Command Manager Multiple Concurrent SPP and vSP

BT900

Application Note *v1.0*

INTRODUCTION

The goal of this document includes the following:

- Use command manager sample application to demonstrate multiple concurrent SPP connections
- Use command manager sample application to demonstrate multiple concurrent vSP connections
- Use command manager sample application to demonstrate multiple concurrent SPP and vSP connections

OVERVIEW

Command manager provides a command interface over the UART for many common BT900 Bluetooth operations. It allows you to configure, pair, and connect with other Bluetooth devices using SPP for classic Bluetooth (BTC) and Laird's Virtual Serial Port service (vSP) for Bluetooth Low Energy (BLE). It allows the BT900 to bridge the radio to the UART and allows data arriving over the radio to be presented to the UART via unsolicited messages identified by connection handle. Data from the host intended for the radio is presented by way of text based commands. This allows for multiple concurrent SPP and vSP connections to be demonstrated.

In this document we assume the BT900 is the master and that it will discover, pair, and connect with multiple remote devices. The following screen shots show the commands used on the BT900 master and assume you are already familiar with the configuration of the remote devices. You may need to use various types of devices to have enough to achieve multiple concurrent connections. We used a mixture of devices during the preparation of this application note, including BTM411 Bluetooth classic development kits and BL600 development kits but you could also use additional BT900 development kits, if available. Example setups are shown in Figure 1.

Note: Laird provides a library of sample *smart*BASIC applications including command manager, to provide a simple, easy-to-use guide for implementing a range of different functionality within your applications. The sample application library on GitHub is never intended to be a completely robust, end-customer application for use in real world applications.

Note: Although the BT900 supports up to seven BTC clients and up to five BLE clients, it may not be possible to implement the maximum BTC and BLE clients simultaneously. Developers are urged to extensively test any scenarios.

© Copyright 2015 Laird. All Rights Reserved

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

REQUIREMENTS

- BT900 development kit
- Two or more Bluetooth SPP devices such as Laird BT411 development kits
- Two or more Bluetooth Laird vSP such as BL600 development kits
- PC with enough USB ports for all of the above
- UWTerminalX – https://github.com/LairdCP/UwTerminalX/releases
- Cmd.manager sample app – https://github.com/LairdCP/BT900-Applications/blob/master/cmd.manager.sb

Note: For the purposes of this document, we assume you have familiarised yourself with compiling/loading *smart*BASIC and with making single vSP and vSP connections using the How to Set Up vSP and SPP with the BT900 application note.

Multiple concurrent SPP and vSP connections are possible, depending on the equipment you use. For example, in Figure 1 we illustrate a sample setup of two vSP and two SPP connections, all simultaneously connected to the BT900. Other combinations are possible, but the exact number of simultaneous connections is limited by the Bluetooth specification and the BT900 resources. Test your intended setup extensively to be sure that your desired functionality is achieved.

