

WL18xx Bluetopia PM Bluetooth RF Testing

Contents

Introduction

Connection-less RF tests

- Continuous TX
- Tester Packet TX RX
- Continuous RX
- BER meter test
 - BER Test - Between Two Wilink Devices
 - BER Test - Between Wilink (RX) and Litepoint iQ2010 (TX)

BT SIG RF Testmode

BLE testing

- BT LE Test Mode
- Connection Test

Conversion of HCI Commands to SS1Tool or hcitool format

- Examples of SS1Tool or Enhanced Bluetopia PM Test Format conversion
 - First start by disabling the sleep protocol
 - Connection-less RF tests
 - Continuous TX
 - Packet TX/RX
 - Continuous RX
 - RX BER meter test
 - BT SIG RF Testmode
 - BLE testing

Introduction

PLT tests are used at the production line after designing the board with the Wilink8 device. This is to sanity check the board works properly and detects any major design issues and manufacturing defects.

Basic RF commands like continuous TX/RX are required as well as part of the Regulatory certification process.

The Connection-less RF tests modes are connection-less RF spectrum tests which uses a spectrum analyzer for verification.

The BT SIG RF Testmode is meant for connecting with a Bluetooth tester where the Wilink8 device is controlled over the LMP (Link Management Protocol)

There are two ways of issuing these test commands to Wilink8 device

- From a PC that runs HCITester. The commands in sections 2 to 4 are run in HCITester
- On the target using either the BluetopiaPM tool **SS1Tool** or the Bluez tool **hcitool** . These commands are used the same way and section 5 shows how to run the tests. **NOTE** : You will have to run `./SS1BTM &` before running `./SS1Tool cmd`

NOTE : The firmware initialization script must be run prior to any scripts. The BTS file that is part of the BT Service Pack (referenced below) is the initialization script. To use it in HCITester, it needs to be modified and converted into text format. For information on this process, follow the instructions described in the *Firmware Initialization Script* section of the [HCITester User's Guide](#).

Useful Links:

- [TI Wireless Tools \(HCITester software included\)](http://www.ti.com/tool/WILINK-BT_WIFI-WIRELESS_TOOLS) (http://www.ti.com/tool/WILINK-BT_WIFI-WIRELESS_TOOLS)
- [TI Bluetooth Service Pack \(XML file - HCItester, BTS file - initialization script downloads\)](http://www.ti.com/tool/WL18XX-BT-SP) (<http://www.ti.com/tool/WL18XX-BT-SP>)
- [HCITester User's Guide](#)

Connection-less RF tests

The Connection-less RF tests are connection-less RF spectrum tests which uses a spectrum analyzer for verification.

Note: The Wilink_X device must be power cycled when switched between different FCC modes.

Continuous TX

This is a non-packet continuous transmission with either GFSK (BR), $\pi/4$ -DQPSK (2-EDR) or 8DPSK (3-EDR).

Steps and the HCI commands to put the device in the constant TX mode are:

Step 1:

run init script.

Step 2:

```
#set TxMode: 0x0 = un-modulated Single carrier, 0x1 = GFSK, 0x2 = EDR2, 0x3 = EDR3, 0x4 = BLE, 0x5 = ANT
TxMode = 1
```

Step 3:

```
Send_HCI_VS_DRPb_Enable_RF_Calibration_Enhanced 0xFDFB, 0x1, 0xFF, 0x00000000, 0x01
Wait_HCI_Command_Complete_VS_DRPb_Enable_RF_Calibration_Enhanced_Event 5000, any, HCI_VS_DRPb_Enable_RF_Calibration_Enhanced, 0x00
```

Step 4:

```
Send_HCI_VS_DRPb_Tester_Con_TX 0xFDCA, 2402, TxMode, 0, 7, 0x00000000, 0x00000000 // For TxMode relevant value please see step2
Wait_HCI_Command_Complete_VS_DRPb_Tester_Con_TX_Event 5000, any, HCI_VS_DRPb_Tester_Con_TX, 0x00
```

The parameters for the HCI_VS_DRPb_Tester_Con_TX are:

1. Opcode

- 0xFDCA

2. Frequency

- f=2402 ~ 2479

3. Modulation scheme

- 0=CW
- 1=GFSK (BR)
- 2= π /4-DQPSK (2-EDR)
- 3=8DPSK (3-EDR)
- 4=BLE
- 5=ANT

4. Test pattern

- 0=PN9
- 1=PN15
- 2=Z0Z0
- 3=all 1
- 4=all 0
- 5=F0F0
- 6=FF00

5. Power level index

- 0-7 where 0 is the lowest and 7 is the highest

6. Internal setting

- 0x00000000

7. Internal setting

- 0x00000000

To exit Continuous TX

The HCI commands for Stopping continuous Tx mode are:

