Steps to Execute Demo_67

Silicon Labs Product Documentation

Exported on 07/14/2023

Table of Contents

Overview

This example demonstrates throughput measurements of WLAN/BLE/BT in alone or with combination of other protocol activities.

Using this application unidirectional/bi-directional throughputs of TCP/UDP/SSL/HTTP/HTTPs can be measured directly from application in the presence of BT/BLE activities.

Provides a configuration parameters for measuring BT/BLE throughput using sniffer.

This application has provision to control the other protocol activities while measuring WLAN/BLE throughputs.

Sequence of Events:
WLAN Task:

This application explains user how to,

1. Configure silabs module in station mode
2. Scan for access point and connect to it.
3. Configure the IP address
4. Configure the module in SSL/TCP/UDP/HTTP/HTTPS client or server
5. Open multiple sockets for bidirectional data transfer
6. Transmit/ Receive the data packets to/from remote server
7. Measure the unidirectional / bidirectional throughputs individually or combinedly with other protocol activities

BLE task:

This application explains user how to,

1. Configure module in advertising mode.
2. Manage the connection between remote device and module.
3. Transmit/receive the data packets to/from remote device
4. Measure the BLE throughput in sniffer

BT task:

This application explains user how to,

1. Configure module in Initiator or acceptor role.
2. Manage the AVRCP connections
3. Stream the A2DP packets
4. Measure the BT throughput in sniffer

Application Setup

The WiSeConnect parts require that the host processor is connected to the WiSeConnect using either SPI, UART, or USB host interface. The host processor firmware needs to properly initialize the selected host interface. The Silicon Labs Wireless SAPI framework provides necessary HAL APIs to enable a variety of host processors.

WiSeConnect based Setup Requirements

- Windows / Linux PC with Host interface(UART/ USB-CDC/ SPI/ USB) in case of WiSeConnect
- Silicon Labs module
- Bluetooth headset
- Smart phone/tablet with BLE Application (Ex: Light Blue APP / n-RF connect)
- WLAN Access Point and a Windows PC with iperf application

Configuration and Steps for Execution

Configuration of Application:

WLAN throughputs:

TCP/UDP/SSL throughputs :

This section explains user how to configure the application for measuring TCP/UDP/SSL unidirectional/ Bidirectional throughputs in alone or with combination of BT/BLE activities.

1. Open '**rsi_common_config.h**' file provided in the release package at '**.. \RSI_SDK_vx\apps\demo\COEX_THROUGHPUT_DEMO_67**' and configure below macros.
   set below macros to 1 to measure **WLAN** alone throughput

```
#define RSI_ENABLE_WLAN_TEST    1 //Set this to 0 to disable WLAN
#define WLAN THROUGHPUT TEST    1 //Set this to 0 for HTTP download
```

   set below macros to 1 to measure **WLAN** throughput along with specific **BLE** activity.

```
#define RSI_ENABLE_WLAN_TEST    1 //Set this to 0 to disable WLAN
#define WLAN_THROUGHPUT_TEST    1 //Set this to 0 for HTTP download
#define RSI_ENABLE_BLE_TEST     1 //Set this to 0 to disable BLE
```

   Note: While measuring WLAN throughput along with BLE activities, ensure 'BLE_THROUGHPUT_TEST' is set to '0'

   set below macros to 1 to measure **WLAN** throughput along with specific **BT** activity.

```
#define RSI_ENABLE_WLAN_TEST    1 //Set this to 0 to disable WLAN
#define WLAN_THROUGHPUT_TEST    1 //Set this to 0 for HTTP download
#define RSI_ENABLE_BT_TEST      1 //Set this to 0 to enable BT
```

   set below macros to 1 to measure **WLAN** throughput along with **BT and BLE** activities.

```
#define RSI_ENABLE_WLAN_TEST    1 //Set this to 0 to disable WLAN
#define WLAN_THROUGHPUT_TEST    1//Set this to 0 for HTTP download
#define RSI_ENABLE_BLE_TEST     1 //Set this to 0 to enable BLE
#define RSI ENABLE BT TEST      1 //Set this to 0 to enable BT
```

   Note: While measuring WLAN throughput with BT and BLE activities ensure 'BLE_THROUGHPUT_TEST' is set to '0'

   If BT/BLE is enabled, configure specific activity of BT/BLE using below macros.
   BLE operations: set any one of below macros to choose desired BLE activity

```
#define BLE_INIT_DONE                       0           //! make it 1 for BLE
init only
#define BLE_CONNECTED                       0           //! make it 1 for BLE
connection only
#define BLE_DATA_TRANSFER_START             1           //! make it 1 for BLE
data transfer
```

BT A2DP operations: set any of one of below 3 macros to choose desired BT A2DP activity

```
#define BT_A2DP_INIT_DONE                   0           //! make it 1 for BT
init only
#define BT_A2DP_CONNECTED                   0           //! make it 1 for BT
A2DP connection only
#define BT_A2DP_START                       1           //! make it 1 for BT
A2DP streaming
```

By default BT_A2DP_START and BLE_DATA_TRANSFER_START are set to '1', that means WLAN measurement will start after BT A2DP and BLE Gatt data transfers

choose the required **operational mode** of RS9116

```
#define RSI_COEX_MODE           9
```

valid configurations:
0 - WLAN alone mode
5 - BT alone mode
9 - WLAN + BT + BLE mode
13 - BLE alone mode

Note: By default opermode set to WLAN+BT+BLE

2. Select WLAN configurations in '**rsi_wlan_config.h**' file provided in the release package at '**..\RSI_SDK_vx\apps\demo\COEX_THROUGHPUT_DEMO_67**'
Enter the AP Connectivity essentials configs as the value to SSID, SECURITY_TYPE and PSK

```
#define     SSID                        "Hotspot"
#define     SECURITY_TYPE               RSI_WPA2
#define     PSK                         "12345678"
```

Channel no in which device should scan choose

```
#define     CHANNEL_NO                  0       //! 0 - scan all channels
```

