How to Set Up vSP (Serial UART) Service

BL6xx

Application Note *v1.0*

1 INTRODUCTION

The goal of this document is to demonstrate how to set up Laird's custom vSP (Serial UART) service.

2 OVERVIEW

Virtual Serial Port (vSP) is a method of sending data over Bluetooth Low Energy (Bluetooth LE), similar to how SPP works in Classic Bluetooth. Laird developed vSP as a custom Bluetooth LE Service at the request of our customers. vSP does not take advantage of the low power features of Bluetooth LE. For further details about vSP, please see the Laird Custom Bluetooth LE Serial Port Service application note available from the Documentation section of the applicable BLE product page.

This guide explains possible vSP scenarios using Laird BL6xx module and describes how to connect devices in each possible scenario. The guide is divided into the following sections:

3 REQUIREMENTS

The following are required for this process:

- PC running Windows XP or later
- UwTerminalX v1.10a or later – SSL version recommended
  (available from https://github.com/LairdCP/UwTerminalX/releases)
- One peripheral and one central role device using Laird Bluetooth development kits (BL654, BL652 or BL600) or dongles (BL654/BL620 only)

Note: BL600 supports peripheral role only/BL620 supports central role only

DVK-BL600 can be flashed with the BL620 firmware

BL600 **Breakout boards ONLY** support BL600 (peripheral) firmware

---

- Android or iOS device with Bluetooth LE support (Android 4.3 Jelly Bean or newer)
- Nordic nRF Connect mobile App (available on the Google Play Store/App Store)
- FTDI Drivers http://www.ftdichip.com/Drivers/VCP.htm (if not included with the OS)

4 GENERAL SETUP PREPARATION

This application note demonstrates the process using two DVK-BL652.

Note: If using a DVK board (BL654/BL653/BL652/BL600) ensure that, if current measurement track is cut, a jumper is fitted across the correct pins to complete the junction.

To prepare your setup, follow these steps:

1. Visit the Laird Github website for the device you are working with to download the sample applications:

https://github.com/LairdCP/BL654-Applications	Supports peripheral and central roles
https://github.com/LairdCP/BL653-Applications	Supports peripheral and central roles
https://github.com/LairdCP/BL652-Applications	Supports peripheral and central roles
https://github.com/LairdCP/BL600-Applications	Supports peripheral role only
https://github.com/LairdCP/BL620-Applications	Supports central role only

Note: For newer BL6xx devices, visit the device-appropriate GitHub

2. On the Github pages, click **Clone or Download** and select **Download ZIP.** Unzip/extract the files to a folder on your computer (such as C:\Laird\BL652) (Figure 1).
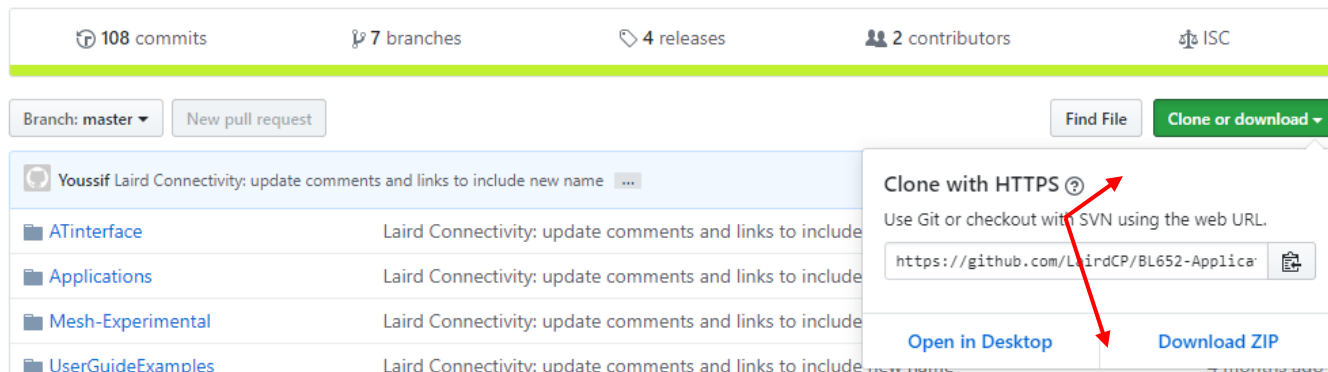

Figure 1: Github download

3. Download UwTerminalX from the UwTerminalx Github – Releases tab.
https://github.com/LairdCP/UwTerminalX/releases (UwTerminalX_vx.xx_Windows_SSL.zip recommended)

4. Plug Device 1 into a PC.

Note: The DVK has an FTDI USB-UART chip on board. When connected, Windows 7 or later should install the driver automatically. The driver is also available from the FTDI website.

5. Locate and note the COM port assigned in Windows Device Manager. The COM port number varies on each computer (Figure 2).
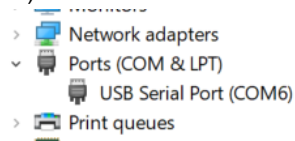

Figure 2: Device manager

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

6. Open UwTerminalX.
   a. On the Config tab, set the port to the COM port discovered in Step 5.
   b. Select the BL6xx device which preselects the default settings of device.