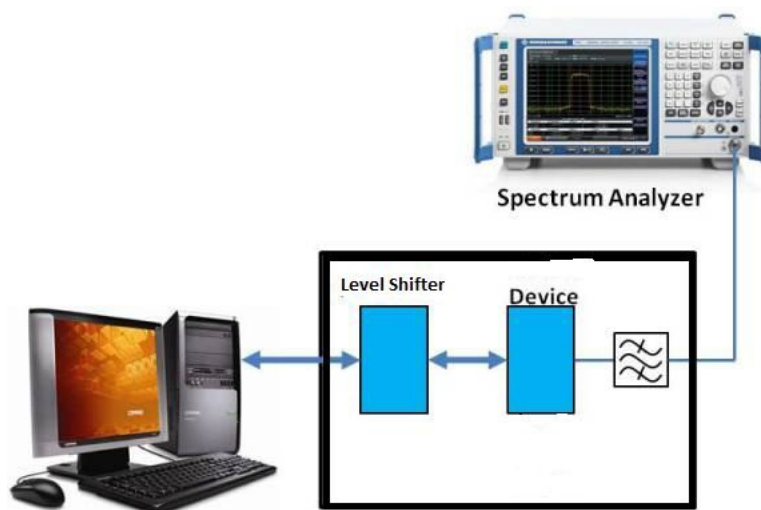
TX Stop :

```
Send_HCI_VS_Write_Hardware_Register 0xFF01, 0x20014044, 0x0010
Wait_HCI_Command_Complete_VS_Write_Hardware_Register_Event 5000, any, HCI_VS_Write_Hardware_Register, 0x00

Send_HCI_VS_Write_Hardware_Register 0xFF01, 0x20014048, 0x0014
Wait_HCI_Command_Complete_VS_Write_Hardware_Register_Event 5000, any, HCI_VS_Write_Hardware_Register, 0x00
```

Stop PN Generator:

```
Send_HCI_VS_Write_Hardware_Register 0xFF01, 0x2001900C, 0x0000
Wait_HCI_Command_Complete_VS_Write_Hardware_Register_Event 5000, any, HCI_VS_Write_Hardware_Register, 0x00
```

Tester Packet TX RX

Typical test setup for Continuous TX mode

```
Send_HCI_VS_DRPb_Tester_Packet_TX_RX 0xFDCC, 0, 3, 2402, 0xFFFF, 2, 0, 27, 7, 1, 0x01FF
Wait_HCI_Command_Complete_VS_DRPb_Tester_Packet_TX_RX_Event 5000, any, HCI_VS_DRPb_Tester_Packet_TX_RX, 0x00
```

The parameters for the HCI_VS_DRPb_Tester_Packet_TX_RX are:

1. Opcode

- 0xFDCC

2. ACL TX Packet Type

- 0x0 - DM1
- 0x1 - DH1
- 0x2 - DM3
- 0x3 - DH3
- 0x4 - DM5
- 0x5 - DH5
- 0x6 - 2-DH1
- 0x7 - 2-DH3
- 0x8 - 2-DH5
- 0x9 - 3-DH1
- 0xA - 3-DH3
- 0xB - 3-DH5

3. Frequency Mode

- 0x00 - Hopping
- 0x03 - Single Frequency

4. Tx Single Frequency

- 2402 ~ 2480
- 0xFFFF - No Tx

5. Rx Single Frequency

- 2402 ~ 2480
- 0xFFFF - No Rx

6. Test pattern

- 0x0= All 0
- 0x1= All 1
- 0x2= ZOZO (0101)
- 0x3= FOFO (11110000)
- 0x4= Ordered
- 0x5= PRBS9 Random

7. Extended features

- 0 - disabled
- 1 - Enabled

8. Packet data length

- 0-17 -> DM1, 0-27 -> DH1, 0-121 -> DM3, 0-183 -> DH3, 0-224 -> DM5, 0-339 -> DH5

9. Power Level

- 0-7 where 0 is the lowest and 7 is the highest

10. Diabie Whitning

- 0x1 - Disabled
- 0x0 - Enabled

11. PRBS9 Init Value

- 0x01FF

Continuous RX

Step 1:

```
run init script.
```

Step 2:

```
Send_HCI_VS_DRPb_Enable_RF_Calibration_Enhanced 0xFDFB, 0x1, 0xFF, 0x00000000, 0x01
Wait_HCI_Command_Complete_VS_DRPb_Enable_RF_Calibration_Enhanced_Event 5000, any, HCI_VS_DRPb_Enable_RF_Calibration_Enhanced, 0x00
```

Step 3: To turn on the receiver portion of the chip the following command needs to be sent

```
#0x2 = Open Loop , 0x1 = Close Loop
Send_HCI_VS_DRPb_Tester_Con_RX 0xFDCB, 2402, 0x01, 0x01
Wait_HCI_Command_Complete_VS_DRPb_Tester_Con_RX_Event 5000, any, HCI_VS_DRPb_Tester_Con_RX, 0x00
```

The parameters for the HCI_VS_DRPb_Tester_Con_RX are:

1. Opcode

- 0xFDCB

2. Frequency

- f=2402 ~ 2480

3. Internal Setting (Rx Mode)

- 0- Connection mode
- 1- ADPLL Closed loop
- 2 – ADPLL Open loop
- 3 – Scan mode

4. Modulation Type

- 0=CW
- 1=GFSK (BR)
- 2= $\pi/4$ -DQPSK (2-EDR)
- 3=8DPSK (3-EDR)

The setup would be the same as for Continuous TX.

BER meter test

A bit error rate (BER) meter test of the DUT is designed primarily for testing of BT sensitivity by the BT sig spec (-70dBm per BT standard).

