

University of Southern Queensland
Faculty of Engineering and Surveying

**MEASURING WITH BEES II –
IN HOME POWER AND WATER MONITORING**

A dissertation submitted by
Yian Yian Tan
Student Number: 0050039354

In fulfilment of the requirements of
Courses ENG4111 and 4112 Research Project

Towards the degree of
Bachelor of Engineering
(Electrical and Electronic Engineering)

October, 2011

Abstract

Wireless Sensor Network (WSN) has been developing rapidly due to its many features, such as low power consumption, node network theory and suitability in large and harsh environment. The areas of application and implementation of WSN has been growing, and ZigBee is one of the WSN specifications commonly utilised. The ZigBee specification is developed by ZigBee Alliance, which covers various ZigBee implementations. It provides a low power, cost effective, reliable, WSN standard, which is suitable in radio frequency application that required low data rate and secure networking.

One of the applications for ZigBee specification is household energy monitoring, and it is commonly accomplished by using smart meter which transmit the captured digital data in the ZigBee network. However, the board usage of analog rolling number type of utilities meters at the residential unit, the energy usage information from this type of meter cannot be easily adapted to the ZigBee network.

This dissertation covers a research project aimed to develop a home power and water monitoring system, which adapts to the analog metering infrastructure and communicates wirelessly in a ZigBee network. The purpose of this project was to make the energy usage information accessible through web interface.

The project is achieved by capturing the analog metering information with a camera module attached to the ZigBee modem at the remote sensor node, and the consumption information, which is the data of the picture file, is transmitted wirelessly in the ZigBee network formed by a ZigBee application kit. A Rabbit processor, which is part of the application kit, accomplished the web interface function of the project. It is linked up with TCP/IP connection, and programmed to generate the web page, which provided the activation of image capturing of the utilities meter in the monitoring system.

University of Southern Queensland

Faculty of Engineering and Surveying

ENG4111 Research Project Part 1 &
ENG4112 Research Project Part 2

Limitations of Use

The Council of the University of Southern Queensland, its Faculty of Engineering and Surveying, and the staff of the University of Southern Queensland, do not accept any responsibility for the truth, accuracy or completeness of material contained within or associated with this dissertation.

Persons using all or any part of this material do so at their own risk, and not at the risk of the Council of the University of Southern Queensland, its Faculty of Engineering and Surveying, or the staff of the University of Southern Queensland.

This dissertation reports an educational exercise and has no purpose or validity beyond this exercise. The sole purpose of the course “Project and Dissertation” is to contribute to the overall education within the student’s chosen degree programme. This document, the associated hardware, software, drawings and other material set out in the associated appendices should not be used for any other purpose: if they are so used, it is entirely at the risk of the user.



Professor Frank Bullen
Dean
Faculty of Engineering and Surveying

Certification of Dissertation

I certify that the ideas, designs and experimental work, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.

I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

Tan Yian Yian

Student Number: 0050039354

Signature

Date

Acknowledgements

The undertaking of this project had been one of the most academic challenges that I had encountered. This project together with the dissertation would not be possible without the support from my Supervisor Dr Alexander Kist and my family members. I am grateful to have them around me to overcome this challenge.

I would like to thank Dr Alexander Kist, my Project Supervisor, for providing suggestions and guidance, and encouraging me when I faced with a bottleneck situation during the development of the project.

I would like to thank my family members for their understanding that I was unable to spend much time with them, and for their patience in lending me a listening ear when I hit with difficult problems on the project.

Tan Yian Yian

University of Southern Queensland

Oct 2011

Table of Contents

Abstract	i
Certification of Dissertation	iii
Acknowledgements	iv
Table of Contents	v
List of Tables	viii
List of Figures	ix
List of Appendices	x
Abbreviations	xi
Chapter 1. Introduction	1
1.1 Overview of the Project	2
1.2 Project Methodology	4
1.2.1 Overall Project Block Diagram	4
1.2.2 Capturing of Power and Water Meter Data	5
1.2.3 Transmitting Data through ZigBee	5
1.2.4 Displaying information on a Web server.....	6
1.3 Overview of the Dissertation	7
1.4 Chapter Summary	7
Chapter 2. Home Utilities Metering	8
2.1 Power Meter	10
2.2 Water Meter	11
2.3 Metering Development.....	13
2.4 Chapter Summary	14

Chapter 3. Wireless Sensor Network.....	15
3.1 Standards	16
3.2 ZigBee.....	17
3.3 Applications of ZigBee Specification	19
3.4 Chapter Summary	21
Chapter 4. Hardware Development.....	22
4.1 Camera Module.....	22
4.2 ZigBee Application Kit	25
4.3 Miscellaneous Hardware.....	28
4.4 Hardware Connection	29
4.4.1 <i>Connection between Rabbit Processor and ZigBee Modem</i>	<i>29</i>
4.4.2 <i>Connection between Camera Module and ZigBee Modem</i>	<i>30</i>
4.5 Chapter Summary	31
Chapter 5. Software Development.....	32
5.1 Camera Module Programming.....	33
5.2 ZigBee Modem Programming	37
5.3 Web Server Programming.....	38
5.4 Web Page Interface.....	40
5.4.1 <i>Web Page for Front Page Display</i>	<i>40</i>
5.4.2 <i>Web Interface for Image Capturing Activation</i>	<i>43</i>
5.4.3 <i>Web Page for Metering Information Display</i>	<i>43</i>
5.5 Rabbit Module Programming.....	45
5.6 Chapter Summary	46

Measuring with Bees II

– In Home Power and Water Metering

Chapter 6. Conclusion	47
6.1 Achievement of Project Objectives.....	49
6.2 Future Enhancement.....	50
6.2.1 <i>ZigBee Network</i>	51
6.2.2 <i>Image Processing</i>	51
List of References	52

List of Tables

Table 1: Pins Connection for RF interface board to Camera Module	30
Table 2: Pins Connection for the Camera Module and the ZigBee Modem	30
Table 3: Commands for the u-Cam Camera Module (2010, pg 8)	34
Table 4 : ZigBee Modem Configurations and Settings.....	37
Table 5 : MIME type Mapping Table	42

List of Figures

Figure 1: ZigBee Network for Measuring with Bees II	3
Figure 2: Block Diagram of the Project	4
Figure 3: Different Types of Metering Display	8
Figure 4 : Population of Electricity Meters and Annual Demand by Region (2006, p26)	9
Figure 5 : Population of Water Meters and Annual Demand by Region (2006, p29)	9
Figure 6: Electromechanical Electricity Meter in Singapore	11
Figure 7: Displacement Water Meter utilised in Singapore.....	12
Figure 8: ZigBee Network Layer Diagram.....	18
Figure 9: The 4D System Serial JPEG Camera Module	23
Figure 10: Connection for Camera Module to Computer	24
Figure 11 : Pins Configuration from Rabbit Processor to ZigBee Modem	29
Figure 12: Message Sequence Chart for Camera Module Synchronisation Process (2010, pg 14)	34
Figure 13: Block Diagram of a Data Package from the Camera Module (2010, pg 10)	35
Figure 14: Message Sequence Chart for Snapshot Picture Activation on the Camera Module (2010, pg 16)	36
Figure 15 : The Front Page for Web Page Interface.....	40

List of Appendices

Appendix A - Project Specification	55
Appendix B - Rabbit Module Programming Codes	57
Appendix C - Web Interface Codes	75
Appendix D - Schematic Diagram.....	77
Appendix D.1 - RCM3720 Prototyping Board Schematic Diagram	78
Appendix D.2 - RF Module Interface Board Schematic Diagram	79

Abbreviations

API – Application Interface Programming

CGI – Common Gateway Interface

CODEC – Coder - Decoder

CRC – Cyclic Redundancy Check

CSMA-CA – Carrier Sense Multiple Access Collision Avoidance

DSSS – Direct Sequence Spread Spectrum

GIF – Graphics Interchange Format

HTTP – Hypertext Transfer Protocol

HTML - Hypertext Markup Language

IEEE – Institute of Electrical and Electronics Engineers

IP – Internet Protocol

JPEG – Joint Photographic Experts Group

LAN – Local Area Network

LED – Light Emitting Diode

MAC – Media Access Control

MIME - Multipurpose Internet Mail Extensions

OCR - Optical Character Recognition

O-QPSK – Offset – Quadrature Phase-Shift Keying

QPSK – Quadrature Phase-Shift Keying

RAM – Random Access Memory

RF – Radio Frequency

RF4CE – Radio Frequency for Consumer Electronics

RS232 – Recommended Standard 232

TCP – Transmission Control Protocol

TTL – Time To Live

USB – Universal Serial Bus

WSN – Wireless Sensor Network

WWW – World Wide Web

Chapter 1.

Introduction

With the rapid development in the computing industries in the last 20 years, the usage of computer technology has adapted into many different industries, such as manufacturing industries, household product industries, control and monitoring industries, etc. The implementation of computing technology in the control and monitoring industries enhanced the desire of automated remote control and monitoring system. An automated remote control and monitoring system allows user to have access on the condition and environment of the remote area without the need to access the physical area. It reduces the labour requirement in the control and monitoring industries, furthermore it increases the safety of the job by reducing the frequent access on the hazardous environment. The growth on the automated remote control and monitoring system encouraged the development of wireless sensor network.

Wireless Sensor Network (WSN) is a communication network consists of many individual sensor nodes which connect together wirelessly. One of the characteristics of the WSN, which deployable to large scale environment with automated and unattended operation, enforces its suitability in the applications for the automated remote monitoring system.

ZigBee is one of the WSN specifications developed by ZigBee Alliance. It is a low power, cost effective, reliable wireless networking specification which adapts well with the WSN requirements. ZigBee Alliance is an industry association of over 200 major electronic, semiconductor and network equipment companies. It governs a set of ZigBee standards which cover wide scope of implementation, such as ZigBee Smart Energy, ZigBee Home Automation, etc. Therefore, ZigBee WSN specification is a common choice of standards for wireless sensor network application.

With the increasing emphasis on energy conservation, the implementation of wireless sensor network in energy usage monitoring has been developing rapidly. The implementation caters for the large scale industries, at the same time; it can also be implemented in normal household. The analog rolling number type of utilities meters are commonly installed at the residential units and the metering information can only be accessed by carrying out manual recording from the meters on site. Therefore, the application of the wireless sensor network at the household energy monitoring allows user to gain information on the energy usage remotely, and also reduces the labour requirement on meter recording.

1.1 Overview of the Project

“Measuring with Bees II – In Home Power and Water Monitoring” – A project developing a wireless sensor network using the ZigBee protocol, which allows the power and water metering information to transmit wirelessly and makes the information accessible via a web interface.

The aim for the project is to develop a power and water monitoring system in normal household, which allows user to monitor the utilities consumption on an internet interface. This project provides a mean for energy conservation through monitoring; it also acts as a remote monitoring system which reduces the manual labour required for recording on the metering information on site.

This In-home power and water monitoring system utilises a wireless sensor network with individual sensor node, for capturing the utilities metering information and communicating through the ZigBee specification. The ZigBee network allows multiple sensor nodes connected together, thus the utilities metering information was captured and communicated in the network concurrently. The network is connected to the computer through the Rabbit processor, which acts as the web server and processing unit. The ZigBee modem connected to the Rabbit processor operates as the coordinator for the network. Figure 1 illustrates the ZigBee network for the project.

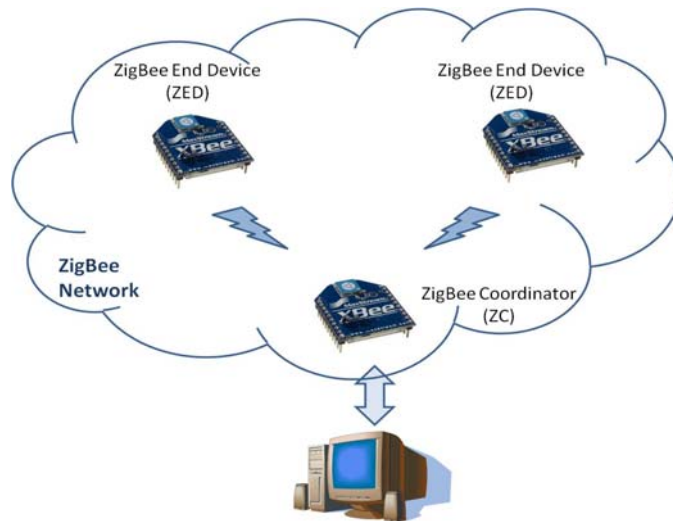


Figure 1: ZigBee Network for Measuring with Bees II

The objectives of the project included the background research of home power and water metering system, and identified the current metering methods and the different types of meters available. The wireless sensor network protocols, in particular ZigBee, are studied for understanding of the wireless sensor network and comparing the advantages and features of ZigBee to other wireless sensor network protocols.

Suitable methods of capturing power and water usage information were determined with the ease in adaptability on common home usage. This project evaluated and selected suitable power and water sensors for home monitoring, thus the metering information could be captured and transmitted through the wireless sensor node. The monitoring unit was developed and built to ensure individual selected hardware can be integrated and functioned together.

The possibilities for making the information available on World Wide Web (www) were investigated to facilitate the development on the web interface program. An interface program which allows sensors to communicate through ZigBee was created, which is capable in data capturing of the power and water meters information in the ZigBee network. This interface program carried out the required task of capturing the metering information and communicating the information to the main sensor node at the computer side wirelessly. Lastly, a web interface program was developed to display the power and water monitoring information on internet.

1.2 Project Methodology

The project was divided into 3 major parts; the capturing of the water and power meter information, the transmitting of the meter data through ZigBee, and the displaying of information on a web server for access on World Wide Web.

The theories and methods used for each part are elaborated in the following sub sections together with the brief explanation on its function and sub function. The overall block diagram of the project is also shown.

1.2.1 Overall Project Block Diagram

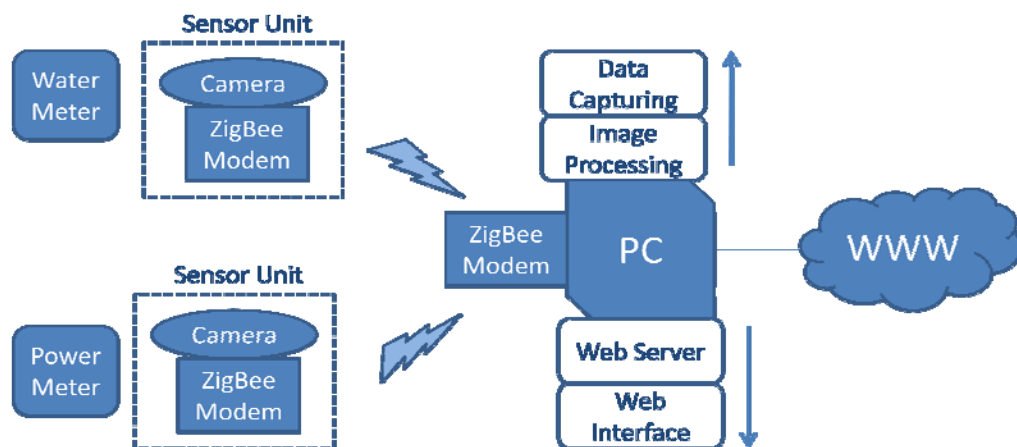


Figure 2: Block Diagram of the Project

The block diagram shown in Figure 2 illustrates how the different parts of the project collaborate with each other, and it also highlights the sub functions of each device and parts.

The data capturing of the metering data was carried out by a camera at the meter side. The camera linked up with the ZigBee modem for communication, and both of them formed the sensor unit. The ZigBee modems were used for communicating the metering information through the ZigBee network, and the information was passed to the computer which holds the web server. The computer handled the web interface and data processing functions.

1.2.2 Capturing of Power and Water Meter Data

With the research and study carried out on the theory and measuring methods for power and water meter, it has concluded that it would be difficult for this project to restructure and develop the power and water meter that able to capture the metering data digitally for transmission through ZigBee. Furthermore, a newly built meter would be hard to adapt and implement into current power and water distribution infrastructure system. Therefore, the methodology for capturing the metering data was by using a camera to take a snapshot of the metering display on the meter, and followed by data-processing on the picture to record and display the actual value of the power and water usage.

