**Bluetooth Low Energy (BLE)**



*By: Johnson Domacasse (#4471709)*

*13 March 2024*

Contents

[1. Introduction: 3](#_Toc160658707)

[2. Bibliography: 3](#_Toc160658708)

# Introduction:

Bluetooth is a wireless technology standard for the exchanging data over short distances. Just like Wi-Fi, Bluetooth also works on 2.4 GHz. It is used in many different application that require wireless communication and control. Like transmitting audio to headphones or a car. Devices use Bluetooth in a point-to-point communication to **continuously** transmit data. This is not optimal for power usage. Bluetooth low energy (BLE) is a power-conserving alternative to this. Its primary application is short distance transmission of small amounts of data. Unlike Bluetooth that is always on, BLE is always in sleep mode constantly, unless a connection is initiated. This conserves power. This technology is extremely useful in machine to machine communication.

The aim of this challenge is to give us a better understanding of how this technology works.

Some important terminology will be discussed, such as UUID, characteristics services etc**[1].**

A diagram of a battery level

Description automatically generatedAn example of these can be seen in the figure on the right. Here we can see the **profile** of the device labelled as the “Bluetooth device”. The **service** that this device provides is the battery service. The **characteristic** of this service is the battery level. Each of these (except for the **profile**) have a predefined UUID. It is used for uniquely identifying information. For example, the service that is provided by a specific device**[1].**

# BLE Scanner:

We begin by setting up the environment we will be working with. Then the Arduino BLE library is added to the project so that we can work with the technology. Then we use the already made example and add it into our main file.

Finally we take a look at the serial monitor and there we can already see some devices in the area that are running on BLE. See example below.

A screenshot of a computer program

Description automatically generated

Here we can see the addresses of the different devices, some of them have local names assigned to them. Others have service UUIDs. What they all have in common is that each one has an RSSI value. Which is simply a measure that represents the relative quality level of the Bluetooth signal received on the device. The signal is better when this value is closer to zero. So a pretty good value is -50 or below. A reasonable value is between -70 and -80 while -100 indicates no signal at all **[2].**

Another fun detail we can notice is that using the assigned numbers document, we can determine that the service the last two devices provide is from Google LLC (0XFE9F)**[3].**

**Note: I tried uploading the .ino file to the board. Although it did compile and upload, the serial print didn’t seem to work, even after adding the Arduino header file.**

# BLE Server:

# Bibliography:

**[1] -** Santos, R. (2019, June 4). *Esp32 Bluetooth Low Energy (BLE) on Arduino Ide*. Random Nerd Tutorials. <https://randomnerdtutorials.com/esp32-bluetooth-low-energy-ble-arduino-ide/>

**[2] -** Li, M. (2023, March 16). *Understanding the measures of bluetooth RSSI*. MOKOBlue. <https://www.mokoblue.com/measures-of-bluetooth-rssi/>

**[3] -** Assigned numbers | bluetooth SIG. (n.d.-a). <https://www.bluetooth.com/wp-content/uploads/Files/Specification/HTML/Assigned_Numbers/out/en/Assigned_Numbers.pdf?v=1705536000119>