Bikash Shrestha

Measurement of power consumption of BLE (Bluetooth Low Energy)

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Thesis

6th October 2016

Helsinki
Metropolia
University of Applied Sciences

Author(s) Title Number of Pages Date	Bikash Shrestha Measurement of power consumption of Bluetooth low energy (BLE) 36 pages 6th October 2016
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Embedded Systems
Instructor(s)	Anssi Ikkonen, Department Head of Information Technology Antti Laurikainen, Project Supervisor Sampo Nurmentaus, Project Supervisor

The purpose of the project was to measure the power consumption of Bluetooth low energy (BLE). The main goal of the project was to study the possibility of using the BLE with the power generated by various energy harvesters to create Internet of Things (IoT). This paper also contains some theoretical research on various energy harvesting systems.

The project was carried out with nRF51422 Development Kit and nRF-Software Development Kit version 11 developed by Nordic Semiconductors. The BLE embedded in the development kit was used to measure the power consumption under different circumstances. The BLE device goes through various phases due to its protocol stack and consumes various amounts of power at each stage. Although the power consumption level is at its peak while the BLE stack is transmitting or receiving, the total power consumption is calculated as the average of various stages.

As a result, power consumption of the BLE was measured under different conditions and life expectancy of the energy source, supplied to run the BLE, was calculated. The calculations were studied to find out the feasibility of the BLE on the IoT.

Keywords	energy harvesters, power consumption, Internet of Things, wireless sensor network, Bluetooth low energy, power metrics

Contents

Abbreviations

BLE	Bluetooth Low Energy
GAP	Generic Access Profile
GATT	Generic Attribute Profile
IoT	Internet of Things
NFC	Near Field Communication
6LoWPAN	IPv6 over Low-power Wireless Personal Area Networks
GSM	Global System for Mobile communication
SIG	Bluetooth Special Interest Group
ULP	Ultra Low Power
RF	Radio Frequency
LC	Link Control
LM	Link Manager
L2CAP	Logical Link Control and Adaptation Protocol
HCI	Host Controller Interface
UART	Universal Asynchronous Receiver and Transmitter
QoS	Quality of Service
EHS	Energy Harvesting System
EH	Energy Harvesters
PMU	Power Management Unit
ESD	Energy Storage Device
ED	Embedded Device
SMD	Surface mount Device
SOC	System-On-Chip
SDK	Software Development Kit
TWI	Two Wire Interface
SPI	Serial Peripheral Interface
RSSI	Received Signal Strength Indicator
WSN	Wireless Sensor Network

1 Introduction

The Bluetooth low energy (BLE) has been very popular in the radio frequency wireless communication system. It has been a vital cable replacement technology in recent years especially for transmitting data on the Internet of Things. The Bluetooth low energy consumes very low energy in comparison to other wireless technologies. Large amounts of energy are wasted in our daily life. Energy is produced even while walking due to the body pressure on the land. Blowing a whistle also produces some energy. Even pressing a light switch generates some energy due to mechanical stress and this is enough for a BLE communication node to run for a few days. It is important to preserve those energies to save the world from an energy crisis. So, various energy harvesters are developed to utilize such energies.

The project was carried out to see the efficiency of energy harvested by means of various energy harvesters. Various types of the energy harvesting methods were researched theoretically and the power consumption of the BLE was measured to see the life expectancy of the BLE communication node with respect to the energy harvesting methods. Thus, the purpose of the project was to find out the effectiveness of energy harvesters on the Internet of Things.

The project was done based on a company's interest in the Internet of Things. The project was completed with the tools and devices supplied by the company. Furthermore, the nRF51422 development kit and nRF-software development kit version 11 developed by Nordic Semiconductors were used to measure the power consumption under different circumstances in the project. In addition to this, different types of BLE applications were analysed in relation to energy harvesting methods.

2 Internet of Things and Bluetooth Low Energy

The Internet of Things (IoT) starts with making of things and data come together in new ways and with actionable intelligence. The IoT links smart objects to the Internet. In other words, it is a network of data-gathering sensors communicating to other devices via mobile, virtual or instantaneous connection. In a way, it is making our lives smart. Most population on the Earth have heard about wearable devices and appliances with sensors, but what has not been marketed enough is the real business value that these “things” create [1].

While talking about the IoT, energy-efficient communication has always been the main topic to consider. The IoT devices are mostly supposed to be autonomous for years. So, the energy consumption of such devices should be minimized to save the world from an energy crisis in the future. There are various wireless protocols focused for the IoT with low energy. Some of the renowned protocols are Bluetooth Low Energy, Zigbee, WiFi, NFC, 6LoWPAN, Z-Wave and GSM. However, I will be discussing here about Bluetooth Low Energy (BLE) (nRF51422 by Nordic Semiconductors) and its power consumption measurement to estimate the life of harvested energy to run BLE applications.

The Classic Bluetooth was connection-oriented and used to maintain a link even if there was no data flowing. Even though it has been independently shown to be lower power that other radio standards, it was still not low enough power for coin-cell and energy harvesting applications. So, BLE (also called Bluetooth Smart or Version 4.0+ of the Bluetooth specification) was introduced as a new power saving alternative to the classic Bluetooth technology, by Bluetooth Special Interest Group (Bluetooth SIG) especially for proficient communication of IoT devices so that they can run for a year using a single coin-cell battery. The first low-energy chip was introduced in April 2009 but with the name Wibree. Again, it was renamed as Ultra Low Power (ULP) Bluetooth Technology and finally ended up with the name Bluetooth Low Energy (BLE) by the end of 2009. The BLE is designed aiming at low power, low latency and low throughput applications with a completely different protocol which is not compatible with the other legacy of Bluetooth protocols [2].

There are several versions of BLE such as BLE 4.0, 4.1 and 4.2. The Bluetooth Special Interest Group (SIG) has promised to deliver solid performance with a recent version of BLE (version 4.2). The benefits of BLE 4.2 over the previous version include sending data over the internet (IPv6/6LoWPAN), tracking and pairing of devices confidentially by trusted owners, 250% faster data transmission and 10 times more packet capacity.

2.1 Core System Architecture of BLE

The BLE core system architecture includes an RF transceiver, baseband and protocol stacks that enable devices to connect and exchange data. BLE devices exchange protocol signaling according to the Bluetooth specifications. The core system protocols are the radio (RF) protocol, link control (LC) protocol, link manager (LM) protocol and logical link control and adaptation protocol (L2CAP).

The lowest three system layers, the radio, link control and link manager protocols, are often grouped into a subsystem known as the Bluetooth controller. This is a common implementation that uses an optional standard interface, the Host Controller Interface (HCI) that enables two-way communication with the remainder of the Bluetooth system, called the Bluetooth host. [16.]

2.1.1 Layers of BLE Stack

The core architecture of BLE consists of many layers which are described below.

Physical layer: It controls transmission or receiving of the 2.4 GHz radio with Bluetooth communication channels. BLE uses fewer channels but broader bandwidth.

Link layer: It defines packet structure/channels, discovery/connection procedure and send/receive data.

Direct Test Mode: It allows testers to instruct the physical layer to transmit or receive a given sequence of packets, submitting commands to it either via the HCI or via a 2-wire UART interface.

Host Controller Interface (HCI): It is an optional standard interface between the Bluetooth controller subsystem and the Bluetooth host.

Logical Link Control and Adaptation Protocol (L2CAP) Layer: It is a packet-based protocol that transmits packets to the HCI or directly to the Link Manager in a hostless system. It is also responsible for higher-level protocol multiplexing, packet segmentation for the LL controller and reassembly, and the conveying of the quality of service information to higher layers.

Attribute Protocol (ATT): It allows an attribute server to expose a set of attributes and their associated values to an attribute client. These attributes can be discovered, read and written by peer devices.

Security Manager: It is responsible for device pairing and key distribution. It also provides additional cryptographic functions that may be used by other components of the stack.

Generic Attribute Profile (GATT): Using the Attribute Protocol, it describes a service framework for discovering services and for reading and writing characteristic values on a peer device. It interfaces with the application through the application's profiles.

Generic Access Profile (GAP): It works in conjunction with GATT to define the generic procedures and roles related to the discovery of Bluetooth devices and sharing information, and link management aspects of connecting to Bluetooth devices.

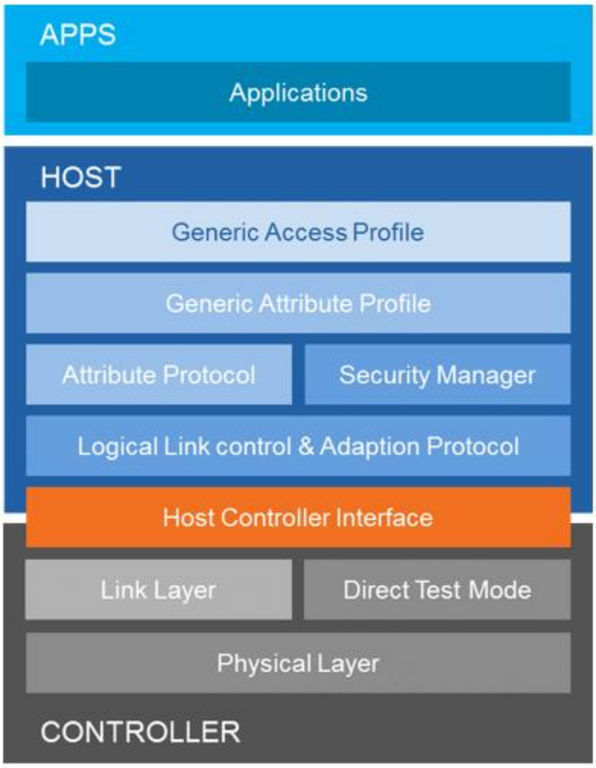

Figure 1. BLE stack. Reprinted from [16]

Figure 1 shows various layers of a BLE stack.

