Assignments T7 and T8

DGMD E-14

Wearable Devices and Computer Vision

Professor Jose Luis Ramirez Herran

October 12, 2021

Austin High, Ping Ji, Yusheng Jiang, Blake Graham

_____**Tutorial 7**_____

Part 1. Please follow the tutorial 7 (https://drive.google.com/file/d/1IQ7VI9qYdqsyPQAd-_jTx7GHQTQbHwRy/view)

Answer in the context of BlueZ Gatttool:

1. What is a **handle** and give an example of one of them.

The handle is a 16 bit address that acts as a unique identifier for a UUID

Example: Attr handle: 0x001

2. What is a **characteristic value** and give an example of one of them.

A characteristic is a basic data element used to construct a GATT service

Example: handle: 0x0011, char properties: 0x02, char value handle: 0x0012, uuid: 00002a23-0000-1000-8000-00805f9b34fb

3. Create your own versions of the figures 1 to 23.

Figure 1: BLE debug interface via USB streaming.

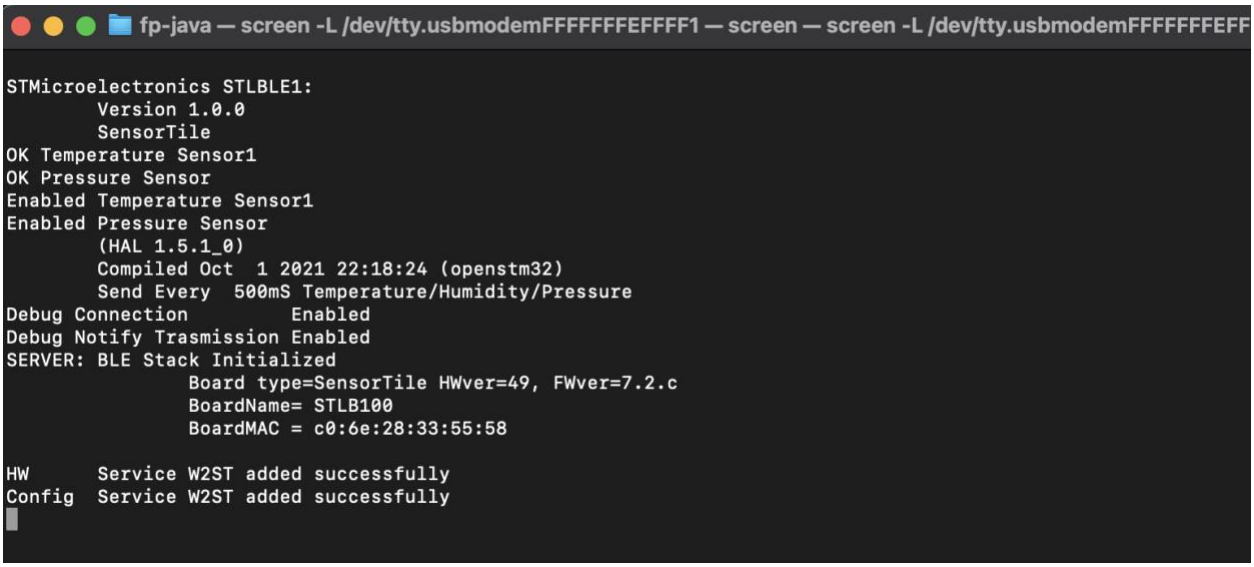


Figure 2: Use bluetootctl in BeagleBone

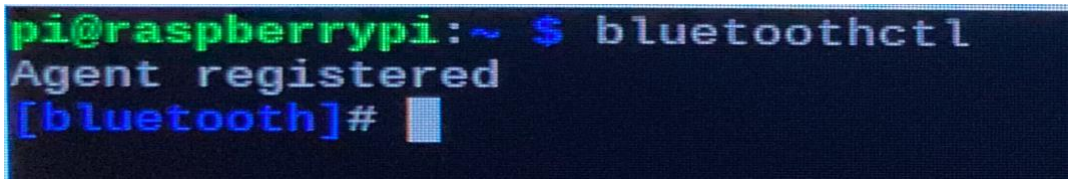


Figure 3: Use bluetoothctl to check if the SensorTile is paired and disconnected

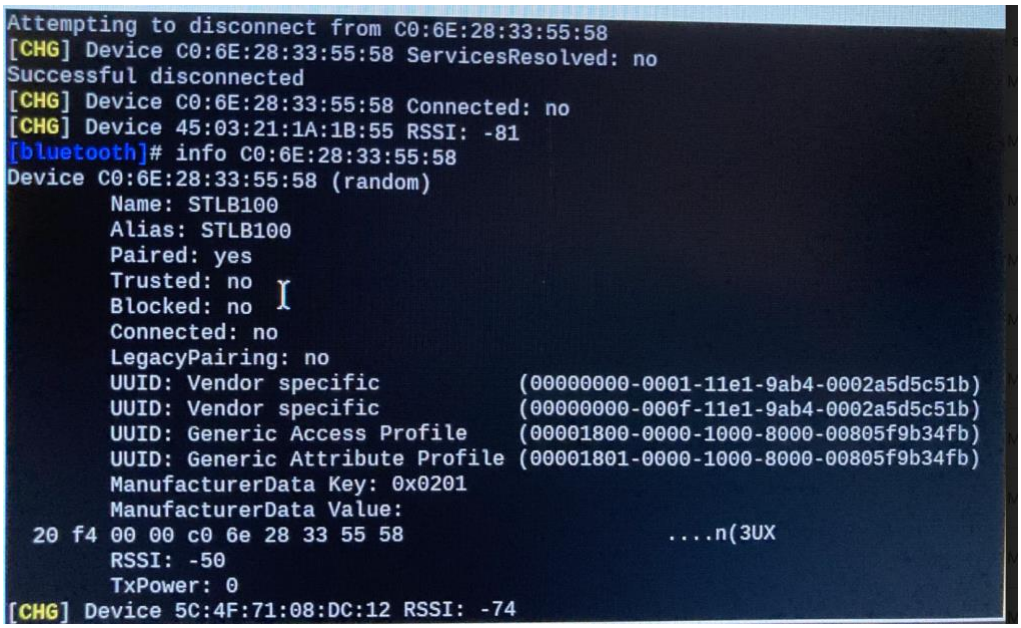


Figure 4: Exit bluetoothctl

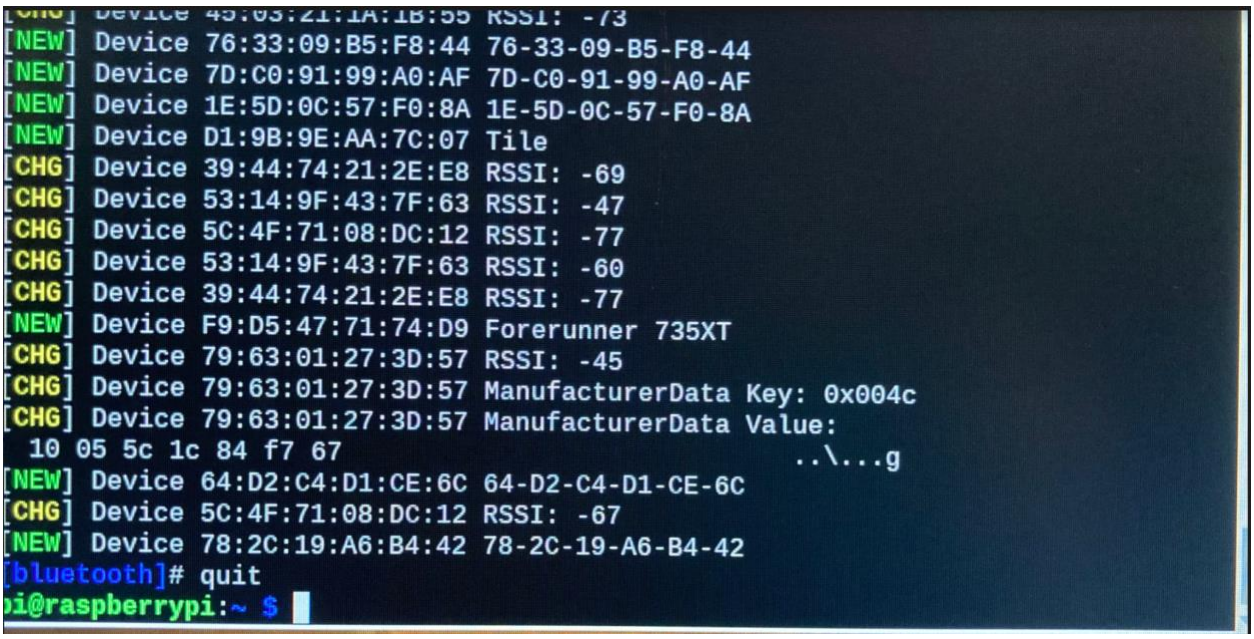


Figure 5: Use gatttool with proper options in interactive mode

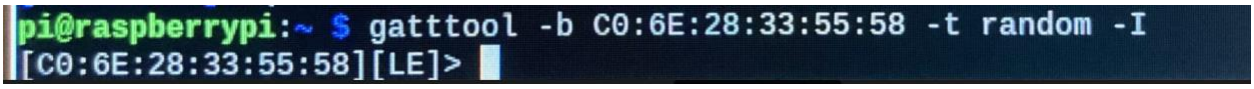


Figure 6: Connect with SensorTile in gatttool interactive mode

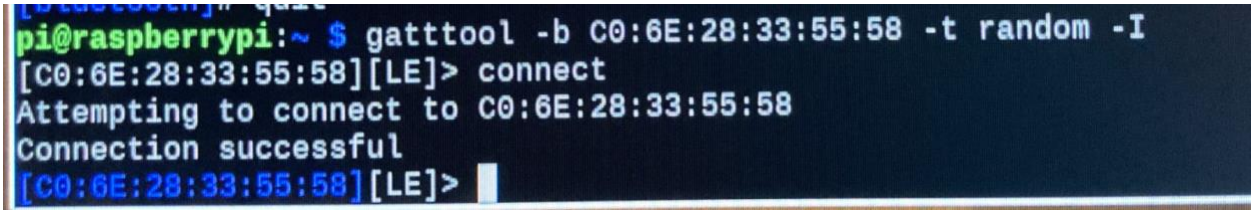


Figure 7: BLE_SampleApp USB debug interface for successful connection

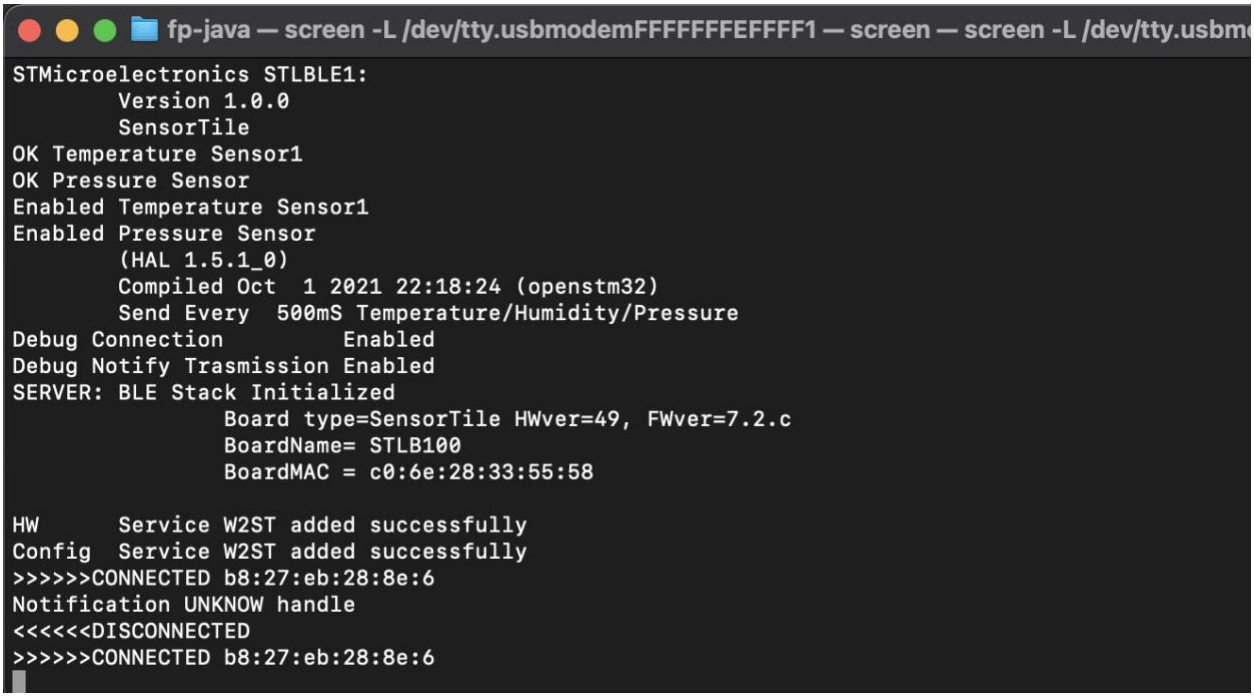


Figure 8: GATT primary services discovery in gatttool

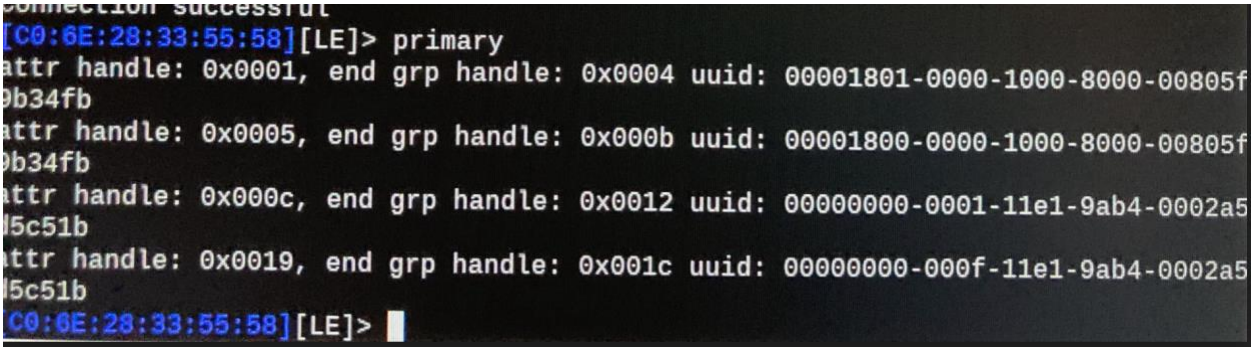


Figure 10: GATT characteristic descriptors discovery in gatttool

```
