SMARTZ SAMPLE APP

Application Note *v1.2*

INTRODUCTION

The goals of this document include the following:

- To give an overview of the smartZ application
- To explain how to use the smartZ application on Laird's BT900 module

OVERVIEW

smartZ is a *smart*BASIC application that provides command-line interface to interact with the BT900 development kit (DVK). The application provides commands that are similar to the BlueZ commands used on Linux which makes it easy for BlueZ users to interact with the BT900 module through this application. In addition, the simplicity of the commands makes using the smartZ application relatively easy for all users.

There are four main commands that are available through this application:

- hcitool – Provides an interface to basic HCI functions such as inquiry, scanning, connecting, and disconnecting.
- gatttool – Allows you to communicate with the GATT server on remote devices. This includes service/characteristic/descriptor discovery, writing and reading characteristic values, and enabling notifications/indications.
- rfcomm – Used to connect the BT900 to a remote SPP device.
- hciconfig – Used to configure the BT900.

REQUIREMENTS

The following are required for this application:

- BT900 DVK with firmware version 9.1.8.0 or newer
- Mini USB cable – Included with BT900 DVKs
- Windows PC
- UwTerminal – Available as a free download from Laird
- smartZ.sb – Sample *smart*BASIC application (provided in the *smart*BASIC folder of the BT900 firmware zip file and available at Github)

Note: This application note assumes that you are familiar with how to download *smart*BASIC applications into the BT900 using the UwTerminal utility. If not, please refer to the BT900 product page and documentation for additional information.

COMMANDS

hcitool

Command	Description
dev	Display local devices
inq	Inquire remote devices
scan	Scan for remote devices
name	Get name from remote device

Command	Description
con	Display active connections
pair	Pair with remote device
unpair	Unpair from remote device
leadv	Start LE advertisement
noleadv	Stop LE advertisement
lescan	Start LE scan
lecc	Create a LE Connection
ledc	Disconnect a LE Connection

For more information on the use of each command, use **hcitool <command> --help**.

gatttool

Command	Description
--primary	Primary Service Discovery
--characteristics	Characteristics Discovery
--char-read	Characteristics Value/Descriptor Read
--char-write	Characteristics Value/Descriptor Write
--char-desc	Characteristics Descriptor Discovery
--listen	Listen for notifications and indications
Options	
-b, --device=MAC	Specify remote Bluetooth address
--sec-level=Level	Set security level. Default: low [low | medium | high]

For more information on the usage of each command, use **gatttool <command> --help**.

rfcomm

Command	Description
connect	Connect to a device running SPP
^^^	Disconnect from device

hciconfig

Command	Options	Description
auth		Enable Authentication
noauth		Disable Authentication
piscan		Enable Page and Inquiry scan
noscan		Disable scan
iscan		Enable Inquiry scan
pscan		Enable Page scan
name	[name]	Get/Set local name
inqmode	[mode]	Get/Set inquiry mode
inqtype	[type]	Get/Set inquiry scan type
inqtpl	[level]	Get/Set inquiry transmit power level

Command	Options	Description
letpl	[level]	Get/Set LE transmit power level
locap	[iocap]	Get/Set the IO capability
putkey	<bdaddr>	Store link key on the device
delkey	<bdaddr>	Delete link key from the device
oobdata		Display local OOB data
version		Display version information
leadv		Enable LE advertising
noleadv		Disable LE advertising

COMMAND USAGE

Classic Bluetooth Commands

Display local Bluetooth device

```
hcitool  dev
```

Example:

```
------->hcitool dev
Device:
      smartZ-0016A4093A89    0016A4093A89
------->
```

Inquiring for remote Bluetooth devices

```
hcitool  inq [--length=t]
             [--numrsp=N]
```

Example:

```
------->hcitool inq
------->
Inquiring ...
------->
      0016A4093A64
      0016A4FEF130
------->
------->hcitool inq --length=10 --numrsp=1
------->
Inquiring ...
------->
      0016A4093A64
------->
```