Port number of remote server

```
#define     SERVER_PORT             5001
```

while verifying SSL RX throughput , port number of remote server should be configured below

```
#define     SSL_RX_SERVER_PORT      5002
```

IP address of remote server

```
#define     SERVER_IP_ADDRESS       "192.168.0.102"
```

Port number of module

```
#define     DEVICE_PORT             5001
```

To select the ip getting configure below macros

```
#define     DHCP_MODE               1 //0 enable or disable
#if !DHCP_MODE // Need to configure manually if dhcp disabled
#define   DEVICE_IP          0x6500A8C0  //192.168.0.101
#define   GATEWAY            0x0100A8C0  //192.168.0.1
#define   NETMASK            0x00FFFFFF  //255.255.255.0
#endif
```

choose the throughput type by configuring below macro

```
#define     THROUGHPUT_TYPE           TCP_TX
```

valid configurations of THROUGHPUT_TYPE:
UDP_TX → UDP transmit
UDP_RX → UDP receive
TCP_TX → TCP transmit
TCP_RX → TCP receive
SSL_TX → SSL transmit
SSL_RX → SSL receive
UDP_BIDIRECTIONAL → UDP transmit and receive
TCP_BIDIRECTIONAL → TCP transmit and receive
SSL_BIDIRECTIONAL → SSL transmit and receive

 BY default 'THROUGHPUT_TYPE' is set to TCP_TX

Average time required to measure UDP_TX/TCP_TX throughputs

```
#define     THROUGHPUT_AVG_TIME     60000    //60sec of throughput numbers
average
```

Maximum no. of packets required to measure UDP_RX/UDP_BIDIRECTIONAL

```
#define     MAX_TX_PKTS             10000
```

3. Open '**rsi_ble_config.h**' file provided in the release package at '**.. \RSI_SDK_vx\apps\demo\COEX_THROUGHPUT_DEMO_67**' and choose **BLE** application configurations BLE Advertise name

```
#define RSI_BLE_APP_GATT_TEST    (void *)"SI_COEX_MAX_DEMO"
```

Configure BLE advertising interval

```
#define RSI_BLE_ADV_INT_MIN         0x06a8 //! 1065ms
#define RSI_BLE_ADV_INT_MAX         0x06a8 //! 1065ms
```

Configure below macros to set connection interval, connection latency and connection supervision timeout

```
#define CONN_INTERVAL_M1            1600    // connection interval:2s
#define CONN LATENCY M1             0       // latency : 0
#define CONN SUPERVISION TIMEOUT M1 1600
```

4. Select BT configurations in '**rsi_bt_config.h**' file provided in the release package at '**.. \RSI_SDK_vx\apps\demo\COEX_THROUGHPUT_DEMO_67**'
RSI_BT_LOCAL_NAME refers name of the Silicon module to appear during scanning by remote devices.

```
#define RSI_BT_LOCAL_NAME                               "A2DP_AVRCP_SOURCE_G"
```

RSI_BT_REMOTE_BD_ADDR refers BD address of the Remote device to which Silicon device has to connect.

```
#define RSI_BT_REMOTE_BD_ADDR                           "00:1E:7C:25:E9:6D"
```

RSI_APP_AVDTP_ROLE refers role of Silicon Device, valid configurations

```
#define     RSI APP AVDTP ROLE                          INITIATOR ROLE
```

Valid configurations:

```
INITIATOR_ROLE -> Module acts as source

ACCEPTOR_ROLE ->  Module acts as sink

Configure below macro for scan and connection
```

```
#define     INQUIRY_ENABLE      1       // 0- connect without scanning, 1-
scan and connect
```

- By default INQUIRY_ENABLE is set to '1'

Configure below macro to call remote name request explicitly followed by connection

```
#define     INQ_REMOTE_NAME_REQUEST     0 // 0- no specific call for remote
name request after connection, 1- inquiry for remote name request after
connection
```

- By default INQ_REMOTE_NAME_REQUEST is set to '1'

Configure below macro, to check inquiry+connection

```
#define     INQUIRY_CONNECTION_SIMULTANEOUS     0 //0 - initiate connection
after scanned device found, 1 - initiate connection independent of scanning
```

- By default INQUIRY_CONNECTION_SIMULTANEOUS is set to '1'

HTTP/HTTPs download throughputs:

This section explains user how to configure the application for measuring HTTP/HTTPs unidirectional throughputs in alone or with combination of BT/BLE.

1. Open '**rsi_common_config.h**' file provided in the release package at '**.. \RSI_SDK_vx\apps\demo\COEX_THROUGHPUT_DEMO_67**' and configure below macros.
   set below macros to 1 to measure **WLAN** alone throughput

```
#define RSI_ENABLE_WLAN_TEST    1 //Set this to 0 to disable WLAN
#define WLAN_THROUGHPUT_TEST    0 //Set this to 0 for HTTP download
```

set below macros to 1 to measure **WLAN** throughput along with **BLE** connection/data transfer.

```
#define RSI_ENABLE_WLAN_TEST    1 //Set this to 0 to disable WLAN
#define WLAN_THROUGHPUT_TEST    0 //Set this to 0 for HTTP download
#define RSI_ENABLE_BLE_TEST     1 //Set this to 0 to disable BLE
```

- **Note:** While measuring WLAN throughput along with BLE activities, ensure 'BLE_THROUGHPUT_TEST' is set to '0'

set below macros to 1 to measure **WLAN** throughput along with **BT** connection/data transfer.

```
#define RSI_ENABLE_WLAN_TEST    1 //Set this to 0 to disable WLAN
#define WLAN_THROUGHPUT_TEST    0 //Set this to 0 for HTTP download
#define RSI_ENABLE_BT_TEST      1 //Set this to 0 to enable BT
```

set below macros to 1 to measure **WLAN** throughput along with **BT and BLE** activities.

```
#define RSI_ENABLE_WLAN_TEST     1 //Set this to 0 to disable WLAN
#define WLAN_THROUGHPUT_TEST     0//Set this to 0 for HTTP download
#define RSI_ENABLE_BLE_TEST      1 //Set this to 0 to enable BLE
#define RSI_ENABLE_BT_TEST       1 //Set this to 0 to enable BT
```