The context below describes the **HCI_VS_DRPb_BER_Meter_Start** command and parameters used in this example:

1. Op code: 0xFD8B

2. Frequency Channel

- 0-39: $f=2402+(2*i)$ MHz
- 40-78: $f=2403+2(i-40)$ MHz

3. Reserved (0x00)

4. BD Address

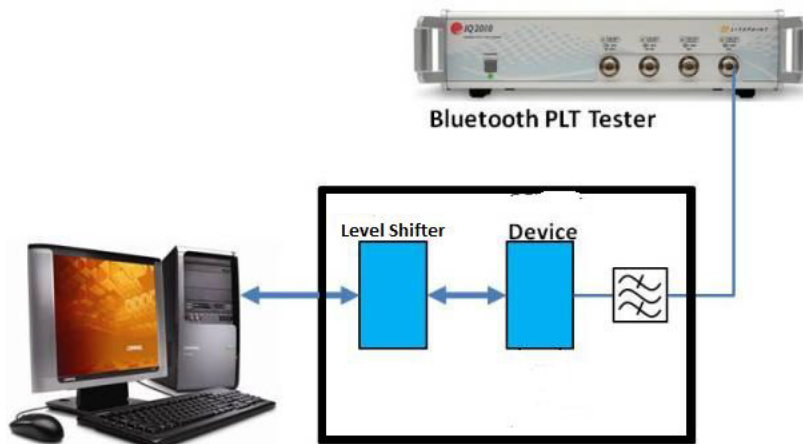
- Is expressed as Little Endian
- Must match the MAC (BD) address as specified by the transmitter
- Actual MAC address of 12:34:56:78:12:34 would be 34:12:78:56:34:12

5. LT Address (0x01) - address of the device in the piconet (not applicable, set to 1)

6. ACL TX packet type

- 0=DM1
- 1=DH1
- 2=DM3
- 3=DH3
- 4=DM5
- 5=DH5
- 6=2-DH1
- 7=2-DH3
- 8=2-DH5
- 9=3-DH1
- A=3-DH3
- B=3-DH5

7. Packet length



WiLink_X Bluetooth PLT mode test setup

- DM1: 0-17
- DH1: 0-27
- DM3: 0-121
- DH3: 0-183
- DM5: 0-224
- DH5: 0-339

8. Number of packets to be used for the BER test (0x0000-0xFFFF) [See below chart and explanation]

9. PRBS initialize (0x1FF)

10. Poll period = 0x1

- Poll period of 5 means that every fifth packet received is used in the BER calculation (longer test time)
- Poll period of 1 means every packet received is used in the BER calculation (shortest test time)

Per the BT spec, if 1.6Mbits are sent, the BER must be less than 10^{-5} in order to pass, which is equivalent to 16 bit errors in this case.

Because there are different ACL packet types that are comprised of different packet lengths, the number of packets transmitted must be changed to accommodate the different packet sizes in order to still send 1.6Mbits. The chart below illustrates the different packet types and number of packets needed to conduct a BER test as per the BT 4.1 spec:

Type	User Payload (bytes)	Num packets to send for BER test
DM1	0-17	11765
DH1	0-27	7408
DM3	0-121	1653
DH3	0-183	1093
DM5	0-224	893
DH5	0-339	590
AUX1	0-29	6897
2-DH1	0-54	3704
2-DH3	0-367	545
2-DH5	0-679	295
3-DH1	0-83	2410
3-DH3	0-552	363
3-DH5	0-1021	196

The BER test can be achieved in two different configurations, either:

1. Two WiLink devices (one designated as the TX and the other as the RX) OR
2. One WiLink device and 1 Litepoint IQ2010 Tester (WiLink as the RX and Litepoint as the TX)

The sections below describe these configurations:

BER Test - Between Two WiLink Devices

Note: Observe maximum RF input power during PLT test (-5dBm for BR and -10dBm for EDR)

The BER meter test can be performed using two WiLink_X devices. One of the devices will be classified as the TX board, and the other as the RX board.

Step 1:

Run the INIT script on each board.

NOTE: You can have two instances of HCITester running, as long as the COM ports are configured correctly.

Step 2:

Start the BER function on the RX board using the following script example:

```

$!10
count = 0

#DH5, 339 packet size x 8 bits x 590 packets = 1.6Mbits
#BD address = that of the TX board
Send_HCI_VS_DRPp_BER_Meter_Start 0xFD8B, 0, 0, 0x222222222222, 1, 0x5, 339, 590, 0x1FF, 0x1
Wait_HCI_Command_Complete_VS_DRPp_BER_Meter_Start_Event 5000, any, HCI_VS_DRPp_BER_Meter_Start, 0x00

#
$!11
Sleep 100
count++

Send_HCI_VS_DRP_Read_BER_Meter_Status 0xFD5B
Wait_HCI_Command_Complete_VS_DRP_Read_BER_Meter_Status_Event 5000, any, HCI_VS_DRP_Read_BER_Meter_Status, 0x00, &syncd, &finished, any, &total_bits, &bit_errors, &packets
if finished== 0 then
    if syncd == 0 then
        log"not synced yet %d count",count
        #Fail "no SYNC"
    else
        log"not finished yet %d packets, count %d",packets,count

