Imperial College London

Department of Electrical and Electronic Engineering

Final Year Project Report 2014

Project Title:	**A Wireless Low Energy Ambulatory Electroencephalogram**
Student:	**Thomas Alexander Morrison**
CID:	**00642176**
Course:	**ISE4**
Project Supervisor:	**James Mardell**
Second Marker:	**Dr. Pants Georgiou**

Abstract

Certain neurological medical disorders require continuous monitoring to fully understand and diagnose. Examples of medical interest include epilepsy, syncope, multiple sclerosis, migraines, strokes, Parkinson's and Alzheimer's disease. Electroencephalography (EEG) is the recording of electrical activity along the scale, resulting from ionic current flows within the neurons of the brain and is useful for both diagnostic and monitoring such aforementioned conditions. Monitoring brain activity can help physicians understand certain characteristics, triggers, and the severity of the disorder. It may be possible to gauge the regions of the brain where the condition is originating and if the patient is a suitable candidate for treatment.

However, symptoms from neurological disorders often appear sporadically and with little to no warning. Seizures vary from minutes to years apart, and sometimes are not realised or detected without proper equipment. Hospitalising patients for long periods of time is a costly option, and in such circumstances the patient could remain in hospital indefinitely. Such circumstances lend themselves to an ambulatory system, where an outpatient can be monitored continuously without discomfort or hospitalisation, improving quality of life while decreasing costs.

With the emergence of low power wireless technologies coupled with portable devices such as tablets and phones, it is a natural technological step to bring care and monitoring out of the hospital and into the home. Through leveraging low energy radio capable platforms, i.e. smartphones and tablets, in the context of an ambulatory EEG, it is possible to empower the patient to inexpensively take health care into their own home and out of the hospital. This project looks at maximising transmission of EEG signals over the emerging wireless technology, Bluetooth Low Energy (BLE). Further, while this project is targeted at EEG signals, there is no reason why this research and technology cannot be applied to other signals and systems, examples including glucose monitoring, electrocardiography and spirometers.

[3]

Contents

1 Acknowledgements

The opporunity is taken here to express gratitude to all those who have directly contributed towards the project.

Firstly, the project surperivsors, James Mardell, Chen Guangwei and Esther Rodriguez-Villegas from the Circuit and Systems deparmental group, for the opportunity to undertake this piece of work.

Cambridge Silicon Radio (CSR) provided hardware and technical support in the spirit of academia. Particular acknowledgements go to employees Adam Hill, Martin Spikings, Mark Wade, Neil Stewart and Simon. CSR have requested that this project report or relevant parts of it be made available to them.

Guan Yang and Jacob Rosenthal from New York, United States, for pubically making available code to drive nRF8001 radio. Imperial College student Stelius Ioakim for sharing experience of printed circuit board (PCB) assembly. My girlfriend for dainty and dexterious hands, invaluable during PCB assembly.

Dr. Nissim Zur, CEO of Vitelix Limited is an expert in low power wireless technologies, and has conversed over many aspects of the CSR1010 chip used. Further, he has also taken an interest in this project's work in regards to maximising the speed, and has requested the results be shared with him.

Finally, special thanks go towards Mike Harbour and Victor Boddy for their efforts in printed circuit board manufacture and assembly. Countless hours were spent in the lab pushing the department's PCB fabrication facilities outside specifiction and assembling the boards.

2 Introduction

Patients, however, are unlikely to suffer from neurological disorder while at the clinic as they often appear sporadically during day to day life and with little to no warning. In such circumstances the patient could remain in hospital indefinitely, however seizures can be minutes to years apart, and sometimes are not realised or detected without proper equipment. Such circumstances lend themselves to an ambulatory system, where the patient can be monitored continuously without discomfort or hospitalisation, thus improving quality of life.

2.1 Problem Landscape and Motivation

Neurological disorders and their sequelae are currently estimated to affect upto one seventh of the world's population, a figure which currently stands at 1 billion. With the ever increasing life expectency and decreasing (relative) fertility rates, the population age demographic has shifted towards an ageing population. As neurological disorders are much more present in aging people, the rates for neurological disorders has increased and is expected to increase further. Diagnosing, monitoring and treating many of these diseases is currently a costly procedure due to the time and resource allocation (i.e. relatively expensive equipment).

Unfortunately many neurlogical disorder symptons occur sporadically with little to no inidication of an event, such as in the cause of a seizure for epilepsy.

The vast majority of modern smart phones are now being equiped with BLE or "Bluetooth Smart" technology. This is a relatively new technology design branded off the popular Bluetooth technology which gained popularity in the early part of last decade. While the two technologies share the same name, their design and operation is very much different. Although origianl Bluetooth's and BLE's use cases may overlap in some scenarios, they were designed to perform well under different circumstances.

By coupling cheap low power sensors and radios, with powerful ubiquitous consumer technology, it is possible not only to cheaply and efficiently monitor patients, improve the quality of life for patients but also help physicians further their understanding of neurological disorders.

Currently BLE is the only radio technology that is currently being built into all smart phones and tablet devices while offering power consumption low enough to enable a long lifetimes from a lightweight power source. In comparison classic Bluetooth's power consumption is typically 1 to 2 orders of magnitude higher than BLE. Through leveraging widely popular and familiar smart phone devices with this new technology, in the context of an ambulatory EEG, it is possible to empower the patient to inexpensively take health care into their own home and out of the hospital.

2.2 Existing Technologies and Products

With the plethora of emerging low power wireless technologies coupled with portable devices such as tablets and phones, it is a natural technological step to bring care and monitoring out of the

hospital and into the home. The consumer fitness sector is being targeted quite strongly, and many devices already exist that utilise lower power technologies to act as gateways for real-time data logging. For example, there already exists a competitive market between heartbeat monitors, cadence monitors and pedometers. These 'activity trackers' use low power electronics and radios to log user's activities and update the user in real time with activity information through the user's phone or smart watch. Popular products on the market at the time of writing include the Fitbit, Fuelband and Jawbone, which all use the BLE technology to connect to smartphones.

2.3 Project End Goals

Originally the project was introduced as "Maximising Bluetooth Low Energy throughput of EEG signals", however

Currently, the Imperial College Circuit and System's group has a wired EEG measurement device. The wired connection between the EEG sensors and a computer cause patients to remain fairly immobile and hence are impracticable for long periods of use. This project will explore using BLE technology for electrocengraphy, and build a prototype system capabale of interfacing with an analouge front to transmit EEG data to a portable device such as a tablet or smart phone. The project will explore the maximum throughput of such a device along with its power consumption.

Hard requirements of the project include

- Running time of atleast 12 hours
- Channel resolution of 8 bits (albeit number of channels undefined)
- A weight of less than 7 grams
- 10 meter range
- BLE wireless technology
- Ability to communicate with a smart phone or tablet
- Real time update of signals

2.4 Structure

3 Theory and Technology

3.1 Bluetooth Low Energy

The original Bluetooth, hereforth refered to a Bluetooth Classic (BTC), was intially concieved as the solution to wired communication over short distances (typically less than 100m). The original specification had an air over-the-air rate of 1MBps, though this has increased to around 3 MBps in the latest version of BTC. Similarly, BLE has an over-the-air rate of 1MBps. Despite the odd realisation that BLE, a much newer technology, has the same over the air rate of last the first incarnation of BTC, the maximum theoretical throughput of BTC is 700kBps, compared to less than 250kBps for BLE - roughly one third of the maximum throughput BTC was capable of (the latest version of BTC brings the disparity to one ninth). While intuatively it may seem that BLE is a less efficient technology, BLE can be orders of magnitudes more efficient than BTC in particular use cases.

Applications where BLE excels in are ones where communication between two devices is only required intermitently, and the volume of information sent is small. An example would be a thermometer in a greenhouse connected by radio to a visual display unit inside the home. Temperature changes at a rate slow enough that it is only necessary to check the temperature every 10 minutes. Once every 10 minutes the radio thermometer device can wake up take a measurement, send a notification of a measurement then return to a deep sleep. BLE does this much better than BTC, taking only a few milliseconds to connect. BTC takes between a few hundred milliseconds to several seconds to reconnect. While both millisecond orders of magnitude and second orders of magnitude are small when compared to an order of magnitude of minutes, over time it adds up to a significant amount, and BLE devices can last many years of a small, single coin cell. BLE is excellent for applications which involve small episodic transmission of data. In the scenario described a BTC system would have a lifetime of approximately 100 days from a typical 3v lithium cell. Off the same cell, a BLE system would have a lifetime of many years. In fact, in this scenario the BLE system lifetime can be extended further as BLE can support connectionless communication, whereby it simply wakes up and transmits the thermometer state to any device that's listening without the need for acknowledgement.

The reason the reconnection times are much faster for BLE is as far as the communication devices are concerned, they never disconnected. Rather in the BLE protocol, the devices agree to meet a specified periods known as connection interval (CI). The devices are free to perform any operations in the mean time, though typically enter a state of hibernation. In many scenarios between two radios, one device will be much more power conscious. In the example above, the battery powered device in the greenhouse would typically be the power conscious device while the visual disuplay unit inside the home will likely be powered from the grid, and have no concern as to its power consupmption. In these situations, the power conscious device is defined to be the slave and the remaining device to be the master.

The slave device also has the ability to skip connection intervals. That is, the slave to skip a upto a predetermined number of connection intervals, known as the slave latency. While the master must always check back to see if the slave has sent anything, the slave, if it has nothing to send, doesn't need to wake up. For example, if the slave latency is 120, and the connection interval is 1000ms, then the slave is not obliged to communicate with the master for upto 2 minutes, despite the master having

to check every 1000ms. If the slave is not able to make contact with the master after 2 minutes, then the master will begin counting the number of times the slave has missed the obliged connection period (that is the CI multiplied by the time slave latency). If this value reaches above a certain threshold (a typically recommened value of 6), the master will consider the slave disconnected, and be required to go through a connection process again[UNLESS USING BROADCASTING TO SEND DATA]. Continuing with the scenario, the slave will be considered disconnected if it hasn't made contact with the master after 12 minutes.

Another contribution to reduced power over BTC is the reduction in the number of channels used in communicaton. BTC, BLE along with other wireless technologies that operate in the 2.4GHz ISM band such as a WiFi and ZigBee all make use of spread spectrum techniques to achieve a sufficient level of noise immunity. That is, the available bandwidth is split (spread) into smaller channels and radios communicate between one another using a pre-dertmined channel hop sequence. As the number of channels decreases, the channel band size grows requiring less accurate and complex modulation hardware, hence decreasing power consumption. BTC originally used 79 channels, and BLE reduces this to 39.