Note: While measuring WLAN throughput with BT and BLE activities ensure 'BLE_THROUGHPUT_TEST' is set to '0'

choose the required **operational mode** of RS9116

```
#define RSI_COEX_MODE           9
```

valid configurations:
0 - WLAN alone mode
5 - BT alone mode
9 - WLAN + BT + BLE mode
13 - BLE alone mode

Note: By default opermode set to WLAN+BT+BLE

2. Select WLAN configurations in '**rsi_wlan_config.h**' file provided in the release package at '**..\RSI_SDK_vx\apps\demo\COEX_THROUGHPUT_DEMO_67**' Parameter for choosing HTTP download throughput test or normal HTTP download

```
#define     HTTP_DOWNLOAD_THROUGPUT_TEST    1       //! 1- measure HTTP/HTTPS
download throughputs
                                                    //! 0 - default HTTP/HTTPS
download
```

Enter the AP Connectivity essentials configs as the value to SSID, SECURITY_TYPE and PSK

```
#define     SSID                        "Hotspot"
#define     SECURITY_TYPE               RSI_WPA2
#define     PSK                         "12345678"
```

Channel no in which device should scan choose

```
#define     CHANNEL_NO                  0       //! 0 - scan all channels
```

Configure below macros to make Use of Local HTTP server to download the files.

```
#define     RSI_DNS_CLIENT    0                 // set to '1' only if using
server name instead of server ip address, by default it is set to '0'
```

```
#define     RX_DATA           1                 // set to '1' to RX data from
remote server
#define     HTTPS_DOWNLOAD    0                 // set to '0' to choose HTTP
download
#define     SERVER_PORT       80                // by default http runs on port
80
#define     SERVER_IP_ADDRESS "192.168.0.101"   //Local server ip address
#define     DOWNLOAD_FILENAME   "dltestdata32.txt"  // File to download, by
default this file is provided in the demo
#define     BYTES_TO_RECEIVE    1048576             // size of file configured
under 'DOWNLOAD_FILENAME'
#define     CONTINUOUS_HTTP_DOWNLOAD    1           // set to '1' to download
continuously, if reset download happens only once.
```

Configure below macros to make Use of Local HTTPS server to download the files.

```
#define     RSI_DNS_CLIENT    0                 // set to '1' only if using
server name instead of server ip address, by default it is set to '0'
#define     RX_DATA           1                 // set to '1' to RX data from
remote server
#define     HTTPS_DOWNLOAD    1                 // set to '1' to choose HTTPs
download
#define     SERVER_PORT       443               // by default https runs on
port 443
#define     SERVER_IP_ADDRESS "192.168.0.101"   //Local server ip address
#define     DOWNLOAD_FILENAME   "dltest.txt"    // File to download, by default
this file is provided in the demo
#define     BYTES_TO_RECEIVE    6144                // size of file configured
under 'DOWNLOAD_FILENAME'
#define     CONTINUOUS_HTTP_DOWNLOAD    1           // set to '1' to download
continuously, if reset download happens only once.
```

Follow below steps to configure local https server

1. Download and install SSL server from https://slproweb.com/products/Win32OpenSSL.html
2. Add the installed location (ex: "C:\Program Files\OpenSSL-Win64\bin") in environment variable 'PATH' and restart the pc to reflect the changes.

3. Open '**rsi_ble_config.h**' file provided in the release package at '**..\RSI_SDK_vx\apps\demo\COEX_THROUGHPUT_DEMO_67**' and choose **BLE** application configurations BLE Advertise name

```
#define RSI_BLE_APP_GATT_TEST    (void *)"SI COEX MAX DEMO"
```

Configure BLE advertising interval

```
#define RSI_BLE_ADV_INT_MIN         0x06a8 //! 1065ms
#define RSI_BLE_ADV_INT_MAX         0x06a8 //! 1065ms
```

Configure below macros to set connection interval, connection latency and connection supervision timeout

```
#define CONN_INTERVAL_M1            1600    // connection interval:2s
#define CONN_LATENCY_M1             0       // latency : 0
#define CONN_SUPERVISION_TIMEOUT_M1 1600
```

4. Select BT configurations in '**rsi_bt_config.h**' file provided in the release package at '**.. \RSI_SDK_vx\apps\demo\COEX_THROUGHPUT_DEMO_67**'
RSI_BT_LOCAL_NAME refers name of the Silicon module to appear during scanning by remote devices.

```
#define RSI_BT_LOCAL_NAME                                "A2DP_AVRCP_SOURCE_G"
```

RSI_BT_REMOTE_BD_ADDR refers BD address of the Remote device to which Silicon device has to connect.

```
#define RSI_BT_REMOTE_BD_ADDR                           "00:1E:7C:25:E9:6D"
```

RSI_APP_AVDTP_ROLE refers role of Silicon Device, valid configurations

```
#define     RSI APP AVDTP ROLE                          INITIATOR ROLE
```

Valid configurations:

```
INITIATOR_ROLE -> Module acts as source

ACCEPTOR_ROLE ->  Module acts as sink

Configure below macro for scan and connection
```

```
#define     INQUIRY_ENABLE      1        // 0- connect without scanning, 1-
scan and connect
```

By default INQUIRY_ENABLE is set to '1'

Configure below macro to call remote name request explicitly followed by connection

```
#define     INQ REMOTE NAME REQUEST     0 // 0- no specific call for remote
name request after connection, 1- inquiry for remote name request after
connection
```

By default INQ_REMOTE_NAME_REQUEST is set to '1'

Configure below macro, to check inquiry+connection

```
#define     INQUIRY_CONNECTION_SIMULTANEOUS     0 //0 - initiate connection
after scanned device found, 1 - initiate connection independent of scanning
```

> By default INQUIRY_CONNECTION_SIMULTANEOUS is set to '1'

BLE throughputs:

This section explains user how to configure the application for measuring BLE Tx/Rx throughputs using sniffer in alone or with combination of WLAN/BT.