2.1.2 GAP Roles

GAP, Generic Access Profile, makes a BLE device visible and determines how two devices can communicate with each other. In other words, it controls connection and advertising in Bluetooth. It defines various roles for the devices in a BLE network, listed below.

- **Broadcaster:** It can work as an advertiser that is non-connectible.
- **Observer:** It can work as a scanner for advertisements but cannot initiate connections.
- **Peripheral:** It can work as an advertiser that is connectible and can operate as a slave in single-layer connections. Peripherals are low-powered as only have send small chunks of data. Once connected with the central device, it can ask the central device to change the connection parameters but cannot modify itself. However, a peripheral device can terminate the connection with the central device intentionally.
- **Central:** It can work as a scanner for advertisements and initiate connections with a peripheral. When a central device wants to connect, it sends a request connection data packet to the peripheral device. If the peripheral device accepts the request from central device, a connection will be established. The central device can only modify the connection parameters and can terminate the connection with its peripheral intentionally. It operates as a master in single or multiple link layer connections. [8.]

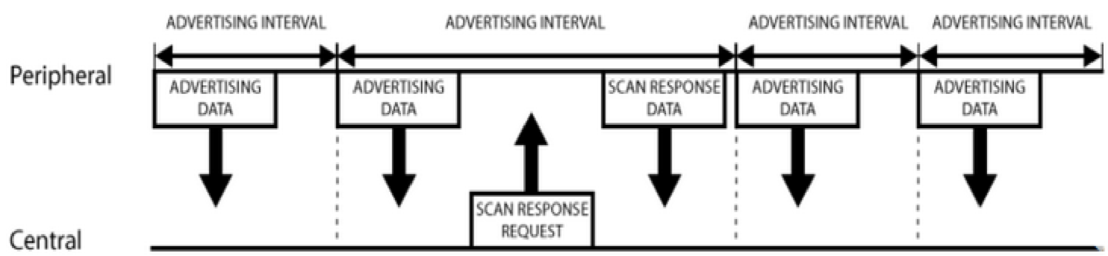

Figure 2. GAP Roles. Reprinted from [8]

Figure 2 shows the GAP roles of a BLE, peripheral and central.

2.1.3 GATT Roles

GATT, Generic Attribute Profile, makes use of the Attribute Protocol (ATT) to transfer meaningful data between BLE devices using concepts called services and characteristics. GATT is defined only when there is a dedicated connection between BLE devices. A BLE peripheral device can be connected to one central device only at a time but a central device can be connected to multiple peripherals. The peripheral, the GATT server, holds the ATT lookup data and service and characteristics definitions while the central, the GATT client, sends requests to the server. All the transactions are started by the master device, the GATT client, which receives a response in return from the slave device, the GATT server. [8.]

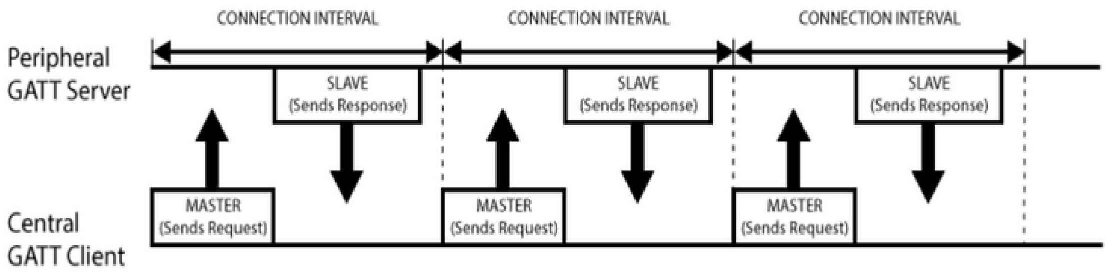

Figure 3. GATT Roles. Reprinted from [8]

Figure 3 shows the GATT roles of a BLE, GATT server and GATT client.

2.2 Applications of BLE

There are several use cases of the BLE. Some of the popular use cases are discussed below.

2.2.1 Beacon

Beacons are tiny computers that constantly send out radio signals to smartphones and tablets within the range specified, containing small chunks of data and are powered by Bluetooth low energy. The beacon technology allows a mobile application to receive and interpret the beacon's position on a micro-scale and deliver hyper-contextual content to users based on location. They are platform-independent and do not need the internet for communication. The signal strength and time between each signal can be configured for a desired coverage. Beacons can communicate only in one direction, meaning that they can broadcast but cannot read data from phones or tablets.

Beacons are designed to use very little power and send small data, typically 1-20% of the standard Bluetooth power and 15-50% of the speed. The data sent by beacons consists of a unique identifier (UUID) for the beacon vendor and a customizable “Major” and “Minor” ID which signifies different areas like stores, rooms, buildings. The mobile application triggers an appropriate action after reading data from the beacons. Usually they are powered by batteries and the battery life can be expected from 1 month to 3 years depending on the application. A beacon with high broadcasting power will last for a shorter period than the one with low broadcasting power. Also, the broadcast interval affects the battery life. The higher the broadcast interval, the higher the battery life, but a very high broadcast interval decreases the beacon’s performance. A broadcast interval larger than 1000 ms will make the beacon almost useless for most scenarios.

There are various potential applications of beacons. These includes indoor tracking, location-based messages, trigger requests for payments and automatic locking/unlocking of computers. Beacons can determine the distance to an object using the Received Signal Strength Indicator (RSSI). RSSI is the strength of the beacon's signal as seen on the receiving device like a smartphone or tablet and it depends on the distance and broadcasting power value. [10.]

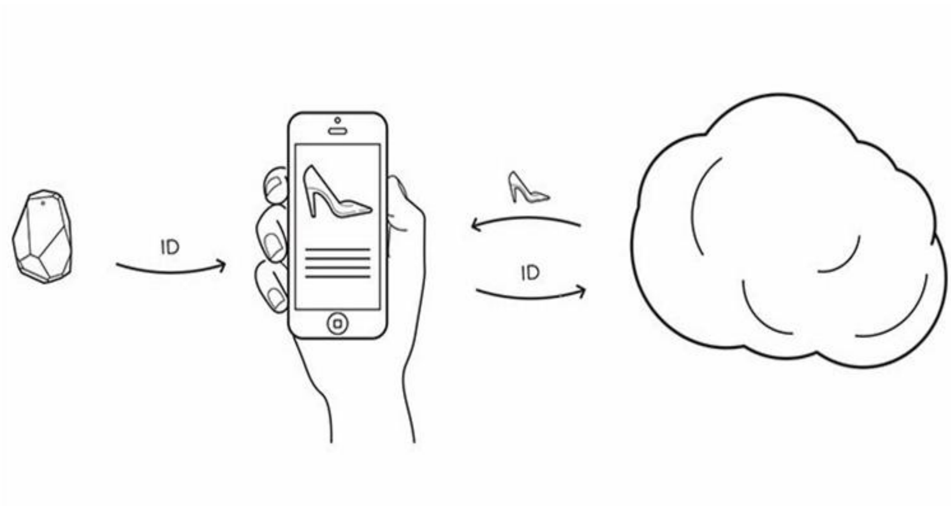

Figure 4. Beacon communication system. Reprinted from [19]

As shown in figure 4, smart devices within a range receive the signals from the beacon and response accordingly with the help of an application installed for this purpose. The cloud ties the system together by granting full access to the metadata like beacon ownership, object type and precise location. The cloud also manages the security and the other services on the top of the beacon.

2.2.2 Wireless Sensor Network

The wireless sensor network (WSN) is a group of spatially distributed sensor nodes with low maintenance requirements, which can automatically monitor environmental conditions and transfer the data collected by the sensor through a gateway to the main database using wireless networking. The networks can have different topologies; the most common are star and mesh topologies. Wireless sensor networks are designed for a long run in a “not-reachable place” and have limited battery power, computation power and memory, and these had been the challenges for wireless sensor networks in the past. So, the BLE or Bluetooth 4.0 protocol was developed with ultra-low power consumption and an increased communication range, aiming at wireless sensor networks [3]. However, the BLE cannot run for life time on just a single battery. Hence the energy harvesting techniques were proposed to slow down the depletion of battery energy.

2.2.3 Mesh Topology

In a mesh network, all devices can talk with each other either directly if within the range or indirectly via one or more intermediate nodes. The mesh networks do not have any central hubs like other topologies and offer multiple ways of getting information from one node to another. This makes an inherently reliable network design. Mesh networks can use a specific route through the network from one node to another or an approach called “flooding” where every message is sent to every node within the range and in turn, they will relay the message to each device they can reach. This is a way to relay the messages in all nodes in the network. The most important benefit of the flooding approach over routing is that less memory and processing power is required and thus, less energy is consumed. [12.]