C0:6E:28:33:55:58][LE]> char-desc
andle: 0x0001, uuid: 00002800-0000-1000-8000-00805f9b34fb
andle: 0x0002, uuid: 00002803-0000-1000-8000-00805f9b34fb
andle: 0x0003, uuid: 00002a05-0000-1000-8000-00805f9b34fb
andle: 0x0004, uuid: 00002902-0000-1000-8000-00805f9b34fb
andle: 0x0005, uuid: 00002800-0000-1000-8000-00805f9b34fb
andle: 0x0006, uuid: 00002803-0000-1000-8000-00805f9b34fb
andle: 0x0007, uuid: 00002a00-0000-1000-8000-00805f9b34fb
andle: 0x0008, uuid: 00002803-0000-1000-8000-00805f9b34fb
andle: 0x0009, uuid: 00002a01-0000-1000-8000-00805f9b34fb
andle: 0x000a, uuid: 00002803-0000-1000-8000-00805f9b34fb
andle: 0x000b, uuid: 00002a04-0000-1000-8000-00805f9b34fb
andle: 0x000c, uuid: 00002800-0000-1000-8000-00805f9b34fb
andle: 0x000d, uuid: 00002803-0000-1000-8000-00805f9b34fb
andle: 0x000e, uuid: 00140000-0001-11e1-ac36-0002a5d5c51b
andle: 0x000f, uuid: 00002902-0000-1000-8000-00805f9b34fb
