PORTABLE ELECTRONICS QUEUE CONTROL SYSTEM

WONG CHUN YUAN

UNIVERSITY MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS♦

JUDUL **PORTABLE ELECTRONICS QUEUE CONTROL SYSTEM**

SESI PENGAJIAN: **2011/2012**

Saya **WONG CHUN YUAN (890507-06-5397)**
(HURUF BESAR)

mengaku membenarkan tesis (Sarjana Muda/~~Sarjana~~ /~~Doktor Falsafah~~)* ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Universiti Malaysia Pahang (UMP).
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan (√)

[ ] **SULIT** (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

[ ] **TERHAD** (Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

[√] **TIDAK TERHAD**

Disahkan oleh:

______________________	______________________
(TANDATANGAN PENULIS)	(TANDATANGAN PENYELIA)

Alamat Tetap:

E-2778,TAMAN MURNI, **JALAN KAMPUNG BARU,** **25100, KUANTAN,** **PAHANG**	**HAZIZULDEN BIN ABDUL AZIZ** (Nama Penyelia)
Tarikh:**20 JUN 2012**	Tarikh: **20 JUN 2012**

CATATAN:
* Potong yang tidak berkenaan.
** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh tesis ini perlu dikelaskan sebagai atau TERHAD.
♦ Tesis dimaksudkan sebagai tesis bagi Ijazah doktor Falsafah dan Sarjana secara Penyelidikan, atau disertasi bagi pengajian secara kerja kursus dan penyelidikan, atau Laporan Projek Sarjana Muda (PSM).

“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 : HAZIZULDEN BIN ABDUL AZIZ

Date : 20 JUNE 2012

"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 : WONG CHUN YUAN

Date : 20/6/2012

To my beloved parents and friends

ACKNOWLEDGEMENT

Foremost, I would like to express my sincere gratitude to my supervisor Mr. Hazizulden bin Abdul Aziz for the continuous support, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research, especially his continuous support and correction in the development of this thesis.

Besides my supervisor, I would like to thank Mr. Azri from the laboratory store for giving me some advice in choosing the components and the development of the circuit. Apart from that, the encouragement, insightful comments, and willingness to help me in my project from other staffs in the university are much appreciated.

I would like to thank my course mates, Chong, Goh and Chou for the assistance in my project. The advise in using XBEE module, in learning C language after I changed from MC68HC11 to PIC18F4620 as well as the guidance on the design of circuit as well as for the sleepless night we worked together before deadlines. And for all the fun we had in the last three and a half years, I would like to thank all my fellow friends and course mates.

Last but not least, I would like to thank my parents for constantly supporting me from behind throughout my life and also their continuous supervision on my progress throughout the year.

ABSTRACT

A queue control system is developed to control the queue in businesses and in servicing counters. By using queue control system, the company or service provider will be able to control the queue of the customers and ensure the provision of a better service. Due to the increase of the demand from people for a better service, it is more likely that a queue control system is required to be allocated at all time for all companies. However, looking the queue control systems currently available in the market, most of them are big in size and unable to be moved here and there. These portable queue systems are mainly connected to big screen or static panel at the entrance for customers to obtain their numbers. This will cause problem in moving counters such as ticketing counters outdoor or servicing counters in a shopping mall. Hence, there is a demand for the portable queue control system which is able to be moved here and there. The purpose of this project is to develop a portable queue control system which is able to fulfill the request discussed above. Looking alike as the normal queue control system, portable queue control system is a small-sized system which can be used to control queue during outdoor events. Unlike static queue control system, portable queue control system is able to function in small range using 9V battery and is convenient to be carried here and there. In this simple system, one push button will be allocated at the display panel, allowing customer to input their arrivals, and a push button will be used to allow the counters to call for the next customer to be served. The display panel and the counters will be communicating with each other via XBEE wireless communication. For further enhancement, the system will be able to control the absence of more customers and sound notification which will call out the customer number and counter number is able to be installed. Besides that, duplication of the display can be made and the multi segments which are big in size can be used.

ABSTRAK

Sistem kawalan giliran dicipta untuk mengawal giliran pelanggan dalam perniagaan dan di kaunter-kaunter perkhidmatan. Dengan menggunakan sistem kawalan giliran, syarikat atau pembekal perkhidmatan akan dapat menjaminkan perkhidmatan yang lebih baik dengan system yang lebih baik kepada pelanggan untuk beratur. Atas permintaan orang ramai agar perkhidmatan yang lebih baik disediakan yang kian meningkat, sudah tidak dapat dinafikan bahawa sistem kawalan giliran perlu disediakan oleh semua syarikat pada bila-bila masa. Walau bagaimanapun, kebanyakan sistem kawalan giliran yang kini boleh didapati di pasaran adalah besar dari segi saiz dan tidak dapat dibawa ke sana sini. Sistem-sistem beratur ini biasanya disambungkan ke skrin besar atau panel statik di pintu masuk bagi pelanggan untuk mendapatkan nombor giliran mereka. Ini adalah kurang sesuai bagi kaunter bergerak seperti kaunter tiket atau perkhidmatan gerakan di pusat membeli belah. Maka, terdapat permintaan untuk sistem kawalan beratur mudah alih yang dapat bergerak di sana sini. Tujuan projek ini adalah untuk membangunkan sistem kawalan giliran mudah alih yang dapat memenuhi permintaan yang dibincangkan di atas. Sistem kawalan beratur mudah alih adalah satu sistem yang bersaiz kecil dan boleh digunakan untuk mengawal pelanggan supaya beratur apabila perkhidmatan disediakan di luar premis perniagaan. Berbeza dengan sistem kawalan beratur statik, sistem kawalan beratur mudah alih dapat berfungsi dalam rangkaian dekat dengan menggunakan bateri 9V dan mudah alih. Dalam sistem yang mudah ini, satu butang tekan akan disediakan di panel paparan bagi pelanggan untuk mendapatkan nombor giliran mereka. Di kaunter, sisten komunikasi tanpa wayar akan digunakan untuk membenarkan kaunter untuk memanggil pelanggan yang seterusnya. Bagi perbaikan, sistem ini boleh mengawal kehadiran lebih banyak pelanggan dan bunyi notifikasi yang akan memanggil nombor pelanggan dan bilangan kaunter boleh dipasang. Selain itu, system paparan dalam size yang lebih besar boleh digunakan.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

This chapter gives an overview of this project, giving readers a general idea of what is this project about. Chapter starts with overview of the project, containing problem statement, objective, scope of project as well as the project applications.

1.1 Overview

Queue control system is widely used in the market to help the business or company providing service to manage the queue of their customers. This is a very common system especially in banks or post office as well as hospital. The implementation of Queue Control system can provide a few advantages to the company as follow [10]:

- Speed up customer throughput and increase sales and profitability
- Improve productivity and operational efficiencies
- Minimize customer waiting times and reduce frustrations
- Increase satisfaction levels and improve customer retention rates
- Make waiting time informative and entertaining

Yet, the current queue control system available in the market is still having a few limitations in size and convenience to be used outdoor. Hence, the aim of the idea of designing a portable queue control system is to solve the problem of controlling queue when certain counters are moving from place to place such as ticketing counters or health consultancy.

1.2 Problem Statement

Throughout the years, the queue systems developed in the market were already widely used. The features provided by the queue system are to press number when arriving at the department and the counter will press the number queuing accordingly to serve them. However, there are still some problems with the queue systems in the market now.

The queue systems available in the market are queue systems which are installed as fixed system and cannot be moved. By using this system, people queuing in the buildings or department can be arranged accordingly. Yet, problems still occur where the crowd is usually wasting time queuing at moving stations like ticket counters. Moreover, the moving counters are not able to control the crowd whether they are queuing properly. In long queue, chaos and disorder might happen. People might be blaming or jump the queue.

In moving counters, the number of people coming is not expected, because the turn-out of the crowd is random. Hence, without a queue control system, the customers attended have to queue up and wait until their turn. This is a waste of time because they cannot do anything instead of waiting at the queue. By implementing queue control system, the customer can get their number and understand the number of people they have to wait for and go shopping or even go to get some food first.

Observing most of the electronic queue systems which are popular nowadays, they are big in size, high in cost and are not able to be moved here and there. This is the reason why most of the moving counters are not provided with any numbering system to manage the queue. Besides that, most of the systems will not show the current number on screen where users always click on the number first to check on numbers before making decision to wait or leave.

1.3 Objective

The main objective of this project is to develop a microcontroller based Portable Queue Management System which is small in size and is able to be moved among places. In order to achieve the main objective, the following sub-objectives were pursued:

1. To develop a queuing algorithm for wirelessly connected queue management system.
2. To design a microcontroller based electronic system connecting a main controller and remove call button for queue management system.

1.4 Scope of Project

The project was divided into two main parts: hardware and software. The hardware emphasized on the architecture and the specifications as well as the circuit connection of the system while the software scope emphasized on the development of the coding which will be designed to meet the hardware specifications.

Hardware Specifications

Table 1.1: Hardware Specifications

Hardware	Specifications
Service Counter	Two service counters were included which will be able to remotely call for the number to serve the customers
Input Panel	One panel was set-up for customer to get their number
Range between controller and remote call button	One hundred and twenty(120) meters as per ZigBee 2 communication range
Queue Number Display	The current numbers are displayed on a LCD screen
Call Number Display	Number being served is displayed on 7-Segment panel

Table 1.2: Software Specifications

Communication	Both remote were able to communicate with base using mesh connection
System Reset	System will reset after reset button was pressed. Reset button was connected in both remote and base circuit
Count Number Range	001 – 999

The design of the system is focusing specifically on the algorithm of the queue number calculation and the design of the communication between the remote and the base panel. Other common features of queue system such as token printing, voice announcement or priority numbers are omitted in the development of this engineering prototype.

1.5 Thesis Outline

Chapter 1 Introduction discussed the overview of the Portable Queue Control System, the problem statement which explained the demand for the design of the project, objective to be achieved in the project as well as the specifications of the prototype which includes the hardware and the software specifications.

In Chapter 2 – Literature Review, related books, journals and articles are used as references as guide to aid finishing this project. Researches were done based on several aspects to study on the features and the design of the project.

Chapter 3 Methodology, describe the process to accomplish this project in flow chart and explanation are available for each phase, also related software used are introduced here generally on what it capable to do. The hardware design is discussed as well.

Chapter 4 is Result and Discussion, of which all works done are presented and clarified proving that the result met the target of achievements. In addition, problem encountered will be discussed to explain some difficulties face and the solution to the problems.

Chapter 5, Conclusion and Recommendation conclude the development of project. Recommendation is included to encourage improvements made to the Portable Queue Control System and the commercialization of the project is discussed.

CHAPTER 2

LITERATURE REVIEW

The queue management system and queue model were discussed to study about the basic operation of the queue management system and how the queue system works. The process of queuing and the queuing theory were discussed here. The studies done on existing queue management system discussed about a few current used methods in developing queue management system. The methods were among the references that contributed to the design of this project. Lastly, the connection between the panels and the counters were discussed as well. The studies were done to do a simple research on the best method to be implemented in the connection of the devices.

2.1 Existing Queue Management System

In daily life, Customer service orientated companies and institutions frequently face the problem of lengthy queues and unpredictable waiting causing tension and stress among both customers and employees which result in efficiency decline [3]. Queue Management has been a problem for many years in many domains including the

Financial, Health Care, Public and Retail Sectors. In this age of technology it is not only important to organize the existing queue, but to gather statistics about the queue in order to identify trends that could be anticipated [5]. Hence, there is a need in the society for the continuous development of a Queue Management System especially in moving counters.

There are several types of Queue Management System products in the market which can be referred for this project. The different types of systems are due to the targeting size of queue and the efficiency as well the size of the targeted company. There are Stand Alone Queue System, Advance Queue System and Centralized Control Queue System.

2.1.1 Stand Alone Queue System

Figure 2.1 shows a model of standalone queue system. The design of this system is targeted for a single counter with a single queue [11]. The system will call queue numbers accordingly or randomly as preset. This system is suitable for small business because it only manages to control one queue with one counter. Excessive customers can cause increase in waiting time while big business with various services is unable to be separated for each service.

Figure 2.1: Stand Alone Queue System [11]

There is only one counter operating in Stand Alone Queue System. All of the customer will be managed at the same counter (Please refer figure 2.2) for the single service operation. The model used is First In First Out queue model (FIFO) [11]. Using the first come first serve concept, the entire customer will be treated equally. This is very suitable for single department or single service operation.

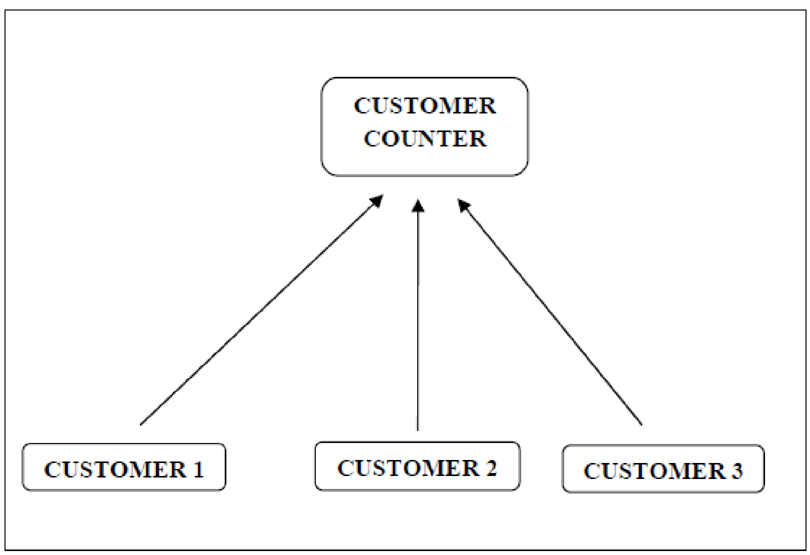


Figure 2.2: Operation of Stand Alone Queue System [11]

2.1.2 Advance Queue System

The system shown in Figure 2.3 refers to an advance queue system. This system is more likely to be used in banks or companies providing several services at once for a few different queues in different counters. In various counters, different services can be sorted accordingly while several counters can be operating at the same moment.

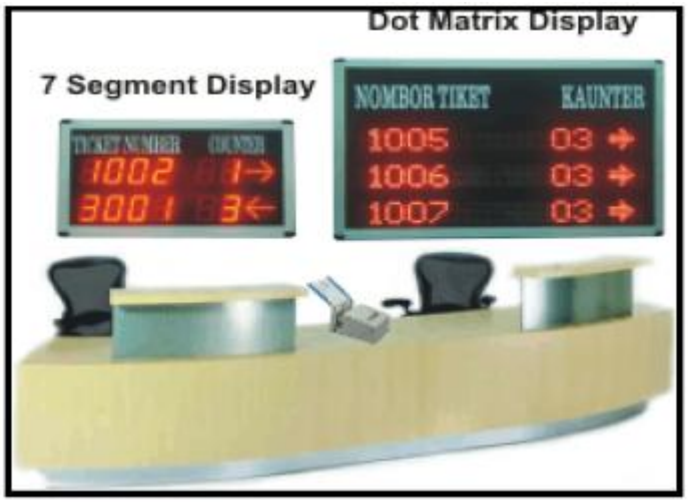


Figure 2.3: Advance Queue System [11]

Figure 2.4 illustrates the system model of the operation in the advance queue system where different customers will go to different counters for different services. In the advance queue system, big number of customers can be supported and various services are applicable. Different customer coming for different purposes can be recognized and called to the proper counter accordingly.

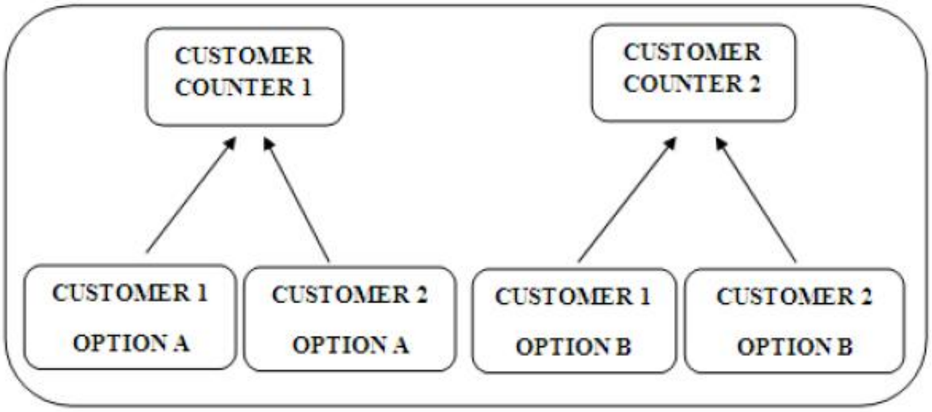


Figure 2.4: Operation of Advance Queue System [11]

2.1.3 Centralized Control Queue System

Centralized control queue system (Figure 2.5) is a high end server-based queue system. It can support more than 20 departments where each department having different services and several counters up to 32 services and 60 counters [11]. The system is compatible in network where each department located at different buildings can be connected through LAN or INTERNET. Centralized control queue system is more to an internet based connection rather than normal wireless connection in stand alone queue system or advanced queue system.

