CYPRESS®
EMBEDDED IN TOMORROW™

RUNNING BLUEZ5 ON CY PARTS

with Ubuntu Linux

July 22, 2020

Scope

- This document aims at providing a quick start guide for users who run Bluez 5.0 tools to initialize Cypress Bluetooth Controller, which is connected to *Ubuntu Linux* machine through **UART** transport.
  - ►For USB transport, check the "*USB*" slides.
- For Bluez 5.0 feature and functionality, it is out of Cypress support scope. Please refer to http://www.bluez.org/release-of-bluez-5-0/ for Bluez information.

Download and build user-space tools

- Source
  - http://www.bluez.org/download/
  - *tar xf bluez-5.??.tar.xz*
- Build
  - *cd bluez-5.??*
  - *./configure*
  - *make*
  - *sudo make install*

Chip and HCI-UART Transport Initialization

Firmware download and HCI UART bring-up

- Bluez5 *hciattach* tool has supported firmware run-time RAM download for CY Bluetooth parts.
  - bluez-5.??/tools/hciattach_bcm43xx.c
  - Applicable on all CY Bluetooth parts supporting firmware run-time RAM download
- Firmware .hcd file
  - The firmware filename should be started with chip's default device name, i.e. the local device name read back from ROM firmware by HCI_Read_Local_Name (OGF 0x03, OCF 0x0014), and be ended with .hcd
    - For example, for BCM89335, a valid filename could be

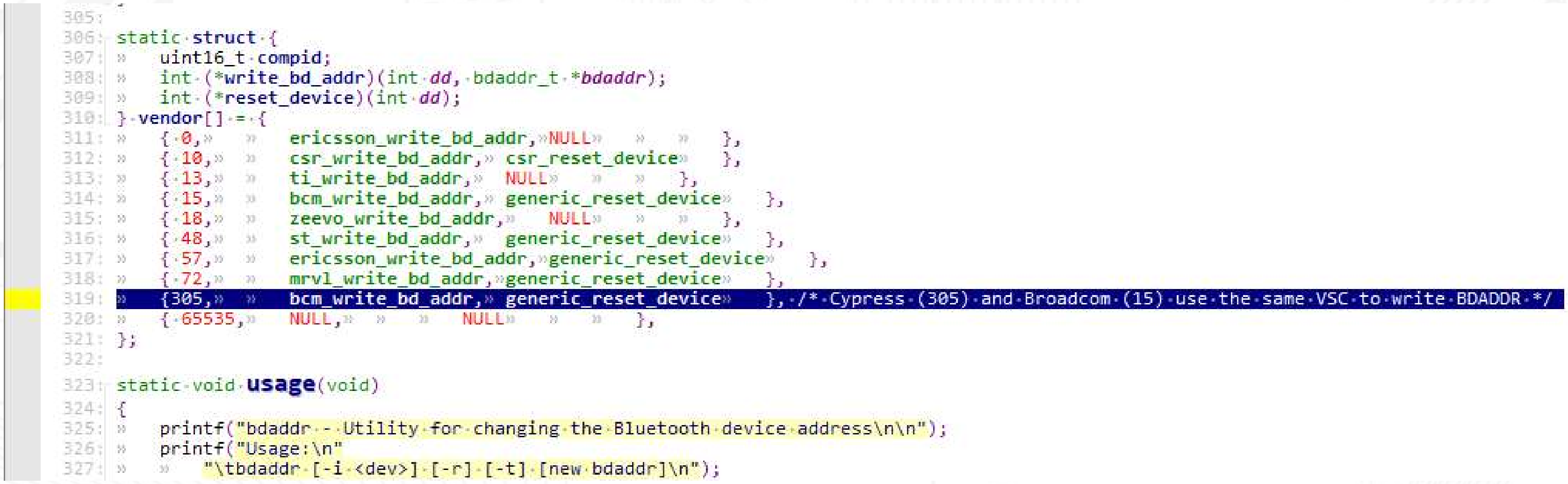
      BCM4335C0_BCM4339_003.001.009.0127.0001.hcd
  - Copy and place the .hcd file in Ubuntu system's /etc/firmware/ folder
- Launch
  - *sudo hciattach -n –p /dev/ttyUSB0 bcm43xx*

Result of launching *hciattach*

```
wcosa@e6400:~$ sudo hciattach -n -p /dev/ttyUSB0 bcm43xx

[sudo] password for wcosa:

bcm43xx_init

Set Controller UART speed to 3000000 bit/s

Flash firmware /etc/firmware/BCM4335C0_BCM4339_003.001.009.0127.0001.hcd

Set Controller UART speed to 3000000 bit/s

Device setup complete
```

Assign Bluetooth device address after firmware download

- Bluez5 ***bdaddr*** tool can be used to write a Bluetooth device address into Bluetooth Controller after run-time RAM firmware has been downloaded.
  - bluez-5.??/tools/bdaddr.c
- Patch to ***bdaddr*** source
  - Add the highlighted code in the *vendor[]* structure:

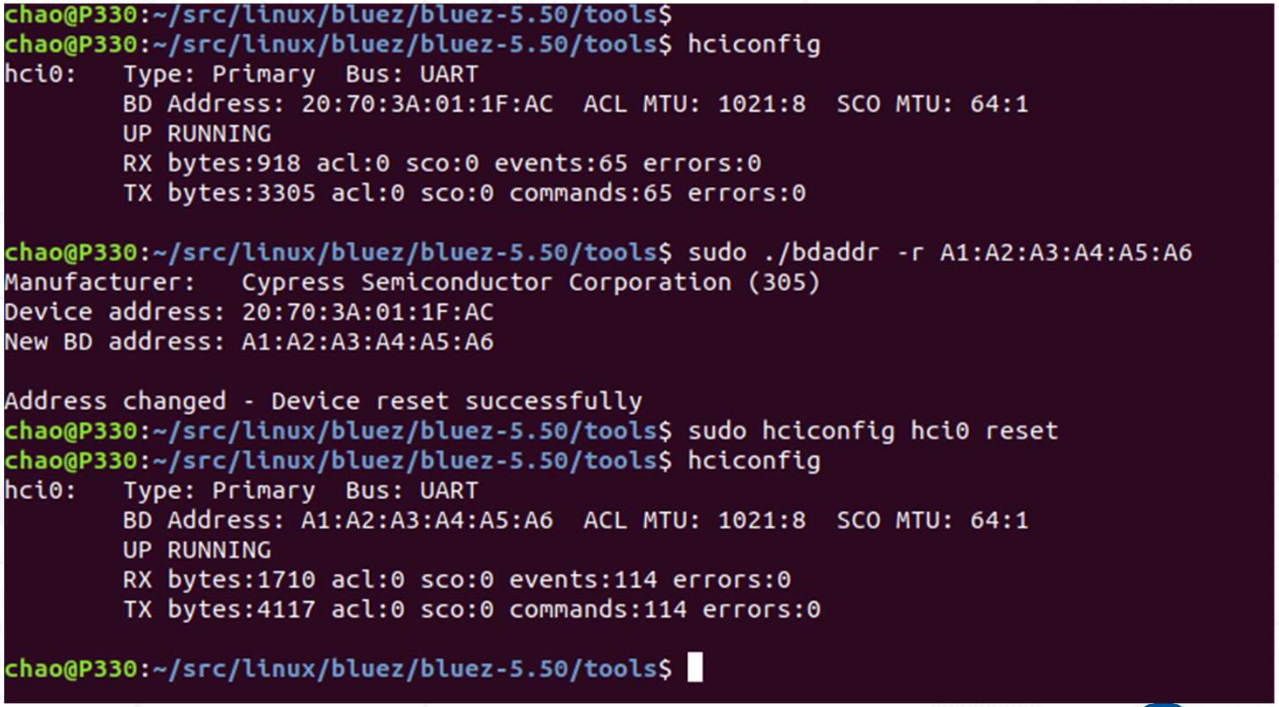


```
static struct {
	uint16_t compid;
	int (*write_bd_addr)(int dd, bdaddr_t *bdaddr);
	int (*reset_device)(int dd);
} vendor[] = {
	{ 0,		ericsson_write_bd_addr,	NULL			},
	{ 10,		csr_write_bd_addr,	csr_reset_device	},
	{ 13,		ti_write_bd_addr,	NULL			},
	{ 15,		bcm_write_bd_addr,	generic_reset_device	},
	{ 18,		zeevo_write_bd_addr,	NULL			},
	{ 48,		st_write_bd_addr,	generic_reset_device	},
	{ 57,		ericsson_write_bd_addr,	generic_reset_device	},
	{ 72,		mrvl_write_bd_addr,	generic_reset_device	},
	{305,		bcm_write_bd_addr,	generic_reset_device	}, /* Cypress (305) and Broadcom (15) use the same VSC to write BDADDR */
	{ 65535,	NULL,			NULL			},
};

static void usage(void)
{
	printf("bdaddr - Utility for changing the Bluetooth device address\n\n");
	printf("Usage:\n"
		"\tbdaddr [-i <dev>] [-r] [-t] [new bdaddr]\n");
```

  - Then rebuild the ***bdaddr*** executable binary

Command sequences to update BDADDR

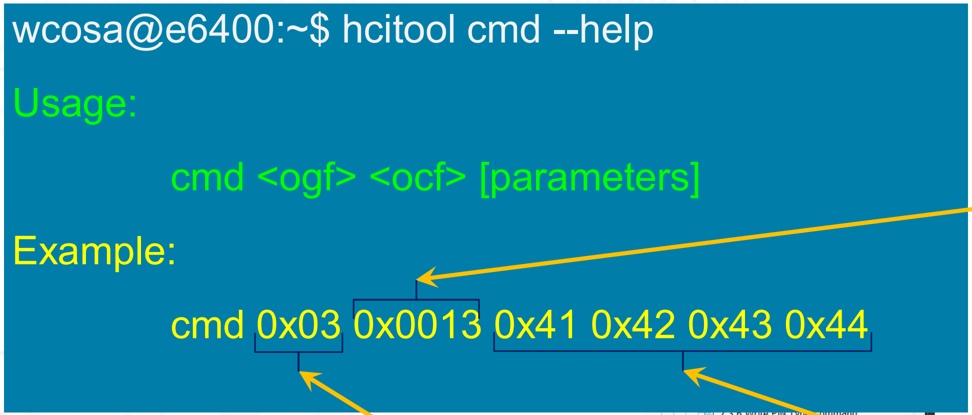
- Set a new BDADDR
  - bdaddr -r <Bluetooth Device Address>
  - Example: *sudo bdaddr -r A1:A2:A3:A4:A5:A6*
- Reset Bluez HCI device
  - *sudo hciconfig hci0 reset*



