KOÇ UNIVERSITY

College of Engineering

ELEC 491 – Electrical Engineering Design Project

Final Report

Bluetooth 5.0
Car-to-Car Communication with OBD-II

Participant information:

Berkan Hiziroglu
Cem Ergin

Project Advisor:
Akın İdil
Hakan Urey

7 Jan 2018

Abstract

Since the discovery of radio, wireless communication technologies are tremendously changing the way we live and the way we interact with our environment. Thanks to the discovery of new and better tools like cellular networks, Wi-Fi and Bluetooth; the world today is a more connected place than it has ever been. Interestingly, these approaches to connectivity that are shaping today's personal electronics and household products, did not penetrate well to some industries; like the automobile industry, due to their limitations.

In the wake of the development of these technologies that offer more efficient, superior connectivity, this project aims to explore how we can incorporate emerging connectivity tools; namely the newly released Bluetooth 5.0 standard which offers greater connection range and throughput compared to its predecessors, to establish connection between moving vehicles aiming to provide motorists with a safer and more pleasant ride. It's two main objectives are establishing a channel for verbal communication between fellow motorists and to make sharing on-board diagnostics data; such as current speed, fuel level or engine status between their vehicles possible.

Table of Contents

Introduction

i. Concept

For the last couple decades, the issue of connectivity has become more and more important for vehicle owners and drivers. Even when connectivity was a smaller problem globally, motorists still went through great lengths to stay connected as they drive, using citizens band radios and humongous car phones. [1] Today car manufacturers are looking to build even better connected cars with integrated Bluetooth and infotainment services but still, most of connectivity desired within vehicles is provided by cellular phones' of the drivers and the passengers. [2]

Although these cellular networks and smart phones offer a great deal of connectivity, they still display shortcomings in various moments such as when the batteries of cellular phones run out or when the cellular connection is not strong enough within the region. In this kinds of situations users can go from having a ton of connectivity to having no connectivity at all in an instant. Additionally, these widespread tools are not specifically designed to provide inter-vehicular connectivity which means that if a driver wants to communicate with another driver driving on the same road with him, he or she still needs to take employ of non-verbal methods such as signalling, waving or honking.

We believe that, as autonomous vehicles and smart transportation is becoming a part of our reality more and more [2], standalone inter-vehicular communication technologies will play an important part in providing motorists and passengers better connectivity while making transportation safer. Our project can be considered as an exploration of this idea and ways we can provide such connection with the tools and technologies that are available today.

ii. Objectives

Fundamentally, this project aims to explore ways we can exploit new communication technologies; in this specific case Bluetooth 5.0 standard, to establish inter-vehicular communication to provide drivers and passengers with safer, more pleasant transportation experiences. To explore this idea, we have picked to distinct features as our main objectives: to establish a communication channel between motorists that they can use to communicate verbally and to enable sharing of on-board diagnostics information between vehicles. In short, this project focuses mostly on establishing Bluetooth connections between various devices and using these connections to send and receive different kinds of data. Therefore using basic Bluetooth protocols and architecture were an integral part of our work.

Displaying our exploration and the capabilities we discover was also an important part of our project. Therefore we've also worked to develop an iOS application that enables iPhones to connect to our simple Bluetooth 5.0 network and send data to other devices.

iii. Background

Ericsson company was working on developing a wireless headset for cell phones as early as 1994 but it wasn't until late 2001 that the wireless communication technology that they developed was ready to be shared with the world. [9] Since then, Bluetooth has become a stable part of personal electronics providing short-range communication between cell phones, headphones, laptops and various electronic products and devices. Over the years, Bluetooth technology has been altered and updated to suit the needs of new products and modern consumers. Currently, Bluetooth 4.x Standard is its most popular version.

Around December 2016, the new Bluetooth 5 Protocol which can be considered a huge leap forward from the earlier version, was announced by Bluetooth SIG and made available to development shortly afterwards. Bluetooth 5.0 provides a greater throughput and longer range compared to its predecessors and extends the use of Bluetooth to newer applications. Bluetooth SIG, claims that this new standard quadruples the range of Bluetooth. [11] Texas Instruments' engineers confirm this claim by establishing a connection over 1 km; which is way above the maximum range of earlier standards. [8] In summary, new developments in Bluetooth technology; especially its greatly extended range is the main reason we chose Bluetooth 5.0 to implement our inter-vehicular communication solution.

