

BSA GATT Guide

AMPAK CONFIDENTIAL

Revision History

Date	Version	Description	Author
2016/2/16	0.1	Initial version	Luke Chen
2016/2/17	0.2	Add how to register notification Add how to send specific ADV data	Luke Chen
2016/3/11	0.3	Fixed typo	Luke Chen

Index

Revision History	ii
Index	iii
GATT server.....	1
Create GATT server	1
Send Notification to GATT client	5
GATT Client.....	6
Connect to GATT server	6
Read/Write information.....	7
Register notification to receive notify from GATT server	8
Wake on LE.....	8
Use BSA to broadcast ADV with specific pattern to test this feature	9

GATT server

Create GATT server

Below is an example to use app_ble and app_manager to create GATT server with 1 service and characteristic.

1. Register server app. We use UUID x9999 as an example in this case.

```
**** Bluetooth Low Energy Server menu ****
29 => Register server app
30 => Deregister server app
31 => Connect to client
32 => Close connection
33 => Create service
34 => Add character
35 => Start service
36 => Configure BLE Advertisement data
37 => Display Servers
38 => Send indication
39 => HIKE WOLE
99 => QUIT
Select action => 29
Bluetooth BLE register menu:
Register app UUID(eg. x9999) => x9999
BSA_trace 11@ 01/10 03h:10m:22s:204ms: BSA_BleSeAppRegisterInit
BSA_trace 12@ 01/10 03h:10m:22s:204ms: BSA_BleSeAppRegister
enabled:1, server_if:5
```

2. Create Service

Select the server index which is created at step 1 and create 16bit service UUID. Also, Enter number of handle you need to have and select primary service in this case.

```

33 => Create service
34 => Add character
35 => Start service
36 => Configure BLE Advertisement data
37 => Display Servers
38 => Send indication
39 => HIKI WOLE
99 => QUIT
Select action => 33
Select Server:
***** BLE SERVER LIST *****
0:BLE Server server_if:5
***** BLE SERVER LIST END *****
Select => 0
Enter Service UUID to create(eg. x180A) => x180A
Enter num of handle(x2) to create.
    For Example, if you will add 5 characteristics in this service
    Use 12 (1 service x 2 + 5 characteristics x 2)
Enter num of handle => 4
Select primary(1) or not(0) => 1
BSA_trace 13@ 01/10 03h:23m:01s:750ms: BSA_BleSeCreateServiceInit
BSA_trace 14@ 01/10 03h:23m:01s:750ms: BSA_BleSeCreateService

```

3. Add characteristic

The characteristic UUID in this case is x9999 and it's NOT a descriptor.

```

33 => Create service
34 => Add character
35 => Start service
36 => Configure BLE Advertisement data
37 => Display Servers
38 => Send indication
39 => HIKI WOLE
99 => QUIT
Select action => DEBUG: app_ble_server_profile_cback: app_ble_server_profile_cback event = 11
BSA_BLE_SE_CREATE_EVT server_if:5 status:0 service_id:40
BSA_BLE_SE_CREATE_EVT if_num:0, attr_num:0
34
Select Server:
***** BLE SERVER LIST *****
0:BLE Server server_if:5
    attr_num:0:uuid:0x180a, is_pri:1, service_id:40 attr_id:0
***** BLE SERVER LIST END *****
Select => 0
Select service's attribute number
Select => 0
Add char UUID(eg. x9999) => x9999
BSA_trace 15@ 01/02 04h:54m:04s:381ms: BSA_BleSeAddCharInit
Select descriptor? (yes=1 or no=0) => 0
Enter Attribute Permissions[Eg: Read-0x1, Write-0x10, Read|Write-0x11]
Enter => 0x11
Enter Characteristic Properties Eg: WRITE-0x08, READ-0x02, Notify-0x10, Indicate-0x20
Eg: For READ|WRITE|NOTIFY|INDICATE enter 0x3A
Enter => 0x3A
BSA_trace 16@ 01/02 04h:54m:08s:699ms: BSA_BleSeAddChar

