DESIGN AND CONSTRUCTION OF DTMF BASED HOME AUTOMATION SYSTEM.

BY
OGUNDELE, OLADIMEJI EMMANUEL
(2005/22070EE)

DEPARTMENT OF ELECTRICAL/COMPUTER ENGINEERING

OCTOBER 2010

DESIGN AND CONSTRUCTION OF DTMF BASED HOME AUTOMATION SYSTEM.

BY:
OGUNDELE, OLADIMEJI EMMANUEL
(2005/22070EE)

A PROJECT SUBMITTED TO THE DEPARTMENT OF ELECTRICAL/COMPUTER ENGINEERING.
FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA
IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD BACHELOR OF ENGINEERING (B.Eng) DEGREE

DECLARATION

I Oladimeji Ogundele Emmanuel declare that this work was done by me and has never been presented elsewhere for the award of a degree. I also relinquish the copyright to Federal University of technology, Minna.

Oladimeji Ogundele Emmanuel

(Name of student)

17/11/10

(Signature and date)

Mr. J.Ajiboye

(Name of supervisor)

17/11/10

(Signature and date)

Eng A.G Raji

(Name of H.O.D)

(Jan. 11, 2011)

(Signature and date)

Engr. Dr. G. L. Igbeka

(Name of external examiner)

1/12/10

(Signature and date)

DEDICATION

This project is dedicated to the Almighty God, my parents and my siblings.

ACKNOWLEDGEMENT

I would like to express sincere gratitude to God Almighty for the gift of life, my family for their support and finally my supervisor; Mr. Ajiboye and Mr. Olawoye for their patience and understanding. God bless you all.

ABSTRACT

This project is a remote control system using DTMF (Dual tone multi-frequency) signals. This home automation project (system) employs the use of telephone lines. The system consists of two systems (the phones and the receiver system); the remote control system (the caller phone) and the phone monitoring system (the receiver system). Any phone could be the remote control system but the receiver system has a phone attached to it. The remote control system uses the dual tone multi-frequency signals to control the operation of the various appliances connected to the phone monitoring system. The devices connected to the phone monitoring system are controlled by predetermined commands. There also a provision for feedback to tell the human if the task was executed or not.

TABLE OF CONTENTS

List of figures

CHAPTER ONE

1.0 INTRODUCTION

1.1 BACKGROUND

A new and essential part of household electronics is domestic automation. Currently, many attempts have been made towards achieving this. This has become inevitable in order to achieve greater control and efficiency. Home automation also reduces the need of humans in the control of appliances.

This can be achieved using a remote means. The method of interest in this project is the remote automation means.

1.2 SCOPE OF WORK

This project work covers the following areas listed below;

- An overview of some remote control means available.
- A description of some essential concepts essential to the project realization.
- A description of the components used in the circuit implementation.
- Circuit design, analysis and implementation.
- Project packaging
- Project application, limitations, problems encountered and recommendations.

1.3 METHODOLOGY

The approach used in the actualization of remote control of appliances in this project is based on Dual tone Multi-Frequency (DTMF) using the global system for mobile communication (GSM). The method employs DTMF signals in a keying system, where DTMF tones are used at the site of automation by an interface device to power the desired electrical/electronic devices.

1.4 CONCEPT OF DUAL TONE MULTI-FREQUENCY

Dual tone multi-frequency (DTMF) signaling is used for telephone signaling over the line in the voice frequency band to the call switching center. DTMF is an example of a multi-frequency shift keying (MFSK) system. Today, DTMF is used for most call setup to the telephone exchange (in most developed regions in the world).

The DTMF keypad is laid out in a 4x4 matrix with each row representing a low frequency, and each column representing a high frequency. Pressing a single key such as "1" will send a sinusoidal tone of the two frequencies 697Hz and 1209Hz. These tones are then decoded by switching center in order to determine which key was pressed. This is done by converting these frequencies into binary codes.

The tone frequencies as defined by the precise tone plan are selected such that harmonics and inter-modulation products will not cause an unreliable signal. No frequency is a multiple of another and the difference between any two frequencies does not equal any of the frequencies. The frequencies were initially designed with a ratio 21/19, which is slightly less than a whole tone.

Dual tone multi-frequency signals are decoded using the Goertzel Algorithm and are also called multi-frequency pulsing and multiple frequency signaling.

1.5 STATEMENT OF PROBLEM

In most modern homes filled with electronic apparatus and appliances, it is useful for the owner to exercise some form of centralized control over the functions in the house. Currently, when the owner needs to turn on the driveway and living room lights before turning on the the alarm and going to bed, he has to walk to the garage, then to the front door, then to the alarm box, then to bed.

When the owner leaves his house to go on holiday, he cannot control any of the devices in his house while he is away and cannot tell whether someone has breached the security or whether he had left the bedroom light on. If the alarm has been triggered at his home, there is no way that the owner can become aware of this until he returns home.

In order to achieve efficiency alongside user-friendliness, the appliances must be able to be moved around the house and still retain their ability to communicate with the system. A degree of automation is needed in a house so that certain functions in the house occur automatically, for example the outside light can turn on when it becomes dark outside. There is a need for a reliable, secure and interactive system that exercises full control over the electric and electronic aspects of the house, with the potential to be accessed from across the globe.

No current solution offers real security in terms of control and intrusion. Most others are expensive and difficult to install.

1.6 SOURCES OF INFORMATION

Different sources of information were consulted during the design and construction of the project. These include past project reports on home automation, as well as websites on the internet. Information was also gathered from fellow students who had knowledge on home automation.

CHAPTER TWO

2.0 LITERATURE REVIEW

A remote control is an electronic device used for the remote operation of a machine. The term sometimes abbreviated to "remote" and also called a flipper or "chicker" is most commonly used to refer to a remote for television or other consumer electronics such as stereo system and DVD players and to turn on and off a main plug. Remote controls for these devices are usually small handheld objects with an array of buttons for adjusting various settings such as television channel, track number, and volume. In fact, for the majority of modern devices with this kind of control, the remote contains all the function controls while the controlled device itself only has a handful of essential primary controls. Most of these remotes communicate to their respective devices via infra-red (IR) signals and a few via radio signals.

2.1 DEVELOPMENT OF REMOTE CONTROL OF APPLIANCES

One of the earliest examples of remote was developed in 1893 by Nikola Tesla, and described in his patent, named method of and apparatus for controlling mechanism of moving vehicles. The first remote-controlled model air-plane flew in 1932. The use of remote control technology for military purposes was worked on intensively during the Second World War. One result of this is the German "wasserfall" missile. The first remote intended to control a television was developed by zenith Radio Corporation in the early 1946. The remote, unofficially called "Lazy bones" used a wire to connect to the television set. To improve the cumbersome setup, a wireless remote control was created in 1955. The remote called flashmatic worked by shinning a beam of light into a photoelectric cell. Unfortunately, the cells did not distinguish between light from the remote and light from other sources. The flashmatic also required that the remote control be pointed accurately to the receiver.

The invention of the transistor made possible cheaper electronic remotes that contained a piezoelectrical crystal that was fed by an oscillating electric current at a frequency near or above the upper threshold of human hearing. The receiver contained a microphone attached to the circuit that was tuned to the same frequency.

In the early 1980s, when semi-conductors for emitting and receiving infra-red radiations were developed, remote controls gradually switched to that technology which as at 2006 was still widely used. December 12, 1901 scientist and Nobel Prize winner Guglielmo Marconi was able to transmit across the Atlantic Ocean for the first time in history. One century later, wireless communication has become widespread. People can now interact even across continents. Voice calls are only one of the options of the functionality of a mobile phone. Other functions include; sending of faxes, exchange of SMS, mails, music and even internet connectivity. No matter the source of the information, it can be reduced to smaller data packets and delivered to a mobile phone.

The twenty-first century has also experienced great advancements in communication and control of devices such as home automation systems, Bluetooth Wi-Fi and tracking devices on cars and mobile phones and more recently the control of devices via global system mobile(GSM).