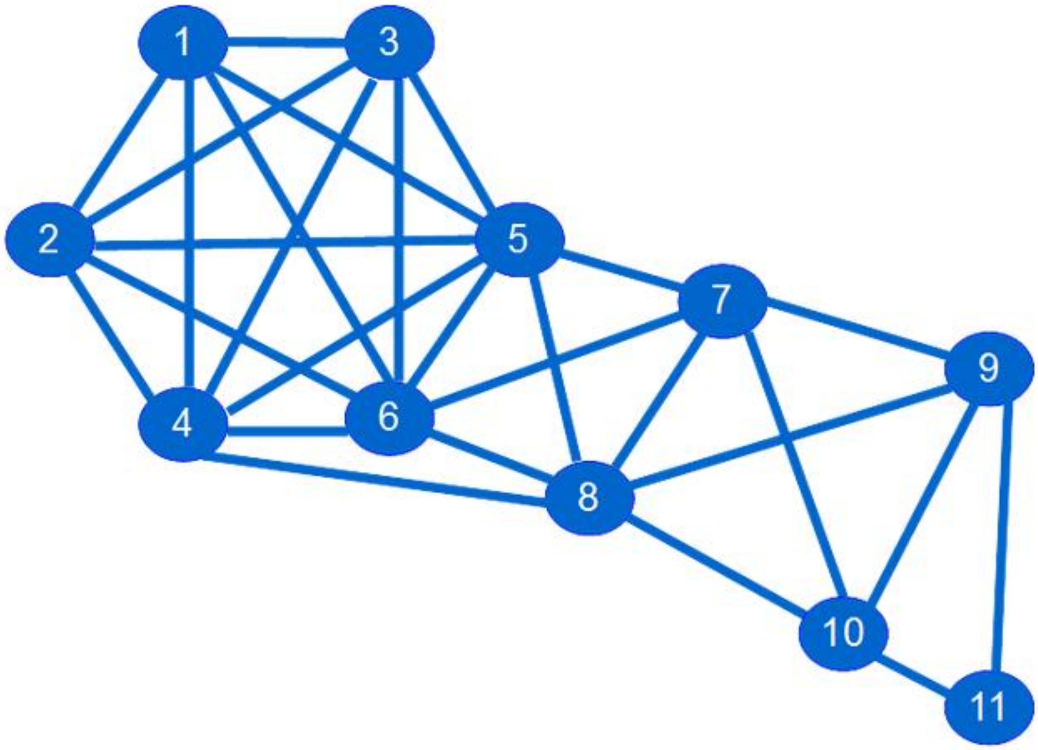

Figure 5. Mesh topology. Reprinted from [17]

The BLE mesh provides the peer-to-peer and robust multi-hop connectivity for many local BLE devices [12]. Many companies are working on mesh networking on top of the BLE protocol. The common implementation is CSRmesh from Cambridge Silicon Radio(CSR) and many companies have selected it for several products, such as smart home lighting products. However, the Bluetooth SIG is also working on the mesh network with its own study group to define an industry standard Bluetooth mesh protocol.

3 Energy Harvesting System and Power Output

3.1 Sensors

Sensors are the objects that detect events or changes in their environment and provide outputs in response. They may provide various types of outputs but typically use electrical or optical signals. They can also form a wireless network to perform a specific task. However, sensors need some energy to do processes from collecting data to sending data to the destination. Sensors are meant to send small chunks of data run for long periods. The data collected by the sensors may contain noise and they need to be filtered out to get the required value. Filtering consumes some additional power. So, they must be built in accordance with the quality of service (QoS). While maintaining the QoS, the power consumption should be considered. Sensors which consumes minimum power are durable [11]. In recent years, there have been some “smart sensors” developed. They are called so because they are smart enough to provide service consuming low energy and can be run with a small amount of energy harvested.

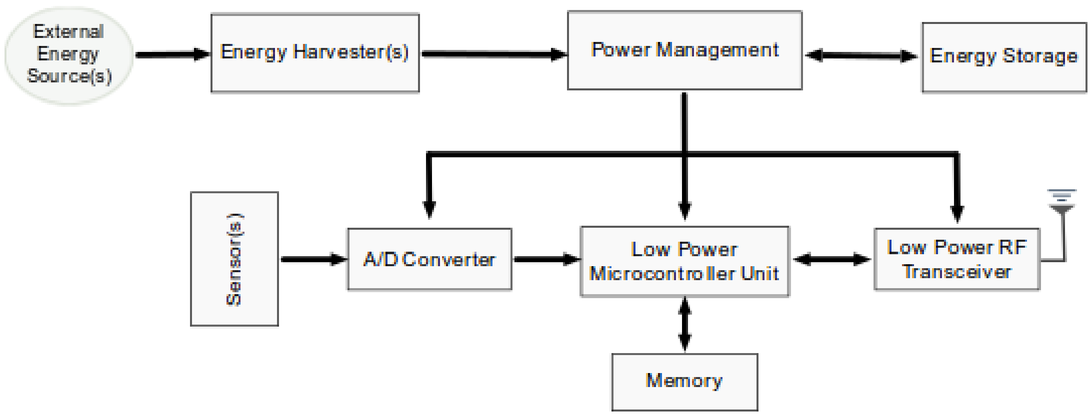

Figure 6. System architecture of energy harvesting capable sensor node. Reprinted from [11]

The system architecture shown in figure 6 consists of an energy harvesting device, a power management module, energy storage, a microcontroller, sensor, an A/D converter, a radio transceiver and a memory. The energy harvester converts external ambient or human generated energy to electrical energy which is then stored in an energy storage device (usually supercapacitors and secondary rechargeable batteries) or delivered to the node for immediate usage by the power management module. The micro-controller has some space to store the sensed data and application code. The sensed data available from the sensor is usually analog which is then digitized with the help of the A/D converter available in the system. The radio transceiver is used for transmitting and receiving the information. The BLE can be used for transmitting and receiving information as it consumes very little power.

3.2 Batteries

Batteries are devices that consist of one or more electrochemical cells with external connections provided to power electrical devices. Batteries store energy well and for a long time. Primary batteries (non-rechargeable) hold more energy than secondary (rechargeable) as the self-discharge is lower. Lead, nickel and lithium based batteries need periodic recharges to compensate for lost energy. Batteries have a higher energy density meaning that they store more energy per unit mass. Battery power is ready to be delivered within a fraction of a second. This is a great advantage over other power sources. Most rechargeable batteries can effectively handle small and large loads. In other words, they have a wide power bandwidth. The rechargeable battery has a relatively short service life even not used. The 3-5-year lifespan is satisfactory for consumer products but this is not acceptable for larger batteries [4]. As we know batteries produce electricity due to the electrochemical reaction; cold temperatures cause a slower reaction resulting in the low performance of the battery.

3.3 Supercapacitors

A capacitor stores energy by means of a static charge as opposed to the electrochemical reaction. A capacitor is charged applying differential voltages on its positive and negative plates. The supercapacitor, also called as ultracapacitor or double-layer capacitor, differs from a regular capacitor in that it has very high capacitance. The capacitance of a capacitor increases as the area of the plates increases and as the distance between the plates decreases. The supercapacitors get their much bigger capacitance from a combination of plates with a bigger, effective surface area and less distance between them. The supercapacitors are rated in Farads which is thousands of times higher than the electrolytic capacitor. The supercapacitor is used for storing energy undergoing frequent charge and discharge cycles at high current and short duration. The supercapacitors have a higher power density meaning that they can release energy more quickly than batteries but batteries are still king for storing large amounts of energy over long periods of time. Although the supercapacitors work at relatively low voltages (may be 2-3 volts), they can be connected in series to produce bigger voltages for use in more powerful equipment [5].

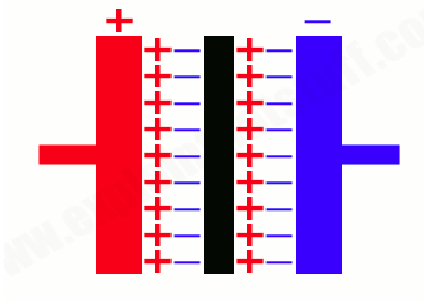

Figure 7. Supercapacitor or double-layer capacitor. Reprinted from [5]

Figure 7 shows a capacitor with two regular capacitors which is known as a double-layer capacitor or a supercapacitor.

3.4 Energy Harvesters

Energy harvesting is the process of capturing and storing the unharnessed energy derived from ambient sources like motion, vibration, heat, light and electromagnetic radiation and to reuse it to power low energy electronics. With the increasing demand of the IoT, energy consumption due to “things” has always been a topic for discussion. Even though the batteries provide nominal energy to power up the low-energy electronic devices, their replacement and recharging introduces a cost and convenience penalty. So as an alternative solution, energy harvesters are introduced providing environmentally friendly energy for free.

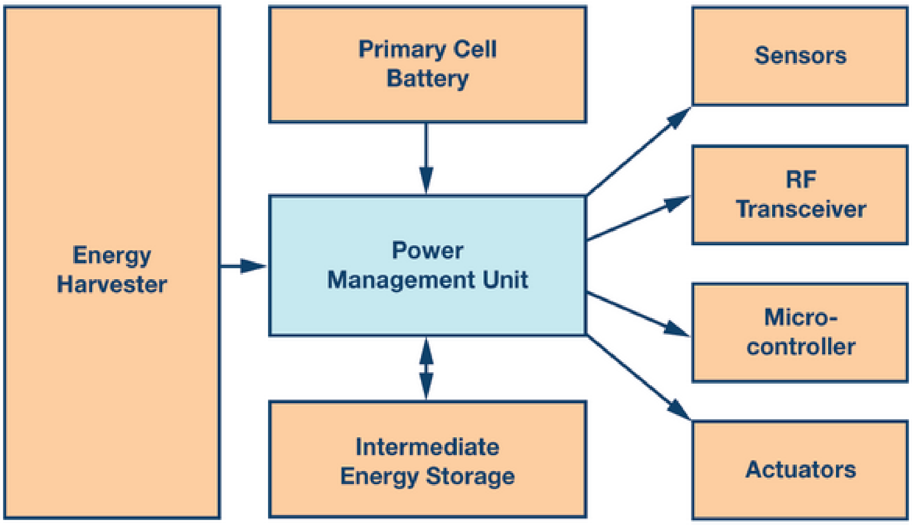

Figure 8. Energy harvesting system setup

An Energy Harvesting System (EHS) consists of Energy Harvesters (EH), Power Management Unit (PMU) and Energy Storage Device (ESD). The PMU charges ESD (typically a capacitor) with energy provided by EH which is later used to power other embedded devices (ED) like sensors, actuators, micro-controller, RF Transceiver. Thus, the ESD is also called as energy buffer in the system. Depending on the state of ED activity, EHS output power varies. When ED is active, the energy is consumed and the voltage from EHS starts to drop. When in a low-power state, the voltage from EHS rises as ESD is being charged faster instead of being depleted. The following graph in figure 9 shows the variation of the EHS output voltage with the ED activity over a period.

