STUDY ON THE TRADE OFF BETWEEN THROUGHPUT AND POWER CONSUMPTION IN THE DESIGN OF BLUETOOTH LOW ENERGY APPLICATIONS

By

Hafiz Ahmed

Approved:

Ahmed Eltom
Professor of Electrical Engineering
(Chair)

Mina Sartipi
Associate Professor of Computer Science
(Thesis Advisor and Committee Member)

Stephen Craven
Adjunct Professor of Electrical Engineering
(Committee Member)

William H. Sutton
Dean of the College of Engineering and Computer Science

A. Jerald Ainsworth
Dean of the Graduate School

STUDY ON THE TRADE OFF BETWEEN THROUGHPUT AND POWER CONSUMPTION IN THE DESIGN OF BLUETOOTH LOW ENERGY APPLICATIONS

By

Hafiz Ahmed

A Thesis
Submitted to the Faculty of the
University of Tennessee at Chattanooga in Partial
Fulfillment of the Requirements
of the Degree of Master of Science
in Electrical Engineering

The University of Tennessee at Chattanooga
Chattanooga, Tennessee

May 2013

ABSTRACT

Bluetooth Low Energy (BLE) is an emerging technology that is considered a breakthrough in the field of wireless communications due to its very low power consumption property. One of the issues of the current practical implementations of BLE is throughput. The theoretical throughput of BLE is around 260 Kbps but the current practical implementations have much lower throughput values. In this work, the effect of important parameters related to the BLE connection and the Generic Attribute Profile (GATT) protocol on the throughput and power consumption of the system has been studied based on practical laboratory experiments.

The throughput was found to increase almost linearly with the number of characteristics and characteristic size used in a BLE application. Additionally, the throughput was found to have an inverse relationship with the connection interval. The average current consumed while the device is connected was found to have a proportional relationship with the number of characteristics and characteristic size and therefore the battery life of a BLE device is greatly affected by these variables. Understanding these results is crucial to BLE system designers and developer because it allows them to design their systems in a way that suits their needs optimally.

DEDICATION

I dedicate this thesis to my parents. No words could ever describe my feelings towards them.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude for Dr. Ahmed Eltom and his wife Shadia for their continuous support during my master degree study and being a second family for me in the United States.

I would also like to express my sincere gratitude for Dr. Mina Sartipi, for her guidance, thoughtful criticism and invaluable assistance.

Special thanks to Dr. Craven, for his guidance and support throughout my master degree study.

My deepest thanks to my family in Sudan, my friends: Halim, Hamza, Ammar, Mohamed, Khalid, Hattan, Alia, Salma, and Suha for always being there whenever I needed them.

Many thanks to Wade Gasior and George Yu from Variable Technologies for their support and invaluable ideas during the implementation of this research project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BLE, Bluetooth Low Energy

GATT, Generic Attribute Profile

TDMA, Time Division Multiple Access

FDMA, Frequency Division Multiple Access

GFSK, Gaussian Frequency Shift Keying

HCI, Host Controller Interface

L2CAP, Logical Link Control and Adaptation Protocol

UUID, Universally Unique Identifier

GAP, Generic Access Profile

CRC, Cyclic Redundancy Check

LLID, Logical Link Identifier

SN, Sequence Number

NESN, Next Expected Sequence Number

MD, More Data

FCS, Frame Check Sequence

RSSI, Received Signal Strength Indication

DMM, Digital Multi-meter

TPT, Throughput-Power Consumption Trade off

CHAPTER 1

INTRODUCTION

Communication systems have changed and will continue on changing our way of life. Information sharing has become available through different channels and technologies of wireless communications. With the availability of information and communications technologies, some challenges such as power consumption have to be addressed to enable more wireless applications. Power consumption has always been a limiting factor in many wireless sensor network applications. Many technologies such as Zigbee and Bluetooth have been competing to provide wireless connectivity with low power consumption. Bluetooth Low Energy (BLE) is a rather new protocol that is expected to enable a revolution in the area of wireless sensor networks because of its low power feature. The low power consumption is achieved by the design of the protocol which keeps the radio off except when it is needed [1]. It allows a device communicating through BLE and running on a coin-cell battery to last more than one year. The low power consumption feature of BLE will enable us to monitor and share information about virtually everything (keys, headsets, etc., can be monitored through the use of BLE). It is expected to enable a large number of internet of things related applications such as health monitors, intelligent shopping, and smart grid applications.

BLE is designed to have a theoretical data rate of 1 Mbps and throughput is around 260 Kbps but practical implementations are much lower. Since the low power consumption of BLE is achieved by sleeping for long period of time and staying awake for sending/receiving data for short amount of time, being capable of sending only very small amount of data in that period makes BLE impractical for many applications. Both the throughput and power consumption of BLE are affected by the connection parameters and other parameters related to the protocols of BLE stack.

In this research, the effect of important parameters related to the BLE connection and the Generic Attribute Profile (GATT) protocol on the throughput and power consumption of the BLE system has been studied. The results of this study are very important to BLE system designers and developers because it helps them make the right decisions when designing their systems and choosing the right set of parameters according to the needs of their systems.

CHAPTER 2

LITERATURE REVIEW

Overview of Bluetooth Low Energy (BLE)

Bluetooth Low Energy (BLE) is a relatively new wireless technology which is designed for applications that require low power consumption such as monitoring and control applications [2].BLE operates in the unlicensed 2.4 GHZ Industrial, Scientific, and Medical (ISM) band. It uses both Time Division Multiple Access (TDMA) and Frequency Division Multiple Access (FDMA) access technologies. To counteract interference and fading, BLE uses a frequency hopping scheme which hops between 40 frequencies separated by 2 MHZ. Three of these channels are advertising channels and the rest are data channels. TDMA is used by assigning time slots for each device to send and receive data on a specific frequency. Two important terms in BLE are physical channels and events:

- Physical Channel: Frequency at which data is sent
- Event: Time unit in which data is sent between BLE devices. There are two types of events: advertising events and connection events.

As the name suggests, advertising packets are sent in advertising events and data is sent in connection events. Figure 1 shows two connection events and one advertising event where the slave and master exchange packets. Since BLE uses frequency hopping, packets of each of the events are sent in a different frequency.

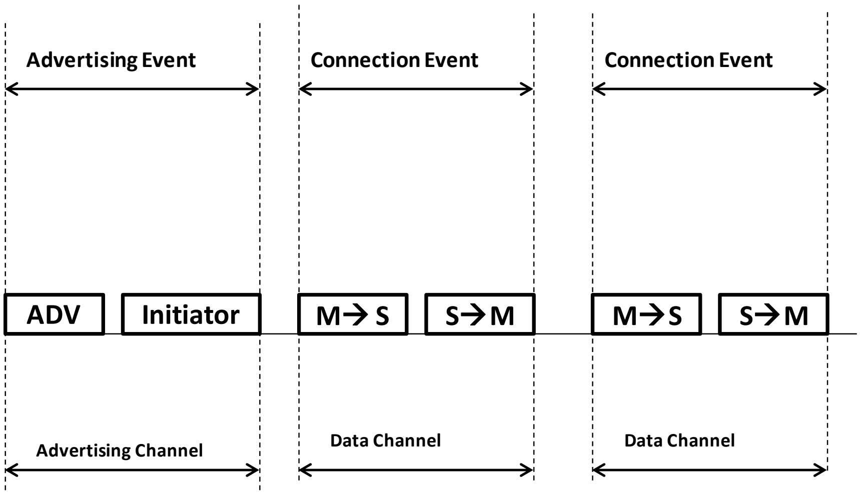

Figure 1 Connection and Advertising Events

Other important terms related to the BLE connection are the connection interval and slave latency:

- Connection interval: the time between two consecutive connection events [3]
- Slave Latency: A parameter that results in power saving by allowing the slave to skip a number of connection events if it does not have data. It specifies the maximum number of connection events that can be skipped. [4]

BLE Stack

One of the design goals of BLE is simplicity. The BLE stack shown in Figure 2 consists of a host and a controller. The controller is a physical device which is capable of sending and receiving data such as a chip. The host is typically a software stack which controls the communication between two or more devices and how different services can be provided over a certain radio channel. Applications use the host and controller to perform certain functions. [5]

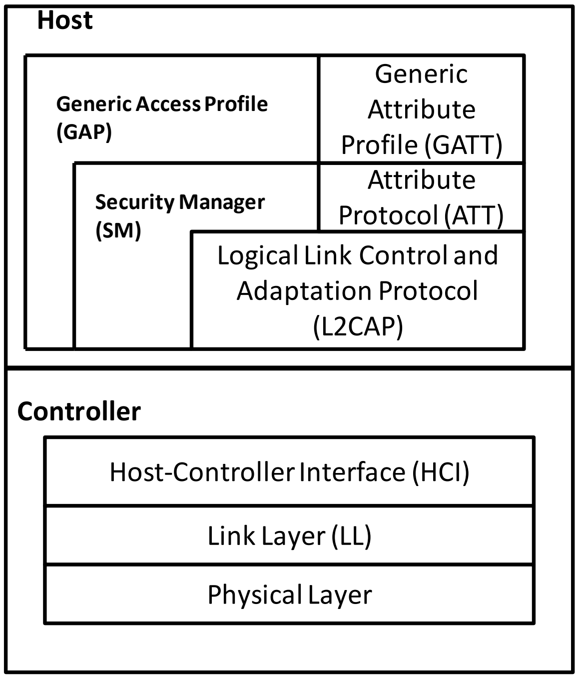

Figure 2 BLE Stack Architecture [4]

Physical Layer: the physical layer is responsible for transmission and reception of data in the 2.4 GHZ frequency. It uses the Gaussian Frequency Shift-Keying (GFSK) modulation scheme.

Link Layer: the link layer is very important because it is responsible for the RF state of the device as well as for advertising, scanning, initiating and connecting.

The Host Controller Interface (HCI) Layer: It enables communication between the host and the controller by allowing a host to send data and commands to the controller and a controller to send data and events to the host [5]. It can be implemented by a software API or hardware such as USB or UART. [4]

Logical Link Control and Adaptation Protocol (L2CAP): It enables end-to-end logical communication by providing data encapsulation services for upper layers.

The Security Manager Protocol: A simple protocol which is responsible for key distribution (for encrypting data using a shared secret), pairing (where devices attempt to trust one another) and other security related functions.

The Attribute Protocol: It defines communication between two devices by allowing them to expose certain pieces of data know as “attributes” and setting rules for accessing data on a peer device [5].In attribute protocol context, the server is the device on which data is stored and the client is the device which reads/writes data from/to the server.

Attributes are data which have an address, type and a value. For example an attribute could be of type weight with value 100 and handle 0x139. An attribute database is a collection of attributes which is stored in an attribute server and the client uses the attribute protocol to access the attributes. Figure 3 shows an example of an attribute database.

Attribute Handle	Attribute Type	Attribute Value
0x0001	Primary Service	GAP Service
0x0002	Characteristic	Read, 0x003,Device Name
0x0003	Device Name	“BLE Reader”
0x0004	Primary Service	GATT Service
0x0005	Primary Service	Electric Measurements Service
0x0006	Characteristic	Read Write,0x0007, Voltage Level
0x0007	Voltage Level	110
0x0008	Characteristic	Read, 0x0009, Frequency Value
0x0009	Frequency Value	54

Figure 3 An example of an attribute database [5]

The Generic Attribute Profile: A protocol that sits on top of the attribute profile and defines the types of attributes and the communication methods between a client and a server by introducing the concepts of characteristics and services. It also defines procedures for discovering characteristics and services, the relationship between services and reading/writing characteristics.

- Service: an encapsulation of some behavior of a device that cannot be changed. For example, a battery level service can be used to deal with battery level in an application.
- Characteristic: data that describes a certain variable. It is labeled with a Universally Unique Identifier (UUID). For example, a characteristic can be the temperature of a body or the rate of heart beat of a patient. A characteristic needs to expose the type of data and the permissions (whether it is read and/or write) and how to configure the data to be notified or broadcast or indicated. A characteristic is composed of three basic elements: declaration, value, and descriptor.
- Characteristic Declaration: This attribute contains three fields: characteristic properties, handle, and type.
- Characteristic Value: an attribute with a type that must match the type in the characteristic declaration attribute. It contains the value of the designated variable.

- Characteristic Descriptors: This attribute can be any number of descriptors on a characteristic for example user description of a characteristic or characteristic presentation format [5]. An example of a characteristic is shown in table 1.