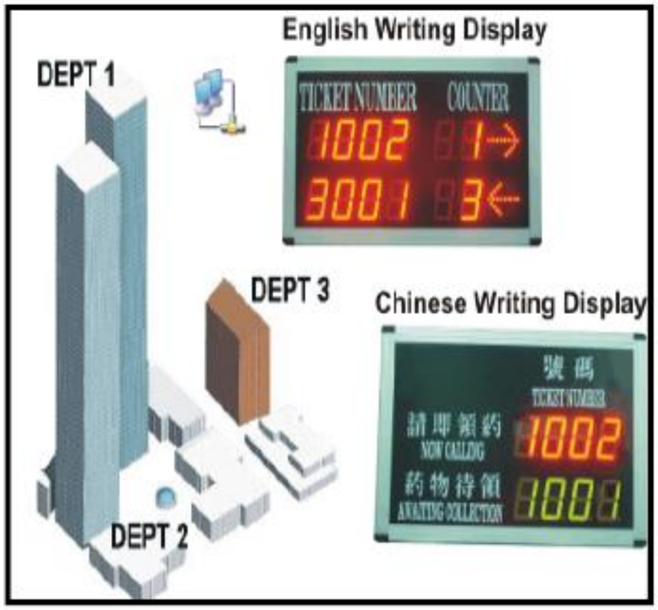


Figure 2.5: Centralized Control Queue Systems [11]

2.2 Queue Model

Queue refers to a first in first out (FIFO)-organized sequence of items, such as data, messages, jobs, or the like, waiting for action [2]. A queue management system is used to control queues. Queues of people form in various situations and locations in a queue area. The process of queue formation and propagation is defined as queuing theory [1].

In queuing theory, there are many types of queue model like first in first out (FIFO), last in first out (LIFO), processor sharing and priority [6]. First in first out (FIFO) principle states that customer are served one at a time and that the customer that has been waiting the longest is served first. Last in first out (LIFO) principle also serves customers one at a time; however the customer with the shortest waiting time will be served first. In processor sharing, customers are served equally. Network capacity is shared between customers and they all effectively experience the same delay. While in priority principal, customers with higher priorities are served prior to the others [6].

Hence, a queuing model has to be established accordingly to suit the queue management system. There are three categories for these types of models based on the number of counters: single server queues, multiple server queues and infinite server queues. Single server queues are most commonly encountered in real life. That is why this system is commonly used in many situations ranging from business, industry, transport, and telecommunications and computing. These are models where there is one server per queue and an item in the queue may have multiple queues to enter. Meanwhile, multiple server queues consist of two or more servers that are identical in serving a single queue of customers. Infinite server is a convenient theoretical model for situations that involve storage or delay, such as parking lots, warehouses and even atomic transitions. In these models there is no queue, as such, instead each arriving customer receives service and this model is rarely encountered in daily life [7].

2.3 Queue Control System Architecture

Figure 2.6 illustrate the architecture for a simple queue management system. It includes a few basic parts: A microcontroller as the main processing controller, a display panel output as the display for the number, a push button as input signal and a token panel to print the paper for the users.

Figure 2.6: Architecture of Simple QMS [3]

In this system, the output display was designed to call out the number to be served by displaying the called number on a 7 segment display panel. The counter that the customer should attend to was displayed too. For the same system, the input was designed via push button, where both of the people in charge at the counter and the customer arrived at the panel to press number and call for number via pressing the push button.

The input and the output were communicating with each other via microcontroller. The microcontroller was programmed to interface between the input and the output port to send the signals while the token panel was designed to print the current number and the number that they obtained.

Yet, the meant of the connection between the input microcontroller and the counter to send data calling for numbers is yet to be defined. Hence, a few approach of connection is to be discussed.

2.4 Data Transmission

The two major parts of the QMS is the counter and the base panel. The panel was designed to come together with a display board, showing number on display board to call for the next customer. While in the counter, a push button was designed to activate the display board which was separated from the counter. Both of the systems at the panel and the counter are independent and standalone. A few data transmission methods using wireless technique will be discussed to configure the method to allow communication between the counter and the panel.

2.4.1 ZigBee

ZigBee is a low-cost, low-power, wireless mesh networking proprietary standard. The low cost allows the technology to be widely deployed in wireless control and monitoring applications, the low power-usage allows longer life with smaller batteries, and the mesh networking provides high reliability and larger range. ZigBee operates in the industrial, scientific and medical (ISM) radio bands according to IEEE 802.11standard [18]. The technology is intended to be simpler and less expensive than other Wireless Personal Area Networks (WPANs) such as Bluetooth. Because ZigBee can activate (go from sleep to active mode) in 15 msec or less, the latency can be very low and devices can be very responsive — particularly compared to Bluetooth wake-up delays, which are typically around three seconds. Because ZigBees can sleep most of the time, average power consumption can be very low, resulting in long battery life [4].

2.4.2 Bluetooth

Bluetooth is a standard and a communications protocol primarily designed for low power usage, with a short range (power-class-dependent: 100m, 10m and 1m, but ranges vary in practice) based on low-cost transceiver microchips in each device. Bluetooth makes it possible for these devices to communicate with each other when they are in range. In most cases the effective range of class 2 devices is extended if they connect to a class 1 transceiver, compared to a pure class 2 network. This is accomplished by the higher sensitivity and transmission power of Class 1 devices. Bluetooth uses a radio technology called frequency-hopping spread spectrum, which chops up the data being sent and transmits chunks of it on up to 79 frequencies. Bluetooth provides a way to connect and exchange information between devices such as mobile phones, telephones, laptops, personal computers, printers, Global Positioning System (GPS) receivers, digital cameras, and video game consoles [4].

2.4.3 Wi-Fi

Wi-Fi or Wifi, is a mechanism for wirelessly connecting electronic devices. A device enabled with Wi-Fi, such as a personal computer, video game console, Smartphone, or digital audio player, can connect to the Internet via a wireless network access point. An access point (or hotspot) has a range of about 20 meters (65 ft) indoors and a greater range outdoors. Multiple overlapping access points can cover large areas. A Wi-Fi enabled device such as a PC, Smartphone, tablet or games console so they can connect to the Internet when within range of a hotspot. Hotspot coverage can comprise an area as small as a single room with wireless-opaque walls or as large as many square miles covered by overlapping access points [8].

2.4.4 Radio Frequency

Radio frequency (RF) is a rate of oscillation in the range of about 3 kHz to 300 GHz, which corresponds to the frequency of radio waves, and the alternating currents which carry radio signals. RF usually refers to electrical rather than mechanical oscillations, although mechanical RF systems do exist. In order to receive radio signals an antenna must be used. However, since the antenna will pick up thousands of radio signals at a time, a radio tuner is necessary to tune in to a particular frequency (or frequency range) [19]. Radio frequency is most commonly used in analogue data transmission.

2.4.5 Comparison

Table 2.1 tabulated the comparison between the four prefaces: ZigBee, Wi-Fi, Bluetooth and Radio Frequency (RF). Among the four transmission technologies, power consumption and limitations will be discussed to suit into the queue management system.

Table 2.1: Comparison between ZigBee, Bluetooth, Wi-Fi & Radio Frequency [4]

	ZigBee	**Wi-Fi**	**Bluetooth**	**Radio Frequency**
Range	10-100 meters	50-100 meters	10 – 100 meters	Long Range
Digital/Analogue	Digital	Digital	Digital	Analogue
Networking Topology	Ad-hoc, peer to peer, star, or mesh	Point to hub	Ad-hoc, very small networks	Tree topology, big network
Complexity (Device and application impact)	Low	High	High	Low
Power Consumption (Battery option and life)	Very low (low power is a design goal)	High	Medium	Medium
Security	128 AES plus application layer security	Wired Equivalent Privacy (WEP)	64 and 128 bit encryption	None
Typical Applications	Industrial control and monitoring, sensor networks, building automation, home control and automation, toys, games	Wireless LAN connectivity, broadband Internet access	Wireless connectivity between devices such as phones, PDA, laptops, headsets	Analogue signal transmission such as radio and walkie-talkie

Referring to the Table 2.1, Wi-Fi is definitely not a suitable protocol to be used as the applicant is more suitable for wireless and internet access. RF is not suitable as well because it is normally used for analogue data transmission unless we use modulator to convert the data. While in the comparison between Bluetooth and ZigBee, the range and the frequency is around the same for both of them. Yet, the power consumption of Bluetooth is higher than ZigBee [4] and Zigbee is more applicable in industrial control and monitoring due to the designed is improved from Bluetooth [4].

2.5 XBEE Module

Xbee is a particular brand of Zigbee compliant radios made by Digi International [14] while Zigbee is a standards organization that requires manufacturers to pay a fee to join in order to sell Zigbee labeled products. XBee is a product name, not a standard. Standard refers to agreed set of specifications and procedures while product referes to an item manufactured by a company/factory. For example, under the standard of Wi-Fi, there are various products manufactured from companies such as Dlink, Aztech or linksys. One of the reasons where XBee is being used widely and mistakenly as ZigBee is because XBee is low cost and easy to use. The popular XBee is XBee 802.15.4 (sometime called XBee series 1), using the same MAC (Medium Access Control) layer as the ZigBee, but XBee's upper layer is proprietary by the manufacturer. MAC is low layer of communication. ZigBee defines 802.15.4 MAC and XBee uses 802.15.4 MAC too. Yet, XBee cannot communicate with other ZigBee products because at higher layer they speak in different language [9]. Hence, in this project. bBee was used because ZigBee is just the protocol organization while XBee is the communication modem to be used.

2.6 XBEE 1, XBEE PRO1, XBEE 2 and XBEE PRO2

The XBee ZigBee modules from Digi International are available in two major classifications: XBee Series 1 and XBee Series 2 modules. The Series 1 and Series 2 modules are quite similar, but selection of a module should be based upon application specific needs.

The XBee Series 1 and Series 2 radios have the same footprint, and for the most part-are pin for pin compatible (few differences in the placement of ADC/IO lines), but are NOT interoperable. Series 1 and Series 2 use different application profiles, which are unique to each radio family. They can however, use the same RS232 or USB interface boards.

Table 2.2: Comparison between XBee 1 and XBee 2 [14]

	XBee Series 1	**XBee Series 2**
Indoor/Urban range	up to 100 ft. (30m)	up to 133 ft. (40m)
Outdoor RF line-of-sight range	up to 300 ft. (100m)	up to 400 ft. (120m)
Transmit Power Output	1 mW (0dbm)	2 mW (+3dbm)
RF Data Rate	250 Kbps	250 Kbps
Receiver Sensitivity	-92dbm (1% PER)	-98dbm (1% PER)
Supply Voltage	2.8 - 3.4 V	2.8 - 3.6 V
Transmit Current (typical)	45 mA (@ 3.3 V)	40 mA (@ 3.3 V)
Idle/Receive Current (typical)	50 mA (@ 3.3 V)	40 mA (@ 3.3 V)
Power-down Current	10 uA	1 uA
Frequency	ISM 2.4 GHz	ISM 2.4 GHz
Dimensions	0.0960" x 1.087"	0.0960" x 1.087"
Operating Temperature	-40 to 85 C	-40 to 85 C
Antenna Options	Chip, Integrated Whip, U.FL	Chip, Integrated Whip, U.FL, RPSMA
Network Topologies	Point to point, Star	Point to point, Star, Mesh
Number of Channels	16 Direct Sequence Channels	16 Direct Sequence Channels
Filtration Options	PAN ID, Channel & Source/Destination	PAN ID, Channel & Source/Destination

The major difference that made the difference is the network topologies. XBEE1 is only able to function in point to point or Star topologies, which meant, one device only can communicate with another device at once only. Meanwhile, for XBee2, Mesh topology is applicable where several devices can be connected and communicating with each other at the same moment.

2.7 Summary

The project will be designed based to the architecture of standalone queue management system. There will be one input panel with the display to show the number called and the counter number. Further discussion on the design will be discussed in the methodology.

For the connection between the devices, wireless connection is chosen and among the few wireless connectivity methods, ZigBee is the best choice due to the applicable in the system. XBEE Series 2 will be used because Mesh topology is more applicable in the communication of the system.

CHAPTER 3

METHODOLOGY

3.1 Introduction

The development of the project was divided into two parts:

i) Hardware design

ii) Software design

3.1.1 Preparation of the Circuit

In this stage, the design of the project was developed. The first part of the circuit preparation stage was the design of the project, where the block diagram and the program flow chart were developed. Next, the idea of the design was implemented, decision on the using of the components and the parts of the system was done such as which microcontroller to use and why, how many microcontroller to be used, how to communicate between the counter and the server, how to display the number and so on.

After the development of the block diagram, flow chart and the component list was done, the circuit was developed.

3.1.2 Proteus Stimulation

In this stage, the circuit was tested via Proteus IDE software. Proteus IDE is a freeware designed to help the students in doing their projects. The desired circuit was designed and in the circuit, the actual value of the components was decided and the connection between the boards was sketched.

One benefit of the usage of Proteus is the real time stimulation. Both of the design architecture and the software stimulation was done via Proteus. The program to run the microcontroller was written in C language and compiled using CCS PCW Compiler. After the compilation of the program, the coding was tested in Proteus accordingly with the circuit.

3.1.3 Preparation of Prototype

In this stage, the work started after the related components are obtained. The components were soldered on the board and the wrapping of the wiring between the components was done. The outcome of this stage was the full prototype to be developed which will be improved after several testing.

The task was divided into two parts: First part was the soldering of the panel to test the input panel and the 7segment display as well as the LCD display. The second part was the soldering of the counter. In the first part, the microcontroller, 7segments, BCD decoder as well as the push button were soldered onto the breadboard. After that, the connections between the components were connected via wrapping using wire. In the second part, the microcontroller and the ZigBee circuit were soldered and connected.

3.1.4 Test Run and Troubleshooting

The first stage of the test run was done by testing the program written in CCS PCW Compiler. The written program was compiled and written into the PIC microcontroller circuit in Proteus. After that, the program will be used to run the seven segments to confirm the validity of the program.

Next, the prototype will be tested for running in the lab. If any problem encountered, the troubleshooting for hardware and software will be carried out to find out the problem and to ensure the project is working.

3.2 Preliminary Design (Hardware)

3.2.1 Block Diagram for System

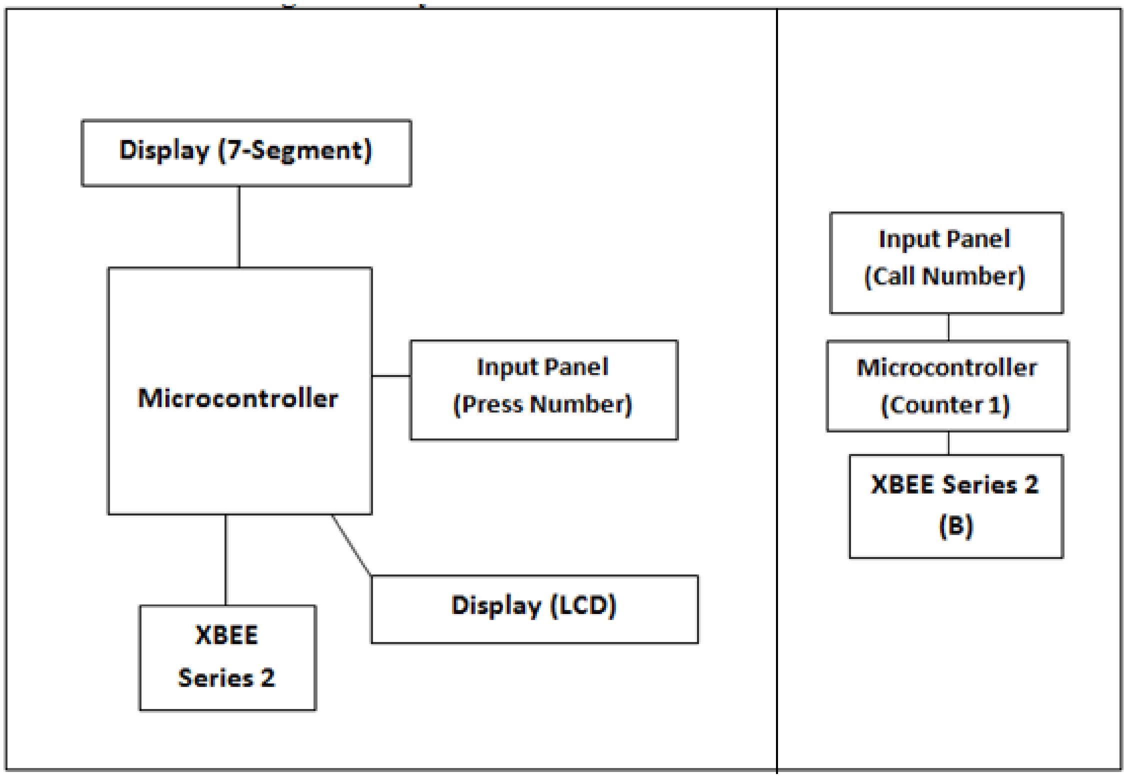


Figure 3.1: Block Diagram of the System

The system was divided into two parts: coordinator and router. The coordinator is the microcontroller allowing the input of the number when the customer arrived. The router was connected to the counter 1 and counter 2. The router will allow the person in charge to press the push button to indicate calling for the customer to be served. At the coordinator, the 7 segment display and the LCD display was shown.