System Design

i. System Overview

We were forced to use PC to design and implement code for our LaunchXL CC2640R2F MCU's since its' Bluetooth software development kits were only available for Windows. We've used Texas Instrument's own integrated development environment; Code Composer Studio, to design and test any code we have loaded into CC2640R2F devices. The coding language used to program CC2640R2f devices was C/C++. On the other hand we've done all of the development for our iOS application on Mac laptops. We've used Apple's integrated development environment Xcode along with its programming language of choice Swift, to design our iOS application. We've used all these tools to establish Bluetooth connection and used it to send and receive data between iPhones. You can find the block diagram of the network below.

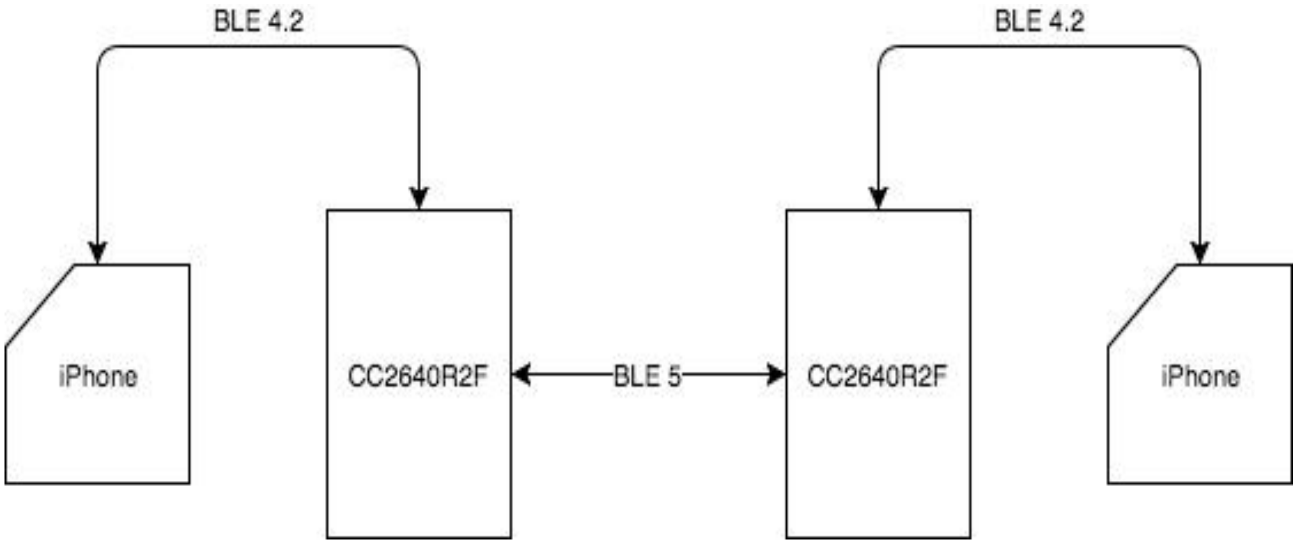

Figure 0: Block Diagram of Prototype

As seen on the diagram above, we connected iPhones to CC2640R2F devices using Bluetooth 4.2 connection and used Bluetooth 5.0 to connect CC2640R2F with each other which enables us to extend the range of this simple network way above the maximum range of Bluetooth 4.2 standard.

ii. Software Design

```
/*********************************************************************
* @fn      multi_role_processCharValueChangeEvt
*
* @brief   Process a pending Simple Profile characteristic value change
*          event.
*
* @param   paramID - parameter ID of the value that was changed.
*
* @return  None.
*/
static void multi_role_processCharValueChangeEvt(uint8_t paramID)
{
  uint8_t newValue;
  // Print new value depending on which characteristic was updated
  switch(paramID)
  {
  case SIMPLEPROFILE_CHAR1:
    // Get new value
    SimpleProfile_GetParameter(SIMPLEPROFILE_CHAR1, &newValue);
    Display_print1(dispHandle, MR_ROW_STATUS2, 0, "Char 1: %d", (uint16_t)newValue);
    //multi_role_performPeriodicTask();
    br_doGattRw2(0,newValue);
    break;
  case SIMPLEPROFILE_CHAR3:
    // Get new value
    SimpleProfile_GetParameter(SIMPLEPROFILE_CHAR3, &newValue);
    br_doGattRw2(0,newValue);
    //multi_role_performPeriodicTask();
    break;
  case SIMPLEPROFILE_CHAR4:
      Display_print1(dispHandle, MR_ROW_STATUS2+1, 0, "Char 4: %d", (uint16_t)newValue);
      break;
  default:
    // Should not reach here!
    break;
  }
}
```

Above is the code from the multi role project. Multi role projects enables TI devices to choose between being a central and a peripheral. This functionality is the basis for our bi-direction property in our system.