1. open '**rsi_common_config.h**' file provided in the release package at '**.. \RSI_SDK_vx\apps\demo\COEX_THROUGHPUT_DEMO_67**' and configure below macros.
   set below macro to 1 to measure **BLE** alone throughput

```
#define RSI_ENABLE_BLE_TEST    1 //Set this to 0 to disable BLE
#define BLE_THROUGHPUT_TEST    1 //Set this to 0 for normal BLE operation
```

   set below macros to 1 to measure **BLE** throughput along with specific **WLAN** activity.

```
#define RSI_ENABLE_BLE_TEST     1 //Set this to 0 to disable BLE
#define BLE_THROUGHPUT_TEST     1 //Set this to 0 for normal BLE operation
#define RSI_ENABLE_WLAN_TEST    1 //Set this to 0 to disable WLAN
```

   > **Note:** While measuring BLE throughput with WLAN activities, ensure 'WLAN_THROUGHPUT_TEST' is set to '0'

   set below macros to 1 to measure **BLE** throughput along with specific **BT** activity.

```
#define RSI_ENABLE_BLE_TEST     1 //Set this to 0 to disable BLE
#define BLE_THROUGHPUT_TEST     1 //Set this to 0 for normal BLE operation
#define RSI_ENABLE_BT_TEST      1 //Set this to 0 to enable BT
```

   set below macros to 1 to measure **BLE** throughput along with **BT and WLAN** activities.

```
#define RSI_ENABLE_BLE_TEST     1 //Set this to 0 to disable BLE
#define BLE_THROUGHPUT_TEST     1 //Set this to 0 for normal BLE operation
#define RSI_ENABLE_WLAN_TEST    1 //Set this to 0 to disable WLAN
#define RSI_ENABLE_BT_TEST      1 //Set this to 0 to enable BT
```

   > **Note:** While measuring BLE throughput with WLAN and BT activities ensure 'WLAN_THROUGHPUT_TEST' is set to '0'

   If WLAN/BT is enabled, configure specific activity of WLAN/BT using below macros.
   WLAN operations: set any one of below macros to choose desired WLAN activity

```
#define WLAN_SCAN_ONLY                    0          //! make it 1 for WLAN
scan only
```

```
#define WLAN_CONNECT_ONLY                    0          //! make it 1 for WLAN
connection only
#define WLAN_DATATRANSFER                    1          //! make it 1 for WLAN
http download
```

BT A2DP operations: set any of one of below 3 macros to choose desired BT A2DP activity

```
#define BT_A2DP_INIT_DONE                    0          //! make it 1 for BT
init only
#define BT_A2DP_CONNECTED                    0          //! make it 1 for BT
A2DP connection only
#define BT_A2DP_START                        1          //! make it 1 for BT
A2DP data transfer
```

- By default BT_A2DP_START and WLAN_DATATRANSFER are set to '1'

choose the required **operational mode** of RS9116

```
#define RSI_COEX_MODE          9
```

valid configurations:
0 - WLAN alone mode
5 - BT alone mode
9 - WLAN + BT + BLE mode
13 - BLE alone mode

- By default opermode set to WLAN+BT+BLE

2. Open '**rsi_ble_config.h**' file provided in the release package at '**.. \RSI_SDK_vx\apps\demo\COEX_THROUGHPUT_DEMO_67**' and choose **BLE** application configurations.
BLE Advertise name

```
#define RSI_BLE_APP_GATT_TEST    (void *)"SI_COEX_MAX_DEMO"
```

Configure BLE advertising interval

```
#define RSI_BLE_ADV_INT_MIN         0x06a8 //! 1065ms
#define RSI_BLE_ADV_INT_MAX         0x06a8 //! 1065ms
```

Configure below macros to set connection interval, connection latency and connection supervision timeout

```
#define CONN_INTERVAL_M1            9   // connection interval:1.125ms
#define CONN_LATENCY_M1             0   // latency : 0
#define CONN_SUPERVISION_TIMEOUT_M1 1600
```

configure below macros to enable/disable data length extension mode

```
#define DLE_ON_M1                   1   // 1- DLE ON, 0- DLE OFF
```

configure BLE data transfer type

```
#define RX_NOTIFICATIONS_FROM_M1    0  //! set below macro to receive 'gatt
notifications' from remote device
#define RX_INDICATIONS_FROM_M1      0  //! set below macro to receive 'gatt
indications' from remote device
#define TX_NOTIFICATIONS_TO_M1      1  //! set below macro to Transmit 'gatt
notifications' to remote device
#define TX_WRITES_TO_M1             0  //! set below macro to Transmit 'gatt
write with response' to remote device
#define TX_WRITES_NO_RESP_TO_M1     0  //! set below macro to Transmit 'gatt
write without response' to remote device
#define TX_INDICATIONS_TO_M1        0 //! set below macro to Transmit 'gatt
indications' to remote device
```

Note: By default all BLE configurations are choosen to get high BLE throughput

3. Select WLAN configurations in '**rsi_wlan_config.h**' file provided in the release package at '**..\RSI_SDK_vx\apps\demo\COEX_THROUGHPUT_DEMO_67**' Enter the AP Connectivity essentials configs as the value to SSID, SECURITY_TYPE and PSK

```
#define     SSID                        "Hotspot"
#define     SECURITY_TYPE               RSI_WPA2
#define     PSK                         "12345678"
```

Channel no in which device should scan choose

```
#define     CHANNEL_NO                  0       //! 0 - scan all channels
```

