

pH Wireless Sensor Network for the meat tenderizing process

by

Vashu Devan



A thesis submitted to Auckland University of Technology in
fulfilment of the requirements for the degree of Master of
Engineering

2010

School of Engineering

Supervisor: Professor Adnan Al-Anbuky

Acknowledgement

I wish to acknowledge and express my sincere thanks and appreciations to the following for all their assistance and support towards my research and thesis:

Professor Adnan Al-Anbuky, Auckland University of Technology, for being my supervisor and mentor. It was through his consistent support, guidance and motivation that lead to the completion of my research and thesis.

Mr. Mihai Costache, Managing Director, Merit of Measurement, New Zealand for sponsoring the project and providing technical guidance.

Foundation for Research, Science and Technology for awarding the programme funding.

Dr Clyde Daly, Carne Technologies Ltd., for arranging the field visit to the Auckland Meat Plant, providing instructions on operations, procedures and guidelines of meat tendrizing systems and proprietary carcass pH analysis algorithm for use in the prototypes.

Meat and Livestock, Australia for co-sponsorship of the project.

Mr Dharam Lingam for proofreading the thesis draft and his invaluable feedback on the content.

Mr Amit Naran for proofreading the thesis before publishing.

Finally my wife Engeleena and children Aashna, Kahveet and Eshaan for their patience, support and encouragement throughout.

Vashu Devan
Student ID: 9048310
2010 Auckland, New Zealand

Abstract

The wireless sensor network is a paradigm shift from the conventional wired system and has made remarkable progress in the last ten years. The system is cost effective, efficient, and user friendly as there is no need for external cables to interconnect devices. There are significant opportunities widely available to assess existing wired systems, and with thorough feasibility studies, most of these could be easily converted into wireless systems. A conceptual pH Wireless Sensor Network based on decentralized architectural paradigm is proposed in this thesis to introduce wireless connectivity and enhance system characteristics of a wired meat tenderising system. The network consists of pH Sensor Nodes and Stimuli Actuator Nodes. The focus of this thesis is the architectural design of these nodes and development of prototypes.

Carcass pH is determined non-intrusively using a proprietary pH analysis algorithm and process. This method enables pH analysis of carcasses in a meat plant without stopping the conveyor. The basis of the design is distributed processing and the collaborative nature of a Wireless Sensor Network. This showed that a network of sensor/actuator nodes could replace the existing wired meat tenderizing system and effectively handle the meat tenderizing process.

The key benefit anticipated from the proposed wireless network node architecture in this thesis, is an intelligent re-configurable system that is compact, modular, cheaper and easier to install. The need for precise and consistent results creates an opportunity for further improvements to signature (spectrum of carcass response to stimuli) sensing and signature analysis algorithm. There is also scope for adding intelligence to the actuator nodes to aid in developing a fault tolerant system with a failsafe mode. While this project is a miniaturised version of real time process control, future studies could target replacement of wired industrial process control entirely with wireless sensor networks.

The objectives of the project were met following the set up of the ZigBee network to simulate meat tenderizing process control, and design of the sensor node and actuator node architecture. A set of standard tools were also determined as part of the project, and are readily available in the market. The major achievement of the project was the development of sensor node and actuator node prototypes, consistent with the expectations of the sponsors and handed over to Merit of Measurement, Auckland.

Table of Contents

ACKNOWLEDGEMENT	II
ABSTRACT	III
TABLE OF CONTENTS	V
LIST OF FIGURES	VII
LIST OF TABLES	IX
STATEMENT OF ORIGINALITY	X
LIST OF ACRONYMS	XII
1 PROJECT OVERVIEW	1
1.1 INTRODUCTION	1
1.2 THE MEAT INDUSTRY	2
1.2.1 <i>Mechanism of meat tenderization using Electrical Stimulation</i>	3
1.2.2 <i>Meat Tenderizing Systems</i>	5
1.2.3 <i>Problems and constraints on the adoption of ES</i>	7
1.3 RESEARCH MOTIVE AND SCOPE FOR WSN	8
1.4 RELATED THEORY FOR SYSTEM DEVELOPMENT	8
1.4.1 <i>Antenna fundamentals</i>	9
1.4.2 <i>Fourier transform</i>	11
1.4.3 <i>Wireless considerations</i>	12
1.4.4 <i>Wireless pH sensor network communication protocol</i>	13
1.4.5 <i>Wireless Network Performance & QoS</i>	14
1.5 SUMMARY	15
2 DEVELOPMENT TOOLS	16
2.1 OVERVIEW ON SELECTION OF TOOLS	16
2.2 MATLAB	16
2.3 OPNET	17
2.3.1 <i>Model Features: ZigBee</i>	17
2.3.2 <i>ZigBee Statistics</i>	18
2.4 PACKET SNIFFER	18
2.5 SOFTWARE APPLICATION DEVELOPMENT TOOLS	19
2.6 HARDWARE DEVELOPMENT PLATFORMS	21
2.6.1 <i>DSK 6713</i>	21
2.6.2 <i>eZ-RF2480</i>	22
2.6.3 <i>CC2430 Demonstration Board</i>	23
2.7 SUMMARY	24
3 ARCHITECTURE OVERVIEW	25
3.1 pH WSN ARCHITECTURE	25
3.2 GENERIC NODES	26
3.2.1 <i>pH Sensor Node</i>	27
3.2.2 <i>Stimuli Actuator Node</i>	27
3.3 COMMUNICATION SYSTEM	28
3.3.1 <i>ZigBee Protocol</i>	28
a. Application Profile	28
b. ZigBee Endpoint registration	29
c. ZigBee Application Framework	30
3.4 SUMMARY	31

4	PH SENSOR NODE DETAIL DESIGN	32
4.1	WIRELESS NODE DESIGN	32
4.2	PH SENSOR NODE	32
4.3	STIMULI ACTUATOR NODE	33
4.4	PH SIGNAL ANALYSIS	35
4.5	WIRELESS RADIO COMMUNICATION	36
4.5.1	<i>Communication protocol</i>	38
4.6	PH SENSOR NETWORK APPLICATION FRAMEWORK	38
4.6.1	<i>Tasks & Function Calls</i>	39
4.6.2	<i>Process of Developing a Private Application Profile: pH Sensor Network.</i>	41
4.7	PH SENSOR NODE AND STIMULI ACTUATOR NODE PROTOTYPE	42
4.8	SUMMARY	44
5	PH SENSOR NETWORK MODELLING AND IMPLEMENTATION	45
5.1	CARCASS REFLEX SIGNAL PROCESSING & PH ANALYSIS	45
5.2	PH SENSOR NETWORK SIMULATION USING OPNET	50
5.3	PH SENSOR NETWORK SIMULATION ON EZ-RF2480 KIT	54
5.3.1	<i>eZ-RF2480 target board program</i>	55
5.3.2	<i>eZ-RF2480 simulation results</i>	57
5.3.3	<i>OPNET and eZ-RF2480 simulation comparison</i>	58
5.3.4	<i>Packet analysis using Packet Sniffer</i>	58
5.4	SUMMARY	61
6	CONCLUSION AND FUTURE WORK	62
6.1	CONCLUSION	62
7	REFERENCES	64
8	APPENDICES	69
8.1	APPENDIX 1 CD CONTENT	69
8.2	APPENDIX 2 'FFT2PH' FUNCTION AND LOAD CELL DATA	70
8.3	APPENDIX 3 CONFIGURING A ZIGBEE DEVICE IN Z-STACK	72

List of Figures

Figure 1.1: Temperature and pH profile of a normal carcass (Source: [15]).....	4
Figure 1.2: SmartStim system architecture	5
Figure 1.3: Stimulation versus Chilling rate (source: [10])	6
Figure 1.4: Processing opportunities (Source: [13]).....	7
Figure 1.5: Types of Antennas, PCB antenna shown was designed for the stimuli actuator node	11
Figure 1.6: ZigBee Stack Architecture (Source: [9]).....	14
Figure 2.1: Screen shot of decoded packet using Packet Sniffer	19
Figure 2.2: Description of fields of the decoded packet frame in Packet Sniffer	19
Figure 2.3: Host MCU and CC2480 interface.....	21
Figure 2.4: DSK 6713 Block diagram (Source: [33]).....	22
Figure 2.5: eZ-RF2480 target board	23
Figure 2.6: CC2430DB Development board.....	23
Figure 3.1: Conceptual pH WSN Architecture	25
Figure 3.2: Key components of a Generic Wireless Node.....	27
Figure 3.3: Communication System	28
Figure 3.4: pH Wireless Sensor Network Application Profile.....	29
Figure 4.1: pH Sensor Node functional modules	33
Figure 4.2: Stimuli Actuator Node functional modules.....	34
Figure 4.3: Generic Communication Packet design	37
Figure 4.4: Command packet	37
Figure 4.5: Sensor Node Application Framework.....	39
Figure 4.6: Sensor Node stack: Z-STACK & CC2420 Transceiver.....	40
Figure 4.7: pH Sensor node prototype block diagram.....	43
Figure 4.8: pH sensor node prototype PCB	43
Figure 4.9: Stimuli Actuator node prototype block diagram	44
Figure 4.10: Stimuli Actuator Node prototype PCB	44
Figure 5.1: pH analysis algorithm	46
Figure 5.2: SmartStim meat tenderization process.....	47
Figure 5.3: Beef carcass response to test pulses at Load cell 1.....	47
Figure 5.4: Beef carcass response to test pulses at Load cell 2.....	48

Figure 5.5: pH analysis on DSP platform	49
Figure 5.6: Lamb carcass response	49
Figure 5.7: OPNET Simulation – pH sensor network	50
Figure 5.8: Trajectory for pH sensor node	52
Figure 5.9: Packets sent by pH sensor node	52
Figure 5.10: Power = 50mW, Sensitivity = -95dBm	53
Figure 5.11: Power = 1mW, Sensitivity = -92dBm (CC2480 Parameters)	54
Figure 5.12: eZ-RF2480 block diagram.....	54
Figure 5.13: pH sensor network simulation setup	55
Figure 5.14: Node_1 program flowchart.....	56
Figure 5.15: Node_2 program flowchart.....	56
Figure 5.16: Node_3 & 4 program flowchart.....	57
Figure 5.17: LQI Chart	57
Figure 5.18: Packets captured by Packet Sniffer tool.....	59
Figure 5.19: SmartStim Installation site at Auckland Meat plant (Source: [17]).....	60
Figure 8.1: Commands for configuring a node as coordinator in a ZigBee network	74
Figure 8.2: Using Simple API to send data/command.....	75
Figure 8.3: Polling method to receive data/command	75

List of Tables

Table 1.1: Types of short range device antennas and properties.....	11
Table 1.2: Wireless network performance parameters	15
Table 4.1: Network message/command design.....	42
Table 5.1: Analysed pH values.....	50
Table 5.2: Node description for sensor network in Figure 5.7.....	51
Table 5.3: Global statistics.....	51
Table 5.4: Node configuration.....	55
Table 8.1: Load cell data for beef carcass (Source: [17]).....	70

Statement of Originality

‘I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the qualification of any degree or diploma of a university or other institution of higher learning, except where due acknowledgement is made in the acknowledgments’.

_____ (Signed)

_____ (Date)

*To my beloved parents
Achaiya and Kamachi*

List of Acronyms

ADC	Analog to Digital Converter
AF	Application Framework
API	Application Programming Interface
APS	Application Support Sub-Layer
DAQ	Data Acquisition
DSP	Digital Signal Processing
ES	Electrical Stimulation
FFD	Fully Functional Device (ZigBee Device)
FFT	Fast Fourier Transform
I ² C	Inter-Integrated Circuit
IDE	Integrated Development Environment
LED	Light Emitting Diode
LQI	Link Quality Indicator
MAC	Medium Access Layer
MCU	Microcontroller
NWK	Network Layer
OTA	Over-The-Air (RF transmission)
PC	Personal Computer
pH	Potentiometric hydrogen ion concentration
PHY	Physical Layer
PSE	Pale, Soft, Exudative
QoS	Quality of Service
RF	Radio Frequency
RFD	Reduced Function Device (ZigBee Device)
RSSI	Received Signal Strength Indicator
RTOS	Real Time Operating System
SoC	System-on-Chip
SPI	Serial Peripheral Interface
TI	Texas Instruments
WSN	Wireless Sensor Network
ZDO	ZigBee Device Object

1 Project Overview

Approximately eighty five per cent (85%) of the world's population eat meat to supplement their protein intake. Most countries, especially in the Middle East and smaller Pacific islands, are fully dependent on imports. This offers great opportunities to meat producing countries, like New Zealand and Australia, to capture the major share of the global market. This can only be achieved through competitiveness and quality of meat produced and processed for export. While assessing the current technology through literature research on wired control systems (cables and complex wiring) and subsequent visits to a meat processing plant, the concept of a wireless sensor network was generated.

An overview of the development of the pH (potentiometric hydrogen ion concentration) sensor network for monitoring and controlling the meat tenderizing process is discussed in this chapter. Any system development process begins with an understanding of the core technologies that are presently available and the theoretical basis on which they are built. Aspects of signal processing and wireless communication system are also highlighted and applied to the design and development of the pH sensor network system. The meat industry background is presented to signify the importance of adopting technology that aids in delivering high quality and consistent meat and meat products.

1.1 Introduction

The technologically dormant meat processing industry is beginning to take advantage of emerging technologies that enhance meat quality. Over three decades, individual or a combination of techniques such as chemical, mechanical, electrical stimulation, conditioning, and ageing have been exploited by researchers to improve meat quality [1-7]. This evidently prompted the Australia and New Zealand meat industries to develop quality standards and procedures for the Producers and Processors to produce quality and consistent product, however only a small percentage of the retail meat industry complied with the quality standards [8]. Meat tenderness could be used as the benchmark to determine the quality of the meat.

Unfortunately, the strategy for quality (tenderness) indicator on the meat packages is yet to be instigated.

Colour, texture, tenderness and pH are all important factors in determining consumer acceptability of meat [5, 9]. Reaching an ultimate pH within a particular processing window exhibits an improvement in tenderness and quality of meat. This is achieved through the use of electrical stimulation which also contributes significant cost benefits to the Processors. The existing electrical stimulation technology consists of traditionally wired systems, which are costly and difficult to install, or retro fit into new and existing meat plants. The pH wireless sensor network is an innovative technology which involves integrating wireless technology into these wired systems which will introduce flexibility and easier installation with reduced cost. The conceptual pH wireless sensor network described in this thesis is the result of scrutinizing an electrical stimulation system with the motive for application of WSN in meat processing. The outcome of the study resulted in the development of pH sensor node and stimuli actuator node prototypes, the exhibits of which are submitted in the CD enclosed [Appendix 1].

1.2 The Meat Industry

Tenderness is considered the most important palatability characteristic of meat. Therefore a critical component of processing meat is to ensure that meat reaches the optimum level of tenderness before consumption. Meat is ‘aged’ up to five days in chillers to improve tenderness before being released to the market for consumption, which constitutes significant cost to the industry. Tenderization can be accelerated by electrical stimulation, whereby currents are used to trigger muscle contractions and accelerate the depletion of muscle energy. *Rigor mortis*, the point at which all muscle energy is depleted, can be accelerated from 12-24 hours post mortem to as little as 2 hours. Reaching *rigor mortis* sooner while the muscles are still warm after slaughter means that the tenderization process is significantly accelerated.

The survival and competitiveness of the New Zealand and Australian meat industries on the global market requires the industries to maintain high standard of quality and consistent product. For example, New Zealand meat is exported to a diverse market. Logistical issues, such as distance and time, can have adverse effects on meat characteristics. Therefore if processing is manipulated to cater for the difference in

distribution chain then optimal product can reach each market [10]. This motivated the Meat & Livestock Australia and Meat & Wool New Zealand to fund programs with the objective to develop cost-effective and flexible technologies to meet the requirements for meat quality [10]. A number of strategies were developed in an effort to create new commercial opportunities for both Producers and Processors. Among these strategies was measurement of meat quality during processing, systems to interpret measurement, and feedback systems that provide information back to the Producers [10]. The outcome was the development of processing methods that resulted in following technologies: Hot boning, Immersion chilling, Muscle stretching and forming, control of post mortem muscle pH decline, and feedback stimulation [10].

NZ Beef and Lamb Quality Mark standard [11] was a New Zealand initiative to introduce a tenderness specification of retail meat. Quality Mark guidelines describe appropriate procedures to handle animals, stunning, ES and chilling conditions. Research conducted on its impact by survey of retail beef, pork and lamb from 1997 to 1999 showed an improvement of tenderness in beef by 21.9% and lamb 17.2% [8].

Similarly a study in Australia suggested that a large number of lamb carcasses do not comply with the Australian Sheep Meat Eating Quality [12]. Based on their findings the authors suggested that electrical stimulation need to be adopted to raise the percentage of carcass compliance to SMEQ. However the factors affecting adoption of electrical stimulation technology are cost, Occupational Health and Safety and difficulty of retrofitting.

1.2.1 Mechanism of meat tenderization using Electrical Stimulation

The pH scale is used as a reference to measure the degree of acidity or alkalinity present in the carcass as illustrated in Fig 1.1. The neutral point on the scale is 7, which is the pH of pure water. Lower pH numbers indicates increasing acidity. Living animal muscle pH is in the range 6.8-7.2. When the animal dies, the pH falls until 24-48 hours after death and reaches 5.6-5.8. This is referred to as ultimate pH. The recommended window is pH of 6.0, between 18°C and 25°C [3].

Numerous studies undertaken globally have shown enhancement in meat quality (tenderness and colour) due to application of high or low voltage electrical stimulation. Meat tenderization is achieved by passing electric current through the carcass to accelerate the rate of pH decline [13, 14]. This process is known as electrical stimulation (ES). However there is still uncertainty as to the exact mechanism by which ES causes tenderness. The most noticeable effect of ES is that it induces vigorous muscular contractions hence increased muscular energy expenditure.

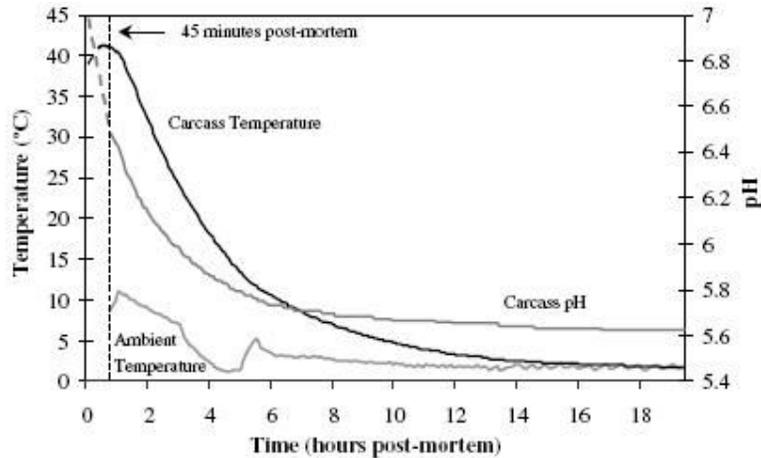


Figure 1.1: Temperature and pH profile of a normal carcass (Source: [15]).

Various researches have either postulated or concluded that ES increases the rate of post mortem glycolysis (the process by which glycogen is broken down to lactic acid) which leads to the rapid decline in muscle pH. The early onset of rigor mortis before the carcass temperature reaches 15°C minimises cold shortening. Cold shortening is a condition, where muscles are exposed to temperatures below 15°C before they have gone into rigor. This causes the muscles to shorten and is likely to be tough. ES causes a release of certain enzymes into the muscle, which break down the protein in the muscle, and physical disruption of muscle structure. Collectively these mechanisms influence meat tenderness, colour, flavour and shelf-life [3, 7, 10, 15]. The end result of the process and the technology is that the meat industry and consumers benefit from uniform, consistent and desirable product.

1.2.2 Meat Tenderizing Systems

There are two known categories of meat tenderizing systems, electrical and non-electrical. The electrical systems are high voltage and extra low voltage. The stimulating period is between 40-90 seconds. The non-electrical systems in use are Mechanical treatment (Stretching or fixation), Wrapping (Tight plastic cling film) and Pi-Vac (Elastic Plastic Tube). Most non-electrical methods are more effective in achieving tenderness but at the cost of prolonged processing [16].

A further sub category of electrical meat tenderizing system is non-intrusive systems. Non-intrusive technologies currently available can only measure meat tenderness. These are near-infrared spectroscopy, ultrasound, image analysis and Colorimeter [5]. A more advanced non-intrusive system is an integrated measurement and tenderizing system. This system can measure pH non-intrusively and at the same time tenderize meat using ES. This is a unique system called SmartStim developed by Carne Technologies Ltd in New Zealand [17]. It provided the platform for the application of the wireless pH sensor network proposed in this thesis. The architecture of the SmartStim system is shown in Figure 1.2.

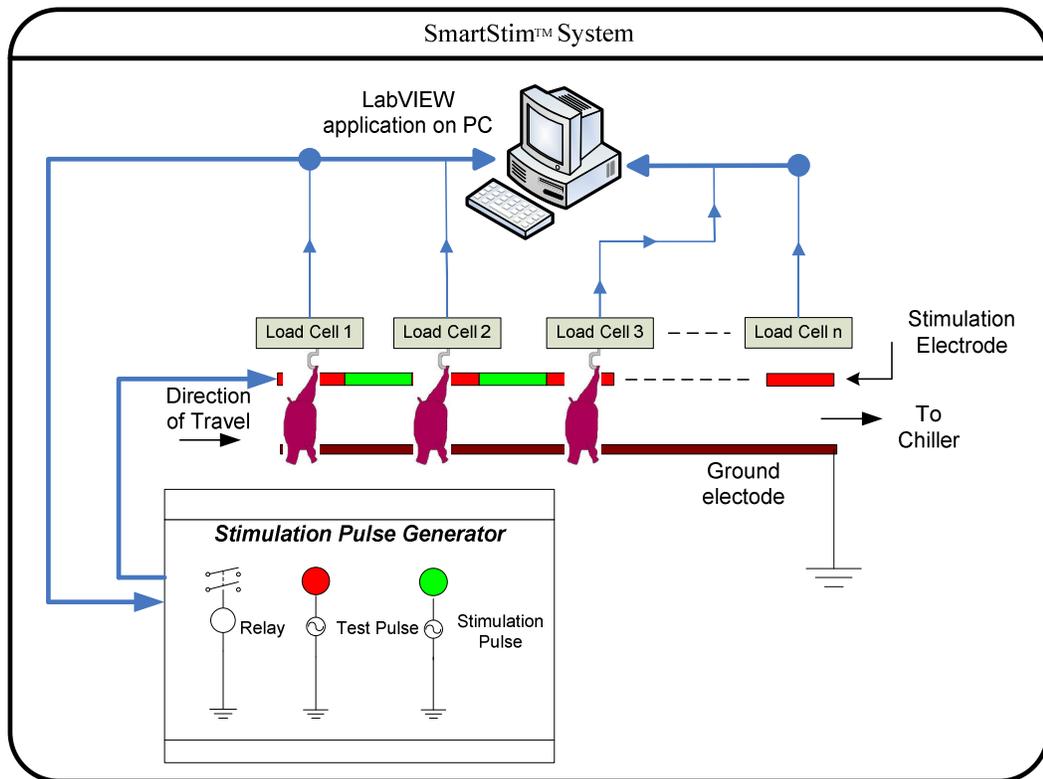


Figure 1.2: SmartStim system architecture.