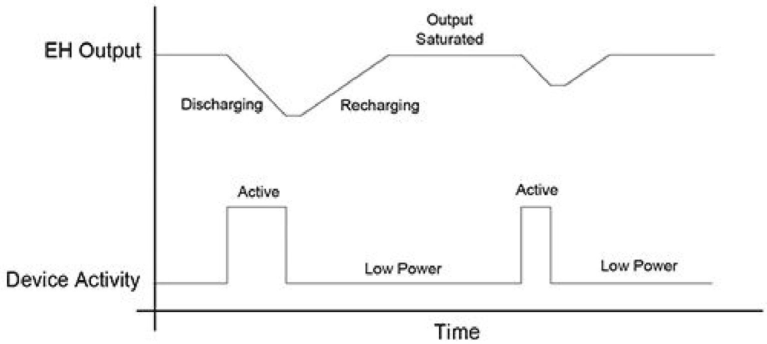

Figure 9. EH output variation with ED activity over a period. Reprinted from [9]

There are various methods of energy harvesting. Some of them are discussed below.

Piezoelectric Method

The piezoelectric effect generates energy in the form of electric signals in response to applied mechanical strain. Human motion, low-frequency vibrations and acoustic noise are some of the potential sources that could be harvested by piezoelectric materials. The power produced by the piezoelectric energy harvesters are the order of milliWatts which is enough for micro-scale devices [6]. The piezoelectric harvesters can produce a voltage ranging from 2V to 10V. The use of piezoelectric materials to harvest power is becoming popular day by day. Piezo elements are being embedded in walkways to capture the energy of people's footsteps and can also be embedded in shoes to capture the energy produced while walking.

Photovoltaic Method

The photovoltaics generate electrical energy from light energy and are commonly made of semiconductors. They may be solar cells, photo sensors or photo diodes. A tiny, inexpensive solar cell can generate 100 Watts of energy at noon time, so it can be relatively plentiful source of energy [6]. Photovoltaic energy conversion is a well-known integrated circuit compatible technology that offers relatively higher power output levels in comparison to other mechanisms. Nevertheless, its power output is strongly dependent on light intensity.

Thermoelectric Method

Temperature differences across the material generates heat flows from high temperatures to low temperatures and consequently results in electrical voltage. This process is known as the thermoelectric effect. Large thermal gradients are essential to produce practical voltage and power levels [6]. However, temperature differences greater than 10^{o}C are rare in micro systems, so such systems generate low voltage and power levels.

Electromagnetic Method

Electromagnetic harvesting system is a technique that uses a magnetic field into convert mechanical energy to electrical energy. This can be achieved using the principle of electromagnetic induction. Electromagnetic induction provides improved reliability and reduced mechanical damping as there would not be any mechanical contact with any parts and no voltage source required [6]. The induced voltage is very small (max. 0.1V) and therefore must be increased to a viable source of energy using a transformer with a more number of turns of the coil or an increased permanent magnetic field. However, these parameters are limited by the size constraints of the microchip as well as its material properties.

Pyroelectric Method

The pyroelectric effect is the phenomena where temperature changes are converted into electrical energy. When the temperature of a pyroelectric material changes, positive and negative charges will move to the opposite ends (polarized) creating an electrical potential. Pyroelectric energy harvesting applications require inputs with time variances, which results in small power outputs. One merit of pyroelectric energy harvesting over thermoelectric energy harvesting is that most of the pyroelectric elements are stable up to 1200^{o} C or more, which increases thermodynamic efficiency.

Electrostatic (Capacitive) Method

Electrostatic method is based on the variable capacitance of vibration-dependent varactors (variable capacitors). An initially charged varactor will separate its plates due to vibration and thus resulting in the transformation of mechanical energy into electrical energy. Electrostatic generators are the mechanical devices that produce electricity using manual power. The significant advantage of electrostatic generators is their ability to integrate with microelectronics while the significant disadvantage includes the need for an additional voltage source to charge the capacitor initially. [6.]

Tidal Method

Tidal (Wind) energy can be converted into electrical energy using a wind turbine to exploit the linear motion coming from wind. The performance of the large-scale wind turbines is highly efficient. However, small-scale wind energy harvesting is challenging due to the fluctuations in the wind strength and unpredicibility of the wind flow source. Also, small-scale wind turbines show inferior efficiency due to the relatively high viscous drag on the blades at low Reynold numbers.

Biochemical Method

The electrical energy can be produced with the help of electrochemical reactions between the biochemical substances. Biofuel cells as active enzymes and catalysts can be used to harvest the biochemical energy in biofluid into electrical energy. Human body fluid contains many biochemical substances that has potential of energy harvesting. Among them, glucose is the most common used fuel source. Theoretically, it releases 24 free electrons per molecule when oxidized into carbon dioxide and water. Even though biochemical energy harvesting can be superior to other harvesting techniques in terms of continuous power output and biocompatibility, its performance depends on the type and availability of fuel cells.

Energy Source	Condition	Power Density(uW/cm^3)
Solar	(Outdoors) (Indoors)	7500 100
Vibrations	1 m/s^2	60
Temperature gradient	ΔT = 5°C	100
RF	Unless near a transmitter	<1

Table 1. Energy harvesting Sources. Reprinted from [18]

Table 1 shows the general estimation of power output for various energy harvesting sources on different conditions. The solar source has been the most powerful source on a bright and shiny day.

4 Power Metrics in BLE

A BLE device achieves low power consumption by keeping radio activity short and allowing the device to reside in the standby mode most of the operating time. The power consumption of a BLE device cannot be compared to another using a single metric. Usually a device gets rated by its peak current but the fact is that a device running on BLE stack consumes current at peak level while a data is transmitted or received. Although the peak current has a vital role in the total power consumption, a BLE device is transmitting or receiving only for a small portion of the total time that the device is connected.

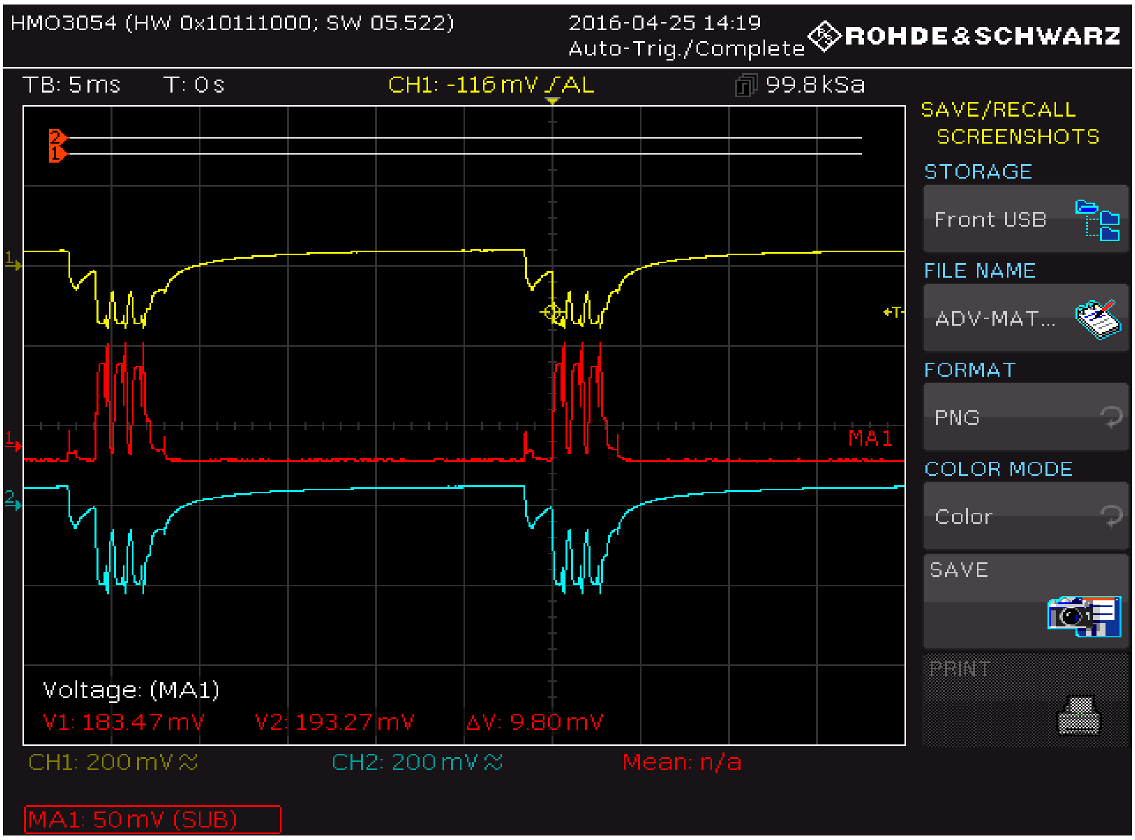

Figure 10. Voltage versus Time during BLE advertisement

Note: In the figure 10, the red graph is the difference between the other two graphs and it is used to measure the required voltage with respect to time.

In addition to transmitting or receiving, a BLE device will go through various states, such as standby, processing, idle etc. Even if a device's current consumption in each different state is known, this is still not enough information to estimate the average total power consumption. The different layers of the BLE stack require a certain amount of processing time to remain connected and comply with the protocol's specifications. The MCU takes time to perform this processing and some current is consumed to do so. Also, the device might go through the transition states which also consumes current. So, to get an accurate estimation, these states should be considered. [20.]

