Mark Arrumm

ZigBee Sensor Network: A Study on Methods of Measuring Energy Consumption

Metropolia University of Applied Sciences

Bachelor of Engineering

Electronics

Thesis

23 September 2018

Author Title Number of Pages Date	Mark Arrumm ZigBee Sensor Network: A Study on Methods of Measuring Energy Consumption 32 pages + 1 appendix 23 September 2018
Degree	Bachelor of Engineering
Degree Programme	Electronics
Instructor(s)	Anssi Ikonen, Senior Lecturer

The main aim of this project was to study the different methods of measuring energy consumption in embedded systems and energy harvesting methods. The objective was to use one of the methods to carry out tests and create the groundwork for further investigations and possibly optimization of energy consumption of ZigBee end nodes.

The measurements performed monitored the drained current from a ZigBee end node when configured to communicate with a ZigBee coordinator in Application Programming Interface (API) mode using the XCTU application to generate the API frames.

The results obtained from the measurements showed noticeable current variations in different communication phases from idle mode to receiving a data packet. Optimizing the energy consumption was beyond the scope of this project.

This study creates a good foundation for further research into optimizing the energy consumption of ZigBee end nodes in a ZigBee wireless network. The information can be useful for design engineers setting up mesh networks as majority of the work existing is based on theoretical studies and simulations.

Keywords	Wireless sensor network, energy harvesting, ZigBee, API, mesh network

Metropolia

Contents

Metropolia

List of Figures

List of Tables

List of Abbreviations

AC – Alternating Current

ADC – Analog-to-Digital Converter

API – Application Programming Interface

APS – Application Support

CSMA/CA – Carrier Sense Multiple Access with Collision Avoidance

DAQ – Digital Acquisition

DC – Direct Current

DSO – Digital Storage Oscilloscope

DUT – Device Under Test

FET – Field Effect Transistor

FFD – Full Function Device

GPIB – General Purpose Interface Bus

GTS – Guaranteed Time Slot

IEEE – The Institute of Electrical and Electronics Engineers

ISM – Industrial Scientific and Medical

LR-WPAN – Low Rate Wireless Personal Area Networks

MAC – Medium Access Control

NWK – Network Layer

OSI – Open Systems Interconnection

PAN – Personal Area Network

PCB – Printed Circuit Board

PHR – Physical Header

PHY – Physical Layer

RFD – Reduced Function Device

SHR – Synchronisation Header

UART – Universal Asynchronous Receiver Transmitter

WPAN – Wireless Personal Area Network

ZDO – ZigBee Device Object

1 Introduction

The last decade has seen great development in the field of wireless sensor technology. A notable entry to this sphere was the ZigBee, that came about shortly after the formation of the ZigBee Alliance in 2002. Since its inception, the way we live, work and play has been transformed and one can only imagine what the next decade promises. ZigBee is a set of protocols based on the physical and Medium Access Control (MAC) layers of the IEEE 802.15.4 standard for wireless communication. It is primarily designed for low-rate, low-cost and low power communications with short to medium range. Its application in smart homes, healthcare, manufacturing industries and commercial buildings have proven to be more economical alternatives to other low-rate wireless personal area networks (LR-WPANs), such as Bluetooth.

The objective of this study was to measure the energy consumption of a ZigBee end node and optimize the energy consumption. An XBee ZigBee Mesh Kit by DIGI was used as the hardware of choice and the XCTU application to generate and transmit messages as they would autonomously in a mesh network.

This thesis covers different methods of measuring energy consumption as well as the ZigBee technology. Energy harvesting is also presented in the theoretical background section as this is a developing area and in relation to optimizing energy, can be considered as a means to extending battery life.

Furthermore, the procedure, set up and configurations used for the measurements are described in more detail and the results, as well as challenges faced are discussed. Finally, the conclusion of this project and recommendations for future work are presented.

2 Theoretical Background

2.1 Methods of Measuring Energy Consumption in Embedded Systems

Most embedded systems are powered by DC power sources and can be run in different operating modes, such as sleep or stand by to conserve the energy. It is necessary to monitor the consumption of energy in embedded systems because of this finite energy from the DC source, cost of replacing batteries over time and the cooling limitation of the device.

Electrical energy depends on three factors: voltage, current and time. This can be expressed as:

$$E = V \times I \times t \qquad (1)$$

However, to get a precise measurement of energy, the following formula would be considered:

$$E = \int_{t_1}^{t_2} i_s(t) \cdot u_s\,(t) \cdot dt \qquad (2)$$

Where $i_s(t)$ is the current drawn from the primary source and $u_s(t)$ the instantaneous voltage between the poles.

Quite often in embedded systems the supply voltage is stable and any ripples that occur are usually very small, lying in the millivolt range. This is the reason some designs incorporate capacitors close to the terminal of the embedded processor to keep the voltage constant.

Some of the ways in which the energy of an embedded system can be measured include: current shunt method, current mirror method, charge transfer method, current probe method, battery status monitoring method and thermal monitoring method.

2.1.1 Current Shunt Method

This is the most common method of measuring power consumption and it basically involves measuring the potential difference across a shunt resistor placed in the high line (closer to the supply) or in the low line (closer to ground). Typical values of shunt resistor used for this are in the range of tens to hundreds mΩ and precision not worse than 1% [1].

The voltage across the shunt resistor can be measured using either a digital multimeter, using DAQ (digital acquisition) devices or using an oscilloscope. Figure 1 below illustrates high line and low line measurements using a DAQ device:

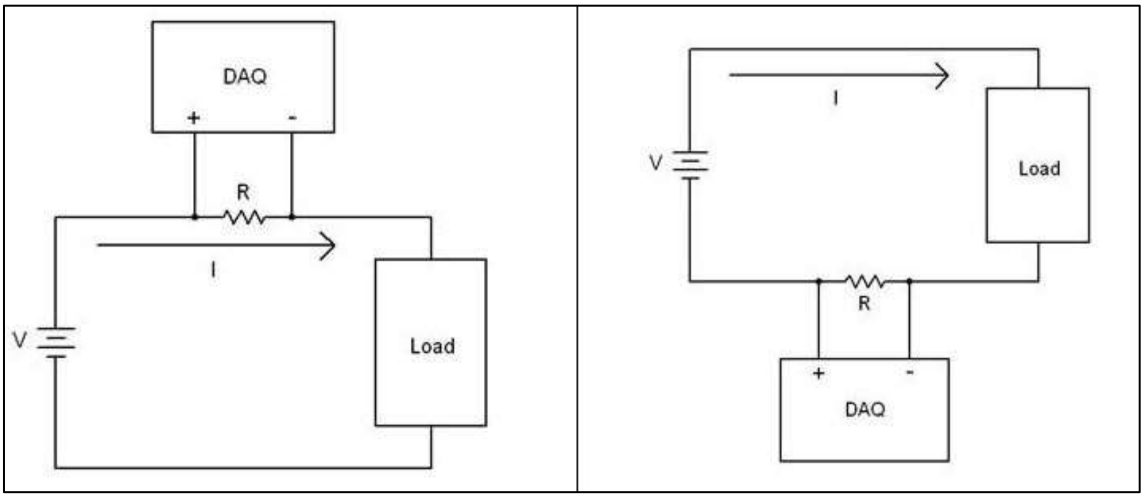

Figure 1. High Side and Low Side Current Measurement [2]

The current shunt method is simple to implement and can be used where the current range is very low as is the case in ultra-low power microcontroller units as well as where the current range is high for example in Bluetooth devices.

This method, however, is not cycle-accurate. In a study by Wolf et. al, this current shunt method was employed and improved to counter this by using a 0.1Ω shunt resistor, differential and operational amplifiers, a power integrator and an 8-bit ADC. In their source code they specified trigger points and executed it in real time recording the measurements on a logic analyser. These measurements could be read offline and could as well be back annotated to when an instruction was executed. [12.]

The use of a precision shunt resistor allows for higher clock frequencies to be reached depending on the choice of active components used. This approach would prove to be expensive if very high frequencies are to be achieved as the active components would be costly.

2.1.2 Current Mirror Method

A current mirror generally refers to a circuit that duplicates the current in one active device to another active device. Simple circuits often utilise two bipolar transistors or FETs. An improved version of this is the double Wilson current mirror that utilises four transistors with the added advantage of cancelling the base current mismatch experienced with the basic current mirror and thus ensures the output current is equal to the input current.

