabilities. They also support very low power state-preserving sleep modes, which makes them well-suited for miniaturized, long-term remote sensing. Unfortunately, the STM32L4 MCU family does not have a native wireless capability. Communicating results in real time to a remote PC or smart device is essential for remote sensing applications. It is also highly attractive to update the MCU firmware of a remotely deployed STM32L4 device using some form of wireless communication link. To address these needs, Tlera Corporation now offers a BLE add-on board and firmware to bring low-power remote connectivity to the STM32L4 "Dragonfly" and "Butterfly" development boards (https://www.tindie.com/products/onehorse/nrf52-add-on-for-butterfly-and-teensy/). It uses Nordic Semiconductor's popular nRF52832 SoC which has embedded BLE capability. This add-on board is sized to plug directly onto the Dragonfly and Butterfly development boards (https://www.tindie.com/products/TleraCorp/dragonfly-stm32l47696-development-board/) (https://www.tindie.com/products/TleraCorp/butterfly-stm32l433-development-board/) as well as the popular "Teensy" 3.X family of MCU products.

The BLE add-on uses the Nordic UART Service (NUS) for data exchange and can be programmed in either the BLE Central or BLE Peripheral roles. Communication between the MCU development board and the nRF52832 is accomplished over simple UART serial. When the nRF52832 receives data from the MCU over UART, it simply sends the data to the BLE device to which it is paired. Similarly, data received by the nRF52832 from a paired NUS device is sent to the MCU over UART.

There are a number of NUS-compatible applications for smart phones and tablets that can be used to view data sent from an nRF52832 peripheral device. However, it has proven difficult to form a BLE data link with a PC. A second Dragonfly/Butterfly/Teensy MCU board with an nRF52832 add-on board programmed for BLE Central role solves this problem. The Central role nRF52832 board pairs with the remote Peripheral role board, making a point-to-point BLE/UART bridge that exchanges data with a PC over a standard USB serial connection.

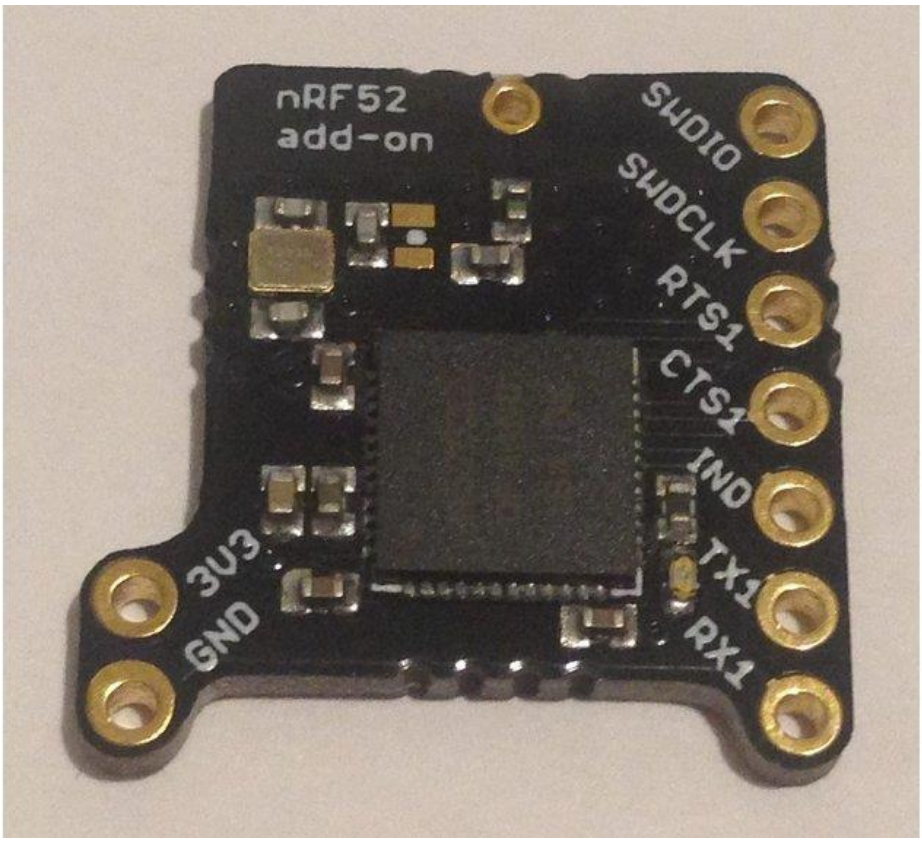
This is an important step forward in IoT device capability. Users can leverage the superior performance of the STM32L4 and Teensy 3.X products with their mature Arduino cores and take advantage of low energy wireless connectivity. A remote sensing device can now directly update a PC serial terminal application or spreadsheet with a serial plug-in. The BLE/UART link can

also support OTA firmware updates for the remote STM32L4 device. In principle, OTA firmware updates could also be done for the Teensy 3.X family of products if a suitable UART-based firmware update capability were to be developed.

The remainder of this document provides the information necessary to install, program and use the nRF52832 add-on board in conjunction with the STM32L4 and Teensy3.X MCU boards. Step-by-step instructions will show how to flash firmware to the nRF52832 add-on boards, use the Arduino-based example sketches to transfer data to a PC from a remote sensing board and update firmware over-the-air to remote STM32L4-based devices.

Teaming up the nRF52832 with the Dragonfly/Butterfly/Teensy 3.X

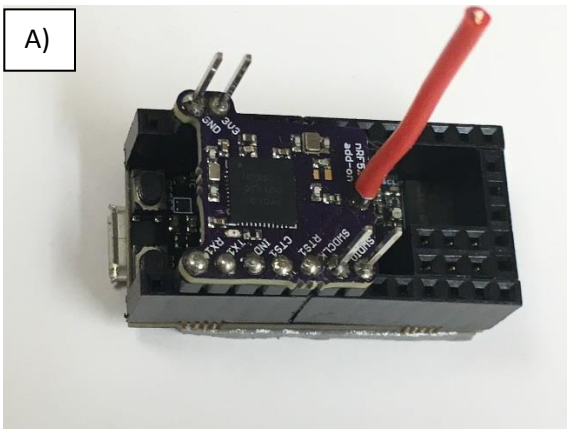
Here is a close-up photo of the nRF52832 add-on board:



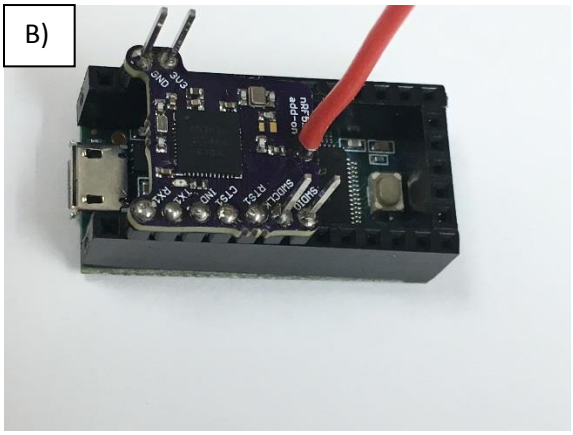
The form factor is designed to fit directly onto the Dragonfly, Butterfly and Teensy 3.X development boards using machine pin headers. It should also be noted that the small connection pad in the center of the add-on board's top edge is for the BLE radio antenna. Typically, the antenna is simply a 1.2" insulated copper wire segment. Connection to the "3V3", "GND", "SWDCLK" and "SWDIO" nodes is necessary to update the add-on board's firmware. Using extended machine pin headers on these nodes is convenient but is not required.

These pictures show the nRF52832 add-on board installed on the A) Dragonfly/Butterfly and B) Teensy 3.2 MCU development boards:

A)



B)



Please verify that the nRF52832 add-on board is installed onto your MCU development board correctly as shown in the photos above.

This table explains the add-on board connections to the MCU board:

Pin Label	Function	Comments
3V3	3.3V VDD	Can be powered by 3V3 LDO or an MCU DO set high
GND	Ground	Can be grounded by MCU GND or an MCU DO set low
RX1	UART Rx	Connects to MCU UART Rx pin
TX1	UART Tx	Connects to MCU UART Tx pin
IND	Indicator Pin	Indicator pin; pin goes low when add-on is BLE paired
CTS1	UART Clear to Send	UART hardware flow control; connects to MCU CTS pin
RTS1	UART Request to Send	UART hardware flow control; connects to MCU RTS pin
SWDCLK	SW Update Clock	Clock connection for nRF52832 firmware programming
SWDIO	SW Update Data	I/O connection for nRF52832 firmware programming

Powering the add-on board is accomplished in two different ways. For the Butterfly and Teensy 3.X, the 3V3 and GND connections are straightforward. For the Dragonfly, the add-on is powered by two GPIO pins. Dragonfly pin 40 is set as a low digital output and pin 39 as a high digital output. The Dragonfly digital outputs are capable of supplying/sinking 20mA, more than enough current to run the nRF52832. These differences are handled in the Arduino sketches for each specific type of MCU development board.

The current implementation of the nRF52832 add-on board is not set up for UART hardware flow control. The nRF52832 firmware can easily be modified to support CTS/RTS flow control but flow control is not yet fully implemented in the STM32L4 Arduino core. When hardware flow control is fully functional, the firmware and documentation will be updated to allow users to take full advantage of this feature.

Before proceeding any further, it is necessary to have the correct IDE software installed on your PC. For Teensy 3.X, install the appropriate version of Arduino to support the latest version of Teensyduino. For the STM32L4, install Arduino 1.6.13 or later and install the STM32L4 core and libraries (https://github.com/GrumpyOldPizza/arduino-STM32L4).

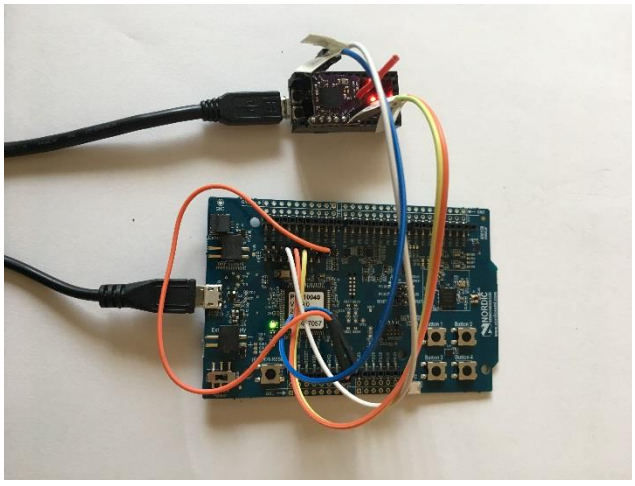
Flashing the nRF52832 Add-on Board with the Desired Firmware

The nRF52832 add-on board can be purchased pre-flashed from Tlera Corp. with either BLE Central role or Peripheral role firmware, ready to use out of the package. If you just want to use the pre-flashed firmware, feel free to skip ahead to the next section. However, some users may wish to reprogram their boards for the opposite BLE role or develop their own custom nRF52832 firmware. This section describes how to flash the add-on board as a Central role device using Nordic Semiconductor's Development Kit (DK) board and "nRFgo Studio" software utility.