Scan for remote devices

```
hcitool scan  [--length=t]
```

[--numrsp=N]

Example:

```
------->hcitool scan
------->
Scanning ...
------->
      0016A4093A64  smartZ-0016A4093A64
      0016A4FEF130  Laird FEF130
------->
------->hcitool scan --length=10 --numrsp=1
------->
      0016A4093A64  smartZ-0016A4093A64
------->
```

Get name from remote device

hcitool name <bdaddr>

Example:

```
------->hcitool name 0016A4093A64
------->
      0016A4093A64  smartZ-0016A4093A64
------->
```

Pairing/unpairing with remote devices

hcitool pair <bdaddr> [--bond]
hcitool unpair <bdaddr>

Example:

```
------->hcitool pair 0016A4093A64
Pairing
------->
Paired
------->hcitool unpair 0016A4093A64
Unpaired
------->
```

Connect to a device running the SPP service

rfcomm connect <bdaddr>

Example:

```
------->rfcomm connect 0016A4093A64
------->
Connected to 0016A4093A89
```

```
Type "^^^" for hangup
test
Sending data over RFCOMM channel 1 2 3 and over
^^^
------->
```

Low Energy Commands

Scanning for BLE devices

hcitool lescan [--length=t]

Example:

```
------->hcitool lescan
------->
LE Scan ...
------->
      000016A4093A64  LAIRD_HRM
-------->hcitool lescan --length=1
-------->
LE Scan ...
------->
      000016A4093A64  LAIRD_HRM
------->
```

Connecting to BLE devices

hcitool lecc <bdaddr>

Example:

```
------->hcitool lecc 000016A4093A64
Connecting ...
------->
Connection handle 130816
------->
```

Displaying active connections

hcitool con

Example:

```
-------->hcitool con
-------->
Connections:
      > 000016A4093A64 handle 130816
-------->
```

Discovering primary services

gatttool -b <bdaddr> --primary

Example:

```
gatttool -b 000016A4093A64 --primary
-------->
attr handle = 0x00000001, end grp handle = 0x00000004, uuid = FE011801
attr handle = 0x00000005, end grp handle = 0x0000000B, uuid = FE011800
attr handle = 0x0000000C, end grp handle = 0x0000001E, uuid = FE01180A
attr handle = 0x0000001F, end grp handle = 0x00000024, uuid = FE01180D
attr handle = 0x00000025, end grp handle = 0x00000027, uuid = FE01180F
-------->
```

Discovering characteristics

gatttool -b <bdaddr> --characteristics

Example:

```
gatttool -b 000016A4093A64 --characteristics
-------->
handle = 0x00000002, char properties = 0x00000022, char value handle = 0x00000003, uuid = FE012A05
handle = 0x00000006, char properties = 0x00000002, char value handle = 0x00000007, uuid = FE012A00
handle = 0x00000008, char properties = 0x00000002, char value handle = 0x00000009, uuid = FE012A01
handle = 0x0000000A, char properties = 0x00000002, char value handle = 0x0000000B, uuid = FE012A04
handle = 0x0000000D, char properties = 0x00000002, char value handle = 0x0000000E, uuid = FE012A29
handle = 0x0000000F, char properties = 0x00000002, char value handle = 0x00000010, uuid = FE012A24
handle = 0x00000011, char properties = 0x00000002, char value handle = 0x00000012, uuid = FE012A25
handle = 0x00000013, char properties = 0x00000002, char value handle = 0x00000014, uuid = FE012A27
handle = 0x00000015, char properties = 0x00000002, char value handle = 0x00000016, uuid = FE012A26
handle = 0x00000017, char properties = 0x00000002, char value handle = 0x00000018, uuid = FE012A28
handle = 0x00000019, char properties = 0x00000002, char value handle = 0x0000001A, uuid = FE012A23
handle = 0x0000001B, char properties = 0x00000002, char value handle = 0x0000001C, uuid = FE012A2A
handle = 0x0000001D, char properties = 0x00000002, char value handle = 0x0000001E, uuid = FE012A50
handle = 0x00000020, char properties = 0x00000002, char value handle = 0x00000021, uuid = FE012A38
handle = 0x00000022, char properties = 0x00000010, char value handle = 0x00000023, uuid = FE012A37
handle = 0x00000026, char properties = 0x00000002, char value handle = 0x00000027, uuid = FE012A19
------->
```