The processCharValueChangeEvt() function is the delegate function for property change events for characteristics in a bluetooth device. This function is called whenever a property is written in the device either by a connected device or by its own. We used the first characteristic as the loop-back and the third characteristic as our data writing characteristic. Whenever a device writes something to the first characteristic of a TI device, it automatically sends that data to the other connected TI device via br_doGattRw2() function. This is the most time consuming part of our system. To send a data from one device to another, TI performs some checks and optimization since it’s main objective is to use less energy, not faster data transmission rate. These checks delay our data transmission for some milliseconds. However, these delays make a significant amount of time when combined at the whole course of a transmission.

The last part of our system, between the second TI device and the mobile device, uses a different transmission system than the previous parts. The first mobile device uses GATT write methods to send the data from its recorder to TI. Below is the first part of our system visualized.

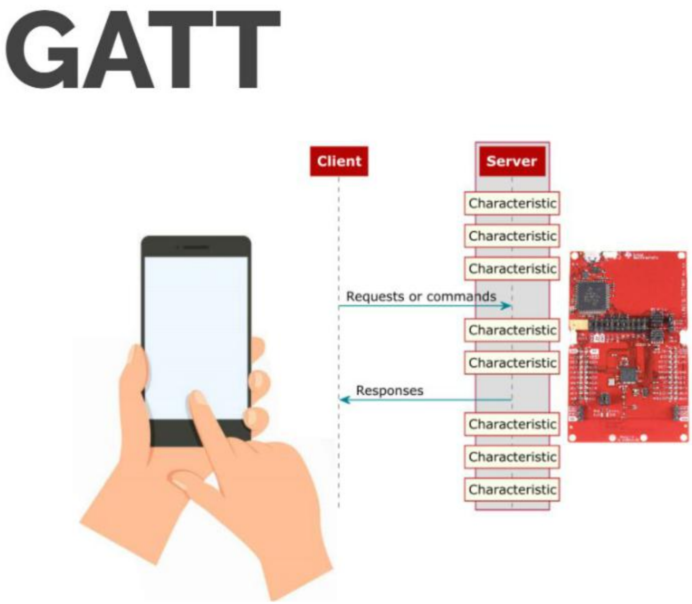

Figure 1: The Generic Attributes scheme

GATT stands for “The Generic Attributes” and it is a hierarchical data structure that is used on Bluetooth Low Energy (LE) devices. It has many functionalities but we mainly used READ and WRITE attributes in our project. As explained above, the first part uses GATT WRITE function to send the data from the mobile device to first TI device.

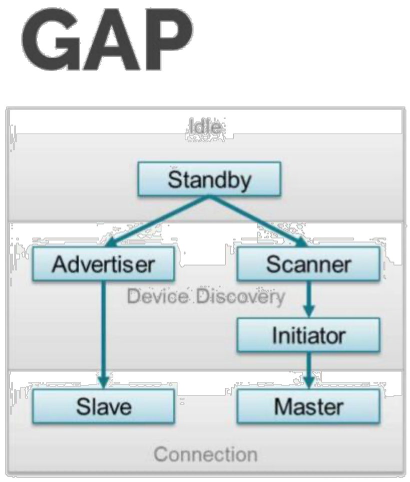

Figure 2: Generic Access Profile scheme

GAP stands for “Generic Access Profile” and is used to establish connection between two BLE devices. A connection can be established between two Bluetooth devices if and only if they conform to the Bluetooth protocol. This means that, they should follow GAP and perform a handshake with each other, so that the connection can be established.

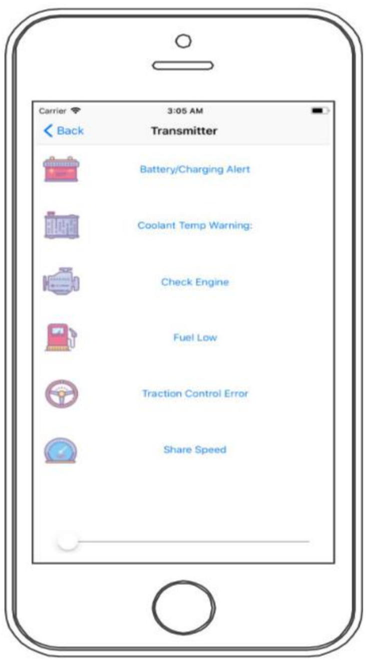

Figure 3: Application screen

iii. Application

We designed our system to be used on Apple iOS devices. The user can download the app to their iPhone device and directly use it. The first screen lets the user select the current TI - BLE devices in the range. The second and the third screens are shown below.

