PORTABLE SECURITY SYSTEM

By

NURZAFIRAH BINTI KHALID

DISSERTATION

Submitted to the Electrical & Electronic Engineering Programme
in Partial Fulfillment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronic Engineering)

Universiti Teknologi PETRONAS

Bandar Seri Iskandar

31750 Tronoh

Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Portable Security System

by

Nurzafirah binti Khalid

A project dissertation submitted to the
Electrical & Electronic Engineering Department
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
BACHELOR OF ENGINEERING (Hons)
(ELECTRICAL & ELECTRONIC ENGINEERING)

Approved by,

(Dr. Mohd Zuki bin Yusoff)
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
DECEMBER 2010

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein has not been undertaken or performed by unspecified sources or persons.

NURZAFIRAH BINTI KHALID

ABSTRACT

A security system is a system used to prevent from and protect against assault, damage, fire, fraud, invasion of privacy, theft, unlawful entry, and other such occurrences caused by deliberate action. A portable security system is designed to be small, easy to carry or move and it is normally being powered by a battery. The operation of a security system is to detect, to delay, and to alarm. Normally, alarm systems use several sensor systems. This project is about designing and developing a Portable Security System that helps to detect a door knob touch, to sense an opening door and to sense a movement in a room. A portable security system can be an option for those who rent apartments and not having the permission of the landlord to begin drilling to install security systems. The system is used to ensure safety and confidential usage of a room too. The significant part of the system is the portability because we make use of a transmitter and receiver, so that an appropriate security breach messages can still be communicated from the door to the person inside the room. The initial stage of the project involves research and feasibility studies of a security system. The system design requires the selection of hardware, including sensors, the transmitter and receiver and a microcontroller. The security system prototype is built part by part. And then, the input and output components of the microcontroller is interfaced with the transmitter and receiver module. The system should trigger the alarm and user will require to key in a special code to stop the alarm.

ACKNOWLEDGEMENTS

Firstly I would like to express my utmost gratitude to Allah the Almighty for giving me the chance and strengths to face the odds and obstacles in completing my Final Year Project (FYP).

A special thank is extended to my FYP supervisor, Dr Mohd Zuki bin Yusoff, senior lecturer in Electrical & Electronic Department, who has supervised, guided and supported me throughout this project. Besides, my special thanks go to Miss Siti, a Microcontroller lab technician who has also given me great guidance in my entire FYP period.

I also would like to thank my parents, siblings, relatives and friends who helped me in the entire duration of this project. Their concerns and compassion towards me during this period are greatly appreciated, and I could not have endured this whole period without them.

My appreciations go to all the people who took their time out of their busy schedules to guide me throughout my FYP journey.

Thank you.

TABLE OF CONTENTS

ABSTRACT v
ACKNOWLEDGEMENTS vi
LIST OF TABLES ix
LIST OF FIGURES x
LIST OF ABBREVIATIONS xi

CHAPTER 1: INTRODUCTION

1.1 Background of Study 1
1.2 Problem Statement 2
1.3 Objectives 2
1.4 Scope of Study 3
1.5 Organization of Report 3

CHAPTER 2: LITERATURE REVIEW

2.1 Background Information 5
2.2 Inputs 6
2.2.1 Capacitance Sensor 6
2.2.2 Accelerometer 7
2.2.3 Passive infra red (PIR) sensor 10
2.3 Input Output Peripherals Devices 12
2.4 Microcontroller 13
2.5 Transmitter and Receiver Module 15
2.6 C Compiler 18
2.7 Summary 19

CHAPTER 3: METHODOLOGY

3.1 Procedure Identification 20

3.2 Procedure Flowchart 21

3.3 Proposed Work for Prototype Installation 24

3.3.1 Transmitter Module 24

3.3.2 Receiver Module 27

3.4 Tools and Equipment Required 29

3.4.1 Software 29

3.4.2 Hardware 29

3.5 Summary 31

CHAPTER 4: RESULT AND DISCUSSION

4.1 Transmitter Module 32

4.1.1 Capacitance Sensor 35

4.1.2 Accelerometer 37

4.2 Receiver Module 38

4.2.1 Problem Analysis 40

4.3 Summary 40

CHAPTER 5: CONCLUSION AND RECOMMENDATION

5.1 Conclusion 41

5.2 Recommendation 43

REFERENCES 44

APPENDICES 45

APPENDIX A – Gantt Chart for FYP I Semester Jan 2010 46

APPENDIX B – Gantt Chart for FYP II Semester July 2010 47

APPENDIX C – Source Code of Transmitter Module 48

APPENDIX D – Source Code of Receiver Module 52

LIST OF TABLES

Table 1 Pinout descriptions of an accelerometer 9

Table 2 Configurations of keys and pins of a keypad... 13

Table 3 Truth table for the simple touch sensor circuit 26

Table 4 Voltage reading while analyzing the behavior of a simple touch sensor circuit 37

LIST OF FIGURES

Figure 1 Flow of a security system 5
Figure 2 Applying a voltage supply causes positive and negative charges to collect on each plate 6
Figure 3 The charges that move back and forth between the plates and voltage supply create an alternating current which is detected by the sensor 7
Figure 4 The g-cell in accelerometer 8
Figure 5 The pin diagram of the MMA1260KEG accelerometer 9
Figure 6 Component of a PIR sensor 11
Figure 7 A MURATA IRS-A200ST01-R1 PIR sensor 11
Figure 8 A 3-by-4 keypad pin layout 12
Figure 9 Basic building blocks of a computer 13
Figure 10 Block diagram of a PIC16F877 microcontroller 14
Figure 11 Transmitter module operation flow 16
Figure 12 Receiver module operation flow 17
Figure 13 Flowchart describing the procedure for designing and building the portable security system 22
Figure 14 Connection at transmitter module 24
Figure 15 Schematic diagram of circuit to replace capacitance sensor 25
Figure 16 Connection at receiver module 27
Figure 17 Schematic diagram shows connection to PIR sensor 28
Figure 18 Developed transmitter module 33
Figure 19 Installation of a transmitter module on the door 34
Figure 20 The touch sensor circuit 36
Figure 21 The accelerometer laminated on a printed board 38
Figure 22 Developed receiver module 39

LIST OF ABBREVIATIONS

PIR	Passive Infra Red
CMOS	Complementary Metal–oxide–semiconductor
ASICs	Application Specific Integrated Circuits
CPU	Central Processing Unit
RAM	Random Access Memory
ROM	Read-only Memory
I/O	Input Output
FYP	Final Year Project
PIC	Peripheral Interface Controller
LED	Light-emitting Diode
IC	Integrated Circuit
LCD	Liquid Crystal Display
USB	Universal Serial Bus

CHAPTER 1

INTRODUCTION

This project titled 'Portable Security System' is conducted to propose a reliable and portable system to ensure safety and confidential usage of a room. The system also provides an option for those who rent in an apartment to install a security system in their apartment without causing damages to the apartment's wall structure. The system enables the detection of three types of security breach that is to detect a door knob touching, to sense door opening and to sense movements in a room. Throughout the implementation process for the project, there is several scope of study involved. This includes determining the appropriate sensors to be utilized, programming the microcontroller and interfacing the inputs and outputs of the system.

1.1 Background of Study

A portable security system is a system that is used to monitor and alert user on intruders. It is used to prevent of and protect against assault, damage, fire, fraud, invasion of privacy, theft, unlawful entry, and other such occurrences caused by deliberate action [1]. The project basically focusing on replicating or imitating the security system that are in the market nowadays. The main focus of this project is to construct a reliable and portable security system that are small in size, has a quick and simple setup, can be moved easily and can connect a module to another module wirelessly in a range of distance. The system comprises sensors, an alarm, a keypad, a transmitter, a receiver and a microcontroller.

1.2 Problem Statement

Anyone living in an apartment might find a portable security system handy since the portable security system provides an option for him or her to install a security system in the apartment without causing damages to the apartment's wall structure. This could help to ensure that the apartment is well protected. The traveler might use the portable security system in his or her hotel for safety purposes as a portable security system is small in size, easy to setup and most importantly mobile. Also, the system is useful when someone wants to use a room to do something that is highly confidential that he or she does not want anyone to interrupt in that moment. Within that moment, one might want to know if someone else is trying to get into the room and one would want enough time to react to the visitor(s). In all these cases, one might find that a portable security system is useful.

1.3 Objectives

This portable security system is able to sense a person touching the doorknob and to sense an open door which will warn and allow the user sufficient time to react to the uninvited visitor, whether to allow the visitor to enter or not. The system also would provide enough time to react if the user accidentally leaves the door opened. Thirdly, one may find it useful if the system is able to sense someone or any movements in the room. Most importantly, the entire system is portable so that an appropriate security breach messages can still be communicated from the door to the person inside the room.

1.4 Scope of Study

The scope of study includes sensors, the transmitter and receiver and the MCU. The feasibility study on the system includes:

1. The appropriate sensors to achieve aim and objectives of project.
2. Integration of the input and the output for control and monitoring.
3. Familiarization to the microcontroller technology and C programming

However, the research on the hardware and software aspects is a continuous activity to build a working prototype and get more design ideas to produce a better product.

1.5 Organization of Report

This report is divided into five chapters namely the introduction, literature review, methodology, results and discussions and conclusion and recommendation. The introduction section includes the background of the project, the problem statement, the objectives and scope of study, which clarify the boundary project work to ensure the feasibility within the given time frame. The literature review provides the background information on the project. It contains all the relevant theories, facts and data which are relevant to the objectives and the findings of this project. The methodology section refers to the methods and procedures used to achieve the objectives of the project. The results and discussion is a section where the finding or outcome of the project work is presented. All the gathered data from the project work is presented in the form of tables and figures. Conclusion includes the most significant findings in relation to the objectives of the project. This section also includes recommendations for future project work.

CHAPTER 2

LITERATURE REVIEW

To achieve the aims and objectives of the project, research and feasibility studies of a security system have been analyze. In this section, a clearer understanding about the software and hardware used will be provided. A portable security system should be able to detect the following security breach:

1) Touching of a door knob.
2) Opening of a door.
3) Movement in a room

There are three types of sensors being used to detect each security breach. A capacitance sensor is used to detect a door knob being touched. An accelerometer is used to detect an opening of a door; meanwhile, a Passive Infra Red (PIR) sensor is used to sense movements in a room. The main equipment used in the system is the three sensors, a transmitter module, a receiver module, a microcontroller unit and input output peripherals (e.g., keypad and buzzer). Meanwhile, the software used to program the microcontroller is C-Compiler.

2.1 Background Information

A portable security system should be small in size, should have a quick and simple setup, should be able to be moved easily can be accessed wirelessly in a range of distance. The flow of a portable security system is illustrated in Figure 1.

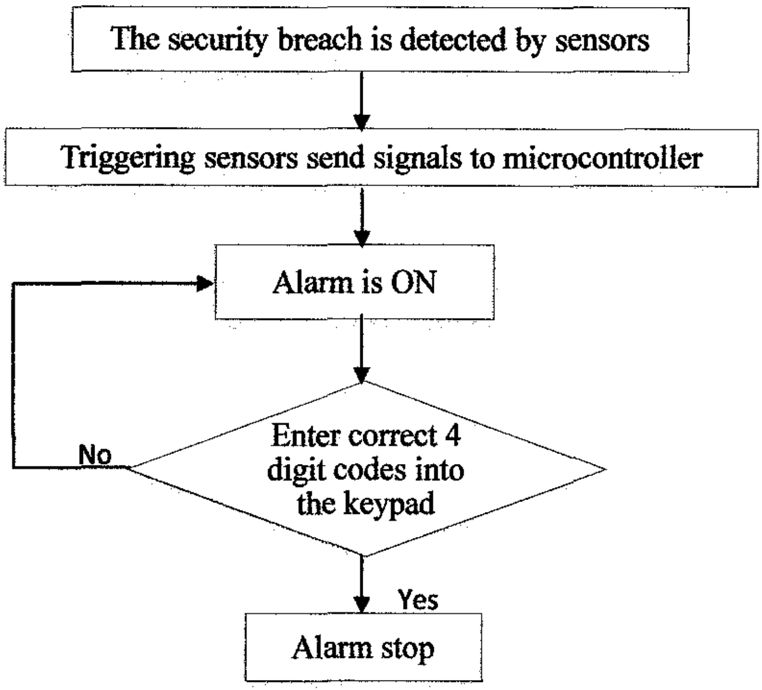


Figure 1 : Flow of a security system

When the security breach is detected by sensors, a signal is sent to the microcontroller. This will activate the alarm to notify the user on the danger and on the intruder in the room. When the user enters the correct four digit codes into the keypad, the alarm will stop. But, when the user does not enter the correct four digit codes into the keypad, the alarm will still be activated for a period of time before it is automatically stopped by the microcontroller.

