

DEVELOPMENT OF WIRELESS TYPE K THERMOCOUPLE TEMPERATURE
MEASUREMENT

NOR ASIAH BINTI AHMAD SUKORI

Thesis submitted in fulfilment of the requirements for the award of the degree of
Bachelor Degree of Electrical Engineering (Electronics)

Faculty of Electrical Engineering

UNIVERSITI MALAYSIA PAHANG

NOVEMBER 2010

CHAPTER 1

INTRODUCTION

1.1 Introduction to temperature measurement using ZigBee

Temperature is a common variation, and temperature monitoring is also an important and basic part of industry field. The traditional temperature monitor system often needs a master node connected with monitor computer and some slave nodes distributed in the environment. And master and slave nodes are connected by such as RS485/RS232 cable to form a monitoring network, data or signal are transmitted between them. However, some factors in complex industrial environment may cause corrosion of cable, such as oil stain. This situation may affect the quality of communications, even lead to the failure. Besides, the rate of change in the temperature monitor system is not fast, which leads to the data transmitting rate is slow, and data quantity is not large, therefore, the temperature monitoring system could adopt wireless transmission technology. ZigBee is an open specification that enables low power consumption, low cost and low data rate (250kb/s) for short-range wireless connections between various electronic devices, which is a proper scheme applied in temperature monitoring system.

The ZigBee standard is built on top of the IEEE 802.15.4 standard, which defines the physical and MAC layer for low rate wireless personal area networks. It

also supports functionalities for channel selection, link quality estimation, energy measurement and clear channel assessment. ZigBee also defines the NWK layer, the application layer and the security layer which are used to form network and ensure security of wireless data transmission.

Two types of devices RFD (Reduced Function Device) and FFD (Full Function Device) could be recognized by ZigBee architecture, and can build three kinds of network topology structure, star topology, tree topology and mesh topology. Each ZigBee network only has a coordinator, which acts as the administrator and takes care of organization of the network. Only the FFD defines the full ZigBee functionality and can become a network coordinator. The RFD has limited resources and does not allow some advanced functions. All main characters of ZigBee are analyzed above is suitable for forming an industrial wireless temperature monitoring system.

1.2 Objectives of the project

There are three main objectives of the project which are:

- i. To develop GUI (Graphical User Interface) application using Visual Basic.

Visual Basic will be use as a main programming language.

- ii. To interface the GUI application and the temperature transmitter output.

The interface process can be done using the Zigbee. ZigBee will be used to interface between instrument and computer.

- iii. To monitor the temperature measurement directly by software application.

Temperature measurement is the way that can be used to measure temperature where data from measurement process can be directly used for other purpose.

1.3 Problem statement

In the industrial, temperature measurement is one of the most frequently measured parameters in process system. Temperature detectors have become industry standards for simple and cost-effective temperature measurement. However, achieving such measurement in an accurate, reliable and cost-effective manner is a challenging problem. If station is far away from the workplace, it is difficult to collect and monitor temperature changes. It wastes time to take and check temperature reading at plant station. They also need to analysis and monitor the data everyday or weekly to make sure the instrument in good condition.

1.4 Scope of project

This project involves designing the software application to analysis the data using Microsoft Visual Basic 2008 Express Edition. Visual Basic 2008 used to develop GUI (Graphical User Interface) application. The interfacing process between GUI application and the temperature transmitter instrument can be done using Zigbee wireless technology. Thermocouple type K is used as a primary transducer to detect temperature changes in Isotech Jupiter temperature bath. Besides Isotech Jupiter and thermocouple, temperature transmitter, hart communicator and digital manometer also will be use as a temperature transmitter instrument for this project.

CHAPTER 2

LITERATURE REVIEW

2.1 Thermocouple

Thermocouples are widely used in industry and in testing and research laboratories for measuring temperature. Thermocouple techniques have been developed to meet the specific requirements of many applications. One of those applications is the measurement of metallic surface temperatures [1]. Two thermocouple wires are attached to the metallic surface, and that metallic surface completes the thermoelectric circuit. Possible advantages of physically separating the junction include the following:

- The effects of the thermocouple wires on the temperature being measured are minimized. For example, conduction of heat away from a point on the surface is less from the attachment of a single wire than from a pair.
- If a standard thermocouple junction is attached to the surface and the thermocouple wires are twisted above the junction, this can sometimes cause an inadvertent short between the wires. The temperature measurement is then made at the location of that short and not on the surface as intended. The separated junction method may reduce the likelihood of such an error [2].

In 1826, Thomas Seebeck discovered that a circuit composed of two dissimilar metals will generate an EMF if the junctions at the ends of those metals are kept at different temperatures. Figure 2.1 shows a thermocouple circuit consisting of two metals A and B, with junctions at temperatures T (test junction) and T_r (reference junction).

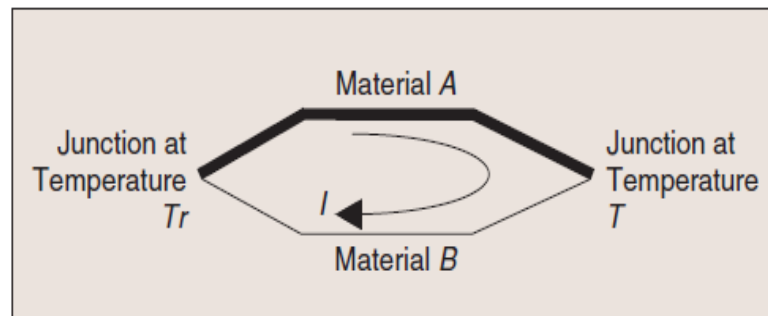


Figure 2.1: Thermocouple circuit of materials A and B [2]

If the reference junction temperature remains constant, the Seebeck EMF (ϵ_{AB}) is a function of the test junction temperature. As the temperature difference between the two junctions increases, ϵ_{AB} increases and a current (I) flows through the circuit. If the junction at T_r is opened and connected to a thermocouple meter, as shown in Figure 2.2 the measured voltage will be a function of temperature T as defined by ϵ_{AB} . This Seebeck voltage is nonlinear with respect to the temperature difference $T - T_r$ [3].

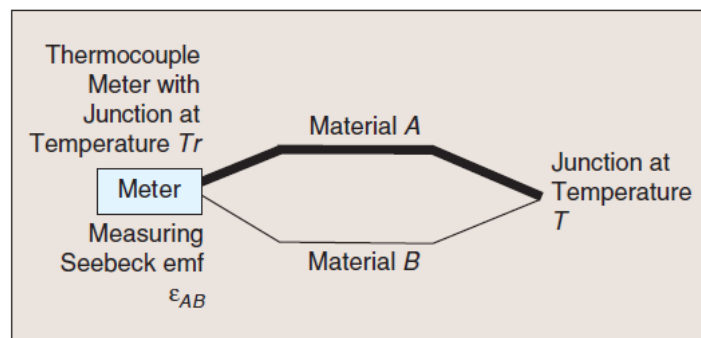


Figure 2.2: Thermocouple circuit with junction of T_r opened [3]

The Seebeck coefficient for materials A and B , also known as the thermoelectric power, is defined in equation (2.1):

$$S_{AB} = \lim_{\Delta T \rightarrow 0} \frac{\Delta \varepsilon_{AB}}{\Delta T} = \frac{d\varepsilon_{AB}}{dT} \quad (2.1)$$

where T is temperature, and ε_{AB} is the Seebeck EMF [3,4].

2.2 Thermocouple type

Thermocouples are available in different combinations of metals or calibrations. The four most common calibrations are J, K, T and E. Each calibration has a different temperature range and environment, although the maximum temperature varies with the diameter of the wire used in the thermocouple.

Some of the thermocouple types have standardized with calibration tables, colour codes and assigned letter-designations. The ASTM Standard E230 provides all the specifications for most of the common industrial grades, including letter designation, colour codes, suggested use limits and the complete voltage versus temperature tables for cold junctions maintained at 32 °F and 0 °C [5].

There are four classes of thermocouples:

- The home body class (called base metal)
- the upper crust class (called rare metal or precious metal)
- the rarified class (refractory metals)
- The exotic class (standards and developmental devices)

Table 2.1 shows the thermocouple type. The home bodies are the Types E, J, K, N and T. The upper crusts are types B, S, and R, platinum all to vary percentages.

Table 2.1: Thermocouple type [5]

J	Iron – constantan	-190 °C to 760 °C
T	Copper - constantan	-200 °C to 371 °C
K	Chromel – alumel	-190 °C to 1260 °C
E	Chromel - constantan	100 °C to 1260 °C
S	90% platinum + 10% rhodium – platinum	0 °C to 1482 °C
R	87% platinum + 13% rhodium – platinum	0 °C to 1482 °C

2.3 Visual Basic

Microsoft Visual Basic is designed for graphical user interface (GUI) programming. It is not a general purpose programming language. For example, you would not want to write a compiler in Visual Basic. It is not a procedural language. Microsoft calls Visual Basic an event-driven programming language. Since the sequence of events that a user chooses is practically unlimited, the programmer must code each event independently in such a way that it can interact with other events. Event-driven programming is ideally suited for object oriented programming techniques.

Visual Studio provides an environment that's common to all languages, which is known as an integrated development environment (IDE). The purpose of the IDE is to enable developer to do as possible with visual tools, before writing code. The IDE provides tools for designing, executing and debugging [6]

Visual Basic objects have properties, methods, and events. Properties define the identity and state of an object. Methods and events define the behaviour of an object [7].

Microsoft defines properties, methods, and events as follows:

1. Property: A named attribute of an object. Properties define object characteristics, such as size, colour, screen location, or whether the object is enabled. Every Visual Basic object has a property called name.

2. Event: An action recognized by an object, such as clicking the mouse or pressing a key. You can write code to respond to events. Events can occur as a result of a user or program action, or they can be triggered by the system.
3. Method: A Subroutine or Function that operates on an object [7].

2.4 PIC Microcontroller

The PIC microcontroller family is manufactured by Microchip Technology Inc. Currently; they are one of the most popular microcontrollers, used in many commercial and industrial applications. Over 120 millions devices are sold each year.

The PIC microcontroller architecture is based on a modified Harvard RISC (Reduced Instruction Set Computer) instruction set with dual – bus architecture, providing fast and flexible design with an easy migration path from only 6 pins to 80 pins and from 384 bytes to 128 Kbytes of program memory [8].

PIC microcontrollers are available with many different specifications depending on:

- Memory Type
 - Flash, OTP(one–time–programmable), ROM, ROMless
- Input – Output pin count
- Memory Size
- Special Features
 - LCD, Motor Control, Radio Frequency, CAN, USB.

Although there are many models of PIC microcontrollers, the nice thing is that they are upward compatible with each other and a program developed for one model very easily, in many cases with no modifications, be run on other models of the family .The basic assembler instruction set of PIC microcontrollers consists of only 33 instructions and most of the family members (except the newly developed devices) use the same instruction set [8].

All PIC microcontrollers offer the following features;

- RISC instruction set with only a handful of instructions to learn
- Digital I/O ports
- On – chip timer with 8 – bit prescaler
- Power – on reset
- Watchdog timer
- Power – saving SLEEP mode
- High source and sink current
- Direct, indirect, and relative addressing modes
- External clock interface
- RAM data memory
- EPROM or Flash program memory

2.4.1 PICAXE 18X

The PICAXE 18X is a PIC16F88 microcontroller loaded with a Basic Stamp style P-Code interpreter. The chip's functionality and development environment are very similar to a Basic Stamp 2's. The quality of documentation is good. The chip is programmed using a minimal version of BASIC or a unique flowcharting system.

It only has 16 bytes of variable space. However, it has another 256 bytes of "data memory" for temporary storage. The chip has enough program space for approximately 600 BASIC commands. It does not have a built-in voltage regulator - but it can operate anywhere from 2v to 5.5v

The PICAXE-18X offers:

- 600 lines memory
- 8 inputs
- 8 outputs
- I2C Interfacing for easy peripheral connecting
- 8/10-bit Analog-to-Digital converters (ADC)
- 8MHz maximum operation speed (4MHz normally)

- Supports
 - Interrupts
 - 12-Bit Digital temperature sensors
 - Servo control
 - Keyboard input
 - IR transmit/receive
 - Plays user-defined musical tones
 - PWM Motor control
 - Input Pulse counting
 - Serial output & debugging via programming cable
 - Higher baud rate for serial work
- Based on the PIC 16F88 IC

The 'PICAXE' system is a powerful, low cost microcontroller programming system designed to simplify educational and hobbyist use of microcontrollers. PICAXE chips can be programmed in a graphical 'flow-chart' environment or in easy to understand BASIC. Figure 2.3 shows the PICAXE -18X pins configuration.