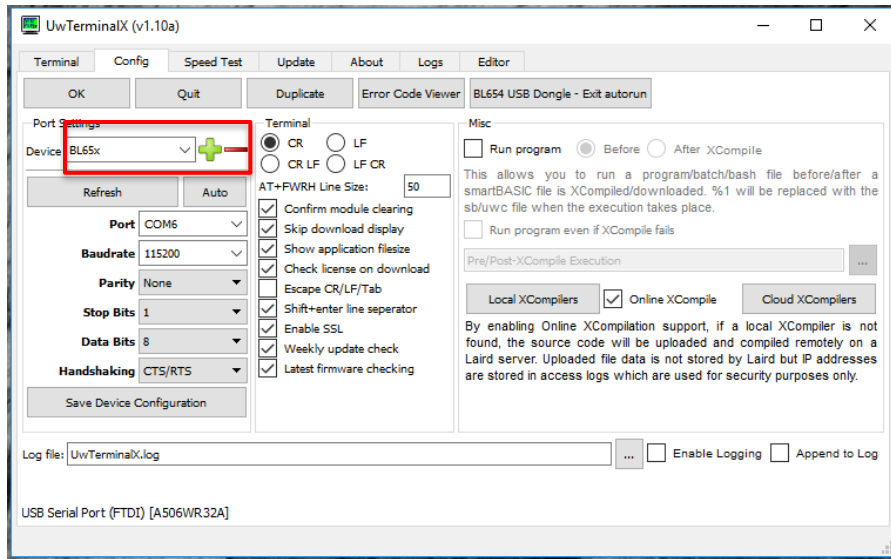| BL65x	115200 bps, none, 1, 8, CTS/RTS
BL600/BL620	9600 bps, none, 1, 8, CTS/RTS

Figure 3: UwTerminal Config tab

7. Type **ati 0** and press **Enter** to confirm the module is accessible. The module then returns the Device Name, if accessible, as shown in Figure 4.
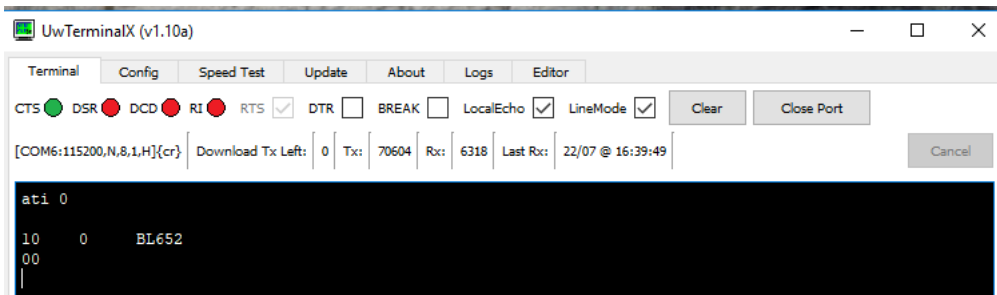

Figure 4: Device name

8. Enter the following commands to obtain the firmware version and BT MAC Address as well as confirm that no *smartBASIC* applications are loaded to the module.

ati 3	Returns *smart*BASIC firmware version
ati 4	Returns the Bluetooth MAC address
at+dir	List applications loaded to module

9. Use **at&f*** to clear the file system and restore default settings if at+dir returns anything other than 00 (Figure 5)
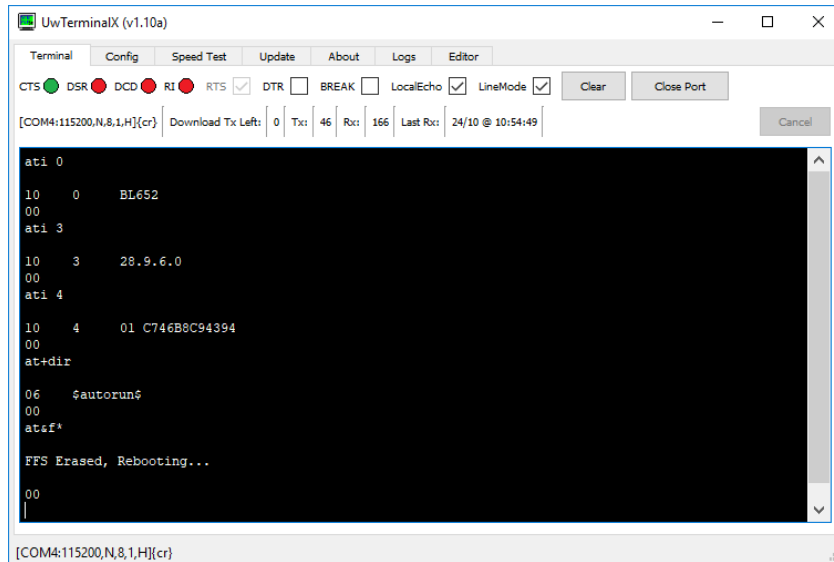
Figure 5: at&f*

10. Open a second UwTerminalX instance and **run through steps 1 – 9 to set up Device 2.**

5 CONNECTION SCENARIOS

Possible vSP scenarios are listed in Table 1. Click the links in the table to see specific instructions for each scenario:

Table 1: vSP scenarios

#	Scenario	Device 1 and Required Sample App (Peripheral Role)	Device 2 and Required Sample App (Central Role)	Supported
A	vSP server/peripheral on BL6xx vSP client/central on BL6xx	DVK-BL65x or DVK-BL600 or BL654 Dongle *$autorun$.vSP.UART.bridge.incoming*	DVK-BL65x or DVK-BL620 or BL620 Dongle *$autorun$.vSP.UART.bridge.outgoing*	✔
B	Hardware vSP server/ peripheral on BL6xx vSP client/central on BL6xx	DVK-BL65x or DVK-BL600 (Dongle cannot be put in HW vSP mode)	DVK-BL65x or BL620 or BL620 Dongle *$autorun$.vSP.UART.bridge.outgoing*	✔
C	vSP server on BL6xx vSP vSP client on Android or iOS	DVK-BL65x or DVK-BL600 or BL654 Dongle *$autorun$.vSP.UART.bridge.incoming*	Android or iOS *Nordic nRF Connect*	✔

5.1 Detailed Setup Instructions for Each Scenario

5.1.1 Scenario A: vSP server/peripheral on BL6xx, vSP client/central on BL6xx

After performing the General Setup, complete the following steps for scenario A:

1. Set up the vSP server/peripheral role device (Device 1) by doing the following:
   a. Select the UwTerminalX terminal connected to Device 1 (DVK1-BL652 in our testing).
   b. Right-click the terminal and select XCompile + Load + Run.
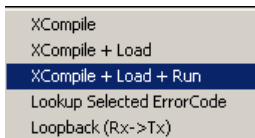
Figure 6: Select XCompile+Load+Run

   c. Select the *$autorun$.vSP.UART.bridge.incoming.sb* file from the Sample Applications downloaded from GitHub previously.
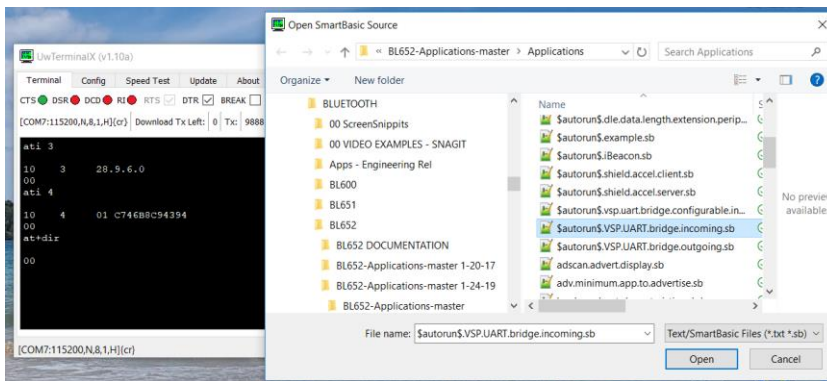
Figure 7: Select the applicable .sb file

2. (*Optional)* Modify the *$autorun$.vSP.UART.bridge.outgoing.sb* sample application to Auto-Connect by doing the following:
   a. Open *$autorun$.vSP.UART.bridge.outgoing.sb* from the previously downloaded *smart*BASIC sample app using a text editor such as Notepad++.
   b. Find and edit the **#define BTAddr** line to have the Bluetooth address of Device 1 (See Step 8 of General Setup Preparation).

   > **Note:** Bluetooth low energy addresses are 7 bytes, not 6 bytes as Bluetooth classic device addresses are. The 01 with a space in the above image is prepended to the displayed Bluetooth classic address: 01C746B8C94394

   c. Find and edit *'#define CONNECT_ON_STARTUP 0* setting it to *1* to enable the module to connect on startup.
   d. Save and close the file.
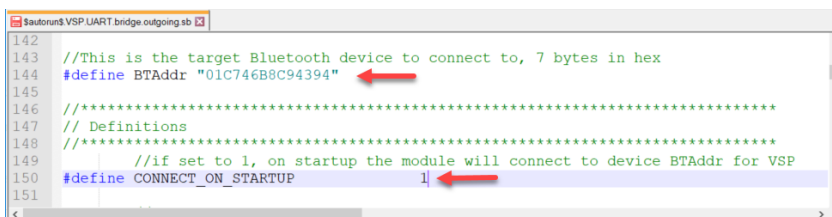
Figure 8: #define CONNECT_ON_STARTUP edits

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

3. Setup the the vSP Client/Central Role Device (Device 2) by doing the following:
   a. Select the UwTerminalX terminal connected to Device 2 (DVK2-BL652 in our testing)
   b. Right-click the terminal window and select **XCompile + Load + Run.**
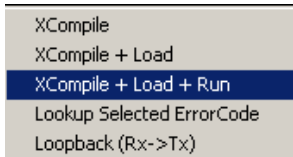
Figure 9: XCompile + Load + Run