This project utilised a small portable camera to capture a picture of the metering display on the power and water meters. The commands for activation of the camera were initialled from the interface program, transmitted through the ZigBee network, and received at the camera connected to the remote ZigBee sensor node. The data of the captured picture file reversed the transmission path, and transmitted wirelessly through the ZigBee network, and received at the computer server side for processing and display on the web interface. The usage of camera for data capturing solved the problem with adaptability on the current power and water infrastructure, and it also provided an easier implementation of the monitoring system on different type of metering devices.

1.2.3 Transmitting Data through ZigBee

Under the review of wireless sensor network for the project development, it has shown that ZigBee was the suitable protocol for the wireless sensor network (WSN), as it fulfilled most of the requirements as a protocol for WSN. Furthermore, ZigBee, the wireless sensor network specification, also proved as a good selection of communication protocol for this project as it matched the requirements of the project.

As the camera and the sensor node modem only turned on when activated and requested, the sensor nodes of the wireless sensor network for this project required a low power usage and long battery life. Concurrently, the network for the project did not require a high data rate transmission, as the data transmitting across the network was only the data packets of the picture file and the commands to activate the camera function. Therefore, it made ZigBee a suitable communication protocol for the project. A ZigBee Application Kit, which consists of the ZigBee modems together with a Rabbit processor as the controller, was utilised in the project for the ease of development of the ZigBee network and the interface program for metering data capturing.

1.2.4 Displaying information on a Web server

The functions of the web server are to communicate with the ZigBee base modem and carry out the data processing of the system. The computer was connected to the ZigBee base modem through a Rabbit processor which has a TCP/IP connection. The Rabbit processor consists of the instructions and commands to activate the camera for picture capturing, and it also linked up to the computer with its TCP/IP connection for display of interface program and activation of data capturing through HTML page. The TCP/IP connection between the rabbit processor and the computer transforms the computer as part of the web server which allowed the interface program to be accessible on World Wide Web (WWW).

The data of the captured picture file located at the ZigBee base modem was recommended for transmitting through a TCP socket to the computer for processing and displaying. The programming at the Rabbit processor allowed individual TCP socket to be open for transmitting data of the picture file. Java programming was considered for generating a Java applet, which enables picture data to be read at the respective TCP socket, and converts it into a jpeg file. Java programming code was selected for this project application because it provides functions for communication over TCP/IP, image input/output, and bytes array conversion.

1.3 Overview of the Dissertation

This dissertation documents the research, analysis, development and implementation of a home power and water monitoring system, which communicates wirelessly, and the information is accessible via a web interface.

Chapter 1 of the dissertation covers the introduction of the project. It provides an overview of the project, and annotates the project methodology. It offers reader a brief idea and understanding of the project.

Chapter 2 covers the research on the home utilities metering. It highlights the technologies and methods of various utilities meter, such as power and water meters.

Chapter 3 covers the research on the Wireless Sensor Network (WSN). It comprises the various WSN standards, and a more in depth understanding on the ZigBee specification, as it is the communication protocol used in the project. The various applications of the WSN are briefly discussed.

Chapter 4 covers the hardware development of the project. The details of the hardware, which include of the camera module, the ZigBee application kit and the computer, are described. The connection between the hardware is illustrated and explained.

Chapter 5 covers the software development of the project. The programming for the various hardware, such as camera module, ZigBee modems, Web server, and Rabbit processor, was explained in detail.

The dissertation concludes on Chapter 6 which consists of the overall achievement of the project and the recommendation for future enhancements.

1.4 Chapter Summary

This chapter provided an introduction of the project, “Measuring with Bee II – In home power and water monitoring system”. It gave a general overview on the aims and objectives as well as the hardware and software utilised.

Chapter 2.

Home Utilities Metering

Utilities metering devices are used to capture the utilities consumption and the common measurement of utilities are electricity, water and gas. The utilities metering devices are mainly used by the utilities supply company to determine the amount of usage and the cost of consumption accordingly. Beside it being used for calculating the usage cost, the utilities meters could also provides a mean to monitor and control the consumption rate, which fulfil the energy conservation purpose.

The utilities meters are generally used in many industries. For instance, the meters in industrial section are deployed to individual plant or machine to monitor its consumption, and in household, the meters are deployed to individual household for the calculation on the cost of usage.

Different types of utilities meter use different mechanisms to measure its value. The detail of the theory and mechanism of the power and water meter are covered in the following sub sections. There are also various types of display on the meter for the measured amount, such as analog needle displacement, analog rolling, and digital number. The analog rolling number is the most common type of display used on the meter currently. Some of the examples of the display type are shown in Figure 3.

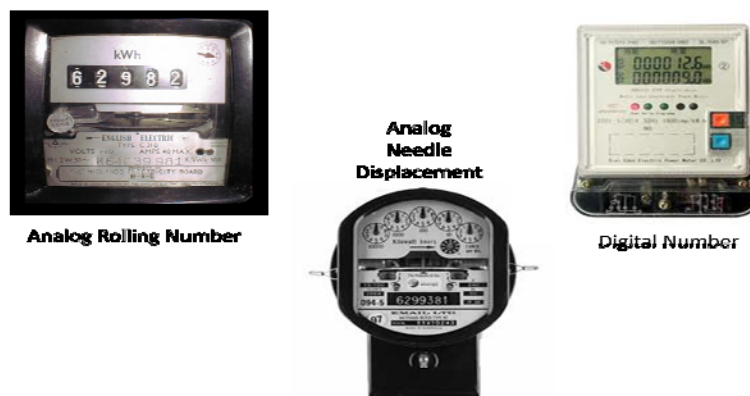


Figure 3: Different Types of Metering Display

– In Home Power and Water Metering

Multi-utility meter report (2006) by ABS Energy research gave a brief overview of the global metering market. It reported on the different type of meters and market trends for different regions in the global market. The report highlighted on the study for the demand and population of the utilities meter, and the population and demand for electricity meter and water meter is shown on Figure 4 and Figure 5 respectively. From the figures, it shown that Asia market made up the largest population and demand for the utilities meters.

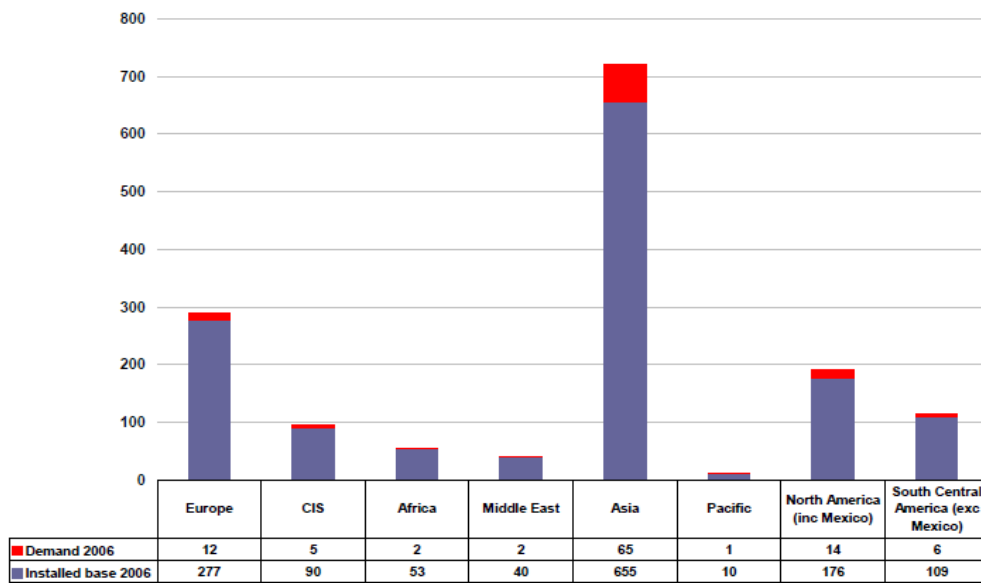


Figure 4 : Population of Electricity Meters and Annual Demand by Region (2006, p26)

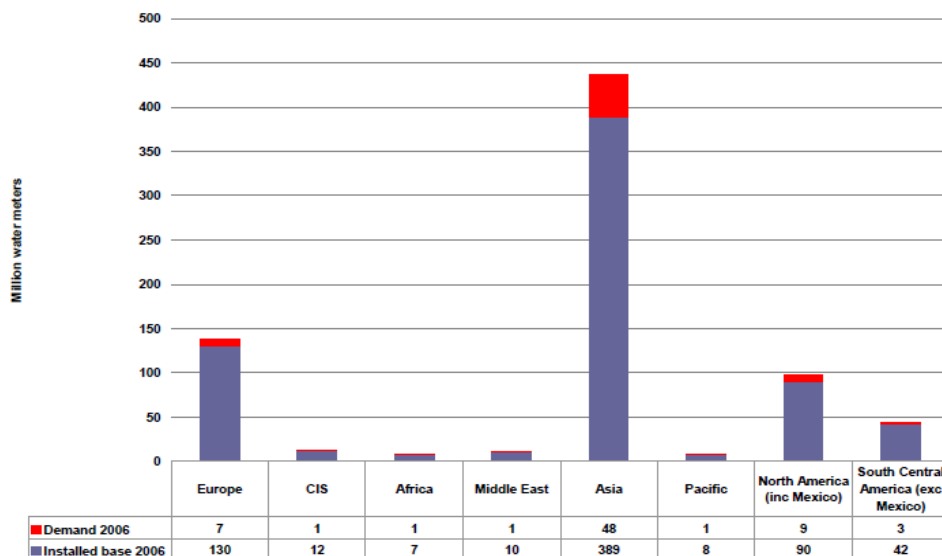


Figure 5 : Population of Water Meters and Annual Demand by Region (2006, p29)

2.1 Power Meter

A power meter, also known as an electricity meter, is used to measure the amount of electricity that passed through it. The meter is connected at the main electricity input point of individual residential or business unit to measure the amount of electricity consumption of the establishment. Most of the household are supplied with a single phase AC current, and the most common unit of measurement for the meter is kilowatt-hour (kWh). Kilowatt-hour is calculated by the formula as $kWh = \frac{Watts \times Time(Hrs)}{1000}$, where it is the general unit used by the power supply company for calculating the cost of electricity consumption with cost per kWh (\$/kWh).

The main measuring mechanism of the meter is continuously measuring the instantaneous voltage and current passing through the meter, and the product of the instantaneous values determines the electrical power (watts). The electrical power value (watts) is integrated against time to give the electricity energy used, which is in the unit of kilowatt-hour. There are 2 basic types of meters which is electromechanical meters and electronic meters. Electronic meter is a newer development on the metering, which capture the consumption value digitally and allow the measured value to be processed. The detail of the electronic meter is covered under the metering development section.

Electromechanical meter utilises the induction theory of the magnetic flux. The meter determines the amount of electricity usage by counting the number of revolutions of an aluminium disc, which is adjusted to rotate at a speed proportional to the power. The aluminium disc is controlled by the magnetic flux generated from 2 coils, voltage and current coils, where their magnetic fluxes are proportionally to the current and voltage respectively. A lag coil on the voltage coil, together with the permanent magnet creating opposite force on the aluminium disc, the disc is made to rotate at a speed proportional to the power being used. This type of meter is commonly used as domestic electricity meter. An example of the electromechanical meter utilised in Singapore's household is shown in Figure 6.



Figure 6: Electromechanical Electricity Meter in Singapore

2.2 Water Meter

A water meter is used to measure the volume of water usage passed through it. Similarly for the electricity meter, it is connected at the main water entry valve for individual residential or business unit to measure the amount of water consumption of the establishment. The common units of measurement for the water meter are cubic feet, cubic meters, and gallons, depends on the country of installation. The unit of measurement used in Singapore is cubic meters, and the charges is based on cost per metre cubic ($\$/\text{m}^3$).

There are many types of water meter around, such as displacement water meter, velocity water meter, electromagnetic meter and ultrasonic meter. Each comes with different mechanisms to measure the amount of water passing through it. There are various types of display on the meter to indicate the amount of usage or volume flowing through the meter. The display types are mainly analog rolling number or digital number, which depend on how the volume flowing through is recorded on a mechanical or electronic register respectively.

Measuring with Bees II

– In Home Power and Water Metering

Displacement water meter is the common type of water meter used in residential unit, and it generally provides an accurate measurement for low and moderate flow rate. It employs a moving measuring element which the physical movement is related to the amount of water passed through the meter. There are 2 common methods for the displacement measurement, oscillating piston and nutating disk. Both methods depend on the water movement to physically move the measuring element to cause a displacement in the direct relation to the amount of water passing through the meter. The piston or disk activates a magnet which drives the register to account for the total volume passed through. An example of the displacement water meter utilised in Singapore's household is shown in Figure 7.



Figure 7: Displacement Water Meter utilised in Singapore

2.3 Metering Development

One of the metering developments was the increasing demand of water meter. As listed in the *Multi-utility meter report* (2006) that water meters are commonly utilised in most household for the Asia region, while the population of the water meter is less common in other large landscape countries, such as Europe, as previously there is no charges for the water consumption. With the study carried out on the effect on water consumption with and without metering reported under *Motivations for Metering*, Magazine of the International Water Association, Lis Stedman (2006) stated that there is an instant drop of 10% of consumption when meter being installed at UK. The report also shown that in Germany, the domestic water consumption for metered apartments (112 litre/capita/day) was 18% lower than for unmetered apartments (137 litre/capita/day) in 1992.

Therefore, the implementation and installation of the water meters at these large landscape countries had been growing. A report (2009) by Ofwat (Water Services Regulation Authority of UK) web site stated that although there is only 40% of household installed with a water meter in UK, but the number had been increasing annually.

Another major metering development was the introduction of Smart meter. With the advanced development on the electronics technology, this new form of utilities meter, which also known as digitised meter, had been expanding its implementation in the metering industry. The smart meter converts the measured value of the utilities into digital format and display the information with digital display.

Besides displaying the consumption values in digital format, the smart meter also provides additional features as compared to the analog meter. It allows the usage data to be processed at the meter or easily transmitted to other processing unit, thus offering a real time or near real-time value indication, instant calculation of the utilities cost, utilities quality monitoring, and outage notification. It also provides a mean to capture the utilities consumption at different time of the day and seasons, so the supply company able to set different tariff for different time period.

There are also various developments on Smart metering with ZigBee wireless communication network, and Luan S, Teng J, Chan S & Hwang (2009) had demonstrated an example with their development of a smart power meter using application message interface on the ZigBee network.

Although with the advanced features provided by the Smart meter, the rate of implementation was still hindered by two main reasons. The first reason was the large amount of household or residential unit is still using the analog meter, which make it difficult for the smart meter to be installed at the current utilities infrastructure. The second reason was the high cost of implementation, which included the cost of the smart meter and the cost of installing the smart meter in the utilities infrastructure.

Alternatively, the development and implementation of the Smart meter was still encouraged by many companies and government sectors due to its long term benefits with reduce labour on metering record, and the energy conservation. The followings are some of the examples:

- Google, as reported by Matthew Wald and Miguel Helft (2009), was encouraging the development of smart meter by providing web interface, known as PowerMeter to link up with the smart meters to provide electricity monitoring.
- A news announced by United Kingdom's Department of Energy and Climate Change (2009) stated on its intention to have smart meters installed in all household by 2020.

2.4 Chapter Summary

This chapter covered the basic of home utilities metering. The measuring mechanisms and display types of the common power and water meter were illustrated. This chapter also identified some of latest metering development, such as Smart meter. With the research carried out for home utilities metering covered under this chapter, it allowed the project to identify and determine the suitable methods for capturing power and water usage information and the adaptability on current metering infrastructure.

Chapter 3.

Wireless Sensor Network

Wireless Sensor Network (WSN) is a communication network that has many individual sensor nodes which connected together wirelessly. Each sensor node consists of 4 main parts, which are sensor, processor, power supply and transceiver. The sensor is a small electronic device that is able to monitor the surrounding environment and condition. The sensor converts the information into data format using the processor, and communicates the information in the network using the transceiver. The power supply provides the power required for the sensor, processor and transceiver.

