EMBEDDED SYSTEM BASED SUBSEA OIL PIPELINE MONITORING SYSTEM USING MULTI-FUNCTIONAL SENSORS

A Thesis

Presented to

The Faculty of the Department of Electrical and Computer Engineering

University of Houston

In Partial Fulfillment

of the Requirements for the Degree of

Master of Science

in Electrical Engineering

by

Deepthi Badam

August 2012

EMBEDDED SYSTEM BASED SUBSEA OIL PIPELINE MONITORING SYSTEM USING MULTI-FUNCTIONAL SENSORS

Deepthi Badam

Approved:

Chair of the Committee
Dr. Yuhua Chen,
Associate Professor
Electrical and Computer Engineering

Co-Chair of the Committee
Dr. Abdelhak Bensaoula,
Research Professor
Electrical and Computer Engineering

Committee Members:

Dr. Mounir Boukadoum,
Professor
Department of Computer Science, UQAM, Canada

Dr. Suresh Khator,
Associate Dean
Cullen College of Engineering

Dr. Badri Roysam,
Professor and Chairman
Electrical and Computer Engineering

ACKNOWLEDGEMENTS

This research would not have been possible without the support of God Almighty and all praise to him. I owe my deepest gratitude to Dr. Abdelhak Bensaoula, for giving me an opportunity to pursue my Master's thesis under his esteemed guidance. Without his support and assistance, this project would not have been possible. His timely support, encouragement, and logical thinking have greatly inspired me to keep my spirits high throughout the research.

I wish to express thanks to Dr. Mounir Boukadoum for his valuable advice regarding the research. His extensive knowledge over the subject was of great help to me. I would also like to thank Dr. Yuhua Chen for accepting and being part of my thesis committee. It gives me pleasure to thank all the members of the Nitride Materials and Devices Laboratory, especially Mr. Alan Price and Mr. Sofiane Amara Madi.

I would like to dedicate my work to my loving husband, Mr. Kishore Potta, and my daughter, Laasya Sree. Without their encouragement and support the research would not have been possible. My special gratitude is due to my parents and other family members. Special thanks to my friends, Sridhar Potta, Ananth Gondu, Shiva Kailaswar, and Radha Krishna Kotti for helping me at various stages to increase the quality of my work.

EMBEDDED SYSTEM BASED SUBSEA OIL PIPELINE MONITORING SYSTEM USING MULTI-FUNCTIONAL SENSORS

An Abstract

of a thesis

Presented to

The Faculty of the Department of Electrical and Computer Engineering

University of Houston

In Partial Fulfillment

of the Requirements for the Degree of

Master of Science

in Electrical Engineering

by

Deepthi Badam

August 2012

ABSTRACT

The concept of employing embedded systems with sensor network for monitoring underwater oil pipelines infrastructures has been investigated by many researchers. This work presents a similar demonstration of a monitoring system detecting temperature and motion of the underwater pipelines, using microcontrollers, sensors and communication via RF.

In this project, Optical communication is employed for monitoring purposes. To support the project’s goals during different phases of pipeline installation, RF communication was found to be the ideal complementary solution. The proposed system provides continuous data regarding the acceleration, the local magnetic field and the temperature of the pipeline using different sensors. An energy harvesting system was also designed to meet the monitoring system’s power requirements.

In my thesis, two low power RF communication protocols, SimpliciTI and ZigBee, have been explored and compared for the pipeline monitoring systems. A C# based Graphical Universal Interface was designed to interact with the monitoring system.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

IEEE	Institute of Electrical and Electronics Engineers
ISM	Industrial Scientific and Medical
SRD	Short Range Devices
MAC	Medium Access Control
MiWi	Microchip Wireless
WHART	Wireless Highway Addressable Remote Transducer
UWB	Ultra Wideband
TI	Texas Instruments Inc.
USCI	Universal Serial Peripheral Interface
UART	Universal Asynchronous Receiver/Transmitter
SPI	Serial Peripheral Interface
I^2C	Inter-Integrated Receiver
IrDA	Infrared Data Association
ADC	Analog to Digital Converter
DAC	Digital to Analog converter
CCS	Code Composer Studio
LF	Low Frequency
HF	High Frequency
LED	Light Emitting diode

CHAPTER 1

INTRODUCTION TO RESEARCH

1.1 Necessity for Subsea Monitoring systems

One of the crucial steps in the oil and gas industries is extracting the oil from seabed underwater to the platform atop the sea through steel pipelines. These pipeline systems range from small diameter, small throughput pipelines systems to large diameter, and large throughput systems. Pipelines also play a predominant role in transporting oil from oil reservoirs located in coastal regions to refineries and from there to redistribution centers to petty regional supply centers. The government (in many countries) regulates the oil pipeline industry. In the U.S, the federal government regulates the overall pipeline system integrity through the U.S. Department of Transportation Office of Pipeline Safety.

The major threats to underwater pipelines are hurricanes, earthquakes, third party interferences, pipeline corrosion etc. [14]. Oil Exploration companies constantly thrive to build a risk free pipeline system with full-scale returns. In particular, offshore oil exploration is more hazard prone and expensive, requiring voluminous research and technically qualified professionals with innovative and modern equipment in the exploration and production steps. Continuous monitoring of the motion of the pipeline, its corrosion level and temperature profile are a major requirement for the safety and integrity of the pipeline system, and also, for the personnel working underwater or atop the sea. Appropriate and effective monitoring systems aim at enhancing the operator's safety and optimizing the maintenance of the pipeline system. This would aid in protecting the environment, identifying potential threats to the pipeline, and securing continuous and cost effective product delivery.

Starting from offshore to onshore, the pipeline system is divided into various monitoring zones, each having specific monitoring equipment and requirements [15]. A view of the various monitoring zones is shown in Figure 1-1.

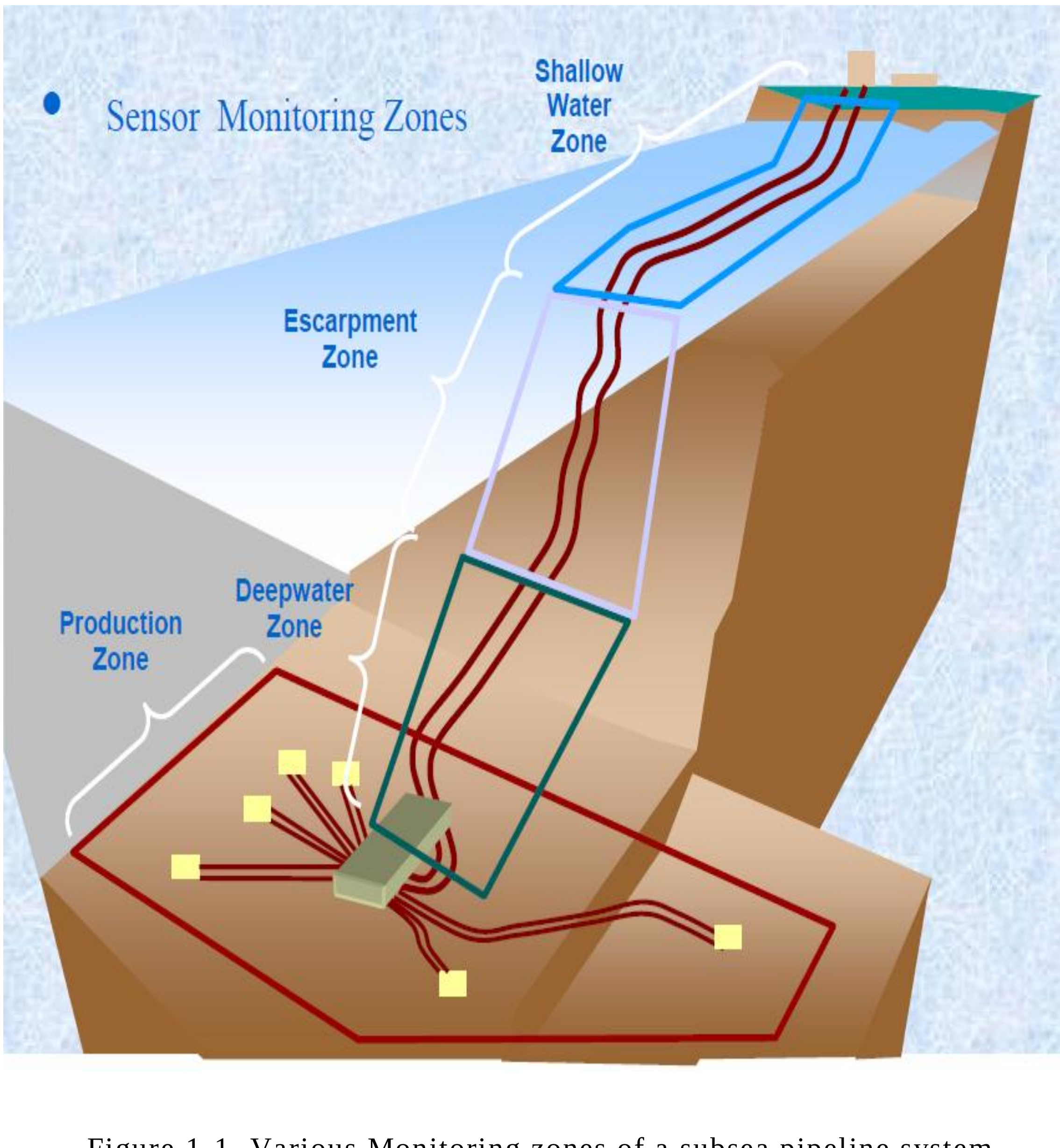


Figure 1-1. Various Monitoring zones of a subsea pipeline system

1.2 Ideal Monitoring System

The essential requirements for an ideal offshore pipeline integrity monitoring system are as follows:

- **Uninterrupted monitoring system** that gives details of pipe stress, leakage of oil, movement of the pipeline, movement of ground due to seismic activity or landslides, etc. along the pipeline length, regardless of weather and pipeline conditions.

- **High sensitivity** to recognize any sort of threats to the pipeline and responding to the same with a good alarm system. Real time information about the pipeline's health would help operators make appropriate decisions based on actual structural conditions and not on assumptions.

- **Third Party Interference (TPI)** like human movement and vehicular intrusions, underwater habitants like fish, whales, sharks, etc. should be detected within the pipeline corridor. These kinds of individual events should be identified and categorized according to the level of alert and reported electronically to the pipeline security operators.

- **Remote Condition Monitoring** of the wear and tear of pumps that keep the oil in motion along the length of pipeline. Such monitoring is usually based on the principle of changes in the acoustic pitches measured by acoustic sensors.

- **Avoid false alarms.**

1.3 Methods for Uninterrupted monitoring system

Different methods of uninterrupted monitoring systems are:

- **Digital pipeline integrity with help of an embedded system using sensors with transmission alternatives: electromagnetics, optics and acoustics, [18]** can be used to detect the temperature, pressure, motion, displacement and orientation of pipes. The information detected can be sent to the central control room for further precautionary measures to be taken in case of emergency.

- **Fiber Optic Sensing** is a promising field which permits collection of much of the information needed by the pipeline operators. Extensive research is being done in this field.

- **Periodical checking** by personnel or remotely controlled vehicles (AUV or ROV) actually going under water to visually check the structural integrity throughout the length of the pipelines [14]. This is currently the most common approach. While useful, this is neither real time nor always possible and cost effective.

1.4 Objective of the Research

This thesis is part of a deep sea pipeline monitoring system under development in my advisor's laboratory. The monitoring system is based on wireless optical communication in deep waters for transfer of data from a sensor attached to a pipeline to a ROV. The system records the temperature, the motion, and magnetic fields surrounding a pipeline system. My work entails the addition of Low Power RF communication to the pipeline monitoring system.

This new capability significantly simplifies the hardware and software subsystems development and testing. It aids in testing a full system in the laboratory and the pre-deployment phase carried at the installation site (offshore). The low power RF monitoring system uses RF transceivers to collect the sensor information. The RF communication is based on two low power RF protocols, SimpliciTI, and ZigBee, which are explained in subsequent chapters. A comparative evaluation between the radio transceivers supporting the two protocols in terms of power consumption, data rate, packet loss, and latency was performed. A C# GUI was developed to display the system's information and collected sensor data.

1.5 Need for a dual communication system

To understand the need for a dual communication system in our project, first it is necessary to get acquainted with the various stages an individual pipe undergoes until integration into a lengthy pipeline. Pipes purchased from the manufacturers are lengths of 40-80ft. Pipes are initially coated with fusion-bonded epoxy and insulating material (e.g., Polyurethane, polypropylene) across their length, but for ~18inch at both ends that is left bare to allow for welding.

Following welding, the sensor monitoring unit is strapped to the pipe and remaining pipe surface is fully covered with insulating materials. Through this installation process, the weld area is subjected to various constraints such a high mechanical strains and large temperature gradients. As such, if a sensor platform is to be installed within that environment, it is critical that post installation full functionality testing is performed prior to deployment under water. As this step occurs under ambient light (on a barge or ship deck), only a visible blind wireless communication system, such as RF, can be utilized.

Usually once the pipes are joined and welded together, they are rolled on to a reel barge and transported to the deployment site via large vessels. While on the reel, these pipes undergo large strain values, sometimes beyond that is allowed by the design rules. An RF multi node capable wireless system, such as the one developed in this thesis, can be utilized to collect in real time the strain profile throughout this step for both quality control and failure analysis.

1.6 Organization of Research

Chapter 2 starts with the description of the project I have been a part of and lays the foundation of my research topic. It also provides information on the sensors I selected to use for the monitoring system.

Chapter 3 discusses the various modern Low Power RF network protocols and technologies. It throws light on the protocols shortlisted to use for our monitoring system. It also discusses the SimpliciTI and ZigBee protocols, the networks they are designed for and which are considered better suited for our particular application.

Chapter 4 describes the hardware and software requirements for each of these protocols. It also discusses the integration of the sensors with the hardware and the protocol built for communicating with the sensors. It also explains the design of the GUI and its functionality.

Chapter 5 gives a comparison between the two chosen protocols in terms of power requirements, packet loss, and range of communication. It describes a good fit from the two protocols selected. It also gives insight into future work.

Chapter 6 gives a summary of the work done.

CHAPTER 2

MONITORING SYSTEM BASED ON OPTICAL COMMUNICATION

In chapter 1, the need for subsea oil pipeline monitoring system was discussed. Also, the organization of this work was presented. In this chapter, the major steps in building a prototype pipeline monitoring system utilizing optical communication is described.

2.1 Introduction to Embedded systems

The evolution of embedded systems dates back to the 1960s, the era of numerous inventions. The advent of transistors around 1930s completely revolutionized the electronics industry. These transistors played the key role in the rapid advancement of the embedded systems field. An embedded system, a.k.a embedded computer, is a computer designed to perform categorical and/or definite functions for the purpose of monitoring and/or control.

Embedded systems are categorized based on the processors they have, namely microprocessors and microcontrollers. The former is designed to be a general-purpose central processing unit (CPU) like those used in a PC (Personal Computer). It is a general-purpose device which when integrated with other external circuitry turns it into a microcomputer. A few examples of microprocessors are TMS1000, Intel4004/8088, Motorola 6809, Zilog Z80, Motorola MC68010/68020/68040 etc. In a microcontroller, all the components required for the controller are integrated into a single chip. The microcontroller is a specialized form of microprocessor found in innumerable electronics systems.

2.2 Monitoring system Design

The need for uninterrupted monitoring of underwater pipelines requires a stable system which continuously monitors their condition at regular intervals. Figure 2-1 depicts the monitoring system designed for our purpose which is an embedded system with required circuitry. The following steps were carried in designing the embedded monitoring system:

1) Embedded System Development

2) Electronics

3) Design of algorithm, Software & Raw Data Processing

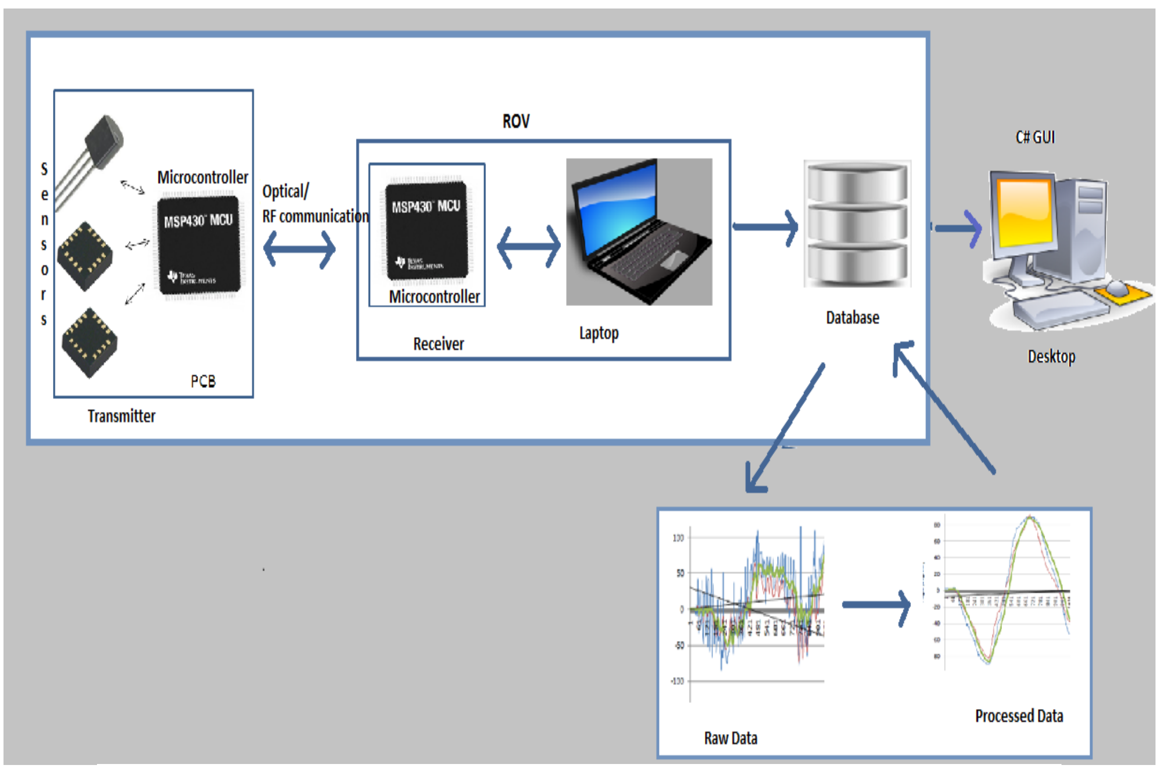


Figure 2-1. Monitoring System Design Steps

2.2.1 Embedded System Development

The whole embedded system development is mainly divided into two parts as follows:

I. Retrieve data from the sensor via microcontroller. Transfer the sensor data via optical/RF communication to the MCU connected to a laptop or to a processing platform on an ROV/remote site for further processing.

II. Build a GUI to collect raw data and save in database. Process the raw data to interpret underwater pipe movements and display them graphically.

2.2.2 Electronics

The electronics required for the monitoring system should fulfill a basic set of requirements as described below.

- The microcontroller selected must be least expensive meeting our design requirements. It must consume low power and must have the ability to communicate with the sensors at desired operating frequencies.

- The accelerometer/magnetometer sensor must be able to give digital output so that the output can be easily processed when received at the user's end. It should withstand high temperatures and pressures along with high operating frequencies, good dynamic range of measurement and high sensitivity. The sensor should be able to send an interrupt when an event of interest has occurred on the underwater pipeline. The components of monitoring system must be easily programmable. The sensors and the microcontroller selected for building the monitoring system are discussed in the rest of the chapter.

2.2.2.1 Accelerometer Sensor Selection

An accelerometer is a sensor that measures the deviation from free-fall, but not acceleration due to gravity. The MMA8451Q accelerometer sensor from Freescale is selected after studying various sensors from different companies. Figure 2-2 shows the application diagram of MA8451Q. MMA8451Q has several features like static orientation detection, free-fall detection, motion detection, and vibration detection. It communicates with microcontrollers via Inter-Integrated Receiver (I^2C) protocol [26].

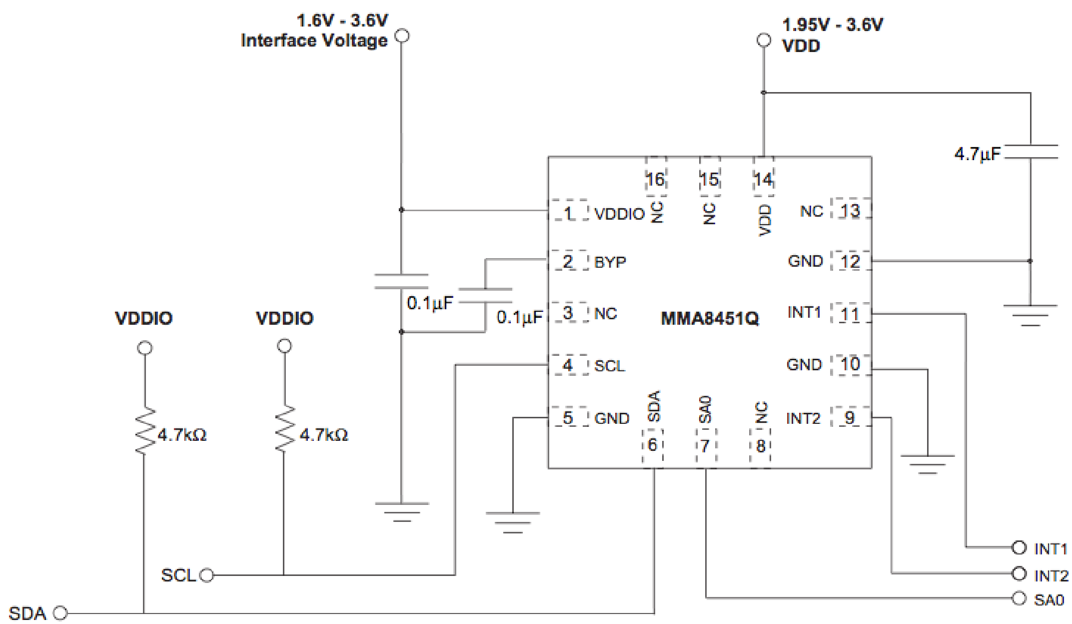


Figure 2-2. Application diagram of MMA8451Q by Freescale

2.2.2.2 Magnetometer sensor selection

A magnetometer sensor is used to measure the strength and/or direction of the magnetic fields in the vicinity of the sensor. The earth's magnetic field keeps changing with place and time due to various reasons, namely, natural time-dependent magnetic anomalies, natural space-dependent magnetic anomalies, and artificial (man-made) magnetic anomalies. MAG3110 magnetometer sensor from Freescale is selected after studying various sensors from different companies [25]. Pin diagrams and measurement coordinate system are shown in Figure 2-3 below.

Figure 2-3. Pin diagram & Measurement coordinate system of MAG3110

2.2.2.3 Microcontroller Selection

For testing and getting acquainted with the working of sensors and MCU's, MSP430FR5739 was considered because it has an FRAM memory and has the capability to communicate with the sensors. Pin designation on MSP430FR5739 is shown in Figure 2-4 below. The microcontroller can be programmed via IAR workbench or Code Composer Studio (CCS).