This Wilson current mirror has been used in a study by Laopoulos et. al [13] to duplicate an embedded system's current consumption and estimate the energy consumed in many clock cycles accurately. In their setup, they used four bipolar transistors and the output from this current mirror circuit was monitored by a precision Digital Storage Oscilloscope. A GPIB bus connection from the DSO then transferred the results to a PC. The location of the current mirror in the setup controls the time resolution as the transistors characterise the frequency bandwidth. The benefits of this method are the high accuracy of the measurements and the ability to measure over different time intervals without being affected by the clock frequencies of the digital circuits [13]. Figure 2 illustrates a setup of the current mirror method and figure 3 the measurement scheme:

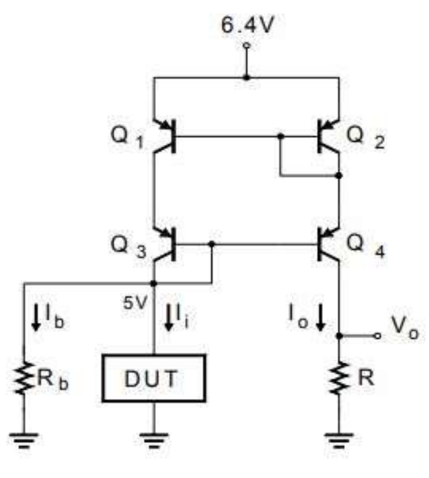

Figure 2. Current mirror. DUT is Device Under Test (i.e. microprocessor) [13]

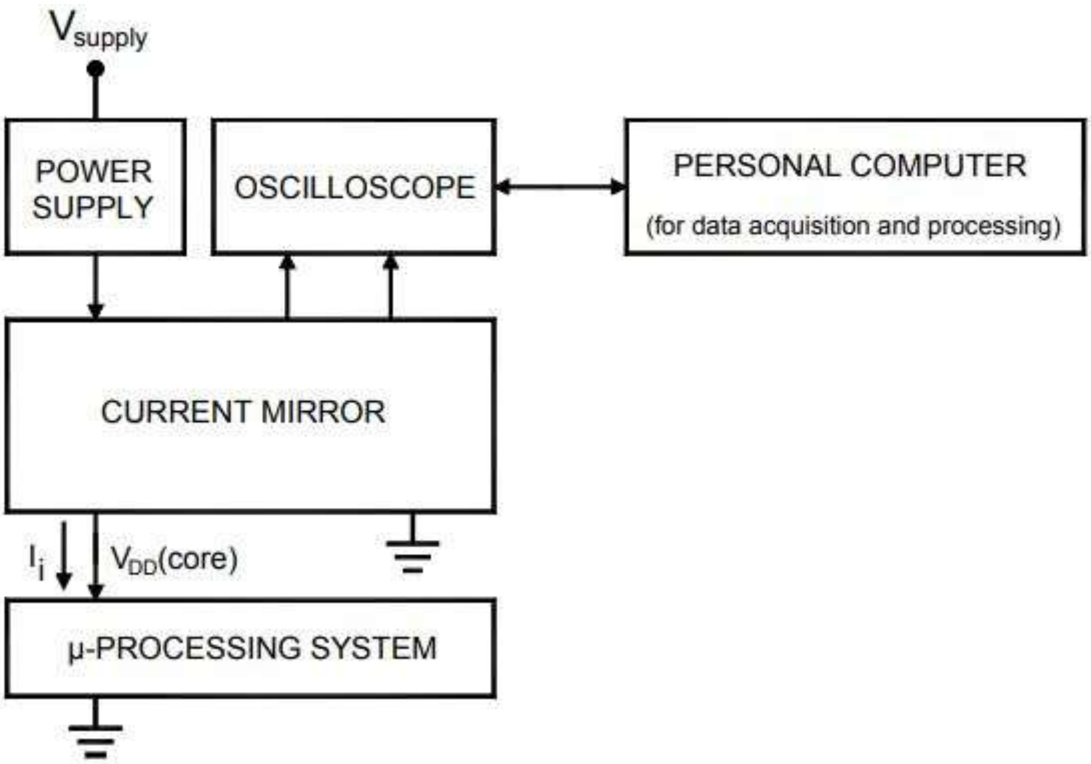

Figure 3. Current measurement scheme [13]

2.1.3 Charge Transfer Method

There exists two ways to accomplish the charge transfer method. The first involves charging a known capacitor from the power supply and discharging it through the embedded system as it consumes the current. The remaining voltage is then measured to note the energy that has been lost. The second method is carried out by charging a known capacitor by the consumption current and measuring the time it takes to completely discharge.

Previous studies have accomplished this by powering up an embedded system during the execution of a program from a charged capacitor and measuring the initial voltage V_1. Once the program has been executed, the final voltage V_2 is measured and the energy consumed can then be expressed as shown in equation (3).

$$E = \frac{1}{2}C(V_1 - V_2)^2 \quad (3)$$

In another study, the second charge transfer method was implemented in conjunction with the current mirror method. A capacitor was charged by the mirrored current for a specified time interval, after which it was cut off from the charging circuit and its discharge time measured. At the same time it was discharging, another capacitor is then getting charged by the mirror current. What was found was that the discharge time was proportional to the current consumed during the specified time interval.

The charge transfer method is beneficial in as far as offering real-time cycle accurate energy measurements. However, some of the drawbacks facing the charge transfer method are that they are complex to implement and, in some cases, require the use of high speed DAQ devices which are expensive.

2.1.4 Current Probe Method

Energy consumption can also be carried out by measuring the current through a wire using current probes. Current probes detect the magnitude and polarity of the electromagnetic field induced by the current in a wire. They are often used in high voltage lines and very rarely used with embedded systems. However, using current probes does not require the supply wire to be cut to insert a shunt resistor or precision ammeter. This explains its benefit for use in embedded systems. The drawback, however, is that they cannot sense current flowing on PCB routes. In a study by Bircher et. al, measuring the instantaneous processor power consumption of a Pentium 4 microprocessor using Agilent 1146A probes, was limited to the input conductors of the microprocessor’s voltage regulator module. The output of the current probe was then sampled by a 10 KHz National Instruments DAQ board and the results saved to a binary file for offline processing. [14.] Current probes are not preferred in many energy consumption measurements because they do not offer cycle accurate results and the probes are also quite expensive.

2.1.5 Battery Status Monitoring Method

Measuring energy consumption by monitoring the battery status is a method that can and has been used in mobile devices. An application can be run and the voltage drop that occurs can be measured to estimate the consumed energy. The code is executed many times and the voltage drop for a specified task can be calculated by dividing the total voltage drop by the number of times the code was run. Running the same task many times can misrepresent the results of software related power consumption as the fast cache memory will be filled by the code. For this reason, a slower external memory is used to store the instructions and the code is called on once. Smart Battery Systems can also be used to regulate the monitored voltage.

2.1.6 Thermal Monitoring Method

Energy consumption in embedded systems could be measured by monitoring thermal changes of the integrated circuits and using the information to estimate the leakage current. A study by Wolf et. al presented the idea with a model that indicated a linear relationship between the leakage current and thermal profile within the integrated circuit's operating temperature range [12]. Further research on thermal monitoring as a method of measuring energy consumption would be needed before this technique can be widely used.

2.2 Energy Harvesting

Energy harvesting, also known as energy scavenging, refers to the process of capturing and converting ambient energy into electrical energy to power small and autonomous electronic devices. [4, 1.]

Energy harvesting is useful for low-voltage and low-power mobile applications in different fields such as medical, military, transport, industry, agriculture and consumer electronics. It can sometimes be costly to replace batteries when they are very many and remotely located. Most of the energy harvesting applications are located close to the source and thus eliminate transmission losses, long cables and in some cases if the energy is enough to power the device, then the embedded system can operate battery-less. [5].

Naturally occurring energy in the environment can be used as a source of energy in energy harvesting. In some cases, energy that is a by-product of other processes, that does not have any other practical use can also be a source. Some common examples include: kinetic energy from vibration, stress and strain; thermal energy from escaped heat in furnaces, combustion engines or geothermal power plants; solar energy from the Sun or artificial light sources. Other sources that can also be used to provide energy are electromagnetic energy from inductors and transformers; wind and fluid energy from air and liquid flow such as rivers; chemical energy from biological processes and RF energy from radio and television broadcasting transmitters.

These sources of energy provide very small packets of energy that are quite difficult to capture for use. New circuits, such as transistors with “zero-threshold” MOSFETS have been developed over time to enable efficient energy management. [6]. They do this by capturing and accumulating these small packets of energy in capacitors or super capacitors, retaining them for a long period of time without losing energy and conditioning the energy to perform the required task. It is also necessary that these circuits can cater for all kinds of conditions, such as overvoltage or overcharge.

Transducers are used to convert the collected energy to electrical energy which can be harvested, stored and conditioned for many low voltage applications, such as wireless sensors. Examples of transducers are piezoelectric (PZT) crystals or fibre composites, photovoltaic cells, thermoelectric generators (TEGs) and electromagnetic inductor coils.

A typical energy harvester system will be composed of the following elements: an energy generator, capture/storage/management electronics and an embedded system such as a wireless sensor network as the load. This is illustrated in figure 4 below:

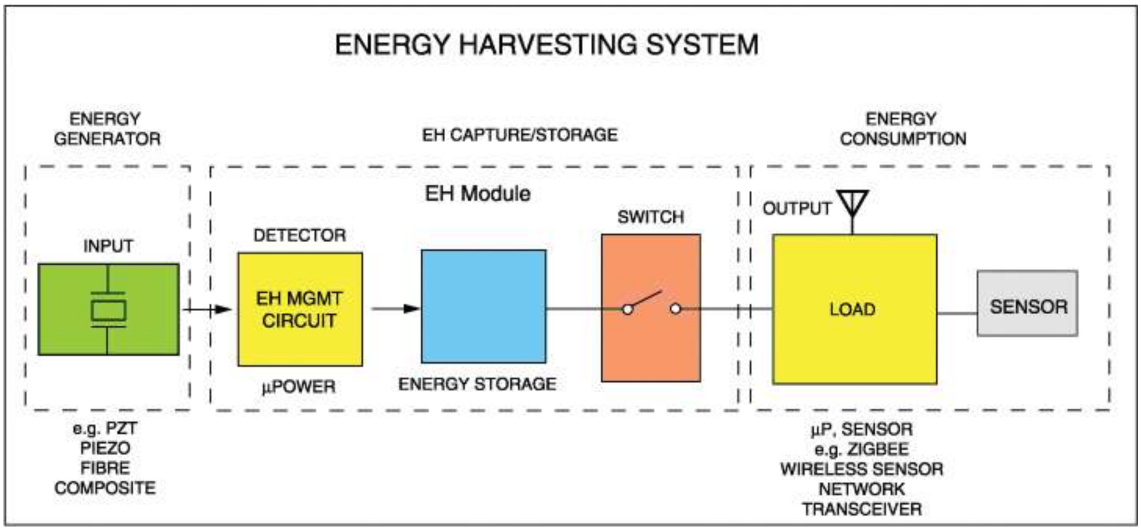

Figure 4. Energy harvesting system [6]

Mechanical vibrations, stress and strain can be converted to electrical energy through piezo fibre composites as shown in figure 4. Different sources can provide mechanical vibrations and stress for example vibrations from an aircraft engine, human or automobile motion on a bridge and low frequency seismic vibrations felt on the earth.

