AN1349: RS9116 BLE Profiles with AT Commands

Version 1.0

August 3, 2021

Table of Contents

1 About

This Application Note describes in detail the creation of "BLE Profiles with AT Commands" using the RS9116W (NCP) module. This document explains how to prepare BLE Profiles with AT Commands and see the functionality of "Proximity Reporter" by running the example "Proximity Profile" script using the RS9116W (NCP) EVK with UART and USB-CDC ports. By using this Application Note, users can design applications according to their host platforms.

2 Prerequisites

1. Required Windows PC
2. WSC EVK
3. Remote Device (Mobile)
4. EFR Connect application in mobile
5. Install Tera-term application using the URL: https://ttssh2.osdn.jp/index.html.en

3 Terminologies

1. EVK - Evaluation Kit
2. UART - Universal Asynchronous Receive Transmit

4 What is Bluetooth Low Energy Technology?

Bluetooth Low Energy (Bluetooth 5.0) is a new, open standard developed by the Bluetooth SIG. It's targeted to address the needs of new modern wireless applications such as ultra-low power consumption, fast connection times, reliability, and security. Bluetooth Low Energy consumes 10-20 times less power and is able to transmit data 50 times quicker than classical Bluetooth solutions. Bluetooth Low Energy is designed for new emerging applications and markets, but it still embraces the very same benefits we already know from the classical, well established Bluetooth technology:

- **Robustness and reliability** - The adaptive frequency hopping technology used by Bluetooth Low Energy allows the device to quickly hop within a wide frequency band, not just to reduce interference but also to identify crowded frequencies and avoid them. In addition to broadcasting Bluetooth Low Energy also provides a reliable, connection-oriented way of transmitting data.
- **Security** - Data privacy and integrity are always a concern is wireless, mission critical applications. Therefore, Bluetooth Low Energy technology is designed to incorporate high level of security including authentication, authorization, encryption, and man-in-the-middle protection.
- **Interoperability** - Bluetooth Low Energy technology is an open standard maintained and developed by the Bluetooth SIG. Strong qualification and interoperability testing processes are included in the development of technology so that wireless device manufacturers can enjoy the benefit of many solution providers and consumers can feel confident that equipment will communicate with other devices regardless of manufacturer.
- **Global availability** - Based on the open, license free 2.4GHz frequency band, Bluetooth Low Energy technology can be used in worldwide applications.

Key features of Bluetooth Low Energy wireless technology include:

- Ultra-low peak, average and idle mode power consumption
- Ability to run for years on standard, coin-cell batteries
- Low cost
- Multi-vendor interoperability
- Enhanced range

Bluetooth Low Energy is also meant for markets and applications, such as:

- Automotive
- Consumer electronics
- Smart energy
- Entertainment
- Home automation
- Security & proximity
- Sports & fitness

5 Bluetooth 5.0 Architecture

5.1 Overview

- **Server**

  Service is the device that provides the information, so these are typically the sensor devices, like thermometers or heart rate sensors. The server exposes implement services, and the services expose the data in characteristics.

- **Client**

  The client is the device that collects the information for one or more sensors and typically either displays it to the user or passes it forward. The client devices typically do not implement any service, but just collect the information from the service provided by the server devices. Clients are typically devices like mobile phones, tablets, and PCs.

The figure below shows the relationship of these two roles.

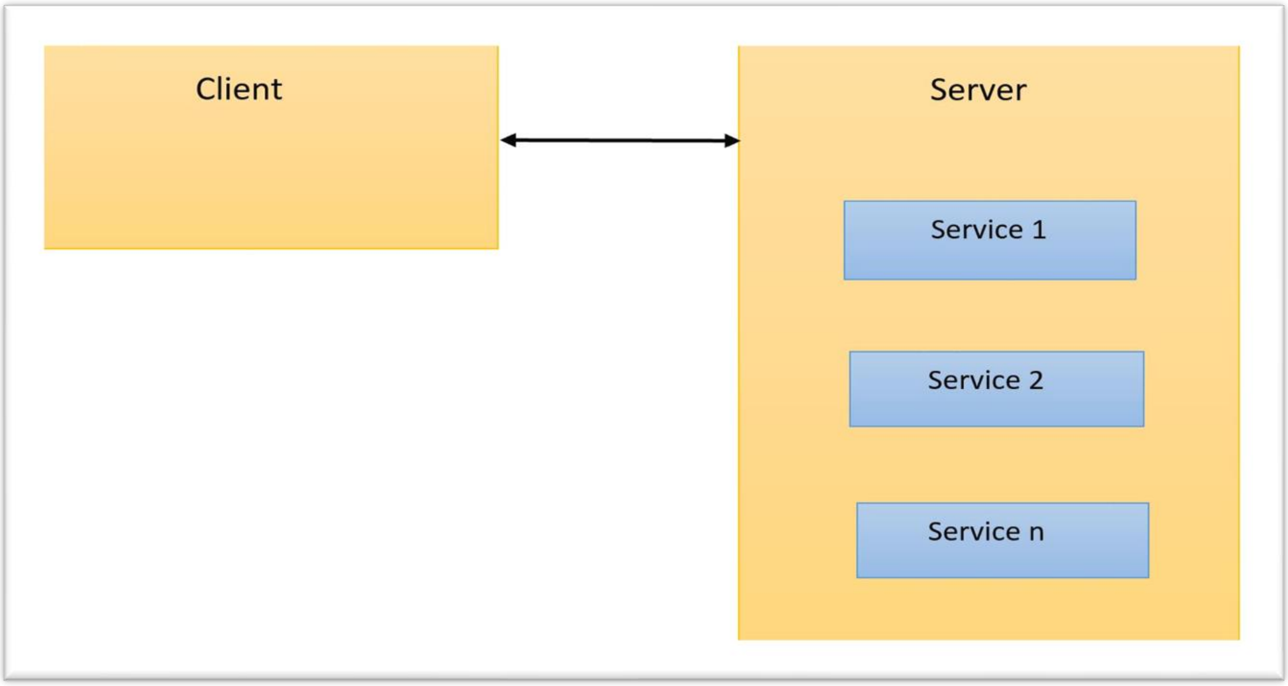


Figure 1: **Bluetooth Low Energy Device Roles**

5.2 Profile

Profiles are used to describe devices and the data they expose and how these devices behave. The data is described by using services, which are explained later, and a profile may implement single or multiple services depending on the profile specification. For example, a Proximity Profile specification mandates that the following services need to be implemented:

- Link Loss Service

Profile specifications might also define other requirements such as security, advertisement intervals, and connection parameters. The purpose of profile specifications is to allow device and software vendors to build standardized interoperable devices and software. Standardized profiles have globally unique 16-bit UUID, so they can easily identify.

5.3 Service

Services such as a Link Loss service describe what kind of data a device exposes, how the data can be accessed, and the security requirements for that data. The data is described using characteristics and a service may contain single or multiple characteristics and some characteristics might be optional whereas some are mandatory.

Two types of services exist:

• **Primary Service**

A primary service is a service that exposes primary usable functionality of this device. A primary service can be included by another service.

• **Secondary Service**

A secondary service is a service that is subservient to another secondary service or primary service. A secondary service is only relevant in the context of another service. Just like the profiles also the services are defined in service specifications. Every service standardized by the Bluetooth SIG has a globally unique 16-bit UUID so just like the profiles also the services can be easily identified. However not every use case can be fulfilled by the standardized service and therefore the Bluetooth Low Energy specification enables device vendors to make proprietary service. The proprietary services are described just as the standardized services, but 128-bit UUIDs need to be used instead of use 16-bit UUIDs reserved for the standard services.

5.4 Characteristic

Characteristics are used to expose the actual data. Characteristic is a value, with a known type (UINT8, UINT16, UTF-8 etc.), a known presentation format. Just like profiles and services also characteristics have unique UUID so they can be easily identified, and the standardized characteristics use 16-bit UUIDs, and vendor specific characteristics use 128-bit UUIDs.

Characteristics consist of:

• **Characteristic Declaration** describing the properties of characteristic value such as:

  • characteristic (UUID)

  • Access control (read, write, indicate etc.)

  • Characteristic value handle (unique handle within a single device)

• **Characteristic Value** containing the value of a characteristic (for example temperature reading).

• **Characteristic Descriptor(s)** which provide additional information about the characteristic (characteristic user description, characteristic client configuration, vendor specific information etc.).

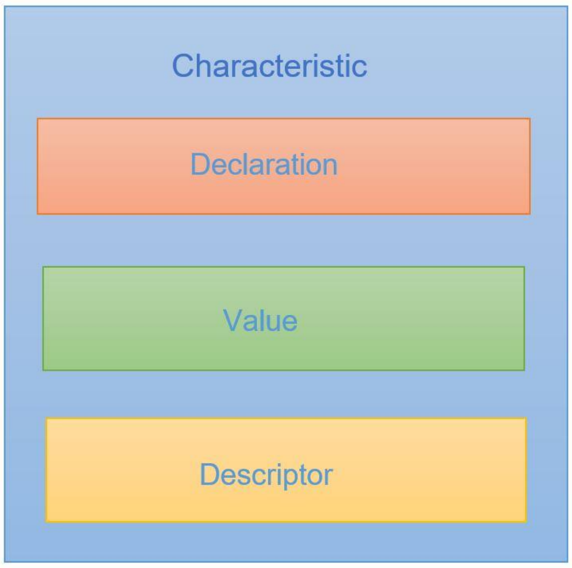


Figure 2: Characteristic Structure

5.5 Relationship Between Profiles, Services, and Characteristics

The illustration below shows the relationship between profiles, services, and characteristics.

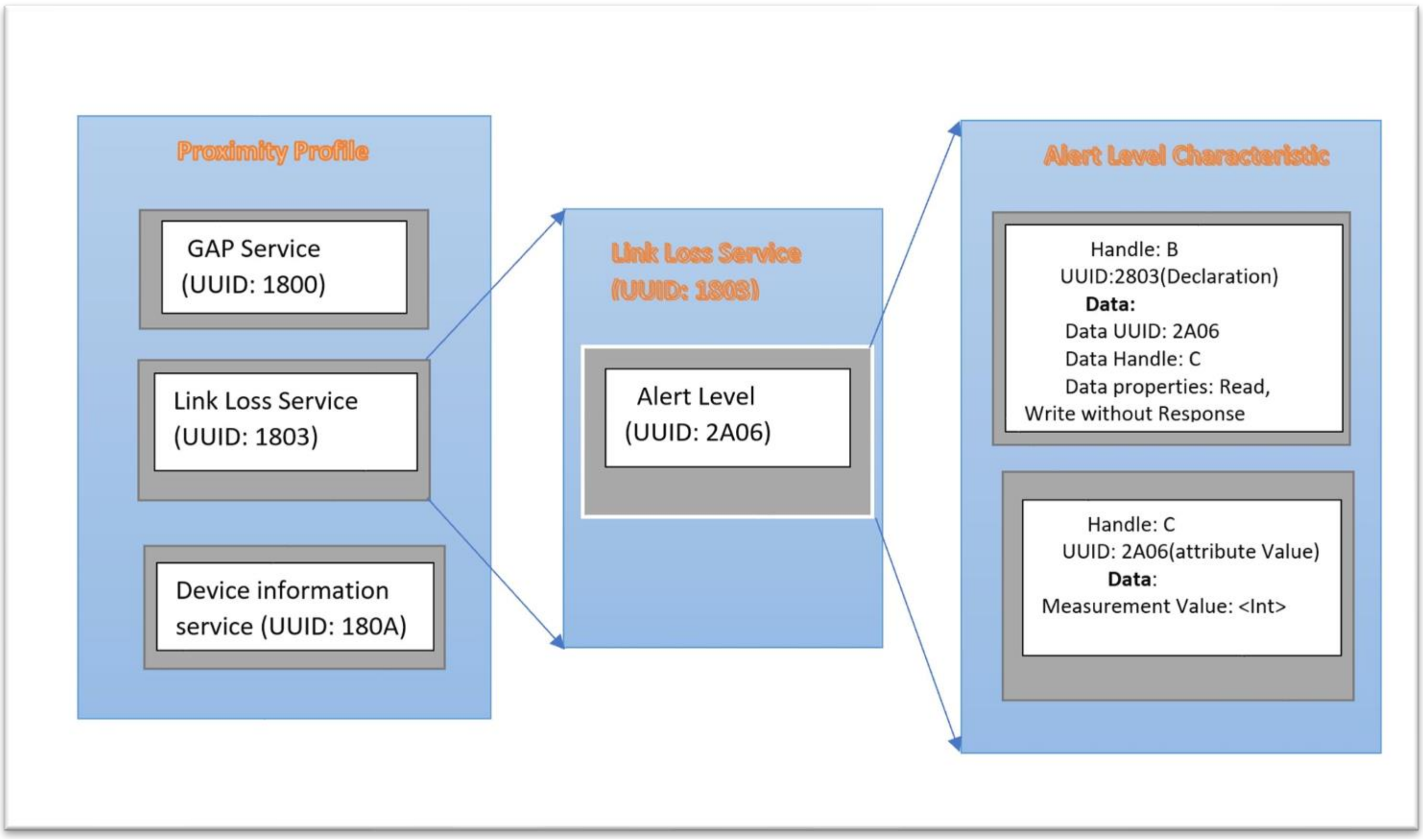

Proximity Profile
GAP Service (UUID: 1800)
Link Loss Service (UUID: 1803)
Device information service (UUID: 180A)
Link Loss Service (UUID: 1803)
Alert Level (UUID: 2A06)
Alert Level Characteristic
Handle: B
UUID:2803(Declaration)
Data:
Data UUID: 2A06
Data Handle: C
Data properties: Read, Write without Response
Handle: C
UUID: 2A06(attribute Value)
Data:
Measurement Value: <Int>

Figure 3: Proximity Profile

6 BLE Profiles and Services

Below is a list of Profiles & Services supported by the RS9116.

S. No	Profile Name	Service UUID
1	Generic Access	0x1800
2	Generic Attribute	0x1801
3	Immediate Alert	0x1802
4	Link Loss	0x1803
5	Tx Power	0x1804
6	Glucose	0x1808
7	Health Thermometer	0x1809
8	Blood Pressure	0x1810
9	HID	0x1812
10	Heart Rate Service	0x180D
11	Battery	0x180F

Notes:

1. Please refer to the below link for the list of standard services and the Characteristics defined by the BT-SIG.

   https://btprodspecificationrefs.blob.core.windows.net/assigned-values/16-bit%20UUID%20Numbers%20Document.pdf

2. Similarly Customized 16bit,32bit, and 128bit UUIDs can be added as a service to RS9116.

7 Proximity Profile v1.0

7.1 Description

The Proximity profile defines the behavior when a device moves away from a peer device so that the connection is dropped or the path loss increases above a preset level, causing an immediate alert. This alert can be used to notify the user that the devices have become separated.

