RS9113 WiseConnect BT/BT- LE PER Usage Guide

Version 1.7.9
May 2020

1 Bluetooth Performance Test AT Command Usage

1.1 Opermode:

To validate BT/BT-LE PER mode use below operating modes.

- For BT - 327680,0,1,0
- For BLE – 851968,0,1,0

Note:
Please use the same BT folder structure format for BLE because PER is common for both BT and BLE, but needs to take care the Opermode in configuration file.

NOTE: In context of binary mode we don't have any separate BT/BLE PER commands to perform.

1.2 BT Transmit Tests

This test allows the configuration of the following parameters and starts the transmission of packets.

- PER enable/disable bit
- Device Address
- Packet Type
- Packet Length
- Link Type
- BR/EDR Mode
- Receive Channel Index
- Transmit Channel Index
- Scrambler Seed
- Number of Packets
- Payload Type
- Classic/LE Mode
- LE Channel Type
- Transmit Power
- Transmit Mode
- Hopping Type
- Antenna Select

Command Usage

The command usage is explained below.

at+rsibt_pertransmit=<per_enable/disablebit>,<dev_addr>,<pkt_type>,<pkt_length>,<link_type>,<br_edr_mode>,<rx_channel_index>,<tx_channel_index>,<scrambler_seed>,<no_of_packets>,<payload_type>,<classic_le_mode>,<le_channel_type>,<tx_power>,<tx_mode>,<hopping_type>,<ant_sel>

<per_enable/disablebit>:used to set per mode

1-enable per mode.

2-disable per mode.

<dev_addr>: Device address. It is a 48-bit address in hexadecimal format, e.g., 00-23-A7-01-02-03.

<pkt_type>: Type of the packet to be transmitted, as per the Bluetooth standard.

<pkt_length>: Length of the packet, in bytes, to be transmitted.

<link_type>: Link Type – ACL, SCO, eSCO. Valid only in the Classic mode and invalid in LE mode.
'0' – SCO
'1' – ACL
'2' – eSCO

<br_edr_mode>: Decides whether the transmission has to happen in Basic Rate or Enhanced Data Rate in Classic mode. It is invalid in LE mode.
'1' – Basic data Rate (1Mbps)
'2' or '3' – Enhanced Data Rate(2 Mbps or 3 Mbps)

<rx_channel_index>: Receive channel index, as per the Bluetooth standard.

<tx_channel_index>: Transmit channel index, as per the Bluetooth standard.

<scrambler_seed>: Initial seed to be used for whitening. It should be set to '0' to disable whitening.

<no_of_packets>: Number of packets to be transmitted. It is valid only when the <tx_mode> is set to Burst mode (0).

<payload_type>: Type of payload to be transmitted.
'0' – Payload consists of all zeros.
'1' – Payload consists of all 0xFF's.
'2' – Payload consists of all 0x55's
'3' – Payload consists of all 0xF0's.
'4' – Payload consists of PN9 sequence.

<classic_le_mode>: Choose between Bluetooth Classic and LE modes for the packet transmission.
'1' – Classic mode
'2' – LE Mode

<le_channel_type>: Channel type in LE mode. It is invalid in Classic mode
'0' – Advertising channel
'1' – Data channel

<tx_power>: Transmit power (in dBm) to be used by the module. The value should be between 0 and 18.

<tx_mode>: Choose between Burst and Continuous modes of transmission.
'0' – Burst mode
'1' – Continuous mode

<hopping_type>: Choose the hopping pattern.
'0' – No hopping
'1' – Fixed hopping
'2' – Random hopping

<ant_sel>: Select one of the two RF ports. For the modules without integrated antenna, it is used to select between pins RF_OUT_1 and RF_OUT_2. For the modules with integrated antenna and U.FL connector, it is used to select between the two.
'2' – RF_OUT_2/Antenna
'3' – RF_OUT_1/U.FL

Note:
1. After the transmission starts, the following command can be given to stop the transmission.
at+rsibt_pertransmit=0
2. In binary mode issue the per_transmit command with 'enable' variable as '0'