The system consists of three major components, the SmartStim application software, stimulation electrode, and pulse generator. The carcass is carried on the conveyor and makes contact with the stimulation electrode and ground electrode. The stimulation electrode is segmented. The red segments are used to apply test pulses, and the green segments to apply stimulation pulses. The ground electrode provides the return path for the electric currents. The carcass response to these pulses is sensed using load cells. The application software running on the PC controls the whole process of pH analysis, stimulation and data storage.

The potential commercial benefits of ES are: reduced inventory for ageing the meat (usually 2-5 days for frozen product), reduced product shrinkage (between 0.5-1% of product weight during storage of chilled product), and increased retail display life in supermarkets (up to 1 day for chilled export lamb) [18]. A further benefit is carcass grading, so that different grades can be used for different end products such as fresh packed, sausages, vacuum packing and export quality. Figure 1.3 shows the processing space for creating commercial opportunities for Producers and Processors. It is of critical importance that there is a balance between the stimulation and the chilling rate, which is to be achieved in the space shaded green. If processed more towards the fast rigour process in the red zone would exacerbate PSE (Pale, Soft Exudative) meat and processing in the slow rigour process towards cold short would result in shrinking of muscles and toughening of the meat.

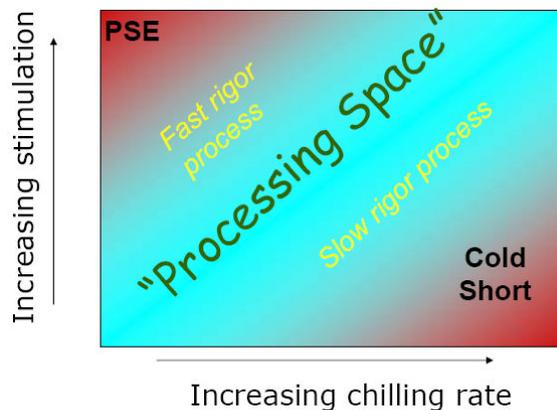


Figure 1.3: Stimulation versus Chilling rate (source: [10]).

Figure 1.4 below further explains the opportunities for fast and slow rigor processes. The processing opportunities that exist for processors and technology innovators, as shown in Figure 1.4, indicates that ES eliminates unnecessary toughening of meat with

only moderately effective commercial chillers. For example, a local supplier may opt for ES and therefore will reduce inventory for ageing meat and adopt moderate chilling requirements.

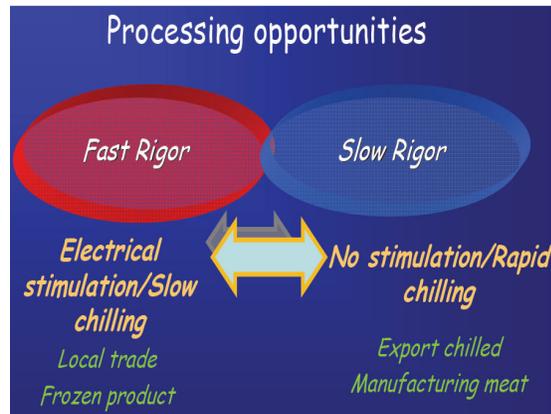


Figure 1.4: Processing opportunities (Source: [13]).

1.2.3 Problems and constraints on the adoption of ES

Other forms of electrical inputs are applied at various stages on the slaughter floor. These are Electronic Stunning, Immobilization and Bleeding. The additive effect of electrical inputs at each point on the slaughter floor in addition to ES may increase the risk of damaging the meat quality due to excessive ES [15]. Geesink and others concluded that pH fall should be monitored immediately after slaughter and electrical parameters of the ES system be adjusted accordingly to optimize meat quality characteristics [4, 15].

Initial adoption of ES technology in the meat industry has experienced a slow response. The main reasons for lack of adoption were cost, Occupational Health and Safety, and the difficulty of retrofitting legacy systems. In recent years use of ES systems is becoming widespread due to advancements made in ES technology and new research has substantiated its usefulness [18]. ES systems have remained primitive while leading edge technology is available to develop intelligent systems through leveraging research findings and recommendations. These systems have a fixed duration of 40-90 seconds of stimulation and ES parameters are selected in relation to the size of plant and animal (beef, sheep). There is no further processing or measurement done to determine the pH or tenderness. Hence it is assumed that the carcass has been appropriately treated based on empirical data.

1.3 Research Motive and Scope for WSN

The idea and motivation for this research is derived from studying the SmartStim meat tenderizing system, which is a completely wired pH analysis system. This study resulted in the formulation of a conceptual pH wireless sensor network to analyse pH and manage the meat tenderizing process. This is the focus of this research and discussion. The application environment in this study is confined to the SmartStim system. However the concept is generic and can be easily configured for a number of application areas.

The most important and obvious advantage of WSN is that it is wireless and dramatically reduces time and cost in the system wiring. Moreover, wireless systems are extremely versatile. The SmartStim architecture shown in Figure 1.2 has wired connections. The notable feature of the architecture is that it does not support portability and scalability. A wired connection requires a physical wire to be laid to the desired location of the central controller. This incurs significant materials and labour costs. Similarly relocating the controller or expanding the system will add additional costs. Long lengths of signal cable can easily pick up noise through inductive and capacitive coupling. Loss of signal quality and excessive variability of samples or data points are forms of noise that can degrade system reliability and performance.

In general a rigid platform such as SmartStim illustrated in Figure 1.2 can be transformed into a compact and modular design by integrating with WSN. The outcome is small form factor, ideal for space constrained installations. Other benefits include, portable nodes that can be located in the vicinity of the sensing element and reduced sensor cable lengths thus reduced signal degradation. System expansion can be facilitated effortlessly by the addition of generic nodes that find and join the network automatically. The resulting architecture is an intelligent re-configurable system that is compact, modular, cheaper and easier to install. This can be made as retro fits or can be included in the design of a new plant.

1.4 Related theory for system development

The most prominent feature that justifies the motive for this project is wireless communication and control. It is essential to highlight the theoretical basis upon which the system is built. Radio, Antenna and Fourier theory discussed below provide the

platform for the development of the pH sensor network for the management of the meat tenderizing process. Since a pH sensor network approach is adopted to address the issues identified in this project, the IEEE 802.15.4 compliant radio is the choice for the RF front end that operates in the unlicensed ISM band. The additional benefits of using standards built on IEEE 802.15.4 such as ZigBee are readily available Zigbee compliant software stacks and IEEE 802.15.4 compliant RF devices from many semiconductor vendors.

The output power, Transmit/Receive current, receiver sensitivity and range are typical parameters considered for RF applications [57]. Antennas also influence the effective range hence types and characteristics of few antennas for short-range devices will be discussed here. Range is one of the most important parameters of any radio system. Data rate, output power, receiver sensitivity, antennas and the intended operation environment all influence the practical range of the radio link. The communication range that can be achieved in a radio system depends very much on the antenna solution [60, 61].

1.4.1 Antenna fundamentals

a. Antenna gain

The antenna gain describes the antenna's ability to radiate power in a certain direction when connected to a source. The gain in the direction of maximum radiation is:

$$G = D\mu m \quad (1)$$

Where, D is directivity, μ is efficiency and m is mismatch loss.

The gain is usually expressed in dBi or dBd. Here dBi means that the directivity D, is measured compared to an isotropic radiator (equal radiation in all directions). dBd is used when D is referring to the directivity of a dipole antenna. A dipole antenna itself has a gain above that of an isotropic radiator, which can be expressed as dBd = 2.15 dBi [62].

b. Range: Friis-Equation

Range in radio communication is generally described by Friis-Equation. Range is the distance at which the link is operating with a signal level equal to the receiver sensitivity level.

$$P_R = P_T \frac{G_T G_R \lambda^2}{(4\pi)^2 d^n} \quad n = 2 \quad (2)$$

P_R : Power available from receiving antenna.

P_T : Power supplied to the transmitting antenna.

G_R : Gain in receiving antenna.

G_T : Gain in transmitting antenna.

λ : wavelength, where $\lambda = \frac{c}{f}$, c = speed of light and f = frequency.

d : Distance.

c : Speed of light in vacuum.

n : path loss exponent, experimentally determined ($n=2$ for free space).

Using Friis-Equation, in free-space the path loss is 80.2dB over a distance of 100m when operating at 2.445MHz. The Friis-Equation is often referred to as the link budget [65].

c. Directivity

The *directivity* of the antenna describes the radiation pattern. The antenna can “see” and radiate better in some directions than others. In this application the transmitter and receiver placement is likely to encounter signal path obstruction due to the nature of industrial buildings. Hence the radiation needs to be equal in all directions. That is, the antenna is to be *Omnidirectional*.

d. EIRP

Effective Isotropic Radiated Power (EIRP) is a term used to describe the effective radiated power from an antenna taking the gain of the antenna into account.

$$EIRP = GP \quad (3)$$

Where, G is the antenna gain and P is the output power from transmitter.

e. Efficiency

The most important term when talking about small antennas is the *efficiency*. The efficiency expresses the ratio of power radiated from the antenna and the power dissipated in the antenna structure (heat) [65].

The efficiency is

$$\eta = \frac{R_r}{R_d + R_r} \quad (4)$$

R_r : radiation resistance (wanted).

R_d : dissipation resistance (unwanted).

f. Types & Features of antennas

There are several antenna types to choose from when deciding on the type of antenna to be used in an RF product. The size, cost, and performance are the most important factors when choosing an antenna. The three most commonly used antenna types for short range devices are, PCB antennas, chip antennas, and whip antennas with a connector. These are shown in Figure 1.5. The properties of these antennas are listed in Table 1.1.

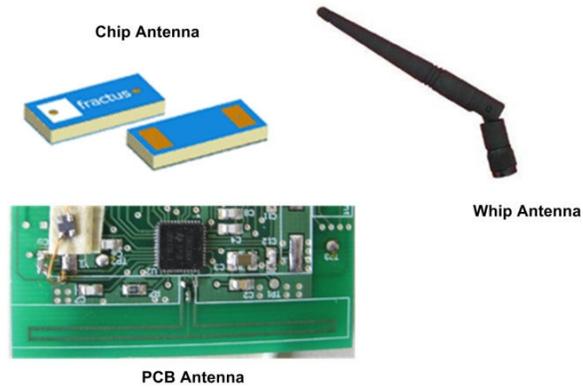


Figure 1.5: Types of Antennas, PCB antenna shown was designed for the stimuli actuator node.

<i>Antenna Types</i>	<i>Benefits</i>	<i>Weaknesses</i>
PCB Antenna	Low cost Good performance is possible Small size is possible at high frequencies	Difficult to design small and efficient PCB antennas Large size at low frequencies
Chip Antenna	Small size	Medium performance Medium cost
Whip Antenna	Good performance	High cost Not suitable for space constraint installations

Table 1.1: Types of short range device antennas and properties.

1.4.2 Fourier transform

The Fourier transform is the mathematical procedure to convert a signal in the time domain into its frequency domain counterpart. The inverse Fourier transform then

returns the frequency representation back to the time domain. Therefore it is possible to form any continuous function $f(x)$ as a summation of a series of sine and cosine terms of increasing frequency.

However to apply this for real world applications Discrete Fourier Transform (DFT) is used. A continuous time domain signal $x(t)$ is sampled in uniform steps giving a discrete time domain signal $x[n]$. For N discrete samples, the N -point DFT is given by

$$X[k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-\frac{j2\pi kn}{N}} \quad (5)$$

For $k, n = 0, 1, 2 \dots N-1$

The number of complex multiplications and additions required to implement Eq. (5) is proportional to N^2 . Therefore it is very inefficient algorithm for computation on a digital signal processor. The Fast Fourier transform (FFT) algorithm reduces the multiplication and additions to $N \log_2 N$. Radix-2 FFT algorithm is widely used for digital signal processing [42, 58].

1.4.3 Wireless considerations

The essential requirements for wireless technology in an industrial environment are identified below.

a. Reliability

Signal Reliability between Transmitter/Receiver is affected by the following:

- Path loss (wall, obstructions)
- RF Interference (ambient noise)
- Transmission power

Collectively these factors determine the bit error rate.

b. Adaptability

The network should adapt to the existing environment. If there are large structures that obstruct the signal path then communication should be routed via alternate path to its destination.

c. Scalability

The wireless network should be scalable. The network should expand effortlessly when more nodes are added to accommodate environment changes such as factory floor plan changes or addition of new plant and machinery.

d. Choice of Network

It is important to identify the best network topology for the application. The topologies are listed below.

- Point-to-Point Links
- Point-to-Multipoint Links (1 base station or access point)
- Mesh
 - More nodes = greater reliability
 - Self-Configuring
 - Self-Healing
 - Scalable to thousands of nodes

A feasibility study of the installation environment needs to be carried out to address the above issues. The more popular protocols for wireless sensor network are ZigBee, Bluetooth, and wireless HART [56]. ZigBee and wireless HART are built on IEEE 802.15.4 standard [37].

1.4.4 Wireless pH sensor network communication protocol

ZigBee is one of the standards developed for wireless sensor networks [48]. It is maintained and published by ZigBee Alliance. ZigBee protocols are intended for use in embedded applications requiring low data rates and low power consumption. It can support 65536 network nodes in a star, tree, or mesh topology. The mesh network topology has three important properties, self-configuring, self-healing and scalability [19]. The ZigBee mesh network forms itself and does not require a manual configuration. Moreover once one node fails, the network reroutes the message through alternative path.

The application framework in ZigBee is the environment in which application objects are hosted on ZigBee devices. Up to 240 distinct application objects can be defined, each identified by an endpoint address from 1 to 240.

This is illustrated in Figure 1.6 below which shows the ZigBee stack architecture. ZigBee provides the network (NWK) layer and the framework for the application layer which is built on IEEE 802.15.4 foundation. The application layer framework consists of the application support sub-layer (APS) and the ZigBee device objects (ZDO). The IEEE 802.15.4-2003 standard [37] defines the two lower layers: the physical (PHY) layer and the medium access control (MAC) sub-layer. There are two PHY layers that operate in two separate frequency ranges: 868/915 MHz and 2.4 GHz. The higher frequency PHY layer is used virtually worldwide [56, 57].

Evidently the ZigBee wireless sensor network developed for industrial monitoring showed good performance and reliability for monitoring non-critical systems [20]. A near real time industrial control and monitoring system with ZigBee was also successfully demonstrated [21]. A closer resemblance to this project is a system proposed using ZigBee, DSP and web services for power monitoring and load control [22].

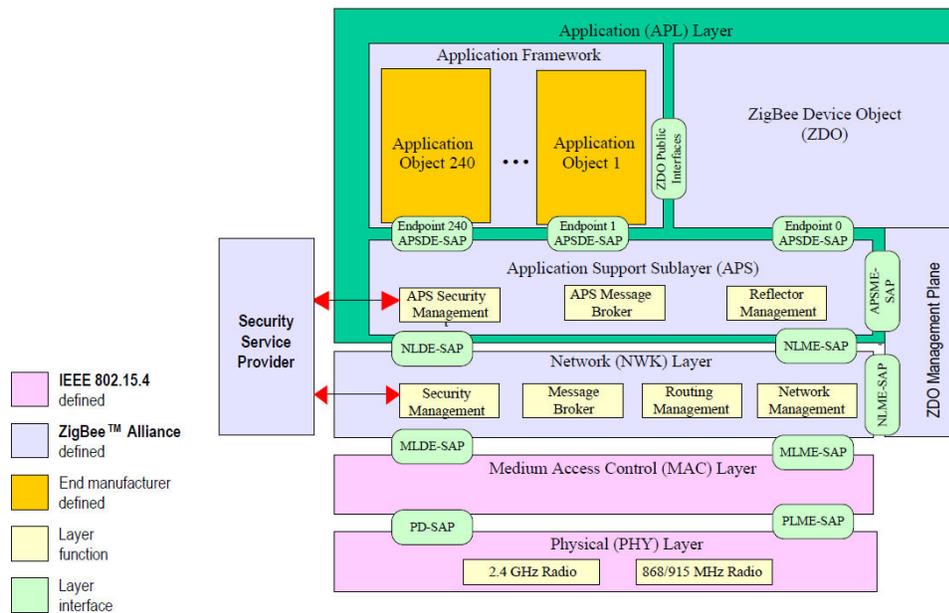


Figure 1.6: ZigBee Stack Architecture (Source: [9]).

1.4.5 Wireless Network Performance & QoS

A measurement of network performance can be defined from characterizing the QoS requirements. In an industrial environment the QoS requirements for a sensor network are: throughput, latency, reliability, security, adaptability and affordability [39] which is

described in Table 1.2. These can then be used to define measurable and quantifiable parameters based on mapping functional needs onto operational requirements. Sensor network simulators can produce all the required parameters to assess sensor network performance based on ZigBee protocol. In addition to the network performance, pH calculation can be quantified as a measure of the total processing time for pH analysis.

<i>Parameters</i>		<i>Description</i>
Throughput	Traffic sent/received (packets/s)	The ability of the network to handle the application traffic in the desired environment.
Latency	End-to-End delay (seconds)	Addresses the timing considerations for the applications including the time delay and variation in the time delay (jitter).
Reliability	Establishing a network (seconds)	Ability of the network to carry information under varying operating conditions such as communication disruptions, power cycling etc.
Security		Ability to prevent unauthorized access or attacks
Adaptability		Ability to adapt to new configurations and new tasks.
Affordability		Cost of ownership, implementation cost, replacement cost and service cost.

Table 1.2: Wireless network performance parameters

1.5 Summary

Chapter 1 provides the general background and framework for the discussion of the related technology to show the significance of the development of pH sensor network for managing the meat tenderizing process. Signal processing techniques are well known with readily available algorithms. Sensor networks are a more recent emerging technology. Sensor network and ES significantly helps in improving tenderizing precision in the plant through the networked control process. This is the essence of WSN, where network of sensor nodes with their own dedicated task collaborate to accomplish control of a complex process.

The project concludes by reviewing the sensor and actuator node architectures. Development of generic sensors, system development and signal processing are part of the discussion. The study further suggests for a collaborative system for process control that would be appropriate to achieve precise, consistent and predictable processing.

2 Development Tools

This chapter introduces the development tools used for proof-of-concept and prototyping. These tools comprise of hardware prototyping tools, software development platforms and simulation packages. The method for generation and collection of specific data is presented here. More detailed description of simulation setup scenario and prototyping is provided in Chapter 5.

2.1 Overview on selection of tools

A wide range of hardware and software tools are available for rapid prototyping and developmental purposes. An informed decision is necessary to select the best tools for the application. The tools specific to this project will be highlighted in this chapter including the development setup and simulation scenarios. A number of factors need to be considered before selecting the tools and these are cost, familiarity, availability, support and training. Trial and code limited versions of software development platforms are available from various semiconductor vendors and third party firms as an option to minimize cost [27, 28, 30]. The learning curve can be quite significant if no prior experience exists and therefore this is accounted for in the project plan. Software tools can be downloaded from the Web but hardware evaluation and development kits can take from 4-12 weeks to arrive due to limited stock held by vendors and suppliers which may run out of stock due to high demand. Hardware lead-time issues are also considered in preparing the project plan. Training and support is an essential requirement to increase productivity. This may be included as part of the research program by the Training Institute with added support from the vendors. Initial consultation was made with staff and fellow students at the AUT for an assessment of access to resource, experience and knowledge base related to the tools for this project. Below are examples of the hardware and software tools used in this research.

2.2 MATLAB

MATLAB is a high-performance language for technical computing that integrates computation, visualization, and programming [23]. In this project MATLAB is used to

simulate the pH analysis algorithm. An m-file is written which has two data files and a pH constant as its arguments. The data files consist of raw carcass response data which is included in the enclosed CD (Appendix 1). It is sampled at 500 Samples/sec using National Instruments 16-Bit data acquisition card [72]. The pH constant and a set of filter coefficients specific to the animal (lamb/beef) and meat plant is proprietary data.

pH calculated using this algorithm is provided in Table 5.1 in chapter 5. The purpose of simulating the algorithm in MATLAB is to use it as a reference for developing the application program for the embedded DSP platform.

2.3 OPNET

OPNET is a software package for network modelling, simulation and analysis of a broad range of networks for fixed and wireless communication technologies [24]. The built in ZigBee library features are outlined below. A number of parameters can be analysed for a desired network configuration. OPNET simulations were performed to produce data by varying receiver sensitivity, transmit power and distance which are discussed in chapter 5 and Figures 5.7-5.9.

2.3.1 Model Features: ZigBee

a. Application layer features

- Generating and receiving application traffic.
- Initiating network discovery and network join.
- Failing and recovering ZigBee devices.

OPNET simulation is also used to further retrieve data from network layer as given below.

b. Network layer features

- Establishing a network.
- Joining a network and permitting network joins.
- Assigning an address.
- Tree routing process.
- Transmitting and receiving data.

2.3.2 ZigBee Statistics

ZigBee statistics can be collected on a global and per-node basis. The statistics is collected as follows.

a. Global Statistics (Full system operation)

- i. ZigBee application statistics
 - Traffic Sent (packets/sec).
 - Traffic Received (packets/sec).
 - End-to-End Delay (seconds).

b. Node Statistics (Sensor/Actuator node only)

- i. ZigBee application statistics
 - Traffic Sent (packets/sec).
 - End-to-End Delay (seconds).
 - Traffic Received (packets/sec).
 - Traffic Dropped (packets).

The meat tenderizing process was simulated using tree network topology. This is done to get an estimate of some of the essential parameters that determine the capability of the network to effectively manage the tenderizing process. These are the global and node statistics produced from running the simulation.