Table 1 Characteristic Example

Handle	Type	Value
0x11	Primary Service	GAP Service
0x12	Characteristic	Read write, 0x13, body temperature
0x13	Body Temperature	98
0x14	Characteristic	Read, 0x15, glucose level
0x15	Glucose level	50

A service is a group of characteristics and the associated behaviors. A service may reference another service. There are two kinds of services:

- Primary service: A service that performs a basic task of the device
- Secondary Service: A service that performs an auxiliary task of the device and must be referenced in at least one primary service [3]. Figure 4 shows an example of two services, service A references service B

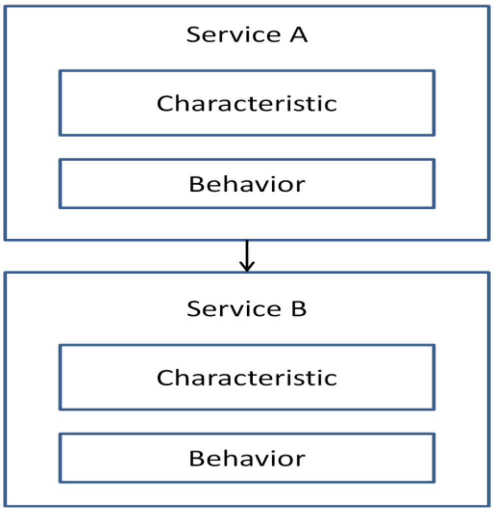

Figure 4 Service A references Service B [5]

A profile sits on top of the hierarchy. It is composed of one or more services used to perform certain functions. Figure 5 shows a general layout of a profile.

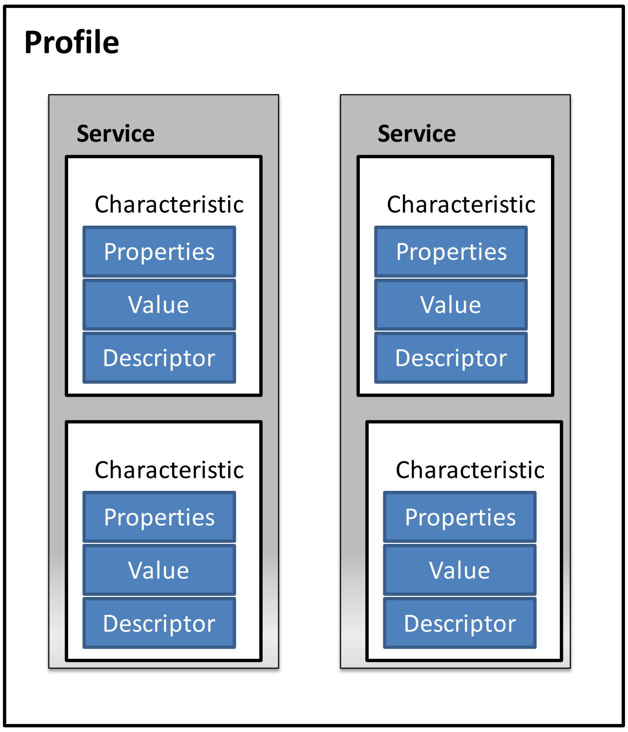

Figure 5 Profile Hierarchy [3]

The Generic Access Profile: It is responsible for device discovery and connection-related services through interfacing with applications and profiles. It also handles the GAP roles of peripheral, central, broadcaster and observer. [3]

- Peripheral is a role that supports a single connection.
- Central is a role that supports multiple connections and initiates connections with devices in peripheral role.
- Broadcaster is a device that sends advertising packets
- Observer is a device that listens to broadcasters and reports to an application. [3]

Profile Hierarchy

Bluetooth profiles enables interoperability in the Bluetooth system. They define interaction methods between layers, the required functions and features of each layer in the

Bluetooth system, data formats and application behaviors. All Bluetooth devices are required to implement GAP and therefore application profiles become supersets of GAP. [3]

Figure 6 shows an application profile

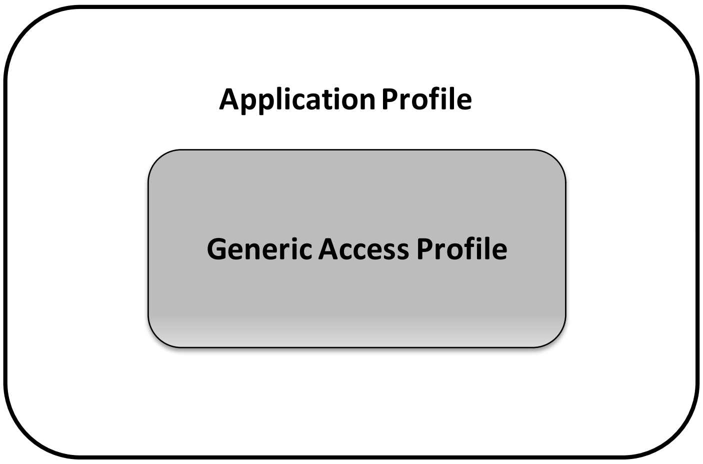

Figure 6 Application Profile Hierarchy [3]

BLE Packets

There are two types of packets in Bluetooth Low Energy: advertising packets and data packets. The difference between data packets and advertising packets is that advertising packets are sent to any listening device or a specific device whereas data packets can only be understood by two devices: a master and a slave [5]. Figure 7 shows the general structure of a BLE packet.

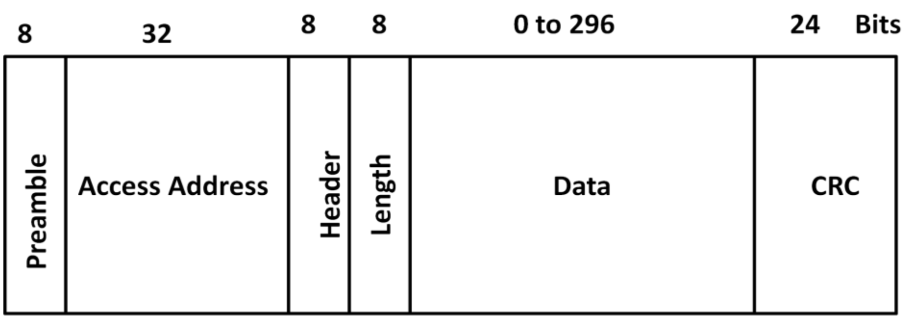

Figure 7 Packet Structure [5]

The fields of the packet are:

- Preamble: a short training sequence placed at the start of the packet
- Access address: This field is used to distinguish the packet from the background noise because the probability of the background noise bits matching the preamble is fairly high but when adding the access address this probability becomes very low. There are two types of access addresses: advertising access address and data access address. Advertising access address is used when the device is not in a connection state whereas the data access address is used when the device is connected.
- Header: The header field differs from advertising packets to data packets. In advertising packets, it contains the advertising packet type and some flag bits. In data packets, it contains four fields: link layer identifier, next sequence number, sequence number and more data. Figure 8 shows the contents of a data packet header.

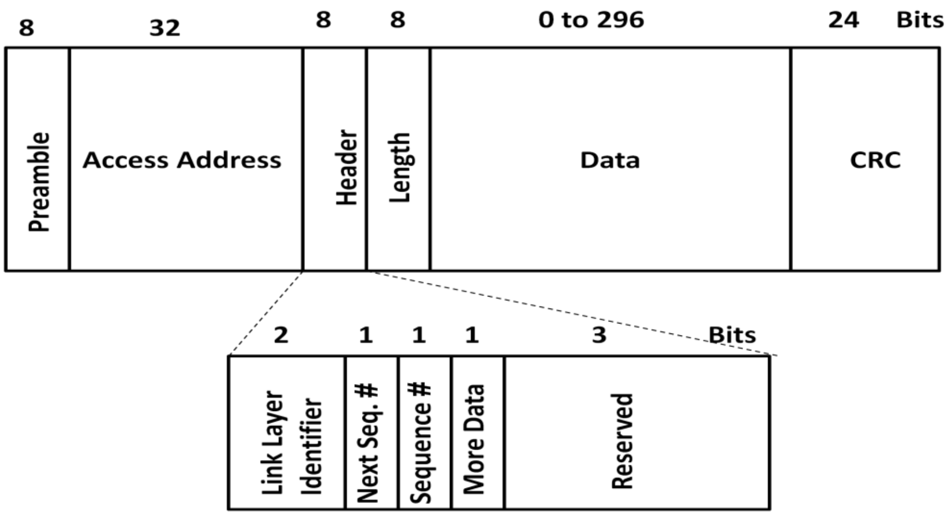

Figure 8 Data Packet Header [5]

- Length: This field contains the length of the payload in bytes
- Data: The actual data that is being transmitted. The length of this field in bytes must be the same as the values in the length field.

- Cyclic Redundancy Check (CRC): This field is used to detect error. It is calculated over the header, the length and the payload fields.

The fields of the data packet header are:

- Logical Link Identifier (LLID): specifies the type of packet among:
    - Link Layer control packet (11): used for connection management
    - Start of a higher-layer packet or a complete packet (10)
    - Continuation of a higher-layer packet (01)
- Sequence Number (SN): A single bit used to identify packets by assigning them sequence numbers. The first packet must have a sequence number of zero and then the packets alternate in sequence numbers between 0 and 1. If the receiver receives a packet with a sequence number similar to the previous one, it assumes that it is a retransmission.
- Next Expected Sequence Number (NESN): used to acknowledge reception of a packet using a single bit. If the sequence number is zero the NESN must be one.
- More Data (MD): This field signals to the receiving device that the transmitting device has more data ready to send which causes connection events to be extended. If it is set to zero, connection events can be closed to save power. [5]

Figure 9 shows a simple connection event that explains the use of the fields of the header in data packets. The master sends the first packet with SN of zero and NESN of zero and MD field set to one. The slave replies with a packet that has sequence number set to zero, NESN set to 1 and MD to 1 but the packet is not sent correctly, so the master retransmits the first packet because it did not receive an acknowledgment. Since the slave did not receive an acknowledgement of its first packet it retransmits the packet again. The master receives the first packet from the slave and replies with a packet that has SN set to 1, NESN to 1 and MD to zero, which acknowledges reception of packet with SN =0 and tells the slave that it does not have more data to send. The slave replies with a packet that has SN set to 1, NESN to 0 and MD to 0, which acknowledges reception of the packet with SN=1 and tells the master that it does not have more data ready to be sent and this concludes the connection event.

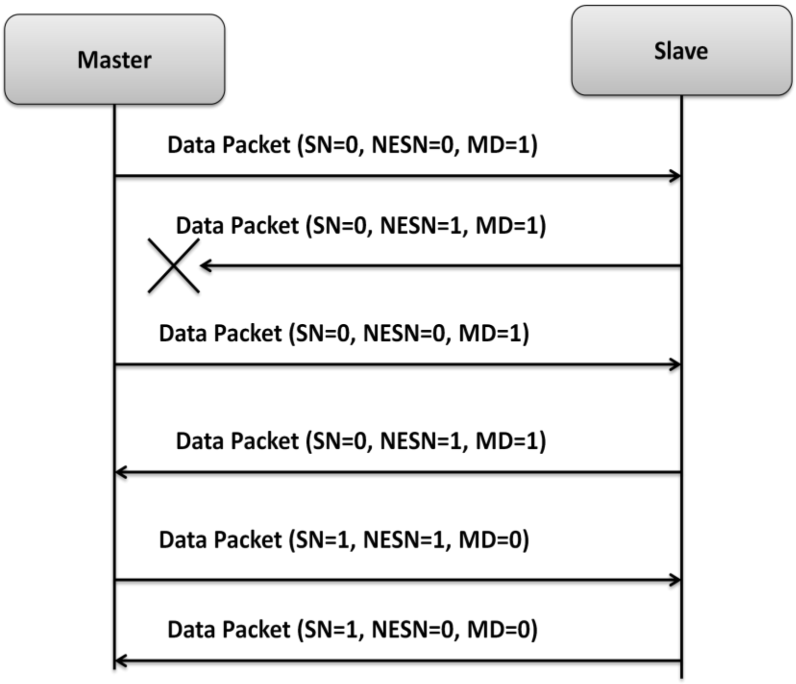

Figure 9 Fields of Data Packet Header in a connection event [5]

Previous Related Work

The authors of reference [2] studied the throughput of BLE and provided a simulation that predicts the values of the throughput of BLE for different values of the connection interval and bit error rates.

In September 2012, at the IEEE International symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems in Germany, a paper presented the results of the performance analysis study on BLE. The authors of the paper measured the throughput of data using GATT notifications and the power consumption of a BLE device while in advertising state. [6]

CHAPTER 3

METHODOLOGY

To study the effects of the GATT protocol parameters and connection parameters on the throughput of BLE we used Texas Instruments' CC2540 development kit. Two devices were used one as a master and the other as a slave as shown in figure 10.

Figure 10 Master and Slave Devices

When developing a BLE application we need to define one or more profiles which are composed of services and characteristics. The choice of the specifications of those characteristics can greatly affect the throughput and power consumption. The current implementation of BLE stack using Texas Instruments' CC2540 development kit allows only four packets to be transmitted in a connection event that is why we limited our experiments to four characteristics at a time.

In our study of the throughput we considered three variables which are: the connection interval, the characteristic size and the number of characteristics.

Nine Characteristics were defined in a profile called simpleGATTprofile as part of a service called simpleProfileService. The UUIDs for the defined characteristics are shown in Table 2.