WSN utilises the node network theory, thus it can cope with node failures, allows nodes mobility and has dynamic topology. As the WSN normally activates data collection from its nodes through the base station at a fixed interval, the nodes do not need to be in power-on state most of the time. On top of that, with the advancement in technology, small sensor node size can be produced; therefore, WSN has the characteristic of low power consumption.

In addition, WSN also has the characteristics of deployable to large scale environment with automated and unattended operation at the sensor nodes. Together with the node ability to function in harsh environment condition and ease in implementation, the development of WSN in the industrial and environmental industry is growing; as it is suitable in the large coverage area network and with rough environmental condition.

The base station of a WSN is the main controlling and processing part of the whole network. It has higher computing power and requires more power. It acts as a gateway for the network to transfer the data from the sensor nodes to the end user's program, such as a server. It also carries out the routing calculation for the network.

3.1 Standards

There were different standards defining the physical layer of the WSN, which includes the operating frequency and the modulation scheme. Chris Townsend & Steven Arms (2004) stated in the Wireless Sensor Networks: Principles and Applications document, these standards include of IEEE 802.11X, Bluetooth, IEEE1451.5, IEEE 802.15.4 and ZigBee. Each of the standards has its own features and suitability for WSN and a brief description on these standards are listed below.

IEEE 802.11X has a high data rate around 1Mbps to 50Mbps, and the transmission range is around 300 feet from the antenna. Due to high data rate, it has a higher power requirement.

Bluetooth requires less power than IEEE 802.11X standard, and it uses the star network topology. Bluetooth based on IEEE 802.15.1 standard with additional specification on the application protocols. However, it is not widely utilised in WSN due to comparable higher power for short transmission distance, lower number of nodes, longer duration to get out from sleep mode, and a complex Media Access Controller (MAC) layer.

IEEE802.15.4 is designed from the requirement of WSN applications. It is a flexible standard with specified multiple data rates and transmission frequencies. It has relatively low power consumption as it allows the node to go into sleep mode and has fast synchronization from sleep mode. It supports star and mesh network topology. It is a widely accepted WSN standard.

IEEE1451.5 aims to standardize the interface of sensors to a WSN and to enhance the standard, IEEE802.15.4, which does not specify the sensor interface.

ZigBee is a standard which specifies its physical and Media Access Control (MAC) layer from IEEE 802.15.4, and it comes with its own standardisation on the higher level applications. ZigBee is the network standard exploited in this project and the detail on ZigBee is covered in the next section.

3.2 ZigBee

ZigBee is a wireless sensor network specification developed by ZigBee Alliance, which is an industry association of semiconductor and network equipment companies such as Honeywell, Mitsubishi Electric, Motorola, Samsung and Philips. The ZigBee Alliance was formed in 1997 and with over 200 member companies currently. Its aim is to develop, maintain and control the ZigBee specifications, ZigBee and ZigBee RF4CE, and standards, such as ZigBee remote control, ZigBee smart energy, ZigBee home automation etc. The detail information on the ZigBee Alliance is available on its web site <<http://www.zigbee.org/>>

ZigBee is a low power, cost effective, reliable, wireless mesh networking standard, which adapts well with the WSN requirements. As the ZigBee modem can be activated from sleep mode in a very short duration, the modem able to stay in sleep mode most of the time to have low power consumption, while still retains the system responsive. This low power usage feature allows the sensor unit to have longer life usage with a smaller battery. Due to low cost of the ZigBee devices, ZigBee is a cost effective means to deploy widely in wireless control and monitoring applications. The mesh networking employed by ZigBee allows its maximum range to be extended through multi-hop. Ad-hoc formation of network, automatic route discovery and self healing are some of the features provided by mesh networking. ZigBee is mainly suitable for radio frequency application that requires long battery life, low data rate and a secure networking.

Some of the technical characteristics of ZigBee specification are:

- Frequency bands: 868MHz, 915MHz and 2.4GHz
- Data transfer rate up to 250Kbps
- Signal transmission range of 10 to 100m
- Advanced Encryption Standard (AES) on data transmission
- Able to work with different ZigBee standards
- Low power usage
- High reliability

– In Home Power and Water Metering

The high reliability of the ZigBee specification is achieved by the following features of its communication protocol:

- IEEE 802.15.4, a very low range wireless technology, with QPSK (Quadrature Phase-Shift Keying), O-QPSK (Offset-QPSK) and DSSS (Direct Sequence Spread Spectrum)
- CSMA-CA (Carrier Sense Multiple Access Collision Avoidance)
- 16-bit CRC (Cyclic Redundancy Check)
- Acknowledgments at each hop
- Using mesh networking to define the most reliable route
- End-to-end acknowledgments to verify data arrived at its destination, with each packet re-transmit up to 3 times.

As stated under the WSN standard section, ZigBee based its physical and MAC layer from IEEE802.15.4, and it functions as a compliance arm to IEEE 802.15.4 specification. This is same as how Wi-Fi Alliance works with IEEE 802.11 specification. ZigBee alliance expanded the ZigBee specification on the network specification and application interface. The network layer diagram for ZigBee is as shown in Figure 8.

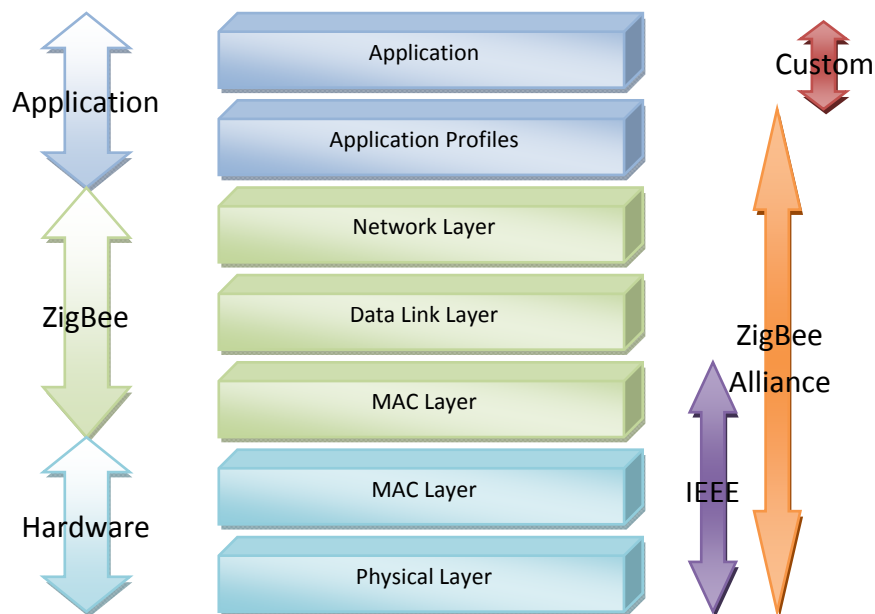


Figure 8: ZigBee Network Layer Diagram

The application profiles defined in ZigBee are used to define the application level compatibility. It describes how the different types of application objects connect and work together, and it also specifies a test suite for verifying compatibility among applications from various vendors.

There are 3 basic types of ZigBee devices, which are Coordinator (ZC), Router (ZR) and End Device (ZED).

- ZigBee Coordinator (ZC) is the main brain of the whole network, and it has the most processing power. There is only one coordinator in each network, and it is the only device that can form a network. It controls and defines routing information, and stores information about the network.
- ZigBee Router (ZR) is used to pass on data from other devices, at the same time; it also runs the application function.
- ZigBee End Device (ZED) is the functional nodes of the ZigBee network. It does not relay data from other devices, thus it able to still in sleep mode most of the time to reduce the power consumption amount. It has only sufficient functionality for communicating with the ZR or ZC.

3.3 Applications of ZigBee Specification

ZigBee Alliance specified the various ZigBee standards, such as ZigBee building automation, ZigBee health care, to cover the various applications implemented with ZigBee specification.

One of recent major development for ZigBee Alliance is the agreement (2009) with RF4CE (Radio Frequency for Consumer Electronics) Consortium to jointly deliver a standard specification for radio frequency-based remote controls. This introduces a new ZigBee specification, known as ZigBee RF4CE, which its application covered a wide range of remote-controlled audio/visual consumer electronic products, such as TVs, recording system and home entertainment system. ZigBee RF4CE provides better communication, increased reliability, enhanced features and flexibility, interoperability and no line-of-sight barrier. It promises many advantages as compared to the existing remote control solutions, such as infra-red, blue-tooth.

Another application of the ZigBee specification which related to the development of this project is the ZigBee Smart Energy standard. As defined on the ZigBee Alliance web site, “ZigBee Smart Energy is the world's leading standard for interoperable products that monitor, control, inform and automate the delivery and use of energy and water.” It develops to allow consumer to have access to the energy usage information and automation control, thus to reduce energy consumption, improve energy conservation and save money. There are many products available currently developed from the ZigBee smart energy standard, such as Smart Meter incorporated with ZigBee wireless communication. However, the use of smart meter was not recommended under this project due to difficulty in implementation on current metering infrastructure.

The recent development and research on applicability of the ZigBee network with the monitoring and control system was cited in a number of IEEE papers.

The followings are some of the example:

- Li Pengfei, Li JiaKun and Jing Junfeng (2010) discussed the technical characteristics, network topology structures and application prospects of ZigBee, and its implementation to the temperature monitoring system during Computer Engineering and Technology International Conference, with their paper title as Wireless Temperature Monitoring System based on the ZigBee Technology.
- Jinsoo Han, Chang-Sic Choi and Ilwoo Lee (2011) described the development of a more efficient home energy management system with the implementation of ZigBee network and infrared system on their IEEE paper, More Efficient Home Energy Management System based on ZigBee Communication and Infrared Remote Controls.
- Gill K, Shuang-Hua Yang, Fang Yao and Xin Lu (2009) evaluated the potential of ZigBee network in designing and implementing flexible home automation architecture on their IEEE paper, a ZigBee-based Home Automation System.

– In Home Power and Water Metering

- Zhou Yimin, Yang XiangLong, Guo Xishan, Zhou Mingang & Wang Liren (2007) wrote an IEEE paper, A Design of Greenhouse Monitoring & Control System based on ZigBee Wireless Sensor Network, covered the explorative application of ZigBee Wireless sensor network in protected agriculture and its applied design for greenhouse management.

3.4 Chapter Summary

This chapter covered the basic information on the wireless sensor network. It gave a brief overview on the Wireless Sensor Network (WSN) and the various standards defining the WSN. As the project was developed with the ZigBee specification for the Wireless Sensor Network, a more detail explanation on the ZigBee specification was covered. The development on various ZigBee projects and applications were researched and listed in the chapter to provide further understanding on the usage of the ZigBee specification.

ZigBee with its features of low data rate and low power consumption make it an ideal WSN specification for this project. Furthermore, with its easy adaptability to other ZigBee standards, it allowed this project to be able to communicate with other ZigBee standards, such as home automation, for future enhancement on the project.

Chapter 4.

Hardware Development

The hardware that formed this project “Measuring with Bees II” consisted of the camera module used for picturing capturing, the ZigBee Application kit, which has the hardware that made up the ZigBee wireless sensor network, and the computer, which acted as the web server. Beside the hardware that formed the final prototype of the project, there was other hardware utilised during the development phase of the project. The detail and functionality of each of them is illustrated in the chapter, and the connections and interfaces between each hardware were also covered.

4.1 Camera Module

The function of the camera module for the project was to take a snapshot on the analog rolling number display of the utilities meters, which is the consumption rate of the utilities. The data of the taken picture file was transmitted through the ZigBee network by formed by ZigBee modems. With the functionality required for the camera module of the project identified, the followings were the requirements when choosing the suitable camera module for this project:

- Required low power (Comparable to ZigBee modem)
- Ease in software configuration and programming
- Ease in hardware set-up
- Able to link up with ZigBee modem for communication

The 4D System uCam-TTL Serial JPEG Camera Module was selected as the camera module which formed part of the hardware for this project. The information and the detail of the camera module are listed in uCam Serial JPEG Camera Module Data Sheet (2010). The selected module is a highly integrated serial camera module with a 32mm x 28mm chip size, and a 4 wire connector. A picture of the camera module is shown in Figure 9.



Figure 9: The 4D System Serial JPEG Camera Module

The 4D u-Cam camera module fulfilled the requirement of the project with its following features:

- Low powered for the TTL model with a 3.3VDC Supply. It matched the voltage requirement of the ZigBee modem, therefore, both the camera module and the ZigBee modem could used the same power supply at the sensor unit of the project.
- Programmable with simple command interface through TTL serial link, with each single command consists of six continuous single byte serial transmissions. The on-board EEPROM of the module offered a simple command based interface to external host via TTL link. This allowed easy programming of the camera module through the Rabbit processor.
- Highly integrated camera module which has on board colour sensor and JPEG CODEC, and all the necessary built-in image processing circuitry, such as colour conversion, down sampling, clamping. Thus, there is no additional hardware requirement and it allowed minimum circuitry design at the sensor node for this project.
- On-board low voltage level TTL serial interface allowed direct connection to the UART or serial port of the microcontroller. This feature of the camera module made it ideal for connection with ZigBee modem UART data in/out pins. The data from the source ZigBee modem IN pin would be transmitting out from the destination ZigBee modem OUT pin to form the communication path in the network.

The camera module came with a dedicated hardware UART chip that provided the serial communication path for commanding and data transmitting for the module. It offered a full-duplex 8 bit data transmission and reception through the TX and RX pins of the module, and the transmitting data was in the format of 8 bits, no parity, and 1 stop bit. The camera module also came with auto baud rates detection features.

The camera module provided various image settings as follows:

- Compressed or Uncompressed picture files, such as RAW or JPEG
- Colour types, such as different bit of RAW gray scale, different bit of RAW colour, JPEG
- Resolutions, from 80 x 60 up to 640 x 480 for both RAW and JPEG

During the development phase of the project, a USB to serial UART converter was required for direct connection of the camera module to the computer, so that the functionality of the camera was verified through the computer, and it also provided understanding to the programming commands of the camera. A 4D system μ USB-MB5 USB to Serial UART Bridge Converter was used to provide direct connection of the camera module to the computer. The connection for the camera module, the convertor and the computer is shown in Figure 10. μ USB-MB5 Datasheet (4D Systems) covers the detail of the converter.

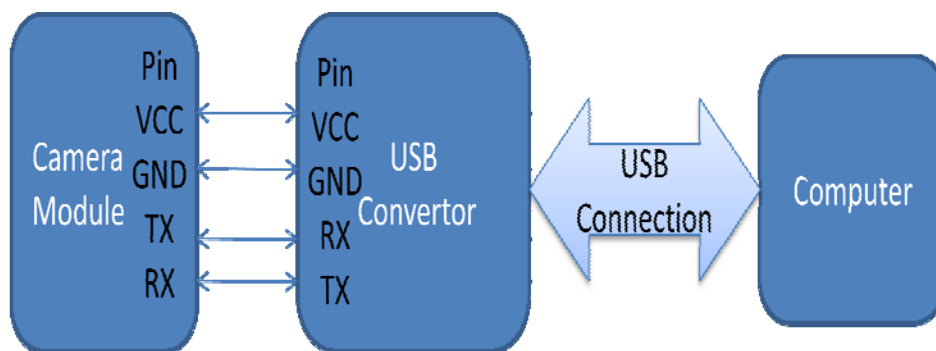


Figure 10: Connection for Camera Module to Computer

4.2 ZigBee Application Kit

The ZigBee Application kit was selected for developing the ZigBee network in the project, instead of individual network components, such as ZigBee modems chip, microcontroller, etc, because the application kit provided easy development on the ZigBee network with the inclusion of programming software, and hardware interface and prototype boards.

Rabbit Semiconductor AN413 ZigBee/802.15.4 application kit was used for the project development, as the application kit able to fulfil the requirement of the project on developing a wireless sensor network communicating in ZigBee. Furthermore, the application kit was also readily available from the development of the previous project “Measuring with Bee”.

