STMicroelectronics SensorTile Tutorial: BLE Communication via BlueZ and the GATT Profile

Table of Contents

1. Introduction to This Tutorial

In ***STMicroelectronics SensorTile Tutorial: Introduction to Bluetooth Low Energy (BLE) Interfaces***, we introduced how to use BlueZ to discover, pair, and connect with SensorTile on Linux platform. In this tutorial, we will use BlueZ gatttool utility to read environmental data from SensorTile Bluetooth Generic Attribute Profile.

The Generic Attributes (GATT) define a hierarchical data structure that is exposed to connected Bluetooth Low Energy devices. GATT is built on top of the Attribute Protocol (ATT), which uses GATT data to define the way that two Bluetooth Low Energy devices send and receive standard messages. In this tutorial, we will use SensorTile as the GATT server to provide services and use BeagleBone as the GATT client to send requests to the GATT server to get environmental data collected on SensorTile.

The Tutorial steps provide:

1. An introduction to *gatttool*
2. An introduction to request environmental data using gatttool
3. An introduction to control the LED on SensorTile using gatttool

For more information regarding the SensorTile board, please open the following link.
www.st.com/sensortile

For more information regarding Bluetooth, please refer to Bluetooth official website.

1.1. List of Required Equipment and Materials

1) 1x STMicroelectronics SensorTile kit.
2) 1x STMicroelectronics Nucleo Board.
3) 1x Personal Computer with two USB type-A inputs OR you must have a powered USB hub.
4) 1x USB 2.0 A-Male to Micro-B Cable (micro USB cable).
5) 1x USB 2.0 A-Male to Mini-B Cable (mini USB cable).
6) Network access to the Internet.
7) A Linux machine (BeagleBone, Intel Edison, Rpi, and etc)

1.2. Prerequisite Tutorials

It is recommended that users have completed and are familiar with the contents of the following tutorials before proceeding.

1. STMicroelectronics SensorTile Tutorial: Introduction to Bluetooth Low Energy (BLE) Interfaces

Your instructor will ensure you have the required background regarding Bluetooth.

2. Introduction to Gatttool

Gatttool is the BlueZ utility that handles BLE communication with GATT. We will use it to discover all the services and characteristics provided by SensorTile, which is the GATT server. Before you start this part, make sure you have successfully installed BlueZ on your Linux machine and flashed the BLE_SampleApp firmware to SensorTile as instructed in ***STMicroelectronics SensorTile Tutorial: Introduction to Bluetooth Low Energy (BLE) Interfaces.***

1. Connect your SensorTile with your computer and use terminal screen command (MAC user) or Putty serial connection (Windows user) to access the USB debug interface of BLE_SampleApp firmware as instructed in ***STMicroelectronics SensorTile Tutorial: Introduction to STMicroelectronics Development Environment and DataLog Project Example -> Example Data Logging Project -> Debug***. See Figure 1.