Table 2 Characteristics UUIDs

Characteristic	UUID
CHAR 1	0xfff1
CHAR 2	0xfff2
CHAR 3	0xfff3
CHAR 4	0xfff4
CHAR 5	0xfff5
CHAR 6	0xfff6
CHAR 7	0xfff7
CHAR 8	0xfff8
CHAR 9	0xfff9

To write a characteristic value to a GATT server, Texas Instruments Host Controller Interface (HCI) commands offer two basic types of requests:

- Acknowledged GATT Write Request: a write request which requires an acknowledgement and it only allows sending one packet in a connection event.
- Unacknowledged GATT Write Request: a write request which does not require an acknowledgement. The current BLE stack implementation of Texas Instruments allows sending up to four packets in a connection interval using this type of requests.

Because we want to send more than one packet we chose the unacknowledged GATT write request in our experiments. `GATT_WriteNoRsp(BLEConnectionHandle, Request)` subprocedure was used to send the request. [7] This subprocedure has two input parameters which are the BLE connection handle and the request. The BLE connection handle is a parameter used to identify the connection, sometimes a master device can have multiple slaves and this parameter will be useful to identify different connections. The second parameter is the request of type `attWriteReq_t` which is type defined by Texas Instruments for write requests. The `attWriteReq_t` is based on a C programming language struct type which has the following members:

- Handle: handle of the attribute to be written
- Len: length of the value to be written
- Value: value of the attribute to be written
- Sig: authentication signature status (not included (0), valid (1), invalid (2)).
- Cmd: command Flag.

The second parameter in `GATT_WriteNoRsp` subprocedure requires the handle of the attribute to be defined. The handles for the attributes were discovered using Texas Instruments' BTool application. Figure 11 shows Texas Instruments' BTool application.

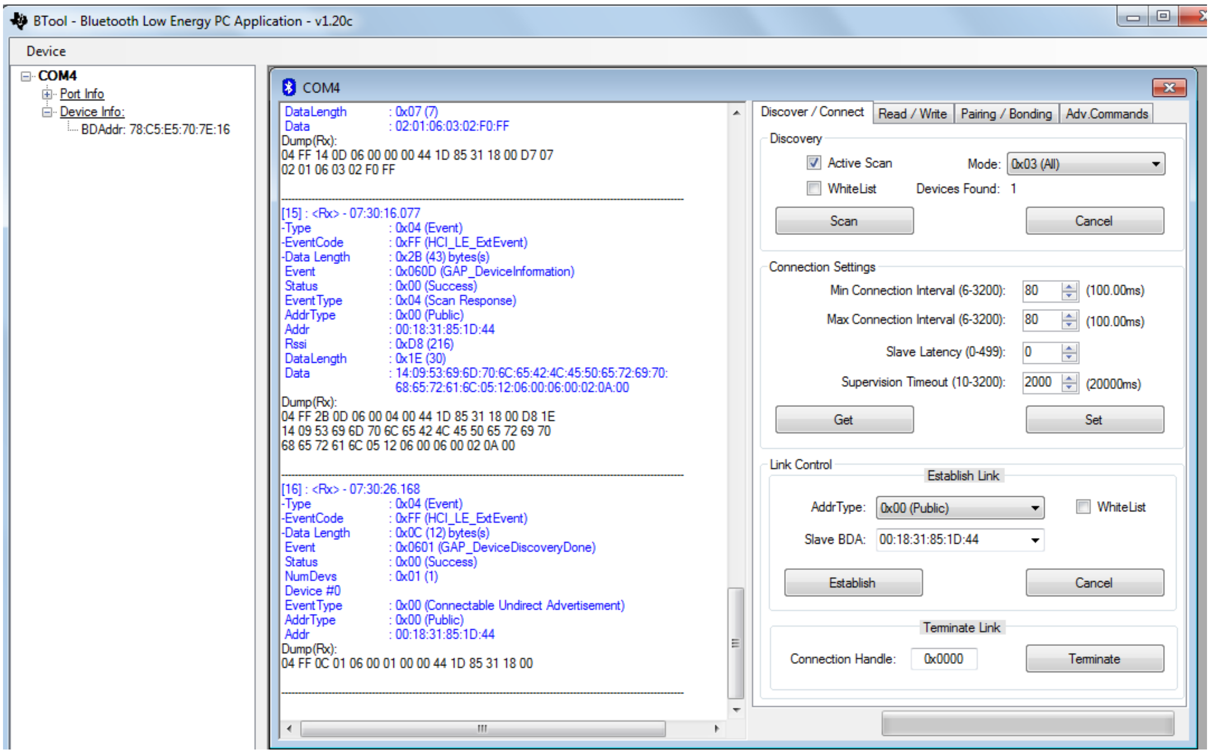

Figure 11 Texas Instruments' BTool application

BTool uses `GATT_DiscCharsByUUID` subprocedure to discover characteristics using the Unique Universal Identifier (UUID) which should be defined in the profile. Figure 12 shows the output of BTool to discover CHAR 1 handle.

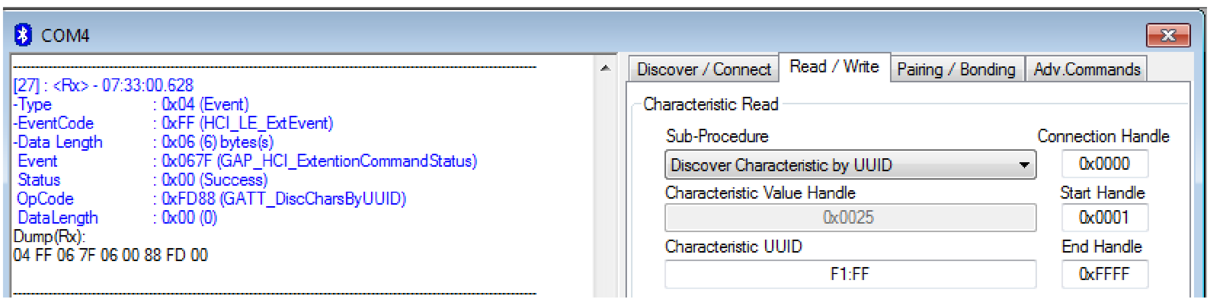

Figure 12 CHAR 1 Handle discovered with BTool

The handles for the nine defined characteristics were discovered using BTool. Table 3 shows the handles for the nine characteristics.

Table 3 Characteristics Handles

Characteristic	Handle
CHAR 1	0x0025
CHAR 2	0x0028
CHAR 3	0x002B
CHAR 4	0x002E
CHAR 5	0x0032
CHAR 6	0x0035
CHAR 7	0x0038
CHAR 8	0x003B
CHAR 9	0x003E

Studying the Effect of Connection Interval and Characteristics' Specifications on the Throughput

To measure the effect of the connection interval and the characteristics' specifications on the throughput, we performed three types of experiments: using a fixed connection interval and characteristic size and varying the number of characteristics, using a fixed connection interval and number of characteristics and varying the characteristic size, and using a fixed number of characteristics and characteristic size and varying the connection interval. We calculated the throughput for each case using Texas Instruments packet sniffer application. Figure 13 shows advertising packets captured with the packet sniffer and Figure 14 shows connection packets.

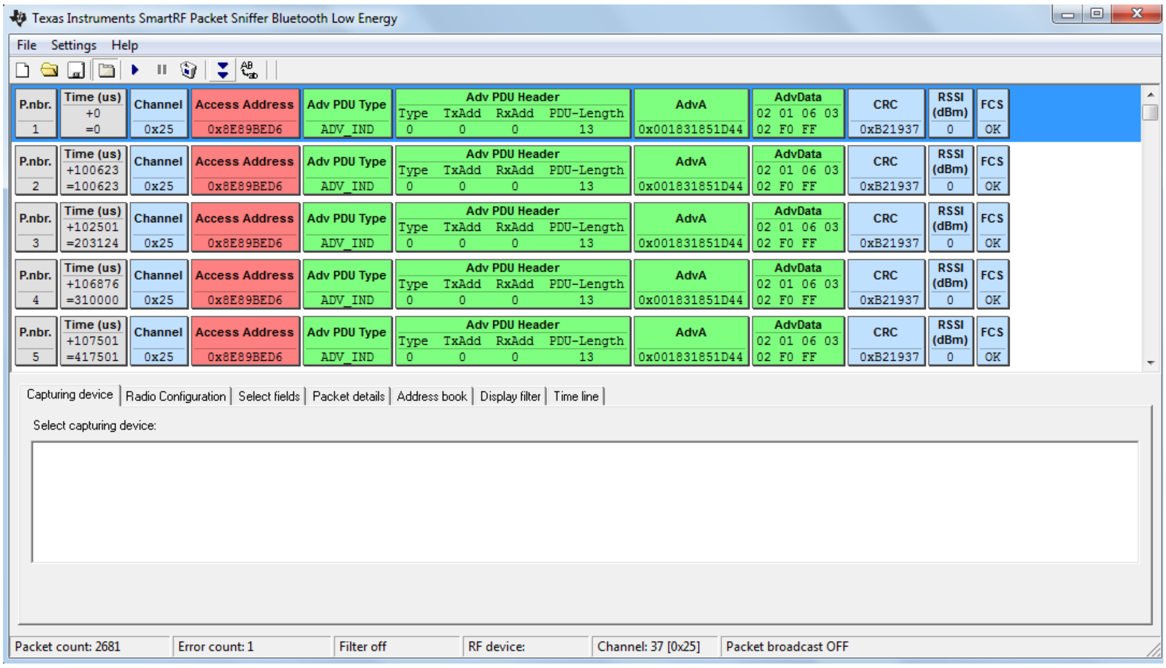

Figure 13 Texas Instruments Packet Sniffer Capturing Advertising Packets

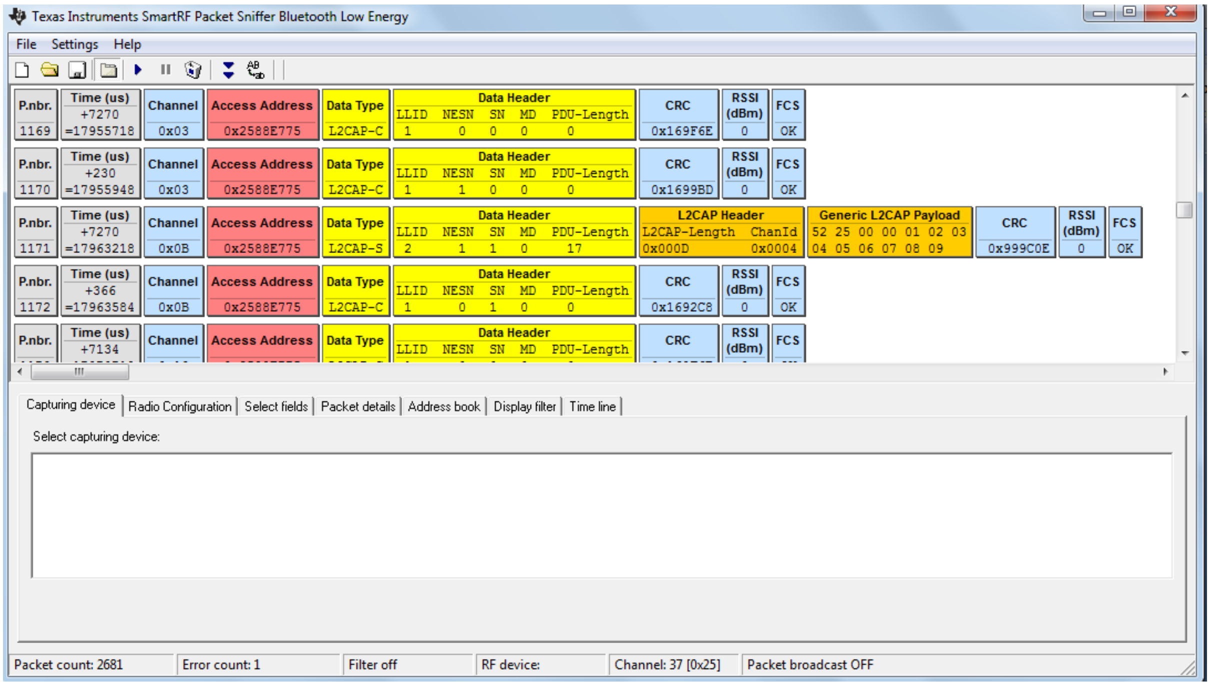

Figure 14 Connection Packets Captured with Texas Instruments' Packet Sniffer

Figure 13 shows advertising packets in green. It shows the packet number, packet time stamp, channel on which packet was sent, access address, advertising packet header type, the address of the advertising device, the advertised data, cyclic redundancy check (CRC), frame check sequence (FCS), and received signal strength indication (RSSI). It shows five packets sent on the same channel. Figure 14 shows data packets sent on different channels (0x03, 0x0B) because BLE uses frequency hopping. It shows the packet number, packet time stamp, channel on which packet was sent, access address, data type, data packet header, CRC, FCS and RSSI. The third packet contains an L2CAP header and L2CAP payload. The L2CAP payload contains the actual user data and the characteristic handle which was sent using the unacknowledged GATT write request. The data header contains the fields which were explained in chapter 2. The first packet has an LLID set to 1 because it is used to manage the connection, NESN set to 0, SN set to 0 and MD set to 0. The second packet is a reply for the first one, it has an LLID set to 1, NESN set to 1 (which acknowledges reception of packet with SN of 0), SN set to 0 and MD set to 0. The connection event is then closed and a channel is changed. The third packet contains the actual user data, we notice the time stamp difference between this packet and the previous one is 7270 microseconds which is huge compared to time stamp differences between packets sent in the same channel and this is due to the processing required for changing the channel. It has an LLID set to 2 because it is a complete packet used to send user data, NESN set to 1 (which acknowledges reception of packet with SN of 0), SN set to 1 and MD set to 0. The slave replies with the fourth packet which has an LLID set to 1, NESN set to 0 (which acknowledges reception of packet with SN of 1, SN set to 0 and MD set to 0. The connection event is then closed and the channel is changed.