In 2005, a student by the name Akande Adeleke (99/9312EE) was able to control and monitor devices (e.g. fluorescent light, motor or air conditioner) using a mobile phone but via sms. Two phones were used; one the master, the other the slave. The slave was interfaced or configured to a tiny planet circuit through a data cable connected to DB9 of the circuit, it uses AT command protocol to communicate with the ATtiny12 (microcontroller), and also transmit and receive sms to and fro the master phone. The design also consisted of a relay which was connected to the device to be controlled and a detector connected to the device to be monitored. The detector is a transducer to convert the input signal from the device to be monitored into an electrical equivalent signal which was fed back to the microcontroller and appropriate actions were taken. The project had no form of security for the primary user (owner).

In 2007, a student named Yakubu Mohammed (2001/12145EE) also worked on something similar. He was able to achieve control by connecting the device to be controlled to a GSM and interfaced with an Ericsson T105 mobile phone and a computer to a microcontroller (ATmel98515). The serial port of the microcontroller was interfaced with the serial port of the phone, while the parallel port of the microcontroller (Port B) was interfaced to the parallel port of the computer. A program which had been developed to send commands to the microcontroller just as the phone does was loaded on the computer. The program on the computer was updated by the microcontroller after a change had been effected on the states of each of the devices connected to the computer, whether it had been implemented by the phone or computer.

The microcontroller was able to access the serial port of the mobile phone via its AT command instruction set. When the device was powered on, the microcontroller initialized the phone and selected the phone memory as the default storage location in preference to that of the simcard. At periodic intervals the microcontroller accesses the phone for new text messages at the pre-assigned memory location. There are two categories of messages obtainable; a valid message (command) and an invalid message (command). If invalid, it will be deleted from the phone memory. If valid, the microcontroller will check to see what the command it is and then execute it.

Also in 2007, Ubah T. Ubah (2002/12131EE) designed a similar circuit but using a network (the internet) to control devices. In his design the various loads were connected to a terminal through a custom built interface circuitry, which acted as the switching or control device. The computer used RS-232 interface which is a serial communication port for sending and receiving control states. The computer ran an application which acted as a server software, this software directly communicated with the serial port to send or receive data. Also a web server was used to serve the page containing the client or command console would have been installed on the system.

The server was connected to a network where other computers on the network would be able to access the command user interface which was used to control the devices; both client and server were written in java. The design was able to control three devices out of which two were static while the other was dynamic, dynamic in the sense that it was able work with varying voltage levels. Controlled devices were interfaced with the host computer using a network (the internet). The switching circuitry is made up of an AT89C2051 microcontroller which is used to receive and decode commands signals sent from the server computer via its RS-232 interface. The microcontroller is equipped with a full duplex UART or serial port which enables it receives and sends serial data. These command signals are byte-wise data and contain bits which indicate whether a particular device is to be switched on or off. A logic 1 turns on the device while a zero (0) turns off the device. The components used for switching on and off were triacs and relays.

This project did not provide for a security system that would bar intruders.

CHAPTER THREE

DESIGN AND ANALYSIS

3.0 DESIGN OBJECTIVES

In implementing this project, certain objectives were borne in mind. They are;

1) The operation of the project should not have any distance limitation (except for areas where there's no network coverage).

2) It should work as a non line of sight (NLOS) device.

3) It should completely reject any kind of audio noise.

4) It should control multiple appliances.

5) The circuit should be powered from the mains to reduce the cost of changing batteries in the long run.

6) It should be able to withstand environmental conditions especially in the tropical region (i.e. Nigeria)

3.1 DESIGN CONSIDERATION

While embarking on the cell phone remote control, a number of factors were considered with obvious emphasis on the following:

1) **COST:** This was a major consideration in the choice of the components used except where the particular components were cardinal to the realization of the project as in the DTMF decoder.

2) **AVAILABILITY:** This was given the duly considered because without due consideration, the whole design process would be futile. Thus, prior to the final circuit

design, a market survey to determine the availability of the various components was carried out.

3) **MAINTAINABILITY:** To ensure easy maintenance of the project, the modular design philosophy was adopted. The casing was made such that it could be dismantled as any time the need arises.

4) **AESTHETICS:** Beauty and user friendliness were also considered in the choice of the package used.

5) **RELIABILITY:** To ensure the project worked well, the components were well tested with extreme precision. The casing materials used were ones that can withstand varying temperatures, humidity, vibration, and dust were used.

3.2 DESIGN CALCULATIONS

3.2.1 POWER SUPPLY

Adequate voltage supply is of paramount importance for the proper functioning of all the components. Hence proper provision was made to ensure a stable power supply system.

From the circuit diagrams(fig 3.3 and fig 3.4) the voltage requirements from the power supply to the DTMF decoder, microcontroller, and relays are either +5v or +12v. The current requirements for the components are as outlined below;

1) DTMF decoder (CM8870DE)............. 9.0mA

2) Microcontroller (Atmel 8952)................10.0mA

3) Relays .... 30.0mA (each)

(With four in parallel).......................120.0mA

Total...139.0mA

Knowing these, the following components and values were chosen

a) 220/12v, 500mA transformer: this also falls within the 6v-24V input limit of the 7805 voltage regulator and has a supply current(500mA) greater than the circuit requirement.

b) 7805 voltage regulator to produce a fixed +5v output.

c) Two 100µf capacitors for filtering.

d) A bridge rectifier (W08m)

e) 500mA fuse.

For a better understanding of the power supply circuit, its block diagram is shown below.

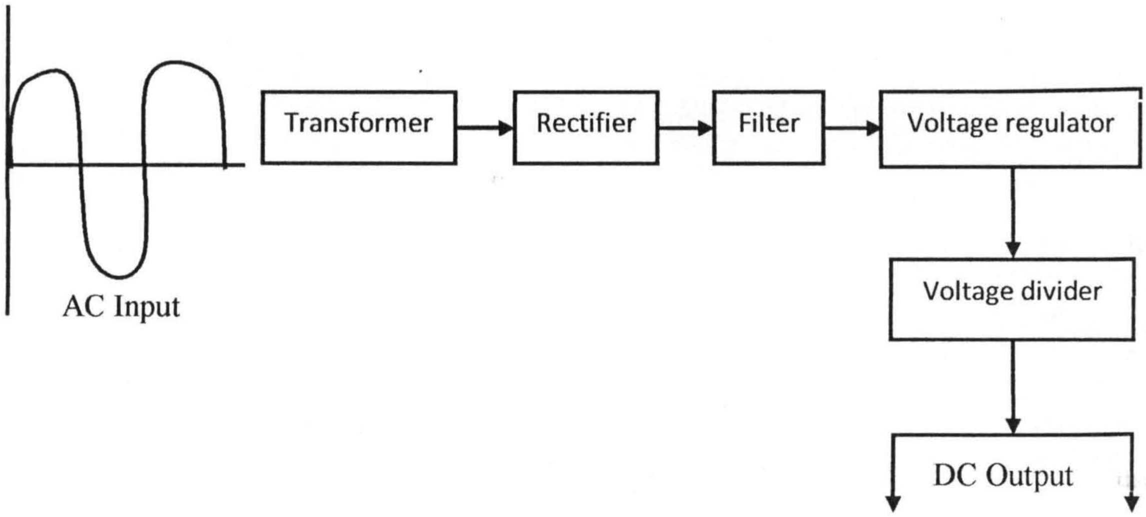
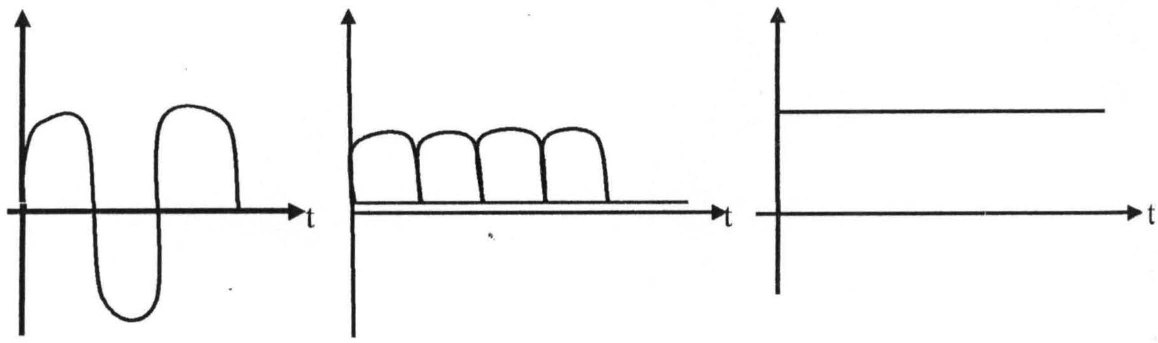