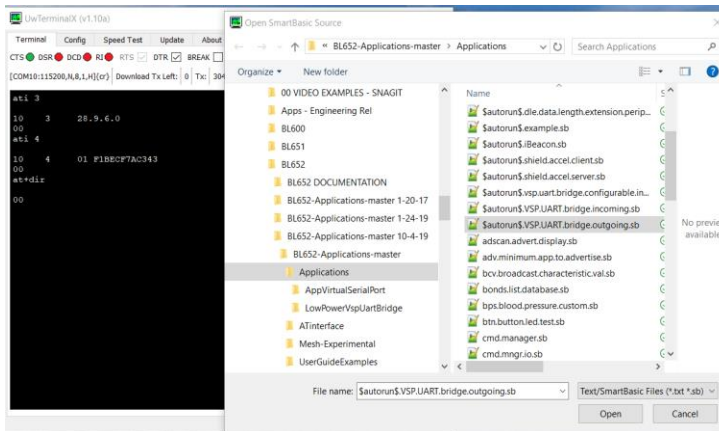   c. Select the *$autorun$.vSP.UART.bridge.outgoing.sb* file.

Figure 10: Select applicable .sb file

4. Connect the devices (Auto-connect/Scan and Initiate Connection) by doing the following:
   a. Auto-Connect (**Requires completion of Step 2**):
      ▪ If Central application was modified as per step 2, the devices should connect as soon as the application starts.

      Note: It may be necessary to power-cycle Device One if advertisements have timed out.

   b. Manual Connection - Scan and Initiate Connection
      ▪ If they do not auto-connect, Device One should now be advertising for a connection. You will not see anything in Device One terminal until the devices are connected.
      ▪ In terminal window for Device 2, with the application running, you should see **OK** followed by **>** indicating it is waiting for a command to be entered.
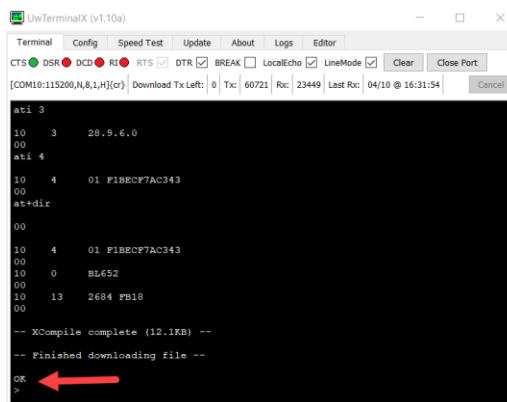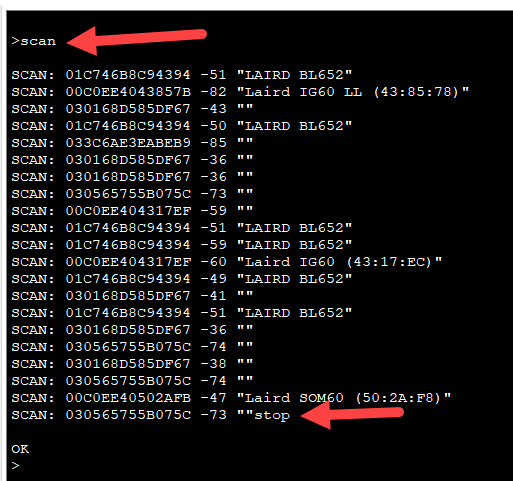
Figure 11: UxTerminal
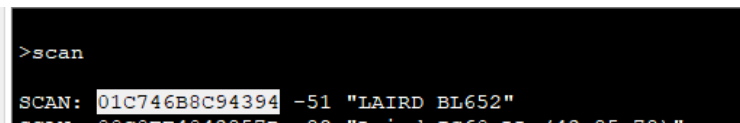
Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

c. Enter *scan* to scan for devices and locate the advertising LAIRD BL652 device.

d. Enter *stop* to stop scanning (enter it between the scanning output).

Figure 12: Enter scan and stop

e. Highlight the BT MAC Address of the device, right-click in the terminal, and select **COPY**.

Figure 13: Highlight BT MAC address

f. Enter the command *connect* followed by the BT MAC Address (right-click and select PASTE).

You should receive a response indicating the devices are connected and ready to transmit/receive.

Figure 14: Read to connect response

Date entered in either of the terminals is sent via the vSP connection and displayed in the connected terminal window.

Figure 15: Sent data

Note: If line mode is enabled (checked) in UwTerminalX, you must hit ENTER before the data is sent.

5.1.2 Scenario B: Hardware vSP server/peripheral on DVK-BL6xx vSP client/central on DVK-BL6xx

After performing the General Setup, complete the following steps for scenario B:

1. Setup the vSP Server/Peripheral Role Device (Device 1) in **Hardware vSP Bridge Mode** as per the applicable datasheet for your module. See the **vSP Command Mode** section for more information.

   a. Clear the module of any $autorun$ applications in the file system using **AT&F*.**

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

nAutorun Pin should be held **HIGH** (tied to VCC)

Note: If nAutorun pin is held low while vSP pin is high, the module enters vSP Command Mode which is used for OTA loading *smart*BASIC applications.

b. vSP Pin should be held HIGH (tied to VCC)
c. Reset or power cycle the module to start vSP adverts.

Note: The device *only* advertises for about ten seconds. If you scan and do not see the device, reset or power cycle the module to restart adverts.

