5-2016

Bluetooth Low Energy Platform with Simblee

Calvin W. Freeman
University of Arkansas, Fayetteville

Bluetooth Low Energy Platform with Simblee

An Undergraduate Honors College Thesis

in the

Department of Computer Science and Computer Engineering
College of Engineering
University of Arkansas
Fayetteville, AR
May 2016

by

Calvin Wade Freeman

This thesis is approved.

_____
Dr. Parkerson
Thesis Director

_____
Dr. Thompson
Committee Member

_____
Dr. Li
Committee Member

ABSTRACT

Bluetooth Low Energy provides a platform for many developers to implement low power communication for a wide range of applications. The Internet of Things (IoT) is an emerging concept that is gaining traction in the world of embedded systems. Bluetooth Low Energy is one of the latest enablers of this movement and the Simblee module improves BLE accessibility. This thesis provides an evaluation of the average power consumption the Simblee uses in different use cases. By leveraging one of the Simblee's most notable features, the Ultra Low Power mode, very low power consumption can be achieved, potentially increasing battery life from weeks to years. The Simblee module can be programmatically put into Ultra Low Power mode in intervals during normal execution to achieve different levels of power consumption at the cost of increased latency. Depending on the use case, low latency can be sacrificed in favor of lower power consumption.

Table of Contents

1. INTRODUCTION 1

1.1 Problem 1

1.2 Objective 2

1.3 Approach 2

1.4 Organization of this Thesis 2

2. BACKGROUND 4

2.1 Key Concepts 4

2.1.1 The Internet of Things (IoT) 4

2.1.2 Bluetooth Low Energy (BLE) 5

2.1.3 GAP 6

2.1.4 GATT 8

2.1.5 Simblee 10

2.2 Related Work 13

2.2.1 Power Consumption Analysis of Bluetooth Low Energy, ZigBee and ANT Sensor Nodes in a Cyclic Sleep Scenario [11] 13

3. ARCHITECTURE 14

3.1 High Level Design 14

3.2 Tools and Documentation 15

3.3 Implementation 15

4. METHODOLOGY, RESULTS AND ANALYSIS 18

4.1 Methodology 18

4.2 Results 19

4.3 Analysis 23

5. CONCLUSIONS 24

5.1 Summary 24

5.2 Contributions 24

5.3 Future Work 24

REFERENCES 26

LIST OF FIGURES

Figure 1: IoT Value Chain [4] 4

Figure 2: Peripheral advertising to several central devices [8] 7

Figure 3: BLE Advertising Process [8] 7

Figure 4: Connected Central and Peripheral devices [8] 8

Figure 5: Communication during connection [8] 9

Figure 6: BLE Data Structure [8] 9

Figure 7: Block Diagram of Simblee [10] 11

Figure 8: Bidirectional communication between Simblee and phone/tablet [11] 12

Figure 9: Ad-hoc communication between modules and phone/tablet [11] 12

Figure 10: Simblee Mobile Application Functionality 14

Figure 11: Advertisement Interval Code 16

Figure 12: Simblee Mobile Application 17

Figure 13: Advertising Interval 20ms-900ms 19

Figure 14: Advertising Interval 1000ms-10000ms 19

Figure 15: Connection Interval 200ms-1100ms 20

Figure 16: ULP Sleep Cycle 100ms-900ms 21

Figure 17: ULP Sleep Cycle 1000ms-10000ms 21

Figure 18: Advertising Interval - Battery Life 22

Figure 19: ULP Sleep Cycle - Battery Life 22

1. INTRODUCTION

1.1 Problem

Bluetooth Low Energy (also known as Bluetooth Smart, Bluetooth 4.0, or BLE) was created to address the problem of high energy consumption that the previous Bluetooth specifications suffered from. It also aimed to provide lower cost and to maintain a similar communication range compared to previous Bluetooth specifications. BLE specifically targeted applications within the realm of health, fitness, sports, and home automation while leveraging the mobility of tablets and smartphones. [1]

Many modern day applications have been made possible with harnessing low energy consumption that BLE provides. Examples include: enhanced prosthetics, Fitbit, Jawbone products, and iBeacon technology. [2] Even though Bluetooth Low Energy offers inherent improvements in energy consumption, it's up to the programmer to implement the applications correctly to completely harness BLE's capabilities of low energy consumption. Several different parameters can be tuned within a BLE application to achieve very low power consumption, but changing these parameters often come with trade-offs. Many BLE devices feature a sleep mode that uses incredibly low power consumption.

The Simblee module offers two low power states: Ultra Low Power (ULP) mode and off. Determining when to enter ULP mode, how often to enter ULP mode, and how much energy is saved is often difficult. Putting the Simblee in ULP mode in different intervals is one effective method in saving power. There are also other ways to save energy by changing how often the

Simblee advertises and changing the connection interval. This thesis looks at the effect that changing these parameters has on power consumptions. By defining clear data of how these parameters affect power consumption, this will help future developers make more informed decisions when programming with the Simblee module.

1.2 Objective

In order to fully leverage the Simblee's capabilities to save energy, the programmer must implement and tune certain parameters that the Simblee offers. By tuning the advertisement intervals, connection intervals, and implementing ULP sleep cycles, the Simblee can drastically reduce its overall power consumption, in turn increasing battery life.

1.3 Approach

To accurately test the Simblee under several different use cases, test applications need to be written to evaluate the Simblee. The main parameters the Simblee API provides to tune is the advertisement interval, the connection interval, and the transmit power. A ULP sleep cycle can also be implemented within the code to provide lower energy consumption. In some situations, the Simblee can operate on as little as 5 µA. Different cycle delays can be tested to evaluate how well each interval saves energy. Tuning these parameters can enable the Simblee to improve battery life time of an average coin cell from a few days to a few months.

1.4 Organization of this Thesis

The organization of this thesis is as follows. Chapter 2 covers the background and key concepts the reader needs to know to fully understand the research. The background covers what IoT is and how it's important to today's society. It also defines what Bluetooth low energy is and

why it was initially created. A few BLE specific topics are covered, and finally a brief overview of the Simblee module is given.

Chapter 3 goes over the project itself, gives a high level design, and some technical implementation details. It provides a high level diagram of how the mobile application works and also briefly discusses the advertisement interval application. It also gives an overview of the tools used to develop the testing applications and to test the Simblee.