```

```

endif
goto $!11
else
BER = 1000000*bit_errors/total_bits
log"BER is %d e-6 (%d bits out of %d)",1000000*bit_errors/total_bits,bit_errors,total_bits
if BER > 100 then
Fail "BER is too high"
else
pause "Finished"
endif
endif
endif
#pause"
goto $!10

```

Step 3:

The BD addresses between the TX and RX boards need to match in order for the BER Meter to function. Set the BD_ADDR on the TX board using the following command:

```

BD_ADDR="222222222222"
Send_HCI_VS_Write_BD_ADDR 0xFC06, BD_ADDR
Wait_HCI_Command_Complete_VS_Write_BD_ADDR_Event 5000, any, HCI_VS_Write_BD_ADDR, 0x00

```

Step 4:

Send TX packets with the TX board using the following commands:

```

Send_HCI_VS_DRPb_Enable_RF_Calibration_Enhanced 0xFDFB, 1, 0xFF, 0x00000000, 0x01
Wait_HCI_Command_Complete_VS_DRPb_Enable_RF_Calibration_Enhanced_Event 5000, any, HCI_VS_DRPb_Enable_RF_Calibration_Enhanced, 0x00
sleep 1000

# Sent Tx Packets
Send_HCI_VS_DRPb_Tester_Packet_TX_RX 0xFDCC, 0x5, 3, 2402, 0xFFFF, 5, 0, 339, 5, 1, 0x01FF
Wait_HCI_Command_Complete_VS_DRPb_Tester_Packet_TX_RX_Event 5000, any, HCI_VS_DRPb_Tester_Packet_TX_RX, 0x00

```

BER Test - Between Wilink (RX) and Litepoint iQ2010 (TX)

In this configuration, the Wilink (RX) device will run the BER meter script as the Litepoint iQ2010 (TX) sends the packets with an embedded BD address.

The waveforms for different packet types with an embedded BD address (0x11111110000) can be downloaded [here \(https://gforge.ti.com/gf/project/wilink/docman/WiLink8/IQ%20Waveforms/\)](https://gforge.ti.com/gf/project/wilink/docman/WiLink8/IQ%20Waveforms/).

Before the BER meter test is initiated, the Litepoint iQ2010 needs to be configured following these steps:

1. Open iQSignal (Litepoint Software).
2. **Connect** Litepoint iQ2010 test equipment to the PC using default settings.
3. Click **Bluetooth -> Vector Signal Generator**.
4. Click **Open Generator File...** and open the *1dh5_11111110000.mod* file that was downloaded above.
5. VSG Control = continuous
6. RF Channel = 0/2402
7. Trigger Type = Free Run
8. Connect an antenna to the RF2 port on the Litepoint iQ2010 Test Equipment.

NOTE: If using the WL18XXMODCOM8I + SDMMCADAPTER boards, you can use the on-board antenna.

The Litepoint iQ2010 and hardware are now properly configured. Do not start the RF signal until you have followed the steps below on the Wilink (RX) board.

Step 1:

```

Run the INIT script Wilink board.

```

Step 2:

Start the BER function on the Wilink (RX) board using the following script example:

```

$!10
count = 0
#DH5, 339 packet size x 8 bits x 590 packets = 1.6Mbits
#BD address = the one embedded in the IQ waveform
Send_HCI_VS_DRPb_BER_Meter_Start 0xFD8B, 0, 0, 0x11111110000, 1, 0x5, 339, 590, 0x1FF, 0x1
Wait_HCI_Command_Complete_VS_DRPb_BER_Meter_Start_Event 5000, any, HCI_VS_DRPb_BER_Meter_Start, 0x00

#
$!11
sleep 2000
count++

Send_HCI_VS_DRP_Read_BER_Meter_Status 0xFD5B
Wait_HCI_Command_Complete_VS_DRP_Read_BER_Meter_Status_Event 5000, any, HCI_VS_DRP_Read_BER_Meter_Status, 0x00, &syncd, &finished, any, &total_bits, &bit_errors, &packets
if finished== 0 then
if syncd == 0 then
log"not syncd yet %d count",count
#Fail "no SYNC"
else
log"not finished yet %d packets, count %d",packets,count
endif
goto $!11
else
BER = 1000000*bit_errors/total_bits

```

```

log"BER is %d e-6 (%d bits out of %d)",1000000*bit_errors/total_bits,bit_errors,total_bits
if BER > 100 then
    Fail "BER is too high"
else
    pause "Finished"
endif
endif
#pause""
goto $110

```

Step 3:

Start the RF signal on the iQSignal GUI by clicking **RF ON/OFF** and confirm that the indication text turns green.

Step 4:

Let the BER script finish on the Wilink board, and observe the results in the HCITester Trace Log.

BT SIG RF Testmode

The BT SIG RF Testmode is meant for connecting with a Bluetooth tester where the Wilink_X is controlled over the LMP (Link Management Protocol). The procedure for enabling the Wilink_X for BT SIG mode is:

1. Proper device power-up
2. Load the correct service pack
3. Load the DUT script (three HCI commands)

Once the device (DUT) is in test mode, the BT tester will, through the RF connection (LMP), take control of the DUT.