2.4 PACKET SNIFFER

The SmartRF Packet Sniffer [25] is a PC software application used to display and store RF packets captured with a listening RF hardware node. Various RF protocols are supported. The Packet Sniffer filters and decodes packets and displays them in a convenient way, with options for filtering and storage to a binary file format. The Packet Sniffer requires a single CC2400EB [26] as shown in Figure 2.6 to be connected to the PC by a USB cable. Figure 2.1 shows a screenshot of the decoded packets and is further explained in Figure 2.2. During application development for the sensor/actuator nodes, Packet Sniffer was used extensively to view the effects of various device configurations, and related network activity.

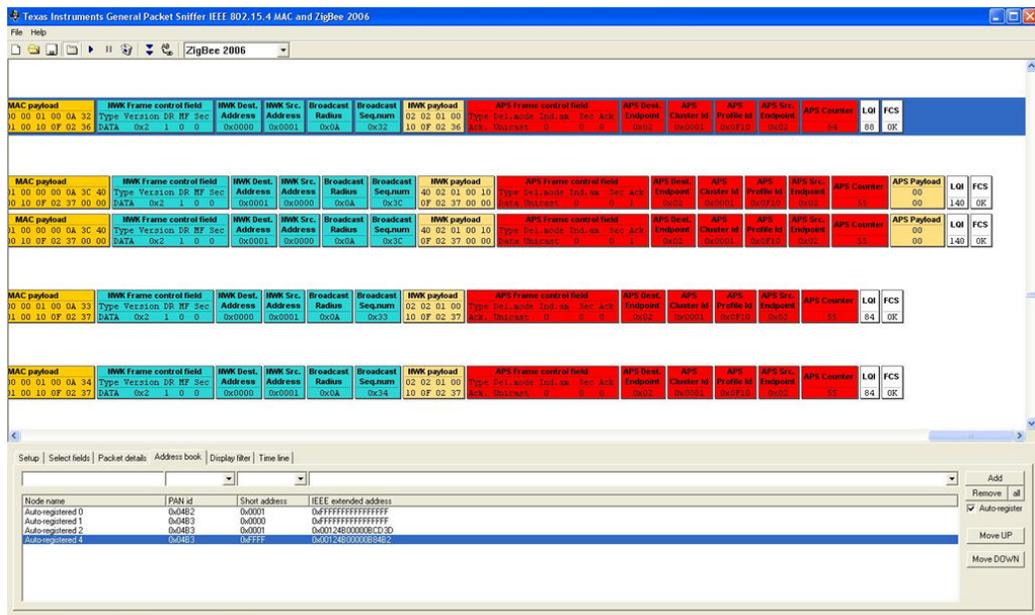


Figure 2.1: Screen shot of decoded packet using Packet Sniffer.

Packet Sniffer decodes the captured packet into frames as shown in Figure 2.2. The frames consist of fields and the format of the fields is specified by IEEE 802.15.4 [37] since ZigBee [48] is built on IEEE 802.15.4. The frame format and details of each field is provided in the ZigBee and IEEE 802.15.4 documents. Packet Sniffer can also provide more detailed information for MAC Payload, Network Layer Information and APS Layer fields.

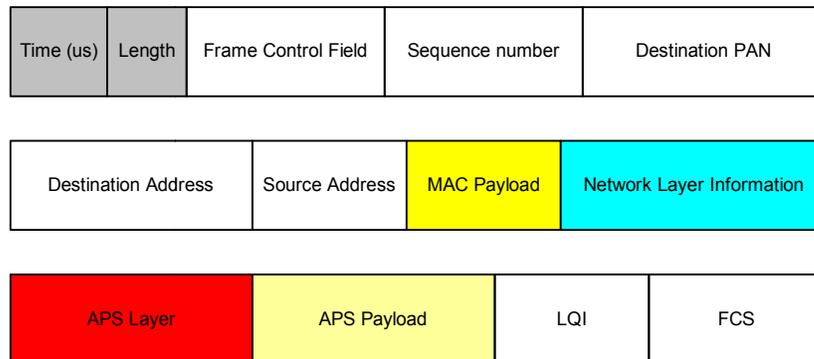


Figure 2.2: Description of the decoded packet frame fields shown in Figure 2.1.

2.5 Software Application Development Tools

a. IAR Kickstart Embedded Workbench

IAR Embedded Workbench is a set of development tools for embedded applications. It integrates the IAR C/C++ Compiler, assembler, linker, librarian, text editor, project

manager, and C-SPY Debugger in an Integrated Development Environment [27]. The IAR Kickstart version for MSP430 devices was used to develop an application program to simulate the meat tenderizing process on eZ-RF2480 platform. The target microcontroller for the application program is MSP430F2274 [35].

b. Code Composer Studio

Code Composer Studio (CCStudio) IDE delivers all of the host tools and runtime software support for the TMS320 DSP based real-time embedded applications [28]. The IDE includes DSP/BIOS support, real-time analysis capabilities, debugger and optimization tools, C/C++ Compiler, Assembler, Linker, integrated CodeWright editor, and a variety of simulators and emulation drivers which were utilized in developing and testing the pH analysis algorithm on the DSK6713 DSP platform.

c. Digital Signal Processing Libraries

The TI C67x DSPLIB [29] is an optimized DSP Function Library for programmers using TMS320C67x devices [40]. It includes C-callable, assembly-optimized general-purpose signal-processing routines. These routines are typically used in computationally intensive real-time applications where optimal execution speed is critical. The TI DSPLIB library includes commonly used DSP routines. Source code is provided that allows modifying functions to match specific needs. The routines contained in the library are first classified in to single and double precision functions and then they are organized into seven different functional categories: Adaptive filtering, Correlation, FFT, Filtering and convolution, Math, Matrix, Miscellaneous [29]. The FFT routine was utilised in this project.

d. CodeVision

CodeVision IDE from HP Info Tech is a high performance ANSI C compiler, automatic program generator and in-system programmer for the Atmel AVR family of microcontrollers [30]. CodeVision was used to develop an application for the stimuli actuator node that has an Atmel Attiny88 microcontroller [71] as the host processor that interfaces with the CC2480 ZigBee processor [38].

e. Texas Instruments Z-Stack

Z-Stack [31] is TI's ZigBee compliant protocol stack for IEEE 802.15.4 products and platforms. Z-Stack is compliant with both the ZigBee 2007 and ZigBee 2006 specification, supporting both ZigBee and ZigBee PRO feature sets. The CC2480

device is preloaded with ZigBee 2006 stack. The host MCU communicates with CC2480 via SPI/UART as illustrated in Figure 2.3. Various commands are provided in the Simple API [70] developed by Texas Instruments.

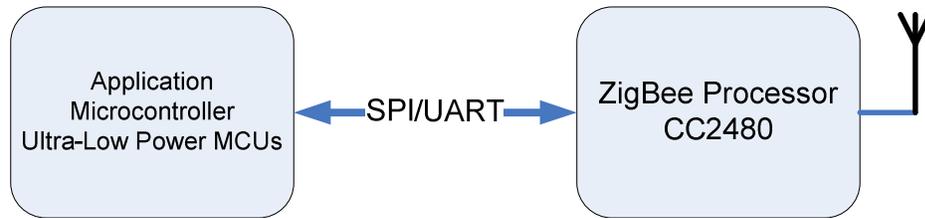


Figure 2.3: Host MCU and CC2480 interface.

The complete ZigBee interface is provided via the AF and ZDO interfaces [31]. Successful exchange of data and commands was established using the simple API. The simple API presents a simplified ZigBee API which is ideal for this project since the application does not use the full feature set available in ZigBee. It contains the necessary interface to commission a ZigBee network, perform bindings between devices and send/receive data. This is further explained in Appendix 3.

f. Altium Designer 6

Altium Designer is a unified electronics design solution to capture and design the physical PCB platform, create the programmable system hardware, develop the embedded software, and manage all the projects that make up an electronic product, all within a single unified design environment [32]. The wireless sensor/actuator node design which consists of schematic capture, PCB layout and component library included in Appendix 1 was realized using Altium Designer.

2.6 Hardware Development Platforms

2.6.1 DSK 6713

The C6713 DSK [33] is a low-cost standalone development platform that enables users to evaluate and develop applications for the TI C67xx DSP family. The pH analysis algorithm was tested on the C6713 DSK. The block diagram is given in Figure 2.4. The functional blocks utilized in the pH analysis were ‘6713 DSP’, ‘Flash’, Embedded JTAG’ and ‘USB’. The communication interface to PC was via USB and program download and debugging via embedded JTAG. The DSK also served as a hardware reference design for the pH sensor node. The schematics and application notes for

C6713 DSK are available to simplify hardware development and were used as a reference guideline for the sensor node PCB layout given in Figure 4.8.

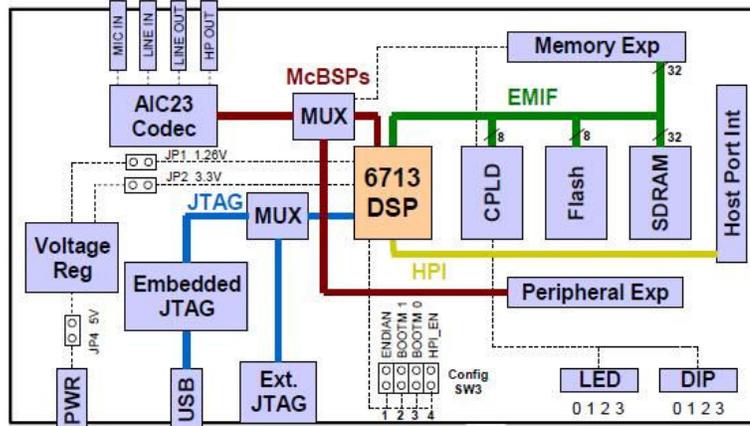


Figure 2.4: DSK 6713 Block diagram (Source: [33]).

2.6.2 eZ-RF2480

Z-Accel is a new concept by Texas Instruments to make the design of a new wireless product or the addition of wireless into existing products easy. Z-Accel devices function as ZigBee network processors, encapsulating all of the complexities of wireless into a black box. A host processor running the application simply communicates to a Z-Accel device via a simple serial interface and well defined API. CC2480 provides a fully compliant ZigBee solution and includes a unique pre-programmed IEEE address for each device. This includes full mesh capabilities, Coordinator/Router/End Device functionality, security, power savings, reliable communication, interoperability, and more. The eZ430-RF2480 Demo Kit [34] is a complete USB-based MSP430 wireless demonstration tool providing all the hardware and software to evaluate the CC2480 2.4GHz ZigBee network processor [38] and the MSP430F2274 microcontroller [35]. The eZ430-RF2480 Demo Kit utilises a free Kickstart version of the IAR Embedded IDE to write, download, and debug application program. The pH sensor network was simulated using four target boards similar to that shown in Figure 2.5.

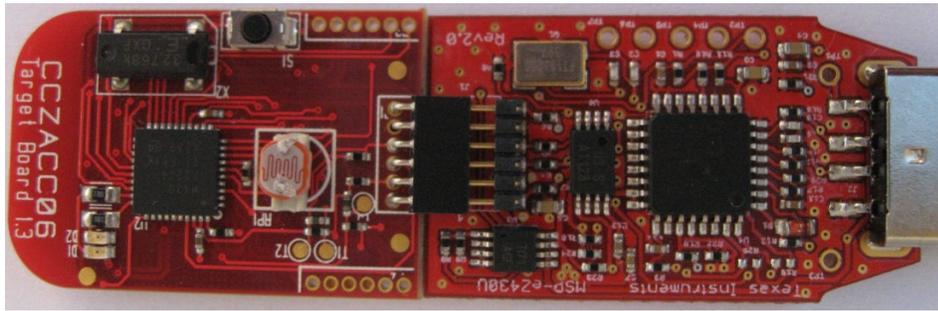


Figure 2.5: eZ-RF2480 target board.

2.6.3 CC2430 Demonstration Board

CC2430 Demonstration board shown in Figure 2.6 is based on TI's CC2430 [36] System-on-Chip solution for IEEE 802.15.4 and ZigBee applications. All I/O signals are available on header connectors on the CC2430DB to allow easy access for connecting external target boards, oscilloscope probes or logic analysers. In addition to this it was used as a 'listening' RF hardware node together with Packet Sniffer software to monitor over-the-air traffic during pH sensor network simulation.

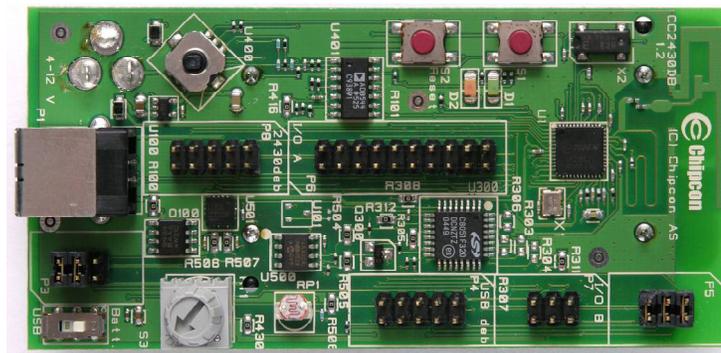


Figure 2.6: CC2430DB Development board.

Development kits are also needed for prototyping purposes. DSK 6713 kit and eZ-RF2480 kits were selected for this purpose. DSK 6713 kit was used to implement an embedded version of the pH analysis algorithm and eZ-RF2480 for establishing a ZigBee network. Packet Sniffer was used to capture over-the-air packets for ZigBee network debugging and analysis. Code limited C/C++ compilers, Code Composer Studio and IAR MSP430 were used to develop embedded applications for DSK6713 and eZ-2480 kits as part of the prototyping stage. These are included in the enclosed CD.

2.7 Summary

Various software and hardware tools were introduced in this chapter which were used in the simulation and development of pH sensor node and stimuli actuator node prototypes. Familiarities with these tools are essential to maximise benefit. The usage and significance of these tools are summarized here.

MATLAB: A function written in MATLAB to calculate the pH of the carcass from load cell data. This function is explained in Appendix 2.

OPNET: Simulation of ZigBee network to see the effects of varying receiver sensitivity transmitter power, distance on transmission of packets.

CCStudio/TI C67x DSPLIB/DSK6713: pH calculation program written in CCStudio and downloaded to DSK6713 to test it on an embedded DSP platform. The FFT functions were used from the TI C67x DSPLIB [29].

IAR Kickstart/Packet Sniffer/eZ-RF2480/CC2430DB: The application program was developed using IAR Kickstart and downloaded to the eZ-RF2480 target boards to simulate the pH sensor network. Packet Sniffer was used to decode the over-the-air packets captured by CC2430DB ‘listening’ RF board.

CodeVision: The application program for the stimuli actuator node prototype was developed using CodeVision. The target device was Atmel’s AVR Attiny88 [71].

Altium Designer 6: All electronic circuit design, PCB layout and component library was developed using Altium Designer 6.

3 Architecture Overview

The conceptual pH wireless sensor network architecture overview is presented in this chapter with emphasis on functional aspects of various components that establish the pH sensor network architecture. The chapter begins with introducing the pH sensor network architecture, the generic wireless nodes that form the sensor network, a description of pH sensor and actuator node architectures and concludes with the communication system that will enable wireless communication between the nodes in the network.

3.1 pH WSN Architecture

The conceptual pH WSN architecture is shown in Figure 3.1. This architecture is developed in order to replace the existing hardwired system and introduce wireless connectivity. It is a decentralised architecture where a group of sensor/actuator nodes manage the meat tenderising process through distributed processing. A portable and scalable system is anticipated through integration of WSN with distributed process control.

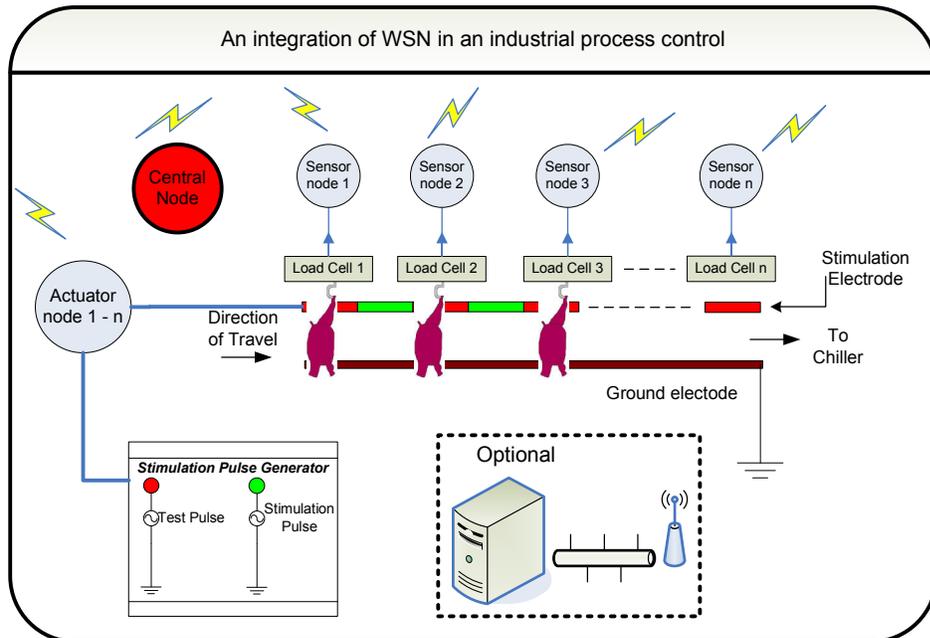


Figure 3.1: Conceptual pH WSN Architecture.

Each sensor node in Figure 3.1 will analyse the pH of the carcass detected on its load cell. A test command will be sent to the actuator node to apply test pulses to the carcass. The carcass response is acquired using the load cells and the pH of the carcass is then calculated. A stimulation activation command will be sent to the actuator node if the pH value is outside a predefined range. The sensor nodes will also send the carcass pH value to the central node for storage and utilisation in further stages of the process.

Sensing, computation and communication are implemented at node level in line with the decentralised architectural paradigm. Referring to Figure 3.1 above, Sensor node1 will calculate the carcass pH at stage 1, compare the pH value with a predefined limit and send the activation command to its child node, Actuator node 1. It will also send a message to the central node to record the pH value and carcass number. The central node is a gateway that interfaces with the meat plant network and provides virtual Ethernet access to the sensor nodes. If a bad carcass is detected by sensor node 1 whose pH value is way out of the pre-defined range it sends a message to subsequent sensor nodes so that no further action is taken. The central node tags the carcass as bad in the record.

Under normal circumstances, each node operates independently and performs its dedicated task as described above. Only during detection of abnormal conditions of the carcass, the nodes communicate with each other to resolve the issue or take necessary actions such as trigger an alarm etc. The nodes work in pairs, every sensor node has an associated actuator node. The actuator node controls the application of test/stimulation signals based on the control command from its parent node. Test/stimulation pulses are applied via a cluster of relays

3.2 Generic Nodes

The common task of a wireless sensor network node is wireless communication with other nodes that are part of the same network. The components of such a generic node are a central controller, communication controller and peripheral interface. Figure 3.2 illustrates the key components of a generic node. This is a standard configuration for most embedded applications. It also provides the base for development of the architecture for pH sensor node and stimuli actuator node.

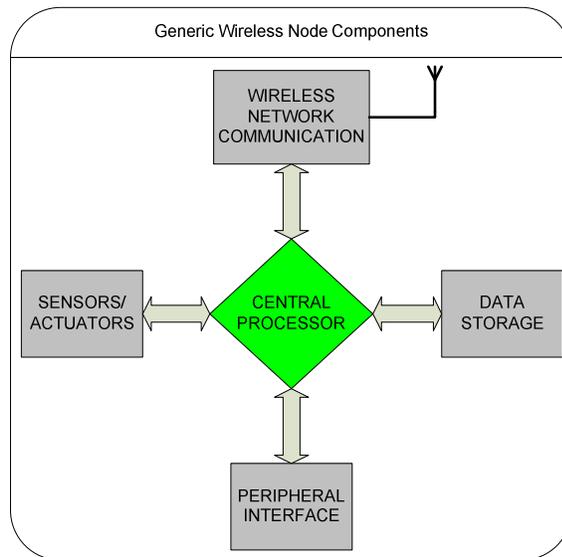


Figure 3.2: Key components of a Generic Wireless Node.

3.2.1 pH Sensor Node

The sensor node plays the key role in the system. It has the capability to independently calculate the pH value at its assigned stage and decide if further stimulation is required. The pH computation requires Fourier analysis of the carcass response spectrum therefore this node requires digital signal processing capability in addition to detecting and joining a desired wireless network. As indicated in Figure 3.2, the architecture for the pH sensor node have similar generic configuration which are sensing, computation and communication on board. The central processor is a digital signal controller capable of executing signal processing algorithms. It interfaces with four separate modules via SPI/I²C. These modules handle data storage, sensor conditioning, external peripherals such as ADC and wireless communication. This is discussed in more detail in Chapter 4.

3.2.2 Stimuli Actuator Node

The role of the stimuli actuator node is to activate the stimulation segments (red and green bars in Figure 3.1) based on the command received from its parent node. The activation of the segments is achieved through a bank of relays which switch high voltage pulses to each segment of the stimulation electrode. In case of communication failure, a timeout feature ensures the stimulation segments are never activated for more than three seconds. There are no intensive signal processing requirements for this node. The architecture of the actuator node is very similar to the sensor node except that

instead of separate modules it is a System-on-Chip (SoC) solution. Since there are no requirements for signal processing, the SoC architecture provides a compact cost effective solution. More details on the architecture are provided in Chapter 4.

3.3 Communication system

The communication system layout is shown in Figure 3.3 based on Texas Instrument's devices and ZigBee protocol. The CC2480 [38] is equipped with IEEE 802.15.4 compliant radio and ZigBee compliant Z-Stack [31]. An external microcontroller is required to host the application program that configure CC2480 and control communication.

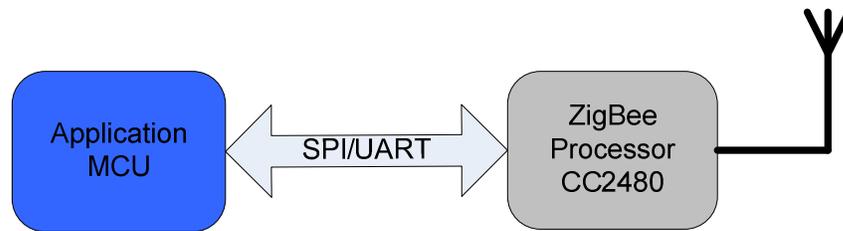


Figure 3.3: Communication System.