Fig 3.1: A Typical Dc Power Supply Block Diagram.

a) TRANSFORMER: A 220/12v, 500mA transformer was used to step down the A.C. supply voltage to 12v so as to suit the requirement of the solid-state electronic devices and circuits fed by the DC power supply. The transformer has a 500mA fuse connected to it, to prevent damage to circuit due to excessive current from mains supply. The transformer output is fed into the rectifier section.

b) RECTIFIER: The full wave bridge rectifier employs four IN4001 diodes to convert the transformer 12v AC output into pulsating DC voltage as illustrated below;



Transformer Output Rectifier output Filter Output

(a) (b) (c)

Fig 3.2: Voltage output at the transformer, rectifier and filter.

c) FILTER: The capacitor removes the fluctuations (ripples) present in the output voltage supplied by the rectifier. A capacitor-filter was preferred to clock filter because it has a higher DC output voltage, low peak inverse voltage (PIV) requirements for diodes and are smaller weight than inductors and clock filters. Basically, a high capacitance (1000µf) was used in order to:

i) Reduce the time of flow of current pulse through the diode.

ii) Increase the peak current in the diode.

iii) Reduce ripple magnitude (ripple factor)

This is illustrated below:

Output frequency = 2 x input frequency (3.0)

But, input frequency from the mains= 50Hz

Thus, output frequency = ripple frequency

= 2 x 50Hz= 100Hz.

The resistive load from the DTMF decoder, microcontroller and relays connected to the power supply as shown.

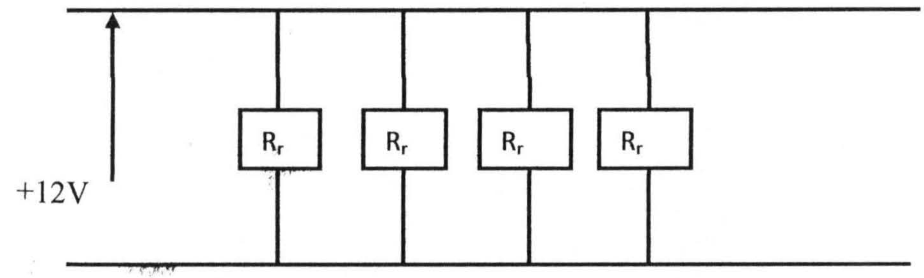


Fig 3.3: Load resistance of relays in the circuit

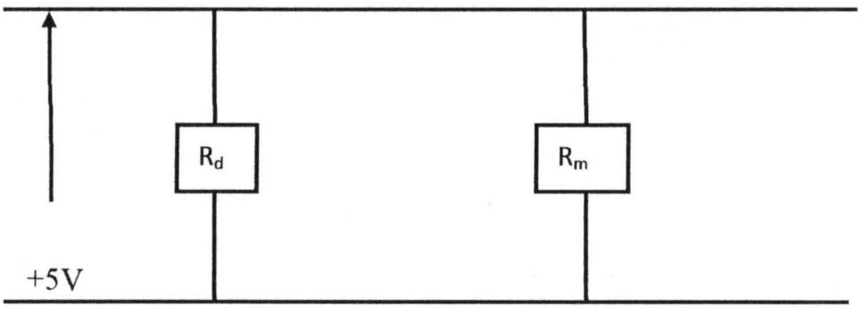


Fig 3.4: Load resistance of the decoder and microcontroller.

Where R_r, R_D and R_m are the resistances of the relays, decoder and microcontroller.

From manufacturer's specifications

Rr = 400Ω

Rd (gain select resistive load) = 50KΩ

Current into the microcontroller = Im = 28mA

Therefore, $R_m = V_{DD}/I_m$ (3.1)

$= 5 \times 10^3 / 28 = 178.6\Omega$

Equivalent relay resistance = R_{rT}

$1/R_{rT} = 1/400 + 1/400 + 1/400 + 1/400 = 1/80$

$R_{rT} = 80\Omega$

Total equivalent load resistance = R_L

$1/R_L = 1/R_{rT} + 1/R_D + 1/R_m = 1/80 + 1/50000 + 1/178.6$

$1/R_L = 0.0125 + 0.00002 + 0.005599 = 0.018119$

$R_L = 55.19\Omega$

But ripple factor,

$Y = 1/4\sqrt{(3fCR_L)}$ for a full wave rectifier (19) (3.2)

By substituting our parameters, we have

$Y = 1/4\sqrt{3(100)(1000 \times 10^{-6}) \times 55.19\Omega}$

$Y = 1/4\sqrt{3 \times 5.510} = 1/38.2368 = 0.026$

This is a very low ripple factor of 2.6%.

d) REGULATION: A monolithic integrated circuit voltage regulator(7805) was used to keep the terminal input voltage to the voltage sensitive ICs constant even when

i) AC input voltage to the transformer varies.

ii) The load varies.

A zener diode or a transistor could have been used for this purpose but for simplicity of implementation.

e) VOLTAGE DIVIDER: Voltage dividing was also an essential part of the supply to provide the +5V and +12V dc needed in the project.

3.2.2 DRIVE CIRCUITRY

The drive circuitry consists of all other parts of the project except the power supply.

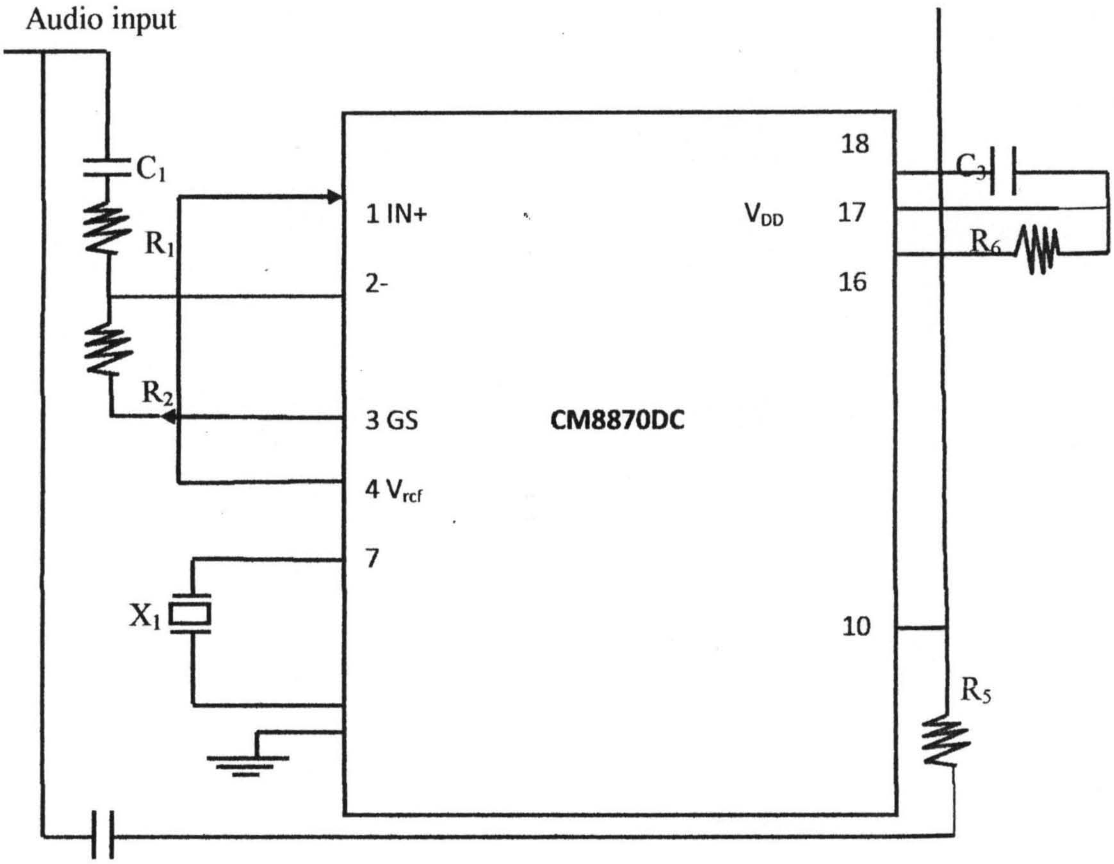


Fig 3.5: A part of the drive circuit; the tone decoder.