Configure below macros to make use of Local HTTP server to download the files.

```
#define     RSI_DNS_CLIENT    0                 // set to '1' only if using
server name instead of server ip address, by default it is set to '0'
#define     RX_DATA           1                 // set to '1' to RX data from
remote server
#define     HTTPS_DOWNLOAD    0                 // set to '0' to choose HTTP
download
#define     SERVER_PORT       80                // by default http runs on port
80
#define     SERVER_IP_ADDRESS "192.168.0.101"   //Local server ip address
#define     DOWNLOAD_FILENAME   "dltestdata32.txt"  // File to download, by
default this file is provided in the demo
#define     BYTES_TO_RECEIVE    1048576             // size of file configured
under 'DOWNLOAD_FILENAME'
```

```
#define     CONTINUOUS_HTTP_DOWNLOAD    1           // set to '1' to download
continuously, if reset download happens only once.
```

Configure below macros to make use of Local HTTPS server to download the files.

```
#define     RSI_DNS_CLIENT    0                 // set to '1' only if using
server name instead of server ip address, by default it is set to '0'
#define     RX_DATA           1                 // set to '1' to RX data from
remote server
#define     HTTPS_DOWNLOAD    1                 // set to '1' to choose HTTPs
download
#define     SERVER_PORT       443               // by default https runs on
port 443
#define     SERVER_IP_ADDRESS "192.168.0.101"   //Local server ip address
#define     DOWNLOAD_FILENAME   "dltest.txt"    // File to download, by default
this file is provided in the demo
#define     BYTES_TO_RECEIVE    6144                // size of file configured
under 'DOWNLOAD_FILENAME'
#define     CONTINUOUS_HTTP_DOWNLOAD    1           // set to '1' to download
continuously, if reset download happens only once.
```

Follow below steps to configure local https server

1. Download and install SSL server from https://slproweb.com/products/Win32OpenSSL.html
2. Add the installed location (ex: "C:\Program Files\OpenSSL-Win64\bin") in environment variable 'PATH' and restart the pc to reflect the changes.

4. Select BT configurations in '**rsi_bt_config.h**' file provided in the release package at '**.. \RSI_SDK_vx\apps\demo\COEX_THROUGHPUT_DEMO_67**'
RSI_BT_LOCAL_NAME refers name of the Silicon module to appear during scanning by remote devices.

```
#define RSI_BT_LOCAL_NAME                                "A2DP_AVRCP_SOURCE_G"
```

RSI_BT_REMOTE_BD_ADDR refers BD address of the Remote device to which Silicon device has to connect.

```
#define RSI_BT_REMOTE_BD_ADDR                           "00:1E:7C:25:E9:6D"
```

RSI_APP_AVDTP_ROLE refers role of Silicon Device, valid configurations

```
#define     RSI_APP_AVDTP_ROLE                          INITIATOR_ROLE
```

Valid configurations:

INITIATOR_ROLE -> Module acts as source

ACCEPTOR_ROLE -> Module acts as sink

Configure below macro for scan and connection

```
#define     INQUIRY_ENABLE      1        // 0- connect without scanning, 1-
scan and connect
```

By default INQUIRY_ENABLE is set to '1'

Configure below macro to call remote name request explicitly followed by connection

```
#define     INQ_REMOTE_NAME_REQUEST     0 // 0- no specific call for remote
name request after connection, 1- inquiry for remote name request after
connection
```

By default INQ_REMOTE_NAME_REQUEST is set to '1'

Configure below macro, to check inquiry+connection

```
#define     INQUIRY_CONNECTION_SIMULTANEOUS     0 //0 - initiate connection
after scanned device found, 1 - initiate connection independent of scanning
```

By default INQUIRY_CONNECTION_SIMULTANEOUS is set to '1'

BT throughputs:

This section explains user how to configure the application for measuring BT audio streaming throughputs from sniffer in alone or with combination of WLAN/BLE

1. open '**rsi_common_config.h**' file provided in the release package at '**.. \RSI_SDK_vx\apps\demo\COEX_THROUGHPUT_DEMO_67**' and configure below macros.
   set below macro to 1 to measure **BT** alone throughput

```
#define RSI_ENABLE_BT_TEST    1 //Set this to 0 to disable BT
#define BT_THROUGHPUT_TEST    1 //Set this to 0 for normal BT operation
```

   set below macros to 1 to measure **BT** throughput along with specific **WLAN** activity.

```
#define RSI_ENABLE_BT_TEST      1 //Set this to 0 to disable BT
#define BT_THROUGHPUT_TEST      1 //Set this to 0 for normal BT operation
#define RSI ENABLE WLAN TEST    1 //Set this to 0 to disable WLAN
```

   Note: While measuring BT throughput with WLAN activities, ensure 'WLAN_THROUGHPUT_TEST' is set to '0'

   set below macros to 1 to measure **BT** throughput along with specific **BLE** activity.

```
#define RSI_ENABLE_BT_TEST      1 //Set this to 0 to disable BT
#define BT_THROUGHPUT_TEST      1 //Set this to 0 for normal BT operation
#define RSI_ENABLE_BLE_TEST     1 //Set this to 0 to enable BLE
```

set below macros to 1 to measure **BT** throughput along with **BT and WLAN** activities.

```
#define RSI_ENABLE_BT_TEST      1 //Set this to 0 to disable BT
#define BT_THROUGHPUT_TEST      1 //Set this to 0 for normal BT operation
#define RSI_ENABLE_WLAN_TEST    1 //Set this to 0 to disable WLAN
#define RSI_ENABLE_BLE_TEST     1 //Set this to 0 to enable BLE
```

Note: While measuring BT throughput with WLAN and Ble activities ensure 'WLAN_THROUGHPUT_TEST' and 'BLE_THROUGHPUT_TEST' are set to '0'

If WLAN/BLE is enabled, configure specific activity of WLAN/BLE using below macros.
WLAN operations: set any one of below macros to choose desired WLAN activity

```
#define WLAN_SCAN_ONLY                  0           //! make it 1 for WLAN
scan only
#define WLAN_CONNECT_ONLY               0           //! make it 1 for WLAN
connection only
#define WLAN_DATATRANSFER               1           //! make it 1 for WLAN
http download
```

