**BSA README**

BSA VERSON: bsa0106\_00.40.00

CHIP: BCM20705(usb port)

**\*\*How to setup build enviorment**

Now, the code can be built on x86 and arm platform, before building the sample code you need to set ARMGCC and PATH of your bulid

system.

Ex:

export ARMGCC=arm-openwrt-linux-gcc

export PATH=[yourtoolchian]/toolchain-arm\_cortex-a5+vfpv3\_gcc-4.8-linaro\_uClibc-0.9.33.2\_eabi/bin:$PATH

**\*\*How to buid sample code**

To build sample code you should build as following:

cd 3rdparty/embedded/bsa\_examples/linux/build\_folder/build

make CPU=platform\_name

Ex: if you want to buld app\_mananger for local build machine, you can do as following:

cd 3rdparty/embedded/bsa\_examples/linux/libbsa/build

make

cd 3rdparty/embedded/bsa\_examples/linux/app\_manager/build

make

if you want to build app\_manager:

cd 3rdparty/embedded/bsa\_examples/linux/app\_manager/build

make CPU=arm

**\*\*Drivers**

btusb: 20705 driver

in 3rdparty/embedded/brcm/linux/btusb folder, please refers to Release\_btusb.txt on how to build and use the driver

bthid: The BTHID driver/module allows a user mode applicatio to submit HID events to the Linux kernel. This is only used for app\_hh

in 3rdparty/embedded/brcm/linux/bthid folder, please refers to Release\_bthid.txt on how to build and use the driver.

**\*\*Application Demo List**

release\_notes/bsa\_examples/Release\_app\_xx.txt

app\_hh -- HH (HID Host): Used to connect to HID Devices (Mouse, Keyboard, Remote Control,)

app\_hd -- HD(HID Device): To act HID device

app\_av -- AV (Audio/Video): Used to stream audio to stereo headset

app\_avk -- AVK (Audio/Video Sink): To act like a stereo headset

app\_ag -- HS/HF –AG (Audio Gateway): Used in a phone or device connected to network

app\_hs -- HS/HF –HS (HeadSet/HandsFree): To act like a mono headset (used by cellular)

app\_fts -- FTP Server -- FTS (File Transfer Server): Used by remote devices (cellular, PC) to access files/folders.

app\_ftc -- FTP Client

app\_ops -- OPP Server -- OPS Object Push Server : Used by remote devices to push/pull files (e.g. business card)

app\_opc -- OPP Client

app\_pbs -- PBS (Phone Book Server): Used by remote devices to access local phone book.

app\_pbc -- Phone Book Profile Client

app\_pam -- Personal Area Networking Profile (PAN)

app\_hl -- HDP (Health Device Profile): Used for exchange of medical device data

app\_mce – MAP(Message Access Profile) client

app\_3d – 3D Synchronization Profile

app\_tm – Test Mode, for RF test

app\_dg -- SPP (Serial Port Profile): Used for wireless replacement of serial cable

app\_ble -- GATT

app\_ble\_cscc – BLE CSC(Cycling speed and cadence) controller

app\_ble\_hrc -- BLE Heart Rate Controller

app\_ble\_pm -- BLE Proximity Monitor

app\_ble\_rscc -- BLE RSC(Running speed and cadence) controller

app\_hogp -- HOGP host

**\*\*How to run the application**

In 3rdparty/embedded/bsa\_examples/linux/app\_xx/build/Release\_app\_xx.txt, you can find the notes of sample applicattion.

To run the sample application, You need to make sure the following step:

1. 20705 dongle is connected with you system

2. insmod btusb. btusb driver is loaded accodrding to Release\_btusb.txt

insmod /udata/nfs/btusb.ko

mknod /dev/btusb0 c 180 194

chmod 666 /dev/btusb0

3. bsa\_server is released in binary, please check the bsa\_server for different platform

bsa\_server stored path: 3rdparty/embedded/bsa\_examples/linux/server/build

4. bsa\_server is started as following:

./bsa\_server -d /dev/btusb0 -p your\_patchram\_path/patchram\_file

And you can use ./bsa\_server --help to check the option that can be used for bsa\_server.

-d para: uart -- /dev/ttyXX usb -- /dev/btusb0

If you use uart port, you can set baudrate by ‘-r’.

./bsa\_server -d /dev/ttyXX -p xx.hcd -r 10

-d device set the hci device interface (e.g. /dev/ttyHS0)

-p patchfile force the patchfile (RAM) to use

-r baudrate set baud rate for opening port

Baud rate codes:

0 = 300 baud

1 = 600

2 = 1200

3 = 2400

4 = 9600

5 = 19200

6 = 57600

7 = 115200

8 = 230400

9 = 460800

10 = 921600

11 = 1M

12 = 1.5M

13 = 2M

14 = 3M

15 = 4M

5. run ./app\_manager

6. run sample application that you need, and you can try the sample application

**\*\*How to catch log**

For issue tracking, bsa log and btsnoop are necessary.

Ex. ./bsa\_server -d /dev/btusb0 -p xx.hcd **-all=5 -b btsnoop.cfa > bsa.log**

1. bsa log

You call use “-all=#” to set trace level to # for all layers.

Usually you can set “all=5” to catch issue trace for developing.

And you can set “all=0” to disable all trace for publishing.

2. btsnoop

You can use “-b file” to specify the BT Snoop file to save serial packets.