3.2.2.1 CRYSTAL OSCILLATOR

The internal clock circuit of the decoder is completed with the addition of an external 3.579545MHz crystal with 0.1% tolerance. The tolerance value which is very low must be adhered to during selection for proper functioning of the decoder.

A logic high applied to pin 6(power down) will power down the device to minimize the power consumption in a standby mode. It stops the oscillator and functions of the filters. So too, logic high input to pin 5 (inhibit mode) will inhibit the detection of tones representing characters A, B, C and D. Since these were not necessary, pins 6 and 5 were left unconnected.

3.2.2.2 STEERING CIRCUIT

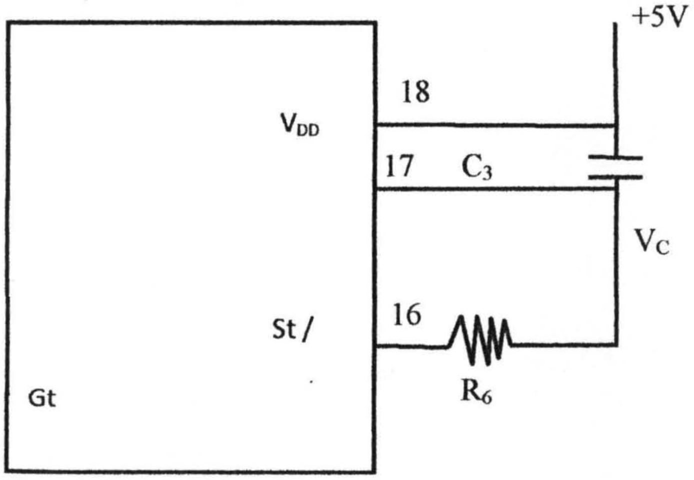


Fig 3.6: The steering circuit of the decoder

The minimum DTMF signal duration required for valid recognition t_{REC} is 40mS.

$$t_{REC} = t_{DP} + t_{GTP} \tag{3.3}$$

If the time to detect the presence of valid DTMF signals, t_{DP} is 14mS (which is within the 5-14mS specified by the manufacturer) then,

$$t_{REC} = 14mS + t_{GTP} \tag{3.4}$$

but t_{GTP} = Guide time, tone present is given as;

$$RCln[V_{DD}/(V_{DD}-V_{TSt})] \tag{3.5}$$

Where, V_{TSt} = Threshold voltage of V_c shown above 0.1μf for C_3 is usually recommended for most applications. Thus, the value of R_6 which gave an acceptable value of t_{GTP} was derived as shown below.

$t_{GTP} = RCln[V_{DD}/(V_{DD}-V_{TSt})]$

$V_{DD}-V_{TSt}$ is taken as approximately IV from experimentation.

Therefore,

$t_{GTP} = RCln[V_{DD}]$

If R = 100k, C = 0.1μf, then

$t_{GTP} = 0.1 \times 10^{-6}(10^{5})\ln[5]$

$= 10^{-2}(1.609)$

$=16.1mS$

Substituting into eqn (3.5),gives

$t_{REC} = 14mS + 16.1mS = 30.1mS$

Since 30.1mS value for t_{REC} is less than the maximum that can be acceptable, our assumed value of R_6 and C_3 are in order.

3.2.2.3 MICROCONTROLLER

By the internal design configuration, the microcontroller operates with an external crystal oscillator X_2, in this case a 5.5MHz crystal oscillator, to provide proper timing of the microcontroller operations.

The resistors R_7-R_{12} were experimented and a suitable value of 4.7k was chosen for them . These enabled keeping the memory clear(MCLR) pin of the microcontroller (pin 4) high and trigging of the switching transistors BC945. They also ensure that the maximum load current of the microcontroller (28mA) is not exceeded.

3.3 WORKING OPERATION OF THE INTERFACE DEVICE.

In describing the operation of the project, a block diagram model has been developed. These are the different functional modules that are critical to the realization of the cell phone remote control. The first and of course the most vital part is the dual tone multi-frequency(DTMF) decoder. This receives signals from the mobile phones through the phone's audio channels. The tones that are decoded are sent by the mobile phone through any service provider network that has such license and compatibility. The decoder has an in built intelligent algorithm which enables it to reject noise and stray tones.

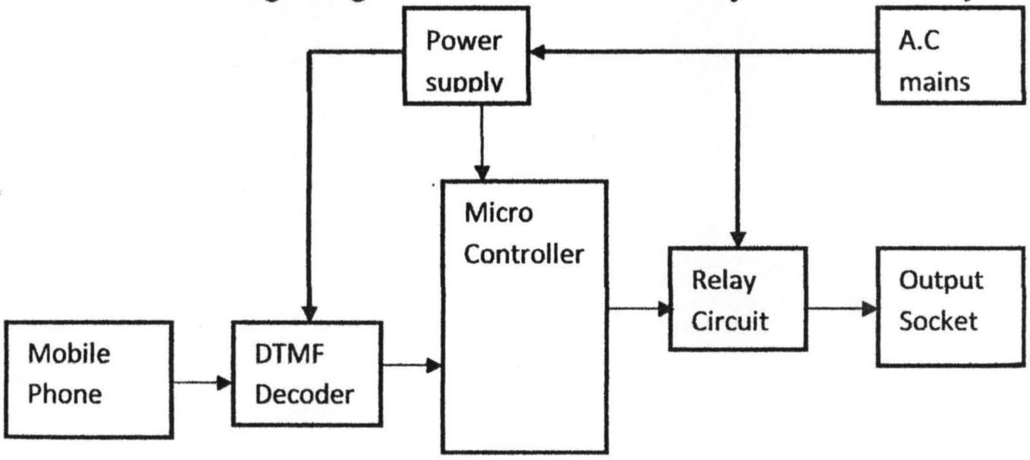


Fig 3.7: Block diagram description of the cell phone remote control

Simultaneously, 0.3-0.9V audio output of the phone is fed to a comparator, LM358 and the DTMF decoder. The output of the comparator is sent to the microcontroller. This enables the microcontroller to trigger the relay circuit that automatically accepts the call after a fixed time. (This eliminates false calls i.e. flashing). Thus, a high from the comparator means a phone call is in progress while a low indicates that the phone is dormant. Eventually, when a valid call is detected by the tone decoder, it outputs binary values representing the codes (i.e. numeric or alphanumeric values) sent by the operator. The microcontroller coordinates every signal within the circuitry. The binary signals it receives from the tone decoder are processed and its output are used to trigger the relay circuit according to how it has been programmed. Each of the appliances has its own unique code, programmed into the microcontroller which is used to control them remotely. The binary signals from the decoder are simultaneously fed to the parallel port of the computer which is used to run the software.

The relay circuitry contains switching transistors and relays that receive signals from the microcontroller. It switches AC from the mains to the output socket. The transistors aside from been switching devices also boost the signal level into the relays in order for them to be triggered.

The power supply circuit is made up of a step down transformer, a rectifier and a regulator. It delivers regulated DC voltage from the AC mains supply to the tone decoder, microcontroller and the comparator.

The output socket consists of sockets used to connect the appliances to the interface device. These are 13-amps sockets. The AC input mains is a regular 220V supply. A two (2) pin plug is used to feed in the 220V supply to the power supply and relays.

3.4 SOFTWARE DESIGN

The construction of this project has two basic parts namely:

- The hardware and
- The software parts.