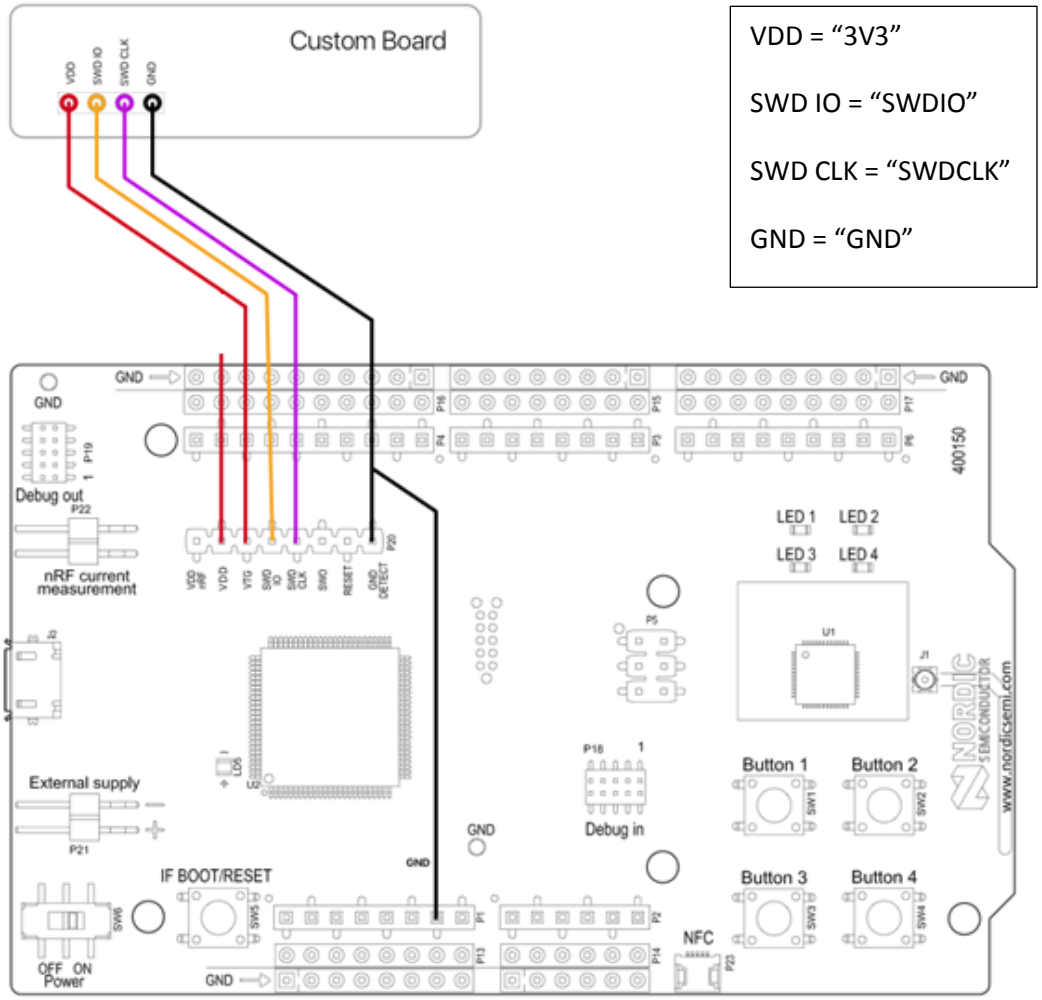
In order to program the nRF52832 add-on board with pre-compiled firmware, the Nordic nRF52 pca10040 DK board is probably the best option. Information and additional links can be found at (https://www.nordicsemi.com/eng/Products/Bluetooth-low-energy/nRF52-DK). The "Downloads" tab in the previous link gives access to the nRFgo Studio install package and can be found in the "PC software" section. Download and install the appropriate version for your particular Windows OS.

Before proceeding, it is important to ensure that the attached microcontroller board is running an Arduino sketch that properly supplies power to the add-on board and does not interfere with any of the nRF52832's I/O during the programming process. The repository includes several simple Arduino "Blink" sketches for the Dragonfly, Butterfly and Teensy 3.X boards that meet these requirements. Before proceeding further, upload the appropriate "Blink" sketch for your MCU board and verify board's LED(s) indeed blink.

This photo shows the setup for flashing firmware on the nRF52832 add-on board:



Details of the nRF52832 add-on board connections to the pca10040 DK board are shown below:



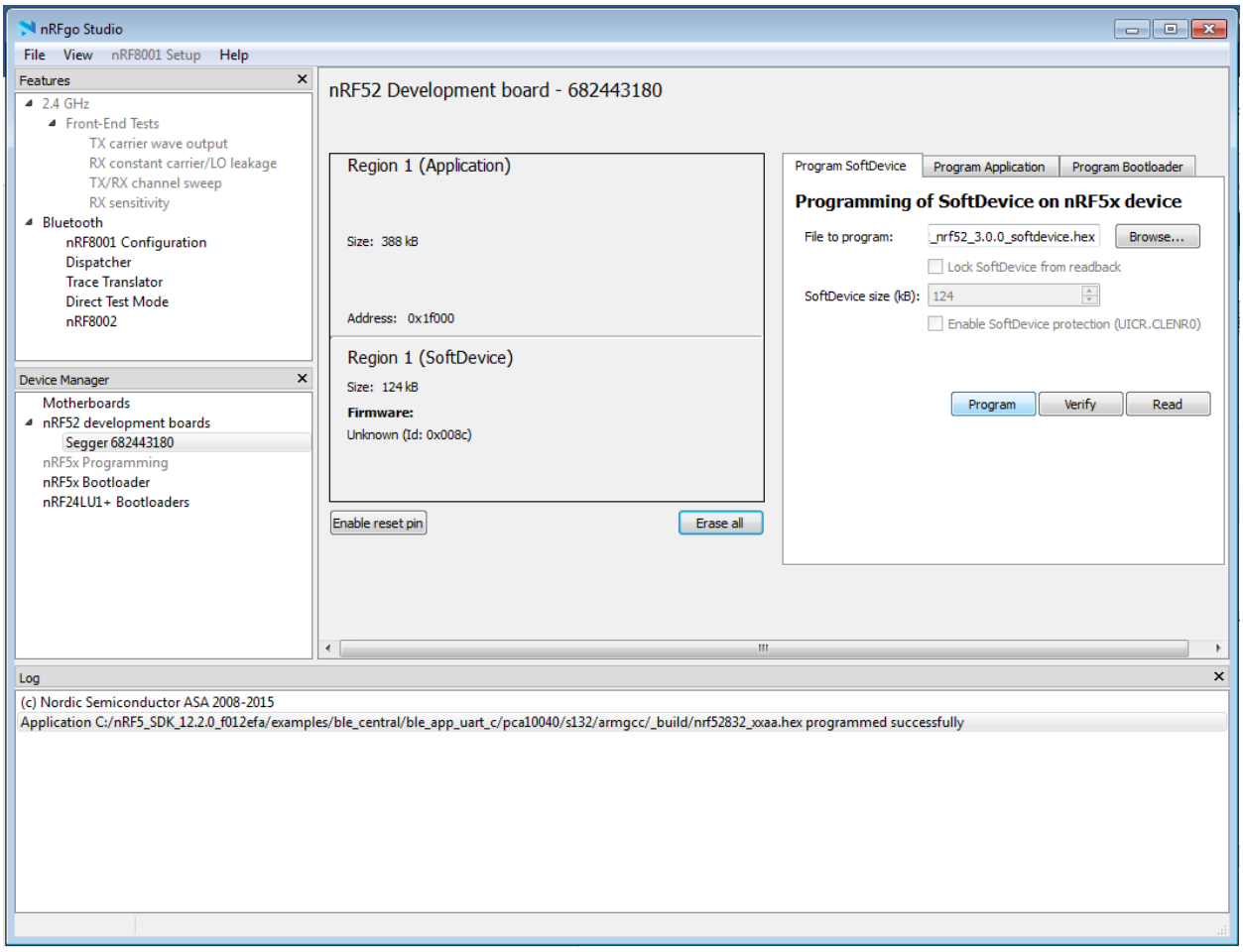
The pin designations in quotes refer to those stenciled on the nRF52832 add-on board.