The AN413 ZigBee application kit came with:

- Rabbit Processor RCM3720 Rabbit core module with RCM3720 prototype board
- 3 MaxStream XBee™ series 1 RF modems with individual RF interface boards
- Complete Dynamic C software CD and Supplemental CD with ZigBee sample programs and reference
- A 12VDC, 1A AC adapter for supplying the power requirement to the RCM3720 module and the prototype board
- 3 battery holders for 3 AAA batteries, which act as the power supply to the ZigBee modems and its RF interface boards
- A 10-pin header and DB9 connections programming cable together with a level-matching adapter board, which provide the RS232 connection from the Rabbit Processor to computer. It allows the downloading of programs to the Rabbit processor using Dynamic C and monitoring of debugging progress.
- Various cables for connection with the prototype board and the RF interface boards

The AN413 ZigBee/802.15.4 application kit provided the development of ZigBee wireless network integrates with Rabbit-based controller system. There are various implementations for the kit, such as simple remote monitoring, proximity sensor readings, wireless I/O control and data transmission. The AN413 ZigBee/802.15.4 Application Kit Manual (Digi International Inc.) covered the detail of the ZigBee application kit used in this project.

The RCM3720 core module consisted of a Rabbit 3000 microprocessor running at 22.1MHz, 512K bytes of Flash memory, 256K bytes of static RAM, and 1M bytes of serial flash memory. It was a C-programmable core module with 31 configurable for I/O pins and 2 fixed outputs. It could be connected directly to the prototype board, and it also came with Ethernet jack for Ethernet connection.

The RCM3720 prototype board besides providing various circuitries link up with the Rabbit processor for testing and simulating purpose, it also able to link up with one of the supplied ZigBee modem through RS-232 connection. The Rabbit processor communicated direct to the ZigBee modem through one of the serial port, and this ZigBee modem acted as the coordinator modem (ZC) in the network. The schematic diagram of the prototype board is listed on appendix D.1

The RF interface board provided power supply (3.3VDC) to the ZigBee modem through the battery holder. The RF interface board came with an RS232 serial interface with flow control, a connection plug (J4) for direct serial connection to the ZigBee modem chip, and 2 LEDs and 2 switches which connected to various analog/digital Input/output pins of the ZigBee modem. The flow control on the RS232 serial interface was referred to the configuration of the DCE/DTE jumpers on the board, and it allowed user to configure the desired connection for modem communicating to computer or Rabbit processor respectively. The schematic diagram of the RF interface board is listed on appendix D.2.

The 3 MaxStream XBee™ series 1 RF modems (P/N: XB24-ACI-001) were designed to meet the IEEE 802.15.4 and ZigBee standard. It supported the unique needs of Wireless Sensor Network, which is low cost, low power and reliable. It came with a chip antenna with indoor range up to 30m and outdoor range up to 100m. It had pins layout available for analog inputs, digital I/O and

UART data in/out. The UART data in/out pins are the main I/O pins utilised in the project for communicating with the ZigBee modem and commanding the camera module. The detail of wire connection is covered under sub section 4.4 Hardware Connection. The modems that came with the application kit are the first generation (Series 1) ZigBee modems, which only accept AT command, and there was no API command functionality, which means that it did not allow remote network configuration on the modem. However, it did not affect the functionality of the ZigBee network required for this project development, as the individual ZigBee modem was pre-configured before installing in the network and the network configuration on the individual modem is covered under the Software development on Chapter 5. XBee/XBee-PRO OEM RF Modules Product Manual (MaxStream, 2006) covered the technical detail of the ZigBee modem utilised in the project.

The ZigBee network for the project utilised one ZigBee modem at the utility (water/power) meter side acted as the ZigBee End Device (ZED), and another ZigBee modem linked up with the Rabbit processor prototype board and acted as the ZigBee Coordinator (ZC). The communication within the network passed through the UART data in and data out pins of the ZigBee modems and transmitted wirelessly.

The application kit was connected to computer for Dynamic C programming through the RS232 connection with DB9 connector programming cable. With the computer utilised in the project did not have a RS232 port, an additional RS232 to USB convertor was required for this communication path. As stated in the technical manual of the application kit that not all RS232 to USB convertor able to work with Dynamic C software, moreover with the problem of unavailability of the recommended convertor (P/N: 540-0070), thus a correct convertor and driver was identified for this connection after a number of testing and troubleshooting with different convertors and driver software.

4.3 Miscellaneous Hardware

Beside the main hardware of this project that formed up the monitoring system, there were also various other hardware required in the development of the project, such as the computer used and the hardware for installing the sensor unit at the meter site.

A personnel laptop, NEC S3300 with Microsoft Window Vista Home Premium installed, was utilised for the development of the project. The laptop came with CD/DVD drive, USB port and Ethernet port. The main function of the computer was to act as a web server linking up with the Rabbit processor through the Ethernet crossover cable connection. The computer in charged on displaying the web interface, image data capturing and processing. Another function of the computer was to program and debug the Rabbit processor through the USB connection. As the Rabbit processor only has a RS-232 connection for programming, thus a RS232 to USB convertor was utilised for this programming connection. The RS232 to USB convertor was also used for programming the ZigBee modems. Both the ZigBee modem firmware and network configuration were programmed through the X-CTU software from Digi International with the convertor connection. The detail of the modem programming is covered under the subsection 5.2 ZigBee Programming. The CD/DVD drive allowed the installation of the necessary software for the application kit, such as Dynamic C, ZigBee library for Dynamic C and X-CTU.

For the hardware required for installing the sensor unit at the meter site, it was recommended that a plastic casing for holding the camera module, ZigBee modem and the power supply to form a complete sensor unit at the meter site. A LED, shared the same power supply on the sensor unit, was suggested to provide the necessary lighting required for the image capturing. As the sensor unit was required to mount in front of the consumption display window of the meter for capturing of the usage information, an L metal bracket was recommended to hold the sensor unit in place. The development of the hardware for holding the sensor unit was under the design concept phase for this project, thus there is no prototype built for this hardware.

4.4 Hardware Connection

The connection and interface required for various individual hardware utilised is part of hardware development of this project. This section covers the 2 main connections, which are the Rabbit processor to the ZigBee modem, and the camera module to the ZigBee modem.

4.4.1 Connection between Rabbit Processor and ZigBee Modem

The ZigBee modem connected to the Rabbit processor functioned as the ZigBee coordinator, which was the main controller of the ZigBee wireless sensor network. The connection between these two components was through the RS 232 connection cable between the processor prototype board and the RF interface board. The direct pin to pin connection was illustrated to allow future implementation of the project without the supporting board. Serial port D of the Rabbit processor was selected as the transmitting and receiving port with the ZigBee modem. Pin 21 (PC0), which is the transmitting pin of the serial port D, and Pin 22 (PC1), which is the receiving pin of the serial port D, were connected to the ZigBee modem Pin 3 (DIN), which is the UART data in pin, and Pin 2 (DOUT), which is the UART data out pin respectively. A detail pin routing including the additional components on the prototype board and RF interface board is shown in Figure 11 and the figure could be cross referenced with the attached schematic diagram on Appendix 4.

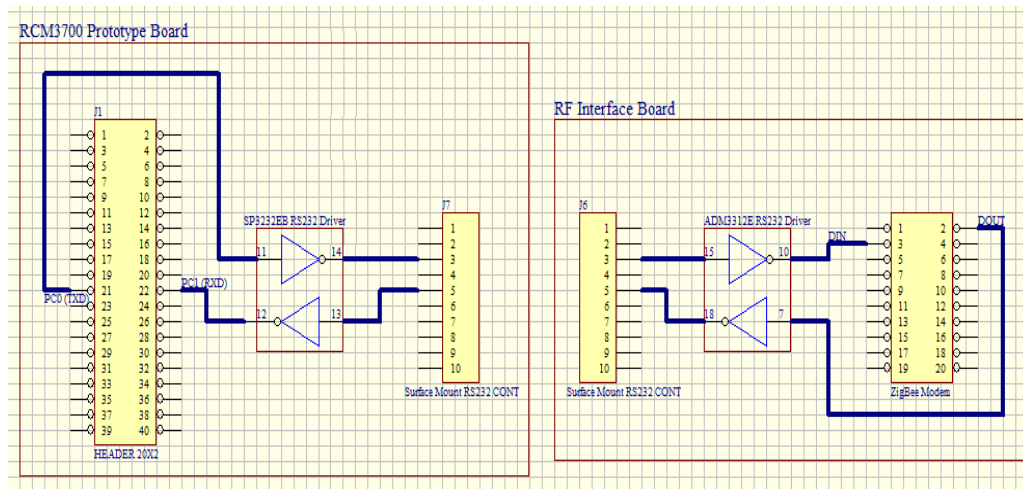


Figure 11 : Pins Configuration from Rabbit Processor to ZigBee Modem

4.4.2 Connection between Camera Module and ZigBee Modem

The ZigBee modem connected to the camera module functioned as the ZigBee End Device (ZED) in the ZigBee wireless sensor network, which only in charge of receiving and transmitting data. The ZigBee modem basically acted as an extended serial asynchronous transmit/receive port for the camera module. For this project development, the connection between these two components was through the serial communication connector (J4) available on the RF interface board. The pins connection of the J4 connector on RF interface board to the camera module is listed on Table 1.

RF Interface Board J4 Connector		Camera Module	
<u>Pin</u>	<u>Function</u>	<u>Pin</u>	<u>Function</u>
3	+3.3VDC	2	+3.3VDC
1	GND	3	GND
2	TX_OUT (UART Data Out)	4	TX
5	RX_IN (UART Data In)	5	RX

Table 1: Pins Connection for RF interface board to Camera Module

The transmission and reception pins connection for the camera module was mainly connecting the transmitting pin (TX) of the camera module to the receiving data pin of the microcontroller, and vice versa for the receiving pin (RX) of the camera module. A direct pins connection between the camera module and ZigBee is listed on Table 2 to allow future implementation of the project without the RF interface board.

Camera Module		ZigBee Modem	
<u>Pin</u>	<u>Function</u>	<u>Pin</u>	<u>Function</u>
2	+3.3VDC	1	+3.3VC (Shared the same power supply)
3	GND	10	GND (Shared the same GND)
4	TX	2	UART Data Out
5	RX	3	UART Data In

Table 2: Pins Connection for the Camera Module and the ZigBee Modem

4.5 Chapter Summary

This chapter covered the hardware development of the project. From selection, identification, understanding to utilising of each individual hardware was illustrated. The characteristic and functionality of the 2 main components of the project, which were the camera module and the ZigBee application kit, were covered in detail. The hardware interface and connection between various components was also explained to provide a better understanding on the relationship between the hardware.

The camera module performed one of the main methodologies of the project, which was capturing the power and water meter consumption information. The many integrating features of the camera module allowed it to be easily incorporate with the ZigBee modem, and together with the ZigBee network formed by the Rabbit processor to perform the picture and data capturing function.

The ZigBee application kit that came with both hardware and software allowed easy development of the project, with programmable processor and built-in hardware interface circuitry. The ZigBee application kit fulfilled the second methodology of the project, which was transmitting the data through ZigBee network. The ZigBee modem and the RF interface board provided by the application kit allowed formation of the ZigBee wireless sensor network as the communication channel for the project.

The application kit together with the computer performed the last methodologies of the project, which was displaying the information on web server. The Ethernet connection of the Rabbit processor linked up with the computer to act as a web server. The computer was in charge of displaying the web interface, image data capturing and processing.

Chapter 5.

Software Development

Software development of the project formed an important portion on the overall development of the project. Software development for this project included programming and software configuration of each individual hardware, and this provided a mean for interfacing between them and ensured the objectives and functionalities of the project were fulfilled. This chapter is divided into individual sections to describe the software development for individual hardware or for the functionalities of the project.

The camera module programming section covers the instructions and commands programmed to activate, configure and control the camera module. It also illustrates on how the microcontroller serial port was utilised in transmitting and receiving the commands and data for the camera module. The ZigBee modem programming section touches on the configuration and programming for the network setting on each modem with the X-CTU software from Digi International. The web server programming section comprises of the Dynamic C programming required for the Rabbit processor application kit to function as a web server, with linkage to the computer through Ethernet connection. Web page interface section contains the creation and programming of HTML web page and also its relationship to the programming in the Rabbit processor. This section also explains the interface portion of the web page, which include activation of image capturing and display of metering data.

As the overall project was developed mainly from the ZigBee application kit with a programmable Rabbit processor RCM3720, thus the Rabbit module programming section covered all the Dynamic C programming codes in the processor required for this project. The programming development in the Rabbit processor functioned as the main programming brain of the whole project and it included software linkage to the functions of camera module, ZigBee modem, web server and web interface page.

5.1 Camera Module Programming

The camera module utilised in the project was the u-Cam TTL serial JPEG camera module from 4D system, it has a command based interface through asynchronous serial transmission with TTL voltage level. The camera module could be programmable through any serial port of a microcontroller, and for this project development, the camera module was programmed through a serial port of the Rabbit processor. The followings are the serial port communication programming setting in the Rabbit processor with Dynamic C software:

- Serial Port D was used as it was connected to the UART data in/out pin of the ZigBee modem through the prototype board to the RF interface board
- Input and output buffer size of the serial port D was set as 256 bytes to allow a image package data size of 250 bytes to be received at the processor
- Baud rate of 57600 was selected to in sync with the ZigBee modem selected baud rate and the camera module auto detection baud rate.

Each of the camera module commands consists of 6 continuous single byte serial transmissions, thus character array of 6 bytes was used to store the commands. Each byte of the command was programmed with a 2 hex numbers, which was same as illustrated in the camera module manual. The commands were transmitted out from the serial port by writing to the port with “serDwrite”. For receiving data from the camera module, character array of various sizes was used to store the received date from the serial port of the Rabbit processor with “serDread”. The exact detail of the Dynamic C programming on the camera module function is listed under the ‘activate_cam_controller’ function inside Rabbit Module Programming Code attached on the Appendix B.

There were a total of 10 commands available for instructing the camera module, and 1 acknowledgement command. The acknowledge command must be return for each accepted command with the matching command ID and counter indicated. The commands were breakdown into parameters and listed in Table 3.

Command	ID Number	Parameter1	Parameter2	Parameter3	Parameter4
INITIAL	AA01h	00h	Colour Type	RAW Resolution (Still Image only)	JPEG Resolution
GET PICTURE	AA04h	Picture Type	00h	00h	00h
SNAPSHOT	AA05h	Snapshot Type	Skip Frame (Low Byte)	Skip Frame (High Byte)	00h
SET PACKAGE SIZE	AA06h	08h	Package Size (Low Byte)	Package Size (High Byte)	00h
Set Baud Rate	AA07h	1 st Divider	2 nd Divider	00h	00h
RESET	AA08h	Reset Type	00h	00h	XXh*
DATA	AA0Ah	Data Type	Length Byte 0	Length Byte 1	Length Byte 2
SYNC	AA0Dh	00h	00h	00h	00h
ACK	AA0Eh	Command ID	ACK Counter	00h / Package ID Byte 0	00h / Package ID Byte 1
NAK	AA0Fh	00h	NAK Counter	Error Number	00h
LIGHT	AA13h	Frequency Type	00h	00h	00h

Table 3: Commands for the u-Cam Camera Module (2010, pg 8)

The “SYNC” command was used to establish a connection between the microcontroller and the camera module, and “ACK” command must be send out after receiving the “SYNC” command. The “SYNC” command was the first command to send to the camera module after the camera powered on. The microcontroller might send out the “SYNC” command up to 60 times before an “ACK” command is returned. Figure 12 illustrated the message sequence chart of the commands between camera module and microcontroller during synchronisation process.