Example:
at+rsibt_pertransmit=1,00-23-a7-01-02-03,0,20,0,1,37,37,0,0,1,2,0,10,0,0,3

The above command starts transmitting ACL packets in burst mode with no hopping at 1 Mbps with the following configuration in LE mode.
Per enable -1
Device address – 00-23-A7-01-02-03
Packet type – 0
Packet length – 20 bytes
Link type – 0
BR/EDR mode – 1 (Basic data Rate with 1Mbps)
Rx channel index – 37(Advertising Channel)
Tx channel index – 37(Advertising Channel)
Scrambler seed – 0 (Disable whitening)
No of packets – 0(Since tx_mode is burst)
Payload type – 1(Payload consists of all 0xFF's)
Classic/LE mode – 2(LE mode)
LE channel type – 0(Advertising Channel)
Tx power – 10(10dBm)
Tx mode – 0(Burst mode)
Hopping type – 0(no hopping)
Antenna select – 2(RF_OUT_2/Antenna)

Refer to the table below for more details.

Standard Packet	pkt_ type	br_edr_mode	classic/le Mode	Packet Length	Link type
DM1	3	1	1	0-17	1
DH1	4	1	1	0-27	1

DH3	11	1	1	0-183	1
DM3	10	1	1	0-121	1
DH5	15	1	1	0-339	1
DM5	14	1	1	0-224	1
2-DH1	4	2	1	0-54	1
2-DH3	10	2	1	0-367	1
2-DH5	14	2	1	0-679	1
3-DH1	8	3	1	0-83	1
3-DH3	11	3	1	0-552	1
3-DH5	15	3	1	0-1021	1
Any Value	Any Value	1	2	0-37	Any Value

Table 1: BT Packet lengths

Binary Command Index: 0x0098

Binary Structure format:

```
typedef union {
  struct {
    UINT08 type;
    UINT08 enable;
    UINT08 bt_addr[6];
    UINT08 pkt_type;
    UINT08 pkt_length[2];
    UINT08 link_type;
    UINT08 edr_ind;
    UINT08 rx_channel;
    UINT08 tx_channel;
    UINT08 scrambled_seed ;
    UINT32 num_pkts;
    UINT08 payload_type;
    UINT08 protocol_mode;
    UINT08 le_channel;
```

```
  UINT08 tx_power_index ;
  UINT08 tx_mode;
  UINT08 frequency_hop;
  UINT08 ant_sel;
 }PerTransmitFrameSnd;
 UINT08 uPerTransmitBuf[27];
} RSI_BT_CMD_PER_TRANSMIT;
```

Note:
In binary structure, 'type' variable should be always value '4'. This is not configurable.

1.3 BT Receive Tests

This test, allows the configuration of the parameters below.

- Per mode enable /disable bit
- Device Address
- Link Type
- Packet Type
- Packet Length
- Scrambler Seed
- BR/EDR Mode
- Receive Channel Index
- Transmit Channel Index
- Classic/LE Mode
- LE Channel Type
- Hopping Type
- Antenna Select

Command Usage

The "bt_receive" command usage is explained below.

```
at+rsibt_perreceive=<perenable/disable>,<dev_addr>,<link_type>,<
pkt_type>,<pkt_length>,<scrambler_seed>,<br_edr_mode>,<rx_channe
l_index>,<tx_channel_index>,<classic_le_mode>,<le_channel_type>,
<hopping_type>,<ant_sel>
```

Parameters for the "bt_receive" command have the same definition as the ones for the "bt_transmit" command.