It is important to power up the nRF52832 add-on board BEFORE powering on the nRF52-DK board so that J-Link programmer will program the add-on board and not the nRF52832 chip residing on the DK board. Double-check all of the connections and power up the MCU and nRF52832 add-on board pair by plugging into an unused USB port. Next, plug the nRF52-DK board into another unused USB port and turn it on. Let the various drivers load; when finished, a virtual drive window will open for the Segger J-Link programmer.

The following instructions show how to program the pre-compiled BLE Central role firmware and SoftDevice to the nRF52832 add-on board as an example. Before proceeding, make sure that the desired firmware “.hex” files are unzipped from the repository and are available on your hard drive. It should be noted that the “SoftDevice” (Real-time BLE stack drivers and management software) version must match the SDK version used to build the nRF52832 application firmware.

If you are building your own firmware using the Nordic SDK, be sure you are using the correct SoftDevice version for your SDK version… Next ensure that both the MCU/add-on board pair and the DK board are powered up in the proper order.

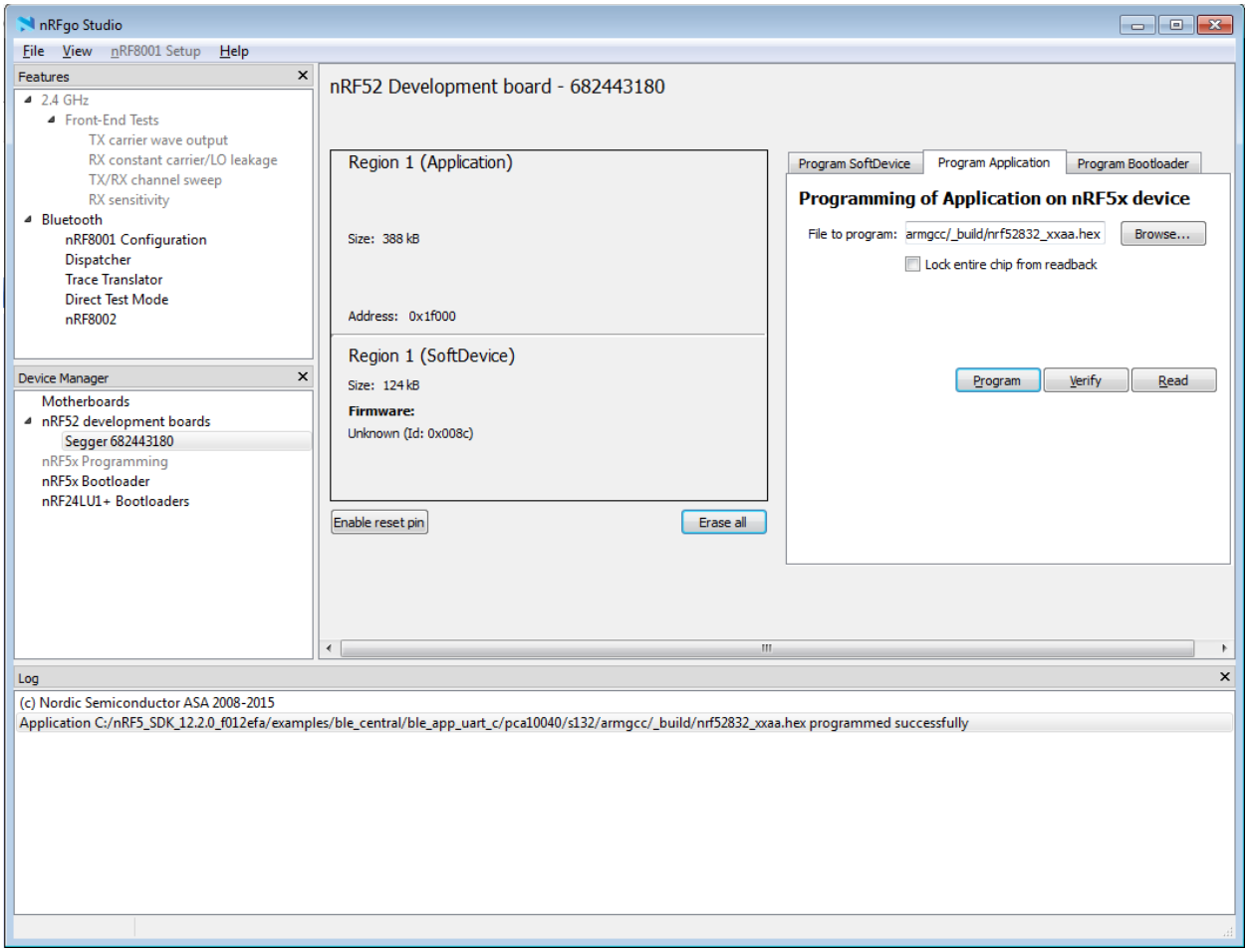
Open nRFgo Studio and look at the main view of the window:



Select the “Segger XXXXXXXXX” entry under “nRF52 development boards”. Unless you are completely sure of the prior programming state of your add-on board, it is recommended that you erase the nRF52’s flash memory. Simply click on the “Erase all” button and wait for a log message confirming successful completion of the erasure.

The SoftDevice needs to be programmed first. Click on the “Program SoftDevice” tab. You should see screen shown above. Click the “Browse…” button and navigate to the “s132_nrf5_3.0.0_softdevice.hex” executable file. Click the “Program” button and nRFgo Studio will flash the SoftDevice to the nRF52832 add-on board. Next, click on the “Program Application” tab, click the “Browse…” button and navigate to the “Tlera_nrf52832_Add_On_BLE_UART_Central.hex” executable file.

You should see:



Click the “Program” button again to flash the BLE Central Role UART application to nRF52832 add-on board.