The software requirement is the assembly language HEX code necessary to enable the microcontroller to combine the operations of the hardware constituent parts.

The software design was developed using the basic principle of first drawing out the flow chart before writing the source code. The flow chart is shown below and the assembly language source code is shown in the appendix.

The objectives of the software design include;

1) To ensure that the memory requirement is as low as possible.

2) To ensure that adequate delay for the proper functioning of the circuitry was incorporated.

3) To reduce the number of errors to be debugged as much as possible.

4) To simplify the codes for easy understanding.

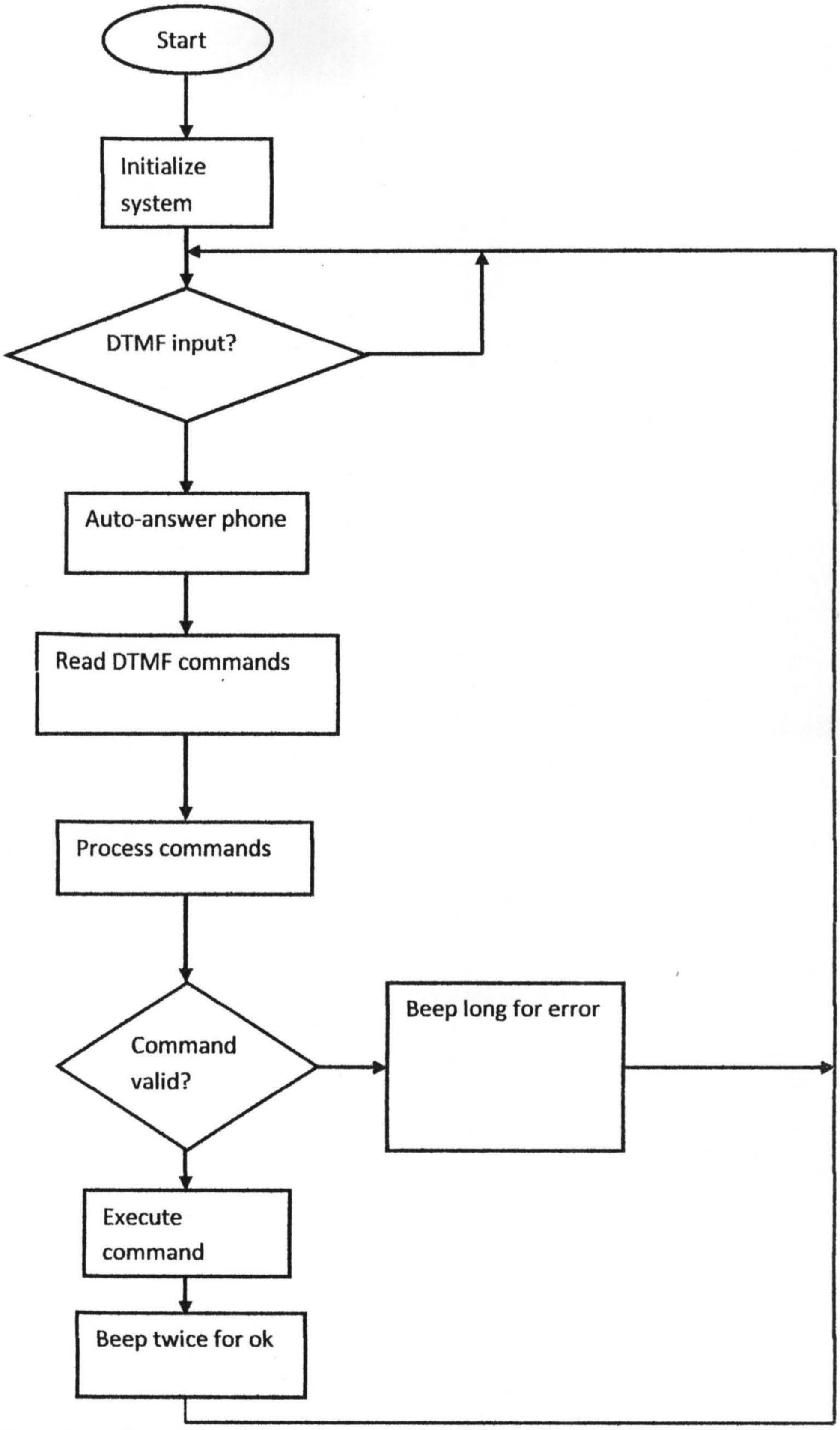


Fig 3.8: A flowchart description of the microcontroller program

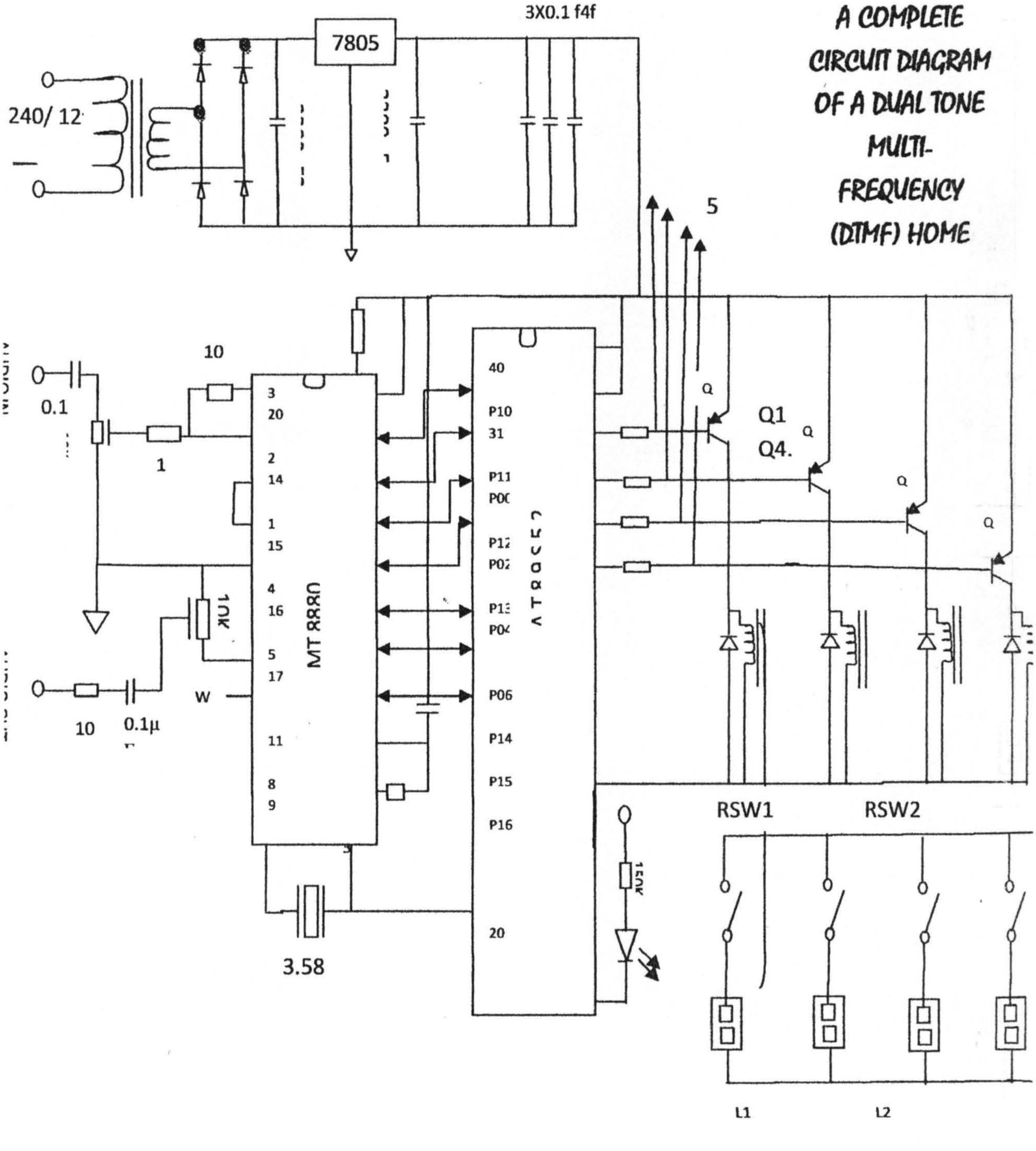


Fig 3.9 the circuit diagram of the device.