3.3.1 ZigBee Protocol

a. Application Profile

An application profile is a common language for exchanging application data [48]. It consists of:

- A set of application devices within that profile
- A set of commands exchanged by the devices at the application layer
- Format of the data that is exchanged between devices in each command.

The application profile for the pH sensor network is shown in Figure 3.4. A messaging strategy is developed to achieve higher throughput by restricting command data type to single characters. The commands are decoded by the application program at the receiver end.

Each sensor node in the network will analyse the carcass pH first and activate stimulation accordingly. This approach is self-correcting any issues that may have resulted at the previous node. For example if no stimulation took place due to communication failure between sensor/actuator node at stage 1 than stage 2 sensor node will analyse pH and deliver the required stimulation.

Peak traffic occurs when all stages are occupied and test/stimulation is in progress. The communication is taking place between sensor/actuator node pairs and only commands are issued by the sensor nodes and acknowledgement from actuator nodes. Acknowledgement is automatically handled by the network layer. The central node has the capacity to record carcass number and carcass pH, hours of operation and total output of carcass.

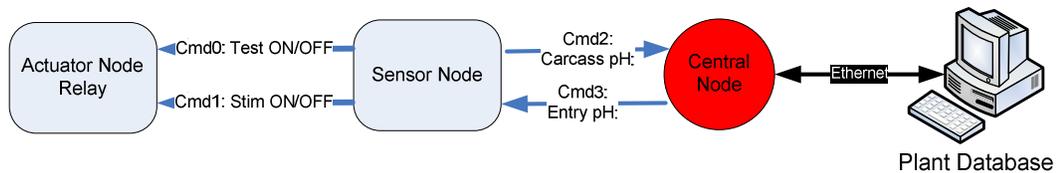


Figure 3.4: pH Wireless Sensor Network Application Profile.

b. ZigBee Endpoint registration

i. Endpoint

Within each node are endpoints which define each application running in a ZigBee node, therefore a single ZigBee node can run multiple applications. The pH sensor/stimuli actuator node have a single application running so both have an Endpoint = 1. These endpoints have to be registered with the ZigBee stack by the application program. The endpoint registration parameters for the nodes are given below. There are actual parameters taken from the application program developed for the nodes in this project. The format is for use with Z-Stack and explained in detail in Chapter 4.

1. pH Sensor Node

MASTER_ENDPOINT_ID	1
ZBee_PROFILE_ID	0x0F10 (Texas Instrument Development Profile ID)
MASTER_DEVICE_ID	1
MASTER_DEVICE_VERSION	1
RESERVED	0
MASTER_CLUSTER_IN_CNT	1
MASTER_CLUSTER_OUT_CNT	1
MRLY_ON_ID	1

2. Stimuli Actuator Node

ZBRLY_ENDPOINT_ID	1
ZBee_PROFILE_ID	0x0F10 (Texas Instrument Development Profile ID)
ZBRLY_DEVICE_ID	2
ZBRLY_DEVICE_VERSION	1
RESERVED	0
ZBRLY_CLUSTER_IN_CNT	1
MSLA_ID	2
ZBRLY_CLUSTER_OUT_CNT	1

ii. Profile ID

Every data request in ZigBee is sent/received on an application profile. Application profile IDs are 16-bit numbers and range from 0x0000-0x7fff for public profiles and 0xbf00-0xffff for manufacturer specific (private) profiles [19, 48]. This is allocated by the ZigBee Alliance. The eZ-RF2480 has a profile ID for development purpose which is 0x0F10.

iii. Device ID and Device version

These are given arbitrary IDs (1, 2 etc.) since this is a private application and will not operate in a public domain.

iv. Cluster In and Cluster Out

These are set of clusters and cross-cluster commands used in either public or private profiles which have already been defined by the ZigBee Alliance as ZigBee Cluster Library [66]. The clusters developed for the sensor/actuator nodes specific to this project is listed in Table 4.1, Chapter 4. Commands that are consumed by the node are summed as Cluster In and those that are produced by the node as Cluster Out. For Example, the Stimuli Actuator node consumes ON/OFF command so Cluster In = 2 and produces none so Cluster Out = 0.

c. ZigBee Application Framework

The key to communicating between devices on a ZigBee network is the agreement on a profile. There are two separate classes: manufacture-specific and public. Every profile is allocated by the ZigBee Alliance upon filing a request.

The profile identifier **0x0F10** is provided by Texas Instruments for developmental purpose only and is used for this project. It provides a pool of device descriptions described by 16-bit value and a pool of cluster identifiers described by a 16-bit value. This means there are 65,536 device descriptions and cluster identifiers within each profile. A single ZigBee device may contain support for many profiles. This capability is defined by using a hierarchy of addressing within the device as follows:

- Device: the entire device is supported by a single radio with a unique IEEE and NWK address.
- Endpoints: this is an 8-bit field that describes different applications that are supported by a single radio.

Endpoint 0x00 is used to address the device profile; endpoint 0xFF is used to address all active endpoints. A single physical ZigBee radio can support up to 240 applications on endpoints 0x01-0xF0.

3.4 Summary

This chapter presented the general overview of the architecture for the pH Wireless Sensor Network for the meat tenderizing system. It also introduced essential components of the communication system to support the wireless architecture. The generic architecture of a wireless sensor node is illustrated in Figure 3.3. The pH sensor network illustrated in Figure 3.1 consists of pH sensor nodes and stimuli actuator nodes. These nodes were described to show their role in the network. The pH wireless network is a ZigBee network and the ZigBee protocol was briefly introduced here.

4 pH sensor node detail design

This chapter details architecture design of various components outlined in Chapter 3. Functional block diagrams of sensor/actuator nodes are discussed including hardware solutions for the node architecture. The wireless communication system is based on ZigBee protocol. Relevant aspects of the protocol are discussed that leads to the development of an application program that is executed on the eZ-RF2480 platform. This chapter concludes with the development of the pH sensor and stimuli actuator node prototypes.

4.1 Wireless Node Design

The design of the sensor node is based upon a decentralized architectural paradigm [41, 69]. This type of architecture produces sensor nodes with on-board processing and communication facilities. The system that originates from this consists of a network of autonomous nodes, either fully connected or connected to only local nodes. Any node in the network can be assigned the role of a coordinator to form the network and sink data for real time visualization and tracking. The decentralized architecture offers a number of advantages, including modularity, scalability and fault tolerance [67, 69].

4.2 pH Sensor Node

In Chapter 3, Figure 3.1, each node is responsible for a dedicated task while the central node has the added task of handling data management and remote access via Ethernet. This is indicated as an optional block in which the central node could be connected to a central database via Ethernet or wireless gateway. The functional modules of the sensor node are shown in Figure 4.1. The operations that are executed by these modules are power conversion, data acquisition, spectrum analysis, handling communication protocols and wireless communication.

The controller takes the role of system management. It handles user interrupts, peripheral interrupts and provide data interface. It acts as a bridge between the DSP and Radio modules by taking pH values from the DSP module and packaging it using the communication protocol module to send to the Radio module for wireless transmission. The process reverses when message/command is received wirelessly. The system management software runs on the controller utilising the controller’s resources to manage the functionality of the sensor node. The sensor node application framework is illustrated in Figure 4.5. Software for the sensor node functional modules is developed using this framework.

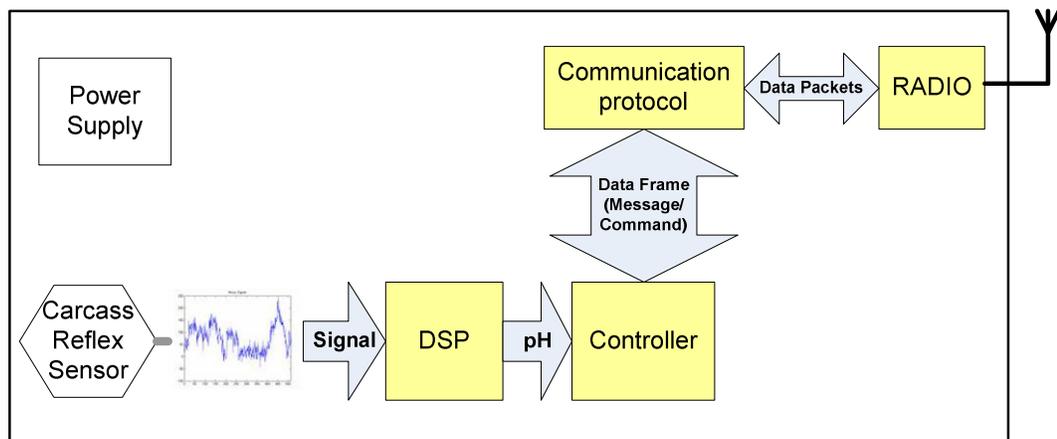


Figure 4.1: pH Sensor Node functional modules.

4.3 Stimuli Actuator Node

The functional modules of the actuator node are shown in Figure 4.2. The communication protocol and the Radio are similar to the sensor node except packets contain commands rather than pH values. These commands are given in Table 4.1.

a. Controller

The controller performs wireless communication utilising the communication protocol and Radio. There is no computational activity at the actuator node therefore the controller option is an 8-bit processor. Its primary role is to decode the commands and activate the respective relays. The other important function is to handle interrupts received from the failsafe monitor. These tasks are handled by the application program running on the controller.

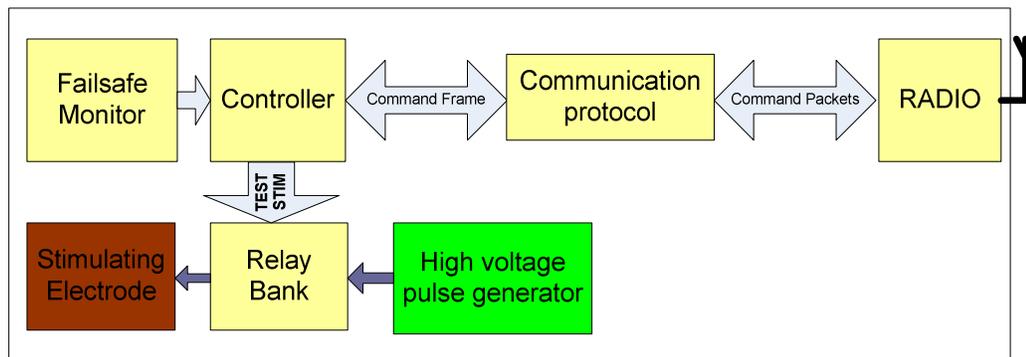


Figure 4.2: Stimuli Actuator Node functional modules.

b. Failsafe Monitor

The maximum allowable duration for stimulation is 3 seconds. This is monitored by using the failsafe monitor module. It is a timer module which is SET/RESET by the controller under normal operation. An “ON” command from the controller to the relay bank sets the failsafe timer and an “OFF” command resets the failsafe timer. In the event of relay or communication failure causing the stimulation state to exceed 3 seconds the failsafe monitor will activate the fail safe relay to disconnect the pulse generator. Relay status is sensed by the controller and if it is a relay failure then an alarm indicator is activated. At this stage it is a simple precautionary measure to prevent carcass damage from over stimulation.

c. Relay Bank

The relay bank consists of relays used for connecting stimulating pulses and test pulses from the pulse generator to the electrodes. The controller module decodes the command and activates the relevant relay. The control signals for the relay bank are STIM ON/OFF and TEST ON/OFF.

d. High Voltage Pulse Generator

This is a third party system developed to generate stimulating pulses and test pulses. Currently it is preconfigured for lamb/beef carcass and the parameters cannot be changed at run time. Future developments for feedback system will enable real time configuration of the pulse generator parameters.

4.4 pH Signal Analysis

The sensing element for carcass reflex to electrical stimulation is a load cell. A load cell enables non-intrusive data acquisition and carcass detection while the carcass is moving on the conveyor. Online data acquisition eliminates plant process flow stoppages which may happen in case of using intrusive sensors such as pH probes or surface electrodes. As a detection element, a load ON/OFF trigger is generated using simple voltage threshold detector circuitry. For example, the output of the detector is logic HIGH when the carcass is on the load cell and logic LOW when it moves out of the load cell. Currently load cells are the most suitable contactless sensing element for this application due to its capability to act as a sensor and detector in this project.

The pH analysis algorithm is proprietary and patented [17]. In this project the algorithm is ported to an embedded DSP platform. The algorithm is outlined below.

1. Take FFT of carcass reflex spectrum.
2. Filter frequencies of interest (10 – 200Hz).
3. Calculate magnitudes of the filtered frequencies.
4. Sum the magnitudes and add carcass pH constant.

Carcass reflex spectrum is comprised of many frequencies. FFT is needed to identify these frequency components and its magnitudes. Only a certain range of frequencies is of interest which represents cellular activity in the muscle tissue during stimulation. The pH is analysed from these frequencies. Converting to the frequency domain, the discrete Fourier transform of the response spectrum is found by taking the fast Fourier transform (FFT). Radix-2 FFT algorithm is widely used for frequency domain processing and spectrum analysis. Radix-2 FFT first computes the DFTs of the even-indexed inputs ($2n$) and of the odd-indexed inputs ($2n+1$), and then combines those two results to produce the DFT of the whole sequence. The Radix-2 algorithm rearranges the DFT of the function $x[n]$ into two parts, a sum over the even-numbered indices $k = 2n$ and a sum over the odd-numbered indices $k = 2n + 1$. It is a computationally efficient discrete Fourier transform [42, 58] defined as:

$$X(2k) = \sum_{n=0}^{(N/2)-1} x[n] + x\left(n + \frac{N}{2}\right) W_{N/2}^{kn}, \quad k = 0, \dots, \frac{N}{2} - 1 \quad (1)$$

$$X(2k + 1) = \sum_{n=0}^{(N/2)-1} \left\{ \left[x[n] - x \left(n + \frac{N}{2} \right) \right] W_N^n \right\} W_{N/2}^{kn}, \quad k = 0, \dots, \frac{N}{2} - 1 \quad (2)$$

$$\text{Where, } W_N^{kn} = e^{-\frac{j2\pi nk}{N}}$$

However for embedding on the DSP hardware platform it is computed using FFT routine included in Texas Instrument DSP library [29]. It is a function implemented in embedded C for the TMS320C6000 series DSP processor. The implementation of the FFT function on the DSK 6713 platform is as follows:

1. Assign a pointer to a floating point array of length 2*n which holds n complex samples.
2. Assign a pointer to a floating point array of length n which holds the n/2 twiddle factors. Radix 2 twiddle generation is performed using tw_gen2fft function found in the FFT support library [68]. After creating this array it is bit-reversed using bit_rev function also in the FFT support library.
3. n is the length of the FFT in complex samples, n must be a power of 2 and greater than or equal to 32. This is the minimum length allowed in the FFT routine in the DSP library [68].

The FIR filter is not implemented. Instead the filter coefficients are provided as proprietary coefficients for beef/lamb carcass. It may be included in future work to have on-board digital filtering. The breakdown of the pH analysis algorithm shows it is a computational intensive process for an embedded processor. Therefore the sensor node must have signal processing capability which is proposed in the architecture to be implemented at node level. The DSP and Controller modules complement each other and would normally be found as an integrated module or device.

4.5 Wireless Radio Communication

The Radio is the heart of the wireless communication system. The obvious choice for this application is unlicensed ISM frequency band. The ISM band is globally used for Industrial, Scientific and Medical communication purposes [44]. IEEE 802.15.4 based radios operate in the ISM band and is largely supported by a number of semiconductor manufacturers.

IEEE 802.15.4 offers the lower network layers which focuses on low cost, low speed ubiquitous communication between devices in contrast with other approaches such as Wi-Fi and Bluetooth [45, 56]. The communication radius is within 10m with a maximum transfer rate of 250kbit/sec. The main feature of IEEE 802.15.4 is the importance of achieving extremely low manufacturing and operation costs,

technological simplicity, guaranteed access time slot using CSMA/CA and integrated support for secure communications [37, 57].

The following strategy for wireless communication incorporating a network of sensor nodes is taken; Network ID, Device ID, Message ID, Command ID, Data Payload. This is shown in Figure 4.3. The required network topology for this application is tree topology since the sensor/actuator node pairs communicate with each other only and all the sensor nodes communicate with a central node for data storage purpose. It is a fairly straight forward master/slave type scenario, so the tree topology is ideal compared to either star or mesh. Star is not suitable since the sensor/actuator node will communicate with each other and to the central node. Mesh network would be more appropriate when all the nodes in the network are required to communicate with each other.

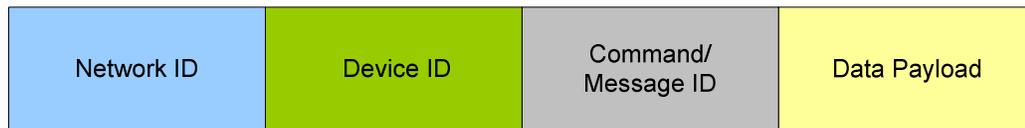


Figure 4.3: Generic Communication Packet design.

Figure 4.4 below shows the actual communication packet sent by the sensor node over-the-air to the actuator node to turn ‘ON’ the Test/Stimulation relay. ‘1’ is ON and ‘0’ is OFF.



Figure 4.4: Command packet.

The packet design strategy highlights the following:

Network ID or PAN (Personal Area Network) ID to be generated so that all the nodes with this ID can join the network initiated by the network coordinator. This would prevent other devices in any other existing wireless network with the same protocol interfering with the pH sensor network and vice versa. Each device has its own ID for identification purposes in the network and message routing. The commands/messages will have respective IDs as endpoints on the sensor/actuator nodes to resemble a

predefine task. Finally a data payload for transfer of data such as pH values, carcass numbers and commands such as ON/OFF.

4.5.1 Communication protocol

The ZigBee communication protocol is an approved standard that is defined to facilitate wireless communication between nodes in a wireless mesh network. ZigBee is one of the many wireless communication protocols and standards that automatically construct an Ad-hoc network of nodes. ZigBee protocol is proposed for this application based on the following key aspects:

- Standards-based wireless platform.
- Open global standard that also addresses industrial automation.
- Secure and reliable.
- Low cost.

The most logical comparison of ZigBee with any other standard is the difference in approach to their respective application area and target market [46, 47]. In the case of ZigBee, its application area is low power sensor networks for remote monitoring and control. Interoperability is not considered for this project hence most of the complexities of developing such a device are reduced. Only peer-to-peer communication is maintained at this stage.

4.6 pH sensor network application framework

The proposed application program development framework shown in Figure 4.5 for the sensor node presents the overall software architecture. It highlights the flow between the modules for the design and development of the system software. Each of these modules has underlying components that are described below. Corresponding software routines are included in the enclosed CD. Figure 4.5 is discussed below.

a. DSP

The pH analysis algorithm is the single component running inside the DSP module and the algorithm is as explained in section 4.4.

b. Communication Controller

The Communication controller comprises of two modules, Communication protocol module and Radio module. The communication protocol is ZigBee and the ZigBee compliant software stack is Z-Stack. Z-Stack handles data frame formatting for transmit/receive via the Radio which in turn handles packet transmission over-the-air. Z-Stack has 10 simple API's to interface with the

system controller module. The Radio is an IEEE 802.15.4 compliant transceiver. This is a physical semiconductor device CC2420 integrated with CC2480.

c. System Controller

The system controller module executes after system power-up and its functionality replicates a pseudo RTOS. It provides the interface for any system resource requirement by the DSP and Communication controller modules through Tasks and Function Calls, which are embedded in the Hardware Abstraction Layer (HAL).

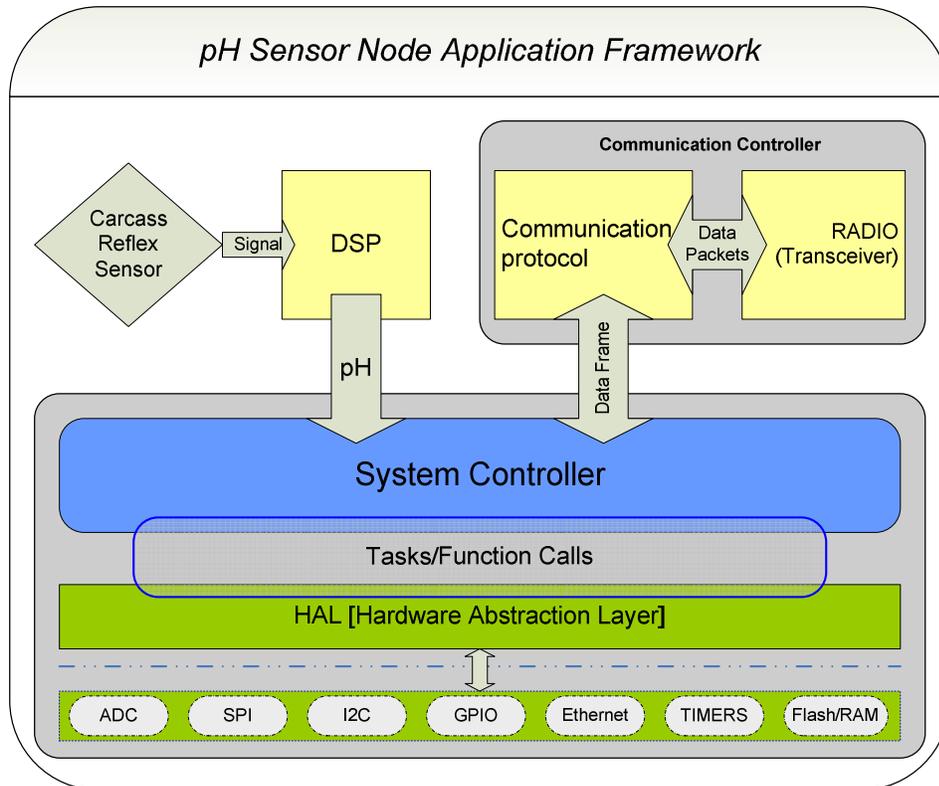


Figure 4.5: Sensor Node Application Framework.

4.6.1 Tasks & Function Calls

The Tasks and Function Calls are subdivided into following categories:

a. Initialization Function Calls

Initialization function calls are used to initialize a service and /or to setup optional parameters for the hardware platform. Initialization functions are often called at the beginning stage when the device power up.

b. Service Access Function Calls

Service access function calls can directly access peripheral registers to set a flag, read a value (for example read ADC registers after a conversion) or control a device (for example turn ‘ON’ an LED). This is made possible by HAL which provides the interface between peripherals and Tasks/Function calls.

c. **Call-back Function Calls**

Call-back function calls are implemented by the application program and are used to pass events that are generated by the hardware (interrupts, counters, timers) or by polling mechanism (UART poll, Timer poll) to upper layer. There are number of call back functions written for the application program to support wireless communication using Z-Stack and simple API during simulation of the pH sensor network on eZ-RF2480 platform.

d. **Services**

HAL drivers provide Timer, GPIO, LEDs, Switch, UART, and ADC service for upper layers. It provides an abstraction layer between the system controller and peripheral devices. The upper layers request use of the hardware resources and peripherals by calling routines in the HAL layer, for example to read a temperature sensor or activate an alarm.

