

# BLE Interface Specification and Simulator Description

| Date    | Version | Editor |
|---------|---------|--------|
| 2.2.21  | V1.0    | Teiimo |
| 8.2.21  | V1.1    | Teiimo |
| 12.2.21 | V1.2    | Teiimo |

## Content

|                                  |   |
|----------------------------------|---|
| 1. Application / Use Case.....   | 2 |
| 1.1. Heating.....                | 2 |
| 1.2. Motion.....                 | 2 |
| 2. The BLE interface.....        | 2 |
| 2.1. General Description.....    | 2 |
| 2.2. Messages.....               | 2 |
| 2.2.1. Heating.....              | 2 |
| 2.2.2. Motion.....               | 3 |
| 3. Interface Simulator.....      | 3 |
| 3.1. Heating.....                | 3 |
| 3.2. Motion.....                 | 4 |
| 3.3. Android Simulation App..... | 4 |

## 1. Application / Use Case

Two applications are considered here, which run on different devices.

### 1.1. Heating

The BLE interface controls the heating level of the jacket. It further returns status information on the current heating level as well as the temperature of the heat pad as well as the total operation time of the product.

### 1.2. Motion

The BLE interface allows to receive the number of steps the user walked as well as his current activity level.

## 2. The BLE interface

### 2.1. General Description

The BLE interface is based on the BLE Gatt protocol<sup>1</sup> and the device is configured as a server in the peripheral mode. The interface supports the BLE 5.0 specification and as such an extended packet length (MTU=247 B), as well as the enhanced data rate of 2Mbit/s. If compatibility with BLE 4.x is required changes in the message length need to be made. For messages that are longer than 20 bytes. In this interface we defined characteristics, which are maximum 20 bytes long.

The security of the connection is set to “just works” to ensure a simple connection in the prototyping stage.

### 2.2. Messages

In the following the messages in terms of service and characteristics for the heating and the motion project will be detailed.

#### 2.2.1. Heating

**Service:** 0x0000fe40cc7a482a984a7f2ed5b3e58f

**Characteristic:** 0x0000fe418e2245419d4c21edae82ed19

**Property:** notify, read, write

|         |       |
|---------|-------|
| byte    | uint8 |
| Heating | count |

**Heating:**

0x00 off  
0x01 33%  
0x02 66%  
0x03 100%

**Count:** Message counter

**Service:** 0x0000fe40cc7a482a984a7f2ed5b3e58f

**Characteristic:** 0x0000fe428e2245419d4c21edae82ed19

**Property:** notify, read

|       |         |       |
|-------|---------|-------|
| int16 | uint32  | uint8 |
| Temp  | op_time | count |

**Temp:** Temperature in °C

**op\_time:** Total time of operation in [min]

**count:** Message counter

<sup>1</sup> <https://www.bluetooth.com/de/specifications/gatt/>; <https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gatt>

### 2.2.2. Motion

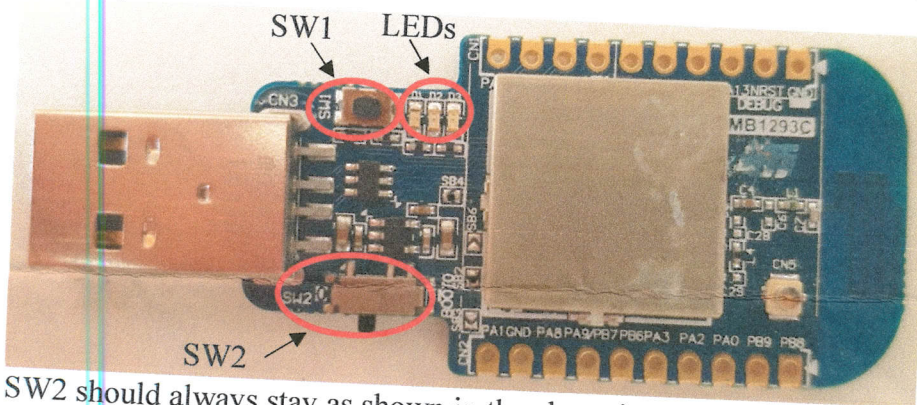
Service: 0x0000fe43cc7a482a984a7f2ed5b3e58f  
Characteristic: 0x0000fe448e2245419d4c21edae82ed19  
Property: notify, read

|          |          |       |
|----------|----------|-------|
| float    | uint32   | Uint8 |
| Activity | nb_steps | count |