2. Setup the the vSP Client/Central Role Device (Device 2) by doing the following:
   a. Select the UwTerminalX terminal connected to Device 2 (DVK2-BL652 in our testing)
   b. Right-click the terminal window and select **XCompile + Load + Run** (Figure 16).
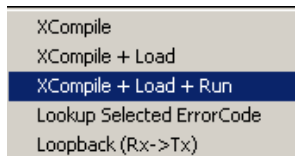
Figure 16: Right-click XCompile + Load + Run

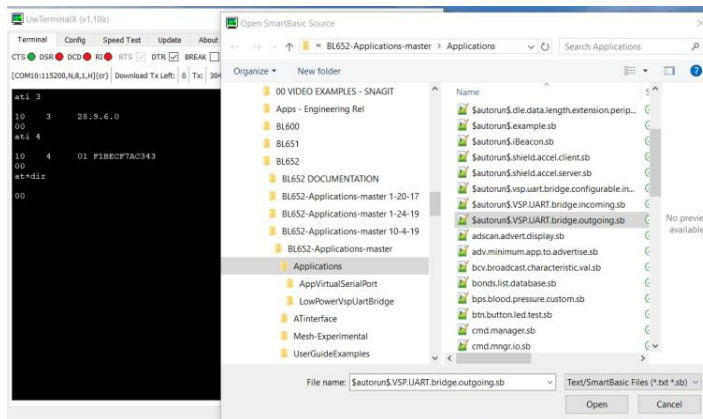c. Select the *$autorun$.vSP.UART.bridge.outgoing.sb* file ().

Figure 17: $autorun$.vSP.UART.bridge.outgoing.sb file

3. Connect the devices by doing the following:
   a. Device 1 should now be advertising for a connection – you will not see anything in this terminal until the devices are connected.
   b. In terminal window for Device 2, with the application running, you should see the following which indicates that it is waiting for you to enter a command (Figure 18).

   OK
   >

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

Figure 18: Waiting for a command to be entered

c. Enter *scan* to scan for devices and locate the advertising **LAIRD BL652** device.
d. Enter *stop* to stop scanning (it is entered between the scanning output).

Figure 19: Enter scan and stop

e. Highlight the BT MAC address of the device and right-click in the terminal – select COPY

Figure 20: Highlight and copy the BT MAC address

4. Enter the command *connect* followed by the BT MAC address (right-click and select **PASTE**).

You should receive a response indicating the devices are connected and ready to transmit/receive. Any data entered in either of the terminals is displayed in the connected terminal window (Figure 21).
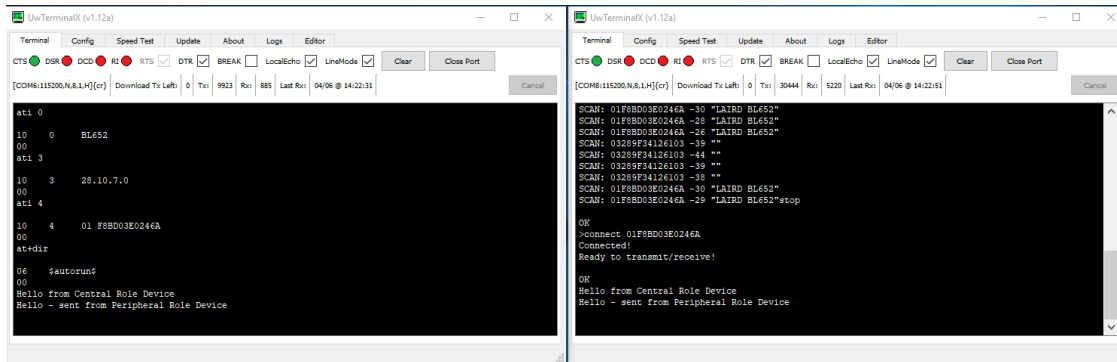
Figure 21: Devices connected - passing data

Note: If line mode is enabled (checked) in UwTerminalX you will need to hit ENTER before the data is sent.

5.1.3 Scenario C: vSP server on BL6xx, vSP client on Android or iOS Device (nRF Connect Application)

1. Set up the vSP Server/Peripheral Role Device (Devi Setup the vSP Server/Peripheral Role Device (Device 1) by doing the following:
   a. Select the UwTerminalX terminal connected to Device 1 (DVK1-BL652 in our testing).
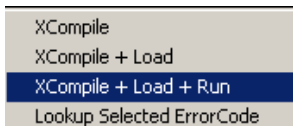   b. Right-click the terminal and select **XCompile + Load + Run.**