In most cases, piezoelectric generators produce high voltages but extremely low currents. Thus, the power generated is too low to power electronics but that does not hinder the energy harvesting capability.

The capture and storage process is initiated by the detector circuit that converts the collected AC voltage to DC. It typically accepts instantaneous input voltages in a steady stream of pulses or in irregular bursts ready for storage. Early energy harvesters required an input of at least 4V to capture and store energy. However, modern designs feature a front-end voltage booster that could initiate the capture and storage process with input voltages that are less than 100mV. [6].

The energy storage is implemented using capacitors or ultracapacitors which are designed to operate at two threshold voltages: +V_low DC and +V_high DC. These threshold voltages typically correspond to the load's threshold minimum and maximum voltages, that is, VL and VH respectively. The capacitor or ultracapacitor will get charged until it reaches VH and the output will switch "on demand" to power the embedded device. As the output gets depleted and reaches VL, it will switch off and restart the charging cycle back to VH.

For optimum efficiency and long energy retention times, energy harvesting systems often include micro power electronics to ensure that the energy consumed by the harvesting electronics is much less compared to the energy input by the generating source. It is also common to include other energy generators to supplement any energy leakage or loss within the system.

2.3 ZigBee Technology

ZigBee technology is a wireless communication standard that defines a set of protocols for use in low data rate, short to medium range wireless networking devices like sensors and control networks [7].

ZigBee, a standard developed by the ZigBee Alliance in 2002, is based on the IEEE 802.15.4 standard for wireless personal area networks (WPAN). The frequencies at which the ZigBee WPANs operate at are: 868 MHz, 902 – 928 MHz and 2.4 GHz [8]. These frequencies determine the data rate of the ZigBee technology, with the 868 MHz band supporting a data transfer rate of 20 Kbps whereas the unlicensed 2.4 GHz ISM (Industrial Scientific and Medical) band supports data rates up to 250 Kbps. A typical ZigBee modem is as shown in figure 5:

Figure 5. ZigBee Modem [9]

2.3.1 ZigBee Architecture

The ZigBee architecture or ZigBee stack consists of various layers. The PHY (Physical) and MAC (Medium Access Control) layers are defined by the IEEE 802.15.4 standard, offering it similar functionality as the physical and data link layers of the OSI model. The NWK (Network) and application layers are defined by the ZigBee Alliance. Figure 6 below illustrates this architecture:

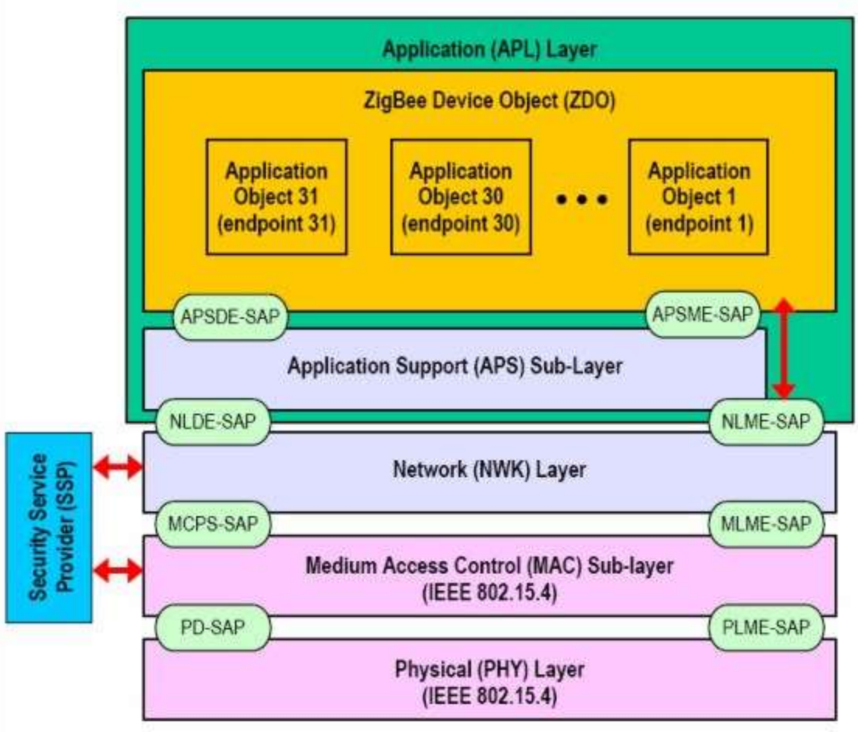

Figure 6. ZigBee Architecture [10]

Physical Layer: The functions of this layer include control and activation of the radio transceiver. The physical layer communicates with the radio devices through packets. It is made up of a Synchronisation Header (SHR) that synchronises the receiver, a Physical Header (PHR) that contains the frame length information and a PHY Payload that includes data or commands and functions as a frame provided by the upper layers.

MAC Layer: Data addressing is one of the functions of this layer. This is done to establish where the frame is going or coming from. The MAC layer also ensures reliable transmission of data by accessing different networks through carrier sense multiple access with collision avoidance (CSMA/CA). It is also responsible for beaconing and this is defined by four frame structures: a beacon frame, that is used by a co-ordinator to transmit beacons; a data frame that is used for all data transfers; an acknowledgment frame, that is used to confirm successful transmission of a frame and a MAC command frame, that handles all MAC peer entity control transfers. This MAC layer can be exploited by higher layers to attain secure communication for multi-hop messaging [10].

Network Layer: The network layer interfaces the MAC layer and the application layer and its function is to provide network management by establishing a new network, the ability to join a network and selecting the network topology. It also provides routing management and security management of the network. This layer is also the network message broker.

Application Layer: This is the highest layer of the ZigBee stack and contains the application support sub layer, the ZigBee device object and up to two hundred and forty application objects. The application objects are manufacturer defined and their function is to implement the user applications as defined by the ZigBee application descriptions. The functions of the APS and ZDO are as follows:

Application Support Sub Layer: This layer has two specific functions, that is, binding and discovery. Binding refers to matching two devices together by their services and needs and relaying messages between them. Discovery is the ability to identify which other devices are operating in another device's personal operating space.

ZigBee Device Object: The role of the ZDO is to interface between application objects, device profiles and the APS. The ZDO also defines the role of the device in the network, responds and/or initiates binding requests and launches secure connection between devices in a network.

2.3.2 ZigBee Devices

There are two types of devices that are specified by the IEEE 802.15.4 standard and these are: Full Function Device (FFD) and Reduced-Function Device (RFD). An FFD can function in any network topology, take up any role in the network and talk to any device in the network. An RFD on the other hand, is limited as it can only talk to the co-ordinator or network co-ordinator depending on the network topology.

These two devices, FFD and RFD can take up different roles within the network: Personal Area Network (PAN) co-ordinator, which is the main controller in the network that sends beacon frames and provides routing information; co-ordinator which relays messages and device which can be a normal end device, either FFD or RFD.

The ZigBee standard used this same IEEE 802.15.4 specification to define its protocol's devices as:

- ZigBee Co-ordinator: which performs same functions as a PAN Co-ordinator.

- ZigBee Router: which enhances the range of the network making it possible for more devices to be added to the network.

- ZigBee End Device: which can serve as a FFD or RFD.

2.3.3 ZigBee Network Topologies

The network topologies used in the ZigBee standard are based off the star and peer-to-peer topologies used in the IEEE 802.15.4 standard. This leads to three topologies used in ZigBee which are: star, tree and mesh topologies, as described below.

