

Bluetooth® 5.0  
Dual-Mode Module

EYSGCCAXX / EYSGCCSXX  
(EBSGCCAXX / EBSGCCSXX)

HCI Users Guide

The Bluetooth® word mark and logos are owned by the Bluetooth SIG, Inc.  
And any use of such marks by TAIYO YUDEN CO., LTD. is under license.

## EYSGCCAXX / EYSGCCSXX

TAIYO YUDEN CO., LTD.

## Revision History

Version	Date	Description
0.1	2017/06/28	First Release
0.7	2020/09/03	Official version candidate Ubuntu Version: 16.04 Update log
1.0	2020/09/17	First official version

## Content

Content .....	3
1. Configuration of the equipment.....	4
2. Preparation.....	4
3. Test Scenario .....	5
3.1 Bluetooth® .....	5
3.1.1 Inquiry & Page Scan enabled .....	6
3.1.2 Check the status .....	6
3.1.3 Inquiry.....	7
3.1.4 RFCOMM communication.....	7
3.1.5 Change the communication parameters.....	8
3.1.6 Change the communication mode .....	8
3.1.7 Power Saving Mode .....	10
3.1.8 Disconnection.....	11
3.2 BLE .....	12
3.2.1 Check the status .....	12
3.2.2 Advertising.....	12
3.2.3 Scan .....	13
3.2.4 Connection .....	14
3.2.5 Disconnection.....	15

## EYSGCCAXX / EYSGCCSXX

TAIYO YUDEN CO., LTD.

## 1. Configuration of the equipment

One Ubuntu 16.04 LTS PC is used. Connect two evaluation boards to PC via USB cables.

USB I/F Type: EBSGCCSXX

UART I/F Type: EBSGCCAXX



Note: For convenience, two evaluation boards are used, but the PC equipped with Bluetooth® can be used as an opposite equipment.

## 2. Preparation

Check the version of BlueZ.

```
$ dpkg --status bluez | grep '^Version:'
Version: 5.37-0ubuntu5.1
```

Configure Super User access level rights.

```
$ sudo su
[sudo] password for user:
```

Note: BD addresses shown in this document are fictitious. Please match them with the module actually used.

## EBSGCCSXX

When connecting by USB, the evaluation board is recognized as HCI0.

Note: If PC equipped with Bluetooth® module be used, the Number on the HCI0 will be change to 1.

```
# hciconfig
hci0:    Type: BR/EDR  Bus: USB
        BD Address: 00:00:A4:17:04:18 ACL MTU: 310:10 SCO MTU: 64:8
        DOWN
        RX bytes:580 acl:0 sco:0 events:31 errors:0
        TX bytes:368 acl:0 sco:0 commands:30 errors:0
```

## EBSGCCAXX

When connecting with UART, the evaluation board is recognized as the serial device “/dev/ttyUSB0”.

Note: If other serial devices are connected to the PC, the Number on the ttyUSB0 will be change to 1 or 2.

## EYSGCCAXX / EYSGCCSXX

TAIYO YUDEN CO., LTD.

Please check the device as shown below.

```
# ls /dev/ttyU*
/dev/ttyUSB0
```

Recognize as an HCI device with the following command.

```
# hciattach /dev/ttyUSB0 any 115200
Device setup complete

# hciconfig
hci1:    Type: BR/EDR  Bus: UART
        BD Address: 00:00:A4:17:04:17 ACL MTU: 310:10 SCO MTU: 64:8
        DOWN
        RX bytes:604 acl:0 sco:0 events:30 errors:0
        TX bytes:398 acl:0 sco:0 commands:30 errors:0

hci0:    Type: BR/EDR  Bus: USB
        BD Address: 00:00:A4:17:04:18 ACL MTU: 310:10 SCO MTU: 64:8
        DOWN
        RX bytes:580 acl:0 sco:0 events:31 errors:0
        TX bytes:368 acl:0 sco:0 commands:30 errors:0
```

Because the device cannot be "UP" directly, please use the following command.

```
# rfkill unblock bluetooth
```