Figure 22: Select XCompile + Load + Run

   c. Select the *$autorun$.vSP.UART.bridge.incoming.sb* file from the Sample Applications downloaded from GitHub previously.
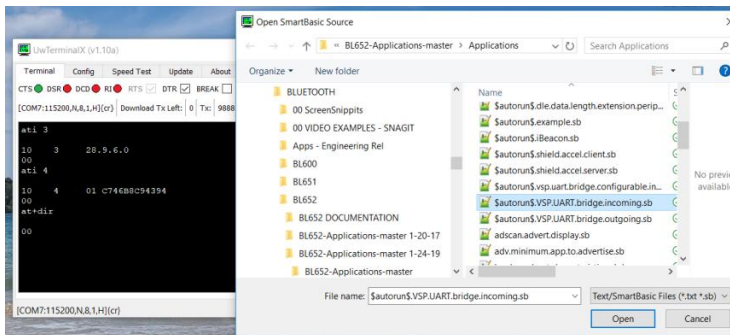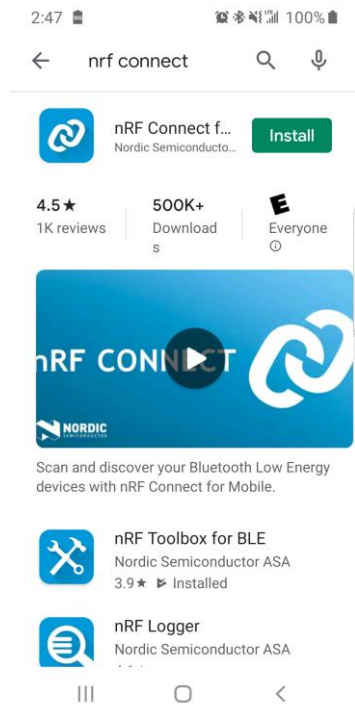

Figure 23: Select the applicable .sb file

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

2. Set up vSP Client on **Android/iOS** Device – Using ***Nordic nRF Connect for Mobile*** by doing the following:
   a. Install and Open Nordic nRF Connect.

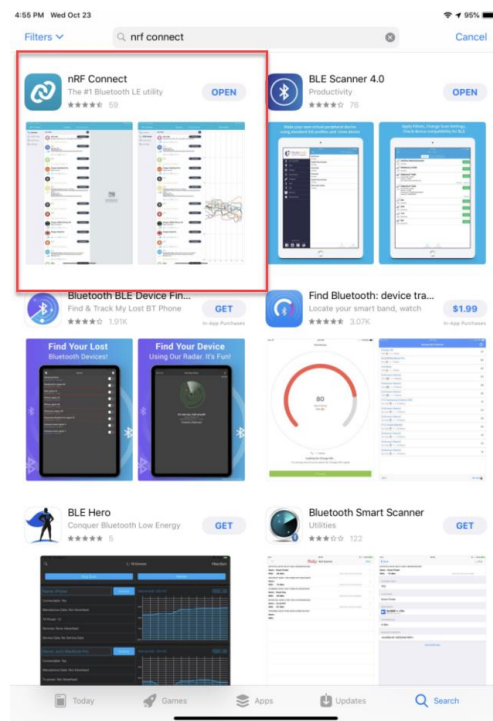*Figure 24: Install and open Nordic nRF Connect*

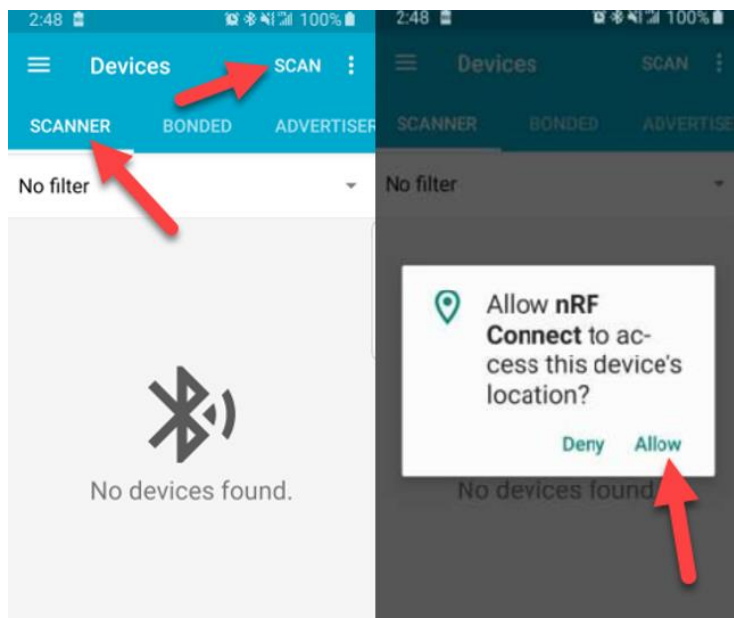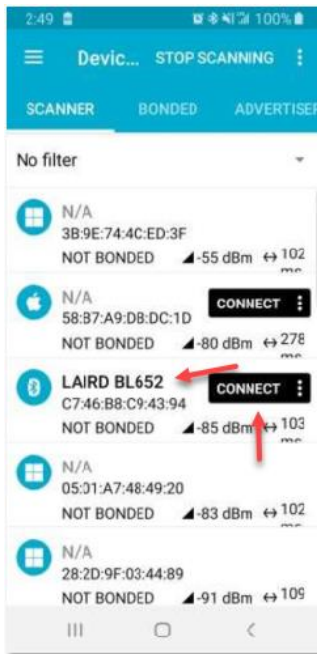3. If prompted select Allow for nRF Connect to access device's location.