In the second screen (Application screen 1), there are two text views. The first one is the text version of the incoming voice data. Second one is for the user to record their voice and send it to the other party.

Application screen 2 is the emulator for on-board diagnostics data. It lets the user inform the other parties instantly about their current speed or any other crucial data. Since these diagnostics can be represented as a single integer, their transmission rate is much faster than voice data, specifically, less than 10 milliseconds.

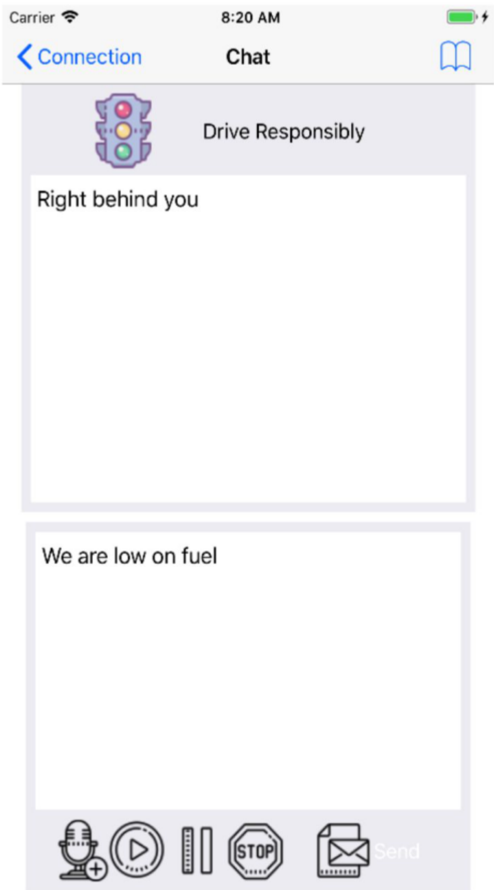

Figure 4: Application screen 1

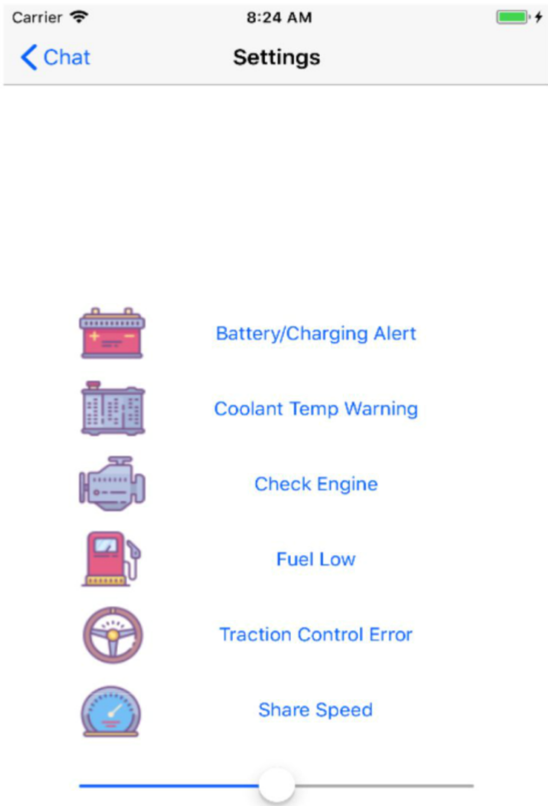

Figure 5: Application screen 2

Analysis and Results

Looking at it in a general perspective, we have met most of the goals we have set. Our first task was to transfer data between to TI devices using Bluetooth 5.0 connections. We successfully sent data using Bluetooth between TI devices that using Code Composer Studio [3], TeraTerm [4] and TI - CC2640R2 devices [5]. Our second task was to enable iPhone's to communicate with TI devices over Bluetooth 4.2 connections. Using XCode [6] and two iPhone 7 devices, we established connections between iPhones and CC2640R2Fs using the aforementioned tools, C and Swift codes.