andle: 0x0010, uuid: 00002803-0000-1000-8000-00805f9b34fb
andle: 0x0011, uuid: 20000000-0001-11e1-ac36-0002a5d5c51b
andle: 0x0012, uuid: 00002902-0000-1000-8000-00805f9b34fb
andle: 0x0019, uuid: 00002800-0000-1000-8000-00805f9b34fb
andle: 0x001a, uuid: 00002803-0000-1000-8000-00805f9b34fb
andle: 0x001b, uuid: 00000002-000f-11e1-ac36-0002a5d5c51b
andle: 0x001c, uuid: 00002902-0000-1000-8000-00805f9b34fb
[C0:6E:28:33:55:58][LE]>
```

Figure 11: Read characteristic value (environmental data) by handle 0x000e

```
[C0:6E:28:33:55:58][LE]> char-read-hnd 000e
Characteristic value/descriptor: a8 0a 03 8a 01 00 15 01
```

Figure 12: Characteristic write to enable environmental data notification in gatttool

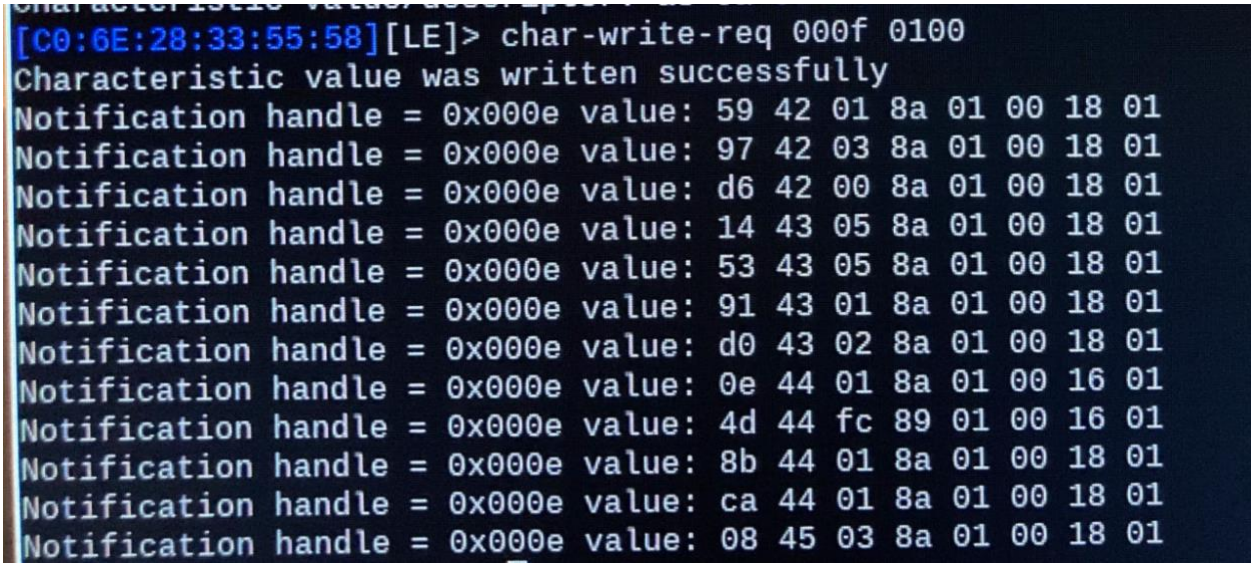


Figure 13: BLE_SampleApp USB debug interface for enabling environmental data notification