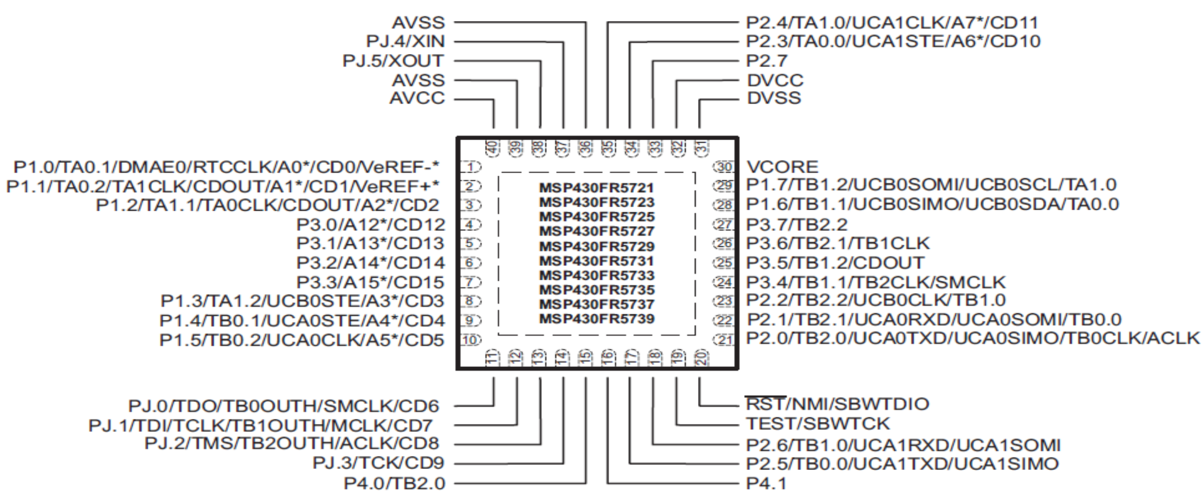


Figure 2-4. Pin Diagram of MSP430FR5739

2.2.2.4 Experimenter's boards with MSP430FR5739 & sensors MAG3110 & MMA8451Q.

The Texas Instruments (TI) demonstration board (Figure 2-5) with MSP430FR5739, used to demonstrate the basic pipeline monitoring system, is shown below. The TI board also comes with an accelerometer sensor ADXL335 and a temperature sensor [28]. There are two demonstration boards from two different vendors in the market which contains both accelerometer and magnetometer sensors on a single board. They are Freescale and ST Microelectronics. **Freescale** board **LFSTBEB3110** [32] contains **MAG3110** and **MMA8451Q** on it. The image of the Freescale board is shown in Figure 2-6.

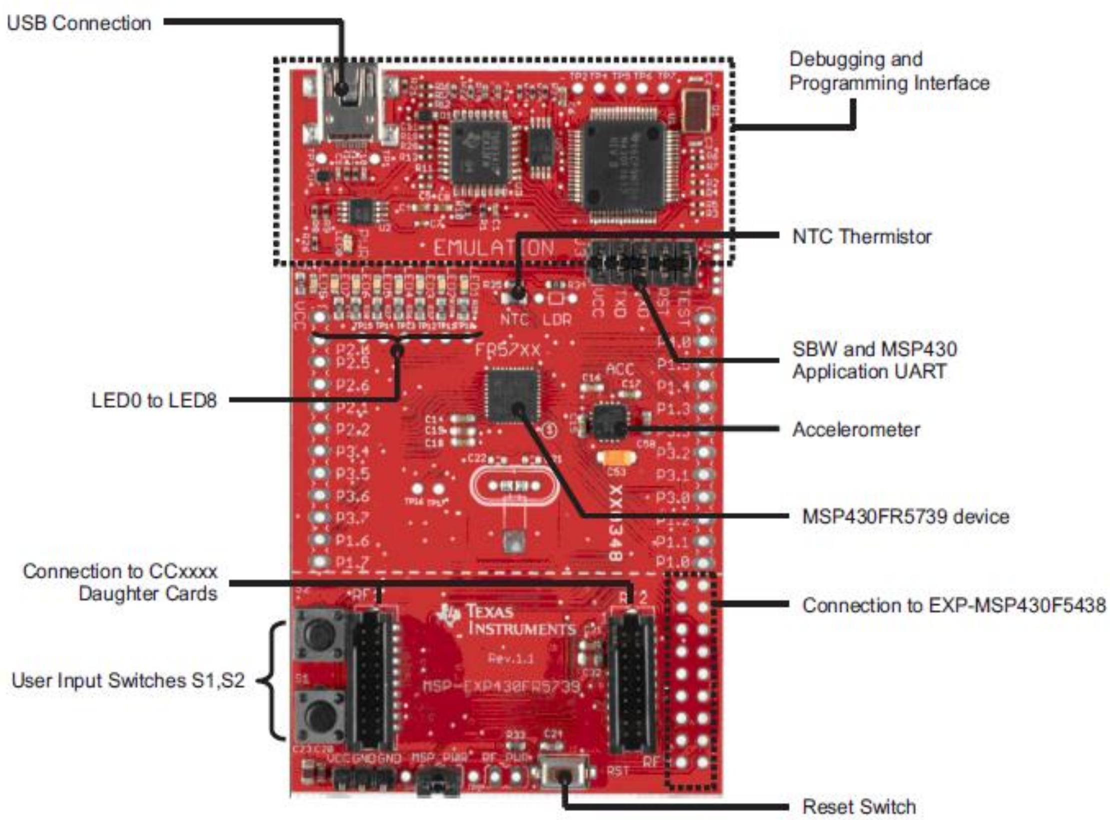


Figure 2-5. TI Evaluation Board for MSP430FR5739

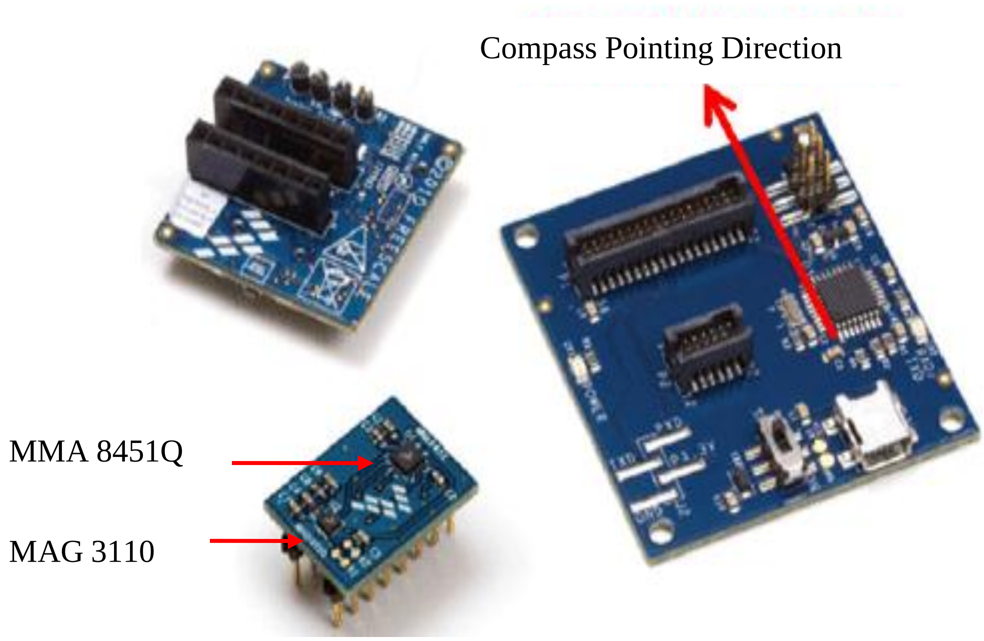


Figure 2-6. Freescale Board with sensors MAG3110 and MMA8451Q

The basic pin diagram of the LFSTBEB3110, the board that contains the fusion of MAG3110 and MMA8451Q can be found in [31]. Note: Vdd and Vddio =2.5Vdc, and the I^2C pull-up resistors are off board. Flow diagram of general write to selected sensors is shown in Figure 2-7.

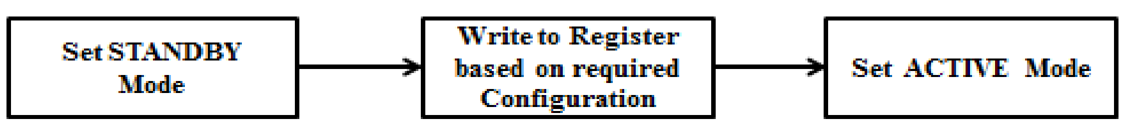


Figure 2-7. General write to the sensors MAG3110 and MMA8451Q

2.2.2.5 Pin Description and flowcharts of the system.

The pin diagram of the whole Integrated System of the basic and simple monitoring system after microcontroller, accelerometer and magnetometer are selected, is shown in Figure 2-8 and Figure 2-9.

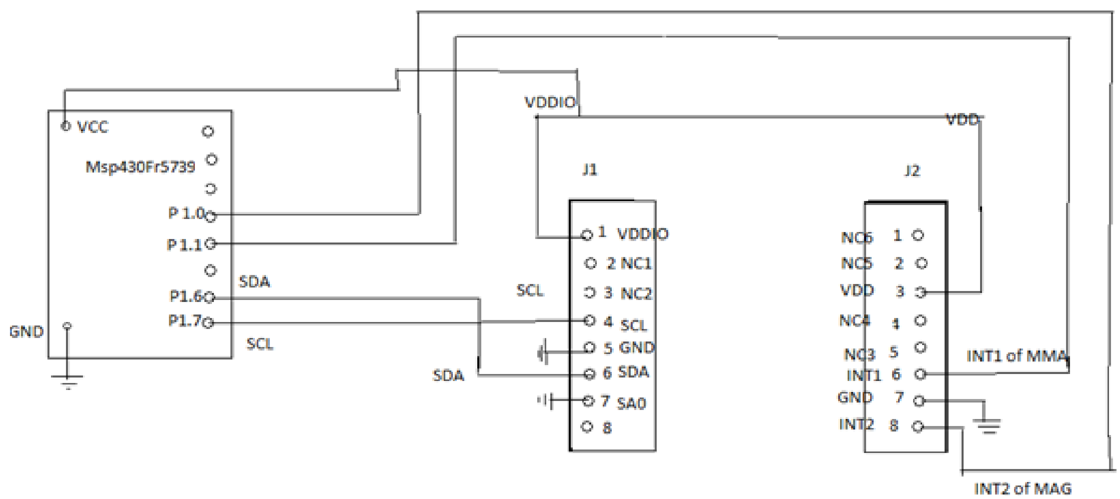


Figure 2-8. Pin Connections for TI board with MSP430Fr5739 and free scale Board

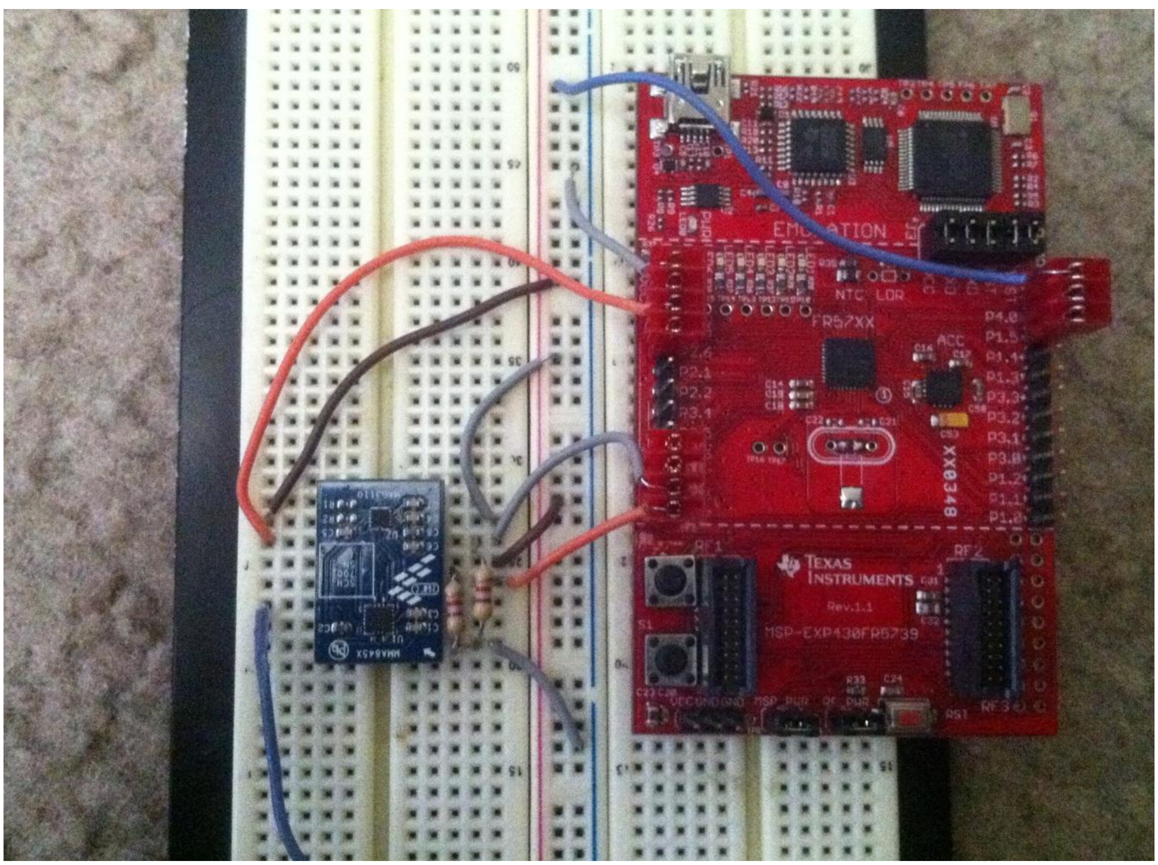


Figure 2-9. Pin connections between MSP430FR5739 and LFSTBEB3110

2.2.3 Design of algorithm, Software & Raw Data Processing

1. The details of the communication between TI Board (MSP430 FR 5739) & Freescale Board (MAG3110 & MMA8451) are given below.

- IAR Workbench, an Integrated Development Environment, is used for developing the code that runs the communication between TI Board and Freescale Board.
- The protocol for communication between the two boards is I2C – data transfer at rate of 100 KHz or a software emulator for I^2C – data transfer at rate of 1MHz.
- Once the communication is successful, we can get raw data from MAG & MMA.
- Accelerometer sensor provides 14-bit wide data on all three axes stored in 2 bytes in 2's complement form [6]. Say the readings from accelerometer are Ax, Ay and Az.
- The sampling rate and type of interrupts to which accelerometer responds can be programmed.
- Proper high pass and low pass filtering should be applied by software to reduce the noise [20].
- Magnetometer has 16-bit wide data on all the three axis stored in 2 bytes in 2's complement form. Say the readings from magnetometer are Bx, By and Bz [7].
- The temperature is measured through the temperature sensor on the microcontroller.

2. For testing purposes, MCU is connected to the computer via serial port and communication between the microcontroller and the laptop is established via Universal Asynchronous Receiver/Transmitter (UART) protocol for monitoring system.

- UART communication is done at 9600 baud rate (bits/sec).
- The data is sent in series of bytes, accelerometer data is sent first followed by the magnetometer data. The temperature sensor can be programmed to measure temperature every ten seconds or every minute. The data sent over UART is decoded, interpreted and processed using a C# GUI.

3. Data transfer to the PC.

- Process the values accordingly and convert the accelerometer values in terms of 'mg' [6] and magnetometer values in terms of 'μT' [7].
- The received raw data must be processed to make them meaningful and to visualize the movement of the pipeline [15] [20]. This part was achieved by another team member on this project.
- The processed data is used and graphically represented in the XY and Z plane in two different graphs. C# is used for post processing of raw data.

4. Good compression techniques should be applied for storing the data on the microcontroller if the data is not collected very often.

The testing of the sensor was done using MSP430FR739. The original optical communication was designed using MSP430F2274. MSP430F2274 is a mixed signal microcontroller with 32KB and 256B of Flash memory and 1KB of RAM memory [37]. The schematic of the optical communication could not be added to this work as it is proprietary property. The pin diagram of MSP430F2274 is shown in Figure 2-10.

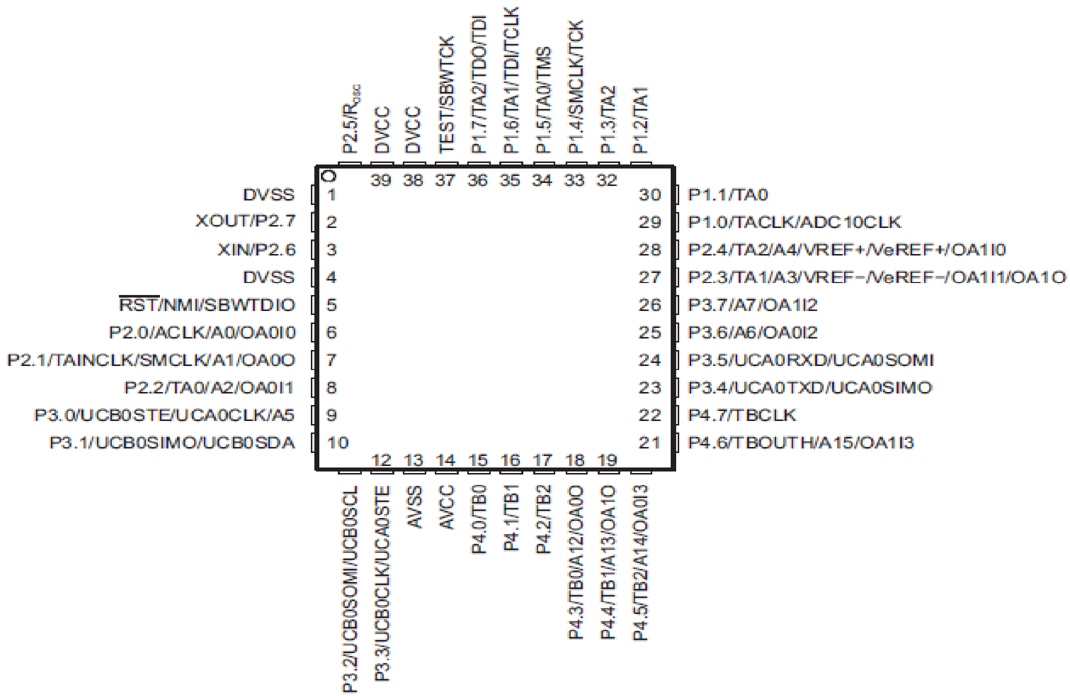


Figure 2-10. Device Pinout of MSP430 F2274

2.2.4 Energy harvesting system to meet Power Requirements

Energy harvesting system was designed to meet the power requirements of the microcontroller and communication functions. The harvesting system utilizes the difference of temperature between the fluid flowing in the pipe and the surrounding atmosphere to generate power. The thermoelectric generator used for energy harvesting system is a Seebeck thermoelectric generator [21]. The product specification sheet can be found in the website [21]. An image of a Seebeck thermoelectric generator is shown in Figure 2-11.

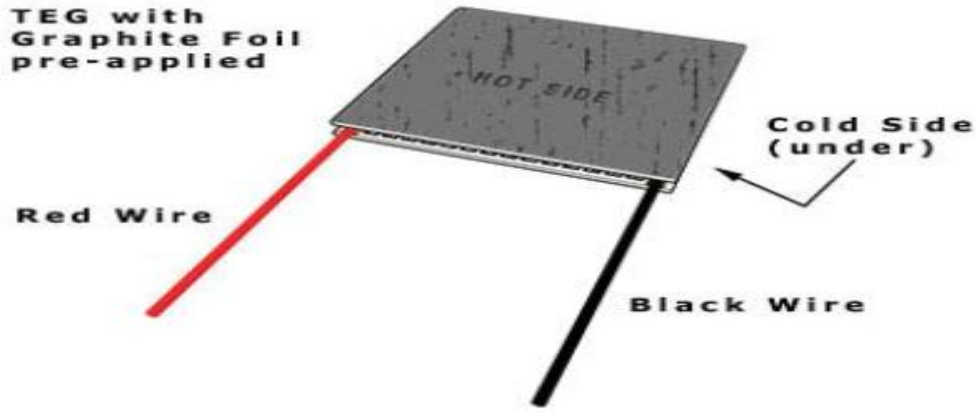


Figure 2-11. Seebeck TEG

2.3 Mode of communication

The monitoring system and ROV are connected through a photo detector and a light emitting diode link. The system connected to the MCU on ROV communicates with sensor units on pipelines by sending message via optical light. A green Light Emitting Diode (LED) is used for communication, as green light can travel in water for longer distance without much attenuation. Also, optical communication was achieved (without data loss) up to a distance of 10ft.

Our sensor unit with MSP430F2274 requires around 310-320 mA of current while communicating optically and around 7.5 mA while responding to interrupts from sensors. Our energy harvesting system can supply current during interrupts and sleep mode of the sensor unit. The rest of the current is supplied by 3.6V Li-ion, 900mAhr rated battery.

This chapter summarizes the monitoring system that utilizes optical communication for transfer of sensor data. It also discusses the energy harvesting system utilized to supply power required by the sensor unit.

CHAPTER 3

WIRELESS SENSOR NETWORKS & PROTOCOLS

The previous chapter described the wireless monitoring system using optical communication. The microcontrollers, sensors, and energy harvesting system used in the monitoring system were discussed. This chapter gives an introduction to wireless sensor networks and the selected low power RF communication protocols that are utilized in our monitoring system.

3.1Introduction to Wireless Sensor Networks

A typical sensor network is an alliance of several hundreds of sensor nodes with each node capable of sensing, collecting data, computing, and communicating with the external world [23]. A Wireless Sensor Network (WSN) consists of a typical sensor network with each node equipped with a radio transceiver along with an internal/external antenna, a microcontroller, electronic circuitry for communicating with the sensors, and an energy harvesting source or a battery. WSN's were initially built for military applications; lately, they are designed for several applications like [5] [30]: home automation [11], surveillance [19], vehicular tracking [8], structural monitoring [39], etc. Figure 3-1 shows an example of a wireless sensor network that combines both functions of collecting sensor information and processing of the collected data. As seen from Figure 3-1 the sensors communicate with each other and with the base station to transfer the data to the remote processing center via internet [38].