Chapter 4 reviews the results collected from the testing and offers an analysis of the results. It also covers how the results were collected. Several graphs with numerical analysis are given to help understand the data presented.

Chapter 5 wraps up the thesis by concluding with a few final thoughts. It gives a general review of thesis as a whole and why this research is important. This chapter also gives a few directions to build off this research in the future.

2. BACKGROUND

2.1 Key Concepts

2.1.1 The Internet of Things (IoT)

The emerging concept of The Internet of Things focuses on the vast network of connected physical devices. With the increased availability of cloud computing, the low energy cost of operation, and the decreasing size of devices and chips, the magnitude of IoT is becoming greater every day. A few examples of technologies that are enabling IoT are RFID and near-field communication, optical tags and quick-response codes (QR codes), and Bluetooth Low Energy. [3]

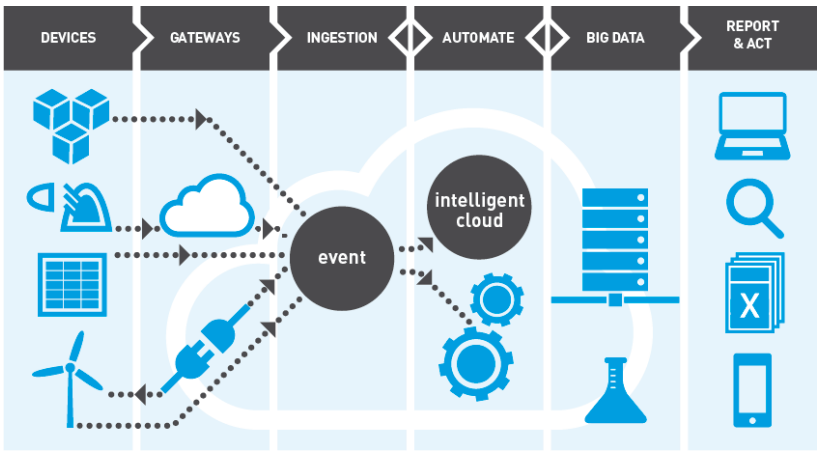


Figure 1: IoT Value Chain [4]

IoT consists of devices such as wearable technology, smartphones, tablets, and various sensors. IoT even goes beyond that with businesses using it within their own specific business models or even shifting their business models using the power of IoT. Companies that were previously focused on selling products are now shifting focus to providing a service with their products. One example includes jet engine manufacturers; instead of selling their product they

can rent their engines and charge airlines a specific rate depending on the engine load and time used. IoT provides the technology to monitor the connected engine components to make better engineering decisions when manufacturing. The benefits of IoT also include enhanced customer service, increased revenue from services and/or products, improved use of assets in the field, and more information to feed big data and information analytic efforts. [5]

Even though IoT can benefit companies and the world in several different areas, there are many challenges that need to be addressed and overcome. A few of the biggest concerns are ensuring data privacy, effectively using the data acquired by IoT, and managing the large influx of data created by IoT.

2.1.2 Bluetooth Low Energy (BLE)

As mentioned in the previous section, BLE is one of the enablers of IoT. Bluetooth Low Energy was created to address the problem of high energy consumption of the earlier generations of Bluetooth and is found in a wide variety of devices. Specifically, BLE operates a peak current consumption of approximately 15 mA whereas Bluetooth Classic operates at a peak current of approximately 30 mA. Low energy consumptions is not the only feature it improves upon, BLE also boasts lower implementation costs, mutli-vendor interoperability, and enhanced range. Both lowering the implementation costs and enabling BLE applications to be developed for several different vendors simultaneously makes BLE much more accessible. Enhanced range is also useful for when sensors are located where power and access are limited. BLE is able to operate at a range of up to 250 meters. [6]

Another detail to address concerning BLE is that it is a subset of Classic Bluetooth, meaning that it is not backwards compatible with Classic Bluetooth devices. However, many Bluetooth modules and devices include dual-mode operation: they can both operate as a

Bluetooth Classic device and a BLE device. Devices that only operate as a BLE device are called single-mode devices.

2.1.3 GAP

In order to understand how a BLE device establishes a connection with another device, clear device roles must be defined. Peripheral devices are traditionally low power, resource constrained BLE devices that connect to a more powerful central device. Central devices possess greater processing power and memory, which is usually a tablet or a smartphone. The first step in establishing a connection between a BLE device and a central device (such as a phone or tablet) is the advertising process. The Generic Access Profile (GAP) controls the connections and advertisement within BLE. This allows the BLE device to be visible to the outside world and determines how the central and peripheral devices communicate with each other. When advertising, a peripheral device can send advertising data to several different central devices. Figure 2 illustrates this concept.

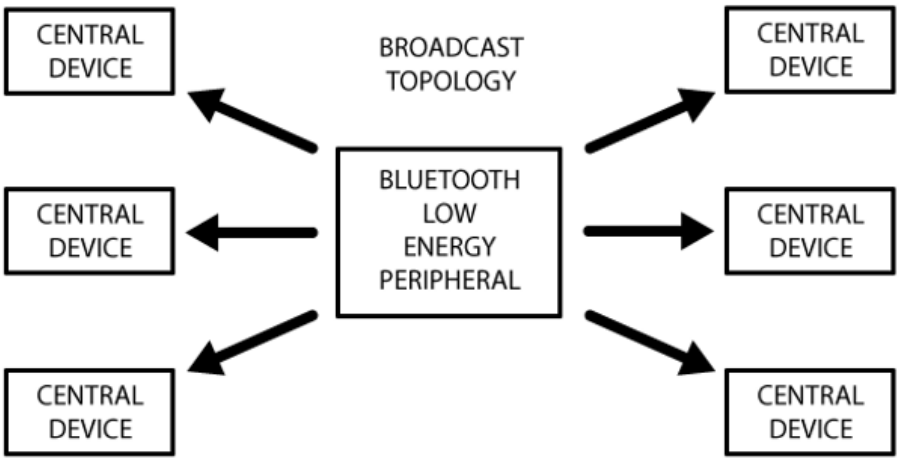


Figure 2: Peripheral advertising to several central devices [8]