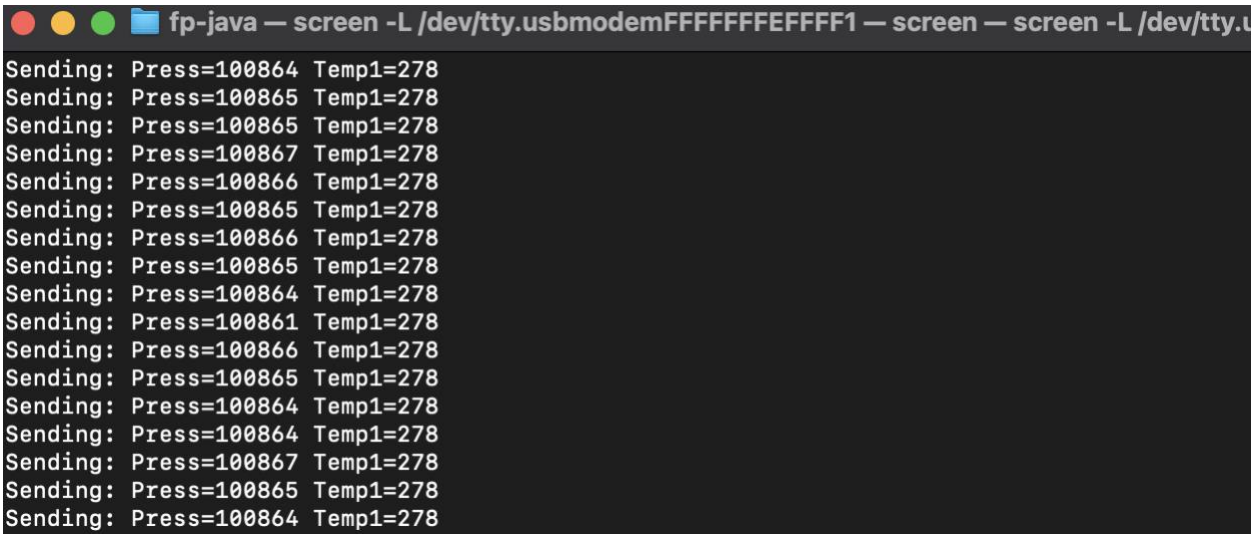


Figure 14: Characteristic write to disable environmental data notification in gatttool

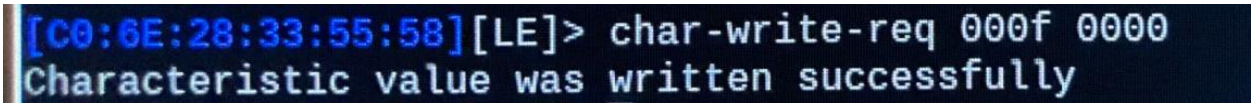


Figure 15: BLE_SampleApp USB debug interface for disabling environmental data notification