2.2 Inputs

2.2.1 Capacitance sensor

A capacitance sensor operates based on the principle of a capacitor which consists of two conductive plates with a space between them that responds to the voltage difference applied to them. The sensor can detect a change in capacitance when something or someone approaches or touches the sensing surface of the sensor. When a given voltage is applied between the two plates of any capacitor, an electric field is created between the plates. This causes the positive and negative charges to collect on each plate as illustrated in Figure 2.

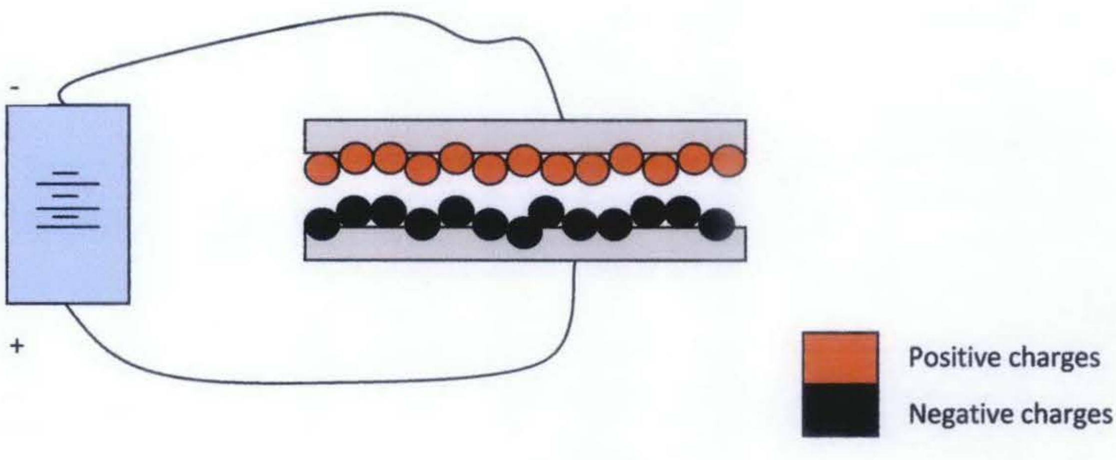


Figure 2 : Applying a voltage supply causes positive and negative charges to collect on each plate

When an object nears the sensing surface, it enters the electrostatic field of the plates and causes an alternating voltage in the circuit. An alternating voltage causes the positive and negative charges on the plates to move back and forth between the objects. The moving of the charges creates an alternating electric current which is detected by the sensor as illustrated in Figure 3 [2]. As the target moves away from the sensing area the sensor output switches back to its original state.

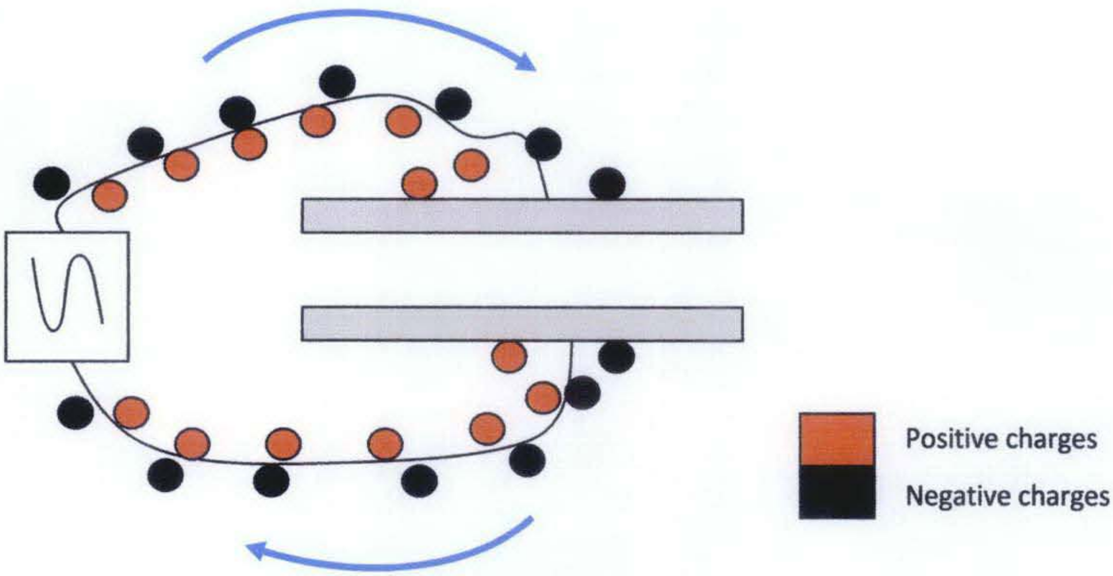


Figure 3: The charges that move back and forth between the plates and voltage supply create an alternating current which is detected by the sensor

2.2.2 Accelerometer

An accelerometer is used to detect the opening of a door. It is an instrument used to measure acceleration forces. The concept of accelerometer is to measure the force exerted by restraints placed on a reference mass to hold its position fixed in an accelerating body, since it is difficult to measure acceleration directly [3]. The output is usually either a varying electrical voltage or displacement of a moving pointer over a fixed scale.

The accelerometer consists of a capacitive sensing cell (g-cell) and a CMOS signal conditioning ASIC contained in a single integrated circuit package. The g-cell can be modeled as two stationary plates with a moveable plate in-between. The center plate can be deflected from its rest position by subjecting the system to an acceleration (illustrated in Figure 4). When the center plate deflects, the distance from it to one fixed plate will increase by the same amount that the distance to the other plate decreases. The change in distance is a measure of acceleration [4].

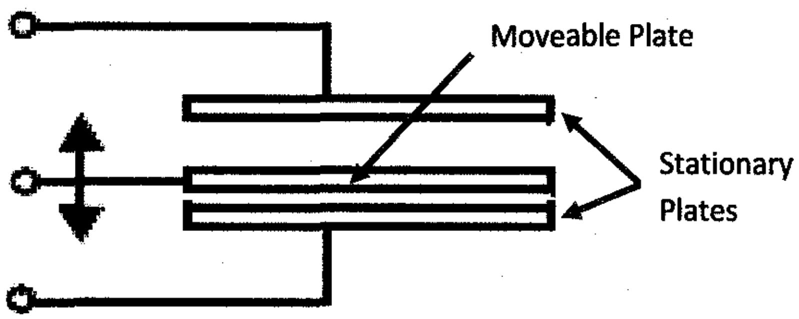


Figure 4 : The g-cell in accelerometer

The accelerometer MMA1260KEG by Freescale Semiconductor is used to detect an open door. This type of accelerometer is used because it is able to produce an output voltage that is linearly related to the acceleration. For this project, the only thing needed is to detect fluctuations from a stable value to detect an open door. The configuration of an accelerometer is described in Figure 5 and Table 1 as below.

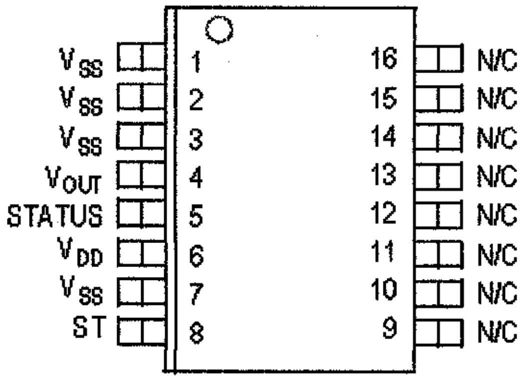


Figure 5: The pin diagram of the MMA1260KEG accelerometer [4]

Table 1 : Pinout descriptions of an accelerometer [4]

Pin No	**Pin Name**	**Description**	**Connection in module**
1 to 3	V_{ss}	Redundant connections to the internal V_{SS}.	Unconnected
4	V_{out}	Output voltage of an accelerometer.	Connected to port A.0 in PIC Microcontroller
5	STATUS	Logic output pin to indicate fault.	Connected to port B.7 in PIC Microcontroller
6	V_{DD}	The power supply input.	Connected to 5V source
7	V_{SS}	The power supply ground	Connected to ground
8	ST	Logic input pin used to initiate self-test.	Connected to port B.6 in PIC Microcontroller
9 to 13	Trim Pins	Used for factory trim	Unconnected
14 to 16	-	No internal connections	Unconnected

2.2.3 Passive infra red (PIR) sensor

To sense movements in a room, a Passive Infra Red (PIR) sensor is used. A PIR sensor is an electronic device which measures infrared light radiating from objects in its field of view. The PIR device can be thought of as a kind of infrared 'camera' which remembers the amount of infrared energy focused on its surface [5]. A PIR sensor is sensitive to 'heat', or rather the infrared light that is emitted by warm or hot objects. If the amount of infrared energy focused on the sensor changes within a configured time period, the device will switch the state of the alarm output relay from low (OFF state) to high (ON state).

Figure 6 below shows the component of a PIR sensor. From theory, a Fresnel lens will focused on the desired protected area determine by users. Once a person entered the monitored area, the Fresnel lens will sense an increase of temperature on the protected area. The increase of temperature will activates the detection input of the PIR sensor. In this project, we applied this theory when a human (an infrared source with one temperature) passed in front of a wall (an infrared source with another temperature); the PIR sensor will detect the changing patterns of a temperature on its protected area.

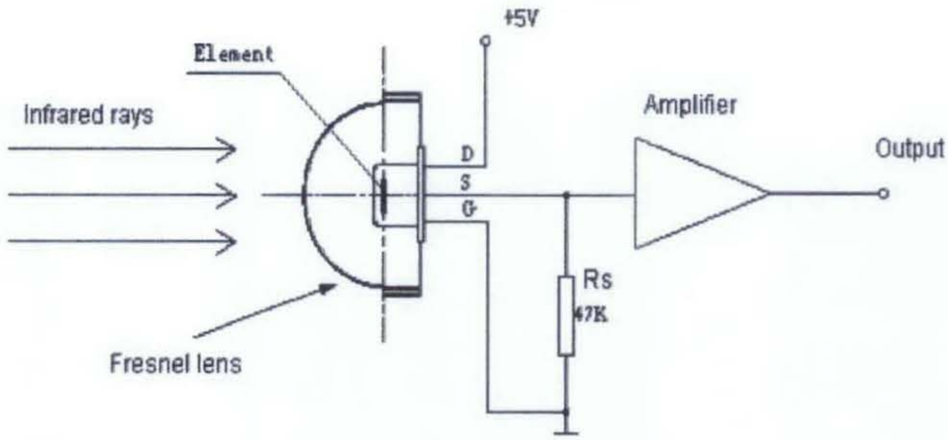


Figure 6: Component of a PIR sensor

The PIR sensor used in the project is IRS-A200ST01-R1 from MURATA (as illustrated in Figure 7). Its working principle is that the higher the sensed temperature is, the lower the output voltage from the PIR sensor will be.

Figure 7 : A MURATA IRS-A200ST01-R1 PIR sensor

2.3 Input Output Peripherals Devices

The input output peripherals devices used in the system are an alarm and a keypad. The alarm is triggered when one or all security breach is detected. When the security breach is detected, a signal or a beeper is triggered off, which will not stop making a sound until an authorized person provides a key. In this context, the key is the correct 4 digit codes being entered in the keypad. When the user enters the correct 4 digit codes into the keypad, the alarm will stop. But, when the user does not enter the correct 4 digit codes into the keypad, the alarm will still be turned on for a period of time before it is stopped automatically by the microcontroller.

A 3-by-4 keypad is used as the output channel of the portable security system. The keypad is used as an interface between the user and the system itself. When the user had initiates the correct 4 digit codes into the keypad, the alarm will stop. As illustrated in Figure 8 below, the keypad itself has 7 pins. The configurations of the keys and pins of the 3-by-4 keypad are described in Table 2 below.