The coordinator and router were communicating with each other via XBEE. XBEE wireless communication allows the data to be sent to each other when calling for number to be displayed. The router delivered data to the coordinator to execute the program. In the coordinator, the microcontroller was connected to XBEE modem to receive data, input panel to allow input from customer to call for number, and display to show the number called out.

3.2.2 Coordinator

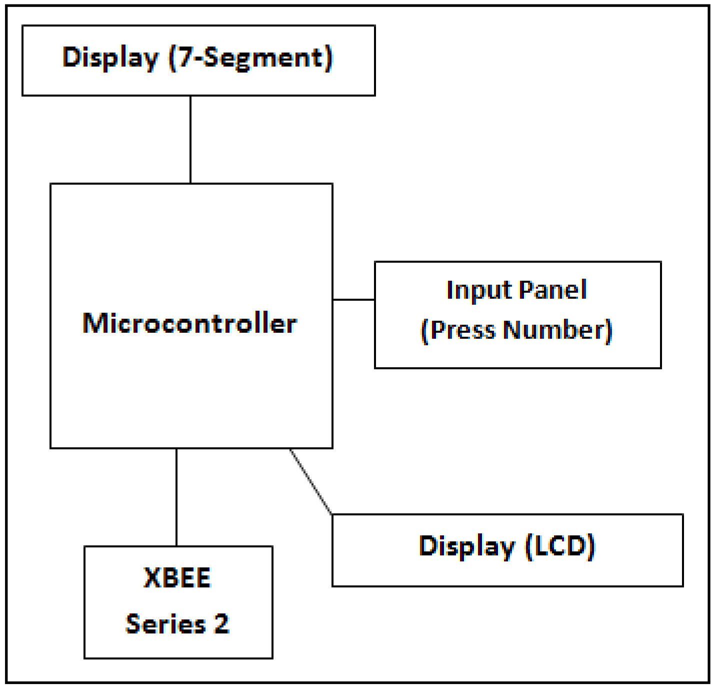


Figure 3.2: Block Diagram of the Coordinator

In the coordinator, the microcontroller was connected to a few devices:

1) Input Panel

The input panel was designed to allow the customer to input their presence upon arrival. The number of the people queuing will be increased and the customer will be called to be served

2) Display – 7 Segment

7 Segment display was designed to display the number called to be serviced. Once the counter called for the next customer, the next queue number was displayed here.

3) Display – LCD

LCD will show the customer their number and the current number when they are pressing the input button.

4) XBEE

XBEE was implemented to allow the communication between the coordinator and the router (counters)

3.2.3 Router (Counters)

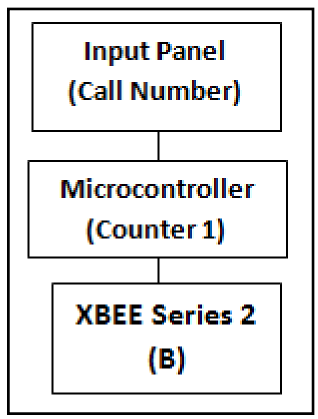


Figure 3.3: Block Diagram of the Router

In the router, the microcontroller was connected to two devices only – the input panel and the XBEE.

1) Input Panel

Input panel will activate the system, calling the coordinator to call the next number and display it on the seven segments.

2) XBEE

XBEE functioned as the wireless connection device with other counter and the coordinator. This is a high range communication where it can be in range of more than 120meters.

3.2.4 Circuit Connection - Coordinator

Figure 3.4 shows the connection interfacing between microcontroller PIC18F4620 and the seven segments. It was a testing circuit by using push button as input to trigger the signal.

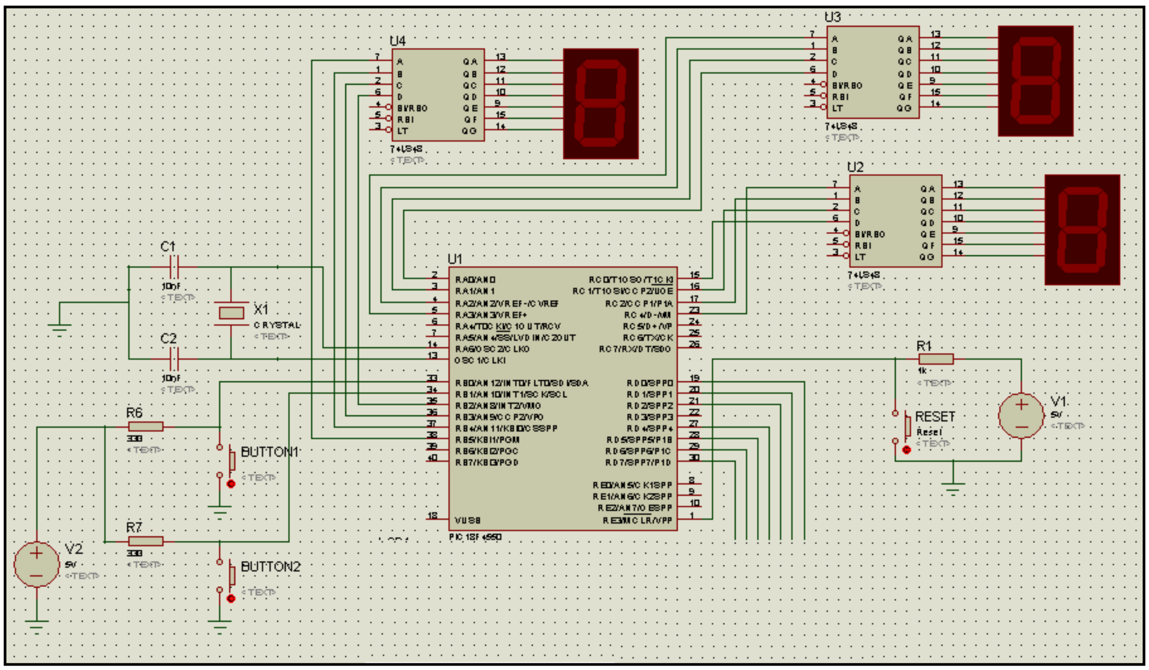


Figure 3.4: Circuit Diagram for Coordinator

PIC18F4620 is having 40 ports. The ports can be initialized to be input or output via programming. The entire circuit, including the input button and the 7 segments display were connected directly to the Port A, B and C. Port D is reserved for LCD. Meanwhile, reset button was connected directly to reset port.

3.2.5 Circuit Connection – Counter

The counter was designed to send command to the coordinator to call for the number to be displayed. The connection between the microcontroller and the input buttons are shown in Figure 3.5.

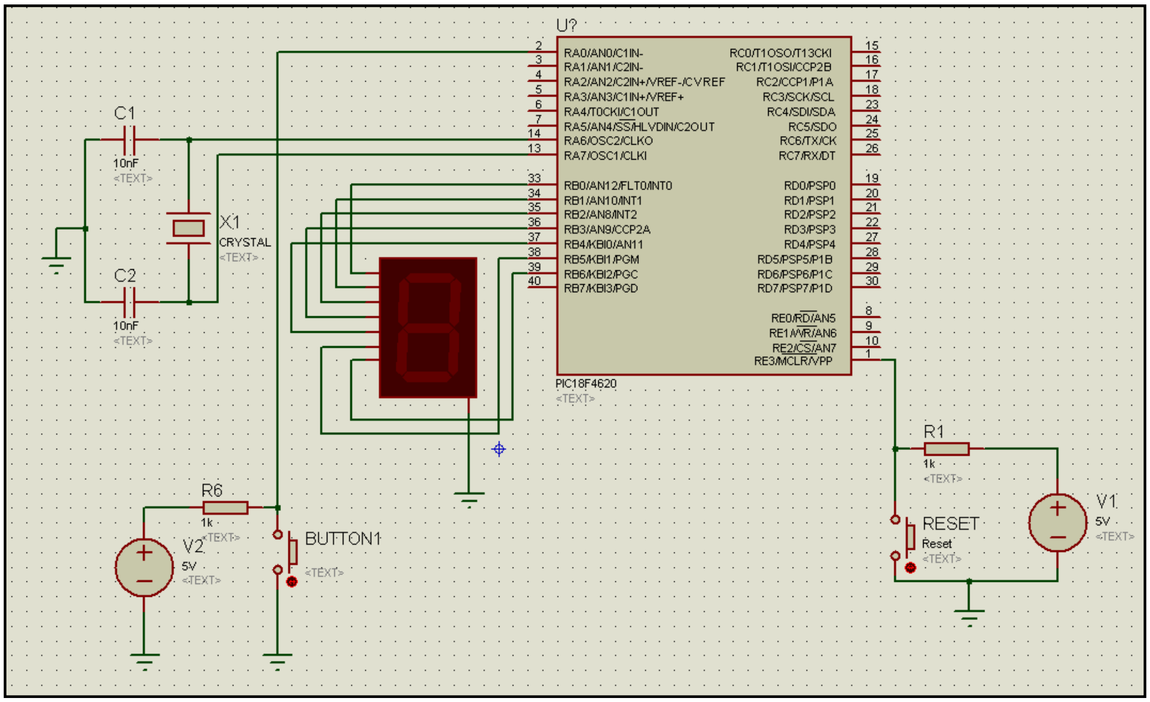


Figure 3.5: Circuit Diagram for Counter

Button 1 acts as the button for the person in charge at the counter to press and call for the next customer. After the button was pressed, the 7segment on the counter displaying the number will be blinking, and a command will be sent to coordinator via XBEE.

3.2.6 7-Segment Latch (Decoding)

One of the problems with microcontroller is the limited input and output port. In this project, there were three seven segments used in the coordinator panel. As shown in figure 3.6, connecting three 7 segments to the microcontroller directly will consume up 24 output ports. Hence, 7448 BCD Decoder was used to control the number of ports used in the project. In figure 3.7, using one 7448 decoder will save up to 4 output ports of the microcontroller.

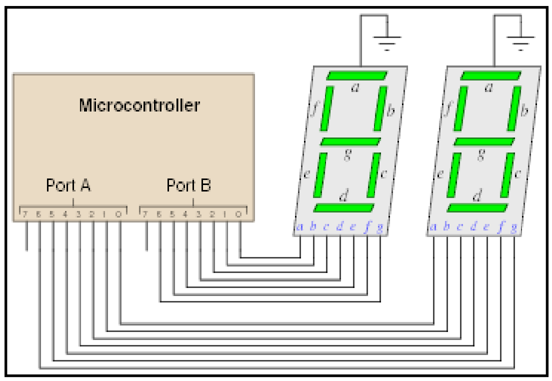


Figure 3.6: Seven Segments –Without Decoder [17]

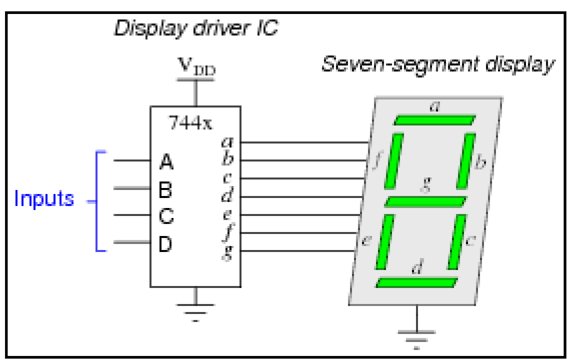


Figure 3.7: Seven Segments –With Decoder [17]

Rather than using multiplexing technique, using 7448 Decoder is more efficient and less complicated. In multiplexing, several seven segments were connected to the same output port, sharing the same output. To activate the selected seven segments, the latch was activated. However, in multiplexing, the circuit is much more complicated due to the connections to the latch. Hence, since number of port is allowable, decoding technique was used.

3.2.7 XBEE Configuration

To configure XBEE, the XBEE has to be connected to the module via breakout board. The XBEE input voltage is not 5V, but its 3.33V, so another regulator has to be setup separately.

Step 1: Construct the circuit

Build the XBEE Breakout board and plug one of the XBEE's into it. Set up the circuit on the bread board. [12]

Step 2: Power-up the XBEE and prep the software

After double checking the connections, use the USB-miniB cable to plug the FT232 Breakout into a USB port on your computer. Both LEDs connected to the XBEE should light up and stay on (assuming your XBEE shipped as a Router/End Device and there is no Coordinator around to connect to). Download the X-CTU software from Digi and install it. [12]

Step 3: Run X-CTU and connect to the XBEE

Run the X-CTU and click on the USB COM port that the XBEE is connected to. If not sure, click on the "Test/Query" button to read each COM port to discover which one has the XBee. [12]

Figure 3.8: XBEE Series 2 [13]

Step 4: Update the firmware

Click on the “Modem Configuration” tab and then click on the “Download new versions…” button to download all of the updated firmware. After the firmware downloads have completed, click on the “Read” button. The window will display all of the current settings of the attached XBEE. [12]

Step 5: Test the XBEE

Click on the terminal tab and type “+++” in the window to enter command mode. The XBEE will respond in a second or two with “OK”. Type “ATVR” to check the firmware version on the XBEE. Type “ATID” to check the PAN Network ID that the XBEE is using. Type “ATNI” to check the Node Identifier. Type “ATCN” to exit command mode. [12]

3.3 Preliminary Design (Software)

Two sets of coding were written, one for the base panel which will be functioning as the coordinator and another set for the counter panel, which the coding is similar for both of the routers. The coding was written in C language using PIC C Compiler, using the header file for PIC18F4620 (the microcontroller obtained from the laboratory). As shown in Figure 3.9, the header file was included in the program to enable the microcontroller to be activated.

```
#include <18F4620.h>                    // PIC18F4620 HEADER FILE
#include <stdlib.h>
#fuses HS,NOWDT,NOLVP,NOPROTECT         // EXTERNAL CLOCK, NO WATCH DOG TIMER, NO LOW VOLTAGE PROGRAMMING
#use delay (clock=20M)                  // 20 MHZ CRYSTAL
#use rs232(baud=9600, UART1)            // USE UART1 FOR XBEE COMMUNICATION
```

Figure 3.9: Programming Header File

3.3.1 Counter Panel

The counter panel functions as the system activation transmitter and router. The counter will power up the system at the base panel when it is activated. The system flow of the counter panel was shown in Figure 3.10

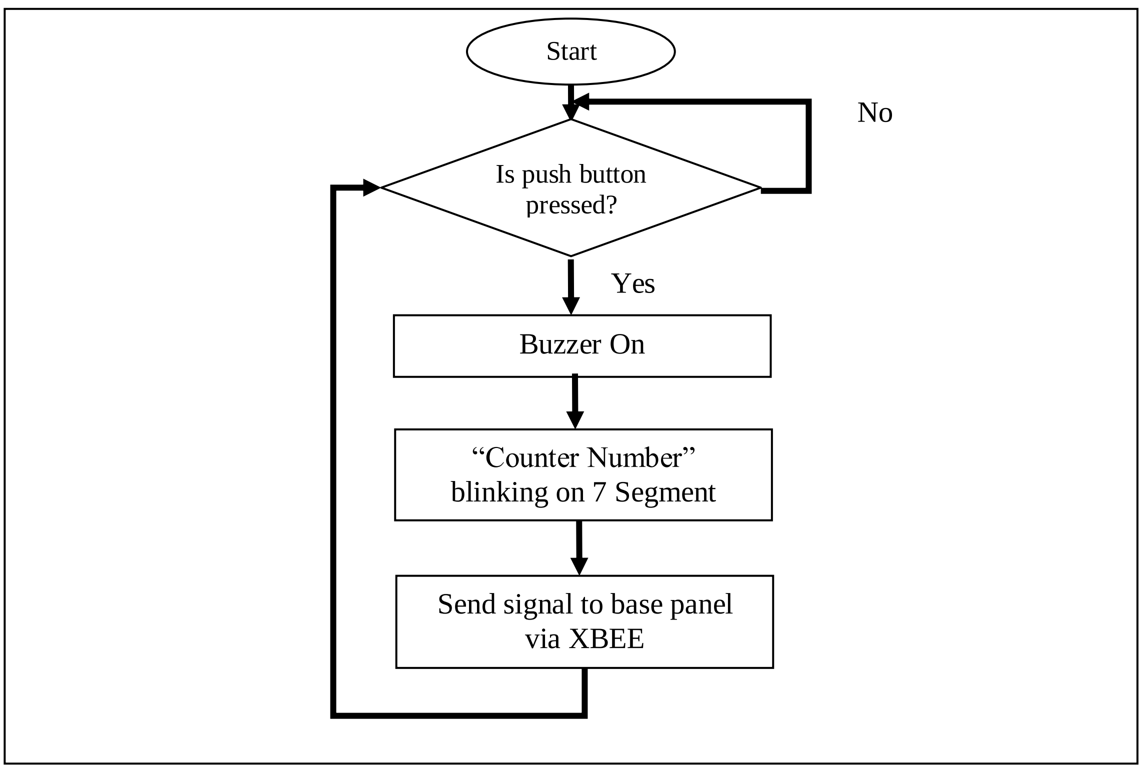


Figure 3.10: Flow Chart of Counter Panel

After the system is powered up, the system will await for the input from the push button. The system will remain at the same state if there is no input. When the push button is pressed, the buzzer will be turned on and the 7-segment connected to the counter panel will be blinking, showing the "Counter Number". At the same time, output signal will be sent to the base panel (coordinator) to call for the run of the program. The detailed description on the base panel will be discussed in Chapter 3.3.2. After that, the system will return to the sub-routine waiting for the input from the button again.