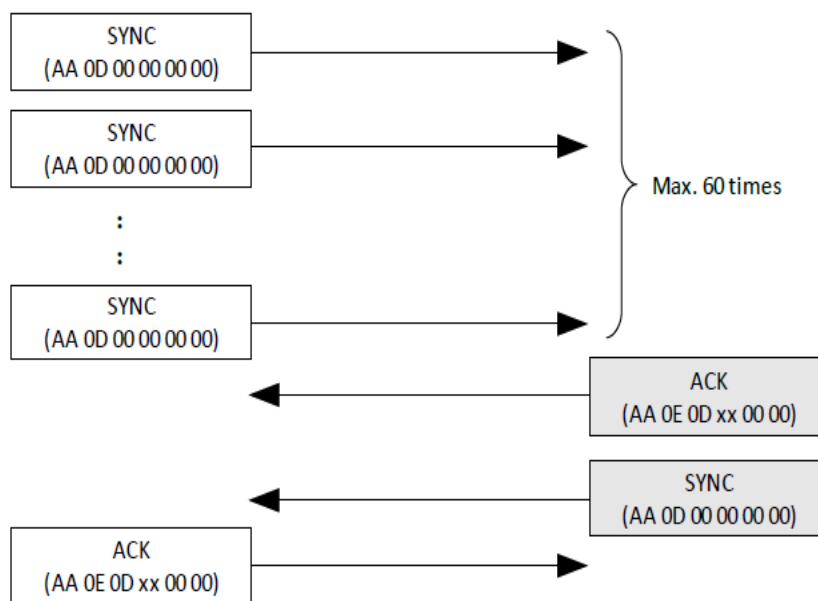


Figure 12: Message Sequence Chart for Camera Module Synchronisation Process (2010, pg 14)

– In Home Power and Water Metering

There were various picture settings available on the camera module and the setting were defined by different commands. For this project development, the picture settings were as listed below:

- Colour Type : JPEG (defined as 07h in programming)
- JPEG Resolution: 320 x 240 (defined as 05h in programming)
- Package Size : 256 bytes (defined as FAh for the lower package size byte, 00h for the higher byte)

Each data package transmitted out from the camera module consisted of its individual identification, data size, the actual image data and a verify code. The block diagram at Figure 13 shows how the image data was encapsulated into a data package. Therefore, a package size of 256 bytes consisted of 250 bytes of image data and the other 6 bytes was for information purpose.



Figure 13: Block Diagram of a Data Package from the Camera Module (2010, pg 10)

For instructing the camera module to take a snapshot picture, there were a sequence of commands to be instructed to the camera module, and each of the commands should be accompanied with an acknowledgement reply. The message sequence chart in Figure 14 defined the commands and responses from the camera modules to allow a successful snapshot picture taken and returning the picture data to the microcontroller. The sequence chart shown was defined for a JPEG photo of 640 x 480 resolution and a 512 bytes data packages.

During the software development, the programming codes ensured the pre-get picture commands, such as initial, set package size, snapshot setting, were received by the camera module with an acknowledgement. If there is no acknowledgement received at the Rabbit processor, then the respective command needed to be re-issued. When a get-picture command was issued to the camera module, an acknowledgement together with a DATA message, which consisted the picture data type and the length of the picture data, would returned from the

– In Home Power and Water Metering

camera module. Calculation in the programming code used the data length bytes in the DATA message to determine the number of expected data package to be received at the Rabbit processor. A for loop used to count down and transfer the image data for a JPEG picture file into a char array. A suitable time delay loop was inserted to provide sufficient time between transmitting out the commands and reading on the serial receiving port of the Rabbit processor.

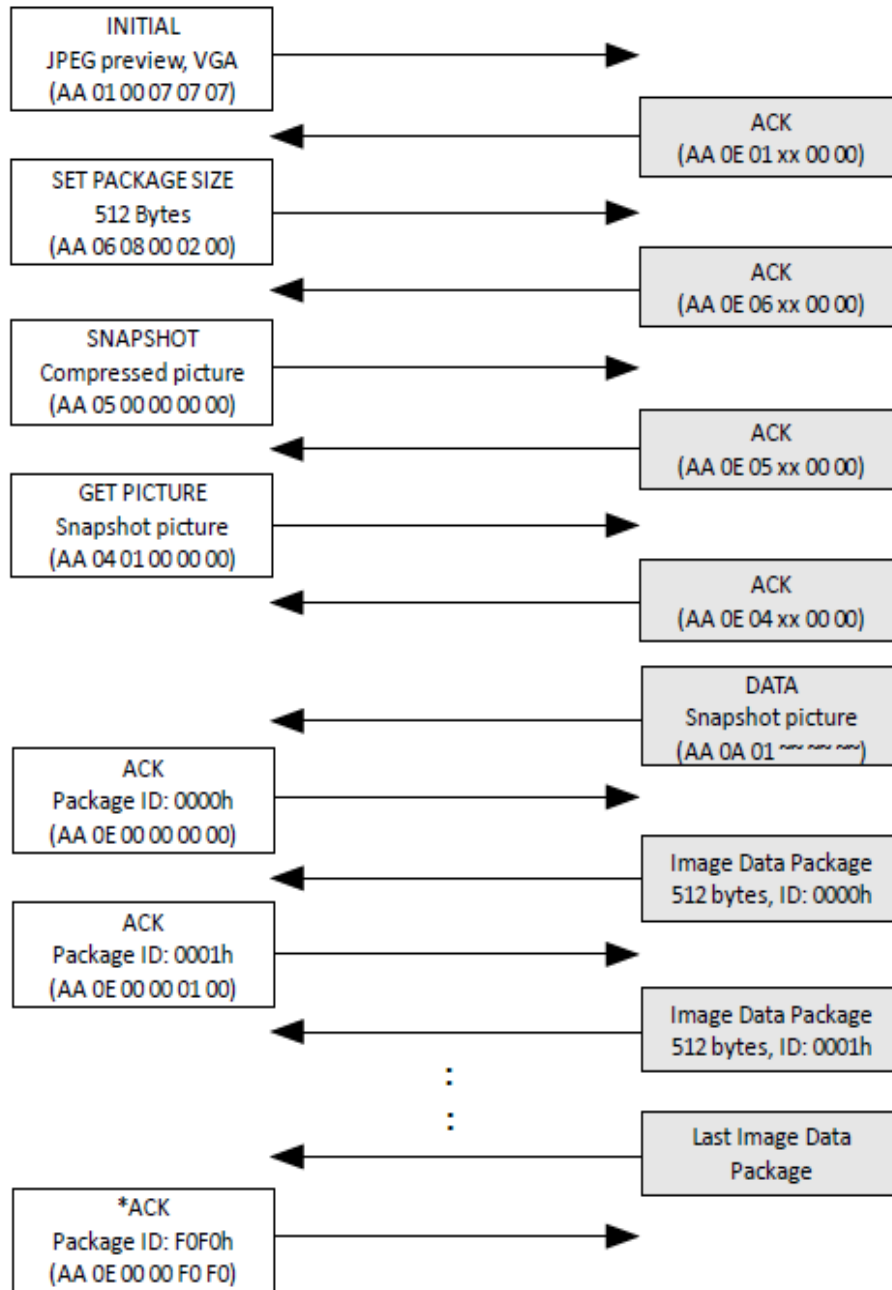


Figure 14: Message Sequence Chart for Snapshot Picture Activation on the Camera Module

(2010, pg 16)

5.2 ZigBee Modem Programming

The network configuration and setting of the ZigBee modems utilised in the project was first defined through the X-CTU software provided by Digi International. The X-CTU program allowed the firmware of modem to be uploaded, and it also provided various modem configurations to be carried out. While selecting the baud rate of ZigBee modem, the baud rate for the serial port of the Rabbit processor was ensured to be same as the modem, so to provide proper communication between them. On the Dynamic C programming side, the xbee.lib was included for the ZigBee modem to function properly in the network. Table 4 listed down the modem configuration for the modems located at the processor side and also at the sensor unit.

Configuration	Base Modem (ZC)	Remote Modem (ZED)	Remark
Firmware Version	10E8	10E8	The latest Firmware for the XBee 802.15.4 function set
Channel	C	C	Define a unique and same channel number
Pan ID	3332	3332	Define a same personnel area network ID
XBee Retries	5	5	Number of retries the modem will execute
Source Address	FFFF	FFFF	Disable reception of packets with 16 bit addresses
Interface Data Rate	6	6	6 = 57600 Same with the serial port
Source Address (High)	13A200	13A200	Original unique modem serial number
Source Address (Low)	400631A7	400630C2	
Destin. Address (High)	13A200	13A200	The serial number of the communicating end modem
Destin. Address (Low)	400630C2	400631A7	
Coordinator Enable	1	0	0 – End Device 1 - Coordinator
Node Identifier	Base	Camera	Name for identification
Coordinator Association	4-100b	NA	Allow Association
End Device Association	NA	12 – 1100b	Poll coordinator on pin wake & Auto associate

Table 4 : ZigBee Modem Configurations and Settings

5.3 Web Server Programming

The development of the web server programming formed an important part of the project development as it partially fulfilled one of the objectives of the monitoring system, which was displaying the information on a web server. The Rabbit processor with the ZigBee application kit acted as the web server for this project development. It linked up to the computer and to the World Wide Web and it allowed users to access the metering information transmitted across the ZigBee network.

The Dynamic C programming for the Rabbit processor has various pre-defined functions and libraries that provided ease on programming the Rabbit processor as a web server. There were 2 main functions while setting up the Rabbit Processor as a web server, which were the TCP/IP connection and HTTP server. The details on using the TCP/IP connection and HTTP server function with the Dynamic C programming are listed in Dynamic C TCP/IP Volume 1 User's Manual (Digi International Inc, 2009) and Dynamic C TCP/IP Volume 2 User's Manual (Digi International Inc, 2010) respectively. The manuals covered the explanations and samples of the various pre-defined functions.

The TCP/IP connection configuration was using one of the predefined configurations with Dynamic C tcp_config library, which was defined by the coding "#define TCPCONFIG 1". This selection set the Rabbit server with the following network configuration:

- Server Static IP Address : 10.10.6.100
- Network Mask : 255.255.255.0
- Gateway Address and NameServer : 10.10.6.1

The IP address setting on the Local Area Network card at the computer needed to be configured with a static IP address as 10.10.6.1 to allow the computer to connect directly with the Rabbit processor through an Ethernet connection with a crossover cable.

– In Home Power and Water Metering

To have the TCP/IP connection to function properly, the coding “#use "dcrtcp.lib"” was included at the beginning of the Dynamic C programming code to call for and activate the dcrtcp library. Within this library file, the tcp_config.lib was called to activate the predefined TCP/IP configuration. Thus, with the programming to include the necessary libraries and functions, the TCP/IP connection between the Rabbit processor and the computer was established.

The HTTP server function was included in the programming code with “#use "http.lib"”, which initiated the http library. The library provided the usage of http functions, such as http_handler() which was called continuously in the main loop to maintain the activation of the HTTP server, and to allow the HTTP server to process requests from the network. In the main loop of the Rabbit programming code, the 3 initialization functions, sock_init(), http_init(), and tcp_reserveport(80), were mandatory to call before the http_handler continuous loop to prepare the Rabbit processor to function as the HTTP server. “sock_init” was used to initialize the TCP/IP networking system. “http_init” was used to initialize the HTTP server. “tcp_reserveport(80)” enabled the smooth handling of multiple HTTP requests by reserving the TCP port 80, which is the port for HTTP.

With the predefined functions and libraries available of the Dynamic C, the configuration of the Rabbit processor as the web server was achievable with the understanding on the usage of the TCP/IP and HTTP functions and libraries. Therefore, the Rabbit processor was programmed as the web server for this project with the suitable programming codes as explained for TCP/IP connection and HTTP server function. The exact detail of the web server function Dynamic C programming is included under Appendix B Rabbit Module Programming Code.

5.4 Web Page Interface

To completely fulfil the objective of the monitoring system, which was displaying the information on the web server; it was achieved by the programming creation on the web page interface. The software development on the web page interface consisted of 3 main parts; the project front web page display, the web interface for image capturing activation and the web page for metering information display. Each of the parts made up of both the Dynamic C Rabbit processor programming and the HTML coding. The Dynamic C programming consisted of the functions and libraries utilised for linkage to web page display and common gateway interface function. The HTML coding consisted of coding for the HTML page in displaying on the web browser and its relationship with the Rabbit processor programming.

5.4.1 Web Page for Front Page Display

The front web page of this project displayed a HTML page which demonstrated its linkage to the Rabbit processor. The layout of the front web page is as shown in Figure 15. The Rabbit processor was programmed with the various pre-defined functions and libraries of the Dynamic C environment, and it created the web page interface or web page application on the web server function of the Rabbit processor.

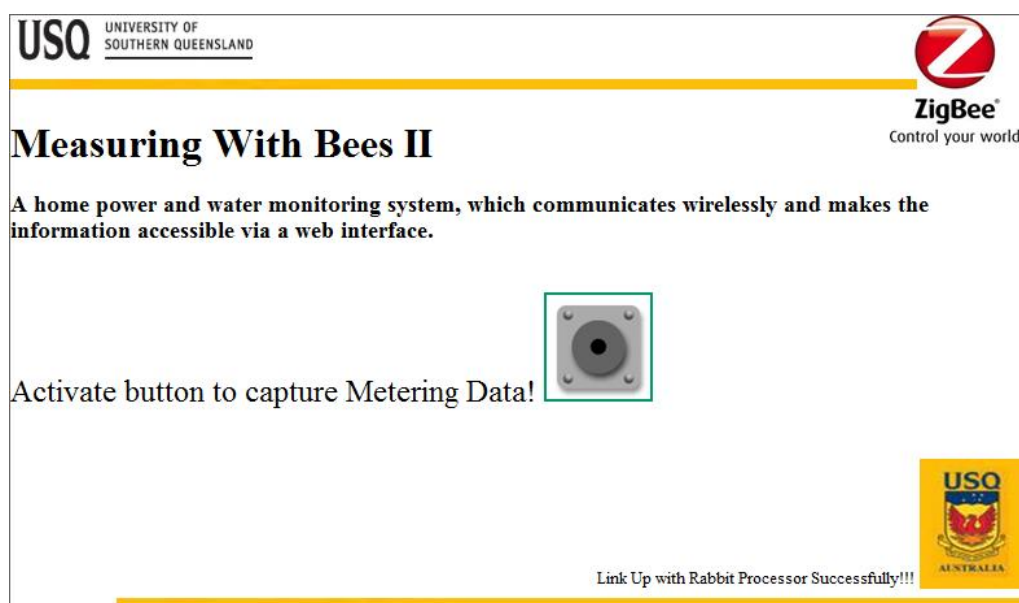


Figure 15 : The Front Page for Web Page Interface

The web page display with the Dynamic C programming mainly made up of 3 sections, which were importing the necessary files for web page display, defining the Multipurpose Internet Mail Extension (MIME) table, and setting up of the static resource table. Each of the section is explained in detail in the following paragraphs, and their functions allowed the web page display to function properly as a whole. The reference for programming related to the web page interface development is included in Dynamic C TCP/IP Volume 2 User's Manual Chapter 2 Web-Enabling the Application (Digi International Inc, 2010).

The first part of the Dynamic C programming for web page display was to import all the necessary files, including the html file which had the coding for the overall web page display. The HTML coding for the front web page display of the project is listed under Appendix C Web Interface Codes. On the Dynamic C programming, "#ximport" was used to include and import the specific file into the program flash memory. "#ximport" was defined with filename and file location as the first parameter and a symbol name as the second parameter for the programming referencing. An example was "#ximport "/ENG4111/Web/Pages/final/front.html" index_html", which imported the main HTML page of the program (front.html) and labelled it with index_html. All the jpg or gif files used on the web page interface were required to be imported individually, so it could be displayed correctly on the web browser.

The definition and initialisation of the MIME table was used to associates a specific file extension with a MIME type and a function that handles the MIME types. It was equivalent to mapping the correct file extensions to its respective MIME types. All the MIME types are registered with the relevant standards body, and they must be correctly specified in the programming, so that the web browser did not be confused.

A standard layout of the MIME table was as follow:

```
SSPEC_MIMETABLE_START  
SSPEC_MIME(extension, type)  
SSPEC_MIMETABLE_END
```

The '*extension*' defined the individual file extension for different files used in the programming and '*type*' defined the MIME type, such as text/html. With the above MIME table coding, the MIME type mapping table was set up for the web page interface. In this project development, 4 rows of MIME types definitions were listed, and they are shown on Table 5.

Extension	Type
.html	text/html
.gif	image/gif
.jpg	image/jpg
.cgi	blank

Table 5 : MIME type Mapping Table

The next step of setting up the web page display in Dynamic C environment was to set up the static resource table. "SSPEC_RESOURCETABLE_START" was used to define the beginning of the static resource table. This resource table contained individual entry that associates the resource name with a file that was imported previously through the symbol name. Individual files, variables and functions used on the web page interface needed to be specified on each entry with respective types, such as xmemfile, function, to have correct corresponding reference on the html file. The following are some of the static resource table entries used in this project software development. The complete list for the static resource table defined for this project is under Appendix B Rabbit Module Programming Code.