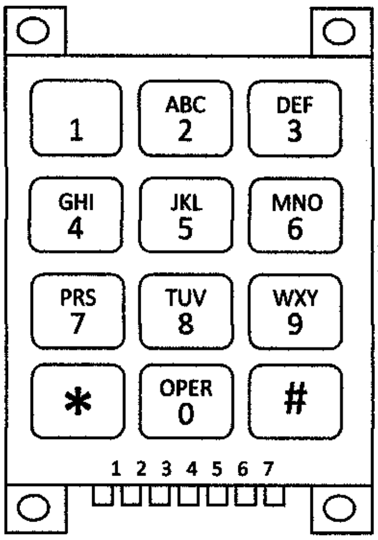


Figure 8 : A 3-by-4 keypad pin layout

Table 2 : Configurations of keys and pins of a keypad

Pin No.	Key Configurations
1	1, 2, 3
2	4, 5, 6
3	7, 8, 9
4	*, 0, #
5	1, 4, 7, *
6	2, 5, 8, 0
7	3, 6, 9, #

2.4 Microcontroller

A microcontroller is a small computer on a single integrated circuit which consists of the following building blocks[6] as shown in Figure 9:

- CPU – the part that does all logic and arithmetic functions
- RAM – storage for programs and/or program variables
- ROM – read-only parts of program
- I/O - connection to external devices

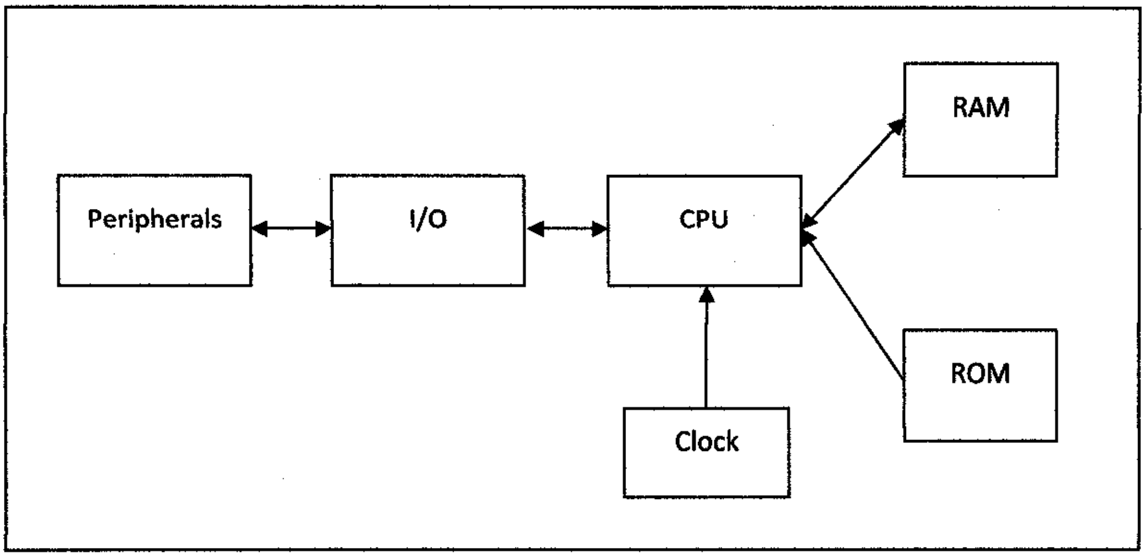


Figure 9: Basic building blocks of a computer

A PIC microcontroller used is the PIC16F877. The PIC16F877 is used because it has 40 pins on it as the system has many input and output connected to it. A PIC microcontroller block diagram is illustrated in Figure 10.

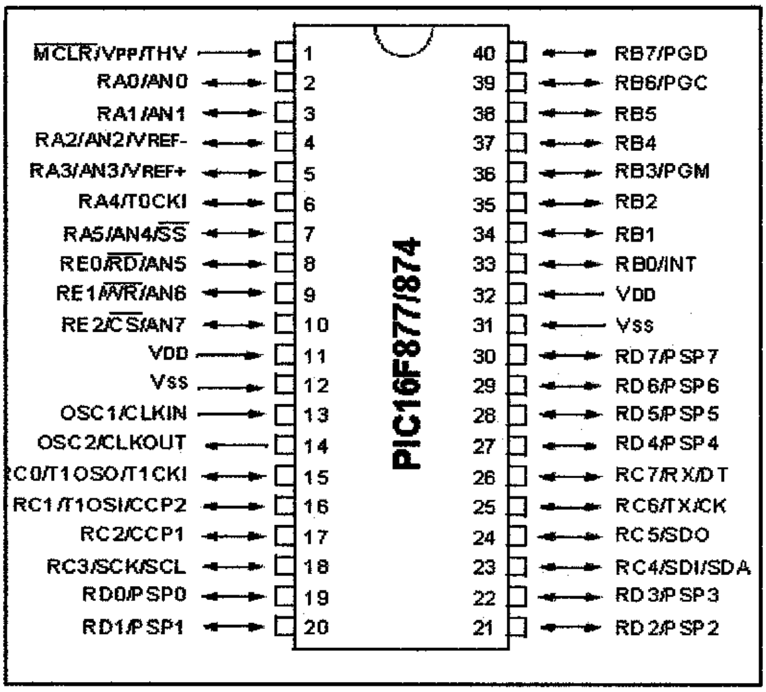


Figure 10 : Block diagram of a PIC16F877 microcontroller[7]

In the system, two microcontrollers are used. One is used at the transmitter module, and another is used at the receiver module. Following are the reasons why microcontrollers are used in this system:

- Cost: Microcontrollers with the supplementary circuit components are much cheaper than a computer.
- Size and weight: Microcontrollers are compact and light compared to computers. This is a requirement to classify the system as a portable system.

- Simple applications: The system requires very few number of I/O and the code is relatively small, which does not require extended amount of memory and a simple LCD display is sufficient as a user interface.

All the three sensors are connected to the microcontroller as the input components. When any of these sensors is triggered, a signal will be send to the microcontroller. The microcontroller will send a signal to activate the alarm, which acts as an output. Also the microcontroller is programmed to interface with the keypad. The keypad is used to stop the alarm by entering the 4 digit code.

2.5 Transmitter and Receiver Module

Frequency shift keying transmitter and receiver modules are used to establish communication between the two PIC microcontrollers for both modules. The operation frequency for both modules operates from 300 to1000 MHz.

Figure 11 below indicates a transmitter operation flow. Firstly, the module is initializing. The transmitter module is opened when a transmitter is ready to transmit data. The module should wait for the nIRQ line to go low before sending data to the receiver. If sending data to the receiver is successful, then the module will start over to send another data. But if the package cannot be sent, the module will have to wait for the nIRQ pin to go low again before trying to send the data again.

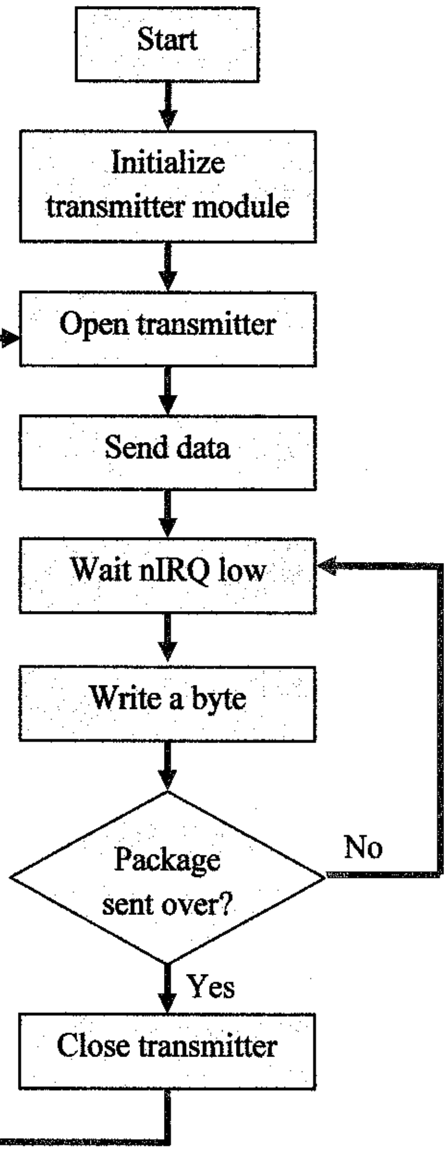


Figure 11 : Transmitter module operation flow

The operation of a transmitter module and receiver module is similar. Figure 12 below indicates a receiver operation flow. Firstly, the module is initializing. The receiver module is opened when the module is ready to receive data. The nIRQ line has to go low before the module receives data. If receiving data is successful, then the module will check pass through the data. If the correct package of data is received, the module will indicate receiving data before enabling the module to receive another data. But if the module does not receive any data, the module will have to wait for the nIRQ pin to go low again before trying to receive the same data again.

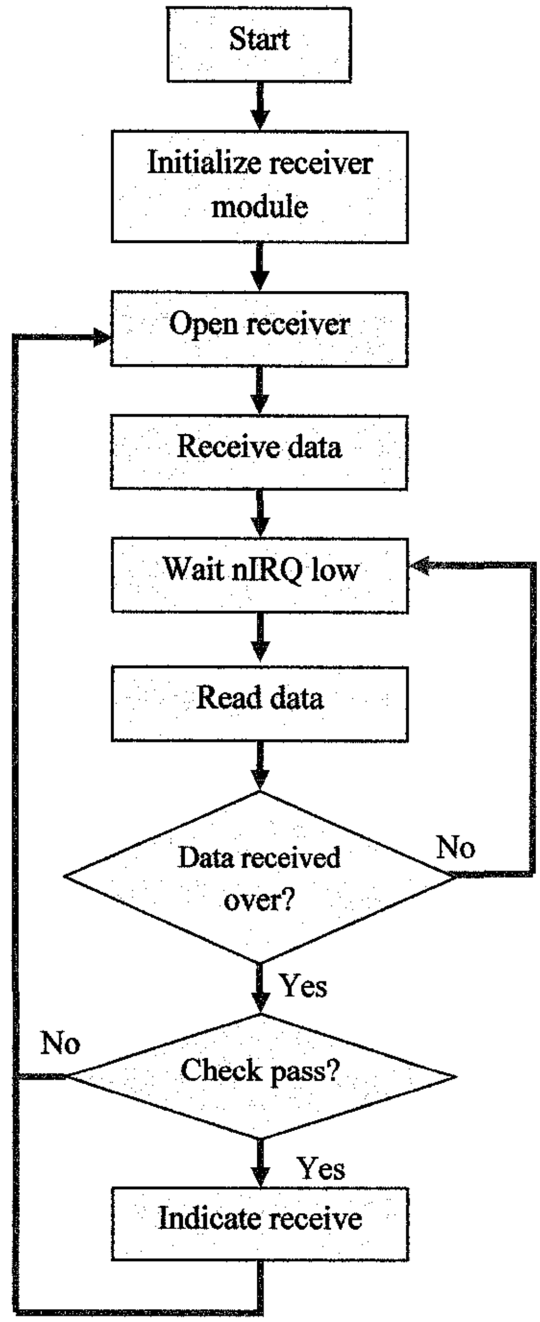


Figure 12 : Receiver module operation flow

The system would consist of two parts, the source (transmitter) module and the destination (receiver) module. The transmitter module would be placed at the doorknob. This module would consist of a capacitance sensor, an accelerometer and a transmitter; meanwhile, the receiver module would consist of a receiver and a PIR sensor.

When a person touches the doorknob, a signal will be sent from the capacitance sensor to the transmitter. The same happens when someone opens the door, where the accelerometer will send a signal to the transmitter. The transmitter will send the appropriate message from the transmitter to the receiver (destination module). The receiver will be programmed to handle signals from sensors, to trigger an alarm, and to interface with the computer to handle the general administrative actions.

2.6 C Compiler

A C compiler is a computer program whose only job is to convert the C program from our form to a form the computer can read and execute. A computer cannot understand the spoken or written language that we humans use in our day to day conversations; likewise, we cannot understand the binary language that the computer uses to do its tasks. It is therefore necessary for us to write instructions in some specially defined language, in this case C, which we can understand, then have that very precise language converted into the very terse language that the computer can understand. The computer prefers a string of 1's and 0's that mean very little to us, but can be very quickly and accurately understood by the computer. The original C program is called the "source code", and the resulting compiled code produced by the compiler is usually called an "object file".

The C Compiler is used to program the PIC microcontroller in the system because the C program used in the C Compiler is more user friendly and not lengthy in words in order to perform a task.

2.7 Summary

Basically this chapter gave the in depth idea and knowledge regarding the hardware and software used to build a portable security system. This chapter contains all the relevant theories, facts and data which are relevant to the objectives of the project. There are three types of input being used. A capacitance sensor is used to detect the touching of a door knob. An accelerometer is used to detect an open door; meanwhile, a Passive Infra Red (PIR) sensor is used to sense movements in a room. The main equipment used in the system is the three sensors, a transmitter module, a receiver module, a microcontroller unit and inpit output peripherals devices. In addition, the software used to program the microcontroller is C-Compiler.

The next chapter will emphasize on the methods and procedures used to achieve the objectives of the projects. Several steps have been identified as a guide on completing the project flow of the whole process. Also, a proposed work for prototype installation is included in the methodology section.

CHAPTER 3

METHODOLOGY

Through research and feasibility studies, the background information and equipment used to build a portable security system is analyzed. In this project, a procedure on project work flow and management needs to be followed. Several steps were identified as a guide on completing the project flow of the whole process. Identifying the project work flow is important to ensure successful outcome of the project. The methodology of designing and building a portable security system will be discussed in this chapter.

3.1 Procedure Identifications

Step 1. Define problem – define security breach needed to be implemented in the system.

Step 2. Research – research is conducted to generate alternatives and solutions to the problems discussed. All findings are summarized in the literature review section which will include the solution to detect the security breach.

Step 3. Identification of hardware required – the process includes the identification of components that meet the requirements for the project.