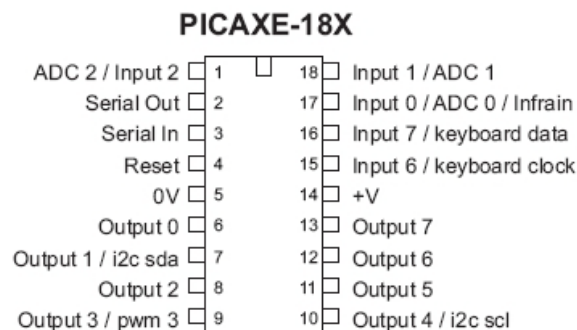


Figure 2.3: PICAXE-18x pins

2.5 Wireless Measurement System based on ZigBee Transmission Technology.

The ZigBee (IEEE 802.15.4) is a new technology that permits the implementation of Wireless Personal Area Networks (WPAN). It is very suitable for wireless sensor networks due to the very low power consumption. This was one of the reasons why it was chosen for the implementation of the system presented in this paper [10]. Summarizing, the main advantages of ZigBee in comparison with other technologies such as Bluetooth or WiFi are the following:

- flexible network architecture
- low cost
- low power consumption
- large number of nodes (≤ 65.536)
- compatibility of equipments from diverse producers

The main disadvantages are:

- low transmission speed
- the existence of a single point of failure represented by
ZigBee coordinator

The ZigBee technology allows the operation in so-called mesh networks that are low cost, self-organizing networks of ZigBee devices. The components of the mesh networks can operate over extended periods of time, even years, without changing the original battery. The ZigBee devices operate in unlicensed radio frequency bands (ISM). These unlicensed bands are not the same in all regions of the world; those the ZigBee devices can operate in three frequency bands centered on 868, 915 and 2400 MHz. The most advantageous frequency band is at 2400 MHz because of higher data rate (250 kb/s) and worldwide availability. In the 2402–2480 MHz frequency band is used offset quadrature phase-shift keying (O-QPSK) modulation technique. In the 868 and 902–928 frequency bands are used DSSS (Direct sequence spread spectrum) and BPSK (Binary phase shift keying) [11].

In the structure of ZigBee networks the devices can be of three types: ZigBee Coordinator, ZigBee Router and the ZigBee End Device.

1. *Zigbee Coordinator (ZC)* – has the function to initiate the network structure by configuring the channels and establishing an ID for that network. It stores the security keys and is capable to bridge to other networks the devices from its own network.
2. *Zigbee Router (ZR)* – act as an intermediate device, its main function is to participate in multi-hop / mesh routing of network messages. It maintains a routing table and manages local address allocation.
3. *Zigbee End Device (ZED)* – does not participate in routing. It contains only the functionality to communicate with its parent node (the network coordinator or a router).

With these types of devices, a ZigBee network can be configured in three topologies: star, a mesh (peer-to-peer), or a cluster tree. Figure 2.4 shows the Topology types for wireless sensor networks.

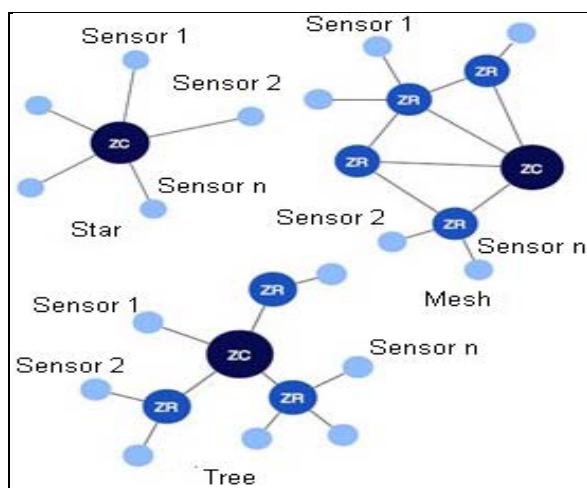


Figure 2.4: Topology types for wireless sensor networks [11]

The radiation level measurement system is composed by a controller board and multiple remote boards that are connected to the sensors through analog signal conditioning boards. The controller board and remote boards was realized around JN5139 wireless microcontroller a high performance, low power device that in

suitable for IEEE802.15.4 and ZigBee applications. The JN5139 wireless microcontroller integrates a 32-bit RISC processor with a 2.4GHz IEEE802.15.4 transceiver. It contains also read only memory (ROM) and random access memory (RAM), a 12 bit analog to digital convertor, an 11 bit digital to analog convertor timer and few comparators. The microcontroller disposes also by UART and SPI interfaces that offer a very useful support for interconnections with other devices and systems. In the design of the proposed system was used this characteristics of this complex device to realize a reliable system. In the design of the proposed system was used this characteristics of this complex device to realize a reliable system [11].

2.6 Previous Research

There are a lot of researches before this that has similarity with this project. Those researches can be reference for this project so that, this project can achieve the objectives.

2.6.1 Temperature Measurement Technology with Sapphire Fiber Blackbody Sensor

This research was done by Gao Wang and Xiao Yan Wang from Key Laboratory of Instrumentation Science & Dynamic Measurement (North University of China) Ministry of Education during 2010 International Conference on Optics, Photonics and Energy Engineering. This paperwork discussed about the new sapphire fiber high temperature sensor sputtered the ceramics film. The system includes blackbody sapphire fiber sensor, taper fiber, transmission optical fiber, narrow bandpass filter, integrated photoamplifier and data acquisition device. The adoption of sapphire fiber makes the measured high temperature up to 2000. The static calibration is performed by comparing the outputs between the calibrated Tungstenrhenium thermocouple and the sapphire fiber sensor, both placed in a stability high temperature field. The dynamics character is done by focusing a CO2 laser with high peak power into the sensor. Dynamics respondent curve shows that its respondent time is less than 20 ms, which is suitable for measuring the transient

high temperature such as the burning and exploration of powders as well as the firing of engine [12].

Accurate measurement of transient high temperature is essential for the design, operation and control of a range of industrial processes, particularly for those transient processes such as the burning and exploration of powders as well as the firing of engine in the field of national defence. Many of these processes involve harsh environments, such as strong shock, high pressure, chemical corrosion gases, toxicity, strong electromagnetic interference, and high-energy radiation exposure. These extreme physical conditions often prevent conventional temperature sensors and measurement devices, such as high temperature thermocouple, optical and acoustic pyrometers, from being used or make them difficult to be used [13].

Figure 2.5 shows the transient high temperature sapphire fiber sensor with a cylindrical blackbody cavity and its metal shell. The wall of the cavity is a thin layer of thermostable film ceramics sintered. Since the film is very thin, its heat capacity is very small and there is no temperature gradient inside. Because the absorption within the sapphire fiber is also small, we can take this small fiber cavity as a cylindrical hollow cavity.



Figure 2.5: Sapphire fiber blackbody cavity and its metal shell [13]

The measuring system consists of blackbody sapphire fiber sensor, tapered fiber, coupler (with narrow band-pass filter), transmission optical fiber, photoamplifier, data acquisition device and software.

According to the Plank' law the emission of spectral radiant flux ϕ (A, T) of the sapphire fiber blackbody cavity at the temperature T is given in equation 2.2:

$$\phi(\lambda, T) = \frac{ac_1}{\lambda^5} [\exp(c_2/\lambda T) - 1] \quad (2.2)$$

where a is the area of the sapphire fiber blackbody cavity, c and C2 are the first and second radiation constants respectively ($c = 3.74 \times 10^{-16} \text{ Wmz}$, $c_2 = 1.43879 \times 10^{-2} \text{ mK}$).

2.7.2 Fuzzy Logic-Based Digital Filtering for Temperature Measurement via K-type Thermocouple

This paper was presented during Computational Intelligence for Measurement Systems and Applications, 2003. In this paper, the noise on the temperature data measured via a K-type thermocouple mounted on the lab set suitably realized for the purpose of measuring thermal expansion of metal samples is reduced by using two cascade band-pass filters (BPF) whose bandwidth can be set by taking advantage of Fuzzy Logic Controller's (FLC) relative error membership function [15].

A heater driver circuit is used to control the inner furnace temperature. The triac of 40 A used in circuits shown in figure 2.6 is drive by TCA785 phase control IC. It produces noise on the temperature data. It effects the calculation of relative error (e) and relative error change (Ae) membership functions on FLC directly. A Fuzzy-Logic based digital filter is applied to acquired temperature to reduce the noise. In design of FLC, three variables such as relative error (e), relative error change (Ae) and reference (r) are fuzzified and a rule base composed of 90 rules is utilized. Relative error is defined by using 6 sets, relative error change by 5 sets, reference by 3 sets [16, 17].

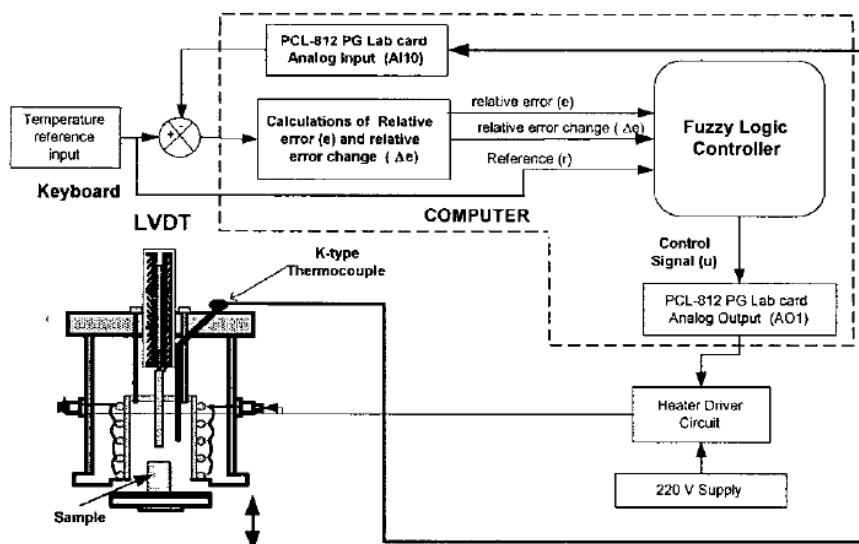


Figure 2.6: Realized FLC system [17]

Because of the noise on the acquired temperature via a K-type thermocouple, the performance of FLC was very low. Two cascade BPFs having settable bandwidth were used to reduce the noise and to increase the performance of FLC. The change of rate of inner temperature is not the same during the experiment. Therefore BPF's bandwidths must be table. The temperature rising curve was divided into 11 sections by using the relative error membership function (REMF). It is shown in Fig. 2.7. Each section has its own parameters such as bandwidth [18].

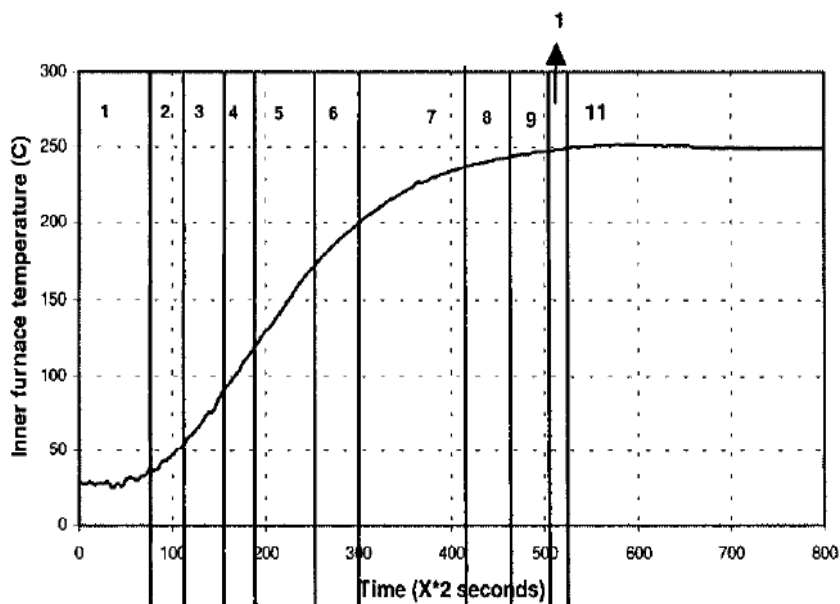


Figure 2.7: Dividing into different sections of the rising curve by using REMF [18]

CHAPTER 3

METHODOLOGY

3.1 Hardware

This project is used Thermocouple Type K as a temperature sensor where it detects temperature changes in Isotech Jupiter temperature bath. Besides that, HART 375 field communicator also being used in doing the measurement. Zigbee wireless system is use to interface between computer and temperature instrument.