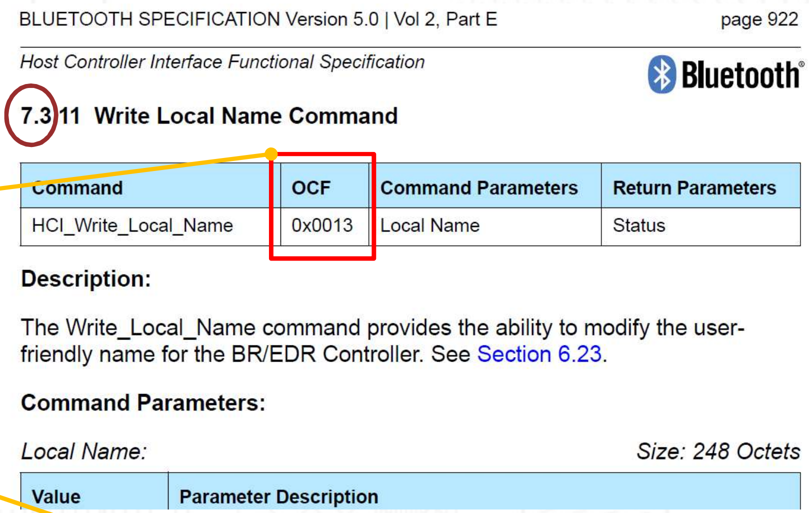
```
chao@P330:~/src/linux/bluez/bluez-5.50/tools$
chao@P330:~/src/linux/bluez/bluez-5.50/tools$ hciconfig
hci0:   Type: Primary  Bus: UART
        BD Address: 20:70:3A:01:1F:AC  ACL MTU: 1021:8  SCO MTU: 64:1
        UP RUNNING
        RX bytes:918 acl:0 sco:0 events:65 errors:0
        TX bytes:3305 acl:0 sco:0 commands:65 errors:0

chao@P330:~/src/linux/bluez/bluez-5.50/tools$ sudo ./bdaddr -r A1:A2:A3:A4:A5:A6
Manufacturer:   Cypress Semiconductor Corporation (305)
Device address: 20:70:3A:01:1F:AC
New BD address: A1:A2:A3:A4:A5:A6

Address changed - Device reset successfully
chao@P330:~/src/linux/bluez/bluez-5.50/tools$ sudo hciconfig hci0 reset
chao@P330:~/src/linux/bluez/bluez-5.50/tools$ hciconfig
hci0:   Type: Primary  Bus: UART
        BD Address: A1:A2:A3:A4:A5:A6  ACL MTU: 1021:8  SCO MTU: 64:1
        UP RUNNING
        RX bytes:1710 acl:0 sco:0 events:114 errors:0
        TX bytes:4117 acl:0 sco:0 commands:114 errors:0

chao@P330:~/src/linux/bluez/bluez-5.50/tools$ 
```

HCI Commands Injection and Logging

HCI commands injection



```
wcosa@e6400:~$ hcitool cmd --help
Usage:
        cmd <ogf> <ocf> [parameters]
Example:
        cmd 0x03 0x0013 0x41 0x42 0x43 0x44
```



BLUETOOTH SPECIFICATION Version 5.0 | Vol 2, Part E page 922

Host Controller Interface Functional Specification

7.3.11 Write Local Name Command

Command	OCF	Command Parameters	Return Parameters
HCI_Write_Local_Name	0x0013	Local Name	Status

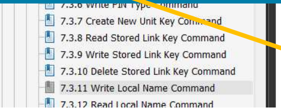
Description:

The Write_Local_Name command provides the ability to modify the user-friendly name for the BR/EDR Controller. See Section 6.23.

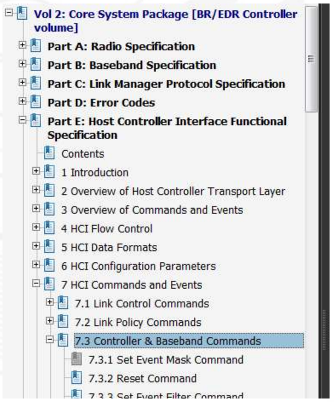
Command Parameters:

Local Name: *Size: 248 Octets*

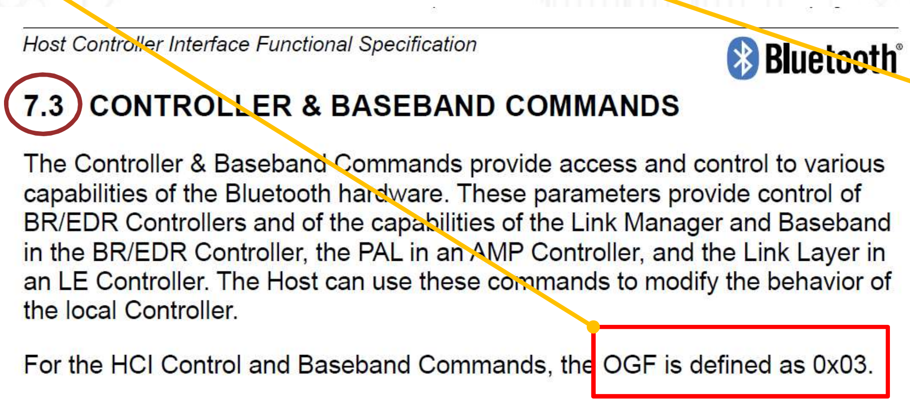
Value	Parameter Description



- 7.3.6 Write PIN Type Command
- 7.3.7 Create New Unit Key Command
- 7.3.8 Read Stored Link Key Command
- 7.3.9 Write Stored Link Key Command
- 7.3.10 Delete Stored Link Key Command
- 7.3.11 Write Local Name Command
- 7.3.12 Read Local Name Command



- Vol 2: Core System Package [BR/EDR Controller volume]
  - Part A: Radio Specification
  - Part B: Baseband Specification
  - Part C: Link Manager Protocol Specification
  - Part D: Error Codes
  - Part E: Host Controller Interface Functional Specification
    - Contents
    - 1 Introduction
    - 2 Overview of Host Controller Transport Layer
    - 3 Overview of Commands and Events
    - 4 HCI Flow Control
    - 5 HCI Data Formats
    - 6 HCI Configuration Parameters
    - 7 HCI Commands and Events
      - 7.1 Link Control Commands
      - 7.2 Link Policy Commands
      - 7.3 Controller & Baseband Commands
        - 7.3.1 Set Event Mask Command
        - 7.3.2 Reset Command
        - 7.3.3 Set Event Filter Command



Host Controller Interface Functional Specification

7.3 CONTROLLER & BASEBAND COMMANDS

The Controller & Baseband Commands provide access and control to various capabilities of the Bluetooth hardware. These parameters provide control of BR/EDR Controllers and of the capabilities of the Link Manager and Baseband in the BR/EDR Controller, the PAL in an AMP Controller, and the Link Layer in an LE Controller. The Host can use these commands to modify the behavior of the local Controller.

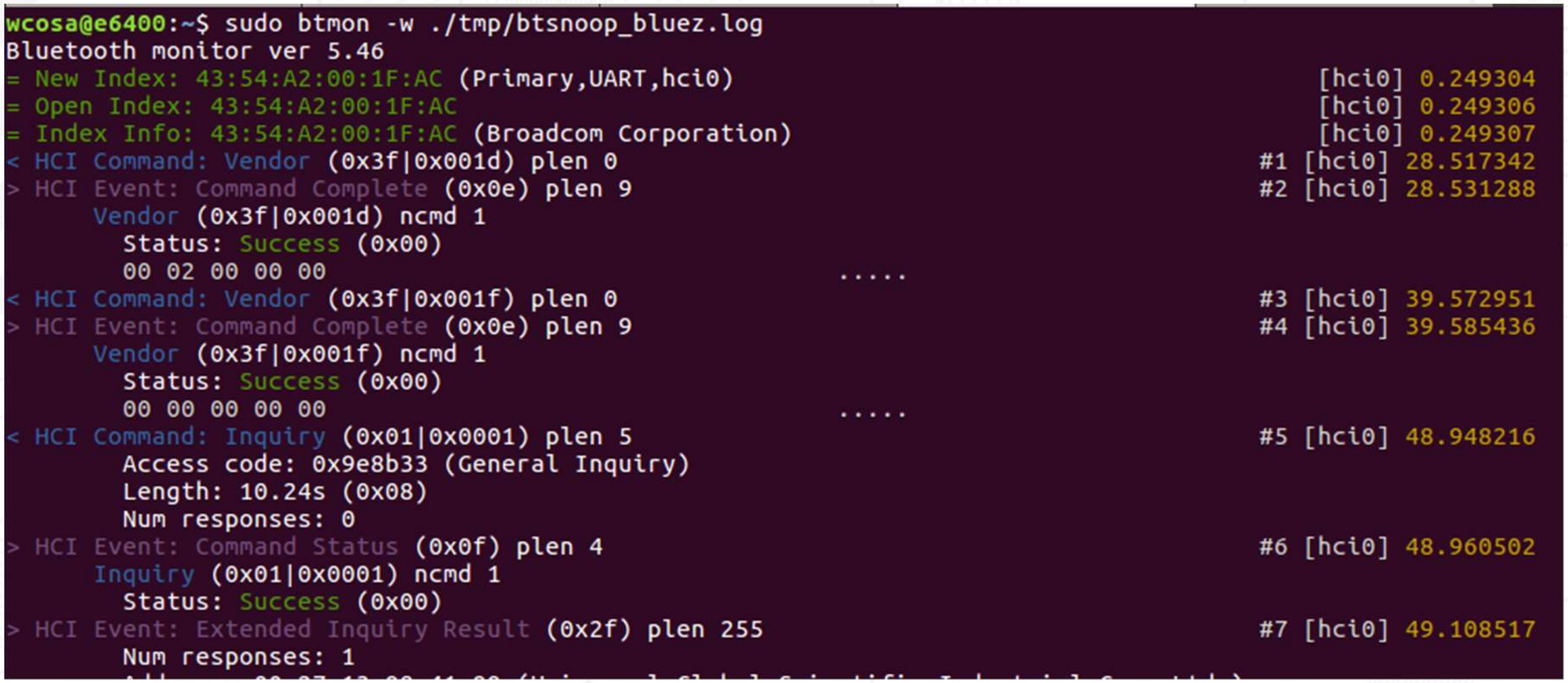
For the HCI Control and Baseband Commands, the OGF is defined as 0x03.