BLE operations: set any one of below macros to choose desired BLE activity

```
#define BLE_INIT_DONE                   0           //! make it 1 for BLE
init only
#define BLE_CONNECTED                   0           //! make it 1 for BLE
connection only
#define BLE_DATA_TRANSFER_START         1           //! make it 1 for BLE
data transfer
```

By default BLE_DATA_TRANSFER_START and WLAN_DATATRANSFER are set to '1

choose the required **operational mode** of RS9116

```
#define RSI_COEX_MODE           9
```

valid configurations:
0 - WLAN alone mode
5 - BT alone mode
9 - WLAN + BT + BLE mode

13 - BLE alone mode

By default opermode set to WLAN+BT+BLE

2. Select BT configurations in '**rsi_bt_config.h**' file provided in the release package at '**.. \RSI_SDK_vx\apps\demo\COEX_THROUGHPUT_DEMO_67**'
RSI_BT_LOCAL_NAME refers name of the Silicon module to appear during scanning by remote devices.

```
#define RSI BT LOCAL NAME                                "A2DP AVRCP SOURCE G"
```

RSI_BT_REMOTE_BD_ADDR refers BD address of the Remote device to which Silicon device has to connect.

```
#define RSI_BT_REMOTE_BD_ADDR                           "00:1E:7C:25:E9:6D"
```

RSI_APP_AVDTP_ROLE refers role of Silicon Device, valid configurations

```
#define     RSI_APP_AVDTP_ROLE                          INITIATOR_ROLE
```

Valid configurations:

```
INITIATOR_ROLE -> Module acts as source

ACCEPTOR_ROLE ->  Module acts as sink

Configure below macros for controlling BT throughput
```

```
#define     ALLOCMETHOD         SBC_ALLOCATION_LOUDNESS             //! valid
configurations: SBC_ALLOCATION_SNR and SBC_ALLOCATION_LOUDNESS
#define     SUBBANDS            SBC_SUBBANDS_8                      //! valid
configs: SBC_SUBBANDS_4 and SBC_SUBBANDS_8
#define     BLOCKLENGTH         SBC_BLOCK_LENGTH_16                 //! valid
configs: SBC_BLOCK_LENGTH_4, SBC_BLOCK_LENGTH_8, SBC_BLOCK_LENGTH_16 and
SBC BLOCK LENGTH 32
#define     MAXBITPOOL          35                                  //! range :
19 to 53
#define     BT PACKET TYPE SEL  BT EDR 2DH3 PACKETS ONLY            //! valid
configs : BT_EDR_2DH3_PACKETS_ONLY, BT_EDR_2DH5_PACKETS_ONLY,
BT_EDR_3DH1_PACKETS_ONLY,

BT_EDR_3DH3_PACKETS_ONLY and BT_EDR_3DH5_PACKETS_ONLY
```

```
Configure below macro for scan and connection
```

```
#define     INQUIRY_ENABLE      1       // 0- connect without scanning, 1-
scan and connect
```

By default INQUIRY_ENABLE is set to '1'

3. Open '**rsi_ble_config.h**' file provided in the release package at '**..\RSI_SDK_vx\apps\demo\COEX_THROUGHPUT_DEMO_67**' and choose **BLE** application configurations. BLE Advertise name

```
#define RSI_BLE_APP_GATT_TEST    (void *)"SI_COEX_MAX_DEMO"
```

Configure BLE advertising interval

```
#define RSI_BLE_ADV_INT_MIN         0x06a8 //! 1065ms
#define RSI BLE ADV INT MAX         0x06a8 //! 1065ms
```

Configure below macros to set connection interval, connection latency and connection supervision timeout

```
#define CONN_INTERVAL_M1            9   // connection interval:1.125ms
#define CONN_LATENCY_M1             0   // latency : 0
#define CONN_SUPERVISION_TIMEOUT_M1 1600
```

configure below macros to enable/disable data length extension mode

```
#define DLE ON M1                   1   // 1- DLE ON, 0- DLE OFF
```

configure BLE data transfer type

```
#define RX_NOTIFICATIONS_FROM_M1    0  //! set below macro to receive 'gatt
notifications' from remote device
#define RX_INDICATIONS_FROM_M1      0  //! set below macro to receive 'gatt
indications' from remote device
#define TX_NOTIFICATIONS_TO_M1      1  //! set below macro to Transmit 'gatt
notifications' to remote device
#define TX_WRITES_TO_M1             0  //! set below macro to Transmit 'gatt
write with response' to remote device
#define TX_WRITES_NO_RESP_TO_M1     0  //! set below macro to Transmit 'gatt
write without response' to remote device
#define TX_INDICATIONS_TO_M1        0 //! set below macro to Transmit 'gatt
indications' to remote device
```

Note: By default all BLE configurations are choosen to get high BLE throughput

4. Select WLAN configurations in '**rsi_wlan_config.h**' file provided in the release package at '**..\RSI_SDK_vx\apps\demo\COEX_THROUGHPUT_DEMO_67**' Enter the AP Connectivity essentials configs as the value to SSID, SECURITY_TYPE and PSK

```
#define     SSID                        "Hotspot"
#define     SECURITY_TYPE               RSI_WPA2
```

```
#define     PSK                         "12345678"
```

Channel no in which device should scan choose

```
#define     CHANNEL NO                  0       //! 0 - scan all channels
```