Star Topology: In the star topology, there exists only one co-ordinator that is responsible for starting and managing the devices in the network. There can be several end devices. However, they are all physically and electrically isolated, so communication between any two devices must go through the co-ordinator. The advantage of the star topology is that it is easy to implement. However, a huge drawback to it lies in the fact that a minor co-ordinator fault could compromise the entire network. Figure 7 below illustrates a star topology:

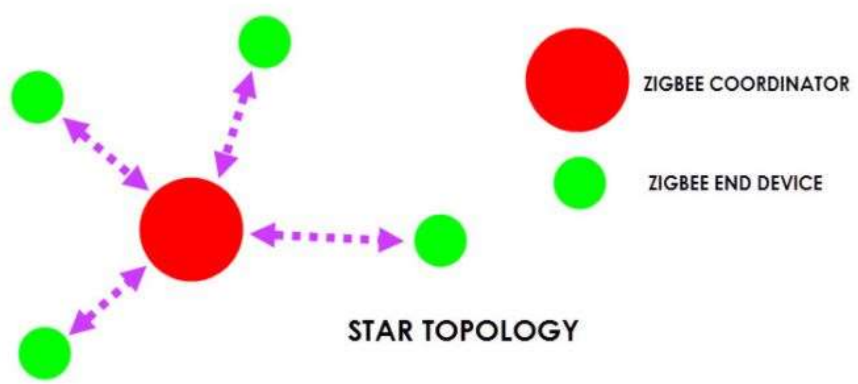

Figure 7. Star Topology [7]

Tree Topology: This topology is a type of peer-to-peer topology. The ZigBee co-ordinator is considered the parent while the end devices are the children. Routers are used to enhance the range of the network, becoming intermediate parents to the end devices connected to them. Communication between two end devices must go through the router to the ZigBee co-ordinator. This topology is not very efficient, however, because as much as the range is increased, a router fault can cut off the end devices connected to it from the network. A tree topology is as shown in figure 8 below:

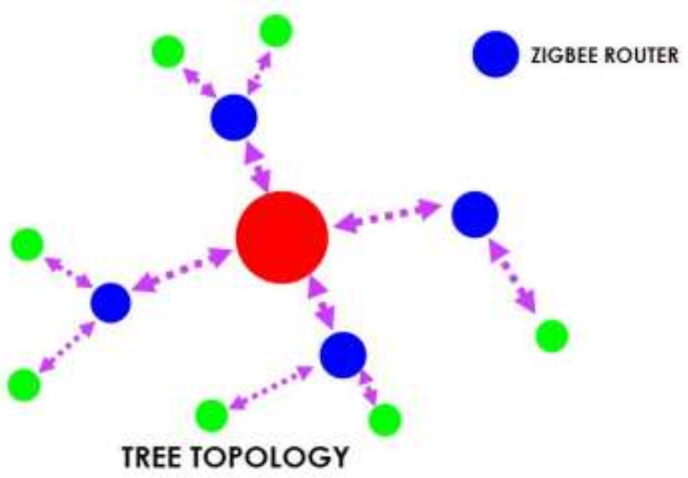

Figure 8. Tree Topology [7]

Mesh Topology: This is also based on peer-to-peer technology and is similar to the tree topology but with added flexibility. The routers in mesh topology are FFD and can thus communicate with other routers in range and to the co-ordinator as well. ZigBee has a route discovery feature that thrives in this topology, as an end device finds the best route to a given node, thus making the network “self-healing” in the case of a failure of a router. Figure 9 below illustrates this topology:

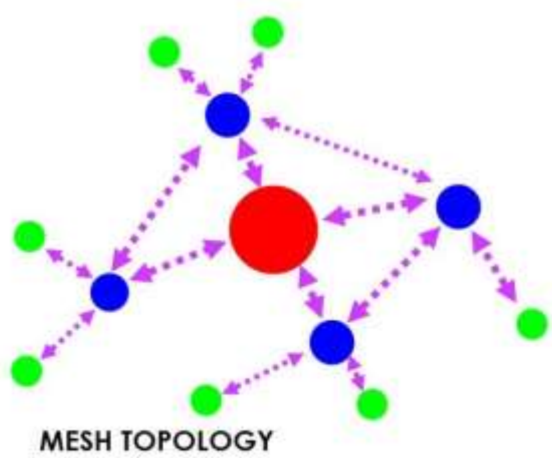

Figure 9. Mesh Topology [7]

2.3.4 ZigBee Applications

The benefits of ZigBee being a low cost, low power technology with low data rate make it possible for it to be used in different applications to make them more effective. For example, in smart homes, HVAC (heating, ventilation, air conditioning) and lighting can be managed with automated control of the systems related. This in turn improves the conservation of energy by creating "greener" homes as well as allocate utility costs in relation to actual consumption. Home monitoring is also achievable and thus an added security measure.

ZigBee has also been used in retail services to enhance the shopping experience. This has been done by creating faster check-outs and availability of in-store assistance. The retailer also benefits from reducing stock-outs, as they can manage the inventory better through electronic shelf labels and even monitor incidents such as spills.

Health care automation has also been achieved using ZigBee technology. Wearable ZigBee devices worn on a patient can be used to monitor body parameters and transmit them to the healthcare staff remotely, who can then prescribe the needed medication.

2.3.5 Energy Saving Methods with ZigBee End Nodes

There are two main modes of operation in which ZigBee devices operate in, that is, beacon mode and non-beacon mode. Co-ordinators that are battery-powered typically operate in the beacon mode while those that are powered by the mains supply fall in the non-beacon mode category.

In the beacon mode, the co-ordinator wakes up periodically and transmits beacons to the routers/devices in the network. On receipt of the beacon, the device/router wakes up and checks if there's incoming data. Once the message transmission is complete, or if there was no message sent, the co-ordinator dictates a schedule for the next beacon and the device/router, as well as the co-ordinator, go to sleep. This is known as a GTS (Guaranteed Time Slot) and can be adjusted to an applications minimum response time or desired rate. Thus, having the ZigBee devices on sleep mode and adjusting the beacon intervals according to an application's minimum response time or desired rate to achieve shorter duty cycles can promise long battery lives.

In the non-beacon mode, the co-ordinators and routers are always active while the devices are asleep. Unlike in beacon mode where the co-ordinator would dictate a schedule for the device to turn on again, in non-beacon mode the devices can wake up randomly and talk to the co-ordinator or router. Overall power consumption is saved by the end nodes in non-beacon mode as the devices are asleep for long periods of time.

The RF power rating of a ZigBee device specifies its transmission power which influences the power the device will consume and its battery life over time. An example from Cirronet's ZigBee solutions states that a 100 mW RF powered node will typically consume 150 mA at 3.3 V when transmitting, compared to 75 mA at 3.3 V for a 1 mW RF powered node [11]. The 100 mW node consumes double the power of the 1 mW node when actively transmitting. In Cirronet's study, if the high RF radio is awake and transmitting 5% of the time, suggesting a very active node, then the extra average power consumption is about 5%. This means that, a battery that would last five years with a 1 mW node would last four years and nine months with a 100 mW node.

3 Methods and Materials

To be able to monitor the power consumption of a ZigBee end node, an XBee ZigBee Mesh Kit by DIGI was used. The kit can be used to create a mesh network to help understand how RF modules connect and transmit messages in the ZigBee protocol. The kit contains two hardware footprints, that is, Through-hole Technology (THT) and Surface-mount Technology (SMT). Some versions of the kit may contain antennas or extra THT or SMT modules. For this project, the kit contents are as presented in the table 1:

Table 1. Digi XBee ZigBee Mesh Kit Contents

S2C ZigBee Kit Qty	Part	
1	Xbee Grove Development Board	
2	XBee Grove Development Board	
1	XBee ZigBee THT module	
2	XBee ZigBee SMT module	
3	Micro USB cables	

XBee devices can be connected to PCs or other microcontrollers such as Arduino or Raspberry Pi to generate data for wireless transfer to other XBee devices. When connected to these intelligent devices, the data is transmitted serially in two modes, that is: Application Transparent (transparent mode) or Application Programming Interface (API mode).

In the transparent mode, the XBee module will send the data from the PC or microcontroller exactly as it is wirelessly to the other XBee module. Upon receiving the data, the receiver module serially transmits to its host device exactly what was sent. This mode provides a simple interface for two-way communication but it has very limited functionality.

In the API mode, a protocol defines the exchange of communication. The data is communicated serially in organised packets (also known as API frames) and in a determined order. This makes it possible to establish complex communication as the data can be transmitted to multiple destinations and in a much faster way.

For this project, the THT module and one SMT module were used. The modules were plugged in to the boards and as a cautionary measure, the board should not be powered when doing so. For the THT module, the footprint is matched to the white lines on the board and for the SMT module, the pins are aligned with those on the board and pushed until it hooks in as illustrated in figure 10 below:

Figure 10. How to plug in ZigBee Modules [15]

The loopback jumper on the board should be kept in the UART position and the board can be plugged in to the PC for configuration. The XCTU application enables one to interact with the XBee modules through a simple and easy to use graphical user interface. The application can be downloaded from the Digi website and is compatible with 32-bit or 64-bit Windows Vista/7/8/10 versions, 64-bit Mac OS X v10.6 and higher versions, 32-bit or 64-bit Linux with KDE or GNOME window managers. The system requirements for the XCTU are as in the table 2:

Table 2. XCTU system requirements [15]

Property	Minimum	Recommended
HDD space	500 MB	1 GB
RAM memory	2 GB	4 GB
CPU	Dual-core processor	Quad-core processor

The XCTU application works in three different modes. The first is the configuration working mode which is the default mode that is presented when the application is started and it is used to configure the XBee modules added. The second mode is the consoles working mode which is used to interact with the selected devices and finally is the network working mode which is used to visualise the topology and interconnections of your network. [15.]