“ABCD”

* The OGF of 0x3F is reserved for vendor-specific commands.

Bluez HCI logging

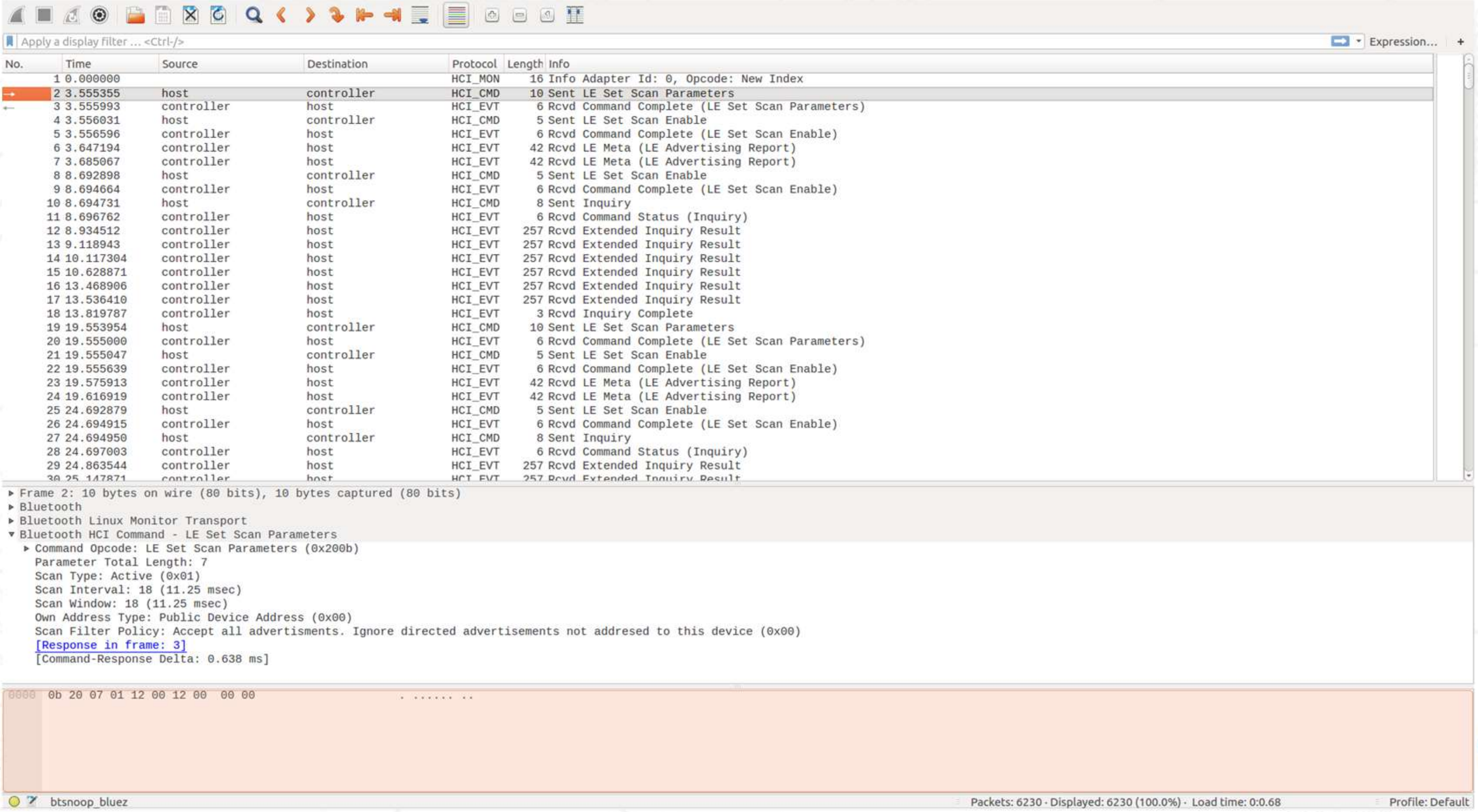
- *btmon -w ./tmp/snoop_bluez.log*
  - HCI packets display on console, and
  - snoop format output in background



```
wcosa@e6400:~$ sudo btmon -w ./tmp/btsnoop_bluez.log
Bluetooth monitor ver 5.46
= New Index: 43:54:A2:00:1F:AC (Primary,UART,hci0)                      [hci0] 0.249304
= Open Index: 43:54:A2:00:1F:AC                                         [hci0] 0.249306
= Index Info: 43:54:A2:00:1F:AC (Broadcom Corporation)                  [hci0] 0.249307
< HCI Command: Vendor (0x3f|0x001d) plen 0                           #1 [hci0] 28.517342
> HCI Event: Command Complete (0x0e) plen 9                          #2 [hci0] 28.531288
      Vendor (0x3f|0x001d) ncmd 1
        Status: Success (0x00)
        00 02 00 00 00                                    .....
< HCI Command: Vendor (0x3f|0x001f) plen 0                           #3 [hci0] 39.572951
> HCI Event: Command Complete (0x0e) plen 9                          #4 [hci0] 39.585436
      Vendor (0x3f|0x001f) ncmd 1
        Status: Success (0x00)
        00 00 00 00 00                                    .....
< HCI Command: Inquiry (0x01|0x0001) plen 5                          #5 [hci0] 48.948216
        Access code: 0x9e8b33 (General Inquiry)
        Length: 10.24s (0x08)
        Num responses: 0
> HCI Event: Command Status (0x0f) plen 4                            #6 [hci0] 48.960502
      Inquiry (0x01|0x0001) ncmd 1
        Status: Success (0x00)
> HCI Event: Extended Inquiry Result (0x2f) plen 255                 #7 [hci0] 49.108517
        Num responses: 1
```

Bluez HCI logging viewer

- To view the snoop file captured by *btmon* tool, you may
  - Use *btmon* tool to view, e.g. $***btmon -r ./tmp/snoop_bluez.log***
  - Use *Wireshark* to view



```
No.   Time        Source       Destination  Protocol  Length Info
    1 0.000000                              HCI_MON       16 Info Adapter Id: 0, Opcode: New Index
    2 3.555355    host         controller   HCI_CMD       10 Sent LE Set Scan Parameters
    3 3.555993    controller   host         HCI_EVT        6 Rcvd Command Complete (LE Set Scan Parameters)
    4 3.556031    host         controller   HCI_CMD        5 Sent LE Set Scan Enable
    5 3.556596    controller   host         HCI_EVT        6 Rcvd Command Complete (LE Set Scan Enable)
    6 3.647194    controller   host         HCI_EVT       42 Rcvd LE Meta (LE Advertising Report)
    7 3.685067    controller   host         HCI_EVT       42 Rcvd LE Meta (LE Advertising Report)
    8 8.692898    host         controller   HCI_CMD        5 Sent LE Set Scan Enable
    9 8.694664    controller   host         HCI_EVT        6 Rcvd Command Complete (LE Set Scan Enable)
   10 8.694731    host         controller   HCI_CMD        8 Sent Inquiry
   11 8.696762    controller   host         HCI_EVT        6 Rcvd Command Status (Inquiry)
   12 8.934512    controller   host         HCI_EVT      257 Rcvd Extended Inquiry Result
   13 9.118943    controller   host         HCI_EVT      257 Rcvd Extended Inquiry Result
   14 10.117304   controller   host         HCI_EVT      257 Rcvd Extended Inquiry Result
   15 10.628871   controller   host         HCI_EVT      257 Rcvd Extended Inquiry Result
   16 13.468906   controller   host         HCI_EVT      257 Rcvd Extended Inquiry Result
   17 13.536410   controller   host         HCI_EVT      257 Rcvd Extended Inquiry Result
   18 13.819787   controller   host         HCI_EVT        3 Rcvd Inquiry Complete
   19 19.553954   host         controller   HCI_CMD       10 Sent LE Set Scan Parameters
   20 19.555000   controller   host         HCI_EVT        6 Rcvd Command Complete (LE Set Scan Parameters)
   21 19.555047   host         controller   HCI_CMD        5 Sent LE Set Scan Enable
   22 19.555639   controller   host         HCI_EVT        6 Rcvd Command Complete (LE Set Scan Enable)
   23 19.575913   controller   host         HCI_EVT       42 Rcvd LE Meta (LE Advertising Report)
   24 19.616919   controller   host         HCI_EVT       42 Rcvd LE Meta (LE Advertising Report)
   25 24.692879   host         controller   HCI_CMD        5 Sent LE Set Scan Enable
   26 24.694915   controller   host         HCI_EVT        6 Rcvd Command Complete (LE Set Scan Enable)
   27 24.694950   host         controller   HCI_CMD        8 Sent Inquiry
   28 24.697003   controller   host         HCI_EVT        6 Rcvd Command Status (Inquiry)
   29 24.863544   controller   host         HCI_EVT      257 Rcvd Extended Inquiry Result
   30 25.147871   controller   host         HCI_EVT      257 Rcvd Extended Inquiry Result
```