Two different types of payloads exist when advertising with GAP: advertising data payload and scan response payload. Advertising payloads are required and is the data that is advertised; scan response payloads are optional and are requested by the central device. Scan response payloads allow the developer to fit more information within the payload such as a device name. Advertising intervals can also be increased as well to lower energy consumption at the cost of latency. Figure 3 illustrates the advertising process of both an advertising payload and a scan response payload. [8]

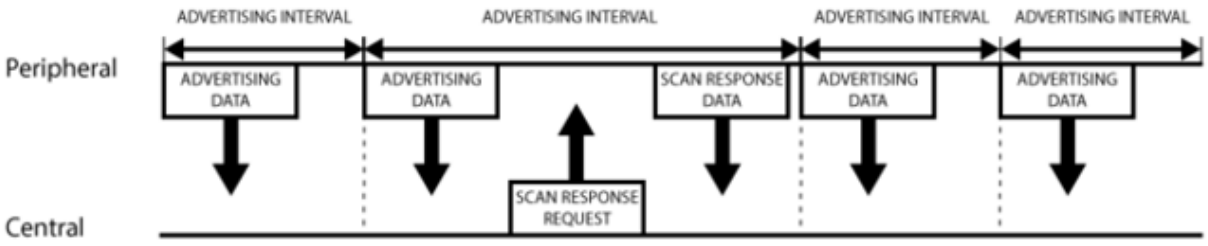


Figure 3: BLE Advertising Process [8]

2.1.4 GATT

In most use cases, after the central device receives an advertisement, the central and peripheral device will create a dedicated connection. The advertising process is then terminated and the peripheral device can only communicate with one central device. Compared with advertising this communication can be bi-directional whereas when advertising the peripheral device can only send data to the central device. This connection schema is shown in Figure 4.

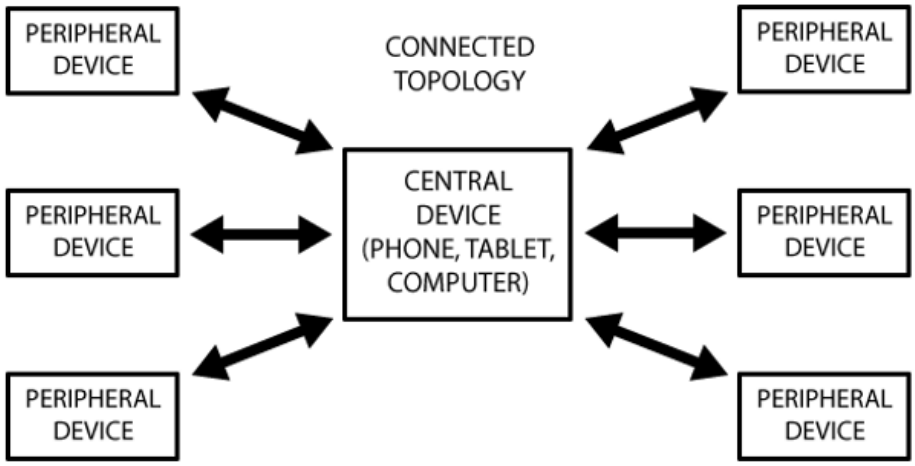


Figure 4: Connected Central and Peripheral devices [8]

Generic Attribute Profile (GATT) defines the way two BLE devices communicate with each other. With a GATT connection, the GATT protocol treats the peripheral device as a GATT sever where the central device is considered a GATT client. The peripheral contains how the data are stored and therefore is defined as a server. When the connection is established the peripheral will suggest a connection interval and the central device will either accept or reject the requested interval based on system resources and existing connections. Figure 5 illustrates the process of communication during a connection.

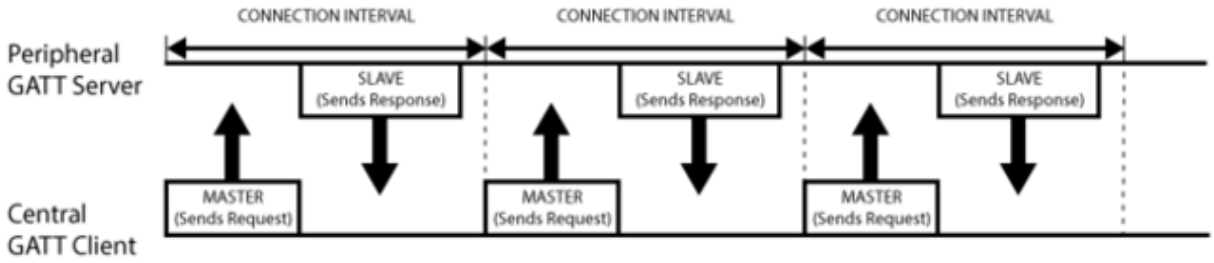


Figure 5: Communication during connection [8]

The data that is being transferred is represented by a nested data structure containing different profiles, services, and characteristics. This data structure is illustrated in Figure 6.

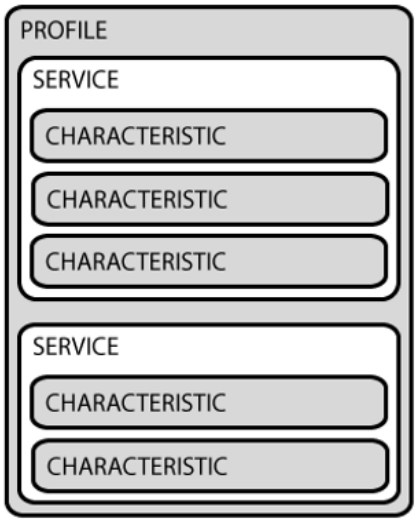


Figure 6: BLE Data Structure [8]

Profiles are a collection of services and characteristics. Profiles are defined either by the developer or by the Bluetooth Special Interest Group (SIG). Each service serves a logical separation between each group of characteristics. Each characteristics contained within a service represents one single data point. One example of a profile defined by Bluetooth SIG is the Blood Pressure Profile that contains a Blood Pressure Service. The Blood Pressure Service exposes

three data characteristics: blood pressure measurement, intermediate cuff pressure, and blood pressure feature. [7]

2.1.5 Simblee

Simblee is designed and developed by Rfduino, a company focused on creating wireless devices that can be programmed with the Arduino IDE. The Simblee is a BLE module that contains both a BLE radio and a 32-bit ARM Cortex M0 processor with 128 KB of flash memory, 24KB of RAM, and operates at 16 MHZ. With the processor and memory packaged with the Bluetooth Smart Radio, the programmer can load a UI for a phone or tablet application instead of using the designated tools to program the phone/tablet. This essentially means you can create an entire iOS app self-contained on the Simblee without using Apple development tools. A block diagram of the Simblee is shown in Figure 7.

