Using Movesense CustomGATTService

6.11.2018

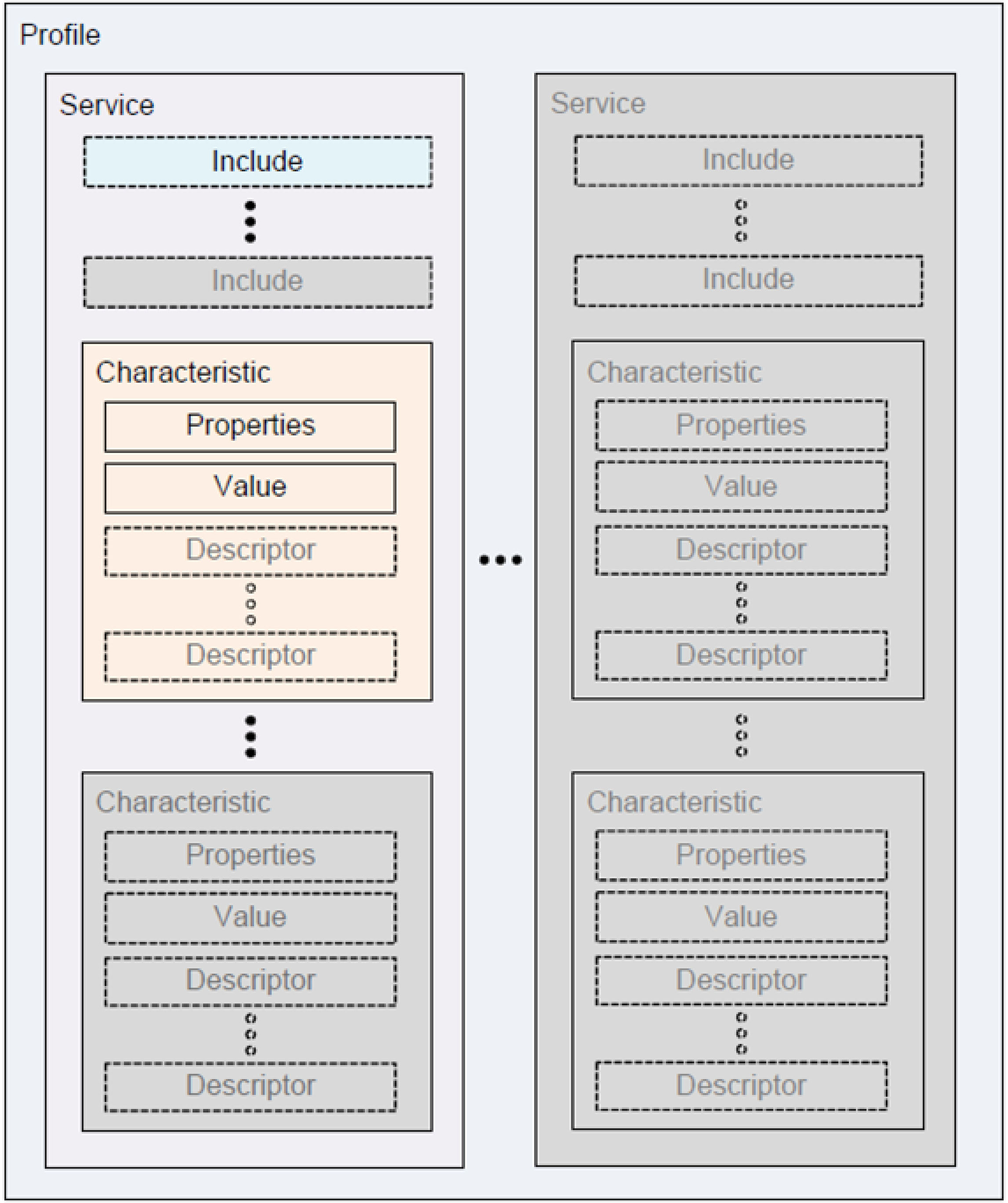
Petri Lipponen

Content

- BLE & GATT
- What is *CustomGATTService*
- Benefits
- Limitations
- How to use it?
- DEMO

BLE & GATT

- GATT: *”The Generic Attributes (GATT) define a hierarchical data structure that is exposed to connected Bluetooth Low Energy (LE) devices.”*
- Services have Characteristics that have Properties & Value
- Value can be up-to MTU-3 bytes. On Movesense sensor the MTU is up-to 158 bytes => max 155 bytes



What is CustomGATTService?

- Movesense core module (optional) that allows creation of a GATT service on the Movesense sensor
- Does not support all the details of GATT, but almost.
- Can be used locally on the device, the mobile (via MDS) and *wbcmd*
- API defined in the *MovesenseCoreLib/resources/movesense-api/comm/ble_gattsvc.yaml*

Benefits

- By-passes Whiteboard communication
  - Much faster connection time since avoids sending WB metadata
  - Get max possible transfer rate
- Accessible from any BLE device
  - Linux, Windows, Embedded devices, Arduino, etc.
- Possible to implement one of standard profiles or a 3rd party GATT API on your sensor

Limitations

- For now the following limitations apply:
  - Max 4 characteristics can be created (in one or more services)
- Only one 128bit base-UUID can be registered
  - Note: That enables >65000 UUIDS
  - Later when WB starts to use the officially registered 16bit service UUID, there will be 3 or 4 available.
- Services cannot be deleted or disabled once created

How to use it? (1/2)

- Create a GATT service with POST: */Comm/Ble/GattSvc*
    - Creates and enables the new GATT service
    - Returns the handle of the new service
    - IMPORTANT! Check for errors in onPostResult!
    - You can see the new service with nrfConnect mobile app
    - Stays there until next reset

- GET: /*Comm/Ble/GattSvc* or /*Comm/Ble/GattSvc/{svcHandle}*
    - Returns GATT service(s) ***created with CustomGATTService***
    - NOTE: Not any other services available on sensor!

How to use it? (2/2)

- For seeing writes to WRITE characteristics:
  - SUBSCRIBE: */Comm/Ble/GattSvc/{svcHandle}/{charHandle}*
  - *Writes will be visible as whiteboard onNotify callbacks*
- *For updating a READ, NOTIFY or INDICATE characteristic:*
  - PUT: */Comm/Ble/GattSvc/{svcHandle}/{charHandle}*
- *For noticing if GATT client enables NOTIFY or INDICATE characteristic notifications:*
  - SUBSCRIBE: */Comm/Ble/GattSvc/{svcHandle}/{charHandle}*
  - *onNotify* event will include a flag that tells if the characteristic notification is enabled or disabled

Questions? Comments?