



WOLKITE UNIVERSITY
COLLEGE OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF ELECTRICAL AND COMPUTER
ENGINEERING
COMPUTER ENGINEERING STREAM

Final year thesis on
REMOTE CONTROL HOME AUTOMATION THROUGH
INTERNET

GROUP MEMBERS

- | | |
|--------------------|-------------|
| 1. HABTAMU MULUNEH | ENGR/539/07 |
| 2. SAMUEL GORSHI | ENGR/953/07 |
| 3. MULUGETA ABATE | ENGR/856/07 |

ID NO

ADVISOR

Mr. ABEBE B.

A Final Year Project Submitted to College of Engineering and Technology on Department of Electrical and Computer Engineering -Wolkite University in Partial Fulfillment of the Requirement for Degree of Bachelor of Science in Electrical and Computer Engineering.

Submission date: - 15/10/2011 EC

Wolkite, Ethiopia

Declaration

We, the undersigned, declare that this project is our won original work, has not been presented for a degree in this or any other universities, and all sources of materials used for the project have been fully acknowledged.

This is to certify that the above declaration made by our effort is correct to the best of our knowledge.

In doing so, we assure that we agree with all written above with our signature as follows.

| | |
|--------------------|-------------|
| 1. Habtamu Muluneh | _____x_____ |
| Student Name | Signature |
| 2. Samuel Gorshī | _____x_____ |
| Student Name | Signature |
| 3. Mulugeta Abate | _____x_____ |
| Student Name | Signature |

This Project has been submitted for examination with my approval as a university advisor.

Project Advisor

Signature

ACKNOWLEDGMENTS

First and fore most great thanks to God for his blessing and helps from the beginning to end and We would like to take this opportunity to express our heartfelt gratitude to our advisor **Mr. ABEBE B.**for his guidance, and instruction that he has given us to complete our thesis project in such a short time.

We wish to express our heartfelt thanks to all the people who directly or indirectly supported throughout our project.

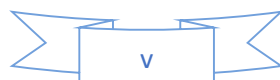
ABSTRACT

Our project basically is an IP based control system, which can be accessed where the internet facility available and it provide you remote control of any device without any physical contact. By doing this project we learned learn a lot of about the working of IP based systems, micro controllers, and many more applications of electronic subjects.

The basic idea is to take advantage of the vast explosion of cell phones and internet access through cell phones and to extend human's reach and possibilities. With Internet Protocol as the backbone of an enterprise-wide network, everything that requires a standalone network today simply becomes subsumed into the IP master. This is interoperability at the very highest level. In the building management sector of this new environment, controllers are designed for the IP network. More importantly, they are also designed to thrive on the IP network.

LIST OF FIGURES

| | |
|---|----|
| Figure 3. 1 sections | 7 |
| Figure 3. 2 Power Supply..... | 10 |
| Figure 3. 3 pin configurations of 9 pin serial port | 11 |
| Figure 3. 4 MAX 232 top view | 14 |
| Figure 3. 5 Top View of PIC 16F877A | 16 |
| Figure 4. 1 Block Diagram of Project..... | 17 |
| Figure 4. 2 Server GUI..... | 18 |
| Figure 4. 3 Flow Chart of Server Application | 19 |
| Figure 4. 4 Client GUI | 20 |
| Figure 4. 5 Flow Chart of Client Application | 21 |
| Figure 4. 6 Error in Connectivity | 22 |
| Figure 4. 7 Socket programming | 25 |
| Figure 4. 8 Schematics of Hardware..... | 27 |
| Figure 5. 0 Simulation Result of project | 29 |



LIST OF TABLES

| | |
|--|----|
| Table 3. 1 shows components used in Hardware | 9 |
| Table 3. 2 Pin description of 9 pin serial port..... | 12 |
| Table 4. 1 Protocol Layer..... | 22 |

ABBREVIATIONS

| | |
|--------------|--|
| ASP.Net----- | -----Active service page |
| API----- | -----Application Programming Interface |
| DPDT----- | -----double pole double through |
| EMF----- | -----Electromotive Force |
| SPDT----- | -----Single pole double through |
| TCP----- | -----Transmission Control Protocol |
| XML----- | -----Extensible Markup Language |
| UDP--- | -----User Datagram Protocol |
| GUI----- | -----graphical user interface |