The HCI commands to put the device in test RF SIG mode are:

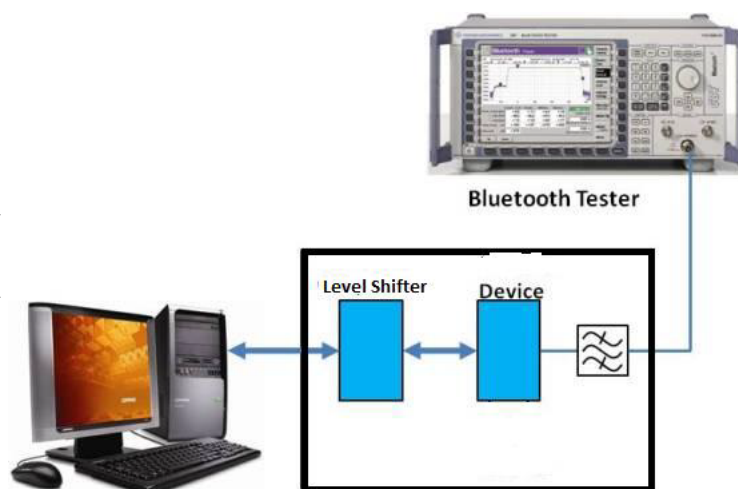
```

HCI_Enable_Device_Under_Test_Mode
HCI_Set_Event_Filter 0x02, 0x00, 0x03
HCI_Write_Scan_Enable 0x03

```

This script will make the device visible and connectable (HCI_Write_Scan_Enable), auto-accept all connections (HCI_Set_Event_Filter) and put the DUT in test mode (HCI_Enable_Device_Under_Test_Mode).

Once the proper sequence has been completed (first loading the BT service pack and then the DUT script), then the BT tester will take control of the device through the RF link (LMP).



Wilink_X Bluetooth RF SIG mode test setup

BLE testing

BT LE Test Mode

Below are the HCI commands from the Spec to place the device in BT LE test mode :

Transmitter Test

```

Send_HCI_BLE_Transmitter_Test 0x201e, 0x00, 0x00, 0
Wait_HCI_Command_Complete_BLE_Transmitter_Test_Event 5000, any, HCI_BLE_Transmitter_Test, 0x00

Send_HCI_BLE_Test_End 0x201f
Wait_HCI_Command_Complete_BLE_Test_End_Event 5000, any, HCI_BLE_Test_End, 0x00, 0

```

Receiver Test

```

Send_HCI_BLE_Receiver_Test 0x201d, 0x00
Wait_HCI_Command_Complete_BLE_Receiver_Test_Event 5000, any, HCI_BLE_Receiver_Test, 0x00

Send_HCI_BLE_Test_End 0x201f
Wait_HCI_Command_Complete_BLE_Test_End_Event 5000, any, HCI_BLE_Test_End, 0x00, 0

```

To disable RF Calibration

```

Send_HCI_VS_DRPb_Enable_RF_Calibration_Enhanced 0xFDFB, 0x1, 0xFF, 0x80000000, 0x00
Wait_HCI_Command_Complete_VS_DRPb_Enable_RF_Calibration_Enhanced_Event 5000, any, HCI_VS_DRPb_Enable_RF_Calibration_Enhanced, 0x00

```

Connection Test

This example will create a BLE connection between two WLink8 boards, but only if the device's BD address is on the 'White List'- meaning it will not connect with devices that are not on the White List. It is optional to put the Slave to sleep and have it wake up to a connection request from a BD address on it's White List.

The hardware used in this example is two WLink8 boards (one as the "slave" and the other as the "master") and a PC running two instances of the HCITester tool (one for each board).

Step 1:

```
Run the INIT script on both the Slave board and the Master board.
```

Step 2:

Write a known BD address on the Slave, so that the master knows which device to connect to.

```
Send_HCI_VS_Write_BD_ADDR 0xFC06,"0708090A0B0D"
Wait_HCI_Command_Complete_VS_Write_BD_ADDR_Event 5000, any, HCI_VS_Write_BD_ADDR, 0x00
```

Step 3:

On the Slave: Clear the device white list (to remove previous data) and then add the Master's BD address to the White List.

```
## CLEAR DEVICE WHITE LIST
Send_HCI_LE_Clear_White_List
Wait_HCI_Command_Complete_LE_Clear_White_List_Event 5000, any, HCI_LE_Clear_White_List, 0x00

## ADD DEVICE ADDRESS : PUBLIC TO WHITE LIST
Send_HCI_LE_Add_Device_To_White_List 0x00, "010203040506"
Wait_HCI_Command_Complete_LE_Add_Device_To_White_List_Event 5000, any, HCI_LE_Add_Device_To_White_List, 0x00
```

Step 4:

On the Slave: Configure the advertising parameters to match the BD address of the Master, the channel to use and set it to only accept scans from the White List.

```
Send_HCI_LE_Write_Advertising_Parameters 0x120, 0x150, 0x00, 0x00, 0x00, "010203040506", 7, 0x03
Wait_HCI_Command_Complete_LE_Write_Advertising_Parameters_Event 5000, any, HCI_LE_Write_Advertising_Parameters, 0x00
```

Step 5:

Enable advertising on the Slave.

```
Send_HCI_LE_Write_Advertise_Enable 0x01
Wait_HCI_Command_Complete_LE_Write_Advertise_Enable_Event 5000, any, HCI_LE_Write_Advertise_Enable, 0, Any
```

NOTE: At this point the Slave may be put to sleep by utilizing the sleep buttons on HCITester.