Proximity Profile defines two roles:

- **The Proximity Reporter**

  The Proximity Reporter measures the RSSI of the Remote Device and compare with the threshold which is fixed in application. If the RSSI cross the threshold level, then reporter will be alarmed with the selected Alert level. The sensor also contains the Device Information Service, which contains information for example about the manufacturer of the device. The Proximity Reporter is the GATT Server.

- **The Proximity Monitor**

  After a successful connection proximity monitor will select the required alert level to indicate the alarm, whenever the proximity reporter moves away and the proximity monitor access the information exposed by the "Proximity Reporter" and can for example display it to the end user. The Proximity Monitor is the GATT Client.

The figure below shows the relationship of these two roles.

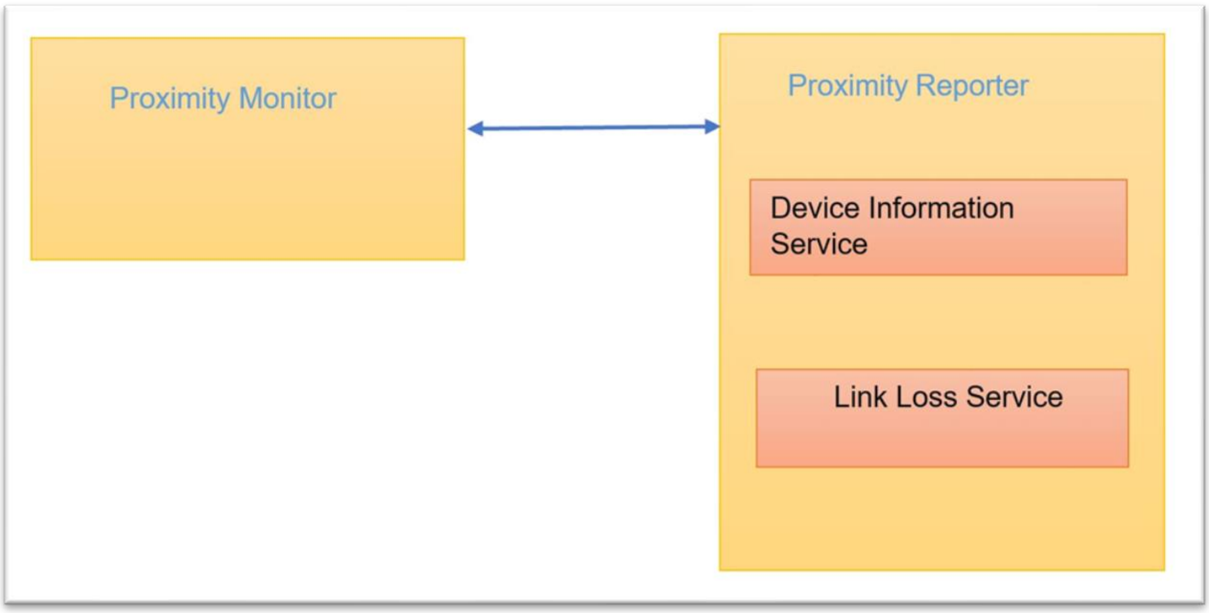


Figure 4. Proximity Profile Roles

Note: The Proximity Reporter has an instance of the Link Loss service, and optionally both the Immediate Alert and the Tx Power service.

8 Profile Usage

Typically, the Proximity Profile is used in most industries. Below is a short list.

- Offices (smart office, checking-in employees, booking conference rooms)
- Retail (loyalty programs, proximity marketing)
- Hospitality (personalization, loyalty, automatic ordering, proximity marketing)
- Museums (wayfinding, audio guides, notifications with extra context)
- Events (checking-in visitors, heat maps, personalization, treasure hunts)
- Logistics (checking-in employees, presence verification, automating deliveries)
- Security (presence verification, checking-in employees)
- Accessibility (wayfinding, visit insights)
- Airports (automated payments, check-in, wayfinding, heat maps)
- Transportation (driver's verification, automatic ticket validation)
- Sports (personalization, presence verification)
- Real estate & showrooms (notifications with product details, interactive experience)

9 Command Sequence

This section details the Command sequence that we are going to use for the Proximity profile.

9.1 Opermode

This is the first command that needs to be sent from the Host after receiving card ready frame from module. This command configures the module in different functional modes.

Command for the Opermode:

```
at+rsi_opermode=<oper_mode>, <feature_bit_map>,
<tcp_ip_feature_bit_map>,<custom_feature_bit_map>,<ext_custom_feature_bit_map>,<bt_custom_feature_bit_ma
p>,<ext_tcp_ip_feature_bit_map>,<ble_custom_feature_bit_map>,<ble_custom_ext
_feature_bit_map>,<config_feature_bit_map>\r\n
```

Note:

If BIT (31) is set to ‘1’ in custom_feature_bitmap at+rsi_opermode=<oper_mode>, <feature_bit_map>, <tcp_ip_feature_bit_map>, <custom_feature_bitmap><ext_ custom_feature_bit_map>\r\n

If BIT (31) is set to ‘1’ in tcp_ip_feature_bit_map at+rsi_opermode=<oper_mode>, <feature_bit_map>, <tcp_ip_feature_bit_map>, <custom_feature_bitmap><ext_ tcp_ip_feature_bit_map>\r\n

If BIT (31) is set to ‘1’ in both custom_feature and ext_custom_feature bit maps
at+rsi_opermode=<oper_mode>, <feature_bit_map>, <tcp_ip_feature_bit_map>,<custom_feature_bitmap><ext_custom_feature_bit_map> <bt_custom_feature_bit_map>\r\n

If BIT(31) is set to 1 in bt_custom_feature_bit_map
at+rsi_opermode=<oper_mode>,<feature_bit_map>,<tcp_ip_feature_bit_map>,<custom_feature_bitmap><ext_custom_feature_bit_map><bt_custom_feature_bit_map><ext_tcp_ip_feature_bit_map><ble_custom_feature_bit_map>\r\n

If BIT(31) is set to 1 in ble_custom_feature_bit_map
at+rsi_opermode=<oper_mode>,<feature_bit_map>,<tcp_ip_feature_bit_map>,<custom_feature_bitmap><ext_custom_feature_bit_map><bt_custom_feature_bit_map><ext_tcp_ip_feature_bit_map><ble_custom_feature_bit_map>,<ble_custom_ext_feature_bit_map>\r\n

If BIT(31) is set to '1' in both tcp_ip_feature_bit_map and ext_tcp_ip_feature_bit_map
at+rsi_opermode=<oper_mode>,<feature_bit_map>,<tcp_ip_feature_bit_map>,<custom_feature_bitmap><ext_custom_feature_bit_map><bt_custom_feature_bit_map><ext_tcp_ip_feature_bit_map><ble_custom_feature_bit_map>,<ble_custom_ext_feature_bit_map>,<config_feature_bit_map>\r\n

Parameters:

Oper_mode (4 bytes): Sets the mode of operation. oper_mode contains two parts <wifi_oper_mode, coex_mode>. Lower two bytes represent wifi_oper_mode and higher two bytes represent coex_mode. oper_mode = ((wifi_oper_mode) | (coex_mode << 16))

Custom_feature_bit_map: Enable "bit (31)" to enable the "ext_custom_feature_bit_map".

Ext_custom_feature_bit_map: Enable the "bit (20) & bit (21)" for enabling the 384K memory. Enable

"bit (31)" for "bt_custom_feature_bit_map".

Bt_custom_feature_bit_map: Enable "bit (30)" to enable the "internal Rf type selection" & Enable "bit (31)" to enable the "Valid_ble_feature_bit_map".

Ble_custom_feature_bit_map: In this parameter enabled the "bit (16) to bit (20)" this is for BLE_TX_POWER selection.

Give 31 as ble tx power index (e.g., 31<<16)

This variable is used to select the ble tx power index value. The following are the possible values.

Default Value for BLE Tx Power Index is 31

The range for the BLE Tx Power Index is 1 to 75 (0, 32 index is invalid)

1 - 31 BLE -0DBM Mode

33 - 63 BLE- 10DBM Mode

64- 75 BLE - HP Mode

Example: -

at+rsi_opermode=851968,0,1,2147483648,2150629376,3221225472,0,1075773440\r\n

OK\r\n

bt_loaded\r\n

9.2 Set Local Name

This is used to set name to the local device.

Command for the "SetlocalName":

at+rsibt_setlocalname=<Name Length>, <Name> \r\n

Parameters:

Name Length	Length of the name of local device
Name (50 bytes)	Name of the local device

Example:

at+rsibt_setlocalname=1, RS9116W_BLE\r\n

OK\r\n

9.3 Get Local Address

This is used to query the BD address of the local device.

Command for the "Get_localbdaddr":

at+rsibt_getlocalbdaddr? \r\n

Example:

OK 88-DA-1A-68-69-70\r\n

9.4 Service Requirements

This is used to add the new service Record in BLE GATT record list. If service is created successfully service record handle is returned, else error value is returned.

Service creation command:

at+rsibt_addservice=<uuid_size>, <ServiceUUID>, <NbrAttributes>, <MaxAttDataSize>\r\n

Parameters:

ServiceUUID – BLE supporting service UUID value.

NbrAttributes (2 bytes) – number of attributes need to add for this service.

MaxAttDataSize (2 bytes) – maximum number of data length that can be used in attribute list.

Result Code	Description
OK, < ServiceHndlerPtr >, < StartHndl >	Command Success.
ERROR <Error code>	Command Fail.

Response Parameters:

ServiceHndlerPtr – created GATT service record handle.

StartHndl (2 bytes) - service record starting attribute handle value.

Example:
The table below describes the service requirements.

Service	UUID	Heart Rate Sensor
Link Loss Service	1803	Mandatory

Table 1: Service Requirements

at+rsibt_addservice=2,1803,3,30

OK 5B650, A

9.5 Add Attribute

This is used to add the attribute record to the specific service using service record handle.

Command for Add Attribute Record:

at+rsibt_addattribute=<ServiceHndlerPtr>, <Hndl>, <AttUUIDsize>, <AttUUID>, <prop>, <DataLen>, <ConfigBitmap>, <Data>\r\n

Parameters:

ServiceHndlerPtr – service record handle.

Hndl (2 bytes) – handle of the attribute record.

AttUUID – attribute record UUID.

Prop (1 byte) – property of the attribute. Below is the description of the Property parameter.

Properties	Value	Description

Broadcast	0x01	If set, permits broadcasts of the Characteristic Value using Server Characteristic Configuration Descriptor. If set, the Server Characteristic Configuration Descriptor shall exist.
Read	0x02	If set, permits reads of the Characteristic Value using procedures
Write without response	0x04	If set, permit writes of the Characteristic Value without response using procedures
Write	0x08	If set, permits writes of the Characteristic Value with response using procedures
Notify	0x10	If set, permits notifications of a Characteristic Value without acknowledgment using the procedure. If set, the Client Characteristic Configuration Descriptor shall exist.
Indicate	0x20	If set, permits indications of a Characteristic Value with acknowledgment using the procedure. If set, the Client Characteristic Configuration Descriptor shall exist.
Authenticated Signed Writes	0x40	If set, permits signed writes to the Characteristic Value using the procedure
Extended properties	0x80	If set, additional characteristic properties are defined in the Characteristic Extended Properties Descriptor. If set, the Characteristic Extended Properties Descriptor shall exist.

DataLen (2 bytes) – attribute record data length.

Data (max 20 bytes) – attribute record data value.

Example:

The table below describes the structure and requirements for the Alert Level characteristic.

Characteristic	**UUID**	**Type**	**Support**	**Security**	**Properties**
Alert Level	2A06	8bit	Mandatory	none	Read, Write_without_Response

Table 2: Alert Level Characteristic Structure

```
at+rsibt_addattribute=5B650, B,2,2803,10,6,0,8,0,0C,00,06,2A
OK\r\n
```

9.6 BLE Set_Advertise_Data

This command is used to set the advertise data to expose remote devices.

Command for "Set_advertise_data":

```
at+rsibt_setadvertisedata=<DataLen>, <Data>\r\n
```

Parameters: -

Length (1 byte)	Data length. Max advertises data length is 31
Data (31 bytes)	Actual data

Example:

at+rsibt_setadvertisedata=10,2,1,6, C,9,52,53,39,31,31,36,57,5F,42,4C,45\r\n

OK\r\n.

9.7 Advertise

This is used to expose or advertise about the local device to the remote BT devices.

Command for "Advertise":

at+rsibt_advertise=< Status >, < AdvertiseType >,< FilterType >,<DirectAddrType>,<DirectAddr>,< adv_int_min >,< adv_int_max >,< own_add_type >,< adv_channel_map >\r\n

Parameters:

Note: All parameters should be in decimal except DirectAddr, it should be in hexadecimal.

Status (1 byte) – To enable/disable Advertising.

1	Enable Advertising
0	Disable Advertising

Advertise Type (1 byte)

State	Description
0x80	Connectable undirected
0x81	Connectable directed with high duty cycle
0x82	Scannable undirected
0x83	Non connectable undirected
0x84	Connectable directed with low duty cycle

Filter Type (1 byte)

Filter Type	Description
0	Allow Scan Request from Any, Allow Connect Request from Any.
1	Allow Scan Request from White List Only, Allow Connect Request from Any.
2	Allow Scan Request from Any, Allow Connect Request from White List Only.
3	Allow Scan Request from White List Only, Allow Connect Request from White List Only.

DirectAddrType (1 byte)

0	Public Address
1	Random Address

DirectAddr (1 byte)- Remote device BD Address

adv_int_min (2 bytes)

Value	Parameter Description
N = 0xXXXX	Minimum advertising interval for non-directed advertising. Range: 0x0020 to 0x4000 Default: N = 0x0800 (1.28 second) Time = N * 0.625 msec Time Range: 20 ms to 10.24 sec.

adv_int_max (2 bytes)

Value	Parameter Description
N = 0xXXXX	Minimum advertising interval for non-directed advertising. Range: 0x0020 to 0x4000 Default: N = 0x0800 (1.28 second) Time = N * 0.625 msec Time Range: 20 ms to 10.24 sec.

own_add_type (1 byte)

Value	Parameter Description
0x00	Public Device Address (default)
0x01	Random Device Address
0x02 – 0xFF	Reserved for future use

adv_channel_map (1 byte)

Value	Parameter Description
00000000b	Reserved for future use
xxxxxxx1b	Enable channel 37 use
xxxxxx1xb	Enable channel 38 use
xxxxx1xxb	Enable channel 39 use
00000111b	Default (all channels enabled)

Example:

at+rsibt_advertise=1,128,0,0,0,60,70,0,7\r\n.

OK\r\n.

Note: For scannable undirected and non-connectable undirected advertising modes, minimum advertising interval should be 41 ms and maximum advertising interval should be 1.28 s.