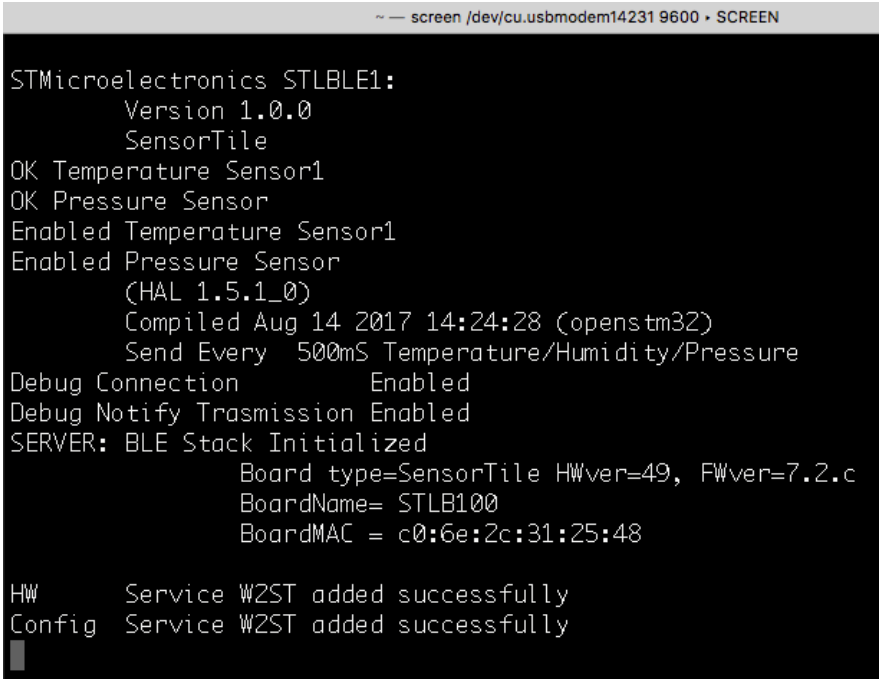

Figure 1: BLE debug interface via USB streaming.

2. Connect your BeagleBone with your computer and log into BeagleBone.

3. In BeagleBone, type command ***$ bluetoothctl*** to use BlueZ in interactive mode. See Figure 2.

```
root@beaglebone:~# bluetoothctl
[NEW] Controller 5C:F8:21:D6:9D:2D beaglebone [default]
[NEW] Device C0:6E:2C:31:25:48 STLB100
[bluetooth]#
```

Figure 2: Use bluetootctl in BeagleBone

4. As instructed in ***STMicroelectronics SensorTile Tutorial: Introduction to Bluetooth Low Energy (BLE) Interfaces,*** discover, pair, and **disconnect** your SensorTile device in bluetoothctl. Use command ***info*** in bluetoothctl to check that your SensorTile is paired but **disconnected**. See Figure 3.

```
[bluetooth]# info C0:6E:2C:31:25:48
Device C0:6E:2C:31:25:48
        Name: STLB100
        Alias: STLB100
        Paired: yes
        Trusted: no
        Blocked: no
        Connected: no
        LegacyPairing: no
        UUID: Vendor specific           (00000000-0001-11e1-9ab4-0002a5d5c51b)
        UUID: Vendor specific           (00000000-000f-11e1-9ab4-0002a5d5c51b)
        UUID: Generic Access Profile    (00001800-0000-1000-8000-00805f9b34fb)
        UUID: Generic Attribute Profile (00001801-0000-1000-8000-00805f9b34fb)
```

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

5. Copy the MAC address of SensorTile and exit bluetoothctl. See Figure 4.

```
[bluetooth]# exit
[DEL] Controller 5C:F8:21:D6:9D:2D beaglebone [default]
root@beaglebone:~#
```

Figure 4: Exit bluetoothctl

6. In BeagleBone, use command ***$ gatttool –b <MAC_ADDRESS_SENSORTILE> –t random –I*** to run gatttool in interactive mode. **–b** is the option to indicate the mac address. **–t** is the option to declare the Bluetooth MAC address type. In BLE_SampleApp firmware, this is set to type random. **–I** is the option to use gatttool in interactive mode. See Figure 5.

```
root@beaglebone:~# gatttool -b C0:6E:2C:31:25:48 -t random -I
[C0:6E:2C:31:25:48][LE]>
```

Figure 5: Use gatttool with proper options in interactive mode

7. In gatttool, use command ***connect*** to connect BeagleBone with SensorTile. If the connection is successful, the MAC address should turn into color of blue. See Figure 6.

```
[C0:6E:2C:31:25:48][LE]> connect
Attempting to connect to C0:6E:2C:31:25:48
Connection successful
[C0:6E:2C:31:25:48][LE]>
```

Figure 6: Connect with SensorTile in gatttool interactive mode

8. In USB debug interface, you can also see the successful connection information. See Figure 7.

```
STMicroelectronics STLBLE1:
        Version 1.0.0
        SensorTile
OK Temperature Sensor1
OK Pressure Sensor
Enabled Temperature Sensor1
Enabled Pressure Sensor
        (HAL 1.5.1_0)
        Compiled Aug 14 2017 14:24:28 (openstm32)
        Send Every  500mS Temperature/Humidity/Pressure
Debug Connection         Enabled
Debug Notify Trasmission Enabled
SERVER: BLE Stack Initialized
                Board type=SensorTile HWver=49, FWver=7.2.c
                BoardName= STLB100
                BoardMAC = c0:6e:2c:31:25:48

HW      Service W2ST added successfully
Config  Service W2ST added successfully
>>>>>CONNECTED 5c:f8:21:d6:9d:2d
```