SSPEC_RESOURCETABLE_START

SSPEC_RESOURCE_XMEMFILE("/index.html", index_html),

SSPEC_RESOURCE_XMEMFILE("/button.gif", button_gif),

SSPEC_RESOURCE_XMEMFILE("/USQlogo.jpg", USQlogo_jpg),

SSPEC_RESOURCE_FUNCTION("/button.cgi", activate_cam_controller),

SSPEC_RESOURCETABLE_END

5.4.2 Web Interface for Image Capturing Activation

The front web page display of this project included a web interface that allow user to activate image capturing on the utilities meter. This web interface was activated with a button icon as shown in Figure 15 and it was achieved with Dynamic C common gateway interface (cgi) function.

The initialisation of the cgi function for activation of the image capturing was carried out by listing the resource name 'button.cgi' related to the symbol name 'activate_cam_controller' as a function entry on the static resource table of the Dynamic C programming code. The resource name 'button.cgi' was used in the HTML page to provide linkage to the 'activate_cam_controller' function loop in the Dynamic C programming code. The HTML code "" was included next to the image file of the button icon to provide the web interface function.

The 'activate_cam_controller' function associated with the resource name 'button.cgi' beginning the function loop coding with "int activate_cam_controller(HttpState* state)". Within this function loop, it consisted of the camera module coding which triggered the image capturing through the ZigBee network, and it also copied the image data into a character array. At the end of this function loop, "cgi_redirectto(state, address)" was used to redirect the web server to a new web page display. 'address' is the http address of the new web page display.

5.4.3 Web Page for Metering Information Display

The project was intended to display the metering consumption information on the web page for access on the World Wide Web. However, during the project development, the conversion of the image data stored in the character array to a JPEG file for displaying on the new web page was unachieved. Therefore, the "cgi_redirectto" coding inside the 'activate_cam_controller' function loop did not link to any web page display on the current project development.

Various methods and ways had been researched and attempted to convert the captured image data, which was in hexadecimal format and stored in the character array, into a JPEG file for displaying on the web page. The following listed a number of ways attempted:

- Using the root variable in the Dynamic C static resource table to define the character array as a variable that is able to access in the HTML coding. However, there are only 5 types of variable available in static resource table entry, which are 8 bits integer, 16 bits integer, 32 bits integer, 16 bits pointer and 32 bits float. It was impossible to convert the hexadecimal character image data to any of the available variable types, and be understandable in the HTML coding as a JPEG file for display.
- Using the “cgi_sendstring” function in the Dynamic C environment to direct generate the HTML page in buffer with the HTML coding written as string in the Dynamic C programming. The project able to generate an web page display using the “cgi_sendstring” function, however, it was unable to direct convert the character array data into a JPEG file with HTML coding.
- Using the function entry of the Dynamic C static resource table to define a resource name ‘image.jpg’, which link to a function loop that send out the image data that stored in the character array data as output. However, during the project development, the HTML coding unable to identify the image.jpg file and display it on the web page.
- Using the TCP socket function in the Dynamic C programming to transmit out the data of the image file stored in the character array through an identified TCP socket to the computer. Java programming was used at the computer side to process the data. ‘java.net.socket’ was used to read the socket data, ‘imageIO.read’ was used to convert the input stream from the socket to an image data, and ‘imageIO.write’ was used to write the image data to a JPEG file. A Java applet was generated with the above Java programming codes, but while including the applet on the HTML coding, the applet did not function properly to generate the JPEG file for display.

5.5 Rabbit Module Programming

The Rabbit processor acted as the main web server for the project and it allowed requests to send to it, and it returned the data through displaying the information on the web browser for accessing anywhere with internet connection. The Rabbit processor was the main brain of the whole project software development, the programming for different functions was coded under the Dynamic C software environment and they were compiled together into the Rabbit processor. Therefore, Dynamic C programming was the major programming codes used in this project development, and the understanding on the Dynamic C programming had a key important for the overall project development.

The main manual for Dynamic C programming, Dynamic C Integrated C Development System for Rabbit 2000 and 3000 Microprocessors user's Manual (Digi International Inc, 2008), introduced the Dynamic C software and stated its operations and functions. The supplement manual, Dynamic C Integrated C Development System for Rabbit Microprocessors Function Reference Manual (Digi International Inc, 2008), listed the common pre-defined functions. Both of the manual provided the information and knowledge to perform the programming in the Dynamic C environment.

With the Rabbit module programming made up of different sections for different functions, such as web server, camera module commands, ZigBee network, web page interface, the detail for each individual functions had been explained, and the unique Dynamic C functions and libraries utilised for each functions were also covered and illustrated. Each individual section of the Dynamic C programming contributed its own function to the overall Rabbit module programming development, and the Rabbit module programming in Dynamic C combined all the different functions into a single programming code. The exact Dynamic C coding of this project is listed under Appendix B Rabbit Module Programming Codes. The programming codes were properly commented to compartmentalise different functions, and also explained the purposes and functions for the coding.

5.6 Chapter Summary

This chapter covered the software development of the project. The software development was an important part for the overall project development. The imaging capturing, the ZigBee network, the web server function and the web page interface, all required the programming of respective hardware to function properly and fulfilled its requirements. As a large portion of the programming codes were written in the Dynamic C software, thus the Rabbit module programming was the major part of the software development, and it consisted of all the Dynamic C programming codes, which each of the unique functions were explained in detail in each individual sections.

Chapter 6.

Conclusion

“Measuring with Bees II – In Home Power and Water Monitoring” aimed to develop a home power and water monitoring system, which communicates wirelessly and makes the information accessible via a web interface. The project utilised a camera module at the sensor side to capture an image of the consumption window on the utility meters, it made the information of the utilities metering to be accessible and allowed the data of the image file to be transmitted wirelessly in the monitoring network. The project demonstrated the usage of ZigBee specification as the protocol for the wireless sensor network, thus the individual sensor nodes of the monitoring system communicated wirelessly through the ZigBee network. The ZigBee network was formed up by the ZigBee application kit which included of the ZigBee modems, Rabbit processor, which functioned as the main processing and programming brain of the project, and various interface and prototype boards. The programming of the camera module was carried out through the serial port of the Rabbit processor, as the camera module utilised was programmable through the asynchronous serial transmission commands. Both the serial port of the Rabbit processor, and the camera module TX/RX pins were connected to the data in/out ports of the ZigBee modem for communicating wirelessly. As the Rabbit processor contained most of the processing and programming function for this project and it was programmable with the Dynamic C software that came with the ZigBee application kit, most of the software development for the project was written in the Dynamic C environment. The various pre-defined functions for the Dynamic C software, such as HTTP server and web enable function, allowed the creation of a web interface with the Rabbit processor as the web server. The Rabbit processor was linked up with the computer through the TCP/IP connection for access to the World Wide Web. The web interface page linked up with the Rabbit web server allowed user to activate the image capturing of the consumption window of the utilities meter through the commands and data transmitting wirelessly through the ZigBee network.

The project development provided user to have an easier and more accessible way to monitor the utilities usage, such as electricity and water, thus the user able to have a better view and knowledge on the energy consumption, which lead to practise of improved energy usage control and conservation.

The project also aimed to provide the ability to access the utilities metering information remotely from the actual metering location, so to minimise the manpower required for carrying out the monthly or bi-monthly metering recordings on site. Although the latest development on the metering industry had been focused on the smart utility meter, where the metering information could be easily accessed through the pre-installed Wi-Fi or ZigBee connection, but due to the high cost of implementation for the smart meter, in both the cost of the meter and the cost of installation, the actual implementation of the smart meter on the household utilities metering infrastructure had been hindered. With this project developed using image capturing to record the metering information, it reduced the cost of implementation of the system, which provided remote monitoring and recording of the utilities meter information.

Furthermore, the project was developed with the ZigBee wireless sensor network, the number of the sensor nodes in the monitoring network could be increased to provide multiple meters monitoring due to the node network theory and the wireless mesh networking standard of the ZigBee specification. This enhanced the area of implementation of the project to include in the industrial environment where there is multiple utilities meters to monitor machine or system energy consumption. The usage of ZigBee wireless sensor network for this project development also provided an easier way to incorporate this project into other ZigBee network or ZigBee standards defined by the ZigBee Alliance, such as home automation standard, and remote control standard. With the flexibility of the ZigBee network, it allowed the project to be implemented with other ZigBee system to provide an overall energy monitoring and controlling network.

6.1 Achievement of Project Objectives

The project achieved in developing a prototype for a home power and water monitoring system. The project designed to capture the consumption of the utilities through image capturing by a camera module at the meter site, and the data of the image file was transmitted wirelessly in the ZigBee network. The camera module was linked up with the remote sensor node of the system, which communicated wirelessly through the ZigBee network formed up from the ZigBee application kit. The Rabbit processor which was part of the ZigBee application kit was programmed in Dynamic C environment to carry out the main processing function of the project, such as commanding the camera module for image capturing and displaying the web page interface. The project achieved to provide a web page interface as a linkage to the ZigBee wireless network, and to provide activation for image capturing on the utility meter.

The following were the objectives identified in the project specification that had been addressed in the project development:

- Research and study had been carried out on the background of home power and water metering, and the wireless sensor network protocols, in particular ZigBee. Chapter 2 and 3 explained the information for the research on these 2 areas.
- Imaging capturing on the consumption display of the utilities meter had been determined as the suitable method to obtain the power and water usage information for the project.
- The ZigBee application kit was used to develop the wireless sensor network for the monitoring system, and the measurement was made available on World Wide Web through the Rabbit processor which is part of the ZigBee application kit. The detail of the ZigBee application kit was covered under Chapter 4.2.
- A camera module was selected to carry out the image capturing function, and it was linked up with the ZigBee modem to pass the commands and data through the ZigBee network. The detail of the camera module was covered under Chapter 4.1.

– In Home Power and Water Metering

- The ZigBee application kit, the camera module and a computer were used to develop and build the monitoring unit. The detail of the hardware development for the monitoring unit was covered under Chapter 4.
- An interface program written in the Dynamic C environment to allow sensors to communicate through ZigBee, and implemented data capture of the power and water meters. The detail of the interface program was covered under software development in Chapter 5.
- A web interface had been developed with the Rabbit processor and the computer to allow activation of the image capturing. The detail of the web interface program was covered under Chapter 5.4. However, the project was unable to display the captured image on the web interface. The data of the image file was stored as hexadecimal form in the character array in the Dynamic C programming, and the data was verified with the hex editor to prove that the image data was correctly captured and transmitted through the ZigBee network.

6.2 Future Enhancement

As the current project development was unable to convert the image data from the hexadecimal format in the Dynamic C environment to a JPEG file for displaying on the web page, thus the major improvement of the project was to carry out the conversion and display the picture file on the web page for user to access the metering information.

There were also other areas of the project that could further enhanced to improve the features of the monitoring system. As the project was developed using the ZigBee wireless network, thus the project could be enhanced by implementing with other ZigBee enabled devices. Another enhancement was to image process the captured picture of the consumption window, and to convert the utility rate into digital format for storage in database and display on the web page. Details of these 2 main recommended enhancement areas were listed in the following sections.

6.2.1 ZigBee Network

ZigBee standard had been widely implemented in many systems, especially home automation and energy control, thus with the project developed based on the ZigBee network, the project could be enhanced by implementing with other ZigBee control devices, such as electricity appliances, lighting switches. Hence, with one web page, the user able to monitor the utilities usage, and at the same time, control the appliances in the ZigBee network.

6.2.2 Image Processing

Image processing on the captured picture of the utility meter allowed the consumption rate to be converted and stored in digital format, which enhanced the features of the project with measurement storage, and historical data on usage rate. The image processing could be carried out by the Java programming at the computer with the Java Optical Character Recognition (OCR) library, and the converted data could be stored at the database for access anytime on the web interface and for historical reference.

List of References

- AN413 ZigBee/802.15.4 Application Kit, Revision A, technical manual, Digi International Inc., viewed 06 Apr 2011, < http://ftp1.digi.com/support/documentation/0220112_a.pdf>
- DECC 2011, Department of Energy and Climate Change, United Kingdom, viewed 05 Sep 2011, < <http://www.decc.gov.uk/>>
- Elahi, A & Gschwender, A, 2009, *ZigBee Wireless Sensor and Control Network*, Prentice Hall, Safari Books Online, viewed 2 Apr 2011, < <http://proquestcombo.safaribooksonline.com.libproxy.nlb.gov.sg/book/electrical-engineering/communications-engineering/9780137059355>>
- Dynamic C Integrated C Development System for Rabbit 2000 and 3000 Microprocessors User's Manual 2008, technical manual, Digi International Inc., viewed 15 May 2011, < <http://ftp1.digi.com/support/documentation/019-0125K.pdf>>
- Dynamic C Integrated C Development System for Rabbit Microprocessors Function Reference Manual 2008, technical manual, Digi International Inc., viewed 18 May 2011, < http://ftp1.digi.com/support/documentation/0190113_m.pdf >
- Dynamic C TCP/IP Volume 1 User's Manual 2009, technical manual, Digi International Inc., viewed on 6 Jun 2011, < http://ftp1.digi.com/support/documentation/0190143_h.pdf>
- Dynamic C TCP/IP Volume 2 User's Manual 2010, technical manual, Digi International Inc., viewed on 6 Jun 2011, < http://ftp1.digi.com/support/documentation/019-0144_G.pdf >
- Gill K, Shuang H, Fang Y & Xin L 2009, 'A ZigBee-based Home Automation System', *IEEE Journal*, Vol 55, Issue 2, P. 422-430, viewed 12 Sep 2011, IEEE Xplore Digital Library, < http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5174403>
- Gislaon, D, 2008, *ZigBee Wireless Networking*, Newnes, Safari Books Online, viewed 29 Mar 2011, < <http://proquestcombo.safaribooksonline.com.libproxy.nlb.gov.sg/book/electrical-engineering/communications-engineering/9780750685979>>
- Jinsoo H, Chang S & Ilwoo L 2011, 'More Efficient Home Energy Management System based on ZigBee Communication and Infrared Remote Controls', *IEEE Journal*, Vol 57, Issue 1, P. 85-89, viewed 28 Sep 2011, IEEE Xplore Digital Library, < http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5735485>
- Kiefel R 2009, *Measuring with Bees – Wireless Web Temperature Sensor*, Master of Engineering Science, University of Southern Queensland, Toowoomba, Australia.
- Kinney P 2003, 'ZigBee Technology: Wireless Control that Simply Works', Communications design conference, viewed 27 May 2011, < <https://docs.zigbee.org/zigbee-docs/dcn/03-1418.doc>>

Measuring with Bees II

– In Home Power and Water Metering

Li P, Li J & Jing J 2010, 'Wireless Temperature Monitoring System based on the ZigBee Technology', *IEEE Journal, Computer Engineering and Technology (ICCET)*, 2010 2nd International Conference, viewed 12 Sep 2011, IEEE Xplore Digital Library, < http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5486248>

Luan S, Teng J, Chan S & Hwang 2009, 'Development of a smart power meter for AMI based on ZigBee Communication', *IEEE Journal, Power Electronics and Drive Systems 2009 International Conference*, viewed 1 Sep 2011, IEEE Xplore Digital Library, < http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5385726>

Multi-Utility Meter Report 2010, 8th Edition, Annual Report, ABS Energy Research, UK, viewed 20 Oct 2011, < <http://web.ebscohost.com.ezproxy.usq.edu.au/ehost/pdfviewer/pdfviewer?sid=22e0efcc-0333-4180-9505-38ebe27465be%40sessionmgr12&vid=1&hid=21>>

Multi-Utility Meter Report 2006, 5th Edition, Annual Report, ABS Energy Research, UK, viewed 20 Oct 2011, < <http://web.ebscohost.com.ezproxy.usq.edu.au/ehost/pdfviewer/pdfviewer?sid=664034b8-b184-4134-996e-3118742f12a1%40sessionmgr11&vid=2&hid=105>>