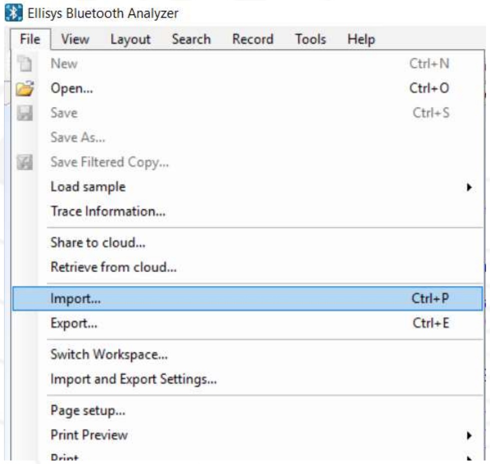
```
▸ Frame 2: 10 bytes on wire (80 bits), 10 bytes captured (80 bits)
▸ Bluetooth
▸ Bluetooth Linux Monitor Transport
▾ Bluetooth HCI Command - LE Set Scan Parameters
  ▸ Command Opcode: LE Set Scan Parameters (0x200b)
    Parameter Total Length: 7
    Scan Type: Active (0x01)
    Scan Interval: 18 (11.25 msec)
    Scan Window: 18 (11.25 msec)
    Own Address Type: Public Device Address (0x00)
    Scan Filter Policy: Accept all advertisments. Ignore directed advertisements not addresed to this device (0x00)
    [Response in frame: 3]
    [Command-Response Delta: 0.638 ms]
```

```
0000  0b 20 07 01 12 00 12 00  00 00                     . ....... ..
```

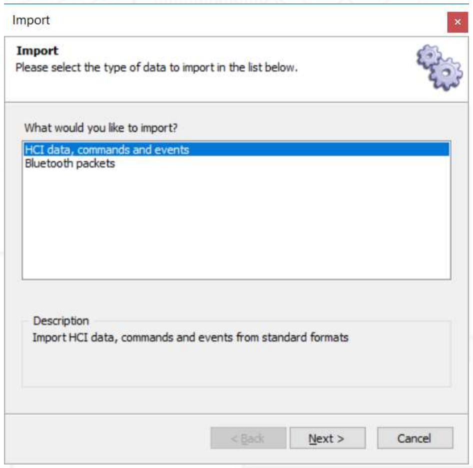
```
btsnoop_bluez        Packets: 6230 · Displayed: 6230 (100.0%) · Load time: 0:0.68        Profile: Default
```

Bluez HCI logging viewer (continue)

- Import content of *btmon* captured file into *Ellisys Bluetooth Analyzer* tool to view



Step-1. **File –> Import…**


Import

Import

Please select the type of data to import in the list below.

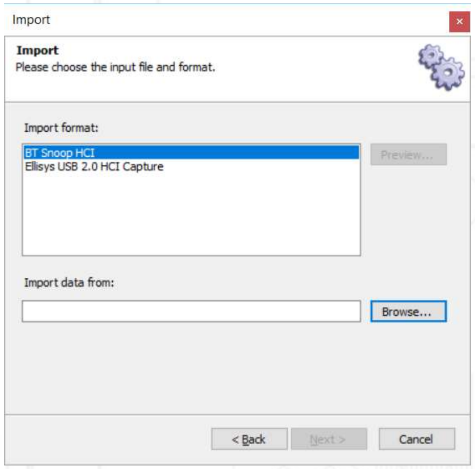
What would you like to import?

- HCI data, commands and events
- Bluetooth packets

Description

Import HCI data, commands and events from standard formats

Step-2. **HCI data, commands and events**


Import

Import

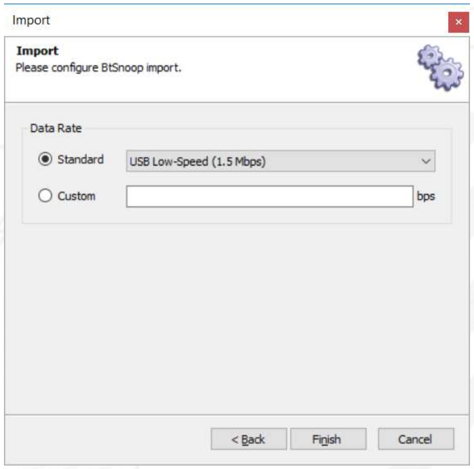
Please choose the input file and format.

Import format:

- BT Snoop HCI
- Ellisys USB 2.0 HCI Capture

Import data from:

Step-3. **BT Snoop HCI**


Import

Import

Please configure BtSnoop import.

Data Rate

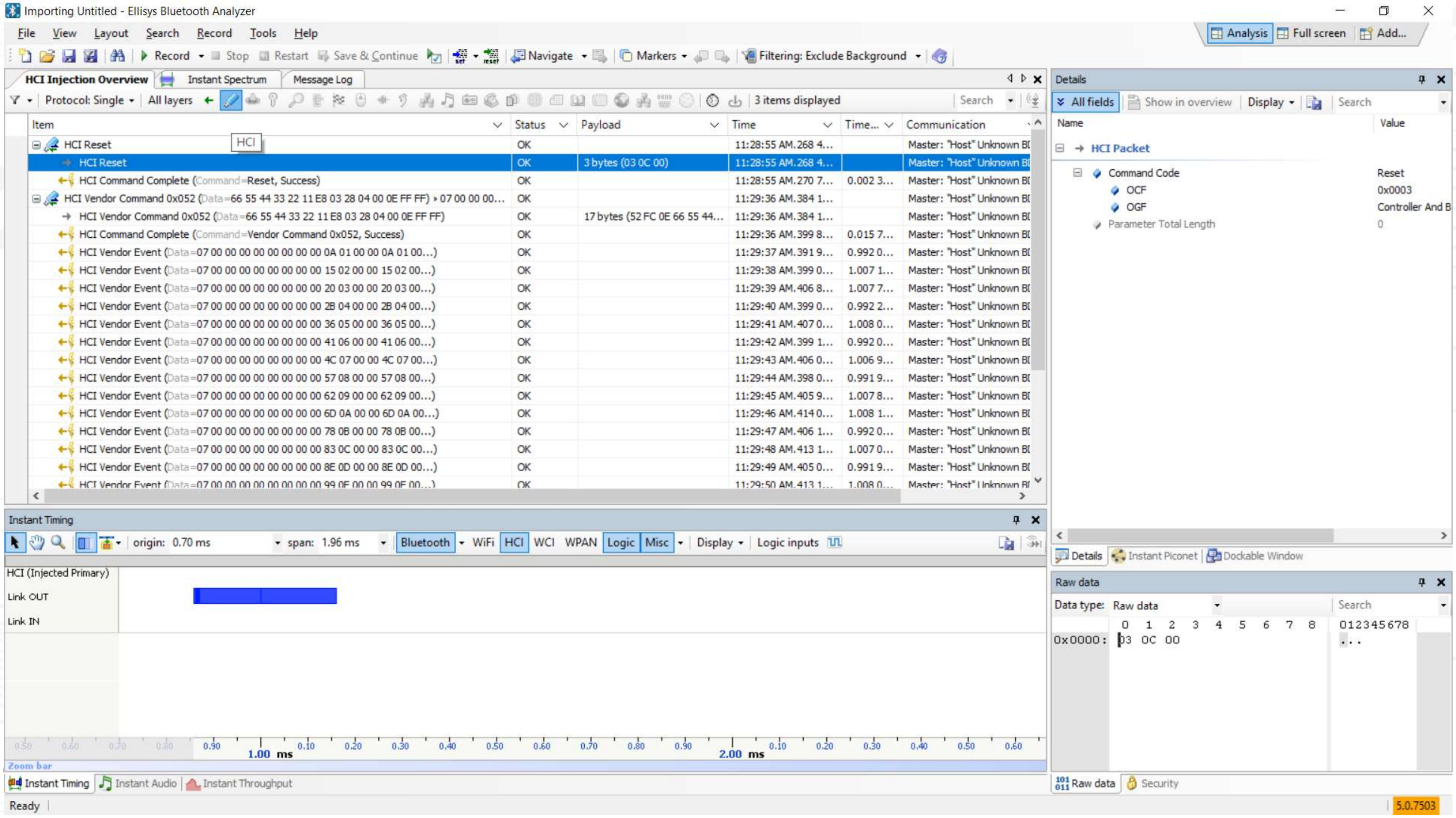
● Standard USB Low-Speed (1.5 Mbps)

○ Custom bps

Step-4. **Standard and** any speed

Bluez HCI logging viewer (continue)