The nRF52832 add-on board is now properly flashed to act as a Central role NUS device. All boards can be powered down and the nRF52832 add-on board may be disconnected from the nRF52-DK board. Now the MCU/nRF52832 add-on board pair can be used as a BLE Central role link to your PC.

If you wish to program the nRF52832 add-on board as a peripheral role device, the procedure is identical to that just described. The only difference is that “Tlera_nrf52832_Add_On_BLE_UART_Peripheral.hex” will be the target executable file in the “Program Application” tab of nRFgo Studio when you flash the application code.

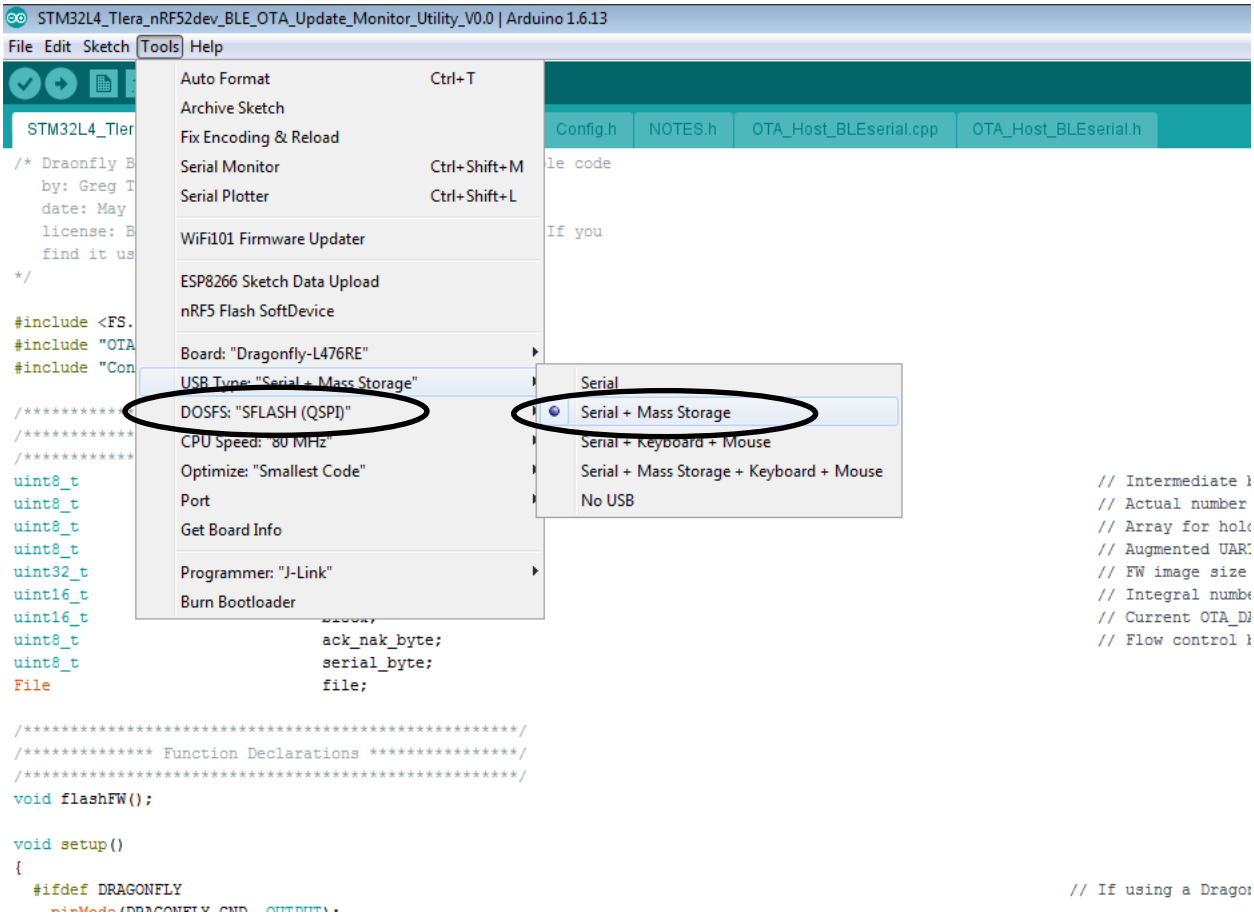
The following sections will show you how to take advantage of the nRF52832 add-on boards to stream data to your PC and do OTA firmware updates to remote MCU boards as well as use Arduino to program your MCU boards.

Central Role BLE Data Monitor and OTA Firmware Update Utility

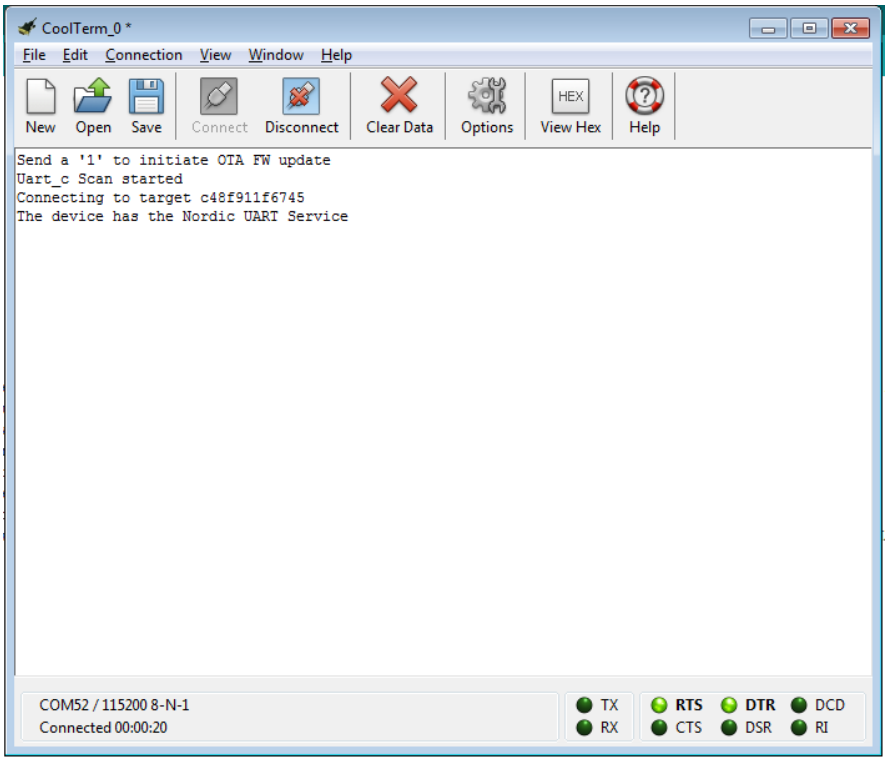
(I) Viewing data sent from a remote NUS device

This section will describe how to use the Central role nRF52832 add-on board with a Dragonfly STM32L4 development board to make a data transfer utility that connects directly to your PC. This utility actually performs two functions: 1) It receives data updates from any NUS-capable BLE peripheral device and passes it through the Dragonfly's USB serial port to a host PC and 2) It acts as a Central Role BLE host that can send OTA firmware updates to a target STM32L4 MCU equipped with a Peripheral role nRF52832 add-on board.