To measure the throughput we used the below formula:

$$Throughput = User\ Data / Consumed\ Time$$

To get the consumed time to send the data we used packets time stamps. User data depends on the characteristic size and number of characteristics defined in the application.

Variable Number of Characteristics

To perform this test, we used the unacknowledged GATT write request to send data between the master and slave. We defined the characteristic size to be 10 bytes and the connection interval to be 7.5 ms (which is the minimum allowable value for the connection interval) and varied the number of characteristics from one to four and calculated the throughput in each case.

Variable Characteristic Size

This test consists of using a fixed connection interval and number of characteristics and varying the characteristic size. To perform the test we sent four unacknowledged GATT Write requests from a master device to a slave device and used a fixed connection interval of 7.5 ms. We varied the characteristic size and observed the corresponding throughput. The current maximum size of characteristics that can be used in Texas Instruments' CC2540 is 20 bytes. In our tests we were not able to send 20 bytes per characteristic and the maximum characteristic size we were able to write is 19 bytes. In this test we varied the characteristic size from a maximum value of 19 bytes to a minimum value of 5 bytes. The corresponding user data size would equal characteristic size × number of characteristics. Number of characteristics used in this case was four so user data in case of 19 bytes characteristics are 76 bytes and 20 bytes in case of 5 bytes of characteristics. The throughput was calculated by dividing the user data in bytes over the actual consumed time to send user data.

Variable Connection Interval

This test consists of using a fixed characteristic size and number of characteristics and varying the connection interval. To perform this test we sent four unacknowledged GATT write requests from a master device to a slave device using a fixed characteristic size of 19 bytes and varied the connection interval parameter from a minimum value of 7.5 ms to a maximum value of 60 ms and calculated the corresponding throughput.

Studying the Effect of Characteristics' Specifications on the Power Consumption in BLE applications

In BLE, power saving is achieved from the fact that the device will be sleeping most of the time between connection events. The power consumption varies during a connection event and an advertising event. To make adequate calculations regarding power consumption in BLE, we must find a parameter that takes into consideration the variability of current consumption between sleeping and connection modes and during a connection event. This parameter is called "average current while device is connected". [8]

Average current while connected can be calculated from the below formula:

Average current while connected =

$$\frac{[(connection\ Interval - Total\ Awake\ time) * (Average\ Sleep\ Current) + (Total\ Awake\ Time) * (Average\ Current\ during\ Connection\ Event)]}{Connection\ Interval}$$

[8]

The average current which can be used for battery life calculations depends greatly on the connection interval.

Test Bed Description

To study the effect of characteristics specifications on the power consumption we used Texas Instruments Bluetooth Low Energy CC2540 mini development Kit. Two devices were used one as a master and the other one as a slave. Using a multi-meter for average current consumption is not enough because current needs to be measured with respect to time that is why we used an oscilloscope in our measurements. We used an oscilloscope with voltage probe and a 10 ohms resistor to convert the voltage to current. Figure 15 shows the test bed block diagram using an oscilloscope and a voltage probe.

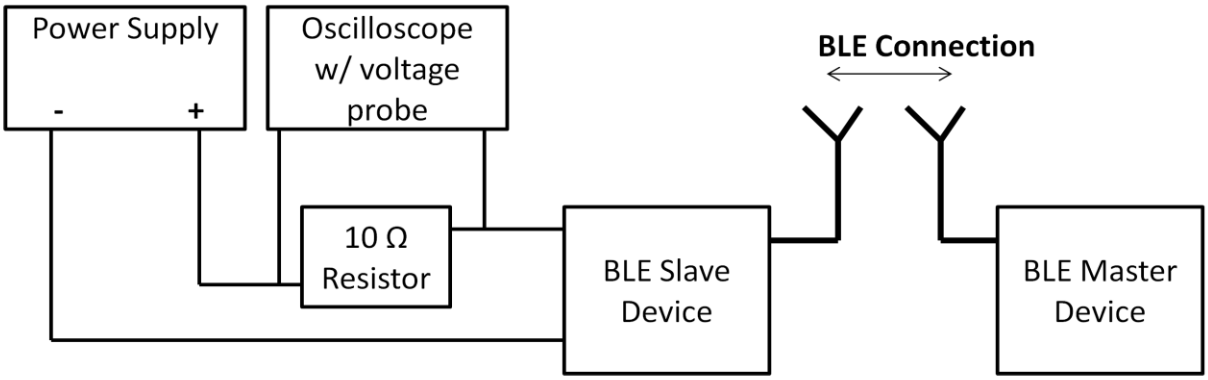

Figure 15 Test bed using an oscilloscope and a voltage probe [8]

Figure 16 shows Texas Instruments CC2540 mini key fob used a slave device in the power consumption measurements.

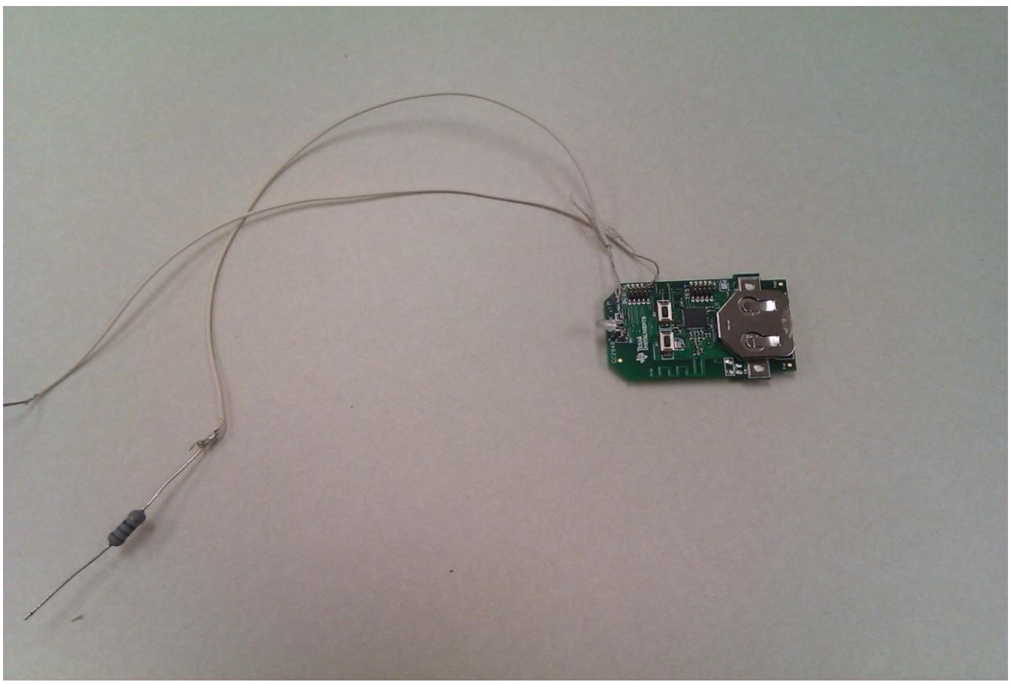

Figure 16 Texas Instruments CC2540 mini Key fob

We used the unacknowledged GATT write requests to write four characteristics from a master device running an application based on Texas Instruments SimpleBLECentral application to a slave device running an application based on Texas Instruments SimpleBLEPeripheral application and we observed the current signal on the oscilloscope. A fixed connection interval of 25 ms was chosen for the experiments and the characteristic size was changed each time in the slave device application and the length of user data was changed accordingly in the master device application and the corresponding current signal was recorded from the oscilloscope.

To calculate the average current while connected, we have to calculate the average current during connection event first. To calculate the average current during connection event we multiply the average value of amplitude for each state with the duration and divide by the total connection event duration. Figure 17 shows a typical current waveform during a connection event.

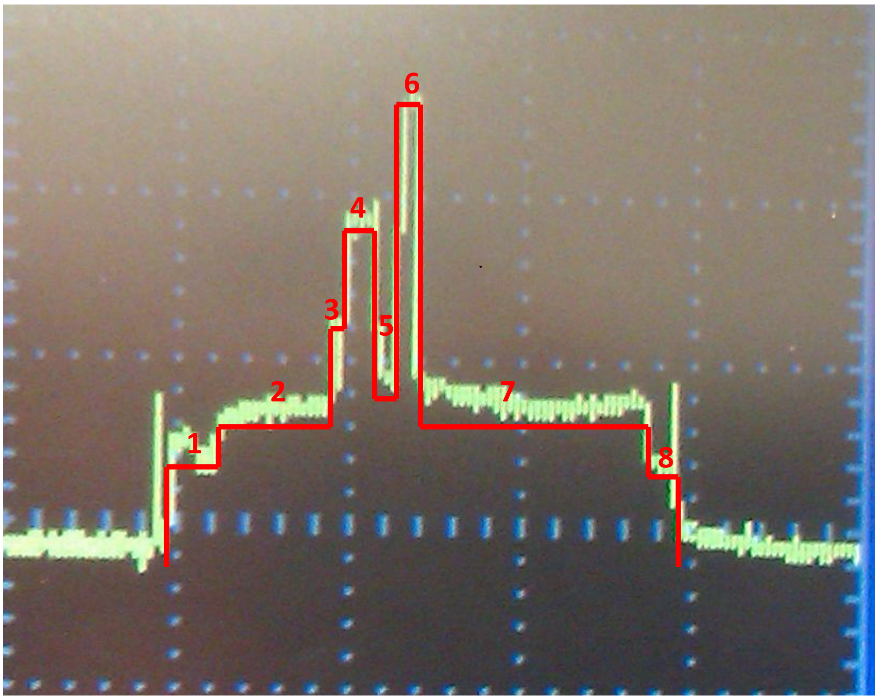

Figure 17 Current Waveform during a Connection Event

In Figure 17, there are eight states in that connection event. Each of these states has an average current and duration, to get the average current during a connection event we multiply the current of each state with its duration and divide by the total duration of the connection event. Below are the states shown in figure 17.

- State 1: Wake-up
- State 2: Pre-processing where the BLE stack prepares for transmission and reception of packets
- State 3: Pre-RX where the kit radio prepares for reception of packets
- State 4: RX where radio listens for packets
- State 5: RX-to-TX where radio switches from reception to transmission of packets
- State 6: TX where radio transmits packets
- State 7: Post-processing where the radio processes received packets and prepares for the next connection event by setting up the sleep timers
- State 8: Pre-sleep where sleep mode preparation is performed [8]

The sleep current was measured by using digital multi-meter (DMM) and setting the connection interval to 4 seconds. For each characteristic size the average current during a connection event was calculated and then the average current while connected was calculated.

To study the possibility of tradeoff between throughput and power consumption when increasing the characteristic size and number of characteristics in a BLE application a new parameter was introduced. This parameter is called BLE Throughput-Power Consumption Tradeoff (TPT) factor. TPT factor for a specific number of characteristics with a specific characteristic size= $\frac{Throughput}{Average\ Current\ while\ Connected}$

The TPT factor indicates the throughput that can be achieved by a specific current unit. The TPT factor was calculated for all characteristic sizes and number of characteristics to study the possibility of obtaining a higher value for the TPT factor with less number of characteristics or characteristic sizes. Additionally, the battery life of a typical button-cell battery with capacity of 230 mAh was calculated for different values of characteristic sizes and number of characteristics using the below formula:

Expected Battery Life when running continuously while in a connected state=

$$\frac{Battery\ Capacity}{Average\ Current\ Consumption\ while\ Connected}$$

CHAPTER 4

ANALYSIS OF RESULTS AND CALCULATIONS

In this chapter we present the results and calculations of our study.

Effect of Number of Characteristics, Characteristic Size and Connection Interval on the Throughput

As mentioned in Chapter 3, to study the effect of the connection interval and the characteristics' specifications on the throughput, we performed three types of experiments: using a fixed connection interval and characteristic size and varying the number of characteristics, using a fixed connection interval and number of characteristics and varying the characteristic size, and using a fixed number of characteristics and characteristic size and varying the connection interval.