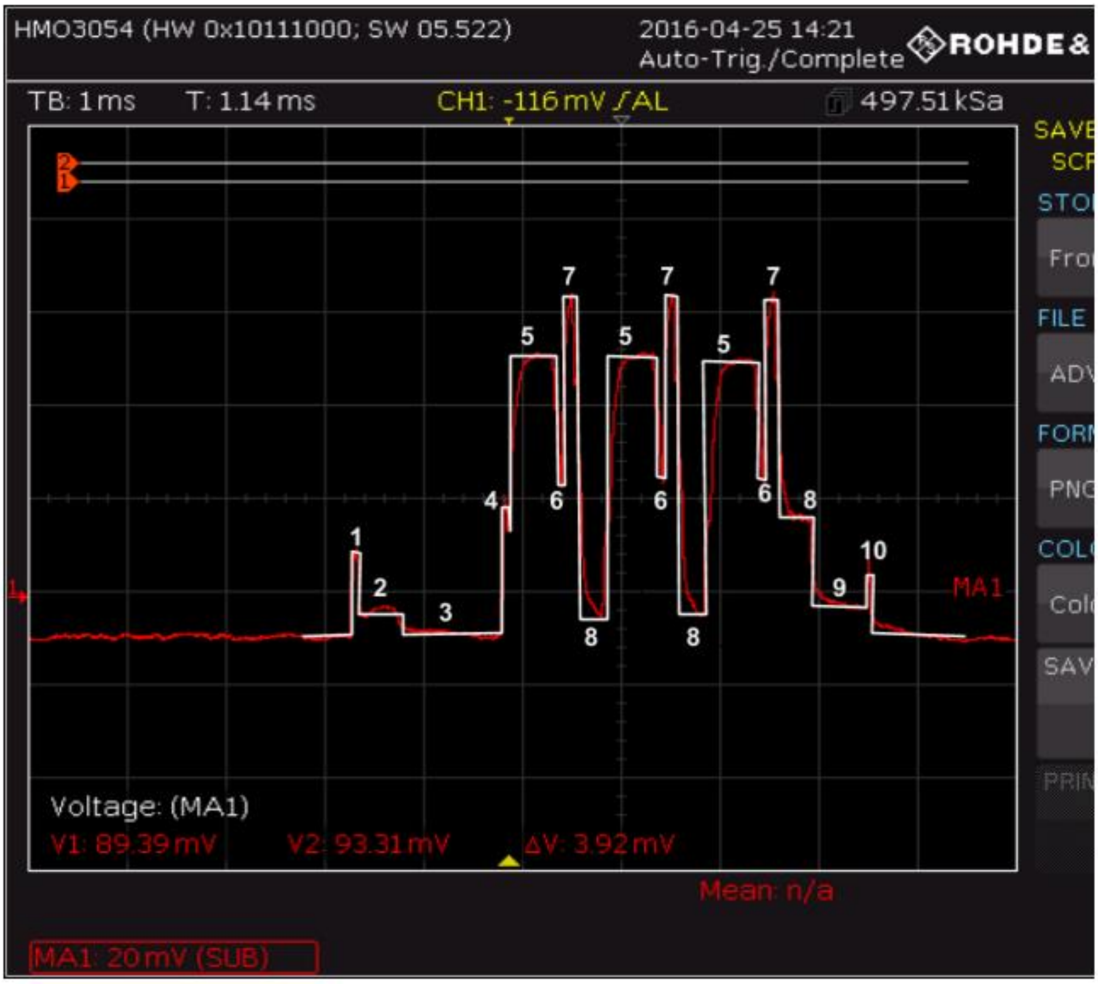

Figure 11. Voltage versus Time during a single advertising event

Figure 11 shows the various states that occurs in a single advertising event. The current consumption level at each state might be different as the BLE stack performs various task consuming different amounts of current.

4.1 Advertising Event-State Analysis

A short comparison on several states of an advertising event is provided below.

	State	Comments
1	Pre-processing	IC wake-up from sleep and radio setup
2	Standby+XOramp	No active clock management and Crystal oscillator ramped. This value depends on the type of crystal.
3	Standby	No clock management active. This value depends on the type of crystal.
4	Pre-to-Tx	Radio turns on for Tx and Rx
5	Tx	Radio transmits an advertisement packet. Time is dependent on the amount of transmitted data.
6	Tx-to-Rx	Tx to Rx transition
7	Rx	Radio listens for a packet from the master
8	Post-processing	BLE protocol stack processes the received packets and set up for idle timer for the next event.
9	Pre-Idle	Transition from post processing to Pre-idle-to-idle
10	Pre-idle-to-Idle	Transition for Pre-idle state to Idle state

Table 2. Advertising Event-State Analysis

Moreover, a device running the BLE stack will spend most of the time in the idle state between the advertising or connection events. In nRF51422, the power management system is highly flexible with functional blocks such as the CPU, radio transceiver and peripherals having separate power state control in addition to the global System ON and OFF modes. In System OFF mode, RAM can be retained and the device state can be changed to System ON through reset or a GPIO signal. In the System ON mode, all functional blocks will be independently in the IDLE or RUN mode depending on the needed functionality. The primary metrics is the average current for the advertising and connected mode. These values determine the life of the energy supplied. [20.]

5 Measurements

The nRF51422 Development Kit and nRF-Software Development Kit version 11 (nRF5_SDK_11.0.0_89a8197) was used for the current measurement. The current should be measured with respect to time to get average current consumption. Hence, an ampere-meter could be a solution for the measurement. But the ampere-meter does not provide enough information to see the behavior with respect to time so, an oscilloscope is required.

5.1 Test Setup

The oscilloscopes are voltage sensitive. So, a resistor in a series was required to measure the voltage drop across it. The voltage drop across the resistor was measured with the oscilloscope connecting its two channels at PIN 22 and using “MATH” function to find the difference between voltages measured by those two channels. At first, the board nRF51422 was prepared by cutting the shorting of the solder bridge SB9 and then a SMD resistor of value 6.6 Ω was mounted on the footprint for R6[13,13].

The 3V from a regulated DC power supply was supplied to the board to eliminate variables that might be caused by a defective or low battery.

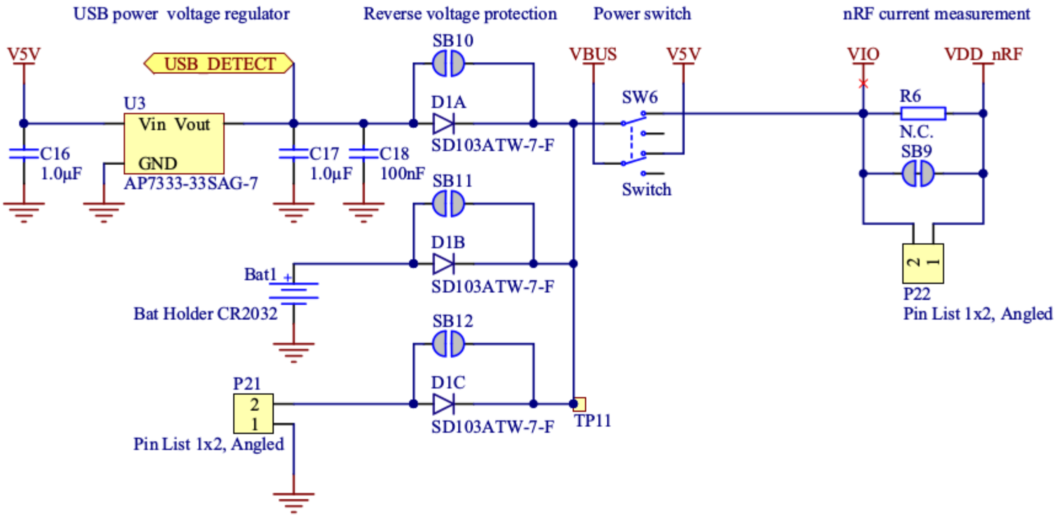

Figure 12. Power Supply Circuitry. Reprinted from [13]

In figure 12, the power supply circuitry for nRF51-pca10028 board is shown where R6 is the place for the resistor to be mounted for voltage measurement and SB9 is the solder bridge that needs to be cut [13;18].

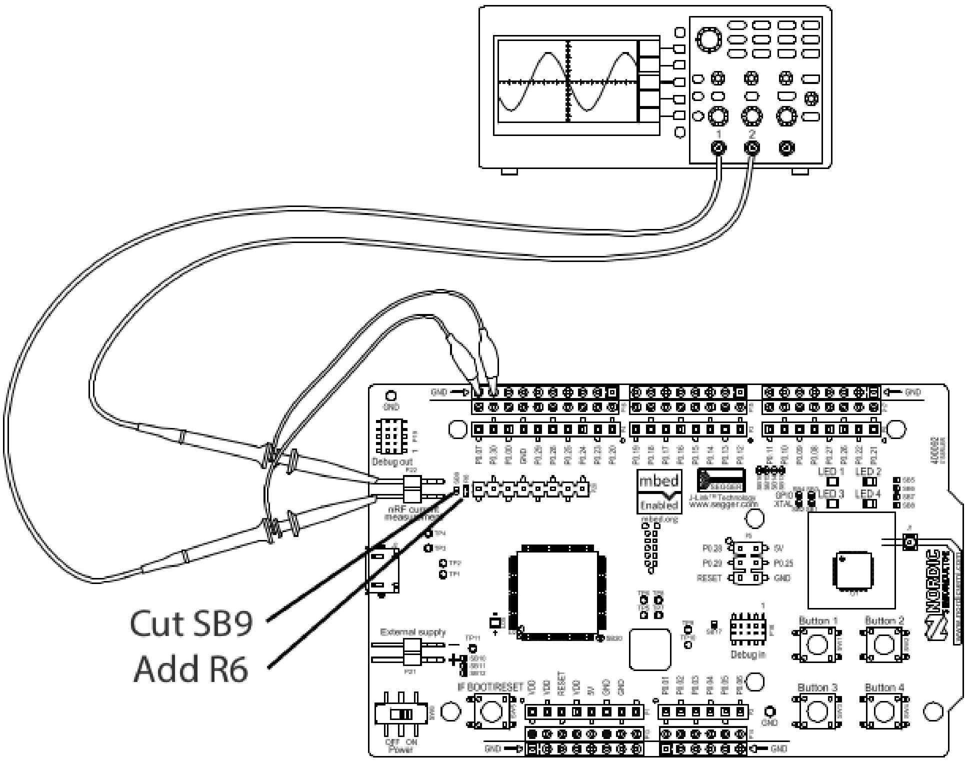

Figure 13. Current Measurement with Oscilloscope. Reprinted from [13]

Figure 13 shows the setup overview for the current measurement for the board (pca10028). Pin 22 on the board is connected to the oscilloscope using two different channels that measure voltages at both ends of the resistor and the difference of those voltages gives the voltage drop across the resistor.