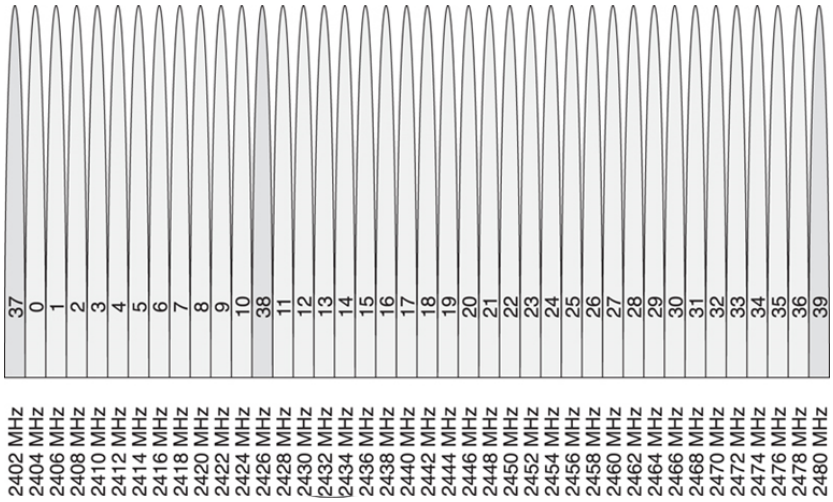


Figure 1: BLE channels (advertisement channels render darker)

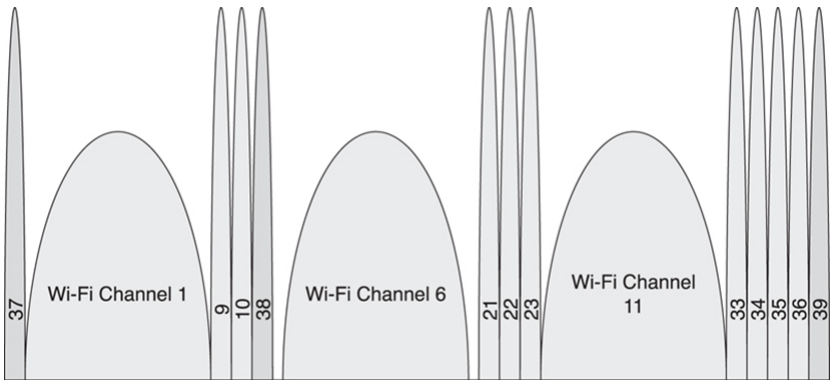


Figure 2: BLE channels with WiFi channels overlaid

The number of channels dedicated to advertising has also decreased, meaing less time is spent searching for discoverable devices. The advertisement channels have been specifically chosen not

to interfere with the common WiFi channels. Finally, the radio characteristics are very dependent on temperature. Complex mechanisms are used to compensate and recalibrate on-the-fly radio parameters. Due to the episodic nature of BLE the radio does not come under such thermal extremes. All these design changes have positive hardware ramifications. The reduced compelxity of BLE means reduced hardware requirements (notably memory), in turn reducing the leakage current.

BTC was designed with the idea that it would be used to do many common jobs, and hence particular configurations were built into it. In BTC, these configurations are known as profiles. Example profiles include the audio distribution profile (A2DP), which is used in by many Bluetooth product manufacturers to allow a device, such as a phone to interact with an audio system, such as in a car. Another example would be the serial port profile (SPP), meant to emulate the highly popular and robust RS-232 serial standard for data transfer (recall that BTC was concieved as a solution to wires). This is all built into what is known as the Bluetooth stack – a software framework that interacts between the physical layer and the application layer[1].

BLE also makes use of this paradigm but is often superficially depicted as BTC operating at lower speeds and power consumption. It is not currently compatible with BTC and there are no plans for it to be. Like BTC, the BLE architecture has 3 over-arching parts: Application, Host and Controller. The controller, simply put, is the radio and related hardware controllers and the application the use case, which could be a cadence monitor, thermometer or even an electroencephalogram. It is the host controller interface (HCI), commonly known as the "stack" that provides the necessary software to enable the application layer to communicate with the radio (see Figure 3).

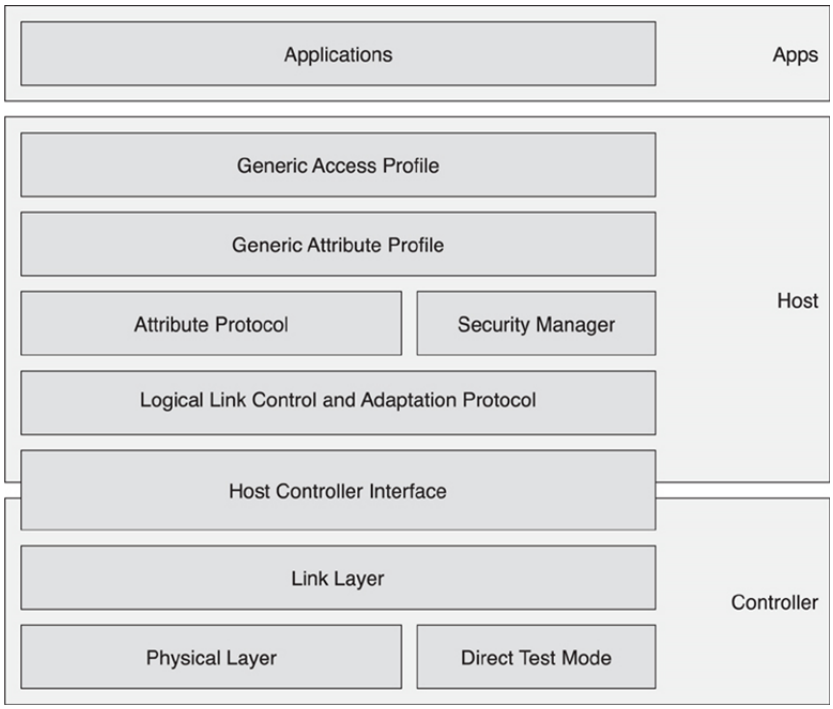


Figure 3: BLE Architecture

In an attempt to be economical with time and space components of the stack deemed irrelevant

[1]Depending on what level of the stack one is working, master can take the names central or client, while slave can take peripheral or server.

will not be discussed further here. For example a description of the security manager is not relevant as this project is not concerned with sending data over encrypted links. Similarly, the Logical Link Control and Adaption Protocol, while used extenivesly in all radio communication will not be discussed in detail as it doesn't provide any insight into maximising throughput or minimising power.

In BTC profiles were diverse and large enough to warrant chip designers releasing tailored chips to perform well for a specific profile, i.e. a chip supporting A2DP may contain coder/decoder (CODEC) hardware for real time audio streaming. In BLE the profile framework is far lighter. BLE's profiles are all built ontop of the Generic Attribute Profile (GATT), which in turn is built upon the Attribute Protocol (ATT), a protocol optimised to run on BLE devices. Attributes are an umbrella term, being the atomic unit of data communicated between BLE devices. Profiles are hierarchical constructions of attributes, in the top down order of profile, service, characteristic and descriptors, as shown in Figure 4. A BLE device implements atleast one profile, Generic Access Profile (GAP) along one more which is typically the purpose of the device. GAP contains important information such the the device name and prefered connection properties. This second profile can be one of the standard profiles as defined by the SIG group or a bespoke profile for the application, such as the case for a EEG. Popular, SIG defined examples of BLE profiles include the heart rate profile (HRP), health thermometer profile (HTP) and even a glucose profile (GLP) with room for many more to be incorporated into the core BLE Special Interests Group (SIG) defined GATT specifications.

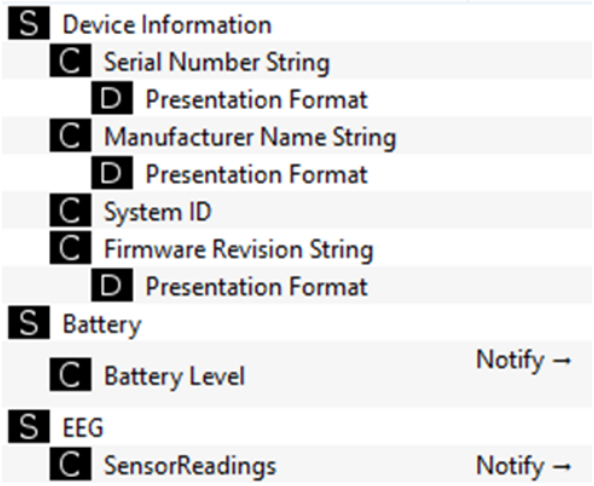


Figure 4: Example GATT profile consisting of 12 attribute - 3 services, 6 characteristics and 3 descriptors.

Attributes represent information about state. In the case of the greenhouse, the state would include the measured temperature. A suitable profile name which encapsulates the state is "thermometer", which contains the services "device information", "battery service", and "thermometer". As in figure 4, the greenhouse thermometer service may contain the same device information and battery services, but replace the "EEG" service with the thermometer service, and the "SensorReadings" service with temperature. The other profile, GAP, will contain the attributes which define how BLE unit discover and establish connection to one another. This includes the device name, perhaps "greenhouse thermometer", along with the preferred slave connection properties, e.g. a connection interval of 4 seconds, with a slave latency of 150 (10 minute window). All this information is contained within the GATT database, and shared as needed to other BLE units.

When communicating attributes, there are four data operations available. Read and write typically require one device to access the others characteristic. In the case of the greenhouse, the house device would request to read the thermometer. Notification and indications differ to read and write, in that the device(s) after information subscribes to the changes of state of the characteristics. For example, when greenhouse slave device wakes up, if the thermometer measurement changed, it will send a notification or indication alert the master device inside the house. The former methods can be thought of as synchronous means of communication while the latter asynchronous. Notifications differ from indications in that indications require a application level acknowledgment. That is, the indication is bubbled up to the user code, which then either accepts or rejects the indications. Notifications are acknowledge near the bottom of the BLE stack, verifying correct receipt and message integrity. Therefore notifications are suitable for higher throughput applications. As shown in figure 4 shows notifications are operation configured for battery and EEG sensor readings. In this example, whenever the battery level changes, any devices subscribed to the characteristic battery level will receive a notification.

In BTC the network topology used pico-nets, whereby one device has one master, but could be a master of another device. BLE operates a simpler network topology, star, whereby a device can either be a master or slave, but not both. The master has the responsibility of organising itself between the slaves. If one slave requires large amounts of bandwidth, it my impact the quality of service encountered by the other devices.

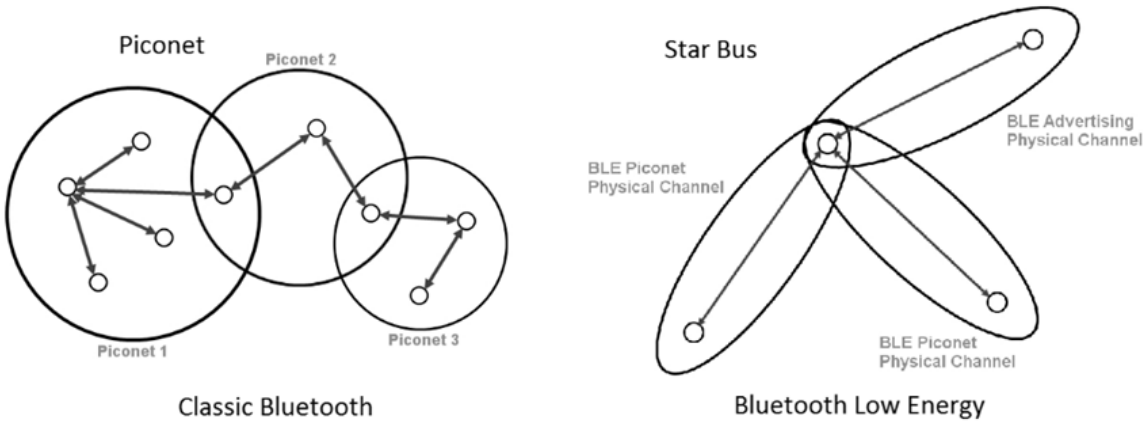


Figure 5: Bluetooth Network Topologies

Both BLE and BTC devices move through generic system states. The same abstract view can be applied to both technologies and is shown in. Note that depending on the role of the device, the state moves right (master) or left (slave) from standby. It may appearing confusing to have another state for scanning which can only move into the standby state, but this is useful for searching and discovering devices with no commitment to connecting. Such a use case might be suitable for devices that intermittently broadcast small amounts of information. Assuming that the master BLE device is in the initiating state, searching for a connectable device, and at the same time the BLE slave is in the advertising stage, periodically broadcasting information using advertisement packets; the two devices will find one another and may initiate a connection request.