Step 4. Development of internal circuit – build the internal circuit for the prototype. It includes a transmitter module, a touch sensor circuit, the receiver module and the connection to the PIR sensor.

Step 5. Program design – programmed the transmitter and receiver module to communicate and transmit appropriate signals. Also, the microcontroller is programmed using the control program.

Step 6. Test and run the control program – the sequence of the operation is also checked.

Step 7. Build the prototype – include the necessary modifications on the hardware, the control equipment and the system.

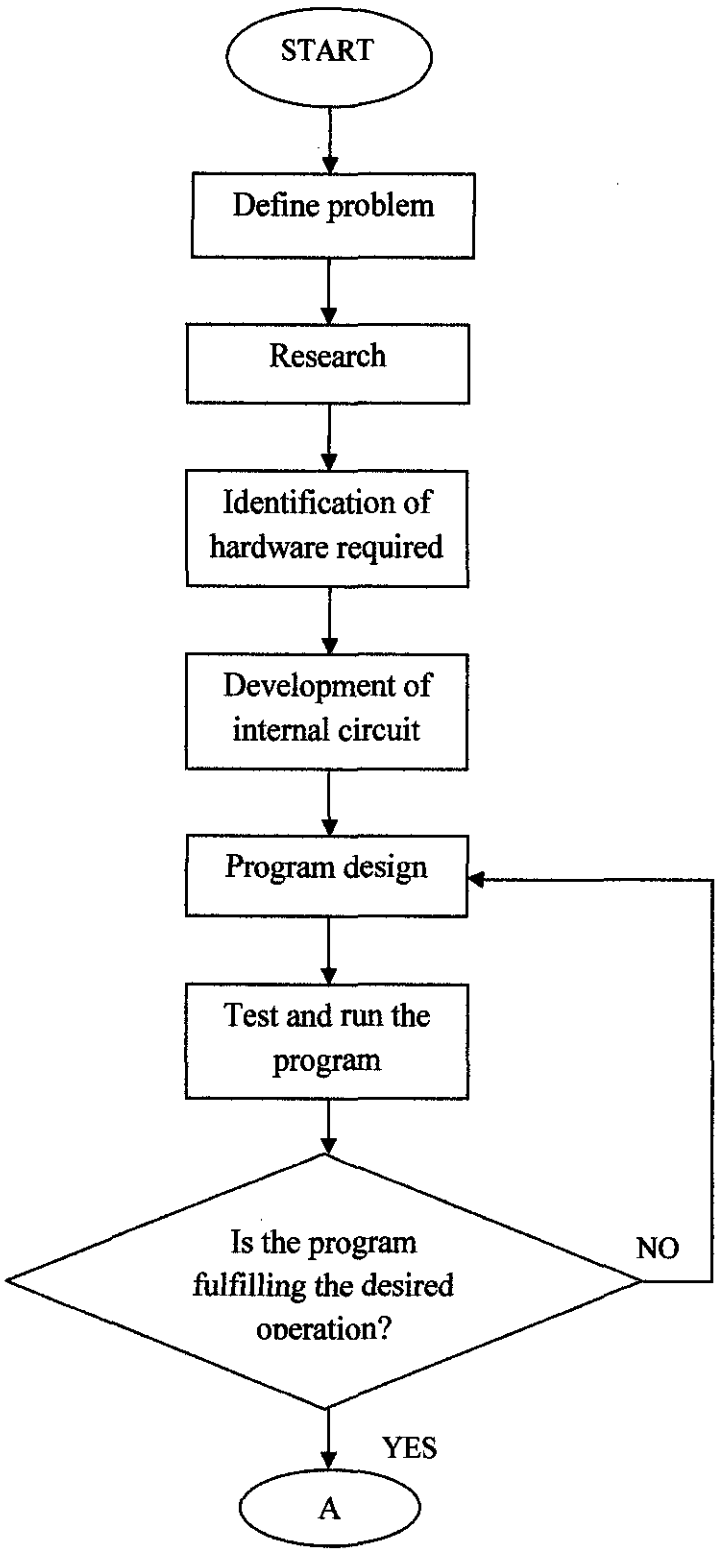
Step 8. Integration between the internal circuit and system – the integration between the internal circuit and system must be programmed to handle signals from sensors, to trigger an alarm, and to interface with the computer to handle the general administrative actions.

Step. 9 Test and run the security system - the system must meet the specification required to trigger an alarm when it detects the security breach stated in the problem.

Step 10. Modification – if necessary, a modification has to be made to achieve building a working prototype.

3.2 Procedure Flowchart

The methodology is classified into two parts that is the process done during FYP 1 period and FYP 2 period. Step 1 until Step 6, which involves research, identification of hardware, development of circuit and program design are implemented during the FYP 1 period. Meanwhile, during the FYP 2 period, the program will be tested and modifications will be made if necessary. A flowchart as shown in Figure 13 explains the procedure used in the development of the system.



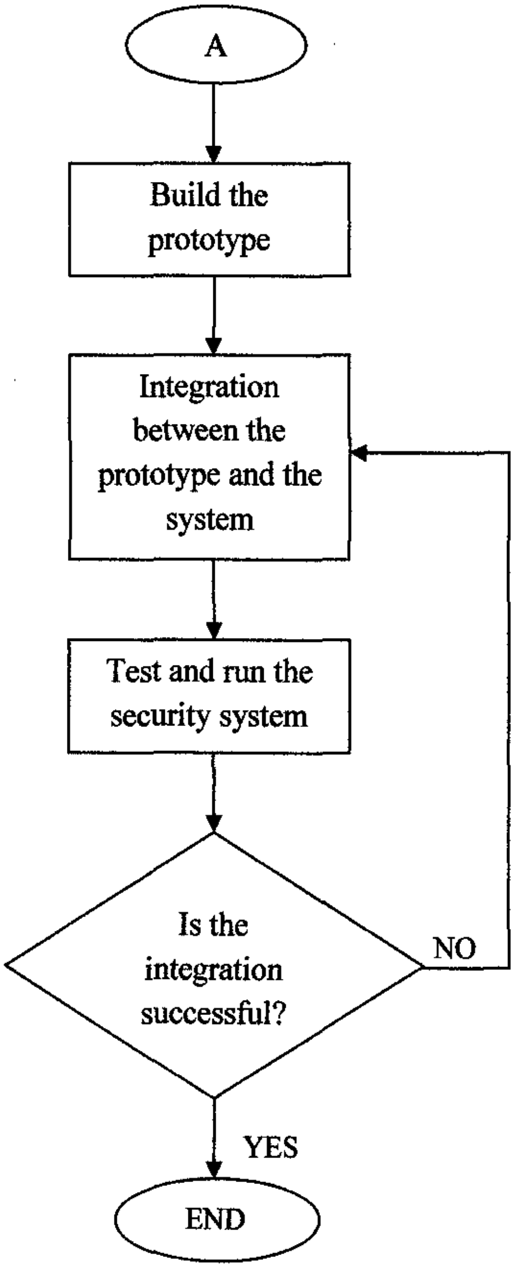


Figure 13 : Flowchart describing the procedure for designing and building the portable security system

3.3 Proposed Work for Prototype Installation

3.3.1 Transmitter Module

The connection for a transmitter module is illustrated in Figure 14. It consists of an accelerometer, a capacitance sensor, a transmitter module connected to various pins of a PIC Microcontroller (PIC16F877). The transmitter module will be placed on the door to detect the door knob being touched and on occasion of an opened door.

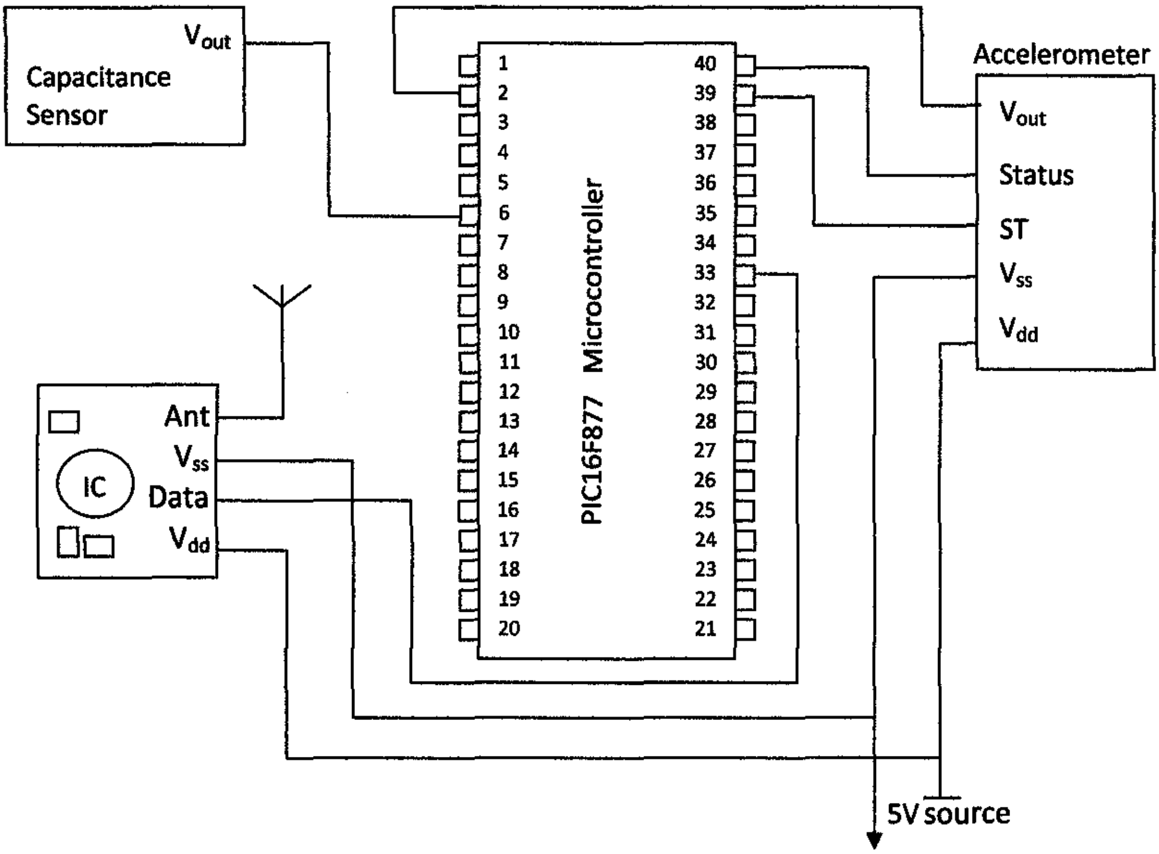


Figure 14 : Connection at transmitter module

In the case of this project, the capacitance sensor will amplify the resulting voltage due to the capacitance difference of our body and ground. Unfortunately, the capacitance sensor needed in the system to detect a touched door knob is too expensive for this project. So, a simple circuit is designed to replace the capacitance sensor. The circuit would have the basic operation of a capacitance sensor.

Figure 15 below shows a simple circuit being used to replace the capacitance sensor. The circuit used a Quadruple 2 input NAND gate IC, a loop hooked to a door knob and other electrical components. Using the properties of capacitors, touch can be used as an extremely effective method of input and control. What allows the human body to be sensed are the conductive electrolytes that it contains, which allows charge to be held and transferred.