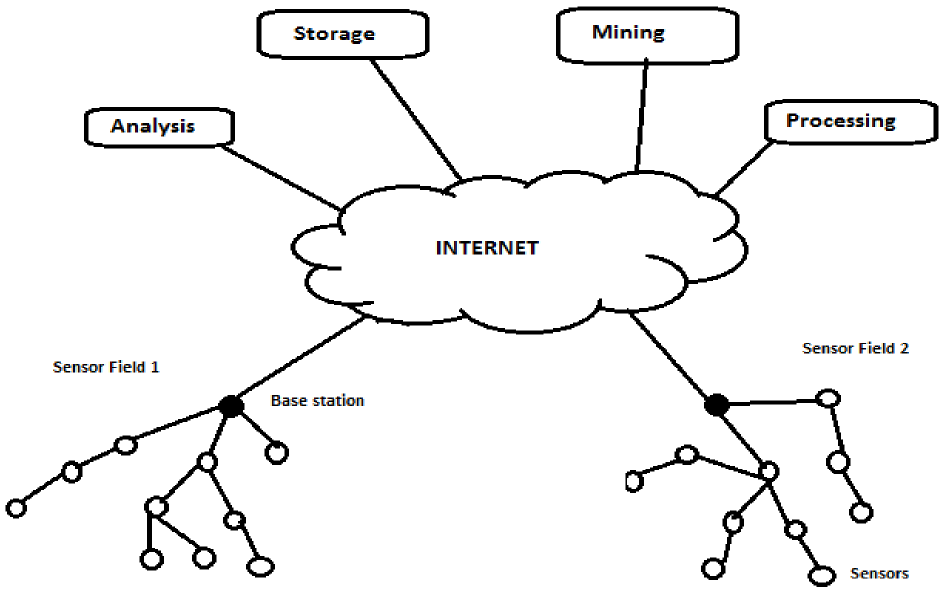


Figure 3-1. Wireless Sensor Networks

3.2 Constraints for implementing WSN's

WSN's, like any other distributed system network, are subjected to several constraints dependent on their working environment. The basic constraints that affect the WSN's design and protocol are discussed below.

3.2.1 Energy Efficiency

WSN's can be powered by non-rechargeable/rechargeable battery packs or energy harvesting systems. Some critical environments demand the sensors to monitor and report continuously which consume energy continuously while other does not. Hence, one of the major constraints in implementing WSN's is energy efficiency [38].

3.2.2 Self-Management

WSN's are deployed in stochastic or deterministic manner depending on the working environment [12]. The deployed sensor node should be capable of adapting to the surroundings, and maintaining maximum coverage and connectivity. Sensors nodes should also possess self-protection against alien attacks.

3.2.3Range of Communication

Radio frequency waves suffer attenuation as they traverse through the medium and passes over obstacles. The relationship between transmitted power P_t, received power P_r and the distance between transmitters d is

$$P_r \propto \frac{P_t}{d^2}. \tag{3-1}$$

From Equation 3-1 it is understood that P_r is inversely proportional to the square of the distance of the source signal. As a consequence, the transmitted power should increase as the distance between the sensor node and base station increases. Increased distance between the nodes demands for communication over multi-hop or router environment to increase the transmission range. As the number of nodes increase in a network, they should be programmed to work periodically or wake on demand to conserve energy if they are deployed in critical environments [38].

3.2.4 Design and Security Constraints

The necessity to build sensor nodes that are cost-effective, easily portable, good processing speeds, storage capacities, etc. underlies numerous constraints in designing the sensor node. These factors also affect the software design at various stages. Sometimes, some sensitive information carried over wireless network may be subjected to attack and the service could be disrupted. If the sensor nodes are placed in a mobile environment it should be capable of adapting to the environment by continuously studying the environment and updating itself [38]. Hence the design of the wireless sensors network is impacted by various factors

3.3 Basics of an RF System

Radio Frequencies (RF) range from few KHz to 300 GHz including microwave frequencies. Industrial Scientific and Medical (ISM) / Short Range Devices (SRD) frequency bands are unlicensed bands (Table 3-1) in which the low power wireless RF protocols were built in different countries. Figure 3-2 shows the electromagnetic spectrum and the range of frequencies we use for RF communication.

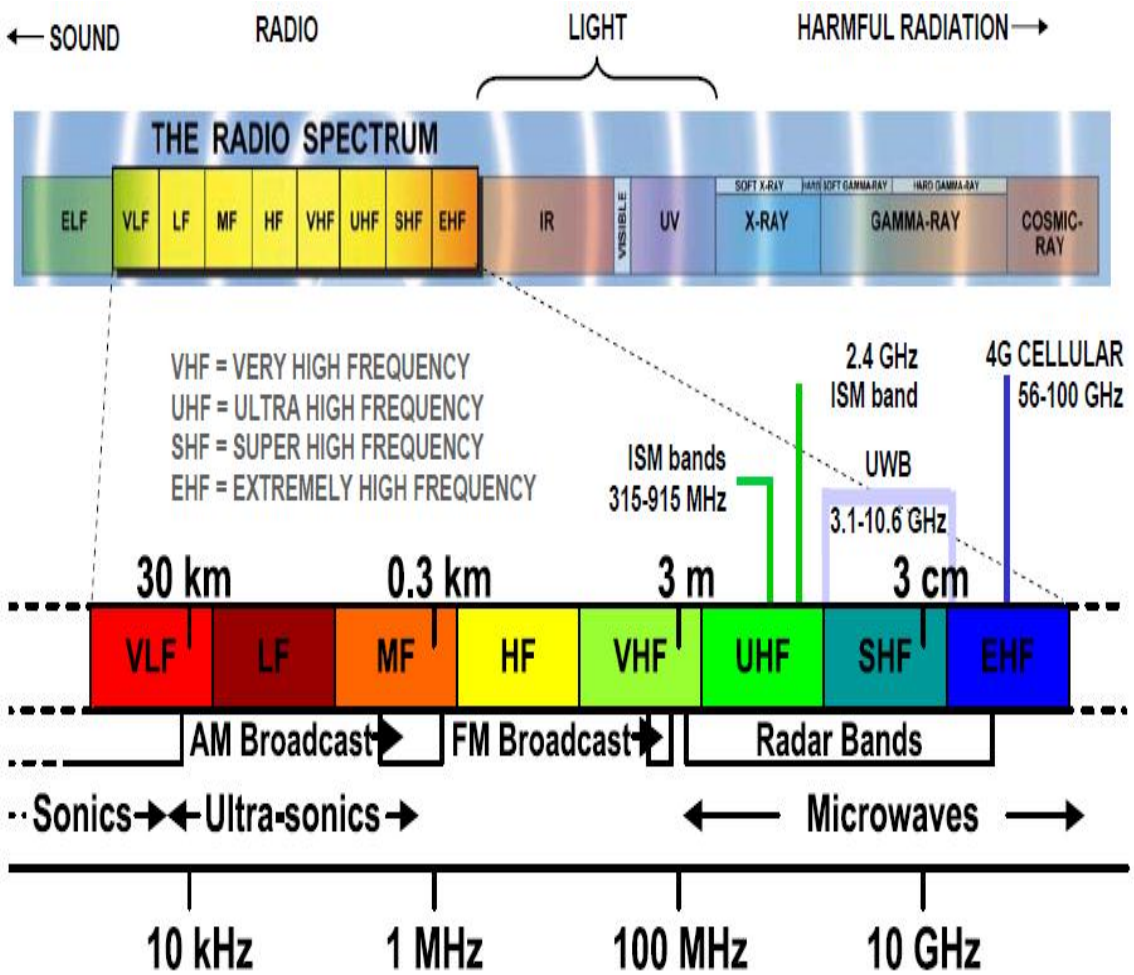


Figure 3-2. Electromagnetic Spectrum (Source 2006 Texas Instruments Inc.)

Table 3-1. Range of communication of different frequencies in the RF frequency range

Frequency f	Wavelength λ	Band	Description
30-300 Hz	10^4 - 10^3 Km	ELF	Extremely Low frequency
300-3000Hz	10^3 - 10^2 Km	VF	Voice Frequency
3-30 KHz	100 - 10 Km	VLF	Very Low Frequency
0.3 – 3 MHz	10 - 1 Km	LF	Low Frequency
3 – 30 MHz	1 - 0.1 Km	MF	Medium frequency
30 - 300 MHz	10 - 1 m	VHF	Very High Frequency
300 - 3000 MHz	100 - 10 cm	UHF	Ultra-high Frequency
3 - 30 GHz	10 - 1 cm	SHF	Super high Frequency
30 – 300 GHz	10 – 1 mm	EHF	Extremely High Frequency

3.3.1 RF Communication Systems

Different RF Communication Systems are as follows:

- A simplex RF system allows only one-way communication from a transmitter to receiver. Examples: FM radio, TV.
- A half-duplex RF system has a communication system in which each node in the network can transmit and receive, but not simultaneously. Examples: Wireless Keyboard, Walkie – talkie.
- A full-Duplex RF system is a system where each node transmits and receives simultaneously. Examples: Satellite Communication, Cellular Phones.

3.3.2 Basic building blocks of an RF System

Figure 3-3 describes the basic building blocks of RF communication system. The Low Frequency (LF) Information signal is the required signal to be sent over the medium and the High Frequency (HF) Carrier is the frequency signal over which we would like to superimpose the LF signal. A modulator superimposes the LF signal over the HF signal using techniques such as Amplitude Modulation (AM) or Frequency Modulation (FM) or Phase Modulation (PM). An amplifier is used to improve the signal level or power level.

A demodulator on the receiver end separates the information signal from the carrier wave.

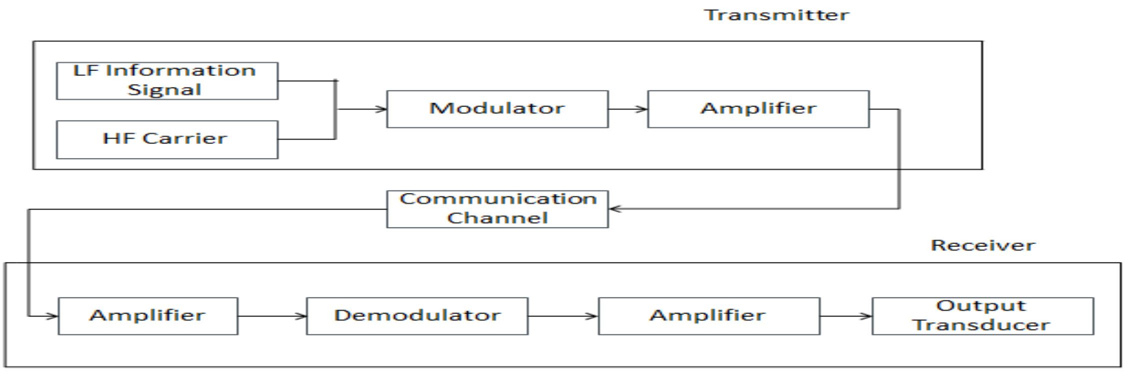


Figure 3-3. Basic Building blocks of an RF communication system

3.4 Low Power Wireless RF protocol

Several Low Power Wireless RF protocols have been built based on Institute of Electrical and Electronics Engineers (IEEE) 802.11x standards (a set of standards for implementing local area wireless networks in 2.4, 3.6 and 5 GHz frequency bands). Examples: Wi-Fi, Bluetooth, ZigBee, 802.15.4 Medium Access Control (MAC). Some proprietary wireless standards are 6LoWPan, SimpliciTI, Microchip Wireless (MiWi), and Wireless Highway Addressable Remote Transducer (WHART). Figure 3-4 shows the short range wireless communication protocols with the distance vs date rate. Table 3-2 shows the comparison of various protocol standards available in today's market.

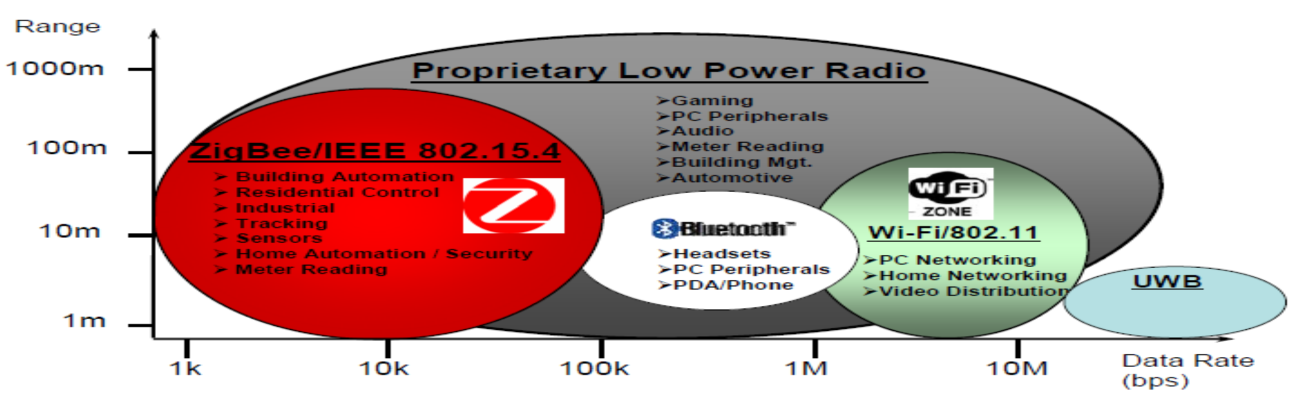


Figure 3-4. Most Common Short Range Wireless Communication Systems

Table 3-2. Comparison of various Low Power RF communication Protocols

Wireless Standard	**Power**	**Frequency**	**Data Rates**	**Standards Based**	**Transmission Range**	**Interop with 3rd parties**	**Cost/Complexity**
Bluetooth	Medium	2.4 GHz	1–3 Mbps	IEEE 802.11	1-100m	Yes	Medium
Wi-Fi	Medium	2.4 GHz	10-100Mbps	IEEE 802.11	100 – several Km	Yes	Medium
ZigBee	Low	2.4GHz	20-250 Kbps	IEEE 802.15.4	10-100m	Yes	Medium
TI MAC	Medium	2.4 GHz	20-250 Kbps	IEEE 802.15.4	10 -100m	Yes	Medium
SimpliciTI	Low	1 GHz & 2.4 GHz	20-250 Kbps	TI proprietary	10-20m	No	Low
6LoWPan	Low	868/915 MHz	20-250 Kbps	IEEE 802.15.4	10-100m	Yes	Medium

After investigating various low power protocols in terms of hardware, software, and cost/ complexity requirements, I decided to choose two standards: TI proprietary standard process SimpliciTI and IEEE proprietary standard ZigBee. The rest of the chapter deals with the protocols, software, and hardware required for these two standards.

3.5 RF Network Basics &Topologies

All wired/wireless network protocols can be conceptually described by what is referred to as the Network Open Systems Interconnect (OSI) Basic Reference Model, as seen in Figure 3-5 [16].

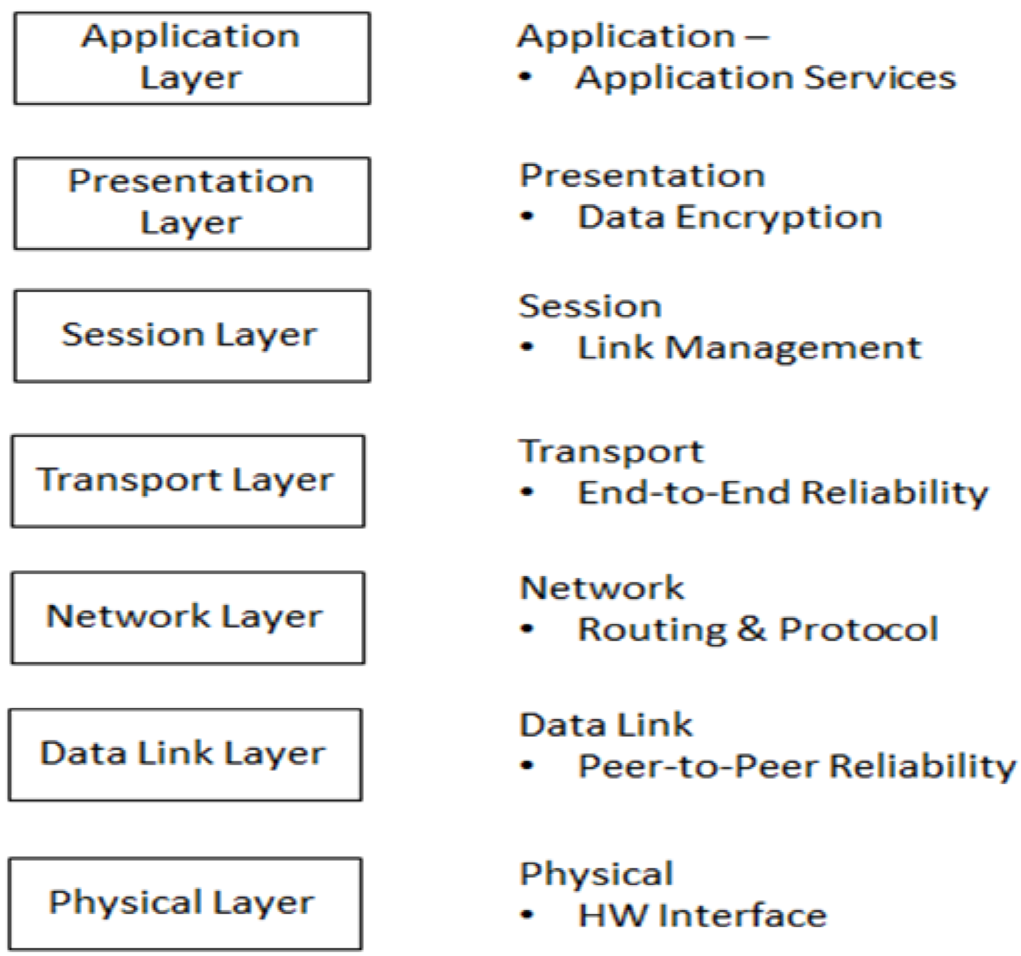


Figure 3-5. Elements of a network

The model divides the components of network protocol implementation into software layers. Each layer communicates with the layer above or below it [16]. The four most common network topologies are shown in Figure 3-6.

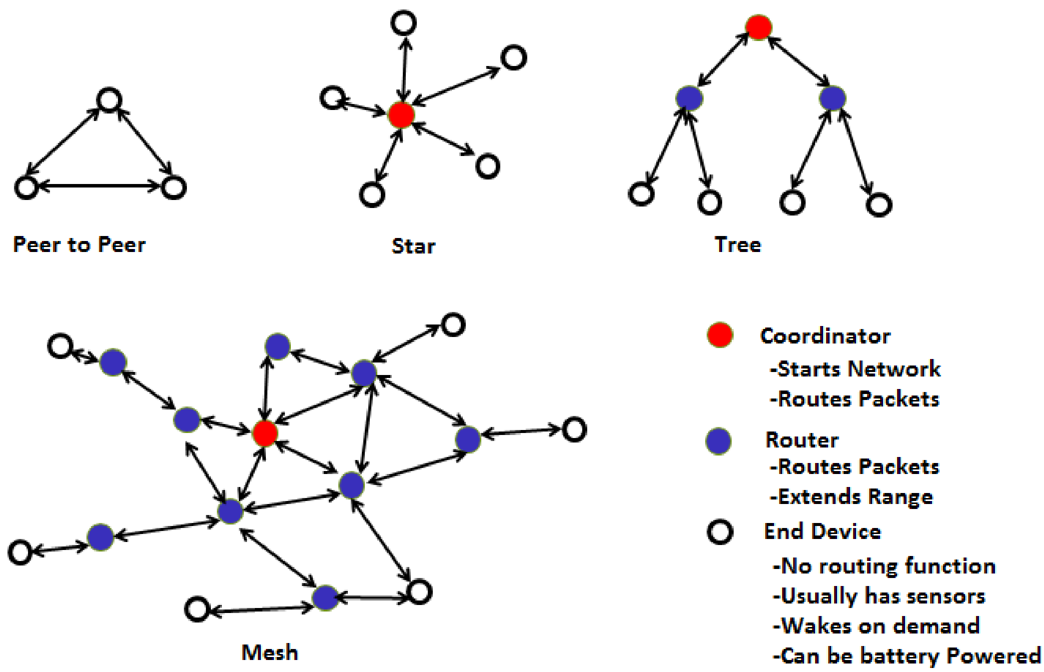


Figure 3-6. Low Power Network Topologies

The Figure 3-6 shows the following topologies:

- A peer–to–peer network topology allows bidirectional or unidirectional data transfer between each node in the network when they are in range of each other.
- In tree topology, the coordinator is the parent node which can be connected to other routers and other end devices. Routers generally extend the range of the network.
- In star topology, the coordinator is the parent node and all other nodes (called end devices) communicate with each other via the coordinator. The disadvantage of this method is there is no alternative link between end devices if the router fails.

- A mesh network is an extended tree topology with the router layer extending downwards with each node having link with every other node in the network. Mesh networks have much efficient propagation because alternative routes exist if one of the routes fails through route – discovery feature.

3.6 Low Power RF Protocol – SimpliciTI

SimpliciTI is a connection based peer-to-peer proprietary low power RF network protocol targeted for small RF networks from TI. This protocol is the simplest protocol to start educating oneself with wireless networking. The network topologies supported by SimpliciTI are peer-to-peer, star with/without repeaters, and the tree network with just one layer of repeaters. SimpliciTI allows devices of three types, namely, Access Point, Repeater and End Device [34]. These devices are explained below.

- Access Point is primarily concerned with network management in terms of membership permissions, linking permissions, and security tokens. TI also supports functionalities like store and forward support for sleeping End devices. It acts as data hub in star network [34]. It can also support end Device functionality i.e., it can by itself instantiate sensors or actuators in the network [33].
- An End Device is a base element of the network which contains sensors or actuators. End Device can be a sleeping device which responds to Access Point on request or keeps transmitting data to Access Point continuously. A peer-to-peer network is strictly made of end devices with repeaters added to them occasionally.
- A Repeater or Range Extender simply replays any frame it receives to improve the range of communication in the network.

Radios supporting SimpliciTI protocol form TI are CC1100/CC2500. CC1100 operates at 1 GHz while CC2500 operates at 2.4 GHz. The code size of the protocol is less than 4Kbytes. Applications of SimpliciTI protocol are found in:

- Industries in smoke detectors, light sensors, occupancy sensors, carbon monoxide sensors.
- Home automation in temperature monitoring, garage door openers, etc.
- Automatic meter readings for gas meters, water meters and e-meters.

3.6.1 Architecture overview of SimpliciTI

The protocol is realized in simple Application Peripheral Interface (API) calls, organized in three layers of the OSI Basic Reference Model as shown in Figure 3-7.