BLE is principally composed of two types of packets, advertisement (Figure 7) and data

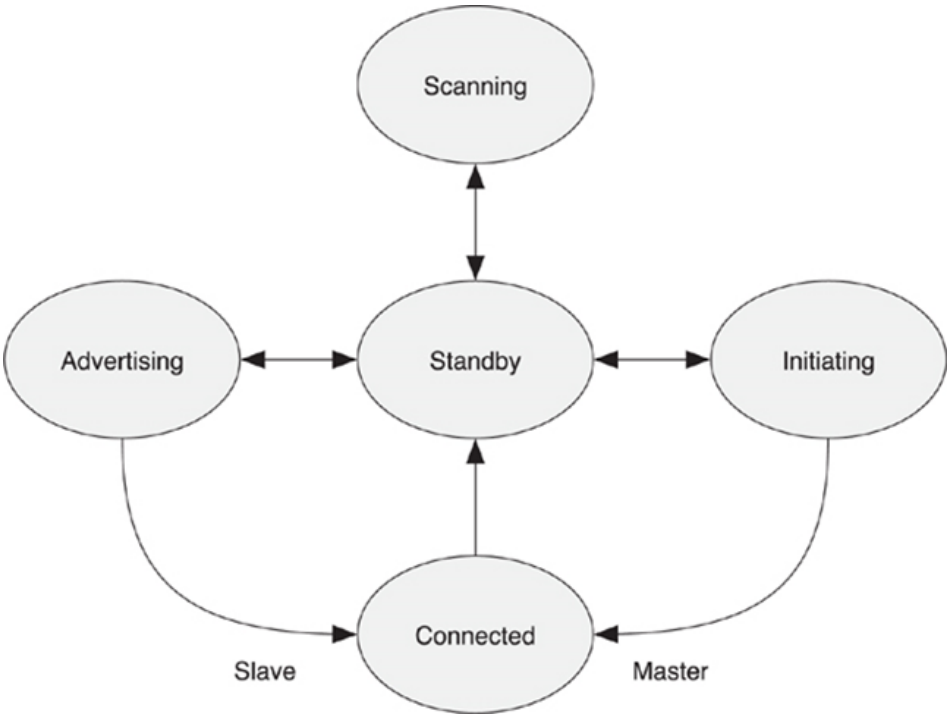


Figure 6: Device State Transition Diagram

(Figure 8). Both packet types vary in length dependent on the payload but share a 32 bit advertising access address field, a 24 bit Cyclic Redundancy Check (CRC) field, an 8 bit header and an 8 bit length field which defines the Packet Data Unit (PDU) size (the last 2 field are often collectively called the header field). Advertisement PDUs range from 0 to 29 bytes (232bits), meaning the total packet size can vary between 72 and 320 bits. The over-the-air data rate is 1Mbit/s, meaning advertisement packets are transmitted at a rate between 72 and 320μs (a single bit is transmitted every 1μs). In addition to the access code and CRC fields, data packets consists of an 8 bit preamble and a PDU varying between 0 and 27 bytes (4 of these bytes are reserved for encryption). Hence, the packet length varies from 80bits to 296 bits (328 including encryption).

Figure 9 shows a connection between two devices being initiated. The first packet shown is is an indirect advertisement packet, available to all listening BLE devices. The second packet is a connection request from the master device, communicating in the payload its address, the address to establish a connection with, and random access address, a connection interval length, the hop sequence, channel map, the connection timetout, slave latency and other things. Here the advertisement packets are rendered in green, the data in (predominantly) yellow (the magenta also represent a type of data packet)

- The channel map bit pattern corresponds to a contiguous stream of 37 ones, corresponding to all 37 channels being operational (the master has no reason to prevent transmission).
- The hop byte indicates between each CI how many channels the device pair will increment. From packet 242 onwards, the channel map increments every CI (2 packets/30ms).
- The access address is a randomly chosen number (by the master) which acts as a identifier for a connection between two devices.

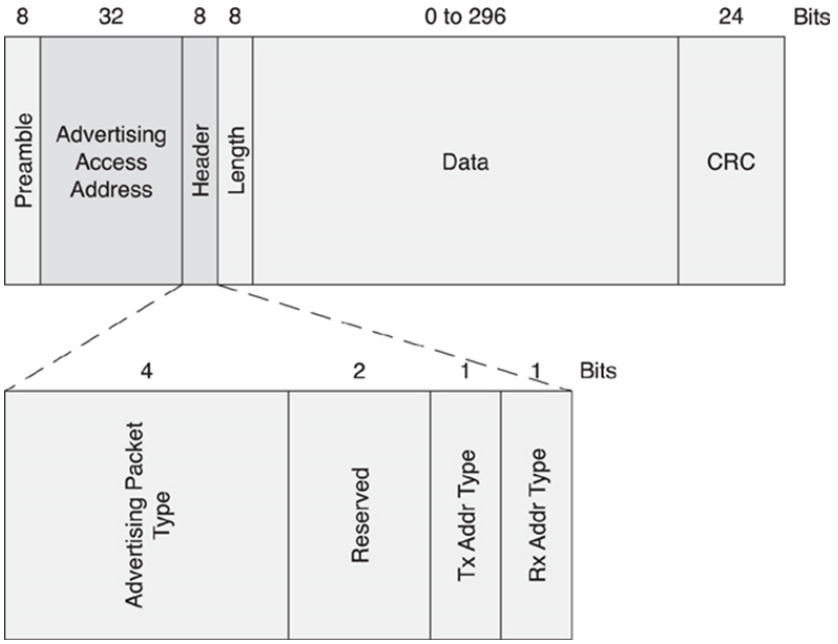


Figure 7: BLE Advertisement packet

Img advertisment

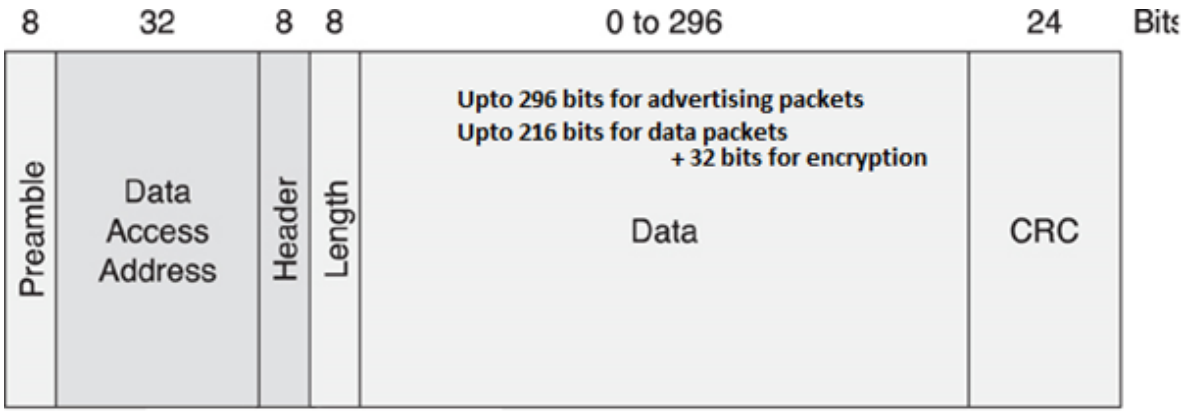


Figure 8: BLE Data packet

- The interval of 0x18 (24 decimal) represents the CI length in units of 1.25ms. Therefore 0x18 represents 30ms between connection intervals. Between each master to slave directional data packet (every 2 packets), the total sum of the time is 30000μs

It is highlighted that (at least initally) the slave interval is 0, meaning the slave should wake up every CI. In packets numbers 243 and 244, the slave and master respectively have nothing to send, and simply acknowledge on another with an empty PDU. The protocol realises flow control through the the lazy acknowledgment of Sequenc Number (SN) and Next Expected Sequence Number (NESN) bits, embedded into the header of every data channel.

While explained for completeness, advertisement and packets used in the initial establishment of a connection are of no concern to this project since they do not contribute to data throughput and once a connection has been established, do not contribute to the power consumption of the device.

Figure 9: BLE connection initialisation

Hence, their contribution is removed from measurements as it is assumed the long term amortised cost

is zero. The (incomplete) connection initialisation process shown in Figure 9 is known as "Just Works" pairing.

As previously mentioned, to achieve high throughputs notifications are used. To achieve the maximum throughput, it is desirable to send notifications as fast as possible. However, data cannot be sent out without flow control, and all notification packets must first be received then acknowledged before the next packet is sent. When a packet is sent, there is a mandatory 150μs inter-frame period. For the maximum throughput the connection receiving device would just acknowledge the data in an 80 bit acknowledgment response. Another 150μs frame would occur between this responding device and a transmitting device, bringing the total time up to:

$$296\mu s + 150\mu s + 80\mu s + 150\mu s = 676\mu s$$

This is shown graphically in Figure 10 (encryption is enabled). The maximum duty cycle is obtained when encryption is used

$$\frac{328\mu s + 80\mu s}{708\mu s} \approx 55.6\%$$

which is relatively low when compared radio technology duty cycles, e.g. BTC. The 150μs inter frame period exists to prevents the silicon from heating to excessively, preventing power consuming hardware being required to recalibrate the radio.

From the figure it is possible to calculate the upper bound for the number of packets that can be transmitted per second is

$$\frac{1000000\mu s}{676\mu s} \approx 1479.29\ packetes/second$$

Of the 37 bytes (296 bits) sent for a notification packet, 27 of them are known as the data payload. Even if encryption is not used, 4 bytes of the 27 are required for the L2CAP header use. The remaining 23 bytes are quotes as the available application bytes.

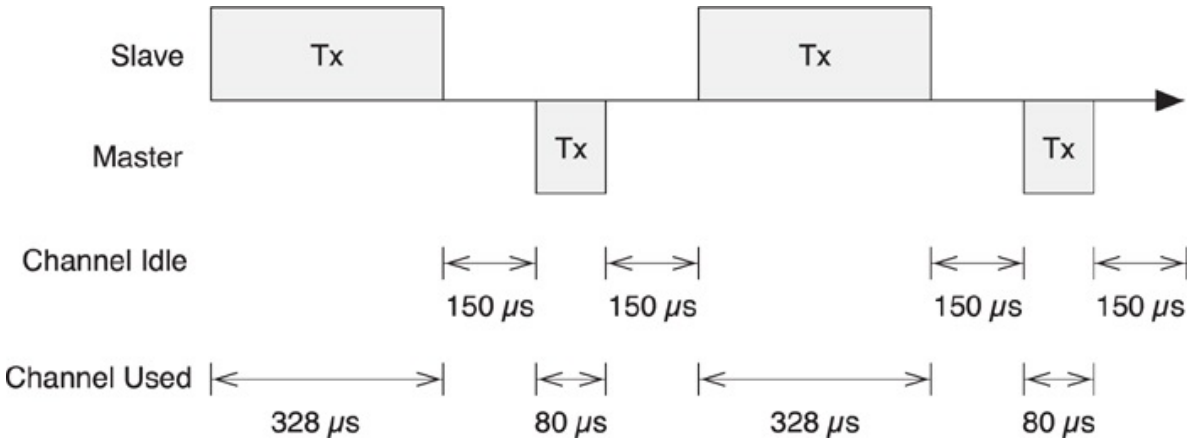
$$1479.29\frac{packets}{s} \times 23 \times 8bits \approx 270kbp/s$$

While a lot of literate quoteCITATION PLEASE] this as the maximum usable data rate of this 23 bytes 3 bytes are required to identify the command type (notification) through an op-code (1 bytes) and which attribute it belongs to through an attribute handle (2 bytes). This means that only 20 bytes of the original 27 reserved for the application are usable, reducing the upper bound for throughput to

$$1479.29\frac{packets}{s} \times 20 \times 8bits \approx 236.7kbp/s$$

Here forth, the usable payload of a "notification" or "data packet" is 20 bytes. When aiming for either low power and/or high efficiency, it is important fill the usable payload with as much data as

possible, as every notification packet sent has a fixed cost of 80 bits associated with it. TALK ABOUT PCK EFFICIENCY



centering

Figure 10: BLE's packet flow for maximum throughput

The BLE specification defines no maximum

The BLE protocol was not designed for high throughput applications, but rather for low latency episodic communication devices transferring small data payloads of representing device state. The energy per bit is low compared other popular technologies, and should not be used as the bottom-line metric for performance. Rather, a holistic metric of the whole system energy consumption for a specific use case will form the basis of the performance metric in this report.

4 Preliminary Research

This section deals with evaluating the technology and radios Equipment used to evaluate

Wireshark etc

4.1 Radio Evaluation

There are a number of competing companies int the BLE consume space. The most popular chip manufacturers at the time of evaluation areTexas Instruments (TI), Nordic Semiconductor (NS) and CSR, although other manufacturers exist, they normally design combo-wireless technology chips, e.g. Broadcomm, which only offers WiFi, BTC and BLE combined system on chip (SoC). These chips are not suitable for this project due to their complexity, packaging and intend application - the typical use case of such combo-chips are in high powered devices such as tablets, smart phones and notebooks.