9.8 Query RSSI

This is used to query RSSI of the connected remote BD device.

Command for "Query RSSI":

at+rsibt_getrssi=<BDAddress? \r\n

Parameters:

BDAddress(6 bytes) - BT Address of the connected remote device.

Result Code	Description
OK <rssi value>	Command Success
ERROR <Error_code>	Command Fail.

Response parameters:

RSSI – RSSI value of the connected remote device

Example:

at+rsibt_getrssi=AA-BB-CC-DD-EE-FF? \r\n

OK 130\r\n

9.9 Set Local Attribute Value

This is used to set/change the local attribute record value to the specific service using service record handle.

Command for "Set Local Attvalue":

at+rsibt_setlocalattvalue=<Hndl>, <DataLen>, <Data>\r\n

Parameters: -

Parameter	Description
Hndl (2 bytes)	handle of the attribute record.
DataLen (2 bytes)	attribute record data length
Data (31 bytes)	attribute record data value.

Example:

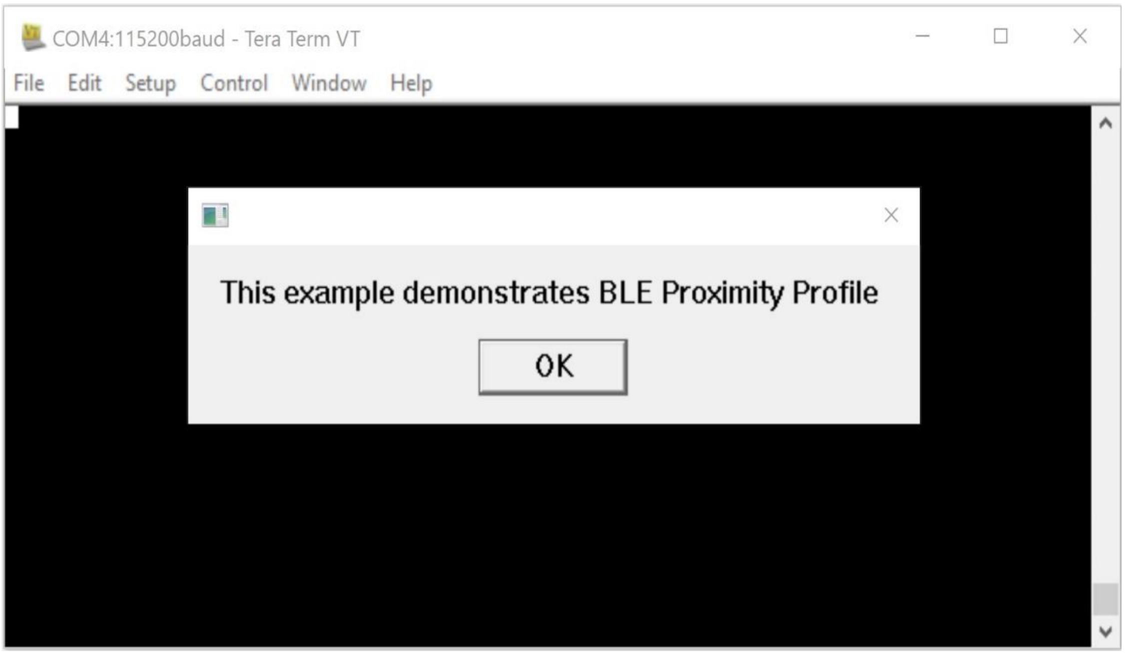
at+rsibt_setlocalattvalue=C,1,1\r\n

OK\r\n

10 Tera-term Script Execution

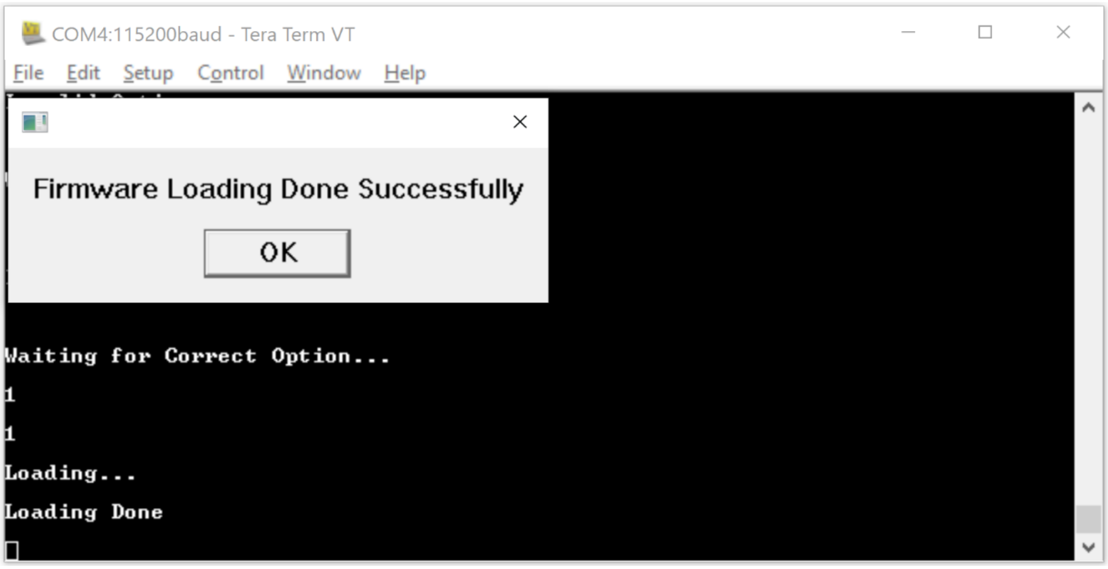
Download and install Tera Term using the link:https://ttssh2.osdn.jp/index.html.en.

1. When we run the "Proximity_Profile.ttl" script mentioned in "Appendix" using the Tera Term, following things will execute one after another.


COM4:115200baud - Tera Term VT
File Edit Setup Control Window Help
OK

This example demonstrates BLE Proximity Profile

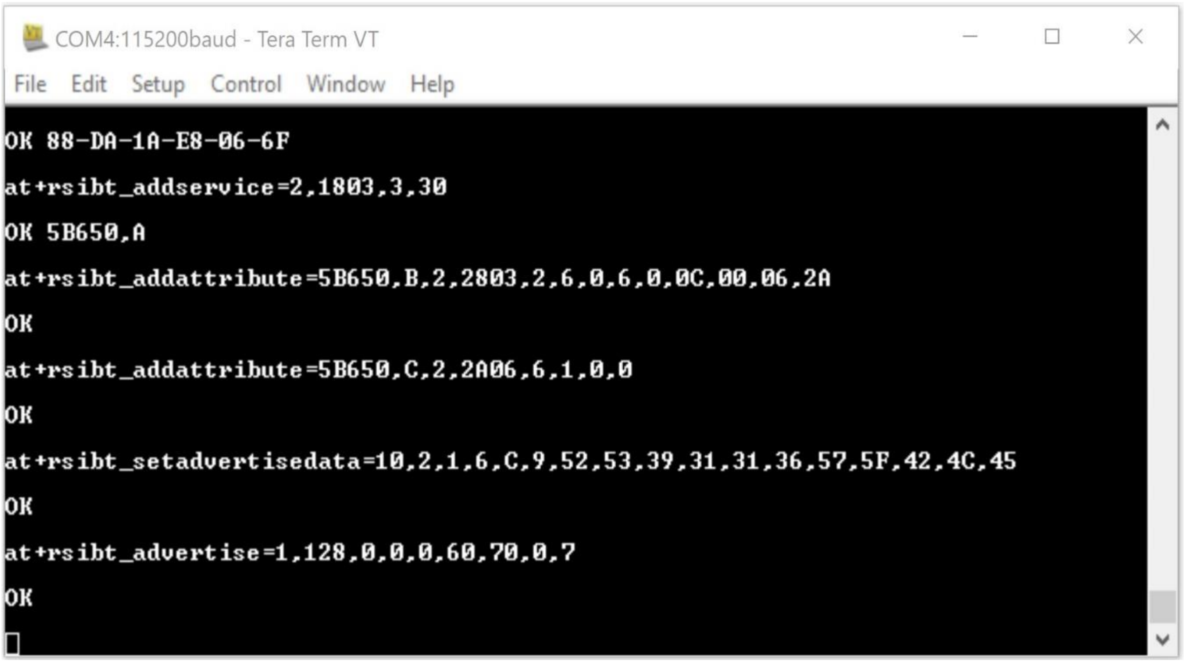
2. The ABRD process will run initially, and you will see a pop-up as "Firmware Loading Done "


COM4:115200baud - Tera Term VT
File Edit Setup Control Window Help
OK

Firmware Loading Done Successfully

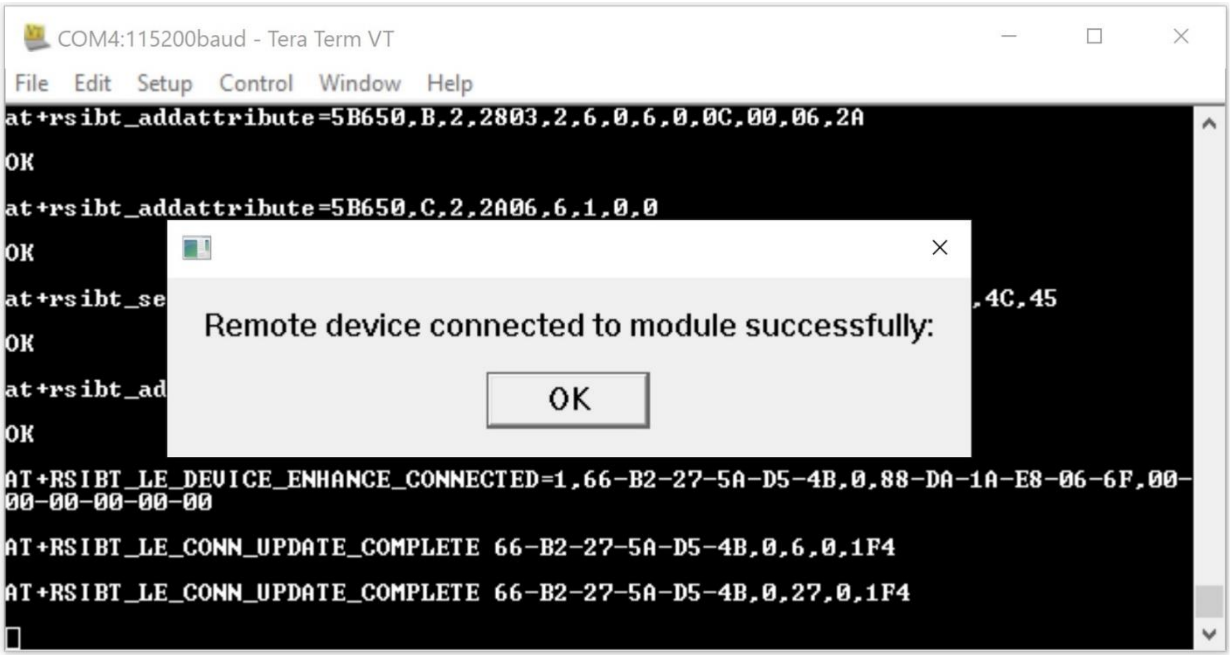
```
Waiting for Correct Option...
1
1
Loading...
Loading Done
```

3. Command sequence will execute one by one, the device is now in advertising state.


COM4:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
OK 88-DA-1A-E8-06-6F
at+rsibt_addservice=2,1803,3,30
OK 5B650,A
at+rsibt_addattribute=5B650,B,2,2803,2,6,0,6,0,0C,00,06,2A
OK
at+rsibt_addattribute=5B650,C,2,2A06,6,1,0,0
OK
at+rsibt_setadvertisedata=10,2,1,6,C,9,52,53,39,31,31,36,57,5F,42,4C,45
OK
at+rsibt_advertise=1,128,0,0,0,60,70,0,7
OK
```

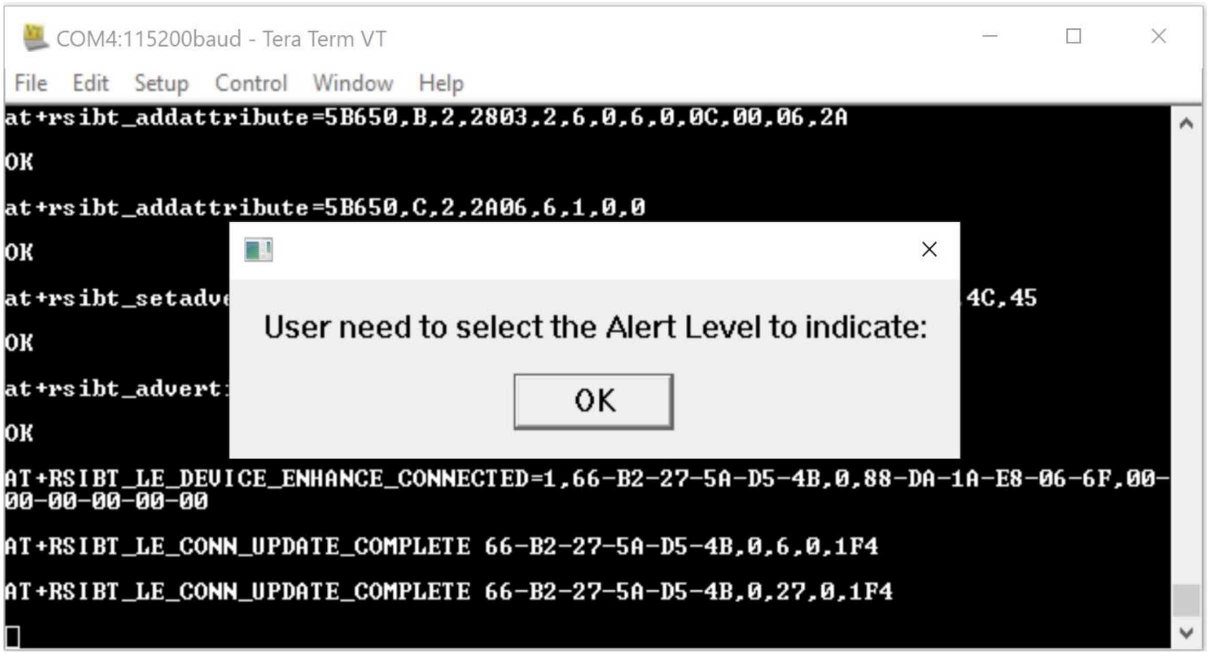
4. When you scan through the Remote BT device, it will show the scan results. Once you initiate the connection, it will connect.


COM4:115200baud - Tera Term VT
File Edit Setup Control Window Help
OK