Variable Number of Characteristics

The throughput was found to increase almost linearly with the number of characteristics. Figure 18 shows two graphs that explain the effect of increasing the number of characteristics on the throughput. The size of characteristic used in the green graph is 19 bytes and in the blue one is 10 bytes. We also notice that the throughput values for the 19 bytes characteristic graph are much higher than the 10 bytes characteristic graph.

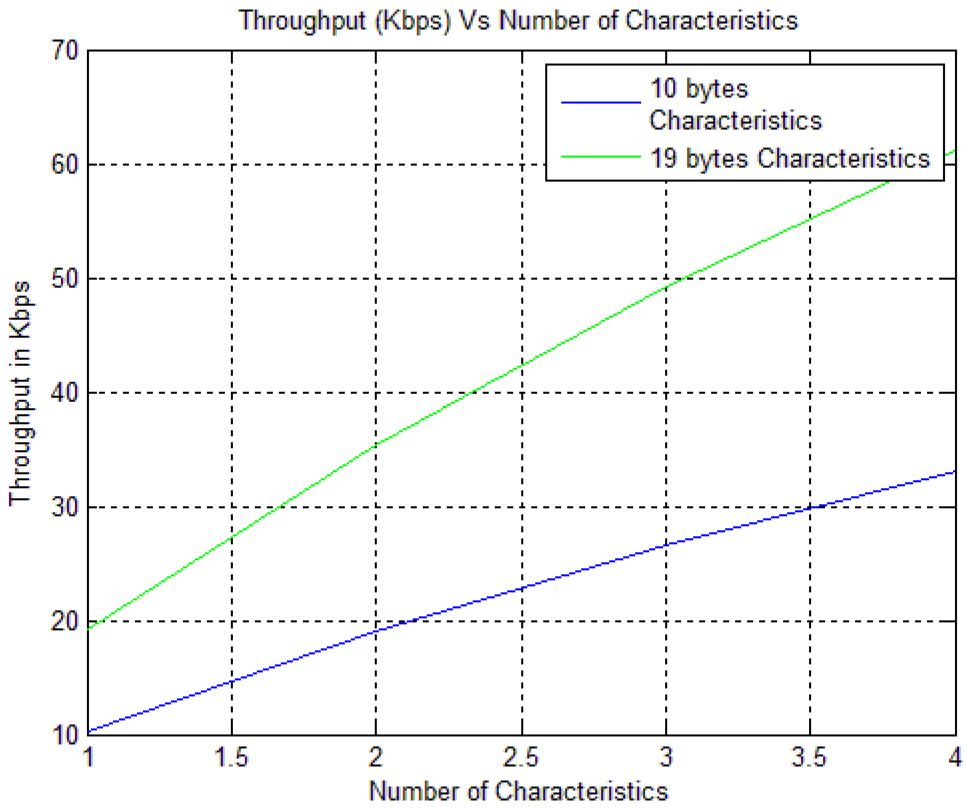

Figure 18 Throughput Vs Number of Characteristics

This result is very important to BLE systems developers and designers because it allows them to consider the effect of the number of characteristics on the throughput when designing their applications and systems.

Variable Characteristic Size

The throughput was found to increase almost linearly with the characteristic size used. Table 3 shows the throughput values for different characteristic sizes. It shows the characteristic size used ,the size of the user data in bytes which is equal to the characteristic size multiplied by the number of characteristics, the actual time consumed to send the data and the calculated throughput in Kilo Bytes per second (KBps) and Kilo bits per second (Kbps). This result is very important to BLE system developers and designers because it allows them to consider the effect of characteristic size on the throughput when designing their applications and systems.

Table 4 Throughput Vs Characteristic Size

Char Size	User Data (Bytes)	Actual Consumed Time (ms)	Throughput (KBps)	Throughput (Kbps)
19	76	9.715	7.64	61.12
17	68	9.652	6.88	55.04
14	56	9.556	5.72	45.78
11	44	9.460	4.54	36.33
8	32	9.365	3.34	26.70
5	20	9.268	2.11	16.86

Variable Connection Interval

The throughput was found to have an inverse relationship with the connection interval. The maximum value of throughput was obtained when the minimum allowed value of connection interval of 7.5 ms was used. Figure 19 shows the relationship between the throughput and connection interval.

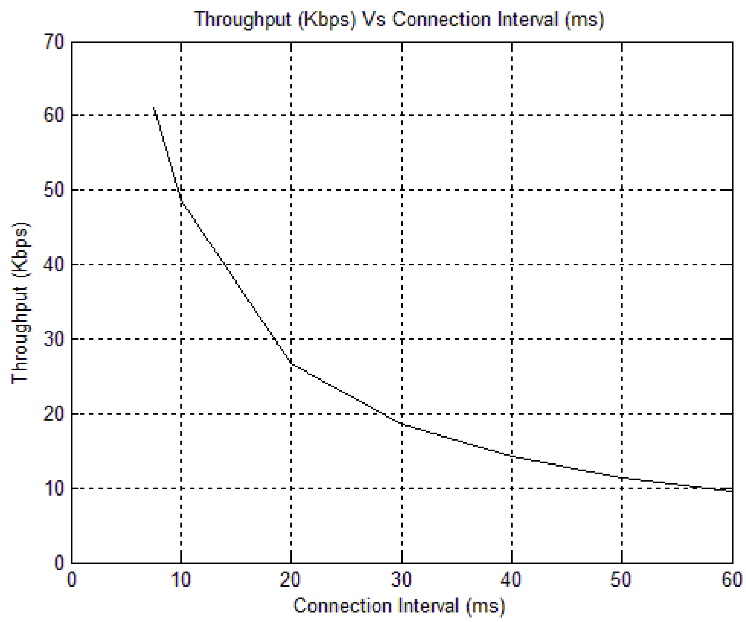

Figure 19 Throughput Vs Connection Interval

This result agrees with the simulation results provided by authors of reference [2] which states that connection interval and the throughput are inversely proportional to each other.

Effect of Characteristics Specifications on the Power Consumption

In this section we present the results and calculations of the effect of characteristic size and number of characteristics on the power consumption study.

As mentioned in chapter 3, we used four characteristics and a connection interval of 25 ms and changed the characteristic size and recorded the consumed current from the oscilloscope. Figure 20 shows the current waveform when writing four characteristics and using a characteristic size of 1 byte.

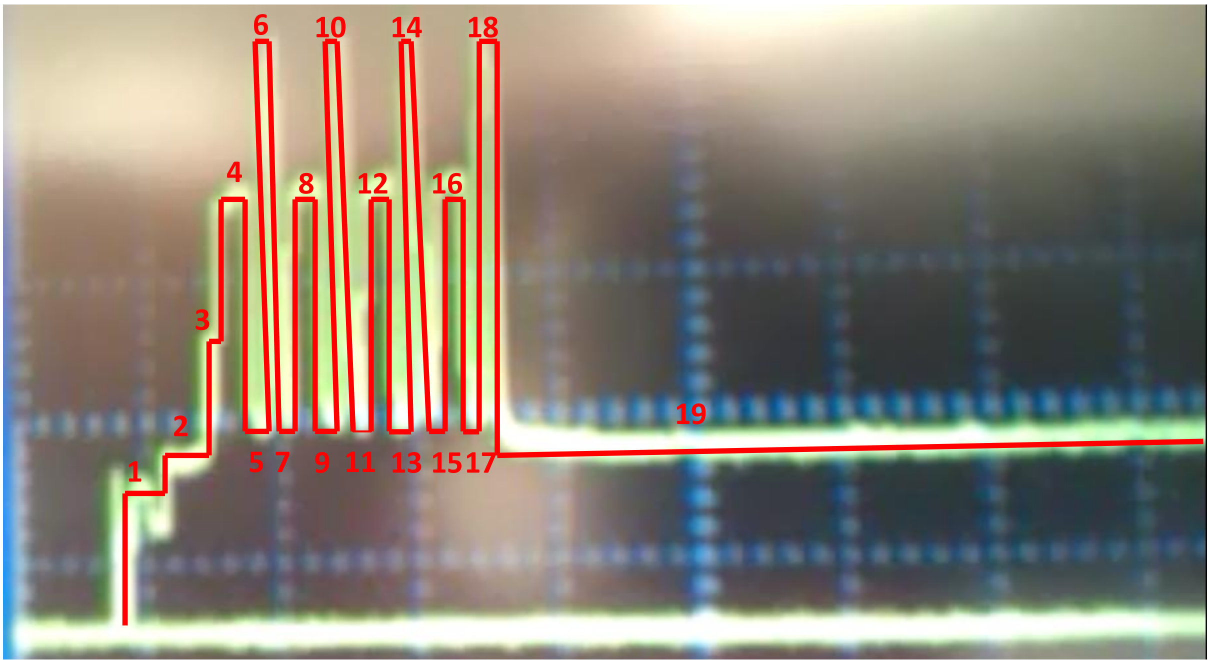

Figure 20 Writing Four 1 Byte Characteristics

Figure 21 shows the rest of the current waveform shown in figure 20.

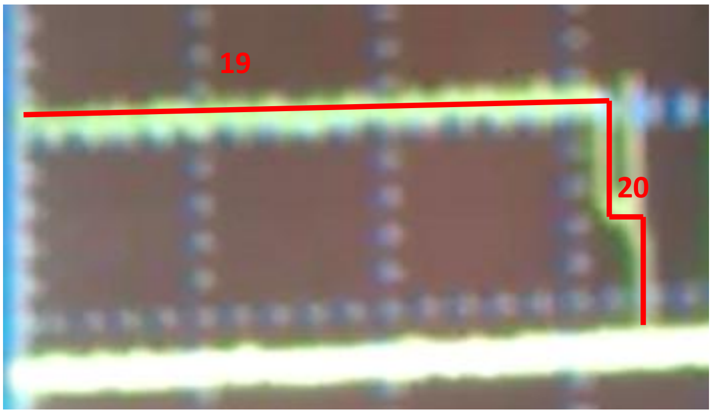

Figure 21 Post-processing and Pre-sleep States

Figures 20 and 21 shows a current waveform which has twenty states. Table 4 shows the different states of the waveform shown in figures 20 and 21.

Table 5 States of a Four Characteristics Current Waveform

State Number	State
1	Wake-up
2	Pre-processing
3	Pre-Rx
4	Rx (Char 1)
5	Rx to Tx (Char 1)
6	Tx (Char 1)
7	Tx to Rx (Char 1 to Char 2)
8	Rx (Char 2)
9	Rx to Tx (Char 2)
10	Tx (Char 2)
11	Tx to Rx (Char 2 to Char 3)
12	Rx (Char 3)
13	Rx to Tx (Char 3)
14	Tx (Char 3)
15	Tx to Rx (Char 3 to Char 4)
16	Rx (Char 4)
17	Rx to Tx (Char 4)
18	Tx (Char 4)
19	Post-Processing
20	Pre-Sleep

To calculate the average current during connection event we multiply the amplitude of the pulse of the specific state with its during and add the total values of the currents of each state and divide it by the total connection event duration as shown in the formula below.

$$\frac{\sum_{i=1}^{n} Ii * Ti}{Ttotal}$$

Where *Ii* is the amplitude of the pulse of state (*i*) and *Ti* is the duration of the pulse of state (*i*) and *Ttotal* is the total duration of the connection event.

To get the average current while the device is connected assuming that it is sending four 1-byte characteristics in each connection event we use the formula:

Average current while connected =

$$\frac{[(connection\ Interval - Total\ Awake\ time) * (Average\ Sleep\ Current) + (Total\ Awake\ Time) * (Average\ Current\ during\ Connection\ Event)]}{Connection\ Interval}$$

The average sleep current was found to be 0.001 mA and the connection interval used for all current measurement experiments is 25 ms. To get the average current while connected we substitute the average current during connection event and the total awake time in the above formula for each of the measurements of the characteristics' specifications effect on the power consumption.

Writing Four 1-Byte characteristics

Table 6 shows the calculated values for each of the current states for the waveform shown in figures 20 and 21.

Table 6 States for a four 1-Byte Characteristics Waveform

State	Time(Microseconds)	Current (mA)	Current * Time (mA * micro sec)
State 1 (Wake-up)	240	5.2	1248
State 2 (pre-processing)	326	6.05	1972.3
State 3 (pre-RX)	70	10.57	739.9
State 4 (Rx) :Char1	195	15.19	2962.05
State 5 (Rx-to-Tx): Char 1	90	7	630
State 6 (Tx):Char 1	140	20	2800
State 7 (Tx-to-RX) Char 1 to Char 2	60	7.2	432
State 8 (Rx) :Char2	195	15.19	2962.05
State 9 (Rx-to-Tx): Char 2	90	7	630
State 10 (Tx):Char 2	140	20	2800
State 11 (Tx-to-Rx): Char 2 to Char 3	60	7.2	432
State 12 (Rx) :Char3	195	15.19	2962.05
State 13 (Rx-to-Tx): Char 3	90	7	630
State 14 (Tx):Char 3	140	20	2800
State 15 (Tx-to-Rx) Char 3 to Char 4	60	7.2	432
State 16 (Rx) :Char4	195	15.19	2962.05
State 17 (Rx-to-Tx): Char 4	90	7	630
State 18 (Tx):Char 4	140	20	2800
State 19 (post-processing)	8461	3.46	29275.06
State 20 (pre-Sleep)	250	1.92	480
	11227		60579.46
Average Current (mA)	5.39587245		

The average current while connected was calculated for two values of the connection interval which are 25 ms and 100 ms. Table 7 shows the calculated average current while connected values.