**\*\* Detailed Example for Application running**

Chip: Based on 20705 usb port

AV Headset : Motorola Elite Flip

HID device: Microsoft Sculpt Mobile Keyboard

BLE HID device: Xiaomi Remote Controller

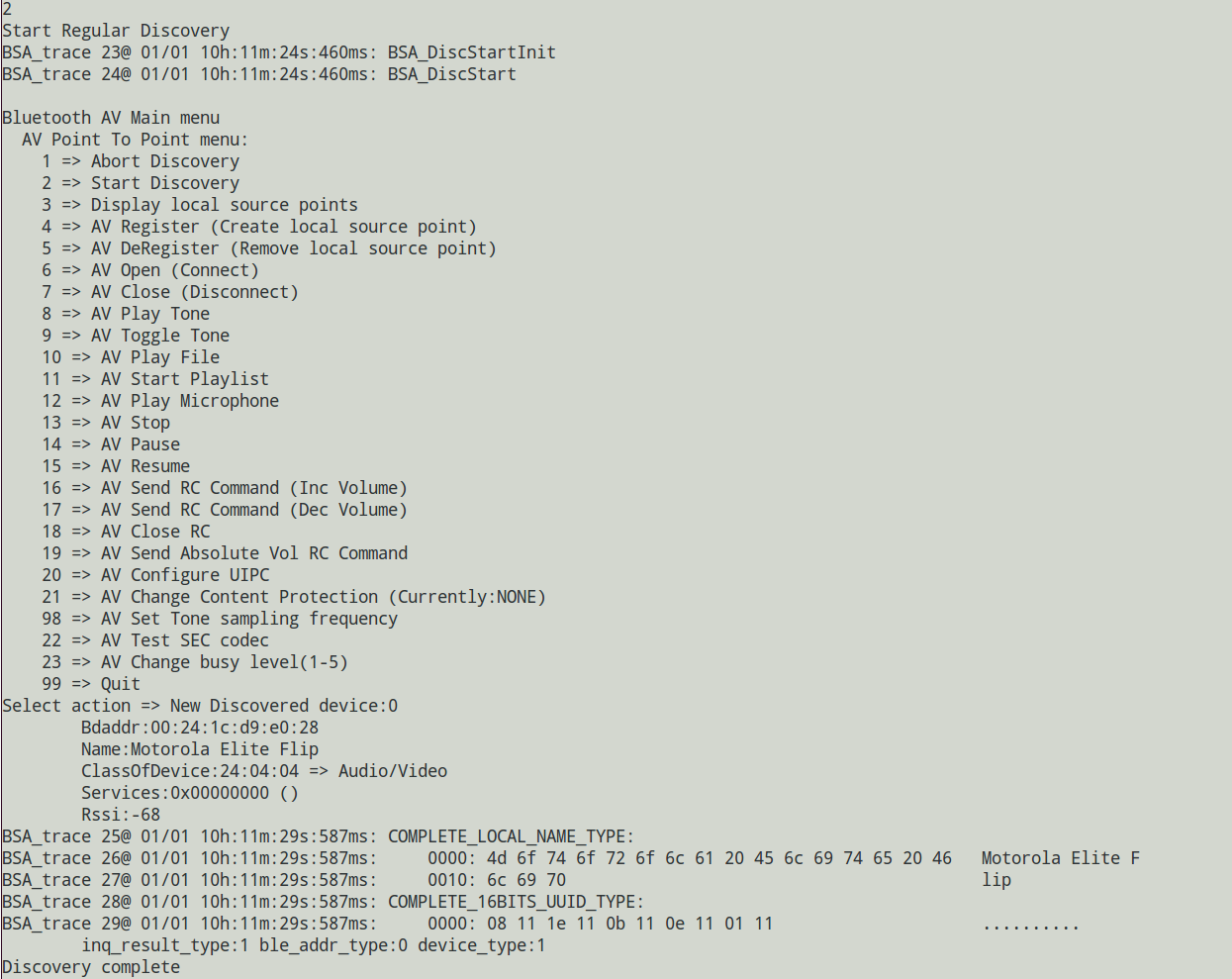
1. **AV -- connect stereo headset and play music**
2. **First pair and play:**

Preparatory work:

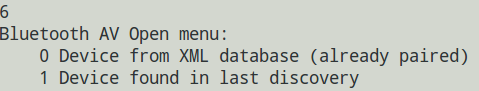
1. put music to Music folder path: ./test\_files/av
2. Open Headset, entry Bluetooth pair mode.

Start:

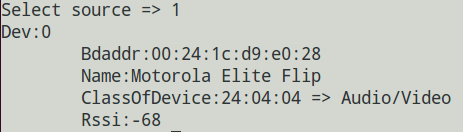
1. Run ./bsa\_server -d /dev/btusb0 -p patchram.hcd
2. When bsa\_server started successfully, run ./app\_manager, ./app\_av in sequence.
3. app\_av menu: input **2**  (Start Discovery)



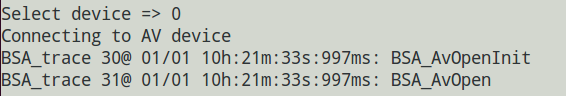
1. input  **6**  (start connect)