Reading characteristics/descriptors values

gatttool -b <bdaddr> --char-read --handle=0x0000

Example:

```
------->gatttool -b 000016A4093A64 --char-read --handle=0x07
------->
Characteristic value/descriptor: 4C414952445F48524D (LAIRD_HRM)
------->gatttool -b 000016A4093A64 --char-read --handle=0x10
------->
Characteristic value/descriptor: 424C363030 (BL600)
------->gatttool -b 000016A4093A64 --char-read --handle=0x0E
------->
Characteristic value/descriptor: 4C6169726420546563686E6F6C6F67696573 (Laird
Technologies)
------->
```

Writing characteristic/descriptor values

gatttool -b <bdaddr> --char-write --handle=0x00 --value=0100

Example:

```
------->gatttool -b 000016A4093A64 --char-write --handle=0x24 value=0100
------->
Characteristic value was written successfully
------->
```

Enabling notifications/indications

Discovering characteristics and locating value handle of notifiable/indicatable characteristic

Example:

```
------->gatttool -b 000016A4093A64 --characteristics
------->
handle = 0x00000002, char properties = 0x00000022, char value handle = 0x00000003, uuid = FE012A05
handle = 0x00000006, char properties = 0x00000002, char value handle = 0x00000007, uuid = FE012A00
handle = 0x00000008, char properties = 0x00000002, char value handle = 0x00000009, uuid = FE012A01
handle = 0x0000000A, char properties = 0x00000002, char value handle = 0x0000000B, uuid = FE012A04
handle = 0x0000000D, char properties = 0x00000002, char value handle = 0x0000000E, uuid = FE012A29
handle = 0x0000000F, char properties = 0x00000002, char value handle = 0x00000010, uuid = FE012A24
handle = 0x00000011, char properties = 0x00000002, char value handle = 0x00000012, uuid = FE012A25
handle = 0x00000013, char properties = 0x00000002, char value handle = 0x00000014, uuid = FE012A27
handle = 0x00000015, char properties = 0x00000002, char value handle = 0x00000016, uuid = FE012A26
handle = 0x00000017, char properties = 0x00000002, char value handle = 0x00000018, uuid = FE012A28
handle = 0x00000019, char properties = 0x00000002, char value handle = 0x0000001A, uuid = FE012A23
handle = 0x0000001B, char properties = 0x00000002, char value handle = 0x0000001C, uuid = FE012A2A
handle = 0x0000001D, char properties = 0x00000002, char value handle = 0x0000001E, uuid = FE012A50
handle = 0x00000020, char properties = 0x00000002, char value handle = 0x00000021, uuid = FE012A38
handle = 0x00000022, char properties = 0x00000010, char value handle = 0x00000023, uuid = FE012A37
handle = 0x00000026, char properties = 0x00000002, char value handle = 0x00000027, uuid = FE012A19
------->
```

Finding handle of the CCCD descriptor

Example:

```
------->gatttool -b 000016A4093A64 --char-desc --handle=0x23
------->
handle : 0x00000024 uuid : FE012902
------->
```

Enabling notifications (by writing 0100 to CCCD) or indications (by writing 0200 to CCCD)

Example:

```
------->gatttool -b 000016A4093A64 --char-write --handle=0x24 --value=0100 --
listen
------->
Listening for notifications/indications. For hangup, type "stop"
------->
Characteristic value was written successfully
```