3.3.2 Base Panel

The base panel functions as a receiver and coordinator. The base panel receives input from customer to get their number, and receives signal from the counter panel to activate the display panel to show the number called to be served. The flow chart for the base panel is shown in Figure 3.11. There are a few components connected to the base panel: the input button for customer to input their number and the 7-segment panel to display the number to be served.

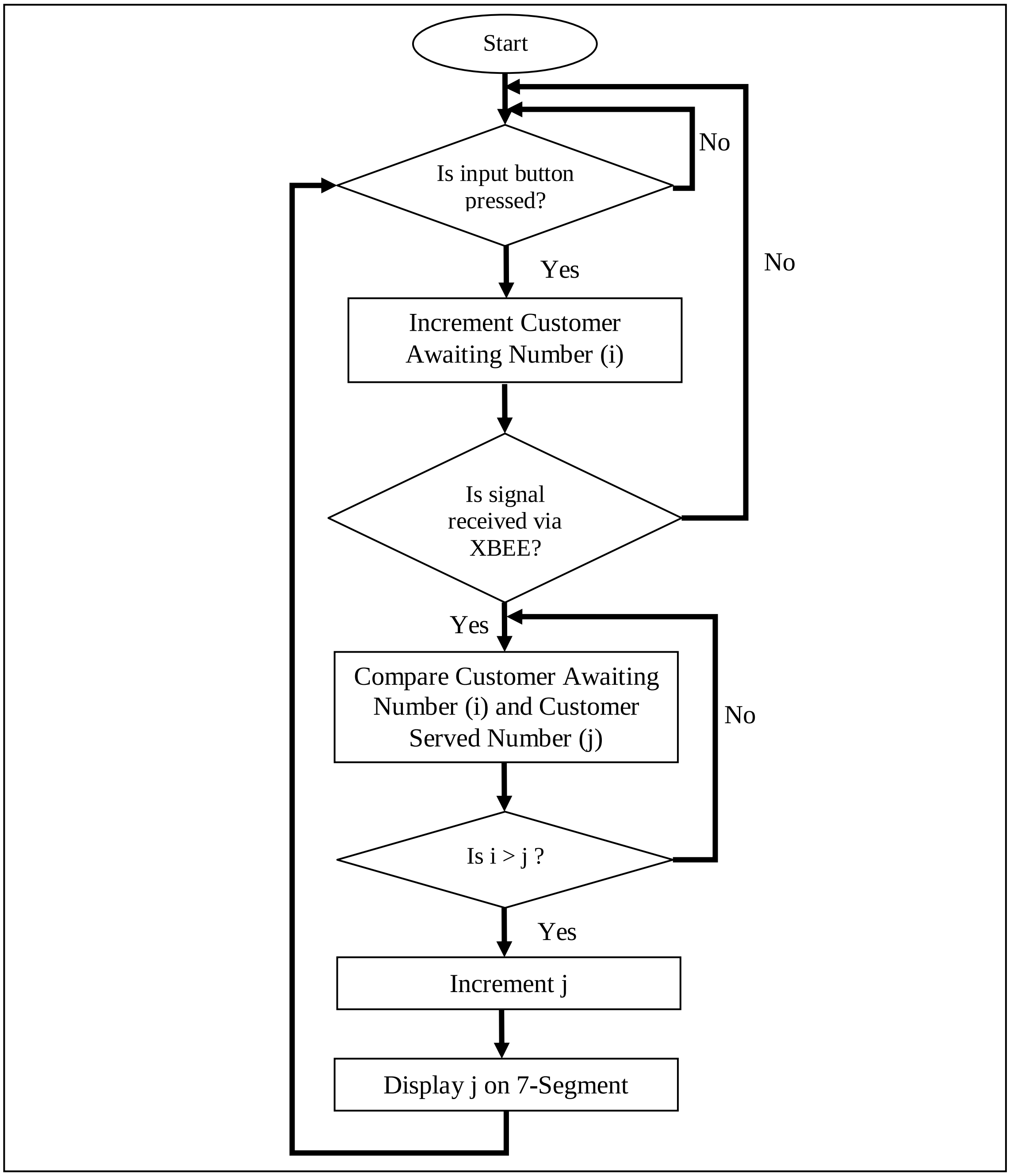


Figure 3.11: Flow Chart for Base Panel

When the system is powered up, the system will await for input number by customer. When a customer came to the panel and pressed the input button, the microcontroller will increment a number in the memory (i), recording the number of the customer. Next, if there is input from the XBEE router at the counter panel, another algorithm will be activated.

The system will compare another number in the memory (j), with the number (i). If (i) > (j), meaning the number of customer waiting is greater than the number of customer being served, there is some customer waiting to be served. Then, the system will increment (i) and load the value of (i) to be displayed on the 7-Segment display panel. The system will repeat the algorithm until reset or achieved maximum value of nine hundred and ninety nine (999).

CHAPTER 4

RESULT & DISCUSSIONS

This chapter discusses about the outcome of the project, the testing of the prototype as well as some discussions on the project. In this chapter, the photos of the outcome of the project will be shown.

5.1 XBEE Communication

In the portable queue control system, there are two important parts. First part is the algorithm of the counting and display in the base panel, while second part is the communication method between counter and panel. In this part, the communication between the XBEE's of the counters and the base panel will be discussed.

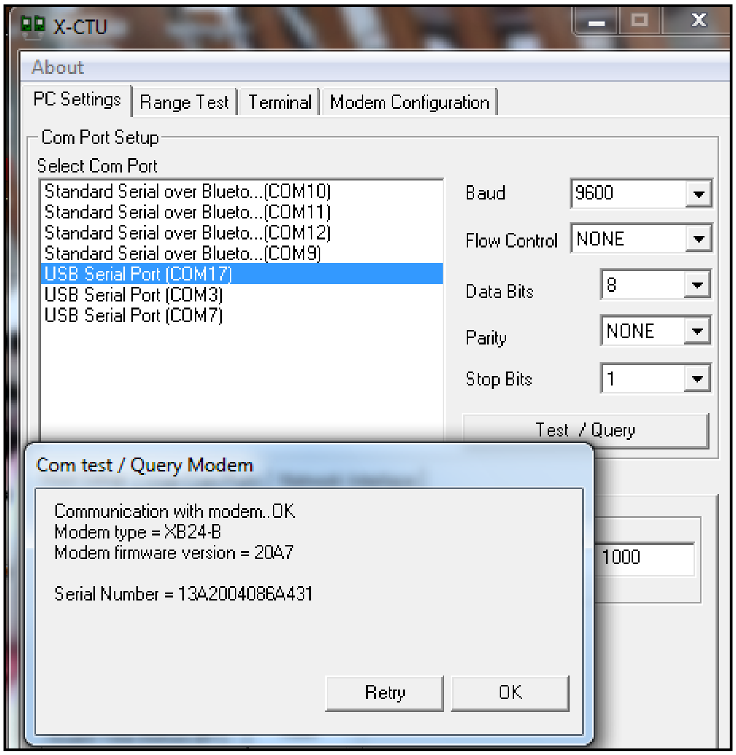


Figure 4.1: Test/Query

As shown in Figure 4.1, three of the XBEE Series 2's were tested via Com test / Query Modem to ensure that the firmware can be loaded. Next, the setting was configured accordingly. In this project, one XBEE will function as coordinator while another 2 XBEE's will be set to function as Router.

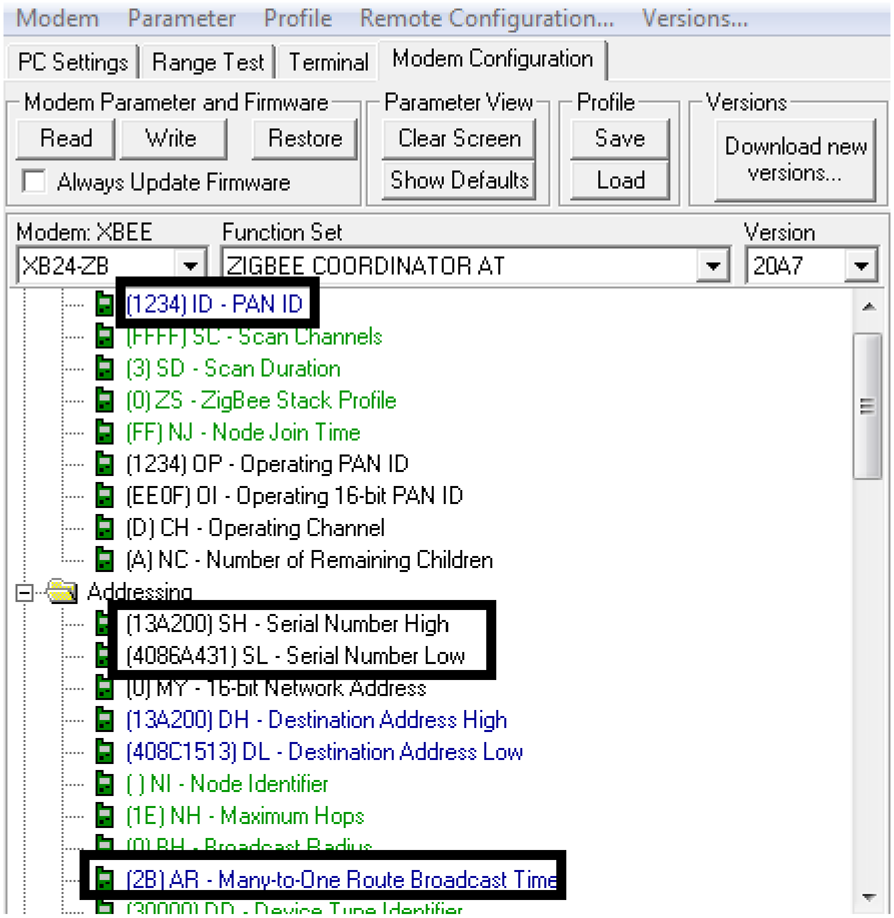


Figure 4.2: Coordinator Setting

Figure 4.2 shows a few important setting for XBEE to communicate. For the first XBEE which acts as the coordinator, the PAN ID was set to 1234 (default), and the Mant-to-One Route Broadcast Time was set to 2B. The same setting will go for the XBEE router configuration.

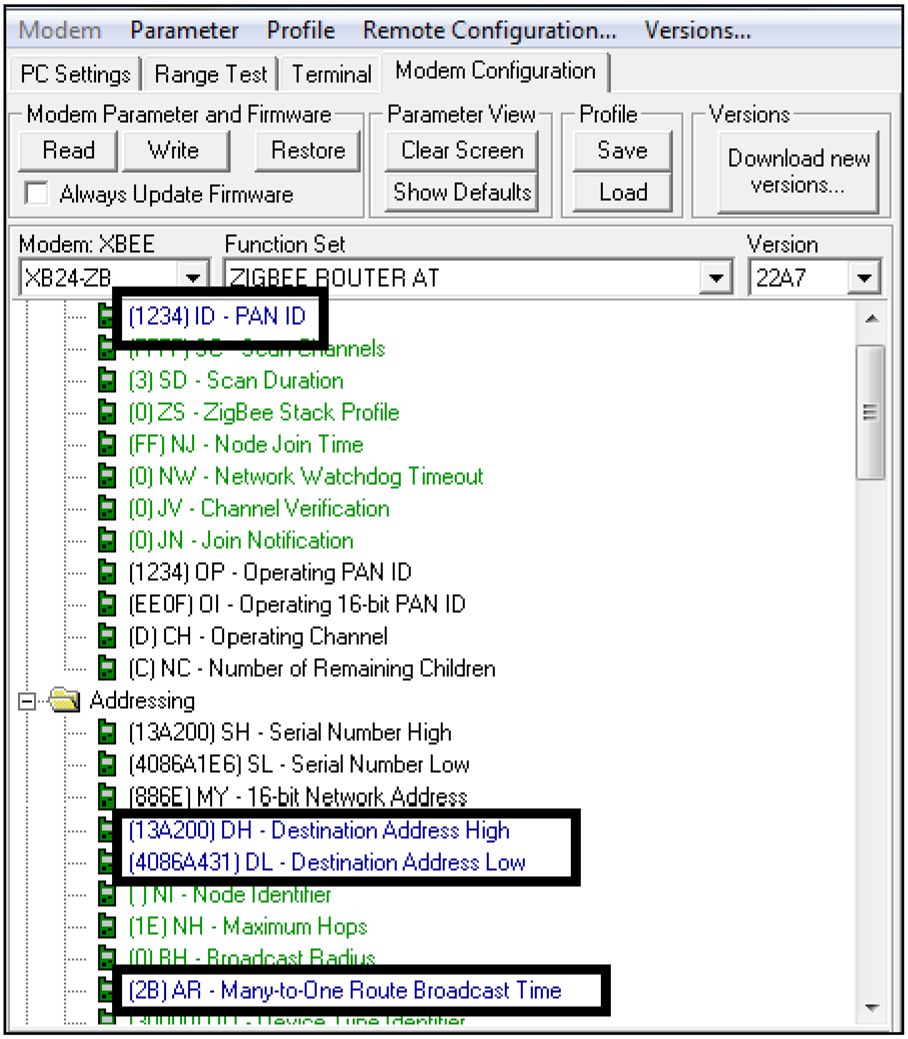


Figure 4.3: Router Setting

The setting for PAN ID and Many-to-One Route Broadcast Time was duplicated for both of the XBEE's as the router. However, different from the coordinator, the Destination Address High (DH) and Destination Address Low (DL) were set as the same, copying the Serial Number High (SH) and Serial Number Low (SL) of the coordinator. The router XBEE's were able to communicate with the coordinator.