5.2 Environment and Software Setup

The nRF51 is a System-On-Chip (SOC) with a cortex-M0 and a BLE radio chip all in one. The software that runs on nRF51 includes a softdevice and applications. The softdevice is a binary from Nordic Semiconductors that provides the BLE capabilities. The application calls the functions from the softdevice that helps to access BLE functions and other capabilities. To program the device, some software (toolchain) such as armgcc, nrfjprog are need to be installed and paths need to be edited in the “Makefile” that comes with nRF5 SDK (Software Development Kit).

Installation of ARM-GCC toolchain

- Downloaded linux installation tarball (gcc-arm-none-eabi-5_2-2015q4-20151219-linux.tar.bz2 from https://launchpad.net/gcc-arm-embedded/+download)
- Unpacked the archieve file in /usr/local directory
- Tested if the compiler is functional using the actual install path.

```
metropolia@Metropolia:   ~$/usr/local/gcc-arm-none-eabi-5_2-
2015q4/bin/arm-none-eabi-gcc --version
```

- Toolchain executables are 32-bit apps, so while running on 64-bits machines, 32-bit libraries were installed using command;

```
$ sudo apt-get -y install lib32z1 lib32ncurses5 lib32bz2-1.0
```

The command-line-tools, nrfjprog and mergehex, for Linux 64-bits were downloaded from Nordic semiconductors’ page (*https://www.nordicsemi.com/eng/Products/ANT/nRF51422*) and placed in /home folder.
The makefile is responsible for defining the compiler and linker. The Nordic SDK comes with makefiles where the paths and the version number of armgcc need to be edited.

The location for makefiles which need to be edited is given below.

```
metropolia@Metropolia:      ~/nRF5_SDK_11.0.0_89a8197/compo-
nents/toolchain/gcc$ ls

gcc_nrf51_common.ld    gcc_startup_nrf52.s    Makefile.posix
nrf51_common.ld  nrf51_xxab.ld  nrf52_common.ld  nrf5x_com-
mon.ld

gcc_startup_nrf51.s  Makefile.common       Makefile.windows
nrf51_xxaa.ld    nrf51_xxac.ld  nrf52_xxaa.ld
```

Here, Makefile.posix and Makefile.windows should be edited as follows [7].

Makefile.posix

```
GNU_INSTALL_ROOT := /usr/local/gcc-arm-none-eabi-5_2-2015q4

GNU_VERSION := 5.2.1

GNU_PREFIX := arm-none-eabi
```

Makefile.windows

```
feq ($(findstring 86, $(ProgramFiles)), )

        PROGFILES := C:/Program Files

else

        PROGFILES := C:/Program Files (x86)

endif

GNU_INSTALL_ROOT := $(PROGFILES)/GNU Tools ARM Embedded/5.2
2015q1

GNU_VERSION := 5.2.1

GNU_PREFIX := arm-none-eabi
```

For the measurement purpose, “ble_app_pwr_profiling” example that is in *nRF5_SDK/examples/ble_peripheral/* was used. After compiling the makefile, the softdevice and board need to be flashed. The command-line tool “nrfjprog” helps in flashing and resetting pin. So, the path for nrfjprog in the makefile that is inside the folder `nRF5_SDK_11.0.0_89a8197/examples/ble_peripheral/ble_app_pwr_profiling/pca10028/s130/armgcc` should be edited.

The following codes can be found at the end of the makefile. The part of the code was added in my case as the nrfjprog folder was in the /home directory.

```
flash: nrf51422_xxac_s130
@echo Flashing: $(OUTPUT_BINARY_DIRECTORY)/$<.hex
~/nrfjprog/nrfjprog     --program     $(OUTPUT_BINARY_DIREC-
TORY)/$<.hex     -f nrf51  --sectorerase
~/nrfjprog/nrfjprog --reset -f nrf51


## Flash softdevice
flash_softdevice:
@echo Flashing: s130_nrf51_2.0.0_softdevice.hex
~/nrfjprog/nrfjprog    --program    ../../../../../../compo-
nents/softdevice/s130/hex/s130_nrf51_2.0.0_softdevice.hex -f
nrf51 --chiperase
~/nrfjprog/nrfjprog --reset -f nrf51
```

The application “ble_app_pwr_profiling” is an example application that is customized for measurement of minimum power consumption. There are several parameters in the application which can be tuned by the user to see its effect on current consumption. They are discussed below.

4.2.1 Clock source

The softdevice needs a low-frequency clock for protocol timing. There are three choices for a low frequency clock source: external crystal, internal RC oscillator and synthesized clock. Among these, the external 32.768 kHz crystal clock source is the best option to get the lowest current consumption when the softdevice used and appropriate accuracy should be chosen so that the softdevice can take the value into consideration. The board (pca10028) that I used for measurement, has the crystal accuracy of ±20 ppm installed so it should be used for the measurement and cannot be changed.

The crystal accuracy can be set with the following commands.

With the softdevice: `NRF_CLOCK_LFCLKSRC-XTAL_x_PPM` where “x” is the accuracy of the crystal.

Without the softdevice, the external 32 kHz crystal is done with the following code:

```
    NRF_CLOCK->LFCLKSRC      =      (CLOCK_LFCLKSRC_SRC_Xtal      <<
CLOCK_LFCLKSRC_SRC_Pos);
    NRF_CLOCK->EVENTS_LFCLKSTARTED = 0;
    NRF_CLOCK->TASKS_LFCLKSTART = 1;

    // Wait for the low frequency clock to start
    while (NRF_CLOCK->EVENTS_LFCLKSTARTED == 0) {}
    NRF_CLOCK->EVENTS_LFCLKSTARTED = 0;
```

The use of a synthesized clock is not helpful in terms of current measurement as it requires 16 MHz clock source to be constantly enabled. The internal 32kHz RC oscillator can also be used to save space on PCB but it needs to be calibrated every 4 seconds to maintain accuracy within 250 ppm. It consumes about 10 uA more current in addition to a 20 ppm crystal.

4.2.2 Low Power Mode with BLE Softdevice

There are two functions which call the softdevice to enter a certain power mode. They are: `sd_app_event_wait()` which enables System on low power mode to keep current consumption minimum and `sd_power_system_off()` which keeps device in System off mode. Those functions are included in the main function of the application. The softdevice uses RTC0 for keeping the information and to know when to wake up for the next BLE connection event. When the RTC0 wakes the chip up, the softdevice will carry out the tasks assigned for the BLE connection event that is receiving and sending packages.

4.2.3 Advertising Interval

When the advertising interval is large, the device will consume less current during advertising. It is more convenient to advertise periodically instead of advertising continuously, which also saves current. These parameters can be changed using the following constants.

Advertising Interval:

```
#define APP_ADV_INTERVAL                      40  /**< The advertising
interval (in units of 0.625 ms. This value corresponds to 25 ms).
*/
```

Advertising periodically:

```
#define APP_ADV_TIMEOUT_IN_SECONDS           180   /**< The adver-
tising timeout in units of seconds. */
```

The advertising interval can be in the range of 20 ms and 10.24 s.

4.2.4 Connection Interval

When the connection interval is large, there will be longer packet delay resulting in slow communication. This will consume less current in comparison to a situation with a short connection interval. The connection interval can be tuned using the following definition in the application.

```
#define MIN_CONN_INTERVAL                              MSEC_TO_UNITS(500,
UNIT_1_25_MS)
#define MAX_CONN_INTERVAL                             MSEC_TO_UNITS(1000,
UNIT_1_25_MS)
```

The above parameters set the connection interval at 500-1000 ms but it depends on the central device to set the connection interval.

The connection interval can be in the range of 7.5 ms and 4.0 s.

4.2.5 Slave Latency

The slave latency is the number of connect events that a slave can skip. This is done to save power on the slave side. It can skip some connection event if there is no data to save power but the sleeping period should not be too long so that the connection will timeout. It is adjusted by modifying the constant given below.

```
#define SLAVE_LATENCY                                              0
```

It can be any value between 0 and 499 but it cannot exceed:

```
((Supervision Timeout/Connection Interval) -1)
```

4.2.6 Supervision Timeout

Supervision timeout is the timeout from the last data exchange till a link is considered lost. If this time passes without a successful connection event, the device will terminate the connection and return to an unconnected state. It is represented in unit of 10 ms. It can be in the range of 100 ms to 32 s. However, it must be larger than:

```
((1+ Slave Latency) * Connection Interval)
```

A central device will not start trying to reconnect before the timeout has passed. So, if a device goes in and out of range often, longer timeout fixes the problem.

4.2.7 Transmitting Power

The radio transmit power also affects the current consumption. The higher the value, the higher the current consumption. It can be set with softdevice using the function `sd_ble_gap_tx_power_set ()` after `ble_stack_init()` in `main()`. The accepted values for nRF51 are -40, -30, -20, -16, -12, -8, -4, 0, and 4 dBm.

4.2.8 SPI, UART, TWI Interfaces

While using SPI, UART and TWI services, it is more energy-efficient to choose the maximum transmission speed and enable the services for a minimal time. A peripheral can be disabled by assigning a disabled value to the peripheral's enable register so that the services do not keep the clock running when they are not needed.