```
at+rsibt_addattribute=5B650,B,2,2803,2,6,0,6,0,0C,00,06,2A
OK
at+rsibt_addattribute=5B650,C,2,2A06,6,1,0,0
OK
at+rsibt_se                                                    ,4C,45
OK
at+rsibt_ad
OK
AT+RSIBT_LE_DEVICE_ENHANCE_CONNECTED=1,66-B2-27-5A-D5-4B,0,88-DA-1A-E8-06-6F,00-
00-00-00-00-00
AT+RSIBT_LE_CONN_UPDATE_COMPLETE 66-B2-27-5A-D5-4B,0,6,0,1F4
AT+RSIBT_LE_CONN_UPDATE_COMPLETE 66-B2-27-5A-D5-4B,0,27,0,1F4
```

Remote device connected to module successfully:

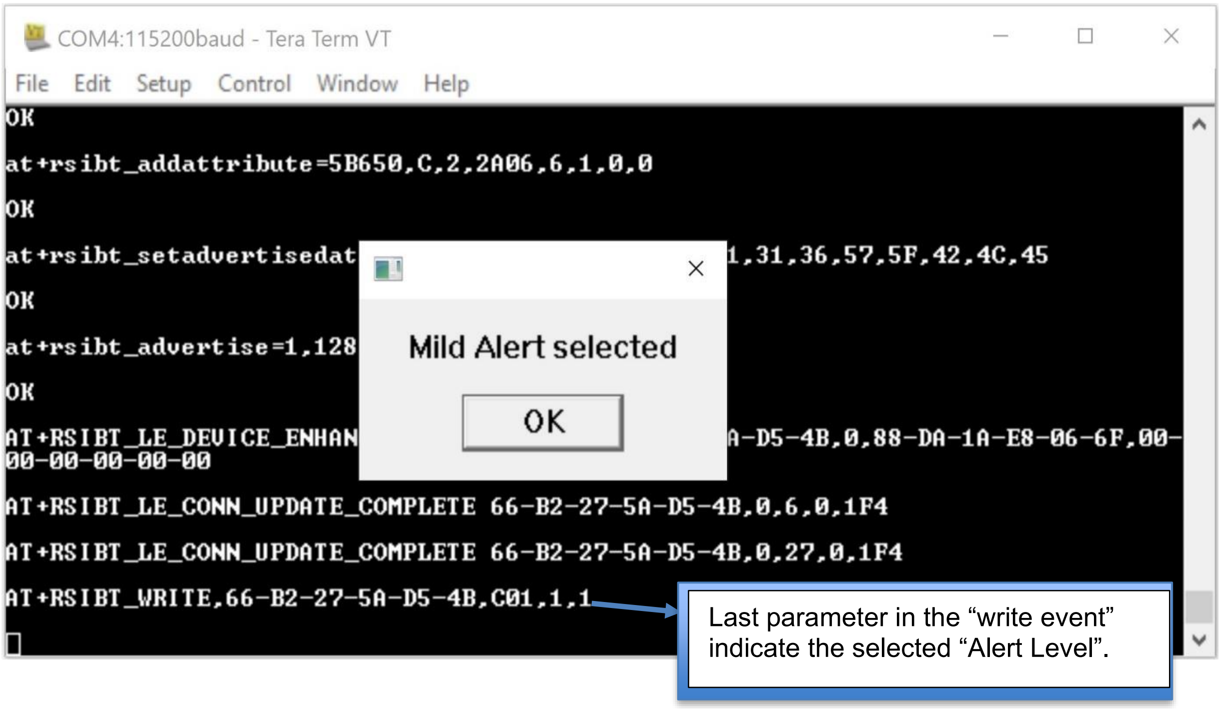
5. After successful connection, the script will ask for selection of “Alert Level” for Indication.


COM4:115200baud - Tera Term VT
File Edit Setup Control Window Help
OK

```
at+rsibt_addattribute=5B650,B,2,2803,2,6,0,6,0,0C,00,06,2A
OK
at+rsibt_addattribute=5B650,C,2,2A06,6,1,0,0
OK
at+rsibt_setadve                                        4C,45
OK
at+rsibt_advert
OK
AT+RSIBT_LE_DEVICE_ENHANCE_CONNECTED=1,66-B2-27-5A-D5-4B,0,88-DA-1A-E8-06-6F,00-
00-00-00-00-00
AT+RSIBT_LE_CONN_UPDATE_COMPLETE 66-B2-27-5A-D5-4B,0,6,0,1F4
AT+RSIBT_LE_CONN_UPDATE_COMPLETE 66-B2-27-5A-D5-4B,0,27,0,1F4
```

User need to select the Alert Level to indicate:

Once select the “Alert Level” in Remote device, write event comes here. In Remote Device when you click “Write_without_Response” property it will show 3 Alert Levels. 1. No Alert 2. Mild Alert 3. High Alert.


COM4:115200baud - Tera Term VT
File Edit Setup Control Window Help
OK

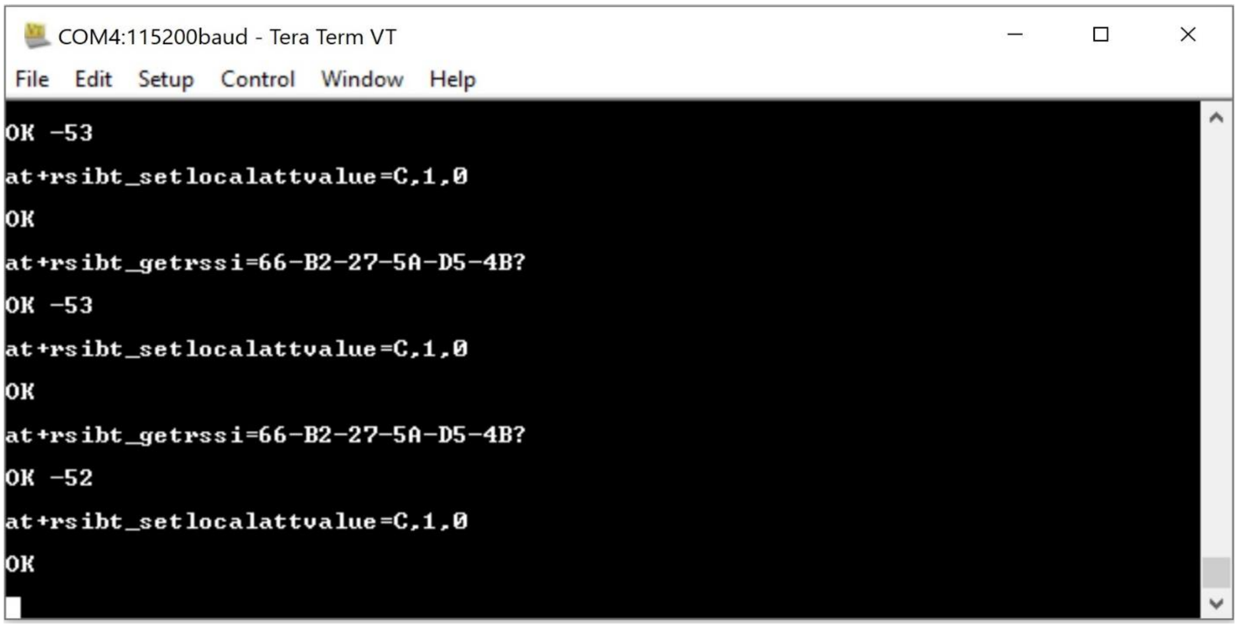
```
OK
at+rsibt_addattribute=5B650,C,2,2A06,6,1,0,0
OK
at+rsibt_setadvertisedat      1,31,36,57,5F,42,4C,45
OK
at+rsibt_advertise=1,128
OK
AT+RSIBT_LE_DEVICE_ENHAN      A-D5-4B,0,88-DA-1A-E8-06-6F,00-
00-00-00-00-00
AT+RSIBT_LE_CONN_UPDATE_COMPLETE 66-B2-27-5A-D5-4B,0,6,0,1F4
AT+RSIBT_LE_CONN_UPDATE_COMPLETE 66-B2-27-5A-D5-4B,0,27,0,1F4
AT+RSIBT_WRITE,66-B2-27-5A-D5-4B,C01,1,1
```

Mild Alert selected

Last parameter in the “write event” indicate the selected “Alert Level”.

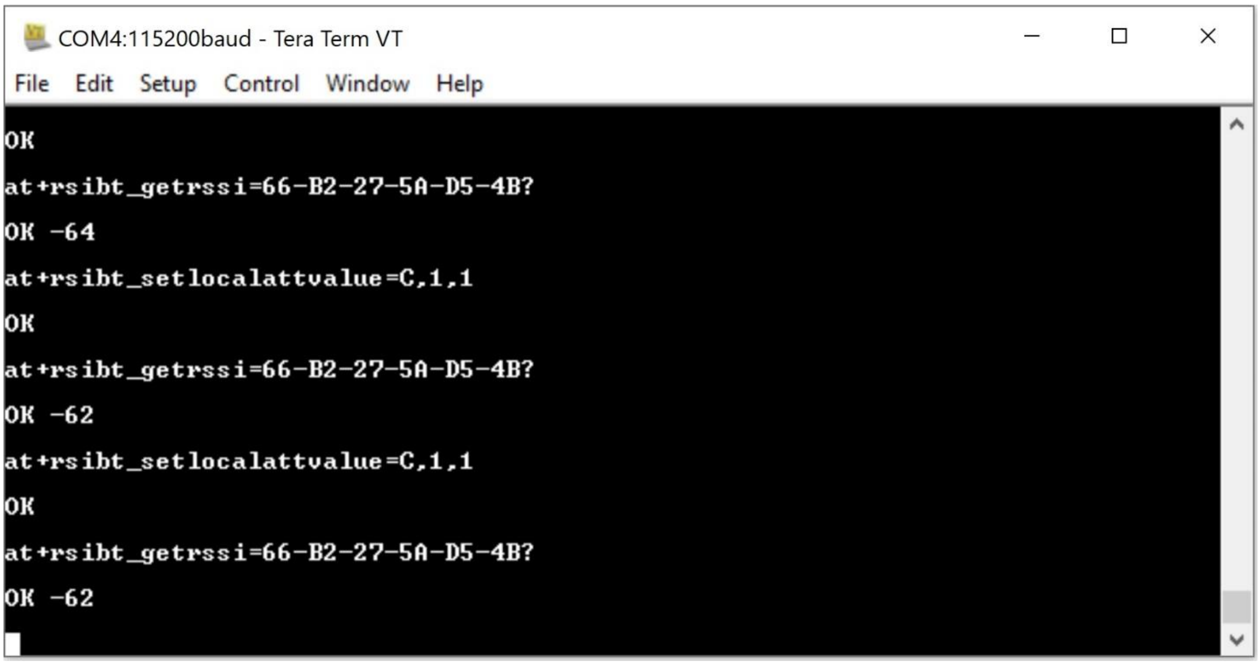
6. After selection of the “Alert Level”, script will continuously run-in loop and based on the RSSI of the Remote Device, Proximity Reporter (RS9116) will indicate the mentioned Alert level to the Remote device. This can check in remote device using “Read” property.

Note: - Threshold mentioned in script is -85.


COM4:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
OK -53
at+rsibt_setlocalattvalue=C,1,0
OK
at+rsibt_getrssi=66-B2-27-5A-D5-4B?
OK -53
at+rsibt_setlocalattvalue=C,1,0
OK
at+rsibt_getrssi=66-B2-27-5A-D5-4B?
OK -52
at+rsibt_setlocalattvalue=C,1,0
OK
```

Here RSSI is "-52", which is less than "threshold" So when you read in the Remote device on the "Alert Level" characteristic it will show the value as "0x00" means "No Alert".