3.2 Instrument

Figure 3.1 shows the picture of the basic instrument connection where the instrument consists of Isotech Jupiter 650B, Yokogawa Temperature transmitter PT100, and Hart 375 Field Communicator. Thermocouple type K is used as an input device to detect temperature changes. Type K (chromel – alumel) is the most commonly used general purpose thermocouple. It is inexpensive and owing to its popularity, available in a wide variety of probes. They are available in the -200 °C to +1350 °C range. Isotech Jupiter 650B is an artificial heater used to simulate temperature changes. Thermocouple will detect the temperature changes and transmit the signal to Yokogawa Temperature Transmitter .Then, Yokogawa Temperature Transmitter will convert the signal to current value and current can be read by Hart 375 Field Communicator.



Figure 3.1: The basic instrument connection

3.2.1 The HART 375 Field Communicator

The 375 Field Communicator has established itself as the standard in handheld communicators. Figure 3.2 shows the 375 Field Communicator where it runs on Window CE, a robust, real-time, operating system. The touch screen display uses transfective technology, making it easy to read in both bright sunlight and in normal lighting. To make sure all conditions are covered, a multi-level backlight is included, allowing the display to be viewed in those areas of plant with dim light. The Graphics Option available for the 375 Field Communicator uses powerful EDDL technology that allows you to read data from field devices in a graphical manner. Charts, graphs, gauges, and product images are just a few of the ways in which important device data can be displayed using the Graphics Option.



Figure 3.2: 375 Field Communicator [19]

The touch sensitive display and large physical navigation buttons provide for efficient use both on the bench and in the field. The weight of the 375 Field Communicator is evenly distributed for comfortable one-handed operation in the field. The 375 Field Communicator has plenty of memory to allow for future expansion. It has 32 MB of application memory and 256 MB of memory on its System Card. The Configuration Expansion Module holds hundreds of device configurations, and it is externally accessible.

3.2.2 Thermocouple type K

Type K thermocouples are the most common and universal type of temperature sensors today. Used for their sensitivity and wide temperature range of -200°C to 1350°C , type K thermocouples are considered general purpose devices because of their many uses in common, everyday applications. They usually have a high sensitivity, about $41\mu\text{v}/^{\circ}\text{C}$. Type K exhibits high thermo power and good resistance to oxidation but remains inexpensive while lasting a long time. They are used in all sorts of different heat and temperature applications, since they have good temperature precision and a wide measurement range. This kind of thermocouple is popular and available in a wide variety of probes.

Since it has such a high temperature range, type K are especially used within the steel and iron fabrication industries. They measure the metal's chemical composition and temperature during machining or casting to know

when it has melted to a certain temperature and is ready to form. Other industrial applications, which use type K thermocouples' temperature range, include kilns, ovens, plastic extrusion machines, pressure chambers, water tanks, heat exchangers and parts washers. Heated appliances like gas stoves also use type K thermocouples to keep from overheating, which is a safety concern and could cause damage. In the home, standard thermostats and temperature switches are also monitored by type K thermocouples because they are so inexpensive and easy to install. Figure 3.3 shows the temperature type K.



Figure 3.3: Thermocouple type k

All type K thermocouples are made of a series of thin, bendable wire constructions. Each wire is made of either chromel, which is a chromium and nickel alloy, or alumel, which is an aluminum and nickel alloy. They are joined with digital devices for fast, easy readout and always use metal or ceramic protective tubes, especially when reducing atmospheres because type K performs poorly in this application. They are considered homebody thermocouples, as opposed to uppercrust, rarified and exotic. To differentiate between these different types, the wires are always labeled with a colored connector; type K is yellow. All thermocouples are divided into temperature capacity categories. Type K, because they measure up to around 2000° C, is considered noble metal thermocouples. Because thermocouples are simple in design, fabrication and assembly processes are quick and inexpensive. The wires are all extruded, which involves heating a metal alloy preform billet and forcing it through successively smaller and smaller molds until the

desired wire shape and size is accomplished. The wires are easily bendable, and can twist together without the application of heat or much force.

3.2.3 YOKOGAWA Temperature Transmitter

A temperature transmitter is a device used to sense a temperature and transmit an output representative of the sensed temperature. Process control transmitters are used to measure process parameters in a process control system. Temperature transmitters are used in controlling industrial processes by sensing a temperature of the process and transmitting the information to a remote location. A process temperature transmitter provides an output related to a sensed process substance temperature. The temperature transmitter output can be communicated over the loop to a control room, or the output can be communicated to another process device such that the process can be monitored and controlled. Figure 3.4 shows the YOKOGAWA Temperature Transmitter. The temperature transmitter injects a current into the temperature sensor and the resultant voltage drop across the temperature sensor is used to measure resistance. The voltage is converted into a digital format using an analogue to digital converter and provided to a microprocessor. The microprocessor converts the measured voltage into a digital value representative of temperature. The temperature transmitter generally includes housing and a temperature probe which attaches to the housing. In order to monitor a process temperature, the transmitter includes a sensor, such as a resistance temperature device (RTD) or a thermocouple. An RTD changes resistance in response to a change in temperature. By measuring the resistance of the RTD, temperature can be calculated. A thermocouple provides a voltage in response to a temperature change.



Figure 3.4: YOKOGAWA Temperature Transmitter

Typically, the temperature transmitter is located in a remote location and coupled to a control room over a 4-20 mA current loop. A temperature sensor is placed in the process fluid and provides an output related to temperature of the process fluid.

3.2.5 ZigBee

ZigBee and IEEE 802.15.4 are standards-based protocols that provide the network infrastructure required for wireless sensor network applications. 802.15.4 defines the physical and MAC layers, and ZigBee defines the network and application layers. For sensor network applications, key design requirements revolve around long battery life, low cost, small footprint, and mesh networking to support communication between large numbers of devices in an interoperable and multi-application environment.

The ZigBee standard was developed to address the following needs:

- Low cost
- Secure
- Reliable and self healing
- Flexible and extendable
- Low power consumption
- Easy and inexpensive to deploy
- Global with use of unlicensed radio bands

ZigBee is the only standards-based technology that addresses the unique needs of most remote monitoring and control sensory network applications.

3.2.5.1 ZigBee Network

Figure 3.5 shows the ZigBee Network Coordinator. ZigBee networks include the following device types:

- Coordinators

This device starts and controls the network. The coordinator stores information about the network, which includes acting as the Trust Center and being the repository for security keys.

- Routers

These devices extend network area coverage, dynamically route around obstacles, and provide backup routes in case of network congestion or device failure. They can connect to the coordinator and other routers, and also support child devices.

- End devices

These devices can transmit or receive a message, but cannot perform any routing operations. They must be connected to either the coordinator or a router, and do not support child devices.

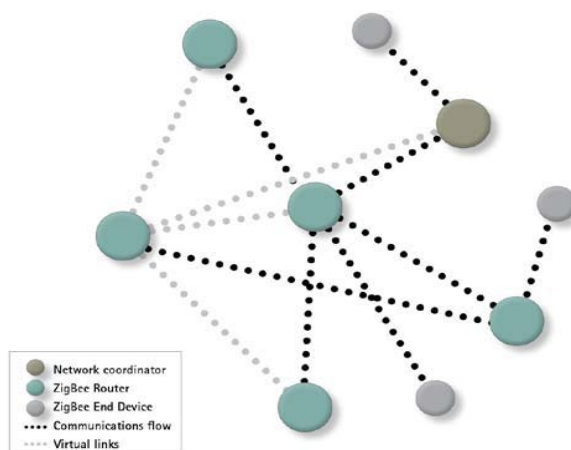


Figure 3.5: Network Coordinator [20]

3.2.5.2 OSI Overview

The Open System Interconnection (OSI) reference model, was developed by the International Organization for Standardization (ISO) as a model for the computer protocol architecture, and as a framework for developing protocol standards. The entire point of the model is to separate networking into several distinct functions that operate at different levels. Each layer is responsible for performing a special task or set of tasks, and dealing with the layers above and below it. An illustration of the general OSI-model and where ZigBee is defined in the model can be seen in Figure 3.6

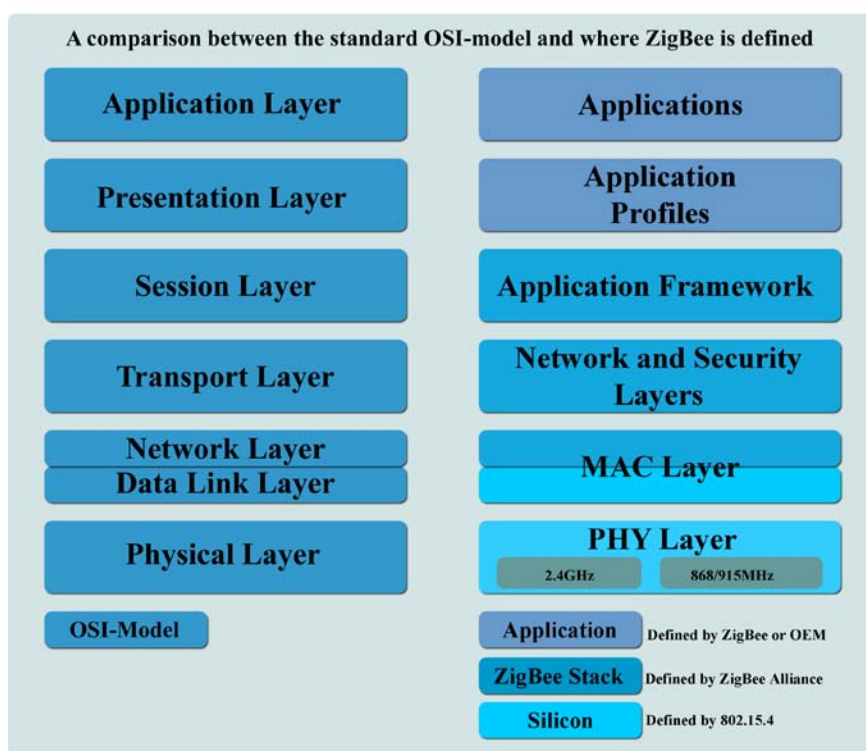


Figure 3.6: OSI model [20]

3.2.5.3 Application (APL) Layer

The top layer in the ZigBee protocol stack consists of the Application Framework, ZigBee Device Object (ZDO), and Application Support (APS) Sublayer.

3.2.5.4 Application Framework

Provides a description on how to build a profile onto the ZigBee stack to help ensure that profiles can be generated in a consistent manner. It also specifies a range of standard data types for profiles, descriptors to assist in service discovery, frame formats for transporting data, and a key value pair constructs to rapidly develop simple attribute-based profiles.

3.2.5.5 Application Objects

Software at an endpoint that controls the ZigBee device. A single ZigBee node supports up to 240 application objects. Each application object supports endpoints numbered between 1 and 240 with endpoint 0 reserved for the ZigBee Device Object (ZDO).

3.2.5.6 ZigBee Device Object (ZDO)

Defines the role of a device within the network (coordinator, router or end device), initiates and/or responds to binding and discovery requests, and establishes a secure relationship between network devices. It also provides a rich set of management commands defined in the ZigBee Device Profile used in ZigBee commissioning. The ZDO is always endpoint zero.

3.2.5.7 ZDO Management Plane

Facilitates communication between the APS and NWK layers with the ZDO. Allows the ZDO to deal with requests from applications for network access and security using ZDP (ZigBee Device Profile) messages.

3.2.5.8 Application Support (APS) Sublayer

Responsible for providing a data service to the application and ZigBee device profiles. It also provides a management service to maintain binding links and the storage of the binding table itself.

3.2.5.9 Security Service Provider (SSP)

Provides security mechanisms for layers that use encryption (NWK and APS). Initialized and configured through the ZDO.

3.3 Visual Basic

Visual Basic is the main language programming for this project. Visual Basic was designed to be easily learned and used by beginner programmers. The language not only allows programmers to create simple GUI applications, but can also develop complex applications. Programming in VB is a combination of visually arranging components or controls on a form, specifying attributes and actions of those components, and writing additional lines of code for more functionality. Since default attributes and actions are defined for the components, a simple program can be created without the programmer having to write many lines of code. Performance problems were experienced by earlier versions, but with faster computers and native code compilation this has become less of an issue.