Table of Contents

| | |
|--|-----|
| Declaration..... | ii |
| ACKNOWLEDGMENTS | iii |
| ABSTRACT..... | iv |
| LIST OF FIGURES | v |
| LIST OF TABLES..... | vi |
| ABBREVIATIONS..... | vii |
| CHAPTER ONE..... | 1 |
| INTRODUCTION | 1 |
| 1.1 Background | 1 |
| 1.2 Statement of Problem..... | 2 |
| 1.3 Proposed System | 2 |
| 1.3 Objectives..... | 3 |
| 1.3.1 General Objectives | 3 |
| 1.3.2 Specific Objectives | 3 |
| 1.4 Methodology | 3 |
| 1.5 Scope of the project..... | 3 |
| 1.6 Significance of the project..... | 4 |
| CHAPTER TWO | 5 |
| Literature Review..... | 5 |
| CHAPTER THREE | 7 |
| TOOLS AND COMPONENTS USED..... | 7 |
| 3.1 Tools..... | 7 |
| 3.1.1 Visual Studio | 7 |
| 3.1.2 PCHWD IDE Compiler for Microchip PIC | 8 |
| 3.2 Components Used in Hardware..... | 9 |
| 3.2.1 Power Supply..... | 10 |
| 3.2.2 Serial Port | 11 |
| 3.2.3 MAX232- Level Converter | 13 |
| 3.2.4 Switches..... | 14 |
| 3.2.5 Microcontroller PIC..... | 15 |
| CHAPTER FOUR..... | 17 |
| SYSTEM DESIGN AND ANALYSIS..... | 17 |

| | |
|--|----|
| 4.1 Sections | 17 |
| 4.2 Windows Application..... | 18 |
| 4.2.1 Server Application..... | 18 |
| 4.2.2 Client Application..... | 20 |
| 4.3 Structure of TCP/IP..... | 22 |
| 4.3.1 Main Features of TCP/IP..... | 23 |
| 4.4 Socket Programming..... | 25 |
| 4.4.1 C# Server Socket Program..... | 25 |
| 4.4.2 C# Client Socket Program..... | 26 |
| 4.5 Hardware..... | 26 |
| 4.5.1 Serial Programming in PIC..... | 26 |
| 4.5.2 Hardware Description..... | 27 |
| 4.5 Simulation Result..... | 29 |
| CHAPTER FIVE..... | 30 |
| CONCLUSION AND RECOMMENDATION..... | 30 |
| 5.1 CONCLUSION..... | 30 |
| 5.2 RECOMMENDATION..... | 30 |
| References..... | 31 |
| APPENDIX-A..... | 33 |
| A-1 Source Code of Server Application..... | 33 |
| A-2 Source Code of Client Application..... | 35 |
| A-3 Code of Microcontroller..... | 36 |

CHAPTER ONE

INTRODUCTION

1.1 Background

Home automation system is a project used to control any devices in home or in office or in other places can switch activate or deactivate. The goal of this project is to control a large number of electronic devices easily and remotely. Our system is a prototype designed only to activate / deactivate electrical appliances at home and to get the current status of the device. But the utilization of this type of remote control is very vast. The primary idea is to exploit the vast explosion of cellular or mobile phones and other portable and mobile internet devices. This project provides a lot of learning edge about the working of IP based systems, microcontrollers, and utilization of electronic components. The scope and users of this internet remote control system will grow with the expansion of the number of internet consumers around the world. Our system remote control home automation through internet connects the user to devices at home. The user can be anywhere in the world, and can control the home appliances through internet at a remote place.

The user who wants to access his/her home from a remote place, he/she should have to install the client software and server software will be installed at home's pc. User will access his / her home through this client software. We developed the Client and Server application on Visual Studio, both Client and Server.

Applications are user friendly, anyone having a little knowledge of computer can easily use it. A Client and Server application will enable a user (client) to send a request to the Server and in response server will activate / deactivate any device.

We have assigned an IP address to our server, so that it can be accessed by using an internet cloud. Server is connected locally with the microcontroller (PIC) which acts as a decoder and eventually control the home devices through electrically operated switches.

1.2 Statement of Problem

Many people are always on the move from place to place due to business demands. Some people can spend a couple of days away from their home leaving all their household appliances without any kind of monitoring and control. Some devices are left plugged into power sockets whereas others are supposed to be plugged into and out of power sockets at different intervals depending on the time of the day.

All this requires an individual to manually attend to each of the devices independently from time to time. All such monitoring and control can be done without necessarily being around or inside the home. Some devices if not controlled properly consume a lot of energy which leads to extra expenditure on electricity. Therefore, we propose to design an internet-based home automation system which will enable one to remotely manage his/her appliances from anywhere, anytime.