Note:

1. After the reception starts using bt_receive, the following command can be given to stop the reception.

2. In binary mode issue the per_receive command with 'enable' variable as '0'

```
at+rsibt_perreceive=0
```

Example:

```
at+rsibt_perreceive=1,00-23-a7-01-02-03,0,0,20,0,1,37,37,2,0,0,2
```

The above command starts receiving ACL packets with no hopping at 1 Mbps with the following configuration in BT-LE mode.
Per enable-1
Device address – 00-23-A7-01-02-03
Link type – 0
Packet type – 0
Packet length – 20 bytes
Scrambler seed – 0(Disable whitening)
BR/EDR mode – 1 (Basic data Rate with 1 Mbps)
Rx channel index – 37(Advertising Channel)
Tx channel index – 37(Advertising Channel)
Classic/LE mode – 2(LE mode)
LE channel type –0 (Advertising Channel)
Hopping type – 0(no hopping)
Antenna select – 2(RF_OUT_2/Antenna)

Binary Command Index: 0x0099
Binary Command Structure:

```
typedef union {
  struct {
    UINT08 type;
    UINT08 enable;
    UINT08 bt_addr[6];
    UINT08 link_type;
    UINT08 pkt_type;
    UINT16 pkt_length;
    UINT08 scrambled_seed ;
    UINT08 edr_ind;
    UINT08 rx_channel;
    UINT08 tx_channel;
    UINT08 protocol_mode;
    UINT08 le_channel;
    UINT08 frequency_hop;
    UINT08 ant_sel;
   }PerReceiveFrameSnd;
   UINT08 uPerReceiveBuf[20];
} RSI_BT_CMD_PER_RECEIVE;
```

Note:
In binary structure, 'type' variable should be always value '5'. This is not configurable.

1.4 BT STATS TEST

The stats command usage is explained below.

```
at+rsibt_perstats
```

This command gives the following stats after running the receive test.
crc_pass: The number of CRC passed packets received
crc_fail: The number of CRC failed packets received
rssi: The RSSI value of the last received packet

Binary Command Index: 0x009A
Binary Structure Format:

```
typedef union {
   struct {
   UINT08 type;
   UINT08 enable;
  }PerStatsFrameSnd;
  UINT08 uPerStatsBuf[2];
} RSI_BT_CMD_PER_STATS;
```

Binary Response structure Format:

```
typedef struct {
   UINT16 crc_fail;
   UINT16 crc_pass;
   UINT8 RSSI;
} RSI_BT_RESP_PER_STATS;
```

Note:
1. For continuously getting the stats repeatedly give the stats command.
2. In binary structure, 'type' variable should be always value '8'. This is not configurable.

1.4.1.1 Continuous Wave Transmit Mode

This command is used to configure the device to transmit a continuous wave. The following parameters can be configured.

1. Channel Index
2. Start/Stop
3. Antenna Select

Command Usage

The command usage is explained below.

```
at+rsibt_percwmode=<channel_index>,<start/stop>,<ant_sel>
```

<channel_index>: Channel index, as per the Bluetooth standard.

<start/stop>: Start or Stop the Continuous Wave mode transmission.

- '0' – start the cw mode transmission
- '2' – stop the cw mode transmission

<ant_sel>: Select one of the two RF ports. For the modules without integrated antenna, it is used to select between pins RF_OUT_1 and RF_OUT_2. For the modules with integrated antenna and U.FL connector, it is used to select between the two.

- '2' – RF_OUT_1/U.FL
- '3' – RF_OUT_2/Antenna

Example

```
at+rsibt_percwmode=10,0,3
```

The above command starts continuous wave transmission with the following configuration

Channel index – 10

Start - 0 (starts the transmission)

Antennal Select – 3(RF_OUT_2/Antenna)

Binary Command Index: 0x009B

Binary Structure Format:

```
typedef union {
 struct {
  UINT08 type;
  UINT08 channel;
  UINT08 cw_mode;
  UINT08 cw_type;
  UINT08 ant_sel;
 }PerCwFrameSnd;
 UINT08 uPerCwBuf[5];
} RSI_BT_CMD_PER_CW_MODE;
```

1.5 AFH_MAP

Description:
This command mode is used to configure the following parameters..
Command Usage