**Activity:** Current activity index of the user  
**Nb\_steps:** Number of steps the user did  
**Count:** Message counter

## 3. Interface Simulator

The BLE interface simulator will provide the interface of both devices (Heating and Motion) at once. This way, both interfaces can be simulated with this device. The simulator device has two switches (SW1, SW2) and three LEDs (D1=blue, D2=green, D3=red).



SW2 should always stay as shown in the above image and should not be touched. If SW is moved to the right, the device will enter the update mode and the application will not run. SW1 is used as user switch to switch among the heating levels.

The LEDs are used to show:

1. The BLE activity: Flickering green LED
2. The current heating state:
  - 0% all LEDs disabled
  - 33% blue LED blinks
  - 66% blue and red LED blink
  - 100% red and blue LED on

### 3.1. Heating

#### Button Press:

The user button allows to increase the heating levels step wise. Once 100% is reached, the heating level goes back to 0 by the next press.

#### Heating Level Indication:

The current heating level of the device will be signaled by two LEDs:

- heating=0: 0% all LEDs disabled
- heating=1: 33% blue LED blinks
- heating=2: 66% blue and red LED blink
- heating=3: 100% red and blue LED on



Once the heating level changes based on a BLE request, or a button press, the current heating level will be sent out and the message counter will be increased. If a wrong heating level request is sending (heating>3) the device will go to heating level 0.

### Operation Time:

The operation time of the heating device will be updated once per second. The temperature is simulated by a sawtooth signal, that varies from 20 to 50. The message counter increases by each notification of the message.

**Note:** The message counter is a uint8. So, it can go up to 255. Once this is reached, it restarts by 0. Each update of the message from the device side will increase this counter

### 3.2. Motion

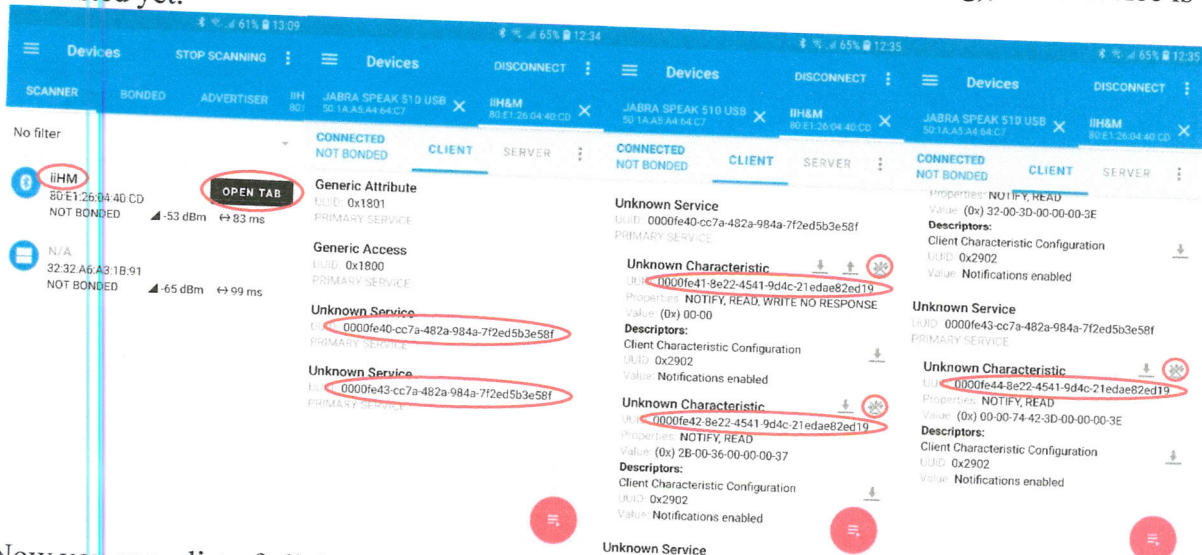
The motion characteristic gets updated each second. The activity is simulated by a sawtooth signal from 0 to 100. The number of steps is increased once per second from 0 to  $2^{32}$ . The message counter gets automatically increased each time a new message is filed.