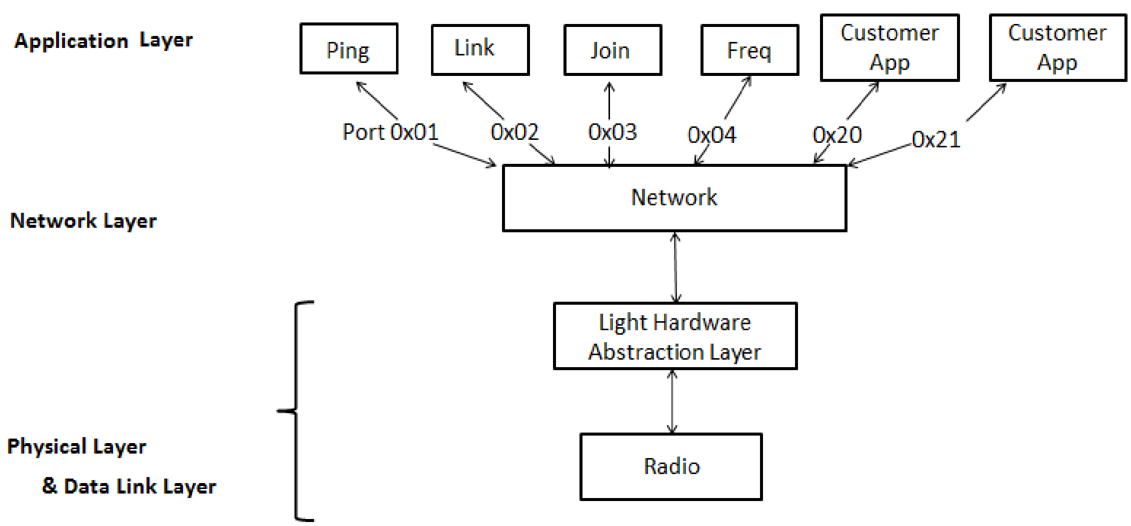


Figure 3-7. SimpliciTI Protocol Layers

3.6.1.1 Application layer

Application layer is where the developer needs to build his application. If required transportation layer can be built as there is no Transport Layer.

3.6.1.2 Network Layer

Network applications like linking, joining and pinging work on a predefined port. The user may intend to change the joining, linking tokens, and ping application according to the requirement. The layer manages the received and transmitted messages to reach their correct destination. The destination is always an application designated by the port number. The network applications, along with corresponding port numbers, are shown in Table 3-3 below.

Table 3-3. Network Management Port list

Application	**Port**	**Description**
Ping	0x01	Just like TCP/IP application, it tries to detect a specific device.
Link	0x02	Used to associate two peers in a network.
Join	0x03	To guard entry to the network in topologies with Aps
Security	0x04	Used to exchange security information such as encryption keys.
Frequency	0x05	Used to perform channel change requests.
Management	0x06	General network management port.

3.6.1.3 Physical & Data Link Layer

Light Hardware Abstraction layer entity called Board Support Package (BSP) is used to abstract the SPI interface from the network layer. BSP supports LEDs and button/switch peripherals attached to the GPIO pins.

3.6.2 Message Acknowledgements

Message Acknowledge is not supported in the basic version of SimpliciTI. As a consequence, it has no accountancy for missing or redundant data. If a success is returned for an acknowledgement request on the transmitter end, it means that only the peer's network layer received the frame, but not the peer application.

3.6.3 Network Discipline

SimpliciTI protocol has a unique procedure in forming the peer-to-peer network or allowing other members to join the network in presence of AP's.

3.6.3.1 Linking

An application's peer-to-peer connection is established via linking. One application waits for link message from the other device with which it should be paired. Additional constraints can be added to link message if an Access Point is present in the network [33].

3.6.3.2 Joining

Joining is the initial step taken by any node in network. The Join message is 4 bytes in length, set by customer during build time. The unique join token ensures no two access points respond to an end device at the same time. The joining is followed by linking to Access Point. The Access Point tracks the type of end device such as whether it is a polling device or not [33].

3.6.3.3 Frequency Agility

The protocol provides a feature for SimpliciTI compliant devices to migrate between different channels if the channel is very noisy [9]. A channel table is provided during build time for the protocol to select among them in case a "noisy channel" event occurs.

3.6.3.4 Security

Message encryption is provided for each message being carried in the network. This option is set by the customer during build time by defining the macro SMPL_SECURE in the configuration file of the program. A detailed description of how the feature works is better given in the Application note [10].

3.6.3.5 Sleeping End Devices

End Devices can either poll the Access Point to check for waiting messages or simply respond to the calls from Access Points when required [33].

3.6.4 Application Programming

IAR Embedded Workbench and Code Composer Studio (CCS) provide the design and development environment for SimplciTI protocol. IAR supports dual chip (MSP430 +Radio) as well as 8051 SoC (System On-Chip) targets, while CCS supports only the dual chip target board. The minimum hardware abstraction layer (BSP) supports only buttons and LED’s, but do not support the timers, UART, LCSs or other on-chip resources. SimplciTI protocol utilizes the flash and RAM of the MCU. Heap memory is not utilized, as it does not support dynamic memory allocation. Stack is used as all memory locations are static [33].

3.6.5 Threading Model

SimpliciTI protocol is designed to work in conjunction with the threading model of the customer application. The protocol provides a general callback capability for received frames, which runs in Interrupt Service Routine (ISR) thread. In case of network applications, the receiving and transmission of frames are done in a same thread [33].

3.6.7 RX/TX Packet Format

The packet or the frame format for the SimplciTI protocol is described in Figure 3-8. The preamble is a fixed number of bytes containing alternating 1's and 0's usually sent at the beginning of the packet followed by a sync word. The detection of the packet at the node can be realized when a sync word or preamble is received. The payload is the maximum number of bytes of data that could be sent in one packet excluding the preamble and sync words. Further chapter describes how the SimplciTI protocol was used for the monitoring system application discussed in chapter 1.

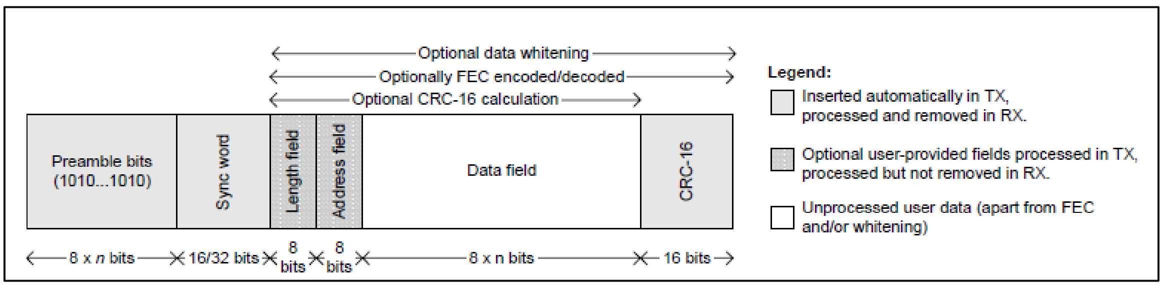


Figure 3-8. SimplciTI frame format

3.7 Low Power RF Protocol - ZigBee

ZigBee is a low power RF protocol designed by the ZigBee Alliance, an organization of companies targeting to build low power RF networking technologies in ISM/SRD bands based on IEEE 802.15.4 standards. ZigBee protocol is designed to work in star, tree, and mesh networks. ZigBee is nest suitable in areas with low data rates and where network nodes sit idle for longer periods of time.

ZigBee Alliance provides profiles with unique profile IDs to ensure application level interoperability. Proprietary profiles are defined by customers. Few standard Alliance Profiles are Home Automation, Smart Energy, Commercial Building Automation, Personal Health and Hospital Care, Vehicle Monitoring, and Agriculture.

The three common devices that work in a ZigBee network are Coordinator, Router, and Reduced function end Devices. Let's see the functionalities of each device in detail.

- A Coordinator mainly initializes the Personal Area Network (PAN) and allows binding and maintains address-table services. It allows end devices and routers to join the network. It routes messages, dynamically repairs routing, and manages security.
- An End Device is basically a sleeping device polling coordinator for message or wake up when requested. It does not route message nor initializes a network. It initially scans to find a network and joins it. It can either talk to a Router or a Coordinator but not to an End Device. Most of the time End Devices are in sleep mode to minimize power consumption and promote long battery life.
- A Router basically talks to a Coordinator and an End Device and routes messages between them with support for secure messaging. It does not initialize a PAN, but allows other devices to join the network. It can buffer messages for sleeping end devices.

3.7.1 ZigBee Architecture

The ZigBee architecture is shown in Figure 3-9. This protocol offers high level functionalities concerned with network structure, message routing, and security [42]. The ZigBee software layer is above the Physical/Data Link Level based on IEEE 802.15.4 standard, which encompasses fundamental principles of ZigBee Networks as ultra-low power consumption, use of unlicensed radio bands, low cost and easy installation [42].

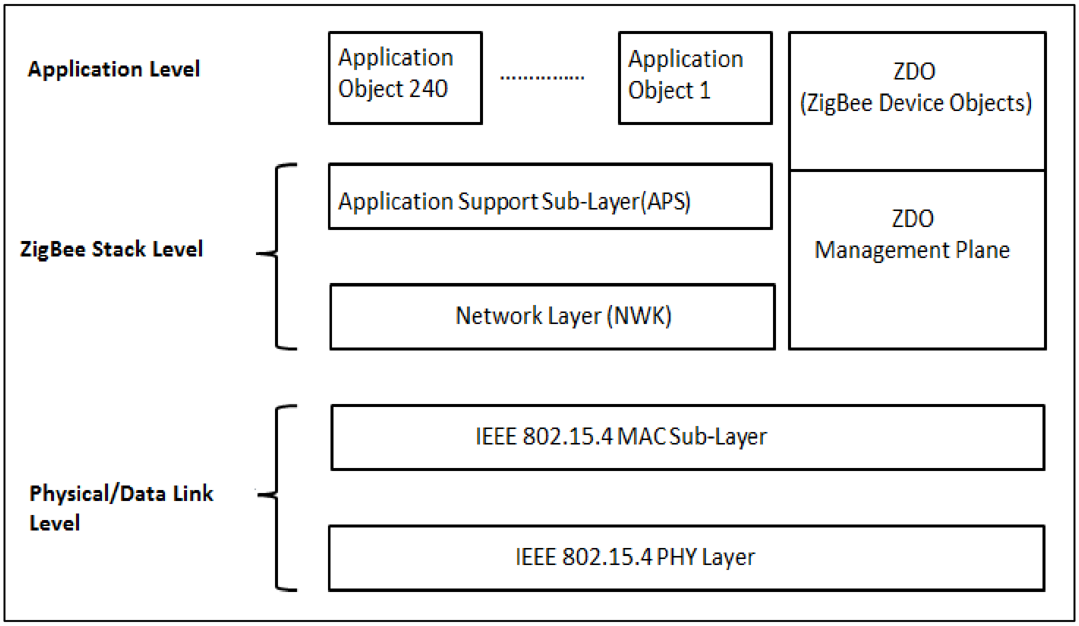


Figure 3-9. Architecture of the ZigBee software Stack

3.7.1.1 Application Level

The application layer is defined by public (certified by ZigBee Alliance for interoperability purposes) and private (for use in closed systems) profiles [41]. The application objects are end devices that can be supported by a single ZigBee node (Coordinator). A single ZigBee node can support up to 240 devices. The ZDO is a generic name for a ZigBee device of any type i.e., it could be a Coordinator, Router or End device.

3.7.1.2 ZigBee stack Level

The ZigBee Stack is the firmware stack that takes the responsibility for initializing and maintaining mesh network, routing messages in a secure manner, and storing and updating the route table information. Network Layer is responsible for all the functions mentioned above. The firmware stack software is provided either by chip vendor or third party source for specific development tool with combination of the radio and the microcontroller. Application Support sub-Layer basically communicates the messages to the right application object and maintains the binding tables. ZDO Management Plane accounts for the APS and NWK layers to communicate with the right ZDO [40] [42].

3.7.1.3 Physical/Data Link Level

IEEE 802.15.4 MAC Sub-Layer basically dispatches the data to be sent or decomposes the received data by addressing them properly. IEEE 802.15.4 PHY Layer is concerned with the interface to the physical transmission medium.

3.7.2 ZigBee Stack Terminology & Application Framework

ZigBee **Profiles** have already been described in the previous section. End Points send messages to other End Points via predefined data types called **Clusters**. Basically, data sent or received over the network have several attributes concerning the device they are sent from and are grouped into a cluster. For example, Home Control Lighting profile has a cluster called OnOffSrc with attributes Auto, Override, and Factory Default. **Task** is a piece of code written to implement any function. **Event** is an occurrence used to trigger a task to run. **Message** is information exchanged between tasks. **Descriptors** basically stores information about the ZDO and its application. Descriptions are of four types which are simple, endpoint, node power, and node [42].

3.7.3 Binding

Binding is a mechanism by which a relationship is established between nodes to communicate with each other. Information between the nodes is exchanged in terms of clusters to establish a relationship. Binding tables store the clusters IDs, binding information such as the network addresses and application end point for each association. There are different type of bindings based on the application we use [43] [24].

3.7.4 ZDO API's

ZDO API's are present for several services like:

- Device and Service Discovery
- End Device Bind, Bind and Unbind service
- Network Management Service
- Device Network Startup.

3.7.5 ZigBee Reliability & Security

ZigBee provides a safe and reliable operating environment even in the presence of other ZigBee networks operating at same frequencies using several techniques like Listen Before Send, Acknowledgements, and Alternative Routes. Some security measures are AES-based Encryption, Message Timeout, and Access Control Lists.

3.7.6 Routing

ZigBee employs several mesh routing techniques for secure transfer of data to the right device. This routing capability distinguishes ZigBee from other protocols as they can work in Ad Hoc environments where the network is unknown initially. ZigBee has capability of self-healing on route failure [24].

This chapter described the two protocols chosen to monitor pipeline system using a WSN. The following chapters describe the software and hardware required for the implementation such a monitoring system.

CHAPTER 4

MONITORING SYSTEM USING SIMPLCITI & ZIGBEE WIRELESS PROTOCOLS

The previous chapter gave a brief introduction to the wireless sensor networks and described the working of the RF protocols shortlisted for the purpose of building the monitoring system. This chapter will cover the design of the monitoring system and the network topology in which both the RF protocols would work along with the hardware and software requirements.

4.1 Hardware requirements & Software Design for SimpliciTI and ZigBee

SimpliciTI is supported by 5 families of radios from TI which include CC1100, CC1101, CC2500, CC2510, CC2511, CC1110, CC1111, CC2520, CC2430 and CC2530.

ZigBee is supported by radios from different vendors. Few of them are TI's CC2530 and CC2520, Digi International's XBee XB24CZ7PIS-004, Freescale's MC13224, MC13226, ST Microelectronics's STM32W, Microchip Technology's MRF24J40MA, MRF24J40MB, MRF24J40MC, etc.

Development Kits selected for SimpliciTI is TI's EZ430RF2500 [36] and for ZigBee CC2530 Mini ZNP Kit [35] from TI. The reasons for selecting the development kits from TI, their hardware description, and integration of the sensors with these kits are explained in the subsequent sections. The sensors selected for monitoring systems are the one's described in Chapter 2, MMA8451Q accelerometer, MAG3110 magnetometer and the embedded sensor within the microcontroller for temperature.

4.1.1 Description of EZ430RF2500 Development Kit

EZ430RF2500 is a wireless development kit from TI which combines a microcontroller MSP430F2274 and a radio CC2500 from Chipcon, as well as the software required to build a low power RF project.

MSP430F2274 is a mixed signal microcontroller with 32KB and 256B of Flash memory and 1KB of RAM memory [37]. The pin diagram of MSP430F2274 is shown in Figure 2-10.

CC2500 from Chipcon is a 2.4GHz transceiver designed for low-cost and low-power wireless RF applications as in consumer electronics, wireless game controllers, RF enabled remote controls, 2400-2483.5 MHz ISM/SRD band system etc., [13]. CC2500 has configurable data rates between 20-500 kBaud and is integrated with a configurable baseband modem that supports various modulation formats. CC2500 has a 64-byte transmit/receive FIFO buffer. CC2500 provides hardware support for packet handling, data buffering, burst transmissions, Clear Channel Assessment, link quality indication, and wake on radio [13]. Several other features of CC2500 are found in its datasheet [13].

A combination of an MSP430F2274 and a CC2500 is found on the EZ430RF2500 development kit from TI, shown in Figure 4-1. The development kit comes with two EZ430 RF2500 target boards, EZ430RF USB debugging interface and battery pack with expansion boards (Figure 4-1) [17].

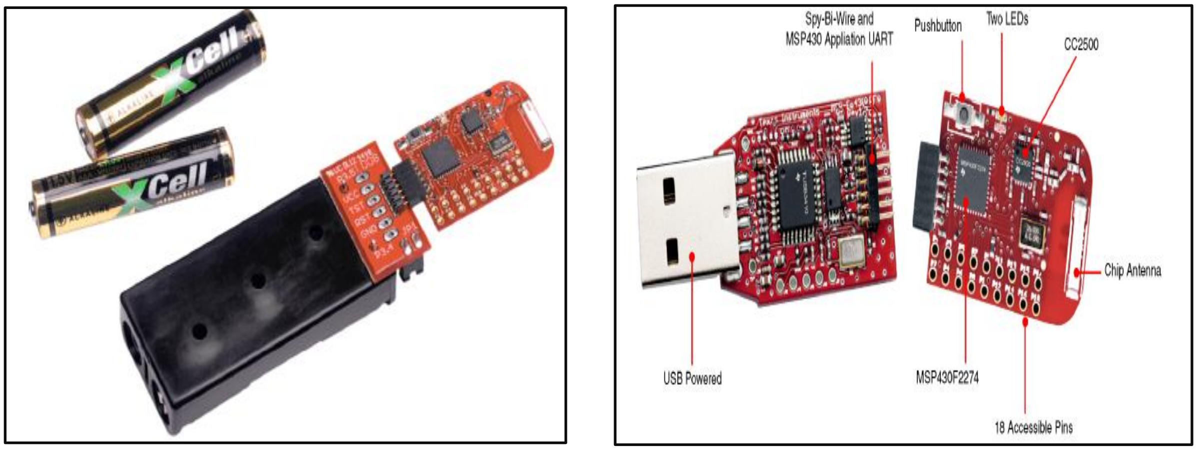


Figure 4-1. EZ430 RF2500 & Battery Expansion Board

4.1.1.1 Sensor integration with EZ430RF2500

Figures 4-2, 4-3 describe the sensor integration with the microcontroller on the EZ430RF2500 board.

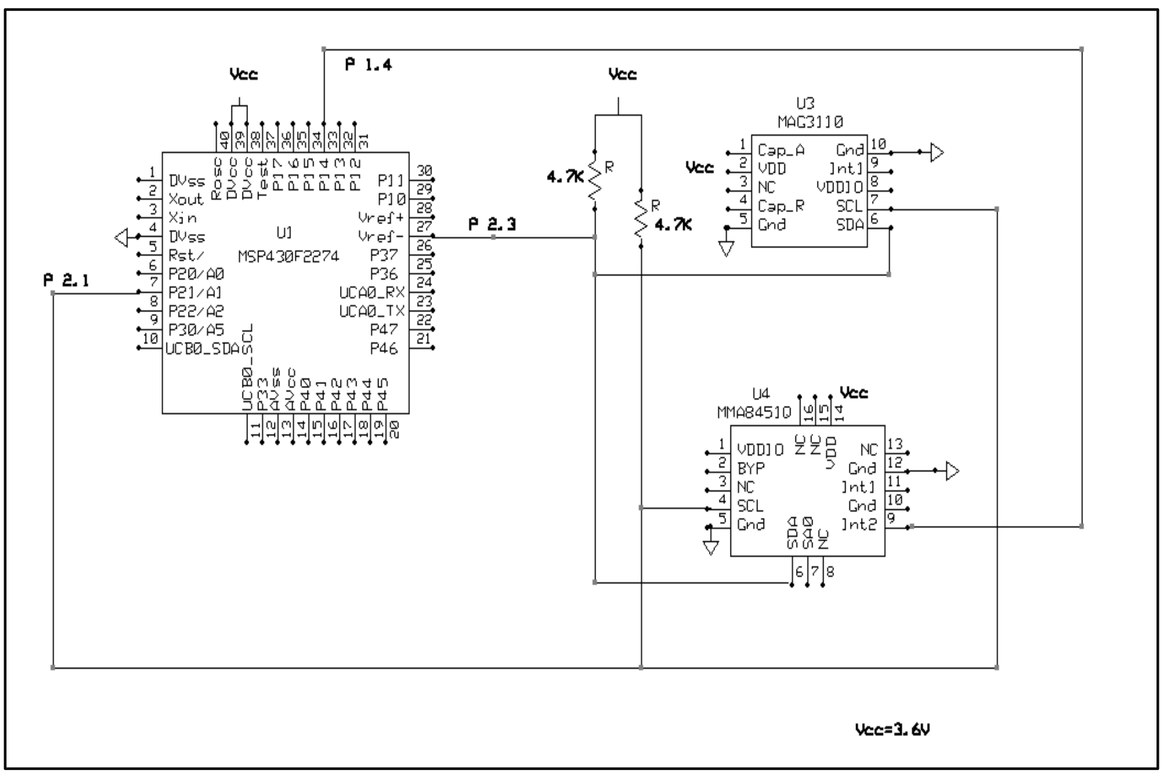


Figure 4-2. Connections between MCUMSP430F2274 on Ez430Rf2500 and Sensors MAG3110 and MMA8451Q

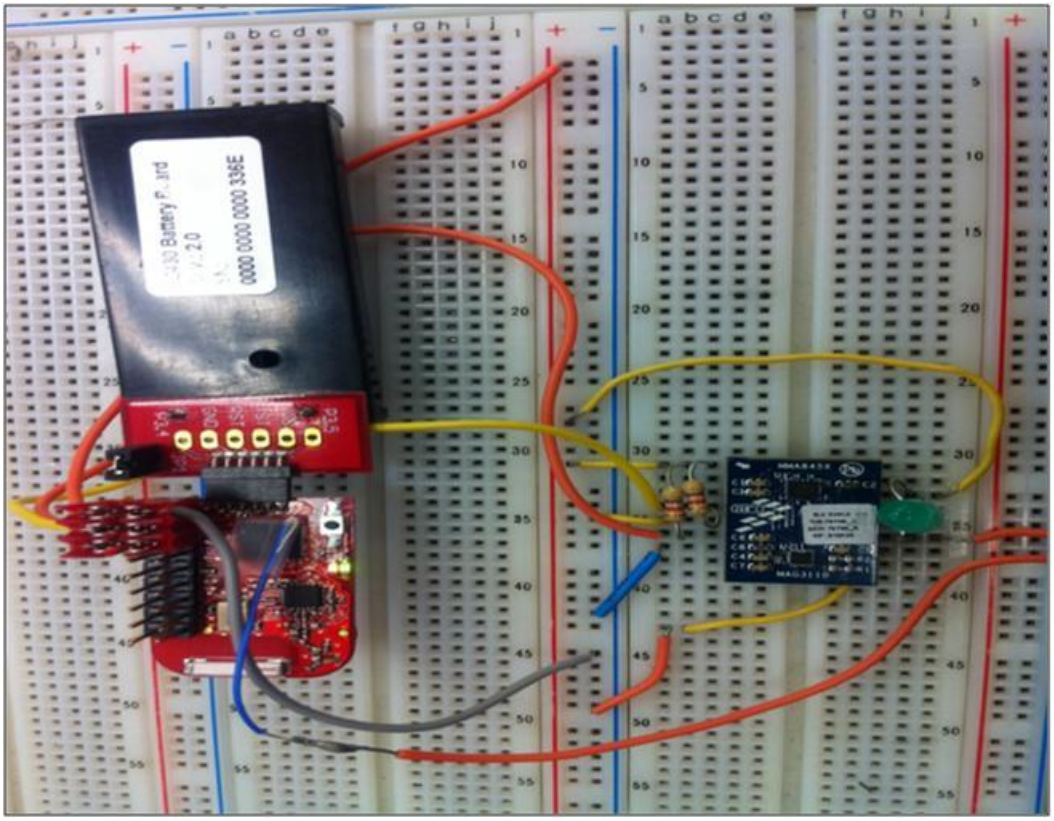


Figure 4-3. End Device of SimpliciTI with sensors

4.1.1.2 Modes of communication between components of monitoring system

Table 4-1 provides the modes of communication used by MCU to communicate with the radio and the Sensors along with their speeds and modules used for communication.