Configure below macros to make use of Local HTTP server to download the files.

```
#define     RSI_DNS_CLIENT    0                 // set to '1' only if using
server name instead of server ip address, by default it is set to '0'
#define     RX_DATA           1                 // set to '1' to RX data from
remote server
#define     HTTPS_DOWNLOAD    0                 // set to '0' to choose HTTP
download
#define     SERVER_PORT       80                // by default http runs on port
80
#define     SERVER_IP_ADDRESS "192.168.0.101"   //Local server ip address
#define     DOWNLOAD_FILENAME   "dltestdata32.txt"  // File to download, by
default this file is provided in the demo
#define     BYTES_TO_RECEIVE    1048576             // size of file configured
under 'DOWNLOAD_FILENAME'
#define     CONTINUOUS_HTTP_DOWNLOAD    1           // set to '1' to download
continuously, if reset download happens only once.
```

Configure below macros to make use of Local HTTPS server to download the files.

```
#define     RSI DNS CLIENT    0                 // set to '1' only if using
server name instead of server ip address, by default it is set to '0'
#define     RX_DATA           1                 // set to '1' to RX data from
remote server
#define     HTTPS DOWNLOAD    1                 // set to '1' to choose HTTPs
download
#define     SERVER PORT       443               // by default https runs on
port 443
#define     SERVER IP ADDRESS "192.168.0.101"   //Local server ip address
#define     DOWNLOAD FILENAME   "dltest.txt"    // File to download, by default
this file is provided in the demo
#define     BYTES_TO_RECEIVE    6144                // size of file configured
under 'DOWNLOAD_FILENAME'
#define     CONTINUOUS_HTTP_DOWNLOAD    1           // set to '1' to download
continuously, if reset download happens only once.
```

Follow below steps to configure local https server

1. Download and install SSL server from https://slproweb.com/products/Win32OpenSSL.html
2. Add the installed location (ex: "C:\Program Files\OpenSSL-Win64\bin") in environment variable 'PATH' and restart the pc to reflect the changes.

Executing the Application:

WLAN throughputs:

UDP/TCP/SSL unidirectional/bidirectional:

1. Compile the project and flash the binary.
2. To measure **WLAN throughput**, run the below iperf commands or ssl scripts
   a. To measure **UDP Tx** throughput, configure module as UDP client and open UDP server in remote port using below command
   iperf.exe -s -u -p <SERVER_PORT> -i 1
   ex: iperf.exe -s -u -p 5001 -i 1
   b. To measure **UDP Rx** througput, configure module as UDP server and open UDP client in remote port using below command
   iperf.exe -c <Module_IP> -u -p <DEVICE_PORT> -i 1 -b<Bandwidth> -t <duration in sec>
   ex: iperf.exe -c 192.168.0.1 -u -p 5001 -i 1 -b50M -t 100
   c. To measure **TCP Tx** throughput, configure module as TCP client and open TCP server in remote port using below command
   iperf.exe -s -p <SERVER_PORT> -i 1
   ex: iperf.exe -s -p 5001 -i 1
   d. To measure **TCP Rx** througput, configure module as TCP server and open TCP client in remote port using below command
   iperf.exe -c <Module_IP> -p <DEVICE_PORT> -i 1 -t <duration in sec>
   ex: iperf.exe -c 192.168.0.1 -p 5001 -i 1 -t 100
   e. To measure **SSL Tx** throughput, configure module in SSL client and follow below steps to run SSL server in windows
   → copy SSL_Server_throughput_d.py from host/sapis/examples/utilities/scripts/ to host/sapis/examples/ utilities/certificates/
   → Open command prompt in folder host/sapis/examples/utilities/certificates/ and run below command
   → **python SSL_Server_throughput_d.py**
   f. To measure **SSL Rx** throughput, configure module in SSL client and follow below steps to run SSL server in windows
   → copy SSL_tx_throughput.py from host/sapis/examples/utilities/scripts/ to host/sapis/examples/utilities/ certificate
   → change port no. from "5001" to the value configured in "SSL_RX_SERVER_PORT"
   → Open command prompt in folder host/sapis/examples/utilities/certificates/ and run below command
   → **python SSL_tx_throughput.py**
   g. To measure TCP bidirectional throughput, configure module as both TCP client and server
   → open TCP server in remote port : **iperf.exe -s -p <SERVER_PORT> -i 1**
   → open TCP client in remote port : **iperf.exe -c <Module_IP> -p <DEVICE_PORT> -i 1 -t <duration in sec>**
   h. To measure UDP bidirectional throughput, configure module as both UDP client and server
   → open UDP server in remote port : **iperf.exe -s -u -p <SERVER_PORT> -i 1**
   → open UDP client in remote port : **iperf.exe -c <Module_IP> -u -p <DEVICE_PORT> -i 1 -b<Bandwidth> -t <duration in sec>**
   i. To measure SSL bidirectional throughput, configure module as both SSL client and server
   → open SSL server in remote port : **python SSL_Server_throughput_d.py**
   → open SSL client in remote port : **python SSL_tx_throughput.py**

   1. while verifying SSL bidirectional throughput, open "**SSL_tx_throughput.py**" and change port no. from "5001" to the value configured in "SSL_RX_SERVER_PORT" 2. Run the UDP /TCP clients after getting the Module IP from application logs.

3. After the program gets executed, Module scans for the configured Access point, connects to it.
4. Acquires the ip address and waits for bt/ble activities to complete if configured.
5. **If BT A2DP is configured**, module inquires for configured bluetooth headset
6. If Headset found, initiate A2DP/AVRCP connections and streams the audio over headset.
7. **If BLE is configured**, Scan for BLE advertise name (RSI_BLE_APP_GATT_TEST) using nRF connect (Android app)/ BLE dongles and initiate ble connection if found.

8. For BLE data transfer, enable Gatt notifications of Module on service characteristic RSI_BLE_ATTRIBUTE_1_UUID (0x1AA1) using nRF connect.
9. If Gatt Notification are enabled, module continuously transmits notifications per connection interval of size 20/232 bytes.
10. Once above activities are completed, Module transmits/receives WLAN packets based on the configuration selected and measures the WLAN throughput

> Note: Verify that all connections are stable and throughput is as expected.