Step 6:

Write a known BD address to the Master, matching that of which was used in Step 4.

```
Send_HCI_VS_Write_BD_ADDR 0xFC06,"010203040506"
Wait_HCI_Command_Complete_VS_Write_BD_ADDR_Event 5000, any, HCI_VS_Write_BD_ADDR, 0x00
```

Step 7:

Clear the device White List and add the Slave's BD address to it:

```
Send_HCI_LE_Clear_White_List
Wait_HCI_Command_Complete_LE_Clear_White_List_Event 5000, any, HCI_LE_Clear_White_List, 0x00

Send_HCI_LE_Add_Device_To_White_List 0x00, "0708090A0B0D"
Wait_HCI_Command_Complete_LE_Add_Device_To_White_List_Event 5000, any, HCI_LE_Add_Device_To_White_List, 0x00
```

Step 8:

Configure the Scan parameters to allow advertising from the White List:

```
Send_HCI_LE_Write_Scan_Parameters 0x00, 0x01000, 0x1000, 0x00, 0x01
Wait_HCI_Command_Complete_LE_Write_Scan_Parameters_Event 5000, any, HCI_LE_Write_Scan_Parameters,
```

Step 9:

Create a connection using a White List and the Slave'd BD address:

```
Send_HCI_LE_Create_Connection 0x0190, 0x0190, 0x01, 0x00, "0708090A0B0C", 0x00, 800, 800, 0, 0x0c80, 0x0000, 0x0000
```

Conversion of HCI Commands to SS1Tool or hcitool format

- SS1Tool is a tool used to send HCI commands to the WLink8, when using BluetopiaPM Bluetooth stack.
- hcitool is a tool used to send HCI commands to the WLink8, when using Bluez stack.

Both tools use the same syntax so in the examples the commands SS1Tool and hcitool are interchangeable.

For a focus on hci command that are not readily available. Here is the format to use for both tools as an example.

```
Send_HCI_VS_DRPb_Tester_Con_TX 0xFDCA, 2402, 0, 0, 7, 0x00000000, 0x00000000
SS1Tool cmd <ogf> <ocf> [Parameters]
```


or

```
Send_HCI_VS_DRPb_Tester_Con_TX 0xFDCA, 2402, 0, 0, 7, 0x00000000, 0x00000000
hcitool cmd <ogf> <ocf> [Parameters]
```

<ogf> and <ocf>, along with parameters are all hexadecimal values.

Every HCI commands have an opcode; this opcode is a combination of the <ogf> and <ocf>.

The same format applies to HCI and HCI_VS commands. See the snapshot below.

For instance, given this command: # Enable Page and inquiry scan:

```
Send_HCI_Write_Scan_Enable 0x03
```

This command has an opcode of 0x0c1a

```
0x0c1a -> 0000 1100 0001 1010
OCF (10 bit opcode cmd field) -> 00 0001 1010 -> 0x01a
OGF (6 bit opcode group field) -> 0000 11 -> 0x03
```

Thus: SS1Tool cmd 0x03 0x01a 0x03 -> Enable page and inquiry scan

```
FDCA -> 1111 11 01 1100 1010
0011 1111 -> 3f
0001 1100 1010 -> 1ca
SS1Tool cmd 3f 1ca
```

Table 3. HCI Command Packet Structure

First					Last
OCF	OGF	Length	Para0	Para1	-----
OCF:	10-bit opcode command field, range: 0x00–0x3F (0x3E reserved for Bluetooth logo testing and 0x3F reserved for VS debug commands)				
OGF:	6-bit opcode group field				
Length:	Length of all parameters in this packet measured in bytes (total length of parameters, not number of parameters)				
Para0 – ParaN:	Specific parameters are associated with each command. These parameters and the size of each parameter are defined for each command. The size of each parameter is an integer number of bytes.				

Examples of SS1Tool or Enhanced Bluetopia PM Test Format conversion

First start by disabling the sleep protocol

HCI Tester command format

```
Send_HCI_VS_Sleep_Mode_Configurations 0xFD0C, 0, 0, 0xFF, 0xFF, 0xFF, 0xFF, 100
```

Bluetopia PM SS1 Tool format

```
SS1Tool cmd 0x3f 0x10c 0x00 0x00 0xff 0xff 0xff 0xff 0x64
```

Connection-less RF tests

The Connection-less RF tests are connection-less RF spectrum tests which uses a spectrum analyzer for verification.

Test that the commands are being sent to the device.

HCI Tester command format

```
Send_HCI_Inquiry "9E8B33", 0x04, 0x00
```

Bluetopia PM SS1 Tool format

```
SS1Tool cmd 0x01 0x001 0x33 0x8b 0x9e 0x04 0x00
```

Continuous TX

This is a non-packet continuous transmission with either GFSK (BR), $\pi/4$ -DQPSK (2-EDR) or 8DPSK (3-EDR). It can also be used with BLE and ANT modulations. Please refer to the [Continuous_TX](#) description above.