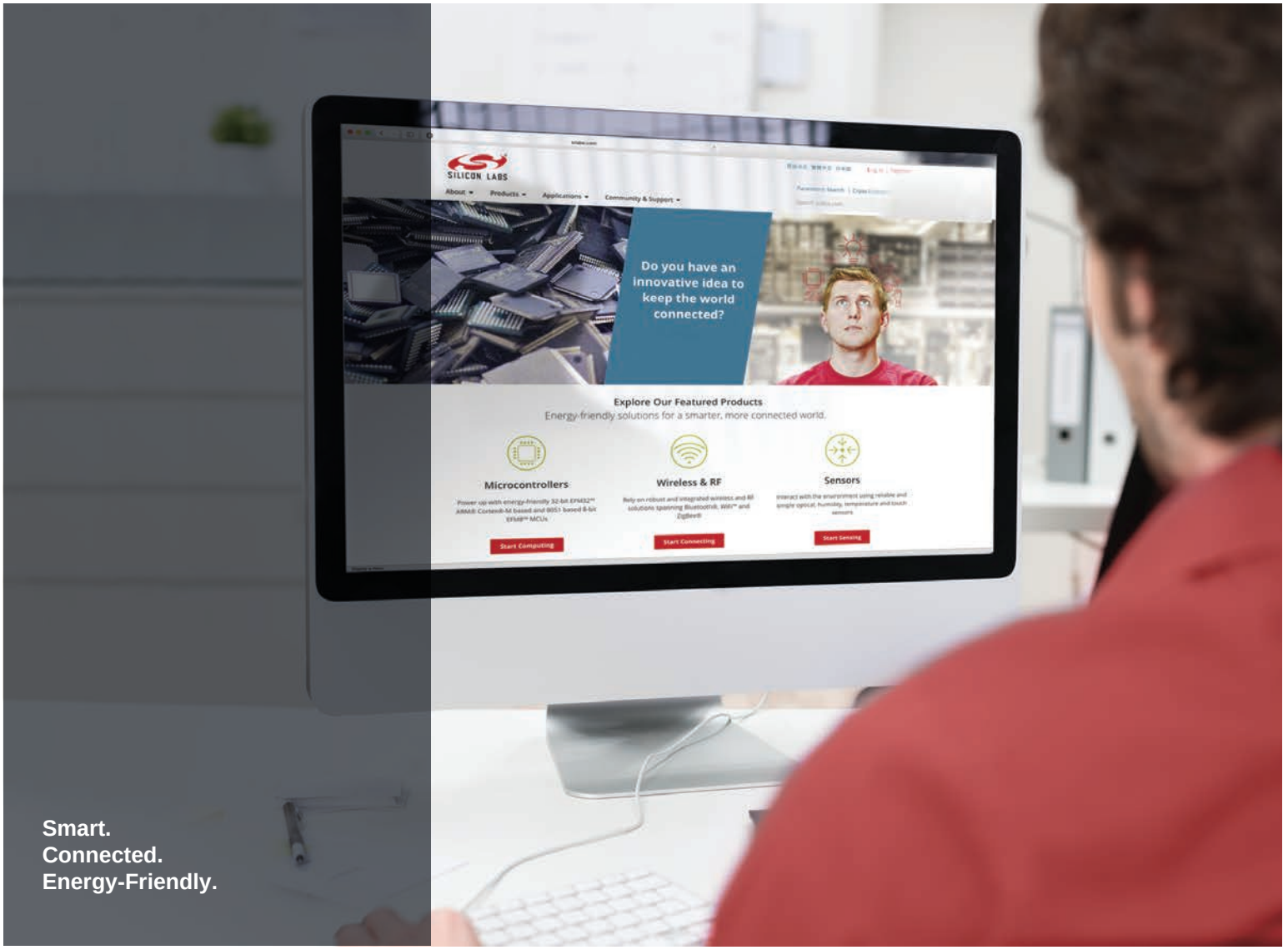
```
at+rsibt_afhmap=<startingchannel>,<endingchannel>
```

Parameters
<startchannel>: This parameter indicates the starting channel index, as per the Bluetooth standard. The range is from 0 to 78.
<endchannel>: This parameter indicates the ending channel index, as per the Bluetooth standard. The range is from 0 to 78.

Example

```
at+rsibt_afhmap=12,30
```

> **Note:**
> In binary mode, this support is not present.



Products
www.silabs.com/products

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

SILICON LABS

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

http://www.silabs.com