HTTP/HTTPs Unidirectional:

1. Compile the project and flash the binary.
2. Copy the files 'dltestdata32.txt', 'dltest.txt' from below source path and paste in to the destination path.

```
[source path:- ..\RSI SDK vx\apps\demo\COEX THROUGHPUT DEMO 67\]
```

```
[destination path:- ..\RSI SDK vx\resources\scripts\]
```

To download the files from local http server, navigate to below folder and run below command.

```
[File path:- ..\RSI_SDK_vx\resources\scripts\]
```

```
#python simple_http_server.py 80
```

To download the files from local https server, copy ssl certificates 'server-cert.pem' , 'server-key.pem' from below 'source path' and paste in to 'destination path'.

```
[source path:- ..\RSI SDK vx\resources\certificates\]
```

```
[destination path:- ..\RSI SDK vx\resources\scripts\]
```

open command prompt, navigate to above destination path and run below command.

```
#openssl s_server -accept 443 -cert server-cert.pem -key server-key.pem -tls1
-WWW
```

3. After the program gets executed, Module scans for the configured Accesspoint, connects to it and acquires the ip address
4. After acquiring ip address, connects to remote server.(ex: simple_http_server.py running in same network where Module is also connected) and starts http/https download.
5. While downloading, http/https throughput numbers are displayed in serial console.
6. **If BT A2DP is configured**, module inquires for configured bluetooth headset
7. If Headset found, initiate A2DP/AVRCP connections and streams the audio over headset.

8. **IF BLE is configured,** Scan for BLE advertise name (RSI_BLE_APP_GATT_TEST) using nRF connect (Android app)/ BLE dongles and initiate ble connection if found.
9. After successfull connection, enable Gatt notifications of Module on service characteristic RSI_BLE_ATTRIBUTE_1_UUID (0x1AA1) using nRF connect.
10. If Gatt Notification are enabled, module continuously transmits notifications per connection interval of size 232 bytes.
11. While transmitting, observe the new WLAN throughput numbers in console.

> Note: Verify that all connections are stable and throughput is as expected.

BLE throughputs:

1. Compile the project and flash the binary.
2. **If WLAN is configured,** then copy the files 'dltestdata32.txt', 'dltest.txt' from below source path and paste in to the destination path.

```
[source path:- ../RSI SDK vx/apps/demo/COEX THROUGHPUT DEMO 67]
```

```
[destination path:- ..\RSI SDK vx\resources\scripts\]
```

To download the files from local http server, navigate to below folder and run below command.

```
[File path:- ..\RSI_SDK_vx\resources\scripts\]
```

```
#python simple_http_server.py 80
```

To download the files from local https server, copy ssl certificates 'server-cert.pem' , 'server-key.pem' from below 'source path' and paste in to 'destination path'.

```
[source path:- ..\RSI SDK vx\resources\certificates\]
```

```
[destination path:- ..\RSI SDK vx\resources\scripts\]
```

open command prompt, navigate to above destination path and run below command.

```
#openssl s_server -accept 443 -cert server-cert.pem -key server-key.pem -tls1
-WWW
```

3. After the program gets executed, Module scans for the configured Accesspoint, connects to it and acquires the ip address
4. After acquiring ip address, connects to remote server.(ex: simple_http_server.py running in same network where Module is also connected) and starts http download
5. **If BT A2DP is configured**, module inquires for configured bluetooth headset

6. If Headset found, initiate A2DP/AVRCP connections and streams the audio over headset.
7. **After completion of WLAN and BT activites** , Scan for BLE advertise name (RSI_BLE_APP_GATT_TEST) using nRF connect (Android app)/ BLE dongles and initiate ble connection if found.
8. After successfull connection, enable Gatt notifications of Module on service characteristic RSI_BLE_ATTRIBUTE_1_UUID (0x1AA1) using nRF connect.
9. If Gatt Notification are enabled, module continuously transmits notifications per connection interval of size 232 bytes.
10. While transmitting, measure the throughput using BLE sniffer

> Note: Verify that all connections are stable and throughput is as expected.

BT throughputs:

1. Compile the project and flash the binary.
2. **If WLAN is configured,** then copy the files 'dltestdata32.txt', 'dltest.txt' from below source path and paste in to the destination path.

```
[source path:- ../RSI SDK vx/apps/demo/COEX THROUGHPUT DEMO 67]
```

```
[destination path:- ..\RSI_SDK_vx\resources\scripts]
```

To download the files from local http server, navigate to below folder and run below command.

```
[File path:- ..\RSI_SDK_vx\resources\scripts\]
```

```
#python simple_http_server.py 80
```

To download the files from local https server, copy ssl certificates 'server-cert.pem' , 'server-key.pem' from below 'source path' and paste in to 'destination path'.

```
[source path:-..\RSI SDK vx\resources\certificates\]
```

```
[destination path:-..\RSI_SDK_vx\resources\scripts\]
```

open command prompt, navigate to above destination path and run below command.

```
#openssl s_server -accept 443 -cert server-cert.pem -key server-key.pem -tls1
-WWW
```

3. After the program gets executed, Module scans for the configured Accesspoint, connects to it and acquires the ip address
4. After acquiring ip address, connects to remote server.(ex: simple_http_server.py running in same network where Module is also connected) and starts http download

5. **If BLE is configured,** Scan for BLE advertise name (RSI_BLE_APP_GATT_TEST) using nRF connect (Android app)/ BLE dongles and initiate ble connection if found.
6. After successfull connection, enable Gatt notifications of Module on service characteristic RSI_BLE_ATTRIBUTE_1_UUID (0x1AA1) using nRF connect.
7. If Gatt Notification are enabled, module continuously transmits notifications per connection interval of size 232 bytes.
8. Along with BLE and WLAN activities, simultaneously module starts BT inquiry.
9. If device found, initiates A2DP/AVRCP connections and transmits the PCM packets (in SBC encoded format) to headset.
10. While Module is transmitting packets to headset and if configured BLE/WLAN activities are running , measure the BT throughput in sniffer.

> Note: Verify that all connections are stable and throughput is as expected.