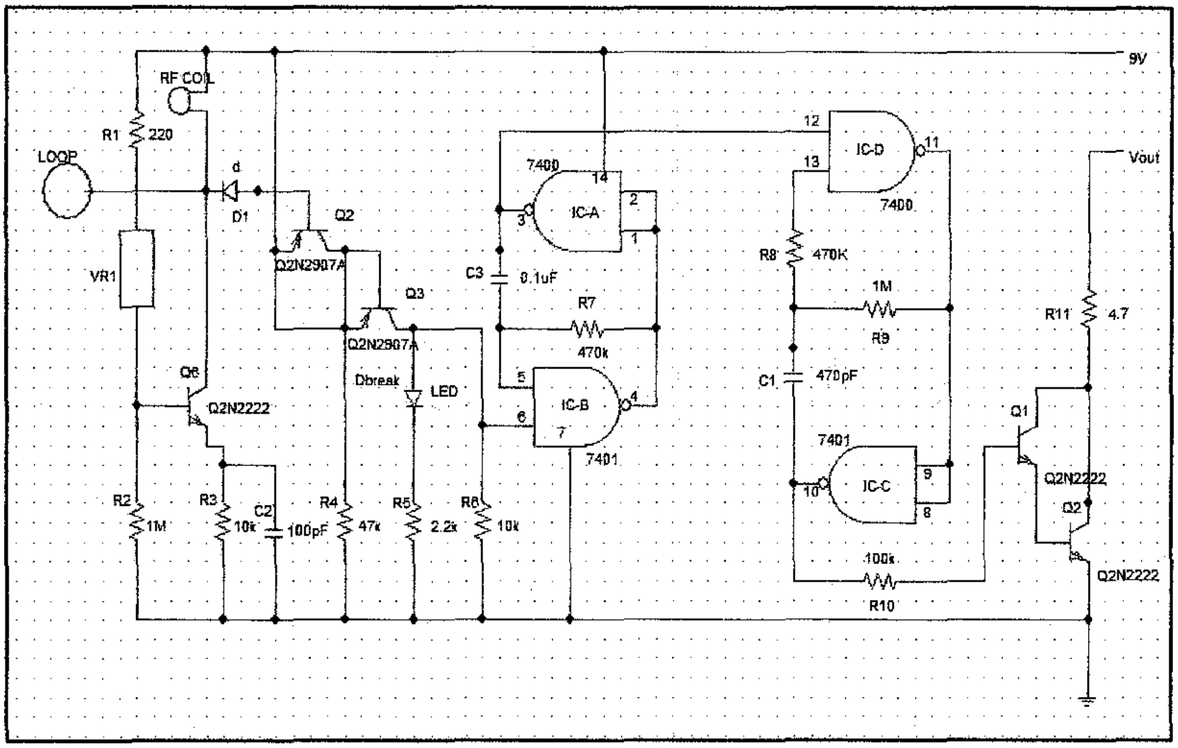


Figure 15 : Schematic diagram of circuit to replace capacitance sensor

Gates are some of the fundamental building blocks of digital electronics. This IC has four two input NAND gates in one package. The loop acts as the sensor input and is hooked to the doorknob. It is connected to the IC. Both inputs are grounded so that the capacitance difference can be sensed each time the wire is touched. This is performed in order to saturate the voltage levels to produce a digital high or digital low. The truth table for this type of circuit is given below (see Table 3). Inputs are taken as A and B. The state for each input is determined by user. If both or any one of the inputs has been touched (HIGH state), the output goes high. On the other hand, if both inputs has not been touched (LOW state), the output goes low.

Table 3 : Truth table for the simple touch sensor circuit

INPUT		**OUTPUT**
A	**B**	**Y**
L	L	L
L	H	H
H	L	H
H	H	H

Two NAND gates are used for touch sensing and other two gates are fed to pin A0 at the MCU as an output. Gates consisting of IC-A and IC-B act as a timer for 10 seconds which is activated by touching the touch probes that is the loop hooked to the doorknob. These can be conveniently concealed in a door handle. If anybody touches the input gate at IC-A and IC-B, it results in a high signal at its output. The other two gates comprising IC-C and IC-D make a high frequency oscillator.

3.3.2 Receiver Module

The connection for the receiver module is illustrated in Figure 16. It consists of a PIR sensor, a receiver (with its own antenna), a keypad and a buzzer connected to various pin of a PIC Microcontroller (PIC16F877). The receiver module will be placed in the room itself to detect movements in the room and to alert user on the intrusion.

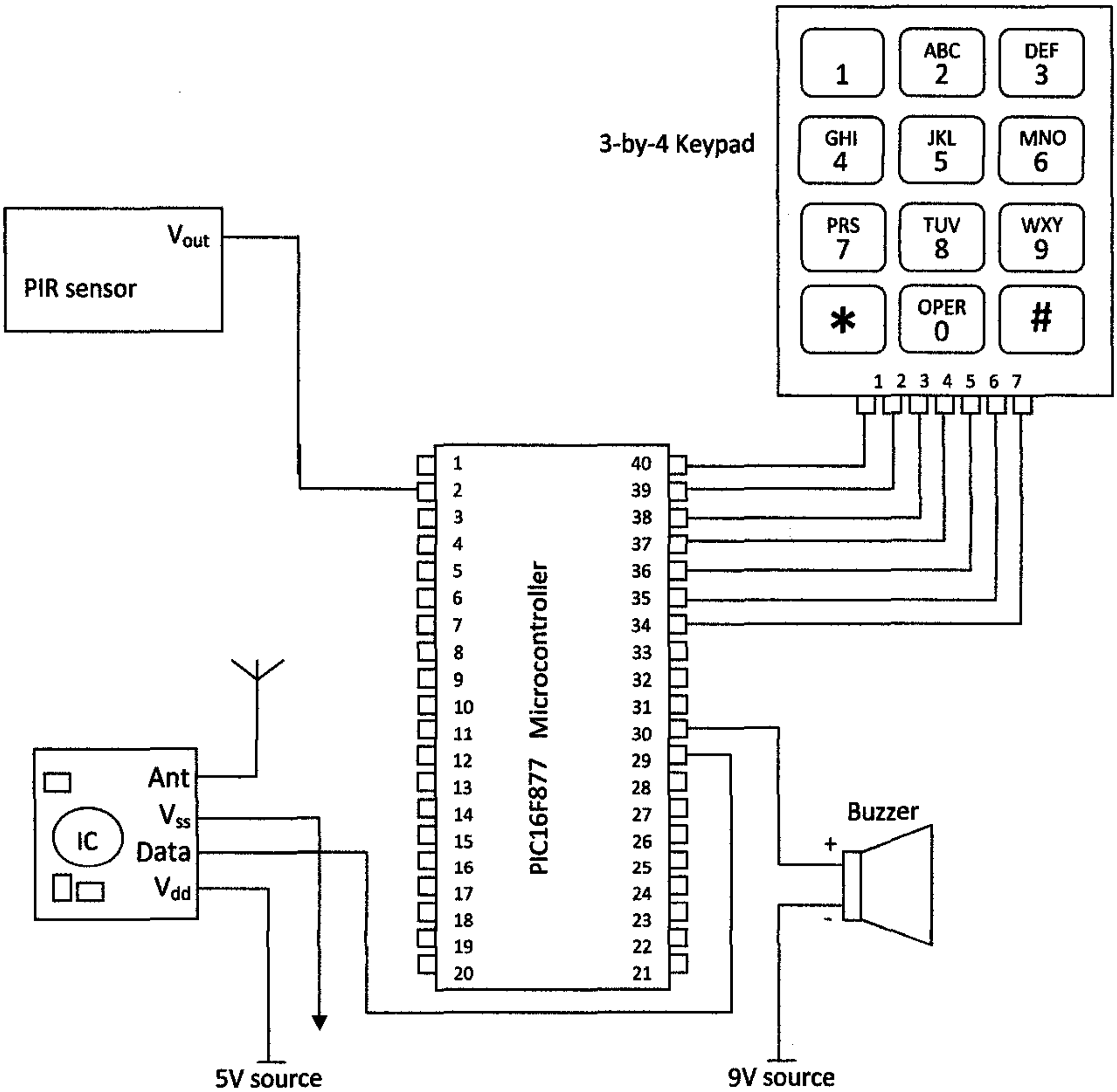


Figure 16: Connection at receiver module

The PIR sensor used is the sensor with a model number IRS-A200ST01-R1 from MURATA. Its working principle is that the higher the sensed temperature, the lower the output voltage from the PIR sensor. As illustrated in Figure 17, a few resistors and transistor are used to produce a 3V source to drive the PIR sensor. The output voltage from the PIR sensor is fed into pin A.0 of the PIC microcontroller.

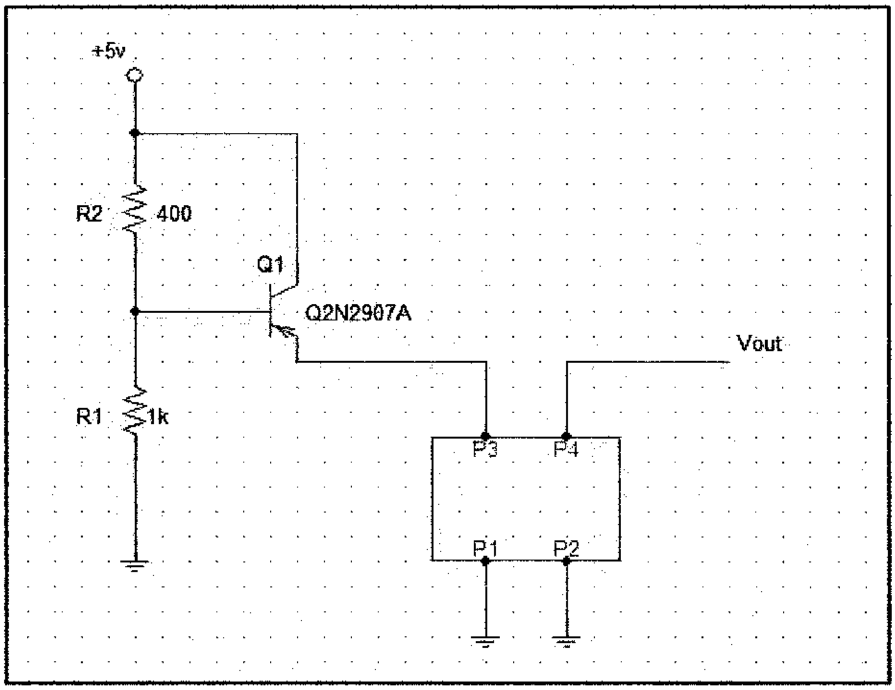


Figure 17 : Schematic diagram shows connection to PIR sensor

3.4 Tools and Equipment Required

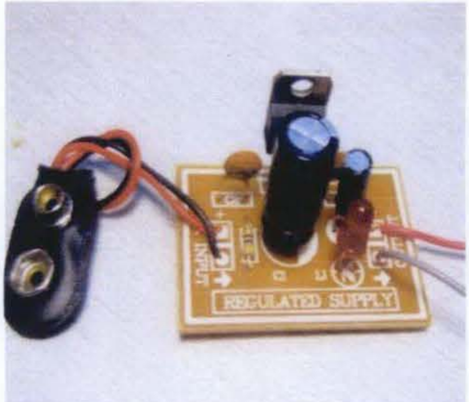
In this section the software and hardware being used to construct the system is being mentioned. The software and hardware used are the most commonly used applications in an education based simulation.

3.4.1 Software

C-Compiler (used to program the PIC16F877a microcontroller)

3.4.2 Hardware

a) <u>Capacitance Sensor</u> - Resistors (220Ω, 1MΩ, 10kΩ, 47kΩ, 2.2 kΩ, 470kΩ, 100kΩ) - IC 4011 (14 pins) - Voltage Regulator - Capacitors (470pF, 100pF, 0.1uF) - NPN and PNP Transistors - RF Coil	
b) <u>Accelerometer</u> Freescale Semiconductor MMA1260KEG	
c) <u>PIR Sensor</u> MURATA IRS-A200ST01- R1	

d)	<u>5V Regulated Power Supply</u> - Regulator IC (7805) - Resistor (470Ω) - Capacitors (1000uF, 47 uF, 0.1uF)	
e)	<u>PIC Microcontroller</u> PIC16F877	
f)	<u>Transmitter Module</u> ALPHA-TX433S	
g)	<u>Receiver Module</u> ALPHA-TX433S	

<table>
<tr><td>h) 3-by-4 Keypad</td><td>
</td></tr>
<tr><td>i) Buzzer</td><td>
</td></tr>
</table>

3.5 Summary

In this chapter, the methods and procedures used to achieve the objectives of the projects have been discussed. Several steps were identified as a guide on completing the project flow of the whole process. Also, a proposed work for prototype installation is included in this section. The prototypes needed to be built are the transmitter module, the touch sensor circuit, the receiver module and the connection to the PIR sensor. Also, the hardware and software required have been identified in this section. Next, the finding and outcome of the project will be presented in the next chapter.

CHAPTER 4

RESULT AND DISCUSSION

The results and findings are described further in this section. It includes the findings, observations and results of the project. All the gathered data from the project work is presented in the form of tables and figures. This section includes the findings, observations and results of the project.

4.1 Transmitter Module

As proposed, the transmitter module is built. Figure 18 shows the transmitter module where it consists of an accelerometer, a capacitance sensor, and a transmitter module connected to the various pin of a PIC Microcontroller (PIC16F877). All the inputs of the module are connected to the PIC Microcontroller. The transmitter module has been placed on the door to detect the touched on the door knob and on occasion the opening of the door.

