SensorTag humidity data corrupt?

Not Answered

Ben Adler

Recently, I've played with the SensorTag's firmware in order to enable indefinite advertising. And everything seemed to work well.

But when I now try reading humidity and temperature from the humidity sensor, it seems I am getting corrupt data. Let me explain using linux's gatttool:

```
# gatttool -b 90:59:AF:0B:8A:7D -I
[   ][90:59:AF:0B:8A:7D][LE]> connect
[CON][90:59:AF:0B:8A:7D][LE]> char-write-cmd 0x3c 01
[CON][90:59:AF:0B:8A:7D][LE]> char-read-hnd 0x38
[CON][90:59:AF:0B:8A:7D][LE]>
Characteristic value/descriptor: 41 63 63 65 6c 2e 20 50 65 72 69 6f 64
```

So I am enabling the humidity sensor (writing 01 to 0x3c) and then reading its data handle (0x38). But instead of 4 bytes, I am receiving 13 bytes.

```
[CON][90:59:AF:0B:8A:7D][LE]> char-write-cmd 0x29 01
[CON][90:59:AF:0B:8A:7D][LE]> char-read-hnd 0x25
[CON][90:59:AF:0B:8A:7D][LE]>
Characteristic value/descriptor: ed fe 60 0b
[CON][90:59:AF:0B:8A:7D][LE]> char-read-hnd 0x25
[CON][90:59:AF:0B:8A:7D][LE]>
Characteristic value/descriptor: c6 fe 5c 0b
```

It seems that the IR temperature can be read correctly.

```
[CON][90:59:AF:0B:8A:7D][LE]> char-read-hnd 0x38
[CON][90:59:AF:0B:8A:7D][LE]>
Characteristic value/descriptor: 41 63 63 65 6c 2e 20 50 65 72 69 6f 64
[CON][90:59:AF:0B:8A:7D][LE]> exit
```

The SAME 13 bytes again?!

Thinking that I might have created a bug in the firmware, I compiled the original SensorTag source from the BLE-CC254x-1.4.0 folder using IAR 8.30.1 and flashed it using the CC debugger. But the error remains.

What might be wrong here?

Thanks!

Ben Adler

9 Replies

Ben Adler

I still haven't managed to solve this problem. By now, I have access to a Nexus5, and while the BLE SensorTag app works so-so in terms of reliability, it does read the humidty sensor correctly.

Further, I've been trying to read the Test Service, which should reply with the result from the power-on self-test. As http://processors.wiki.ti.com/index.php/SensorTag_User_Guide#Test_Service notes, this value should always be 0x3F. In my case, it is always 00:

```
# gatttool -b 90:59:AF:0B:8A:7D --interactive
[   ][90:59:AF:0B:8A:7D][LE]> connect
[CON][90:59:AF:0B:8A:7D][LE]> char-read-hnd 0x64
[CON][90:59:AF:0B:8A:7D][LE]>
Characteristic value/descriptor: 00
[CON][90:59:AF:0B:8A:7D][LE]> char-write-cmd 0x67 0x01
[CON][90:59:AF:0B:8A:7D][LE]> char-read-hnd 0x64
[CON][90:59:AF:0B:8A:7D][LE]>
Characteristic value/descriptor: 00
[CON][90:59:AF:0B:8A:7D][LE]> char-write-cmd 0x29 01
[CON][90:59:AF:0B:8A:7D][LE]> char-read-hnd 0x25
[CON][90:59:AF:0B:8A:7D][LE]>
Characteristic value/descriptor: cc fe ec 0a
[CON][90:59:AF:0B:8A:7D][LE]>
```

So, the self-test seems to have failed, but reading the temperature sensor works.

What is going on here? Please give me an idea!

thanks!

ben

Michael Vartanian

Hi Ben,

I was able to read the humidity sensor and it returned 4 bytes using the default firmware loaded on the sensor tag. I have not modified it. I am not sure the firmware version as I do not know how to read that. If you know how to read that and let me know, I would be glad to tell you which version of firmware I am running.

I used BlueZ 5.12.

Below is the gatttool commands and results from linux on the raspberry pi...

```
pi@raspberrypi3 ~ $ sudo gatttool -b 34:B1:F7:D1:40:56 --interactive
[34:B1:F7:D1:40:56][LE]> connect
Attempting to connect to 34:B1:F7:D1:40:56
Connection successful
[34:B1:F7:D1:40:56][LE]> char-write-cmd 0x3c 01
[34:B1:F7:D1:40:56][LE]> char-read-hnd 0x38
Characteristic value/descriptor: 28 64 ce 4c
[34:B1:F7:D1:40:56][LE]>
```

Hope this helps. Let me know if you need any further information from me.

Ben Adler

In reply to Michael Vartanian:

Michael,

thank you so much for your answer! It made me revisit the problem and look at an ASCII table. The data I get is actually a string:

41 A
63 c
63 c

65 e
6c l
2e .
20 SPACE
50 P
65 e
72 r
69 i
6f o
64 d

Really weird. Usually, I'd says that there must be something wrong with the firmware. But this happens even with the original firmware flashed onto the SensorTag. Maybe I've done something wrong while flashing, or maybe the Tag is broken.

I have ordered another SensorTag and will test it when it arrives. Then I'll post an update.