Once the XCTU is installed, and with the devices connected to the PC, they can be added to the application by selecting the discover devices icon on the toolbar as shown in figure 11 below:

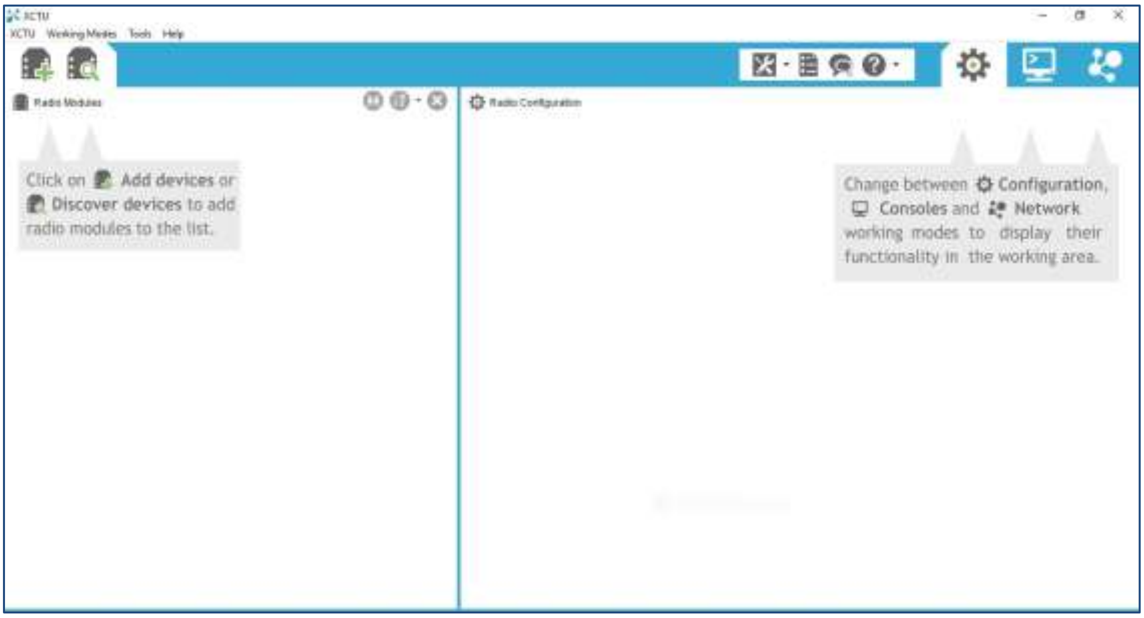

Figure 11. XCTU opening screen

It is preferred to select all when the XCTU dialog box opens asking to select the ports to be searched. The values in the port parameters window can be maintained and the XCTU will carry out its process and eventually the devices connected will be displayed as in figure 12 to be added to the application.

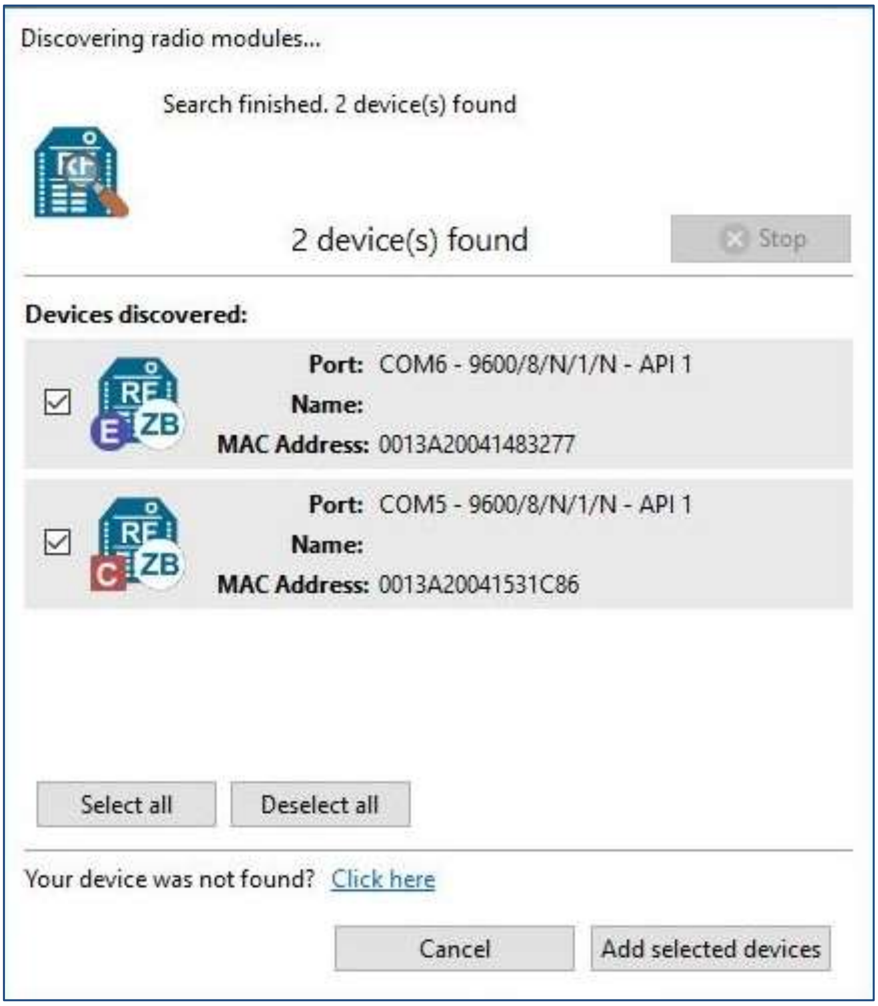

Figure 12. Discover radio modules screen

For the devices to be able to communicate with each other, they must be in the same network and in any network, there must be a coordinator. The default settings of the device can be restored by clicking the load default settings icon in the Radio Configuration section. Table 3 illustrates the settings used for this project.

Table 3. XCTU configuration settings

Param	XBee A	XBee B	Effect
ID	2015	2015	Defines the network that a radio will attach to.
JV	–	Enabled [1]	Verifies if a coordinator exists on the same channel to join the network or to leave if it cannot be found.

CE	Enabled [1]	–	Sets the device as the coordinator.
NI	COORDINATOR	END_DEVICE	Defines the node identifier, a human friendly name for the module. Note: The default NI value is a blank space and should be deleted when changing the value.
AP	API Enabled [1]	API Enabled [1]	Enables the API mode.
SP	3E8	3E8	Defines the duration of time spent sleeping. 3E8 (hexadecimal) = 1000 (decimal) x 10 ms = 10 seconds.
SM	–	Cyclic sleep [4]	Enables cyclic sleep mode in the end device.
SO	–	1	Keeps the module awake for the entire period.

Note: The dash (–) in the table means to maintain the default value.

After writing the new settings into the devices, switch to the consoles working mode and open the serial connection of both devices. To generate the transmit request frame, the coordinator’s console is detached so that two consoles can be viewed at the same time. The add new frame to the list icon is selected and the frame is created using the frames generator tool which will basically open a window and the filled in settings for this project are as shown in figure 13:

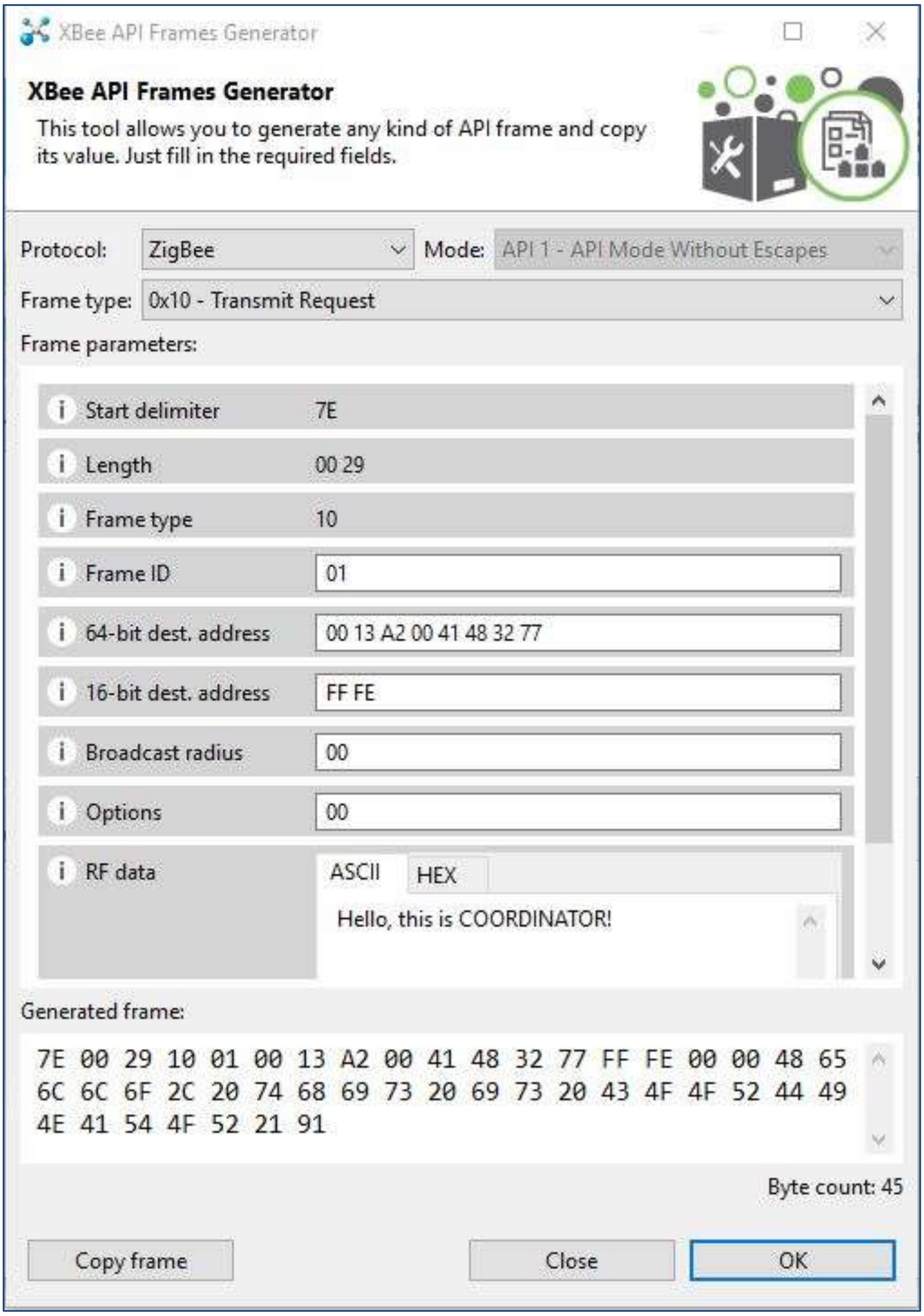

Figure 13. XBee API Frames Generator

After clicking ‘OK’ and adding the frame, it is now ready to be transmitted simply by clicking the ‘send selected frame’ button in the consoles working mode. The measurement setup was based on the current shunt type using a 1Ω resistor in the high side of the end node's serial connection and voltage measurements were taken by an oscilloscope. Figure 14 illustrates this:

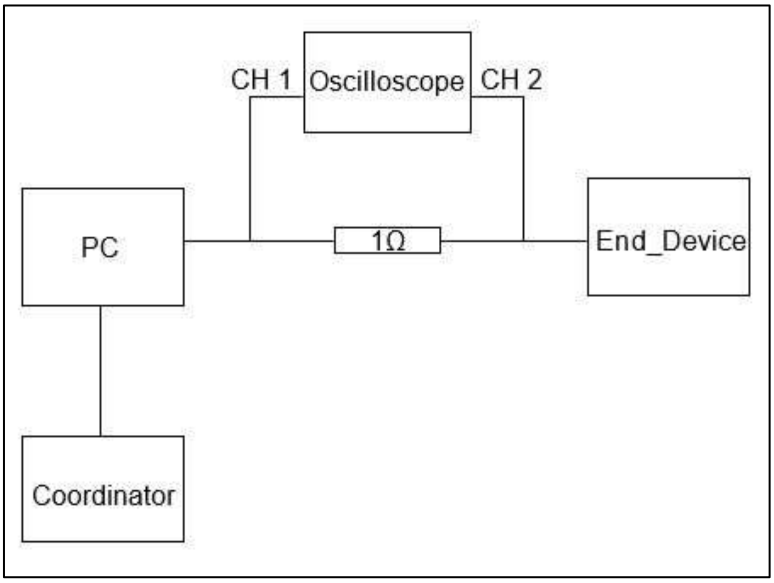

Figure 14. Measurement Setup

4 Results and Discussion

Measurements of the voltage drop over the known resistor corresponding to the current drain have been arranged as follows:

4.1 Consumption When Idle

After the configuration settings have been set and before the serial connection is opened on the XCTU Consoles working mode, the end node is in an idle state. During this idle state, the ZigBee end node checks for valid data. However, it does not transmit or receive anything. Figure 15 illustrates the measurement as seen on the oscilloscope.

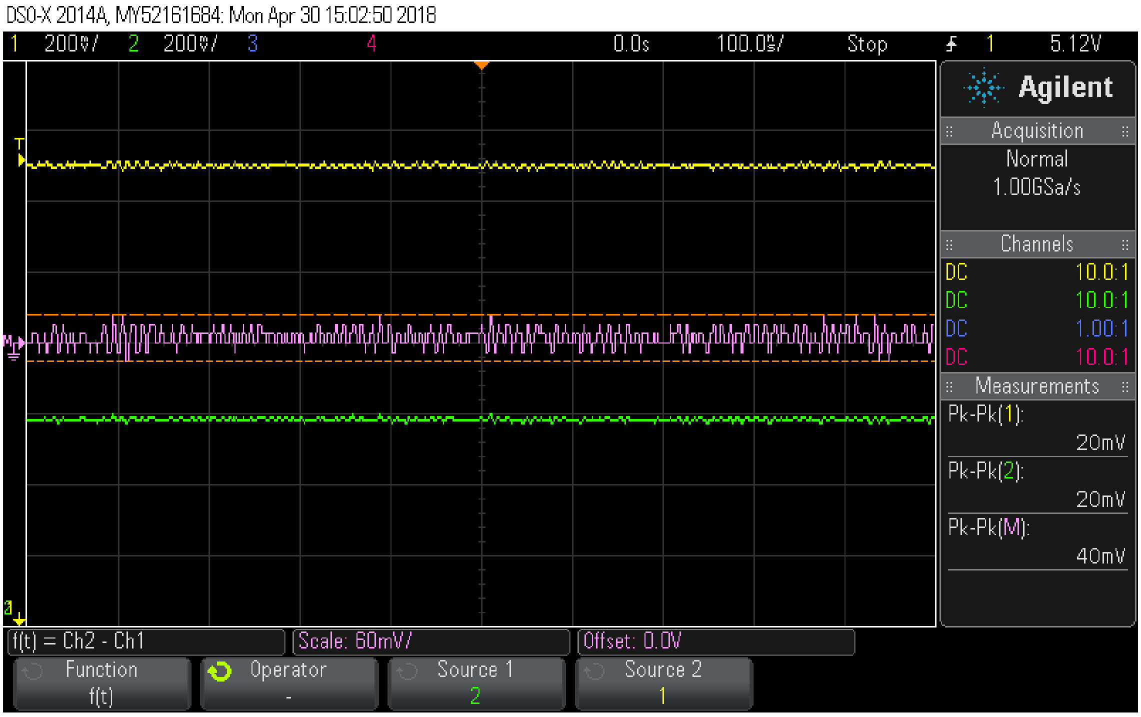

Figure 15. Consumption when idle

Since the value of the resistor is 1Ω and there is a linear relationship between voltage and current, i.e. Ohm's Law, the current can be read directly from the oscilloscope measurement and thus 40 mA is the current drained when idle.

4.2 Consumption during Start-up

When the serial connection is opened in XCTU Consoles working mode, the transition from idle to start-up takes about 0.45 µs as marked by phase 1 in the oscilloscope image figure 16. After that, the voltage variations that occur are due to the end node initiating an active scan for a coordinator, and the current consumption in this phase as measured by the oscilloscope was 121 mA.

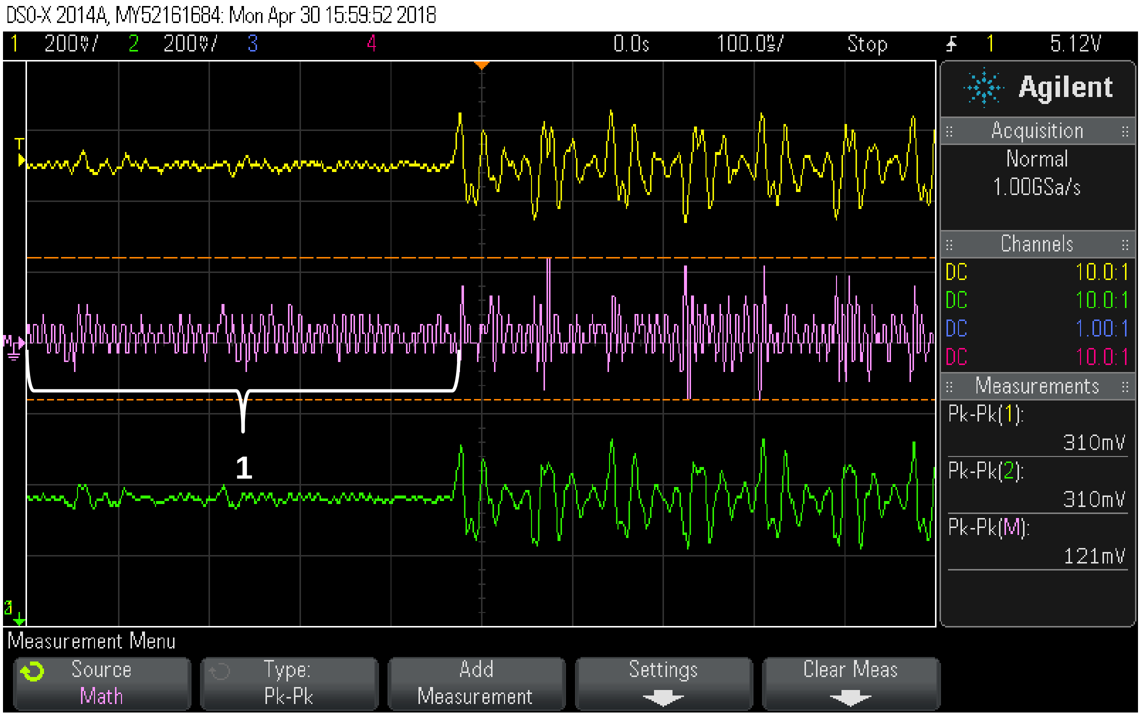

Figure 16. Consumption during start-up

4.3 Consumption during Association

The active scan ends when the end device discovers the coordinators PAN ID. Once that happens an association request is sent by the end device to the coordinator and waits for confirmation. The coordinator then sends the association confirmation resulting in the ‘ASSOC’ LED to start blinking. Figure 17 below illustrates the oscilloscope image of this phase with the current consumption