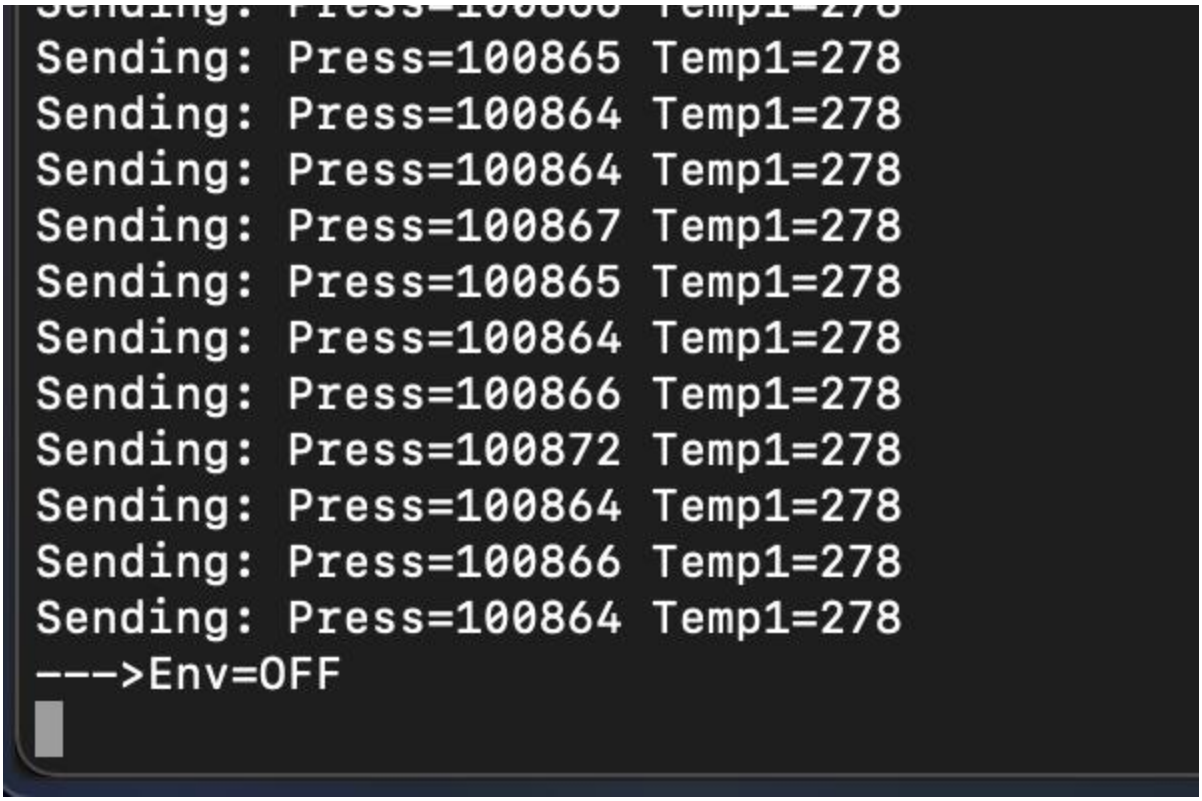


Figure 16: Characteristic write to handle 0x001b with command 2000000001

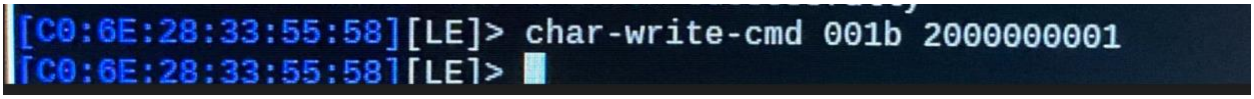


Figure 17: BLE_SampleApp USB debug interface for turn on LED

Figure 18: Characteristic write to hand le 0x001b with command 2000000000

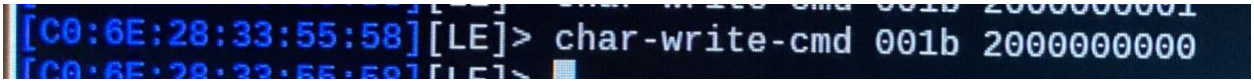


Figure 19: BLE_SampleApp USB debug interface for turn off LED

Figure 20: Disconnect SensorTile and exit gatttool

```
[C0:6E:28:33:55:58][LE]> disconnect

(gatttool:2250): GLib-WARNING **: 07:30:19.410: Invalid file descriptor.
[C0:6E:28:33:55:58][LE]> exit
pi@raspberrypi:~ $
```

Figure 21: Use gatttool to request environmental data

```
pi@raspberrypi:~ $ gatttool -b C0:6E:28:33:55:58 -t random --char-write-req --ha
ndle=0x000f --value=0100 --listen
Characteristic value was written successfully
Notification handle = 0x000e value: fe 5c fb 89 01 00 19 01
Notification handle = 0x000e value: 3d 5d 00 8a 01 00 19 01
Notification handle = 0x000e value: 7b 5d fc 89 01 00 19 01
Notification handle = 0x000e value: ba 5d 06 8a 01 00 19 01
Notification handle = 0x000e value: f8 5d fd 89 01 00 19 01
Notification handle = 0x000e value: 37 5e 01 8a 01 00 19 01
Notification handle = 0x000e value: 75 5e fd 89 01 00 19 01
Notification handle = 0x000e value: b4 5e fb 89 01 00 18 01
Notification handle = 0x000e value: f2 5e fb 89 01 00 18 01
Notification handle = 0x000e value: 31 5f ff 89 01 00 18 01
Notification handle = 0x000e value: 6f 5f 00 8a 01 00 18 01
```

Figure 22: Request environmental data and save data in test.txt

```
pi@raspberrypi:~ $ gatttool -b C0:6E:28:33:55:58 -t random --char-write-req --ha
ndle=0x000f --value=0100 --listen > test.txt
^C
pi@raspberrypi:~ $ ls
Bookshelf  Documents  Music     Public     test.txt
Desktop    Downloads  Pictures  Templates  Videos
```

Figure 23: Verify saved environmental data in test.txt

```
pi@raspberrypi:~ $ cat test.txt
Characteristic value was written successfully
Notification handle = 0x000e value: f9 8e fb 89 01 00 19 01
Notification handle = 0x000e value: 37 8f 00 8a 01 00 19 01
Notification handle = 0x000e value: 76 8f 00 8a 01 00 19 01
Notification handle = 0x000e value: b4 8f ff 89 01 00 19 01
Notification handle = 0x000e value: f3 8f fc 89 01 00 18 01
Notification handle = 0x000e value: 31 90 01 8a 01 00 18 01
Notification handle = 0x000e value: 70 90 00 8a 01 00 18 01
Notification handle = 0x000e value: ae 90 fd 89 01 00 18 01
Notification handle = 0x000e value: ed 90 00 8a 01 00 18 01
Notification handle = 0x000e value: 2b 91 fd 89 01 00 18 01
pi@raspberrypi:~ $
```

4. submit your test.txt file as in Figure 23.