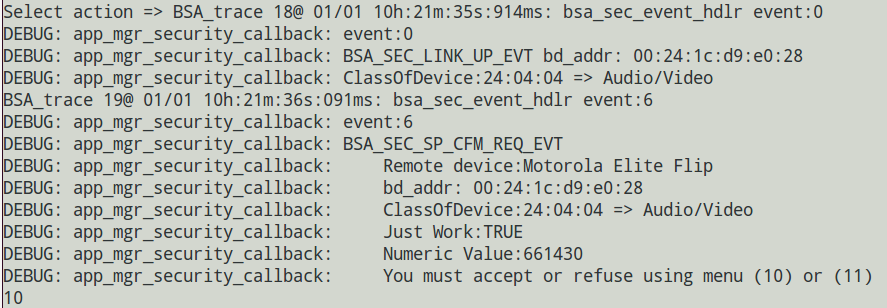
1. input **1**



1. input **0**

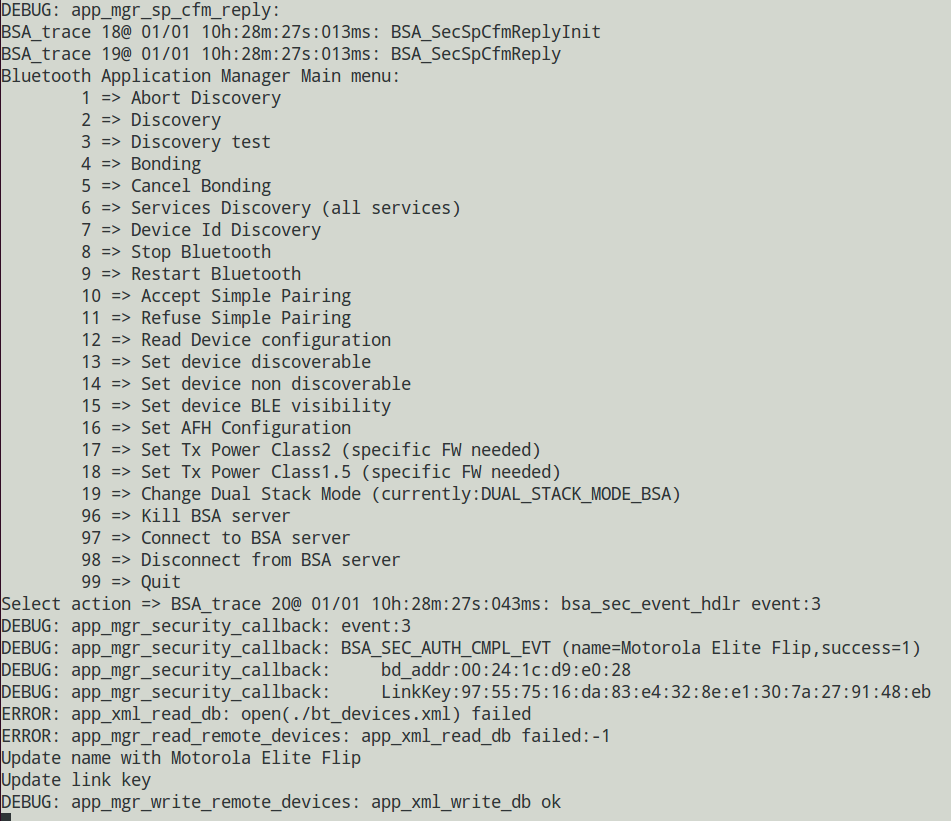


1. app\_manager input **10**  (accept pair)

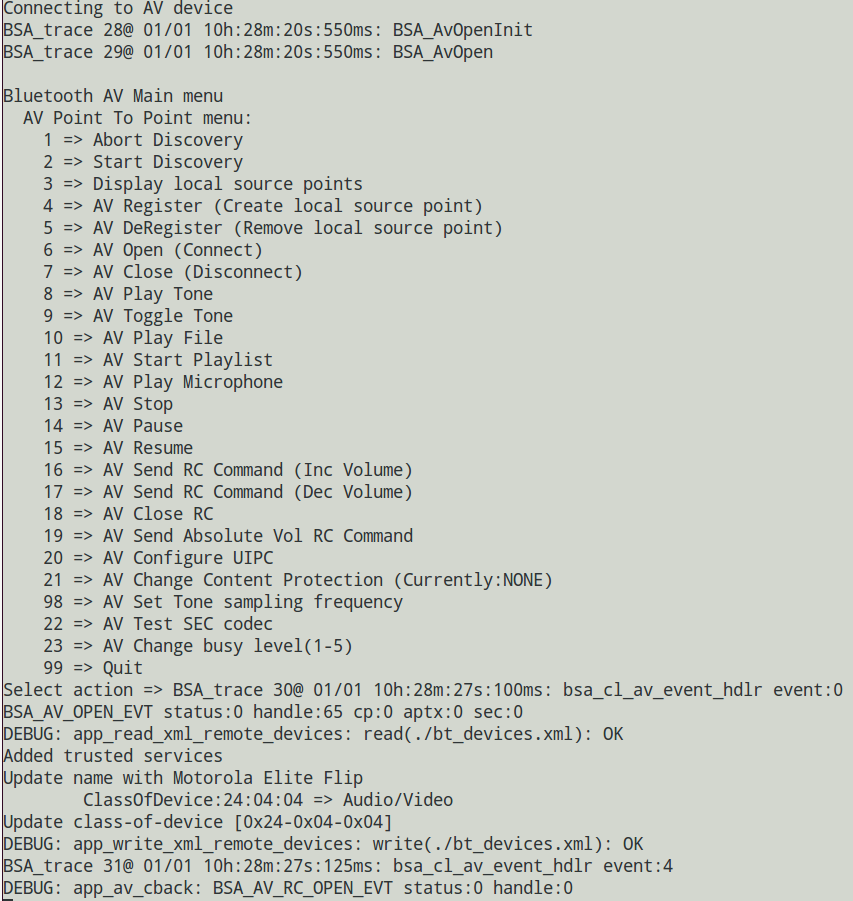


1. you can hear the headset hint connected, and find:

**app\_manager:**



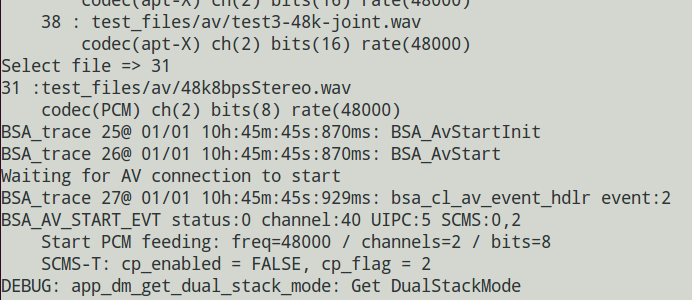
app\_av:

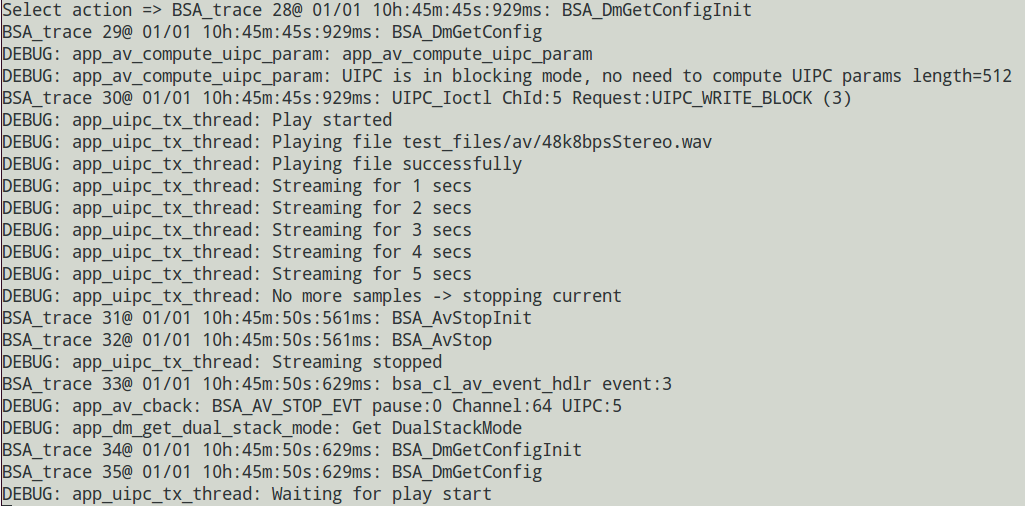


1. Play File

input **10** to play specified music **OR** input **11** to play music list in folder.

If input **10**, and then select specified music, as below:





1. **Reconnect Headset**

If you disconnect Headset and want to reconnect:

1. run ./bsa\_server -d /dev/btusb0 -p patchram.hcd (if not running)
2. run ./app\_manager, ./app\_av (if not running)
3. open BT headset

**2. Audio gateway(connect to headset)**

a. Put "sco\_ag\_out.wav"  into ./test\_files/ag

b. Run ./bsa\_server -d /dev/btusb0 -p patchram.hcd

c. When bsa\_server started successfully, run ./app\_manager, ./app\_ag in sequence.

d. app\_ag menu(See the bold red part):

./app\_ag

BSA\_trace 1@ 01/01 23h:14m:13s:748ms: BSA\_MgtOpenInit

BSA\_trace 2@ 01/01 23h:14m:13s:748ms: BSA\_MgtOpen (./)

Created thread with thread\_id=  df18c700

BSA\_trace 3@ 01/01 23h:14m:13s:748ms: bsa\_cl\_mgt\_init

BSA\_trace 4@ 01/01 23h:14m:13s:748ms: UIPC\_Init

BSA\_trace 5@ 01/01 23h:14m:13s:748ms: UIPC\_Open ChId:3

DEBUG: app\_ag\_enable: Entering

BSA\_trace 6@ 01/01 23h:14m:13s:748ms: BSA\_AgEnableInit

BSA\_trace 7@ 01/01 23h:14m:13s:748ms: BSA\_AgEnable

DEBUG: app\_ag\_register: Entering

BSA\_trace 8@ 01/01 23h:14m:13s:749ms: BSA\_AgRegisterInit

BSA\_trace 9@ 01/01 23h:14m:13s:749ms: BSA\_AgRegister

DEBUG: app\_ag\_register: Register complete handle 1, status 0/success

Bluetooth Headset/HandsFree Main menu:

    1  => Discover Headset

    2  => Enable

    3  => Disable

    4  => Register

    5  => Deregister

    6  => Connect

    7  => Disconnect

    8  => Open audio

    9  => Close audio

    10  => Record audio file

    11  => Stop record audio file

    12  => Play audio file

    13  => Stop Playing audio file

    14  => Display SCO route config

    15  => Indicate incoming call

    16  => PickUp call

    17  => HangUp call

    99  => Quit

**Select action => 1**

Start COD filtered Discovery

Look for COD service:2000 major:4 minor:0

BSA\_trace 10@ 01/01 23h:14m:15s:804ms: BSA\_DiscStartInit

COD: 0x20 0x04 0x00

COD Mask: 0x20 0x1f 0x00

BSA\_trace 11@ 01/01 23h:14m:15s:804ms: BSA\_DiscStart

Bluetooth Headset/HandsFree Main menu:

    1  => Discover Headset

    2  => Enable

    3  => Disable

    4  => Register

    5  => Deregister

    6  => Connect

    7  => Disconnect

    8  => Open audio

    9  => Close audio

    10  => Record audio file

    11  => Stop record audio file

    12  => Play audio file

    13  => Stop Playing audio file

    14  => Display SCO route config

    15  => Indicate incoming call

    16  => PickUp call

    17  => HangUp call

    99  => Quit

Select action => New Discovered device:0

Bdaddr:00:0c:8a:3c:13:2d

Name:Bose BT2 01.05.00

ClassOfDevice:24:04:04 => Audio/Video

Services:0x00000000 ()

Rssi:-51

DeviceType:BR/EDR InquiryType:BR AddressType:Public

Extended Information:

    ShortName: Bose BT2 01

    TxPower:4 dB

    Complete Service [UUID16]:

        0x111E [Handsfree]

        0x1108 [Headset]

        0x110D [Advanced Audio Distribution]

        0x110B [Audio Sink]

Discovery complete

**6**

DEBUG: app\_ag\_open: Entering

Bluetooth AG menu:

    0 Device from XML database (already paired)

    1 Device found in last discovery

**Select source => 1**

Dev:0

Bdaddr:00:0c:8a:3c:13:2d

Name:Bose BT2 01.05.00

ClassOfDevice:24:04:04 => Audio/Video

Rssi:-51

**Select device => 0**

BSA\_trace 12@ 01/01 23h:14m:25s:140ms: BSA\_AgOpenInit

BSA\_trace 13@ 01/01 23h:14m:25s:140ms: BSA\_AgOpen

BSA\_trace 14@ 01/01 23h:14m:25s:140ms: BSA\_AgOpen BDA: 00:0c:8a:3c:13:2d

Bluetooth Headset/HandsFree Main menu:

    1  => Discover Headset

    2  => Enable

    3  => Disable

    4  => Register

    5  => Deregister

    6  => Connect

    7  => Disconnect

    8  => Open audio

    9  => Close audio

    10  => Record audio file

    11  => Stop record audio file

    12  => Play audio file

    13  => Stop Playing audio file

    14  => Display SCO route config

    15  => Indicate incoming call

    16  => PickUp call

    17  => HangUp call

    99  => Quit

Select action => BSA\_trace 15@ 01/01 23h:14m:27s:956ms: bsa\_ag\_event\_hdlr event:0

DEBUG: app\_ag\_cback: event:0 for handle: 1, index: 0

DEBUG: app\_ag\_cback: BSA\_AG\_OPEN\_EVT: s=0(success)

DEBUG: app\_ag\_cback: 00:0c:8a:3c:13:2d

BSA\_trace 16@ 01/01 23h:14m:28s:186ms: bsa\_ag\_event\_hdlr event:13

DEBUG: app\_ag\_cback: event:13 for handle: 1, index: 0

DEBUG: app\_ag\_cback: BSA\_AG\_AT\_CIND\_EVT

BSA\_trace 17@ 01/01 23h:14m:28s:186ms: BSA\_AgResult

BSA\_trace 18@ 01/01 23h:14m:28s:302ms: bsa\_ag\_event\_hdlr event:2

DEBUG: app\_ag\_cback: event:2 for handle: 1, index: 0

DEBUG: app\_ag\_cback: BSA\_AG\_CONN\_EVT: 00:0c:8a:3c:13:2d

BSA\_trace 19@ 01/01 23h:14m:28s:369ms: bsa\_ag\_event\_hdlr event:18

DEBUG: app\_ag\_cback: event:18 for handle: 1, index: 0

DEBUG: app\_ag\_cback: BSA\_AG\_AT\_NREC\_EVT

BSA\_trace 20@ 01/01 23h:14m:28s:439ms: bsa\_ag\_event\_hdlr event:6

DEBUG: app\_ag\_cback: event:6 for handle: 1, index: 0

DEBUG: app\_ag\_cback: BSA\_AG\_SPK\_EVT 12

BSA\_trace 21@ 01/01 23h:14m:28s:590ms: bsa\_ag\_event\_hdlr event:21

DEBUG: app\_ag\_cback: event:21 for handle: 1, index: 0

DEBUG: app\_ag\_cback: BSA\_AG\_AT\_CLCC\_EVT

BSA\_trace 22@ 01/01 23h:14m:28s:590ms: BSA\_AgResult

**8**

DEBUG: app\_ag\_open\_audio: Entering

BSA\_trace 23@ 01/01 23h:14m:32s:628ms: BSA\_AgAudioOpenInit

BSA\_trace 24@ 01/01 23h:14m:32s:628ms: BSA\_AgAudioOpen

Bluetooth Headset/HandsFree Main menu:

    1  => Discover Headset

    2  => Enable

    3  => Disable

    4  => Register

    5  => Deregister

    6  => Connect

    7  => Disconnect

    8  => Open audio

    9  => Close audio

    10  => Record audio file

    11  => Stop record audio file

    12  => Play audio file

    13  => Stop Playing audio file

    14  => Display SCO route config

    15  => Indicate incoming call

    16  => PickUp call

    17  => HangUp call

    99  => Quit

Select action => BSA\_trace 25@ 01/01 23h:14m:32s:678ms: bsa\_ag\_event\_hdlr event:3

DEBUG: app\_ag\_cback: event:3 for handle: 1, index: 0

DEBUG: app\_ag\_cback: BSA\_AG\_AUDIO\_OPEN\_EVT: s=0(success)

BSA\_trace 26@ 01/01 23h:14m:32s:678ms: UIPC\_Open ChId:9

BSA\_trace 27@ 01/01 23h:14m:32s:678ms: uipc\_rb\_open name:sco-tx-rb id:9 size:2401 flags:x2

BSA\_trace 28@ 01/01 23h:14m:32s:678ms: uipc\_rb\_open: Size = 2401

BSA\_trace 29@ 01/01 23h:14m:32s:679ms: uipc\_rb\_open: Buffer = 0x7f92dfb9d010

BSA\_trace 30@ 01/01 23h:14m:32s:679ms: uipc\_rb\_open: semid = 720896

BSA\_trace 31@ 01/01 23h:14m:32s:679ms: uipc\_rb\_open: wr\_lockid = 753665

BSA\_trace 32@ 01/01 23h:14m:32s:679ms: uipc\_rb\_open: wr\_lockid.val = 2400

BSA\_trace 33@ 01/01 23h:14m:32s:679ms: uipc\_cl\_sco\_channel\_open SCO channels opened

BSA\_trace 34@ 01/01 23h:14m:32s:679ms: uipc\_rb\_open name:sco-rx-rb id:10 size:2401 flags:x1

BSA\_trace 35@ 01/01 23h:14m:32s:679ms: uipc\_rb\_open: Size = 2401

BSA\_trace 36@ 01/01 23h:14m:32s:679ms: uipc\_rb\_open: Buffer = 0x7f92dfb9c010

BSA\_trace 37@ 01/01 23h:14m:32s:679ms: uipc\_rb\_open: semid = 819203

BSA\_trace 38@ 01/01 23h:14m:32s:679ms: uipc\_rb\_open: wr\_lockid = 851972

BSA\_trace 39@ 01/01 23h:14m:32s:679ms: uipc\_rb\_open: wr\_lockid.val = 2400

BSA\_trace 40@ 01/01 23h:14m:32s:679ms: uipc\_cl\_sco\_channel\_open SCO channels opened

BSA\_trace 41@ 01/01 23h:14m:32s:679ms: UIPC\_Ioctl ChId:9 Request:UIPC\_REG\_CBACK (5)

BSA\_trace 42@ 01/01 23h:14m:32s:679ms: uipc\_cl\_sco\_ioctl UIPC\_REG\_CBACK 0

BSA\_trace 43@ 01/01 23h:14m:32s:679ms: uipc\_cl\_sco\_read\_task p\_cback 0x4023b0 , arg 0x4023b0

**12**

Bluetooth Headset/HandsFree Main menu:

    1  => Discover Headset

    2  => Enable

    3  => Disable

    4  => Register

    5  => Deregister

    6  => Connect

    7  => Disconnect

    8  => Open audio

    9  => Close audio

    10  => Record audio file

    11  => Stop record audio file

    12  => Play audio file

    13  => Stop Playing audio file

    14  => Display SCO route config

    15  => Indicate incoming call

    16  => PickUp call

    17  => HangUp call

    99  => Quit

Select action => DEBUG: app\_ag\_play\_file\_thread: Entering

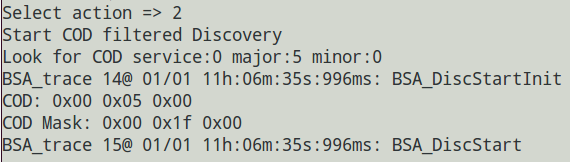
**3. HID Device**

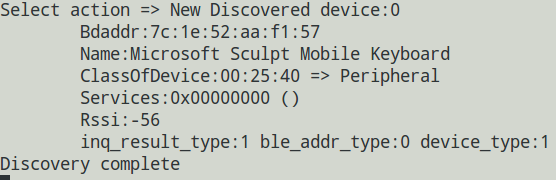
Preparatory work:

MS Keyboard entry Bluetooth pair mode.