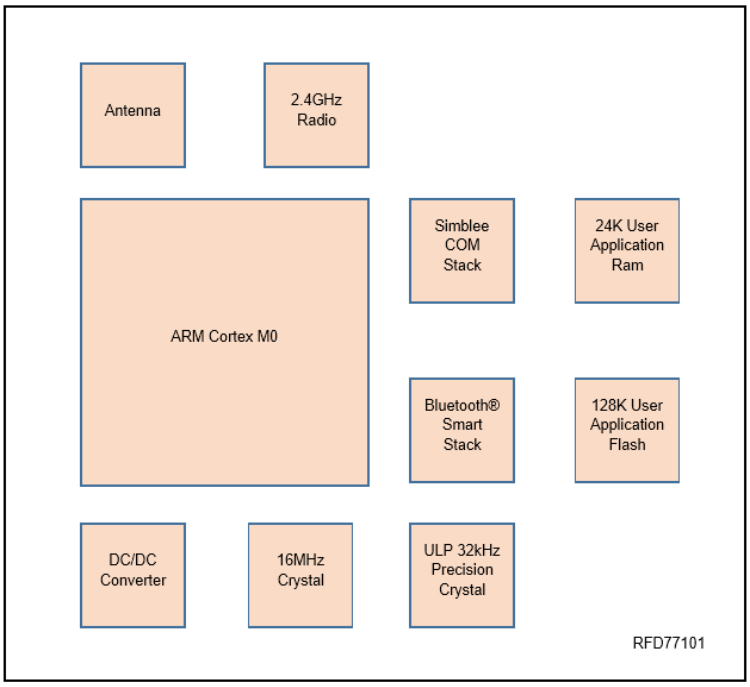


Figure 7: Block Diagram of Simblee [10]

Simblee is a dual-mode Bluetooth device, including both a Bluetooth Smart Stack and Simblee COM (which is the Bluetooth Classic Stack). Several of Simblee functions such as ULP_Delay(), require precise time measurements to operate which makes use of the precision crystal. Simblee can either be used as a standalone microcontroller since it has a built in microcontroller, or a Bluetooth module.

Simblee offers several APIs to develop with including: SimbleeBLE, SimbleeCOM, SimbleeAES, SimbleeCloud, SimbleeForMobile, SimbleeForMobileClient, and a few others. The most robust and well-documented APIs are SimbleeForMobile, SimbleeCloud, SimbleeCOM and SimbleBLE. SimbleeForMobile focuses on creating apps for the iOS and offers an extensive API for creating UI. SimbleeCOM focuses on Simblee-to-Simblee communication, and SimbleeCloud focuses on sending data to a cloud server from the Simblee. SimbleBLE offers a lower-level API and allows you to customize the payload and the

Universally Unique Identifier (UUID) of the Simblee. The API also allows the Simblee to be put into an Ultra Low Power mode or to be turned off. When in ULP mode the Simblee can be awoken by a specified event or can be awoken by a pin transition. When the Simblee is in system off mode, it uses less energy (consumes less than 600 nA) but can only be awoken by a pin event. [9] The Simblee documentation provides two main use cases for the Simblee module illustrated in Figures 8 and 9.

Figure 8: Bidirectional communication between Simblee and phone/tablet [10]

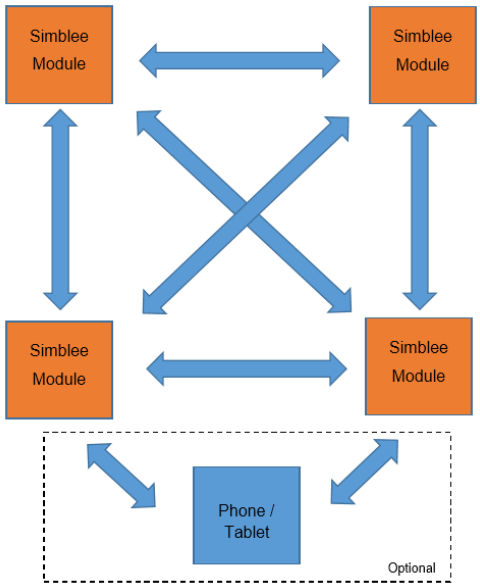


Figure 9: Ad-hoc communication between modules and phone/tablet [10]

2.2 Related Work

2.2.1 Power Consumption Analysis of Bluetooth Low Energy, ZigBee and ANT Sensor Nodes in a Cyclic Sleep Scenario [11]

The Electrical Engineering department of the University of Washington, Seattle published a paper concerning power analysis of BLE, ZigBee, and ANT. Both ZigBee and ANT are low energy wireless protocols that are similar to BLE. The research involved focused on testing how sleep cycles affected the power consumption of each protocol by sending a fixed number of bytes from the slave module (wireless protocol module) to a master module. The results demonstrated that the optimal sleep cycle for BLE was 10 seconds and that at 10 seconds it was consuming ~0.06 mA. This value corresponds relatively closely to the data collected by this thesis. They also tested wider ranges of sleep cycle intervals, ranging from 1 second to 120 seconds. Their data showed that the current consumption kept decreasing but began to reach an asymptote at 30 seconds. Compared to ZigBee and ANT, BLE performed better in lower power consumption while retaining a smaller sleep interval.

3. ARCHITECTURE

3.1 High Level Design

In order to evaluate how power consumption changes due to changing different parameters, two testing applications were created to test different advertising intervals, connection intervals, and ULP sleep cycle intervals. To evaluate different advertising intervals, the Simblee API exposes an advertising interval variable to change within its SimbleeBLE class. To test the connection intervals and ULP sleep intervals, a mobile application was created to operate on the iOS Simblee app. Figure 10 illustrates how the mobile application functions.