COM4:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
OK
at+rsibt_getrssi=66-B2-27-5A-D5-4B?
OK -64
at+rsibt_setlocalattvalue=C,1,1
OK
at+rsibt_getrssi=66-B2-27-5A-D5-4B?
OK -62
at+rsibt_setlocalattvalue=C,1,1
OK
at+rsibt_getrssi=66-B2-27-5A-D5-4B?
OK -62
```

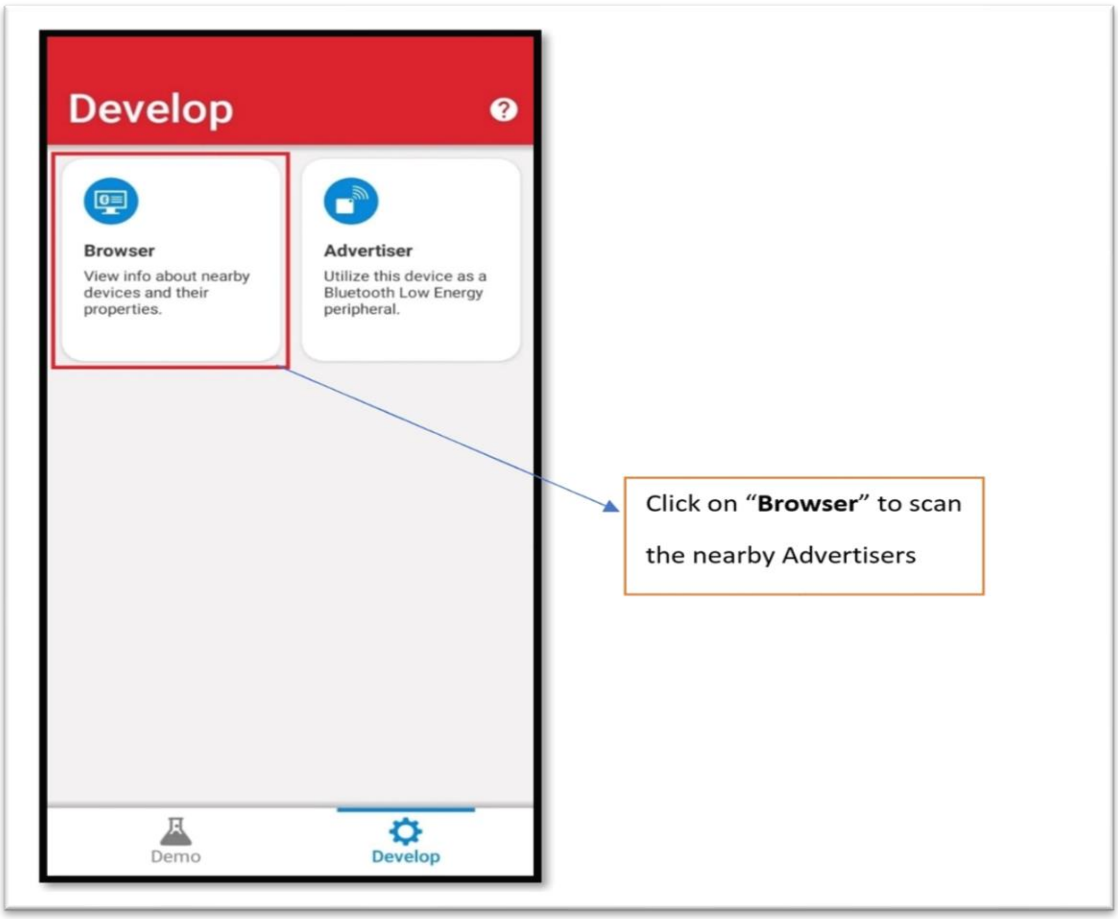
Here RSSI is "-62", which is more than "threshold" So when you read in the Remote device on the "Alert Level" characteristic it will show the value as "0x01" means "Mild Alert".

.

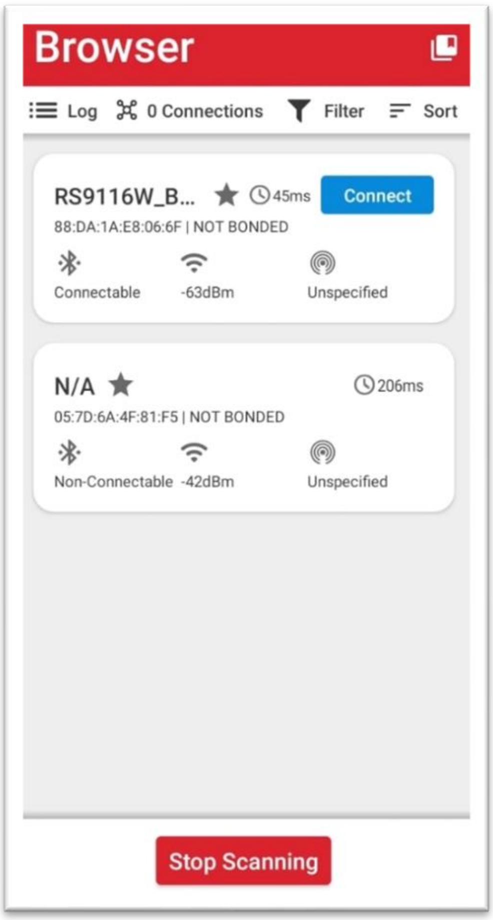
11 Connection Using the Remote Device

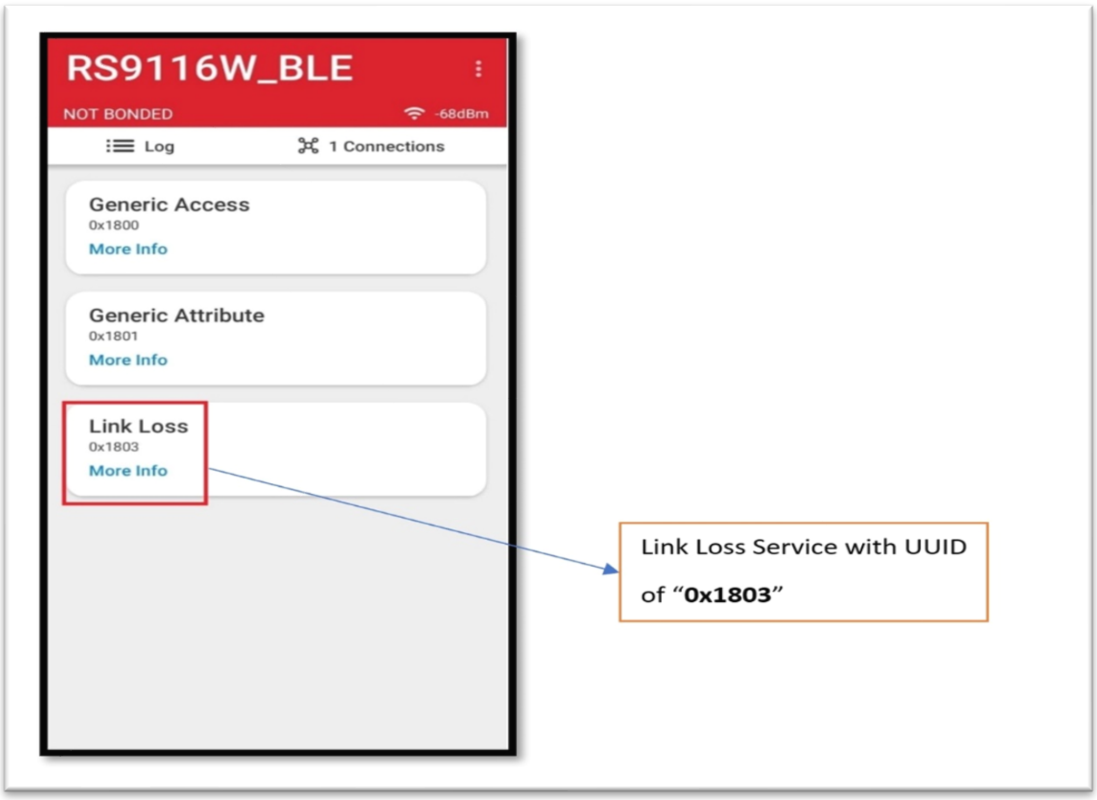
We can connect using the “EFR Connect” app in the mobile.

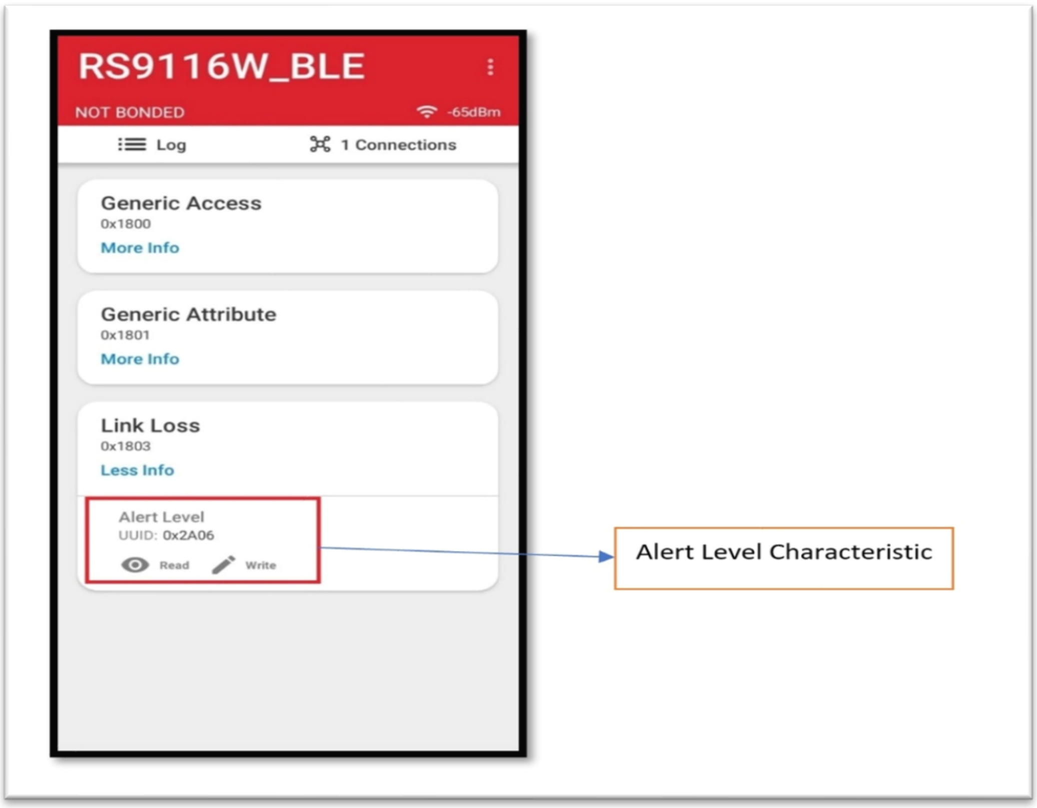
1. Open the “EFR Connect” app in the mobile (Remote Device) and scan for the devices. Below you can see the advertisers list.


Develop
Browser
View info about nearby devices and their properties.
Advertiser
Utilize this device as a Bluetooth Low Energy peripheral.
Demo
Develop
Click on “**Browser**” to scan the nearby Advertisers

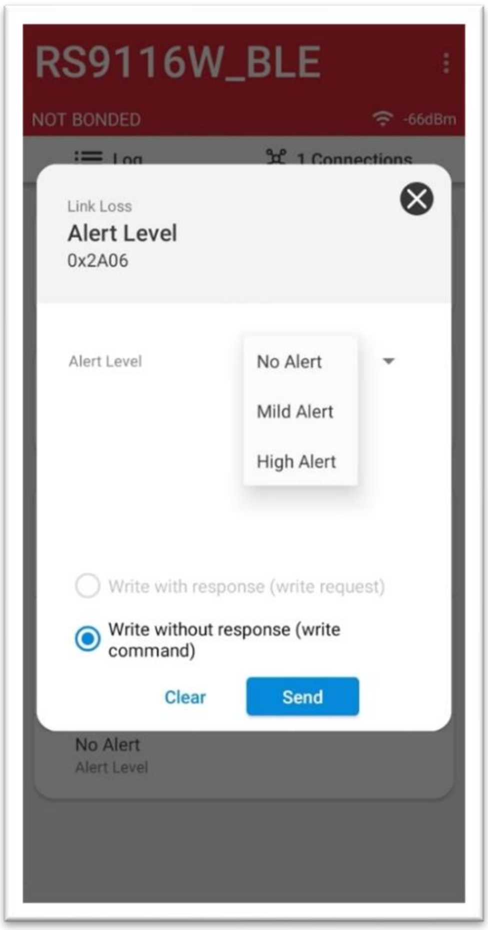
2. All the advertising devices list we can be able to see in the scanning list. Once we get the required Advertiser, we can directly give the connection initiation to the RS9116 and make sure connection should be successful.

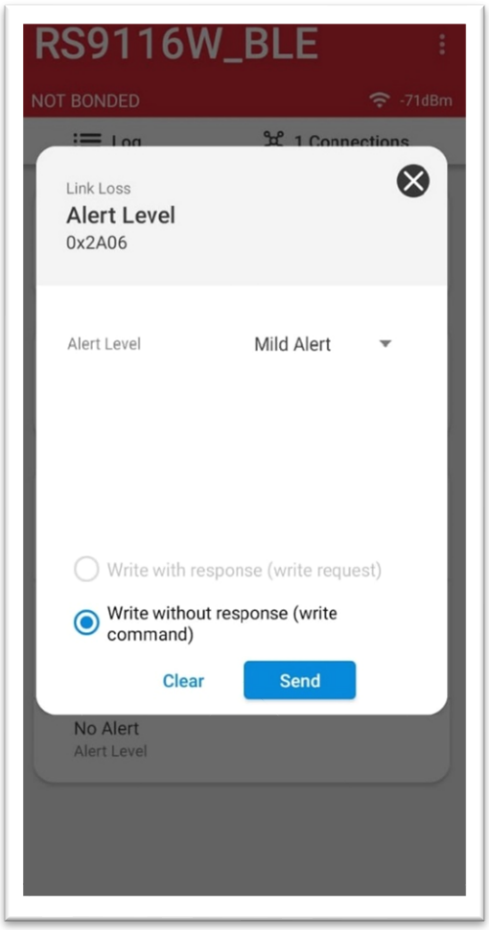

Browser
Log
0 Connections
Filter
Sort
RS9116W_B...
45ms
Connect
88:DA:1A:E8:06:6F | NOT BONDED
Connectable
-63dBm
Unspecified
N/A
206ms
05:7D:6A:4F:81:F5 | NOT BONDED
Non-Connectable
-42dBm
Unspecified
Stop Scanning


RS9116W_BLE
NOT BONDED
-68dBm
Log
1 Connections
Generic Access
0x1800
More Info
Generic Attribute
0x1801
More Info
Link Loss
0x1803
More Info
Link Loss Service with UUID of "0x1803"



3. After successful connection script will ask to select the "Alert Level" for the indication purpose. By selecting the Alert Level, it will pop-up on the tera term.





Link Loss
Alert Level
0x2A06

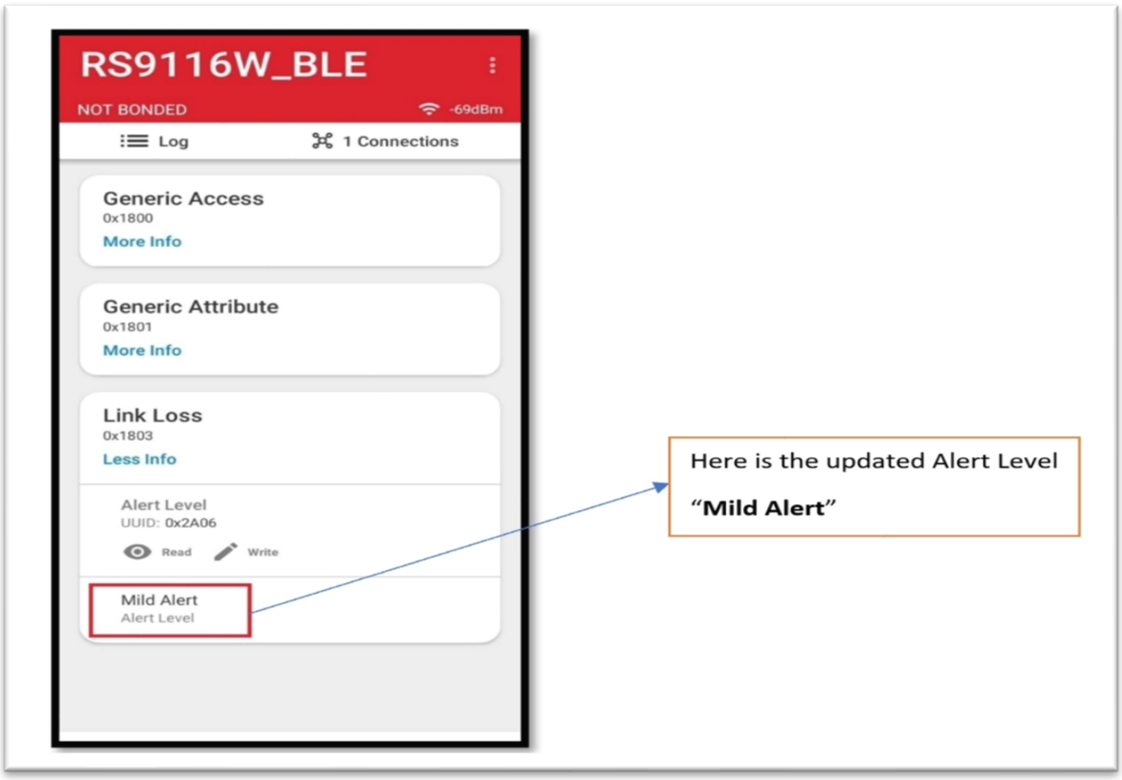
Alert Level Mild Alert

Write with response (write request)
Write without response (write command)

No Alert
Alert Level

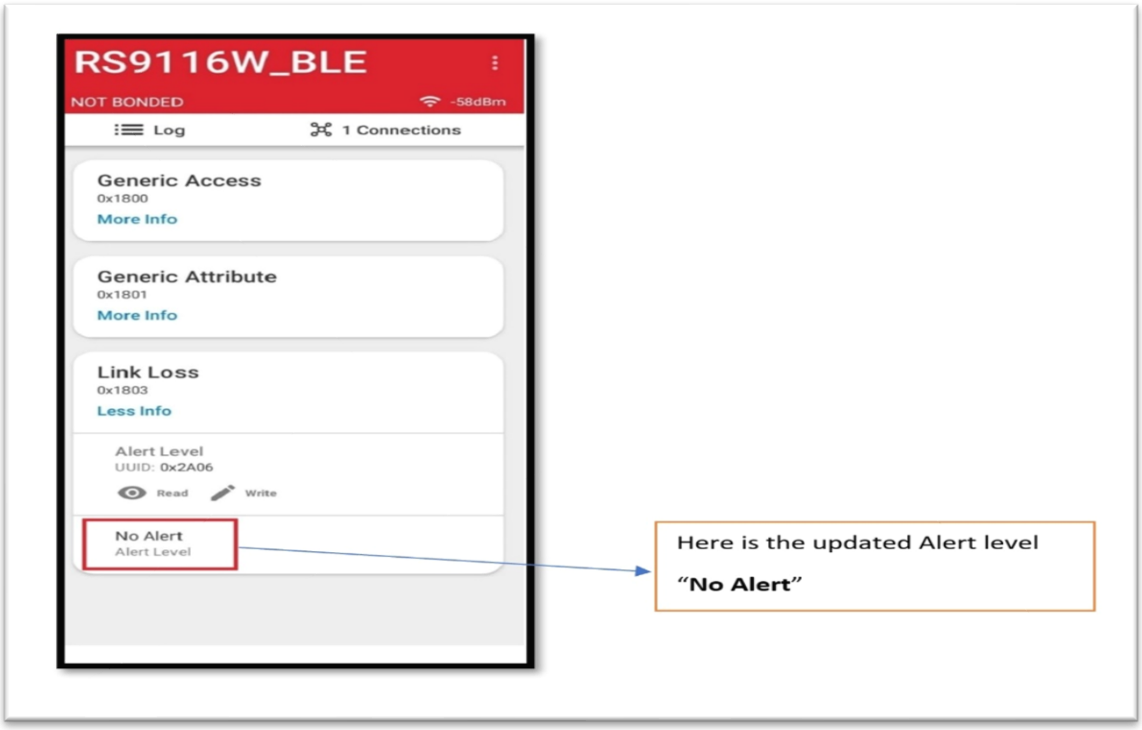
Note: - Threshold mentioned in the application script is -85dbm.

Here we selected the “Mild Alert”, So it is updated on the script and updated value on the same characteristic.


RS9116W_BLE
NOT BONDED -69dBm
Log 1 Connections
Generic Access
0x1800
More Info
Generic Attribute
0x1801
More Info
Link Loss
0x1803
Less Info
Alert Level
UUID: 0x2A06
Read Write
Mild Alert
Alert Level

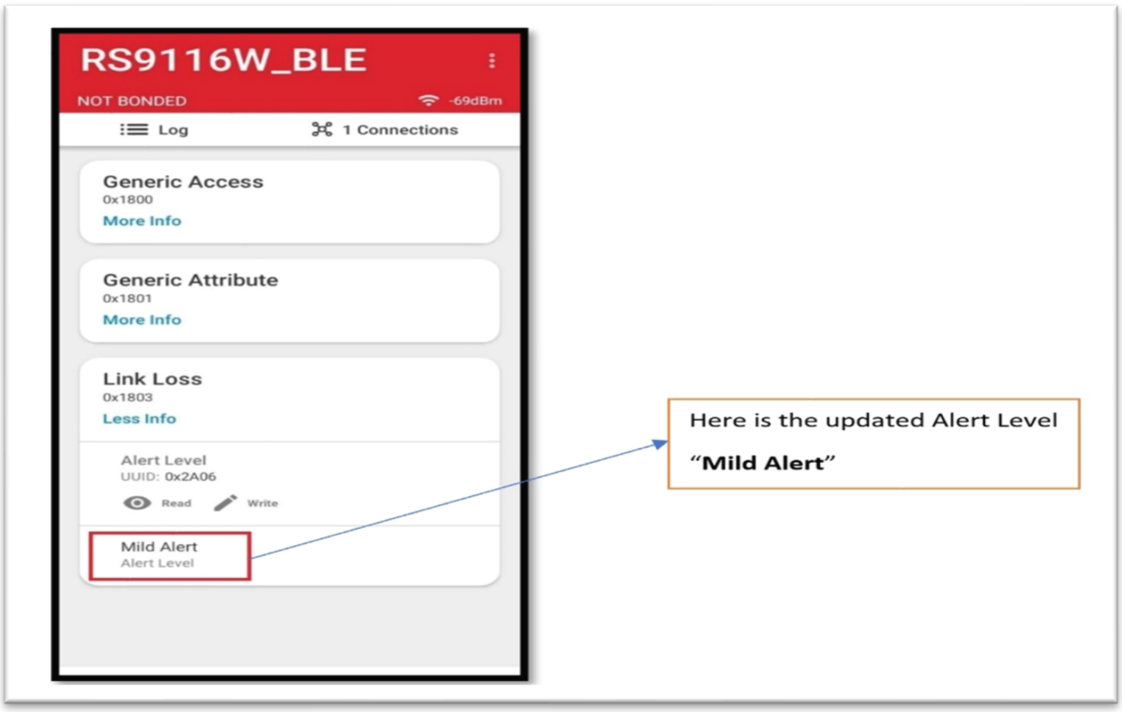
Here is the updated Alert Level
"Mild Alert"

4. After selection of the "Alert Level" from the "Remote device (Mobile)" script will run continuously, then whenever the RSSI does not cross the threshold level it will have "Alert Level" as "No Alert" we can see using the "Read" Property.


RS9116W_BLE
NOT BONDED -58dBm
Log 1 Connections
Generic Access
0x1800
More Info
Generic Attribute
0x1801
More Info
Link Loss
0x1803
Less Info
Alert Level
UUID: 0x2A06
Read Write
No Alert
Alert Level

Here is the updated Alert level
"No Alert"

5. Wherever the RSSI cross threshold level it will have “Alert Level” as “Mild Alert” we can see using the “Read” Property.


RS9116W_BLE
NOT BONDED -69dBm
Log 1 Connections
Generic Access
0x1800
More Info
Generic Attribute
0x1801
More Info
Link Loss
0x1803
Less Info
Alert Level
UUID: 0x2A06
Read Write
Mild Alert
Alert Level

Here is the updated Alert Level
“Mild Alert”

12 Summary

By using the above procedure RS9116W module will work as a Proximity Reporter, whenever the remote device moves away from the "Proximity Reporter (RS9116W)" it will indicate the alarm with selected “Alert Level".

13 References and Related Documentation

- Please refer to “RS9116W BLE AT Command Programming Reference Manual.pdf “ at https://docs.silabs.com/rs9116 for more details on BLE commands.
- Please refer to the link for Bluetooth core specification https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?doc_id=421043
- Please refer the below path for the Release package.

  https://www.silabs.com/documents/login/software/RS9116.NB0.WC.GENR.OSI.X.X.X.zip

14 Troubleshooting

1. If the UART port is not detected by the PC, please reset the module, and try again.
2. Make sure the Remote device should be enabled with Bluetooth.

15 Revision History

Revision No	Version No	Date	Change
1	v1.0	August 2021	Initial Version

16 Appendix A

Here is the “Proximity_profile” script code.

