Steps to Execute Coex_Max_Demo_57

Silicon Labs Product Documentation

Exported on 07/06/2023

Table of Contents

1 Introduction:

The purpose of this demo is to execute different protocols(BT/BLE/WIFI) individually/combinedly under opermode 0x109.

2 Configurations of Application

1. Select demo from 'rsi_common_app.h'.

   ```
   [File path:- RSI_SDK_WEARABLES_PROJECT/examples/inc/rsi_common_app.h]
   ```

   ```
   #define COEX MAX APP        1 //Set this to 1
   ```

   Note:
   Make sure to set remaining all demo Macros to 0

2. Add **RSI_WITH_OS** macro in below path

   ```
   [File path:- RSI_SDK_WEARABLES_PROJECT/Properties/C/c++Build/Settings/
   GNU ARM CROSS C++ Compiler/ Preprocessor/]
   ```

3 Compile time Configurations:

1. To enable/disable individual protocols configure below macros in 'rsi_common_app.h'

```
#define RSI_ENABLE_BLE_TEST     1 //Set this to 0 to disable BLE
#define RSI ENABLE BT TEST      1 //Set this to 0 to disable BT
#define RSI_ENABLE_PROP_PROTOCOL_TEST       0 //To be kept 0
#define RSI ENABLE WIFI TEST    1 //Set this to 0 to disable WLAN
```

Note:
By default all protocols are selected except WLAN.

2. Select BLE common configurations in 'rsi_ble_config_DEMO_57.h'

```
[File path:- RSI_SDK_WEARABLES_PROJECT/examples/COEX_MAX_DEMO_57/
rsi_ble_config_DEMO_57.h]
```

To select number of BLE connections, configure below macros
Set below macro to required slave connections

```
#define RSI_BLE_MAX_NBR_SLAVES   3
```

Set below macro to required master connections

```
#define RSI_BLE_MAX_NBR_MASTERS  2
```

To identify remote device with BD Address/device name

```
#define CONNECT_OPTION CONN_BY_NAME //CONN_BY_NAME or CONN_BY_ADDR
```

If CONNECT_OPTION is set to CONN_BY_NAME, configure below macros.

```
Add the remote BLE device name to connect

#define RSI_REMOTE_DEVICE_NAME1                       "slave1"
#define RSI_REMOTE_DEVICE_NAME2                       "slave2"
#define RSI_REMOTE_DEVICE_NAME3                       "slave3"
```

If CONNECT_OPTION is set to CONN_BY_ADDR, configure the below macros

```
Configure the address type of remote device as either Public Address or Random
Address
```

```
#define RSI_BLE_DEV_ADDR_TYPE LE_PUBLIC_ADDRESS //!LE_PUBLIC_ADDRESS or
LE_RANDOM_ADDRESS

Add the BD Address of remote BLE device to connect

#define RSI_BLE_DEV_1_ADDR "88:DA:1A:FE:2A:2C"
#define RSI_BLE_DEV_2_ADDR "7E:E6:5E:30:77:6F"
#define RSI_BLE_DEV_3_ADDR "70:1A:69:32:7C:8E
```

Configure below macros to select the profile characteristics uuid for data transfer

```
#define RSI_BLE_CLIENT_WRITE_SERVICE_UUID_M1                0x180D //! Heart
Rate service uuid
#define RSI_BLE_CLIENT_WRITE_CHAR_UUID_M1                   0x2A39 //! Heart
Rate control Point
#define RSI_BLE_CLIENT_WRITE_NO_RESP_SERVICE_UUID_M1        0x1802 //!
Immediate Alert service uuid
#define RSI_BLE_CLIENT_WRITE_NO_RESP_CHAR_UUID_M1           0x2A06 //! Alert
level char uuid
#define RSI_BLE_CLIENT_INIDCATIONS_SERVICE_UUID_M1          0x1809 //! Health
thermometer Alert service uuid
#define RSI_BLE_CLIENT_INIDCATIONS_CHAR_UUID_M1             0x2A1C //!
Temperature measurement
#define RSI_BLE_CLIENT_NOTIFICATIONS_SERVICE_UUID_M1        0x180D //! Heart
Rate service uuid
#define RSI_BLE_CLIENT_NOTIFICATIONS_CHAR_UUID_M1           0x2A37 //! Heart
Rate measurement
```