3.3.1 Start the Visual Basic 2008

1. On the Windows taskbar, click Start, click All Programs, and then click the Microsoft Visual Studio 2008 folder.

The folders and icons in the Microsoft Visual Studio 2008 folder appear in a list.

2. Click the Microsoft Visual Studio 2008 icon.

If this is the first time you are starting Visual Studio, it might take a few minutes to configure the environment. If you are prompted to specify the settings to use, select the Visual Basic development settings. When Visual Studio starts, figure 3.7 show the development environment on the screen with its many menus, tools, and component windows. These windows are sometimes called tool windows.

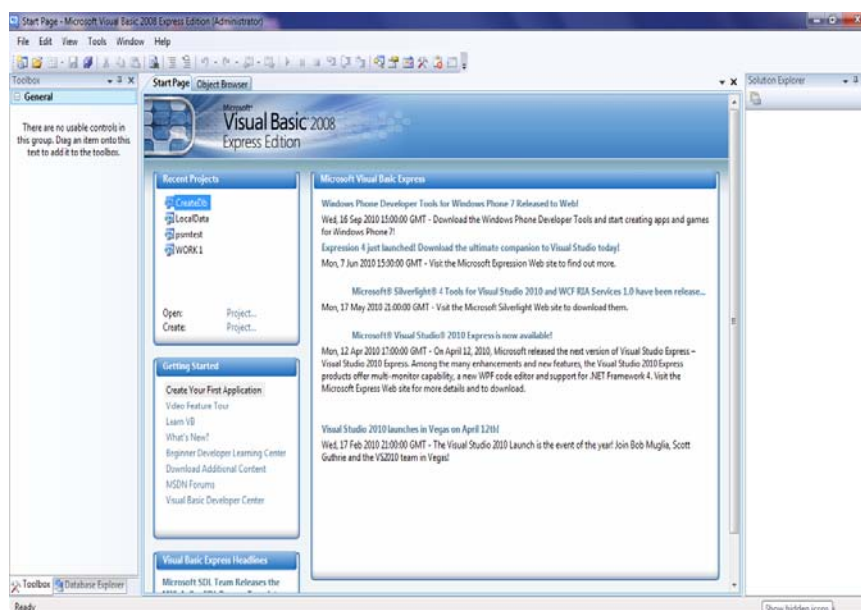


Figure 3.7: Tool windows

The Start Page is a comprehensive source of information about the project, as well as resources within the Visual Basic development community. This is one avenue for receiving new information about Visual Studio after you purchase the software.

Then, from the tool windows go to File and choose New Project. The New Project dialog box opens as shown in the figure 3.8. Select Windows Application and Input project Name.

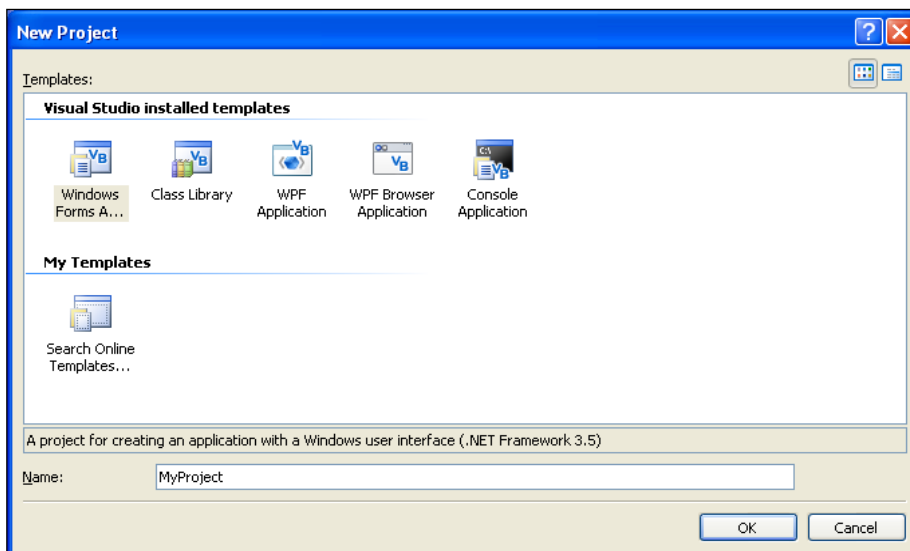


Figure 3.8 : New Project dialog box

3.3.2 The Visual Studio Tools

The Toolbox is a container for all the controls that can be add to a Windows Forms application or a Windows Presentation Foundation (WPF) application. Figure 3.9 shows the Toolbox Window where the Toolbox is located on the left side of the integrated development environment (IDE). If the Toolbox is not visible, click the View menu, and then click Toolbox to display it. The following illustration shows the common controls in the Toolbox.

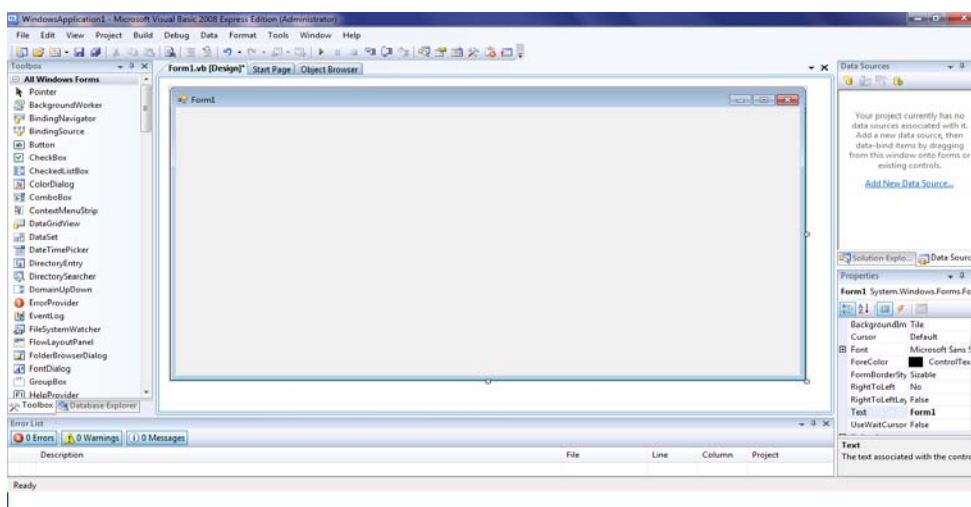


Figure 3.9: Toolbox Window

3.4 X-CTU

X-CTU is a Windows-based application provided by Digi. This program was designed to interact with the firmware files found on Digi's RF products and to provide a simple-to-use graphical user interface to them.

X-CTU is designed to function with all Windows-based computers running Microsoft Windows 98 SE and above. X-CTU can either be downloaded from Digi's Web site or an installation CD. When launched, four tabs across the top of the program can be seen as a Figure 3.10. Each of these tabs has a different function. The four tabs are:

- **PC Settings:** Allows a customer to select the desired COM port and configure that port to fit the radios settings.
- **Range Test:** Allows a customer to perform a range test between two radio.
- **Terminal:** Allows access to the computers COM port with a terminal emulation program. This tab also allows the ability to access the radios' firmware using AT.
- **Modem Configuration:** Allows the ability to program the radios' firmware settings via a graphical user interface. This tab also allows customers the ability to change firmware versions.

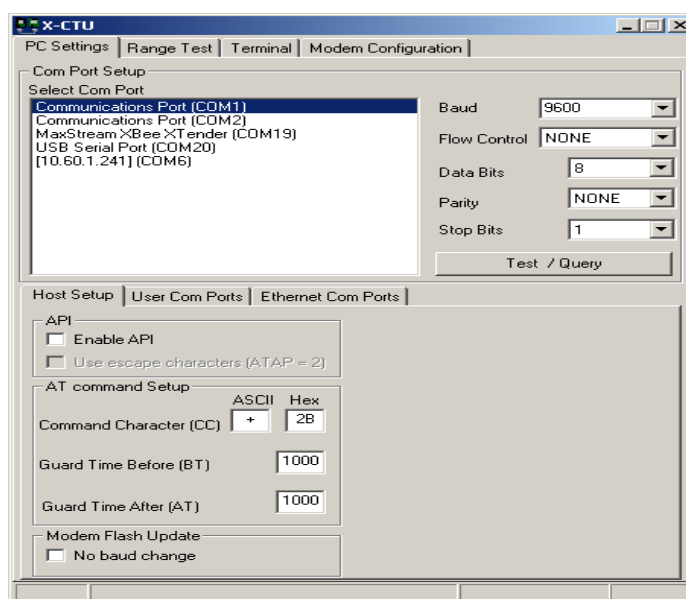


Figure 3.10: X-CTU front program

3.4.1 PC Settings Tab

When the program is launched, the default tab selected is the “PC Settings” tab. The PC Settings tab is broken down into three basic areas: The COM port setup, the Host Setup, and the User Com ports. COM port setup:

- i. The PC settings tab allows the user to select a COM port and configure the selected COM port settings when accessing the port. Some of these settings include:
 - Baud Rate: Both standard and non-standard
 - Flow Control: Hardware, Software (Xon/Xoff), None
 - Data bits: 4, 5, 6, 7, and 8 data bits
 - Parity: None, Odd, Even, Mark and Space
 - Stop bit: 1, 1.5, and 2
- ii. To change any of the above settings, select the pull down menu on the left of the value and select the desired setting. To enter a non-standard baud rate, type the baud rate into the baud rate box to the left. The Test / Query button is used to test the selected COM port and PC settings. If the settings and COM port are correct, the response similar to the one depicted in Figure 3.11.

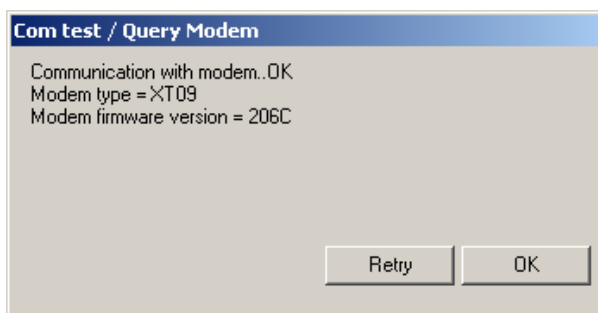


Figure 3.11: Com test /Query Modem

The Host Setup tab allows the user to configure how the X-CTU program is to interface with a radio's firmware. This includes determining whether API or AT command mode will be used to access the module's firmware as well as the proper command mode character and sequence.

The user COM port option allows the user to “Add” or “Delete” a user-created COM port. This is only for temporary use. Once the program has closed, the user-created COM port will disappear and is no longer accessible to the program.

3.4.2 Range Test Tab

The range test tab is designed to verify the range of the radio link by sending a user-specified data packet and verifying the response packet is the same, within the time specified. For performing a standard range test, please follow the steps found in most Quick Start or Getting Started Guides that ship with the product.

3.4.2.1 Packet Data and Size

By default, the size of the data packet sent is 32 bytes. This data packet specified can be adjusted in either size or the text sent. Modifying the size of the packet sent, change the value next to the “Create Data” box and click on the “Create Data” button as shown in figure 3.12. If want to change the data sent, delete the text in the transmit window and place the desired text. By modifying the text, data packet size, packet delay and the data receive timeout; the user is able to simulate a wide range of scenarios.

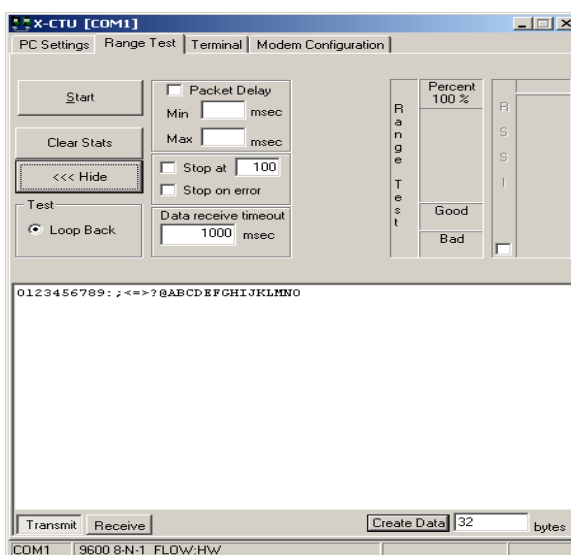


Figure 3.12: Range test tab

3.4.2.2 RSSI

The RSSI option of the X-CTU allows the user to see the RSSI (Received Signal Strength Indicator) of a received packet when performing a range test.