```
DEVICE_NAME = 'RS9116W_BLE'

errorcode = 'Unknown' ; used in error reports

message = 'This example demonstrates BLE Proximity Profile '
messagebox message ''

send '|'
wait 'U'
sendln 'U'
mpause 200
sendln '1'
wait 'Loading Done'
message = 'Firmware Loading Done Successfully'
messagebox message ''

sendln 'at+rsi_opermode=851968,0,1,2147483648,2150629376,3221225472,0,1075773440'
wait 'bt_loaded' 'ERROR'
if result == 2 then
    goto error
endif

strlen DEVICE_NAME
int2str device_name_length result
sendln 'at+rsibt_setlocalname='device_name_length','DEVICE_NAME
waitregex 'OK' 'ERROR (\w{4})'
if result == 2 then
    errorcode = groupmatchstr1
    goto error
endif

sendln 'at+rsibt_getlocalbdaddr?'
waitregex 'OK' 'ERROR (\w{4})'
if result == 2 then
    errorcode = groupmatchstr1
    goto error
endif
```

```
sendln 'at+rsibt_addservice=2,1803,3,30'
waitregex 'OK (.+),' 'ERROR (\w{4})'
if result == 1 then
    service_id = groupmatchstr1
else
    errorcode = groupmatchstr1
    goto error
endif

sendln 'at+rsibt_addattribute='service_id',B,2,2803,2,6,0,6,0,0C,00,06,2A'
waitregex 'OK' 'ERROR (\w{4})'
if result == 2 then
    errorcode = groupmatchstr1
    goto error
endif

sendln 'at+rsibt_addattribute='service_id',C,2,2A06,6,1,0,0'
waitregex 'OK' 'ERROR (\w{4})'
if result == 2 then
    errorcode = groupmatchstr1
    goto error
endif

sendln 'at+rsibt_setadvertisedata=10,2,1,6,C,9,52,53,39,31,31,36,57,5F,42,4C,45'
waitregex 'OK' 'ERROR (\w{4})'
if result == 2 then
    errorcode = groupmatchstr1
    goto error
endif

sendln 'at+rsibt_advertise=1,128,0,0,0,60,70,0,7'
waitregex 'OK' 'ERROR (\w{4})'
if result == 2 then
    errorcode = groupmatchstr1
    goto error
endif

waitln 'AT+RSIBT_LE_DEVICE_ENHANCE_CONNECTED'
if result == 1 then
```

```
    messagebox 'Remote device connected to module successfully:'  ''
    ;goto endscript
else
   goto error
endif

strsplit inputstr ','
Mac_addr = groupmatchstr2

messagebox 'User need to select the Alert Level to indicate:' ''
waitln 'AT+RSIBT_WRITE'
strsplit inputstr ','
str2int value groupmatchstr5
if value == 0 then
    messagebox 'No ALert selected' ''
elseif value == 1 then
    messagebox 'Mild Alert selected' ''
elseif value ==2 then
    messagebox 'High Alert selected' ''
endif

Threshould = -85
While 1
    sendln 'at+rsibt_getrssi='Mac_addr'?'
    waitregex 'OK (.+)' 'ERROR (\w{4})'
    if result == 1 then
        RSSI = groupmatchstr1
    else
        errorcode = groupmatchstr1
        goto error
    endif

    int2str str1 value
    str2int Item RSSI
    if Item > Threshould then
        sendln 'at+rsibt_setlocalattvalue=C,1,0'
        waitregex 'OK' 'ERROR (\w{4})'
        if result == 2 then
            errorcode = groupmatchstr1
            goto error
        endif
    elseif Item < Threshould then
        sendln 'at+rsibt_setlocalattvalue=C,1,'str1''
        waitregex 'OK' 'ERROR (\w{4})'
        if result == 2 then
            errorcode = groupmatchstr1
            goto error
        endif
    else
    endif
endwhile

goto endscript

:error
errormsg = 'Error '
strconcat errormsg errorcode
strconcat errormsg '. Please check your configuration and try again.'
messagebox errormsg 'ERROR'

:endscript

exit
```

17 Appendix B

This section explains the "Heart Rate Profile" with commands used and script code along with tools used for a connection. This also helps understanding how to use above explained commands to create any new profile.

17.1 Command Sequence

For any profile to be created except the *addservice* (respective service UUID) and *addattribute* (respective Characteristics) commands other commands are about the same as above used for the "Proximity Profile" example.

17.1.1 Service Requirements

The table below describes the service requirements.

Service	UUID	Heart Rate Sensor
Heart Rate Service	180D	Mandatory

Table 1: service requirements

The Heart Rate Sensor implements one and only one instance of Heart Rate Service.