3. Configure below macros to select each connection configurations, by default this configurations matches with coex max app requirement provided by garden customer.
   Master1 configurations: (where XX=M1)
   Set below macro to enable secure connection between SiLabs device(peripheral) and remote ble device(central)

```
#define SMP ENABLE XX           1

//By default this macro is set to '0'
```

Set below macro to add remote device to whitelist

```
#define ADD TO WHITELIST XX     1

//By default this macro is set to '0'
```

Set below macro to discover remote profiles.

```
#define PROFILE_QUERY_XX     1
```

```
//By default this macro is set to '1'
```

Set below macro to enable data transfer between devices

```
#define DATA TRANSFER XX      1

//By default this macro is set to '1'
```

To select the type of data transfer configure below macros
Set below macro to receive 'gatt notifications' from remote device

```
#define RX NOTIFICATIONS FROM XX      1

//By default this macro is set to '1'
```

Note:
Make sure to set below macros to 0

```
#define     RX INDICATIONS FROM XX     0 //Set this to 0
```

Set below macro to receive 'gatt indications' from remote device

```
#define RX INDICATIONS FROM XX      1

//By default this macro is set to '0'
```

Note:
Make sure to set below macros to 0

```
#define     RX NOTIFICATIONS FROM XX     0 //Set this to 0
```

Set below macro to Transmit 'gatt notifications' to remote device

```
#define TX_NOTIFICATIONS_TO_XX      1

//By default this macro is set to '1'
```

Note:
Make sure to set below macros to 0

```
#define     TX_WRITES_TO_XX                 0 //Set this to 0
#define     TX_WRITES_NO_RESP_TO_XX         0 //Set this to 0
#define     TX_INDICATIONS_TO_XX         0 //Set this to 0
```

Set below macro to Transmit 'gatt write with response' to remote device

```
#define TX_WRITES_TO_XX     1

//By default this macro is set to '0'
```

Note:
Make sure to set below macros to 0

```
#define     TX_NOTIFICATIONS_TO_XX         0 //Set this to 0
#define     TX_WRITES_NO_RESP_TO_XX         0 //Set this to 0
#define     TX_INDICATIONS_TO_XX         0 //Set this to 0
```

Set below macro to Transmit 'gatt write without response' to remote device

```
#define TX WRITES NO RESP TO XX     1

//By default this macro is set to '0'
```

Note:
Make sure to set below macros to 0

```
#define     TX_WRITES_TO_XX                 0 //Set this to 0
#define     TX_NOTIFICATIONS_TO_XX         0 //Set this to 0
#define     TX_INDICATIONS_TO_XX         0 //Set this to 0
```

Set below macro to Transmit 'gatt indications to remote device

```
#define TX INDICATIONS TO XX         1

//By default this macro is set to '0'
```

Note:
Make sure to set below macros to 0

```
#define    TX_WRITES_TO_XX                0 //Set this to 0
#define    TX_WRITES_NO_RESP_TO_XX        0 //Set this to 0
#define    TX_NOTIFICATIONS_TO_XX        0 //Set this to 0
```

To select data length extension for each connection configure below macro
Set below macro to enable data length extension

```
#define DLE_ON_XX        1

//By default this macro is set to '0'
```

Configure below macros to set connection interval, connection latency and connection supervision timeout
Below configuration is for connection interval of 45ms, latency 0 and timeout:400ms

```
#define CONN_INTERVAL_XX            36
#define CONN_LATENCY_XX             0
#define CONN_SUPERVISION_TIMEOUT_XX 400
```