The test.txt would be submitted separately.

5. Translate the data recorded in the final text file: test.txt to physical values. (you could write a python program to automate this otherwise you can do it manually).

The following table was converted from test.txt by a block of python code.

Hanldes	Timestamps (sec)	Pressure (mbar)	Temperature (degree C)
Notification handle = 0x000evalue:	366.01	1008.59	28.1
Notification handle = 0x000evalue:	366.63	1008.64	28.1
Notification handle = 0x000evalue:	367.26	1008.64	28.1
Notification handle = 0x000evalue:	367.88	1008.63	28.1
Notification handle = 0x000evalue:	368.51	1008.6	28
Notification handle = 0x000evalue:	369.13	1008.65	28
Notification handle = 0x000evalue:	369.76	1008.64	28
Notification handle = 0x000evalue:	370.38	1008.61	28
Notification handle = 0x000evalue:	371.01	1008.64	28
Notification handle = 0x000evalue:	371.63	1008.61	28

_____Tutorial 8_____

Part 1. Please follow the tutorial 8

Answer the following questions:

1. **GATT** is used exclusively after a connection has been established between the two devices

(True or False?)

Answer: **True**

2. Complete the sentence below:

We use SensorTile as the **GATT** _____

a. Server

b. Client

c. Peer

d. Service

Answer: **A.**

3. Complete the sentence below:

We use BeagleBone Black Rev. C/Raspberry Pi3B+/Pi Zero W as the GATT _____

a. Server

b. Client

c. Peer

d. Service

Answer: **B.**

4. Complete the sentence below:

We can discover, read, and write characteristics with gatttool.

GATT stands for **Generic Attribute Profile** and defines a data structure for organizing characteristics and attributes