CHAPTER FOUR

4.0 IMPLEMENTATION AND TESTING

This was crucial part in the realization of the circuit. Testing of the workability of the project was done in practice on a breadboard. This was after the theoretical achievement of the circuit on a workbench. The choice of breadboard was for the following reasons;

1) Easy removal of faulty components.
2) To ensure the whole system worked well.
3) To enhance better planning and space management on the final (Vero) board.

4.1 POWER SUPPLY IMPLEMENTATION AND TESTING

Sequel to the design of a suitable power supply circuit which has the capacity to deliver a regulated +5V and +12V, the actual implementation was achieved through the steps below;

i) The components were assembled and tested with a multi-meter to ensure they were in good working condition.

ii) The board was cleaned to remove dust and oil, in case there were any.

iii) The position of each component was determined.

iv) The components were carefully soldered to avoid dry joints.

v) The output of the power supply was tested using a digital multi-meter to confirm it had nearly constant values of +5V and +12V DC.

4.2 MAIN CIRCUITRY IMPLEMENTATION AND TESTING

This was achieved using the same procedure adopted for the power supply. The only difference was that the microcontroller source code was first tested on a microcontroller simulator to verify its correctness.

Again, the outputs of the tone decoder, comparator and microcontroller were all tested to ensure that the correct signals and signal levels where output from these components.

4.3 SOLDERING

Components had to be soldered onto the veroboard instead of a printed circuit board. Although, printed circuit board is more reliable and durable (thus it is used for commercial circuits), the Vero board was chosen because it is easy to use.

To ensure good quality soldering, the following precautions were adhered to;

1) Prior to soldering, the board was cleaned thoroughly to ensure it was free from oil and dust.

2) All circuitry and contacts that were not to be soldered were covered with heat resistant materials.

3) Care was taken while soldering to avoid damage to components and Vero board by using IC sockets and by not applying heat in localized regions.

4) After cooling, soldered joints were cleaned and made to have a smooth uniform shiny surface.

4.4 CASING CONSTRUCTION AND ASSEMBLY

The casing was first designed with AutoCAD(any graphic software could be used.) This design was then implemented with a cardboard sheet to actually visualize the dimension and shape of the casing.

A plastic casing was used after a number of variables, which fall into two categories were considered. The categories are:

- Physical factors
- Environmental factors

4.4.1 PHYSICAL FACTORS

The physical considerations for choosing a plastic casing are;

1) APPEARANCE: We wanted a final package beautiful and very neat.

2) WEIGHT: A casing that should be light.

4.4.2 ENVIROMENTAL FACTORS

The environmental factors that also influenced the choice of the used casing are;

1) AMBIENT TEMPERATURE: Since the device is to be demonstrated in a high tropical region (i.e. Minna), the casing should be able to withstand high temperature.

2) HUMIDITY: To ensure that the casing does not rust or water does not get into the internal circuitry, the package must be rustproof, resistant and water tight.

3) SHOCK AND VIBRATION: Due to the adverse effects of shock and vibration, the casing should be spacious, damped and must not be a conductor to prevent shock.

4) PESTS AND INSECTS: The casing should be able to withstand the destructive tendencies of pests and insects.

4.5 SYSTEM RELIABILITY AND MAINTENANCE

Due to the increasing complexity of modern engineering systems, the concept of reliability has become a very important factor in the overall system design. Reliability is important because a designer must design an electronic system that would work both in theory and in practice. The general acceptable definition of reliability is the "Reliability" is the characteristic of systems or components that can be expressed in terms of probability that it will continue to perform its function over a specific period of time and under specified operating conditions.

Thus reliability can be viewed as a measure of successful performance of a system. Reliability which is an inherent characteristic of a system, similar to the system's capacity or power rating, needs to be critically addressed at every stage of the product or system development including design, manufacturing and testing. In design phase, proper design method relating to the components, materials, processes, tolerance and so on, are carefully selected. The objectives at this stage is to ensure that well-established design procedure are applied, known materials and processes are used and the areas of uncertainty are highlighted for further actions. When the initial hardware manufacturing are completed, tests are carefully planned, executed and data collected to generate confidence in the design.

Reliability analysis was not left undone in this design. In fact, right from the beginning of the concept through specifications and design to the implementation of the cell phone remote controller, the system was intended to be reliable. Thus in order to improve reliability, the system was made as simple as possible.

4.6 APPLICATIONS

The cell phone remote control is an interesting project which finds a wide range of applications in security and control systems. These include;

a. It could be used in the remote control of industrial and electrical machines without the need to be physically present at the site.

b. It could be used in the remote control of home appliances.

c. It could be used for locking vehicles to check theft.

d. It could be used for emergency exits.

e. It could be used to activate security alert systems.

4.7 PROBLEMS ENCOUNTERED

The various difficulties experienced during the construction of the project include

1) The audio signal levels of different phones vary considerably. Thus it was difficult to set the variable resistor, R_5 value to accommodate a number of mobile phones

2) The DTMF crystal oscillator, X_1 with a value of 3.579545 MHz was not easy to find especially due to the high precision required.

3) The DTMF decoder was not readily available and this caused a great deal of delay in construction.

4) The erratic nature of power supply posed some difficulties during soldering.

CHAPTER FIVE

5.0 CONCLUSION

5.1 SUMMARY OF WORKDONE

From the preceding chapters, an attempt to explain the design and implementation of the cell phone remote control has been made. This involves the description of the concept of mobile phone remote control, the dual tone multi-frequency and the various types of remote control systems and the description of the various components used.

The project also involved understanding the objectives, considerations, design calculation used as well as the microcontroller assembly language.

5.2 CONCLUSION

Mobile phones have become an indispensable part of our life. Our system uses a microcontroller and a GSM mobile phone for its operations. The systems can be used as a test bed for any application that requires on-off switching based applications. Wireless controlled home appliances in the comforts of any environment will revolutionize our way of living.

In this project low cost, secure, ubiquitously accessible, remotely controlled solution for automation of homes has been introduced. The approach discussed in this project is novel and has achieved the aim of controlling home appliances remotely using DTMF tones generated from GSM mobile phone while dialling digits on the GSM mobile phone, satisfying user needs and requirements. Implementing this project in the home provides home owners with the privilege to automate their home so that they can take advantage of the technological advancement available today. In addition, automating one's home is an excellent way to, cut energy costs, and gain increased control over the surroundings. Home automation is an inexpensive project, and the much usefulness is only limited by one's imaginations.

This project has proven to be very interesting and challenging. It has helped to bring together all the theoretical knowledge gained during these years and an appreciation for the practicality. It has also helped in getting more informed about the world of embedtronics and domotics.

5.3 RECCOMMENDATIONS

The aim of students' project work is to enhance the creativity and understanding of engineering concepts by students. The following recommendations can help in this regard.

1) A USB port can be used to further improve the design in order to increase the compatibility of the control unit with other systems.

2) It is also recommended that alternate sources of power should be used to power the circuit and devices to be controlled to ensure efficient control.

3) The department through the university should seek industrial and corporate sponsorship for students' projects and good and quality projects should be recognized so as to encourage them and younger students.