Thanks again!

ben

Ben Adler

In reply to Ben Adler:

Using the linux "strings" tool, I found that the strings

Accel. Data
Accel. Conf.
Accel. Period

are present in the firmware. That was to be expected. Thinking that I might have a) constructed a corrupt firmware or b) made a mistake while flashing it, I thought I should download official firmware from http://processors.wiki.ti.com/index.php/SensorTag_Firmware:

http://processors.wiki.ti.com/images/1/10/SensorTagFW_1_5.zip contains

SensorTagImgA_v1_5.bin
SensorTagImgB_v1_5.bin
CC2541DK_BIM_SensorTagOadImgA_v1_5.hex

and I believe only the .hex file is meant for flashing using the CC debugger and the Ti Smart RF Flash Programmer. After flashing the .hex, the BLE SensorTag app displays an "GATT error code: 13" error.

Using gatttool in linux yields the same behavior and bytes as before.

Then I downloaded http://processors.wiki.ti.com/images/9/9c/SensorTag_accel_8G.zip from the same wiki page, unpacked SensorTag_accel_8G.hex and flashed it. BLE SensorTag app works and shows correct humidity data (~47%). Using gatttool also works:

```
# gatttool -b 90:59:AF:0B:8A:7D --interactive
[   ][90:59:AF:0B:8A:7D][LE]> connect
[CON][90:59:AF:0B:8A:7D][LE]> char-read-hnd 0x38
[CON][90:59:AF:0B:8A:7D][LE]>
Characteristic value/descriptor: 00 00 00 00
[CON][90:59:AF:0B:8A:7D][LE]> char-write-cmd 0x3c 01
[CON][90:59:AF:0B:8A:7D][LE]> char-read-hnd 0x38
[CON][90:59:AF:0B:8A:7D][LE]>
Characteristic value/descriptor: 1c 65 9a 6e
[CON][90:59:AF:0B:8A:7D][LE]> char-write-cmd 0x3c 00
[CON][90:59:AF:0B:8A:7D][LE]> char-read-hnd 0x38
[CON][90:59:AF:0B:8A:7D][LE]>
Characteristic value/descriptor: 00 00 00 00
[CON][90:59:AF:0B:8A:7D][LE]>
```

Using this firmware, suddenly https://github.com/msaunby/ble-sensor-pi also works just fine.

Dear Texas Instruments, why does one official firmware work while the other official firmware doesn't work?

What do I need to do in order to flash CC2541DK_BIM_SensorTagOadImgA_v1_5.hex correctly?

How can I create working firmware using IAR Embedded Workbench? I described my process at http://bensotech.blogspot.de/2013/12/ti-sensortag-bluetooth-low-energy-ble.html and thought that it worked, but I obviously killed the humidity data in the process :(

Edgars Nemse

In reply to Ben Adler:

I encountered the same issue today, trying to modify the minimum sensor read period. At first I thought I had broken something, but even firmware compiled from the unmodified source didn't work properly.

After a long investigation I discovered something interesting - everything still works, except the handles for all of the characteristics after the IR temperature sensor have been increased by 3, relative to what they were in the previous firmware versions, and to what is listed in the user guide.

For example, the accelerometer data is now at handle 0x2D + 3 = 0x30.

I'm not sure what the exact cause of this is, but it appears that the temperature service has a new characteristic compared to previous firmware versions(?)

At any rate, it would be nice of TI to let us know if that's a bug, or a deliberate change, and give some indication of that in the Wiki user guide page.

Meanwhile the fix is as simple as adding 3 to all of the characteristic handles, worked for me at least.

Hope this helps, Ben!

Ben Adler

In reply to Edgars Nemse:

Edgars,

you are a hero! Thank you very much, I'll try this tomorrow and am sure it'll work!

The rest of this message is a rant.

TI, your support is a desaster! I have posted this problem 6 weeks(!) ago and none of you people could be bothered to have a short look to either confirm the problem or even make a suggestion? That's so poor! I thought you're selling these SensorTags to promote your BLE solutions?!

I'm also frustrated because I had to scroll to page 834756389 in this forum to find some frustrated customer suggest you put BLE_CC2540_DeepDive_Training_2011.pdf on the front page IN BIG LETTERS. You still haven't. It is *so* useful, yet hidden so perfectly. That's just stupid. Many people waste a lot of time just because you're not linking to this document and we have to assemble the bits and pieces of this presentation ourselves.

TI, wake up! If there was another company with a similar offering (maybe even using a free compiler!), I'd be gone in a second!

ben

ben wei

In reply to Ben Adler:

Dear Ben

Do you know how to use gatttool read sensortag data such as TI sensortag?

Could you provide me methods?

Thanks a lot

Ben

xie lunzhen2

In reply to Edgars Nemse:

Hi,

Can anyone share how to solve the problem in firmware side?

My setup:

samsung galaxy note3, with andriod 4.42, running the TI BLE Sensortag from the play store

BLE-CC254x-1.4.0

Error code: 3 or 13

Urgent!

thanks

Jason Eller

In reply to Edgars Nemse:

Can someone explain in a little more detail how to change the characteristic handles for the sensors and if that did in fact fix the problem. I am not much of a coder but am trying to get the SensorTag to work with the new firmware using an Android phone.