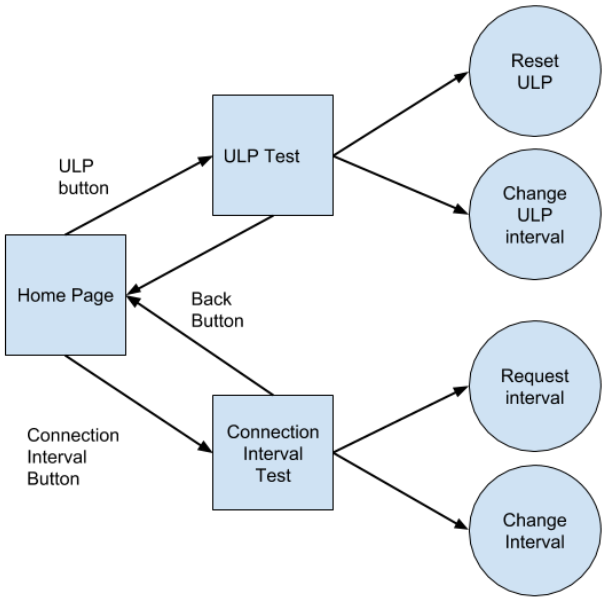


Figure 10: Simblee Mobile Application Functionality

The mobile application allows the connection intervals and ULP sleep cycle intervals to be easily changed without reprogramming the Simblee. In order to monitor how much current the Simblee was using a simple circuit was set up with the battery, the Simblee module, and a multimeter.

3.2 Tools and Documentation

Since Simblee was developed for use in the Arduino environment, the Arduino IDE was used for programming the Simblee. The Simblee API documentation provided by Rfduino, Simblee's manufacturer, is fairly limited so referring to the Simblee API source header files proved useful. All testing was done with the RFD77803 Simblee dev kit with a Fluke 287 true RMS multimeter.

3.3 Implementation

When programming an Arduino device, two functions are required to be present in the code: void setup() and void loop(). The setup function is called when the program is initially loaded onto the Arduino device. The loop function is essentially a while(true) loop, executing its code until an interrupt is encountered or the device turns off. Other functions can be written to be called within setup() or loop().

The Simblee API exposes the attribute `advertisementInterval` from the Simblee class to set the advertisement interval in milliseconds. `advertisementInterval` must be set before the SimbleeBLE or SimbleeForMobile stack has begun and in most cases is initialized within the setup function. A portion of the code implemented is show in Figure 11.

```
48 //   -20, -16, -12, -8, -4, 0, +4
49   SimbleeBLE.txPowerLevel = -8;
50   SimbleeBLE.advertisementInterval = 1000;
51
52   // start the BLE stack
53   SimbleeBLE.begin();
```

Figure 11: Advertisement Interval Code

Within the loop function, Simblee_ULPDelay(INFINITE) is called so when the Simblee is not advertising, it will enter Ultra Low Power mode. The variable `txPowerLevel` allows the programmer to set the BLE transmit power.

The mobile application was developed using the SimbleeForMobile API which is built on top of the SimbleeBLE API. When creating a SimbleeForMobile application, two additional functions are required: ui() and ui_event(). ui() defines how the UI will look on the iPhone and ui_event() handles any events, such as a button press, that occur on the iPhone. To change the connection intervals, the function void updateConnInterval(int min_conn_interval_ms, int max_conn_interval_ms) can be called that accepts two integers, one for the minimum connection interval and one for the maximum connection interval. int getConnInterval() can be called to check if the central device accepted the proposed connection interval. In order to implement the ULP sleep cycles, Simblee_ULPDelay(int delay) was called within the loop function with the delay parameters set to the desired sleep time. Screenshots of the application can be seen in Figure 12.

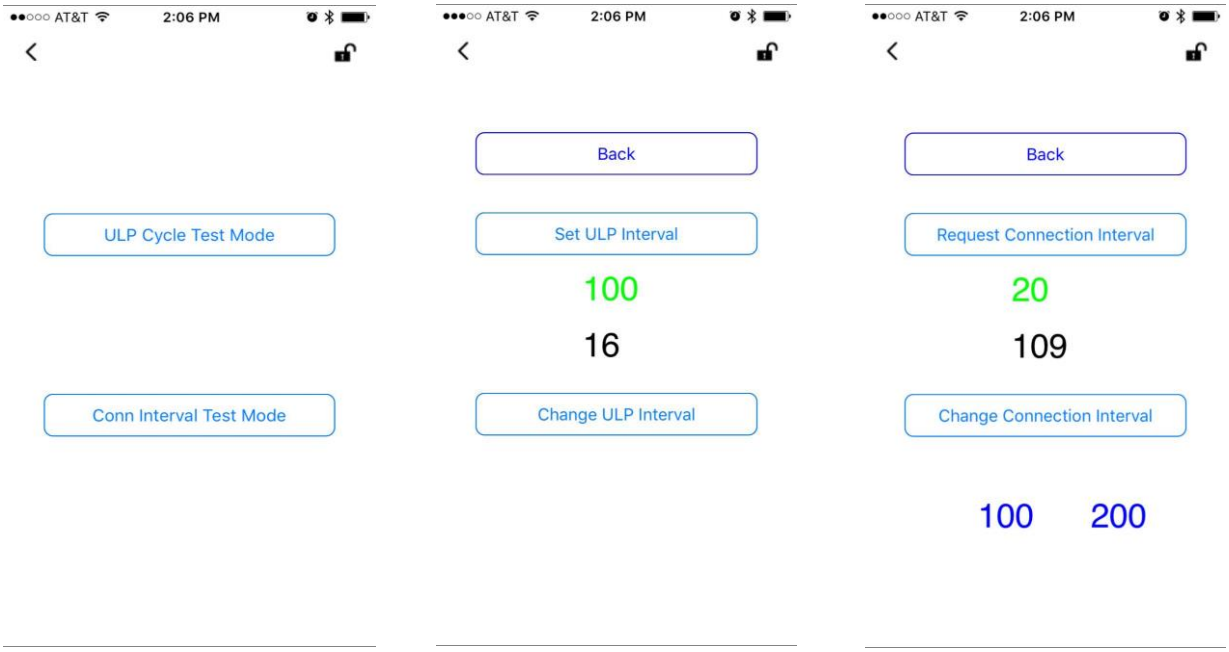


Figure 12: Simblee Mobile Application

The green numbers represent the ULP interval and Connection interval. The black numbers are updated by the Simblee and illustrates how often the Simblee is sending information. The blue numbers on the connection interval screen represent the interval that was requested by the Simblee.