Ofwat 2011, Water Service Regulation Authority, United Kingdom, viewed 15 Apr 2011, < <http://www.ofwat.gov.uk/>>

Singapore Power 2010, Singapore Power Ltd, Singapore, viewed 20 April 2011, < <http://www.singaporepower.com.sg/irj/portal/spservices>>

Stedman L 2006, 'Motivations for Metering', Magazine of the International Water Association, viewed 25 April 2011, < http://www.iwapublishing.com/pdf/Metering_W21_apr06.pdf>

Townsend C & Arms S 2004, *Wireless Sensor Networks: Principles and Applications*, MicroStrain Inc, viewed 21 Apr 2011, < <http://microstrain.com/white/Wilson-chapter-22.pdf>>

uCam Serial JPEG Camera Module Data Sheet 2010, Revision 4, data sheet, 4D Systems Pty Ltd, Australia, viewed 05 Apr 2011, < <http://www.4dsystems.com.au/downloads/micro-CAM/Docs/uCAM-DS-rev4.pdf>>

uUSB-MB5 Data Sheet, Revision 1, data sheet, 4D Systems Pty Ltd, Australia, viewed 25 May 2011, < http://www.4dsystems.com.au/downloads/micro-USB/uUSB-MB5/Docs/Pdf/uUSB-MB5_DataSheet_rev1.pdf>

Wald M & Helft M, 2009, 'Google Taking a step into Power Metering', The New York Times, 09 February, viewed 14 Apr 2011, < <http://www.nytimes.com/2009/02/10/technology/companies/10grid.html>>

Water Utility Meters Report 2003, 2nd Edition, Annual Report, ABS Energy Research, UK, viewed 20 Oct 2011, < <http://web.ebscohost.com.ezproxy.usq.edu.au/ehost/pdfviewer/pdfviewer?sid=ca59a995-b96a-4b0e-a76a-8443d3091502%40sessionmgr10&vid=1&hid=21>>

Measuring with Bees II

– In Home Power and Water Metering

XBee/XBee-PRO OEM RF Modules Product Manual 2006, data sheet, MaxStream Inc, United States, viewed 26 Apr 2011, <
http://ftp1.digi.com/support/documentation/90000982_A.pdf>

Zheng, P, Peterson, L, Davie, B & Farrel, A, 2009, *Wireless Networking Complete*, Morgan Kaufmann, Safari Books Online, viewed 26 Mar 2011, <
<http://proquestcombo.safaribooksonline.com.libproxy.nlb.gov.sg/book/networking/wireless/9780123750778>>

Zhou Y, Yang X, Guo X, Zhou M & Wang L 2007, 'A Design of Greenhouse Monitoring & Control System based on ZigBee Wireless Sensor Network', *IEEE Journal, Wireless Communications, Networking and Mobile Computing*, 2007. WiCom 2007 International Conference, viewed 10 Sep 2011, IEEE Xplore Digital Library, <
http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5661599>

ZigBee Alliance 2011, ZigBee Alliance, California, USA, viewed 12 Mar 2011, <
<http://www.zigbee.org/Home.aspx>>

Appendix A

Project Specification

University of Southern Queensland
FACULTY OF ENGINEERING AND SURVEYING
ENG 4111 / 4112 Research Project

PROJECT SPECIFICATION

For: Yian Yian Tan 0050039354

Topic: MEASURING WITH BEES II – IN HOME POWER AND WATER MONITORING

Supervisor: Dr Alexander A. Kist

Project Aim: To develop a home power and water monitoring system, which communicates wirelessly and makes the information accessible via a web interface.

PROGRAMME: (Issue A, 22 March 2011)

1. Research the background of home power and water metering. Determine the suitable methods of capturing power and water usage information.
2. Research wireless sensor network protocols, in particular ZigBee; and investigate possibilities to make measurement available on the www.
3. Evaluate and select suitable power and water sensors for home monitoring.
4. Develop and built the monitoring unit.
5. Create an interface program to allow sensors to communicate through ZigBee and implement data capture of power and water meters.
6. Develop a web interface to display on the power and water monitoring information.

As time permits:

7. Include additional features on the web interface, such as usage display, usage alert
8. Further analyse and design on the sensor system to allow adaption on different water and power meters.

AGREED:

Yian Yian Tan, Student

Approved

Dr Alexander A.Kist, Supervisor

Date: 21 Mar 2011

Date: 21 Mar 2011

Appendix B

Rabbit Module Programming Codes

Measuring with Bees II

– In Home Power and Water Metering

```
#class auto

//Defining the pre-selected TCP/IP configuration
#define TCPCONFIG 1
#define TCP_BUF_SIZE 2048
#define HTTP_MAXSERVERS 2
#define MAX_TCP_SOCKET_BUFFERS 2

//Define the Labels for directing the response under CGI
#define REDIRECTHOST _PRIMARY_STATIC_IP
#define REDIRECTTO "http://"REDIRECTHOST"/redir.html"

//Defining the Serial Port setting for ZigBee modem communication
#define ATCMDRSP_SP D
#define DINBUFSIZE 255 //PC1 = RxD -- Xbee pin 2 = Dout
#define DOUTBUFSIZE 255 //PC0 = TxD -- Xbee pin 3 = Din
#define DEFAULTBAUD 57600L //xbee factory default baud rate

// Initializing the required libraries
#mmap xmem
#use "drtcp.lib"
#use "http.lib"
#use xbee.lib

//Importing the necessary files for web page interface
#ximport "/ENG4111/Web/Pages/final/front.html" index_html
#ximport "/ENG4111/Web/Pages/final/redir.html" redir_html
#ximport "/ENG4111/Web/Pages/final/button.gif" button_gif
#ximport "/ENG4111/Web/Pages/final/colourbar.jpg" colourbar_jpg
#ximport "/ENG4111/Web/Pages/final/photo.jpg" photo_jpg
#ximport "/ENG4111/Web/Pages/final/USQlogo.jpg" USQlogo_jpg
#ximport "/ENG4111/Web/Pages/final/usqlogo_line.gif" Usqlogoline_jpg
#ximport "/ENG4111/Web/Pages/final/ZigBee.jpg" ZigBee_jpg

//Define to use the Hex string to byte conversion
#define hexstrtobyte

//Defining the MIME table for respective files used.
SSPEC_MIMETABLE_START
SSPEC_MIME(".html", "text/html"),
SSPEC_MIME(".gif", "image/gif"),
SSPEC_MIME(".jpg", "image/jpg"),
SSPEC_MIME(".cgi", "")
SSPEC_MIMETABLE_END
```

Measuring with Bees II

– In Home Power and Water Metering

//The CGI function for activation of image capturing

```
int activate_cam_controller(HttpState* state)
```

```
{
```

```
    //Initializing the char array for the commands to the camera module
```

```
    char sync_cam[6];
```

```
    char ack[6];
```

```
    char reply_long[12];
```

```
    char reply[6];
```

```
    char init[6];
```

```
    char set_pack[6];
```

```
    char snap[6];
```

```
    char get_pic[6];
```

```
    char data[6];
```

```
    char reset[6];
```

```
    char poweroff[6];
```

```
    char set_baud[6];
```

```
    // Initializing a char array to store the image package = 256bytes
```

```
    char image[256];
```

```
    // Initializing a char array to store the final image data from all the packet
```

```
    char photo[30000];
```

```
    // Initializing the camera module commands' ID
```

```
    char command_ID_cam;
```

```
    char command_ID_chip;
```

```
    char command_ID_init;
```

```
    char command_ID_setpack;
```

```
    char command_ID_snap;
```

```
    char command_ID_getpic;
```

```
    char command_ID_reset;
```

```
    char command_ID_poweroff;
```

```
    // Initializing the char variables to store parameters of the image data
```

```
    char ack_counter_cam;
```

```
    char ack_counter_chip;
```

```
    char ack_counter_pack;
```

```
    char pack_ID_0;
```

```
    char pack_ID_1;
```

```
    char package_ID;
```

```
    // Initializing the variables to calculate the number of data packet and image size from  
    the return DATA file
```

```
    int image_size;
```

```
    int number_pack;
```

Measuring with Bees II

– In Home Power and Water Metering

```
char data_len_1, data_len_2, data_len_3;

// Initializing a variable to count the number of wrong packet received
int wrongpack_count;

// Initializing the variables for the delay loop
unsigned long int endtime;
unsigned long int delaystop;

// Initializing the variables for looping in the programming
int count,n,i,j,k;

//Defining the command ID
command_ID_cam='\x0d';
command_ID_chip='\xff';
command_ID_init='\x01';
command_ID_setpack='\x06';
command_ID_snap='\x05';
command_ID_getpic='\x04';
command_ID_reset='\x08';
command_ID_poweroff='\x09';

//Defining the counter
ack_counter_cam='\x00';
ack_counter_chip='\x00';
ack_counter_pack='\x00';

//Defining the variable for normal ack
pack_ID_0='\x00';
pack_ID_1='\x00';

//Defining for package ID counting
package_ID='\x00';

//Defining the variable for loop count
count=1;

//Defining the Sync command
sync_cam[0] = '\xaa';
sync_cam[1] = '\x0d';
sync_cam[2] = '\x00';
sync_cam[3] = '\x00';
sync_cam[4] = '\x00';
sync_cam[5] = '\x00';
```


Measuring with Bees II

– In Home Power and Water Metering

```
//Defining the ACK command for return from camera module
ack[0] = '\xaa';
ack[1] = '\x0e';
ack[2] = command_ID_cam;
ack[3] = ack_counter_cam;
ack[4]= pack_ID_0;
    ack[5]= pack_ID_1;

//Defining the baud rate for 9600bps if the 57600bps baud rate fails the
communication test.
set_baud[0] = '\xaa';
set_baud[1] = '\x07';
set_baud[2] = '\xbf'; //for 9600 baud rate
set_baud[3] = '\x01';
set_baud[4]= '\x00';
set_baud[5]= '\x00';

//Open the serial port D for communication to ZigBee Module
serOpen(ATCMRSP_SP,DEFAULTBAUD); //using Baud rate 57600
serFlowCtrlOn(ATCMRSP_SP); //enable flow control
//Clean away the write and read buffer
serWrFlush(ATCMRSP_SP);
serRdFlush(ATCMRSP_SP);

//To try out the baud rate and communication path on the serial port
printf("\nTrying DEFAULTBAUD (%ld) ",DEFAULTBAUD);
if(xb_atModeOn(1500)<0) // if fails try 9600
{ printf("FAILED, trying (57600L) ");
  serOpen(ATCMRSP_SP,9600L);
  serWrFlush(ATCMRSP_SP);
  serRdFlush(ATCMRSP_SP);
  if(xb_atModeOn(1500)<0)
  { printf("\nTried 57600 baud and 9600 baud and Failed\n");
    exit(0);
  }
  else
  { serDwrite(set_baud, 6);
  }
}
printf("SUCCESS\n");

//=====
//Synchronising with Camera
```

Measuring with Bees II

– In Home Power and Water Metering

```
        while (count<=60)
    {
        costate{
        //Write to serial port D with SYNC command and wait for response
        waitfor (n=serDwrite(sync_cam, 6)>0);
        {
            //Delay loop
            waitfor (DelaySec(2));

            //Enter if loop if receive data from the serial port D
            if ((n=serDread(reply_long,12,50)>0)
            {
                printf("\nAble to read!");
                printf("\n%x %x %x %x %x %x",reply_long[0],reply_long[1],reply_long[2],
reply_long[3],reply_long[4], reply_long[5]);
                printf("\n%x %x %x %x %x %x",reply_long[6],reply_long[7],reply_long[8],
reply_long[9],reply_long[10], reply_long[11]);

                //Comparing to see whether the received data is correct Acknowledgement
                if
                ((reply_long[0]==ack[0])&&(reply_long[1]==ack[1])&&(reply_long[2]==com
mand_ID_cam))
                {
                    printf("\nAcknowledge received from Camera");

                    //Comparing to see whether a SYNC command is received from the camera
                    module
                    if ((reply_long[6]==sync_cam[0])&&(reply_long[7]==sync_cam[1]))
                    {
                        printf("\nSync received from Camera\n");

                        //Defining the ACK command
                        ack[2] = command_ID_chip;
                        ack[3] = ack_counter_chip;

                        //Increment on ack_counter for chip
                        ack_counter_chip = ack_counter_chip +1;

                        //Writing the ACK command to the serial port
                        n=serDwrite(ack, 6);
                        break;
                    }
                }
            }
        }
        else
```

Measuring with Bees II

– In Home Power and Water Metering

```
{
    printf("\nWrong Acknowledge received from Camera");
    serDrdFlush;
}
    break;
}
//Increment on the counting for Sync Command issued
printf("\nCount for sync %d",count);
count=count+1;
}}
}

//=====

//=====
//Initial Camera to capture JPEG image with 320 x 240 (\x05)

//defining Initial command
init[0] = '\xaa';
init[1] = '\x01';
init[2] = '\x00';
init[3] = '\x07';           //colour type -- JPEG
init[4] = '\x05';           // raw resolution -- 320 x 240
init[5] = '\x05';           // jpeg resolution -- 320 x 240

//Clear the serial port read and write buffer
serDwrFlush;
serDrdFlush;

//Resetting the counting loop variable
count=1;

while(count<=30)
{
    costate{
        //Write to serial port D with INIT command and wait for response
        waitfor (n=serDwrite(init, 6)>0);
        {
            //Delay loop
            waitfor (DelaySec(2));

            //Enter if loop if receive data from the serial port D
            if ((n=serDread(reply,6,50))>0)
            {
```

Measuring with Bees II

– In Home Power and Water Metering

```
printf("\nAble to read!");
printf("\n%x %x %x %x %x %x",reply[0],reply[1],reply[2], reply[3],reply[4],
reply[5]);
```

```
//Comparing to see whether the received data is correct Acknowledgement
if ((reply[0]==ack[0])&&(reply[1]==ack[1])&&(reply[2]==command_ID_init))
{
    printf("\nInitial completed\n");
    break;
}
```

```
//Comparing to check whether NAK message is received
else if ((reply[0]==ack[0])&&(reply[1]=='\x0f'))
```

```
{
    printf("\nNAK message received");
}
```

```
else
```

```
{
    printf("\nWrong Acknowledge received from Camera (For init)");
    printf("\nRe-Initial required");
}
```

```
}
```

```
}
```

```
else
```

```
{
    printf("\nWaiting for ACK (For init)");
}
```

```
//Increment on the counting for Init Command issued
```

```
printf("\nCount for init %d",count);
```

```
count=count+1;
```

```
}}
```

```
}
```

```
//=====
```

```
//=====
```

```
//Set Package size 256 bytes
```

```
//defining the Set package command
```

```
set_pack[0] = '\xaa';
```

```
set_pack[1] = '\x06';
```

```
set_pack[2] = '\x08';
```

```
set_pack[3] = '\x00'; //package size (lower bytes)
```

```
set_pack[4] = '\x01'; //package size (higher bytes)
```

```
set_pack[5] = '\x00';
```