The HCI commands to put the device in the Continuous TX mode (after service pack has been loaded) are:

HCI Tester command format

```
Send_HCI_Write_Scan_Enable 0x00
Send_HCI_VS_DRPb_Enable_RF_Calibration_Enhanced 0xFDFB, 1, 0xFF, 0x00000000, 0x01
Send_HCI_VS_DRPb_Tester_Con_TX 0xFDCA, 2402, 0x1, 0, 7, 0x00000000, 0x00000000
```

Bluetopia PM SS1 Tool format

```
SS1Tool cmd 0x03 0x001a 0x00
SS1Tool cmd 0x3f 0x01fb 0x01 0xff 0x00 0x00 0x00 0x00 0x01
SS1Tool cmd 0x3f 0x01ca 0x62 0x09 0x01 0x00 0x07 0x00 0x00 0x00 0x00 0x00 0x00 0x00
```

Note: Having active Page and/or Inquiry scans running in parallel during RF tests, specially during Continuous TX tests, will cause the test signal to appear for a short period of time and the test signal will disappear on the next scan instance. This will appear as a burst rather than the Continuous TX. The BluetopiaPM enables both Page and Inquiry scans by default when the WL18xx device initializes. In order to avoid loosing the test signal after a short burst due to scan instances, the HCI_Write_Scan_Enable command must be issued before running the Continuous TX test using SS1Tool/hcitol

The HCI commands to stop the Continuous TX mode test

HCI Tester command format

```
Send_HCI_VS_Write_Hardware_Register 0xFF01, 0x20014044, 0x0010 #TX Stop
Send_HCI_VS_Write_Hardware_Register 0xFF01, 0x20014048, 0x0014 #TX Stop
Send_HCI_VS_Write_Hardware_Register 0xFF01, 0x2001900C, 0x0000 #Stop PN Generator
```

Bluetopia PM SS1 tool format

```
SS1Tool cmd 0x3f 0x0301 0x44 0x40 0x01 0x20 0x10 0x00
SS1Tool cmd 0x3f 0x0301 0x48 0x40 0x01 0x20 0x14 0x00
SS1Tool cmd 0x3f 0x0301 0x0c 0x90 0x01 0x20 0x00 0x00
```

Packet TX/RX

To enable the continuous packet transmission/receive the following commands need to be sent after the loading of the service pack:

HCI Tester command format

```
Send_HCI_VS_DRPb_Enable_RF_Calibration_Enhanced 0xFDFB, 0x1, 0xFF, 0x00000000, 0x01
Send_HCI_VS_DRPb_Tester_Packet_TX_RX 0xFDCC, 0, 3, 2402, 0xFFFF, 5, 0, 27, 7, 1, 0x01FF
```

Bluetopia PM SS1 tool format

```
SS1Tool cmd 0x3f 0x01fb 0x01 0xff 0x00 0x00 0x00 0x00 0x01
SS1Tool cmd 0x3f 0x01cc 0x00 0x03 0x62 0x09 0xff 0xff 0x05 0x00 0x1b 0x00 0x07 0x01 0xff 0x01
```

Continuous RX

To turn on the receiver portion of the chip the following command needs to be sent after the loading of the service pack:

HCI Tester command format

```
Send_HCI_VS_DRPb_Enable_RF_Calibration_Enhanced 0xFDFB, 0x1, 0xFF, 0x00000000, 0x01
Send_HCI_VS_DRPb_Tester_Con_RX 0xFDCB, 2402, 0x01, 0x01
```

Bluetopia PM SS1 tool format

```
SS1Tool cmd 0x3f 0x01fb 0x01 0xff 0x00 0x00 0x00 0x00 0x01
SS1Tool cmd 0x3f 0x01cb 0x62 0x09 0x01 0x01
```

RX BER meter test

A bit error rate (BER) meter test of the DUT is designed primarily for testing of BT sensitivity by the BT sig spec (-70dBm per BT standard).

HCI Tester command format

```
Send_HCI_VS_DRPb_BER_Meter_Start 0xFD8B, 0x00, 0x00, 0x341278563412, 0x01, 0x01, 0x001b, 0x03e8, 0x01ff, 0x01
```

Bluetopia PM SS1 tool format

```
SS1Tool cmd 0x3f 0x18B 0x00 0x00 0x12 0x34 0x56 0x78 0x12 0x34 0x01 0x01 0x1b 0x00 0xe8 0x03 0xff 0x01 0x01
```

Delay 5000 (ms)

HCI Tester command format

```
Send_HCI_VS_DRP_Read_BER_Meter_Result 0xFD13
```

Bluetopia PM SS1 tool format

```
SS1Tool cmd 0x3f 0x113
```

The procedure is to send the first VS HCI command (HCI_VS_DRPb_BER_Meter_Start) which starts the BER function in the Wilink_X device.

A delay equal to or longer than the time it takes to read the number of packets specified is needed before the second VS HCI command (HCI_VS_DRP_Read_BER_Meter_Result is sent to read back (event) the BER test result.

The frequency channel, BD address, packet type and #bytes/packet must match that of the transmitter.