To enable the UART for both Rx and Tx, the following code should be executed.

```
NRF_UART0->ENABLE = 1;
NRF_UART0->TASKS_STARTTX = 1;
NRF_UART0->TASKS_STARTRX = 1;
```

To disable the UART for both Rx and Tx, the following code should be executed.

```
NRF_UART0->TASKS_STOPTX = 1;
NRF_UART0->TASKS_STOPRX = 1;
NRF_UART0->ENABLE = 0;
```

To enable the SPI peripheral 0, the following code should be executed.

```
NRF_SPI0->ENABLE = 1;
```

To disable the SPI peripheral 0, the following code should be executed.

```
NRF_SPI0->ENABLE = 0;
```

5.3 Formulas

After noting down the time and voltage drop across the resistor for each stage from the oscilloscope using marker, we can calculate the average voltage drop across the resistor during a single connection event using the following formula:

Average voltage drop during a single advertising/connection event (V_{AVG})=

$$\frac{\Sigma(State\ 'X'\ Time)*(State\ 'X'\ Voltage)}{\Sigma(States\ Time)}$$

Then the average current is calculated dividing the average voltage drop by the value of the resistor used as follows.

Average Current drawn during a single advertising/connection event (I_{EVENT}) =

$$\frac{Average\ voltage\ drop\ during\ a\ single\ advertising\ or\ connection\ event(V_{AVG})}{Resistor\ Value(R)}$$

The advertising/connection interval is made up of two blocks: the advertising/connection event and idle state as shown in figure 14. So, the average current during the advertisement or connection state can be defined as the sum of the average current for an advertising or connection event and the average current for idle state over the given advertising or connection interval.

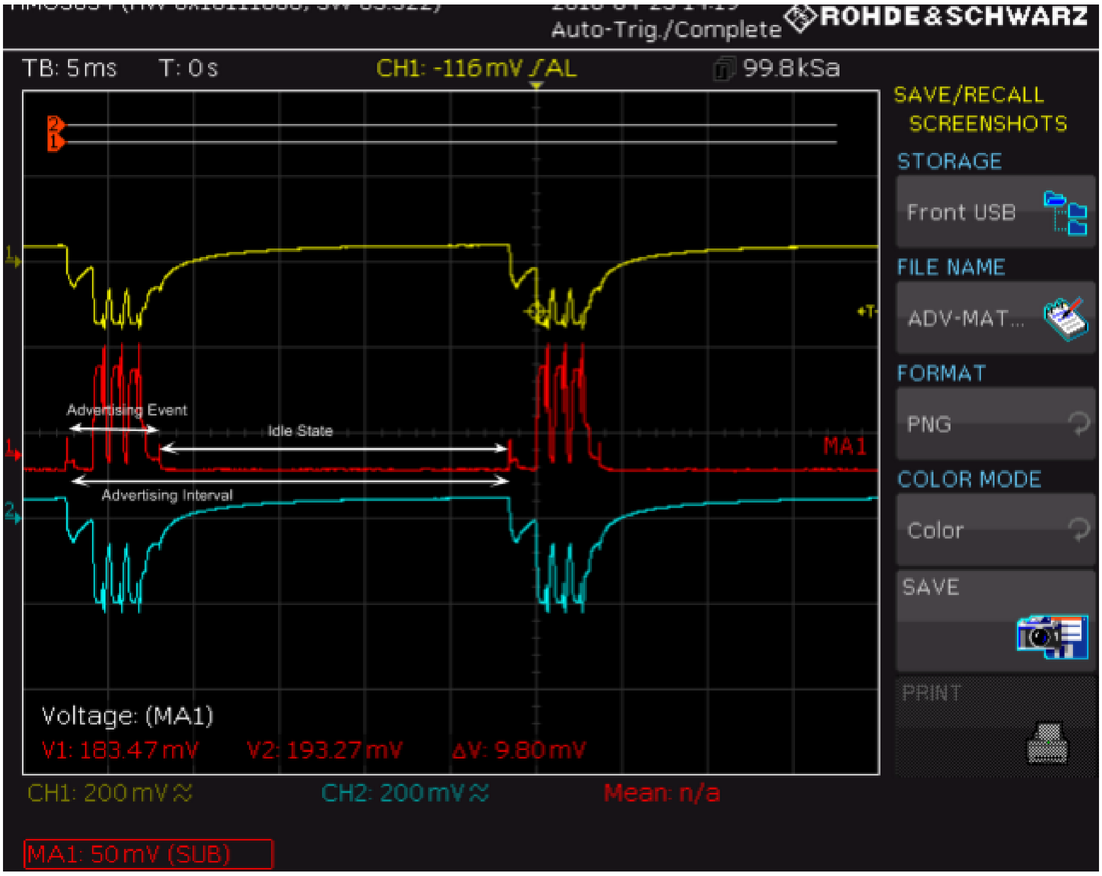

Figure 14. Advertising interval = Advertising event + Idle state

The idle current (I_{IDLE}) for the idle state is usually very small (one or a few micro-Amperes) which is not possible to be measured using the oscilloscope. Thus, a multimeter with micro-Ampere precision is necessary to find out the idle current. Then, the average current during the advertisement/ connection is calculated as below.

Average current during advertisement/connection (I_{AVG}) =

$$\frac{[(Advertisement\ or\ Connection\ Interval) - \Sigma(States\ Time)] * I_{IDLE} + \Sigma(States\ Time) * I_{EVENT}}{Advertisement\ or\ Connection\ Interval}$$

A capacity of the battery is determined in terms of electrical charge that it holds or the amount of current that it can deliver in one hour. It is measured in mAh. Hence, the life of a battery can be estimated as its capacity over the current supplied.

Here, the average current during advertisement/connection (I_{AVG}) determines the life of the battery that is used as a power source.

$$\text{Estimated life of battery (in hours)} = \frac{Capacity\ of\ Battery(mAh)}{Current\ supplied(mA)}$$

Let us see an example demonstrating the formulas above.

There are several states that an advertising event or a connection event goes through and we can measure the corresponding voltages and time for each stage using markers in the oscilloscope.

6 Calculations, Results and Analysis

6.1 Use Case 1

After providing some value to the parameters and measuring the current consumption of the BLE, time differences and voltage differences were noted in the spreadsheet for the calculation. A screenshot of the spreadsheet is shown in figure 15.

Battery capacity (mAh):	230				230			
Connection Interval (ms):	20				20			
Advertising interval (ms):	20				20			
Slave Latency:	0				0			
TX-power (dBm):	0				0			
Payload Size (bytes):	31				31			
Resistor Value(Ohms):	6.6				6.6			
Idle current (uA):	2.6				2.6			
32 KHz clock crystal accuracy (ppm):	20				20			
	Advertisement				Connection			
	Time (us)	Voltage (mV)	Percent of events		Time (us)	Voltage (mV)	Percent of events	
			100				100	
State 1 (Pre-processing)	83.5	18		1503	127.5	18.5		2358.75
State 2 (Standby + XO ramp)	380	4.5		1710	347	5.75		1995.25
State 3 (Standby)	1060	0.9		954	1060	0.9		954
State 4a (Pre-to-Tx)	47	28.9		1358.3	0	0		0
State 4b (Pre-to-Rx)	0	0		0	83.5	26.5		2212.75
State 5 (Tx)	520	60.5		31460	171	59		10089
State 6 (Tx-to-Rx)	50	33		1650	0	0		0
State 6b (Rx-to-Tx)	0	0		0	64.5	53.5		3450.75
State 7 (Rx)	177	72.5		12832.5	175.5	66		11583
State 8 (Post-processing)	238	3.3		785.4	419	23.5		9846.5
State 9 (Tx)	555	60.5		33577.5	0	0		0
State 10 (Tx-to-Rx)	50	33		1650	0	0		0
State 11 (Rx)	191	72.9		13923.9	0	0		0
State 12 (Post-processing)	225	2.5		562.5	0	0		0
State 13 (Tx)	550	58.5		32175	0	0		0
State 14 (Tx-to-Rx)	50	33.7		1685	0	0		0
State 15 (Rx)	147	73.7		10833.9	0	0		0
State 16 (Post-processing)	319	24.5			0	0		0
State 17 (Pre-idle)	894	4.5		4023	0	0		0
State 18 (Pre-idle-to-Idle)	36	14.5		522	0	0		0
State 19	0	0		0	0	0		0
State 20	0	0		0	0	0		0
State 21	0	0		0	0	0		0
State 22	0	0		0	0	0		0
State 23	0	0		0	0	0		0
State 24	0	0		0	0	0		0
State 25	0	0		0	0	0		0
State 26	0	0		0	0	0		0
State 27	0	0		0	0	0		0
State 28	0	0		0	0	0		0
State 29	0	0		0	0	0		0
State 30	0	0		0	0	0		0
				151206				42490
Total time of advertising/connection event	5572.5				2448			
Average Current draw during advertising/connection event (mA):		4.1112607				2.629852		
Average current draw during advertisement/connection (uA):	1147.38				324.18			
Expected battery life (hours):	200.46				709.49			
Expected battery life (days):	8.35				29.56			

Figure 15. Screenshot of sample measurement from Spreadsheet

In the figure 15, there are more states for the advertising event than in the connection packet. The BLE smart uses 40 RF (Radio Frequency) channels in the ISM band (2.4 GHz). Three of them are dedicated to the advertising channel which is channel 37(2402 MHz), 38(2426 MHz) and 39(2480 MHz). They were selected to avoid interference with the busy channels used by WiFi. Each of these channels has their own Rx and Tx state. For the connection event, there might occur several states depending on the number of packets transmitted and received, even though just an empty connection packet is shown in the sample in figure 15.

There are also columns named percentage of event which is assumed as 100 but there might be a case where the advertising event/connection event does not happen 100%. Considering the sample in figure 15 as an example, it is given that the current consumed for 100% of advertising event is 1147.38 uA and for the connection event 324.18 uA. Let us assume that the advertising is done for 10 minutes and then connected for 50 minutes. So, the total time of the events will be 60 minutes or an hour of which 1/6 was spent on advertising and 5/6 on connection.