Step-5. **HCI Injection Overview –> HCI**



Importing Untitled - Ellisys Bluetooth Analyzer

HCI Injection Overview | Protocol: Single | All layers | 3 items displayed

Item	Status	Payload	Time	Time...	Communication
HCI Reset	OK		11:28:55 AM.268 4...		Master: "Host" Unknown BI
HCI Reset	OK	3 bytes (03 0C 00)	11:28:55 AM.268 4...		Master: "Host" Unknown BI
HCI Command Complete (Command=Reset, Success)	OK		11:28:55 AM.270 7...	0.002 3...	Master: "Host" Unknown BI
HCI Vendor Command 0x052 (Data=66 55 44 33 22 11 E8 03 28 04 00 0E FF FF) › 07 00 00 00...	OK		11:29:36 AM.384 1...		Master: "Host" Unknown BI
HCI Vendor Command 0x052 (Data=66 55 44 33 22 11 E8 03 28 04 00 0E FF FF)	OK	17 bytes (52 FC 0E 66 55 44...	11:29:36 AM.384 1...		Master: "Host" Unknown BI
HCI Command Complete (Command=Vendor Command 0x052, Success)	OK		11:29:36 AM.399 8...	0.015 7...	Master: "Host" Unknown BI
HCI Vendor Event (Data=07 00 00 00 00 00 00 00 00 0A 01 00 00 0A 01 00...)	OK		11:29:37 AM.391 9...	0.992 0...	Master: "Host" Unknown BI
HCI Vendor Event (Data=07 00 00 00 00 00 00 00 00 15 02 00 00 15 02 00...)	OK		11:29:38 AM.399 0...	1.007 1...	Master: "Host" Unknown BI
HCI Vendor Event (Data=07 00 00 00 00 00 00 00 00 20 03 00 00 20 03 00...)	OK		11:29:39 AM.406 8...	1.007 7...	Master: "Host" Unknown BI
HCI Vendor Event (Data=07 00 00 00 00 00 00 00 00 2B 04 00 00 2B 04 00...)	OK		11:29:40 AM.399 0...	0.992 2...	Master: "Host" Unknown BI
HCI Vendor Event (Data=07 00 00 00 00 00 00 00 00 36 05 00 00 36 05 00...)	OK		11:29:41 AM.407 0...	1.008 0...	Master: "Host" Unknown BI
HCI Vendor Event (Data=07 00 00 00 00 00 00 00 00 41 06 00 00 41 06 00...)	OK		11:29:42 AM.399 1...	0.992 0...	Master: "Host" Unknown BI
HCI Vendor Event (Data=07 00 00 00 00 00 00 00 00 4C 07 00 00 4C 07 00...)	OK		11:29:43 AM.406 0...	1.006 9...	Master: "Host" Unknown BI
HCI Vendor Event (Data=07 00 00 00 00 00 00 00 00 57 08 00 00 57 08 00...)	OK		11:29:44 AM.398 0...	0.991 9...	Master: "Host" Unknown BI
HCI Vendor Event (Data=07 00 00 00 00 00 00 00 00 62 09 00 00 62 09 00...)	OK		11:29:45 AM.405 9...	1.007 8...	Master: "Host" Unknown BI
HCI Vendor Event (Data=07 00 00 00 00 00 00 00 00 6D 0A 00 00 6D 0A 00...)	OK		11:29:46 AM.414 0...	1.008 1...	Master: "Host" Unknown BI
HCI Vendor Event (Data=07 00 00 00 00 00 00 00 00 78 0B 00 00 78 0B 00...)	OK		11:29:47 AM.406 1...	0.992 0...	Master: "Host" Unknown BI
HCI Vendor Event (Data=07 00 00 00 00 00 00 00 00 83 0C 00 00 83 0C 00...)	OK		11:29:48 AM.413 1...	1.007 0...	Master: "Host" Unknown BI
HCI Vendor Event (Data=07 00 00 00 00 00 00 00 00 8E 0D 00 00 8E 0D 00...)	OK		11:29:49 AM.405 0...	0.991 9...	Master: "Host" Unknown BI
HCI Vendor Event (Data=07 00 00 00 00 00 00 00 00 99 0E 00 00 99 0E 00...)	OK		11:29:50 AM.413 1...	1.008 0...	Master: "Host" Unknown BI

Details

Name	Value
HCI Packet	
Command Code	Reset
OCF	0x0003
OGF	Controller And B
Parameter Total Length	0

Instant Timing — origin: 0.70 ms, span: 1.96 ms

HCI (Injected Primary)
Link OUT
Link IN

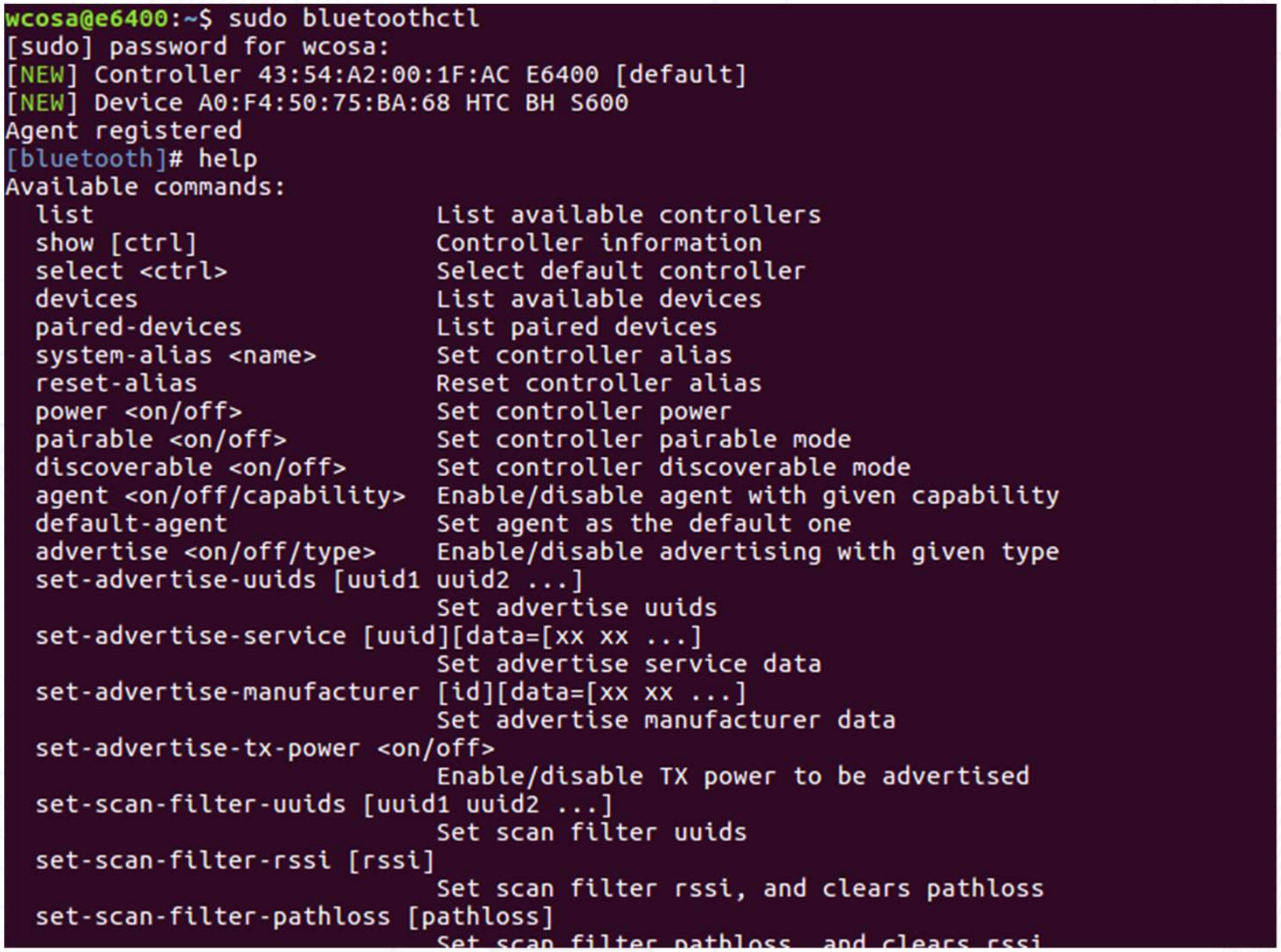
Raw data — Data type: Raw data

```
        0  1  2  3  4  5  6  7  8   012345678
0x0000: 03 0C 00                    ...
```

Ready | 5.0.7503

Control Tool

bluetoothctl

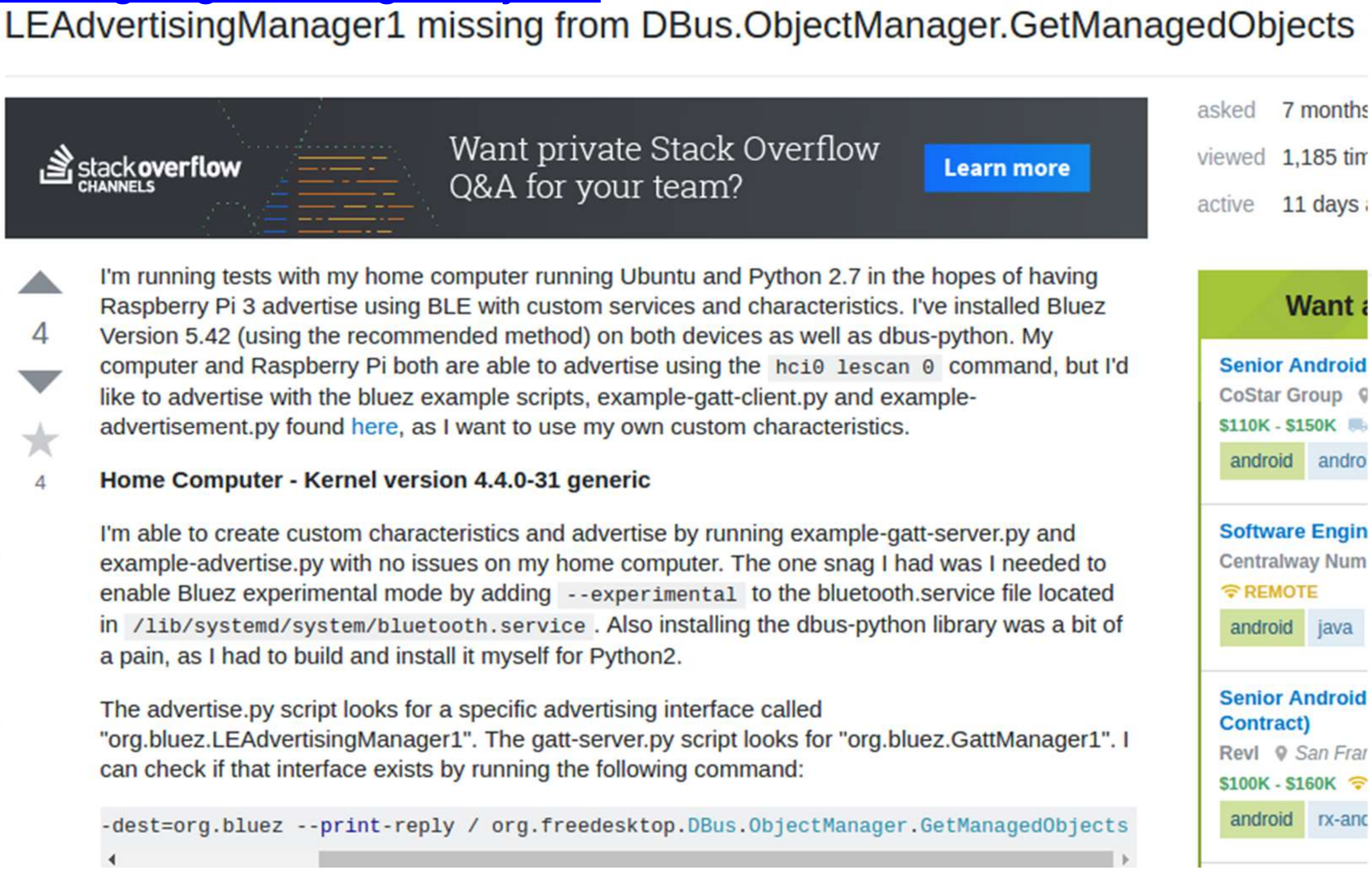


```
wcosa@e6400:~$ sudo bluetoothctl
[sudo] password for wcosa:
[NEW] Controller 43:54:A2:00:1F:AC E6400 [default]
[NEW] Device A0:F4:50:75:BA:68 HTC BH S600
Agent registered
[bluetooth]# help
Available commands:
  list                       List available controllers
  show [ctrl]                Controller information
  select <ctrl>              Select default controller
  devices                    List available devices
  paired-devices             List paired devices
  system-alias <name>        Set controller alias
  reset-alias                Reset controller alias
  power <on/off>             Set controller power
  pairable <on/off>          Set controller pairable mode
  discoverable <on/off>      Set controller discoverable mode
  agent <on/off/capability>  Enable/disable agent with given capability
  default-agent              Set agent as the default one
  advertise <on/off/type>    Enable/disable advertising with given type
  set-advertise-uuids [uuid1 uuid2 ...]
                             Set advertise uuids
  set-advertise-service [uuid][data=[xx xx ...]
                             Set advertise service data
  set-advertise-manufacturer [id][data=[xx xx ...]
                             Set advertise manufacturer data
  set-advertise-tx-power <on/off>
                             Enable/disable TX power to be advertised
  set-scan-filter-uuids [uuid1 uuid2 ...]
                             Set scan filter uuids
  set-scan-filter-rssi [rssi]
                             Set scan filter rssi, and clears pathloss
  set-scan-filter-pathloss [pathloss]
                             Set scan filter pathloss, and clears rssi
```