3.4.2.3 API Function

The X-CTU also allows the user to test the API function of a radio during a range test. To perform a range test with the API function of the radio, follow the steps outlined below:

- 1: Configure the Base with API enabled and a unique 16 bit or 64 bit source address.
- 2: Configure the remote radio with a unique source address and set the Destination address to equal the Base radio's source address.
- 3: Enable the API option of the X-CTU on the PC Settings tab and connect the base radio to the PC (See Figure 3).
- 4: Connect the red loopback adapter to the remote radio and place them a distance apart.
- 5: Enter either the 16 bit or 64 bit destination address of the remote radio into the Destination Address box on the Range Test tab as shown in figure 3.13.
- 6: Create a data packet of your choosing by typing in the data in the Transmit box
- 7: To start a Range test, click on Start.

The TX failures, Purge, CCA, and ACK messages will increment accordingly while the range test is performed. To stop a range test, click on the Stop button.

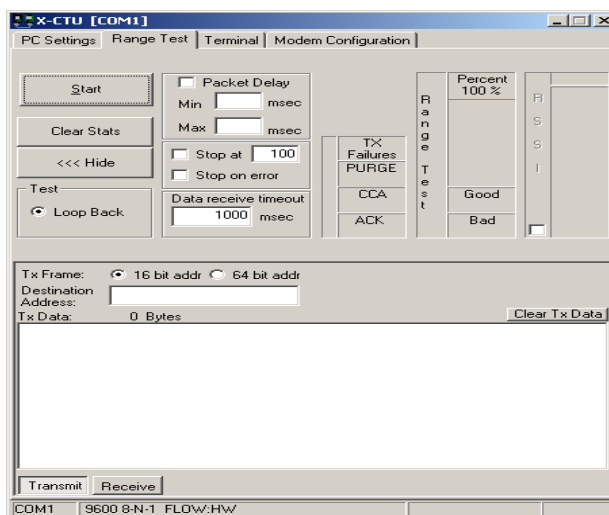


Figure 3.13: Destination address box

3.4.3 The Terminal Tab

The Terminal tab has three basic functions: Terminal emulator Ability to send and receive predefined data packets (Assemble packet) Ability to send and receive data in Hex and ASCII formats (Show/Hide hex)

3.4.3.1 The main terminal window

The main white portion of this tab is where most of the communications information will occur while using X-CTU as a terminal emulator. The text in blue is what has been typed in and directed out to the radio's serial port while the red text is the incoming data from the radio's serial port as shown in figure 3.14

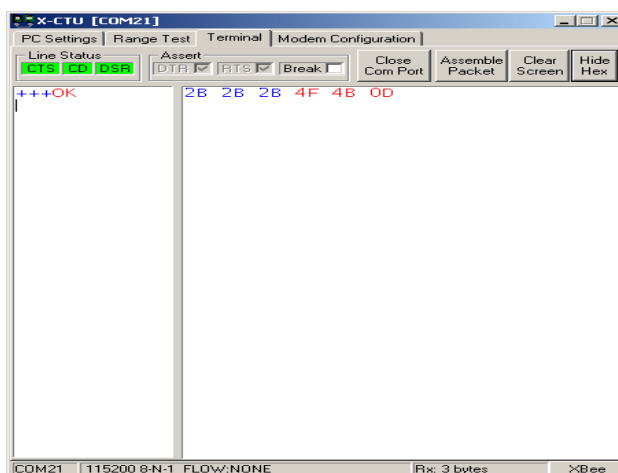


Figure 3.14: The main terminal window

3.4.3.2 Assemble Packet

The Assemble Packet option on the Terminal tab is designed to allow the user to assemble a data packet in either ASCII or Hex characters. This is accomplished by selecting the Assemble packet window and choosing either ASCII (default) or Hex. Once selected, the data packet is assembled by typing in the desired characters as depicted in Figure 3.15.

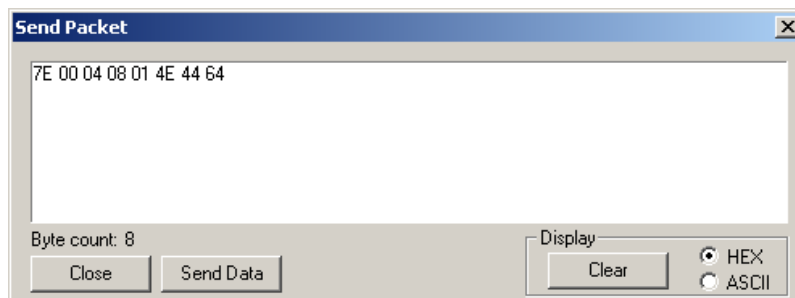


Figure 3.15: Data packet

The **Line Status** indicators show the status of the RS-232 hardware flow control lines. Green indicates the line is asserted while black indicates de-asserted.

The **Break** option is for engaging the serial line break. This can be accomplished by checking or asserting the **Break** option. Asserting the **Break** will place the DI line high and prevent data from being sent to the radio.

3.4.4 Modem Configuration Tab

The Modem configuration tab has four basic functions:

- 1: Provide a Graphical User Interface with a radio's firmware
- 2: Read and Write firmware to the radio's microcontroller
- 3: Download updated firmware files from either the web or from a compressed file
- 4: Saving or loading a modem profile

3.4.4.1 Reading a radio's firmware

To read a radio's firmware, follow the steps outlined below:

- 1: Connect the radio module to the interface board and connect this assembly or a packaged radio (PKG) to the PC's corresponding port (IE: USB, RS232, Ethernet etc.).
- 2: Set the PC Settings to the radio's default settings.
- 3: On the Modem Configuration tab, select "Read" from the Modem Parameters and Firmware section as shown in figure 3.16

3.4.4.2 Making changes to a radio's firmware

Once the radio's firmware has been read, the configuration settings are displayed in three colours:

- Black – not settable or read-only
- Green – Default value
- Blue – User-specified

To modify any of the user-settable parameters, click on the associated command and type in the new value for that parameter. For ease of understanding a specific command, once the command is selected, a quick description along with its limits is provided at the bottom of the screen. Once all of the new values have been entered, the new values are ready to be saved to the radio's non-volatile memory.

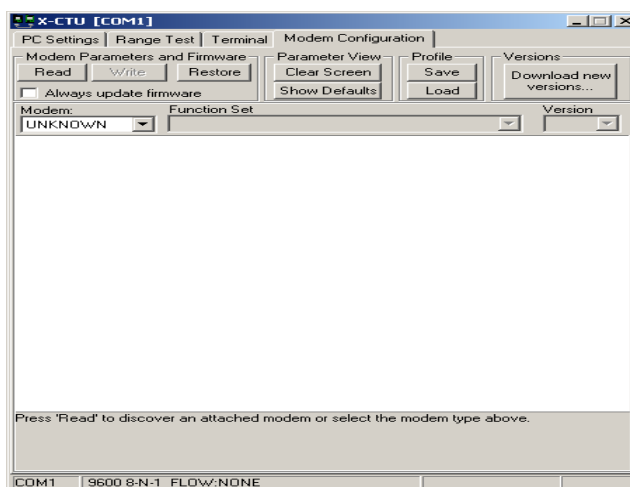


Figure 3.16: Modem Parameters and Firmware section

3.4.4.3 Writing firmware to the Radio

To write the parameter changes to the radio's non-volatile memory, click on the Write button located in the Modem Parameters and Firmware section as shown in figure 3.17 .

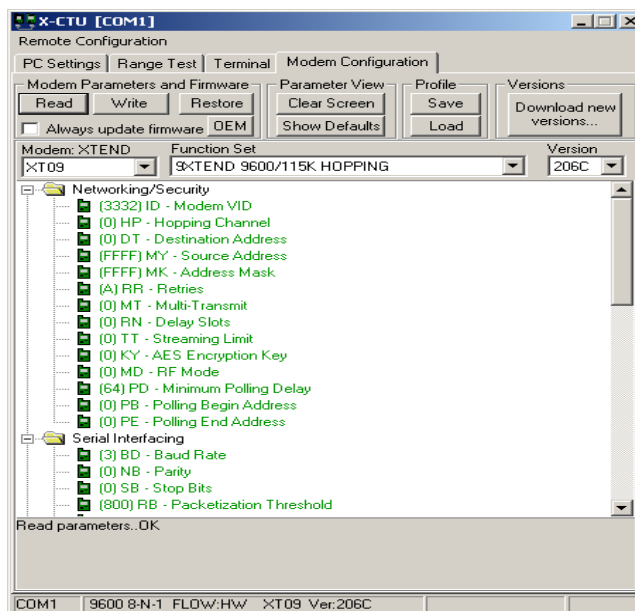


Figure 3.17: Modem Configuration

3.4.4.4 Remote Modem Management

XBee 802.15.4 modules with firmware version 1xCx and above, XBee ZNet 2.5 modules, and XBee ZB modules offer the ability to be configured with over the air commands. With the addition of this new feature, the user is able to configure remote radio parameters with X-CTU or API packets. To use the remote configuration tool, the following is required:

- The radio connected to the PC must be in API mode
- The remote radio must be associated or within range of the base radio

To access remote radios through X-CTU's Modem Configuration tab, perform the steps below:

- Enable API on the PC Settings tab
- Verify the COM port selection and settings

- On the Modem Configuration tab, select the Remote Configuration option on the top left corner of the program
- Select Open Com port
- Select Discover
- Select the desired modem from the discovered node list as shown in figure 3.18
- On the Modem configuration tab, select Read

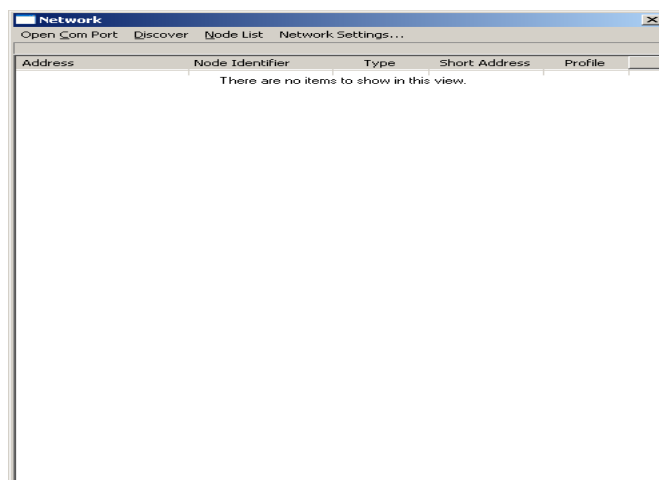


Figure 3.18: Node list

The remote radio's configuration is now displayed on the Modem Configuration tab. At this point, the same options exist with respect to Read and Write parameter changes. Please note that the ability to change firmware versions is still limited to the radio's UART. To clear the discovered node list, click on Node List and Clear.

3.5 Interfacing between hardware and software

Figure 3.19 shows the interfacing between hardware and software. PC is representing GUI application where all the data were send to visual basic via wireless system. The GUI application detected the temperature measurement. Figure 3.14 shows the flow chart of the project.

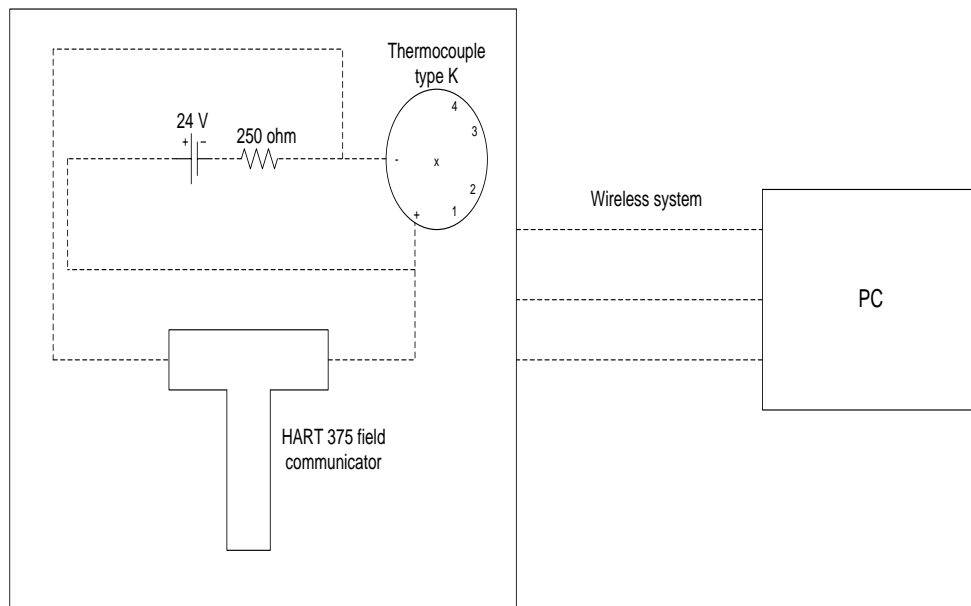


Figure 3.19: Interface between software and hardware

Thermocouple type K inside the Isotech Jupiter used to indicate the temperature. Then, the temperature value will be send to HART 375 field communicator where the temperature value is showed. After that, Xbee from the hardware send the data to other Xbee at the PC and the GUI application will show the temperature value. Zigbee is used to interface between hardware and software. Zigbee is divided into two parts which are transmitter and receiver. Figure 3.20 shows the Xbee transmitter and Xbee receiver.