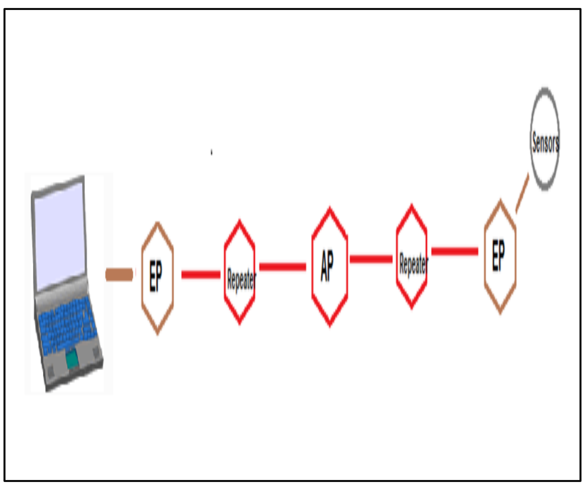
Table 4-1. Modes of communication between components of monitoring system

Components	**Mode of communication**	**Speed**	**Module used**
MCU-Radio	SPI	1 MHz	USCIB0
MCU-Sensors	I^2C	1 MHz	General I/0 Pins

4.1.1.3 Software Design for Monitoring system using SimpliciTI

The SimplciTI program on the development kit boards is a basic version of the original protocol. It needs to be modified according to the user needs and used under TI proprietary name. The following are the steps in the design process.

1. As an initial step in the design process the network topology in which our monitoring system works should be fixed. As SimpliciTI works only in a star and peer-to-peer topology we can use either of them. We fixed our monitoring network topology to star topology. The star topology can have routers if required as in Figure 4-4 or not as in Figure 4-5. Design of software and packet format is different for different devices in the network like Access Point, End Point and Routers.



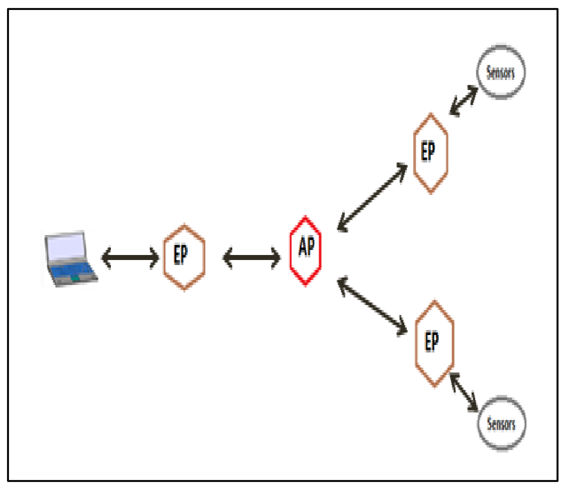


Figure 4-4. Star topology with repeaters Figure 4-5. Star topology w/o Repeaters

2. Data packet size can be fixed to 60 bytes or it can be varied. First 11 bytes of packet always contains the length of the packet, source address, destination address, sequence number of packet in case of secure messaging, LinkID and port number, followed by

sensor data information from 12^{th} byte. Sensor information packet on End Device is described in Table 4-2.

3. Access Point links the end devices and routers to its network. It has to receive information from different types of end devices. Access Point distinguishes the end devices during link time and stores their Link IDs in different buffers. It is built with the capability to distinguish the source of information based on the type of information and routes the information to the correct end device.

Table 4-2. Packet format of Sensor information from End Device

Byte Position	**Information**
1-11	Network Application information
12	Type of information to indicate to Access Point.
13	Extra information if Required. Temporarily LinkID is sent.
14	LinkID.
15	Type of End Device
16	Which sensor information? Temp or Acc or Mag
17	Extra information if any
18-60	Sensor data

4. Access Points receives information from end device on the computer on a separate port defined to differentiate between packets with commands and packets with sensor information. The End Devices are defined by End Device Type ID's and Sensors are defined by Sensor Type ID's are described in Table 4-3 and Table 4-4. Working of Access Point is better described by flowchart in Figure 4-6.

Table 4-3. Type of End Device with values

Type of End Device	Value
End Device with No Sensors	0x01
End Device with all 3 three sensors	0x02
End Device with Temperature sensor	0x03

Table 4-4. Sensor type Recognized by Sensor Type ID

Sensor	Sensor type ID
Temperature Sensor	0x01
Accelerometer Sensor	0x02
Magnetometer Sensor	0x03

5. End Devices in this monitoring system design are basically of two types: End Device connected to GUI and end devices with sensors. End Device on computer takes command from GUI via UART at 9600 baud rate and transmits them over RF to Access point. Message format sent from end device connected to computer to the Access Point is shown in Table 4-5. The working of End Device on computer is explained in Figure 4-7.

Table 4-5. Message format sent from End Device connected to computer

Byte Position	Type of Information
1-11	Network Application information
12	Direction to Access Point –
13	End Device number to which command should be sent.
14	Command to End Device.

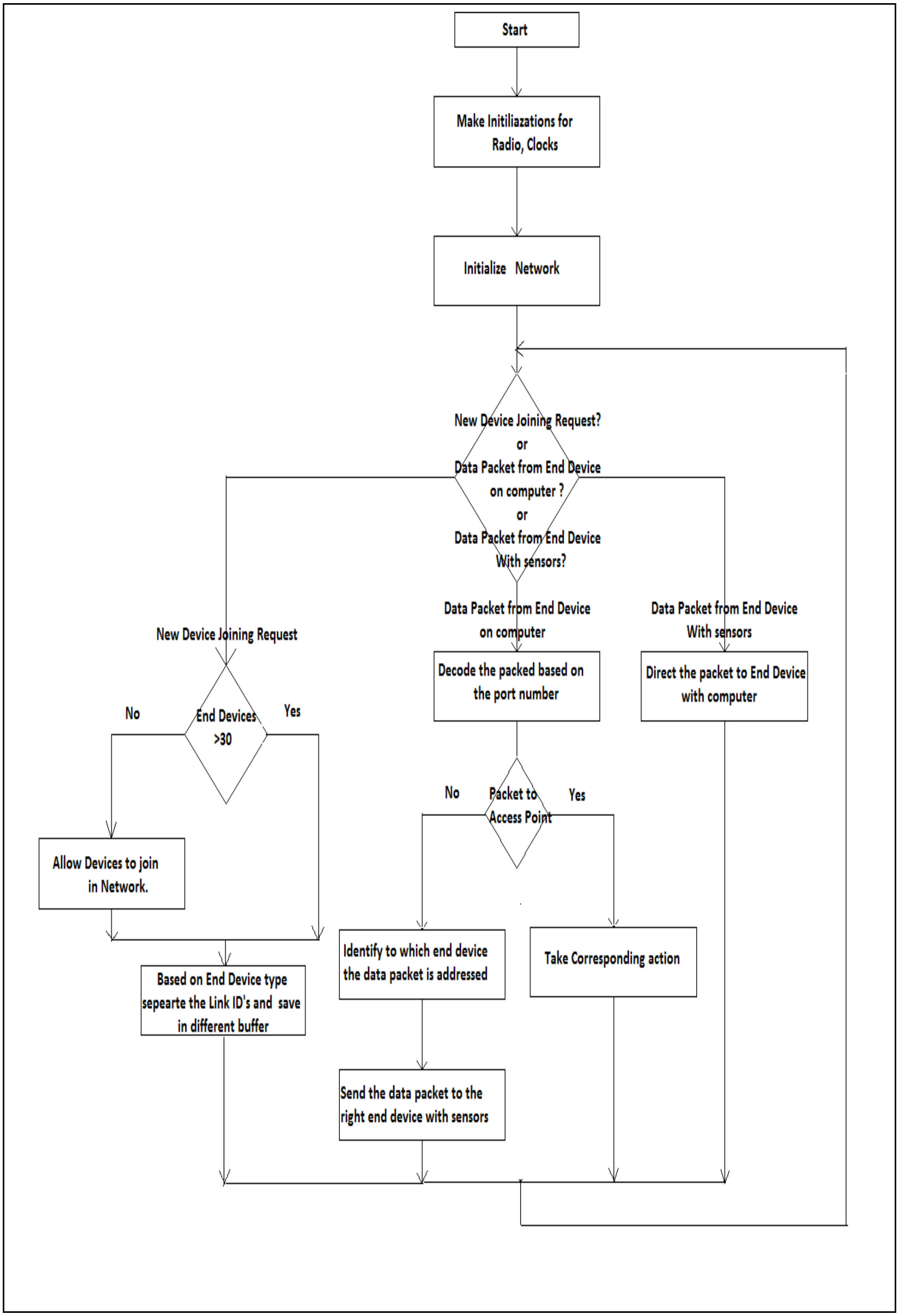


Figure 4-6. Flowchart for working of Access Point

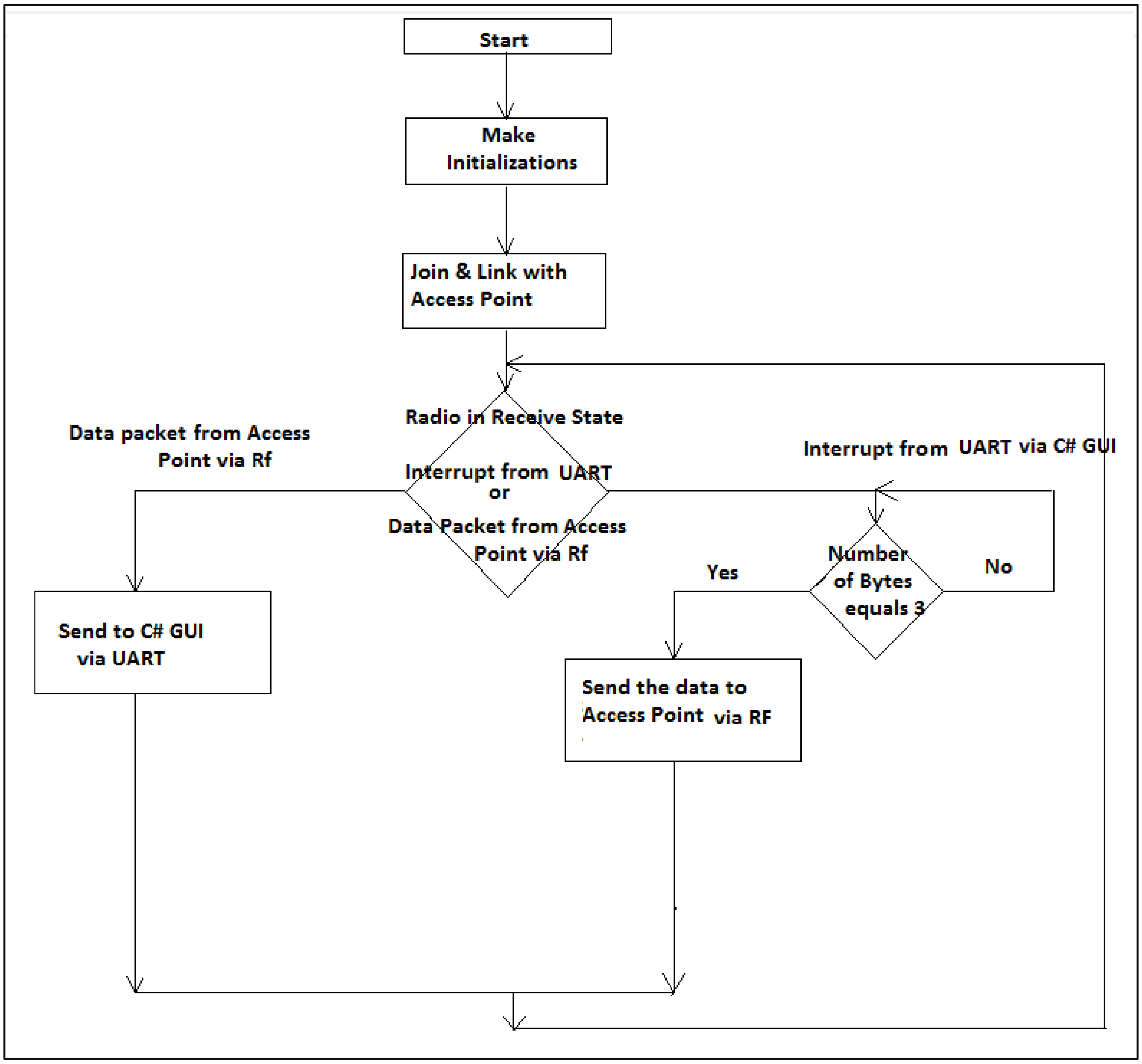


Figure 4-7. Flowchart for working of end device connected to computer

6. End Points are usually sleeping devices that respond to the sensors on interrupt, and access point on request. End points are connected to the sensors as shown in Figure 4-5 and End Device program is loaded into the EZ430RF2500 target board. The End Points communicate with Accelerometer and Magnetometer via I2C. The settings for MMA8451Q & MAG3110 can be modified by configuring the registers. The temperature of the pipelines can be collected using the temperature sensor on the MCU and its unit is in degrees centigrade. The temperature is sampled every ten seconds using ADC module

of MCU. The sensor information is stored in the flash of the MCU. Flowchart in Figure 4-8 describes how End Devices with sensors work.

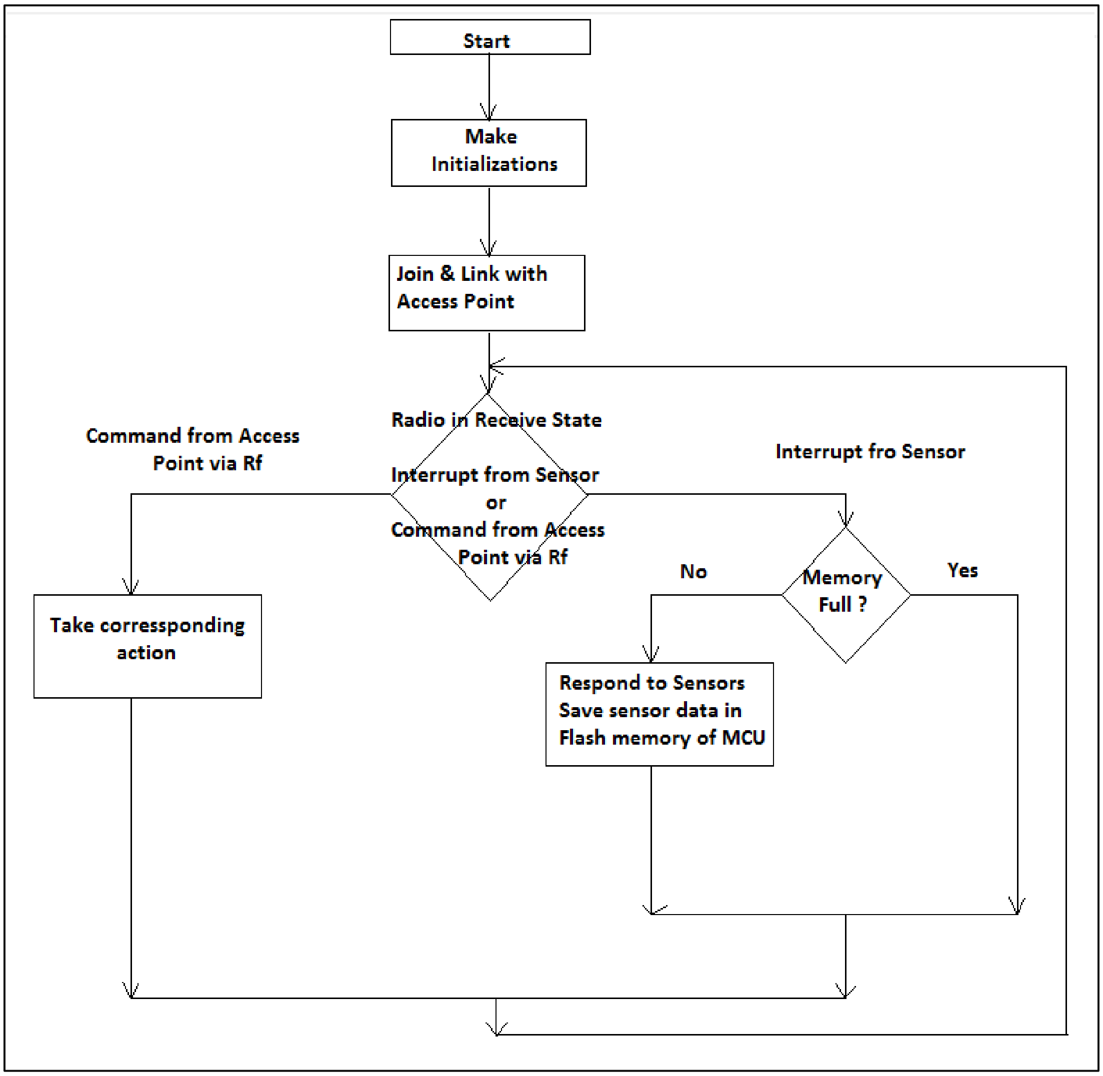


Figure 4-8. Flowchart for working of End Devices with sensors

The end device responds to different commands like:

- Start communication with sensors – Initial step to start communication with end devices.

- Start sensors – Initiates the accelerometer to trigger MCU with an interrupt, whenever motion is detected above a threshold value (say 0.063g or 0.5g or 1g, g being the gravity).
- Stop Sensors – Stops the accelerometer to detect any motion detection.
- Send Temperature Data – This command sends the temperature data recorded over a period of time in packets. The structure of temperature data is three bytes, with two bytes of temperature data and one byte of voltage level of the battery operated device, if required, for determining battery's lifetime. The end device sends 10 samples of temperature in every packet taken at intervals of 10 seconds from the time the microcontroller is powered on. Temperature data is stored in flash of MCU starting at address 0xB000. 66 samples of temperature data can be stored within allocated flash memory.
- Send Accelerometer Data – This command sends accelerometer data stored in the flash along with their time stamps. Each sample of Accelerometer data is ten bytes with 4 bytes of timestamp and 6 bytes of accelerometer data with two bytes of data on each axis. Each packet has 3 samples of accelerometer information. Accelerometer data is stored in flash of MCU starting at address 0xBC80. 1500 samples of accelerometer data can be stored with allocated flash memory.
- Send Magnetometer Data – This command sends magnetometer data stored in flash; each packet can send up to 6 samples of magnetometer information. Magnetometer data is stored in flash of MCU starting at address 0xB0C8. 500 samples of magnetometer data can be stored with allocated flash memory

- There are also commands to change the threshold values for motion detection of accelerometer and to dump the memory at once.

7. Routers just route the information between two peers or between an access point and end point. Communication between all the devices on the network is managed by C# GUI (explained in the subsequent sections).

4.1.2 Description of CC2530 Mini ZNP kit

There are two approaches to using ZigBee Processor from TI from two different development kits namely CC2530ZNPMini Kit and MSP430F5438 Experimenter Board with CC2530. How these both kits can be used to implement the ZigBee wireless task is better explained in Figure 4-9.

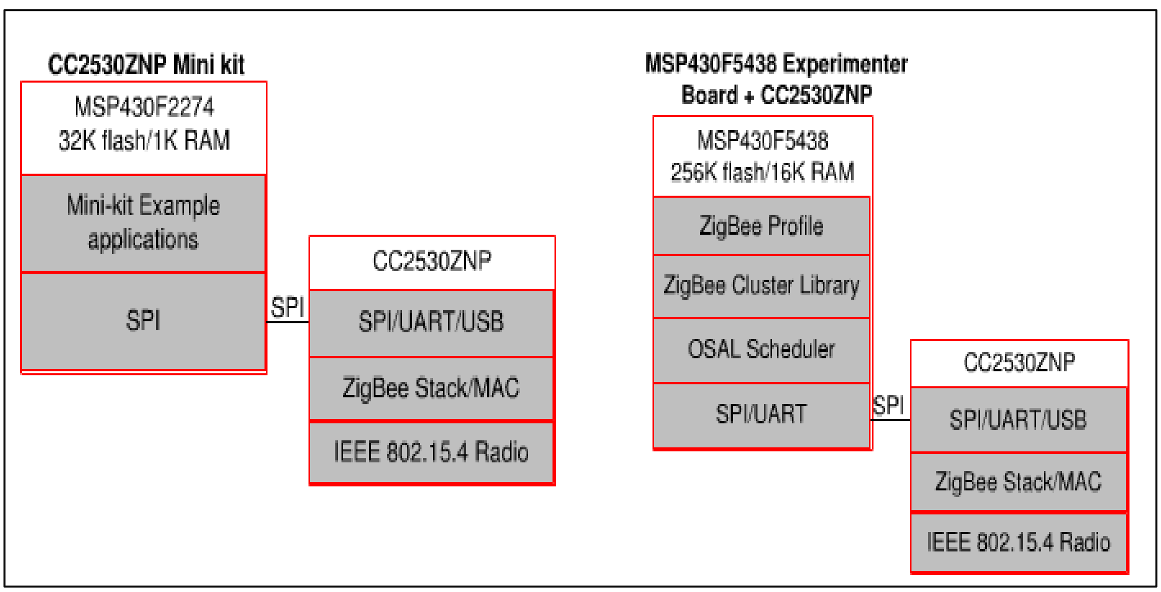


Figure 4-9. Approaches to using ZNP Processor CC2530

ZNP Mini Kit has no ZigBee profile support, but easily portable than other TI platforms. As SimpliciTI is designed to work in star network; we can use the CC2530 ZNP Mini Kit to work in the star network and compare the power consumption by these radios.

CC2530 Mini ZNP Kit possess three target boards to setup a small network, each target board is equipped with an MCU MSP430RF2500 and a ZigBee Network Processor CC2530. It also has eZ430 USB stick and two battery boards as seen in Figure 4-10. Table 4-6 describes the mode of communication between sensors and MCU on ZNP Mini Kit. The target boards are equipped with an accelerometer sensor CMA3000-D01 [2] and an ambient light sensor SFH5711 from OSRAM [3]. CC2530 ZNP has two sets of interfaces-Reduced Interface API and Full-ZigBee API.