Enable Experimental Features

Error – “LEAdvertisingManager not found”

- https://stackoverflow.com/questions/41351514/leadvertisingmanager1-missing-from-dbus-objectmanager-getmanagedobjects



LEAdvertisingManager1 missing from DBus.ObjectManager.GetManagedObjects

I'm running tests with my home computer running Ubuntu and Python 2.7 in the hopes of having Raspberry Pi 3 advertise using BLE with custom services and characteristics. I've installed Bluez Version 5.42 (using the recommended method) on both devices as well as dbus-python. My computer and Raspberry Pi both are able to advertise using the `hci0 lescan 0` command, but I'd like to advertise with the bluez example scripts, example-gatt-client.py and example-advertisement.py found here, as I want to use my own custom characteristics.

Home Computer - Kernel version 4.4.0-31 generic

I'm able to create custom characteristics and advertise by running example-gatt-server.py and example-advertise.py with no issues on my home computer. The one snag I had was I needed to enable Bluez experimental mode by adding `--experimental` to the bluetooth.service file located in `/lib/systemd/system/bluetooth.service`. Also installing the dbus-python library was a bit of a pain, as I had to build and install it myself for Python2.

The advertise.py script looks for a specific advertising interface called "org.bluez.LEAdvertisingManager1". The gatt-server.py script looks for "org.bluez.GattManager1". I can check if that interface exists by running the following command:

```
-dest=org.bluez --print-reply / org.freedesktop.DBus.ObjectManager.GetManagedObjects
```

Cheat-sheet

- Check Bluetooth service running status
  - *systemctl status bluetooth*
- List loaded org.bluez D-Bus interfaces
  - *dbus-send --system --dest=org.bluez --print-reply / org.freedesktop.DBus.ObjectManager.GetManagedObjects*
- Enable experimental features in /lib/systemd/system/bluetooth.service
  - ExecStart=/usr/local/libexec/bluetooth/bluetoothd --experimental
- Restart Bluetooth service
  - *sudo systemctl daemon-reload*
  - *sudo systemctl restart bluetooth*

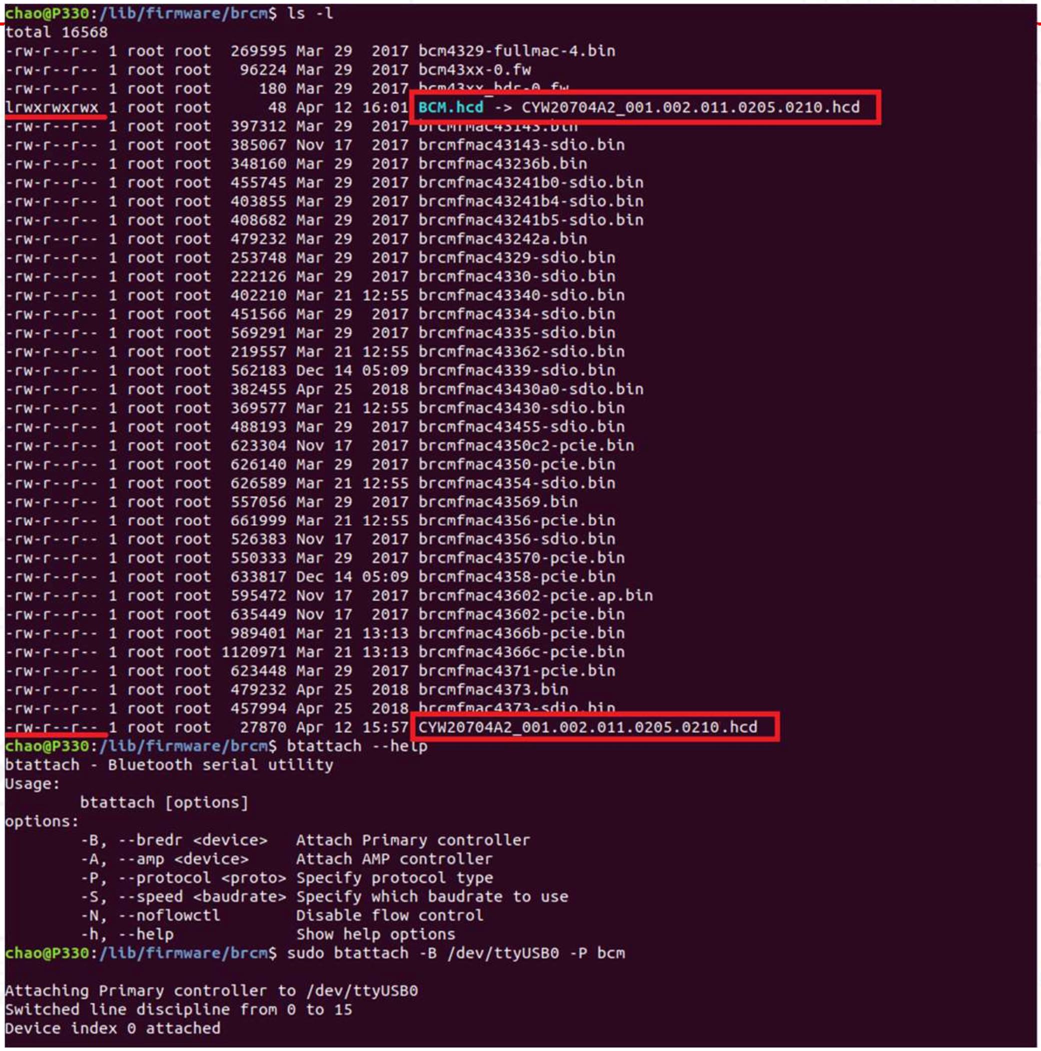
Supplement

Use ‘btattach’ on UART Transport

- Similar to ‘hciattach’, ‘btattach’ is also able to initialize firmware download and apply Bluez HCI protocol on the specified UART port.
- ‘btattach’
  - ‘btattach’ calls (relies on) Bluez kernel driver to conduct firmware download process.
  - Bluez uses hard coded firmware file */lib/firmware/brcm/BCM.hcd* (don’t confuse it with the filename used in USB case, refer to previous slide for details).
  - Command example -- $*sudo btattach -B /dev/ttyUSB0 -P bcm*
- ‘hciattach’
  - Instead of calling Bluez kernel driver to download firmware, ‘hciattach’ implements the whole firmware download procedure in user-space codes.
  - ‘hciattach’ looks for firmware file in */etc/firmware/* folder based on the local device name returned from HCI_Read_Local_Name (OGF 0x03, OCF 0x0014) command.
  - Command example -- $*sudo hciattach -n –p /dev/ttyUSB0 bcm43xx*