Part 2 Please follow the tutorial 8

1. Take screenshots corresponding to Figures 8, 13 and 15

Figure8: FP-SNS-ALLMEMS1 USB debug interface

Figure 13: char-write-req to enable motiondata notification in gatttool

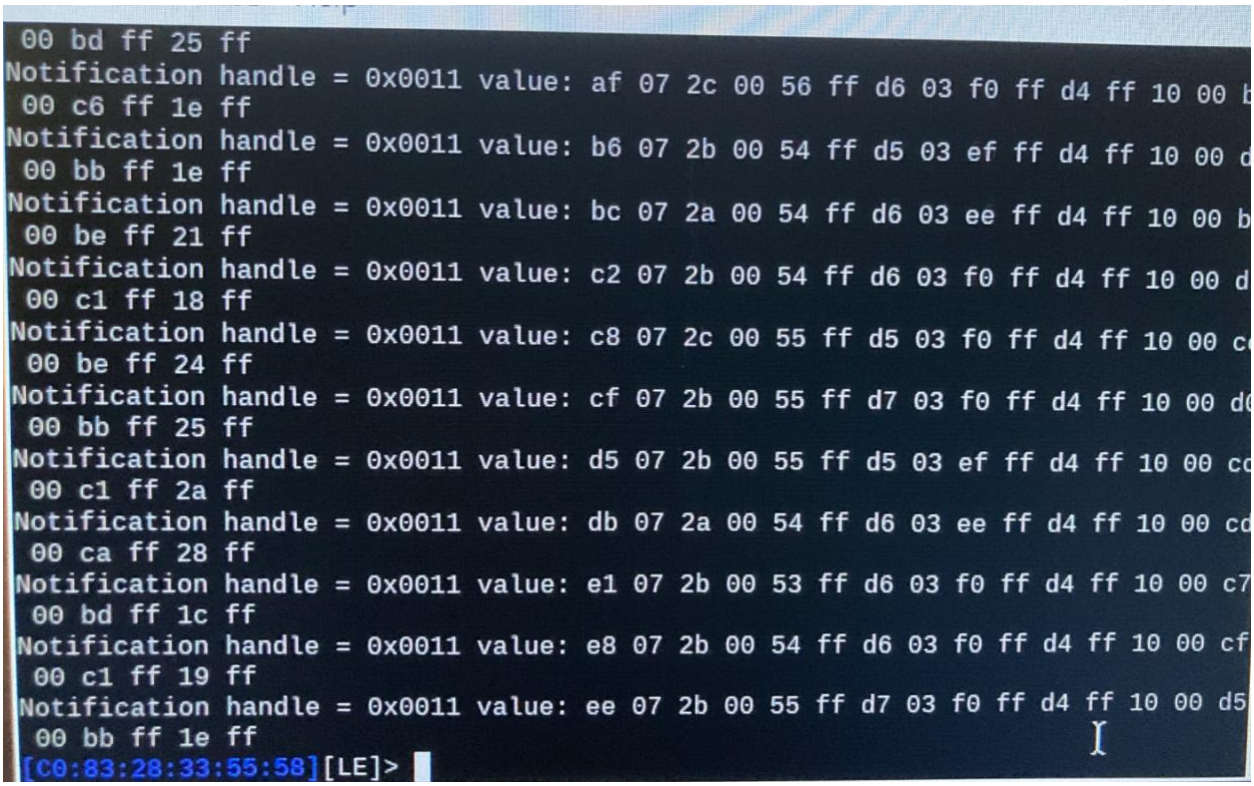
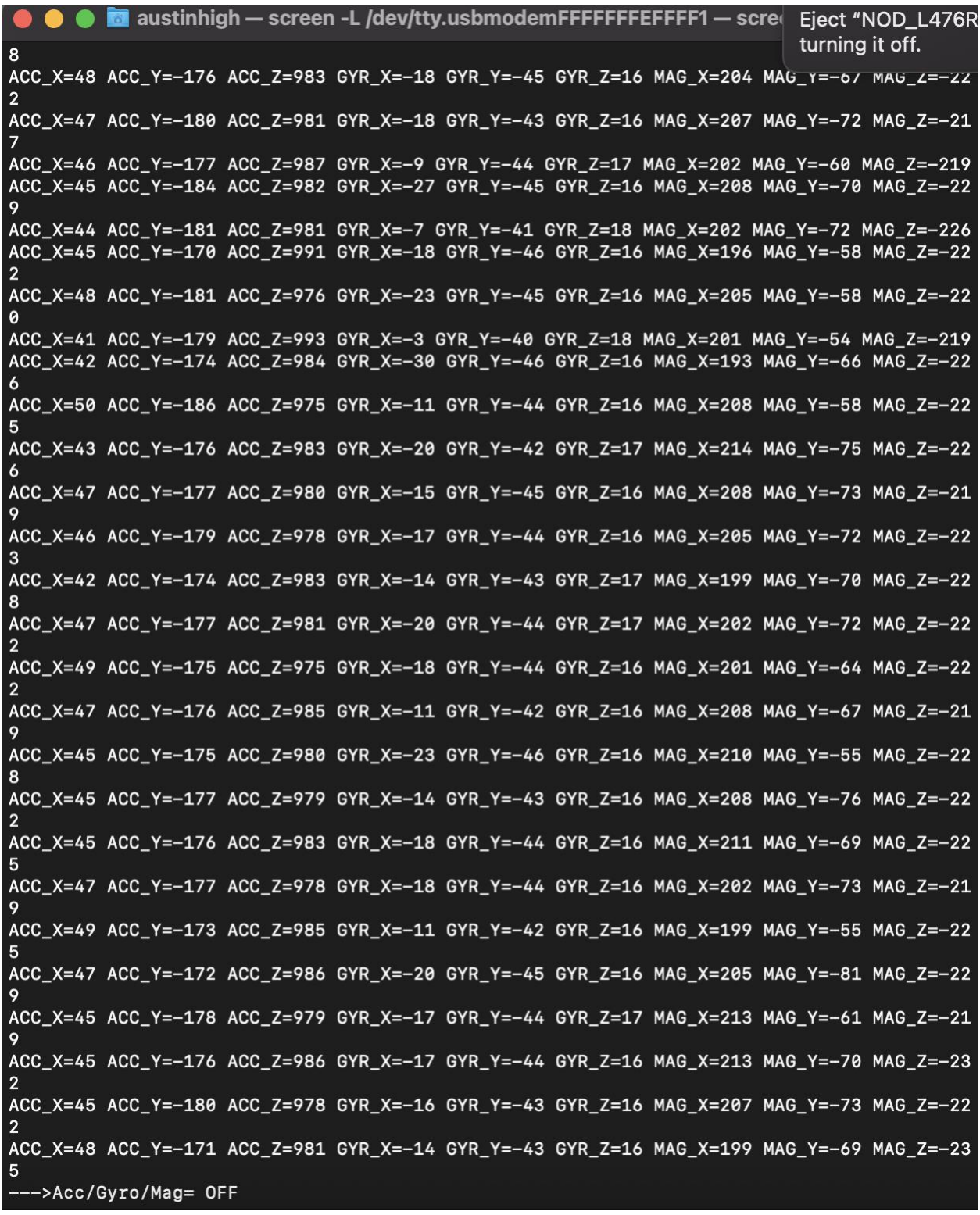


Figure 15: USB debug interface for enable/disable motion data notification



2. Save Requested Motion Data to Text File as in section 4 of the tutorial 8 and submit text file: motion_data.txt

The motion_data.txt will be submitted separately.