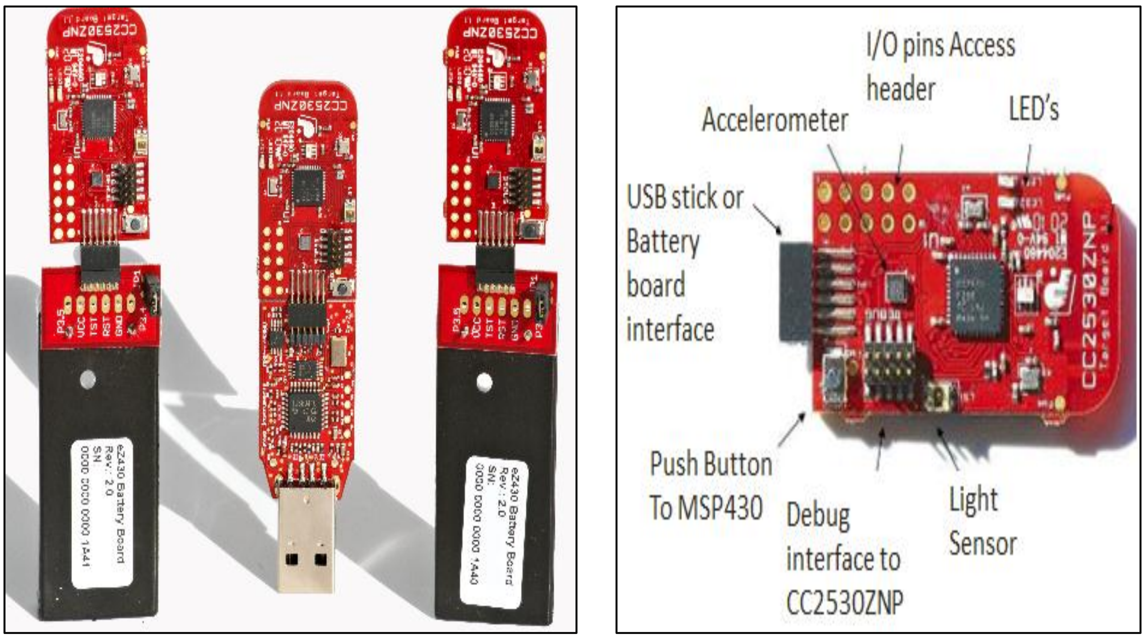


Figure 4-10. EZ430RF2500 ZNP Mini Kit

Table 4-6. Mode of Communication between sensors and MCU on ZNP Mini kit

Components	Mode of communication	Speed	Module used
MCU-Radio	SPI	4 MHz	USCIB0
MCU-Accelerometer	SPI	1 MHz	General I/0 Pins
MCU-Light Sensor	General I/O		Analog to Digital Converter (ADC)

4.1.2.1 Protocol Scenarios

General Frame format is shown in Table 4-7. Length field has the length of the data field, command field has command of the frame which is constructed of two bytes and Data field contains frame data. Command field is constructed of two bytes as seen in Table 4-8. Command types are interpreted in Table 4-9 and 4-10. CC2530 ZNP Interface Specification document explains different command frames in detail [4].

Table 4-7. General Frame format

Bytes 1	2	0-250
Length	Command	Data

Table 4-8. Command fields in ZigBee

Command 0		Command 1
Bits 7-5	Bits 4-0	Bits 7-0
Type	Subsystem	ID

Table 4-9. Command Types in ZigBee Frame

Command Type	Command
0x00 POLL	Poll command is used to retrieve queued data.
0x01 SREQ	A synchronous request requires immediate response.
0x02 AREQ	An asynchronous request generally handling call back events.
0x03 SRSP	A synchronous response sent in response to SREQ command.

Table 4-10. Subsystem Types in ZigBee Frame

Subsystem Value	Subsystem Name
0	RPC Error Interface
1	SYS Interface
2	Reserved
3	Reserved
4	AF Interface
5	ZDO Interface
6	Simple API Interface
7	UTIL Interface
8-32	Reserved

4.1.2.2 Software ZigBee Pro Stack and Integration with the Monitoring System.

CC2530 ZNP Mini kit runs on ZigBee Pro stack. Limitations on the mini kit as explained before, are it has no ZigBee profile support and no operating system abstraction layer. Initially, ZigBee Pro Programs are studied, and our application is built on the ZigBee stack without modifying the joining process of the routers or end devices in the coordinator's PAN network. The following steps are involved in building the monitoring system:

1. The network topology in which ZigBee Mini kit is used to work is star topology.
2. Data packet size is fixed to 100 bytes with 60 bytes of sensor data in it.
3. The flow of data should be made bidirectional.
4. Access Point is connected to the system and a C# code is written to display the information received and transferred to the access point. Any command to the end points can be sent from the C# GUI. Different command frames are sent from Coordinator to the End Devices.
5. If end devices in the network are required, a broadcast message is sent by the coordinator and end devices respond to this request by sending their short address. The coordinator sends the list of short addresses of end devices to the C# GUI and the program would represent the End Devices with their short address in a list box where we can select them and send commands.
6. End Devices have a self-healing capability. If the Coordinator is powered off while transferring data, end device would wait for some time and restart itself to check for a new PAN network.

7. The ambient light sensor data is stored in flash of the end device every 1 minute. Light sensor information is of 4 bytes for each sample. First, two bytes are light sensor information; next, two bytes are battery voltage level to check the lifetime of the battery in case of battery operated devices. Light sensor data is stored in flash of MCU starting at address 0xB000. 800 samples of light data can be stored within allocated flash memory.

8. The accelerometer sensor is programmed to work on detecting motion. Once motion is detected, it sends an interrupt to the MCU and motion information is received from the sensor and stored in the flash along with timestamp. Timestamp and accelerometer data are each 4 bytes long. In accelerometer data, the first byte describes the cause of motion and next three bytes describes motion data on the three axis. Accelerometer data is stored in flash of MCU starting at address 0xBC80. 400 samples of accelerometer data can be stored with allocated flash memory.

9. Few Commands to End Devices from Coordinator are:

   - Start_Sensors – This command initiates the sensors for motion detection.
   - Bye – This command erases flash memory where sensor data is stored and makes the accelerometer stay in motion detection state.
   - Send_light_Sensor_Info – This command directs End Devices to send the light sensor information stored in the flash.
   - Send_Accelerometer_Info – This command directs the End Device to send accelerometer information stored in the flash.

- Dump_memory – This command directs the End Device to send all the information in the flash, i.e., light sensor information followed by accelerometer information.
- Erase_Flash – This command erases flash memory and this command should be sent once Dump_memory is issued in order for memory to accommodate new data.

The subsequent sections explain C# GUI and how it works for both SimpliciTI and ZigBee.

4.2 C# GUI

Initially, there was a requirement for a packet sniffer kind of application to find the data flow in and out of the targets connected to the System either in case of SimplciTI or ZigBee. The packet sniffer developed by TI does not support the Development kits selected for the monitoring system. So a C# GUI was built which displays the data in and out of the target boards and also makes it easy to recognize the end devices in the network and send commands to them when required. Data is transferred in and out between target boards and C# GUI via serial port at 9600 baud. There are two different screens that would appear when we execute the program WIFIConsoleUI. They are explained as follows:

- The first screen called Home screen gives options for selection between SimpliciTI and ZigBee and options to select the COM Port (serial port) over which we would like to communicate with the target board.

- The second screen would popup according to the options selected on the Home Screen.
- In the background, on every button click, a packet is sent from the serial port to target board connected to computer. The target board decodes the packet and performs the required action.
- In case of SimpliciTI end device is connected to computer whereas in case of ZigBee, Coordinator is connected to the computer.
- The Access Point decodes the packet information received from end device on computer and performs the required action in case of SimpliciTI.
- Screenshot of Home Screen of GUI is shown in Figure 4-11.

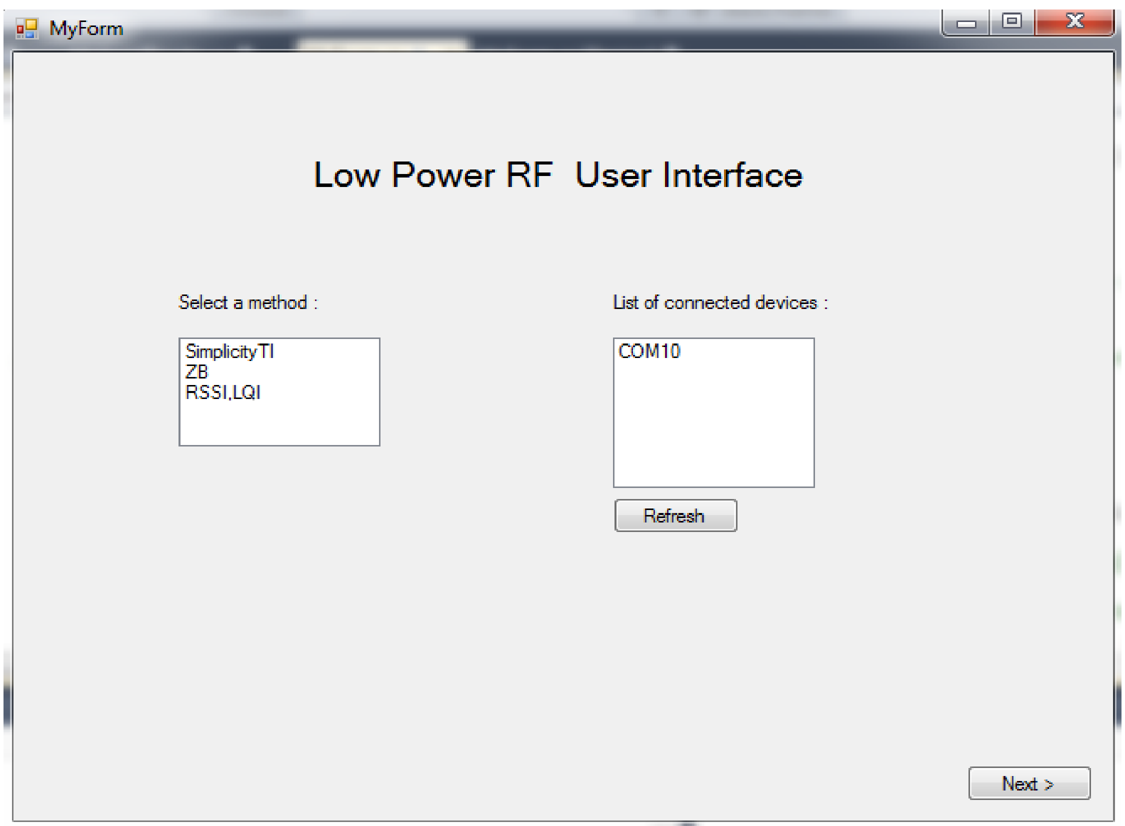


Figure 4-11. Screenshot of Home Screen of GUI

- In case of SimpliciTI, clicking the Refresh button would display the end devices in the network along with the type of sensors present on each end device.
- By selecting the End Device, the commands list box will highlight with different commands that could be sent to the end device. Once the command is sent to the devices, the required action is performed by the devices in the network. The response to commands such as Dump_memory or Send Accelerometer Data would be end device on computer receiving data packets. The data packets received on the target board via RF are sent to the serial port and in the C# program they are decoded according to the frame formats chosen in the SimplciTI format. The received packets can be displayed on the textbox on the screen in human readable format. A screenshot of the GUI when SimpliciTI is implemented is shown in Figure 4-12.

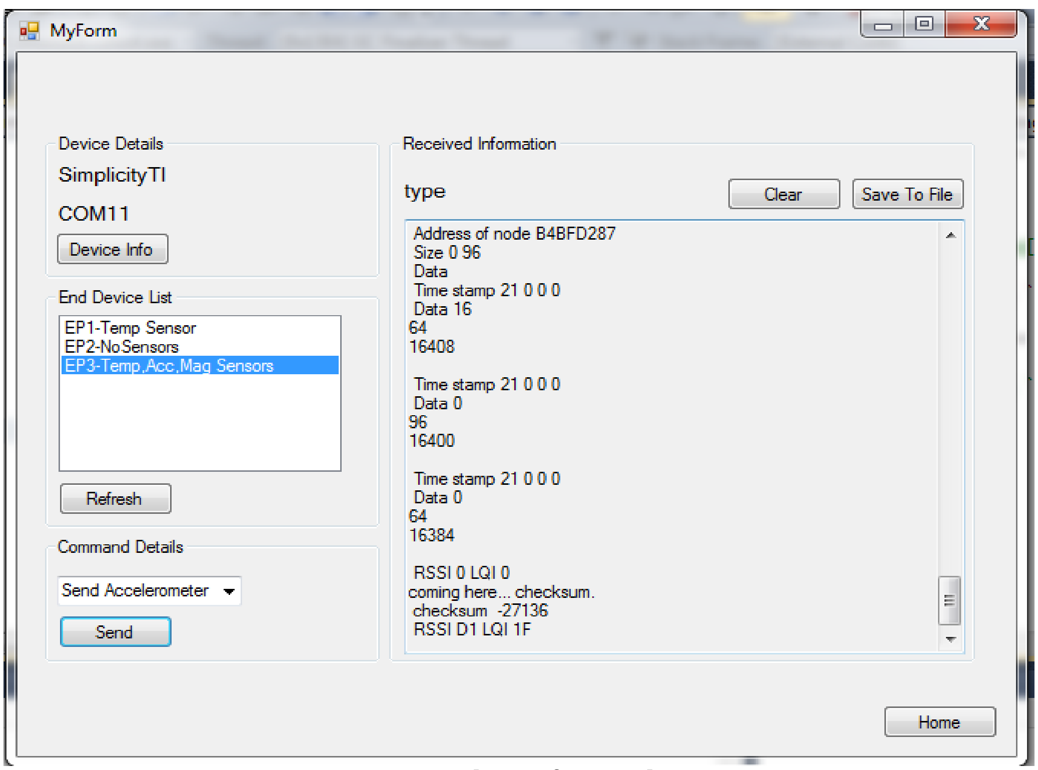


Figure 4-12. Screenshot of SimpliciTI screen

- In case the ZigBee option is selected on the home screen, the second screen would popup with different options that can be implemented on the coordinator connected to the computer. The Device Info button investigates what device is connected to the computer and automatically sends a broadcast message to devices on the network to check which devices are present in the network. The Refresh button refreshes the list of end devices in the network. Different commands described in section 3.2.2.2 can be sent to the end devices and received data can be seen in the text box in the screen. A screenshot of GUI when ZigBee is implemented is shown in Figure 4-13.

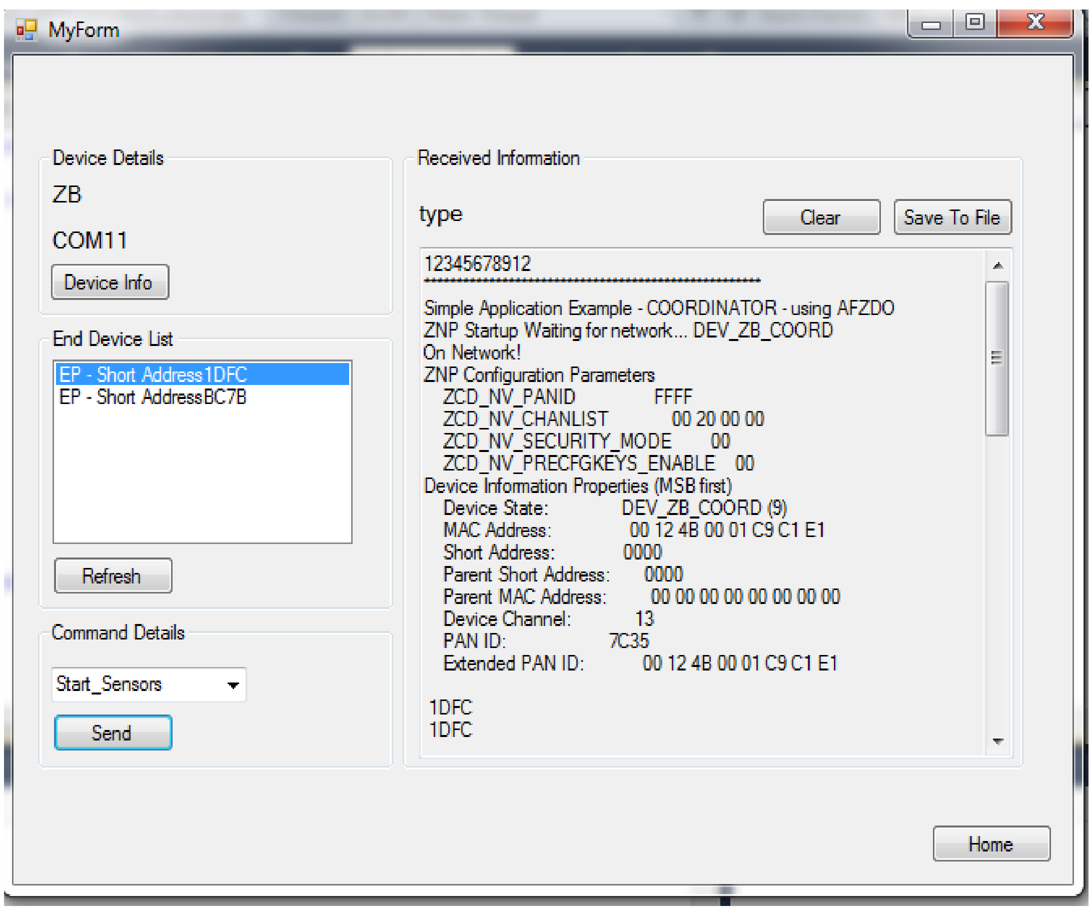


Figure 4-13. Screenshot of GUI with ZigBee implementation

- The data received on the serial port regarding the sensor information is stored in the database as a series of tables. This information can be retrieved when required and can be processed accordingly. The database used for storing information is MySQL Database. A screenshot of the MySQL server with tables used for storing data from sensors is shown in Figure 4-14.

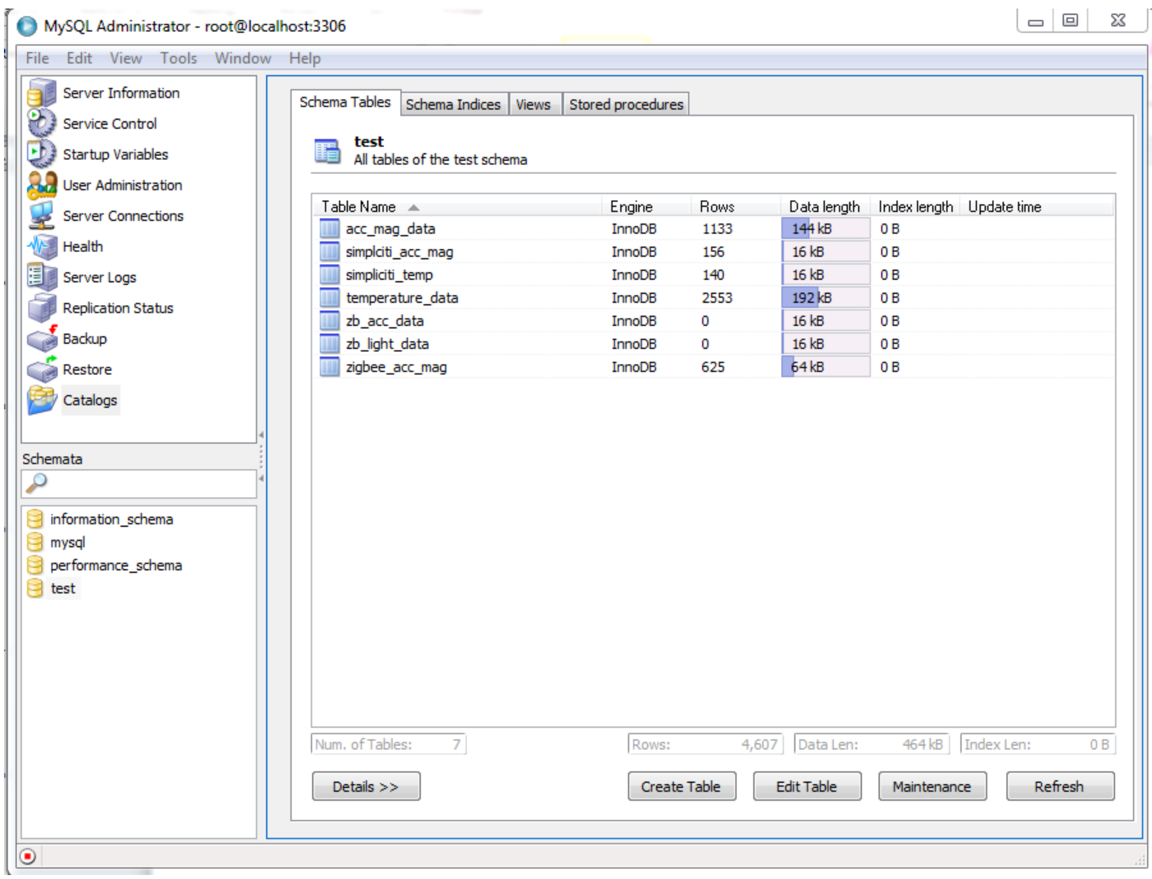


Figure 4-14. Screenshot of MySQL Database Administrator

This chapter discussed the hardware used for our monitoring system. It also described how the application layer is built using SimpliciTI and ZigBee protocols. The next chapter will present a comparative study between the two chosen protocols (SimpliciTI and Zigbee) with respect to our monitoring system application.

CHAPTER 5

COMPARISON OF SIMPLICITI & ZIGBEE PROTOCOLS

The previous chapter discussed the hardware components, design of the software, packet format, and the GUI used for the two protocols. This chapter will compare the two chosen protocols based on software and power requirements.

5.1 Design Freedom for SimpliciTI & ZigBee

After evaluating the Network Open Systems Interconnect (OSI) Basic Reference Model in Chapter 3 and design stack of the two protocols, we can come to few conclusions regarding the degree of freedom one has when designing our monitoring system protocol. SimplciTI stack freely allows the user to implement one's own higher layer protocol and application layer. In the case of ZigBee, the customer, is left with designing just the application layer. The design freedom range is summarized in Table 5-1 [24].

Table 5-1. Design Freedom of SimpliciTI and ZigBee

Design Freedom	SimpliciTI	ZigBee
Application	Design Freedom	Design Freedom
Higher Layer	Design Freedom	Z-Stack & Simple API
Lower Layer	SimpliciTI	TIMAC
Physical Layer	CC111x,CC251x,CC243x,CC253x,CC2430, Msp430 +C1101, CC2500 or CC2520	CC253x, CC243x, CC2480
RF Frequency	2.4GHz , Sub 1 GHz	2.4GHz

5.2 Current Requirements - SimpliciTI

The frame format used for SimpliciTI is shown in Figure 3-8 and the data packet formats are explained in detail in chapter 4. The arrangement required to measure current consumed by the target boards is shown in Figure 5-1.