4. METHODOLOGY, RESULTS AND ANALYSIS

4.1 Methodology

The first step taken to measure current consumption of the Simblee was to set up a circuit with the Simblee module, battery shield, and multimeter. The average current was measured by using the average function the multimeter provided. After the circuit was set up and the multimeter had been on for at least an hour, the Simblee was loaded with the desired program. The average current was measured for approximately two to three minutes then the reading was reported. When testing the advertising intervals, intervals from 20 ms to 10000 ms were measured. Connection intervals were tested from 100 ms to 1100 ms and ULP sleep cycles were tested from 100ms to 10000ms. Transmit powers of -20 dBm, -8 dBm, and 4 dBm were tested for all three intervals tests.

4.2 Results

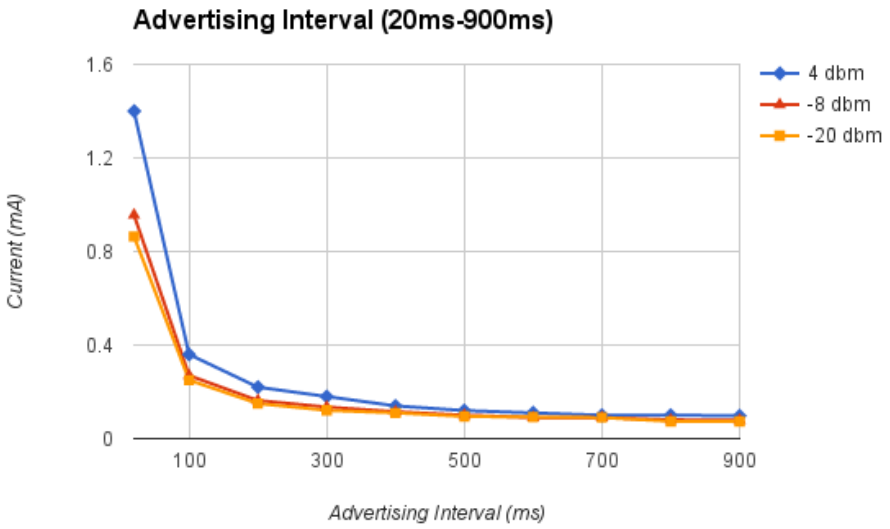


Figure 13: Advertising Interval 20ms-900ms

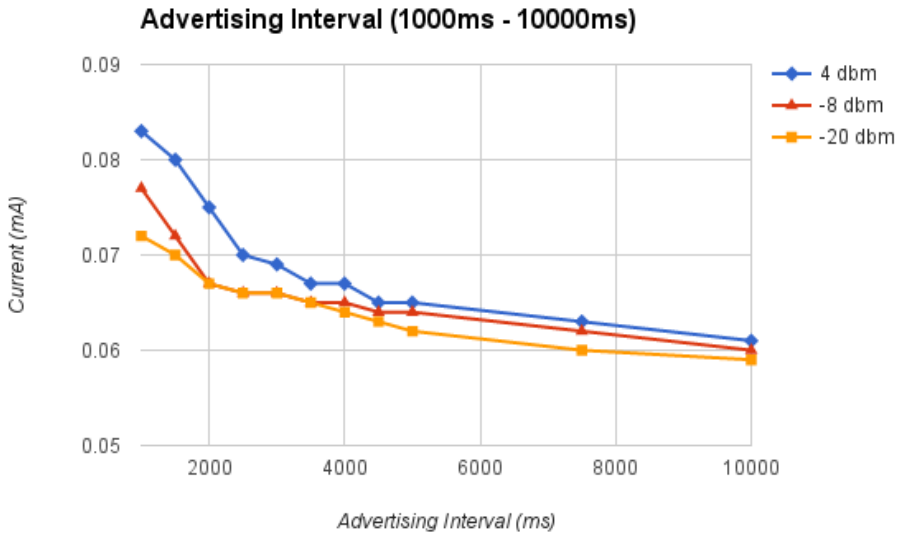


Figure 14: Advertising Interval 1000ms-10000ms

Figure 13 and 14 show the results of changing the advertising intervals. Changing the intervals from 20 ms to 300 ms gives drastic improvements. As the advertisement intervals increase, the improvement in current consumption decreases. At approximately 900 ms it begins to reach a minimum current. Figure 14 shows the advertising intervals from 1000ms to 10000ms with a different scale on the y-axis. The current consumption eventually reaches a minimum at around 0.06 mA.

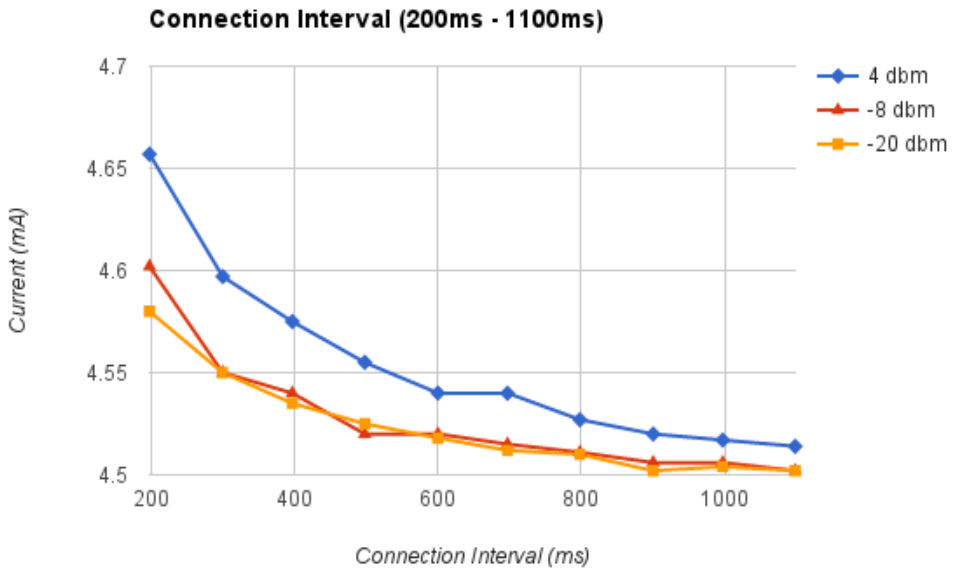
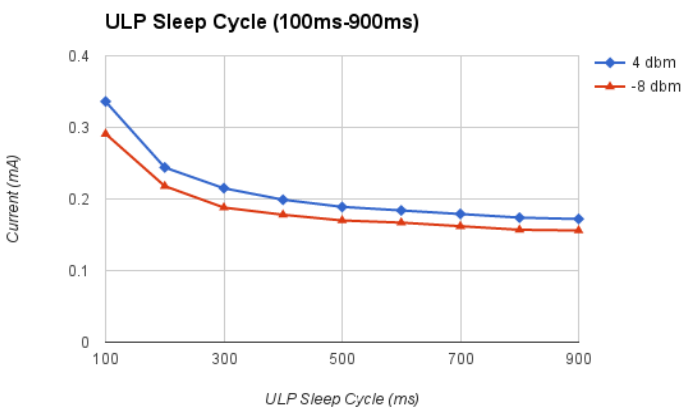