Note: If the char handle requires paring, sec-level should be increased.

```
------->gatttool --sec-level=medium -b 000016A4093A64 --char-write --handle=0x24 -
-value=0100 --listen
------->
Listening for notifications/indications. For hangup, type stop
------->
Characteristic value was written successfully
```

Configurability Commands

Setting the device to be pairable or unpairable

```
hciconfig auth
hciconfig noauth
```

Example:

```
------->hciconfig auth
------->
------->hciconfig noauth
------->
```

Setting the device to be discoverable and connectable

hciconfig piscan

Example:

```
------->hciconfig piscan
```

Setting the device to be neither discoverable nor connectable

hciconfig noscan

Example:

```
------->hciconfig noscan
```

Setting the device to be discoverable

hciconfig iscan

Example:

```
------->hciconfig iscan
```

Setting the device to be connectable

hciconfig pscan

Example:

```
------->hciconfig pscan
```

Set/get device name

hciconfig name [name]

Example:

```
------->hciconfig name test
------->
------->hciconfig name
test
------->
```

Set/get the inquiry mode

hciconfig inqmode [inqmode]

Example:

```
-------->hciconfig inqmode 0
-------->
-------->hciconfig inqmode
Inquiry mode: Standard Inquiry
-------->
-------->hciconfig inqmode 1
-------->
-------->hciconfig inqmode
Inquiry mode: Inquiry with RSSI
-------->
hciconfig inqmode 2
-------->
hciconfig inqmode
Inquiry mode: Inquiry with RSSI or Extended Inquiry
-------->
```

Set/get IO capability

hciconfig iocap

Example:

```
-------->hciconfig iocap
IO capability: Just Works
-------->
-------->hciconfig iocap 1
-------->
-------->hciconfig iocap
IO capability: YES/NO
-------->
-------->hciconfig iocap 2
-------->
-------->hciconfig iocap
IO capability: Keyboard
-------->
-------->hciconfig iocap 3
-------->
-------->hciconfig iocap
IO capability: Display
-------->
-------->hciconfig iocap 0
-------->
-------->hciconfig iocap
IO capability: Just Works
-------->
```

Display local OOBdata

hciconfig oobdata

Example:

```
------->hciconfig oobdata
OOB Hash:   BFAE518C2E8752E3B0290A8891E5D97C
Randomizer: 5B5D935E9FD8B1579C808E77F1CA3C20
------->
```

Enable/disable LE advertisements

hciconfig leadv

hciconfig noleadv

Example:

```
------->hciconfig leadv
------->
------->hciconfig noleadv
------->
```

Get/set the inquiry transmit power level

hciconfig inqtpl [inqtpl]

Example:

```
------->hciconfig inqtpl
Inquiry transmit power level : 4
------->
------->hciconfig inqtpl 20
------->
------->hciconfig inqtpl
Inquiry transmit power level : 20
------->
------->hciconfig inqtpl -20
------->
------->hciconfig inqtpl
Inquiry transmit power level : -20
------->
```

Get/set the LE transmit power level

hciconfig letpl

Example:

```
------->hciconfig letpl 4
------->
------->hciconfig letpl
LE transmit power level : 4
------->hciconfig letpl 20
Invalid power value. Valid range is from -20 to 8 dBm
------->
```

Display version information

hciconfig version

Example:

```
------->hciconfig version
App Version: smartZ Version 1.0
Device Name: LAIRD BT900
Firmware: 9.1.2.0
Manufacturer: Laird Technologies
------->
```

FURTHER INFORMATION

Further information relating to firmware and the use of UWTerminal is available from the Laird website at the Laird Embedded Wireless Support Center BT900 Product Page .

REVISION HISTORY

Revision	Date	Description	Initiated By
1.0	06 Nov 2014	Initial Release	Jonathan Kaye
1.1	20 Jan 2015	Updated links to Laird support site. Added Revision History	Sue White
1.2	9 Nov 2015	Added section on iocap	Jonathan Kaye