```

As we set characteristic properties to support Notify and Indicate, so we need to create x2902

descriptor (Client Characteristic Configuration descriptor) as below

```
34 => Add character
35 => Start service
36 => Configure BLE Advertisement data
37 => Display Servers
38 => Send indication
39 => HIKI WOLE
99 => QUIT
Select action => 34
Select Server:
***** BLE SERVER LIST *****
0:BLE Server server_if:5
    attr_num:0:uuid:0x180a, is_pri:1, service_id:40 attr_id:0
    attr_num:1:uuid:0x9999, is_pri:0, service_id:40 attr_id:42
***** BLE SERVER LIST END *****
Select => 0
Select service's attribute number
Select => 0
Add char UUID(eg. x9999) => x2902
BSA_trace 17@ 01/02 10h:31m:12s:247ms: BSA_BleSeAddCharInit
Select descriptor? (yes=1 or no=0) => 1
Enter Attribute Permissions[Eg: Read-0x1, Write-0x10, Read|Write-0x11]
Enter => 0x11
BSA_trace 18@ 01/02 10h:31m:16s:178ms: BSA_BleSeAddChar
```

4. Start service

Select service index and service attribute index to start.

```
35 => Start service
36 => Configure BLE Advertisement data
37 => Display Servers
38 => Send indication
39 => HIKI WOLE
99 => QUIT
Select action => 35
Select Server:
***** BLE SERVER LIST *****
0:BLE Server server_if:5
    attr_num:0:uuid:0x180a, is_pri:1, service_id:40 attr_id:0
    attr_num:1:uuid:0x9999, is_pri:0, service_id:40 attr_id:42
    attr_num:2:uuid:0x2902, is_pri:0, service_id:40 attr_id:43
***** BLE SERVER LIST END *****
Select => 0
Select Service's attribute number :
Select => 0
BSA_trace 19@ 01/02 10h:32m:45s:081ms: BSA_BleSeStartServiceInit
service_id:40, num:0
BSA_trace 20@ 01/02 10h:32m:45s:081ms: BSA_BleSeStartService
```

5. You can check the service and characteristic you just created

```

36 => Configure BLE Advertisement data
37 => Display Servers
38 => Send indication
39 => HIKE WOLE
99 => QUIT
Select action => 37
***** BLE SERVER LIST *****
0:BLE Server server_if:5
    attr_num:0:uuid:0x180a, is_pri:1, service_id:40 attr_id:0
    attr_num:1:uuid:0x9999, is_pri:0, service_id:40 attr_id:42
    attr_num:2:uuid:0x2902, is_pri:0, service_id:40 attr_id:43
***** BLE SERVER LIST END *****

```

6. Configure BLE advertisement data as below

```

36 => Configure BLE Advertisement data
37 => Display Servers
38 => Send indication
39 => HIKE WOLE
99 => QUIT
Select action => 36
Enter appearance value Eg:0x1122 => 0x1122
Enter number of services between <1-6> Eg:2 => 1
Enter service UUID eg:0xA108 => x180A
Is this scan response? (0:FALSE, 1:TRUE) => 0
BSA_trace 19@ 01/10 03h:27m:06s:537ms: BSA_DmSetConfigInit
BSA_trace 20@ 01/10 03h:27m:06s:537ms: BSA_DmSetConfig