Figure 15: Connection Interval 200ms-1100ms

Changing the connection interval yielded poor results. Even though increasing the connection interval decreases the current, the Simblee API does not offer any way to enter ULP mode in between connections or wake the Simblee from ULP mode every time the Simblee sends information to the mobile application. Figure 16 and 17 shows an alternative method by establishing ULP sleep cycles, it does not change the connection interval but the Simblee only sends data every time it wakes.



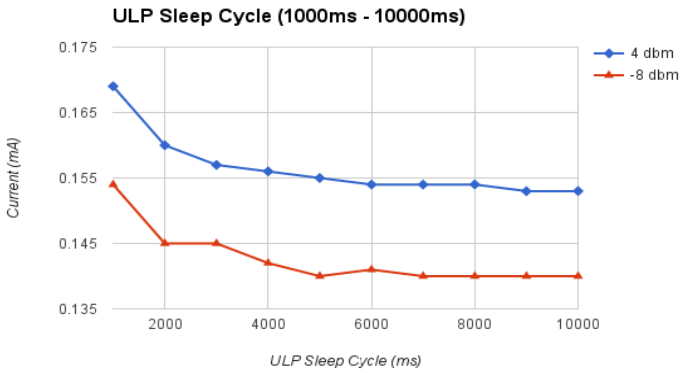


Figure 16: ULP Sleep Cycle 100ms-900ms **Figure 17: ULP Sleep Cycle 1000ms-10000ms**

Using ULP sleep cycles drastically improved upon connection intervals. The current v ULP cycle curve is similar to the current v advertisement interval curve where initially increasing the cycle provides a large benefit in current consumption. Figures 18 and 19 show the theoretical battery life for both advertisement intervals and connection intervals for an AAA battery and an average coin cell battery. -20 dBm was not tested since it did not yield any improvements over -8 dBm. When calculating the battery life the following equation was used.

$$\frac{Battery\ Capacity\ (mAh)}{Device\ Consumption\ (mA)} * 0.7$$

The coefficient 0.7 takes into account external factors affecting the battery life.