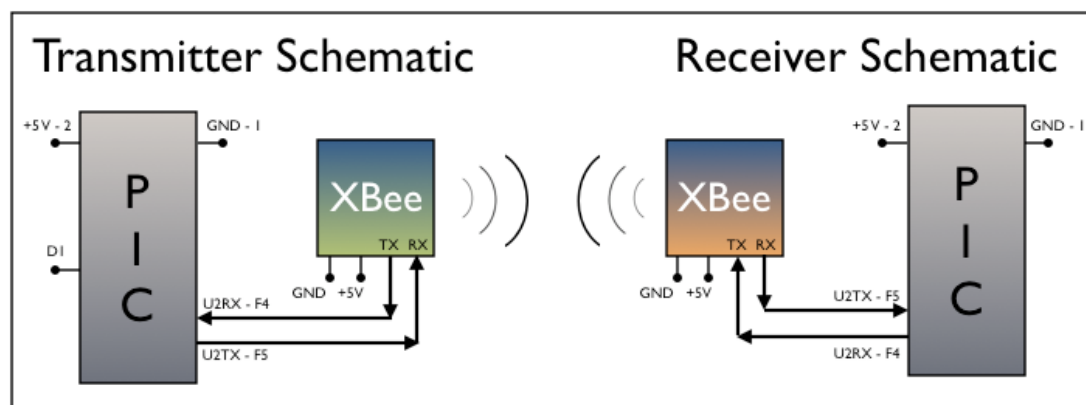


Figure 3.20: Xbee Receiver and Xbee transmitter

3.5.1 Modes of Operation of Xbee

There are two different modes of operations that Xbees can run in. They are:

- Transparent mode
- API mode

3.5.1.1 Transparent mode

By default, XBee/XBee-PRO RF Modules operate in Transparent Mode. When operating in this mode, the modules act as a serial line replacement - all UART data received through the DI pin is queued up for RF transmission. When RF data is received, the data is sent out the DO pin.

Data is buffered in the DI buffer until one of the following causes the data to be packetized and transmitted:

1. No serial characters are received for the amount of time determined by the RO (Packetization Timeout) parameter. If $RO = 0$, packetization begins when a character is received.
2. The maximum number of characters that will fit in an RF packet (100) is received.
3. The Command Mode Sequence (GT + CC + GT) is received. Any character buffered in the DI buffer before the sequence is transmitted.

3.5.1.2 API (Application Programming Interface) mode

During Xbee operation, API (Application Programming Interface) Operation is applied. It is an alternative to the default Transparent Operation. The frame-based API extends the level to which a host application can interact with the networking capabilities of the module. When in API mode, all data entering and leaving the module is contained in frames that define operations or events within the module.

Transmit Data Frames (received through the DI pin (pin 3)) include:

- RF Transmit Data Frame
- Command Frame (equivalent to AT commands)

Receive Data Frames (sent out the DO pin (pin 2)) include:

- RF-received data frame
- Command response
- Event notifications such as reset, associate, disassociate.

The API provides alternative means of configuring modules and routing data at the host application layer. A host application can send data frames to the module that contain address and payload information instead of using command mode to modify addresses. The module will send data frames to the application containing status packets; as well as source, RSSI and payload information from received data packets.

The API operation option facilitates many operations such as the examples below:

- Transmitting data to multiple destinations without entering Command Mode
- Receive success/failure status of each transmitted RF packet
- Identify the source address of each received packet

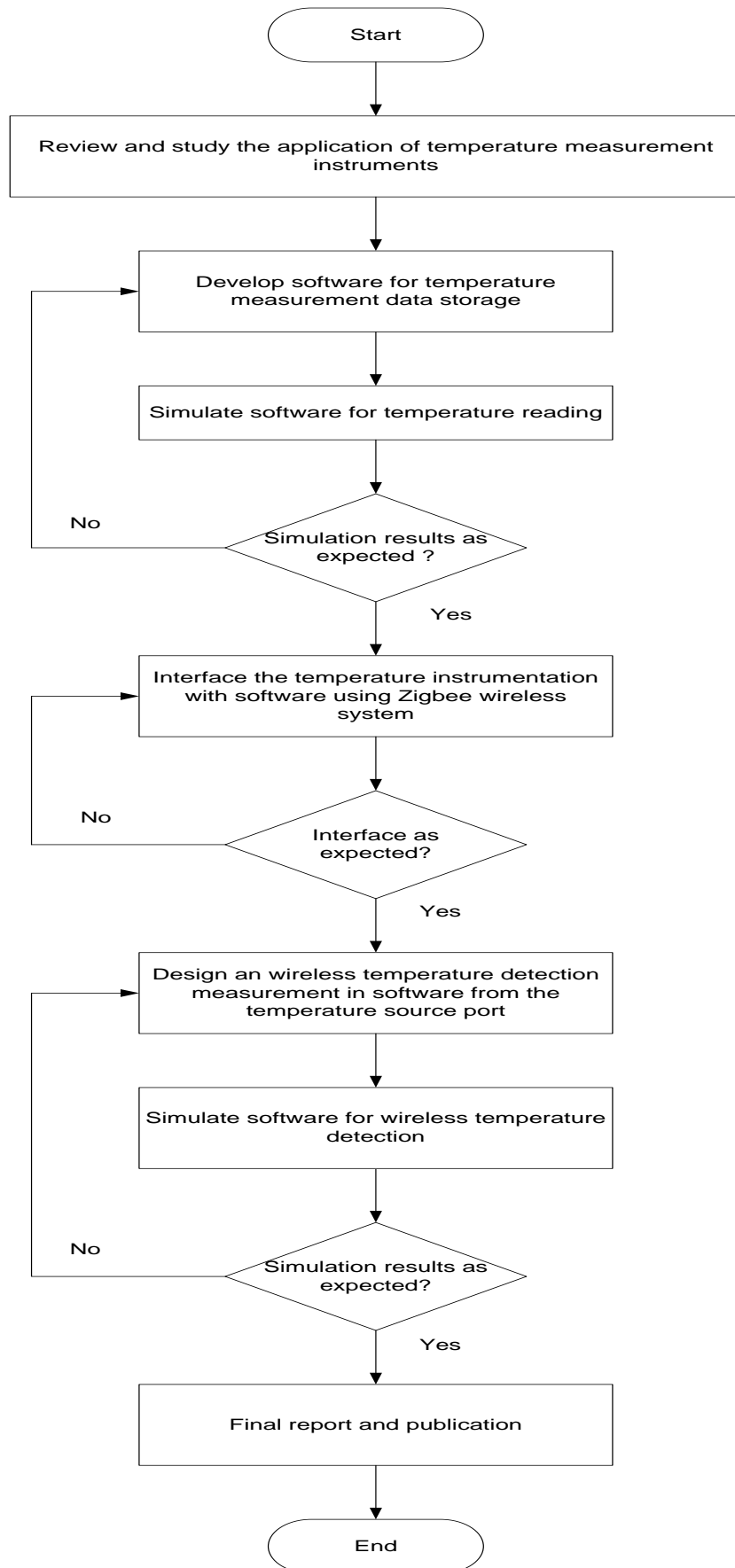


Figure 3.14: Flow chart of the project

CHAPTER 4

RESULT & DISCUSSION

4.1 Result for the GUI application

The GUI application shows the temperature monitoring system for Thermocouple type K. Under the INPUT label, the box is for the miliampere value from the thermocouple type K. Current from the value range of 4 mA – 20 mA is sent to the GUI application. Five readings will be taken during the monitoring data. The monitoring system is divided into 4 parts;

- miliampere (mA)
- Temperature (Temp)
- milivoltage(mV)
- Date and time

In miliampere (mA) section, the value from the INPUT box was keyed in at the mA row every time the data is entered. Then, at the temperature (Temp) section the temperature value was keyed in. The temperature value can be produced by using the equation 4.1

$$\text{Temperature} = \left(\frac{15 (0.1 * \text{current} - 0.4)}{0.16} \right) + 50 \quad (4.1)$$

The date/time section was display current date and time during monitor the temperature for each data. For the mV section, the value mV is taken by using the equation 4.2:

$$\text{Voltage} = 0.04 (\text{Temp} + 2.4) \quad (4.2)$$

Figure 4.1 shows the GUI application where the application system still in the design mode. Design mode shows only the label and box without any value.

Figure 4.1: The design mode for GUI application for monitoring and data record

Then, when run the application, the mode changed from design mode to run mode. During run mode, the value is entered and the result showed. Figure 4.2 shows the run mode for the GUI application.

Figure 4.2: The run mode for GUI application for monitoring and data record

At the run mode , the value of mA was entered at the INPUT box . Only the value 4mA to 20 mA is chosen . After five readings were taken , every data from monitoring system will be recorded . Table 4.1 shows the summary of data collection from GUI application

Table 4.1: Data collection from GUI application

	miliampere (mA)	Temperature (Temp)	milivoltage (mV)	Date / Time
Data 1	4	50	2.202	11/10/2010 3:15:01 AM
Data 2	8	87.5	3.663	11/10/2010 3:15:06 AM
Data 3	12	125	5.123	11/10/2010 3:15:09 AM
Data 4	16	162.5	7.019	11/10/2010 3:15:12 AM
Data 5	20	200	8.400	11/10/2010 3:15:15 AM

From the table, it shows that every time temperatures value is increases, the values of mV also increase. This shows that the relationship between temperature and mV is proportional. Figure 4.3 shows the graph of temperature vs. mV.

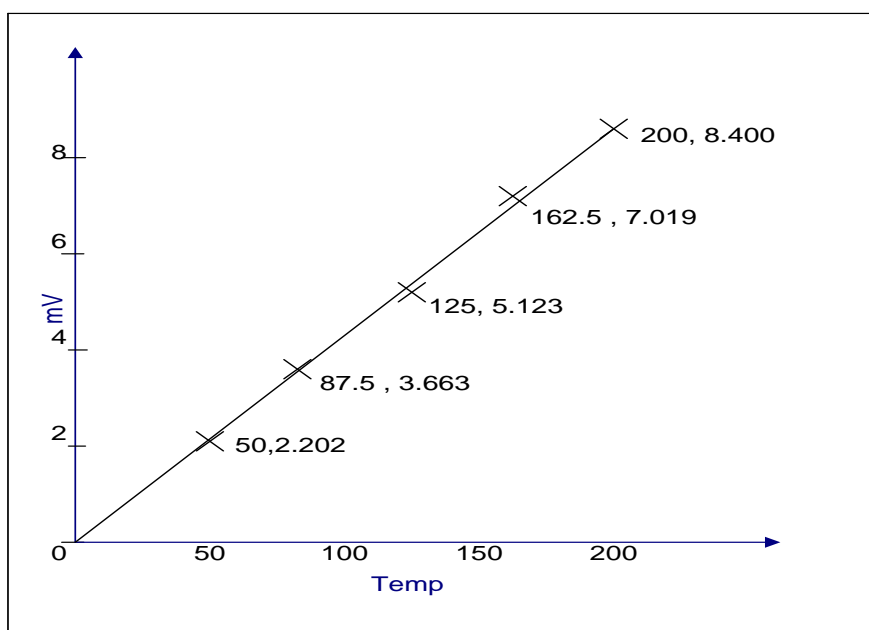


Figure 4.3: Graph Temperature vs. mV

During the monitoring, the measured value from the mV had a big error from the actual value from the Thermocouple type K temperature ($^{\circ}\text{C}$) table given in appendix.

NIST (National Institute of Standards and Technology) has published tables and equations that describe the voltage/temperature relationship. NIST provides two forms of polynomials. The first provides voltage as a function of temperature, and the second provides temperature as a function of voltage. Both of these forms will be used in the procedure described here for doing a thermocouple temperature conversion. The NIST tables and equations are all based on one end of the thermocouple (reference junction end) being at a temperature of 0 degrees Celsius. It is possible to avoid having a 0 degree reference junction. If the temperature of the reference junction is accurately known, then by using both the forward and inverse NIST polynomials the temperature of the thermocouple can be computed.