Device "UP".

```
# hciconfig hci0 up
# hciconfig hci1 up
```

Confirm the device by "hcidtool".

```
$ # hcidtool dev
Devices:
        hci1      00:00:A4:17:04:17
        hci0      00:00:A4:17:04:18
```

### 3. Test Scenario

#### 3.1 Bluetooth®

Check the operation log by executing "hcidump" command in another terminal window.

```
# hcidump -i hci1 hci
HCI sniffer - Bluetooth packet analyzer ver 5.37
device: hci1 snap_len: 1500 filter: 0x2
```

**Note: The hcidump log is shown in yellow.**

Note: In the above command, the HCI event of hci1 is acquired.

## 3.1.1 Inquiry &amp; Page Scan enabled

```
# hcitool -i hci0 cmd 0x3 0x1a 0x3
< HCI Command: ogf 0x03, ocf 0x001a, plen 1
03
> HCI Event: 0x0e plen 4
01 1A 0C 00
# hcitool -i hci1 cmd 0x3 0x1a 0x3
< HCI Command: ogf 0x03, ocf 0x001a, plen 1
03
> HCI Event: 0x0e plen 4
01 1A 0C 00

< HCI Command: Write Scan Enable (0x03|0x001a) plen 1
enable 3
> HCI Event: Command Complete (0x0e) plen 4
Write Scan Enable (0x03|0x001a) ncmd 1
status 0x00
```

Note: "hcitool -i hciX cmd" can issue each HCI command specified in Bluetooth® Specification.

"hcitool -i hci0 cmd 0x3 0x1a 0x3" can issue Write Scan Enable Command.

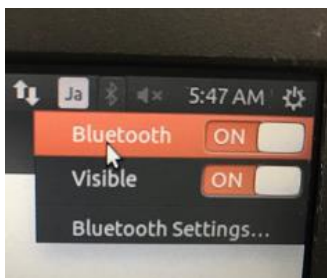
"0x3" specifies OGF. Write Scan Enable Command is HCI Control and Baseband Commands, so OGF=0x3.

"0x1a" specifies OCF. OCF of Write Scan Enable Command is "0x1a".

The last "0x3" is a parameter. Enable both Inquiry Scan and Page Scan.

If want to issue other HCI command, please refer to the Bluetooth® Specification.

Click on Bluetooth® settings at the top right of Desktop, turn on both "Bluetooth" and "Visible".



## 3.1.2 Check the status

```
# hciconfig
hci1: Type: BR/EDR Bus: UART
BD Address: 00:00:A4:17:04:17 ACL MTU: 310:10 SCO MTU: 64:8
UP RUNNING PSCAN ISCAN
RX bytes:53846 acl:0 sco:0 events:339 errors:0
TX bytes:2634 acl:0 sco:0 commands:126 errors:0
```

```
hci0:   Type: BR/EDR  Bus: USB
        BD Address: 00:00:A4:17:04:18  ACL MTU: 310:10  SCO MTU: 64:8
        UP RUNNING PSCAN ISCAN
        RX bytes:316461 acl:0 sco:0 events:1350 errors:0
        TX bytes:1851 acl:0 sco:0 commands:110 errors:0
```

Both modules are in the "UP RUNNING PSCAN ISCAN" status.

### 3.1.3 Inquiry

```
# hcitool -i hci0 inq
Inquiring ...
        00:00:A4:17:04:17  clock offset: 0x1944          class: 0x0c010c
```

```
# hcitool -i hci1 inq
Inquiring ...
        00:00:A4:17:04:18  clock offset: 0x66b9          class: 0x0c010c
```

### 3.1.4 RFCOMM communication

HCI0 is set to Master, HCI1 is set to Slave.