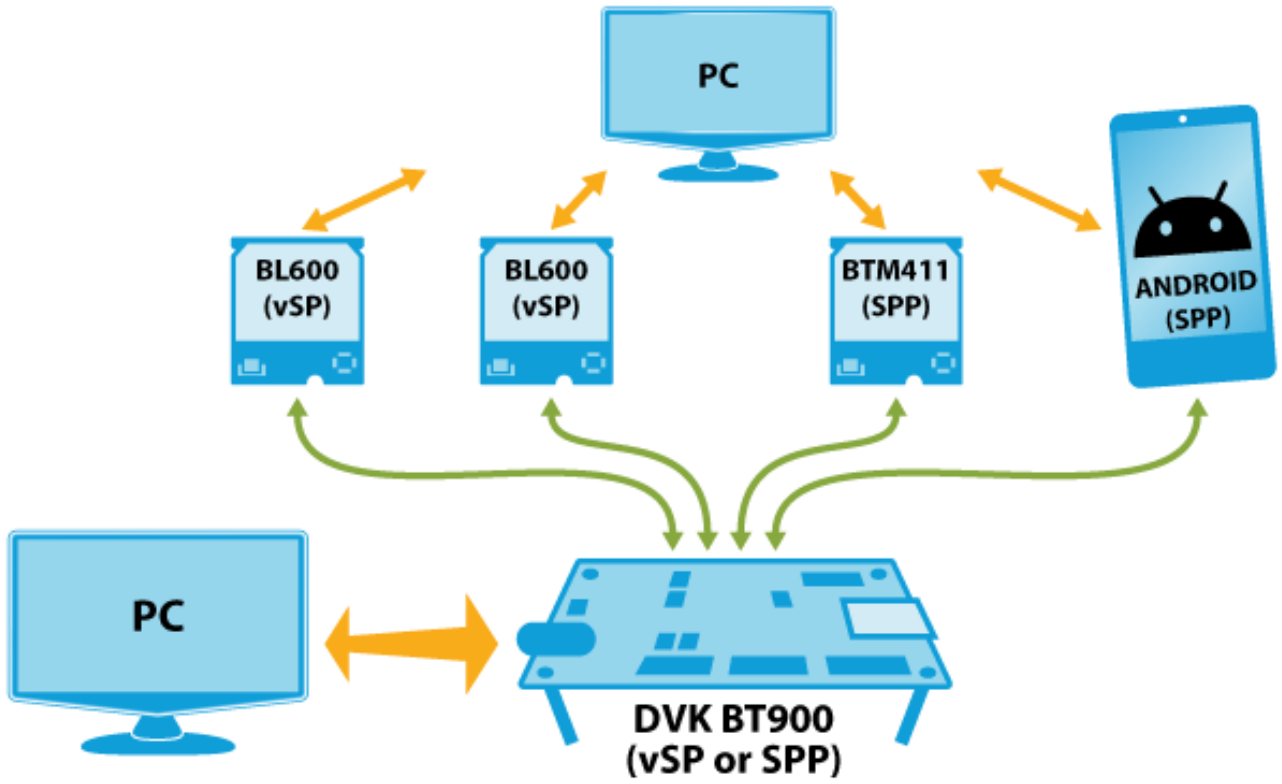

Figure 1: Sample configuration

© Copyright 2015 Laird. All Rights Reserved

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

BT900 Development Kit Setup

To set up the development kit, follow these steps:

1. Use UWTerminalX to return the BT900 dev board to factory defaults using the command at&f* as shown (Figure 2).

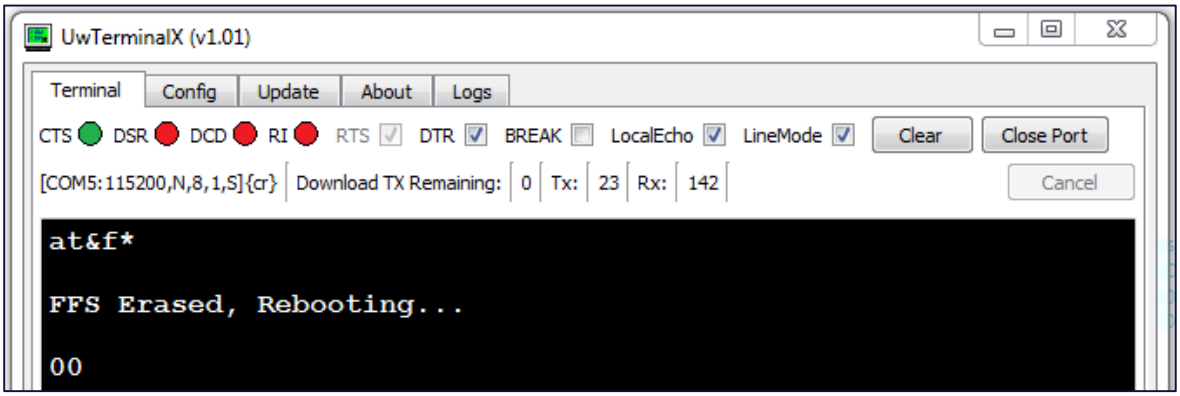

Figure 2: Factory default

2. Load command manager – use the right-click menu to select **Xcompile+load**.

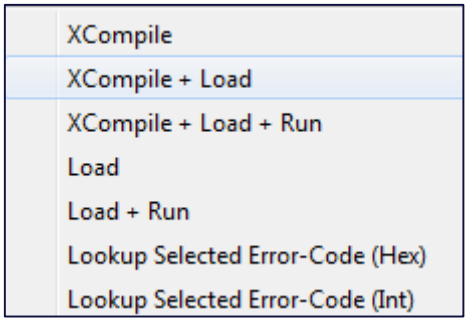

Figure 3: XCompile Load and Run

3. Select the **cmd.manager.sb** file.
4. Wait for the command manager program to load; this should take approximately 25 seconds.