**Note:** The message counter is a uint8. So, it can go up to 255. Once this is reached, it restarts by 0. Each update of the message from the device side will increase this counter

### 3.3. Android Simulation App

As simulation application on the android phone, we use the *nrf connect* app, that can be downloaded from google play store. The source code of this application can be found under github<sup>2</sup>.

Once the device is connected to USB, the device can be found by the nrf connect app under the name *iiHM*. Just click connect or open tab to connect to the device, or refresh the scan by swiping down or selecting scan on the top right (not visible while scanning), if the device is not listed yet.

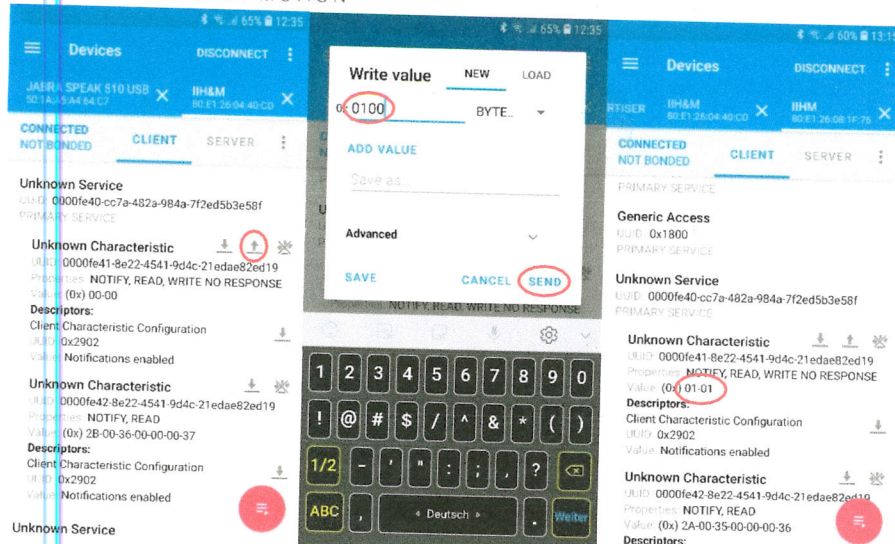


Now you see a list of all the services provided by the device. Click on the two services that are described above and enable the notification for the three available characteristics (click on the three arrows pointing downward). Once activated, the three arrows will be crossed out as shown above.

Now two characteristics will automatically get updated once per second (0000fe42, 0000fe44). You will see this in the value.

The first characteristic (0000fe41) will only get updated once the user has sent a BLE message as below:

<sup>2</sup> <https://github.com/NordicSemiconductor/Android-nRF-Connect>



To send a BLE message to the device just click on the arrow pointing upward. Now enter a heating level and send it to the device. The device will then directly set the heating level and reply with an updated heating message with an increased message counter and the current heating level. As in this case a heating level of 1 was requested, the device replied with 0x0101 as the message counter was increased by 1. Note that, if previously the heating level was modified, the message counter will be different.

The first characteristic (0000fe41) will only get updated if the button SW1 (on simulator device) is pressed.