Figure 4.6 shows the location of the sensor node application program in relation to the ZigBee protocol architecture. The pH data and Relay control application executes through simple API within the Z-Stack. Endpoints 1 and 2 are supported by a single radio. Sensor node application is mounted on endpoint 1 and actuator node is mounted on endpoint 2. The ZigBee processor CC2480 is an integrated device with CC2420 transceiver and Z-Stack preloaded. CC2420 is an IEEE 802.15.4 compliant transceiver and Z-Stack is a ZigBee Alliance certified platform. Both are integrated into a single device, the CC2480.

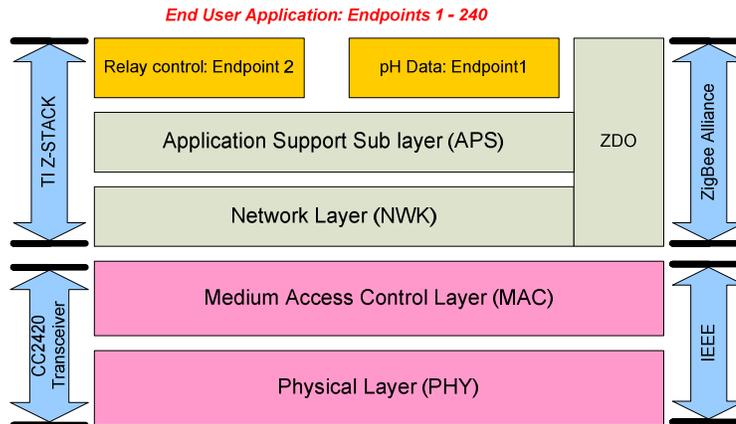


Figure 4.6: Sensor Node stack: Z-STACK & CC2420 Transceiver.

The ZigBee protocol was discussed to an extent to prepare for the sensor node application development. The full detail description of the ZigBee protocol is beyond the scope of this thesis.

4.6.2 Process of Developing a Private Application Profile: pH Sensor Network.

1. Identify all devices in the application.
 - Assign a unique device ID to each device [unique 16 bit identifier].
2. Identify the commands that need to be exchanged between these devices and assign a unique 16-bit command ID to each of them.
3. For each command identify the devices that “*produce*” and “*consume*” it.
4. Create a simple descriptor structure for each device. This includes assigning a device identifier and device version to each of the devices, specify the list of *output* and *input* commands for that device and specify a profile ID. This is a 16-bit value that identifies uniquely the application profile. These are assigned by the ZigBee Alliance.
5. For each command define the format of the message being exchanged and its interpretation.
 - Format: “An 8 bit value”.
 - Interpretation: For example “0” is OFF and “1” is ON.
6. Write the device application program for each device.
 - Device with output commands should be able to generate the packet either periodically or when an external event occurs.
 - Device with input commands should handle the reception of these packets and parse the payload.

Table 4.1 below is a list of commands and message IDs allocated to each device in the ZigBee based pH sensor network. Commands are called clusters in the ZigBee specification and can be private or public. The clusters in Table 4.1 are private developed for this project. Each device (node) is given an ID to identify it in the network. Commands are also given IDs so that the application program (Endpoint) running on the device (node) can identify the command and take necessary action.

pH sensor network Profile ID: 0x0F10					
Devices	Commands	Command ID [16-Bit]	Produce (P)	Consume (C)	Format [8-Bit]
Sensor Node					
Master – Coordinator [MC001 – 0xC001]	MRLY_ON	1002	P		1
	MRLY_OFF	1004	P		0
	MRLY_ACK_ON	1006	C		
	MRLY_ACK_OFF	1008	C		
	ENTRY_pH	1010	P		
	SS_OFF	1012	P		0
	SS_ON	1014	P		1
	SS_ACK_ON	7001		C	
	SS_ACK_OFF	7003		C	
	EXIT_pH	3002		C	
Slave1 – Router [SR1001 – 0xC050]	S1_pH	2002	P		
	S1RLY_ON	2004	P		1
	S1RLY_OFF	2006	P		0
	S1RLY_ACK_ON	5001		C	
	S1RLY_ACK_OFF	5003		C	
	ENTRY_pH	1010		C	
Slave2 – Router [SR2001 – 0xC051]	EXIT_pH	3002	P		
	S2RLY_ON	3004	P		1
	S2RLY_OFF	3006	P		0
	S2RLY_ACK_ON	6001		C	
	S2RLY_ACK_OFF	6003		C	
	S1_pH	2002		C	
Actuator Node					
ZMRelay [ZM001 – 0xED01]	MRLY_ACK_ON	4001	P		1
	MRLY_ACK_OFF	4003			
	MRLY_ON	1002		C	1
	MRLY_OFF	1004		C	0
ZS1Relay [ZS1001 – 0xED02]	S1RLY_ON	2004		C	1
	S1RLY_OFF	2006		C	0
	S1RLY_ACK_ON	5001	P		2
	S1RLY_ACK_OFF	5003	P		
ZS2Relay [ZS2001 – 0xED03]	S2RLY_ON	3004		C	1
	S2RLY_OFF	3006		C	0
	S2RLY_ACK_ON	6001	P		2
	S2RLY_ACK_OFF	6003	P		
ZSSRelay [ZSS001 – 0xED04]	SS_OFF	1012		C	0
	SS_ON	1014		C	1
	SS_ACK_ON	7001	P		2
	SS_ACK_OFF	7003	P		

Table 4.1: Network message/command design.

For example to turn ‘ON’ relay ‘ZMRelay’ the command ID ‘1002’ and Command ‘1’ is sent by the ‘Master’ node as listed in Table 4.1. Only the node that has included command ID ‘1002’ in its list of clusters will respond which in this case is ‘ZMRelay’. All these commands are registered within the application program as explained in section 3.2.1.2. ZigBee Alliance has already released various profiles and cluster libraries for interoperability purposes which can also be used.

4.7 pH Sensor Node and Stimuli Actuator Node Prototype

The formulations of the sensor node architecture and application framework lead to the realization of the pH sensor node. The block diagram is shown in Figure 4.7 which includes actual semiconductor devices chosen for the node. The sensor node prototype

PCB is shown in Figure 4.8. The schematic and PCB design files are included in the attached CD [Appendix 1].

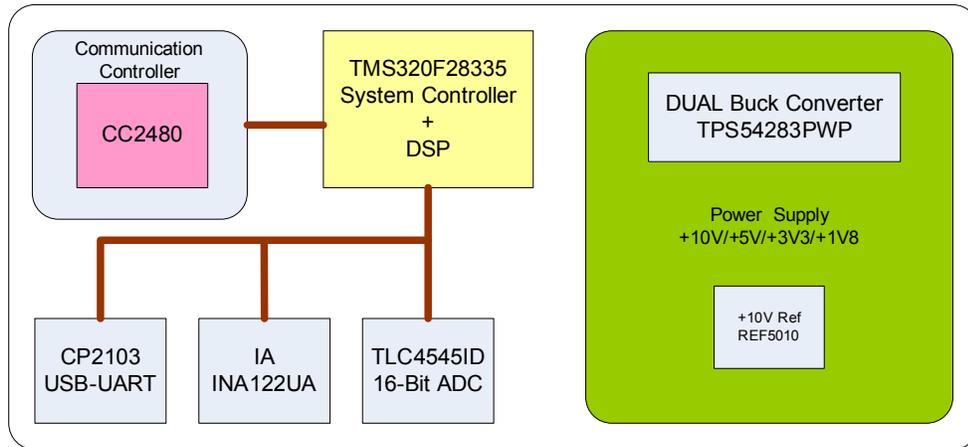


Figure 4.7: pH Sensor node prototype block diagram.

In addition to the application program interface there is a physical link between the communication and system controller modules as shown in Figure 4.7. These are control signal lines polled by the system controller when polling is used. If an interrupt driven system is preferred then the control signal lines can cause an interrupt which the system controller has to service to initiate the wireless communication process. The communication interface between CC2480 and TMS320F28335 [49] is via SPI. Other peripheral devices included in the design are CP2103 [50] for uploading data to PC, INA122UA [51] instrument amplifier for sensor signal amplification and TLC4545ID ADC [52] for data acquisition. REF5010 provides a stable +10Vdc excitation voltage to the load cell.

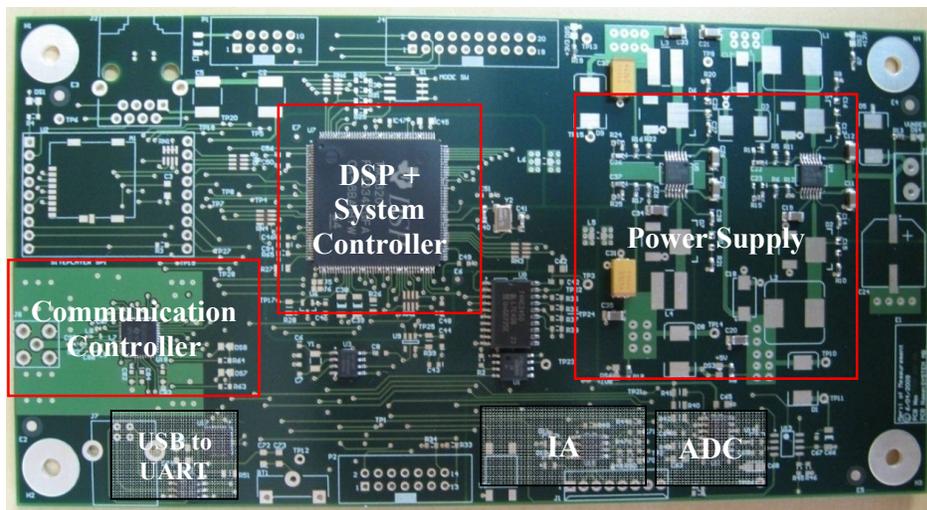


Figure 4.8: pH sensor node prototype PCB.

Figure 4.9 shows the block diagram of the actuator node and Figure 4.10 shows the PCB. The core concepts for this project, pH analysis algorithm and wireless sensor network were developed and tested on development platforms available commercially. However, to fulfil the requirement of the project sponsors the prototypes shown in Figure 4.8 and Figure 4.10 were developed and delivered.

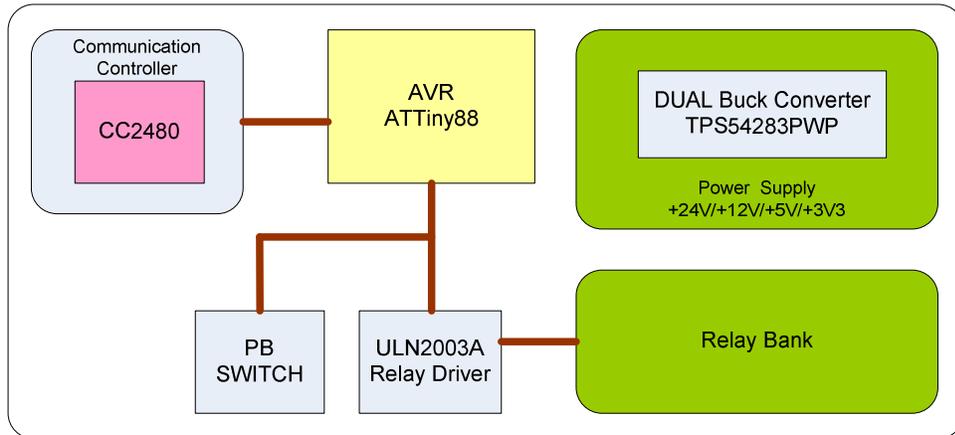


Figure 4.9: Stimuli Actuator node prototype block diagram.

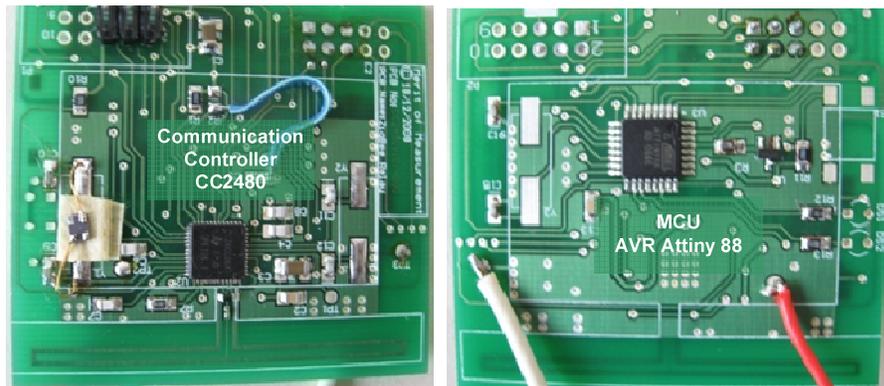


Figure 4.10: Stimuli Actuator Node prototype PCB.

4.8 Summary

The wireless node design is based on decentralized architectural paradigm. This chapter presented the flow of the design process that leads to the development of the pH sensor node, stimuli actuator node architecture, application program development and wireless node prototypes. Modelling and simulation were carried out in parallel with the wireless node development process and are explained in Chapter 5.

5 pH Sensor Network Modelling and Implementation

The experimental/simulation setup and associated equipment are discussed and illustrated in this chapter. Through these simulations it was possible to evaluate evidence based wireless sensor network performance and data that can create opportunities and potential for the development of prototypes. It is necessary to verify the proposed architectures through simulation and prototyping to prove the concept. A wide range of development tools are available from various vendors discussed in Chapter 2. Some of these tools have become the preferred standard in industry and academia therefore the simulation of pH analysis algorithm and pH sensor network were carried out using these tools. The simulation results complement system design flow and realization.

5.1 Carcass reflex signal processing & pH analysis

pH sensor and stimuli actuator node prototype was developed with reference to the SmartStim system. SmartStim is a non-intrusive system which analyses carcass pH by applying 300V pulsed stimulus and calculating pH from the spectrum of the carcass reflex to this stimuli [17]. The reflex is sensed using load cells. Thus the pH analysis and electrical stimulation takes place while the carcass is moving on the conveyor. The pH analysis algorithm is shown in Figure 5.1. However from the digital signal processing perspective, the process decomposes the spectrum of the carcass reflex to stimuli using FFT and removes unwanted frequencies by applying a digital filter. The imported coefficients and pH constant is an intellectual property held by Carne Technologies Ltd.

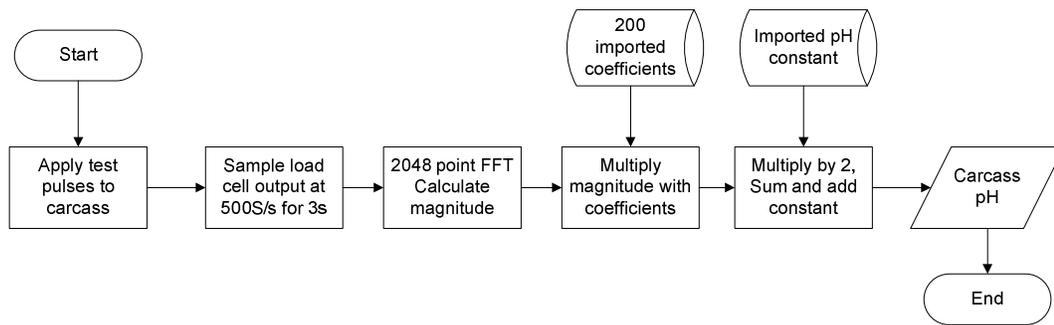


Figure 5.1: pH analysis algorithm.

SmartStim system has multiple stages of stimulation to achieve a target pH. A major commercial shortcoming of existing electrical stimulation technology is that each carcass responds differently to the stimulation. Insufficient stimulation delays tenderization and produces tough meat, while exaggerated stimulation causes deterioration of other meat quality attributes. Measuring pH at each stage of stimulation enables more accurate and controlled treatment on a carcass-by-carcass basis which is the central focus of this research by developing wireless pH sensor network.

MATLAB is selected to simulate pH analysis algorithm. It is a well-established industry standard computation and simulation software package and available at the host institute. The simulation allowed the development of program for pH analysis on an embedded DSP platform DSK6713. Estimation of real time performance was also considered.

Figure 5.2 below shows the flow of the meat tenderizing process. The sensing element is a Tedeo Huntleigh load cell [54]. Load cells effectively measure the muscle contraction force when electric current passes through the muscles. A test pulse is applied when the carcass is detected on the load cell. Test pulses are applied to calculate pH while stimulation pulse causes tenderization.

The Stimulation/Test parameters are as follows:

- Test/Stimulation pulses: 300V.
- Test Duration: 3 seconds (9 pulses with increasing interval up to 150ms).
- Stimulation Duration: variable (15 and 20 pulses/sec).

The data acquisition parameters are as follows:

- F_s (sampling frequency): 500Hz.
- Duration: 3 seconds.
- Data conversion: 16-Bit National Instrument data acquisition PCI card.

The load cell output is sampled at 500Hz using National Instrument's 16-Bit data acquisition card [72]. The acquired data was processed on MATLAB using the "fft2pH" function explained in Appendix 2.

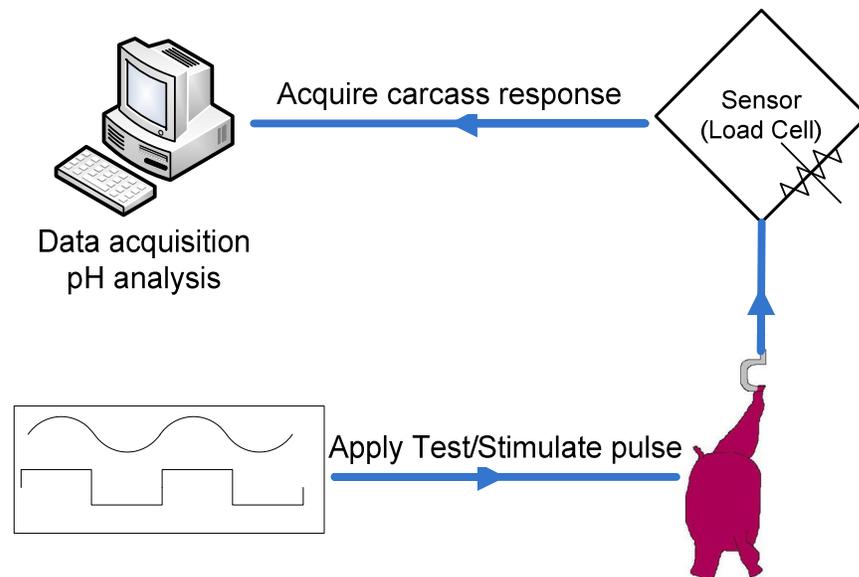


Figure 5.2: SmartStim meat tenderization process.

The actual muscle responses are shown in Figures 5.3 and 5.4 below and reflect two successive conditions of the beef carcass. The Y-axis in Figure 5.3 is the load cell output in volts and the X-axis is the data points sampled at 500Hz.

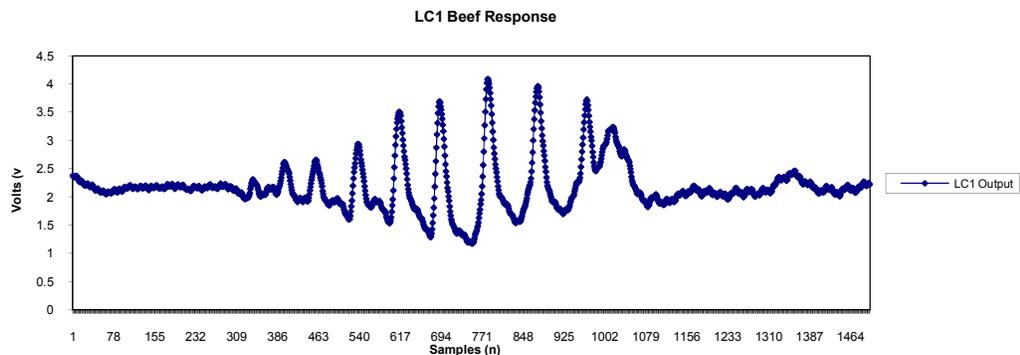


Figure 5.3: Beef carcass response to test pulses at Load cell 1.

The waveform in Figure 5.3 is the muscle response to test pulses. At this stage the carcass is untreated (no ES applied). The pH value was calculated from this data and electrical stimulation was applied accordingly. The decrease in amplitude of the carcass response in Figure 5.4 is the result of decline in pH due to electrical stimulation. No further stimulation is required once the target pH of 5.6-5.8 is reached.

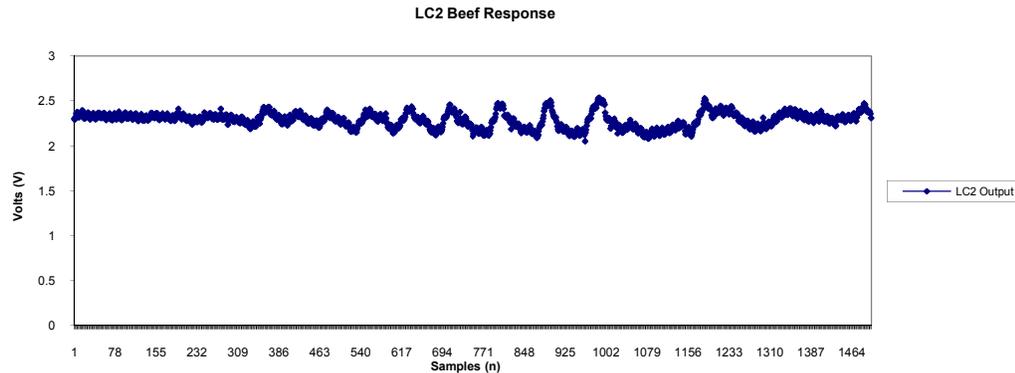


Figure 5.4: Beef carcass response to test pulses at Load cell 2.

The mathematical representation of the pH analysis algorithm developed in MATLAB is a function “fft2pH.m” explained in Appendix 2. The results obtained from this “fft2pH.m” function was verified with data from the meat plant held by Carne Technologies Ltd. Data produced by MATLAB was used as reference values for the DSP platform. The theoretical base is same therefore similar result is obtained from MATLAB and the DSP platform, DSK6713.