Example: ‘btattach’ over UART

- Create a soft link in the /lib/firmware/brcm/ folder to link the ‘BCM.hcd’ name to the desired firmware file.
- Run btattach by specifying UART port and firmware download protocol.



```
chao@P330:/lib/firmware/brcm$ ls -l
total 16568
-rw-r--r-- 1 root root  269595 Mar 29  2017 bcm4329-fullmac-4.bin
-rw-r--r-- 1 root root   96224 Mar 29  2017 bcm43xx-0.fw
-rw-r--r-- 1 root root     180 Mar 29  2017 bcm43xx_hdr-0.fw
lrwxrwxrwx 1 root root      48 Apr 12 16:01 BCM.hcd -> CYW20704A2_001.002.011.0205.0210.hcd
-rw-r--r-- 1 root root  397312 Mar 29  2017 brcmfmac43143.bin
-rw-r--r-- 1 root root  385067 Nov 17  2017 brcmfmac43143-sdio.bin
-rw-r--r-- 1 root root  348160 Mar 29  2017 brcmfmac43236b.bin
-rw-r--r-- 1 root root  455745 Mar 29  2017 brcmfmac43241b0-sdio.bin
-rw-r--r-- 1 root root  403855 Mar 29  2017 brcmfmac43241b4-sdio.bin
-rw-r--r-- 1 root root  408682 Mar 29  2017 brcmfmac43241b5-sdio.bin
-rw-r--r-- 1 root root  479232 Mar 29  2017 brcmfmac43242a.bin
-rw-r--r-- 1 root root  253748 Mar 29  2017 brcmfmac4329-sdio.bin
-rw-r--r-- 1 root root  222126 Mar 29  2017 brcmfmac4330-sdio.bin
-rw-r--r-- 1 root root  402210 Mar 21 12:55 brcmfmac43340-sdio.bin
-rw-r--r-- 1 root root  451566 Mar 29  2017 brcmfmac4334-sdio.bin
-rw-r--r-- 1 root root  569291 Mar 29  2017 brcmfmac4335-sdio.bin
-rw-r--r-- 1 root root  219557 Mar 21 12:55 brcmfmac43362-sdio.bin
-rw-r--r-- 1 root root  562183 Dec 14 05:09 brcmfmac4339-sdio.bin
-rw-r--r-- 1 root root  382455 Apr 25  2018 brcmfmac43430a0-sdio.bin
-rw-r--r-- 1 root root  369577 Mar 21 12:55 brcmfmac43430-sdio.bin
-rw-r--r-- 1 root root  488193 Mar 29  2017 brcmfmac43455-sdio.bin
-rw-r--r-- 1 root root  623304 Nov 17  2017 brcmfmac4350c2-pcie.bin
-rw-r--r-- 1 root root  626140 Mar 29  2017 brcmfmac4350-pcie.bin
-rw-r--r-- 1 root root  626589 Mar 21 12:55 brcmfmac4354-sdio.bin
-rw-r--r-- 1 root root  557056 Mar 29  2017 brcmfmac43569.bin
-rw-r--r-- 1 root root  661999 Mar 21 12:55 brcmfmac4356-pcie.bin
-rw-r--r-- 1 root root  526383 Nov 17  2017 brcmfmac4356-sdio.bin
-rw-r--r-- 1 root root  550333 Mar 29  2017 brcmfmac43570-pcie.bin
-rw-r--r-- 1 root root  633817 Dec 14 05:09 brcmfmac4358-pcie.bin
-rw-r--r-- 1 root root  595472 Nov 17  2017 brcmfmac43602-pcie.ap.bin
-rw-r--r-- 1 root root  635449 Nov 17  2017 brcmfmac43602-pcie.bin
-rw-r--r-- 1 root root  989401 Mar 21 13:13 brcmfmac4366b-pcie.bin
-rw-r--r-- 1 root root 1120971 Mar 21 13:13 brcmfmac4366c-pcie.bin
-rw-r--r-- 1 root root  623448 Mar 29  2017 brcmfmac4371-pcie.bin
-rw-r--r-- 1 root root  479232 Apr 25  2018 brcmfmac4373.bin
-rw-r--r-- 1 root root  457994 Apr 25  2018 brcmfmac4373-sdio.bin
-rw-r--r-- 1 root root   27870 Apr 12 15:57 CYW20704A2_001.002.011.0205.0210.hcd
chao@P330:/lib/firmware/brcm$ btattach --help
btattach - Bluetooth serial utility
Usage:
        btattach [options]
options:
        -B, --bredr <device>   Attach Primary controller
        -A, --amp <device>     Attach AMP controller
        -P, --protocol <proto> Specify protocol type
        -S, --speed <baudrate> Specify which baudrate to use
        -N, --noflowctl        Disable flow control
        -h, --help             Show help options
chao@P330:/lib/firmware/brcm$ sudo btattach -B /dev/ttyUSB0 -P bcm

Attaching Primary controller to /dev/ttyUSB0
Switched line discipline from 0 to 15
Device index 0 attached
```

USB

CYPRESS

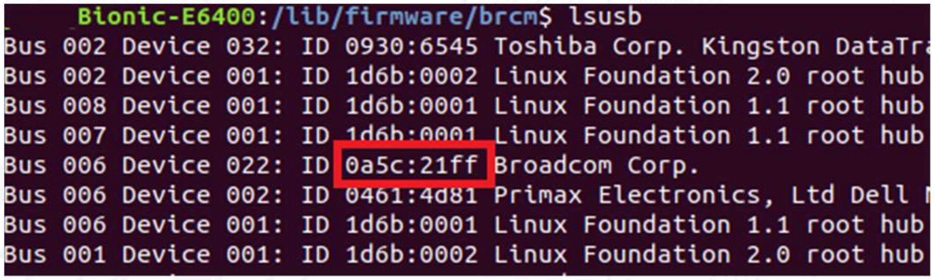
EMBEDDED IN TOMORROW™

Firmware Download via Bluez “btusb” Driver

- Bluez “btusb” USB driver will look for a specific BCM-<vid>-<pid>.hcd firmware file in the /lib/firmware/brcm/ folder and download it, once the driver detects a plugging-in USB Bluetooth dongle with Broadcom VID 0x0a5c.
  - e.g. BCM20704 dongle advertises VID=0x0a5c/PID=0x21ff
  - Looks for /lib/firmware/brcm/BCM-0a5c-21ff.hcd file
- WARNING ! To work with Bluez USB driver, the firmware .hcd must be built with “**NO USB Re-enumeration**” feature.

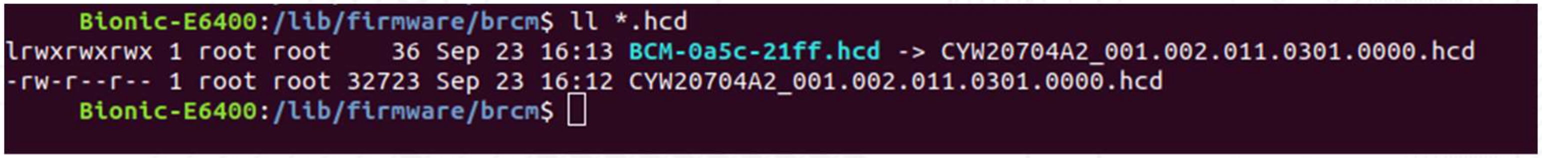
Example: “btusb” Firmware Download

- Identify the VID & PID of Cypress (Broadcom) Bluetooth Controller,



```
_    Bionic-E6400:/lib/firmware/brcm$ lsusb
Bus 002 Device 032: ID 0930:6545 Toshiba Corp. Kingston DataTr
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 008 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 007 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 006 Device 022: ID 0a5c:21ff Broadcom Corp.
Bus 006 Device 002: ID 0461:4d81 Primax Electronics, Ltd Dell
Bus 006 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

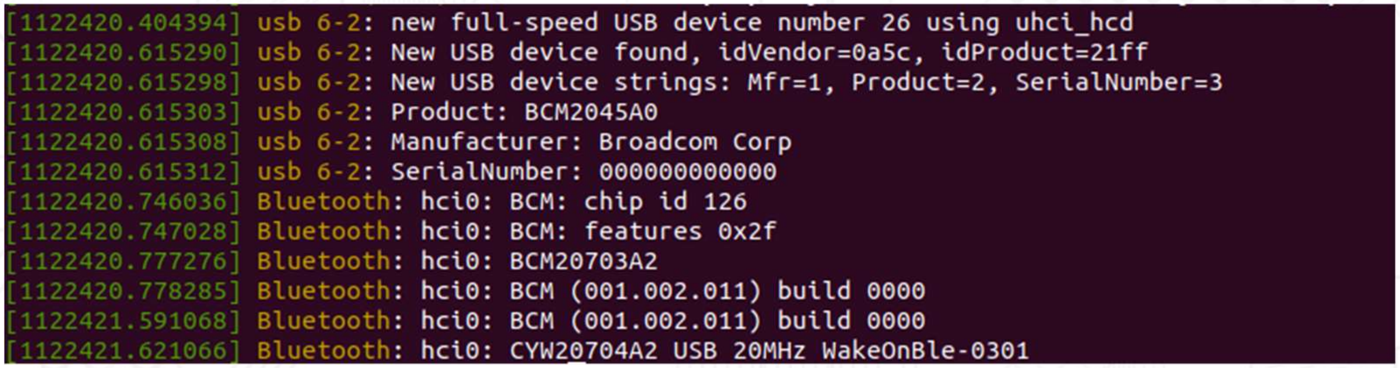
- Create a soft link in the /lib/firmware/brcm/ folder to link the ‘BCM-<vid>-<pid>.hcd’ name to the desired firmware file. For example,