The coefficients for the temperature range can be divided into four parts:

- 0 $^{\circ}\text{C}$ to 1372 $^{\circ}\text{C}$
- -200 $^{\circ}\text{C}$ to 0 $^{\circ}\text{C}$
 - Voltage range -5.891 mV to 0 mV
 - Error Range 0.04 $^{\circ}\text{C}$ to -0.02 $^{\circ}\text{C}$
- 0 $^{\circ}\text{C}$ to 500 $^{\circ}\text{C}$
 - Voltage range 0 mV to 20.644 mV
 - Error range 0.04 $^{\circ}\text{C}$ to -0.05 $^{\circ}\text{C}$
- 500 $^{\circ}\text{C}$ to 1372 $^{\circ}\text{C}$
 - Voltage range 20.644 mV to 54.886 mV
 - Error range 0.06 $^{\circ}\text{C}$ to -0.05 $^{\circ}\text{C}$

Table 4.2 shows the coefficients and error range for each parts of the temperature range.

Table 4.2: The coefficients and error range for the temperature range

0 °C to 1372 °C	-200 °C to 0 °C	0 °C to 500 °C	500°C to 1372 °C
$c1 = 3.8921204975 * 10^{-2}$	$C1 = 2.5173462 * 10^1$	$C1 = 2.508355 * 10^1$	$C1 = 4.830222 * 10^1$
$c2 = 1.8558770032 * 10^{-5}$	$C2 = -1.1662878$	$C2 = 7.860106 * 10^{-2}$	$C2 = -1.646031$
$c3 = -9.9457592874 * 10^{-8}$	$C3 = -1.0833638$	$C3 = -2.503131 * 10^{-1}$	$C3 = 5.464731 * 10^{-2}$
$c4 = 3.1840945719 * 10^{-10}$	$C4 = -8.9773540 * 10^{-1}$	$C4 = 8.315270 * 10^{-2}$	$C4 = -9.650715 * 10^{-4}$
$c5 = -5.6072844889 * 10^{-13}$	$C5 = -3.7342377 * 10^{-1}$	$C5 = -1.228034 * 10^{-2}$	$C5 = 8.802193 * 10^{-6}$
$c6 = 5.6075059059 * 10^{-16}$	$C6 = -8.6632643 * 10^{-2}$	$C6 = 9.804036 * 10^{-4}$	$C6 = -3.110810 * 10^{-8}$
$c7 = -3.2020720003 * 10^{-19}$	$C7 = -1.0450598 * 10^{-2}$	$C7 = -4.413030 * 10^{-5}$	$C7 = 0$
$c8 = 9.7151147152 * 10^{-23}$	$C8 = -5.1920577 * 10^{-4}$	$C8 = 1.057734 * 10^{-6}$	$C8 = 0$
$c9 = -1.2104721275 * 10^{-26}$	$C9 = 0$	$C9 = -1.052755 * 10^{-8}$	$C9 = 0$

Table 4.3 shows the comparison between the actual value of mV and the measured value mV. Since the value of actual value and measured value is different, error is calculated. Percentage Error is the difference between the true value and the measured divided by the true value and the result is multiplied by 100 to make it a percentage. The percent error obviously can be positive or negative.

Table 4.3: comparison between actual value and measured value

Actual value (mV)	Measured value (mV)
2.023	2.202
3.557	3.663
5.124	5.123
6.620	7.019
8.138	8.400

Equation 4.3 can be used to get the error between actual values and measured values where the formula is the measured value of the experimental value minus the actual value divided by actual value times 100.

$$\text{Error percentage} = \frac{\text{measured value} - \text{actual value}}{\text{actual value}} \times 100 \% \quad (4.3)$$

From the Table 4.4 shows the error percentage for each reading. There error percentage for each value is really big. The temperature range for 5 readings is 0 °C to 500°C. If refer to the coefficient temperature range, the error range is error range 0.04 °C to -0.05°C . The error percentage shows that the equation 4.2 is not very accurate.

Table 4.4: Error percentage (%)

1 st reading	8.848
2 nd reading	2.980
3 rd reading	-0.0195
4 th reading	6.0272
5 th reading	3.219

CHAPTER 5

CONCLUSION

5.1 Summary of the work

By using Visual Basic, the development of GUI (Graphical User Interface) application can be done easily. This shows that Visual Basic is a good platform to develop user friendly software to calculate and analyze the temperature data. Besides that, the systems of offer a new learning experience for student. In addition, student can save a lot of time using this software instead of using manual calculation.

In this project, one of the objectives which are to interface the GUI application and the temperature transmitter output cannot be achieved. While interfacing, the problem occurred where the data from the software cannot receive any data from the temperature transmitter. Even though software can read both Xbee transmitter and Xbee receiver but still the data cannot be send.

As a conclusion, after observation, a few problems can be summarized. First, the programming for the ZigBee is not suitable for the interfacing system. The programming only works with the software application. Then, the Zigbee cannot support the current from the temperature transmitter.

5.2 Recommendation

For the future development and enhancement, there are some suggestions that seem can be implemented to improve the system. Here is some enhancement of this project that should be added in future:-

1. Upgrade the program for more function and not limited for analysis the data only. Use the software as a controller to control an actuator such as heater. Various controllers can be implemented such as discontinuous controller or continues controller mode. For instance, P, PD, PID controller can be used to control plant temperature.
2. Write the programming using other language such as LabView, C, and C++. This allows the program can be executed in other platform instead of Microsoft Windows.

5.3 Costing & Commercialization

This software application is suitable for the commercialization especially to student and industrial application. Most of the application in this software has been build to replace manual calculation. Manual calculation requires a lot of time, concentration and focus. Thus, by using this software student can minimize their time in doing analysis instead of using the manual calculation which require a lot of times. In addition, this software can increase productivity of works and make job easier

REFERENCES

- [1] H.D. Baker, E.A. Ryder, and N.H. Baker, “*Temperature Measurement in Engineering*”. Stamford, CT: Omega Press, 1975, vol. 2, p. 179.
- [2] D.D. Pollock, “*The Theory and Properties of Thermocouple Elements*”. (ASTM Special Technical Publication 492) Stamford, CT: Omega Press, 1979, pp. 1–23.
- [3] Thomas J. Bajzek,” *Thermocouples: A Sensor for Measuring Temperature Part 3 in a series of tutorials in instrumentation and measurement*”. IEEE Instrumentation & Measurement Magazine, 2005, pp 35-38
- [4] A.W. Van Herwaarden, P.M. Sarro, Thermal sensors based on the Seebeck effect, Sens. Actuators A 10 (1986) 321–346.
- [5] Norton, P., “*Visual Basic for Windows, 2nd Edition*”, Brady Publishing, New York, N.Y., 1992.
- [6] O.Kochan, R.Kochan, O.Bojko, M.Chyrka, “*Intelligent Data Acquisition and Advanced Computing System Technology and Applications*”. IEEE International Workshop, 2007 .pp 47-50.
- [7] Atkinson, M., Bailey, P., Chisholm, K., Cockshott, P. and Morrison, R., "An Approach to Persistent Programming", *Computer Journal*, Vol. 26 (4), 1983.
- [8] Evangelos Petroustos, “*Mastering Visual Basic 2010*”. Indiana, SYBEX: Wiley Publishing Inc, 2010, pp 8 -13.
- [9] Di Jasio , Wilmshurst , Ibrahim , Marton Bates , J. Smitth , “ *PIC Microcontroller* “. Oxford, Newness: Elsevier Inc, 2008, pp 3 -12.
- [10] Dogan Ibrahim, “*PIC Basic Project: 30 projects using PIC basic and PIC pro*”. Oxford, Newnes: Elsevier Inc, 2006, pp 13 – 46.
- [11] J. Hong, Q. Zhu, J. Xiao, “*Design and Realization of Wireless Sensor Network Gateway Based on ZigBee and GPRS*”. Second International Conference on Information and Computing Science, Manchester, UK, pp. 196–199, 2009.
- [12] Estrin,D. “*Wireless sensor networks application driver for low power distributed systems*”. Low Power Electronics and Design, International Symposium on 2001, Page: 194.

- [13] Yang Shulian, Zhang Yu, “*Temperature Measurement System for Ceramic Kilns Based on Wireless Communication Technology* “.2010 International Conference on Measuring Technology and Mechatronics Automation, China , 2010.
- [14] ZENG De-wen, CAO Jian, DING Jia-feng¹,YE Shao-long, “Kiln temperature monitoring system based on wireless”. *Manufacturing Automation*, Vol. 31, No. 4, pp. 95-97, April 2009.
- [15] Hayriye Korkmaz , Burhanettin Can, “ *Fuzzy Logic-Based Digital Filtering for Temperature Measurement via K-type Thermocouple*” . Computational Intellingence for Measurement Systems and Applications, Lagona , Switzerland ,29 – 31 July 2003
- [16] Korkmar, H., “*Designing and Realizing of a Temperature and Vacuum Rate Controllable Furnace*”. PhD Thesis, Istanbul, 2002
- [17] SUN Ning, SHEN Xiang-heng, “Design of wireless system for multi-channel temperature measurement”, *Control & Automation*,Vol. 19, No. 4, pp. 32-33, April 2009.
- [18] F. L. LEWIS “Wireless Sensor Networks,” Technologies, Protocols,and Applications. New York, 2004.
- [19] www.fieldcommunitor.com
- [20] http://www.daintree.net/downloads/whitepapers/zigbee_primer.pdf

APPENDICS

Appendic A

GUI programming

Public Class Form1

```
Dim oExcel As Object
Dim oBook As Object
Dim oSheet As Object
```

```
Dim Temp As String
Dim Current As Integer
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
End Sub
```

```
Private Sub TextBox3_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox3.TextChanged
End Sub
```

```
Private Sub TextBox7_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox7.TextChanged
End Sub
```

```
Private Sub TextBox8_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox8.TextChanged
End Sub
```

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
End Sub
```

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.
End Sub
```

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
    TextBox5.Text = System.DateTime.Now
End Sub
```

```
Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer2.Tick
    TextBox9.Text = System.DateTime.Now
End Sub
```

```
Private Sub TextBox16_TextChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles TextBox16.TextChanged
```

End Sub

```
Private Sub Label6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Label6.Click
```

```
    Dim Volt As String
    Current = TextBox1.Text
    Temp = TextBox3.Text
    Volt = TextBox4.Text
    Temp = ((3 * (100 * Current - 400)) / 32 + 50)
    Volt = 0.04 * (Temp + 2.4)
    TextBox2.Text = Current
    TextBox3.Text = Temp
    TextBox4.Text = Volt
    TextBox5.Text = System.DateTime.Now
```

End Sub

```
Private Sub Label7_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Label7.Click
```

```
    Dim Volt As String
    Current = TextBox1.Text
    Temp = TextBox3.Text
    Volt = TextBox4.Text
    Temp = ((3 * (100 * Current - 400)) / 32 + 50)
    Volt = 0.04 * (Temp + 2.4)
    TextBox6.Text = Current
    TextBox7.Text = Temp
    TextBox8.Text = Volt
    TextBox9.Text = System.DateTime.Now
```

End Sub

```
Private Sub Timer5_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer5.Tick
```

```
    TextBox21.Text = System.DateTime.Now
```

End Sub

```
Private Sub Timer4_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer4.Tick
```

```
    TextBox17.Text = System.DateTime.Now
```

End Sub

```
Private Sub Timer3_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer3.Tick
```

```
    TextBox13.Text = System.DateTime.Now
```

End Sub

```
Private Sub Label8_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Label8.Click
```

```
    Dim Volt As String
    Current = TextBox1.Text
    Temp = TextBox11.Text
```

```

    Volt = TextBox12.Text
    Temp = ((3 * (100 * Current - 400)) / 32 + 50)
    Volt = 0.04 * (Temp + 2.4)
    TextBox10.Text = Current
    TextBox11.Text = Temp
    TextBox12.Text = Volt
    TextBox13.Text = System.DateTime.Now
End Sub

```

```

Private Sub Label9_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label9.Click
    Dim Volt As String
    Current = TextBox1.Text
    Temp = TextBox15.Text
    Volt = TextBox16.Text
    Temp = ((3 * (100 * Current - 400)) / 32 + 50)
    Volt = 0.04 * (Temp + 2.4)
    TextBox14.Text = Current
    TextBox15.Text = Temp
    TextBox16.Text = Volt
    TextBox17.Text = System.DateTime.Now
End Sub

```

```

Private Sub Label10_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label10.Click
    Dim Volt As String
    Current = TextBox1.Text
    Temp = TextBox19.Text
    Volt = TextBox20.Text
    Temp = ((3 * (100 * Current - 400)) / 32 + 50)
    Volt = 0.04 * (Temp + 2.4)
    TextBox18.Text = Current
    TextBox19.Text = Temp
    TextBox20.Text = Volt
    TextBox21.Text = System.DateTime.Now
End Sub

```

```

Private Sub DataGridView1_CellContentClick(ByVal sender As System.Object,
ByVal e As System.Windows.Forms.DataGridViewCellEventArgs)

```

```

End Sub

```

```

Private Sub Label11_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label11.Click

```

```

End Sub

```

```

Private Sub Button1_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click

```

```

    'Start a new workbook in Excel

```



```

oExcel = CreateObject("Excel.Application")
oExcel.Visible = True
oBook = oExcel.Workbooks.Add()

```

'Add data to cells of the first worksheet in the new workbook.