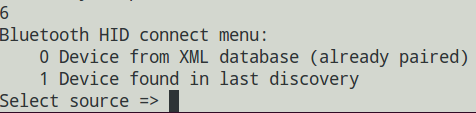
Start:

1. Run ./bsa\_server -d /dev/btusb0 -p patchram.hcd
2. When bsa\_server started successfully, run ./app\_manager, ./app\_hh in sequence.
3. app\_hh menu: input **2**  (COD Discovery (Peripheral Dev))

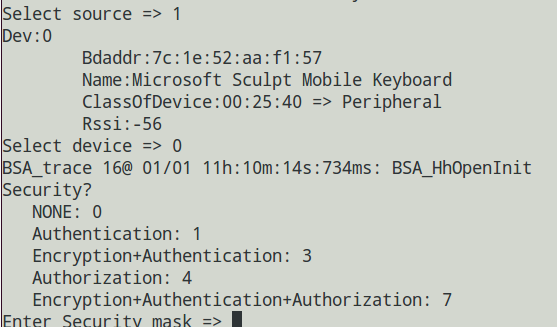




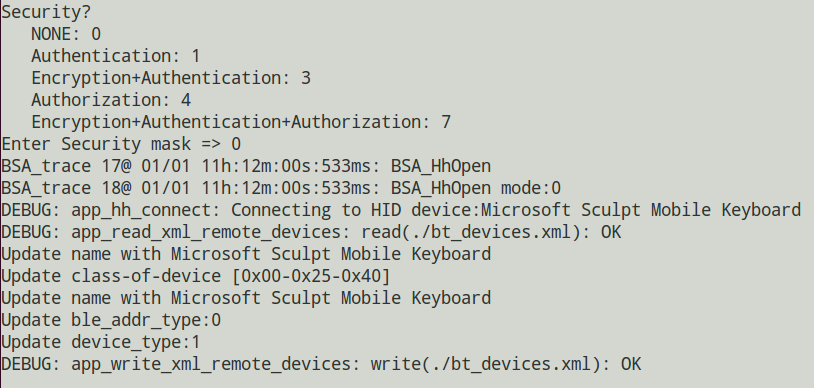
1. input **6** (HID Connect (Report mode))

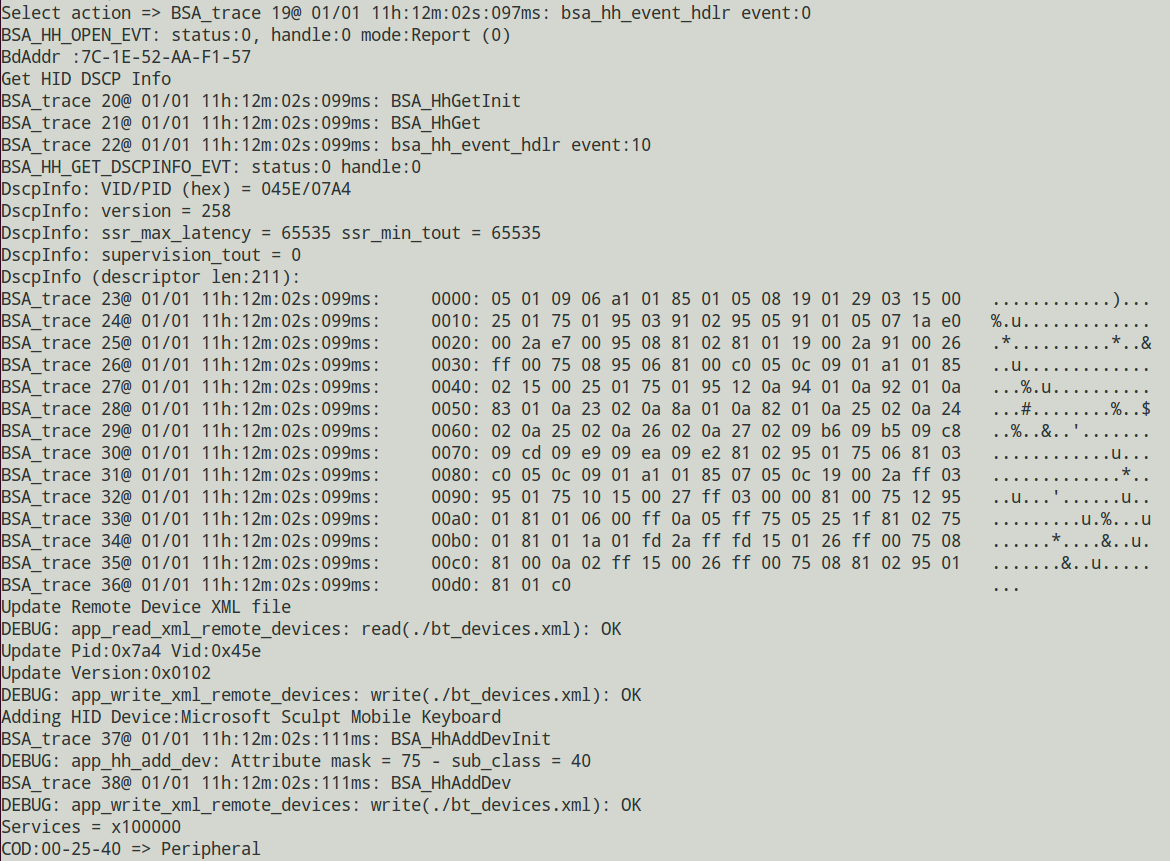


1. input **1** , and then input **0** to select MS Keyboard



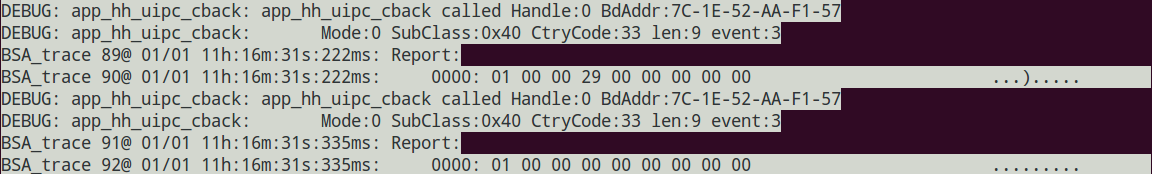
1. input **0**, connect successfully. (Security select depends on HID device)





if you input key in keyboard, you can find data display on app\_hh:

For example, press “ESC”



**4. BLE HID device:**

Preparatory work:

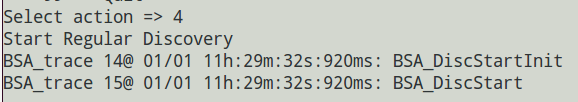
Xiaomi BLE remote controller entry Bluetooth pair mode.