Command manager can now be run by typing ***cmd*** followed by return.

Note: A complete list of commands available through command manager can be found in the source code file.

© Copyright 2015 Laird. All Rights Reserved

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

BTC (SPP) CONFIGURATION

You may need to first configure a number of settings depending on the capabilities of the remote device. Here (Figure 4) we set the BT900 IO capability for pairing as well as the inquiry settings to use when discovering other Bluetooth classic devices.

```
cmd


LAIRD BT900 - 000016A4093AC0


OK
>btc iocap 1

OK
>btc setpairable 1

OK
>inquiry config 1 2

OK
>inquiry start 20

OK
>
MAC: 0016A4001581 -46
MAC: 00027233928F -71
  EIR: 09094D454449412D5043020A0003020B11
  - Tag 0x09: MEDIA-PC
  - Tag 0x0A: 00
  - Tag 0x02: 0B11

MAC: 0016A4001589 -35

>
OK
>
```

Figure 4: BT900 configuration

cmd	Runs the loaded command manager program
btc iocap 1	Sets the BT900 IO capability for pairing to display yes/no
btc setpairable 1	Makes the BT900 pairable
inquiry config 1	Configures the inquiry to return MAC address and RSSI
inquiry start 20	Starts a Bluetooth classic inquiry with a 20-second timeout

© Copyright 2015 Laird. All Rights Reserved

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

BTC (SPP) PAIRING

You may need to first pair with the remote device. The actual process varies depending on the IO capabilities of the remote device but here (Figure 5) we show simple secure pairing with passkey comparison, as the remote device has both keyboard and display IO capabilities.

```
>btc pair 0016a4001581 1

OK
>
Passkey: 000053
>
Pair Req: 0016A4001581
>btc pairresp 1

OK
>
Pair Result: 0 0016A4001581
>btc pair 0016a4001589 1

OK
>
Passkey: 763033
>
Pair Req: 0016A4001589
>btc pairresp 1

OK
>
Pair Result: 0 0016A4001589
>
```

Figure 5: BTC pairing

btc pair 0016a4001581	Initiates pairing with the remote device specified
btc pairresp 1	Accepts the pairing in response to the pair req message

The pairing is successful when the pair result equals 0.

© Copyright 2015 Laird. All Rights Reserved

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

BTC(SPP) CONNECTION

Making a connection is now just a case of using the SPP connect command and the MAC address of the remote device returned from the earlier inquiry (Figure 6).

```
>spp connect 0016a4001581

OK
>
Pair Result: 0 0016A4001589
>
 --- SPP Connect: (00000000) handle = 1
>spp connect 0016a4001589

OK
>
Pair Result: 0 0016A4001589
>
 --- SPP Connect: (00000000) handle = 2
>
```

Figure 6: BTC SPP connection

spp connect 0016a4001581	Initiates a SPP connection with the remote device specified
--- SPP Connect: (00000000)	Shows a successful connection along with a handle to identify the connection
spp connect 0016a4001589	Initiates a SPP connection with the remote device specified
--- SPP Connect: (00000000)	Shows a successful connection along with a handle to identify the connection

Once connected, any incoming data from the remote device is passed to the UART with the associated port/connection handle (Figure 7).

```
Port Handle: 1 Length: 1
Data: t
>
Port Handle: 1 Length: 1
Data: e
>
Port Handle: 1 Length: 1
Data: s
>
Port Handle: 1 Length: 1
Data: t
>
Port Handle: 2 Length: 1
Data: t
>
Port Handle: 2 Length: 1
Data: e
>
Port Handle: 2 Length: 1
Data: s
>
Port Handle: 2 Length: 1
Data: t
```

Figure 7: BTC SPP data received

To send data to a remote device you must write data to the required connection handle (Figure 8).

spp write 1 Laird	***1*** is the connection handle and ***Laird*** is the data to be written to the remote device.

© Copyright 2015 Laird. All Rights Reserved

```
>spp write 1 Laird

OK
>spp write 2 Laird

OK
>
```

Figure 8: BTC SPP data sent

At this point we now have two concurrent SPP connections with remote devices and can send and receive data to/from either device via the UART. Further SPP connection can be added if required using the same procedure detailed previously.

BLE (vSP) Scan

We can now terminate our existing BTC SPP connections and establish BLE vSP connections or we can retain the BTC SPP connections and add BLE vSP connections as desired. Each new connection has a new connection handle.