Figure 7: BLE_SampleApp USB debug interface for successful connection

9. In gatttool, use command ***primary*** to discover all the primary services. See Figure 8.

```
[C0:6E:2C:31:25:48][LE]> primary
attr handle: 0x0001, end grp handle: 0x0004 uuid: 00001801-0000-1000-8000-00805f
9b34fb
attr handle: 0x0005, end grp handle: 0x000b uuid: 00001800-0000-1000-8000-00805f
9b34fb
attr handle: 0x000c, end grp handle: 0x0012 uuid: 00000000-0001-11e1-9ab4-0002a5
d5c51b
attr handle: 0x0019, end grp handle: 0x001c uuid: 00000000-000f-11e1-9ab4-0002a5
d5c51b
```

Figure 8: GATT primary services discovery in gatttool

10. In gatttool, use command ***characteristics*** to discover all the characteristics. See Figure 9.

```
[C0:6E:2C:31:25:48][LE]> characteristics
handle: 0x0002, char properties: 0x20, char value handle: 0x0003, uuid: 00002a05
-0000-1000-8000-00805f9b34fb
handle: 0x0006, char properties: 0x4e, char value handle: 0x0007, uuid: 00002a00
-0000-1000-8000-00805f9b34fb
handle: 0x0008, char properties: 0x4e, char value handle: 0x0009, uuid: 00002a01
-0000-1000-8000-00805f9b34fb
handle: 0x000a, char properties: 0x02, char value handle: 0x000b, uuid: 00002a04
-0000-1000-8000-00805f9b34fb
handle: 0x000d, char properties: 0x12, char value handle: 0x000e, uuid: 00140000
-0001-11e1-ac36-0002a5d5c51b
handle: 0x0010, char properties: 0x12, char value handle: 0x0011, uuid: 20000000
-0001-11e1-ac36-0002a5d5c51b
handle: 0x001a, char properties: 0x14, char value handle: 0x001b, uuid: 00000002
-000f-11e1-ac36-0002a5d5c51b
```

Figure 9: GATT characteristics discovery in gatttool

11. In gatttool, use command ***char-desc*** to discover all the characteristic descriptors. See Figure 10.

```
[C0:6E:2C:31:25:48][LE]> char-desc
handle: 0x0001, uuid: 00002800-0000-1000-8000-00805f9b34fb
handle: 0x0002, uuid: 00002803-0000-1000-8000-00805f9b34fb
handle: 0x0003, uuid: 00002a05-0000-1000-8000-00805f9b34fb
handle: 0x0004, uuid: 00002902-0000-1000-8000-00805f9b34fb
handle: 0x0005, uuid: 00002800-0000-1000-8000-00805f9b34fb
handle: 0x0006, uuid: 00002803-0000-1000-8000-00805f9b34fb
handle: 0x0007, uuid: 00002a00-0000-1000-8000-00805f9b34fb
handle: 0x0008, uuid: 00002803-0000-1000-8000-00805f9b34fb
handle: 0x0009, uuid: 00002a01-0000-1000-8000-00805f9b34fb
handle: 0x000a, uuid: 00002803-0000-1000-8000-00805f9b34fb
handle: 0x000b, uuid: 00002a04-0000-1000-8000-00805f9b34fb
handle: 0x000c, uuid: 00002800-0000-1000-8000-00805f9b34fb
handle: 0x000d, uuid: 00002803-0000-1000-8000-00805f9b34fb
handle: 0x000e, uuid: 00140000-0001-11e1-ac36-0002a5d5c51b
handle: 0x000f, uuid: 00002902-0000-1000-8000-00805f9b34fb
handle: 0x0010, uuid: 00002803-0000-1000-8000-00805f9b34fb
handle: 0x0011, uuid: 20000000-0001-11e1-ac36-0002a5d5c51b
handle: 0x0012, uuid: 00002902-0000-1000-8000-00805f9b34fb
handle: 0x0019, uuid: 00002800-0000-1000-8000-00805f9b34fb
handle: 0x001a, uuid: 00002803-0000-1000-8000-00805f9b34fb
handle: 0x001b, uuid: 00000002-000f-11e1-ac36-0002a5d5c51b
handle: 0x001c, uuid: 00002902-0000-1000-8000-00805f9b34fb
```

Figure 10: GATT characteristic descriptors discovery in gatttool

3. Request Environmental Data Using Gatttool Interactively

Refer to Figure 10, there are different handles corresponding to different uuids. You can use handles to communicate with SensorTile for BLE services and characteristics.