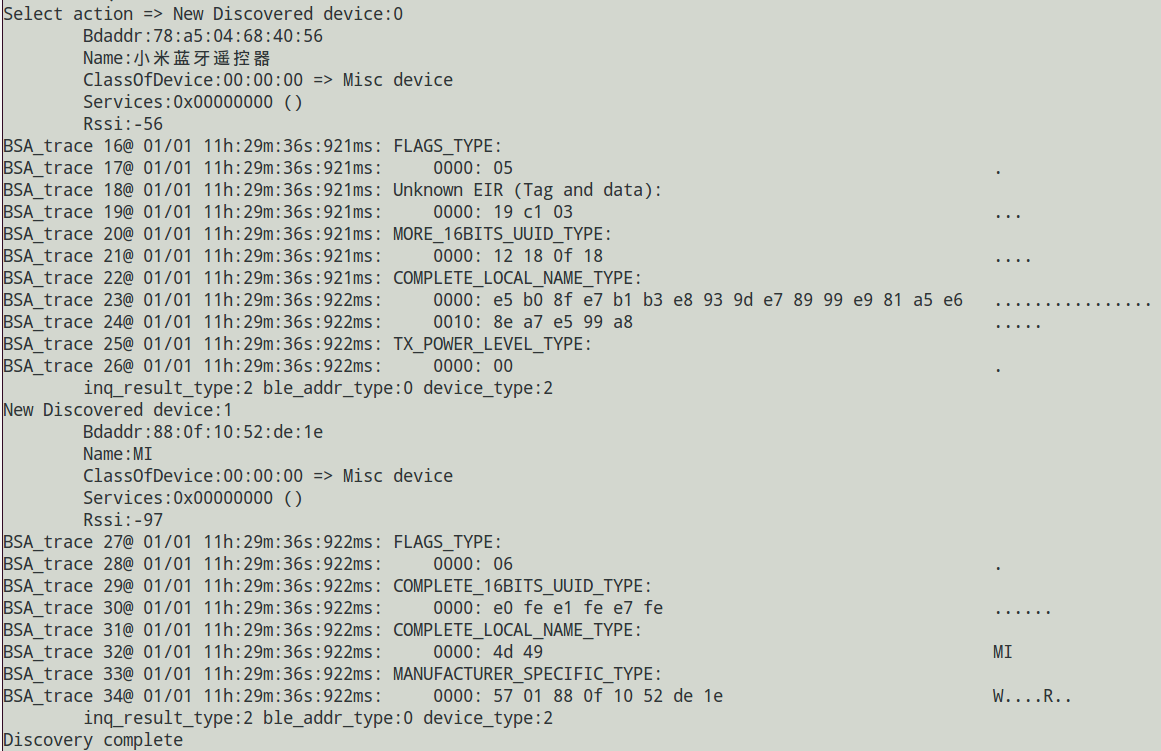
Start:

a. Run ./bsa\_server -d /dev/btusb0 -p patchram.hcd

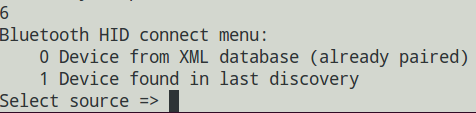
b. When bsa\_server started successfully, run ./app\_manager,./app\_ble, ./app\_hh in sequence**(Or just run app\_hogp)**

c. input **4** (BLE HID deivce Discovery)

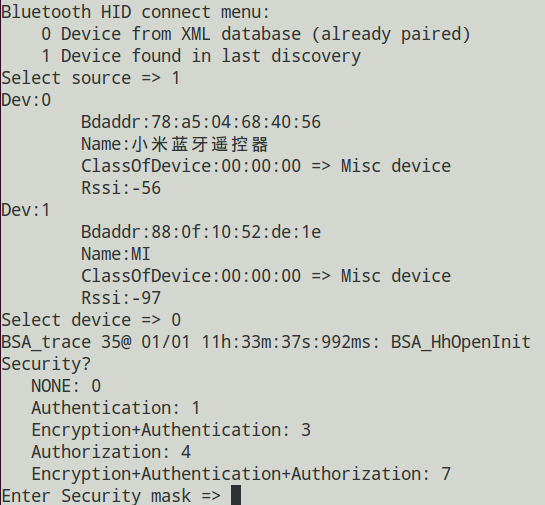




d. input **6** (HID Connect (Report mode))



e. input **1** , and then input **0** to select Xiaomi RC



f. input 0, connect successfully. (Security select depends on HID device)

At app\_hh side, the display about connected successfully and pressed key is similar with BR/EDR HID devices.