The vSP service UUID is 569a1101-b87f-490c-92cb-11ba5ea5167c so we need to look for adverts which contain that UUID (Figure 9). Note that the UUID appears in reverse.

```
>scan start 1000 0

OK
>
ADV:00600308AD7EDF 02011A0BFF4C000906030D00000000 0 -73
ADV:01D931BE6CDC3A 02010609094C545F555041535303030A1811077C16A55EBA11CB920C497FB801119A56 0 -37
ADV:00600308AD7EDF 02011A0BFF4C000906030D00000000 0 -73
ADV:01D931BE6CDC3A 02010609094C545F555041535303030A1811077C16A55EBA11CB920C497FB801119A56 0 -34
ADV:01CB8D99009102 02010609094C545F555041535303030A1811077C16A55EBA11CB920C497FB801119A56 0 -31
ADV:01D931BE6CDC3A 02010609094C545F555041535303030A1811077C16A55EBA11CB920C497FB801119A56 0 -37
ADV:01CB8D99009102 02010609094C545F555041535303030A1811077C16A55EBA11CB920C497FB801119A56 0 -28
ADV:00600308AD7EDF 02011A0BFF4C000906030D00000000 0 -73

Scanning stopped via timeout

>
```

Figure 9: BLE advert scan

scan start 1000 0	Begins the BLE scanning process with the specified timeout and whitelist filter.

The resulting scan data shows two BLE devices adverting with the vSP service, 01D931BE6CDC3a and 01CB8D99009102. You may need to set the timeout longer depending on the advertising interval set for the vSP peripheral devices.

Note: Actual connection handles may vary depending on how many SPP and vSP devices you have currently connected. Do not assume that your connection handle will be the same as those shown in the screenshots contained within this document. The screenshots are only to show what sort of interaction to expect.

© Copyright 2015 Laird. All Rights Reserved

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

vSP Connection

```
>connect 01cb8d99009102 50 30 30 50

OK
>
 --- Connect: (0002FE01) handle=2
Conn Interval 7500
Conn Supervision Timeout 100000
Conn Slave Latency 0
```

Figure 10: BLE vSP connection

Connect 01cb8d99009102 50 30 30 50	Initiates a connection to the specified device with the connection parameters provided (connection timeout, minimum connection interval, maximum connection interval, supervision timeout).

After establishing one vSP connection, you can make further vSP connections if required. You may also add further SPP connections using the procedures detailed previously in this document. Note that each connection has a unique connection handle.

Note: We assume you are already familiar with vSP and have read the following document:
Using Virtual Serial Port Service (vSP) with smart BASIC

BLE (vSP) GATT Table

Once connected to a vSP device, you can query the GATT table using the following:

gattc tablemap 2	***2*** is the connection handle of the device for which you want the GATT table.

```
>gattc tablemap 2

S:1 ,(7) ,FE011800
 C:3 ,0000000A ,FE012A00 ,0
 C:5 ,00000002 ,FE012A01 ,0
 C:7 ,00000002 ,FE012A04 ,0
S:8 ,(11) ,FE011801
 C:10 ,00000020 ,FE012A05 ,0
  D:11 ,FE012902
S:12 ,(24) ,FE01180A
 C:14 ,00000002 ,FE012A29 ,0
 C:16 ,00000002 ,FE012A24 ,0
 C:18 ,00000002 ,FE012A25 ,0
 C:20 ,00000002 ,FE012A27 ,0
 C:22 ,00000002 ,FE012A26 ,0
 C:24 ,00000002 ,FE012A28 ,0
S:25 ,(65535) ,FD021101
 C:27 ,00000010 ,FD022000 ,0
  D:28 ,FE012902
 C:30 ,0000000C ,FD022001 ,0
 C:32 ,00000010 ,FD022002 ,0
  D:33 ,FE012902
 C:35 ,0000000C ,FD022003 ,0
OK
```

Figure 11: BLE vSP GATT table

© Copyright 2015 Laird. All Rights Reserved

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

S	Service handle
C	Characteristic handle
D	Descriptor handle

BLE (vSP) Read/Write

You can also read back individual attributes from the GATT table. Here we read back the value of handle 3 from the GATT table of connection handle 2. Looking at the GATT table above, we can see that the UUID for this characteristic is 0x2A00 which is the device name, in this case being LT_UPASS.

```
>gattc read 2 3 0

OK
>
EVATTRREAD(hConn=196097,handle=3,status=0)
 >BleGattcReadData(data=4C545F5550415353,offset=0)
                  (data=LT_UPASS)

>
```

Figure 12: BLE vSP GATT read

gattc read 2 3 0	***2*** is the connection handle, ***3*** is the attribute handle, and ***0*** is the offset.