```

oSheet = oBook.Worksheets(1)
oSheet.Range("A1").Value = " Temp "
oSheet.Range("B1").Value = " Current "
oSheet.Range("C1").Value = " Voltage "
oSheet.Range("D1").Value = " Date/Time"
oSheet.Range("A1:B1:C1:D1").Font.Bold = True
oSheet.Range("A2").Value = TextBox3.Text
oSheet.Range("B2").Value = TextBox2.Text
oSheet.Range("C2").Value = TextBox4.Text
oSheet.Range("D2").Value = TextBox5.Text
oSheet.Range("A3").Value = TextBox6.Text
oSheet.Range("B3").Value = TextBox7.Text
oSheet.Range("C3").Value = TextBox8.Text
oSheet.Range("D3").Value = TextBox9.Text
oSheet.Range("A4").Value = TextBox10.Text
oSheet.Range("B4").Value = TextBox11.Text
oSheet.Range("C4").Value = TextBox12.Text
oSheet.Range("D4").Value = TextBox13.Text
oSheet.Range("A5").Value = TextBox14.Text
oSheet.Range("B5").Value = TextBox15.Text
oSheet.Range("C5").Value = TextBox16.Text
oSheet.Range("D5").Value = TextBox17.Text
oSheet.Range("A6").Value = TextBox18.Text
oSheet.Range("B6").Value = TextBox19.Text
oSheet.Range("C6").Value = TextBox20.Text
oSheet.Range("D6").Value = TextBox21.Text

```

```

oBook.SaveAs("C:\Users\User\Documents\Book1.xlsx")
oExcel.Quit()

```

End Sub

```

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click

```

```

    SerialPort1.Open()
    Timer6.Enabled = True

```

End Sub

```

Private Sub Timer6_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer6.Tick

```

```

    TextBox1.Text = SerialPort1.ReadByte

```

End Sub

```
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click
    TextBox1.Text = ""
    TextBox22.Text = ""
    TextBox23.Text = ""
End Sub
```

```
Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button6.
End Sub
```

```
Private Sub Button2_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Dim Volt As String

    Current = TextBox1.Text
    Temp = TextBox22.Text
    Volt = TextBox23.Text

    Temp = ((3 * (100 * Current - 400)) / 32 + 50)
    Volt = 0.04 * (Temp + 2.4)
    TextBox1.Text = Current
    TextBox22.Text = Temp
    TextBox23.Text = Volt
End Sub
```


“I hereby acknowledge that the scope and quality of this thesis is qualified for the award of the Bachelor Degree of Electrical Engineering (Electronics)”

Signature : _____

Name : NAJIDAH BINTI HAMBALI

Date : 29 NOVEMBER 2010

“All the trademark and copyrights use herein are property of their respective owner. References of information from other sources are quoted accordingly; otherwise the information presented in this report is solely work of the author.”

Signature : _____

Author : NOR ASIAH BINTI AHMAD SUKORI

Date : 29 NOVEMBER 2010

DEDICATION

Dedicated to my family and friends

ACKNOWLEDGEMENTS

I am grateful and would like to express my sincere gratitude to my supervisor Mdm. Najidah Binti Hambali for her ideas, guidance, continuous encouragement and constant support in making this project possible. She gives me hope and courage and I really appreciate every single thing that she had done to me. I appreciate her consistent support since the Final Project I. I am truly grateful for her tolerance of my naïve mistakes. I also would like to express very special thanks to other lecturers especially Mr. Zamri bin Ibrahim who contribute his ideas while finishing this project.

My sincere thanks go to all my lab mates especially to Olivea and Zaiful and members of the staff of the Mechanical Electrical Engineering Department, UMP, who helped me in many ways and made my stay at UMP pleasant and unforgettable. Many special thanks go to member engine research group for their excellent co-operation, inspirations and supports during this study.

I acknowledge my sincere indebtedness and gratitude to my parents for their love, dream and sacrifice throughout my life. I acknowledge my sincere also to my friends for their helps, love and encouragement. I cannot find the appropriate words that could properly describe my appreciation for their devotion, support and faith in my ability to attain my goals

ABSTRACT

This project will focus on how to develop a wireless type K Thermocouple temperature measurement. Visual Basic is used as a main programming language to develop a GUI (Graphical User Interface) application. Then, the sensor for this project is a Thermocouple type K. Wireless system is used to interface the GUI and the thermocouple sensor and ZigBee will be used for the wireless. Wireless temperature measurement system is developed using ZigBee communication technology. It overcomes the disadvantages of wired measurement system, such as complicated wiring and difficult maintenance. The wireless sensor networks based on ZigBee has the characteristics of insulation, strong electromagnetic immunity, low power and high accuracy, solving the problem of easy breaking, easy aging, cannot resist high temperature in traditional optical fiber temperature measurement technology, saving the trouble of wiring, ensuring the reliability and safety of the operation. For this project, the temperature measured by the thermocouple is transmitted to the PC by transceiver controlled by ZigBee, and the temperature data is managed by PC. The thermocouple sensor detected temperature change and send the input to software system where the input is converted from current into voltage signal. In GUI application, the data is recorded and manipulated to get the appropriate result.

ABSTRAK

Projek ini akan menumpukan pada bagaimana untuk membangunkan sistem pengukuran suhu tanpa wayar menggunakan termogandingan jenis K. Visual Basic digunakan sebagai sistem perisian utama untuk mencipta aplikasi GUI (Graphical User Interface). Kemudian, pengesan yang digunakan untuk projek ini adalah pengesan termogandingan jenis K. Sistem tanpa wayar digunakan untuk menyambungkan antara aplikasi GUI dan pengesan termogandingan dimana ZigBee digunakan sebagai teknologi tanpa wayar. Sistem pengukuran suhu tanpa wayar dibangunkan menggunakan teknologi komunikasi ZigBee. Ini mengatasi kelemahan sistem pengukuran kabel, seperti kabel rumit dan penyelenggaraan yang sukar. Rangkaian pengesan tanpa wayar berdasarkan ZigBee mempunyai ciri-ciri insulasi, kekebalan elektromagnet yang kuat, penggunaan kuasa yang rendah dan ketepatan yang tinggi, menyelesaikan masalah mudah pecah, mudah rosak, tidak boleh menahan suhu yang tinggi dalam teknologi serat suhu tradisional optik pengukuran, menyimpan data gangguan kabel, memastikan ketahanan dan keselamatan operasi. Untuk projek ini, suhu yang diukur dengan termogandingan dihantar ke PC dengan alat yang dikendalikan oleh ZigBee, dan data suhu yang dikendalikan oleh PC. Pengesan termogandingan mengesan perubahan suhu dan menghantar input ke sistem perisian di mana input tersebut ditukar dari arus menjadi isyarat voltan. Dalam aplikasi GUI, data akan dicatat dan dimanipulasi untuk mendapatkan hasil yang sesuai.

TABLE OF CONTENTS

	Page
SUPERVISOR’S DECLARATION	ii
STUDENT’S DECLARATION	iii
ACKNOWLEDGEMENTS	v
ABSTRACT	vi
ABSTRAK	vii
TABLE OF CONTENTS	viii
LIST OF TABLES	xi
LIST OF FIGURES	xi
LIST OF ABBREVIATIONS	xiv

CHAPTER	TITLE	PAGE
1	<u>INTRODUCTION</u>	1
	1.1 Introduction	1
	1.2 Objectives of the project	2
	1.3 Problem Statement	3
	1.4 Scope of the project	3
2	<u>LITERATURE REVIEW</u>	4
	2.1 Thermocouple	4
	2.2 Thermocouple type	6
	2.3 Visual Basic	7
	2.4 PIC Microcontroller	8
	2.4.1 PICAXE 18X	9
	2.5 Wireless Measurement System based on ZigBee Transmission Technology	11

	2.6	Previous Research	13
	2.6.1	Temperature Measurement Technology with Sapphire Fiber Blackbody Sensor	13
	2.6.2	Fuzzy Logic-Based Digital Filtering for Temperature Measurement via K-type Thermocouple	15
3		<u>METHODOLOGY</u>	17
	3.1	Hardware	17
	3.2	Instruments	17
	3.2.1	The HART 375 Field Communicator	18
	3.2.2	Thermocouple type K	19
	3.2.3	YOKOGAWA Temperature Transmitter	21
	3.2.4	ZigBee	22
	3.3	Visual Basic	26
	3.3.1	Start the Visual Basic 2008	26
	3.3.2	The Visual Studio Tools	28
	3.4	X-CTU	29
	3.4.1	PC Settings Tab	30
	3.4.2	Range Test Tab	31
	3.4.3	The Terminal Tab	33
	3.4.4	Modem Configuration Tab	34
	3.5	Interfacing between hardware and software	37
	3.5.1	Modes of Operation of Xbee	39
4		<u>RESULTS AND DISCUSSION</u>	42
	4.1	Result for the GUI application	42
5		<u>CONCLUSION AND RECOMMENDATIONS</u>	48
	5.1	Summary of the work	48
	5.2	Recommendation	49
	5.3	Costing & Commercialization	49

<u>REFERENCES</u>	50
<u>APPENDICES</u>	52
A Programming for GUI Application	53
B Thermocouple type K Table	58

LIST OF TABLES

Table No.	Title	Page
2.1	Thermocouple type	7
4.1	Data collection from GUI application	44
4.2	The coefficients and error range for the temperature range	46
4.3	Comparison between actual value and measured value	46
4.4	Error percentage (%)	47

LIST OF FIGURES

Figure No.	Title	Page
2.1	Thermocouple circuit of materials A and B	5
2.2	Thermocouple circuit with junction of Tr opened	5
2.3	PICAXE-18x pins	9
2.4	Topology types for wireless sensor networks	12
2.5	Sapphire fiber blackbody cavity and its metal shell	14
2.6	Realized FLC system	16
2.7	Dividing into different sections of the rising curve by using REMF	16
3.1	The basic instrument connection	18
3.2	375 Field Communicator	19
3.3	Thermocouple type k	20
3.4	YOKOGAWA Temperature Transmitter	22
3.5	Network Coordinator	23
3.6	OSI model	24
3.7	Tool windows	27
3.8	New Project Dialog	28
3.9	Toolbox Window	28
3.10	X-CTU front program	29
3.11	Com test / Query Modem	30
3.12	Range test tab	31
3.13	Destination address box	33

3.14	The main terminal window	33
3.15	Data packet	34
3.16	Modem Parameters and Firmware section	35
3.17	Modem Configuration	36
3.18	Node list	37
3.19	Interface between software and hardware	38
3.20	Xbee Receiver and Xbee transmitter	38
3.21	Flow chart of the project	41
4.1	The design mode for GUI application for monitoring and data record	43
4.2	The run mode for GUI application for monitoring and data record	43
4.3	Graph Temperature vs. mV	44

LIST OF ABBREVIATIONS

IEEE	Institute of Electrical and Electronics Engineers
RFD	Reduced Function Device
FFD	Full Function Device
GUI	Graphical User Interface
IDE	Integrated Development Environment
WPAN	Wireless Personal Area Networks
DSSS	Direct sequence spread spectrum
BPSK	Binary phase shift keying
REMF	Relative Error Membership Function
OSI	Open System Interconnection
ISO	International Organization for Standardization
ZDO	ZigBee Device Object
ZDP	ZigBee Device Profile
SSP	Security Service Provider
API	Application Programming Interface