```

7. Set BLE visibility to be discoverable and connectable

```

19 => Set device BLE visibility
20 => Set AFH Configuration
21 => Set Tx Power Class2 (specific FW needed)
22 => Set Tx Power Class1.5 (specific FW needed)
23 => Change Dual Stack Mode (currently:DUAL_STACK_MODE_BSA)
24 => Set Link Policy
25 => Enter Passkey
26 => Get Remote Device Name
27 => RSSI Measurement
96 => Kill BSA server
97 => Connect to BSA server
98 => Disconnect from BSA server
99 => Quit
Select action => 19
Set BLE Discoverability (0=Not Discoverable, 1=Discoverable) => 1
    Device will be discoverable
Set BLE Connectability (0=Not Connectable, 1=Connectable) => 1
    Device will be connectable
DEBUG: app_dm_set_ble_visibility: Set BLE Visibility Discoverable:1 Connectable:1
BSA_trace 26@ 01/10 03h:31m:08s:351ms: BSA_DmSetConfigInit
BSA_trace 27@ 01/10 03h:31m:08s:351ms: BSA_DmSetConfig

```

8. Set BR/EDR to be non-discoverable and non-connectable

```

17 => Enter remote Out Of Band data
18 => Set device visibility
19 => Set device BLE visibility
20 => Set AFH Configuration
21 => Set Tx Power Class2 (specific FW needed)
22 => Set Tx Power Class1.5 (specific FW needed)
23 => Change Dual Stack Mode (currently:DUAL_STACK_MODE_BSA)
24 => Set Link Policy
25 => Enter Passkey
26 => Get Remote Device Name
27 => RSSI Measurement
96 => Kill BSA server
97 => Connect to BSA server
98 => Disconnect from BSA server
99 => Quit
Select action => 18
Set BR/EDR Discoverability (0=Not Discoverable, 1=Discoverable) => 0
Device will not be discoverable
Set BR/EDR Connectability (0=Not Connectable, 1=Connectable) => 0
Device will not be connectable
DEBUG: app_dm_set_visibility: Set Visibility Discoverable:0 Connectable:0
BSA_trace 29@ 01/06 03h:37m:21s:127ms: BSA_DmSetConfigInit

```

9. You should be able to connect to this GATT server now. You can use *nRF Master Control* application in Android and use *lightblue* in iOS to connect to GATT server.

Send Notification to GATT client

After you've created GATT server, you can use notification to notify GATT client. Now we use iOS *lightblue* as an example to receive notification from BSA GATT server.

1. Launch iOS *lightblue* to scan nearby peripheral devices and connect to BSA GATT server.
2. Click "Listen for notification"
3. At BSA GATT server side, select "send indication" to send data.


```

38 => Send indication
39 => HIKI WOLE
99 => QUIT
Select action => 38
Select Server:
***** BLE SERVER LIST *****
0:BLE Server server_if:5
    attr_num:0:uuid:0x180a, is_pri:1, service_id:40 attr_id:0
    attr_num:1:uuid:0x9999, is_pri:0, service_id:40 attr_id:42
    attr_num:2:uuid:0x2902, is_pri:0, service_id:40 attr_id:43
***** BLE SERVER LIST END *****
Select => 0
Select Service's attribute number :
Select => 1
BSA_trace 21@ 01/02 10h:48m:16s:058ms: BSA_BleSeSendIndInit
Enter length of data => 1
Enter data in byte => 0xAA

```

GATT Client

Connect to GATT server

BSA can also acts as GATT client

1. Select “Start BLE discovery” to scan nearby devices.
2. Select “Connect to server” to make connection to GATT server

```

Select action => 10
Bluetooth BLE connect menu:
  0 Device from XML database (already paired)
  1 Device found in last discovery
Select source => 1
Dev:0
  Bdaddr:79:ab:54:46:12:8f
  Name:Blank
  ClassOfDevice:00:00:00 => Misc device
  Rssi:-43
Select device => 0
Select Client:
BLE CLIENT LIST
Index:0 client_if:4

Select => 0
BSA_trace 19@ 01/02 11h:08m:21s:791ms: BSA_BleClConnectInit
Direct connection:1, Background connection:0 => 1
BSA_trace 20@ 01/02 11h:08m:23s:425ms: BSA_BleClConnect