First, connect a Central role nRF52832 add-on board to a Dragonfly development board as described earlier in this document. Next, flash the Dragonfly board with the "STM32L4_Tlera_nRF52dev_BLE_OTA_Update_Monitor_Utility_V0.X" Arduino sketch. Be sure that the correct board, "Serial + Mass Storage" and "SFLASH (QSPI)" options are selected before uploading the Dragonfly firmware.

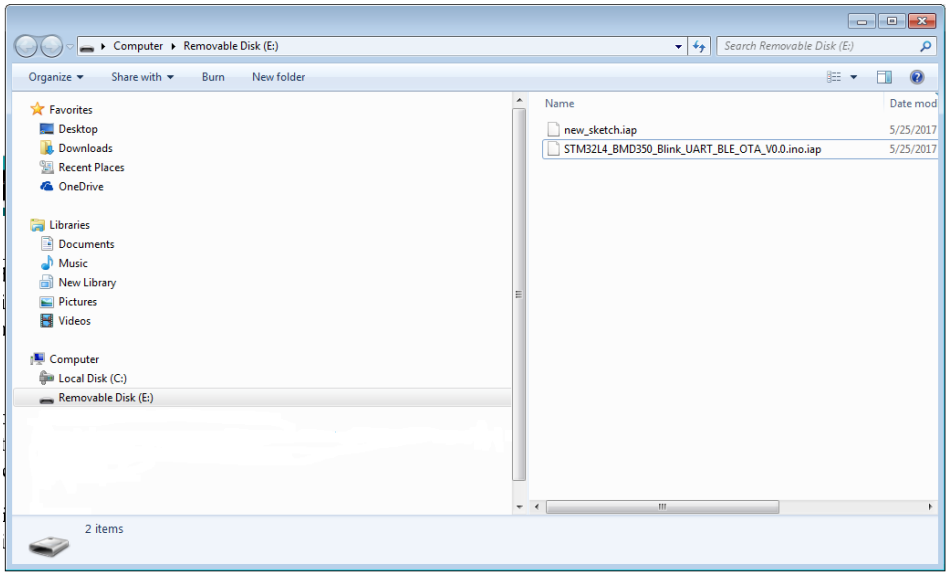


Data transfer from a Peripheral role NUS-capable device to your PC is now quite simple. Plug the Dragonfly into a USB port and open a serial monitor window and once the Dragonfly boots and the nRF52832 pairs with a NUS-capable peripheral device you will see:



Once the BLE NUS link is established, whatever is sent over the NUS from the remote device will appear in the serial window. So data from the remote device can be captured directly to your PC using any data-logging serial terminal application or even a serial plug-in for a spreadsheet application.

It should also be noted that the QSPI flash memory on the Dragonfly board is directly accessible from your PC as well. When plugged into a USB port, drivers resident on the Dragonfly board immediately mount a virtual flash drive on the host PC and open a file explorer window:

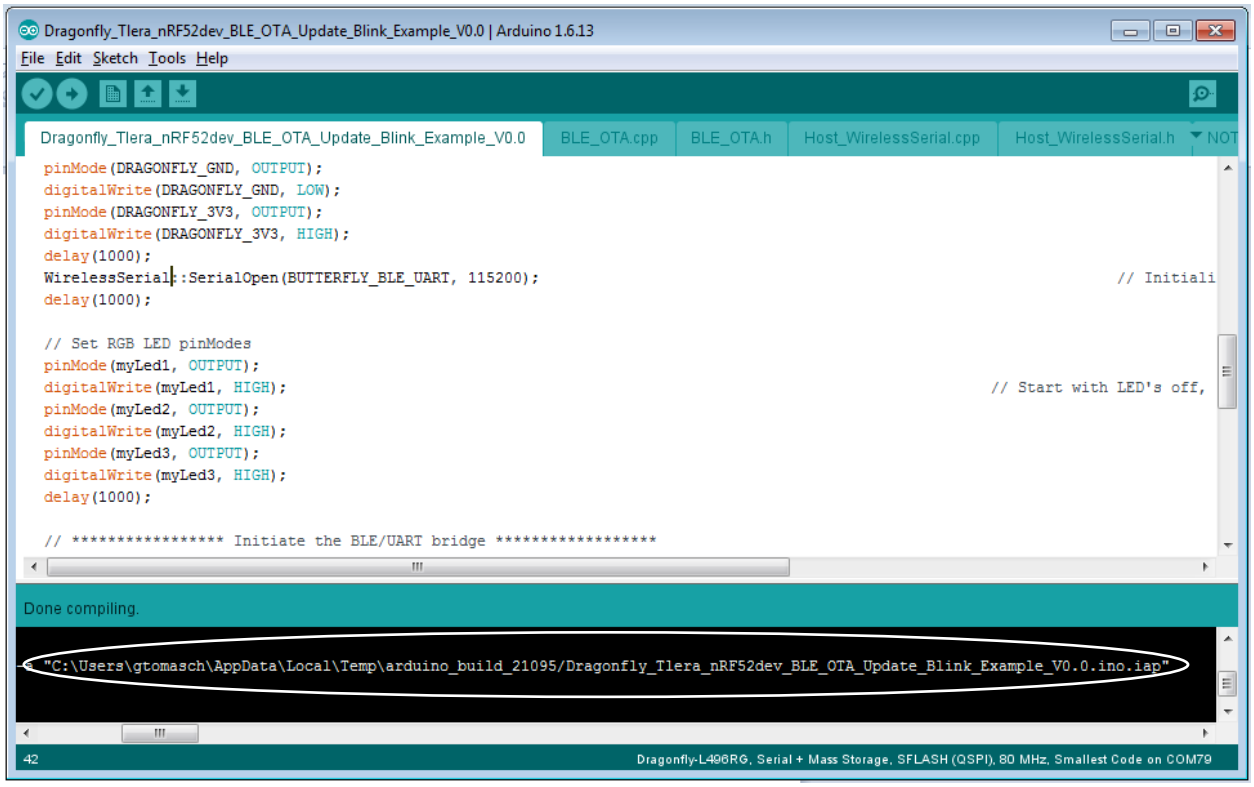


Note that the files listed in the right-hand pane of this explorer window are actually resident on the Dragonfly's QSPI flash chip! This capability is quite powerful; simply drag-and-drop to transfer any file to or from the Dragonfly's QSPI flash memory. The next section will discuss how to use this feature and the BLE/UART bridge to remotely update the Arduino firmware on a remote SLM32L4 device.

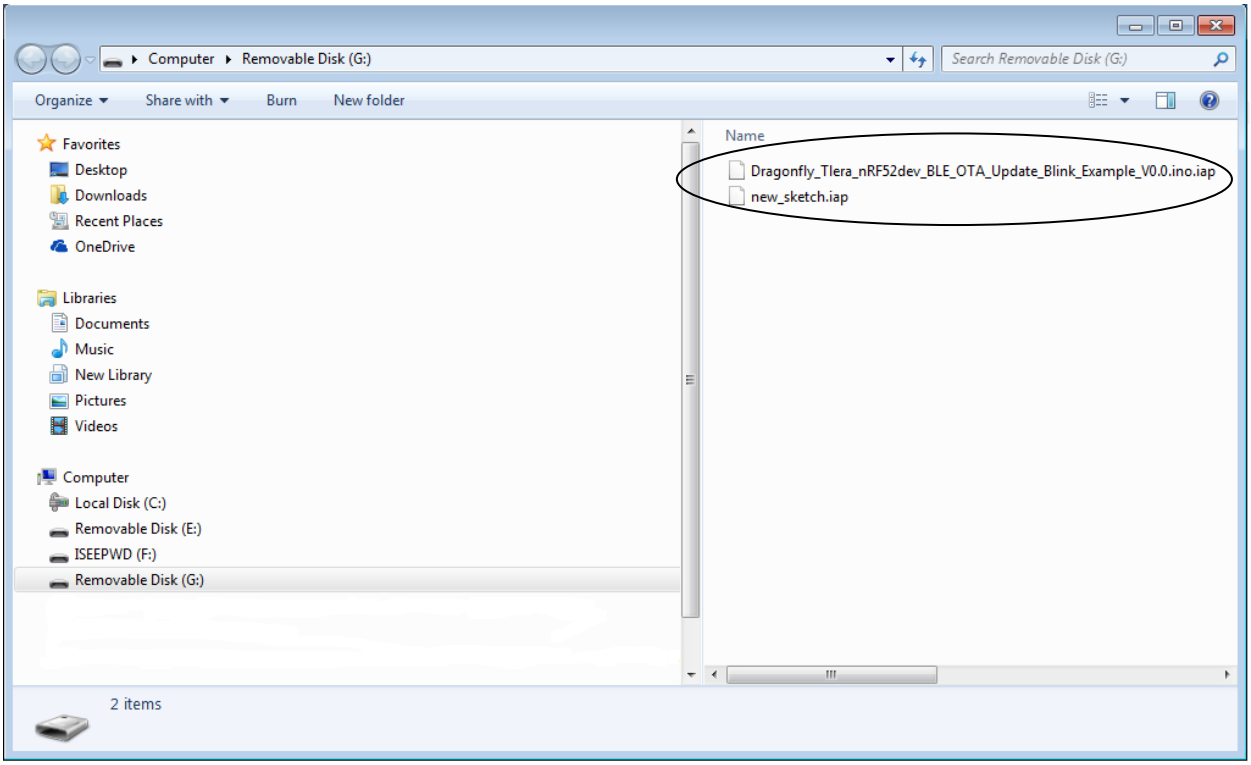
(II) Perform an OTA firmware update on a remote STM32L4 device

It is assumed that the data monitor/firmware update utility will be paired to a remote Dragonfly or Butterfly board equipped with a Peripheral role nRF52832 add-on board. Not only will data from the remote board appear in the serial monitor window, we can update the Arduino firmware of the remote board from the PC over the BLE NUS connection. The following example gives step-by-step instructions to do this.

We will further assume that the data monitor/firmware update utility is ready to go and that the remote board is a Dragonfly. Compile and upload the "Dragonfly_Tlera_nRF52dev_BLE_OTA_Blink_ExampleV0.0" sketch to the remote Dragonfly board using the Arduino IDE… Next, go to the Arduino IDE instance just used to compile the remote board's firmware. Go to the message window at the bottom of the screen and scroll both up and right until you locate the path message as to where the ".iap" firmware image file is located:



In this case, the file is “Dragonfly_Tlera_nRF52dev_BLE_OTA_Blink_Example_V0.0.ino.iap”. Typically, Arduino build files are located in “C:\Users\.....\AppData\Local\Temp\arduino_build_xxxxxx” folder where “xxxxxx” is a numerical designation assigned by the IDE. This is the firmware image file… Navigate to the “.iap” firmware image file and copy it to the Dragonfly virtual drive:

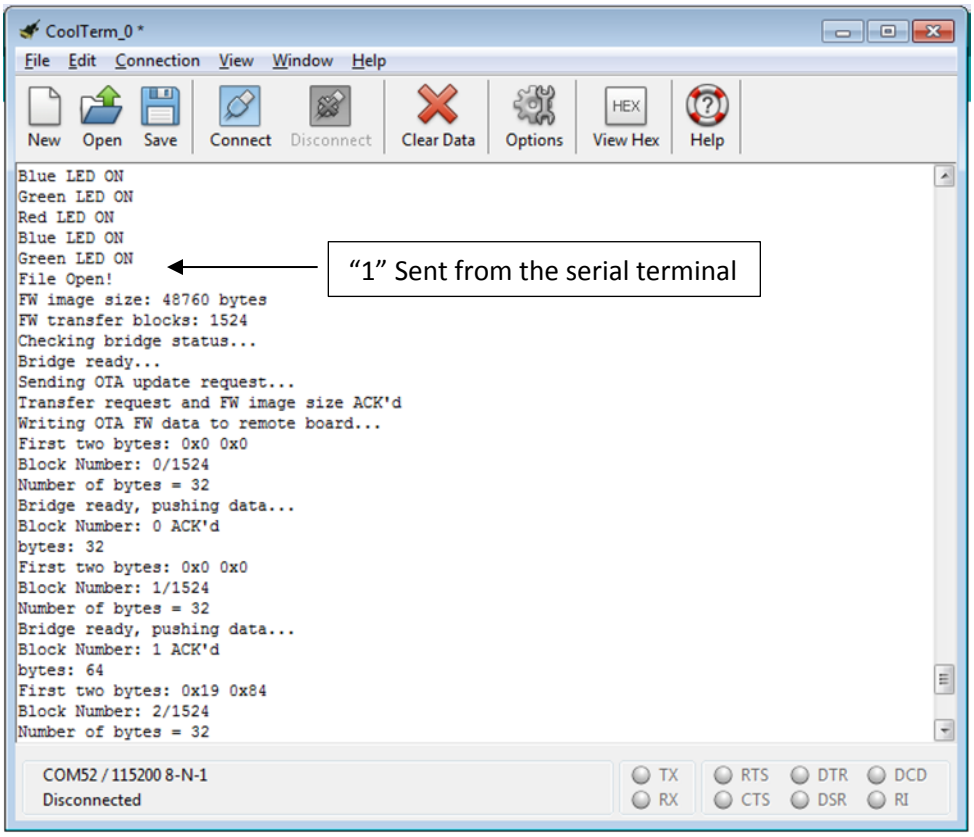


Copy and rename the firmware image file to “new_sketch.iap”. This is necessary because the “DOSFS” file system library used for building the Dragonfly sketch does not support file name wildcarding. The “new_sketch.iap” file will be parsed and transferred to the IU target board during the OTA firmware update. For purposes of this example we are using the same firmware image that is already loaded into the remote Dragonfly board. In general, “new_sketch.iap” can contain any valid Dragonfly firmware image.

Power up all of the boards including the remote Dragonfly board and make sure that there are no other active Central role devices that can pair with the remote nRF52832 add-on board. The blue LED’s on both nRF52832 add-on boards will be on steady when a BLE NUS connection is established. Next, open a serial terminal and connect to the host utility Dragonfly’s USB serial port using 115200 baud, 8-N-1 parity and no flow control.

Serial messages sent from the remote Dragonfly will appear in the serial terminal window. For this example, the remote dragonfly will flash red, blue and green LED’s and send messages over the BLE/UART bridge telling which color LED is turned on.

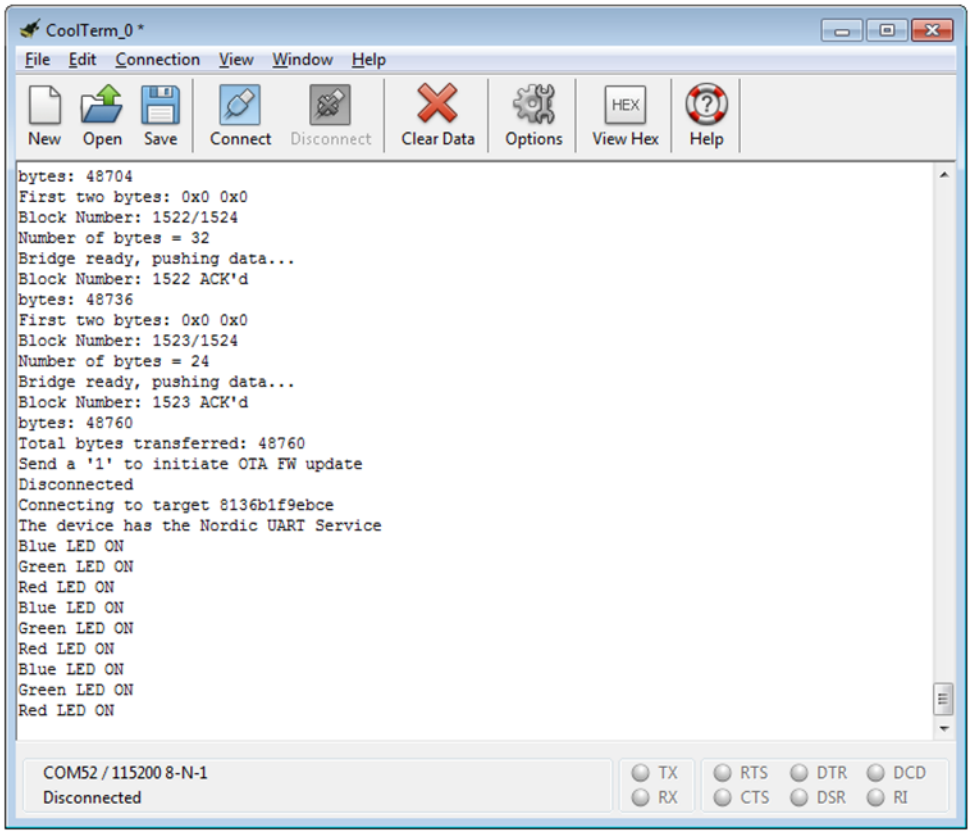
Send a “1” at any time from the serial terminal and the firmware update process will commence.



The terminal messages provide a wealth of information:

- New firmware image size
- Total number of data blocks required to transfer the firmware image
- BLE/UART bridge status
- Message ACK/NAK status
- First two bytes of each data block payload
- Data block payload size
- Transferred byte count

Upon successful completion, the total transferred byte count is indicated followed by a reminder prompt that sending a “1” will initiate another firmware update. Once the firmware update is complete, the remote STM32L4 MCU will reboot and data updates will resume. Any version of firmware for the target STM32L4 board can be copied to the Dragonfly virtual drive, renamed to “new_sketch.iap” and the OTA firmware update process repeated at any time by sending a “1” from the serial terminal window…



Depending upon the size of the new firmware image, it will take approximately 2 to10 minutes to complete the update. Once all desired firmware updates are complete, the utility will resume monitoring of the BLE UART messages coming from the remote STM32L4 board.