REFERENCE

1. Conte, G., & Scaradozzi, D. (2003). Viewing home automat ion systems as multiple agents systems. *RoboCUP2003*, Padova, Italy. Retrieved from : http://www.robosiri.it/ROBOCUP_2003/ROBOCUPSITOSIRI/articles/pdf/Conte.pdf
2. Alkar, A. Z., & Buhur, U. (2005). An Internet Based Wireless Home Automation System for Multifunctional Devices. *IEEE Consumer Electronics, 51*(4), 1169-1174.Retrieved from:http://www.thaieei.com/embedded/pdf/Automation/20022.pdf
3. Murthy, M. V. R. (2008). Mobile based primary health care system for rural India. *W3C workshop on Roleof Mobile Technologies in Fostering Social Development*, June, 2008.
4. Jawarkar, N. P., Ahmed, V., Ladhake, S. A. & Thakare, R. D. (2008). Micro-controller based Remote Monitoring using Mobile through Spoken Commands. Journal Of Networks, 3(2), 58-63. Retrieved from : http://www.academypublisher.com/jnw/vol03/no02/jnw03025863.pdf
5. Potamitis, I., Georgila, K., Fakotakis, N., & Kokkinakis, G. (2003). An integrated system for smart-home control of appliances based on remote speech interaction. *EUROSPEECH 2003, 8th European Conference on Speech Communication and Technology*, pp. 2197-2200, Geneva, Switzerland, Sept. 1-4, 2003. Retrieved from http://www.wcl.ee.upatras.gr/ai/papers/potamit is14.pdf
6. A journal of The Leading Global Real-World Media and Entertainment Company.Copyright 2010 Discovery Communication Inc.

7. Wikimedia Foundation Incorporation (2010). A discussion on Home automation.

8. Wikimedia Foundation Incorporation (2010). A discussion on Microcontroller.

9. Heath, Steve (2003). Embedded system design._EDN series for design engineers (2 ed.). Newnes. pp. 11–12.

10. 2002-2008 Microcontroller Pros Corporation. The Evolution Of Flash Microcontrollers

11. Delgado, A. R., Picking, R., & Grout, V. (2006) Remote-controlled home automation systems with differentnetwork technologies. *Proceedings of the 6th International Network Conference (INC 2006)*, University of Plymouth, 11-14 July 2006, pp. 357-366. Retrieved from http://www.newi.ac.uk/groutv/papers/p5.pdf

12. Ciubotaru-Petrescu, B., Chiciudean, D., Cioarga, R., & Stanescu, D. (2006). Wireless Solutions for Telemetry in Civil Equipment and Infrastructure Monitoring. *3rd Romanian-Hungarian Joint Symposium on Applied Computational Intelligence (SACI)* May 25-26, 2006. Retrieved from : http://www.bmf.hu/conferences/saci2006/Ciubotaru.pdf