Start the RFCOMM service on the Slave side. And using port 22, it opened the port with the device name "rfcomm1".

```
# rfcomm -i hci1 listen /dev/rfcomm1 22 &
[1] 2140
root@taiyoyuden-ThinkPad-T61p:/home/taiyoyuden# Waiting for connection on channel 22
```

Search service from master side and connect.

```
# rfcomm -i hci0 connect /dev/rfcomm0 00:00:A4:17:04:17 22 &
# Connected /dev/rfcomm0 to 00:00:A4:17:04:17 on channel 22
Press CTRL-C for hangup
```

Slave side was also connected.

```
Connection from 00:00:A4:17:04:18 to /dev/rfcomm1
Press CTRL-C for hangup
```

Text is transmitted from the master side and received by the slave side.

On the slave side, use the "rfcomm" command to display received text. For example:

```
# cat /dev/rfcomm1
```

Send text from master side.

```
# echo "1234567890" > /dev/rfcomm0
```

If the text is displayed on the slave side, it is successful.

```
# cat /dev/rfcomm1
1234567890
```

Note: On the slave side, data other than the text sent from the master side may be displayed. After repeating send text several times, only the text sent from master side will be displayed.

### 3.1.5 Change the communication parameters

Check the Connection Handle being connected.

```
# hcitool -i hci0 con
Connections:
    < ACL 00:00:A4:17:04:17 handle 90 state 1 Im SLAVE
# hcitool -i hci1 con
Connections:
    > ACL 00:00:A4:17:04:18 handle 90 state 1 Im MASTER
```

It can be seen that the handle number is "90".

### Inquiry Interval, windows

Sentence

```
# hciconfig hcix inqparms [win:int] /*Get/Set inquiry scan window and interval
```

Get the current value.

```
# # hciconfig hci0 inqparms
hci0:    Type: BR/EDR Bus: USB
         BD Address: 00:00:A4:17:04:18 ACL MTU: 310:10 SCO MTU: 64:8
         Inquiry interval: 4096 slots (2560.00 ms), window: 18 slots (11.25 ms)
```

Note: Default value: Inquiry interval: 4096 slots (2560.00 ms), window: 18 slots (11.25 ms)

Note: When Interval is set to 1280 msec, Interval = 1280 msec = 2048 slots.

```
# hciconfig hci0 inqparms 18:2048
# hciconfig hci0 inqparms
hci0:    Type: BR/EDR Bus: USB
         BD Address: 00:00:A4:17:04:18 ACL MTU: 310:10 SCO MTU: 64:8
         Inquiry interval: 2048 slots (1280.00 ms), window: 18 slots (11.25 ms)
```

### Paging Interval, Windows

Sentence

```
# hciconfig hcix pageparms [win:int] /*Get/Set page scan window and interval
```

Note: Default value: Page interval: 2048 slots (1280.00 ms), window: 18 slots (11.25 ms)

### 3.1.6 Change the communication mode

Check the AFH settings.

Sentence

```
# hciconfig hcix afhmode [mode] /*Get/Set AFH mode
```

Get the current value

```
# hciconfig hci0 afhmode
hci0:    Type: BR/EDR Bus: USB
         BD Address: 00:00:A4:17:04:18 ACL MTU: 310:10 SCO MTU: 64:8
         AFH mode: Enabled
```

Note: Default value: AFH mode: Enable



## DH1

Use the handle number to change the packet type that is connected. Enter the handle number as a hex value.

```
# hcitool -i hci1 cmd 0x01 0x000F yy yy 10 00
```

Note: yy yy: Connection handle

Example:

```
# hcitool -i hci1 cmd 0x01 0x000F 5A 00 10 00
< HCI Command: ogf 0x01, ocf 0x000f, plen 4
  5A 00 10 00
> HCI Event: 0x0f plen 4
  00 01 0F 04

< HCI Command: Change Connection Packet Type (0x01|0x000f) plen 4
  handle 90 ptype 0x0010
  Packet type: DH1
> HCI Event: Command Status (0x0f) plen 4
  Change Connection Packet Type (0x01|0x000f) status 0x00 ncmd 1
> HCI Event: Connection Packet Type Changed (0x1d) plen 5
  status 0x00 handle 91 ptype 0x0010
  Packet type: DH1
```