TI provided an inexpensive BLE packet sniffer which was used extensively throughout this project.

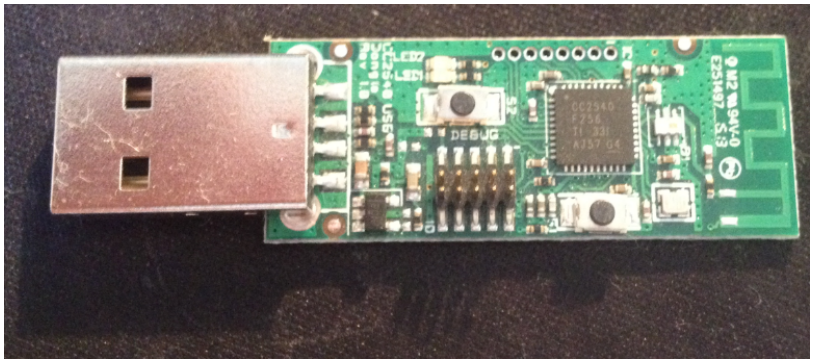


Figure 11: TI's BLE packet sniffer

Different sources [citation] will calculate the application throughput differently.

BLE data packets can carry a payload of upto 20 bytes

After sourcing

4.1.1 nRF8001

The first radio tested was NS's nRF8001, selected due to the data sheet claiming it was the lowest power consuming device. The reason for this was because the chip only contained a controller and host layer. The application layer must be provided by an external micro controller, which is a large amount of effort to write. Fortunately a hobbyist open source project[6] was available that provided the necessary code to get limited connectivity up and running. Figure **??** shows the prototyping setup used to measure the performance of the nrf8001 microchip.

Using a shunt resistor, the measured peak current was approximately 14.5mA, which correlates with the figure found in the data sheet (14.6mA). This was at a

The ATMega328 consumes approximately XmA, however for the remainder. As the device was hardware limited to only 1 packet per connection event further development was abandoned.

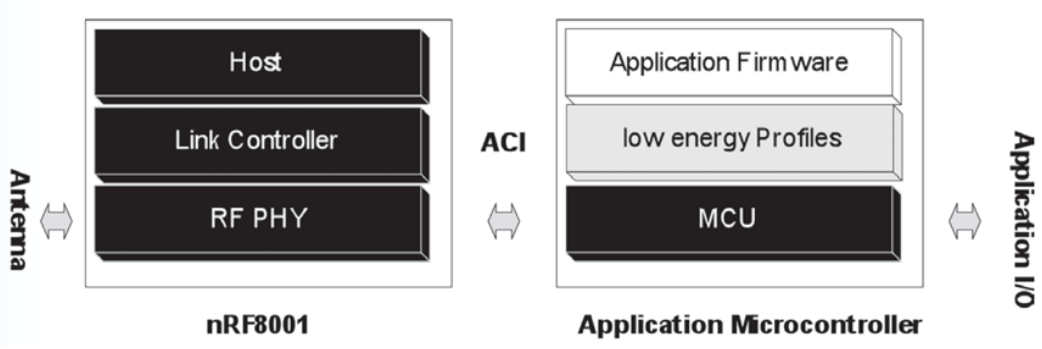


Figure 12: nRF8001 application block diagram

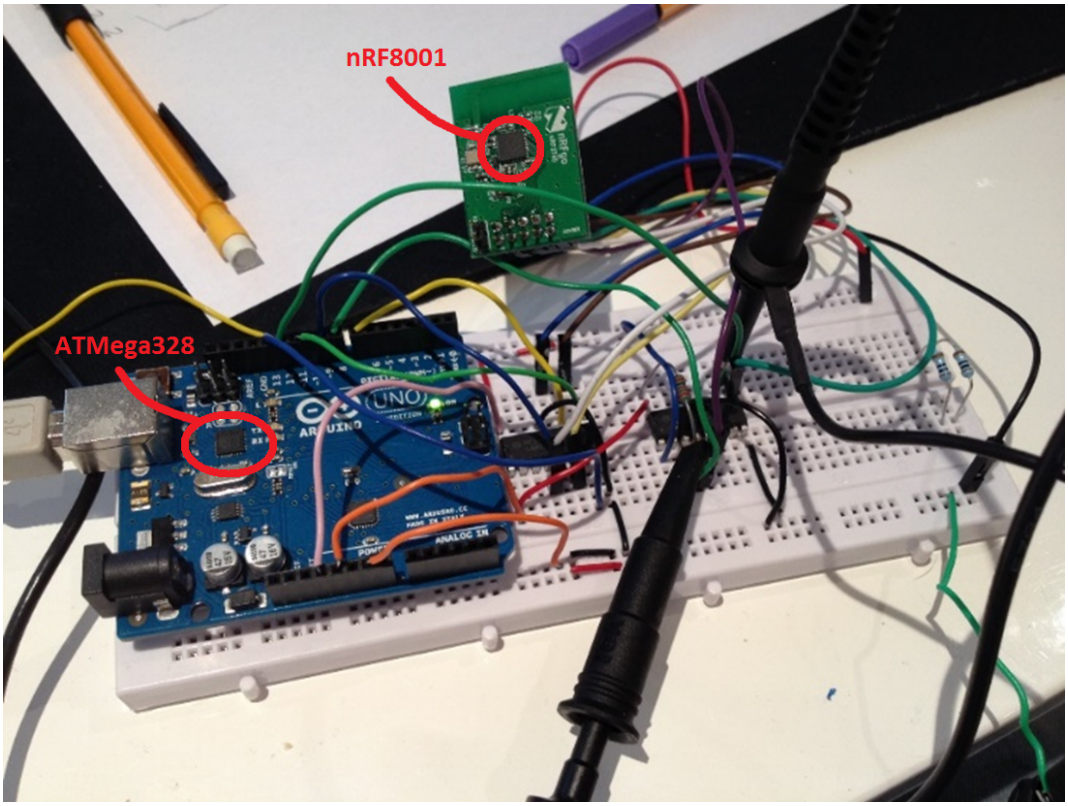


Figure 13: Arduino Uno development board acting hosting the application layer for the nRF8001 radio. Level shifters were required for interfacing with the low-paper chip

P.nbr.	Time (us)	Direction	Data Type	Data Header					L2CAP Header		ATT_Handle_Value_Notify		
				LLID	NESN	SN	MD	PDU-Length	L2CAP-Length	ChanId	Opcode	AttHandle	AttValue
978	+28596	M->S	Empty PDU	1	1	1	0	0					
	=28442594												
979	+230	S->M	L2CAP-S	2	0	1	0	27	0x0017	0x0004	0x1B	0x0017	02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02
	=28442824												
980	+29770	M->S	Empty PDU	1	0	0	0	0					
	=28472594												
981	+230	S->M	L2CAP-S	2	1	0	0	27	0x0017	0x0004	0x1B	0x0017	03 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03
	=28472824												
982	+29771	M->S	Empty PDU	1	1	1	0	0					
	=28502595												
983	+230	S->M	L2CAP-S	2	0	1	0	27	0x0017	0x0004	0x1B	0x0017	04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04 04
	=28502825												
984	+29772	M->S	Empty PDU	1	0	0	0	0					
	=28532597												
985	+230	S->M	L2CAP-S	2	1	0	0	27	0x0017	0x0004	0x1B	0x0017	05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05
	=28532827												
986	+29770	M->S	Empty PDU	1	1	1	0	0					
	=28562597												
987	+229	S->M	L2CAP-S	2	0	1	0	27	0x0017	0x0004	0x1B	0x0017	06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06 06
	=28562826												

Figure 14: Packet-sniffed connection between Apple iPhone 4s and nrf8001. Notification rate 1 packet every 30ms. Some fields removed due to page space constraints

Initially, the device was configured to send a large amount of notification within a single {CE and at the lowest CI. The master device was an Apple iPhone 4S, and it was found the total throughput was 5328 bit/s or 2.3

As the device can only send 1 packet per CI, to achieve the highest throughput, the device must be run at the smallest connection interval of 7.5ms. At this interval the upper bound for throughput becomes

$$\frac{1000}{7.5} \times 20\, bytes \approx 2666\, byte/s$$

With 16 channels at an 8 bit resolution, the highest frequency measurable (according to Nyquist condition) is

$$\frac{12666\, byte/s}{16} \times \frac{1}{2} \approx 83Hz$$

A maximum support frequency running a solo channel is approximately 1328 Hz.

4.1.2 nRF51822

Another NS product tested was the nRF51822. Information on the suppport packets per connection event was also not present in the datasheet[9]. This device was found capable of receiving up to 6 packets per second, and this was verified by an engineer[7]. While a marked improvement over the nRF8001, this meant that the device was only capable of operating at 60

Unfortunately the chip came as part of a larger development kit and accurate power measurements were not attainable, however the support softwares suite included a power calculator [CITE BATTERY SECTION].

4.1.3 CSR1010

Through direct contact with CSR engineers, it was reported a particular single mode BLE radio device, the CSR1010, was measured reaching 250kbps

> *. . . the theoretical maximum throughput for a single BLE connection using a much bandwidth as possible is around 300kbps at the air interface. The usable application data rate is somewhat less, but 80kbps should be possible (Using ATT packets with a 23-byte MTU should provide approximately 200kbps application data rate, if my maths is correct).*
>
> *However, in practise you are limited by what other activities each end of the link is doing. In particular smart-phones (as one end of the BLE link) will likely severely restrict the BLE throughput to account for other Bluetooth activities (such as eSCO links) and co-existence with WiFi etc. But if you controller both ends of the link (e.g. single-mode BLE devices at both ends) then it should be easier to get close to the maximum theoretical rate.* [sic]

[sic]

In the first paragraph the engineer is treating the full 27 bytes in the data payload as the application data. This number should of come from

$$\frac{1000000\mu s}{676\mu s} \times 27 < 320kbps$$

320 to 300 is a generous approximation, it is believed that the engineer actually rounded a smaller number, 305, which was calculated by including an unecessary 32 bit message integreity code (MIC), used only when encryption is enabled:

$$\frac{1000000\mu s}{708\mu s} \times 27 \approx 305kbps \approx 300kbps$$

Many literature pieces and training materials on BLE make this assumption, including those on the SIG website [4] [1].

In the second paragraph the engineer is highlighting some important facets of BLE devices, and that is although one end of the link is determined by the lower of the two device's thorughputs. The engineer cites smart phones as such devices that restrict the link capability. This was seen in the nRF8001 - the iPhone 4S could not handle a connection interval less than 30ms. Combined with that the nRF8001 stack couldn't transmit more than one packet per connection interval on average, the maximum throughput achievable by this device was cut by 4.

When asked how to show the math behind the numbers achieved

> *While testing our uEnergy silicon (CSR1010) at HCI level I can achieve 250kbps throughput using a dedicated test application (i.e. data generated on-chip). HCI packets are 27 bytes, so to get my earlier number I simply did (20/27) * 250 to get an application-level data rate (27 bytes minus 4 bytes L2CAP header minus 3 bytes ATT header, leaving 20 bytes for application in each packet).*

I haven't tested raw throughput at the application level, but as our ATT protocol runs on stack alongside the HCI/controlller level, it should not see any additional processing overheads. [sic]

The HCI is the component between the host and controller layers shown in Figure 3. The engineer is saying that data throughput should no be affected by any high-level stack activity (any stack congestion, if existing, shouldn't affect throughput). Although the engineer has given no commet on latency (data will take time to propagate up the stack).

CSR supplied a development kit consisting of provided a development board (CNS10004V5D) and Universal Serial Bus (USB) dongle. The CSR1010 is part of CSR 's new μEnergy product line, for which CSR have developed their own compiler and integrated development environment (IDE) known as xIDE. For initial throughput testing, an example health thermometer application was used. From exploring and understanding the main features of the operating system abstraction layer (OSAL), it was immediately possible to increase the number of notification packets sent per connection interval from 1 to 8.