4. Follow the above instructions to select configurations for remaining connections (slave1(XX = S1),slave2 (XX =S2),slave3(XX=S3) and master2(XX=M2))
5. Select BT configurations in 'rsi_bt_config_DEMO_57.h'

```
[File path:- RSI_SDK_WEARABLES_PROJECT/examples/Coex_Max_Demo_57/
rsi_bt_config_DEMO_57.h]
```

Enter the remote BT device address as the value to RSI_BT_REMOTE_BD_ADDR

```
#define     RSI_BT_REMOTE_BD_ADDR                        (void
*)"B8:D5:0B:9B:D6:B2"
```

To select the role configure below macro

```
#define     RSI_APP_AVDTP_ROLE                        INITIATOR_ROLE //
ACCEPTOR_ROLE
// By default initiator role is selected
```

Set below macro to start bt inquiry

```
#define     INQUIRY_ENABLE      1 //By default this macro is set to '1'
```

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

```
#define     INQ_REMOTE_NAME_REQUEST     1 //By default this macro is set to '1'
```

Configure below macro, to check inquiry+connection

```
#define     INQUIRY CONNECTION SIMULTANEOUS     0 //By default this macro is
set to '0'
```

Configure below macro to choose type of audio

```
#define     RSI_AUDIO_DATA_TYPE                 PCM_AUDIO //By default this
macro is set to 'SBC_AUDIO'
```

6. **Configuration for Embedded TCP/IP stack :**
   Select WLAN configurations in 'rsi_wlan_config_DEMO_57.h'

```
[File path:- RSI_SDK_PROJECT/apps/demo/COEX_MAX_DEMO_57/
rsi_wlan_config_DEMO_57.h]
```

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

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

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
#define SERVER IP ADDRESS "192.168.75.27" //! Server IP address
```

6a. For SSL Tx:
Configure below macro to check SSL-TX data transfer.

```
#define     SSL TX DATA     1 //! Enable for SSL Data Transfer
#if SSL TX DATA
#define     TX_DATA         1
#define     MAX_TX_PKTS     10000
#else
#define     TX_DATA         0 //! Enable this to test TCP transmit
#endif
```

Run the below python script at server side.

```
python SSL_Server_throughput_d.py
```

6b. For SSL Rx:
Configure below macro to check SSL-RX data transfer.

```
#define     SSL RX DATA     1 //! Enable for SSL RX Data Transfer
#if SSL RX DATA
#define     RX_DATA         1
#else
#define     RX_DATA         0 //! Enable this to test HTTP/S download
#endif
```

Run the below python script at server side.

```
python SSL_tx_throughput.py
```

6c. For HTTP Download:
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 IP ADDRESS "192.168.0.10"    //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.
```

Run 'simple_http_server.py' using below command and makesure file 'dltestdata32.txt' is provided in the same folder as 'simple_http_server,py'

```
[File path:- RSI_SDK_Vx/resources/scripts/simple_http_server.py]
```

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

6d. For HTTPS Download:
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_IP_ADDRESS "192.168.0.10"    //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.
#define     SSL                           1 //! Enable SSL or not
#define     LOAD_CERTIFICATE              1 //! Load certificate to device
flash
```

Go to below path and start SSL by running below command

```
[File path:- RSI_SDK_Vx/resources/scripts/]
```

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

Note:
1. Download and install SSL server from https://slproweb.com/products/Win32OpenSSL.html
2. Update the installed location in 'PATH' variable ex: "C:\Program Files\OpenSSL-Win64\bin"
2. Make sure to copy 'server-cert.pem' and 'server-key.pem' files in the same directory, where SSL started.
3. Makesure 'dltest.txt' is provided in same folder where SSL server started

7 .Configuration for Host-based TCP/IP stack :

7a.Configuration for HTTP file download with Host-based TCP/IP stack

Need to enable "**#define LWIP_TESTMODE 1**" in opt.h