## 3-DH5

Use the handle number to change the packet type that is connected. Enter the handle number as a hex value.

```
# hcitool -i hci1 cmd 0x01 0x000F yy yy 00 20
```

Note: yy yy: Connection handle

Example:

```
# hcitool -i hci1 cmd 0x01 0x000F 5A 00 00 20
< HCI Command: ogf 0x01, ocf 0x000f, plen 4
  5A 00 00 20
> HCI Event: 0x0f plen 4
  00 01 0F 04

< HCI Command: Change Connection Packet Type (0x01|0x000f) plen 4
  handle 90 ptype 0x2000
  Packet type: 3-DH5
> HCI Event: Command Status (0x0f) plen 4
  Change Connection Packet Type (0x01|0x000f) status 0x00 ncmd 1
> HCI Event: Connection Packet Type Changed (0x1d) plen 5
  status 0x00 handle 90 ptype 0x2000
  Packet type: 3-DH5
```

### 3.1.7 Power Saving Mode

#### Hold Mode

On the slave side, enter Hold Mode with the following command:

```
# hcitool -i hci1 cmd 0x02 0x0001 yy yy A0 00 A0 00
```

Note: yy yy: Connection handle

#### Example

```
# hcitool -i hci1 cmd 0x02 0x0001 5A 00 A0 00 A0 00
< HCI Command: ogf 0x02, ocf 0x0001, plen 6
  5A 00 A0 00 A0 00
> HCI Event: 0x0f plen 4
  00 01 01 08

< HCI Command: Hold Mode (0x02|0x0001) plen 6
  handle 90 max 160 min 160
> HCI Event: Command Status (0x0f) plen 4
  Hold Mode (0x02|0x0001) status 0x00 ncmd 1
> HCI Event: Mode Change (0x14) plen 6
  status 0x00 handle 90 mode 0x01 interval 160
  Mode: Hold
> HCI Event: Mode Change (0x14) plen 6
  status 0x00 handle 90 mode 0x00 interval 0
  Mode: Active
```

#### Sniff Mode

On the slave side, enter Sniff Mode with the following command:

10, 500 msec = N / 0.625 = 0x0010, 0x0320

```
# hcitool -i hci1 cmd 0x02 0x0003 yy yy 10 00 10 00 01 00 00 00
```

Note: yy yy: Connection handle

#### Example

```
# hcitool -i hci1 cmd 0x02 0x0003 5A 00 10 00 10 00 01 00 00 00
< HCI Command: ogf 0x02, ocf 0x0003, plen 10
  5A 00 10 00 10 00 01 00 00 00
> HCI Event: 0x0f plen 4
  00 01 03 08

< HCI Command: Sniff Mode (0x02|0x0003) plen 10
  handle 90 max 16 min 16 attempt 1 timeout 0
> HCI Event: Command Status (0x0f) plen 4
  Sniff Mode (0x02|0x0003) status 0x00 ncmd 1
> HCI Event: Mode Change (0x14) plen 6
  status 0x00 handle 90 mode 0x02 interval 16
  Mode: Sniff
```

To return from Sniff Mode to Normal Mode, use the following command:

```
# hcitool -i hci1 cmd 0x02 0x0004 yy yy
```

Note: yy yy: Connection handle

Example:

```
# hcitool -i hci1 cmd 0x02 0x0004 5A 00
< HCI Command: ogf 0x02, ocf 0x0004, plen 2
  5A 00
> HCI Event: 0x0f plen 4
  00 01 04 08

< HCI Command: Exit Sniff Mode (0x02|0x0004) plen 2
  handle 90
> HCI Event: Command Status (0x0f) plen 4
  Exit Sniff Mode (0x02|0x0004) status 0x00 ncmd 1
> HCI Event: Mode Change (0x14) plen 6
  status 0x00 handle 90 mode 0x00 interval 0
  Mode: Active
```

### 3.1.8 Disconnection

On the slave side, enter the following command to disconnect:

```
# hcitool -i hci1 cmd 0x01 0x0006 yy yy 16
```