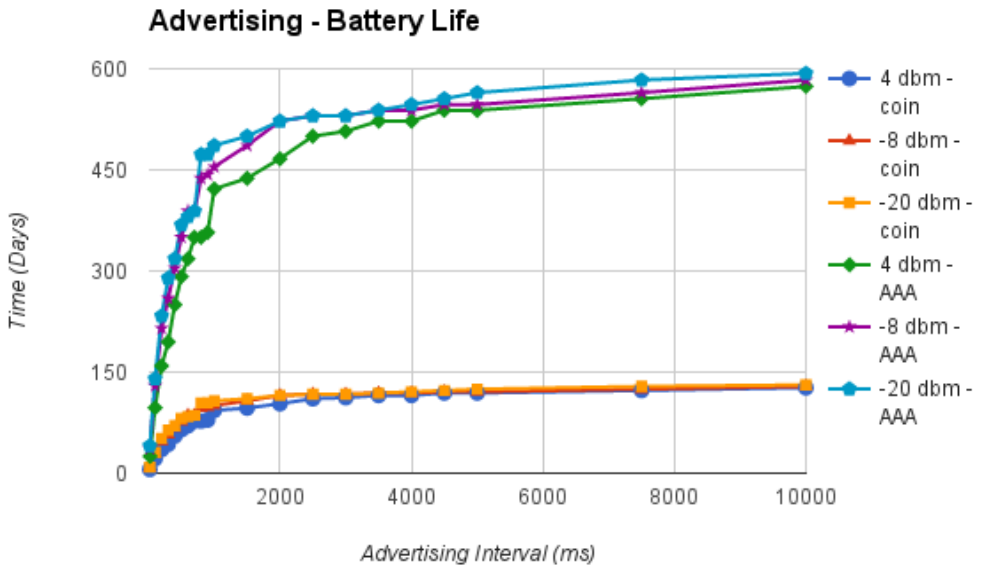


Figure 18: Advertising Interval - Battery Life

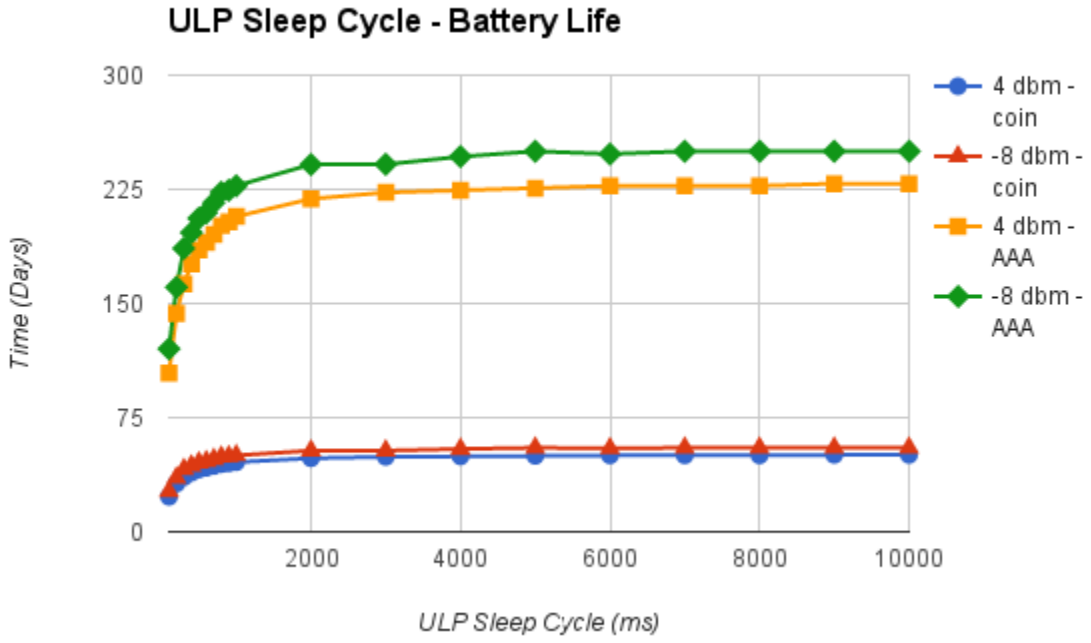


Figure 19: ULP Sleep Cycle - Battery Life

4.3 Analysis

For both advertisement intervals and ULP sleep cycles the benefit of increasing the interval at first greatly improves the battery life but gradually reaches an asymptote as the intervals increase. Simblee's documentation states that the Simblee's ULP mode consumes <4µA (.004 mA). This is true if the BLE stack has not started (SimbleeBLE.begin() has not been called in the program). It seems that the other components within the Simblee (such as the processor and memory) are consuming some kind of current. Processor load could be affecting current consumption too. The results, however, support the fact that increasing the interval in which the Simblee enters ULP mode drastically reduces current consumption and can lead to increased battery life.

5. CONCLUSIONS

5.1 Summary

This thesis evaluates the Simblee's low energy capabilities by changing different parameters and implementations, specifically the advertising interval, connection interval, and ULP sleep interval. Ranges from 20ms intervals to 10000ms intervals were tested with power transmit levels at -20dBm, -8dBm, and +4dBm. Two main applications were built for testing these parameters: one to test the advertisement intervals and the other to test the ULP sleep cycles and connection intervals. The results gathered by this research show that increasing the intervals in which the Simblee sleeps, greatly decreases the power consumption.

5.2 Contributions

The data collected from this thesis will help future Simblee developers make more informed decisions when programming the Simblee. The data gives a clear representations of how the current consumption is affected by different parameters. Implementing a sleep-wake cycle within the program can increase battery life from days to months, which opens up new use cases and possibilities for the Simblee module.

5.3 Future Work

This thesis provides an overview of energy consumption of the Simblee. The research can be expanded upon by testing the different APIs provided by Simblee, most notably SimbleeCloud. SimbleeCloud is a useful API that connects the Simblee to a server to transfer data over the cloud. A more standardized test for SimbleeForMobile could be created at a lower level. As of now, with the SimbleeForMobile API, it is difficult to control how much information

is being sent from the Simblee to the iPhone. A lower-level application could be created to control and standardize the data flow from the Simblee to the iPhone.

Since Simblee is composed of more than just a Bluetooth Smart Radio, each component could be individually tested to see how much power it's using under certain conditions. This could help explain why the Simblee isn't reaching the lower current consumption and it could possibly determine what the SimbleeBLE and SimbleeForMobile API is doing in the background while executing.

REFERENCES

1. "Low Energy." *Bluetooth*. N.p., n.d. Web. 20 Apr. 2016. <https://www.bluetooth.com/what-is-bluetooth-technology/bluetooth-technology-basics/low-energy>.

2. "How Do Fitbit Trackers Sync Their Data?" *Fitbit Help*. N.p., n.d. Web. 20 Apr. 2016. https://help.fitbit.com/articles/en_US/Help_article/1877

3. R. Want, B. N. Schilit and S. Jenson, "Enabling the Internet of Things," in *Computer*, vol. 48, no. 1, pp. 28-35, Jan. 2015.

4. "IoT, Internet of Things." *IoT Internet of Things*. Codit, n.d. Web. 20 Apr. 2016. <http://www.codit.eu/how-can-we-help/internet-of-things/>.

5. *Internet of Things: Science Fiction or Business Fact?* Rep. Verizon / Harvard Business Review, n.d. Web. 20 Apr. 2016. <https://hbr.org/resources/pdfs/comm/verizon/18980_HBR_Verizon_IoT_Nov_14.pdf>.

6. Nillson / CEO of ConnectBlue, Rofl. "Shaping the Wireless Future with Low Energy Applications and Systems." *ConnectBlue*. Ublox, n.d. Web. 20 Apr. 2016. <http://www.connectblue.com/press/articles/shaping-the-wireless-future-with-low-energy-applications-and-systems/>.

7. "Blood Pressure Service (BLS)." *BLS | Bluetooth Development Portal*. Bluetooth SIG, n.d. Web. 20 Apr. 2016. <https://developer.bluetooth.org/TechnologyOverview/Pages/BLS.aspx>.

8. "Introduction to Bluetooth Low Energy." *Introduction*. Adafruit, n.d. Web. 20 Apr. 2016. <https://learn.adafruit.com/introduction-to-bluetooth-low-energy/introduction>.

9. "Simblee Concepts." *Learn at SparkFun Electronics*. SparkFun, n.d. Web. 20 Apr. 2016. <https://learn.sparkfun.com/tutorials/simblee-concepts/low-power-modes>.

10. *Simblee User Guide V 2.05*. N.d. Https://www.simblee.com/Simblee%20User%20Guide%20v2.05.pdf.

11. Dementyev, Artem, Steve Hodges, Stuart Taylor, and Joshua Smith. "Power Consumption Analysis of Bluetooth Low Energy, ZigBee and ANT Sensor Nodes in a Cyclic Sleep Scenario." *2013 IEEE International Wireless Symposium (IWS)* (2013): n. pag. Web.