```
[File path:- RSI_SDK_PROJECT/third_party/lwip/src/src/include/lwip/
opt.h]
```

Need to disable "**#define LWIP_HAVE_LOOPIF 0**" in lwipopts.h

```
[File path:- RSI_SDK_PROJECT/examples/COEX_MAX_DEMO_57/lwipopts.h]
```

Add **ENABLE_RECURSIVE_MUTEXES** macro in the below path

```
[File path:- RSI_SDK_PROJECT/options/"C/C++compiler"/ Preprocessor/
Defined symbols]
```

Select WLAN configurations in 'rsi_wlan_config_DEMO_57.h'

```
[File path:- RSI_SDK_PROJECT/apps/demo/COEX_MAX_DEMO_57/
rsi_wlan_config_DEMO_57.h]
```

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

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

Run the below python script on the server-side.

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

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

```
#define     RSI TCP IP BYPASS             1                // set to '1' only
if using TCP_IP_BYPASS mode (host-based TCP/IP stack), by default it is set to
'0'
#define     RX_DATA                       1                // set to '1' to RX
data from remote server
#define     SERVER IP ADDRESS          "192.168.0.10"      // 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.
#define     VERIFY_RX_DATA                1                //! to verify
receive data, set this to '1'
#define     SOCKTEST INSTANCES MAX        1                //! No. of sockets
to test
#define     SOCKET ASYNC FEATURE          0                //! Set to 1 if
want to receive data asynchronously on TCP socket.Set 0 for synchronous
receive.
```

Note:
HTTP file downloads support only in synchronous mode.

Run 'simple_http_server.py' using the below command and make sure file 'dltestdata32.txt' is provided in the same folder as 'simple_http_server.py'

```
[File path:- RSI_SDK_Vx/resources/scripts/simple_http_server.py]
```

7B.Configuration for TCP TX with Host-based TCP/IP stack :

Need to enable "**#define LWIP_TESTMODE 1**" in opt.h

```
[File path:- RSI_SDK_PROJECT/third_party/lwip/src/src/include/lwip/
opt.h]
```

Need to disable "**#define LWIP_HAVE_LOOPIF 0**" in lwipopts.h

```
[File path:- RSI_SDK_PROJECT/examples/COEX_MAX_DEMO_57/lwipopts.h]
```

Add **ENABLE_RECURSIVE_MUTEXES** macro in the below path

```
[File path:- RSI_SDK_PROJECT/options/"C/C++compiler"/ Preprocessor/
Defined symbols]
```

Select WLAN configurations in 'rsi_wlan_config_DEMO_57.h'

```
[File path:- RSI_SDK_PROJECT/apps/demo/COEX_MAX_DEMO_57/
rsi_wlan_config_DEMO_57.h]
```

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

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

Run the below IPERF server on the server-side.

```
iperf -s -i 1
```

configure the below macros to make use of the transmitting data on the TCP socket.

```
#define     RSI TCP IP BYPASS             1                 // set to '1' only
if using TCP_IP_BYPASS mode (host-based TCP/IP stack), by default it is set to
'0'
#define     TX_DATA                       1                 // set to '1' to RX
data from remote server
#define     SERVER IP ADDRESS          "192.168.0.10"       // Local server ip
address
#define     SOCKTEST INSTANCES MAX        1                 //! No. of sockets
to test
#define     SOCKET ASYNC FEATURE          0                 //! Set to 1 if
want to receive data asynchronously on TCP socket.Set 0 for synchronous
receive.
```

7C.Configuration for Receiving data asynchronously on TCP with Host-based TCP/IP stack :

Need to enable "**#define LWIP_TESTMODE 1**" in opt.h

```
[File path:- RSI_SDK_PROJECT/third_party/lwip/src/src/include/lwip/
opt.h]
```

Need to disable "**#define LWIP_HAVE_LOOPIF 0**" in lwipopts.h