The standard mechanism provided in the example, is for every connection interval a confirmation of the first acknowledge is raised in the application layer. This is used in the program as a point of reference for application layer timers, so schedule the next wake up time, etc. This mechanism provides visibility at the **CE!** (**CE!**) level, but not at the notification packet level - if many notifications are sent and acknowledged, this method doesn't provide any information that the **CE!** may have room for more notifications. This is a problem when large throughputs are desired, as it appears the buffer queue is 8 packets in size. When more than 8 are added to the queue, they are ignored, causing data loss. Hence, through the naive method, whereby notifications are placed into the queue before or after the start of the **CE!**, upto a maximum of 8 notifications are correctly sent in order. While it is possible to continously flood the stack (buffer queues) with more notification requests, without knowing if the stack is capable of dealing with them, data will be lost. This mechanism is visualised in Figure 15.

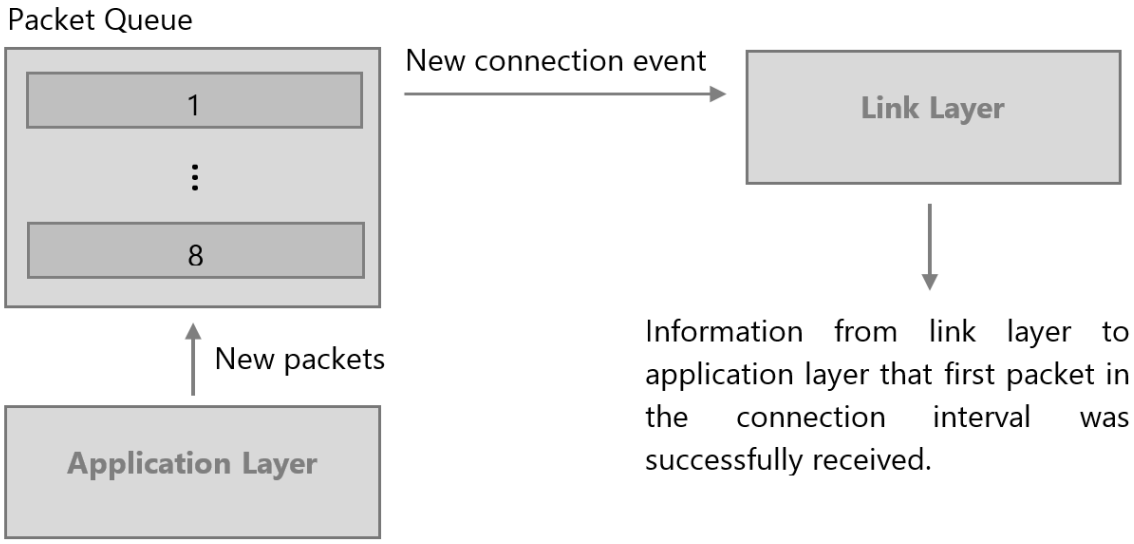


Figure 15: Notification of the first packet

An experiment to test throughput was setup whereby each connection event sends 8 notifications per connection event with the same payload, bar one byte which increments with the connection event. The output can be seen in Figure 16.

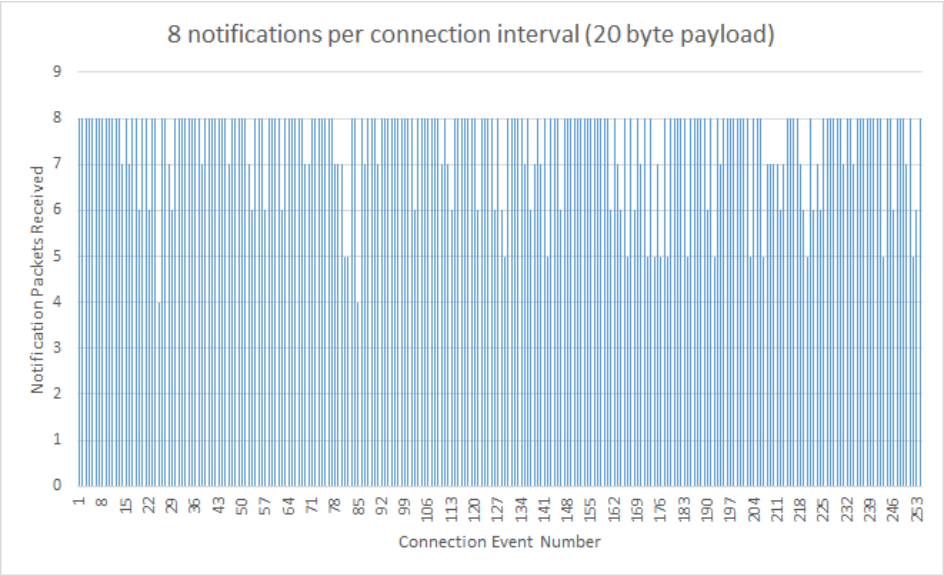


Figure 16: Queuing of 8 packets being lost

From the experiment above, the conclusion was that the link layer stack became flooded with notification requests. Had the requests been accepted into the stack, then they would of been transmitted. If they failed to arrive at their destination, then the flow would of caused a negative acknowledgment, and the packet would of been retransmitted. Provided the notification made it through the link layer stack, it should appear in the output.

After greater exploration of the software system framework, the application programming interface (API) and software mechanisms, it's possible use the message pump to signal whenever a packet is acknowledged. Upon receiving this signal, the device can then dispatch a subsequent packet. Figure 17 shows such a mechanism.

The experiment was repeated again, this time sending a new notification every time the previous one was acknowledged using the message pump signal. The results are shown in Figure 18

A capture of the notification data was taken over a period of approximately 46 seconds. Figure 19 shows part of the capture.

In the 46.060 second period a total of 55286 notification packets were captured, bringing the average throughput to

$$\frac{55286}{46.060} = 1200.30395137\ packets/s$$

For a 7.5ms CI, the number of **CE!** is 133.333 and the number of packets per connection interval is

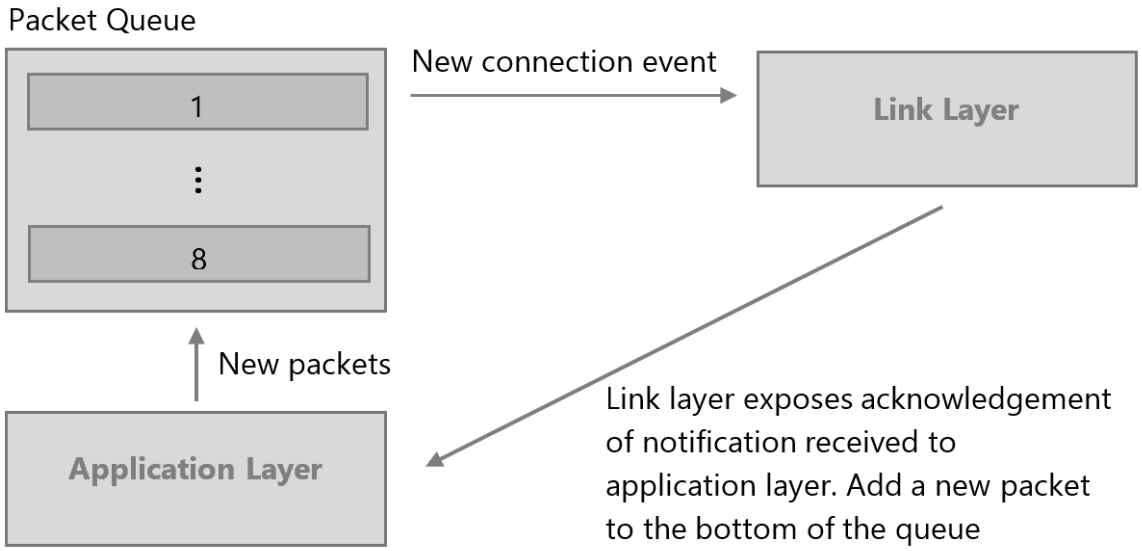


Figure 17: Upon packet acknowledgment

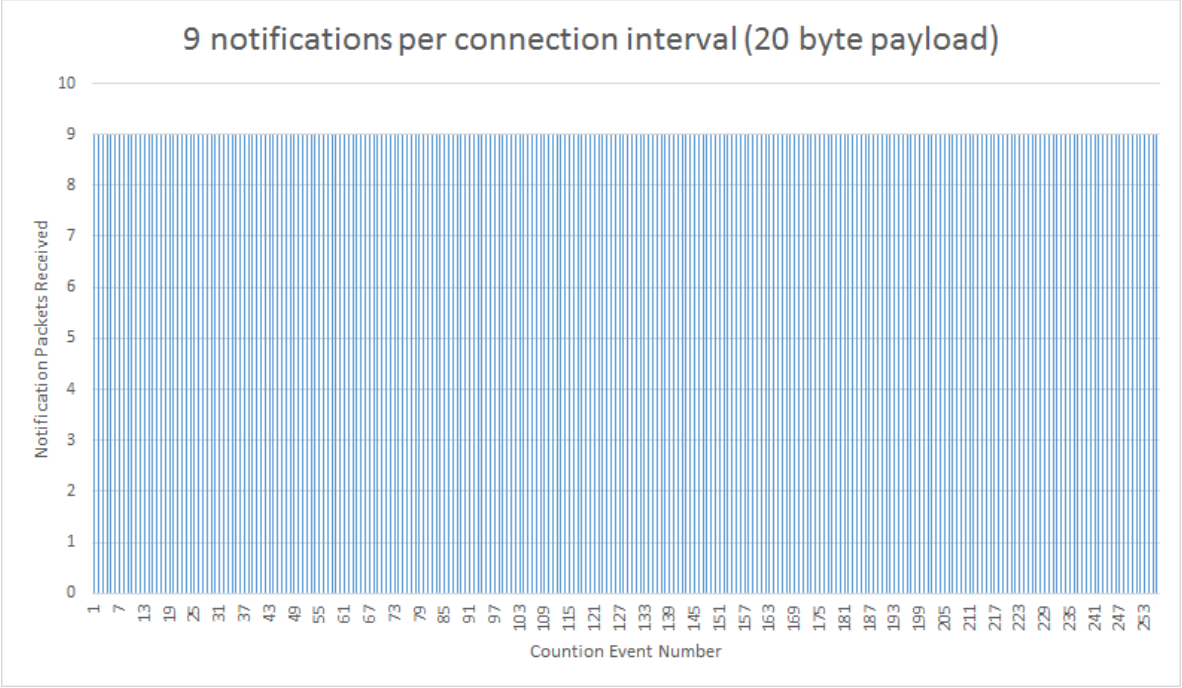


Figure 18: 9 packets being received per connection interval when through acknowledgment feedback mechanism

$$\frac{1200.30395137}{133.3333} = 9.0027\ packets/s$$

Therefore 9 packets per connection interval at 7.5ms. This corresponding to a throughput of 192kbps or 90% of the theoretical maximum. A comprehensive explanation of the code used to achieved this is provided in the later section **??**.