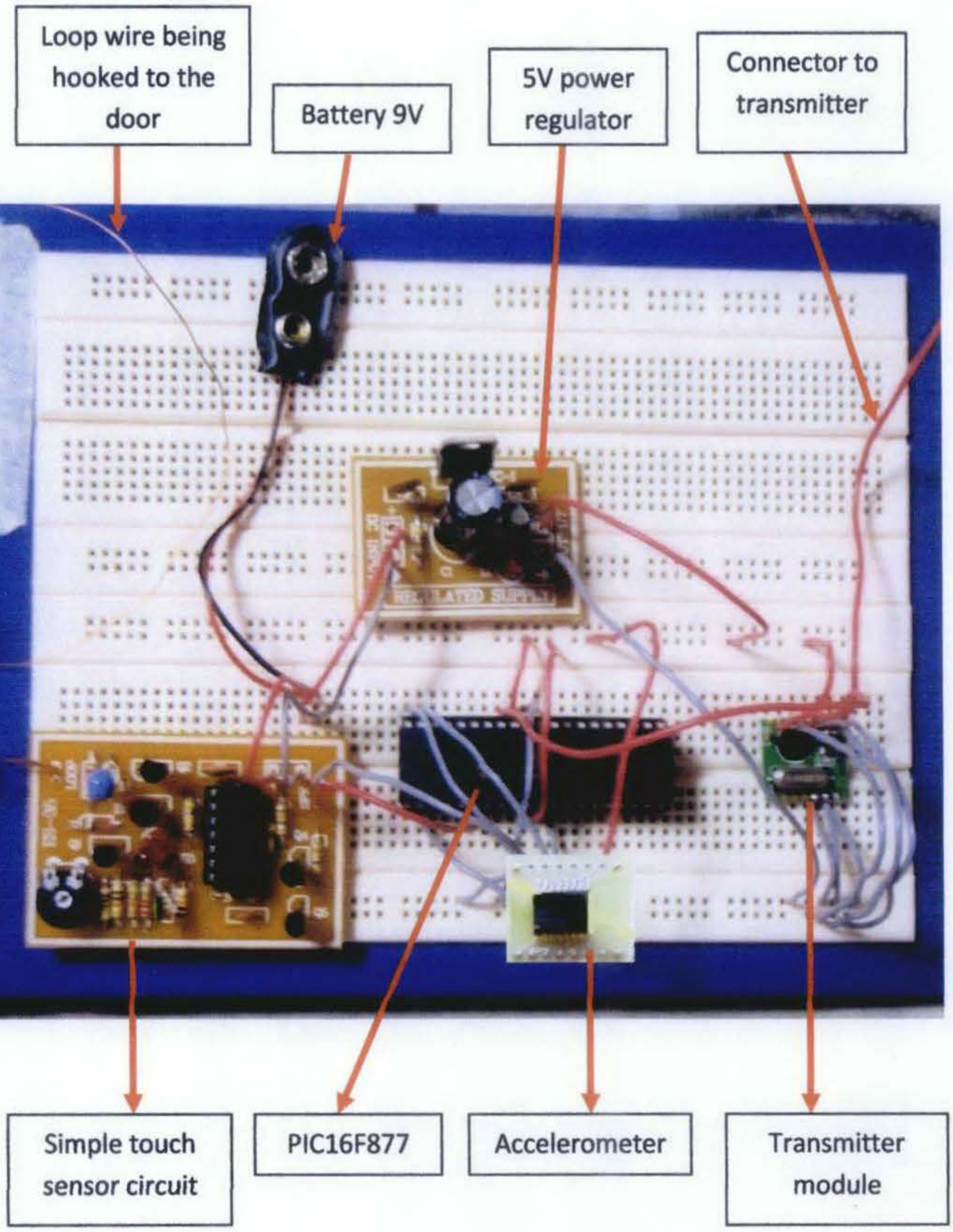


Figure 18: Developed transmitter module

The loop of the simple touch sensor circuit is hanged on the door knob to detect the door knob being touched. The transmitter module will be placed on the door itself. The accelerometer on the transmitter module will sense the opening of a door by detecting the distance of a moved door on z-axis sensitivity. The installation of the transmitter module on the door is illustrated in Figure 19 below:

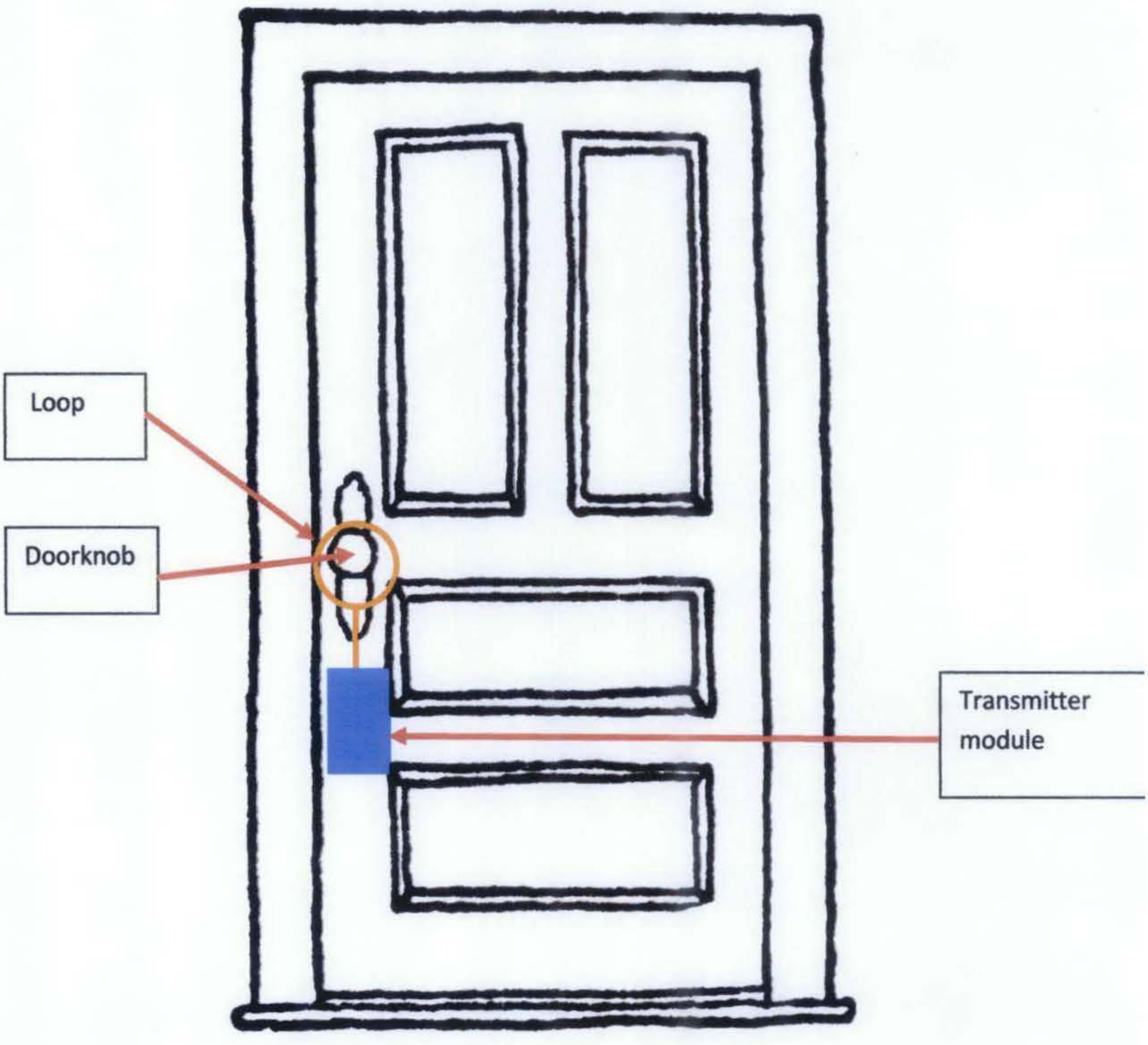


Figure 19 : Installation of a transmitter module on the door

4.1.1 Capacitance Sensor

In this project, a simple circuit is designed to replace the capacitance sensor. The circuit would have the basic operation of a capacitance sensor where the capacitance difference can be sensed each time one touch the wire. The circuit used a Quadruple 2 input NAND gate IC, a loop hooked to a door knob and other electrical components.

The truth table for this type of gate is given in the previous chapter (see Table 3). The inputs are taken as A and B. If both or any one of the inputs goes low, the output goes high. On the other hand, if both inputs go high, the output goes low. If anyone touches the doorknob, it results in a high signal at its output.

Based on the schematic diagram proposed, the simple touch sensor circuit is constructed (see Figure 20). Each electronics component needed to build the circuit is soldered on a laminated circuit board. Wires are used to connect the touch sensor circuit to the output and power supply. Meanwhile, bare wires are used to connect the input terminal of the circuit to the doorknob.

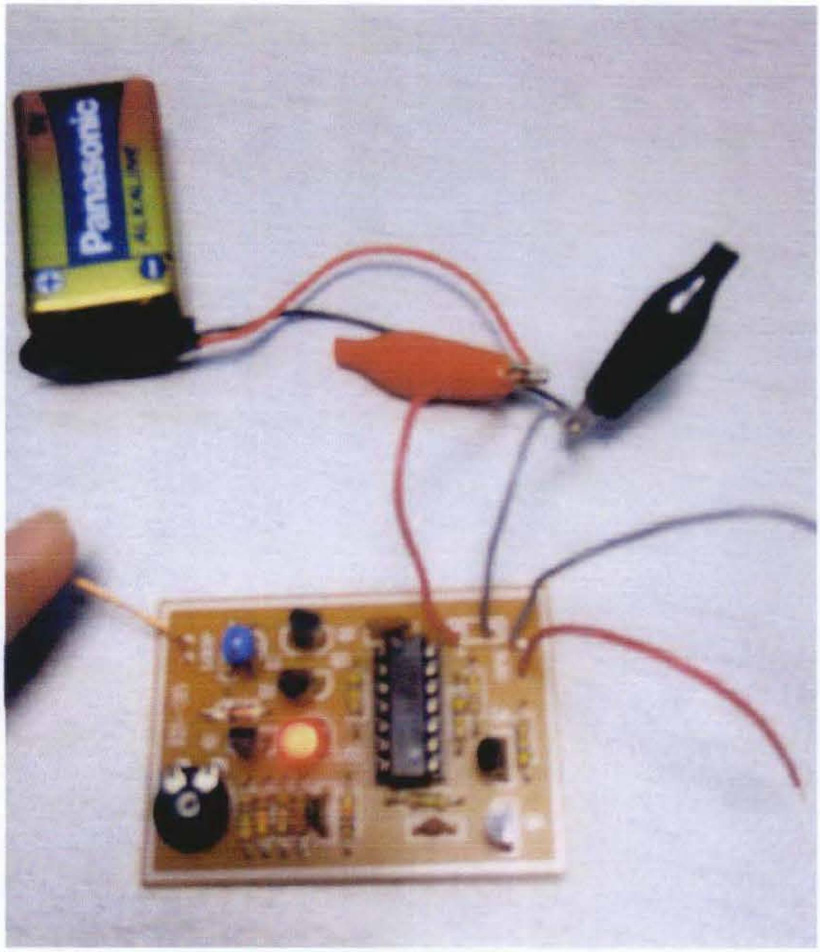


Figure 20 : The touch sensor circuit

The simple touch sensor circuit behavioral has been analyzed and observed. When the doorknob is touched, the high voltage is sensed at Vout; meanwhile, a low voltage is sensed at Vout when the doorknob is not touched. A voltage reading is made at each input and output of the simple touch sensor circuit. The voltage reading for each condition is recorded in Table 4 below.

Table 4 : Voltage reading while analyzing the behavior of a simple touch sensor circuit

Voltage reading at input (V)		**Voltage reading at output (V)**
A	**B**	**Y**
1.50	1.52	0.07
1.50	8.80	4.01
8.82	1.51	4.03
8.80	8.82	4.01

The reading matches the theoretical truth table condition of a simple touch sensor circuit. If both or any one of the inputs has been touched (HIGH state), the output goes high. On the other hand, if both inputs have not been touched (LOW state), the output goes low. A high voltage at Vout will light on the LED and a signal (ON state) will be sent to the microcontroller. At this condition, the alarm will be turned on.

4.1.2 Accelerometer

The MMA1260KEG accelerometer by Freescale Semiconductor is used to detect an opened door. Because the size of an accelerometer is too small and due to its high sensitivity to temperature, the accelerometer is being laminated to a printed board before wires are soldered to connect the accelerometer to the PIC microcontroller; the laminated accelerometer is shown in Figure 21 below.

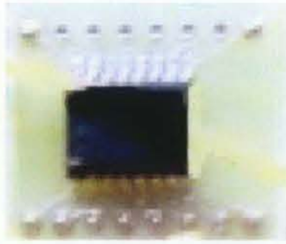


Figure 21 : The accelerometer laminated on a printed board.

The accelerometer on the transmitter module is used to sense an opening of a door by detecting the distance of the moved door on z-axis sensitivity. The accelerometer behavioral has been analyzed and observed. When an open door is sensed, a high voltage is sensed at Vout; meanwhile, a low voltage is sensed at Vout when the door is closed. The output voltage produced is linearly related to the distance of a moved door. A voltage reading is made at Vout when the door is opened and when the door is closed. The voltage reading for each condition is recorded below:

Vout when the door is closed = 0.00V
Vout when the door is half opened = 0.17V
Vout when the door is widely opened = 0.47V

4.2 Receiver Module

The receiver module will be placed in the room itself to detect movements in the room and to alert user on the intrusion. User must be aware that the range of communication of a transmitter module and receiver module should be less than 300m.