So, the total average current consumption is calculated as follows:

= (1/6) * 1147.38 uA + (5/6) * 324.18 uA

= 191.23 uA + 270.15 uA

= 461.38 uA

The calculation above shows, under the given condition, the current consumed by the system is 461.38 uA in an hour. So, if the battery capacity is 230 mAh, the system will run for (230 mAh/ 461 uA), which is about 499 hours.

Then the average power consumption is calculated as the product of average current consumption and voltage supplied (mathematically, P = I*V). Considering the total average current consumption calculated above and the voltage supplied (V_{DD}) as 3V, the total power consumption in an hour will be (461.38 uA * 3V), which gives 1.38414 mW.

6.2 Use Case 2

Another simple use case was carried out. The default ble_app_hts application example available on *nRF5_SDK_11.0.0_89a8197/examples/ble_peripheral/* was modified for the measurement. Basically, the default ble_app_hts application updates the simulated value of battery level every tick and die temperature of the chip on the board (using temperature sensor that is on the board already) to the master device only when Button 1 is pressed. So, the application is modified to update the die temperature in every tick.

After the modification, the current measurement was carried out for the test setup as before, and the parameters set in the code are given below:

Advertising interval= 25 ms

Connection interval= Min (500ms), Max (1000ms)

Tx Power = 0 dBm

Slave Latency= 0

Resistor = 6.6 Ohm

Using the oscilloscope, the current consumption during the advertising event was found to be 869.81 uA. The current consumption during the connection event was found to be 31.23 uA (at min. connection interval = 500 ms) and 16.92 uA (at max. connection interval = 1000 ms). Here, in this case, the average value of the values found for the minimum and maximum connection interval was considered to calculate the total current consumption. So, the average current consumption during the connection event is (31.23 uA + 16.92 uA)/2 = 24.075 uA.

Suppose the application is advertising for 1 minute and then connected for the next 59 minutes. So, to run the application for an hour (as the situation stated), the current consumption is estimated as follows:

The total current consumption in an hour = (869.81 uA * 1/60h) + (24.075 uA * 59/60h)

= 14.496 uAh + 23.674 uAh

= 38.17 uAh

The power consumption is calculated as a product of current and voltage. If 3V is the supplied voltage in the above scenario, then the power consumed is calculated as 3V * 38.17 uAh =114.51 uWh.

Converting Watt-hours into Joule,

114.51 uWh = 114.51 uWh * 3600s/h = 0.41223 uJoule

The size of a solar panel can be estimated using the formula:

$E_{electrical} = E_{solar} * A * R * PV_{efficiency}$

Adapted from [15]

Where,

$E_{electrical}$ = Energy required in electrical form

E_{solar} = Energy available from the sun

A = Area of solar panel

R = Yield of solar panel given by ratio: electrical power (in kWh) of one solar panel divided by the area of one panel

$PV_{efficiency}$ = PV cell's ability to convert light energy into electrical energy

For the system mentioned above,

$E_{electrical}$ = 0.41223 Joule

E_{solar} = 5.69kWh *3600s/h = 20.484 MegaJoules/m^2 (typical June in Vantaa,Finland) Adapted from [14]

R = 15% (assuming)

$PV_{efficiency}$ = 3% for Thin-Film cells and 15% for Crystalline Silicon cells

The required area of the Crystalline Silicon solar cell is given by

$$A = E_{electrical} / (E_{solar} * R * PV_{efficiency})$$

$$= 0.41223 / (20484000 * 0.15 * 0.15)$$

$$= 8.944 * 10^{-7} \ m^2$$

Similarly, the required area of Thin-Film solar cell is given by

$$A = E_{electrical} / (E_{solar} * R * PV_{efficiency})$$

$$= 0.41223 / (20484000 * 0.15 * 0.03)$$

$$= 4.472 * 10^{-6} \ m^2$$

The calculations above show the Crystalline Silicon solar cells have high efficiency and require a smaller size than thin-film solar cells but are more expensive.

7 Conclusion

The goal of the project was to see the efficiency of the harvested energy in the BLE communication in the field of the Internet of Things. So, the power consumption of the BLE was measured. The BLE emerges as a strong low-power wireless technology and with low-complexity peripherals, which contributes to connect many BLE devices to the Internet of Things with a high data transmission rate. The measurements done in the project show the BLE consumes extremely low power and hence, it is named Bluetooth low energy. After analyzing the energy harvesters' power output, it can be concluded that the BLE communication can be done easily for few days even with a very small amount of energy harvested.

Following the approach in this study, BLE system designers can estimate the current consumption with various parameters, choose the energy harvesting system as needed and can design the system accordingly. In the BLE, there exists a trade-off between power consumption, latency and throughput that mainly depends on the connection interval and slave latency parameters. Turning down the BLE to sleep when not needed also helps to save some power but this will create a real-time monitoring issue. Also, it is important to note that BLE communication can be done only within a limited distance.

Overall the BLE, being a strongest wireless technology, has many use cases in recent years and can be operated with a very small amount of energy harvested by small energy harvesters like solar cells on a calculator and a mechanical switch. The use of harvested energies also provides benefits of being maintenance-free and environmentally friendly as energy harvesters scavenge the wasted energy.

References

[1] Taneja, M. (2014). *A Frame for Power Saving in IoT Networks.* Available at: http://ieeexplore.ieee.org/xpls/icp.jsp?arnumber=6968211#article. Accessed 28 March 2016.

[2] Bluetooth SIG (2016). *Bluetooth Low Energy.* Available at: https://www.bluetooth.com/what-is-bluetooth-technology/bluetooth-technology-basics/low-energy. Accessed 1 April 2016.

[3] M.M.El Emary, I. and Ramakrishnan, S. (2013). *Wireless Sensor Networks: From Theory to Applications.* CRC Press, pp. 369-375.

[4] Cadex Electronics Inc. (2016). *Battery University [online].* Available at: http://batteryuniversity.com/learn/article/comparing_the_battery_with_other_power_sources. Accessed 3 April 2016.

[5] Woodford, C. (2015). *Supercapacitors [online]*. Available at: http://www.explainthatstuff.com/how-supercapacitors-work.html. Accessed 3 April 2016.

[6] M.Conrad, J. and Chalasani, S. (2008). *A survey of Energy Harvesting Sources for Embedded Systems.* Available at: http://ziyang.eecs.umich.edu/~dickrp/ides/papers/chalasini-harvest.pdf. Accessed 8 April 2016.

[7] Fagerness, T. (2015). *BLE using nRF51: ARM-GCC Build Environment [online]*. Available at: http://www.allaboutcircuits.com/projects/ble-using-nrf51-arm-gcc-build-environment/. Accessed 22 April 2016.

[8] Townsend,K. (2014). *Introduction to Bluetooth Low Energy [online]*. Available at: https://learn.adafruit.com/introduction-to-bluetooth-low-energy/. Accessed 1 April 2016.

[9] Kumar, R. (2015). *Energy Harvesting and Bluetooth low Energy: Designing a Batteryless Beacon [online].* Available at: http://www.digikey.com/en/articles/techzone/2015/nov/energy-harvesting-and-bluetooth-low-energy-designing-a-batteryless-beacon. Accessed 5 April 2016.

[10] ibeaconinsider (2015). *What is iBeacon? A guide to Beacons [online].* Available at: http://www.ibeacon.com/what-is-ibeacon-a-guide-to-beacons/. Accessed 5 April 2016.

[11] Basagni, S., Yousuf Naderi, M., Petrioli, C. and Spenza, D. *Wireless Sensor Networks with Energy Harvesting [online].* pp. 3,4,12. Available at:

http://www1.ece.neu.edu/~naderi/papers/EHWSN-Book-Chapter.pdf. Accessed 6 April 2016.

[12] Shen, W. and Larmo, A. (2015). *Bluetooth Smart Mesh-why does it make sense for IoT? [online].* Available at: http://www.ericsson.com/research-blog/internet-of-things/bluetooth-smart-mesh-make-sense-iot/. Accessed 18 May 2016.

[13] Nordic Semiconductors. *nRF51 Development Kit (User Guide v1.0).* Available at: http://infocenter.nordicsemi.com/pdf/nRF51_DK_UG_v1.0.pdf. Accessed 15 April 2016.

[14] European Commission-Joint Research Center (JRC). *Photovoltaic Geographical Information System-Interactive Maps [online].* Available at: http://re.jrc.ec.europa.eu/pvgis/apps4/pvest.php?lang=en&map=europe. Accessed 25 May 2016.

[15] RF Monolithics Inc. (2010). *Sizing Solar Energy Harvesters for Wireless Sensor Networks (Application Note M1002).* Available at: http://wireless.murata.com/media/products/apnotes/anm1002.pdf. Accessed 25 May 2016.

[16] Bluetooth SIG (2016). *Bluetooth Core Specification [online].* Available at: https://www.bluetooth.com/specifications/bluetooth-core-specification. Accessed 3 April 2016.

[17] Yoshida, J. (2015). *Mesh Comes to Bluetooth [online].* Available at: http://www.eetimes.com/document.asp?doc_id=1325815. Accessed 18 May 2016.

[18] Digi-Key (2016). *Tips to Getting the Most Out of Energy Harvesting Sources [online].* Available at: http://www.digikey.com/en/articles/techzone/2016/apr/tips-to-getting-the-most-out-of-energy-harvesting-sources. Accessed 22 May 2016.

[19] Estimote, Inc. (2016). *What is ibeacon? [online].* Available at: http://developer.estimote.com/ibeacon/. Accessed 6 April 2016.

[20] Nordic Semiconductors (2016). *Current measurement guide: Introduction [online].* Available at: https://devzone.nordicsemi.com/tutorials/27/. Accessed 17 April 2016.