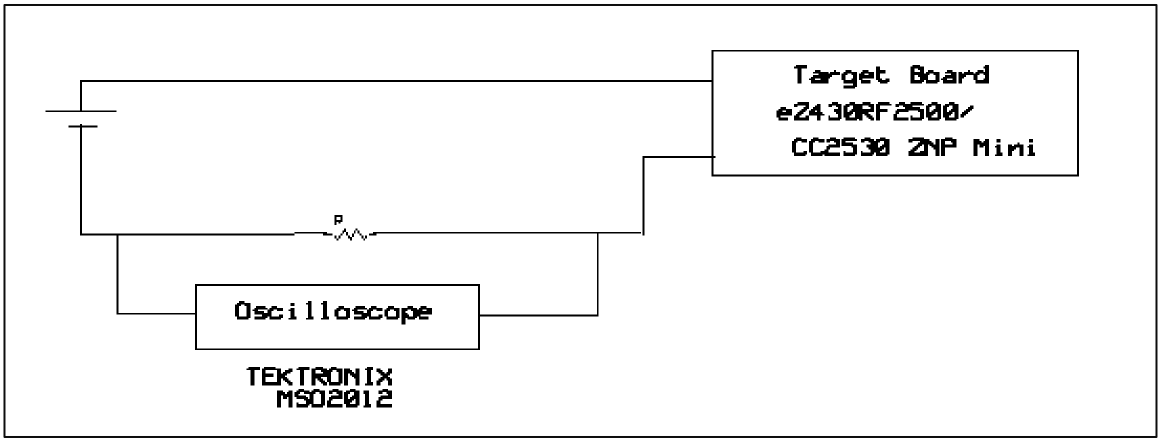


Figure 5-1. Setup for Current measurement by the target board

The current consumed during transmission/reception of packets via RF is the major factor affecting the battery life. In battery-operated target boards, we can visualize the current profile graphically on an oscilloscope by measuring the voltage drop across a fixed resistor of 12Ω. The current profiles of EZ430 RF2500, loaded with End Device program with different packet sizes, sent with a gap of 1 sec time frame are observed and the current consumption is tabulated in Tables 5-1 and 5-2. The transmitting power is changed and the current consumption is observed with two conditions where sent data is acknowledged and not acknowledged. In these scenarios, the packet size is varied and their current profiles are observed and rough estimate of the current consumption is calculated. It is observed that as the transmitting power increases, the range of communication increased. Current profiles with and without acknowledgements are shown in Figure 5-2 and 5-3.

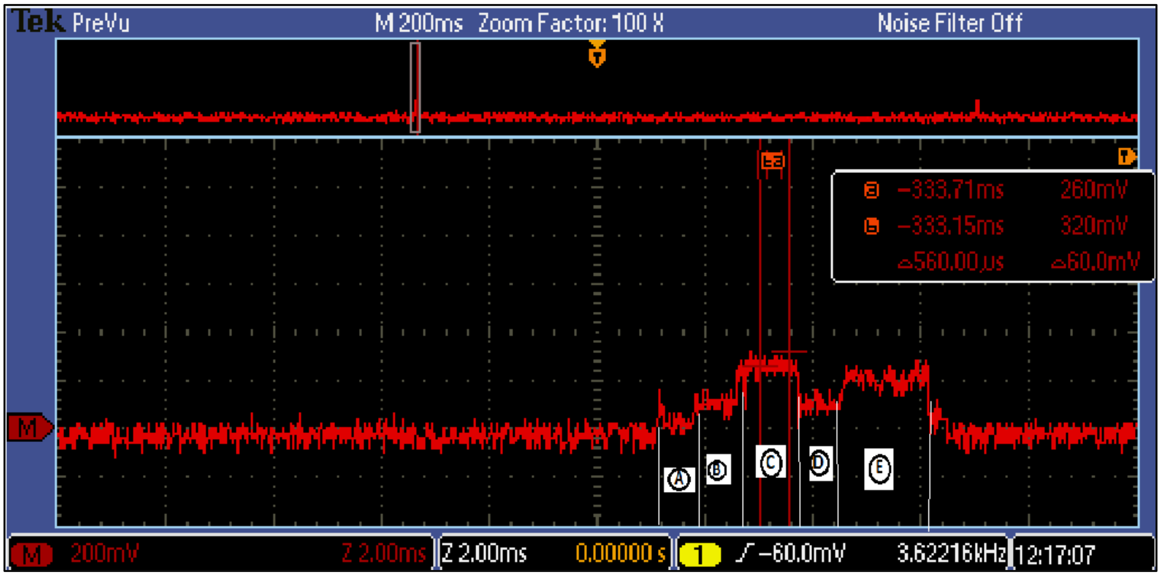


Figure 5-2. Current profile of end Device when transmitting data with acknowledgement

Figure 5-2 shows the current profile when a packet of size 20 bytes is transferred over RF. Current profile seen in Figure 5-3 can be interpreted in five different regions A, B, C, D and E. Each of these regions is shown in Table 5-2 along with the current consumed.

Table 5-2. Current profile interpretation of Figure5-2

Region	Time (msec)	Current(mA)	Amp*sec
A- Wake up Radio	0.6	5.83	3498nA*s
B- Idle mode of radio	0.8	13.33	10664nA*s
C- Transmit data packet	1.2	24.16	28992nA*s
D- Radio sleep and wake for receive	0.88	13.33	11730nA*s
E- Receive data packet and sleep.	1.72	20.83	35827nA*s
Total			90711nA*s

Figure 5-3 describes the current profile of EZ430RF2500 when end Device program is loaded and packet of size 20 bytes is transferred over RF with no data acknowledgement and transmitting power of 1.5 dBm.

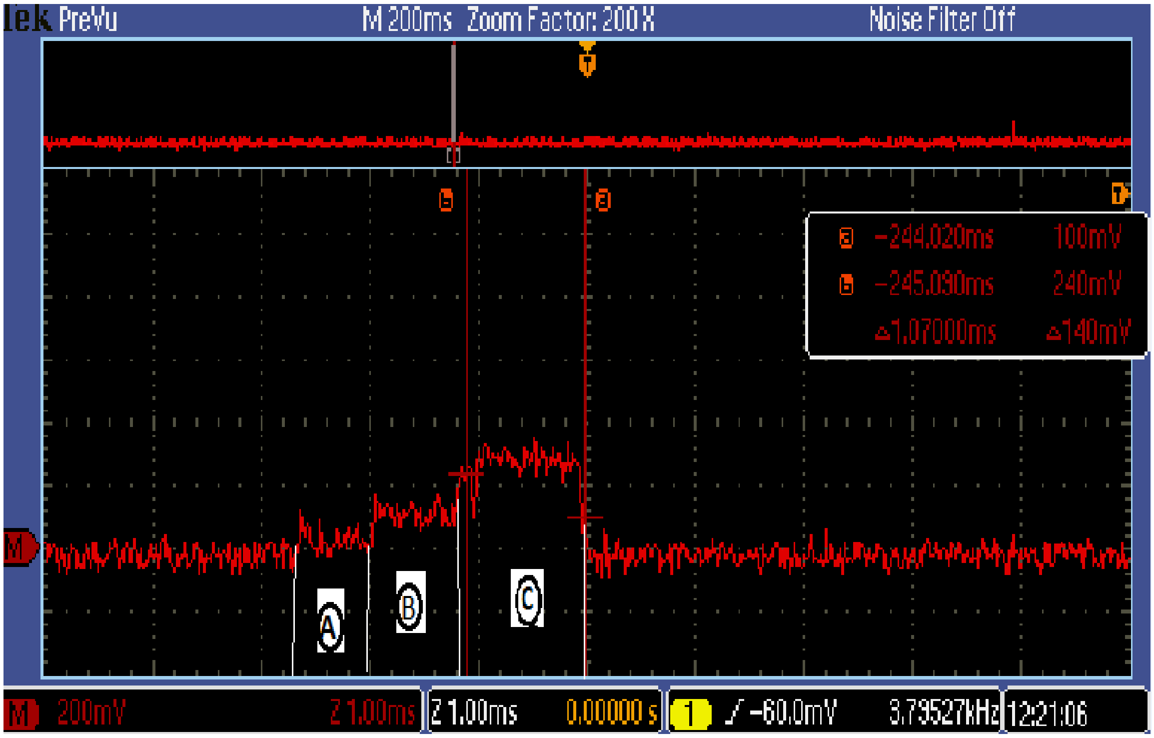


Figure 5-3. Current profile of end Device when transmitting data without acknowledgement

Table 5-3. Current profile interpretation of Figure5-3

Region	Time (msec)	Current(mA)	Amp * Sec
A- Wake up Radio	0.6	5.83	3498 nA*s
B- Idle mode of radio	0.8	13.33	10664 nA*s
C- Transmit data packet and sleep	1.1	24.16	26576 nA*s
Total time duration of transmission	2.85		40738nA*s

The current consumption for different transmission powers are shown in Tables 5-4 and 5-5.

Table 5-4. Current Consumption and transmission and reception time on end Device in SimpliciTI with 0.3dBm transmission power

Number ofBytes sent	ACK	SPI Speed	TX Power	Txtime(msec)	Current(mA)(Apprx)	ACK RX Time	Current (mA)(Apprx)	TX+RX time(msec)
60	Yes	8MHz	0xFE	1.895	21.66	1.945	22.5	7.04
50	Yes	8MHz	0xFE	1.88	23.6	1.96	20.16	6.64
40	Yes	8MHz	0xFE	1.28	24.16	1.6	19.16	6.08
30	Yes	8MHz	0xFE	1.2	22.5	1.48	20	6
20	Yes	8MHz	0xFE	1.2	24.16	1.72	20.83	5.2
60	No	8MHz	0xFE	1.874	23.3			3.67
50	No	8MHz	0xFE	1.8	26.3			3.78
40	No	8MHz	0xFE	1.34	24			3.12
30	No	8MHz	0xFE	1.04	22.5			2.66
20	No	8MHz	0xFE	1.07	24.35			2.85
60	Yes	1MHz	0xFE	2.48	14.6	10.6	20.83	18
50	Yes	1MHz	0xFE	6.1	19.8	3.208	19.6	15.6
40	Yes	1MHz	0xFE	2.17	22.5	8.145	18.75	13.21
30	Yes	1MHz	0xFE	2.12	20	8.2	20.83	12.76
20	Yes	1MHz	0xFE	1.84	23.33	7.36	19.167	12.16
60	No	1MHz	0xFE	2.58	18.3			3.7
50	No	1MHz	0xFE	1.7	25.3			3.8
40	No	1MHz	0xFE	1.45	21.6			2.98
30	No	1MHz	0xFE	1.16	24.16			4.24
20	No	1MHz	0xFE	1.49	20			3.77

Table 5-5. Current Consumption and transmission and reception time on end Device in SimpliciTI with –10.1dBm transmission power

Number ofBytes sent	ACK	SPI Speed	TX Power	Txtime(msec)	Current(mA)(Apprx)	ACK RX Time	Current (mA)(Apprx)	TX+RX time(msec)
60	Yes	8MHz	0x97	2.6	12.5	2.12	20	7.32
50	Yes	8MHz	0x97	1.66	14.16	2.044	20	6.58
40	Yes	8MHz	0x97	1.4	16.67	1.72	20	6.28
30	Yes	8MHz	0x97	1.34	11.66	1.62	21.16	5.78
20	Yes	8MHz	0x97	920usec	24	1.4	20	5.52
60	No	8MHz	0x97	2.28	16.67			4.2
50	No	8MHz	0x97	1.72	15.83			3.36
40	No	8MHz	0x97	1.46	13.33			3.16
30	No	8MHz	0x97	1.3	16.67			3.22
20	No	8MHz	0x97	960usec	17.5			5.12
60	Yes	1MHz	0x97	2.14	13.3	10.74	19.16	15.66
50	Yes	1MHz	0x97	2	13.3	9.64	20.83	14.16
40	Yes	1MHz	0x97	1.52	15.8	8.9	17.5	13.56
30	Yes	1MHz	0x97	1.26	11.67	8.42	16.67	12.98
20	Yes	1MHz	0x97	880usec	14.08	7.68	20	11.52
60	No	1MHz	0x97	3.052	12.5			3.812
50	No	1MHz	0x97	1.56	10.3			3.42
40	No	1MHz	0x97	1.38	10.83			3.36
30	No	1MHz	0x97	1.24	12.5			2.98
20	No	1MHz	0x97	880usec	12.5			2.64

The current consumption of a device in receive mode is shown in Figure 5-4. The current consumption is 21.5mA.

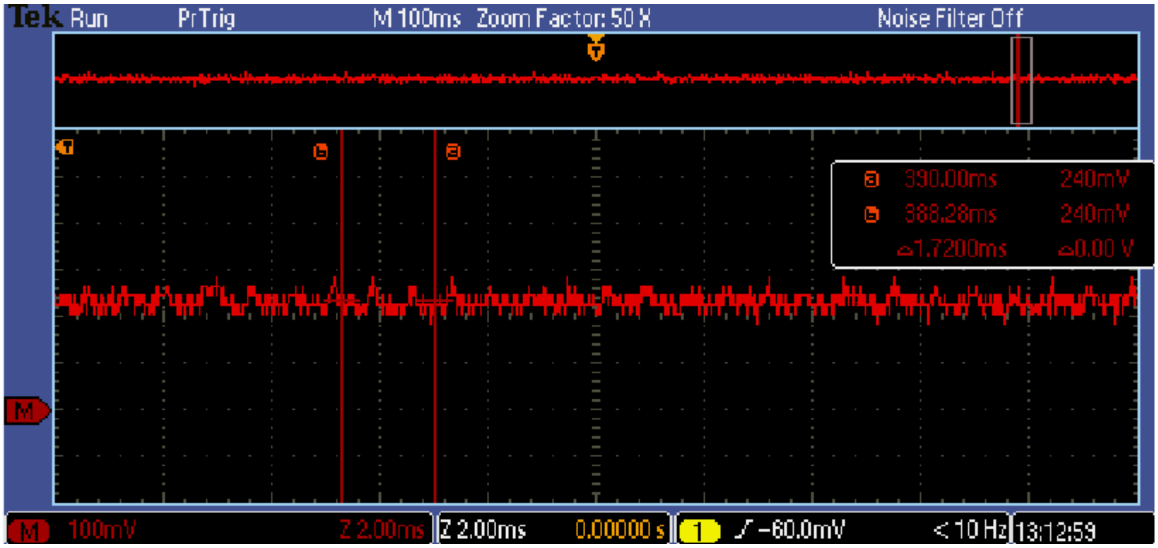


Figure 5-4. Current consumption in continuous receive mode of SimpliciTI target board

Figure 5-5 shows the current consumption when the End Device receives an interrupt from sensors and gets information from them with radio receiver is on.

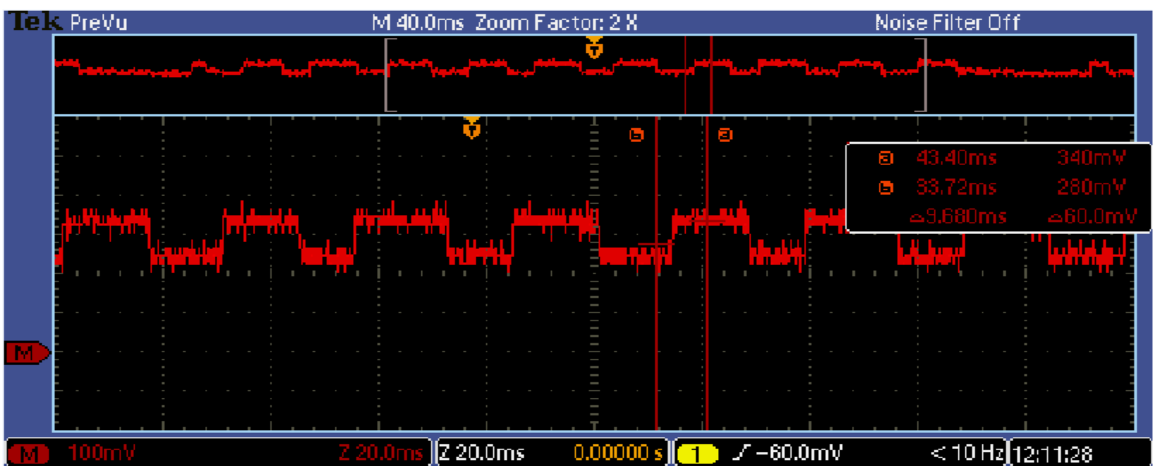


Figure 5-5. Current profile on End Device when attending interrupt from sensor

From Figure 5-5 the current consumption on End Device is 28.33mA (inclusive of radio receiver on current). Hence, the current consumption during interrupt is approximately 6.83mA. From all of the above observations the average current consumption can be estimated in different devices on the network with SimpliciTI. The monitoring system uses end device and access points with no acknowledgements operating at 2.4GHz with transmitting power of 0.3 dBm. The batteries used for monitoring system demo are AAA batteries with a rating of 1000 mAh.

Table 5-6. Life Expectancy of Battery in devices of Monitoring system using SimpliciTI

Device	Scenario	Battery Life(hours)
Access Point	Receive state waiting for end devices to join, link, and communicate between peers by routing.	46.5
End Point	Receive state waiting for commands to start sensors, receive interrupts from sensors.	46.5
End Point	Continuously receiving interrupts from sensors with radio on.	35.33
End Device	Just Interrupted by sensors throughout	111.11

The life of the battery is dependent on different modes of operations of the system. The Receive Signal Strength Indicator (RSSI) on the receiver varies as the distance from the transmitter varies. The range of communication varies with the change in transmitting power as shown in Table 5-7 with direct line of sight in a building.

Another test was performed to check the packet loss. A burst of 1000 packets was sent with a one second time gap and in a network that had a different number of nodes, by gradually increasing the distance between the receiver and end nodes. The results are tabulated in Table 5-8. The data packet sent from end nodes are in standard frame format (Figure 3-8) and a payload size of 45 bytes.

Table 5-7. Range of Communication variation with change in Transmitting Power of SimpliciTI devices in direct line of sight

PATABLE settings	TX Power (dBm)	Current (mA)		Range of communication (m)	
		Tx	Rx	Indoor	Outdoor
0xFF	1.5	25.833	23.75	70-75	46-50
0xFE	0.3	21.6	22.5	70-75	46-48
0xA9	-3.8	19.8	20.58	65-70	45-47
0x97	-10.1	15.5	21.4	15-17	19-23
0x55	-15.8	14.16	22.6	8-9	17-19
0x84	-24	13.08	22	6-8	2-3
0x50	-30	End points could not join network		End points could not join network.	<0.5

Table 5-8. Packet error Rate with different number of nodes in the network with direct line of sight

Distance (m)	1 node		2 nodes				Packet Loss Rate		
			Node1		Node2		1node	Node1	Node2
	Received	Lost	Received	Lost	Received	Lost			
1.5	1000	0	1000	0	1000	0	0	0	0
3.5	1000	9	961	39	1000	0	0.9	3.9	0
7.5	1000	0	1000	0	991	9	0	0	.9
10.5	999	1	956	44	889	111	0.1	4.4	11.1
66	973	27	842	158	863	137	2.7	15.8	13.7
70	205	795	836	164	890	110	79.5	16.4	11.0

5.3 Current Requirements ZigBee

The frame format used for ZigBee and the data packet format were explained in detail in chapter 4. The arrangement was required to measure the current consumed by the target boards as shown in Figure 5-1. The frames were sent over RF between coordinator and end devices are 128 bytes. The current profile of the end device during communication is shown in Figure 5-6 and it is interpreted in Table 5-9.

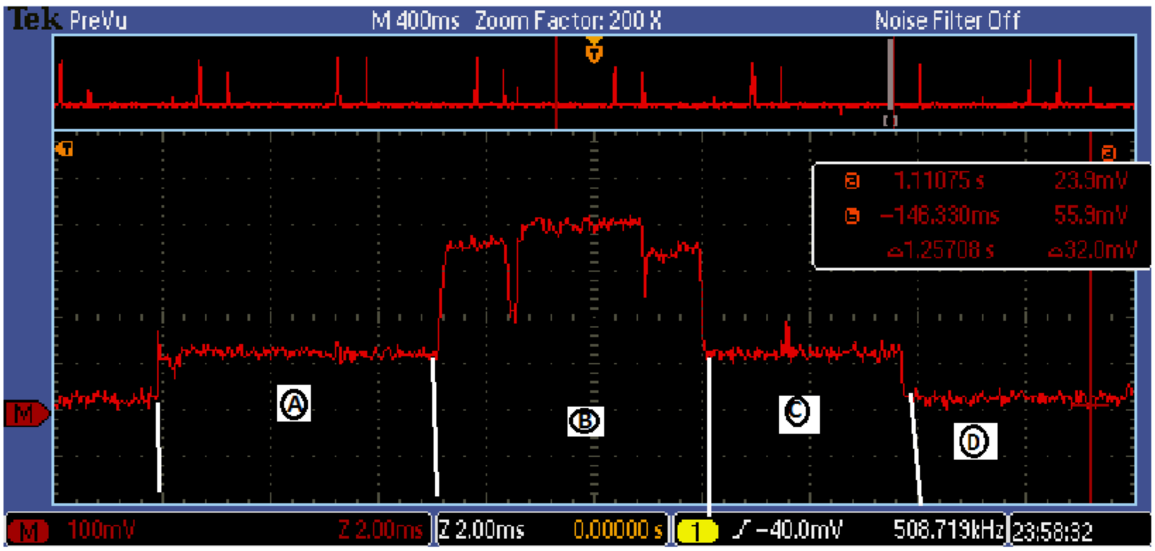


Figure 5-6. Current Profile on End Device in ZigBee network trying to search for network

Table 5-9. Current Profile interpretation of Figure 5-6

Region	Time	Current(mA)	Amp*sec
A -Wake up Radio	5.2	11.33	58916nA*s
B -Packet Transmitted over the air	4.9	31.95	156555nA*s
C -Radio sleep	3.59	11.66	41859nA*s
D -MSP430 and CC2530 in sleep mode		2.667	
Total		57.607	

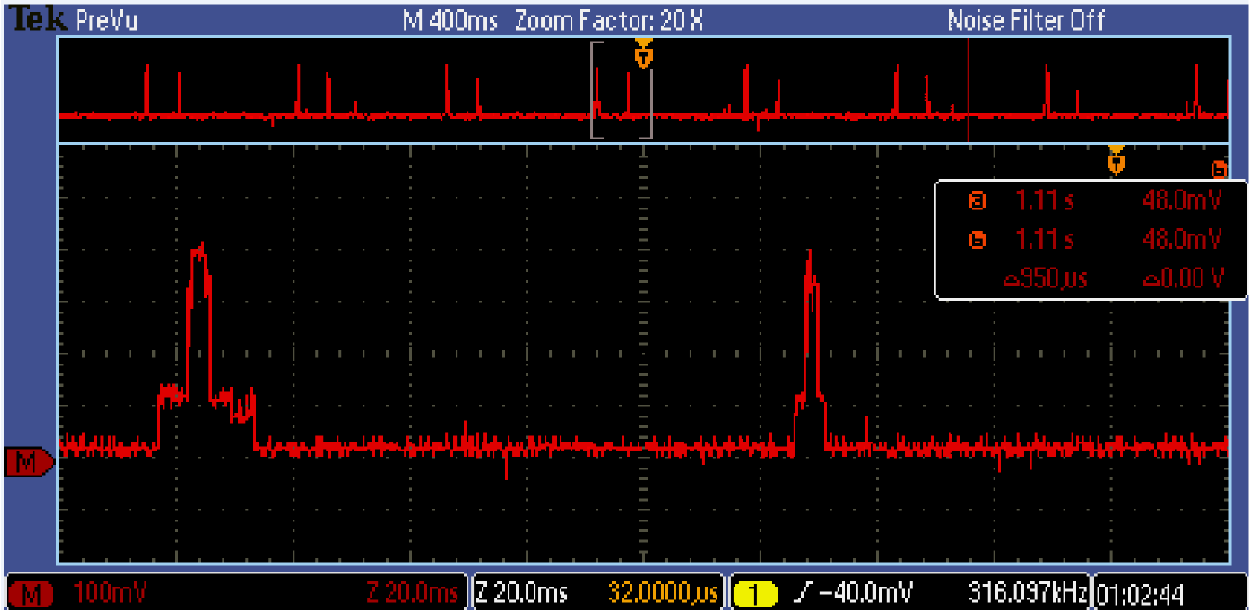


Figure 5-7. Current profile on ZigBee end device with acknowledgement

Along with the current consumption observed in Figure 5-7, there is also the current consumption during reception of acknowledgement. Its current profile is shown in Figure 5-8 and it is interpreted in Table 5-10.