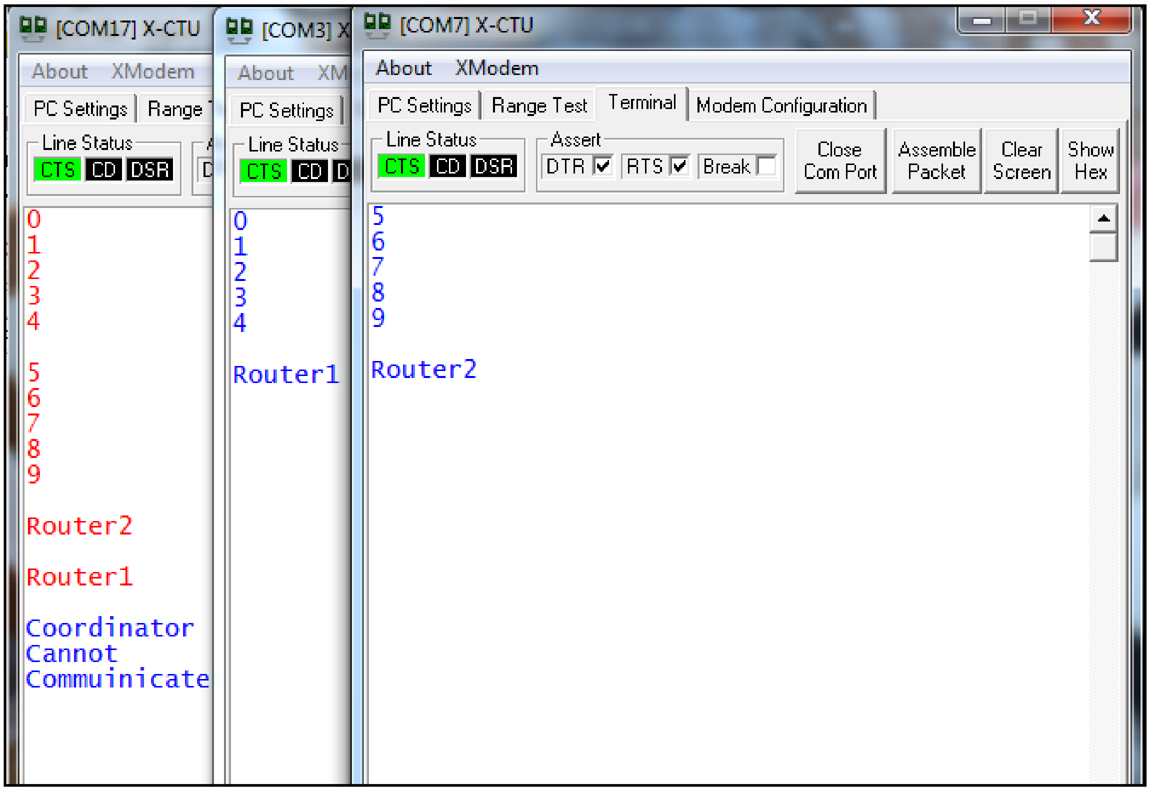


Figure 4.4: Communication between XBEE's

The communication between XBEE's was shown in Figure 4.4. The blue colour represents transmit while red colour represents receive. COM17 (The first part in Figure 4.18) is the coordinator while COM3 and CPM17 are the routers.

When COM3 send data out in string "0 1 2 3 4", they were received and shown in COM17. Then, COM7 send data out "5 6 7 8 9 Router2" and were displayed in the coordinator. However, when COM17 send out "Coordinator Cannot Communicate", they were not displayed by the Routers. Hence, the one way communication between the XBEE's was accomplished.

4.2 Counting Algorithm

The counting was done in PIC18F4620 by using internal memory counter. The input via push button will increment the counter number which was stored in the memory. However, by using the BCD decoder, the number cannot be loaded and sent directly to the 7448 decoder.

For example, when loaded until number 20, if stored directly to the display, it was shown 0001 0100 and if this number was loaded to the 7-segment display, it will show “14” rather than “20”. To show “20”, the input at the BCD decoder has to be “0010 0000” showing 2 and 0 separately.

To solve this problem, different coding was written, loading different number when the counting reaches different value. For example, when the counting loaded until 187, the program will call to load “0001 1000 0111” to store to the BCD decoder.

This solution will cause longer program because each number until 999 has to be defined a separate set of coding. Complete coding of the algorithm for the circuit is available as in Appendix A, Appendix B & Appendix C

4.3 Outcome of the Project

The prototype of the Portable Queue Control System was done. To test the system, three consequences were tested with the system.

4.3.1 Base Panel

Figure 4.5 shows the base panel of the queue control system. The 7-segments panel will show the current number of the customer being served, while the XBEE above the display was used to receive input from counter panels to call the numbers. Initial number is 000 after the system resets.

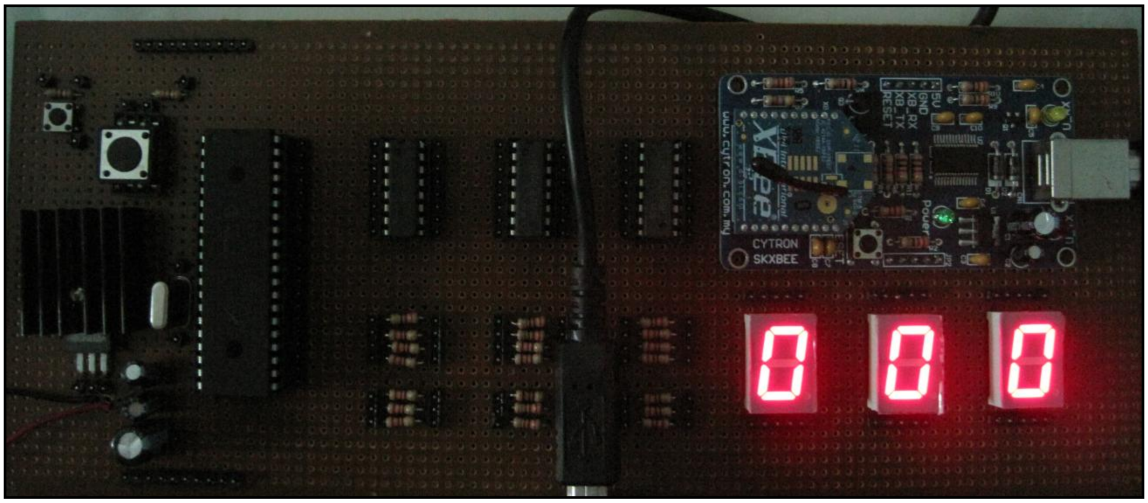


Figure 4.5: Base Panel

4.3.2 Counter Panel

Figure 4.6 shows the counter panel of the system. In this simple panel, one seven segment was located to show the counter number, XBEE module was used to transmit data to the base panel and a buzzer to notify the customer when a number was called.

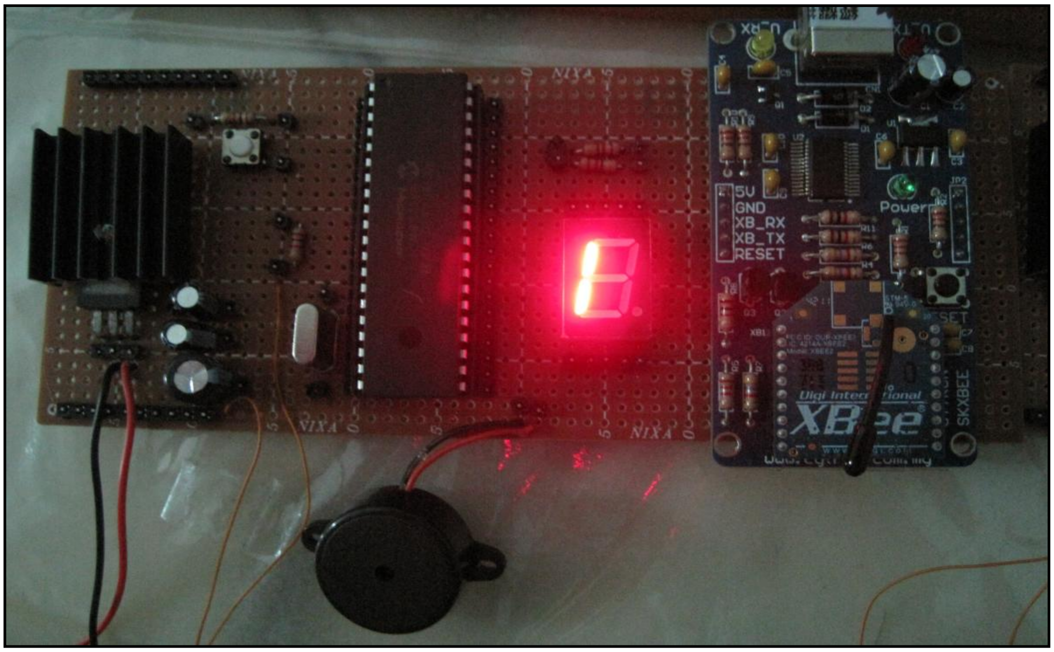


Figure 4.6: Counter Panel

4.3.3 Initial Condition

Initially, after the system is powered up, the base panel will show the number of 000 as in Figure 4.7. Both of the counter panels will show no number on the display and buzzer is not on. However, the system is ready to function and is able to send command to the base panel at anytime.

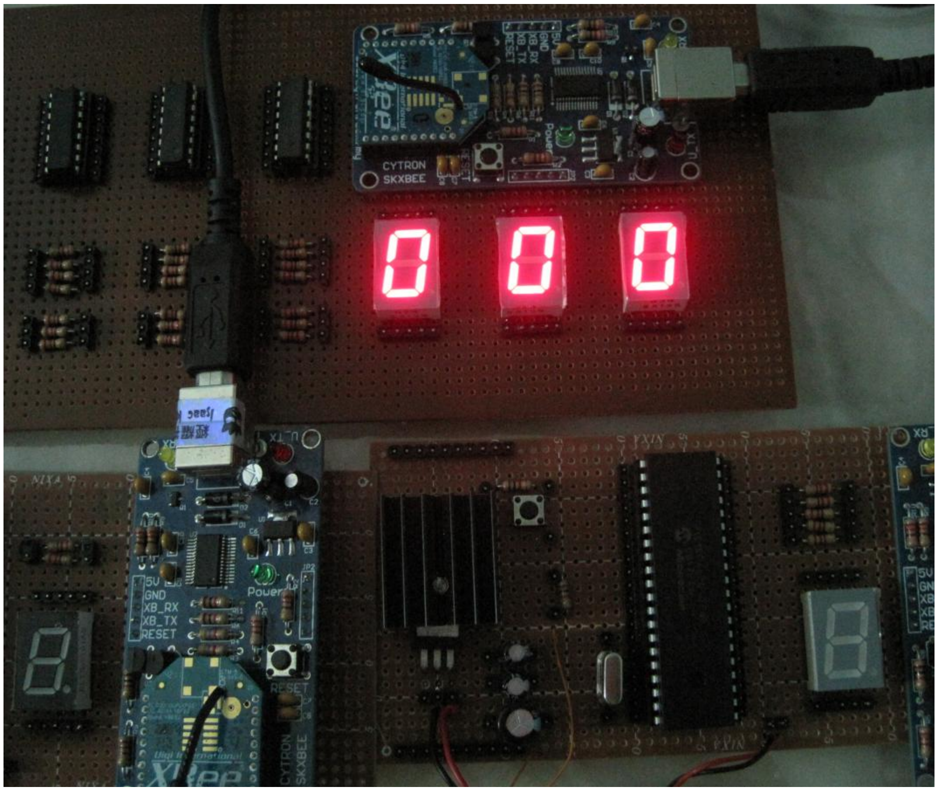


Figure 4.7: Initial Condition

4.4 Test Run

To test the system, the communication between the base panel and the counter were tested. Twenty numbers were input to the base panel, assuming there are twenty customers waiting to be served. Then, the various communications between the panel and the counters were tested.

4.4.1 Continuous Counting in Counter 1

The first test was done to test the communication between counter one and the base panel which will be shown in Figure 4.8, Figure 4.9 and Figure 4.10. The initial condition was shown in Figure 4.7 where initial number is 000 and both counter numbers were not shown.

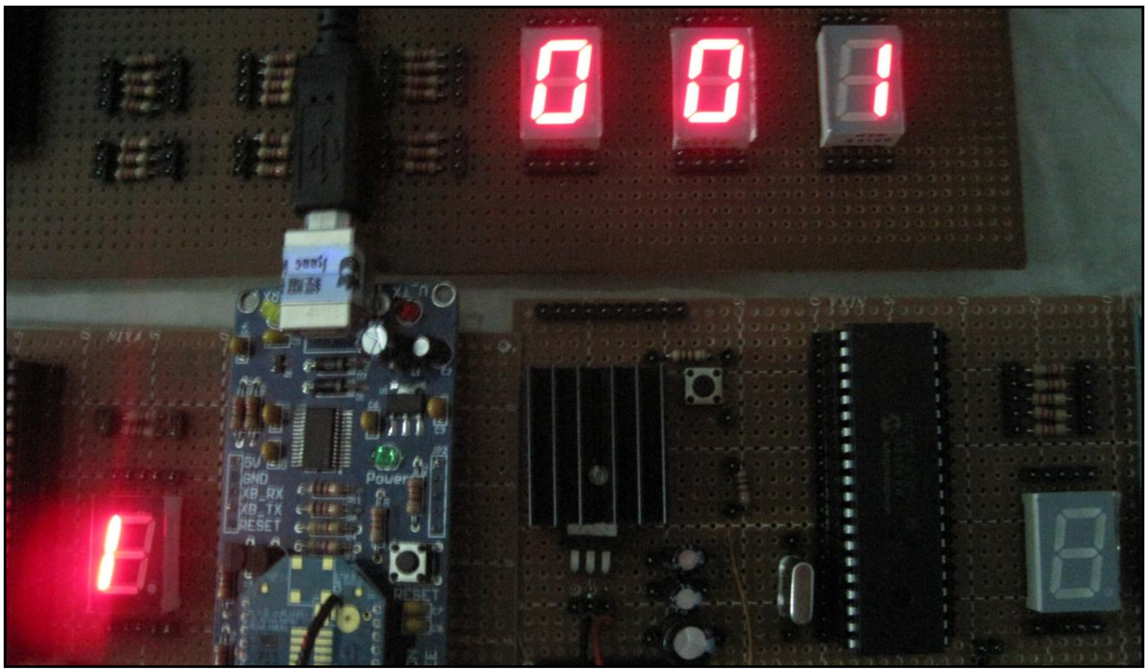


Figure 4.8: Test 1 - 001

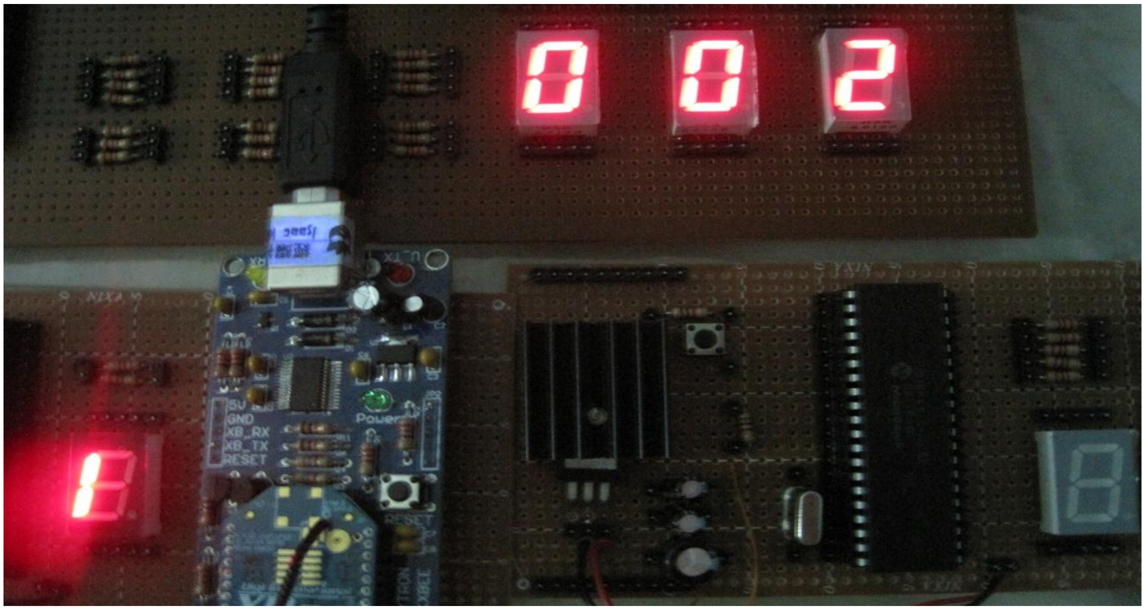


Figure 4.9: Test 1 – 002

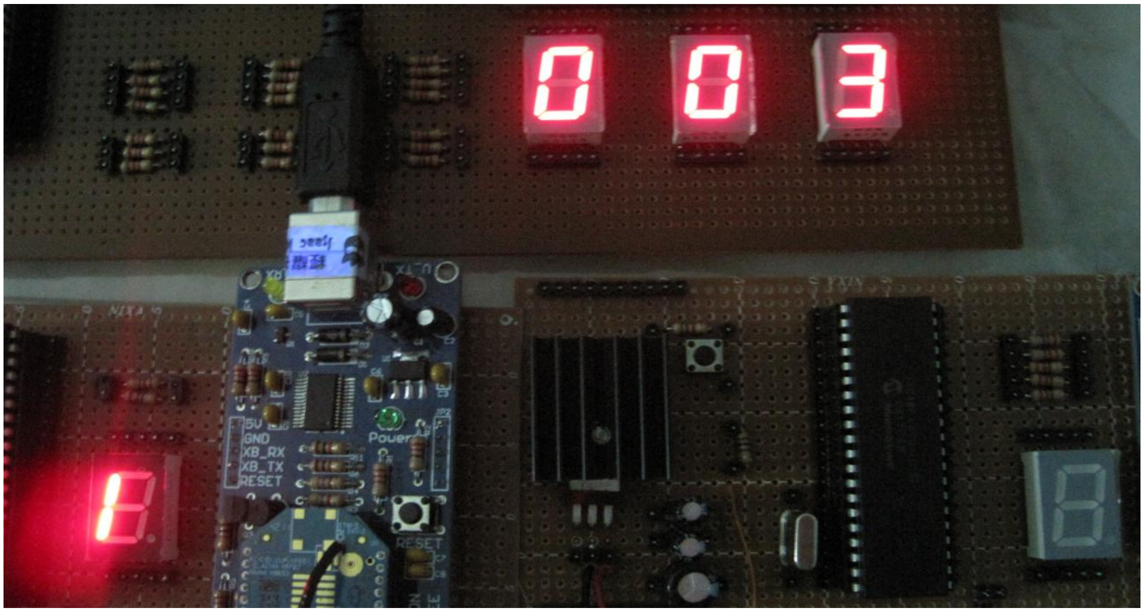


Figure 4.10: Test 1 – 003

As shown in the three figures above, the communication between base panel and counter 1 was tested. The push button at counter 1 was pressed for three times continuously, and the display panel changed like shown in the figures above. Each time when the push button at counter 1 was pressed, the 7-segment at counter 1 was blinking showing number "1" and the buzzer buzzed. Hence, the calling for the counting of numbers for counter 1 is functioning well.