Table 7 Calculated Average Current while connected for four 1-Byte Characteristics

Connection Interval (ms)	Average current while connected (mA)
25	2.423
100	0.606

Writing Four 3-Bytes Characteristics

Table 8 shows the calculated values for the different states of the waveform obtained when writing four 3-Bytes characteristics.

Table 8 States for Four 3-Bytes Characteristics Waveform

State	Time(microseconds)	Current (mA)	Current * Time (mA*micro sec)
State 1 (Wake-up)	227	5.34	1212.18
State 2 (pre-processing)	300	6.02	1806
State 3 (pre-RX)	75	10.22	766.5
State 4 (Rx) :Char1	250	15	3750
State 5 (Rx-to-Tx): Char 1	91	6.6	600.6
State 6 (Tx):Char 1	136	20	2720
State 7 (Tx-to-RX) Char 1 to Char 2	75	6.8	510
State 8 (Rx) :Char2	250	15	3750
State 9 (Rx-to-Tx): Char 2	91	6.6	600.6
State 10 (Tx):Char 2	136	20	2720
State 11 (Tx-to-Rx): Char 2 to Char 3	75	6.8	510
State 12 (Rx) :Char3	250	15	3750
State 13 (Rx-to-Tx): Char 3	91	6.6	600.6
State 14 (Tx):Char 3	136	20	2720
State 15 (Tx-to-Rx) Char 3 to Char 4	75	6.8	510
State 16 (Rx) :Char4	250	15	3750
State 17 (Rx-to-Tx): Char 4	91	6.6	600.6
State 18 (Tx):Char 4	136	20	2720
State 19 (post-processing)	8461	3.46	29275.06
State 20 (pre-Sleep)	250	1.92	480
	11446		63352.14
Average Current (mA)	5.534871571		

The average current while connected was calculated for two values of the connection interval which are 25 ms and 100 ms. Table 9 shows the calculated average current while connected values.

Table 9 Calculated Average Current while Connected for four 3-Bytes Characteristics

Connection Interval (ms)	Average current while connected (mA)
25	2.534
100	0.634

Writing Four 5-Bytes Characteristics

Table 10 shows the calculated values for the different states of the waveform obtained when writing four 5-Bytes characteristics.

Table 10 States for Four 5-Bytes Characteristics Waveform

State	Time(Microseconds)	Current (mA)	Current * Time (mA * micro sec)
State 1 (Wake-up)	265	5	1325
State 2 (pre-processing)	178	5.9	1050.2
State 3 (pre-RX)	54	10.7	577.8
State 4 (Rx) :Char1	212	15.31	3245.72
State 5 (Rx-to-Tx): Char 1	107	7.14	763.98
State 6 (Tx):Char 1	170	20	3400
State 7 (Tx-to-RX) Char 1 to Char 2	54	7.2	388.8
State 8 (Rx) :Char2	212	15.31	3245.72
State 9 (Rx-to-Tx): Char 2	107	7.14	763.98
State 10 (Tx):Char 2	170	20	3400
State 11 (Tx-to-Rx): Char 2 to Char 3	54	7.2	388.8
State 12 (Rx) :Char3	212	15.31	3245.72
State 13 (Rx-to-Tx): Char 3	107	7.14	763.98
State 14 (Tx):Char 3	170	20	3400
State 15 (Tx-to-Rx) Char 3 to Char 4	54	7.2	388.8
State 16 (Rx) :Char4	212	15.31	3245.72
State 17 (Rx-to-Tx): Char 4	107	7.14	763.98
State 18 (Tx):Char 4	170	20	3400
State 19 (post-processing)	8461	3.46	29275.06
State 20 (pre-Sleep)	250	1.92	480
	11326		63513.26
Average Current (mA)	5.607739714		

The average current while connected was calculated for two values of the connection interval which are 25 ms and 100 ms. Table 11 shows the calculated average current while connected values.

Table 11 Calculated Average Current while Connected for four 5-Bytes Characteristics

Connection Interval (ms)	Average current while connected (mA)
25	2.541
100	0.636

Writing Four 7-Bytes Characteristics

Table 12 shows the calculated values for the different states of the waveform obtained when writing four 7-Bytes characteristics.

Table 12 States for a Four 7-Bytes Characteristics Waveform

State	Time(Microseconds)	Current (mA)	Current (mA) * Time (micro sec)
State 1 (Wake-up)	240	5	1200
State 2 (pre-processing)	260	5.8	1508
State 3 (pre-RX)	70	9.4	658
State 4 (Rx) :Char1	240	15.2	3648
State 5 (Rx-to-Tx): Char 1	115	6.8	782
State 6 (Tx):Char 1	155	20	3100
State 7 (Tx-to-RX) Char 1 to Char 2	60	6.8	408
State 8 (Rx) :Char2	240	15.2	3648
State 9 (Rx-to-Tx): Char 2	115	6.8	782
State 10 (Tx):Char 2	155	20	3100
State 11 (Tx-to-Rx): Char 2 to Char 3	60	6.8	408
State 12 (Rx) :Char3	240	15.2	3648
State 13 (Rx-to-Tx): Char 3	115	6.8	782
State 14 (Tx):Char 3	155	20	3100
State 15 (Tx-to-Rx) Char 3 to Char 4	60	6.8	408
State 16 (Rx) :Char4	240	15.2	3648
State 17 (Rx-to-Tx): Char 4	115	6.8	782
State 18 (Tx):Char 4	155	20	3100
State 19 (post-processing)	8461	3.46	29275.06
State 20 (pre-Sleep)	250	1.92	480
	11501		64465.06
Average Current (mA)	5.605169985		

The average current while connected was calculated for two values of the connection interval which are 25 ms and 100 ms. Table 13 shows the calculated average current while connected values.

Table 13 Calculated Average Current while Connected for four 7-Bytes Characteristics

Connection Interval (ms)	Average current while connected (mA)
25	2.579
100	0.645

Writing Four 9-Bytes Characteristics

Table 14 shows the calculated values for the different states of the waveform obtained when writing four 9-Bytes characteristics.

Table 14 States for a Four 9-Bytes Characteristics Waveform

State	Time(Microseconds)	Current (mA)	Current (mA) * Time (micro sec)
State 1 (Wake-up)	308	5.2	1601.6
State 2 (pre-processing)	380	6.2	2356
State 3 (pre-RX)	108	10.4	1123.2
State 4 (Rx) :Char1	240	15.6	3744
State 5 (Rx-to-Tx): Char 1	100	7	700
State 6 (Tx):Char 1	145	20	2900
State 7 (Tx-to-RX) Char 1 to Char 2	85	7.2	612
State 8 (Rx) :Char2	240	15.6	3744
State 9 (Rx-to-Tx): Char 2	100	7	700
State 10 (Tx):Char 2	145	20	2900
State 11 (Tx-to-Rx): Char 2 to Char 3	85	7.2	612
State 12 (Rx) :Char3	240	15.6	3744
State 13 (Rx-to-Tx): Char 3	100	7	700
State 14 (Tx):Char 3	145	20	2900
State 15 (Tx-to-Rx) Char 3 to Char 4	85	7.2	612
State 16 (Rx) :Char4	240	15.6	3744
State 17 (Rx-to-Tx): Char 4	100	7	700
State 18 (Tx):Char 4	145	20	2900
State 19 (post-processing)	8461	3.46	29275.06
State 20 (pre-Sleep)	250	1.92	480
	11702		66047.86
Average Current (mA)	5.644151427		

The average current while connected was calculated for two values of the connection interval which are 25 ms and 100 ms. Table 15 shows the calculated average current while connected values.

Table 15 Calculated Average Current while Connected for four 9-Bytes Characteristics

Connection Interval (ms)	Average current while connected (mA)
25	2.642
100	0.661

Writing Four 11-Bytes Characteristics

Table 16 shows the calculated values for the different states of the waveform obtained when writing four 11-Bytes characteristics.

Table 16 States for a Four 11-Bytes Characteristics Waveform

State	Time(Microseconds)	Current (mA)	Current * Time (mA * micro sec)
State 1 (Wake-up)	265	4.78	1266.7
State 2 (pre-processing)	345	5.55	1914.75
State 3 (pre-RX)	110	10	1100
State 4 (Rx) :Char1	304	15.21	4623.84
State 5 (Rx-to-Tx): Char 1	125	6.73	841.25
State 6 (Tx):Char 1	155	19.4	3007
State 7 (Tx-to-RX) Char 1 to Char 2	100	7.17	717
State 8 (Rx) :Char2	304	15.21	4623.84
State 9 (Rx-to-Tx): Char 2	125	6.73	841.25
State 10 (Tx):Char 2	155	19.4	3007
State 11 (Tx-to-Rx): Char 2 to Char 3	100	7.17	717
State 12 (Rx) :Char3	304	15.21	4623.84
State 13 (Rx-to-Tx): Char 3	125	6.73	841.25
State 14 (Tx):Char 3	155	19.4	3007
State 15 (Tx-to-Rx) Char 3 to Char 4	100	7.17	717
State 16 (Rx) :Char4	304	15.21	4623.84
State 17 (Rx-to-Tx): Char 4	125	6.73	841.25
State 18 (Tx):Char 4	155	19.4	3007
State 19 (post-processing)	8461	3.46	29275.06
State 20 (pre-Sleep)	250	1.92	480
	12067		70075.87
Average Current (mA)	5.807232121		

The average current while connected was calculated for two values of the connection interval which are 25 ms and 100 ms. Table 17 shows the calculated average current while connected values.

Table 17 Calculated Average Current while Connected for four 11-Bytes Characteristics

Connection Interval (ms)	Average current while connected (mA)
25	2.803
100	0.701

Writing Four 13-Bytes Characteristics

Table 18 shows the calculated values for the different states of the waveform obtained when writing four 13-Bytes characteristics.

Table 18 States for a Four 13-Bytes Characteristics Waveform

State	Time(Microseconds)	Current (mA)	Current * Time (mA * micro sec)
State 1 (Wake-up)	275	4.67	1284.25
State 2 (pre-processing)	260	5.65	1469
State 3 (pre-RX)	100	10	1000
State 4 (Rx) :Char1	333	15.19	5058.27
State 5 (Rx-to-Tx): Char 1	125	6.92	865
State 6 (Tx):Char 1	150	19.23	2884.5
State 7 (Tx-to-RX) Char 1 to Char 2	96	6.73	646.08
State 8 (Rx) :Char2	333	15.19	5058.27
State 9 (Rx-to-Tx): Char 2	125	6.92	865
State 10 (Tx):Char 2	150	19.23	2884.5
State 11 (Tx-to-Rx): Char 2 to Char 3	96	6.73	646.08
State 11 (Rx) :Char3	333	15.19	5058.27
State 12 (Rx-to-Tx): Char 3	125	6.92	865
State 13 (Tx):Char 3	150	19.23	2884.5
State 14 (Tx-to-Rx) Char 3 to Char 4	96	6.73	646.08
State 15 (Rx) :Char4	333	15.19	5058.27
State 16 (Rx-to-Tx): Char 4	125	6.92	865
State 16 (Tx):Char 4	150	19.23	2884.5
State 17 (post-processing)	8461	3.46	29275.06
State 18 (pre-Sleep)	250	1.92	480
	12066		70677.63
Average Current (mA)	5.857585778		

The average current while connected was calculated for two values of the connection interval which are 25 ms and 100 ms. Table 19 shows the calculated average current while connected values.

Table 19 Calculated Average Current while Connected for four 13-Bytes Characteristics

Connection Interval (ms)	Average current while connected (mA)
25	2.827
100	0.707

Writing Four 15-Bytes Characteristics

Table 20 shows the calculated values for the different states of the waveform obtained when writing four 15-Bytes characteristics.

Table 20 States for a Four 15-Bytes Characteristics Waveform

State	Time(microseconds)	Current (mA)	Current * Time (mA* micro sec)
State 1 (Wake-up)	275	4.44	1221
State 2 (pre-processing)	333	5.55	1848.15
State 3 (pre-RX)	115	10	1150
State 4 (Rx) :Char1	360	15.18	5464.8
State 5 (Rx-to-Tx): Char 1	110	6.48	712.8
State 6 (Tx):Char 1	160	19.25	3080
State 7 (Tx-to-RX) Char 1 to Char 2	115	6.66	765.9
State 8 (Rx) :Char2	360	15.18	5464.8
State 9 (Rx-to-Tx): Char 2	110	6.48	712.8
State 10 (Tx):Char 2	160	19.25	3080
State 11 (Tx-to-Rx): Char 2 to Char 3	115	6.66	765.9
State 12 (Rx) :Char3	360	15.18	5464.8
State 13 (Rx-to-Tx): Char 3	110	6.48	712.8
State 14 (Tx):Char 3	160	19.25	3080
State 15 (Tx-to-Rx) Char 3 to Char 4	115	6.66	765.9
State 16 (Rx) :Char4	360	15.18	5464.8
State 17 (Rx-to-Tx): Char 4	110	6.48	712.8
State 18 (Tx):Char 4	160	19.25	3080
State 19 (post-processing)	8461	3.46	29275.06
State 20 (pre-Sleep)	250	1.92	480
	12299		73302.31
Average Current (mA)	5.960021953		

The average current while connected was calculated for two values of the connection interval which are 25 ms and 100 ms. Table 21 shows the calculated average current while connected values.