```
at+rsibt_addservice=2,180D,3,30
OK 5B650, A
```

17.1.2 Add Attribute

The table below describes the structure and requirements for the Heart Rate Service.

Characteristic	UUID	Type	Support	Security	Prosperities
Heart Rate Measurement	2A37	8bit	Mandatory	none	Notify

Table 2: Heart Rate Measurement characteristic structure

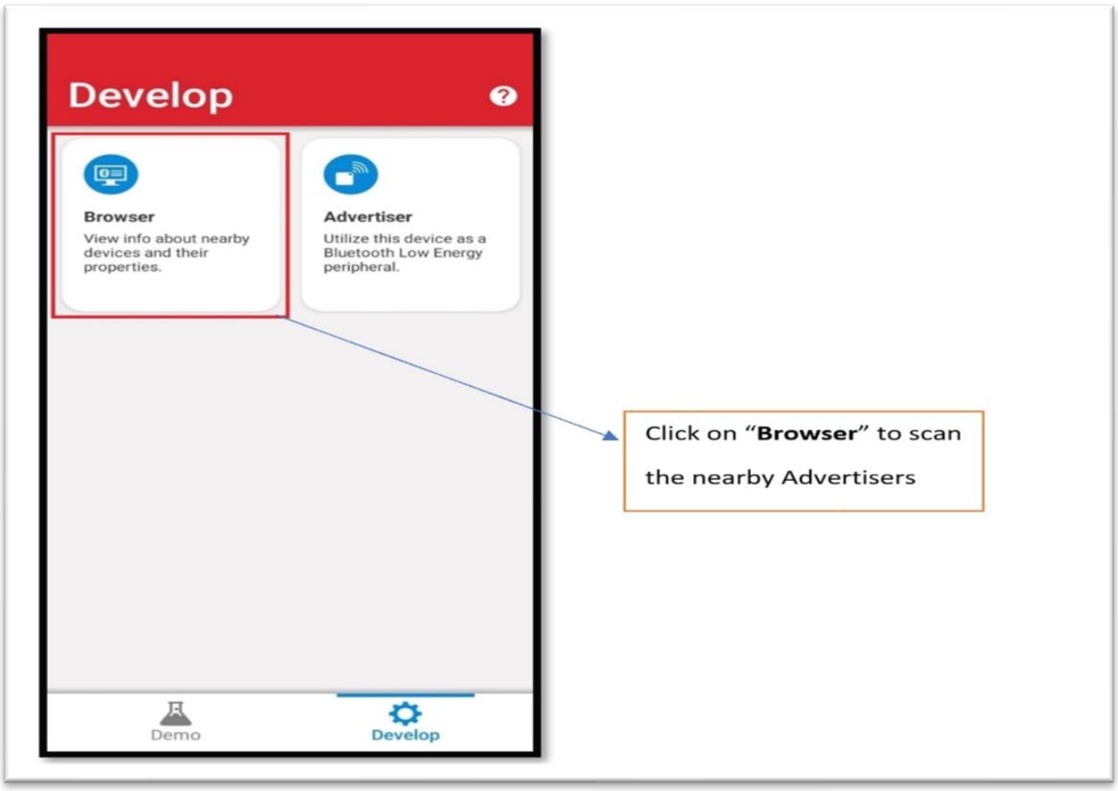
```
at+rsibt_addattribute=5B650, B,2,2803,10,6,0,10,0,0C,00,37,2A
```

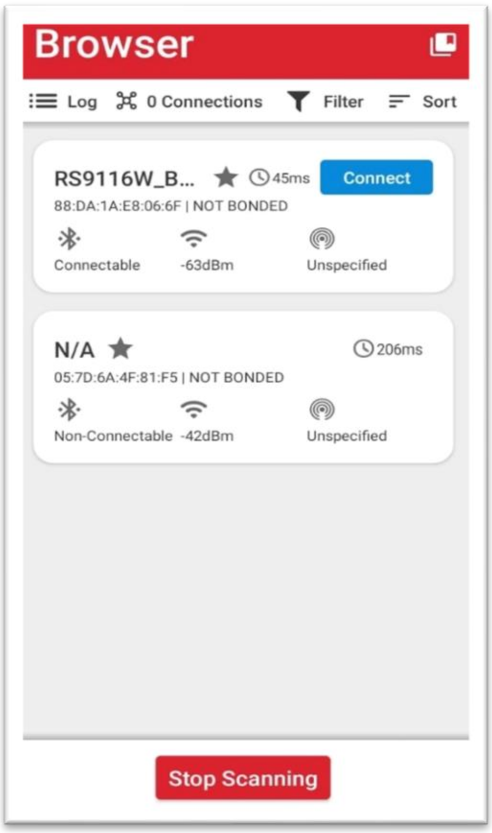
```
OK\r\n
```

17.2 Connection Using the Remote Device

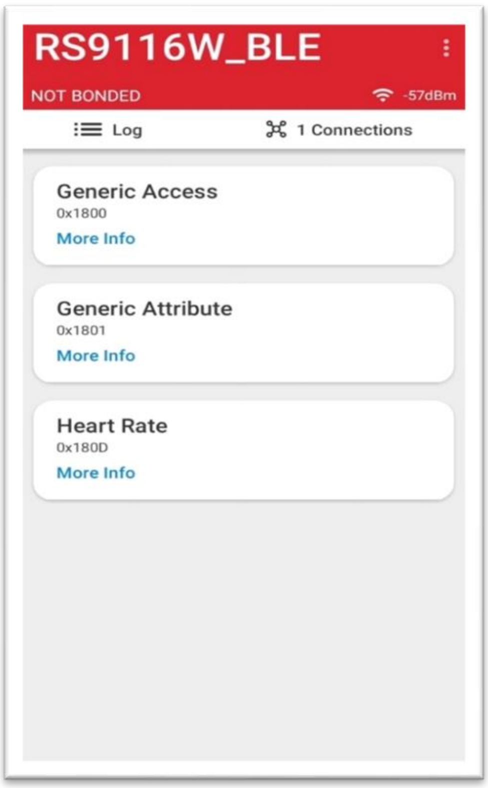
We can connect using the “EFR Connect” app on the mobile.

1. Open the “EFR Connect” app on the mobile (Remote Device) and scan for the devices. Below you can see the advertisers list.

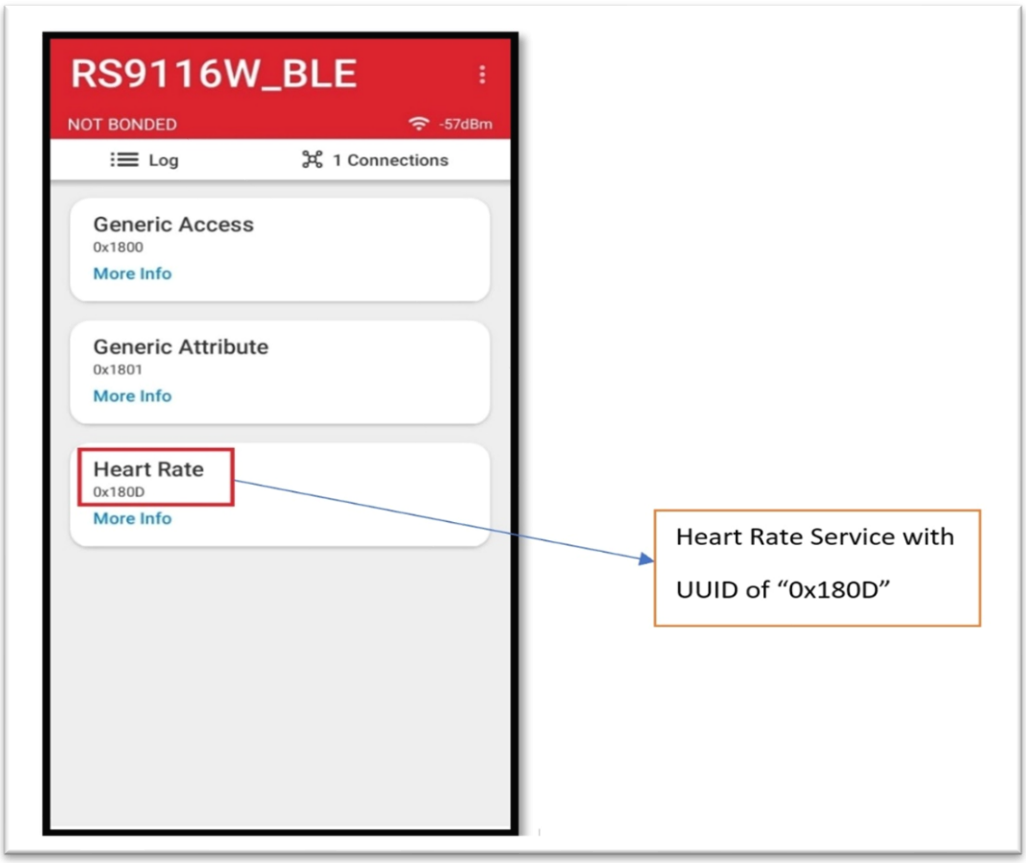

Develop
Browser
View info about nearby devices and their properties.
Advertiser
Utilize this device as a Bluetooth Low Energy peripheral.
Click on “**Browser**” to scan the nearby Advertisers
Demo
Develop

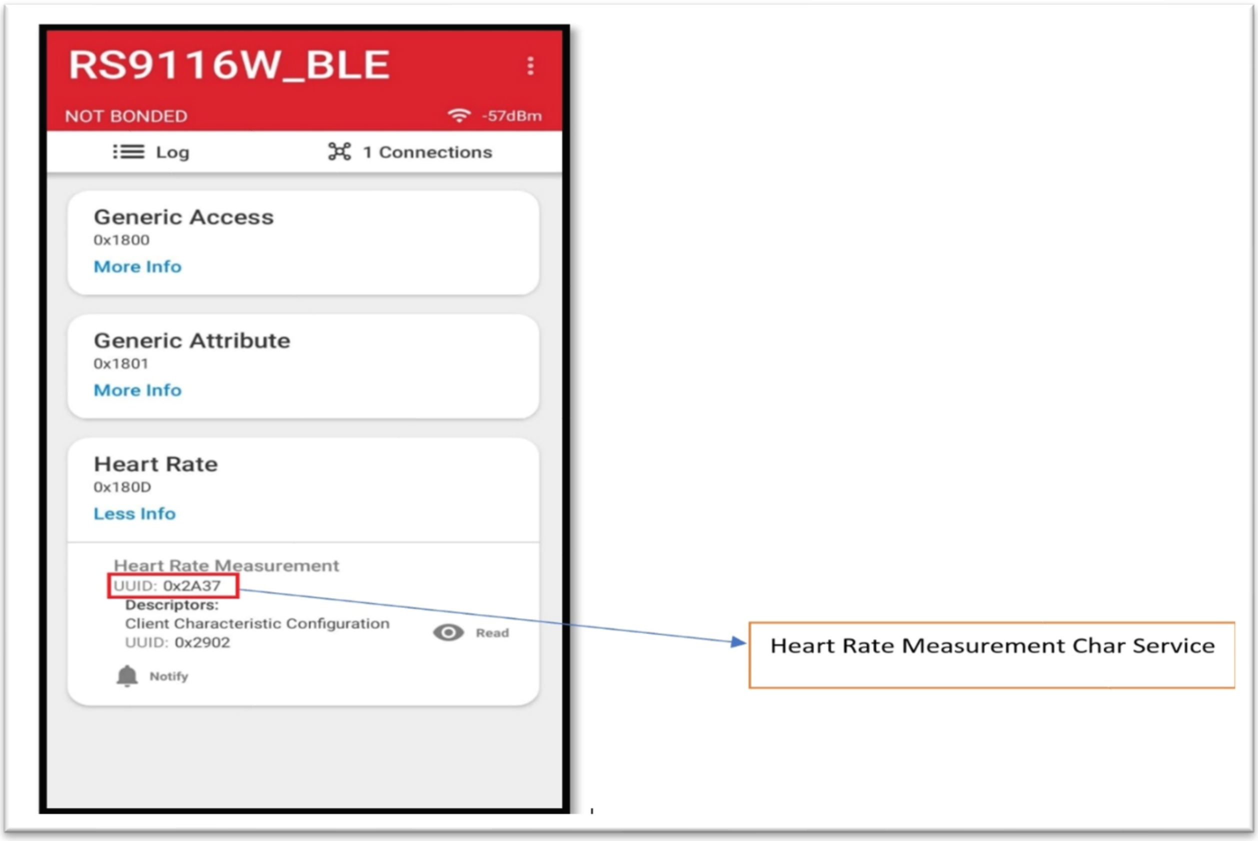


2. All the advertising devices list we can be able to see in the scanning list. Once we get the required Advertiser, we can directly give the connection initiation to the RS9116 and make sure the connection should be successful.

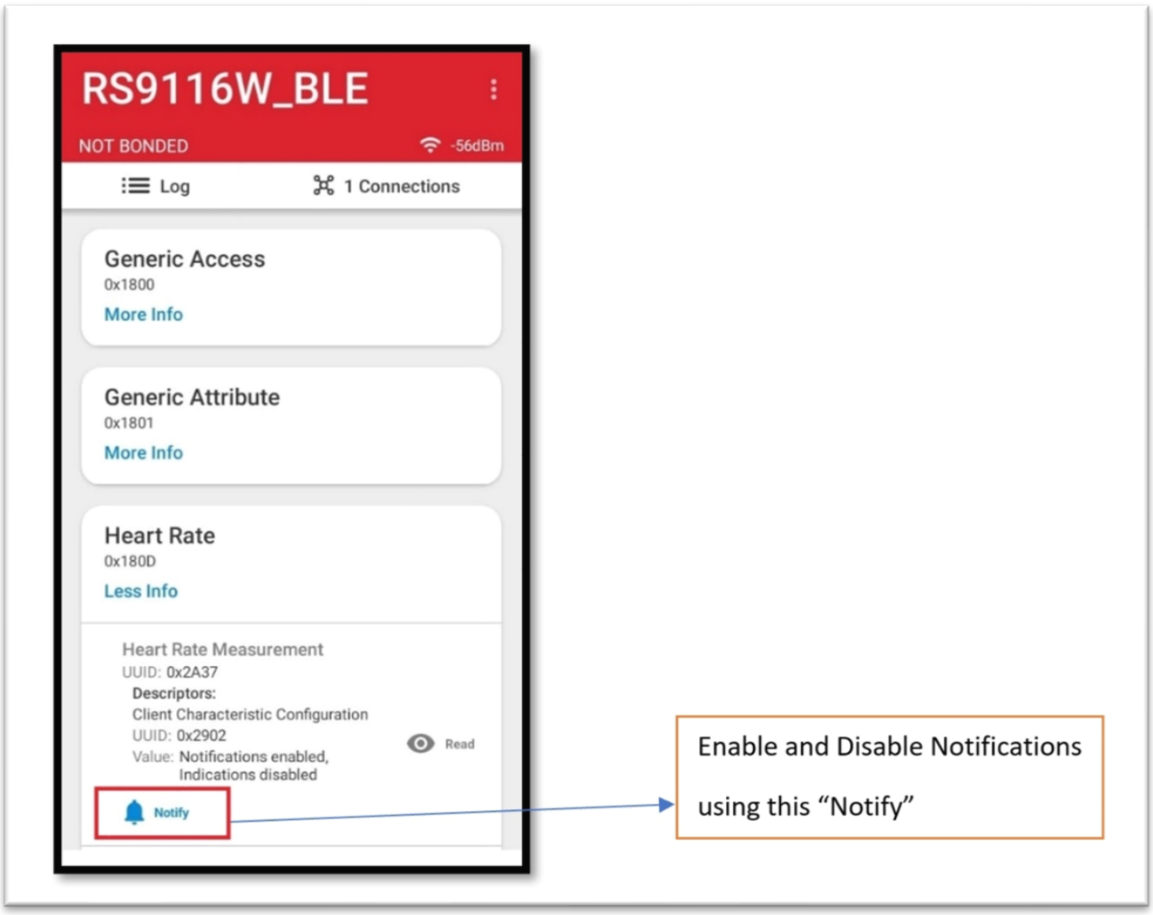


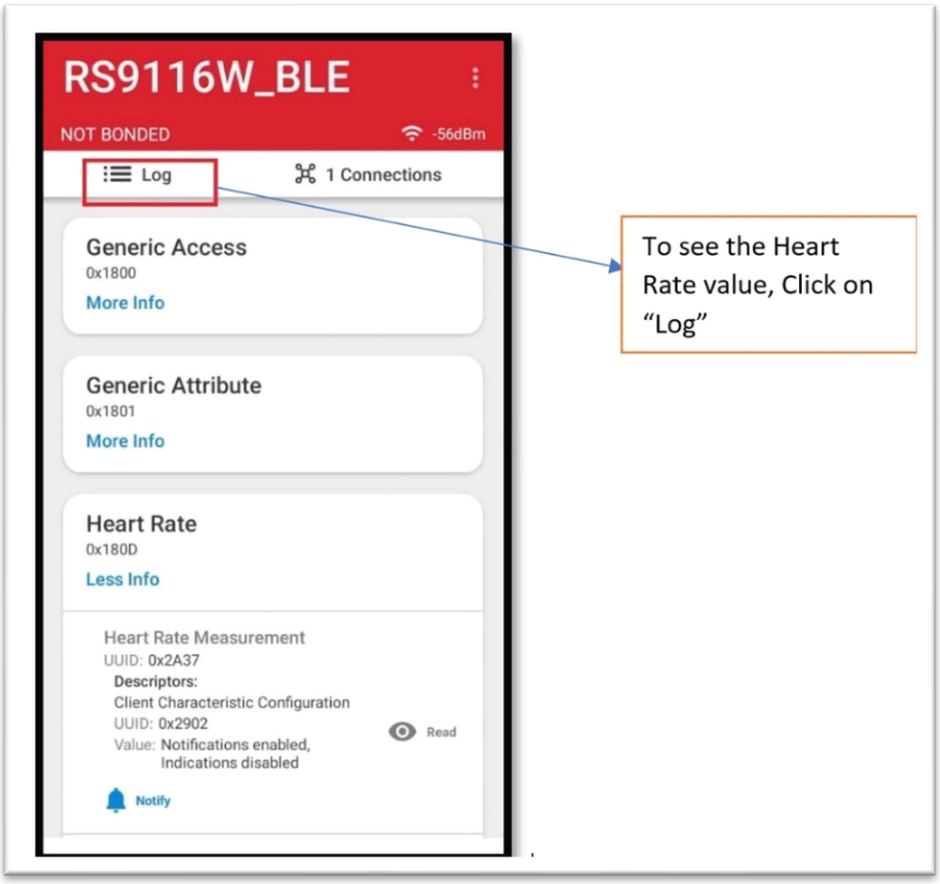
3. Now we can see the “RS9116” Heart rate monitor service “0x180d” in the service list. By clicking on that service, we can see the respective attribute and properties.

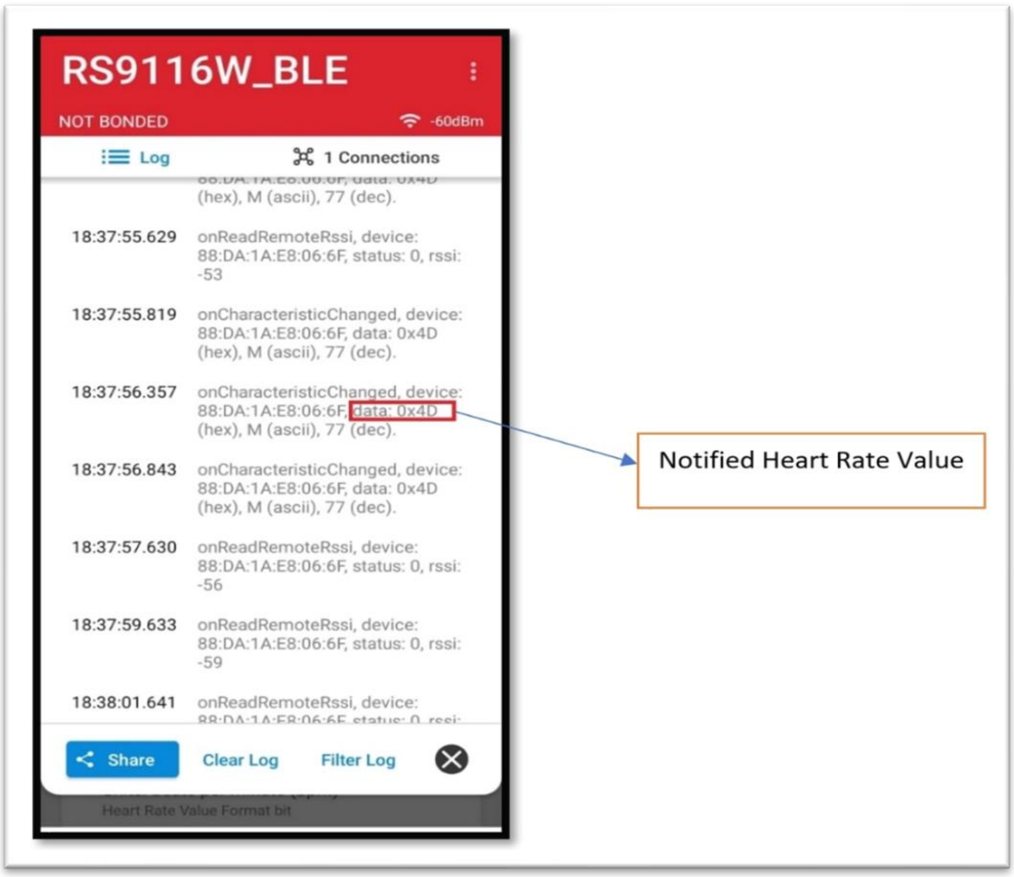




4. After connection, the "Tear-term" script will ask for the "Enable Notification", when you enable the notification in the Remote device, the RS9116 device will send the "Heart Rate" value.


RS9116W_BLE
NOT BONDED
-56dBm
Log
1 Connections
Generic Access
0x1800
More Info
Generic Attribute
0x1801
More Info
Heart Rate
0x180D
Less Info
Heart Rate Measurement
UUID: 0x2A37
Descriptors:
Client Characteristic Configuration
UUID: 0x2902
Value: Notifications enabled, Indications disabled
Read
Notify
Enable and Disable Notifications using this "Notify"


RS9116W_BLE
NOT BONDED
-56dBm
Log
1 Connections
Generic Access
0x1800
More Info
Generic Attribute
0x1801
More Info
Heart Rate
0x180D
Less Info
Heart Rate Measurement
UUID: 0x2A37
Descriptors:
Client Characteristic Configuration
UUID: 0x2902
Value: Notifications enabled, Indications disabled
Read
Notify
To see the Heart Rate value, Click on "Log"



