

## BLE Protocol

The BLE protocol is based on GATT (**Generic Attributes**) and defines the way that two Bluetooth Low Energy devices transfer data using the concepts of **Services** and **Characteristics**.

The data is stored in a simple lookup table with an id for each entry. The protocol defines the READ, WRITE, and NOTIFY actions for these entries.

The protocol defines generic characteristics but also allows us to define private characteristics, which require specific code in the BLE device and the monitor app.

## BLE Pairing

Like standard Bluetooth protocol, BLE requires pairing. But in this case, it is done automatically.

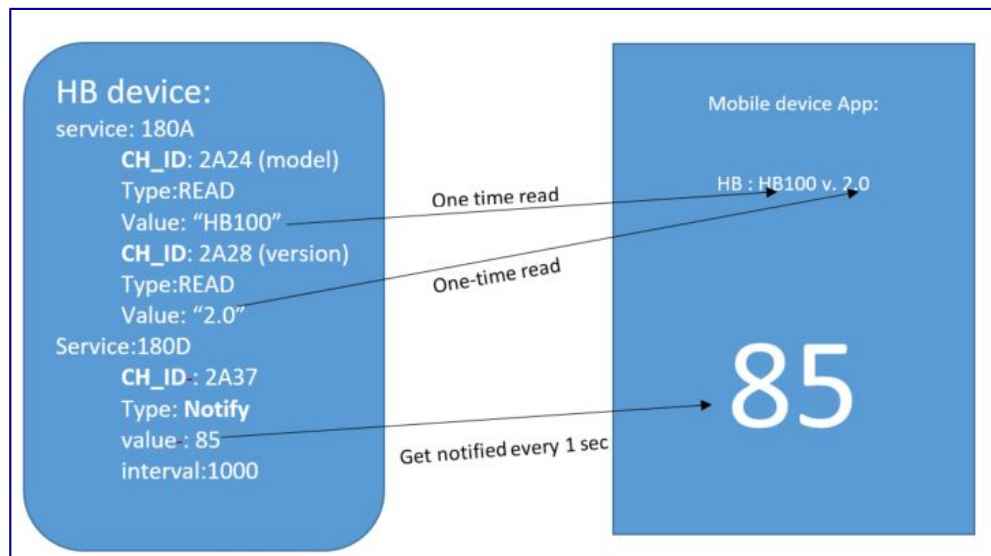
The BLE device broadcasts its service, and the device that receives the broadcast can ask to connect.

Generic devices accept the connection, whereas other devices will not accept the connection without an authentication handshake mechanism, based on setting predefined private characteristics with a unique key.

After pairing, the devices can exchange data based on the characteristics and the allowed actions using the UUID of the table entries.

## Data Flows Over BLE Protocol

The data flow is based on the actions and the characteristics values. For example, here are the heartbeat devices and mobile device data flow (after pairing):



## IoT Testing BLE Challenges

There are a large number of challenges to developing and testing IoT mobile applications:

- Most of the tests have to be done manually and require some form of connection between the IoT device and a real person.

- The IoT and mobile devices should be in the same location, with a manual interaction to connect between them.
- It takes more time to develop the IoT HW device, and you cannot parallel the application (SW) and the IoT device's development and testing.
- The testers cannot control the data sent from the IoT device to the mobile app and set any predefined validation.
- Health devices and applications require FDA approval. That means large numbers of proven tests on a large number of devices.
- In the case of offshore testing, it is always a challenge to send and manage the devices at remote sites.

Let's take an example of a heartbeat device connected to a finger and send the heartbeat to a mobile app.

The app/device tests required:

- Person/tester (With heartbeat and finger, preferably)
- Mobile devices
- IoT device



This process cannot be automated – it cannot be scaled.

## IoT Testing for BLE

Our solution goals are to:

- Automate the test process.
- Access all the mobile devices in the cloud.

The solution is a SW visualization of BLE devices in the cloud.

It allows the user to define and update the characteristics and the services. The services support APIs and are controlled by automation scripts.

At Perfecto, we selected this solution for the following reasons:

- It is fully controlled by SW – key for automation.
- It is located in the cloud near the devices.
- It acknowledges generic characteristics and can be virtually any device
- Multiple threads
- Works on real RF channels
- Allows data to be set by the tester
- Connects to any automation framework.

## How Does It Work?

A new BLE server is added to the lab. This machine contains up to 20 BLE dongles. For each dongle, the machine can set a virtual BLE device. The automation server controls both the IoT and mobile devices and allows us to develop a full scenario in an automation script.



Each BLE service is configured by an external JSON file. The file defines the services. Based on the JSON file, a new virtual BLE service started and exposes **BLE** data.

```

[
  {
    name : "HB_502021179",
    services : [
      {
        name : "HB Service", // D:
        uuid : 46a970e00d5f11e28b5e0002a5d5c51b,

        characteristics : [
          {
            name : "ch1",
            uuid : 0aad7ea00d6011e28e3c0002a5d5c51b,
            value : [ 0x0a4a2000, 0x0F00D660, 0x0048
            ],
            properties : [notify],
            interval : 10000
          },
        ],
      },
    ],
  },
  {
    name : "Device Information",
    uuid : 180A,
    characteristics : [
      {
        name : "Manufacturer Name",
        uuid : 2A29,
        value : "HB ma",
        properties : [read]
      },
      {
        name : "Model Number",
        uuid : 2A24,
        value : "Model 3230",
        properties : [read]
      },
    ],
  },
]

```

The mobile device receives the data over the RF channel (standard BLE HW on a phone), and it works like it arrived from real BLE devices.

The last part of the solution is the commands, which are added to the automation system and allow us to start, stop, and change BLE servers and data from the script.