If another Wilink_X unit is used for the BT transmitter in the PLT BER test, then the following two commands needs to be sent after the successful loading of the BT service pack:

HCI Tester command format

```
Send_HCI_VS_Write_BD_ADDR 0xFC06, "123456781234"
Send_HCI_VS_DRPb_Tester_Packet_TX_RX 0xFDCC, 0x01, 0x03, 0x0962, 0xffff, 0x05, 0x00, 0x001b, 0x07, 0x01, 0x01ff
```

Bluetopia PM SS1 tool format

```
SS1Tool cmd 0x3f 0x006 0x34 0x12 0x78 0x56 0x34 0x12
SS1Tool cmd 0x3f 0x1CC 0x01 0x03 0x62 0x09 0xff 0xff 0x05 0x00 0x1b 0x07 0x01 0xff 0x01
```

Where the first VS HCI command sets the BD address of the TX device, as the DUT device needs to know the specific BD address of the transmitter. This command HCI_VS_DRPb_Tester_Packet_TX_RX sets the transmitter characteristics.

BT SIG RF Testmode

The BT SIG RF Testmode is meant for connecting with a Bluetooth tester where the Wilink_X is controlled over the LMP (Link Management Protocol).

The procedure for enabling the Wilink_X for BT SIG mode is:

1. Proper device power-up
2. Load the correct service pack
3. Load the DUT script (three HCI commands)

Once the device (DUT) is in test mode, the BT tester will, through the RF connection (LMP), take control of the DUT.

The HCI commands to put the device in test RF SIG mode are:

HCI Tester command format

```
HCI_Enable_Device_Under_Test_Mode
HCI_Set_Event_Filter 0x02, 0x00, 0x03
HCI_Write_Scan_Enable 0x03
```

Bluetopia PM SS1 tool format

```
SS1Tool cmd 0x06 0x03
SS1Tool cmd 0x03 0x05 0x02 0x00 0x03
SS1Tool cmd 0x03 0x1a 0x03
```

BLE testing

Transmitter Test

HCI Tester command format

```
Send_HCI_VS_DRPb_Enable_RF_Calibration_Enhanced 0xFDFB, 0x1, 0xFF, 0x80000000, 0x00 //disable RF Calibration
Send_HCI_BLE_Transmitter_Test 0x201e, 0x00, 0x00, 0
Send_HCI_BLE_Test_End 0x201f
```

Bluetopia HCI tool format:

```
SS1Tool cmd 0x3f 0x01fb 0x01 0xff 0x00 0x00 0x00 0x00 //disable RF Calibration
SS1Tool cmd 0x08 0x001e 0x00 0x00 0x00
SS1Tool cmd 0x08 0x001f
```

Receiver Test

HCI Tester command format

```
Send_HCI_VS_DRPb_Enable_RF_Calibration_Enhanced 0xFDFB, 0x1, 0xFF, 0x8000000, 0x00 //disable RF Calibration
Send_HCI_BLE_Receiver_Test 0x201d, 0x00
Send_HCI_BLE_Test_End 0x201f
```

Bluetopia HCI tool format:

```
SSITool cmd 0x3f 0x01fb 0x01 0xff 0x00 0x00 0x00 0x00 0x00 //disable RF Calibration
SSITool cmd 0x08 0x001d 0x00
SSITool cmd 0x08 0x001f
```

<p>Keystone=</p> <ul style="list-style-type: none"> For technical support on MultiCore devices, please post your questions in the C6000 MultiCore Forum For questions related to the BIOS MultiCore SDK (MCSDK), please use the BIOS Forum <p>Please post only comments related to the article WL18xx Bluetopia PM Bluetooth RF Testing here.</p>	<p>C2000=For technical support on the C2000 please post your questions on The C2000 Forum. Please post only comments about the article WL18xx Bluetopia PM Bluetooth RF Testing here.</p>	<p>DaVinci=For technical support on DaVinciplease post your questions on The DaVinci Forum. Please post only comments about the article WL18xx Bluetopia PM Bluetooth RF Testing here.</p>	<p>MSP430=For technical support on MSP430 please post your questions on The MSP430 Forum. Please post only comments about the article WL18xx Bluetopia PM Bluetooth RF Testing here.</p>	<p>OMAP35x=For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article WL18xx Bluetopia PM Bluetooth RF Testing here.</p>	<p>OMAPL1=For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article WL18xx Bluetopia PM Bluetooth RF Testing here.</p>	<p>MAVRK=For technical support on MAVRK please post your questions on The MAVRK Toolbox Forum. Please post only comments about the article WL18xx Bluetopia PM Bluetooth RF Testing here.</p>
---	---	--	--	---	--	---

Links



[Amplifiers & Linear Audio](#)
[Broadband RF/IF & Digital Radio](#)
[Clocks & Timers](#)
[Data Converters](#)

[DLP & MEMS High-Reliability Interface](#)
[Logic](#)
[Power Management](#)

[Processors](#)

- ARM Processors
- Digital Signal Processors (DSP)
- Microcontrollers (MCU)
- OMAP Applications Processors

[Switches & Multiplexers](#)
[Temperature Sensors & Control ICs](#)
[Wireless Connectivity](#)

Retrieved from "https://processors.wiki.ti.com/index.php?title=WL18xx_Bluetopia_PM_Bluetooth_RF_Testing&oldid=235357"

This page was last edited on 20 August 2018, at 19:42.

Content is available under [Creative Commons Attribution-ShareAlike](#) unless otherwise noted.