```
(hex), M (ascii), 77 (dec).

18:37:55.629   onReadRemoteRssi, device:
               88:DA:1A:E8:06:6F, status: 0, rssi:
               -53

18:37:55.819   onCharacteristicChanged, device:
               88:DA:1A:E8:06:6F, data: 0x4D
               (hex), M (ascii), 77 (dec).

18:37:56.357   onCharacteristicChanged, device:
               88:DA:1A:E8:06:6F, data: 0x4D
               (hex), M (ascii), 77 (dec).

18:37:56.843   onCharacteristicChanged, device:
               88:DA:1A:E8:06:6F, data: 0x4D
               (hex), M (ascii), 77 (dec).

18:37:57.630   onReadRemoteRssi, device:
               88:DA:1A:E8:06:6F, status: 0, rssi:
               -56

18:37:59.633   onReadRemoteRssi, device:
               88:DA:1A:E8:06:6F, status: 0, rssi:
               -59

18:38:01.641   onReadRemoteRssi, device:
```

5. After 10 Notifications received from the RS9116, Script will ask for the "Disable notification", then once disabled notification again script will ask for the "Enabled Notification". This process repeat will continuously.

18 Appendix C

Here is the “Heart Rate Profile” script code.

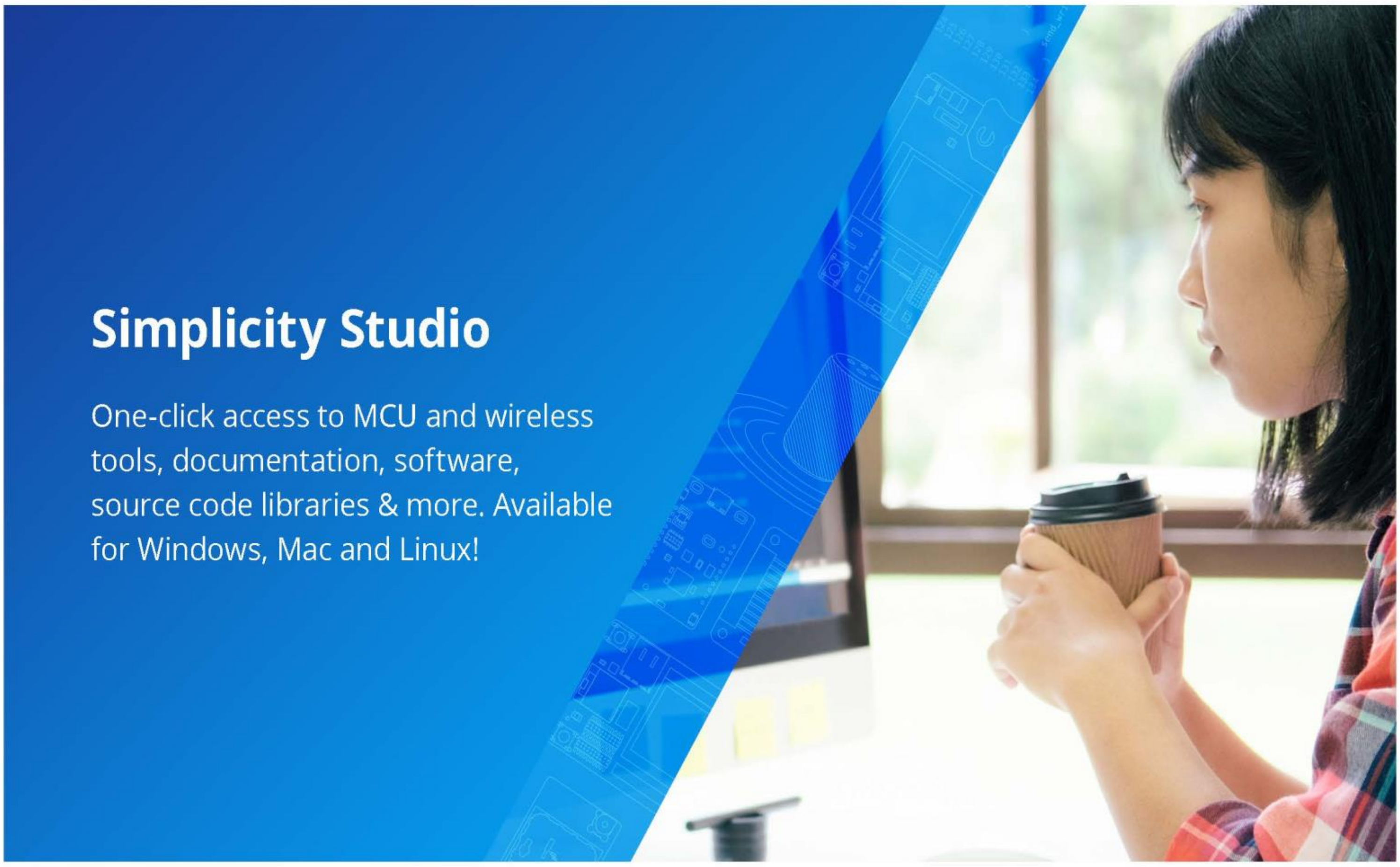
```
DEVICE_NAME = 'RS9116W_BLE'

errorcode = 'Unknown' ; used in error reports

message = 'This example demonstrates BLE_Heart_Rate profile functionality'
messagebox message ''

send '|'
wait 'U'
sendln 'U'
mpause 200
sendln '1'
wait 'Loading Done'
message = 'Firmware Loading Done Successfully'
messagebox message ''



sendln 'at+rsi_opermode=851968,0,1,2147483648,2150629376,3221225472,0,1075773440'
wait 'bt_loaded' 'ERROR'
if result == 2 then
goto error
endif

strlen DEVICE_NAME
int2str device_name_length result
sendln 'at+rsibt_setlocalname='device_name_length','DEVICE_NAME
waitregex 'OK' 'ERROR (\w{4})'
if result == 2 then
errorcode = groupmatchstr1
goto error
endif

sendln 'at+rsibt_getlocalbdaddr?'
waitregex 'OK' 'ERROR (\w{4})'
if result == 2 then
errorcode = groupmatchstr1
goto error
endif

sendln 'at+rsibt_addservice=2,180D,3,30'
waitregex 'OK (.+),' 'ERROR (\w{4})'
if result == 1 then
service_id = groupmatchstr1
else
errorcode = groupmatchstr1
goto error
endif

sendln 'at+rsibt_addattribute='service_id',B,2,2803,10,6,0,10,0,0C,00,37,2A'
waitregex 'OK' 'ERROR (\w{4})'
if result == 2 then
errorcode = groupmatchstr1
goto error
endif

sendln 'at+rsibt_addattribute='service_id',C,2,2A37,1A,3,0,0,0,0'
waitregex 'OK' 'ERROR (\w{4})'
if result == 2 then
```

```
errorcode = groupmatchstr1
goto error
endif


sendln 'at+rsibt_addattribute='service_id',D,2,2902,A,2,0,0,0'
waitregex 'OK' 'ERROR (\w{4})'
if result == 2 then
errorcode = groupmatchstr1
goto error
endif

sendln 'at+rsibt_setadvertisedata=10,2,1,6,C,9,52,53,39,31,31,36,57,5F,42,4C,45'
waitregex 'OK' 'ERROR (\w{4})'
if result == 2 then
errorcode = groupmatchstr1
goto error
endif

sendln 'at+rsi_fwversion?'
waitregex 'OK' 'ERROR (\w{4})'
if result == 2 then
errorcode = groupmatchstr1
goto error
endif

sendln 'at+rsibt_advertise=1,128,0,0,0,60,70,0,7'
waitregex 'OK' 'ERROR (\w{4})'
if result == 2 then
errorcode = groupmatchstr1
goto error
endif

waitln 'AT+RSIBT_LE_DEVICE_ENHANCE_CONNECTED'
if result == 1 then
messagebox 'Remote device connected to module successfully:' ''
;goto endscript
else
goto error
endif


while 1
mpause 500
:Enable
messagebox 'Waiting to enable the notifications' ''
waitln 'AT+RSIBT_WRITE' 'ERROR'
if result == 2 then
goto error
endif
i = 0
strsplit inputstr ','
strcompare groupmatchstr5 '1'
if result=0 then
while (i < 10)
sendln 'at+rsibt_setlocalattvalue=C,1,4D'
waitregex 'OK' 'ERROR (\w{4})'
if result == 2 then
errorcode = groupmatchstr1
goto error
endif
i = i+1
mpause 500
```

```
endwhile
endif
messagebox 'Waiting to disable the notifications' ''
waitln 'AT+RSIBT_WRITE' 'ERROR'
if result == 2 then
goto error
endif
strsplit inputstr ','
;messagebox groupmatchstr5 "groupmatchstr5"
strcompare groupmatchstr5 '0'
if result=0 then
goto Enable
endif
endwhile

goto endscript

mpause 1000

:error
errormsg = 'Error '
strconcat errormsg errorcode
strconcat errormsg '. Please check your configuration and try again.'
messagebox errormsg 'ERROR'




:endscript

exit
```

Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!



IoT Portfolio	**SW/HW**	**Quality**	**Support & Community**
www.silabs.com/IoT	www.silabs.com/simplicity	www.silabs.com/quality	www.silabs.com/community

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

www.silabs.com