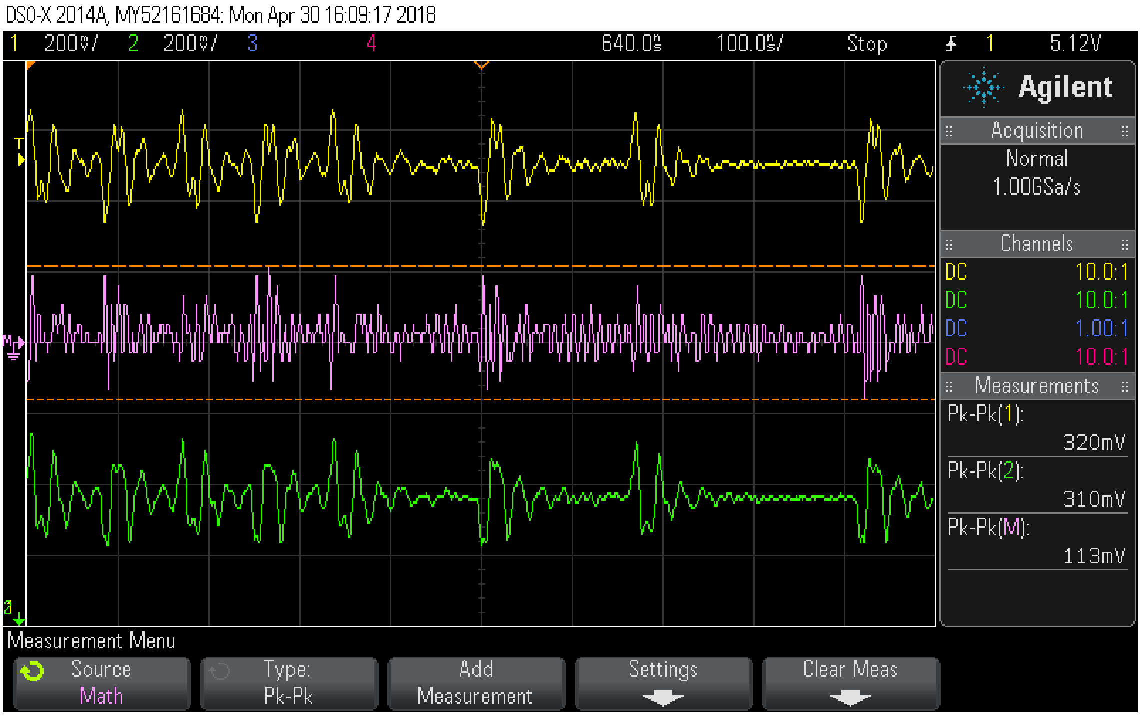

Figure 17. Consumption during association

4.4 Consumption during Packet Transmission

The coordinator sends a packet of data to the end node as a transmit request and this message gets queued when the end device has gone to sleep mode after establishing a connection to the coordinator. When the end device wakes up again, it sends a beacon to the coordinator to request for RF data. The coordinator sends an acknowledgement (ACK) message to the end device followed by the API frame and then the end node finally sends an acknowledgment back to the coordinator. Figure 18 shows the consumption when the data request occurs indicated by the spike.

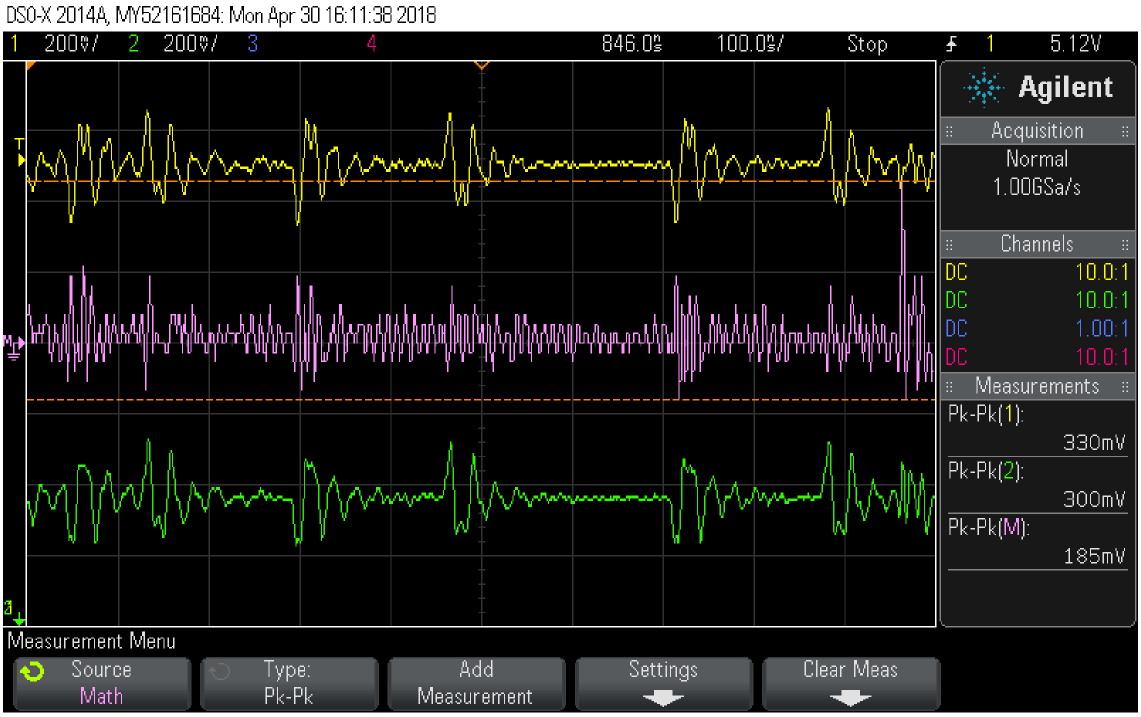

Figure 18. Consumption during data request

When the end node receives the API frame it transfers it serially to the host device then reverts to sleep mode. A confirmation message on the XCTU frames log indicates the receive packet and the consumption that occurs can be seen by the spike that occurs in figure 19.

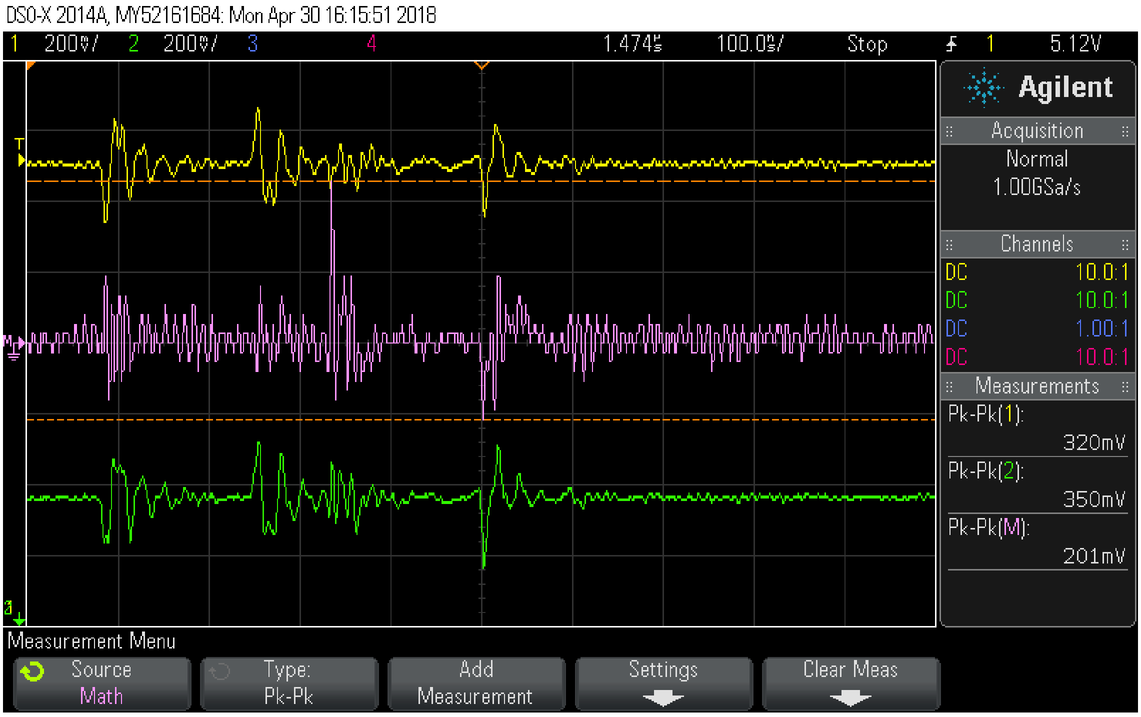

Figure 19. Consumption during serial transfer

The measurements were repeated with different sleep periods set on the end nodes however the results were quite similar. The transmit interval of the frame being sent by the coordinator was also changed and set to loop five times and then ten times, but the results in both cases were still similar to sending the frame once. This challenge thus hindered the possibility to optimize the energy consumption.