4.4.2 Counting Alternately via Counter 1 and Counter 2

The next test is to test on both of the counters functioning together. Both counters were activated in turn to test whether the system is functioning well. The result is shown in figures below.

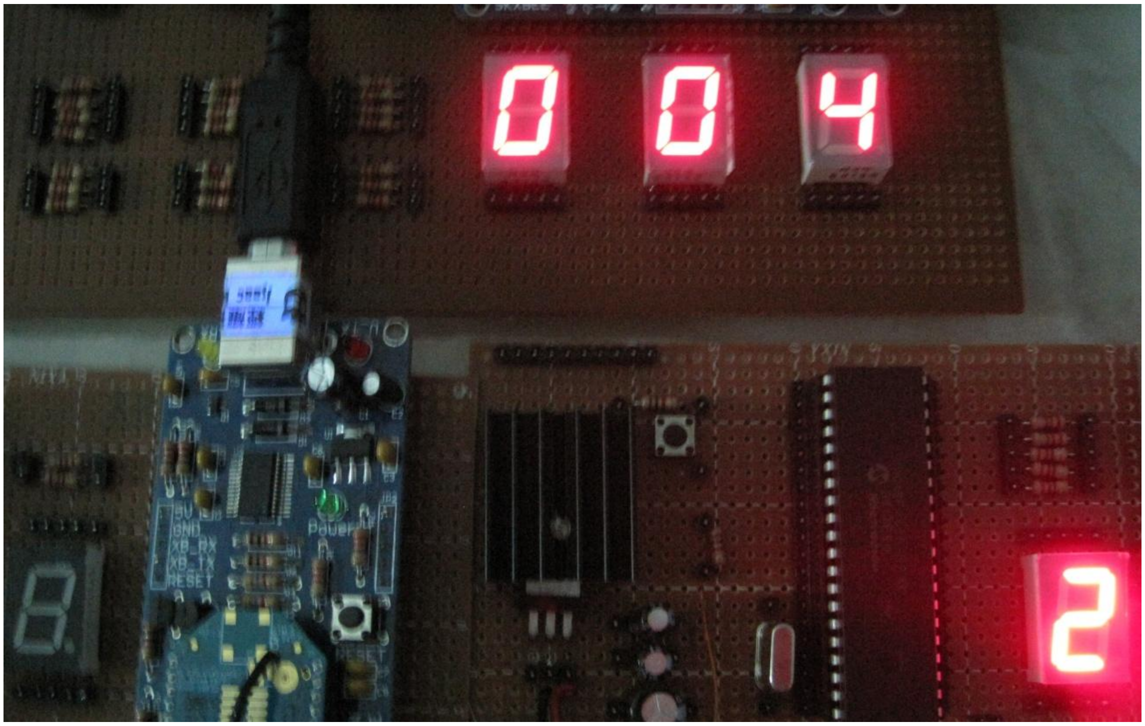


Figure 4.11: Test 2 – 004 (Counter 2)

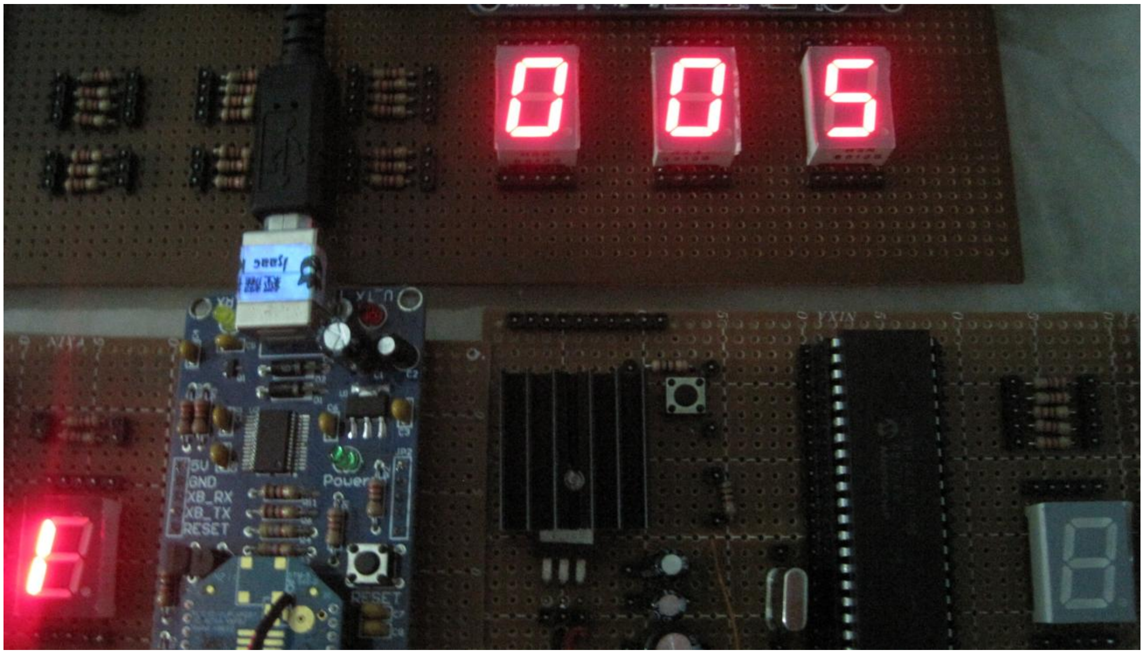


Figure 4.12: Test 2 – 005 (Counter 1)

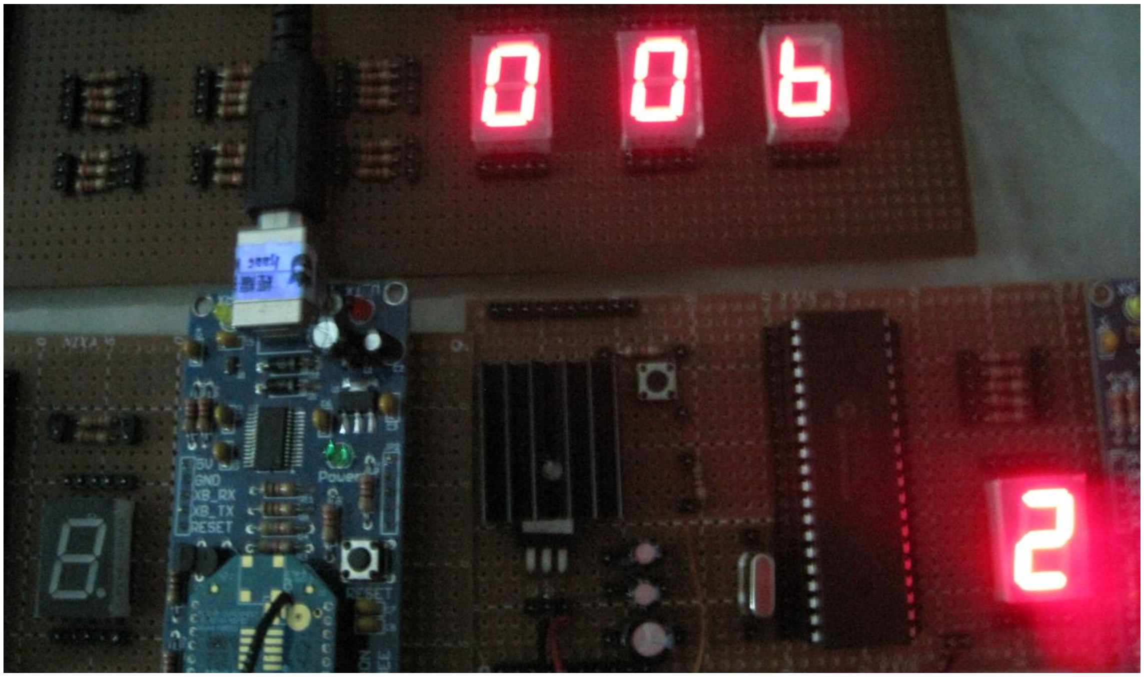


Figure 4.13: Test 2 – 006 (Counter 2)

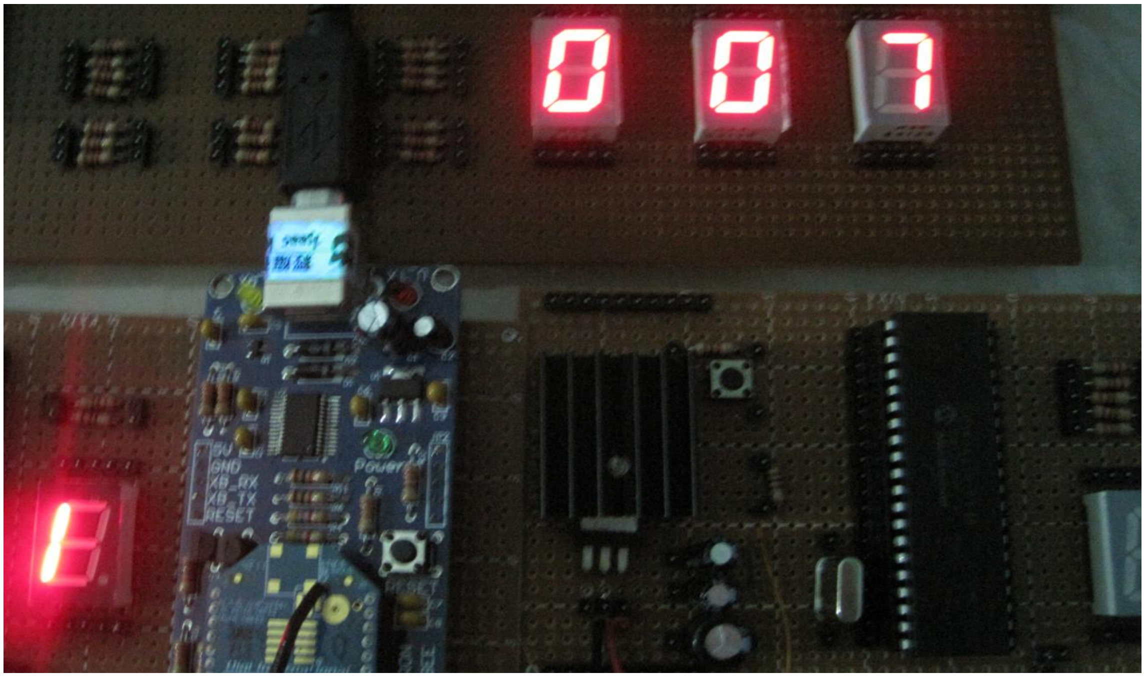


Figure 4.14: Test 2 – 007 (Counter 1)

In Figure 4.11, the push button at the counter 2 was pressed, the number at the base panel jumped from 003 to 004. The test continued with pressing the push button at counter 1, counter 2 and counter 1 again, while the number jumped from 004 to 005, 006 and 007 accordingly. Each time the counter button was pressed, the buzzer will buzz and the counter will show their counter number in blinking. Hence, the communication using both counters was going well.

4.4.3 Continuous Counting in Counter 2

The next test was done using single counter, the counter 2, to test the single communication between counter 2 and the base panel. The result is shown in the figures below:

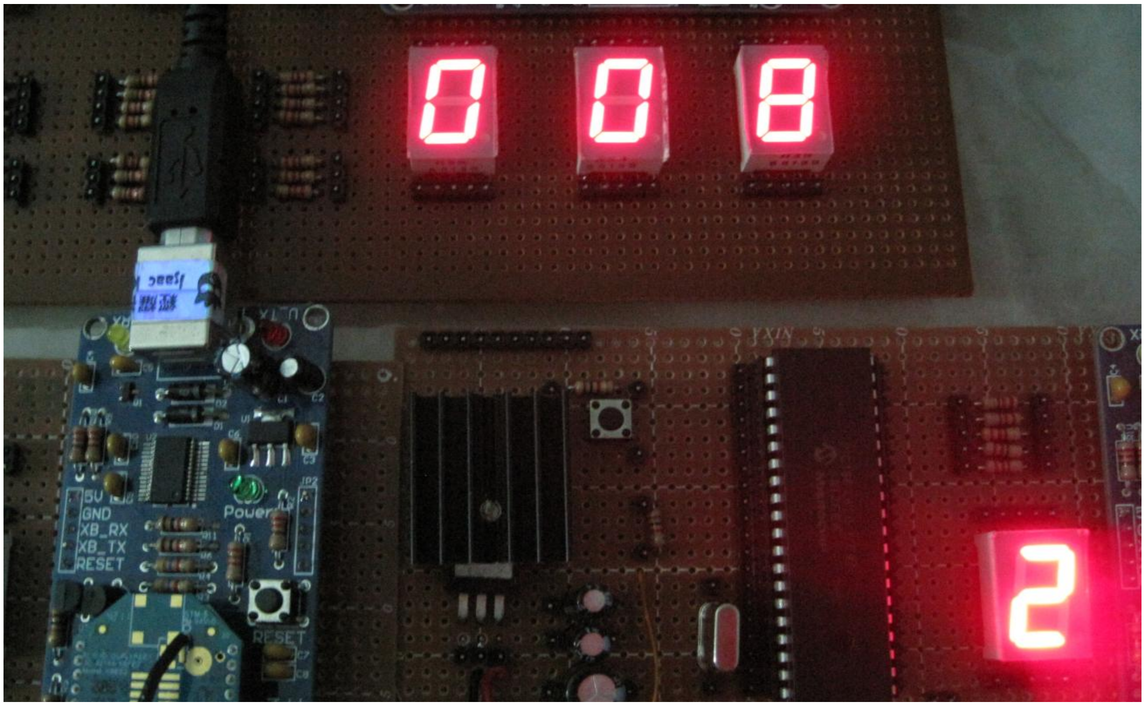


Figure 4.15: Test 3 – 008 (Counter 2)

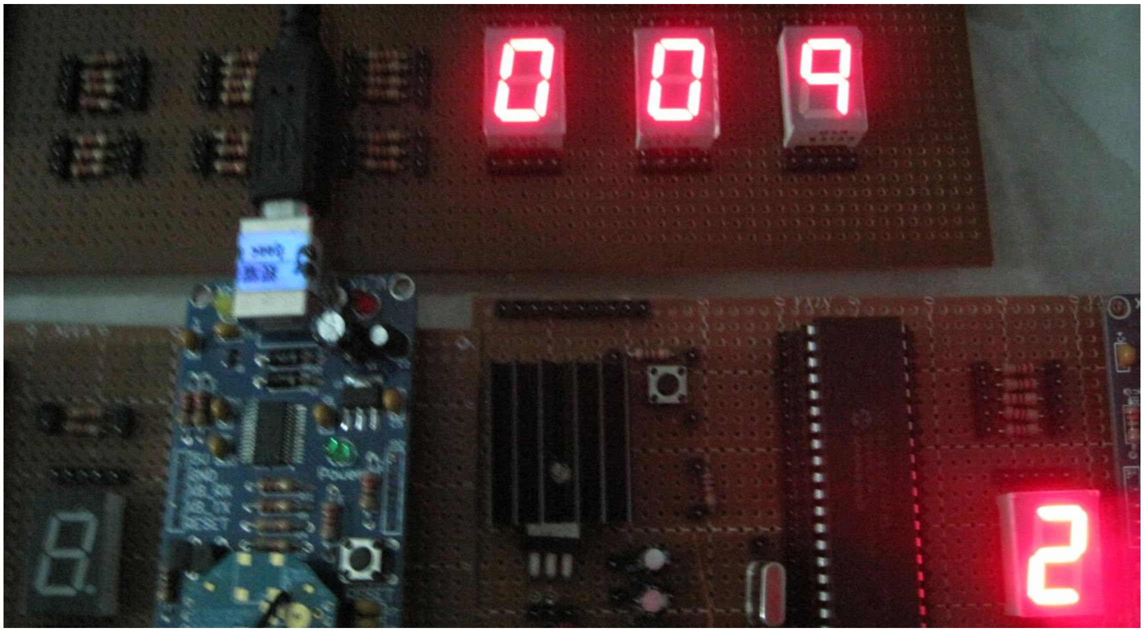


Figure 4.16: Test 3 – 009 (Counter 2)