Handles in the red rectangular are very important:

0x000e is characteristic value handle of environmental data.
0x000f is client characteristic configuration handle for environmental data transmission.
0x001b is SensorTile LED configuration handle.

You will use these three handles for the following sessions.

In this session, you will use gatttool to request environmental data.

1. Make sure that the SensorTile is connected with BeagleBone in gatttool.

2. In gatttool, use command ***char-read-hnd 000e*** to read characteristic value (environmental data) with handle 0x000e. See Figure 11. The data will be explained the step 4.

```
[C0:6E:2C:31:25:48][LE]> char-read-hnd 000e
Characteristic value/descriptor: 58 14 b1 86 01 00 0f 01
```

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

3. In gatttool, use command ***char-write-req 000f 0100*** to modify the characteristic value of handle 0x000f and therefore enable the notification of environmental data (handle 0x000e) on SensorTile. After you enter the command, SensorTile will start to send environmental data continuously and BeagleBone will listen to the data transmission in gatttool. See Figure 12.

```
[C0:6E:2C:31:25:48][LE]> char-write-req 000f 0100
Characteristic value was written successfully
Notification handle = 0x000e value: c2 7a ab 86 01 00 0f 01
Notification handle = 0x000e value: 01 7b ab 86 01 00 0f 01
Notification handle = 0x000e value: 3f 7b aa 86 01 00 0f 01
Notification handle = 0x000e value: 7e 7b aa 86 01 00 0f 01
Notification handle = 0x000e value: bc 7b ab 86 01 00 0f 01
Notification handle = 0x000e value: fb 7b aa 86 01 00 0e 01
Notification handle = 0x000e value: 39 7c ad 86 01 00 0e 01
Notification handle = 0x000e value: 78 7c ab 86 01 00 0e 01
Notification handle = 0x000e value: b6 7c aa 86 01 00 0e 01
Notification handle = 0x000e value: f5 7c aa 86 01 00 0e 01
Notification handle = 0x000e value: 33 7d aa 86 01 00 0e 01
Notification handle = 0x000e value: 72 7d aa 86 01 00 0e 01
Notification handle = 0x000e value: b0 7d aa 86 01 00 0e 01
Notification handle = 0x000e value: ef 7d ab 86 01 00 0e 01
Notification handle = 0x000e value: 2d 7e ab 86 01 00 0f 01
Notification handle = 0x000e value: 6c 7e ad 86 01 00 0f 01
Notification handle = 0x000e value: aa 7e ad 86 01 00 0e 01
```

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

4. Switch to the USB debug interface. The debug information will be similar to figure 13.

```
--->Env=ON
Sending: Press=100011 Temp1=271
Sending: Press=100011 Temp1=271
Sending: Press=100010 Temp1=271
Sending: Press=100010 Temp1=271
Sending: Press=100011 Temp1=271
Sending: Press=100010 Temp1=270
Sending: Press=100013 Temp1=270
Sending: Press=100011 Temp1=270
Sending: Press=100010 Temp1=270
Sending: Press=100010 Temp1=270
Sending: Press=100010 Temp1=270
Sending: Press=100010 Temp1=270
Sending: Press=100010 Temp1=270
Sending: Press=100011 Temp1=270
Sending: Press=100011 Temp1=271
Sending: Press=100013 Temp1=271
Sending: Press=100013 Temp1=270
```

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

5. In gatttool, use command ***char-write-req 000f 0000*** to disable the notification of environmental data. See Figure 14.

```
[C0:6E:2C:31:25:48][LE]> char-write-req 000f 0000
Characteristic value was written successfully
```