The pH analysis algorithm was written in ‘C’ and downloaded to the DSK 6713 kit. The setup for this task is shown in Figure 5.5 below. The response data shown in Figures 5.3, 5.4 and 5.6 was fed into the MATLAB function “fft2pH” to obtained pH values. The response data was then loaded as an input data file for processing on the DSK6713 embedded DSP development platform. Consistent results were obtained for beef and lamb carcasses. The algorithm can be successfully implemented on embedded DSP platform since it is verified here with reference values from MATLAB which were initially verified by data from the meat plant.

The core computational activity required for the pH analysis algorithm is FFT. Therefore DSP capability system processor is used in the sensor node prototype. The DSP based processors normally have onboard multiplier units to handle floating point arithmetic and large number of peripherals to connect to the outside world. The results are shown in Table 5.1

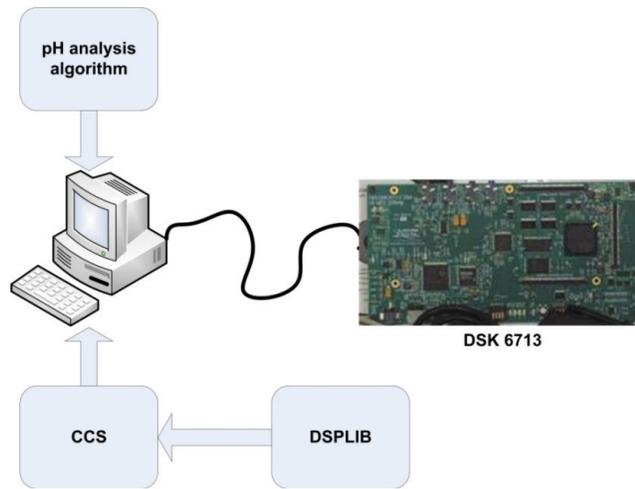


Figure 5.5: pH analysis on DSP platform.

Further analysis was performed on load cell data for lamb carcass as shown in Figure 5.6. The waveforms in Figure 5.6 clearly indicate the gradual decline in muscle response to test pulses. After each load cell there is stimulation electrodes to treat the carcass. Finally at load cell 4 the target pH is reached.

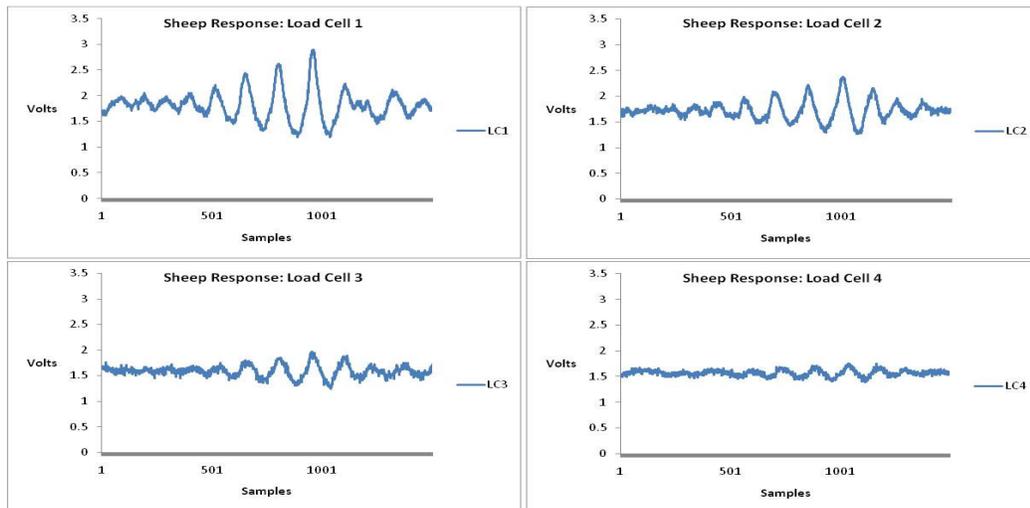


Figure 5.6: Lamb carcass response.

This gradual decrease in response to stimuli can be tracked using the pH sensor network. At each load cell the pH is analysed and the sensor node sends the value to the central node. Therefore the carcass number and its pH value are recorded. This data can be plotted to obtain the pH curve for each carcass.

Analysed pH						
LC - Load cell	Beef		Lamb			
	LC1	LC2	LC1	LC2	LC3	LC4
MATLAB (PC)	6.6054	6.2859	6.0791	5.8526	5.7169	5.695
DSK6713 (Embedded DSP)	6.6054	6.2859	6.0791	5.8526	5.7169	5.695

Table 5.1: Analysed pH values.

5.2 pH sensor network simulation using OPNET

The main objective of the simulation, both in OPNET and on development platform (eZ-RF2480) was to confirm the capability of ZigBee based pH sensor network to manage the meat tenderizing process. Figure 5.7 below shows the pH sensor network setup simulated in OPNET using its built-in ZigBee module. Node '0' is the coordinator that starts and forms the network. Nodes '0', '5', '6' and '8' represents pH sensor node. Nodes '1', '2', '3', '4' and '7' represent stimuli actuator node. These are listed in Table 5.2. Each square grid in the background represents a distance of 50m. Therefore the network radius was approximately 25m.

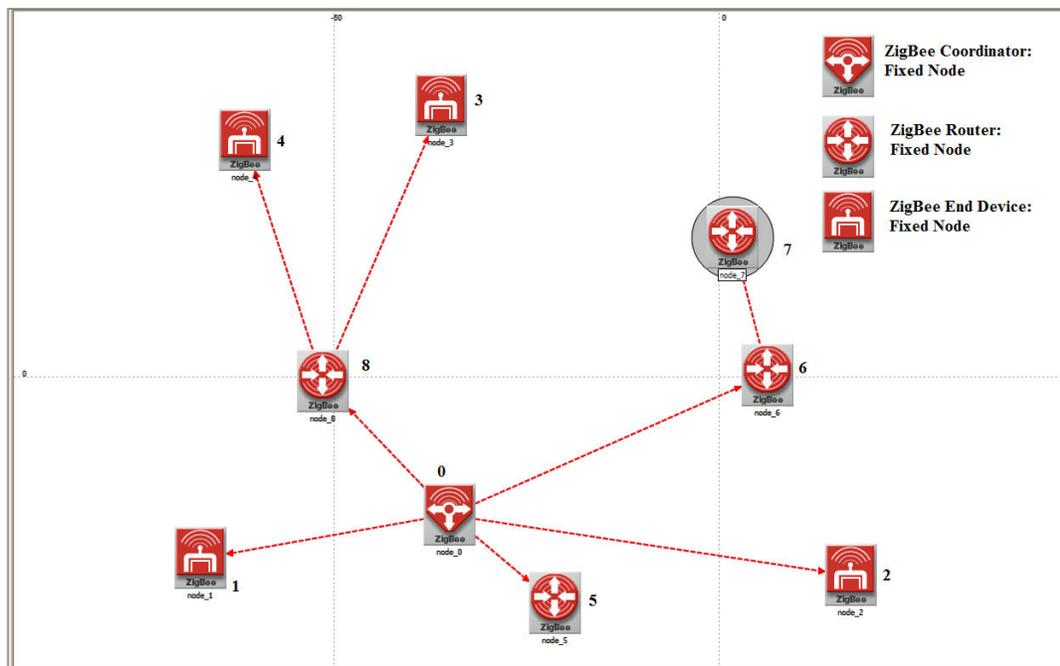


Figure 5.7: OPNET Simulation – pH sensor network.

Stimuli actuator node can be either a router or end device. If it is an end device it needs to activate periodically to check for data with its parent device. This is one of the

features that enables power saving. It will introduce a delay equivalent to the ‘wake-up’ time of the end device but it can also be set to zero so the node does not ‘sleep’. Node ‘0’ communicates with two stimuli actuator nodes, router and an end device. This is to compare the end-to-end delay for single hop and double hop transmissions in this network. The physical environment in this case was set to office, which is a square (10m x 10m) free space. The transmission was Omni directional (line-of-sight). This simulation was run for ten hours to gather node global statistics first.

Coordinator	Depth 1	Depth 2
pH Sensor Node: Node_0 (Coordinator/Sensor)	Stimuli Actuator Node: Node_1 (End Device)	Stimuli Actuator Node: Node_3 (End Device)
	Stimuli Actuator Node: Node_2 (End Device)	Stimuli Actuator Node: Node_4 (End Device)
	pH Sensor Node: Node_5 (Router)	Stimuli Actuator Node: Node_7 (Router)
	pH Sensor Node: Node_6 (Router)	
	pH Sensor Node: Node_8 (Router)	

Table 5.2: Node description for sensor network in Figure 5.7.

a. Simulation results: Global Statistics

The critical network data obtained from the simulation is listed in Table 5.3. This is global statistics of the network operation for ten hours.

PAN ID	Channel	Initial Network Formation (Seconds)	Number of Nodes	Tree Depth	End-to-End Delay (milliseconds) for 2 node network		
					Maximum	Minimum	Average
0	26	11.45	9	2	6.18	6.11	6.16

Table 5.3: Global statistics.

The end-to-end delay indicates that on average each carcass will receive six milliseconds of unwanted stimulation after “OFF” command are sent. This is negligible to cause any adverse effects to the carcass. However, any packet dropped will result in retries which may lead to increased communication delays. Normally this will take a few milliseconds. Some strategies are proposed in further work in Chapter 6 to accommodate transmission failures that may affect the tenderizing process.

b. Simulation results: Node Statistics

The network must be initiated and the nodes must join the network before actual sensor data and actuator commands can be transmitted. A number of activities occur during this network formation and joining stage at the lower levels of the ZigBee protocol, mainly the IEEE 802.15.4 MAC and PHY layers. However the critical data for this

project is the end-to-end delay as discussed earlier. The values obtained from the simulation of the network shown in Figure 5.7 were as follows:

- i. **pH Sensor Node end-to-end delay**
Peak = 6.28 milliseconds, Average = 6.17 milliseconds.
- ii. **Stimuli Actuator Node end-to-end delay**
Peak = 6.20 milliseconds, Average = 6.14 milliseconds.

These values are negligible in comparison with stimulation duration which normally lasts for three seconds. The main concern is the successful transfer of data/command from one node to another. Therefore, a simple ON/OFF command will take 6.20 milliseconds to reach its destination enroute various layers of the ZigBee protocol.

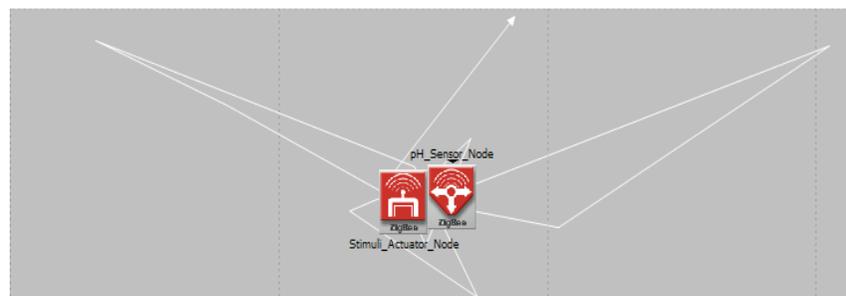


Figure 5.8: Trajectory for pH sensor node.

Figure 5.8 above shows the trajectory for pH sensor node. In this simulation the pH sensor node is made mobile which moves along the path shown by the trajectory line. The stimuli node is fixed. The effect of varying transmitter power, receiver sensitivity and distance on packet transmission was obtained without physically going out in the field to perform these tests. First the transmitter power and receiver sensitivity were set to the maximum value allowable in the simulation:

- Transmit power = 50mW.
- Receiver sensitivity = -95dBm.

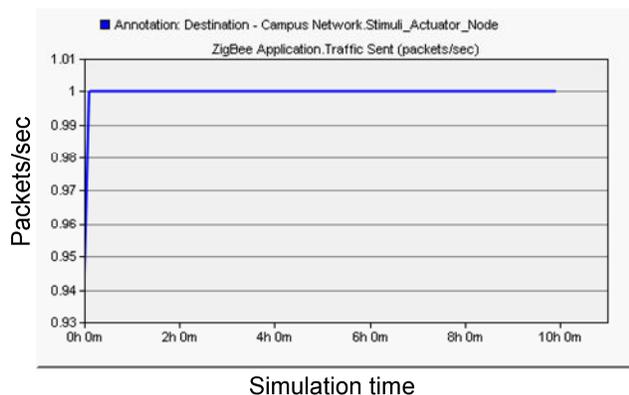


Figure 5.9: Packets sent by pH sensor node.

Figure 5.9 shows traffic sent by the pH sensor node to the stimuli actuator node as it moved along the path shown in Figure 5.8. A packet was sent every second continuously. The packets received by the stimuli actuator node are shown in Figure 5.10. The peaks indicate all packets received which also correspond to the distance between the nodes being smallest and ideal location points for the node. As the distance increased packets are dropped and the curve starts to roll off until it reaches zero where communication is totally lost. It rises again as the nodes get closer and communication is re-established.

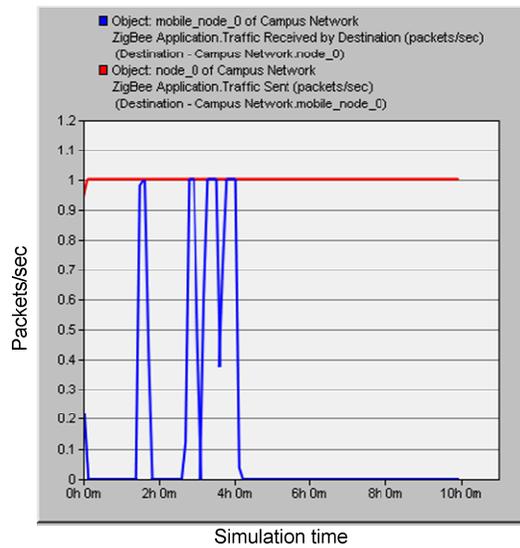


Figure 5.10: Power = 50mW, Sensitivity = -95dBm.

The simulation was repeated with using actual parameters of CC2480 device which is used in the prototypes. These parameters were set as below:

- Transmit power = 1mW.
- Receiver sensitivity = -92dBm.

The plot in Figure 5.11 shows significant loss of packets compared to the plot in Figure 5.10. This is due to lower transmit power level of the CC2480 device which is used for applications in the unlicensed ISM band. There is strict regulation on transmit power level for devices operating in the unlicensed band besides other parameters [59, 63]. This simulation clearly showed that transmitter power and receiver sensitivity are essential parameters to be considered in wireless hardware design. OPNET simulation also provided useful node statistics such as time for establishing a network from power-up (Seconds), Traffic sent/received (packets/sec), End-to-End delays (Seconds) and Retransmission Attempts (packets).

Simulation packages are essential tools for use in the early stages of system design. To get more realistic data the OPNET simulation scenario was repeated using eZ-RF2480 kit. This is discussed in section 5.3 below.

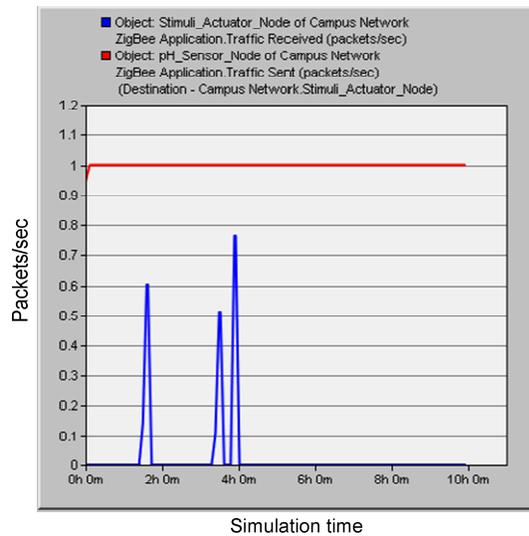


Figure 5.11: Power = 1mW, Sensitivity = -92dBm (CC2480 Parameters).

5.3 pH sensor network simulation on eZ-RF2480 kit

The pH sensor network was simulated in hardware using eZ-RF2480 development kit. It also provided a medium to understand Z-Stack functionality and simple API (simplified Z-Stack API by Texas Instruments). The core components of the eZ-RF2480 are illustrated in Figure 5.12. The node application program development, debugging and programming the MSP430 devices were completed in IAR Embedded Workbench Kickstart IDE.

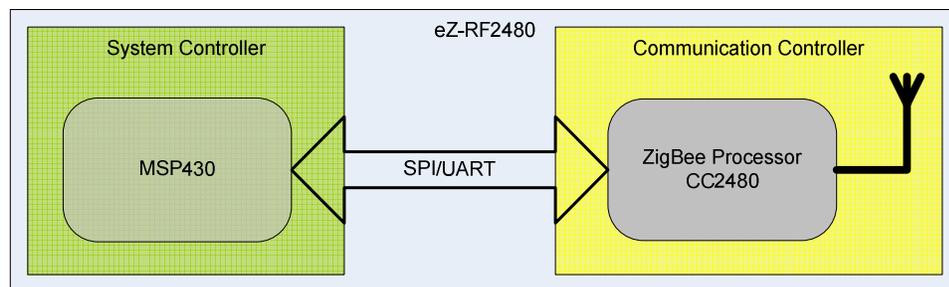


Figure 5.12: eZ-RF2480 block diagram.

First the programs were written for the four target boards included in the eZ-RF2480 kit to setup a ZigBee network of four nodes. Secondly a ZigBee network was setup using two nodes only to repeat the OPNET simulation using real hardware. These are

discussed in the following subsections. The simulation setup is shown in Figure 5.13 and discussed in section 5.3.1. Over-the-air packets during network operation were captured using CC2430DB platform and decoded using Packet Sniffer software. This provided an insight into the activity during network start-up, joining the network and data/command exchange between nodes. This was also an excellent method of understanding the ZigBee protocol.

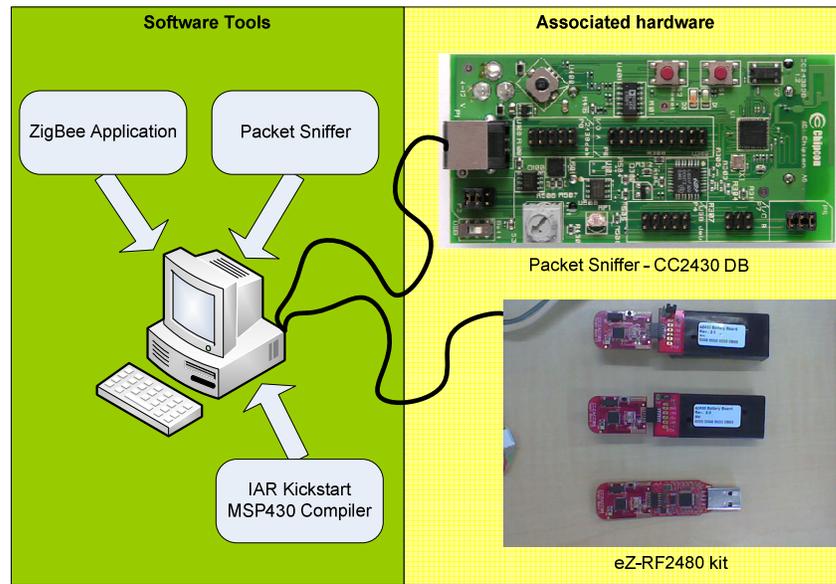


Figure 5.13: pH sensor network simulation setup.

5.3.1 eZ-RF2480 target board program

Four eZ-RF2480 target boards were used. These were programmed and configured as shown in Table 5.4.

Target Board	Node Application	Node Type
1	pH Sensor Node	Coordinator
2	pH Sensor Node	Router
3	Stimuli Actuator Node	Router/End Device
4	Stimuli Actuator Node	Router/End Device

Table 5.4: Node configuration.

The network is commissioned when all four target boards are powered up. Sensor Node 1 is configured as the coordinator so it will form the network. Other nodes then join the network. Sensor Node 1 program flowchart is shown in Figure 5.14. Sensor Node 1 generates a random 8-bit number and sends it to Sensor Node 2. This number depicts carcass pH at load cell 1(Chapter 3, Figure 3.1). It also compares this number with a set range and if it is within range (for example between 50 and 70) it sends “OFF”

command to Actuator Node 3 and if out of range sends “ON” command to Actuator Node 3. This is simulating the switching of stimulation pulses via the relay bank in the SmartStim system. Sensor Node 1 also uploads transmitted/received data to a PC via USB. This represents the recording of carcass pH data at the meat plant. The data is captured using terminal software (Windows Hyper Terminal).

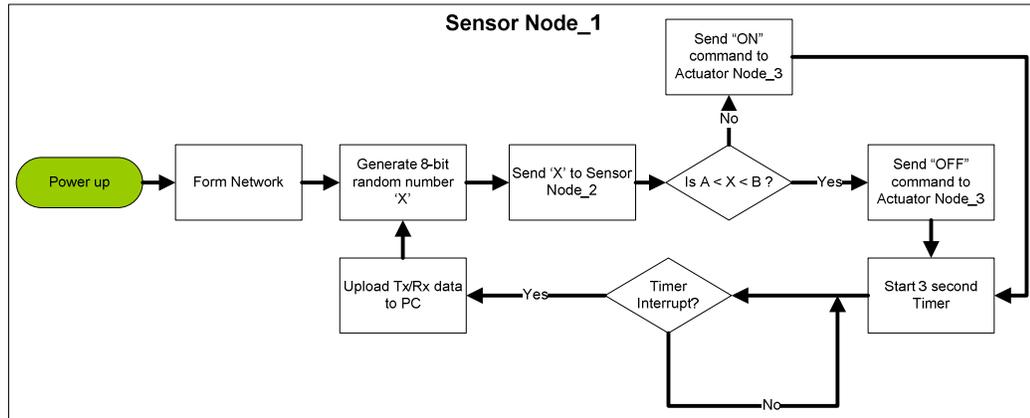


Figure 5.14: Node_1 program flowchart.

Sensor Node 2 program flowchart is shown in Figure 5.15. It represents sensor node at load cell 2 in Figure 3.1. Sensor Node 2 also compares the received data with a set range and accordingly sends either “ON” or “OFF” command to Actuator Node 4. It also sends the received data back to Sensor Node 1. This represents carcass pH at load cell 2.