The receiver module is developed as proposed. Figure 22 shows the transmitter module where it consists of a PIR sensor, a receiver (with its own antenna), a keypad and a buzzer connected to the various pin of a PIC microcontroller (PIC16F877). All inputs and outputs of the module are connected to the PIC microcontroller. The receiver module will be placed in the room itself to detect movements in the room and to alert the user on the intrusion.

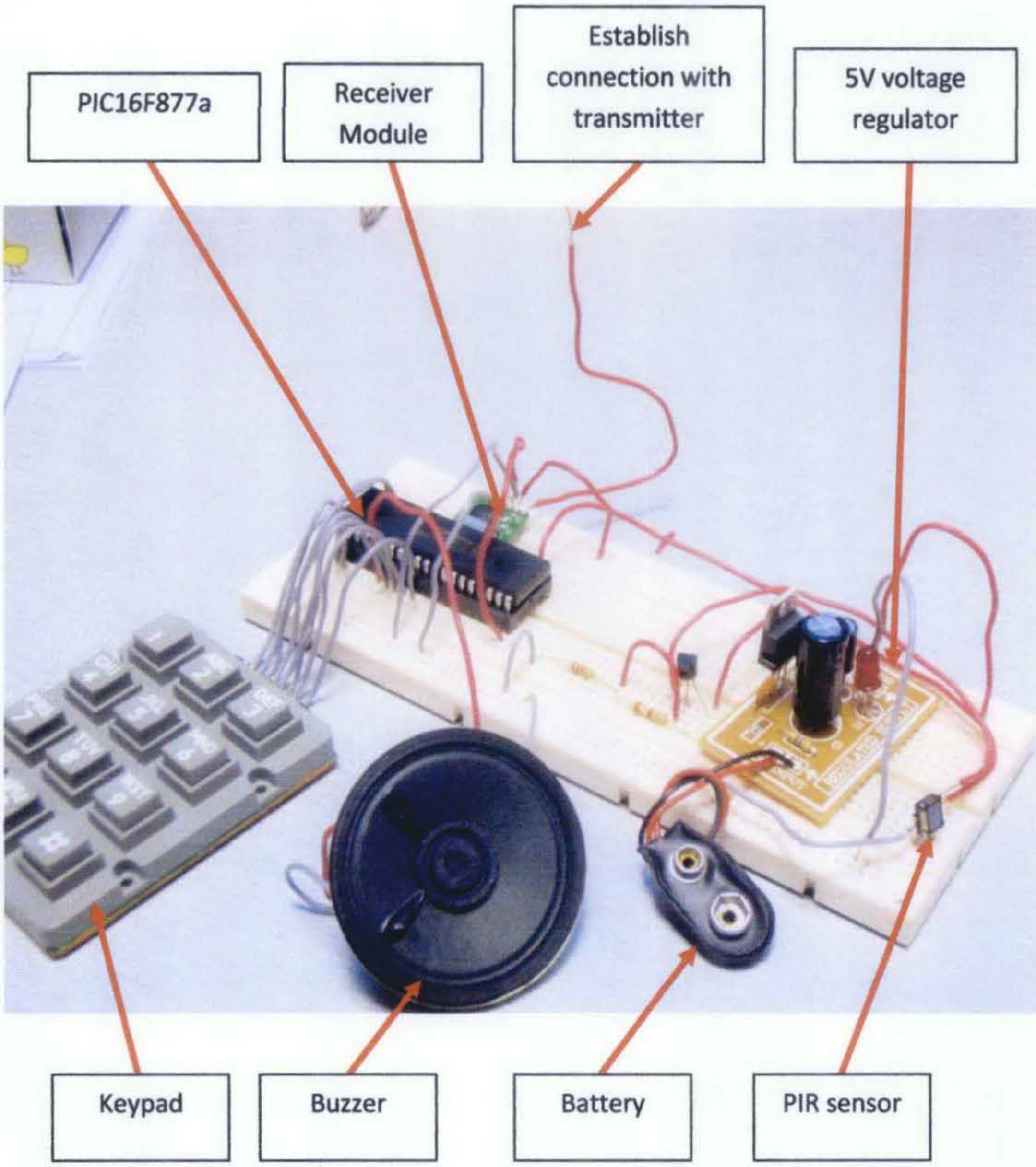


Figure 22 : Developed receiver module

4.2.1 Problem Analysis

The objective to sense movements in a room is not achieved. Although the PIR sensor working principle has been fully understood, there is a hard time to figure out on how the actual device works. The relation between the output voltage and the surrounding temperature sensed by the PIR is still cannot be established as the detail operating characteristics of the PIR sensor is still unknown.

Also, the transmitter module and receiver module had failed to communicate wirelessly. There are errors in the programming code of both modules. There are modifications which need to be done in the programming parts to achieve communications between the transmitter module and the receiver module wirelessly.

4.3 Summary

The finding and outcome of the project is presented in this chapter. All the gathered data from the project work are presented in the form of tables and figures. This section includes the findings, observations and results of the project. Mainly, the circuit needed to build the prototype for the portable security system is built. The built circuits are the transmitter module, the touch sensor circuit, the receiver module and the connection to the PIR sensor.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

The development of a portable security system has been discussed in the previous section. The main focus of this project is to construct a reliable and portable security system that is small in size, has a quick and simple setup, can be moved easily and can be accessed wirelessly in a range of distance. Through the development process of the system, an opportunity to determine the appropriate sensors, programming the microcontroller and interfacing the inputs and outputs of the system is provided. Apart from that, an additional knowledge on software application is gained.

5.1 Conclusion

This portable security system had provides a good learning and practical experience in working with hardware and software. Also, it provides us with the opportunity to interface parts of the communication mechanism that is used between the transmitter and receiver module, as well as to work with protocols. The construction of the prototype starts from the chassis design and implementation, the identification of hardware and software used, the development of circuits, and the program design and interfacing inputs and outputs of the system.

Upon the completion of the project, the transmitter module and also the receiver module had been developed. The transmitter module has a capacitance sensor to detect a door knob touching and an accelerometer to sense the opening of door, a transmitter module and a PIC microcontroller. The receiver module consists of a PIR sensor to sense movements in a room, PIC microcontroller and the outputs of the system. The circuits built for the project are the transmitter module, the touch sensor circuit, the receiver module and the connection to the PIR sensor.

The major objectives of this project have been successfully achieved. The main achievements of this work are as follows:

- The familiarization to the microcontroller technology and C programming
- The construction of a portable security system that enables the detection of three types of security breach that is to detect a touch on a door knob, to sense door opening, and to sense movements in a room.
- The interfacing of all the main components of the project.
- The handling of important tasks in proposing a project up to the completion process.

This portable security system provides an option for those who rent in an apartment to install a security system in their apartments without causing damages to the apartment's wall structure. Also, it helps to ensure safety and confidential usage of a room. The system also benefits travelers who might use the portable security system in his or her hotel for safety purposes, as the portable security system is small in size, easy to setup and most importantly mobile.

5.2 Recommendation

This section presents recommendations for future improvements of this project. Although this project has achieved its objective which includes constructing the transmitter module and the receiver module, designing programs for the microcontrollers to achieve the aim and objectives of the project, it still has a long way to go.

In this project, we have not yet tested and run the system accordingly as the program design part has not been fully completed. Some of the future work that we plan to undertake is to connect the system to the computer via USB or serial port to enable users to retrieve the log information of the intrusion. Also, we plan to add an LCD display as it provides a clear screen function for user.

For future work, the system should be able to be connected to a computer using serial ports or USB ports. The computer should be used to set or change this 4 digit code, as well as to set the time and date. By interfacing the systems to the computer, the user should be able to retrieve the track of the time and date when each of the security system components has been triggered.

Also, an LCD display should be included in the future work as it provides clear screen function for user. An LCD display would have made this portable security system to be more user-friendly in the future.

REFERENCES

[1] "Security. 2010", http://www.businessdictionary.com/definition/ security.html, [retrieved 17 Feb 2010, 10.13 am].

[2] Lion Precision 2009. *Capacitive Sensor Operation and Optimization* Tech Note LT03-0020 91-5, Cornell University, Minessota, USA.

[3] " Accelerometer. 2010 " , http://www.answers.com/topic/accelerometer, [retrieved 17 Feb 2010, 10.45 am].

[4] Freescale Semiconductor 2009. *Low G Micromachined Accelerometer* Technical Data MMA1260KEG, Freescale Semiconductor, Inc., Arizona. USA.

[5] "Passive Infrared Sensor. 2010", http://en.wikipedia.org/wiki/Passive_infrared_sensor, [retrieved 17 Feb 2010, 11.25 am].

[6] Feisal Mohammed. 2002. *EE25M Introduction to microprocessors* http://www.electronicsforu.com/electronicsforu/Articles/ad.asp?url=http:// www.eng.uwi.tt/depts/elec/staff/feisal/ee25m/resources/ee25mlect2.pdf&title=PIC+16F877+ Microcontroller+for+beginners.

[7] Microchip Technology Inc. 1999. *28/40-pin 8-Bit CMOS FLASH Microcontrollers* Technical Data PIC16F87X, Microchip Technology Inc., Arizona. USA.

APPENDICES

APPENDIX A

Gantt Chart for FYP I Semester Jan 2010

No	Detail/ Week	1	2	3	4	5	6	7		8	9	10	11	12	13	14
1	Selection of Project Topic								Mid-semester break							
2	Preliminary Research Work															
3	Submission of Preliminary Report															
4	Project Work; Research on Hardware Design															
5	Submission of Progress Report															
6	Seminar															
7	Project Work Continue; Designing System															
8	Submission of Interim Report Final Draft															
9	Oral Presentation															

Suggested milestones

Process

APPENDIX B

Gantt Chart for FYP II Semester July 2010

No	Detail/ Week	1	2	3	4	5	6	7		8	9	10	11	12	13	14
1	Project Work Continue								Mid-semester break							
2	Submission of Progress Report 1															
3	Project Work Continue															
4	Submission of Progress Report 2															
5	Project Work Continue															
6	Poster Exhibition															
7	Submission of Dissertation (soft bound)															
8	Oral Presentation															
9	Submission of Project Dissertation (hard bound)															

Suggested milestones

Process

APPENDIX C

Source Code of Transmitter Module

```
#include<math.h>
#include<16F877.h>
#include<stdio.h> // sprintf
#include<delay.h>

#define begin {
#define end }
#define preTx 0xaa
#define dataHeader 0x60
#define delta 0.02

unsigned long int TxBuff @0x2f0;
unsigned char TxLL @0x2f0, TxLH @0x2f1, TxHL @0x2f2, TxHH @0x2f3;
unsigned int convH, count;
unsigned char convL, preTxCt, touched, moved, sent, touchDev,
   shiftTx, freqM;
float sensorVal[2], moveInit;
bit touchS, state, Tx;


void init(void);

//400uSec interrupt for transmit
interrupt [TIM2_COMP] void trans(void)
begin
   state=1;
   if(!PINA.4) freqM++;
   else freqM=0;
   if(Tx)
      {
         PORTB.0=(TxHH&0x80)>>7;
          shiftTx++;
         TxBuff=TxBuff<<1;
      }
   else PORTB.0=0;
end

interrupt [ADC_INT] void converter(void)

begin
   state=0;
end

void main(void)
{
init();

   while(1)
   {
   if(state)
```

```
{
   state=0;
   convL=ADCL;
   convH=ADCH;
   //start another conversion

   convH=convH<<8;
   convH+=convL;
   sensorVal[touchS]=((((float)convH)/1023.0)*5.);

   if(touchS)
   {
      touchS=0;
      if(freqM>5)
         touched=0b00001000;
      else
         touched=0b00000100;

   }
   else
   {
      touchS=1;
      //start a conversion
      if(((sensorVal[0]<(moveInit+delta)) ||
(sensorVal[0]<(moveInit-delta))))
         && ((sensorVal[1]<(moveInit+delta)) ||
(sensorVal[1]<(moveInit-delta))))
         moved=0b00000010;
      else
         moved=0b00000001;
   }


   //this part is for the accelerometer STATUS and ST pins. If
STATUS is high, it can be turned low by applying a logic high
to Self Test(ST)
   if(PINB.7) PORTB.6=1;
   else PORTB.6=0;

   //TX code
   if(Tx)
   {
      //PORTB.0=(TxHH&0x80)>>7;
      if(touched>7 || moved>1)    //if touching or opening is
detected
      {
           if(shiftTx<=28)
           {
              TxBuff&=TxBuff|=((long int)touched)<<shiftTx;
              touched=0b00000100;
              TxBuff|=((long int)moved)<<shiftTx;
              moved=0b00000001;
           }
           else
           {
              shiftTx-=8;
```

```
                TxBuff&=TxBuff|=((long int)touched)<<shiftTx;
                touched=0b00000100;
                TxBuff|=((long int)moved)<<shiftTx;
                moved=0b00000001;
            }

         }
         //shiftTx++;
        //TxBuff=TxBuff<<1;
        if(!TxBuff)
        {
           Tx=0;
           shiftTx=0;
        }

    }
    else
    {
        //transmit 24 bits of 1010's and then the data header
  0110 and then 4 bits of information about doorknob touching
  and door opening.
        //bits in transmitter buffer are transmitted in
  following order TxHH, TxHL, TxLH, TxLL.
        if(touched>7 || moved>1)   //if touching or opening is
  detected
        {
            TxLL=dataHeader;
            TxLL|=touched;
            touched=0b00000100;
            TxLL|=moved;
            moved=0b00000001;
            TxLH=preTx;
            TxHL=preTx;
            TxHH=preTx;
            Tx=1;
         }
     }


  }//if(state)

  }//while(1)
}//main


void init(void)//initialise the system
begin
  DDRB=0x7f;//pin 0 for TX, pin 7 accel stat, pin 6 accel cont
  PORTB=0x80;

  TCCR2=0x0c;//prescale timer 2 by 1/64 and clear on compare
  (for 2500 baud)
  OCR2=50;//compare on 50, 400uSec tick


  //init veriables
```

```
state=1;
touchS=0;
touched=0b00000100;
moved=0b00000001;
TxBuff=0;
convH=0;
convL=0;
preTxCt=3;
sensorVal=0.0;
count=250;
Tx=0;
sent=0;
shiftTx=0;
freqM=0;

delay_ms(1000);


//to obtain the initial voltage at start-up for purpose of
detecting door opening by comparing a later voltage.
while(state){}
state=1;
convL=ADCL;
convH=ADCH;

//start a conversion
convH=convH<<8;
convH+=convL;
moveInit=((((float)convH)/1023.0)*5.);

while(state){}

convL=ADCL;
convH=ADCH;
convH=convH<<8;
convH+=convL;
moveInit=(moveInit+((((float)convH)/1023.0)*5.))/2.

end
```