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

6. Switch to USB debug interface. Debug information shows that the environmental data notification is off (disabled). See Figure15.

```
--->Env=ON
Sending: Press=100011 Temp1=271
Sending: Press=100011 Temp1=271
Sending: Press=100010 Temp1=271
Sending: Press=100010 Temp1=271
Sending: Press=100011 Temp1=271
Sending: Press=100010 Temp1=270
Sending: Press=100013 Temp1=270
Sending: Press=100011 Temp1=270
Sending: Press=100010 Temp1=270
Sending: Press=100010 Temp1=270
Sending: Press=100010 Temp1=270
Sending: Press=100010 Temp1=270
Sending: Press=100010 Temp1=270
Sending: Press=100011 Temp1=270
Sending: Press=100011 Temp1=271
Sending: Press=100013 Temp1=271
Sending: Press=100013 Temp1=270
--->Env=OFF
```

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

7. By comparing Figure 12 and Figure 13, we can interpret the environmental data sent from SensorTile.

   In Figure 12, all the data are sent in the format of hexadecimal. The first 4 bits in the yellow rectangle (c2 7a) are timestamps. The next 8 bits in the red rectangle (ab 86 01 00) are air pressure data. The last 4 bits in the blue rectangle (0f 01) are temperature data from temperature sensor 1.

   In Figure 13, pressure data are 100011, which are actually 1000.11 mbar. Temperature data are 271, which are actually 27.1 °C.

The environmental data sent by notification should be read from right to left (little endian rule), which means that the pressure data are 0x000186ab and the temperature data are 0x010f. By transferring hexadecimal to decimal, the pressure data is 11*1 + 10*16 + 6*256 + 8*4096 + 1*65536 = 100011 and the temperature data is 256*1 + 15 = 271, which is the same as the pressure value and the temperature value in USB debug interface (Figure 13).

You can compare other lines of data in gatttool and USB debug interface to verify the data interpretation above.

4. Control SensorTile LED Using Gatttool Interactively

In this session, you will use gatttool to control SensorTile LED.

1. Make sure that that the SensorTile is connected with BeagleBone in gatttool.

2. In gatttool, use command ***char-write-cmd 001b 2000000001*** to write configuration command (2000000***01***) to handle 0x001b to turn **on** the SensorTile LED. You can check if the LED on your SensorTile is turned on. See Figure 16.

```
[C0:6E:2C:31:25:48][LE]> char-write-cmd 001b 2000000001
```

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

3. In USB debug interface, you can see the debug information that the configuration command to turn on the SensorTile LED is composed of feature mask (F), which is 20000000, and command (C), which is 01 (1). See Figure 17.

```
Conf Sig F=20000000 C= 1
```

Figure 17: BLE_SampleApp USB debug interface for turn on LED

4. In gatttool, use command ***char-write-cmd 001b 2000000000*** to write configuration command (2000000***00***) to handle 0x001b to turn off the SensorTile LED. You can check if the LED on your SensorTile is turned off. See Figure 18.

```
[C0:6E:2C:31:25:48][LE]> char-write-cmd 001b 2000000000
```

Figure 18: Characteristic write to handle 0x001b with command 2000000000

5. In USB debug interface, you can see the debug information that the configuration command to turn off the SensorTile LED is composed of feature mask (F), which is 20000000, and command (C), which is 00 (0). See Figure 19.

```
Conf Sig F=20000000 C= 0
```

Figure 19: BLE_SampleApp USB debug interface for turn off LED

5. Save Requested Environmental Data to Text File

In this session, you will use gatttool to request environmental data and then save the data to text file for future development.

1. In gatttool, use command ***disconnect*** and ***exit*** to disconnect BeagleBone and SensorTile and exit gatttool as Figure 20.

```
[C0:6E:2C:31:25:48][LE]> disconnect
[C0:6E:2C:31:25:48][LE]> exit
root@beaglebone:~#
```