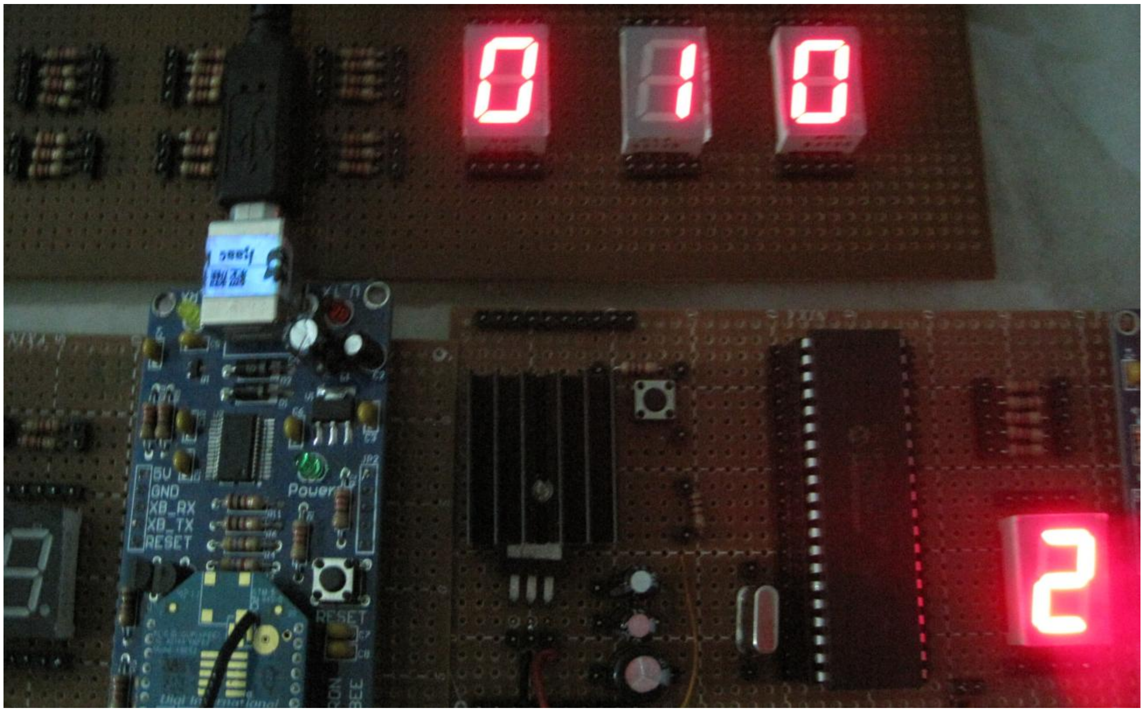


Figure 4.17: Test 3 – 010 (Counter 2)

Testing the counter 2 solely, the 7 segment display did blink accordingly and the buzzer did turn on accordingly, meaning that the counter 2 is functioning. Besides that, when the number counts continuously, the number able to jump to 010, changing the second digit of the number display, showing that the BCD converter counting is working as well.

4.4.4 Counting Range Test

The last test done was the counting range test. This test aims to ensure that the system is able to count until the desired maximum value (999). The test was done by pressing continuously the push button from 000 until the maximum value. The result was shown at Figure 4.18 and Figure 4.19.

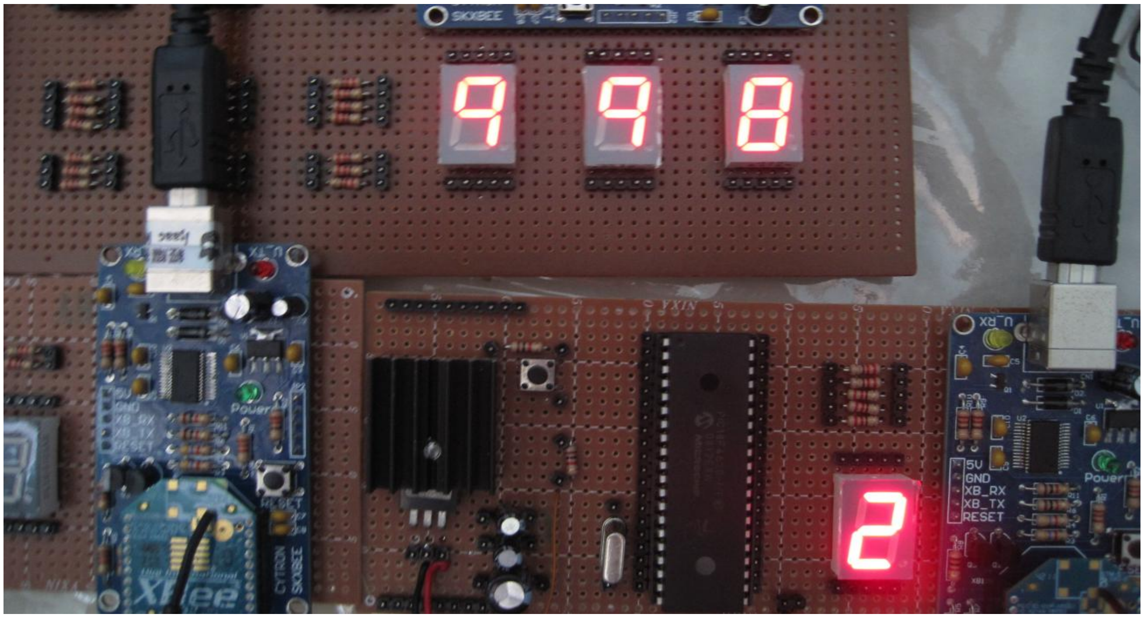


Figure 4.18: Test 4 – 998 (Counter 2)

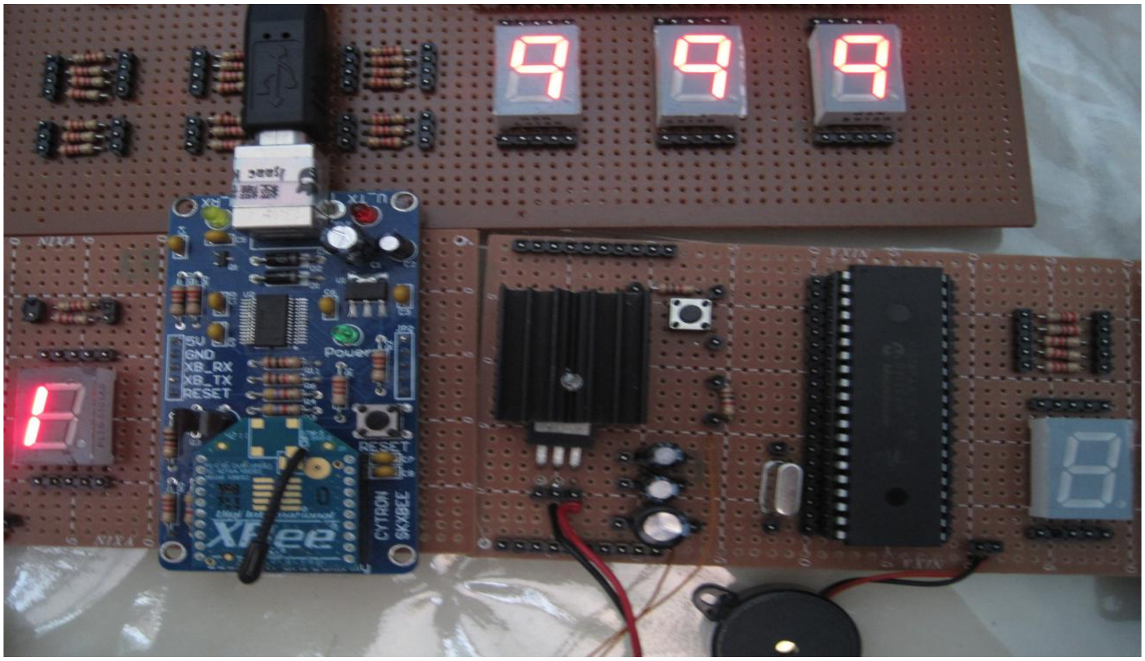


Figure 4.19: Test 5 – 999 (Counter 1)

As shown in Figure 4.18 and Figure 4.19, the counter is able to count until the expected range, 999. Hence, the targeted objective is achieved.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

This chapter concludes on the task done and discusses several recommendations to upgrade the project. Targeted market and commercialization will be discussed as well in the chapter.

5.1 Conclusion

The objective of the project was accomplished. The Portable Queue Management System is fully developed. The main objective of the final year project is to develop a microcontroller based Portable Queue Management System which is small in size and is able to be moved among places. The sub-objectives of the project are to develop a queuing algorithm for wirelessly connected queue management system and to design a microcontroller based electronic system connecting a main controller and remote call button for queue management system.

The prototype of the project consists of one base panel with display and two counter panels to control the display panel. All of the panels are able to communicate with one another via wireless communication. The size of the system is small and it is convenient to be carried to be used during outdoor activities.

For the algorithm of the prototype, the system allows the record of the number of customer waiting and will compare with the number of displayed. When the remote button at the counter was pressed, the counting of the display will be increased and showing the number of the next customer.

Although there is some limitations and space for improvement for this project, the product is complete and is able to function at any outdoor services for trial. By using this Portable Queue Control System, the queue management for the customer can be improved and will bring for increase in the sales or services.

5.2 Recommendations

In order to improve the project, a few limitations can be overcome to provide a better service in practical.

5.2.1 Duplex Communication

In this project, the design between the XBEE's for the panel and the counters were using one way communication. To improve the system, duplex (two way communication) can be established so that the counters and the panels can be communicating with each other. With that, the system will be able to display the number called at the counter.

5.2.2 Priority Queue Sequence

As per discussed in the existing queue control system available in market in literature review (Chapter 2.1), there is some system using priority queue system. If the system is designed using priority queue sequence, different services can be sorted and priority can be given to serve those customers with higher priority or different services can be sorted out to be served at different counters.

5.2.3 Number of Servicing Counters

In real life business, the number of counters providing service might not be limited to two only. Hence, more servicing counters can be installed to provide a better service to the customers.

5.3 Commercialize of Product

The design of the product is ready for sales in the market. Processing in bulk, the cost of one Portable Queue Control System is estimated to be at cost of RM200. Having margin of 50%, the system can be sold to targeted dealers or computer shops at price of RM300.

Investigating the current market share, servicing and business contributes to most of the income of the society. Since the portable queue management system is suitable for any business which aims to provide services outdoor, the product is estimated to be having a big market share in the future after commercialized.

References

[1] *Tescos Queue Management*

[2] http://dictionary.reference.com/browse/queue

[3] *Proceeding of the 2011 IEEE Students' Technology Symposium* 14-16 January, 2011, lIT Kharagpur

[4] The comparison of Wi-Fi, Bluetooth and ZigBee http://www.sena.com/blog/?p=359

[5] *Queue admin: The effects of an advance queue management system on Management System on Barbershop Administration* Kinnis K. Gosha, Master of Science, May 10, 2007, (B.S., Albany State University, May 2005)

[6] Gross, Donald; Carl M. Harris *Fundamentals of Queueing Theory. Wiley* (1998).

[7] *Queuing Model* Wikipedia contributors, http://en.wikipedia.org/wiki/Queueing_model

[8] *Wi-Fi (wireless networking technology)* Encyclopædia Britannica.

[9] *Cytron Contributors, "Is XBee ZigBee?"* http://tutorial.cytron.com.my/2011/03/06/is-xbee-zigbee/

[10] *5 Ways Queue Management Systems Can Help Your Business* http://allyouneedislists.com/world-business/online-business/5-ways-queue-management-systems-can-help-your-business/

[11] Mohamad Fazli bin Alias *Front Desk Customer Service For Queue Management System*, November, 2007

[12] *How-to: Setting up XBee ZNet 2.5 (Series 2) modules*
http://www.embedded.arch.ethz.ch/xbee-setup.pdf

[13] *XBee 1mW Wire Antenna - Series 2*
http://cytron.com.my/viewProduct.php?pcode=XBEE-S2&name=XBee%201mW%20Wire%20Antenna%20-%20Series%202

[14] *What is the difference between ZigBee and XBee?*
http://www.quora.com/What-is-the-difference-between-ZigBee-and-XBee

[15] *The Major Differences in the XBee Series 1 vs. the XBee Series 2*
http://www.digi.com/support/kbase/kbaseresultdetl?id=2213

[16] Roberts, Lawrence G.; Wessler, Barry D. (1970), "Computer network development to achieve resource sharing", AFIPS '70 (Spring): Proceedings of the May 5–7, 1970, spring joint computer conference, New York, NY, USA

[17] Digital Display Circuits
http://www.allaboutcircuits.com/worksheets/display.html

[18] *IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. (2007 revision). IEEE-SA. 12 June 2007.

[19] Jeffrey S. Beasley; Gary M. Miller (2008). Modern Electronic Communication (9 ed.). pp. 4–5.

[20] D.C. Folz; *Microwave and Radio Frequency Application.* Virginia Polytechnic Institute and State University. Proceedings of the Fourth World Congress on Microwave and Radio Frequency Applications. November, 2004. Austin, Texas.