```
[File path:- RSI_SDK_PROJECT/examples/COEX_MAX_DEMO_57/lwipopts.h]
```

Add **ENABLE_RECURSIVE_MUTEXES** macro in the below path

```
[File path:- RSI_SDK_PROJECT/options/"C/C++compiler"/ Preprocessor/
Defined symbols]
```

Select WLAN configurations in 'rsi_wlan_config_DEMO_57.h'

```
[File path:- RSI_SDK_PROJECT/apps/demo/COEX_MAX_DEMO_57/
rsi_wlan_config_DEMO_57.h]
```

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

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

Run TCP server scripts which is capable of transmitting packets.

configure the below macros to make use of the transmitting data on the TCP socket.

```
#define     RSI_TCP_IP_BYPASS             1                // set to '1' only
if using TCP_IP_BYPASS mode (host-based TCP/IP stack), by default it is set to
'0'
#define     RX_DATA                       1                // set to '1' to RX
data from remote server
#define     SOCKTEST_INSTANCES_MAX        1                //! No. of sockets
to test
#define     SOCKET_ASYNC_FEATURE          1                //! Set to 1 if
want to receive data asynchronously on TCP socket.Set 0 for synchronous
receive.
```

Don't modify below macros incase of Host-based TCP/IP stack

Note:
#define SOCKET_ASYNC_FEATURE 0
#define WLAN_THROUGHPUT_ENABLE 0
#define SSL 0 //! Enable SSL or not
#define LOAD_CERTIFICATE 0 //! Load certificate to device flash
#define CONFIGURE_TIMEOUT 0 //! Configuring timeout value

```
#define   HIGH_PERFORMANCE_ENABLE            0
#define WLAN_SYNC_REQ                        0
#define RSI_HTTP_SOCKET_TASK_PRIORITY            2
```

8. Compile the project and flash the binary onto FRDM-K28.
9. Below steps are based on the default configurations provided in the application.
10. Module starts BLE advertising and scanning, starts bt devices inquiry and proprietary protocol simultaneously.
11. Module connects to configured remote device if device in pairing mode and starts playing audio.
12. Module accepts the BLE connections if initiated by remote BLE device(max 2 master connections are accepted) and starts data transfer based on the user configuration.

Phone1: Module enables Gatt notifications of remote device for RSI_BLE_CLIENT_NOTIFICATIONS_CHAR_UUID_M1 (Heart Rate measurement) and receives 20 bytes notifications/45ms connection interval from phone1.

If Phone1 enables notifications of module for service characteristic RSI_BLE_ATTRIBUTE_1_UUID, module transmits 20 bytes notifications continuously

Phone2: Module enables Gatt notifications of remote device for profile characteristic RSI_BLE_CLIENT_NOTIFICATIONS_CHAR_UUID_M1 (Heart Rate measurement) and receives 20 bytes notifications/500ms connection
interval from phone2.

Note: After 2 master connections module advertises with connection interval of 211.25ms in non connectable mode.

13. Connects to configured remote slave devices if devices are advertising and starts data transfer based on the user configuration.

slave1: Module enables Gatt notifications of remote device for RSI_BLE_CLIENT_NOTIFICATIONS_CHAR_UUID_M1 (Heart Rate measurement) and receives 20 bytes notifications/300ms connection interval from slave1.

slave2: Module enables Gatt notifications of remote device for RSI_BLE_CLIENT_NOTIFICATIONS_CHAR_UUID_M1 (Heart Rate measurement) and receives 20 bytes notifications/300ms connection interval from slave2.

slave3: Module enables Gatt notifications of remote device for RSI_BLE_CLIENT_NOTIFICATIONS_CHAR_UUID_M1 (Heart Rate measurement) and receives 20 bytes notifications/300ms connection interval from slave3.

14. Verify that ble connections not degrading the performance of BT audio(there should not be any audio glitches) and ant data transfer.
15. WLAN HTTP/HTTPS download starts based on the flag HTTPS_DOWNLOAD and repeats the download continuously.