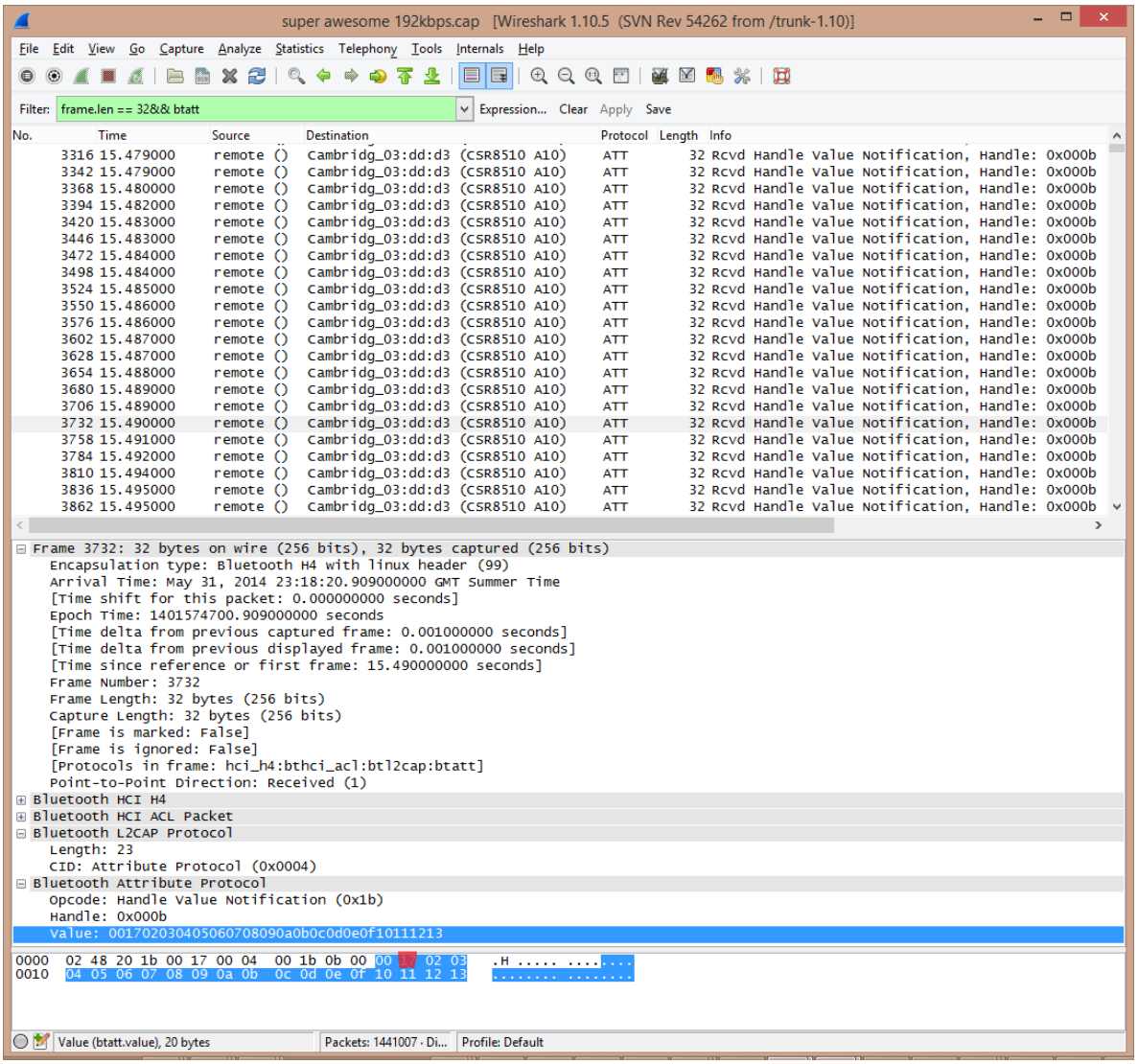


Figure 19: Packet capture. 20 byte data payload (blue) and connection event counter variable highlighted (red). Resolution of time stamps is to the nearest millisecond

4.1.4 CC2540 and CC2541

TI's CC2541 is part of the sensor tag package - a BLE one of the first, and arguably the most popular [need to CITE?] development kits. The CC2540 differs in that it's range and power usage are increased, though it's throughput is still theoretically the same.

Again, the associated white papers did not document the packet throughput per connection event. Through contact with a TI employee [CITE], there is a hardware limitation of 4 buffers, limiting only 4 packets per connection event (each buffer contains 1 packet). Each buffer is 128 bytes in length, while BLE data packets can be a maximum of 41 bytes. Through using an exposed firmware (software) switch it was possible to increase the throughput slightly to achieve 5 packets per connection event by refilling the buffers within the connection event, once the packet has been link layer acknowledged. Again, this was discovered through a Texas instrument employee. This increased the average packets per connection event to approximately 4.6, bringing the total throughput between 10-12.3k byte/s.

The CC2541 (and also CC2540) is a full SoC - a programmable microcontroller (MCU) is packages togeter with the radio hardware. This has certain economical advantages, as common hardware and footprint is shared between the radio and MCU. However, the Nordic company Chipcon who originally developed the hardware (Texas Instruments has since acquired them), relied on the Swedish tool chain provider, IAR, for the development tool suite and chain. TI still relies up this tool software suite, haven't not invested in their own facilities to develop for the platform. A single software license is approximately 2000 USD. While IAR do offer a code-limited version, it is too small to contain all the necessary libraries. IAR also provide a 30 day trail, which is what is currently being used for evaluation, but proceeding with this chip for the prototype may be economically impossible.

Much like the CSR1010, a similar OSAL existed for the CC2540/1,

For 16 channels running simultaneously, at 8 bit resolution, the highest achievable detectable frequency is approximately 384 Hz.

4.2 Batteries

There are many different battery technologies available. For the device a small, low profile low footprint, a lightweight battery is required. Given that BLE can consume upto 15mA at peak current draw, taking this as the upper bound means that for 12 hours of continuous use batteries with capacities of 180mAh should be considered. Although the radio will dominate, other parts of the circuit such as the Microprocessor will consume a small amount of power, so this figure is arbitrarily upped to 200mAh. Note that peak current is an unfair way to measure current consumption in BLE, however it is being used here as a worst case scenario.

Two batteries technologies which satisfy these requirements are Lithium-ion (the popular CR2032 is typically advertised as an energy source with many BLE modules) or Zinc-air batteries with capacities of 600mAh cheaply available. Common voltages are 1.4V to 1.65V (radios require between 1.8 -3.6V). However the batteries are light, less than 2g each (one CR2032 weighs, on average, over 3g), therefore they can be combined in series to provide a suitable voltage and a large capacity (over 1Ah). Further, this high voltage will allow the effective radio range of the BLE device to increase. If a Zinc-air cell exhibits damage, unlike most technologies, no dangerous chemicals are present. The downside to zinc-air batteries is that they must be vented to an oxygen environment (cannot be hermetically sealed) which in turn causes them to lose their charge over time relatively quick once exposed to air. Although the Ambulatory Electroencephalogram (AEEG) use case requirements will require the device to have lifetime of hours to days, not years.

Two Zinc-air batteries weighing in at 1.9g each, can supply over 1000mAh with a combined series voltage of 2.8V. In the worst case of BLE drawing 15mA, the system lifetime is approximately 3 days. Recall 15mA is the peak current and BLE will never continuously draw this current, hence the lifetime of the device should be greater, depending on the throughput. acBLE was designed that current is never continuously consumed, and between radio events, the battery has recovery periods, extending the useful life of the battery.

Nordic semiconductor provides a system lifetime calculator. Using a typical Lithium-Ion battery of 220mAh at a 7.5ms connection interval, yields a lifetime of 163hours. This makes sense

as the current average current consumption is slightly above 1.3mA and the capacity of the battery is 220mA. Accuracy of this model will require vertification in the power analysis of the AEEG, but the simulation looks sensible. Figure 20 shows a notification simulation of the EEG service being sent continuously every 7.5ms. This model is only for 1 notification per connection event. If

$$\frac{7.5ms}{0.676ms} = 11.094\ notificationpacketsperconnecitoninterval \frac{163hoursfor1notificationperconnectioninterval}{11.094packetsperconnectioninterval} \approx 14.7h$$

This should be an overestimation of the power requirements, as there is a fixed cost payed for waking the MCU, which would be amortised for higher throughputs. Further in the model above the idle time contributes to the power consumption, but at full throughput, there will be little idle time. Finally, running at full throughput is not necessairly the end used case, hence, system

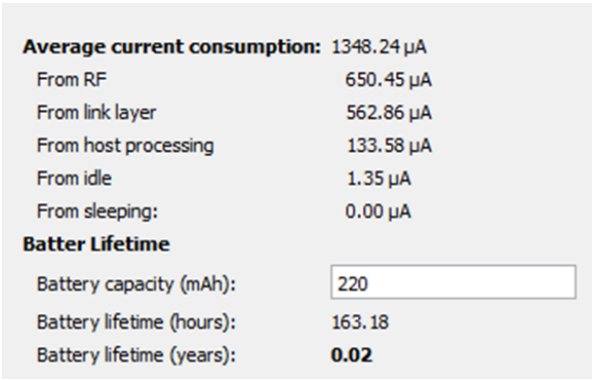


Figure 20: Power consumption and lifetime calculator

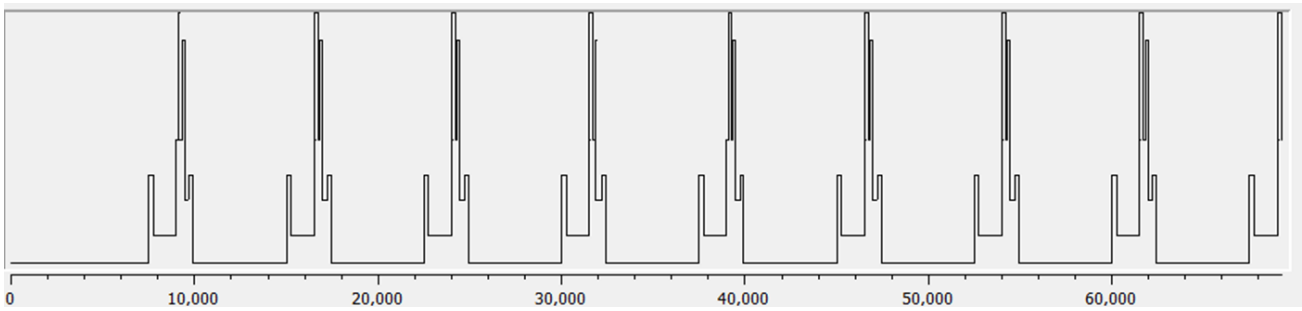


Figure 21: nRF8001 waveform (1 transmission per connection event

Figure 23 - Simulation plot of notifications being sent For 2 zinc-air batteries of 500mAh capacities, this model predicts the device lifetime under nominal operation begins to reach 10 days. An intelligent estimate to how this module is generated is linked to the throughputs calculated at the start of the Radios section. As each bit represents 1 micro second it is very easy to integrate for set parameters the time over time the current profiles to achieve an estimate of power usage. Max drain current is another important concept to consider. Batteries can only safely deliver a maximum amount of current. If the battery is stressed to deliver more current that it is rated, it will degrade the overall lifetime of the battery.

4.3 Microcontroller

There are a couple of options here. Often radios are provided as a system on chip (SoC), whereby the microchip contains not just the radio but also a fully functional microcontroller unit (MCU). Often these devices will consume more power than just the host and controller, but the system as a whole will typically consume more power overall if the application is run on an off chip MCU.

Figure 15 - SoC (a) and a typical 2 chip solution (b) (Heydon, 2012) It really comes down to the application, as having the application on an off-chip MCU may mean that power is saved overall as other features such as ADCs, SPI interfaces and other peripherals can be utilised and integrated into the application without having to run a separate application on both the external MCU and radio chip, which leads to increased overhead, complexity and power.

For development purposes the Arduino family of development boards, which sport many different processors seem like an excellent for development. These processors have many features such as GPIO, inbuilt ADCs, dedicated SPI pins, etc. Further, there is an extensive documentation, large knowledge base and a very active online forum. Of particular interest are the earlier Arduino models which utilise the very low power Atmel 8-bit family of processors. A further plus is that these processors are available in both DIP and SMT packages, meaning they are suitable breadboard prototyping. The ATMega328 seems like a particurly good choice, as a balance between support, power and simplicity.

Currently, Texas Instrument's SensorTag have been investigated along with Nordic semiconductors nRF8001. The former is a SoC solution while the latter only implements the host and controller layers. Implementing an application layer is no small feat, and fortunately it was possible to make contact with someone who has developed libraries to drive the chip using an Arduino microcontroller. Please see the section titled Experimentation and Current for more information.

Currently the general feeling is towards using an Arduino micro controller simply due to the ease and rapid nature of development. Some components (an particular ADC and the nRF8001) have been tested and confirmed to work with the device.