You might also want to turn on notifications for the vSP service in which case you must identify the descriptor from the GATT table. 0x2000 is the 16 bit offset for the vSP TX characteristic so you can see its descriptor handle is 28 from the GATT table above. We use the writecmd to write 0x0100 into the descriptor to enable notifications with an EVNOTIFYBUF if successful. You can also enable notifications for modemout characteristics descriptor located at handle 33.

```
>gattc writecmd 1 28 0100

OK
>
EVNOTIFYBUF

>gattc writecmd 1 33 0100

OK
>
EVNOTIFYBUF
```

Figure 13: BLE vSP GATT write

gattc writecmd 1 28 0100	***1*** is the connection handle, ***28*** is the attribute handle, and ***0100*** is the value.
gattc writecmd 1 33 0100	***1*** is the connection handle, ***33*** is the attribute handle, and ***0100*** is the value.

Now whenever a notification is received from the GATT server, we receive a notification event identifying the connection handle on which it was received, along with the attribute handle and the data received.

© Copyright 2015 Laird. All Rights Reserved

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

The following screenshot (Figure 14) shows data being received from connection handle 1 with the attribute handle being 27.

```
EVATTRNOTIFY()
 >BleGattcNotifyRead(hConn=0001FF00,handle=27,Dumped=0,data=0D)
EVATTRNOTIFY()
 >BleGattcNotifyRead(hConn=0001FF00,handle=27,Dumped=0,data=74)
 >BleGattcNotifyRead(hConn=0001FF00,handle=27,Dumped=0,data=65)
 >BleGattcNotifyRead(hConn=0001FF00,handle=27,Dumped=0,data=73)
EVATTRNOTIFY()
 >BleGattcNotifyRead(hConn=0001FF00,handle=27,Dumped=0,data=74)
 >BleGattcNotifyRead(hConn=0001FF00,handle=27,Dumped=0,data=0D)
EVATTRNOTIFY()
```

Figure 14: BLE vSP Notification event

The next screenshot (Figure 15) shows data being received from connection handle 2

```
>
EVATTRNOTIFY()
 >BleGattcNotifyRead(hConn=0002FE01,handle=27,Dumped=0,data=0D)
EVATTRNOTIFY()
 >BleGattcNotifyRead(hConn=0002FE01,handle=27,Dumped=0,data=74)
EVATTRNOTIFY()
 >BleGattcNotifyRead(hConn=0002FE01,handle=27,Dumped=0,data=65)
 >BleGattcNotifyRead(hConn=0002FE01,handle=27,Dumped=0,data=73)
 >BleGattcNotifyRead(hConn=0002FE01,handle=27,Dumped=0,data=74)
 >BleGattcNotifyRead(hConn=0002FE01,handle=27,Dumped=0,data=0D)
EVATTRNOTIFY()
```

Figure 15: BLE vSP Notification event

BTC (SPP)/BLE (vSP) Disconnections

The following commands can be used to disconnect any existing connection using its connection handle (Figure 16):

spp disconnect 1 n	Causes the BTC connection where ***n*** is the connection handle ***1*** to be terminated.
disconnect n	Causes the BLE connection where ***n*** is the connection handle ***1*** to be terminated.

```
>disconnect 2

OK
>
 --- Disconnect: (0002FE01) handle=2
```

Figure 16: Terminating a BLE connection

© Copyright 2015 Laird. All Rights Reserved

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

REFERENCES

The following documents are also accessible from the BT900 product page of the Laird website (Documentation tab):

- How to Set Up vSP and SPP with the BT900
- BT900 vSP and SPP Server
- BT900 DVK Hardware Integration Guide
- cmd.manager.sb
- UwTerminalX
- Using Virtual Serial Port Service (vSP) with smart BASIC

REVISION HISTORY

Version	Date	Notes	Approver
1.0	11 Mar 2016	Initial Release	Mark Duncombe

© Copyright 2015 Laird. All Rights Reserved

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