APPENDIX

```
#include <reg52.h>


//#define debug

#include <intrins.h>

#define xtal 3580000UL

#define dx_port1 P0

#define dx_port2 P2

#define tone_port P1

//*************************

//*************************

#define max_cmd 15

#define T0_reload 0x06

#define cmd_stop '#'

#define cmd_start '*'

#define t_base 250

#define sec_1 1193

#define timeout_count sec_1*5

#define write_delay 3000UL

#define max_char 16

//***********************

//***********************

#define slave_address 0xa0

#define write_flag 0x00

#define read_flag 0x01

#define pin_lenght_size 4

#define pin_address_start 0x00
```

```
#define presets_address pin_address_start+pin_lenght_size
//************************
//************************
#define load_ctrl 8
#define pin_change 10
#define max_load '4'
#define min_load '1'
#define pin_start 1
#define pin2_start 6
#define load_off '0'
#define load_on '1'
#define load_1 '1'
#define load_2 '2'
#define load_3 '3'
#define load_4 '4'

#define freq_reload 184
//*********************
//*********************
sbit sda= P3^7;
sbit scl=P3^6;
sbit tone_dx=P3^0;
#define freq_dx tone_dx
sbit load1_dx =P2^0;
sbit load2_dx =P2^3;
sbit load3_dx =P2^6;
sbit load4_dx =P0^6;
sbit error_led =P3^5;
```

```
sbit reset= P3^5;

sbit tone_in=P3^2;

//*****************

//*****************

sbit lcd_out = P0^0;

sbit lcd_clock= P0^3;

sbit lcd_load=P0^1;

sbit lcd_en=P0^2;

#define lcd_rs lcd_out

//*******************

sfr T2MOD = 0xc9;

//*********************

//*********************

volatile unsigned int count;

unsigned char data buffer[max_cmd+2];

unsigned char preset,error,timeout;

//**************************

//**************************

code char table[]={"D1234567890*#ABC"};

//**********************

//**********************

unsigned char store_pin(unsigned char data *);

void tone_off(void);

//*****************

//*****************

void tf0_isr(void) interrupt 1

{

        if(!(--count)){    TR0=0;timeout=1;}
```

```
}
//*********************
//*********************
void start_timeout(void)
{
        TR0=0;
        TF0=0;
        TL0=T0_reload;
        count=timeout_count;
        timeout=0;
        TR0=1;
}
//******************
//******************
void stop_timeout(void){TR0=0;}
//******************
//******************
void delay_ms(void)
{
        start_timeout();
        count=1;
        while(!timeout);        //wait for 250 machine clock tickes here
        stop_timeout();
}
//******************
//******************
void write_timeout(void)
{
```

```
        unsigned char z=write_delay/t_base;

        while(z){delay_ms();z--;}
}
//************************
//************************
void delay_sec(float sec)                    //maximum of 16 seconds possible here
{
        unsigned int z=sec_1;

        z*=sec;
        while(z){delay_ms();z--;}
}
//****************
//****************
void serialize(unsigned char c,unsigned char reg_select)
{
        unsigned char z;

        lcd_load=0;
        lcd_clock=0;
        lcd_en=0;

        for(z=0;z<8;z++)
        {
                lcd_out=!(c&0x80);
                lcd_clock=1;
                lcd_clock=0;
```

```
		c<<=1;
	}
	lcd_rs=reg_select;
	lcd_load=1;
	lcd_load=0;
	lcd_en=1;
	lcd_en=0;
}
//********************
//********************
void lcd_data(unsigned char c)
{
	unsigned char z=250;
	serialize(c,1);
	while(--z);
}
//****************
//****************
void lcd_cmd(unsigned char c)
{
	unsigned char z=250;
	serialize(c,0);
	while(--z);
}
//****************
//****************
void lcd_clear(void)
{
```

```
        lcd_cmd(0x01);
}
//****************
//****************
void lcd(char code *ptr)
{
        while(*ptr)lcd_data(*ptr++);
}
//*************************
//*************************
void init_lcd(void)
{
        lcd_cmd(0x38);
        write_timeout();
        lcd_cmd(0x38);
        delay_ms();
        delay_ms();
        lcd_cmd(0x38);
        delay_ms();
        delay_ms();
        lcd_cmd(0x0c);
        delay_ms();
        delay_ms();
        lcd_cmd(0x01);
        delay_ms();
        delay_ms();
        lcd_cmd(0x06);
        delay_ms();
```

```
        delay_ms();
}
//*******************
//*******************
void dly(void)                                                  //
{
        _nop_();
        _nop_();
        _nop_();
        _nop_();
        _nop_();
        _nop_();
}
//********************
//********************
void i2c_start(void)
{
        sda=1;
        scl=1;
        dly();
        sda=0;
        dly();
        scl=0;
        dly();
}
//*****************************
//*****************************
```

```
void i2c_stop(void)
//
{
	sda=0;
	dly();
	scl=1;
	dly();
	sda=1;
}
//************************
//************************
void no_ack(void)                                        //
{
	sda=1;
	_nop_();
	scl=1;
	_nop_();
	scl=0;
}
//**********************
//**********************
unsigned char write_byte(unsigned char p)       //
{
	unsigned char d;
	for(d=0;d<8;d++)
	{
	sda=p&0x80;
	_nop_();
```

```
        scl=1;
        dly();
        scl=0;
        dly();
        p<<=1;
        }

        sda=1;
        _nop_();
        scl=1;
        _nop_();
        p=sda;
        scl=0;
        return p;
}
//************************
//************************
unsigned char read_byte(void)           //
{
        unsigned char x,c;

        c=0x00;
        sda=1;
        for(x=0;x<8;x++)
        {
        c<<=1;
        scl=1;
        dly();
```

```
        c|=sda;
        scl=0;
        dly();
        }
        return c;
}
//************************
//************************
unsigned char write(unsigned char address,unsigned char c)        //
{
        i2c_start();
        if(error=write_byte(slave_address|write_flag))goto write_abort;
        if(error=write_byte(address))goto write_abort;
        if(error=write_byte(c))goto write_abort;
        i2c_stop();
        write_timeout();
        return 0;

write_abort:
        i2c_stop();
        return 1;
}
//************************
//************************
unsigned char read(unsigned char address)                        //
{
        unsigned char data_read;
        i2c_start();
```

```
        if(error=write_byte(slave_address|write_flag))goto read_abort;
        if(error=write_byte(address))goto read_abort;
        i2c_start();
        if(error=write_byte(slave_address|read_flag))goto read_abort;
        data_read=read_byte();
        no_ack();
        i2c_stop();
        error=0;
        return data_read;

read_abort:
        i2c_stop();
        error=1;
        return 0;
}
//;************************
//*************************
void set_presets(void)
{
        dx_port1=0xff;
        dx_port2=0xff;
        if(preset&0x01)load1_dx=0;
        if(preset&0x02)load2_dx=0;
        if(preset&0x04)load3_dx=0;
        if(preset&0x08)load4_dx=0;
}
//*******************
//*******************
```

```
unsigned char read_dtmf(void)
{
	unsigned char z;

	if(tone_in)                             //if STD low grab data
	{
		z=tone_port&0x0f;
		while(tone_in); //wait until STD deasserted
		return table[z];
	}
	else return 0;
}
//*******************
//*******************
unsigned char get_dtmf(void)
{
	unsigned char data *ptr=buffer;
	unsigned char x=0;
	unsigned char c;

	while(1)
	{
		start_timeout();
		while((!(c=read_dtmf()))&&(!timeout));
		if(timeout)
		{
			stop_timeout();
			error=1;
```

```
        return 0;
}

if(c)
{
        stop_timeout();
        if(c==cmd_stop)
        {
                if(!x)
                {
                        error=1;
                        return 0;
                }
                error=0;
                return x;
        }
        *ptr=c;
        ptr++;
        x++;
        #ifdef debug
        lcd_data(c);
        #endif
        if(x>max_cmd)
        {
                error=1;
                return 0;
        }
}
```

```
        }
}
//********************
//********************
unsigned char load_presets(void)
{
        preset=read(presets_address);
        if(error)return 1;
        else return 0;
}
//*******************
//*******************
void init_irq(void)
{
        ET0=1;
        ET2=1;
}
//*******************
//*******************
void init_timer(void)
{
        TCON=0x00;
        TMOD=0x22;
        TH0=T0_reload;
        TL0=T0_reload;

        T2CON=0x00;
        T2MOD=0x00;
```

```
        RCAP2H=(65536-freq_reload)>>8;
        RCAP2L=(65536-freq_reload);
        TH2=RCAP2H;
        TL2=RCAP2L;
}
//******************
//******************
unsigned char reset_pin(void)
 {
        buffer[0]='1';
        buffer[1]='2';
        buffer[2]='3';
        buffer[3]='4';
        if(store_pin(buffer))return 0;
        else return 1;
}
//*******************
//*******************
unsigned char reset_loads(void)
{
        return(write(presets_address,0x00));
}
//******************
//******************
void show_loading(void)
{
        lcd_clear();
        lcd("loading...");
```

```
        delay_sec(1);
}
//****************
//****************
void show_pin(void)
{
        lcd_clear();
        lcd("PIN: ");
        lcd_data(read(0x00));
        lcd_data(read(0x01));
        lcd_data(read(0x02));
        lcd_data(read(0x03));
        delay_sec(1);
}
//****************
//****************
unsigned char sys_init(void)
{
        unsigned char z=0;
        IE=0x00;
        tone_off();
        init_irq();
        init_timer();
        EA=1;
        preset=0x00;
        if(!reset)
        {
                z=reset_pin();
```

```
            z|=reset_loads();
        }
        z|=load_presets();
        #ifdef debug
        init_lcd();
        show_loading();
        show_pin();
        #endif
        if(!z)set_presets();
        return z;
}
//********************
//********************
void beep_error(void)
{
        unsigned char z=6;
        error_led=0;
        while(z)
        {
                delay_sec(1);
                error_led=!error_led;
                z--;
        }
        error_led=1;
}
//*******************
//*******************
void tone_on(void)
```

```
{
        TH2=RCAP2H;
        TL2=RCAP2L;
        TF2=0;
        TR2=1;
}
//****************
//****************
void tone_off(void)
{
        TR2=0;
}
//*****************
//*****************
void tf2_isr(void)          interrupt 5
{
        if(TF2)
        {
                freq_dx=!freq_dx;
                TF2=0;
        }
}
//******************
//******************
/*
void tone_on(void)
{
        tone_dx=1;
```

```
}
//***************
//***************
void tone_off(void)
{
        tone_dx=0;
}
*/
//***************
//***************
void send_error(void)
{
        tone_on();
        delay_sec(1.0);
        tone_off();
}
//*********************
//*********************
void send_ok(void)
{
        tone_on();
        delay_sec(0.8);
        tone_off();
        delay_sec(0.5);
        tone_on();
        delay_sec(0.8);
        tone_off();
}
```

```
//********************
//******************
void turn_off(unsigned char c)
{
        switch(c)
        {
                case load_1:    preset&=~(0x01);
                                        break;
                case load_2:    preset&=~(0x02);
                                        break;
                case load_3:    preset&=~(0x04);
                                        break;
                case load_4:    preset&=~(0x08);
                                        break;
                default:        break;
        }
}
//*******************
//*******************
void turn_on(unsigned char c)
{
        switch(c)
        {
                case load_1:    preset|=0x01;
                                        break;
                case load_2:    preset|=0x02;
                                        break;
                case load_3:    preset|=0x04;
```

```
                                                break;
                case load_4:    preset|=0x08;
                                                break;
                default:        break;
        }
}
//*****************
//*****************
unsigned char compare_pin(unsigned char data *ptr)
{


        unsigned char data s[10];
        unsigned char address=pin_address_start;
        unsigned char x;

        for(x=0;x<pin_lenght_size;x++)
        {
                s[x]=read(address);
                if(error)return 0;
                address++;
        }
        for(x=0;x<pin_lenght_size;x++)
        {
                if(s[x]!=*ptr)return 0;
                ptr++;
        }
        return 1;
}
```

```
//******************
//******************
unsigned char store_pin(unsigned char data *ptr)
{
	unsigned char address;
	unsigned char c;

	for(address=pin_address_start;address<pin_lenght_size;address++)
	{
		c=*ptr;
		if(write(address,c))return 0;
		ptr++;
	}
	return 1;
}
//**********************
//**********************
unsigned char  decode_dtmf(unsigned char c)
{

	if(buffer[0]!=cmd_start)return 0;
	if(buffer[5]!=cmd_start)return 0;
	if(!(compare_pin(buffer+pin_start)))return 0;

	switch(c)
	{
		case load_ctrl:
```

```
        if((buffer[6]>max_load)||(buffer[6]<min_load))return 0;

        if((buffer[7]!=load_off)&&(buffer[7]!=load_on))return 0;
                                        if((buffer[7])==load_off)turn_off(buffer[6]);
                                        else turn_on(buffer[6]);


                                        set_presets();
                                        if(write(presets_address,preset))return 0;
                                        else return 1;


            case pin_change:        return(store_pin(buffer+pin2_start));


            default:                return 0;
        }
}




//******************
//******************
void main(void)
{
        unsigned char c;


        if(sys_init())beep_error();
        #ifdef debug
```

```
while(1)
{
        lcd_clear();
        lcd_cmd(0x80);
        while(!tone_in);
        c=get_dtmf();
        if(error)
        {
                lcd_clear();
                lcd("command error!");
                delay_sec(1);
        }
        else
        {
                if(c=decode_dtmf(c)){
                        lcd_clear();
                        lcd("cmd exec ok!");
                        delay_sec(1);
                }
                else{
                        lcd_clear();
                        lcd("cmd exec error!");
                        delay_sec(1);
                        }
        }
}

#endif
```

```
        #ifndef debug
        while(1)
        {
                while(!tone_in);
                c=get_dtmf();
                if(error)send_error();
                else{
                                if(c=decode_dtmf(c))send_ok();
                                else send_error();
                        }
        }
        #endif
}
```