4.4 Memory

A hard specification is the real time update of the EEG signals. Long term storage is not considered real time. NAND flash chips Further, high capacity memory modules come in small packages, some requiring factory assembly.

An option to achieve large storage capacity without difficult packages is to use popular SD cards. SD cards, and the variation, microSD cards are extremely easy to communication and an experiment was performed with an Arduino to discover power , However, during reading and writing the current consumption is between 25 to 100mA. From reviewing [san disk cite], a write can require 250ms. Assuming data is written in blocks, the average current consumption attributed by the NAND is 2.5mA.

4.5 Analogue to Digital Converter

Acquisition of EEG signals from the front end consists of using an analogue to digital converter (ADC). Many BLE SoC contain ADCs, though, for example the CSR1010 are only capable to gathering 700 samples per second. Operating the ADC requires a high speed controller. For high acqutisition rates (outside the standard operating range of EEG signals), the MCU will likely need to remain in an idle (active) state. For example, it takes the CSR1010 2.2ms to power up from a deep sleep state, meaning for acqisition rates over 450Hz, the chip would be required to remain in an idle (active) state as powering down to a lower power would miss samples.

The CSR1010 has an idle current of approximately than 1mA at 3.3V. The Arduino Uno micro controller is the ATMega328P. While the ATMega328's claims to consume only 0.2mA at 1.8V, this occurs when the device is clocked at 1MHz. This speed may not be sufficient to run the application layer with the responsiveness required, and the MCU is normally clocked at either 8MHz or 16 for most applications. Increasing the clock rate increases the current consumption. For the radio circuit and battery constrains, the expected voltage is expect to be 3V. While a voltage regulator can be used, there will be losses in conversion. Further power is required for running the hardware to enable communication between the radio and ADC. Running the ATMega328 at the same clock rate as the CSR1010, at 3.3Vs the report active (idle) current consumption is

The ATMega328 contains inbulit ADCs capable of suitably high sample rates, however it was found running these increases the current consumption by atleast 1.5mA.

All BLE chips tested without the application layer, was found to exhibit a low throughput. Given this, the only BLE radios worth running already contain a full SoC, whereby the MCU is capable of communicating with an off-chip ADC. This and the power, complexity (communication and technology) and footprint size of using multiple active components, it was decided the best course of action would be too use an off-chip ADC connected to the BLE SoC.

The most obvious and popular protocols for interfacing with an external ADC are Serial Peripheral Interface (SPI) and I2C. The microchip MCP3008 has a resolution of 10 bits and uses the SPI protocol to interface. While confirmed to work with the ATMega328 MCU during preliminary prototyping (a dual in-line package (DIP) package was available), neither the CSR1010 nor the CC2541 support native SPI, containing no generic SPI driver libraries either. The protocol would have to be created in the application layer through modification of the programmable input/output (PIO) pins, commonly known as "bit banging". Achievable throughputs would not be known until implemented, leading to a development risk in the meantime.

The CSR1010 supports I2C natively along with provided a generic driver at the application layer. The most suitable device found was the Analogue Devices' AD7997 10bit 8 channel I2C ADC. The device can operate at different I2C speed, 100KHz and 400KHz. While there is the risk of interface issues, the level is less than that of the MCP3008, and believed tolerable. It is also highlighted that the correct working of the ADC is not crucial to the project's success. This, and the fact that no analogue front end is available.

4.6 Analogue Frontend

The biopoentials measured on the scalp required analogue filters and amplifiers to extract useful information. Ultimately the ADCs should interface with this frontend. At the time of writing, the circuits and systems group at Imperial College have recently taped out a full custom silicon analouge front-end design for manufacture, however these chips will not be available for use before the project deadline. Therefore no analogue front-end electronics will be present on the prototype board.

4.7 Component Selection

The decision for the radio chip

While the CC2541 showed promise as with enough hardware understanding and firmware switches, it may mean the device could approach the theoretical limit, the unavailability of an inexpensive tool chain meant software development was problematic. The CSR1010 is supplied with a free tool chain and IDE, as CSR engineers confirmed experiments with the device operating with high throughputs. The package of the radios were both similar, being of roughly the same size quad flat no-leads (QFN) package type, which while reported outside the department's PCB production capabilities, was still considered of sufficiently large for in-house prototyping.

The application will be developed on a tablet, due to the expected ease of development. It is difficult to source either an Android and iPhone device capable of many notifications per CI, little information available and costly. The target platform is the Surface Pro, running Windows 8.1. While the Surface Pro contains an internal multi-mode wireless chip, BLE capable, it is not known how well this will fare for higher throughputs. CSR provided a dongle, which after preliminary testing, appears to be able to achieve high throughputs. This is another reason for using the tablet, as it contains a USB port. Further, developing for Windows 8.1 on a tablet is similar to developing for Windows Phone 8, hence, although

Ultimately, the decision is to use an off-board ADC module, the AD7997. The prototype will interface with two of these over an I2C bus, to provide 16 channels.

Considering the requirement for real time throughput, footprint, manufacturing and assembly challenges, power consumption and added design complexity, there will be no use of NAND memory in the immediate prototypes, except that used internally inside a MCU or externally as a bootloader (i.e. CSR1010 device requires a small external storage unit which contains the program code, loaded at power on time)

The final component selection for the EEG prototype is

- AD7997 8bit I2C ADC. Two are required for 16 channels
- CSR1010
- P675 Zinc Air batteries. Batteries of capacity 620mAh are readily available and inexpensive. Two are required in series to achieve an operable voltage

- AT24C512C-SSHD 512K Serial EEPROM Flash for CSR's non-volatile storage

The final PCB prototype is expected to consist of 4 active components.

5 Specification

6 Implementation

6.1 Hardware

A challanging part of this project develping hardware. It was desired that the device should be small and light so it can be shown into a EEG hat and worn by the user with no discomfort. To achieve a small size components were placed within close proximity of each other, and at angled orientatinons (tesselated)

Typically,

The first iteration of the board. Due to the process capabilities and materials used, the radio frequency traces were not impedance matched, and the device range was limited to approximately 7 meters within a building or 15 meters line of sight (LOS).

To test both the process capability and variation, lines of decreasing thickness along with convex shapes were used . Total size was 27.5 by 34mm

6.2 Firmware

6.3 Tablet Application

7 Results

Weighing in at a total of

8 Evaluation

The layout can be shurnk further through reducing distances between components. This was not done previously as it significantly increases the difficultly in hand assembly, which was the method used for the prototype. For the analouge front end

8.1 Power Analysis

The hardware

Assuming that the number of

The idea of minimising energy per bit is not the true picture, as BTC and WiFi have higher energy per bit ratios. To achieve the lowest energy per bit, the transmission rate needs to be at the highest throughput achievable on the hardware due to the fixed costs or sunk costs of waking the MCU and preparing the device for radio transmission. However, consistently running at such throughput invalidates the use case for BLE, as there are other technologies capable of transmitting at the same or higher throughputs for smaller energy requirements.