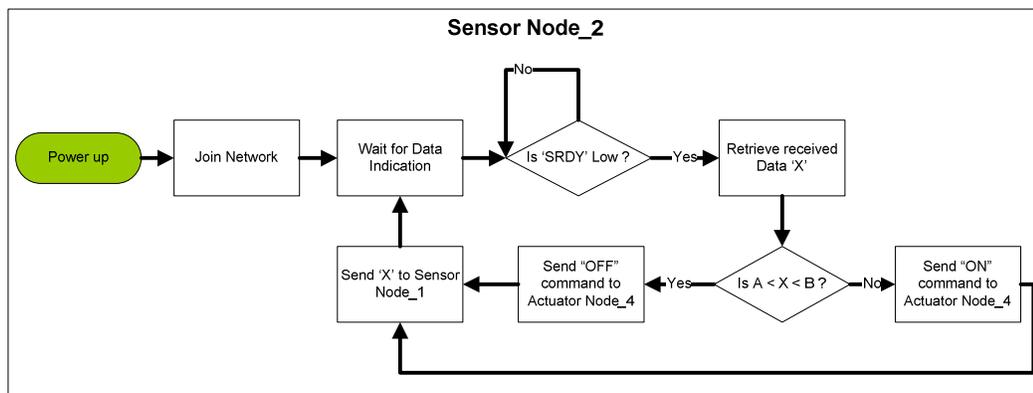


Figure 5.15: Node_2 program flowchart.

Actuator Node 3 and Node 4 have same program as shown in Figure 5.16. The received commands are decoded and either green or red LED is activated. This program represents the activation of TEST/STIM relays. The network traffic, transmission time and related parameters are examined using the packet sniffer tool which is discussed in section 5.2.4.

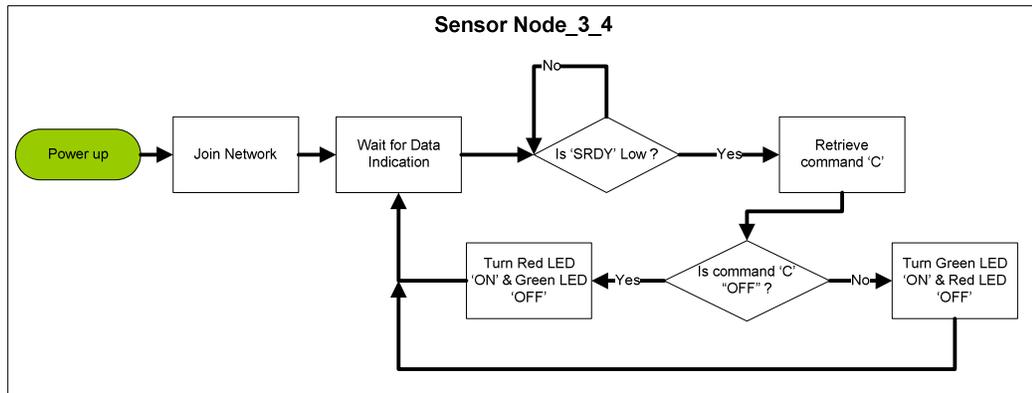


Figure 5.16: Node_3 & 4 program flowchart.

5.3.2 eZ-RF2480 simulation results

The ZigBee network is setup to repeat the OPNET simulation using actual hardware to determine range and link quality. Target board 1(pH sensor node) was fixed and Target board 3 (stimuli actuator node) was moved away until it went out of range and communication ceased. Target board 3 was then moved back towards Target board 1 and it automatically rejoined the network. It was then moved behind a brick wall. The maximum distance between the target boards was 5m and the Link Quality Indicator (LQI) was recorded every 3 seconds using the built-in feature in Z-Stack. LQI is direct indicator of RSSI (Received Signal Strength Indicator) in negative dBm [55, 73, 74]. Figure 5.17 shows the plot of LQI values which is similar to OPNET simulation plot in Figure 5.11.

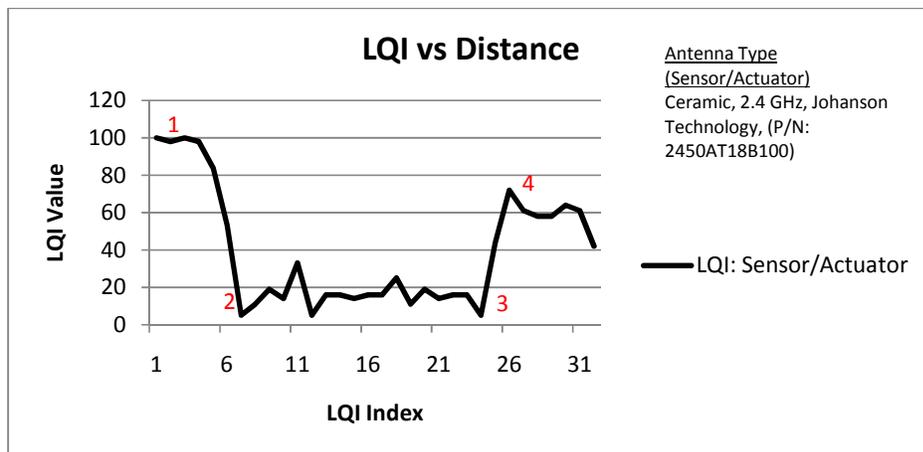


Figure 5.17: LQI Chart.

The data labels in Figure 5.17 are discussed below.

- [1] Distance between Sensor/Actuator: 30cm.
- [2] Distance between Sensor/Actuator: > 5m.
Communication lost.

- [3] Distance between Sensor/Actuator: < 5m.
Node rejoins network.
- [4] Distance between Sensor/Actuator: 30cm.
Node is behind a brick wall.

5.3.3 OPNET and eZ-RF2480 simulation comparison

Essentially two primary objectives were met with the above simulations. Firstly by sitting at the desk and using OPNET on the PC, the pH sensor network was established without any hardware. The network operated within a radius of 25m and the end-to-end delay was 6.28 milliseconds. Secondly the network was setup using eZ-RF2480 kit which consists of CC2480 devices and the environment is real. The network operating radius was 5m beyond which communication was lost as shown in Figure 5.17. The OPNET simulation provided global and nodal statistics based on an ideal environment to evaluate the pH sensor network. The simulation environment was open free space but allowed visualization of influential factors such as transmitter power, receiver sensitivity and distance. The LQI values in Figure 5.17 reflect the performance of the network in a real environment using real hardware. Therefore a significant reduction in network radius is noticed. Finally both simulation results indicated OPNET as a useful tool to extrapolate RF hardware design and performance. However it is noted that actual performance of the network in real operating environment also depends on physical factors such as antenna type, path obstruction and distance between nodes. Specific study related to RF performance, antenna types, and surroundings is beyond the scope of this project.

5.3.4 Packet analysis using Packet Sniffer

Packet Sniffer is a powerful tool for debugging the ZigBee network. For instance communication between devices via SPI is functional but there is no response from remote node. This tool is used to view activity in the invisible communication space. It is also an essential learning tool to understanding the ZigBee protocol and Z-Stack. Figure 5.18 shows packets captured using Packet Sniffer tool during data transfer from sensor node (coordinator) to an actuator node in a beacon-enabled network.

Beacon is a message with specific format that is used to synchronize the clocks of the nodes in the network. Beacon-enabled networks dedicate a specific time slot to a particular device. This is called guaranteed time slot (GTS). In Figure 5.18 an ‘OFF’ command is sent to the actuator node. According to Table 4.1 the ‘OFF’ command is designated ‘0’. Therefore to turn off a relay ‘0’ is sent to the actuator node on endpoint 2 (see Figure 4.5). Referring to Figure 5.18, this captured and decoded unicast frame shows the parameters specific to each protocol layer.

The IEEE 802.15.4 defines four general MAC frame structures. These are the beacon frame, the data frame, the acknowledgment frame and the MAC command frame. All these frames were captured and viewed during the pH sensor network simulation. However, the data frame (Frame 1 in Figure 5.18) which carries the commands and pH values in this application is studied in more detail. It consists of the MAC frame, the NWK frame and the APS frame. Each of these three frames that makeup the data frame is described briefly.

Frame 1: ‘OFF’ command sent from Sensor node to Actuator node

P.nbr.	Time (ms)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	
RX	+3	29	Type	Sec	Pnd	Ack.req	PAN_compr				
251	=274842		DATA	0	0	1	1	0x42	0x04B3	0x0001	0x0000

MAC payload		IWK Frame control field				IWK Dest. Address	IWK Src. Address	Broadcast Radius	Broadcast Seq.num	IWK payload			
48	00	01	00	00	00	0A	3C	40	Type	Version	DR	MF	Sec
02	01	00	10	0F	02	37	00	00	DATA	0x2	1	0	0

APS Frame control field				APS Dest. Endpoint	APS Cluster Id	APS Profile Id	APS Src. Endpoint	APS Counter	APS Payload	LQI	FCS
Type	Del.mode	Ind.am	Sec Ack								
Data	Unicast	0	0	1	0x02	0x0001	0x0F10	0x02	55	00	00

Frame 2: Acknowledgment frame from Actuator

P.nbr.	Time (ms)	Length	Frame control field				Sequence number	LQI	FCS	
RX	+1	5	Type	Sec	Pnd	Ack.req	PAN_compr			
252	=274843		ACK	0	0	0	0	0x42	84	OK

Frame 3: Data Request frame from Actuator node

P.nbr.	Time (ms)	Length	Frame control field				Sequence number	Dest. PAN	Dest. Address	Source Address	
RX	+9	27	Type	Sec	Pnd	Ack.req	PAN_compr				
253	=274853		DATA	0	0	1	1	0x3C	0x04B3	0x0000	0x0001

MAC payload		IWK Frame control field				IWK Dest. Address	IWK Src. Address	Broadcast Radius	Broadcast Seq.num	IWK payload		
48	00	00	00	01	00	0A	33	Type	Version	DR	MF	Sec
02	02	01	00	10	0F	02	37	DATA	0x2	1	0	0

APS Frame control field				APS Dest. Endpoint	APS Cluster Id	APS Profile Id	APS Src. Endpoint	APS Counter	LQI	FCS	
Type	Del.mode	Ind.am	Sec Ack								
ACK	Unicast	0	0	0	0x02	0x0001	0x0F10	0x02	55	84	OK

Frame 4: Acknowledgment frame from Sensor node

P.nbr.	Time (ms)	Length	Frame control field				Sequence number	LQI	FCS	
RX	+1	5	Type	Sec	Pnd	Ack.req	PAN_compr			
254	=274854		ACK	0	0	0	0	0x3C	140	OK

Figure 5.18: Packets captured by Packet Sniffer tool.

The general MAC frame consists of the MAC header (MHR), the MAC payload and the MAC footer (MFR). The first field is the frame control that defines the frame type. This could be either beacon, data, acknowledgement, or MAC command frame. The other fields that are part of the MAC frame control field are Sec (Security Enabled), Pnd (Frame Pending), Ack.req (Ack. Request), PAN_compr (PAN ID Compression).

Similarly the NWK and APS frames have frame control field. The fields for the NWK frame control field are Type (Frame Type), Version (Protocol Version), DR (Discover Route), MF (Multicast Flag) and Sec (Security). The APS frame control fields are Type (Frame Type), Del.mode (Delivery Mode), Ind.am (Indirect Address Mode), Sec (Security) and Ack (Ack Request). The Packet Sniffer tool was indeed a very useful tool to understand the ZigBee protocol.

Figure 5.19 shows the SmartStim system installed at Auckland Meat plant. All the components of the system are shown in this figure. The load cell location is aligned vertically with the short segment of the stimulation bar as shown in Figure 5.19. The load cells are used to sense the response when test pulses are applied since this data is used to analyse the pH. The carcass makes contact with the stimulation bar and return electrode as it is carried on the conveyor.

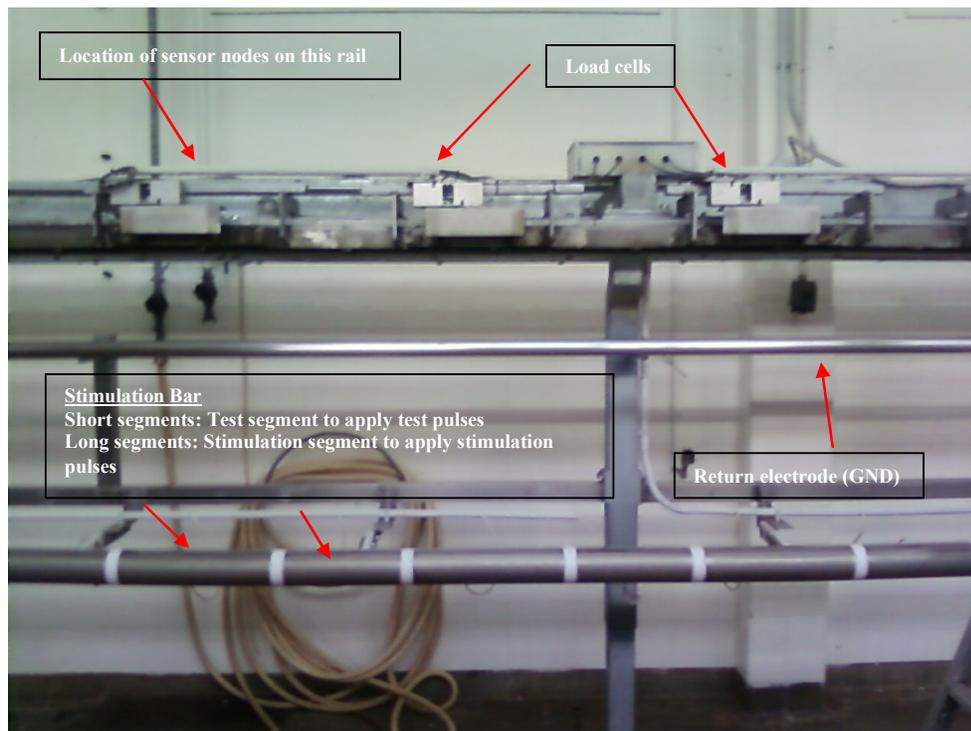


Figure 5.19: SmartStim Installation site at Auckland Meat plant (Source: [17]).

5.4 Summary

MATLAB and OPNET simulations provided vital data that was used in the development of the prototypes. MATLAB results confirmed that the pH calculation function was producing results same as those obtained from Carne Technologies Ltd. This “fft2pH” function was then written in ‘C’ language and tested on the DSP DSK6713. This confirmed the program was producing same results as MATLAB. OPNET simulations showed the effects of receiver sensitivity, transmitter power and distance on packet transmission. Many tedious and repetitive tasks was performed using simulations to minimize development cost, resolve design issues and identify suitable components for actual system development.

6 Conclusion and future work

This chapter sums up the justification for the choice of a wireless system and highlights the issues related to sensor/actuator node architectures. The issues relates to hardware and software development for the system and communication controller. The study concludes with suggestions for future work on possible improvements to pH wireless sensor network application for the meat tenderizing process.

6.1 Conclusion

The research concluded with the development of a pH wireless sensor network for the meat tenderization process as an alternative to the existing conventional wired system. The project achieved its objective of technological innovation in the meat industry for improving the quality and efficiency with a cost effective system. The technology known as SmartStim developed by Carnes Technologies Ltd (proprietorship of the technology) provided the necessary tools and baseline for developing the pH WSN. pH WSN evolved from a detail analysis and assessment of the wired system which is elaborately documented in the five (5) chapters and even leading to the development of pH sensor node and stimuli actuator node prototypes. The prototype circuit board is a multipurpose board which could be used for numerous other applications such as a totally non-intrusive wireless pH sensor.

The main focus of the study was in the slaughter chain of beef and sheep carcass. A proprietary pH analysis algorithm was provided and the tenderising mechanism was explained. Based on this information a conceptual pH wireless sensor network was developed including architectures for the sensor/actuator node. The concept is based on the principles of distributed and collaborative processing. The proof-of-concept was simulated on software simulation tools MATLAB and OPNET. It was verified by developing and testing on commercially available development platforms. Sensor and actuator node prototypes were also developed based on the proposed concept and handed to the sponsor organisation for assembling and testing.

Wireless technique seems to be the best alternative with reference to reliability, adaptability and scalability. Signal reliability between Transmitter/Receiver is affected by path loss (wall, obstructions), RF Interference (ambient noise) and transmit power. Collectively these determine the bit error rate. Sensor network would overcome this by locating router nodes that allow alternate route when communication path is obstructed. ZigBee protocol was chosen for wireless communication because it is built on robust IEEE 802.15.4 standard with free stack available from many vendors. However transmission of power is restricted due to operation in an unlicensed ISM band which can be considered as a good trade-off between cost and licensing.

Adaptability to the network environment is of critical importance. The nature of the network is utilised where location of nodes can overcome the restrictions permitted by the environment. That is plant, pillars or fixed structures cannot be moved to accommodate the systems. The network should adapt to the existing environment. The choice of network is tree topology which can easily be extended to a mesh network by reconfiguring the nodes. The scalability provided by the ZigBee protocol allows up to 65535 nodes. This is far more than what will ever be required for this application but it shows that effortless expansion is possible as number of endpoints increases.

The study provides future direction for research especially with antennas and range as few different types of antennas can be used in various scenarios to select the best antenna for this application that is reliable and cost effective. ZigBee is a certified protocol but a proprietary communication protocol can also be developed. A trade-off between reliability and development cost need to be performed. Field trials and communication issues needs to be further explored. The prototype needs to be operated in actual environment. Performance data to be collected and hardware software changes made accordingly. pH analysis algorithm can be further developed using adaptive techniques so that it can automatically adjust the overall processing and analysis based on carcass responses and carcass type. Mechanisms to handle system failures also need further development.

7 References

1. Chrystall, B.B., & Devine C.E. (1978). Electrical stimulation, muscle tension and glycolysis in bovine Sternomandibularis. *Meat Science*, 2(1), 49-58.
2. Locker, R.H., et al. (1975). New Concepts in Meat Processing. *Advances in Food Research*, 21, 157-222.
3. Dransfield, E. (1987). Fundamental of meat chilling, electrical stimulation and conditioning. *International Journal of Refrigeration*, 10(2), 110-112.
4. Geesink, G.H., et al. (2001). Effects of stress and high voltage electrical stimulation on tenderness of lamb m. longissimus. *Meat Science*, 57(3), 265-271.
5. Koochmaraie, M., Shackelford, S.D., & Wheeler T.L. (2005). Biological base that determines beef tenderness. *British Society of Animal Science*, 21-25.
6. Solomon, M.B. (1986). Effect of different types and locations of the electrode source of an extra low voltage electrical stimulation system on beef quality. *Meat Science*, 16(3), 217-224.
7. Taylor, D.G., & Cornell, J.G. (1986). The effects of electrical stimulation and ageing on beef tenderness. *Meat Science*, 12(4), 243-251.
8. Bickerstaffe, R., et al. (2001). Impact of introducing specifications on the tenderness of retail meat. *Meat Science*, 59(3), 303-315.
9. Monin, G. (1998). Recent methods for predicting quality of whole meat. *Meat Science*, 49(1), 231-243.
10. Simmons, N.J., et al. (2006). Integrated technologies to enhance meat quality - An Australasian perspective. *Meat Science*, 74(1), 172-179.
11. *SMEQ*. (n.d). Retrieved August7, 2008, from <http://www.beeflambnz.co.nz/qualityMark/qualityMark-customer.html>.
12. Toohey, E.S., et al. (2008). The impact of new generation pre-dressing medium-voltage electrical stimulation on tenderness and colour stability in lamb meat. *Meat Science*, 79, 683-691.
13. Simmon, N., et al. (2006). *Integrated Technologies for Enhanced Meat Quality*. Presented at ICoMST, Dublin, Ireland.
14. Seideman, S.C., & Cross, H.R. (1982). Utilization of electrical stimulation to improve meat quality: A Review. *Journal of Food Quality*, 5(4), 247-269.
15. Geesink, G.H., et al. (2001). Electrical stimulation -- when more is less. *Meat Science*, 57(2), 145-151.

16. Sørheim, O., & Hildrum, K.I. (2002). Muscle stretching techniques for improving meat tenderness. *Trends in Food Science & Technology*, 13(4), 127-135.
17. *SmartStim*TM (n.d.). Retrieved August 17, 2008, from <http://www.redmeatinnovation.com.au/innovation-areas/eating-quality/products/smartstim>.
18. Simmons, N.J., et al. (2008). Reassessing the principles of electrical stimulation. *Meat Science*, 80(1), 110-122.
19. Gislason, D. (2008). *ZigBee Wireless Networking*. Oxford: Newnes.
20. Li, Z., (2006). ZigBee Wireless Sensor Network in Industrial Applications. *SICE-ICASE*, 1067-1070.
21. Khakpour, K., & Shenassa, M.H. (2008). Industrial Control using Wireless Sensor Networks [Electronic version]. *Information and Communication Technologies: From Theory to Applications*, 1-5.
22. Jui-Yu, C., Min-Hsiung, H., & Jen-Wei, C. (2007). A ZigBee-Based Power Monitoring System with Direct Load Control Capabilities. *Networking, Sensing and Control*, 895-900.
23. *Getting Started with MATLAB*. Retrieved August 18, 2008, from http://www.mathworks.com/access/helpdesk/help/techdoc/learn_matlab/bqr_2pl.html.
24. *OPNET Technologies Inc.* Retrieved January 12, 2009, from <OPNET\14.5A\doc\modeler\STM\STM-44-01.html>.
25. *PACKET-SNIFFER: SmartRF Protocol Packet Sniffer*. Retrieved February 8, 2009, from <http://focus.ti.com/docs/toolsw/folders/print/packet-sniffer.html>.
26. *CC2430DB:CC2430 Demonstration Board*. Retrieved February 10, 2009, from <http://focus.ti.com/docs/toolsw/folders/print/cc2430db.html>.
27. *IAR Embedded Workbench for TI MSP430*. Retrieved October 3, 2008, from <http://www.iar.com/website1/1.0.1.0/220/1/>.
28. *CCS: Code Composer Studio Integrated Development Environment (IDE)*. Retrieved October 10, 2008, from <http://focus.ti.com/docs/toolsw/folders/print/ccstudio.html>.
29. *SPRC121:TMS320C67x DSP Library*. Retrieved October 12, 2008, from <http://focus.ti.com/docs/toolsw/folders/print/sprc121.html>.
30. *CodeVisionAVR V2.04.5 User Manual*. Retrieved September 5, 2008, from http://www.hpinfotech.ro/html/cvavr_doc.htm.
31. *Z-Stack: ZigBee Protocol Stack*. Retrieved October 7, 2008, from <http://focus.ti.com/docs/toolsw/folders/print/z-stack.html>.
32. *Altium Designer 6 An Introduction*. (n.d.). (2005). Belrose, NSW, Australia: Altium Ltd.