```

3. You can use “Display Clients” to know the service and characteristic UUID

Read/Write information

After connected to GATT server, you can read/write information from/to GATT server. Below is the sample to use “Read” information to read x180A service UUID and x9999 characteristic.

```

Select action => 14
Select Client:
BLE CLIENT LIST
Index:0 client_if:4

Select => 0
Enter Service UUID to read(eg. x1800) => x180A
Enter Is_primary value(eg:0,1) => 1
Enter Instance ID for Service UUID(eg. 0,1,2..) => 0
Enter Char UUID to read(eg. x2a00) => x9999
Enter Instance ID for Char UUID(eg. 0,1,2..) => 0
BSA_trace 27@ 01/02 11h:50m:47s:226ms: BSA_BleClReadInit
Select descriptor? (yes=1 or no=0) => 0
BSA_trace 28@ 01/02 11h:50m:49s:133ms: BSA_BleClRead

```

Register notification to receive notify from GATT server

You can also register notification receive notify from GATT server.

Select “Register notification” in app_ble and enter corresponding service UUID and characteristic UUID. In this example, we can enter service UUID x180A and characteristic UUID x9999 that we created in server side.

```
Select action => 16
Select Client:
BLE CLIENT LIST
Index:0 client_if:4

Select => 0
BSA_trace 31@ 01/03 03h:29m:38s:843ms: BSA_BleClNotifRegisterInit
Enter Service UUID to register(eg. x1800) => x180A
Enter Char UUID to register(eg. x2a00) => x9999
Enter Is_primary value(eg:0,1) => 1
Enter Instance ID for Service UUID(eg. 0,1,2..) => 0
Enter Instance ID for Char UUID(eg. 0,1,2..) => 0
DEBUG: app_ble_client_register_notification: size of ble_notireg_param:64
BSA_trace 32@ 01/03 03h:29m:52s:370ms: BSA_BleClNotifRegister
```

After notification is registered, GATT client side can receive notification/indication if GATT server notify/indicate client.

```
Select action => BSA_BLE_CL_NOTIF_EVT_BDA :BE:EF:BE:EF:14:A5
conn_id:0x4, svrc_id:0x180a, char_id:0x9999, descr_type:0x0000, len:2, is_notify:0
BSA_trace 43@ 01/03 03h:33m:46s:787ms: data:
BSA_trace 44@ 01/03 03h:33m:46s:787ms: 0000: aa bb
Receive Indication! send Indication Confirmation!
BSA_trace 45@ 01/03 03h:33m:46s:787ms: BSA_BleClIndConfInit
BSA_trace 46@ 01/03 03h:33m:46s:787ms: BSA_BleClIndConf
```

Wake on LE

In some user scenario, we may need to wake up host when BT receives some specific wake up

LE advertisement pattern. In this sample, BSA would toggle BT_HOST_WAKE when it receives ADV with manufacture specific data of AD type to “0F 00 57 41 4B 45 55 50”. The pattern is set in app_ble_wake_configure(). “0F 00” is company ID and “57 41 4B 45 55 50” is “WAKEUP” ASCII code. Check app_ble_wake_configure function to know more detail and user can set this wake up pattern accordingly in source code.

Below is the step to enable this feature

1. Add 2 parameters marked as red to start BSA server

```
#./bsa_server -diag=0 -lpm -d /dev/ttyXXX -p /xxx/xxx/xxx/xxx.hcd
```

(-diag=0 is used to prevent LMP message would assert BT_HOST_WAKE and -lpm is used to enable low power mode)

Please note that you should assert BT_WAKE to prevent BT controller get into low power mode to prevent BSA crash.

2. Select “Configure for Wake on BLE” in app_ble to enable WoLE feature.
3. Send BLE ADV with specific pattern from other devices and monitor BT_HOST_WAKE

Use BSA to broadcast ADV with specific pattern to test WoLE feature

We can use other BSA to send ADV for testing if you don't have device that can advertise packet.