The measured results were out of tolerance with values presented in the datasheet (see appendix 1). In this case, the receive current which can be considered as the consumption just after association with the coordinator has occurred was found to be 113 mA while in the datasheet it is mentioned to be 28 mA. This inaccuracy could possibly be attributed to the high frequency components and the oscilloscope not being able to accurately display them. Hence the noisy signal.

The oscilloscope's vertical resolution also hindered the measurement of the consumption of energy during sleep mode. In addition to that, the resistor's low value would not permit an accurate measurement for current that is less than 1 µA as indicated in the datasheet (see appendix 1).

5 Conclusion

The initial goal of this project to study different methods of measuring energy consumption in embedded systems and energy harvesting methods has been achieved. The information presented can be used to carry out in depth studies into any of the methods and compared to the current shunt method used in this project.

There was one major drawback that was faced in this project. The measurements were limited by the vertical resolution of the oscilloscope when trying to measure the energy consumption when the ZigBee was in sleep mode. This can, however, be solved by using higher series oscilloscopes which are quite expensive. For the scope of this project, the results obtained are enough to show the energy consumption at different stages of the ZigBee end node from idle to packet transmission.

As energy harvesting continues its growing trend, optimizing energy consumption becomes a crucial factor to accomplish. This thesis project can thus be further developed and used in the research.

References

1 Nakutis Ž. Embedded Systems Power Consumption Measurement Methods Overview. ISSN 1392-1223 Matavimai, 2009;44(2):29-35.

2 National Instruments. Make Accurate Power Measurements with NI Tools [online]. Amsterdam, NL: National Instruments; 17 March 2008. Available from: http://www.ni.com/white-paper/7077/en/. Accessed 5 March 2018.

3 Naehyuck Chang, Kwanho Kim, Hyung Gyu Lee. Cycle-Accurate Energy Measurement and Characterization With a Case Study of the ARM7TDMI. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2002;10(2):146-154

4 Beeby S, White N. Energy Harvesting for Autonomous Systems. Norwood, MA: Artech House; 2010.

5 Kingatua A. The How and Why of Energy Harvesting for Low-Power Applications [online]. USA: All About Circuits; 23 June 2016. Available from: https://www.allaboutcircuits.com/technical-articles/how-why-of-energy-harvesting-for-low-power-applications/. Accessed 6 March 2018.

6 Kinman M. What is Energy Harvesting and How Does it Work? [online]. Virginia, USA: Amazon; 4 April 2010. Available from: https://www.wirelessdesignmag.com/article/2010/04/what-energy-harvesting-and-how-does-it-work. Accessed 13 March 2018.

7 Ravi. What is Zigbee Technology? Architecture, Topologies and Applications [online]. New Jersey, USA: Electronics Hub; 17 November 2017. Available from: https://www.electronicshub.org/zigbee-technology-architecture-applications/. Accessed 13 March 2018.

8 Electrical Technology. What is ZigBEE Technology and How it works? [online]. Utah, USA: Electrical Technology; 20 September 2017. Available from: https://www.electricaltechnology.org/2017/09/zigbee-technology-wireless-networking-system.html. Accessed 13 March 2018.

9 Agarwal T. ZigBee Wireless Technology Architecture and Applications [online]. Mumbai, India: Elprocus; 2014. Available from: https://www.elprocus.com/what-is-zigbee-technology-architecture-and-its-applications/#comments. Accessed 13 March 2018.

10 Kinney P. ZigBee Technology: Wireless Control that Simply Works [online]. Communication Design Conference; 2 October 2003. Available from: https://www.itk.ntnu.no/fag/TTK4545/TTK2/PDF/ZigBee%20Technology%20-%20Wireless%20control%20that%20simply%20works.pdf. Accessed 14 March 2018.

11 Cirronet. Sorting Out ZigBee Power Options [online]. Massachusetts, USA: EETimes; 3 July 2007. Available from: https://www.eetimes.com/document.asp?doc_id=1273485#msgs. Accessed 14 March 2018.

12 Wolf F, Kruse J, Ernst R. Timing and Power Measurement in Static Software Analysis. Microelectronics Journal 2002;33(1-2):91-100.

13 Laopoulos T, Neofotistos P, Kosmatopoulos C, Nikolaidis S. Measurement of Current Variations for the Estimation of Software-related Power Consumption. IEEE Transactions on Instrumentation and Measurement 2003;52(4):1206-1212.

14 Bircher W.L, Valluri M, Law J, John L.K. Runtime Identification of Microprocessor Energy Saving Opportunities. Proceedings of the International Symposium on Low Power Electronics and Design 2005;ISLPED ’05:275-280.

15 Digi. Xbee Zigbee Mesh Kit User Guide [online]. Minnesota, USA: Digi International; 21 February 2018. Available from: https://docs.digi.com/display/XBeeZigBeeMeshKit. Accessed 4 March 2018.

SPECIFICATIONS	Digi XBee® S2C ZigBee Standard	Programmable	Digi XBee-PRO® S2C ZigBee Standard	Programmable	Digi XBee® S2D ZigBee Thread Ready Standard
PERFORMANCE					
TRANSCEIVER CHIPSET	Silicon Labs EM357 SoC				Silicon Labs EM3587 Soc
DATA RATE	RF 250 Kbps, Serial up to 1 Mbps				
INDOOR/URBAN RANGE	200 ft (60 m)		300 ft (90 m)		200 ft (60 m)
OUTDOOR/RF LINE-OF-SIGHT RANGE	4000 ft (1200 m)		2 miles (3200 m)		4000 ft (1200 m)
TRANSMIT POWER	3.1 mW (+5 dBm) / 6.3 mW (+8 dBm) boost mode		63 mW (+18 dBm)		3.1 mW (+5 dBm) / 6.3 mW (+8 dBm) boost mode
RECEIVER SENSITIVITY (1% PER)	-100 dBm / -102 dBm boost mode		-101 dBm		-100 dBm / -102 dBm boost mode
FEATURES					
SERIAL DATA INTERFACE	UART, SPI				
CONFIGURATION METHOD	API or AT commands, local or over-the-air (OTA)				
FREQUENCY BAND	ISM 2.4 GHz				
FORM FACTOR	Through-Hole, Surface Mount				Surface Mount
INTERFERENCE IMMUNITY	DSSS (Direct Sequence Spread Spectrum)				
ADC INPUTS	(4) 10-bit ADC inputs				
DIGITAL I/O	15				
ANTENNA OPTIONS	Through-Hole: PCB Antenna, U.FL Connector, RPSMA Connector, or Integrated Wire SMT: RF Pad, PCB Antenna, or U.FL Connector				
OPERATING TEMPERATURE	-40º C to +85º C				
DIMENSIONS (L X W X H) AND WEIGHT	Through-Hole: 0.960 x 1.087 in (2.438 x 2.761 cm) SMT: 0.866 x 1.33 x 0.120 in (2.199 x 3.4 x 0.305 cm)		Through-Hole: 0.960 x 1.297 in (2.438 x 3.294 cm) SMT: 0.866 x 1.33 x 0.120 in (2.199 x 3.4 x 0.305 cm)		SMT: 0.866 x 1.33 x 0.120 in (2.199 x 3.4 x 0.305 cm)
PROGRAMMABILITY					
MEMORY	N/A	32 KB Flash/ 2 KB RAM	N/A	32 KB Flash/ 2 KB RAM	N/A
CPU/CLOCK SPEED	N/A	HCS08 / up to 50.33 MHz	N/A	HCS08 / up to 50.33 MHz	N/A
NETWORKING AND SECURITY					
PROTOCOL	ZigBee PRO 2007, HA-Ready with support for binding/multicasting				
ENCRYPTION	128-bit AES				
RELIABLE PACKET DELIVERY	Retries/Acknowledgements				
IDS	PAN ID and addresses, cluster IDs and endpoints (optional)				
CHANNELS	16 channels		15 channels		16 channels
POWER REQUIREMENTS					
SUPPLY VOLTAGE	2.1 to 3.6V		2.7 to 3.6V		2.1 to 3.6V
TRANSMIT CURRENT	33 mA @ 3.3 VDC / 45 mA boost mode	47 mA @ 3.3 VDC / 59 mA boost mode	120 mA @ 3.3 VDC	120 mA @ 3.3 VDC	33 mA @ 3.3 VDC / 45 mA boost mode
RECEIVE CURRENT	28 mA @ 3.3 VDC / 31 mA boost mode	42 mA @ 3.3 VDC / 45 mA boost mode	31 mA @ 3.3 VDC	45 mA @ 3.3 VDC	28 mA @ 3.3 VDC / 31 mA boost mode
POWER-DOWN CURRENT	<1 μA @ 25º C	1.5 μA @ 25º C	<1 μA @ 25º C	1.5 μA @ 25º C	<3 μA at 25º C
REGULATORY APPROVALS					
FCC, IC (NORTH AMERICA)	Yes		Yes		Yes
ETSI (EUROPE)	Yes		No		Yes
RCM (AUSTRALIA AND NEW ZEALAND)	Yes		Yes		No (Coming Soon)

It's the easy and fast way to build a wireless mesh network using Digi XBeemodules.Tolearnmorevisitdocs.digi.com.

877-912-3444 I 952-912-3444