8.2 Bill of materials

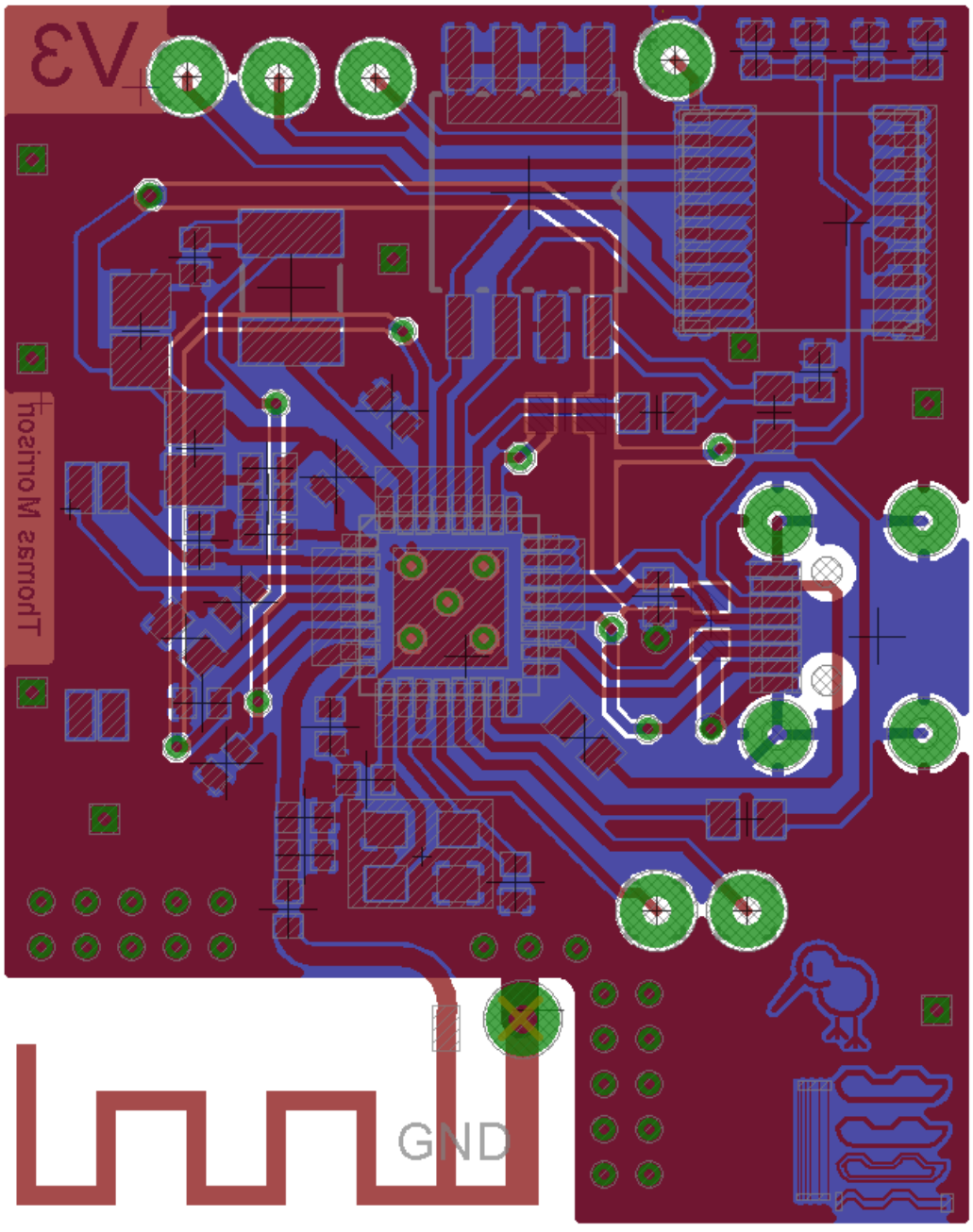


Figure 22: First PCB Design

Figure 23: First (working) PCB iteration

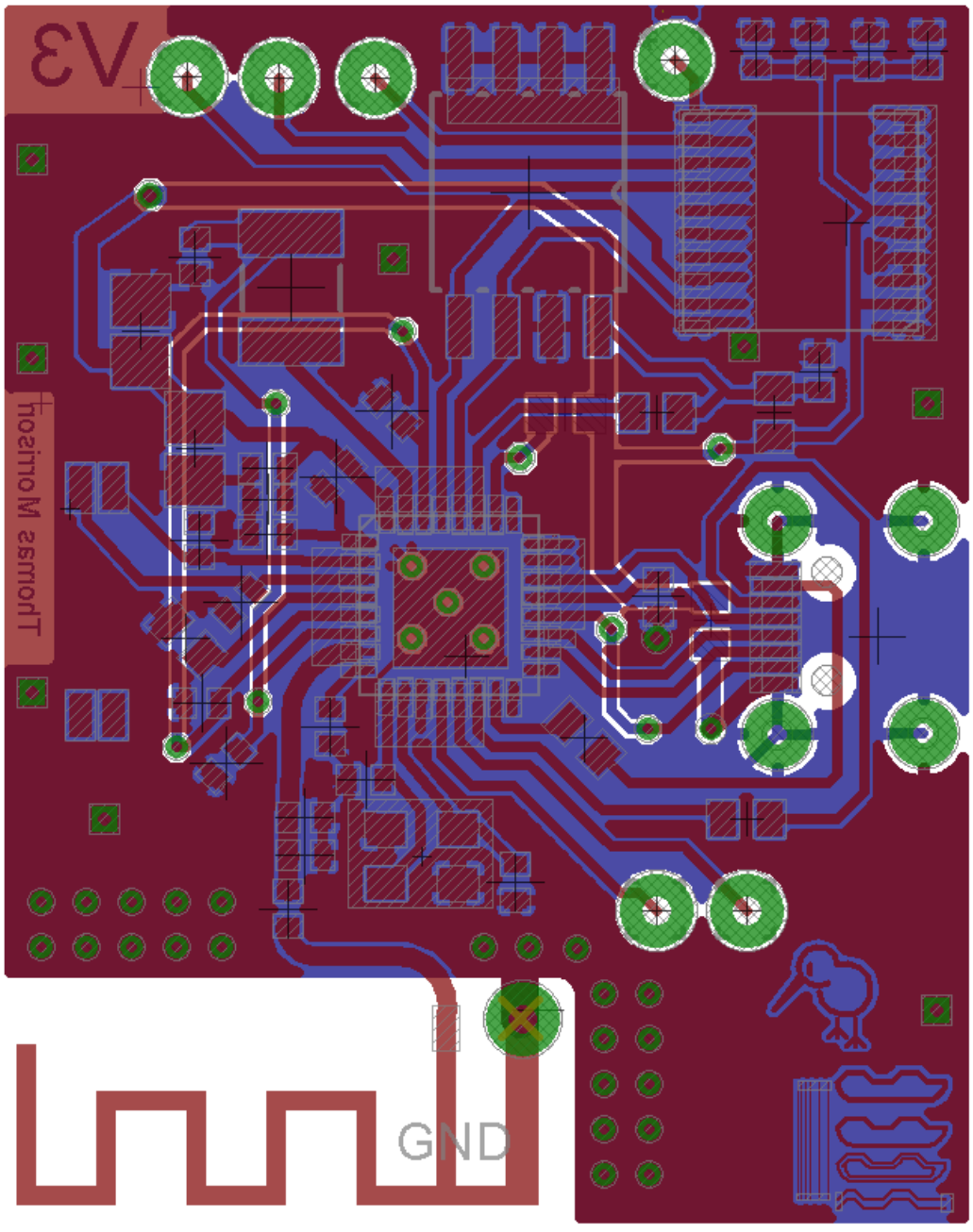


Figure 24: Schematic

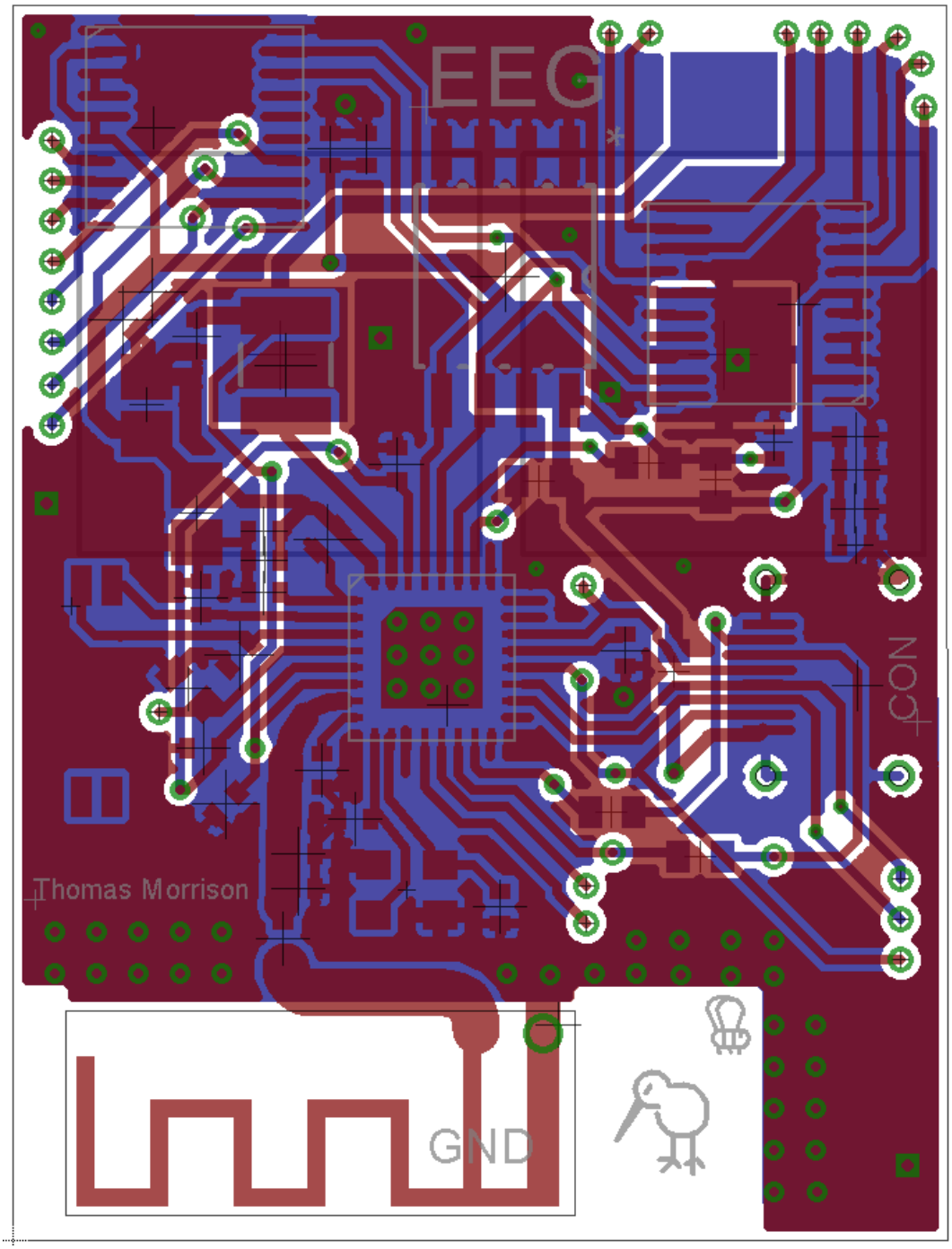


Figure 25: Final PCB Design

9 Conclusions and Future Work

The majority of interesting EEG signals exist between . Running an EEG device at that

The proposed device is comparable to a hearing aid, even using the same battery technology.

Smart phone not single end device -> contain other radios for high throughput, maybe why hardware only supports 4 packets

In the sole pursuit of maximizing throughput, the idea is to

$$\min_x \frac{1250 \times x}{708} - \lfloor \frac{1250 \times x}{708} \rfloor$$

Mathematically this finds the points where

As CI increases chance of CRC error increases. Trade off between CI length, remainder

Track /feedback for optiminum

Unfortunately a connection interval must be between 7.5ms and 4000ms long, in incremental steps of 1.25ms. Therefore x must exist in the range of 6 to 3200.

//but unfortunately, there is an energy cost associated with hardware buffers from operation and leakage current

Hard to classify

On the whole, the project has Both ends of the link

Not all tablets will be able to run at full speed

Ideally, it is desirable to write a conference style paper.

Never list packets per CI

Useful for CSR, monies worth out of me

Duplex

10 Final Remarks

A vast array of technologies and tools were used throughout this project. Below highlights those that are non-trivial and were of significant importance

- Git verisioning control - For versioning and segmenting the workflow
- xIDE - CSRs integrated development suite for compiling and debugging CSR-stack based chips. This is where the firmware for the CSR1010 MCU was written

- Wireshark - Invaluable in analysing the packets and control flow between BLE radios, unfortunately must be used offline
- SmartRF online packet sniffer - Useful as a low-speed packet sniffer. Extremely useful in the early days of the project, however the throughputs eventually obtained rendered the tool and hardware redundant as it was not capable of such speeds
- Visual Studio 2013 - Primairly used for developing the tablet application. Highly useful for "knocking up" quick throughput expirements

The large amount of both written and generated code is to vast to warrant being included in this report. Therefore is has been decided to make it publically available online at the git repositry address `https://github.com/proftom/AmbulatoryEEG`.

Numbers all over the place Alot quote 708 as minimum, but it's 676 and confirmed http://www.ncbi.nlm.nih.gov/

11 Bibliography

References

[1] SIG. *Fast, Simple Debugging for Bluetooth Low Energy Applications*. 2012. URL: https://www.google.co.uk/url?sa=t&rct=j&q=&esrc=s&source=web&cd=5&ved=0CE0QFjAE&url=https%3A%2F%2Fdeveloper.bluetooth.org%2FDevelopmentResources%2FDocuments%2FWebinar_FastSimpleDebugging%2520Frontline-SIG.pdf&ei=lhaTU5OPJqGS7Ab-9IHwAQ&usg=AFQjCNHSkvdA4Vf2aGh-FJP4NpVUOmW0-A&sig2=wKQSG9ZYwzweJomkD6x2jg&bvm=bv.68445247,d.ZGU&cad=rja.

[2] Alex W.; Ault Aaron; Krogmeier James V.; Buckmaster Dennis R. Balmos Andrew D.; Layton. "Investigation of Bluetooth Communications for Low-Power Embedded Sensor Networks in Argiculter". In: 131620559. 2013 ASABE Annual International Meeting. 2013. URL: https://engineering.purdue.edu/oatgroup/docs/pape\r_balmos_1.pdf.

[3] R. Quian; Rosso O. A.; Kochen S. Blanco S; Quiroga. *Time-frequency analysis of electroencephalogram series*. Tech. rep. 1995, pp. 1–3.

[4] Nick Hunn (WiFore) Bluetooth SIG Robin Heydon (CSR). *Bluetooth low energy*. URL: https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?doc_id=227336.

[5] Josep Paradells Carles Gomez Joaquim Oller. "Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology". In: (2012). URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3478807/.

[6] Guan Yang Jacob Rosenthal. *nRF8001 support for Arduino*. https://github.com/guanix/arduinonrf8001. 2013.

[7] Ole Morten. *Nordic Semiconductor employee*. 2013. URL: https://devzone.nordicsemi.com/question/3440/how-do-i-calculate-throughput-for-a-ble-link.

[8] Joakim Lindh Sandeep Kamath. *Measuring Bluetooth Low Energy Power Consumption*. 2013. URL: http://www.ti.com/lit/an/swra347a/swra347a.pdf.

[9] Nordic Semiconductor. *nRF51822 Datasheet*. 2013, pp. 1–67. URL: http://www.100y.com.tw/pdf_file/39-Nordic-NRF51822.pdf.

[10] Nordic Semiconductor. *nRF8001 Datasheet*. Tech. rep. 2013, pp. 1–161. URL: http://www.nordicsemi.com/kor/content/download/2981/38488/file/nRF8001_PS_v1.2.pdf.

12 Appendix

12.1 Acronyms

EEG Electroencephalography

AEEG Ambulatory Electroencephalogram

BLE Bluetooth Low Energy

BTC Bluetooth Classic

CSR Cambridge Silicon Radio

TI Texas Instruments

NS Nordic Semiconductor

GATT Generic Attribute Profile

GAP Generic Access Profile

ATT Attribute Protocol

CI connection interval

SIG Special Interests Group

CODEC coder/decoder

CRC Cyclic Redundancy Check

PDU Packet Data Unit

SN Sequenc Number

NESN Next Expected Sequence Number

LOS line of sight

SoC system on chip

MCU microcontroller

API application programming interface

IDE integrated development environment

OSAL operating system abstraction layer

USB Universal Serial Bus

PCB printed circuit board

QFN quad flat no-leads

ADC analogue to digital converter

DIP dual in-line package

SPI Serial Peripheral Interface

PIO programmable input/output

HCI host controller interface

MIC message integreity code

13 User Guide