Note: yy yy: Connection handle

Example:

```
# hcitool -i hci1 cmd 0x01 0x0006 5A 00 16
< HCI Command: ogf 0x01, ocf 0x0006, plen 3
  5A 00 16
> HCI Event: 0x0f plen 4
  00 01 06 04
```

RFCOMM was also disconnected.

```
# Connection from 00:00:A4:17:04:18 to /dev/rfcomm1
Press CTRL-C for hangup
Disconnected
```

```
# Connected /dev/rfcomm0 to 00:00:A4:17:04:17 on channel 22
Press CTRL-C for hangup
Disconnected
```

### 3.2 BLE

HCI1 as Peripheral, for Advertising.

HCI0 as Central, for Scanning.

#### 3.2.1 Check the status

```
# hciconfig
hci0:    Type: BR/EDR  Bus: USB
        BD Address: 00:00:A4:17:04:18  ACL MTU: 310:10  SCO MTU: 64:8
        UP RUNNING PSCAN ISCAN
        RX bytes:2762 acl:29 sco:0 events:108 errors:0
        TX bytes:3878 acl:29 sco:0 commands:59 errors:0

hci1:    Type: BR/EDR  Bus: UART
        BD Address: 00:00:A4:17:04:17  ACL MTU: 310:10  SCO MTU: 64:8
        UP RUNNING PSCAN ISCAN
        RX bytes:124811 acl:4371 sco:0 events:4815 errors:0
        TX bytes:68578 acl:4536 sco:0 commands:145 errors:0
```

#### 3.2.2 Advertising

The Bluetooth® side stops scanning, the BLE side starts advertising.

```
# hciconfig hci1 noscan

< HCI Command: Write Scan Enable (0x03|0x001a) plen 1
  enable 0
> HCI Event: Command Complete (0x0e) plen 4
  Write Scan Enable (0x03|0x001a) ncmd 1
  status 0x00

# hciconfig hci1
hci1:    Type: BR/EDR  Bus: UART
        BD Address: 00:00:A4:17:04:17  ACL MTU: 310:10  SCO MTU: 64:8
        UP RUNNING
        RX bytes:12142 acl:0 sco:0 events:101 errors:0
        TX bytes:2580 acl:0 sco:0 commands:56 errors:0

# hciconfig hci1 leadv

< HCI Command: LE Set Advertising Parameters (0x08|0x0006) plen 15
  min 1280.000ms, max 1280.000ms
  type 0x00 (ADV_IND - Connectable undirected advertising) ownbdaddr 0x00 (Public)
  directbdaddr 0x00 (Public) 00:00:00:00:00:00
  channelmap 0x07 filterpolicy 0x00 (Allow scan from any, connection from any)
> HCI Event: Command Complete (0x0e) plen 4
  LE Set Advertising Parameters (0x08|0x0006) ncmd 1
  status 0x00
< HCI Command: LE Set Advertise Enable (0x08|0x000a) plen 1
> HCI Event: Command Complete (0x0e) plen 4
  LE Set Advertise Enable (0x08|0x000a) ncmd 1
  status 0x00
```

Advertising parameters can be set with the following command:

advertising interval default = 1.28sec (0x0800)

Time = N \* 0.625msec

30 msec / 0.625 = 48 = 0x0030

1280 msec / 0.625 = 2048 = 0x0800

Note: 30msec

```
# hcitool -i hci1 cmd 0x08 0x0006 30 00 30 00 00 00 00 00 00 00 00 00 07 00
```

Note: 1280msec

```
# hcitool -i hci1 cmd 0x08 0x0006 00 08 00 08 00 00 00 00 00 00 00 00 07 00
```