Figure 20: Disconnect SensorTile and exit gatttool

2. Make sure that the SensorTile is successfully paired but disconnected. In BeagleBone, use command ***$ gatttool -b <MAC_ADDRESS_SENSORTILE> -t random --char-write-req --handle=0x000f --value=0100 --listen*** to enable the notification of environmental data on SensorTile and listen to the data transmission. See Figure 21. Use **Ctrl + C** to **terminate** the data transmission.

```
root@beaglebone:~# gatttool -b C0:6E:2C:31:25:48 -t random --char-write-req --ha
ndle=0x000f --value=0100 --listen
Characteristic value was written successfully
Notification handle = 0x000e value: 52 4b 25 87 01 00 09 01
Notification handle = 0x000e value: 91 4b 25 87 01 00 09 01
Notification handle = 0x000e value: cf 4b 26 87 01 00 09 01
Notification handle = 0x000e value: 0e 4c 24 87 01 00 09 01
Notification handle = 0x000e value: 4c 4c 25 87 01 00 09 01
Notification handle = 0x000e value: 8b 4c 26 87 01 00 09 01
Notification handle = 0x000e value: c9 4c 25 87 01 00 09 01
Notification handle = 0x000e value: 08 4d 24 87 01 00 09 01
Notification handle = 0x000e value: 46 4d 24 87 01 00 09 01
Notification handle = 0x000e value: 85 4d 25 87 01 00 07 01
Notification handle = 0x000e value: c3 4d 24 87 01 00 07 01
Notification handle = 0x000e value: 02 4e 25 87 01 00 07 01
```

Figure 21: Use gatttool to request environmental data

3. In BeagleBone, use command ***$ gatttool -b <MAC_ADDRESS_SENSORTILE> -t random --char-write-req --handle=0x000f --value=0100 --listen > test.txt***, wait 5 seconds, use **Ctrl + C** to terminate the gatttool, and use command ***$ ls*** to check if test.txt is saved. Now, you have saved the 5 seconds environmental data in test.txt file for future use. See Figure 22.

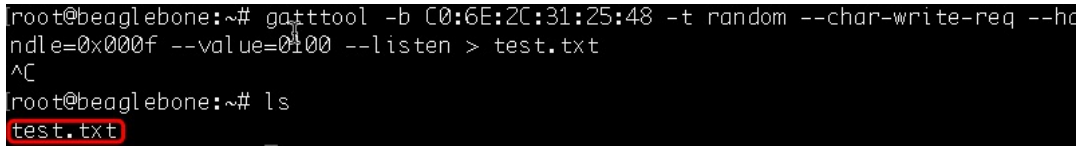

Figure 22: Request environmental data and save data in test.txt

4. You can use command ***$ cat test.txt*** to quickly check the data in test.txt. See Figure 23.

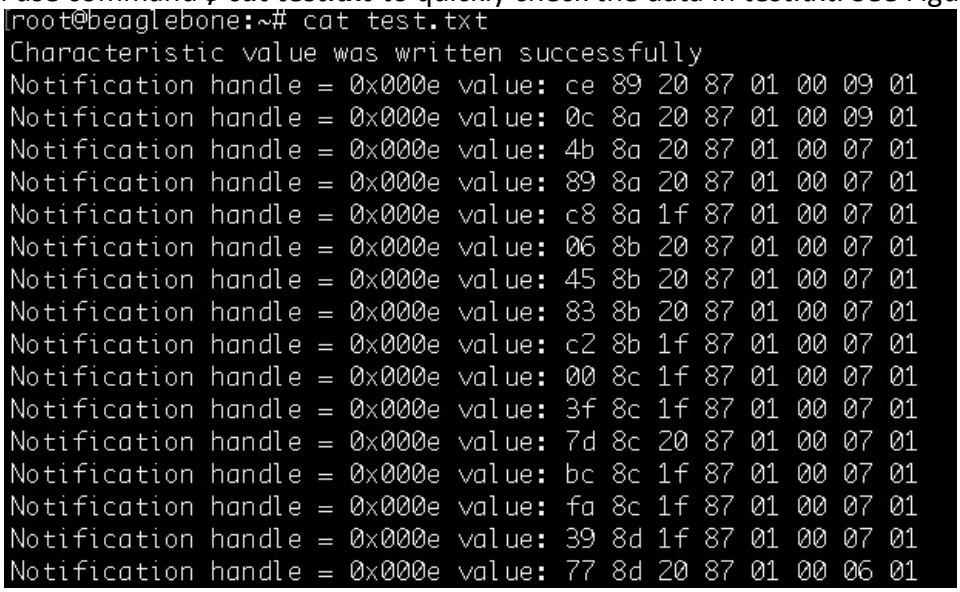

Figure 23: Verify saved environmental data in test.txt