Figure 25: Select Allow

4. Locate the Device Name (Laird BL65x is default) in the list of Bluetooth LE devices located and connect to the device.
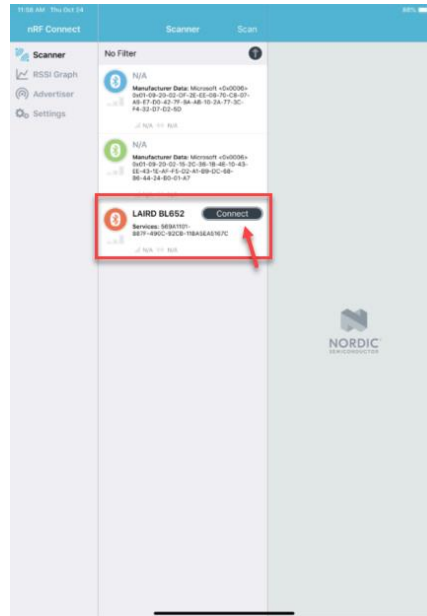
(Android)   **(iOS)**

Figure 26: Locate device name

5. Set up nRF Connect GATT Client to receive notifications by enabling CCCDs.
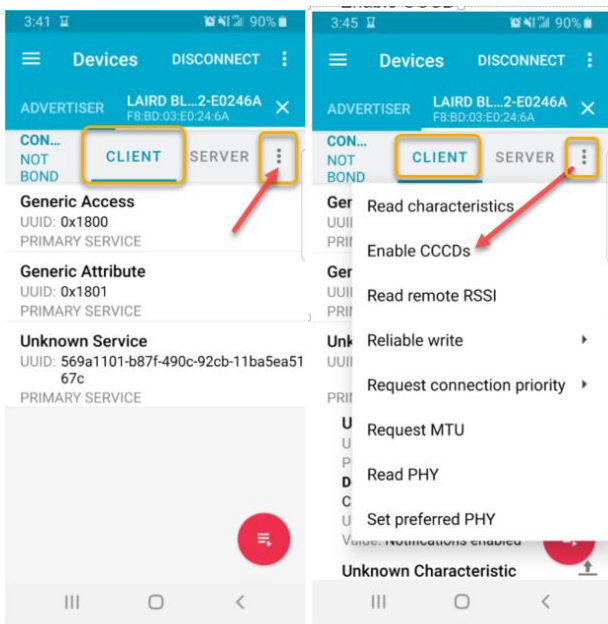**(Android – Enable CCCDs)**

Figure 27: Set up nRF Connect GATT client

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

a. (**Android only**) – Click Unknown Service to expand to view the GATT Table Characteristics and Descriptors.
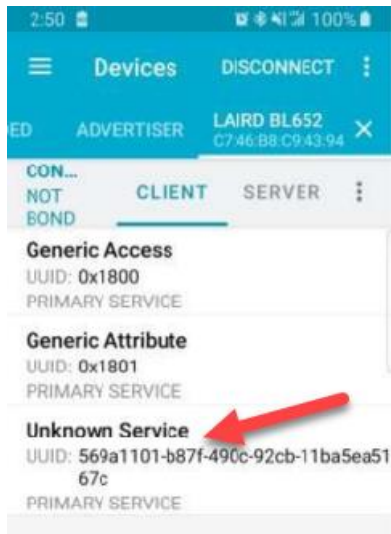

Figure 28: Click Unknown Service

6. Locate the **Rx Characteristic** of the Peripheral Role device by locating the UUID with the 2001 **Offset**. This is the UUID to which you transfer data from the Central role device (nRF Connect App). Click the arrow to open the dialog box.

Virtual Serial Port UUID's

vSP Service UUID
569a**1101**-b87f-490c-92cb-11ba5ea5167c
RX_FIFO characteristic UUID
569a**2001**-b87f-490c-92cb-11ba5ea5167c
TX_FIFO characteristic UUID
569a**2000**-b87f-490c-92cb-11ba5ea5167c
MODEM_IN characteristic UUID
569a**2003**-b87f-490c-92cb-11ba5ea5167c
MODEM_OUT characteristic UUID
569a**2002**-b87f-490c-92cb-11ba5ea5167c

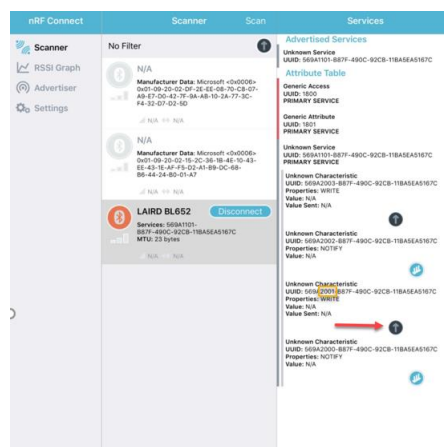*Figure 29: Virtual serial port UUIDs*
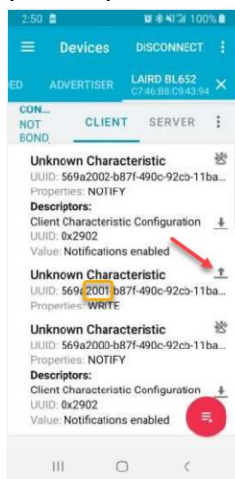
(Android)　　　　　　　　　　**(iOS)**
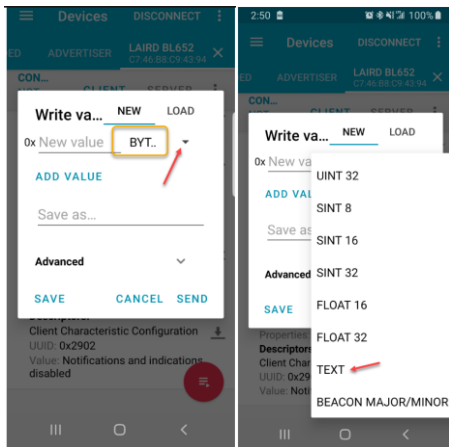