APPENDIX

APPENDIX A

CODING FOR BASE PANEL

```
#include <18F4620.h>                  // PIC18F4620 HEADER FILE
#fuses XT,NOWDT,NOLVP,NOPROTECT        // EXTERNAL CLOCK, NO WATCH
DOG TIMER, NO LOW VOLTAGE PROGRAMMING
                                                      //
#define  SW1   PIN_B0
#define  SW2   PIN_B1

#define  LCD_E   PIN_D0
#define  LCD_RS  PIN_D1
#define  LCD_RW  PIN_D2
#define  LCD_D4  PIN_D4
#define  LCD_D5  PIN_D5
#define  LCD_D6  PIN_D6
#define  LCD_D7  PIN_D7

#define  BCD11   PIN_C0          //
#define  BCD12   PIN_C1          //
#define  BCD13   PIN_C2          //
#define  BCD14   PIN_C4          //

#define  BCD21   PIN_A0          //
#define  BCD22   PIN_A1          //
#define  BCD23   PIN_A2          //
#define  BCD24   PIN_A3          //

#define  BCD31   PIN_B2          //
#define  BCD32   PIN_B3          //
#define  BCD33   PIN_B4          //
#define  BCD34   PIN_B5          //

void zero()
{
  output_low(BCD31);
  output_low(BCD32);
  output_low(BCD33);
  output_low(BCD34);

  output_low(BCD21);
  output_low(BCD22);
  output_low(BCD23);
  output_low(BCD24);
```

```
  output_low(BCD11);
  output_low(BCD12);
  output_low(BCD13);
  output_low(BCD14);
}

void one()
{
  output_low(BCD31);
  output_low(BCD32);
  output_low(BCD33);
  output_low(BCD34);

  output_low(BCD21);
  output_low(BCD22);
  output_low(BCD23);
  output_low(BCD24);

  output_low(BCD11);
  output_low(BCD12);
  output_low(BCD13);
  output_high(BCD14);
}

void two()
{
  output_low(BCD31);
  output_low(BCD32);
  output_low(BCD33);
  output_low(BCD34);

  output_low(BCD21);
  output_low(BCD22);
  output_low(BCD23);
  output_low(BCD24);

  output_low(BCD11);
  output_low(BCD12);
  output_high(BCD13);
  output_low(BCD14);
}

void three()
{
  output_low(BCD31);
  output_low(BCD32);
```

```
  output_low(BCD33);
  output_low(BCD34);

  output_low(BCD21);
  output_low(BCD22);
  output_low(BCD23);
  output_low(BCD24);

  output_low(BCD11);
  output_low(BCD12);
  output_high(BCD13);
  output_high(BCD14);
}

void four()
{
  output_low(BCD31);
  output_low(BCD32);
  output_low(BCD33);
  output_low(BCD34);

  output_low(BCD21);
  output_low(BCD22);
  output_low(BCD23);
  output_low(BCD24);

  output_low(BCD11);
  output_high(BCD12);
  output_low(BCD13);
  output_low(BCD14);
}

void five()
{
  output_low(BCD31);
  output_low(BCD32);
  output_low(BCD33);
  output_low(BCD34);

  output_low(BCD21);
  output_low(BCD22);
  output_low(BCD23);
  output_low(BCD24);

  output_low(BCD11);
  output_high(BCD12);
```

```
  output_low(BCD13);
  output_high(BCD14);
}

void six()
{
  output_low(BCD31);
  output_low(BCD32);
  output_low(BCD33);
  output_low(BCD34);

  output_low(BCD21);
  output_low(BCD22);
  output_low(BCD23);
  output_low(BCD24);

  output_low(BCD11);
  output_high(BCD12);
  output_high(BCD13);
  output_low(BCD14);
}

void seven()
{
  output_low(BCD31);
  output_low(BCD32);
  output_low(BCD33);
  output_low(BCD34);

  output_low(BCD21);
  output_low(BCD22);
  output_low(BCD23);
  output_low(BCD24);

  output_low(BCD11);
  output_high(BCD12);
  output_high(BCD13);
  output_high(BCD14);
}

void eight()
{
  output_low(BCD31);
  output_low(BCD32);
  output_low(BCD33);
  output_low(BCD34);
```

```
  output_low(BCD21);
  output_low(BCD22);
  output_low(BCD23);
  output_low(BCD24);

  output_high(BCD11);
  output_low(BCD12);
  output_low(BCD13);
  output_low(BCD14);
}

void nine()
{
  output_low(BCD31);
  output_low(BCD32);
  output_low(BCD33);
  output_low(BCD34);

  output_low(BCD21);
  output_low(BCD22);
  output_low(BCD23);
  output_low(BCD24);

  output_high(BCD11);
  output_low(BCD12);
  output_low(BCD13);
  output_high(BCD14);
}

void ten()
{
  output_low(BCD31);
  output_low(BCD32);
  output_low(BCD33);
  output_low(BCD34);

  output_low(BCD21);
  output_low(BCD22);
  output_low(BCD23);
  output_high(BCD24);

  output_low(BCD11);
  output_low(BCD12);
  output_low(BCD13);
  output_low(BCD14);
```

```
}

void eleven()
{
  output_low(BCD31);
  output_low(BCD32);
  output_low(BCD33);
  output_low(BCD34);

  output_low(BCD21);
  output_low(BCD22);
  output_low(BCD23);
  output_high(BCD24);

  output_low(BCD11);
  output_low(BCD12);
  output_low(BCD13);
  output_high(BCD14);
}

void twelve()
{
  output_low(BCD31);
  output_low(BCD32);
  output_low(BCD33);
  output_low(BCD34);

  output_low(BCD21);
  output_low(BCD22);
  output_low(BCD23);
  output_high(BCD24);

  output_low(BCD11);
  output_low(BCD12);
  output_high(BCD13);
  output_low(BCD14);
}

void thirteen()
{
  output_low(BCD31);
  output_low(BCD32);
  output_low(BCD33);
  output_low(BCD34);

  output_low(BCD21);
```

```
  output_low(BCD22);
  output_low(BCD23);
  output_high(BCD24);

  output_low(BCD11);
  output_low(BCD12);
  output_high(BCD13);
  output_high(BCD14);
}

void fourteen()
{
  output_low(BCD31);
  output_low(BCD32);
  output_low(BCD33);
  output_low(BCD34);

  output_low(BCD21);
  output_low(BCD22);
  output_low(BCD23);
  output_high(BCD24);


  output_low(BCD11);
  output_high(BCD12);
  output_low(BCD13);
  output_low(BCD14);
}

void fifteen()
{
  output_low(BCD31);
  output_low(BCD32);
  output_low(BCD33);
  output_low(BCD34);

  output_low(BCD21);
  output_low(BCD22);
  output_low(BCD23);
  output_high(BCD24);


  output_low(BCD11);
  output_high(BCD12);
  output_low(BCD13);
  output_high(BCD14);
```

```
}

void sixteen()
{
  output_low(BCD31);
  output_low(BCD32);
  output_low(BCD33);
  output_low(BCD34);

  output_low(BCD21);
  output_low(BCD22);
  output_low(BCD23);
  output_high(BCD24);

  output_low(BCD11);
  output_high(BCD12);
  output_high(BCD13);
  output_low(BCD14);
}

void seventeen()
{
  output_low(BCD31);
  output_low(BCD32);
  output_low(BCD33);
  output_low(BCD34);

  output_low(BCD21);
  output_low(BCD22);
  output_low(BCD23);
  output_high(BCD24);

  output_low(BCD11);
  output_high(BCD12);
  output_high(BCD13);
  output_high(BCD14);
}

void eighteen()
{
  output_low(BCD31);
  output_low(BCD32);
  output_low(BCD33);
  output_low(BCD34);
```

```
  output_low(BCD21);
  output_low(BCD22);
  output_low(BCD23);
  output_high(BCD24);

  output_high(BCD11);
  output_low(BCD12);
  output_low(BCD13);
  output_low(BCD14);
}

void nineteen()
{
  output_low(BCD31);
  output_low(BCD32);
  output_low(BCD33);
  output_low(BCD34);

  output_low(BCD21);
  output_low(BCD22);
  output_low(BCD23);
  output_high(BCD24);

  output_high(BCD11);
  output_low(BCD12);
  output_low(BCD13);
  output_high(BCD14);
}

void twenty()
{
  output_low(BCD31);
  output_low(BCD32);
  output_low(BCD33);
  output_low(BCD34);

  output_low(BCD11);
  output_low(BCD12);
  output_high(BCD13);
  output_low(BCD14);

  output_low(BCD11);
  output_low(BCD12);
  output_low(BCD13);
  output_low(BCD14);
}
```

```
void main()
{
static int i=0;
static int j=0;

set_tris_a(0x00);              // SET ALL PORT A AS OUTPUT PORT
set_tris_b(0x03);
set_tris_c(0x00);              // SET ALL PORT C AS OUTPUT PORT
set_tris_d(0x00);              // SET ALL PORT D AS OUTPUT PORT

output_c(0x00);                // RESET PORT C

lcd_init();

  while(TRUE)
  {
    {

    if (kbhit())
    {
    x = getc();
    }
      i++;
      delay_ms(500);
    }
    if(!input(PIN_B1))
    {
        j++;
      }
      delay_ms(500);
    }

    if(j==0)
    {
      zero();
    }
    if(j==1)
    {
      one();
    }
    if(j==2)
    {
      two();
    }
    if(j==3)
```

```
{
  three();
}
if(j==4)
{
  four();
}
if(j==5)
{
  five();
}
if(j==6)
{
  six();
}
if(j==7)
{
  seven();
}
if(j==8)
{
  eight();
}
if(j==9)
{
  nine();
}
if(j==10)
{
  ten();
}
if(j==11)
{
  eleven();
}
if(j==12)
{
  twelve();
}
if(j==13)
{
  thirteen();
}
if(j==14)
{
  fourteen();
```

```
    }
    if(j==15)
    {
      fifteen();
    }
    if(j==16)
    {
      sixteen();
    }
    if(j==17)
    {
      seventeen();
    }
    if(j==18)
    {
      eighteen();
    }
    if(j==19)
    {
      nineteen();
    }
    if(j==20)
    {
      twenty();
    }
  }
}
```

APPENDIX B

CODING FOR COUNTER 1

```
#include <18F4620.h>                   // PIC18F4620 HEADER FILE
 #include <stdlib.h>
 #fuses  HS,NOWDT,NOLVP,NOPROTECT                  //  EXTERNAL  CLOCK,  NO
WATCH DOG TIMER, NO LOW VOLTAGE PROGRAMMING

 #define    e     PIN_B4   //Enable Pin e 7 segment
 #define    f     PIN_B5   //Enable pin f 7 segment
 #define    buzzer   PIN_B7
 #define    on    PIN_C0
 #define    SW1   PIN_A0

 void run()
 {
  output_low(on);
  delay_ms(800);
  output_high(on);
  output_high(buzzer);
  delay_ms(500);
  output_high(e);
  output_high(f);
  output_low(buzzer);
  delay_ms(500);
  output_low(e);
  output_low(f);
  output_high(buzzer);
  delay_ms(500);
  output_high(e);
  output_high(f);
  output_low(buzzer);
  delay_ms(500);
  output_low(e);
  output_low(f);
  output_high(buzzer);
  delay_ms(500);
  output_high(e);
  output_high(f);
  delay_ms(500);
  output_low(e);
  output_low(f);
  delay_ms(500);
  output_high(e);
  output_high(f);
```

```
 output_low(buzzer);
 delay_ms(500);
 output_low(e);
 output_low(f);
 delay_ms(500);
 output_high(e);
 output_high(f);
 delay_ms(500);
 output_low(e);
 output_low(f);
 delay_ms(500);
 output_high(e);
 output_high(f);
 delay_ms(500);
 output_low(e);
 output_low(f);
 delay_ms(500);
 output_high(e);
 output_high(f);
 delay_ms(500);
 output_low(e);
 output_low(f);
 }

void main()
{
set_tris_c(0x00);          // SET ALL PORT C AS OUTPUT PORT
set_tris_b(0x00);          // SET ALL PORT B AS OUTPUT PORT
set_tris_a(0x01);          // SET FIRST PORT OF PORT A AS INPUT
output_a(0x01);            // RESET PORT A
output_b(0x00);            // RESET PORT B
output_c(0x01);            // RESET PORT C

 while(TRUE)                   // always repeat program below
{
if (!input(SW1))           // when call button pressed
 { while(!input(SW1))            //take action after button released

    delay_ms(100);//short delay to debounce the switch
    run();
}
}
}
```

APPENDIX C

CODING FOR COUNTER 2

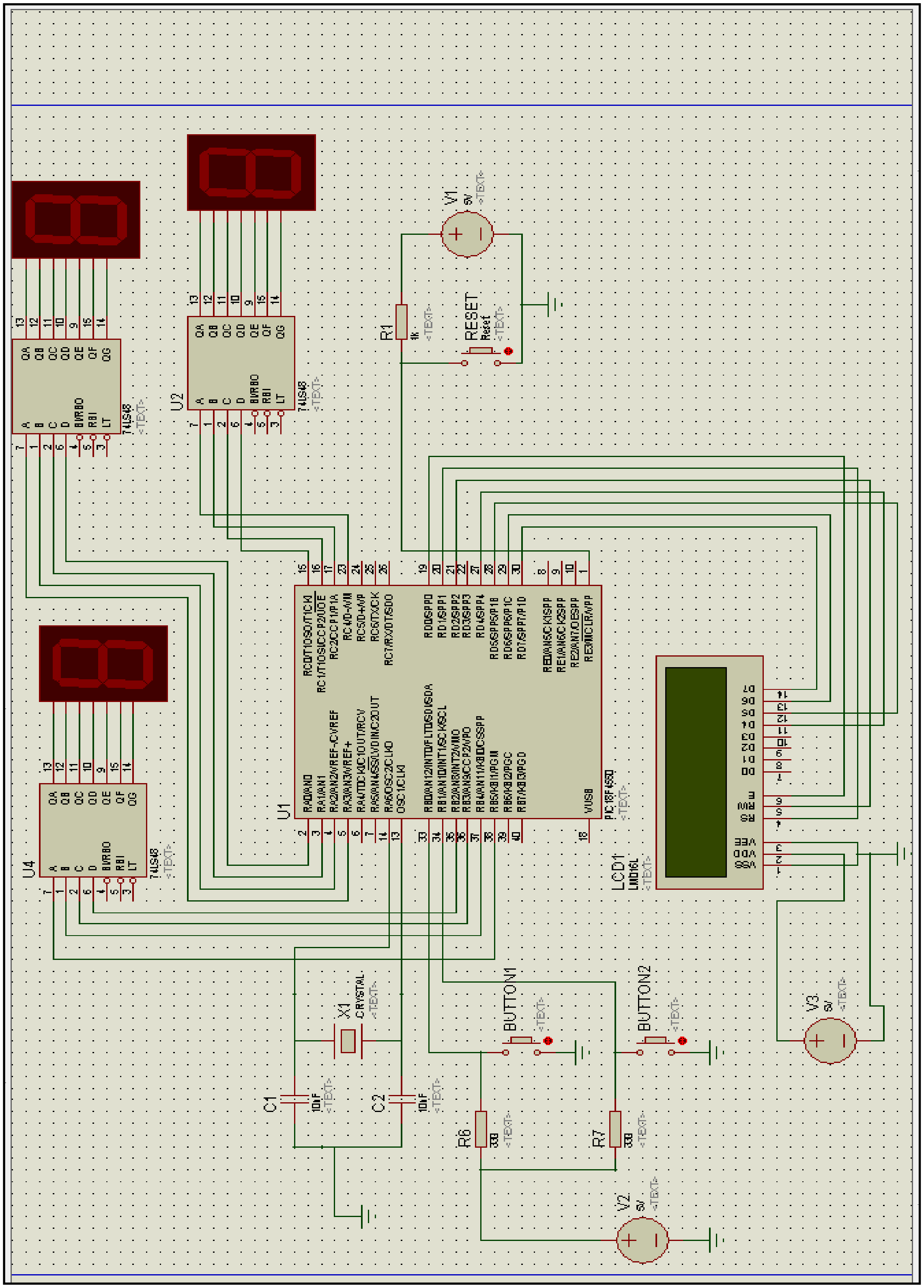
```
#include <18F4620.h>                // PIC18F4620 HEADER FILE
#include <stdlib.h>
#fuses HS,NOWDT,NOLVP,NOPROTECT       // EXTERNAL CLOCK, NO
WATCH DOG TIMER, NO LOW VOLTAGE PROGRAMMING

#define    a  PIN_B0   //Enable pin 1 7 segment
#define    b  PIN_B1   //Enable pin b 7 segment
#define    d  PIN_B3   //Enable pin d 7 segment
#define    e  PIN_B4   //Enable Pin e 7 segment
#define    g  PIN_B6   //Enable pin g 7 segment
#define    buzzer   PIN_B7
#define    on    PIN_C0
#define    SW1   PIN_A0

void run()
{
 output_low(on);
 delay_ms(800);
 output_high(on);
 output_high(buzzer);
 delay_ms(500);
 output_high(a);
 output_high(b);
 output_high(d);
 output_high(e);
 output_high(g);
 output_low(buzzer);
 delay_ms(500);
 output_low(a);
 output_low(b);
 output_low(d);
 output_low(e);
 output_low(g);
 output_high(buzzer);
 delay_ms(500);
 output_high(a);
 output_high(b);
 output_high(d);
 output_high(e);
 output_high(g);
 output_low(buzzer);
 delay_ms(500);
```

```
output_low(a);
output_low(b);
output_low(d);
output_low(e);
output_low(g);
output_high(buzzer);
delay_ms(500);
output_high(a);
output_high(b);
output_high(d);
output_high(e);
output_high(g);
delay_ms(500);
output_low(a);
output_low(b);
output_low(d);
output_low(e);
output_low(g);
delay_ms(500);
output_high(a);
output_high(b);
output_high(d);
output_high(e);
output_high(g);
output_low(buzzer);
delay_ms(500);
output_low(a);
output_low(b);
output_low(d);
output_low(e);
output_low(g);
delay_ms(500);
output_high(a);
output_high(b);
output_high(d);
output_high(e);
output_high(g);
delay_ms(500);
output_low(a);
output_low(b);
output_low(d);
output_low(e);
output_low(g);
delay_ms(500);
output_high(a);
output_high(b);
```

```
 output_high(d);
 output_high(e);
 output_high(g);
 delay_ms(500);
 output_low(a);
 output_low(b);
 output_low(d);
 output_low(e);
 output_low(g);
 delay_ms(500);
 output_high(a);
 output_high(b);
 output_high(d);
 output_high(e);
 output_high(g);
 delay_ms(500);
 output_low(a);
 output_low(b);
 output_low(d);
 output_low(e);
 output_low(g);
}

void main()
{
set_tris_c(0x00);            // SET ALL PORT C AS OUTPUT PORT
set_tris_b(0x00);            // SET ALL PORT B AS OUTPUT PORT
set_tris_a(0x01);
output_a(0x01);
output_b(0x00);              // RESET PORT B
output_c(0x01);              // RESET PORT C


 while(TRUE)                     // always repeat program below
{
if (!input(SW1))             //
 { while(!input(SW1))            //take action after sw1 is released
   {}
    delay_ms(100);//short delay to debounce the switch
    run();

 }
 }}
```

APPENDIX D

CIRCUIT FOR BASE PANEL



APPENDIX E

CIRCUIT FOR COUNTER