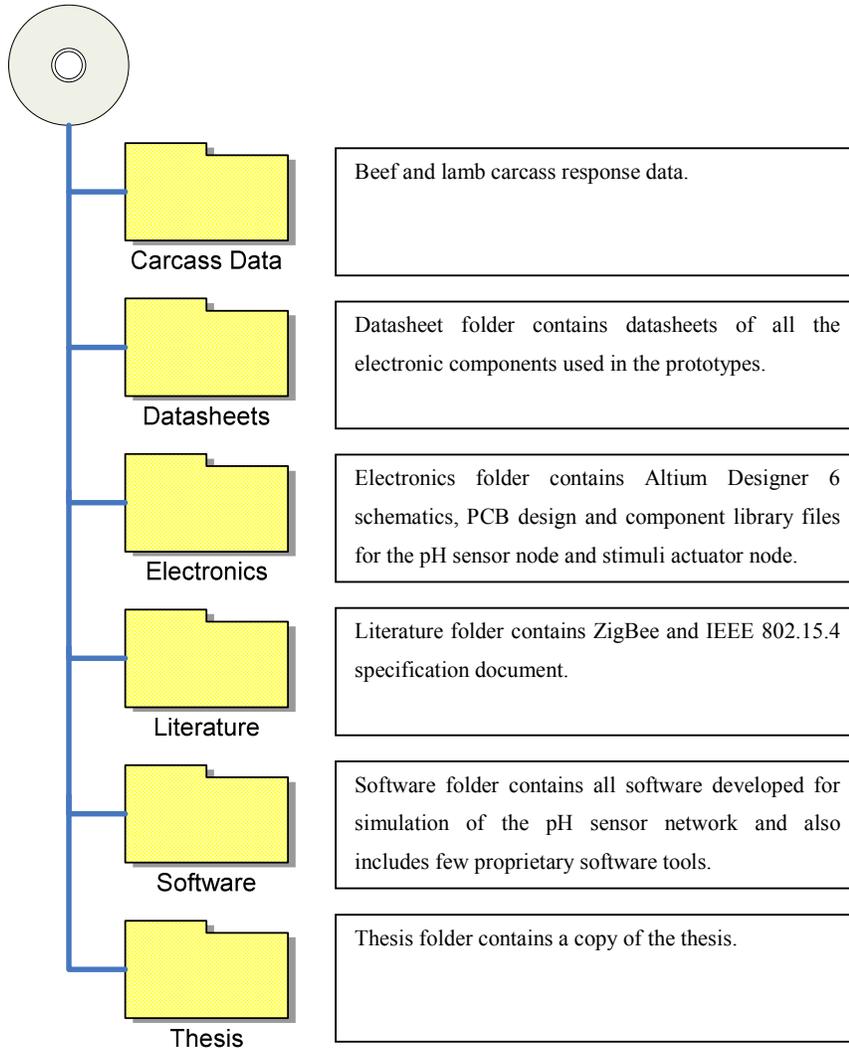
33. *TMDSDSK6713:TMS320C6713 DSP Starter Kit (DSK)*. Retrieved October 23, 2008, from <http://focus.ti.com/docs/toolsw/folders/print/tmstdsk6713.html>.
34. *Z-Accel Demonstration Kit: EZ-RF2480*. Retrieved February 7, 2009, from <http://focus.ti.com/docs/toolsw/folders/print/ez430-rf2480.html>.
35. *MSP430F2274:16-bit Ultra-Low-Power Microcontroller, 32kB Flash, 1K RAM*. Retrieved February 7, 2009, from <http://focus.ti.com/docs/prod/folders/print/msp430-f2274.html>.
36. *CC2430: System-on-Chip Solution for 2.4 GHz IEEE 802.15.4 / ZigBee™*. Retrieved February 8, 2009, from <http://focus.ti.com/docs/prod/folders/print/cc2430.html>.
37. *IEEE 802.15.4*. Retrieved December 10, 2008, from <http://standards.ieee.org/getieee-e802/download/802.15.4d-2009.pdf>.
38. *CC2480 Data Sheet (Rev.A)*. Retrieved December 4, 2008, from <http://focus.ti.com/lit/ds/swrs074a/swrs074a.pdf>.
39. Howitt I., Manges W.W., Kuruganti P.T., Allgood G., Gutierrez, J.A., Conrad J.M. (2006). Wireless industrial sensor networks: Framework for QoS assessment and QoS management, *The Instrumentation, Systems and Automation Society*, 45 (3), 347-359.
40. *TMS320C6713*. Retrieved September 17, 2008, from <http://focus.ti.com/docs/prod/folders/print/tms320c6713>.
41. *Decentralized Computing*.(n.d.). Retrieved August 23, 2008, from http://en.wikipedia.org/wiki/Decentralised_computer_system.
42. Proakis, J.G., & Manolakis, D.G. (1996). *Digital Signal Processing: Principles, Algorithms and Applications*. New Jersey: Prentice Hall.
43. *C2833x/C2823x C/C++ Header Files and Peripheral Examples*. Retrieved March 14, 2009, from <http://focus.ti.com/docs/toolsw/folders/print/sprc530.html>.
44. *ISM band*. Retrieved October 7, 2008, from http://en.wikipedia.org/wiki/ISM_band
45. *Wireless Technology Comparison*. (n.d). Retrieved October 7, 2008, from www.dpactech.com/docs/evaluation_support/Wireless_Technology_Comparison.pdf.
46. Baker, N. (2005). Zigbee and Bluetooth: strengths and weaknesses for industrial applications. *IEE Journal of Computing and Control Engineering*, 20-25.
47. Baker, N. (2005). Bluetooth: Strengths and Weaknesses for Industrial applications. *IEE Journal of Computing and Control Engineering*, 21-25.
48. *ZigBee Specification including the Pro Feature Set*. Retrieved November 2, 2008, from <http://www.zigbee.org/Products/DownloadZigBeeTechnicalDocuments.aspx>.

49. *TMS320F28235/28234/28232: Digital Signal Controllers (DSCs) Data Manual SPRS439E (2008)*. Retrieved February 13, 2009, from <http://focus.ti.com/dsp/docs/dspcontent.tsp?contentId=1107>.
50. *SINGLE-CHIP USB TO UART BRIDGE CP2103, Silicon Laboratories, 2005*. Retrieved March 4, 2009, from <http://www.silabs.com/products/interface/usbtourat/Pages/default.aspx>.
51. *Single Supply, MicroPower INSTRUMENTATION AMPLIFIER, INA122UA. PDS-1388B, BURR- BROWN Corporation, 1997*. Retrieved March 4, 2009, from <http://focus.ti.com/docs/prod/folders/print/ina122.html>.
52. *5-V, Low Power, 16-Bit, 200-KSPS serial Analog-to-Digital converters with Auto-Power down. SLAS293 – DECEMBER 2001 TLC4545*. Retrieved March 4, 2009, from <http://focus.ti.com/docs/prod/folders/print/tlc4545.html>.
53. Hammoodi, I.S., Stewart B.G., Kocian, A., & McMeekin, S.G. (2009). A comprehensive Performance Study of OPNET Modeler For ZigBee™ Wireless Sensor Networks. *NGMAST*, 357-362.
54. *Tedea Huntleigh load cell datasheet*. Retrieved December 18, 2008, from <http://www.vishay.com/test-measurements/list/product-12020/>.
55. *Calculation and usage of LQI and RSSI*. Retrieved January 7, 2009, from http://e2e.ti.com/support/low_power_rf/f/160/t/18233.aspx.
56. *Wireless Sensor Network*. Retrieved August 16, 2008, from http://en.wikipedia.org/wiki/Wireless_sensor_network.
57. Farahani, S. (2008). *ZigBee Wireless Networks and Transceivers*. Oxford: Newnes.
58. Bateman, A., & Patterson-Stephens, I. (2002). *The DSP Handbook*. London: Prentice Hall.
59. *ISM-Band and Short Range Device Regulatory Compliance Overview*. Retrieved February 3, 2009, from <http://focus.ti.com/lit/an/swra048/swra048.pdf>.
60. *Antenna Selection Guide*. Retrieved January 5, 2009, from <http://focus.ti.com/lit/an/swra161a/swra161a.pdf>.
61. *ISM-Band and Short Range Device Antennas (Rev. A)*. Retrieved January 5, 2009, from <http://focus.ti.com/general/docs/litabsmultiplefilelist.tsp?literatureNumber=swra046a>.
62. *AN048: 2.4GHz Chip Antenna (Rev. B)*. Retrieved January 5, 2009, from <http://focus.ti.com/general/docs/litabsmultiplefilelist.tsp?literatureNumber=swra092b>.
63. *AN032: 2.4 GHz Regulations*. Retrieved February 3, 2009, from [http://focus.ti.com-general/docs/litabsmultiplefilelist.tsp?literatureNumber=swra060](http://focus.ti.com/general/docs/litabsmultiplefilelist.tsp?literatureNumber=swra060).
64. *AN003 SRD Antennas*. Retrieved January 5, 2009, from <http://focus.ti.com/general/docs/litabsmultiplefilelist.tsp?literatureNumber=swra088>.

65. *DN018: Range Measurements in an Open Field Environment (Rev. A)*. Retrieved January 7, 2009, from <http://focus.ti.com/general/docs/litabsmultiplefilelist.tsp?literatureNumber=swra169a>.
66. *ZigBee Cluster Library*. Retrieved February 22, 2009, from <http://www.zigbee.org/Products/ZigBeeClusterLibraryDownload.aspx>.
67. Durrant-Whyte, H.F., Rao, B.Y.S., Hu, H. (1991). Toward a fully decentralized architecture for multi-sensor data fusion. *Principles and Applications of Data Fusion, IEE Colloquium*, 2, 1-4.
68. *TMS320C67x DSP Library Programmer's Reference Guide*. Retrieved September 15, 2008, from <http://focus.ti.com/general/docs/litabsmultiplefilelist.tsp?literatureNumber=spru657c>.
69. Wellington, S. (2002). *Algorithms for sensor validation and multisensory fusion*. Nottingham Trent University, Nottingham, United Kingdom.
70. *CC2480 Interface Specification*. Retrieved December 10, 2008, from <http://focus.ti.com/lit/er/swra175a/swra175a.pdf>.
71. *Atmel AVR ATtiny88 – 8 Bit RISC MCU*. Retrieved October 10, 2008, from http://www.atmel.com/dyn/products/product_card.asp?part_id=4351.
72. *National Instruments PCI-6220 DAQ*. Retrieved July 24, 2008, from <http://sine.ni.com/nips/cds/view/p/lang/en/nid/14130>.
73. *CC2480 Developer's Guide*. Retrieved November 7, 2008, from <http://focus.ti.com/general/docs/litabsmultiplefilelist.tsp?literatureNumber=swra176>.
74. *RSSI on eZ430-RF2480*. Retrieved March 10, 2009, from http://e2e.ti.com/support/low_power_rf/f/158/t/17471.aspx?PageIndex=2.
75. *Ceramic Antenna, 2.4 GHz, Johanson Technology, (P/N: 2450AT18B100)*. Retrieved November 7, 2008, from http://kor.johansontechnology.com/products/rfc/ant/JTI_Antenna_2450AT18B100_06-12-2006.pdf.

8 Appendices

8.1 Appendix 1 CD content



8.2 Appendix 2 ‘fft2pH’ function and load cell data

The table below contains raw load cell data for beef response to stimuli. It only contains first twenty data points. 1500 data points are acquired in total for each carcass to calculate its pH. All load cell data is included in the enclosed CD [Appendix 1].

Raw data (Volts) Load cell 1	FFT	Beef Coefficients	Coefficients * FFT	SUM (Coefficients*FFT)	pH
Beef Constant = 6.2104					
2.375072	2.179626	0.0372	0.081082102	0.395046559	Calculated
2.363353	0.07539	-0.6731	-0.050745097		6.605447
2.355541	0.109591	0.8056	0.088286824		Measured
2.347728	0.12247	0.05	0.006123478		6.64
2.351634	0.110264	-1.3081	-0.14423626		
2.355541	0.062751	1.3164	0.082605179		
2.371166	0.035192	3.4152	0.12018847		
2.36726	0.086301	-1.0658	-0.091979755		
2.363353	0.120842	-0.2227	-0.02691158		
2.343822	0.091193	-0.5214	-0.047547811		
2.312571	0.053126	-0.9272	-0.049258854		
2.300851	0.018448	-0.7706	-0.014216306		
2.269601	0.078709	-0.9554	-0.075198951		
2.261788	0.087888	0.2721	0.023914298		
2.269601	0.091933	0.7033	0.064656198		
2.265694	0.125179	0.5057	0.063303248		
2.28132	0.168398	0.5083	0.085596708		
2.28132	0.209655	0.0764	0.016017625		

Table 8.1: Load cell data for beef carcass (Source: [17]).

The data in Table 8.1 was acquired using the following devices:

- Tedeia Huntleigh load cell (Model 1263) for sensing carcass response to stimuli.
- National Instruments PCI-6220 DAQ (16-Bit, 250 kS/s, 16 Analog Inputs) for data acquisition.

The MATLAB function “fft2pH.m” is explained here. The steps shown in Chapter 5, Figure 5.1 were written as “fft2pH.m”, an m-file in MATLAB to calculate the pH of a carcass from raw load cell data. This function was also used to verify results obtained on the DSK6713 DSP platform.

```
function pH = fft2pH(LCoutput, Coff, K)
```

[The arguments of the function are LCoutput = Raw load cell data, Coff = Carcass coefficients, K = carcass constant]

```
format long
```

```
sample = importdata(LCoutput);
```

```
coefficient = importdata(Coff);
```

[This takes FFT of 1500 points to compare with results from Carne Technologies Ltd]

```
fftx = fft(sample);
```

[FFT length of 1024 to verify results on the DSP platform DSK6713 since the FFT length in the DSP library can only be to the power of 2]

```
fftx = fft(sample,1024)
```

```
Rfft = abs(fftx);
```

```
slen = length(sample);
```

[Calculate the magnitude of the frequency components in the carcass response spectrum]

```
magnitude = Rfft/length(Rfft);
```

```
magnitude(2:end)=magnitude(2:end).*2;
```

```
Cofflen = length(coefficient);
```

[Multiply the frequency components with the carcass coefficients]

```
magnitude = magnitude(1:Cofflen).*coefficient;
```

[Calculate pH]

```
pH = sum(magnitude)+K
```

8.3 Appendix 3 Configuring a ZigBee device in Z-Stack

The Application Programming Interface (API) to setup the pH sensor network on eZ-RF2480 platform is explained here. The CC2480 device is preinstalled with Z-Stack and supports the simple API. Simple API is developed by Texas Instruments and consists of ten API calls used by the host microcontroller to communicate with the CC2480 device [70]. Configuring one of the target boards as a coordinator and using the simple API for communication with other nodes in the ZigBee network is discussed here.

Figure 8.1 shows the order of the commands for configuring a node as a coordinator in the ZigBee network. Full code is included in the enclosed CD. *zb_WriteConfiguration* is used to write a configuration parameter to the CC2480 device. The configuration parameters are divided into network-specific and device-specific parameters. The network-specific parameters must be set to the same value for all CC2480 devices in a ZigBee network. These parameters are stored in a non-volatile memory on the CC2480 device. Once the devices are configured as coordinator (only 1 device in a ZigBee network), router or end device, the following simple API commands are issued to the CC2480 device by the host MCU to start up the network.

1. ZB_APP_REGISTER_REQUEST

The host processor must register its application program with the CC2480 device. The registration parameters are Application Endpoint, Application Profile ID, Device ID, Device version, number of input commands or clusters and number of output commands or clusters. These parameters for the pH sensor network are given in section 3.3.1 and Table 4.1.

2. ZB_START_REQUEST

This command starts the ZigBee stack in the CC2480 device.

3. ZB_START_CONFIRM

This is issued by the CC2480 device. If the status return is success, the device is ready to send, receive and route network traffic.

4. ZB_PERMIT_JOINING_REQUEST

This command is used to control the joining permissions of a single device or entire network. It controls the timeout for joining the network.

Once network start up confirmation is received, communication between the nodes in the network can begin. The commands to send data/command to another remote device are shown in Figure 8.2. This can only happen after receiving network start up confirmation. The result of the send request is confirmed by reading the status flag returned by the CC2480 device. '0' is success and '1' is failure.

The CC2480 device communicates with the host processor via SPI/UART. The SRDY is the slave ready line controlled by CC2480. This can be polled by the host processor or it can be connected to one of the interrupt lines on the host processor. Figure 8.3 shows the polling method for receiving data/command from a remote device. The SRDY line is polled and getMSG() function is called to retrieve the received data/command.

SRDY line is driven low by the CC2480 device when it receives a packet over-the-air from a remote device. This packet is decoded by passing through various layers of the ZigBee protocol. The application program retrieves the data from the allocated message buffer. The application program running on the host processor has full control of all the devices in the network.

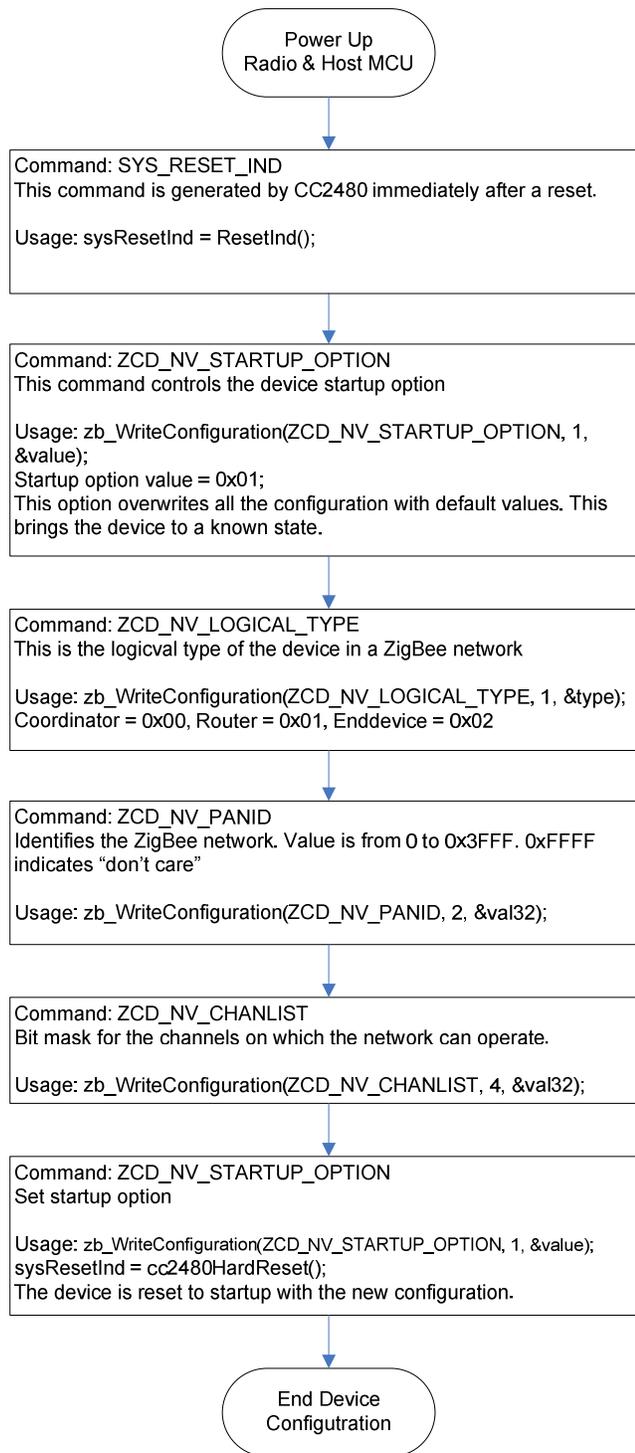


Figure 8.1: Commands for configuring a node as coordinator in a ZigBee network.

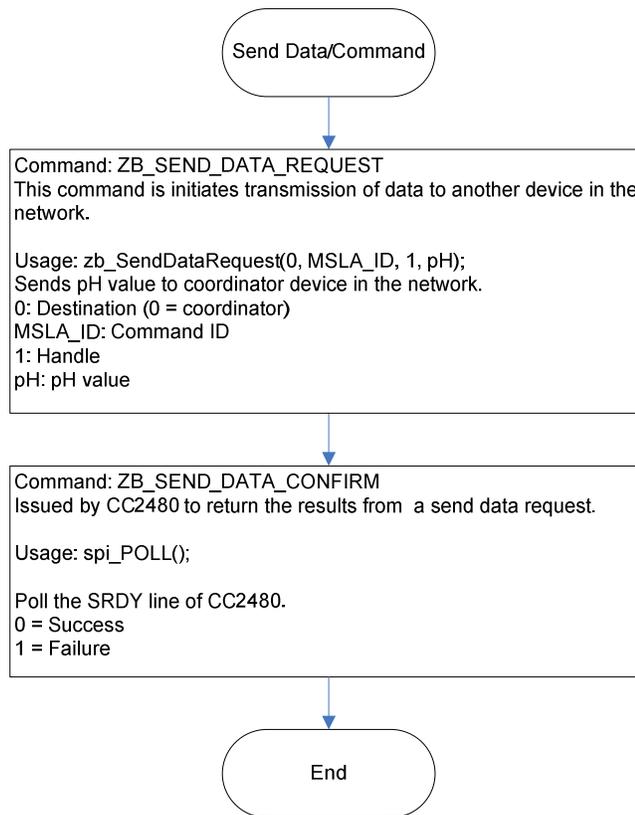


Figure 8.2: Using Simple API to send data/command.

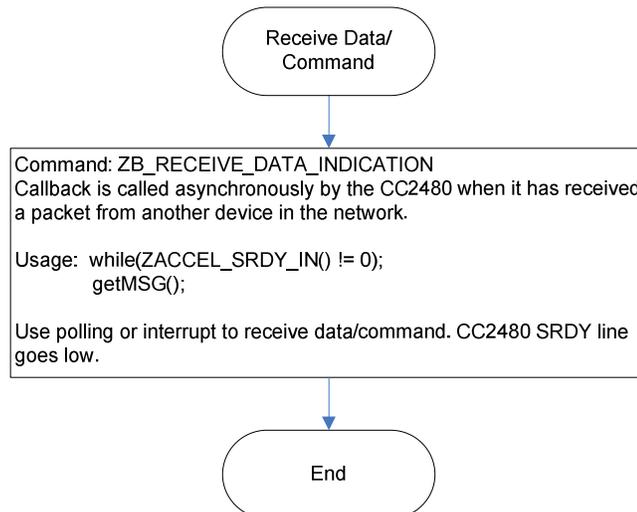


Figure 8.3: Polling method to receive data/command.