Measuring with Bees II

– In Home Power and Water Metering

```
//Clear the serial port read and write buffer
serDwrFlush;
serDrdFlush;

//Resetting the counting loop variable
count=1;

while(count<=10)
{
  costate{
    //Write to serial port D with Set Package command and wait for response
    waitfor(n=serDwrite(set_pack, 6)>0);
    {
      waitfor (DelaySec(2));

      //Enter if loop if receive data from the serial port D
      if ((n=serDread(reply,6,50)>0)
        {
          printf("\nAble to read!");
          printf("\n%x %x %x %x %x %x",reply[0],reply[1],reply[2],
reply[3],reply[4], reply[5]);

          //Comparing to see whether the received data is correct Acknowledgement
          if
((reply[0]==ack[0])&&(reply[1]==ack[1])&&(reply[2]==command_ID_setpack))
          {
            printf("\nSet Package completed\n");
            break;
          }
          //Comparing to check whether NAK message is received
          else if ((reply[0]==ack[0])&&(reply[1]=='\x0f'))
          {
            printf("\nNAK message received");
          }
          else
          {
            printf("\nWrong Acknowledge received from Camera (For Set
package)");
            printf("\nRe-Set Package required");
          }
        }
      else
      {
        printf("\nWaiting for ACK (For Set package)");
      }
    }
  }
}
```

```

//Increment on the counting for Set Package Command issued
    printf("\nCount for Set package %d",count);
    count=count+1;
}}
}
//=====

//=====
//Activation of the SnapShot command

//defining Snapshot command
snap[0] = '\xaa';
snap[1] = '\x05';
snap[2] = '\x00'; //snapshot type
snap[3] = '\x00'; //skip frame(lower bytes)
snap[4] = '\x00'; //skip frame(higher bytes)
snap[5] = '\x00';

//Clear the serial port read and write buffer
serDwrFlush;
serDrdFlush;

//Resetting the counting loop variable
count=1;

while(count<=10)
{
    costate{
//Write to serial port D with Snapshot command and wait for response
        waitFor(n=serDwrite(snap, 6)>0);
        {
            waitFor (DelaySec(2));

//Enter if loop if receive data from the serial port D
            if ((n=serDread(reply,6,50))>0)
            {
                printf("\nAble to read!");
                printf("\n%x %x %x %x %x %x",reply[0],reply[1],reply[2],
reply[3],reply[4], reply[5]);

//Comparing to see whether the received data is correct Acknowledgement
                if
((reply[0]==ack[0])&&(reply[1]==ack[1])&&(reply[2]==command_ID_snap))

```

Measuring with Bees II

– In Home Power and Water Metering

```
    {
        printf("\nSnapShot Requested\n");
        break;
    }
//Comparing to check whether NAK message is received
else if ((reply[0]==ack[0])&&(reply[1]=='\x0f'))
    {
        printf("\nNAK message received");
    }
else
    {
        printf("\nWrong Acknowledge received from Camera (For SnapShot)");
        printf("\nSnapshot request to resend");
    }
}
else
{
    printf("\nWaiting for ACK (For SnapShot)");
}

//Increment on the counting for Snapshot Command issued
printf("\nCount for SnapShot %d",count);
count=count+1;
}}
}

//=====

//=====

//Activation of the Get Picture Command

//Defining Get Picture command
get_pic[0] = '\xaa';
get_pic[1] = '\x04';
get_pic[2] = '\x01'; //picture type (Snapshot type)
get_pic[3] = '\x00';
get_pic[4] = '\x00';
get_pic[5] = '\x00';

//Defining DATA command
data[0] = '\xaa';
data[1] = '\x0a';
data[2] = '\x01'; //data type (Snapshot type)

serDwrFlush;
```

Measuring with Bees II

– In Home Power and Water Metering

```
serDrdFlush;

count=1;

while(count<=50)
{
  costate{
    //Write to serial port D with Get Picture command and wait for response
    waitfor(n=serDwrite(get_pic, 6)>0);
    {
      waitfor (DelaySec(2));

      //Enter if loop if receive data from the serial port D
      if ((n=serDread(reply_long,12,200)>0)
        {
          printf("\nAble to read!");
          printf("\n%x %x %x %x %x %x",reply_long[0],reply_long[1],reply_long[2],
reply_long[3],reply_long[4], reply_long[5]);
          printf("\n%x %x %x %x %x %x",reply_long[6],reply_long[7],reply_long[8],
reply_long[9],reply_long[10], reply_long[11]);

          //Comparing to see whether the received data is correct Acknowledgement
          if
((reply_long[0]==ack[0])&&(reply_long[1]==ack[1])&&(reply_long[2]==command_ID
_getpic))
          {
            printf("\nGet Picture Requested");

            //Comparing to see whether the correct DATA command is received
            if ((reply_long[6]==data[0]) && (reply_long[7]==data[1]) &&
(reply_long[8]==data[2]))
            {
              printf("\nData command received from Camera\n");
              printf("\n%x %x %x", reply_long[9], reply_long[10], reply_long[11]);

              //Copy the data length into variables: data_len_1, data_len_2, data_len_3
              data_len_1 = reply_long[9];
              data_len_2 = reply_long[10];
              data_len_3 = reply_long[11];

              break;
            }
          }
          else
          {
            printf("\nWrong Data command received from Camera");
          }
        }
      }
    }
  }
}
```


Measuring with Bees II

– In Home Power and Water Metering

```
        printf("\n%x %x %x", reply_long[6], reply_long[7], reply_long[8]);
        break;
    }
}
//Comparing to check whether NAK message is received
else if ((reply[0]==ack[0])&&(reply[1]=='\x0f'))
{
    printf("\nNAK message received");
}
else
{
    printf("\nWrong Acknowledge received from Camera (For Get Picture)");
    printf("\nGet Picture request to resend");
}
}
else
{
    printf("\nWaiting for ACK (For Get Picture)");
}

//Increment on the counting for Get Picture Command issued
printf("\nCount for Get Picture %d",count);
count=count+1;
}}
}

//=====

//=====
//Calculate image size and number of data packet using data len from the DATA
command

//Using the hexstrtobyte function to convert the hex string to byte and calculate the
total image size

image_size=hexstrtobyte(data_len_1)+hexstrtobyte(data_len_2)*256+hexstrtobyte(data_
len_3)*65536;

printf("\nImage Size = %d", image_size);

//Calculate the number of data packet to be received the rabbit processor
number_pack = image_size / (256-6);

printf("\nNumber of pack = %d", number_pack);
```

Measuring with Bees II

– In Home Power and Water Metering

```
//=====
//Receive data file

//Clear the serial port read and write buffer
serDwrFlush;
serDrdFlush;

//define command ID for ACK on package
ack[2] = '\x00';    //command ID for package
ack[3] = '\x00';

//define the count for wrong ack or wrong file
wrongpack_count=0;

//define variable for complete photo data
k=0;

//Using for loop to count down and receive the data packet
for (i=0; i<=(number_pack+1); i++)
{
    //Compare last package size to trigger sending special reset
    if (i==number_pack)
    {
        //Set reset command
        reset[0]='\xaa';
        reset[1]='\x08';
        reset[2]='\x01';    //reset type -reset state machine only
        reset[3]='\x00';
        reset[4]='\x00';
        reset[5]='\xff';    //special reset

        //Defining the delay variables
        endtime=MS_TIMER+200;
        delaystop=0;
        //delay loop
        while((endtime-MS_TIMER)>delaystop)
        {
        }

        //Sending special reset command
        if (n=serDwrite(reset,6)>0)
        {
            printf("\nCamera resetted!");
        }
    }
}
```

```
        break;
    }

//Defining the Acknowledge message for package ID
ack[4]= pack_ID_0;
ack[5]= pack_ID_1;

//Sending the data ack to camera module
n=serDwrite(ack,6);

//Defining the delay variables
endtime=MS_TIMER+200;
delaystop=0;
//delay loop
while((endtime-MS_TIMER)>delaystop)
{
}

//Enter if loop if receive data from the serial port D
if ((n=serDread(image,256,100))>0)
{
    //Comparing the package ID to determine whether the correct package is received
    if ((image[0] == pack_ID_0)&& (image[1] == pack_ID_1))
    {
        printf("\nPackage ID matched");

        //To print out the package ID
        printf("\n%x %x", image[0],image[1]);

        //Looping for copying the photo data over
        for(j=0 ; j<(256-6);j++)
        {
            //Copying the image data over to the photo char array which consist of all
            the image file
            photo[k]=image[4+j];

            //Increment on the variable for image size
            k=k+1;

            //Break out from the data copying loop when end of image data reached
            if (k>image_size)
                break;
        }
    }
}
```

```
    printf("\nPhoto data copied");

    //define and increment the package ID
    pack_ID_0 = pack_ID_0+1;

    //Increment the next byte package ID if \xff is reached
    if (pack_ID_0 == '\xff')
    {
        pack_ID_1 = pack_ID_1+1;
        pack_ID_0 = '\x00';
    }

    //Reset the wrong package count if the image data is received
    wrongpack_count=0;

    serDwrFlush;
    serDrdFlush;

}
//If the package ID is unmatch
else
{
    printf("\nPackage ID Un-match\n");
    printf("%x %x\n", image[0],image[1]);

    //Decrement on the counting variable of image packet
    i=i-1;

    //Increment on the wrong package count
    wrongpack_count=wrongpack_count+1;

    serDwrFlush;
    serDrdFlush;

    //Break out from the data receiving loop if more than 10 wrong package
    received
    if (wrongpack_count>=10)
    {
        break;
    }
}
}
```

Measuring with Bees II

– In Home Power and Water Metering

```
//For no data received on the serial port
else
{
    printf("\nDidn't receive photo data!!\n");
    i=i-1;
}
}

//Compare to see whether there are more than 10 wrong packages receive.
if (wrongpack_count>=10)
{
    printf("\nError receiving the file!!");
}
else
{
    printf("\nAll package of data received!");
}

//To close the Serial port D
serDclose;
printf("\nclose");

//Display the image data and converting the display to easy read manner
for (i=0; i<image_size; i++)
{
    if ((photo[i] == '\x01')||(photo[i] == '\x02')||(photo[i] == '\x03')||(photo[i] ==
'\x04')||(photo[i] == '\x05')||(photo[i] == '\x06')||(photo[i] == '\x07')||(photo[i] ==
'\x08')||(photo[i] == '\x09')||(photo[i] == '\x0a')||(photo[i] == '\x0b')||(photo[i] ==
'\x0c')||(photo[i] == '\x0d')||(photo[i] == '\x0e')||(photo[i] == '\x0f')||(photo[i] == '\x00'))
        printf("0%x ", photo[i]);
    else
        printf("%2x ", photo[i]);
}

//Programming to convert the char array to .jpg file and to display on the web page
// To be insert in project future enhancement

//Redirect the web page to required display
cgi_redirectto(state,REDIRECTTO);
return 0;
}
```

Measuring with Bees II

– In Home Power and Water Metering

//Defining the static resource table

SSPEC_RESOURCETABLE_START

```
SSPEC_RESOURCE_XMEMFILE("/index.html", index_html),
SSPEC_RESOURCE_XMEMFILE("/redir.html", redir_html),
SSPEC_RESOURCE_XMEMFILE("/button.gif", button_gif),
SSPEC_RESOURCE_XMEMFILE("/colourbar.jpg", colourbar_jpg),
SSPEC_RESOURCE_XMEMFILE("/photo.jpg", photo_jpg),
SSPEC_RESOURCE_XMEMFILE("/USQlogo.jpg", USQlogo_jpg),
SSPEC_RESOURCE_XMEMFILE("/usqlogo_line.gif", Usqlogoline_jpg),
SSPEC_RESOURCE_XMEMFILE("/ZigBee.jpg", ZigBee_jpg),
SSPEC_RESOURCE_FUNCTION("/button.cgi", activate_cam_controller),
```

SSPEC_RESOURCETABLE_END

//The main programming loop

main()

{

 //initialize board

 brdInit();

 //Initialize the required connection as web server

 sock_init();

 http_init();

 tcp_reserveport(80);

 printf("\nDirect your web browser to http://10.10.6.100/");

 while (1)

 {

 http_handler();

 }

}

Appendix C

Web Interface Codes

Measuring with Bees II

– In Home Power and Water Metering

```
<HTML>
<HEAD>
  <TITLE>Measuring With Bees II</TITLE>
</HEAD>

<BODY topmargin="0" leftmargin="0" marginwidth="0" marginheight="0"
bgcolor="#FFFFFF" link="#009966" vlink="#FFCC00" alink="#006666">

<img SRC="ZigBee.jpg" ALIGN=right width=100 height=100>
<img SRC="usqlogo_line.gif">
<br><img SRC="colourbar.jpg" width=700 height=8>

<h1>Measuring With Bees II</h1>

<font size=4><b>A home power and water monitoring system, which
communicates wirelessly and makes the information accessible via a web
interface.</b> </font>

<br><br><br>
<font size=5>
Activate button to capture Metering Data!
</font>
<A HREF="/button.cgi"> <img SRC="button.gif" width=80 height=80></A>

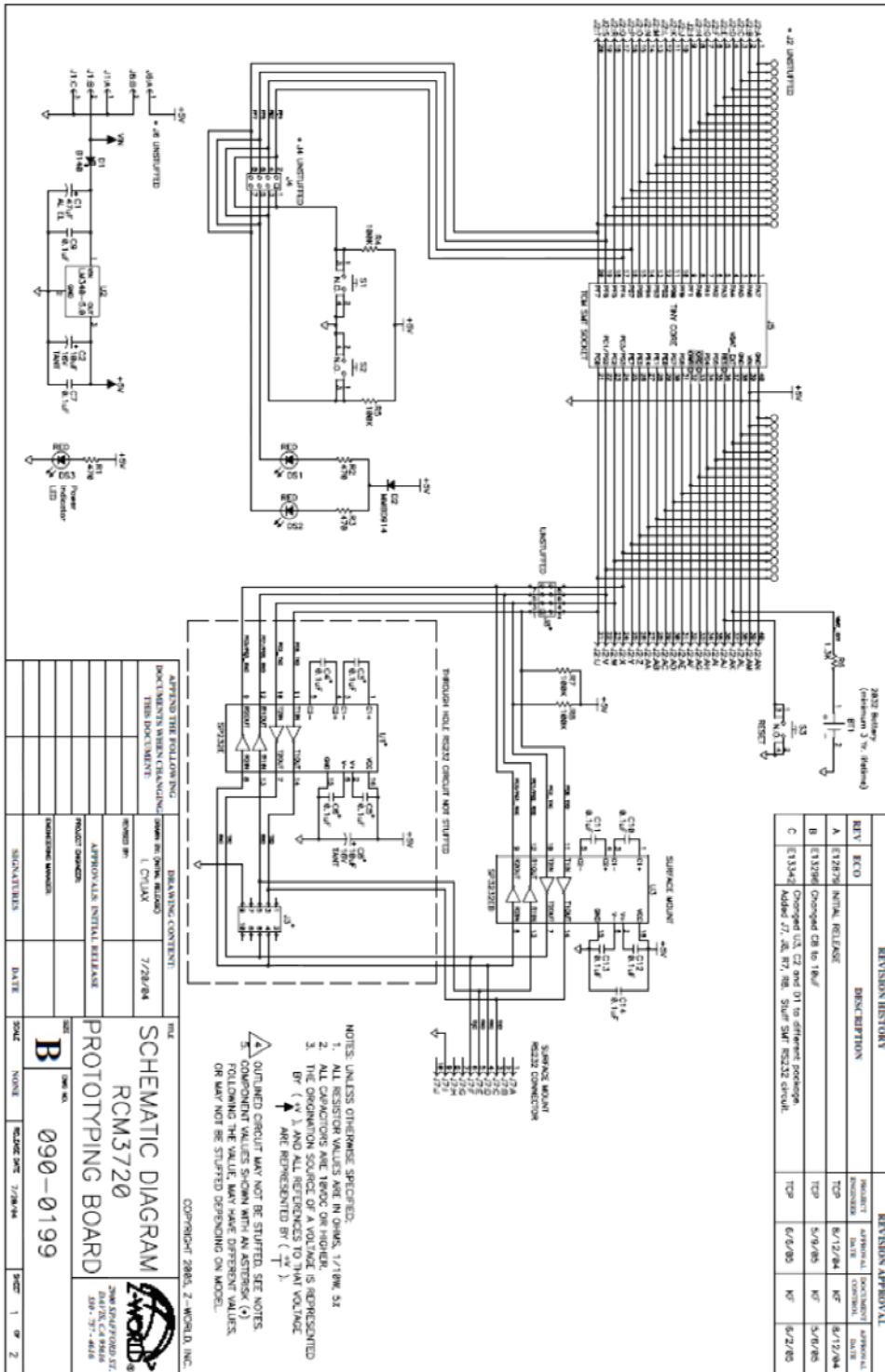
<br><br><br>
<p align=right>
<font size=2>Link Up with Rabbit Processor Successfully!!!</font>
<img SRC="USQlogo.jpg" ALIGN=ABSBOTTOM width=80 height=100>
<br><img SRC="colourbar.jpg" width=700 height=8>
</p>
```


Appendix D

Schematic Diagrams

Appendix D.1

RCM3720 Prototyping Board Schematic Diagram



Appendix D.2

RF Module Interface Board Schematic Diagram