Table 21 Calculated Average Current while Connected for four 15-Bytes Characteristics

Connection Interval (ms)	Average current while connected (mA)
25	2.932
100	0.733

Writing Four 17-Bytes Characteristics

Table 22 shows the calculated values for the different states of the waveform obtained when writing four 17-Bytes characteristics.

Table 22 States for a Four 17-Bytes Characteristics Waveform

State	Time(microseconds)	Current (mA)	Current * Time (mA * micro sec)
State 1 (Wake-up)	285	4.9	1396.5
State 2 (pre-processing)	205	5.92	1213.6
State 3 (pre-RX)	85	9.8	833
State 4 (Rx) :Char1	367	15.3	5615.1
State 5 (Rx-to-Tx): Char 1	125	6.53	816.25
State 6 (Tx):Char 1	163	19.3	3145.9
State 7 (Tx-to-RX) Char 1 to Char 2	85	6.3	535.5
State 8 (Rx) :Char2	367	15.3	5615.1
State 9 (Rx-to-Tx): Char 2	125	6.53	816.25
State 10 (Tx):Char 2	163	19.3	3145.9
State 11 (Tx-to-Rx): Char 2 to Char 3	85	6.3	535.5
State 12 (Rx) :Char3	367	15.3	5615.1
State 13 (Rx-to-Tx): Char 3	125	6.53	816.25
State 14 (Tx):Char 3	163	19.3	3145.9
State 15 (Tx-to-Rx) Char 3 to Char 4	85	6.3	535.5
State 16 (Rx) :Char4	367	15.3	5615.1
State 17 (Rx-to-Tx): Char 4	125	6.53	816.25
State 18 (Tx):Char 4	163	19.3	3145.9
State 19 (post-processing)	8461	3.46	29275.06
State 20 (pre-Sleep)	250	1.92	480
	12161		73113.66
Average Current (mA)	6.012142094		

The average current while connected was calculated for two values of the connection interval which are 25 ms and 100 ms. Table 23 shows the calculated average current while connected values.

Table 23 Calculated Average Current while Connected for four 17-Bytes Characteristics

Connection Interval (ms)	Average current while connected (mA)
25	2.925
100	0.732

Writing Four 19-Bytes Characteristics

Table 24 shows the calculated values for the different states of the waveform obtained when writing four 19-Bytes characteristics.

Table 24 States for a Four 19-Bytes Characteristics Waveform

State	Time(Microseconds)	Current (mA)	Current * Time (mA * micro sec)
State 1 (Wake-up)	333	5	1665
State 2 (pre-processing)	319	6.19	1974.61
State 3 (pre-RX)	95	10.71	1017.45
State 4 (Rx) :Char1	365	16.32	5956.8
State 5 (Rx-to-Tx): Char 1	131	7.02	919.62
State 6 (Tx):Char 1	190	20.95	3980.5
State 7 (Tx-to-RX) Char 1 to Char 2	119	3.3	392.7
State 8 (Rx) :Char2	365	16.32	5956.8
State 9 (Rx-to-Tx): Char 2	131	7.02	919.62
State 10 (Tx):Char 2	190	20.95	3980.5
State 11 (Tx-to-Rx): Char 2 to Char 3	119	3.3	392.7
State 11 (Rx) :Char3	365	16.32	5956.8
State 12 (Rx-to-Tx): Char 3	131	7.02	919.62
State 13 (Tx):Char 3	190	20.95	3980.5
State 14 (Tx-to-Rx) Char 3 to Char 4	119	3.3	392.7
State 15 (Rx) :Char4	365	16.32	5956.8
State 16 (Rx-to-Tx): Char 4	131	7.02	919.62
State 16 (Tx):Char 4	190	20.95	3980.5
State 17 (post-processing)	8461	3.46	29275.06
State 18 (pre-Sleep)	250	1.92	480
	12559		79017.9
Average Current (mA)	6.291735011		

The average current while connected was calculated for two values of the connection interval which are 25 ms and 100 ms. Table 25 shows the calculated average current while connected values.

Table 25 Calculated Average Current while Connected for Four 19-Bytes Characteristics

Connection Interval (ms)	Average current while connected (mA)
25	3.161
100	0.791

The average current during a connection event and the average current consumed while connected increase when we increase the characteristic size in a BLE application. Increasing the connection interval results in a smaller value of average current while connected because the device will be in sleep mode for a longer duration of time.

Figure 22 shows the relationship between the characteristic size and average current while connected when using a connection interval of 25 ms.

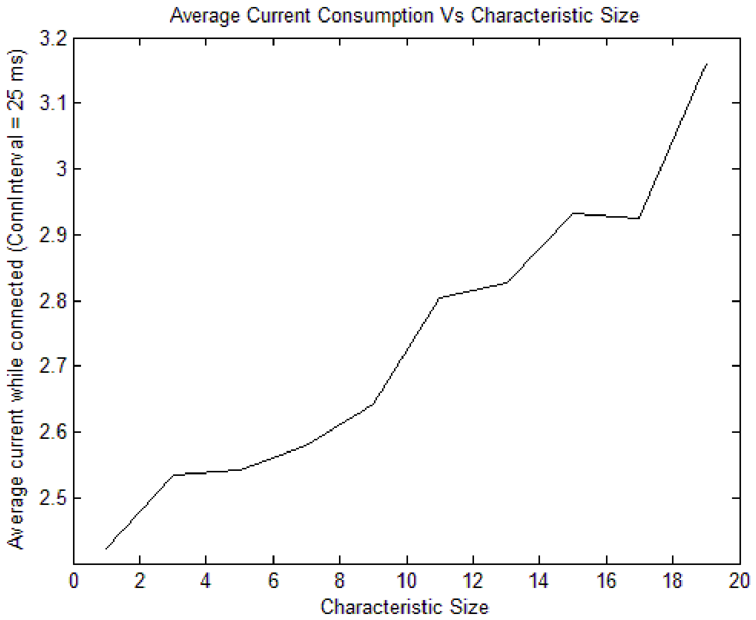

Figure 22 Avg. Current Consumption Vs Characteristic Size (Connection Interval = 25 ms)

Figure 23 shows the relationship between the characteristic size and average current while connected when using a connection interval of 100 ms.

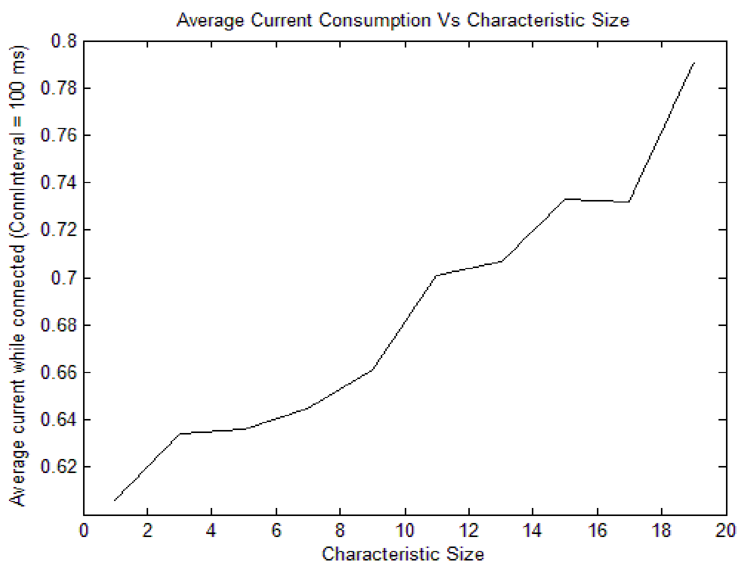

Figure 23 Avg. Current Consumption Vs Characteristic Size (Connection Interval = 100 ms)

Throughput-Power Consumption Tradeoff (TPT) Factor and Battery Life Calculations

In this section we present the results and calculations for the TPT factor and battery life. The TPT factor was calculated using the maximum achievable throughput using the specific number of characteristics and characteristic size and the average current consumption while connected.

TPT Factor for different Characteristic Sizes

Table 26 shows the calculated TPT factor in KBps/mA and in Kbps/mA using the average current consumed while connected calculated using a connection interval of 25 ms.

Table 26 Tradeoff Factor Calculated using connection Interval of 25 ms

Characteristic Size	Throughput (Kbps)	Throughput (KBps)	Avg. Current While Connected (25 ms)	Trade off Factor (KBps/mA)	Trade off Factor (Kbps/mA)
1	1.186	0.14825	2.423	0.061184482	0.489475856
3	3.552	0.444	2.534	0.175217048	1.401736385
5	5.909	0.738625	2.541	0.290682802	2.325462416
7	8.258	1.03225	2.579	0.400252036	3.202016285
9	10.598	1.32475	2.642	0.501419379	4.011355034
11	12.93	1.61625	2.803	0.576614342	4.612914734
13	15.254	1.90675	2.827	0.674478245	5.395825964
15	17.569	2.196125	2.932	0.749019441	5.992155525
17	19.876	2.4845	2.925	0.849401709	6.795213675
19	22.173	2.771625	3.161	0.876819045	7.014552357

Table 27 shows the calculated TPT factor in KBps/mA and in Kbps/mA using the average current consumed while connected calculated using a connection interval of 100 ms.

Table 27 Tradeoff Factor Calculated using connection Interval of 100 ms

Characteristic Size	Throughput (Kbps)	Throughput (KBps)	Avg. Current While Connected (100 ms)	Trade off Factor (KBps/mA)	Trade off Factor (Kbps/mA)
1	1.186	0.14825	0.606	0.244636964	1.95709571
3	3.552	0.444	0.634	0.700315457	5.602523659
5	5.909	0.738625	0.636	1.161360063	9.290880503
7	8.258	1.03225	0.645	1.600387597	12.80310078
9	10.598	1.32475	0.661	2.004160363	16.0332829
11	12.93	1.61625	0.701	2.305634807	18.44507846
13	15.254	1.90675	0.707	2.696958982	21.57567185
15	17.569	2.196125	0.733	2.996077763	23.9686221
17	19.876	2.4845	0.732	3.394125683	27.15300546
19	22.173	2.771625	0.791	3.503950695	28.03160556

As shown in tables 26 and 27, the TPT factor increases significantly by increasing the characteristic size.

TPT Factor for Different Number of Characteristics

Table 28 shows the TPT factor for different number of characteristics obtained using the average current while connected calculated using a connection interval of 25 ms.

Table 28 TPT Factor for Different Number of Characteristics (Connection Interval= 25 ms)

Number of Characteristics	Throughput (Kbps)	Throughput (KBps)	Avg. Current Consumption while Connected(mA) (25 ms)	Trade off Factor (Kbps/mA)	Trade off Factor (KBps/mA)
1	5.9928	0.7491	1.811	3.309110988	0.413638874
2	11.6705	1.4588125	2.261	5.161654135	0.645206767
3	17.05655	2.13206875	2.711	6.291608263	0.786451033
4	22.1747	2.7718375	3.161	7.015090161	0.87688627

Table 29 shows the TPT factor for different number of characteristics obtained using the average current while connected calculated using a connection interval of 100 ms.

Table 29 TPT Factor for Different Number of Characteristics (Connection Interval= 100 ms)

Number of Characteristics	Throughput (Kbps)	Throughput (KBps)	Avg. Current Consumption while Connected(mA) (100 ms)	Trade off Factor (Kbps/mA)	Trade off Factor (KBps/mA)
1	5.9928	0.7491	0.453	13.22913907	1.653642384
2	11.6705	1.4588125	0.566	20.61925795	2.577407244
3	17.05655	2.1320687 5	0.678	25.15715339	3.144644174
4	22.1747	2.7718375	0.791	28.03375474	3.504219343

As shown in tables 28 and 29, the TPT factor increases by increasing the number of characteristics used in a BLE application.

From tables 28, 29, 30, and 31 we notice that the TPT factor increases significantly by increasing the connection interval because of the decrease in the power consumption.

Battery Life Calculations

In this subsection we present the results of the calculated battery life for different characteristic sizes and number of characteristics.

Battery Life for Different Characteristic Sizes

Table 30 shows the battery life values for different characteristics sizes obtained by using the average current calculated with connection intervals of 25 ms and 100 ms.