```
     Bionic-E6400:/lib/firmware/brcm$ ll *.hcd
lrwxrwxrwx 1 root root    36 Sep 23 16:13 BCM-0a5c-21ff.hcd -> CYW20704A2_001.002.011.0301.0000.hcd
-rw-r--r-- 1 root root 32723 Sep 23 16:12 CYW20704A2_001.002.011.0301.0000.hcd
     Bionic-E6400:/lib/firmware/brcm$ 
```

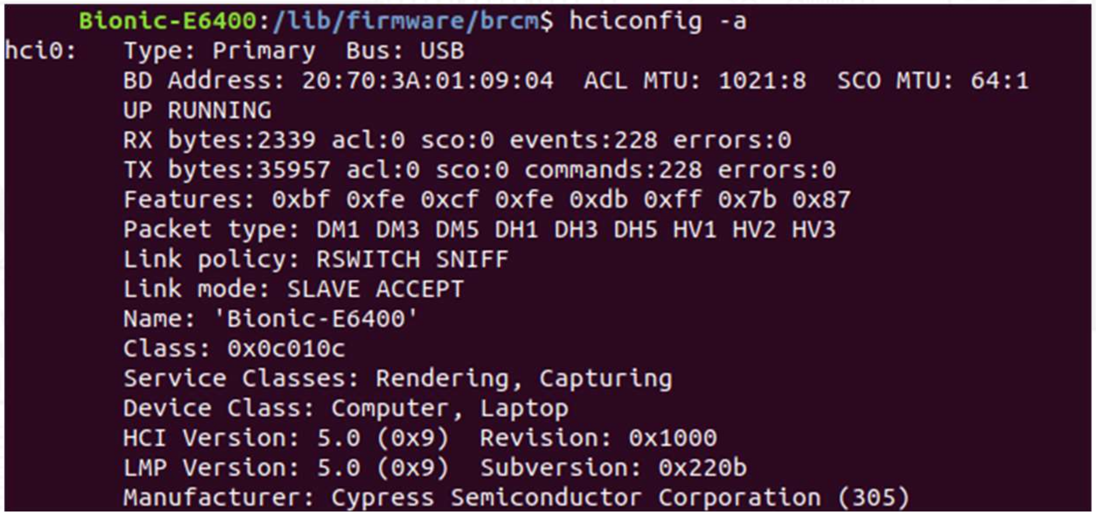
Example: “btusb” Firmware Download (continue)

- The `*dmesg*` example log after CYW20704 was successfully loaded with “BCM20703A2_001.002.011.0301.0000.hcd” firmware,



```
[1122420.404394] usb 6-2: new full-speed USB device number 26 using uhci_hcd
[1122420.615290] usb 6-2: New USB device found, idVendor=0a5c, idProduct=21ff
[1122420.615298] usb 6-2: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[1122420.615303] usb 6-2: Product: BCM2045A0
[1122420.615308] usb 6-2: Manufacturer: Broadcom Corp
[1122420.615312] usb 6-2: SerialNumber: 000000000000
[1122420.746036] Bluetooth: hci0: BCM: chip id 126
[1122420.747028] Bluetooth: hci0: BCM: features 0x2f
[1122420.777276] Bluetooth: hci0: BCM20703A2
[1122420.778285] Bluetooth: hci0: BCM (001.002.011) build 0000
[1122421.591068] Bluetooth: hci0: BCM (001.002.011) build 0000
[1122421.621066] Bluetooth: hci0: CYW20704A2 USB 20MHz WakeOnBle-0301
```

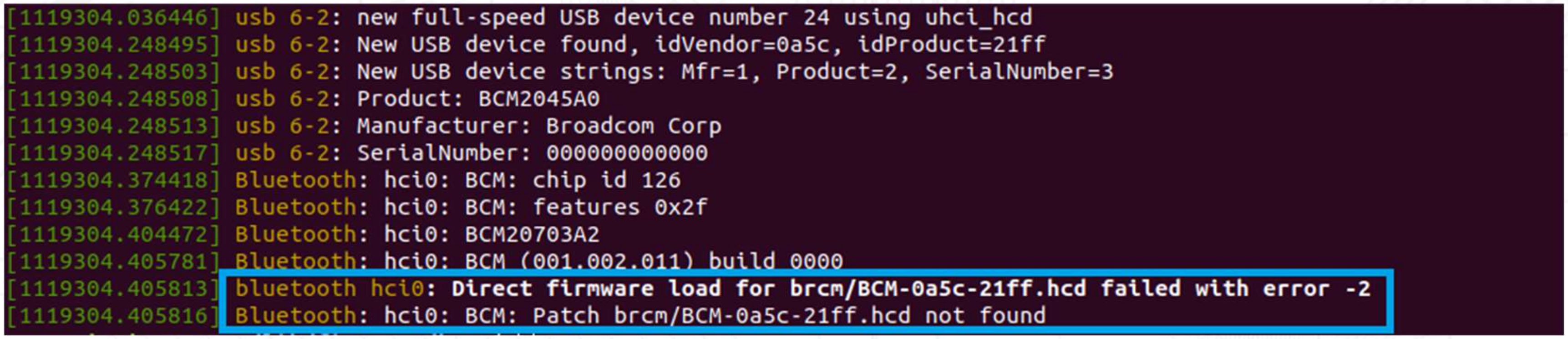
- The result of `*hciconfig -a*`,



```
        Bionic-E6400:/lib/firmware/brcm$ hciconfig -a
hci0:   Type: Primary  Bus: USB
        BD Address: 20:70:3A:01:09:04  ACL MTU: 1021:8  SCO MTU: 64:1
        UP RUNNING
        RX bytes:2339 acl:0 sco:0 events:228 errors:0
        TX bytes:35957 acl:0 sco:0 commands:228 errors:0
        Features: 0xbf 0xfe 0xcf 0xfe 0xdb 0xff 0x7b 0x87
        Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
        Link policy: RSWITCH SNIFF
        Link mode: SLAVE ACCEPT
        Name: 'Bionic-E6400'
        Class: 0x0c010c
        Service Classes: Rendering, Capturing
        Device Class: Computer, Laptop
        HCI Version: 5.0 (0x9)  Revision: 0x1000
        LMP Version: 5.0 (0x9)  Subversion: 0x220b
        Manufacturer: Cypress Semiconductor Corporation (305)
```

“btusb” Initialization without Firmware Download

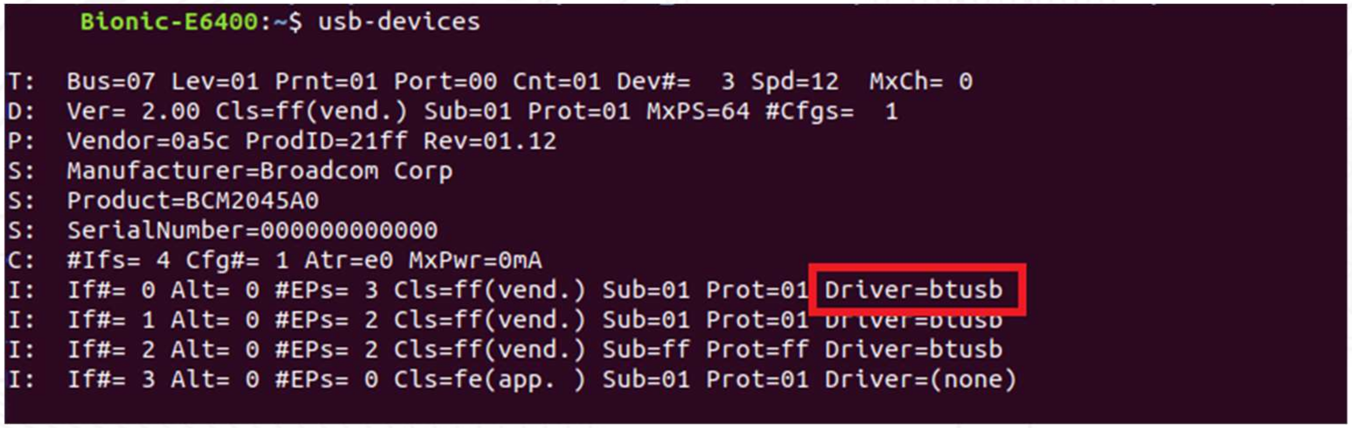
- Cypress Bluetooth Controllers come with default ROM firmware which will still be functioning on most basic Bluetooth operations.
- The `*dmesg*` example log when Bluez USB driver couldn’t find a matching BCM-<vid>-<pid>.hcd firmware file in the /lib/firmware/brcm/ folder,



```
[1119304.036446] usb 6-2: new full-speed USB device number 24 using uhci_hcd
[1119304.248495] usb 6-2: New USB device found, idVendor=0a5c, idProduct=21ff
[1119304.248503] usb 6-2: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[1119304.248508] usb 6-2: Product: BCM2045A0
[1119304.248513] usb 6-2: Manufacturer: Broadcom Corp
[1119304.248517] usb 6-2: SerialNumber: 000000000000
[1119304.374418] Bluetooth: hci0: BCM: chip id 126
[1119304.376422] Bluetooth: hci0: BCM: features 0x2f
[1119304.404472] Bluetooth: hci0: BCM20703A2
[1119304.405781] Bluetooth: hci0: BCM (001.002.011) build 0000
[1119304.405813] bluetooth hci0: Direct firmware load for brcm/BCM-0a5c-21ff.hcd failed with error -2
[1119304.405816] Bluetooth: hci0: BCM: Patch brcm/BCM-0a5c-21ff.hcd not found
```

Check USB device details

- \`cat\` the /sys/kernel/debug/usb/devices† (/proc/bus/usb/devices), if available, for USB device details.
  - Make sure system loads “**btusb**” USB driver to serve the plugged-in Bluetooth device.
    - †Need “root” permission to access. Try mounting the debug filesystem (\`mount -t debugfs none /sys/kernel/debug\`) if it isn’t available.
    - On Ubuntu, \`*usb-devices*\` can be used to display USB devices details.



```
Bionic-E6400:~$ usb-devices

T:  Bus=07 Lev=01 Prnt=01 Port=00 Cnt=01 Dev#=  3 Spd=12  MxCh= 0
D:  Ver= 2.00 Cls=ff(vend.) Sub=01 Prot=01 MxPS=64 #Cfgs=  1
P:  Vendor=0a5c ProdID=21ff Rev=01.12
S:  Manufacturer=Broadcom Corp
S:  Product=BCM2045A0
S:  SerialNumber=000000000000
C:  #Ifs= 4 Cfg#= 1 Atr=e0 MxPwr=0mA
I:  If#= 0 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=01 Prot=01 Driver=btusb
I:  If#= 1 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=01 Prot=01 Driver=btusb
I:  If#= 2 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=ff Prot=ff Driver=btusb
I:  If#= 3 Alt= 0 #EPs= 0 Cls=fe(app. ) Sub=01 Prot=01 Driver=(none)
```

CYPRESS®

EMBEDDED IN TOMORROW™