Later we combined these singular connections to form a simple network between all four devices and successfully transferred data; simple integers; from one iPhone to the other using this network. We then developed an iOS application to act as an on-board diagnostic error generator and display our findings along with the error messages received from the other device. At this point, we have proven our concept of utilizing Bluetooth 5.0 connections to provide inter-vehicular connections.

In the next phase, we focused more on providing verbal communication between users. Our ultimate goal was to upgrade the previous network to provide a bi-directional audio communication to users. Unfortunately, some of the parts we required to implement this system without iPhones were not supplied by any providers in Turkey, so we extended our iOS application to help us present sound recording and playing capabilities. We've used this setup to communicate but due to the implementation of the previous simple network we couldn't reach the throughput we needed to send audio. The data amount was 1 byte (8 bits) per message with about 10 milli seconds of single transmission time which suggests that we were capable of transferring 100 bytes of data every second. Unfortunately, this size was not enough to even send sound files over a reasonable amount of time. Therefore, we incorporated speech recognition and text-to-speech tools in our iOS application to be able to send text data rather than audio data over our network. This way, we were able to send and receive long texts in English with only reasonable delays.

Results

Below is the system we have constructed. The four components of our system are connected to each other at the same time. We have managed to send a character every 0.4 seconds. This is an enough transmission rate for successfully sending on-board diagnostics data along with simple text. It wasn't not enough for a fast voice communication, so we converted the input speech to text before transmitting. After the last mobile device receives the text data, it converts the text to speech and the driver can still get the information. This conversion sped up the process to provide the fast transmission of the voice data. Overall, although this project has its minor flaws, we have successfully managed to provide car to car communication via Bluetooth 5.0.

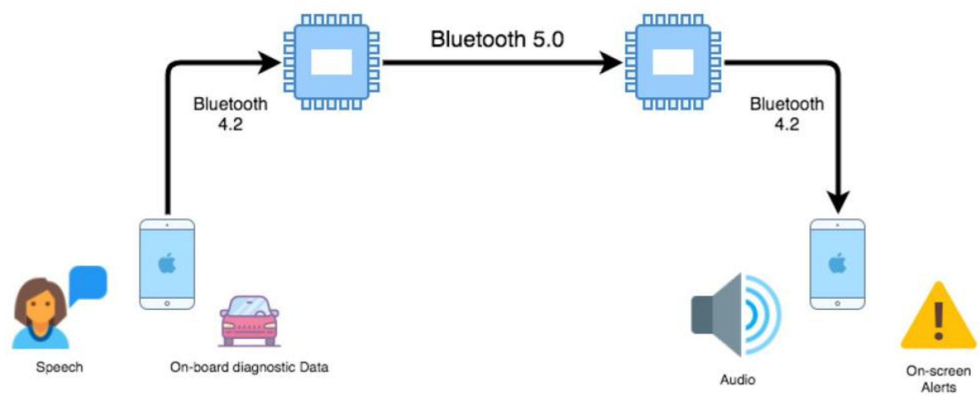

Figure 6: Project diagram

Conclusion

As mentioned earlier, this project is focused on finding solutions to the problem of inter-vehicular communication which is an interesting issue that requires both hardware innovations and software support to be solved. To provide a standalone and efficient inter-vehicular communication, we proposed a system that uses Bluetooth 5.0 to provide verbal communication and data sharing capabilities to motorists.

In its current form, our prototype solution is able to establish a Bluetooth 5.0 connection between Texas Instruments' LaunchXL MCU's and allows them to send and receive data between each other. Furthermore, we've created an iOS application that enables iPhones to connect to these devices and to send & receive information. We are currently able to use this network of 4 devices; 2 LaunchXL's and 2 iPhones, to send and receive text data and emulate on-board diagnostics alert messages within the iOS application.

Even though our prototype cannot currently send and receive simultaneous bi-directional audio data as we've first imagined, it has proved that Bluetooth 5.0 can be used successfully to establish standalone inter-vehicular communication. Therefore, we believe that we have successfully explored the capabilities of Bluetooth 5.0 in a inter-vehicular application and our experiences will serve as an example to students and engineers in our school who are interested in new Bluetooth technologies.

Design improvements

For now, our system includes two peripheral mobile devices, one peripheral TI device and one central TI device. The connection between four of them is stable, connection is re-established in case of a disconnection. The system supports simple bidirectional data transmission. Each mobile device can send and receive data to the other as they wish.