Table 30 Battery Life for Different Characteristic Sizes

Char Size	Avg. Current Consumption While Connected (100 ms)	Avg. Current Consumption While Connected (25 ms)	Battery Life when using 25 ms (hours)	Battery Life when using 100 ms (hours)
1	0.606	2.423	94.924	379.538
3	0.634	2.534	90.766	362.776
5	0.636	2.541	90.516	361.635
7	0.645	2.579	89.182	356.589
9	0.661	2.642	87.055	347.958
11	0.701	2.803	82.055	328.103
13	0.707	2.827	81.358	325.318
15	0.733	2.932	78.445	313.779
17	0.732	2.925	78.632	314.208
19	0.791	3.161	72.761	290.771

As shown in table 31, the battery life of a BLE device depends greatly on both the connection interval and the characteristic size used in a BLE application. Increasing the characteristic size decreases the battery life significantly, almost 24% when increasing the characteristic size from one to nineteen bytes. Additionally, increasing the connection interval increases the battery life significantly for example when using 19 bytes characteristics battery life increases by 300 % when increasing the connection interval from 25 ms to 100 ms.

Battery Life for Different Number of Characteristics

Table 32 shows the battery life values for different number of characteristics obtained by using the average current calculated with connection intervals of 25 ms and 100 ms.

Table 31 Battery Life for Different Number of Characteristics

Number of Characteristics	Avg. Current Consumption While Connected (100 ms)	Avg. Current Consumption While Connected (25 ms)	Battery Life when using 25 ms (hours)	Battery Life when using 100 ms (hours)
1	0.453	1.811	127.0016565	507.7262693
2	0.566	2.261	101.7249005	406.360424
3	0.678	2.711	84.8395426	339.2330383
4	0.791	3.161	72.76178425	290.7711757

As shown in table 31, we observe that the battery life almost goes down to half when we increase the number of characteristics in a BLE application from one to four.

CHAPTER 5

CONCLUSION

This work takes an in-depth exploration into the technology of BLE by studying the effect of different parameters related to the BLE connection and the Generic Attribute Profile (GATT) protocol on the throughput and the power consumption of the system. To achieve this goal the packet structure, the connection parameters, the protocol stack, and the application structure have been studied.

The characteristic size and the number of characteristics used in a BLE application and the throughput were found to be almost linearly related. Additionally, the throughput was found to have an inverse relationship with the connection interval. The average current consumed while the device is connected was measured for different characteristic sizes and number of characteristics and was found to have proportional relationships with these two variables.

One of the goals of the study was to find if the gain in throughput by increasing the number of characteristics and characteristic size can be outperformed by the gain in power saving at a specific number of characteristics or characteristic size used in a BLE application. To achieve this goal, a parameter known as the throughput-power consumption trade off (TPT) factor was defined and measured for different number of characteristics and characteristic size. The TPT factor was found to have a proportional relationship with the number of characteristics and characteristic sizes which means that the gain in throughput improvement obtained by increasing the number of characteristics or characteristic size outperforms the gain in power saving obtained by decreasing the number of characteristics or characteristic size.

The battery life was calculated when using different number of characteristics and characteristic sizes and found to decrease by 24% when increasing the characteristic size from one to nineteen bytes and by 43% when increasing the number of characteristics from one to four. The results of this study are very important to BLE system developers and designers because when taking these facts into consideration they can design their applications in a way that suits their needs optimally.

Following the approach in this study, BLE system designers can calculate different parameters related to the performance of their system and design the system accordingly. For example, a BLE application designer who is not getting the required throughput with a specific number of characteristics or characteristic size can increase the number of characteristics or characteristic size to get a suitable throughput. Another example is a designer who desires to have specific battery lives for the sensors used in a system can calculate the battery lives for the sensors considering different scenarios of design options and choose the design that suits his needs optimally.

Future work includes studying the possibility of throughput improvement by reducing the processing time required for switching between frequencies or using compression techniques and the effect of these techniques on the power consumption of the system.

REFERENCES

[1] "Bluetooth Low Energy",Stollman, Tech. Rep, 2012.

[2] I. D. C. Gomez, and J. Paradells. "Modeling the maximum throughput of Bluetooth low energy in an error-prone link". *IEEE Commun Lett*, vol. 15, NO. 11. November, 2011.

[3] *Specification of the Bluetooth System, Covered Core Package 4th Edition*, ed, June 2010.

[4] *TI CC254x Bluetooth Low Energy Software Developer's Guide*, T. Instruments Inc.,2012.

[5] R. Heydon, *Bluetooth Low Energy, The Developer's Handbook*: Prentice Hall, October 2012.

[6] M. L. Elke Mackensen, and Thomas M. Wendt, "Performance Analysis of an Bluetooth Low Energy Sensor System," presented at *The 1st IEEE International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems,* Offenburg, Germany, 2012.

[7] *TI BLE Vendor Specific HCI reference guide*, T. Instruments Inc., 2011.

[8] S. K. a. J. Lindh, "Application Note 092. Measuring Bluetooth Low Energy Power Consumption, " Texas Instruments, 2012.

APPENDIX A

SOURCE CODE

Code used in both the master and slave devices is based on Texas Instruments

SimpleBLEPeripheral and SimpleBLECentral applications.

CODE FOR THE DEFINITION OF THE CHARACTERISTICS IN THE PROFILE

Definition of the Characteristics UUIDs

```
#define SIMPLEPROFILE_CHAR1_UUID          0xFFF1
#define SIMPLEPROFILE_CHAR2_UUID          0xFFF2
#define SIMPLEPROFILE_CHAR3_UUID          0xFFF3
#define SIMPLEPROFILE_CHAR4_UUID          0xFFF4
#define SIMPLEPROFILE_CHAR5_UUID          0xFFF5
#define SIMPLEPROFILE_CHAR6_UUID          0xFFF6
#define SIMPLEPROFILE_CHAR7_UUID          0xFFF7
#define SIMPLEPROFILE_CHAR8_UUID          0xFFF8
#define SIMPLEPROFILE_CHAR9_UUID          0xFFF9

// Length of Characteristic in bytes
#define SIMPLEPROFILE_CHAR_LEN     19
```

Definition of the Attributes in the Attribute Database

```
/*********************************************************************
 * Profile Attributes - variables
 */

// Simple Profile Service attribute
static CONST gattAttrType_t simpleProfileService = { ATT_BT_UUID_SIZE,
simpleProfileServUUID };


// Simple Profile Characteristic 1 Properties
static uint8 simpleProfileChar1Props = GATT_PROP_READ | GATT_PROP_WRITE;

// Characteristic 1 Value
static uint8 simpleProfileChar1[SIMPLEPROFILE_CHAR_LEN] =
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};

//static uint8 simpleProfileChar1=0;

// Simple Profile Characteristic 1 User Description
static uint8 simpleProfileChar1UserDesp[17] = "Characteristic 1\0";


// Simple Profile Characteristic 2 Properties
static uint8 simpleProfileChar2Props = GATT_PROP_READ | GATT_PROP_WRITE;

// Characteristic  2 Value
```

```
static uint8 simpleProfileChar2[SIMPLEPROFILE_CHAR_LEN] =
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};

//static uint8 simpleProfileChar2=0;

// Simple Profile Characteristic 2 User Description
static uint8 simpleProfileChar2UserDesp[17] = "Characteristic 2\0";


// Simple Profile Characteristic  3 Properties
static uint8 simpleProfileChar3Props = GATT_PROP_READ | GATT_PROP_WRITE;

// Characteristic 3 Value
static uint8 simpleProfileChar3[SIMPLEPROFILE_CHAR_LEN] =
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};

//static uint8 simpleProfileChar3=0;

// Simple Profile Characteristic 3 User Description
static uint8 simpleProfileChar3UserDesp[17] = "Characteristic 3\0";


// Simple Profile Characteristic 4 Properties
static uint8 simpleProfileChar4Props = GATT_PROP_NOTIFY;

// Characteristic 4 Value
static uint8 simpleProfileChar4[SIMPLEPROFILE_CHAR_LEN] =
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};

// Simple Profile Characteristic 4 Configuration Each client has its own
// instantiation of the Client Characteristic Configuration. Reads of the
// Client Characteristic Configuration only shows the configuration for
// that client and writes only affect the configuration of that client.
static gattCharCfg_t simpleProfileChar4Config[GATT_MAX_NUM_CONN];

// Simple Profile Characteristic 4 User Description
static uint8 simpleProfileChar4UserDesp[17] = "Characteristic 4\0";


// Simple Profile Characteristic 5 Properties
static uint8 simpleProfileChar5Props = GATT_PROP_READ | GATT_PROP_WRITE;

// Characteristic 5 Value
static uint8 simpleProfileChar5[SIMPLEPROFILE_CHAR_LEN] =
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};

// Simple Profile Characteristic 5 User Description
static uint8 simpleProfileChar5UserDesp[17] = "Characteristic 5\0";

// Simple Profile Characteristic 6 Properties
```

```
static uint8 simpleProfileChar6Props = GATT_PROP_READ | GATT_PROP_WRITE;

// Characteristic 6 Value
static uint8 simpleProfileChar6=0;
// Simple Profile Characteristic 6 User Description
static uint8 simpleProfileChar6UserDesp[17] = "Characteristic 6\0";

// Simple Profile Characteristic 7 Properties
static uint8 simpleProfileChar7Props = GATT_PROP_READ | GATT_PROP_WRITE;

// Characteristic 7 Value
static uint8 simpleProfileChar7=0;

// Simple Profile Characteristic 7 User Description
static uint8 simpleProfileChar7UserDesp[17] = "Characteristic 7\0";

// Simple Profile Characteristic 8 Properties
static uint8 simpleProfileChar8Props = GATT_PROP_READ | GATT_PROP_WRITE;

// Characteristic 8 Value
static uint8 simpleProfileChar8=0;

// Simple Profile Characteristic 8 User Description
static uint8 simpleProfileChar8UserDesp[17] = "Characteristic 8\0";

// Simple Profile Characteristic 9 Properties
static uint8 simpleProfileChar9Props = GATT_PROP_READ | GATT_PROP_WRITE;

// Characteristic 9 Value
static uint8 simpleProfileChar9=0;

// Simple Profile Characteristic 9 User Description
static uint8 simpleProfileChar9UserDesp[17] = "Characteristic 9\0";
```

CODE FOR WRITING CHARACTERISTICS FROM THE MASTER DEVICE

```
//Using the function WriteToCharacteristic(I,j) to write to characteristic number I values start
//with j
WriteToCharacteristic(1,2);
WriteToCharacteristic(2,3);
WriteToCharacteristic(3,4);
WriteToCharacteristic(5,6);

// Definition of Function WriteToCharacteristic
```

```
static void WriteToCharacteristic( int charnumber, int charvalue)
{
    attWriteReq_t reqw;
    uint8 i;
    uint8 j= charvalue;
    uint8 status;
    switch (charnumber)
    {
    case 1:
      reqw.handle=0x0025;
      for (i=0;i<15;i++)
      {
        reqw.value[i]=i+j;
      }
      reqw.len= 15;
      break;
    case 2:
      reqw.handle= 0x0028;
      for (i=0;i<15;i++)
      {
        reqw.value[i]=i+j;
      }
      reqw.len= 15;
      break;
    case 3:
      reqw.handle=0x002B;
      for (i=0;i<15;i++)
      {
        reqw.value[i]=i+j;
      }
      reqw.len= 15;
      break;
    /*case 4:
      reqw.handle=0x002E;
      break;*/
    case 5:
      reqw.handle=0x0032;
      for (i=0;i<15;i++)
      {
        reqw.value[i]=i+j;
      }
      reqw.len= 15;
      break;

    case 6:
      reqw.handle= 0x0035;
      reqw.value[0]=6;
      reqw.len= 1;
      break;
```

```
    case 7:
      reqw.handle= 0x0038;
      reqw.value[0]=7;
      reqw.len= 1;
      break;

    case 8:
      reqw.handle= 0x003B;
      reqw.value[0]=8;
      reqw.len= 1;
      break;

    case 9:
      reqw.handle= 0x003E;
      reqw.value[0]=9;
      reqw.len= 1;
      break;

    }

    reqw.sig=0;
  reqw.cmd=1;
  status= GATT_WriteNoRsp( simpleBLEConnHandle, &reqw);
}
```

VITA

Hafiz Ahmed was born in Khartoum, Sudan to the parents of Eltayeb Ahmed and Samia Mahmoud. Hafiz has three brothers and two sisters. Hafiz attended school in Khartoum, Sudan through high school which he graduated from as one of the top ten students in Sudanese secondary school examinations around the country and received a full scholarship from the University of Khartoum and the Sudanese national office for students' support for his Bachelor's degree. During his undergraduate study Hafiz was involved in many students' organizations such as the executive committee and the academic council of Electrical and Electronic Engineering Students' Society. Hafiz graduated with a first class Bachelor of Science in Electrical Engineering degree from the University of Khartoum in 2008. After graduation, Hafiz worked as a network management engineer and software service engineer for two years. He was offered a graduate research assistantship in the Electrical Engineering department at the University of Tennessee at Chattanooga and continued to pursue Master of Science in Electrical Engineering.