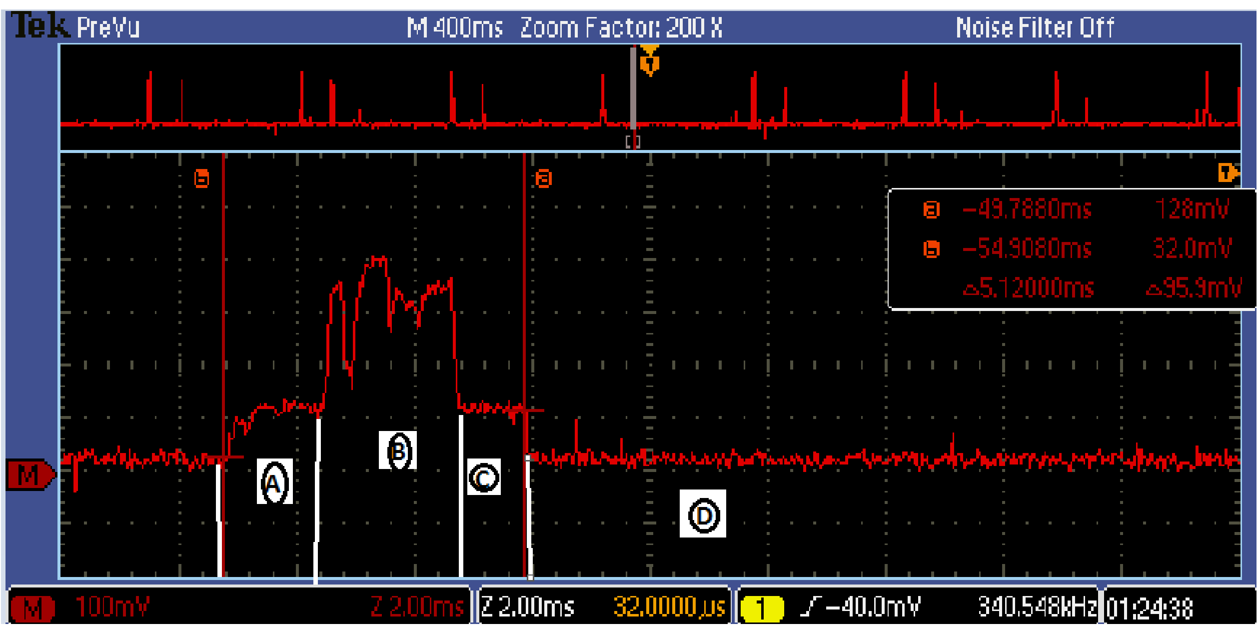


Figure 5-8. Current Profile on ZigBee End Device when receiving acknowledgement after data transmission

Table 5-10. Current Profile interpretation of Figure 5-8

Region	Time (msec)	Current(mA)	Amp*sec
A -Wake up Radio	1.72	11.33	194876nA*s
B -Packet Received over the air	2.24	30.5	6832nA*s
C -Radio sleep	1.24	11.33	14049nA*s
D -MSP430 and CC2530 in sleep mode		2.667	
Total		58.827	

Figure 5-9 shows the current profile when microcontrollers respond to an interrupt from accelerometer. The current consumption when the processor sits idle is approximately 2.583 mA, and 6.83 mA when responding to an interrupt. The LED on the target board blinks when interrupt occurs. So, the current shown in Figure 5-9 is a combination of LED consumption and read from registers of accelerometer.

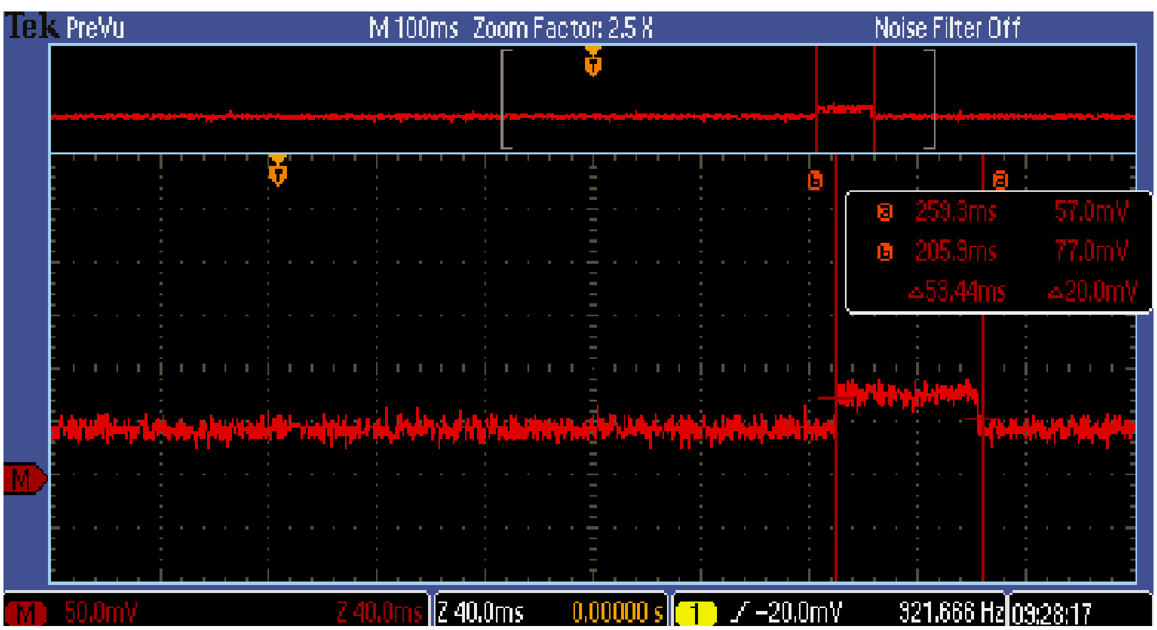


Figure 5-9. Current profile when target board responds to interrupts from sensors

Figure 5-10 is a screenshot of end device receiving a packet from the access point. The reception time is 895-920 msec with average current consumption of 10-11.14 mA.

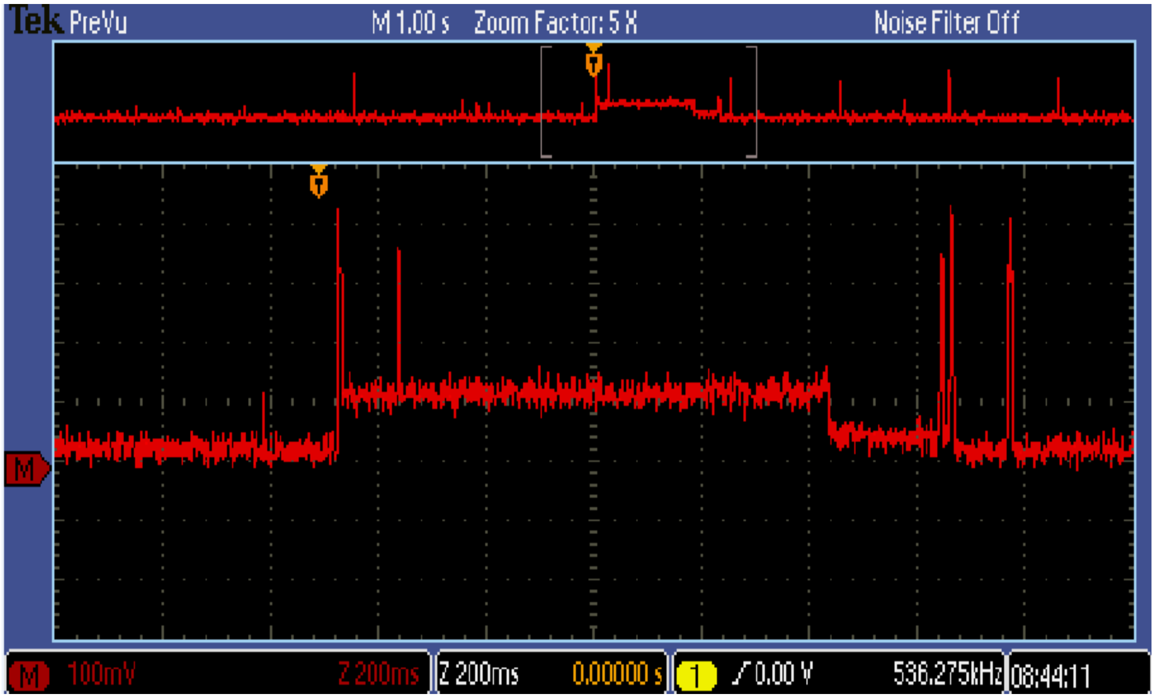


Figure 5-10. Current profile when target board receives data from Access Coordinator

The target boards run on AAA cells of 1000ma rating. Table 5-11 describes the battery life when using end devices under a few scenarios. Range of Communication and packet loss rate are shown in Tables 5-12 and 5-13.

Table 5-11. Life Expectancy of Battery in devices of monitoring system using ZigBee Network Processor

Device	Scenario	Battery Life(hours)
End Point	Joins a network and waits for commands.	387.14
End Point	Continuously in state of receiving interrupts from sensor.	222.22
End Point	Continuously receiving commands from access points.	94.607

Table 5-12. Range of Communication variation with change in Transmitting Power of ZigBee devices in direct line of sight

PATABLE settings	TX Power (dBm)	Current (mA)		Range of communication (m)	
		Tx	Rx	Indoor	Outdoor
0xF5	4.5	30.15	10.95	70-75	10-19
0xE5	2.5	29.15	10.9	70-75	7-10
0xB5	-1.5	27.85	10.83	65-70	6-8
0x95	-4	25.77	11.23	15-17	9-11
0x55	-12	26.44	11.61	8-9	10-15
0x15	-20	25.55	10.98	6-8	9-12

Table 5-13. Packet error Rate with different number of nodes in the network with direct line of sight

Distance(m)	1 node		2 nodes			
			Node1		Node2	
	Received	Lost	Received	Lost	Received	Lost
1	1000	0	1000	0	1000	0
2	1000	0	1000	0	1000	0
4	1000	0	1000	0	1000	9
6	999	1	1000	0	1000	0
10	1000	0	1000	0	991	9

5.4 Comparison between SimpliciTI and ZigBee

Through software implementation and experimental results obtained under SimpliciTI and ZigBee hardware and protocols, a comparative study was performed with the results summarized in Table 5-14.

Table 5-14. Comparison between SimpliciTI and ZigBee protocols used for Monitoring system Design

	SimpliciTI	ZigBee
Frequency	2.4GHz	2.4GHz
Based on standards	No	Yes (802.15.4)
Number of nodes	2 - 30	2 - 240
Network topology	Star, peer-to-peer	Star, Tree, Mesh
Device Objects	Access Point, End Point, Repeater	Coordinator, End Device , Router
Current consumption	Low	Low
Battery Life(1000Amphr)	46.5-111.11 hours	94.6 – 387 hours
Range of communication	Indoor -Up to 70m Outdoor – Up to 48m	Indoor -Up to 70m Outdoor – Up to 20m
Sensors	MMA8451,MAG3110	CMA3000-D01
IDE	IAR WorkBench	IAR WorkBench
Packet Loss Rate	High	Low

ZigBee is found to be a better fit for our monitoring system application as the power requirements are low, as well the ability to expand the network in a mesh topology. ZigBee was also found to have a lower packet loss as well as good communication range.

5.5 Future Work

5.5.1 Design of the dual communication system

In this thesis, I have used two different development kits for the demonstration of a monitoring system application to assist the optical communication system during the development and pre-deployment phases of the pipeline installation. The monitoring system with optical link was developed and tested with custom built boards compatible with insertion with the pipeline system during deployment.

An extension to this project can be a monitoring system with dual communication on custom built boards with supporting radios and optical link circuitry. The custom built board can use the same energy harvesting system for optical link, and be supported by battery backup for RF communication and for operations of the system underwater. The radios supporting ZigBee protocol can be used in combination with the optical link as they consume less power as observed from our experiments. Figure 5-11 depicts the dual combination system.

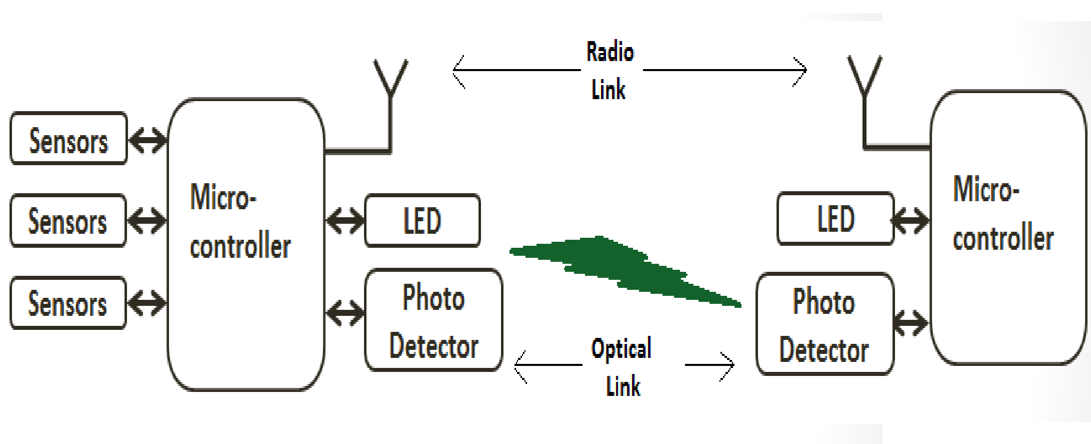


Figure 5-11. Design of Monitoring System with dual combination of Optical and RF links

According to the requirement of the monitoring system, ZigBee can be explored to its full extent by building a customized profile to assist the monitoring system with optical communication for deep water, during development, testing, and pre-deployment phases. If a pressure gauge is added along with existing sensors, the strain of the pipes when sitting on a reel barge can be obtained. The algorithm used for programming can be modified to reduce the power consumption by setting the end devices to poll.

CHPATER 6

SUMMARY

The initial goal of the work was to design a deep water pipeline monitoring system employing optical communication as a means to transfer sensor collected data from the pipeline to an underwater vehicle. However, to meet the constraints during development, testing, and pre-deployment phases of the system as well during pipeline installation, a combination of optical and RF communication was found to be better suited for our application.

In order to meet our constraints, a monitoring system based on RF communication was built based on two low power RF protocols, SimpliciTI and ZigBee. Both protocols were evaluated and our monitoring system application was built using these protocols for sensor data transfer. Application layers using both protocols were developed and demonstrated.

Software protocols, power requirements, battery life expectancies in worst case scenarios, packet loss rate, and the range of communication were measured in both cases by conducting experiments. ZigBee was found to be a better fit for the RF monitoring system considering all the above characteristics. A GUI was designed to send different commands to end devices with sensors in network based on both protocols.

This work can further be extended by building our application in a mesh network utilizing full-length ZigBee features. The algorithms used for RF communication using these protocols can further be revised to lower the power consumption by the end modules.

BIBLIOGRAPHY

[1] PIPELINE INTERNATIONAL- Retrieved July 23, 2012, from http://pipelinesinternational.com/magazine.

[2] Murata Electronics Oy- Retrieved July 23, 2012, from http://www.muratamems.fi/products/accelerometers/cma3000-accelerometers.

[3] OSRAM Opto Semiconductors - Retrieved July 23, 2012, from http://catalog.osram-os.com/catalogue/catalogue.do?favOid=0000000300037a3100ad0023&act=showBookmark.

[4] *CC2530 ZNP Interface Specification* - Retrieved July 23, 2012, from http://zb4osgi.aaloa.org/redmine/attachments/40/CC2530_ZNP_Interface_Specification.pdf.

[5] Ali A. Abed, A. A. (2010). *Building an HMI and Demo Application of WSN-based Industrial Control Systems. Intrenational Conference on Energy, Power & Control*.

[6] *AN4076 - Data Manipulation and Basic Settings of the MMA8451, 2, 3Q*. - Retrieved July 23, 2012, from http://cache.freescale.com/files/sensors/doc/app_note/AN4076.pdf.

[7] *AN4248 - Implementing a Tilt-Compensated eCompass using Accelerometer and Magnetometer Sensors. (n.d.).* Retrieved July 23, 2012, from http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MAG3110.

[8] Angelo Cenedese, G. O. (2010). *Low-Density Wireless Sensor Networks for Localization and Tracking in Critical Environments*. *IEEE Transactions On Vehicular Technology,,* (pp. VOL. 59, NO. 6).

[9] *Application Note on simplciiTI Frequency Agility* - Retrieved July 23, 2012, from www.ti.com.

[10] *Application Note on SimpliciTI Security.pdf* - Retrieved July 23, 2012, from www.ti.com.

[11] Byoung-Kug Kim, S.-H. H.-S.-S. *The Study of Applying Sensor Networks to a Smart Home*. *Fourth International Conference on Networked Computing and Advanced Information Management*.

[12] Carlos E. Otero, I. K. (2011). *Analysis Of Wireless Sensor Networks Deployments Using Vertical Variance Trimming And The Analytic Hierarchy Process*. *International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC)*.

[13] CC2500 - Retrieved July 23, 2012, from http://www.ti.com/product/cc2500, http://www.ti.com/lit/ds/symlink/cc2500.pdf.

[14] PIPELINE INTERNATIONAL - Retrieved July 23, 2012, from http://pipelinesinternational.com/news/continuous_integrity_monitoring_for_pipelines/008029/#.

[15] David V. Brower, N. P. Real Time Subsea Monitoring And Controlsmart Field Solutions. Subsea Rio 2004 Conference.

[16] EETimes. Retrieved July 23, 2012, from http://www.eetimes.com/design/embedded/4007598/Eeny-Meeny-Miney-Moe-choosing-a-low-power-wireless-network-protocol--Part-1?pageNumber=1.

[17] eZ430-RF2500 Development Tool, User's Guide, SLAU227E. Retrieved July 23, 2012, from http://www.ti.com/lit/ug/slau227e/slau227e.pdf.

[18] F. J. L. Ribeiro, A. C. (Jan 2012). *Deepwater Monitoring System Using Logistic-Support Vessels in Underwater Sensor Networks. Latin america Transactions (IEEE) (http://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=9907)*, Volume 10 ,Issue 1 pages 1324-1331.

[19] F. Viani, G. O. (Sept 2010) *WSN-based Solutions for Security and Surveillance. Proceedings of the 3rd European Wireless Technology Conference.Pages 285-288*.

[20] Freescale SemiConductor Application Note AN3397. Retrieved July 23,2012, from http://perso-etis.ensea.fr/~pierandr/cours/M1_SIC/AN3397.pdf.

[21] http://www.customthermoelectric.com. - Retrieved July 23, 2012, from http://www.customthermoelectric.com/powergen/pdf/1261G-7L31-04CL_spec_sht.pdf.

[22] Imad Jawhar, N. M. (n.d.). *A Routing Protocol and Addressing Scheme for Oil, Gas, and Water Pipeline Monitoring Using Wireless Sensor Networks*. Retrieved July 23, 2012, from http://faculty.uaeu.ac.ae/Nader_M/papers/WOCN2008.pdf.

[23] Kulkarni, S. S. (Third Quarter 2011). *Energy Harvesting Sensor Nodes: Survey and Implications.* (pp. VOL. 13, NO. 3). *IEEE Communications Surveys & Tutorials*.

[24] Low_Power_RF_Solutions_Workshop - Retrieved July 23, 2012, from http://processors.wiki.ti.com/index.php/Low_Power_RF_Solutions_Workshop.

[25] MAG3110. - Retrieved July 23, 2012, from http://www.freescale.com/files/sensors/doc/data_sheet/MAG3110.pdf.

[26] MMA8451 DataSheet. Retrieved July 23, 2012,from freescale : http://www.freescale.com.

[27] MSP430F2274, SLAS504F - Retrieved July 23, 2012, from http://www.ti.com/product/msp430f2274,http://www.ti.com/lit/ds/symlink/msp430f2274.pdf.

[28] MSP-EXP430FR5739 FRAM Experimenter Board - slau343.pdf. Retrieved July 23, 2012, from http://www.ti.com.

[29] Nader Mohamed, I. J.-J. (2011). *Sensor Network Architectures for Monitoring Underwater Pipelines. Sensors Vol11*, http://www.mdpi.com/1424-8220/11/11/10738.

[30] Pedram Radmand\ Alex Talevski, S. P. (2010). *Comparison of Industrial WSN Standards . 4th IEEE International Conference on Digital Ecosystems and Technologies, (IEEE DEST 2010).*

[31] Schematic of LFSTBEB3110 from FreeScale. Retrieved July 23, 2012, from http://cache.freescale.com/files/sensors/doc/support_info/LFSTBEB3110_sch.pdf.

[32] Sensor Fusion. - Retrieved July 23, 2012, from http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MAG3110.

[33] SimplciTI Developers Notes . - Retrieved July 23, 2012, from TI: www.ti.com.

[34] SimpliciTI-brief_tutorial.- Retrieved July 23, 2012, from TI: www.ti.com.

[35] TI.- Retrieved July 23, 2012, from http://www.ti.com/tool/cc2530zdk-znp-mini&DCMP=hpa_rf_cc2530&HQS=Other+OT+cc2530znp.

[36] TI.- EZ430 Rf2500 - Retrieved July 23, 2012, from http://www.ti.com/tool/ez430-rf2500.

[37] TI - MSP430 F2274.- Retrieved July 23, 2012, from http://www.ti.com/product/msp430f2274.

[38] Waltenegus Dargie, C. P. (2010). Fundamentals of Wireless Sensor Networks. A John Wiley and Sons, Ltd., Publication.

[39] X. Liu, J. C. (n.d.). *Fault Tolerant WSN-based Structural Health Monitoring.* 978-1-4244-9233-6/11/ IEEE.

[40] *ZigBee concepts* - Retrieved July 23, 2012, from http://www.jennic.com/elearning/zigbee/index.htm.

[41] *ZigBee Engineering Options*. Retrieved July 23, 2012, from http://www.cirronet.com/pdf/wp_zigbee_engineering_options.pdf.

[42] *ZigBee turotial*. Retrieved July 23, 2012, from http://www.jennic.com/index.php.