Unfortunately, our current system is unable not provide enough throughput and data transmission, hence it practically cannot provide smooth voice communication over BLE right now. Instead, it uses a speech to text library to convert the speech into text, transmit it to the other mobile device via BLE and then use a text to speech library to convert the received text into audible sound. With this approach, the long wait time from sending voice data is reduced drastically while the overall concept is altered only slightly.

However, one should note that, our current system is capable of sending voice data or any data for that matter, and can theoretically enable voice communication over BLE. Unfortunately, its current implementation causes the transmission of large files to happen in a timely manner, which makes it unsuitable for practical use. We plan to enhance our system; which currently acts as a proof to our concept, by decreasing the transmission time, increasing the throughput of data and allowing the easy usage of our system for users.

As we increase the transmission rate, we also hope to try some experimental new concepts, such as transferring a video or streaming a live audio/video session through BLE along with creating mesh network type structures to enable sharing of vital information on a larger scale.

Even though our design delivers a complete BLE experience, currently, it is incapable of providing the swiftness that is essential to today's electronic devices. We plan to develop our prototype further, aiming to send any kind of data and fast. Our plan is to approach this issue by doing an optimization to our code and creating custom BLE profiles rather than using the built-in profiles and implementations.

References

1. “How Connectivity is Driving the Future of the Car,” Wired, 10-Aug-2016. [Online]. Available: https://www.wired.com/brandlab/2016/02/how-connectivity-is-driving-the-future-of-the-car/
. [Accessed: 07-Jan-2018].
2. K. Moinzadeh, “Vehicle Connectivity: Paving the Road to the Future of Transportation.,” Entrepreneur, 29-Oct-2017. [Online]. Available: https://www.entrepreneur.com/article/298321. [Accessed: 07-Jan-2018].
3. Texas Instruments, “Code Composer Studio (CCS) Integrated Development Environment (IDE)”, http://www.ti.com/tool/CCSTUDIO. [Accessed: 07-Jan-2018]
4. TeraTerm, https://ttssh2.osdn.jp/index.html.en [Accessed: 07-Jan-2018].
5. Texas Instruments, “SimpleLink™ Bluetooth® low energy CC2640R2F wireless MCU LaunchPad™ development kit”, http://www.ti.com/tool/LAUNCHXL-CC2640R2. [Accessed: 07-Jan-2018]
6. Apple, “XCode”, https://developer.apple.com/xcode/. [Accessed: 07-Jan-2018]
7. Apple, “iPhone”, https://www.apple.com/lae/iphone/. [Accessed: 07-Jan-2018]
8. “How does Bluetooth® 5 increase the achievable range of a Bluetooth low energy connection?,” TI E2E™ Community. [Online]. Available: http://e2e.ti.com/blogs_/b/connecting_wirelessly/archive/2017/01/30/how-does-bluetooth-5-increase-the-achievable-range-of-a-bluetooth-low-energy-connection. [Accessed: 07-Jan-2018].
9. D. Tonner, “The Bluetooth Blues,” Information Age, 24-May-2001. [Online]. Available: https://web.archive.org/web/20071222231740/http://www.information-age.com/article/2001/may/the_bluetooth_blues . [Accessed: 07-Jan-2018].
10. Rohde&Schwardz, “From cable replacement to the IoT Bluetooth 5 White Paper”, https://bluetoothworldevent.com/__media/PDFs/Rohde--26-Schwarz_3e_Bluetooth_WhitePaper.pdf. [Accessed: 07-Jan-2018].
11. “Bluetooth® 5 Quadruples Range, Doubles Speed, Increases Data Broadcasting Capacity by 800% | Bluetooth Technology Website,” Bluetooth. [Online]. Available: https://www.bluetooth.com/news/pressreleases/2016/06/16/-bluetooth-5-quadruples-rangedo ubles-speedincreases-data-broadcasting-capacity-by-800. [Accessed: 07-Jan-2018].

Appendices

Hardware : TI - CC2640R2F

BLUETOOTH 4.2	BLUETOOTH 5.0
Speed: 1 Mbps	**Speed:** 2 Mbps
Range: 50 meters outdoor, 10 meters indoor	**Range:** 200 meters outdoor, 40 meters indoor
Power Consumption: High	**Power Consumption:** Low

Source: http://www.rfwireless-world.com/Terminology/Bluetooth-5-vs-bluetooth-4-2.html