### 3.2.3 Scan

Scan Peripheral from Central side.

```
# hcitool -i hci0 lescan
LE Scan ...
D4:4D:A4:AF:D4:56 (unknown)
D4:4D:A4:AF:D4:56 MS-06J
00:00:A4:17:04:17 (unknown)
00:00:A4:17:04:17 CSR - bc7

< HCI Command: LE Set Scan Parameters (0x08|0x000b) plen 7
  type 0x01 (active)
  interval 10.000ms window 10.000ms
  own address: 0x00 (Public) policy: All
> HCI Event: Command Complete (0x0e) plen 4
  LE Set Scan Parameters (0x08|0x000b) ncmd 1
  status 0x00
< HCI Command: LE Set Scan Enable (0x08|0x000c) plen 2
  value 0x01 (scanning enabled)
  filter duplicates 0x01 (enabled)
> HCI Event: Command Complete (0x0e) plen 4
  LE Set Scan Enable (0x08|0x000c) ncmd 1
  status 0x00
> HCI Event: LE Meta Event (0x3e) plen 12
  LE Advertising Report
    ADV_IND - Connectable undirected advertising (0)
    bdaddr 00:00:A4:17:04:17 (Public)
    RSSI: -42
> HCI Event: LE Meta Event (0x3e) plen 12
  LE Advertising Report
    SCAN_RSP - Scan Response (4)
    bdaddr 00:00:A4:17:04:17 (Public)
    RSSI: -42
```

Unlike Bluetooth®, BLE continues to scan with the above command, if confirm the target device, stop scan with Ctrl + C.

Scan parameters can be set with the following command:

BLE Scan interval and window: default = 10msec, 10msec

Note: Interval: 1280msec = 0x0800, Windows: 11msec = 0x0011

```
# hcitool -i hci0 cmd 0x08 0x000B 00 00 08 11 00 00 00
```

### 3.2.4 Connection

Connect by specifying a BD address from the Center side.

```
# # hcitool -i hci0 lecc 00:00:A4:17:04:17
```

Connection handle 91

```
< HCI Command: LE Create Connection (0x08|0x000d) plen 25
  bdaddr 00:00:A4:17:04:17 type 0
  interval 4 window 4 initiator_filter 0
  own_bdaddr_type 0 min_interval 15 max_interval 15
  latency 0 supervision_to 3200 min_ce 1 max_ce 1
> HCI Event: Command Status (0x0f) plen 4
  LE Create Connection (0x08|0x000d) status 0x00 ncmd 1
> HCI Event: LE Meta Event (0x3e) plen 19
  LE Connection Complete
    status 0x00 handle 91, role master
    bdaddr 00:00:A4:17:04:17 (Public)
< HCI Command: LE Read Remote Used Features (0x08|0x0016) plen 2
> HCI Event: Command Status (0x0f) plen 4
  LE Read Remote Used Features (0x08|0x0016) status 0x00 ncmd 1
> HCI Event: LE Meta Event (0x3e) plen 12
  LE Read Remote Used Features Complete
    status 0x00 handle 91
    Features: 0x01 0x00 0x00 0x00 0x00 0x00 0x00 0x00
< ACL data: handle 91 flags 0x00 dlen 7
> HCI Event: Number of Completed Packets (0x13) plen 5
  handle 91 packets 1
```

Connection parameters can be set with the following command:

BLE Scan interval and window: default = 10msec, 10msec

Note: Interval: 105msec = 0x0054

```
# hcitool -i hci0 lecup --handle 91--min 0x0054 --max 0x0054
```

Note: Interval: 1000msec = 0x0320

```
# hcitool -i hci0 lecup --handle 91 --min 0x0320 --max 0x0320
```

### 3.2.5 Disconnection

To disconnect, specify the Connection Handle and using the following command:

```
# hcitool -i hci0 ledc 91

< HCI Command: Disconnect (0x01|0x0006) plen 3
  handle 91 reason 0x13
  Reason: Remote User Terminated Connection
> HCI Event: Command Status (0x0f) plen 4
  Disconnect (0x01|0x0006) status 0x00 ncmd 1
> HCI Event: Disconn Complete (0x05) plen 4
  status 0x00 handle 91 reason 0x16
  Reason: Connection Terminated by Local Host
```