APPENDIX D

Source Code of Receiver Module

```
#include<math.h>
#include<16F877.h>
#include<stdio.h> // sprintf
#include<delay.h>
#include<String.h>

#define begin {
#define end }
#define maxKey 11
#define minut 60000
#define keypdInt 30
#define Release 1
#define Debounce 2
#define DetectTerm 3
#define stillPressed 4
#define debounceRelease 5
#define res 1500//1.5 Sec
#define ulkT 3000//3 sec unlock time
#define preTx 0xaa
#define dataHeader 0x60
#define delta 0.015

//the  subroutines
void gets_int(void);   //starts getting a string from serial line
void puts_int(void);   //starts a send to serial line
void init(void);    //all the usual mcu stuff
void keypad(void);
void systemInit(void);
void systemStop(void);
void keypadDeb(void);
void RxAnalisys(void);
void readLog(void);
void timeFunc(void);

//RXC ISR variables
unsigned char r_buffer[16]@0x2f0;    //input string
unsigned char func[4]@0x2f0, data[12]@0x2f4;
unsigned char r_index;  //current string index
unsigned char r_ready;  //flag for receive done
unsigned char r_char;  //current character

//TX empth ISR variables
unsigned char t_index;  //current string index
unsigned char t_buffer[64];   //output string
```

```
unsigned char t_ready;  //flag for transmit done
unsigned char t_char;  //current character

flash unsigned char keyTbl[16]={0xd7, 0xee, 0xde, 0xbe, 0xed,
0xdd, 0xbd, 0xeb, 0xdb, 0xbb, 0xb7};//0-9,#
//help/instruction menu
unsigned char keypd[5];//, tmpLog[64];//, keypd[5];
unsigned char tmpLog[16];
unsigned char ii, jj, ready, errorCnt, currState, keyVal, butnum,
convL, keypdIndex, month, maybe, act, funcCode, timeShift;
unsigned int temp, time, days, hours, minuts, restart, code,
convH;
unsigned long keypdTime;
unsigned char RxBuff, count, RxCount, state;
flash unsigned char castM=0b00000011, castT=0b00001100;
bit Rx, RxBit, touched, moved, motion, RxTemp, RxTempO,
touchBlock, moveBlock, motionBlock;
float sensorVal, motionInit;



void main(void)
{
   init();//initialize system
   while(1)
   {
      if(streamA || saveA)//alarm activated
      {
         if(state)//rf char recieved
         {
            state=0;
            if(ADCSRA.0)//wait for adc
            {
               convL=ADCL;
               convH=ADCH;
               //start another conversion
               convH=convH<<8;
               convH+=convL;
               sensorVal=((((float)convH)/1023.0)*5.);
            }
            RxBuff=RxBuff<<1;
            RxBuff|=((unsigned char)RxBit);
            if((RxBuff>>4)==0b00000110)//check rf in buffer for
data stamp
            {
               if((RxBuff & castT)>7) touched=1;
               else touched=0;
               if((RxBuff & castM)>1) moved=1;
               else moved=0;
            }
         }
         else
         {
            touched=0;
            moved=0;
         }
```

```
      if(restart==0)//if user paused for 1.5 sec restart keypad
entry from scratch
         keypdIndex=0;

      if(r_ready)//Rx complete
         RxAnalisys();


      if(keypdTime==0)//keypad state machine
         keypadDeb();


         else if(ready)//string recieved, and analised
         {
            ready=0;
            switch(funcCode)
            {
            case 1://"kpd"://keypad function
               t_index=0;
               keypdIndex=0;//reset keypad
               {
                  if(code==codeTbl)
                  {
                     errorCnt=0;//reset keypad errors
                     //disable alarm
                     streamA=0;
                     saveA=0;
                     systemStop();
                  }
               }

             case 2://"log"://log y/n[touch] y/n[open]
y/n[motion]
                 //touch
                 if(data[0]=='y') enableL[0]=1;
                 else if(data[0]=='n') enableL[0]=0;

                 //move
                 if(data[1]==' ')
                 {
                    if(data[2]=='y') enableL[1]=1;
                    else if(data[2]=='n') enableL[1]=0;
                 }

                 //motion
                 if(data[3]==' ')
                 {
                    if(data[4]=='y') enableL[2]=1;
                    else if(data[4]=='n') enableL[2]=0;
                 }

                 puts_int();//send confirmation to terminal
               break;
```

```
            case 3://"era"://era (clear screen)
                  putchar(0x0c);//clear screen - form feed
               for(ii=0;ii<7;ii++)//display menu on terminal
               {
                  puts_int();
                  while(!t_ready){};
               }
               putchar('\r');
               break;
               default://instruction error
               puts_int();//enable Tx
               while(!t_ready){};//wait untill Tx complete

               for(ii=0;ii<7;ii++)//display menu on terminal
               {
                  puts_int();
                  while(!t_ready){};
               }
               putchar('\r');
               break;
           }//end switch
           while(!t_ready){};//wait untill Tx complete
           PORTD.6=1;
        }

     end   //end else
   }


void init(void)
{
   DDRD=0x7f;//pin 7 Rx
   PORTD=0x80;

   DDRB=0xff;//keypad
   PORTB=0xff;

   delay_ms(1000);

   r_index=0;  //current string index
   for(ii=0;ii<64;ii++)//init all strings \ char buffers to 0
   {
      if(ii<16) r_buffer[ii]=0;
      t_buffer[ii]=0;
      if(ii<5) keypd[ii]=0;
   }
   r_ready=0;  //flag for receive done
   r_char=0;   //recieved char
   t_index=0;  //current string index
   t_ready=0;  //flag for transmit done
   t_char=0;   //char to transmit
   //generic for veriables
   ii=0;
   jj=0;

   ready=0;
```

```
   //keypad operation veriables
   errorCnt=0;
   currState=Release;//keypad debounce state
   keyVal=0xff;
   butnum=maxKey;
   keypdIndex=0;//# of keypad keys entered
   keypdTime=0;//interval for keypad debounce state machine
   maybe=0;
   temp=0;
   //time veriables
   days=0;
   hours=0;
   minuts=0;
   restart=0;

   //alarm system veriables
   code=0;
   act=0;
   funcCode=0;
   state=0;
   RxBuff=0;
   Rx=0;
   RxBit=0;
   touched=0;
   moved=0;
   count=0;
   RxTemp=0;
   RxTemp0=0;
   motion=0;
   touchBlock=0;
   moveBlock=0;
   motionBlock=0;


//serial Rx - not locking
void gets_int(void)
{
  r_ready=0;
  r_index=0;
  UCSRB.7=1;//enable Rx complete interupt
}

//serial Tx - not locking
void puts_int(void)
{
  t_ready=0;
  t_index=0;
  if (t_buffer[0]>0)//if t_buffer not empty
  {
     putchar(t_buffer[0]);//send first char
     UCSRB.5=1;//enable Data register empty interupt
  }
}

//keypad scan function
void keypad(void)
```

```
begin
   //scan top nibble
   DDRB=0x0f;
   PORTB=0xf0;
   delay_us(5);
   keyVal=PINB;
   //scan bottom nibble
   DDRB=0xf0;
   PORTB=0x0f;
   delay_us(5);
   keyVal = keyVal | PINB;
   //detect valid value
   if(keyVal!=0xff)
   begin
      for(butnum=0;butnum<maxKey;butnum++)
      begin
         if(keyTbl[butnum]==keyVal) break;
      end
   end
   else
      butnum=maxKey;
end

void systemInit(void)
{
   //30 secs to leave room
   DDRA.0=0;
   for(ii=30;ii;ii--)
   {
      delay_ms(1000);
      PORTC.7^=1;
      sprintf(t_buffer,"%d\r",ii);
      puts_int();
   }
   TCCR1B=0x0b;//prescale by 1/64 and clear on compare
   OCR1AH=0x00;
   OCR1AL=25;//compare on 25 100uSec
   ADMUX=0b01000000;//PIR
   ADCSRA=0b11000110;
   motionInit=0;
   for(ii=0; ii<20; ii++)
   {
      while(!(ADCSRA.4)){};//wait for ADC
      convL=ADCL;
      convH=ADCH;
      convH=convH<<8;
      convH+=convL;
      motionInit+=((((float)convH)/1023.0)*5.);
   }
   motionInit=motionInit/20.;//avg of 20 readings
   if(streamA) UCSRB =0b00011000;//enable receiver and
transmitter
   PORTD.6=1;

}
```

```
void systemStop(void)
{
   TCCR1B=0x0b;//prescale by 1/64 and clear on compare
   OCR1AH=0x00;
   OCR1AL=250;//compare on 25 1mSec
   UCSRB =0b00011000;//enable receiver and transmitter
   gets_int();//enable Rx
}

void keypadDeb(void)
{
   keypdTime=keypdInt;
         switch(currState)
            begin
            case Release:
               keypad();//scan keypad
               if(butnum<11)//button is pressed
               begin
                  currState=Debounce;
                  maybe=butnum;
               end
               break;
            case Debounce:
               PORTC.0=1;
               keypad();//scan keypad
               if(butnum==maybe)//same button as before
                  currState=DetectTerm;
               else//reset debounce machine
               begin
                  maybe=0;
                  currState=Release;
               end
               break;
            case DetectTerm:
               currState=stillPressed;
               if((butnum<10) && (butnum>=0))
               begin
                  act=0;//reset activation button
                  if(keypdIndex<3)//first three keys, save in
array
                  begin
                     keypd[keypdIndex]=butnum + 0x30;//store &
convert to ASCII
                     putchar(keypd[keypdIndex]);//display on
terminal
                     keypdIndex++;
                        restart=res;//count 1.5sec
                  end
                  else//4th key
                  begin
                     keypd[keypdIndex]=butnum + 0x30;//store &
convert to ASCII
                     putchar(keypd[keypdIndex]);//display on
terminal
                     putchar(' ');
                     keypd[4]=0;//string null
```

```
                    sscanf(keypd,"%d", &code);//convert to int
                    ready=1;//go to alarm state machine
                    funcCode=1;//keypad function
                 end
              end
              else if(act==10 && butnum==10)//activation button
is pressed for 1.2Sec
              {
                 PORTD.6=0;
                 act=0;
                 ready=1;
                 funcCode=8;
              }

              break;
           case stillPressed:
              keypad();
              if(butnum!=maybe)//different button or none
                 {currState=debounceRelease;}
              else if(butnum==10)//remote activation button
              {
                 keypdIndex=0;//reset keypad
                 act++;//inc pressed time
                 currState=Release;
                 PORTD.6^=1;
              }
              break;
           case debounceRelease:
              keypad();
              if(butnum==maybe)
              {
                 currState=stillPressed;
              }
              else
              begin
                 maybe=0;
                 currState=Release;
              end
              break;
           default:
              maybe=0;
              currState=Release;
              keypdIndex=0;
              break;
        end   //end switch

}   //end if
```