Figure 30: Locate Rx characteristic

7. Change the format to Text/UTF8 (or desired format).

(Android)   **(iOS)**

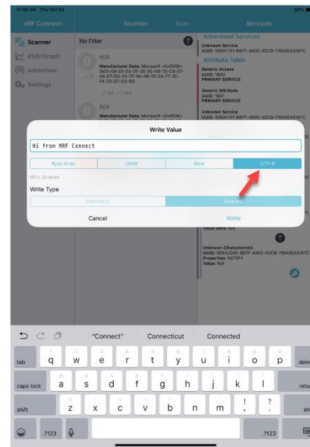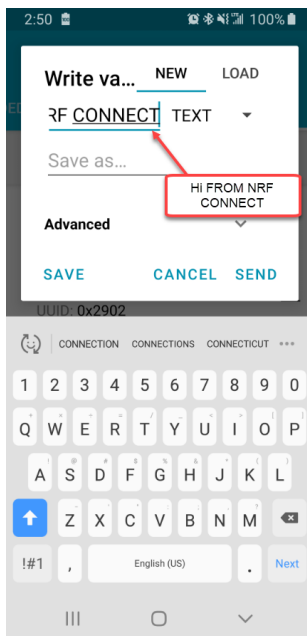**Figure 31: Change format**

8. Enter data to send/write to Device 1 (peripheral device)

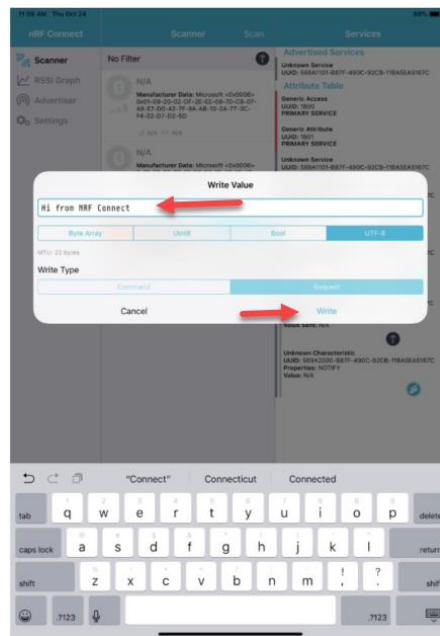**(Android)**   **(iOS)**

Figure 32: Enter data to send/write to Device 1

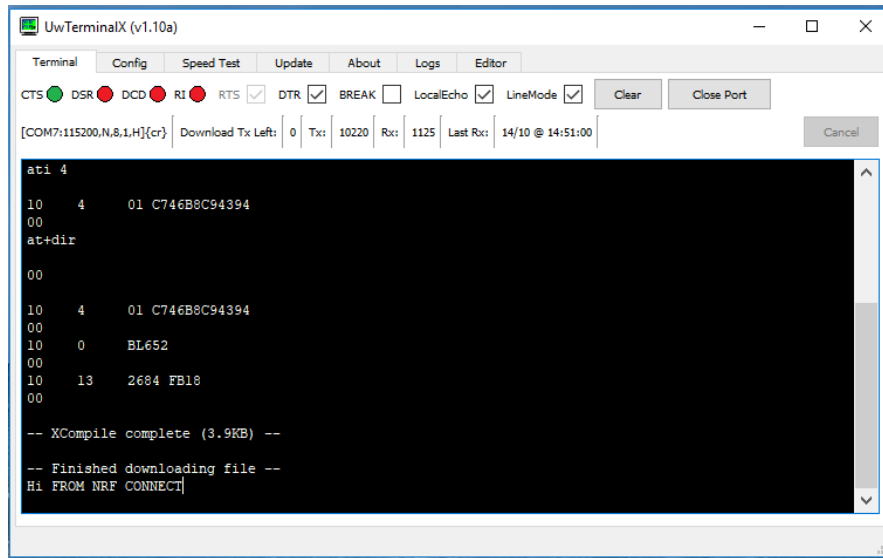Data appears in the terminal window of the connected peripheral device.

Figure 33: Displayed connected peripheral device data

Any data sent from the Terminal to the nRF Connect App is received by the TX Characteristic of the GATT server in nRF Connect (Offset 2000) from Figure 29.

TX_FIFO characteristic UUID
569a**2000**-b87f-490c-92cb-11ba5ea5167c

The data received will display in HEX format. At the time of this writing the nRF Connect App did not have an option for viewing the received data in Text or UTF8 format.

6 REVISION HISTORY

Version	Date	Notes	Contributor(s)	Approver
1.0	31 July 2020	Initial Release	Rikki Horrigan	Jonathan Kaye

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610