1.3 Proposed System

The system we have proposed is an extended approach to automating a control system. With the advancement and breakthroughs in technology over the years, the lives of people have become more complicated and thus, they have become busier than before. With the adoption of our system, we can gain control over certain things that required constant attention.

The application of our system comes in handy when people who forget to do simple things such as turn activate or deactivate devices at their home or in their office, they can now do so without their presence by internet access on mobile phone.

This development, we believe, will ultimately save a lot of time especially when people far from and don't have to come back for simple things such as to turn activate/deactivate switches at their home or at their office once they set out for their respective work. Possible target appliances include (but are not limited to) climate control system, security systems, lights; anything with an electrical interface.

The proposed approach for designing this system is to implement a microcontroller-based control module that receives its instructions and command from a cellular phone over internet. The microcontroller then will carry out the issued commands and then communicate the status of a given appliance or device back to the cellular phone.

1.3 Objectives

1.3.1 General Objectives

To design an internet-based home automation system for controlling home appliances

1.3.2 Specific Objectives

The specific objectives of this project are:

- To know how client and server applications are created.
- To understand how microcontroller (PIC) is interface with server.
- Knowing the relationship between server socket and client socket programming.
- To co-ordinate appliances and other devices through internet

1.4 Methodology

Methods involved during the whole project time are the following:

- **Literature review:** referring the books, websites and other reference materials including datasheet of components to understand how the system can work and know about each component.
- **Design specifications:** listing the required performance of the system
- **System modeling:** formulating the relationship between different parts and components of the system.
- **Designing:** the circuit design of the whole system using proteus software.
- **Simulation:** simulating the designed circuit

1.5 Scope of the project

Our area of interest is remote control of home appliances using the internet. In this project we have cover theoretical analysis, operation and software simulation on the remote control of a fan, a light and a television set (TV).

1.6 Significance of the project

The significances of this project are: -

To eliminate the need of being physically present in any location for tasks involving the operation of appliances within a household/office

User Convenience

The basic advantage is user convenience. It becomes very easy to perform all activities. Even if you are not present at home you can access all your home system remotely. It allows one to keep a track of all the activities taking place at his/her place even in their absence.

Maintaining and Saving energy

This project helps you to manage energy consumption at your place and also provides the benefit of saving energy. For example, it will automate your thermostat to adjust settings throughout the day based on the timing depending on whether someone is at home or not. Some smart devices can be synced up to your appliances with real-time energy information. This helps your home appliances know the most cost-effective voltage operate.

Enhanced Security

Security is always an important factor which should always be taken in to consideration.

Environment and Economical contribution

Home automation system allow you to ensure that you are only using the energy and resources that are necessary while you are home, and you are sustaining resources. Home Automation Systems provide convenience and saves your time and effort performing home activities. When you properly manage your energy, you can reduce your energy consumption, which may help you save money.

Parental Control

Parents can have complete control over their children's.

CHAPTER TWO

Literature Review

We see some project similar with our topics from those good proposed study is from Home Automation System based on GSM. To enable its use in several applications, this design integrates the device to be controlled, the microcontroller, and the GSM module. This paper implements a complete M2M (Machine to machine or man to machine system) over a GSM network. [1]

A low-cost Java-Based Home Automation system based on PC-based home server. Various devices are connected to the input/output ports of the microcontroller and their status is send to the server whether they are on or off. The monitoring and control software engine are based on the combination of Java Server Pages, JavaBeans, and Interactive C. The system is scalable and that is any number of devices can be added with no major changes to its design. Password protection is used to stop unauthorized users from accessing the appliances at home. If the Internet connection or the server is not working, the embedded system board still can control and operate the appliances locally. [2]

Home Automation System Using Arduino employs an embedded micro – web server in Arduino Mega 2560 microcontroller, with IP connectivity for accessing and controlling devices and appliances remotely. These devices can control through a web application or via Bluetooth Android based Smart phone app. [3]

Bluetooth based Home Automation System using Cell phone. This system is a low-cost secure system in which the communication between mobile and home appliances is wireless. Appliances of home connected to the Arduino board. The users expected to acquire password for the Arduino BT and the cell phone to access the home appliances. This adds a protection from unauthorized users. [4]

Home automation control Remote Control System using DTMF (Dual-Tone Multi-Frequency) tones as the control signals. The DTMF remote control unit designed for interface with a standard cellular mobile telephone handset. It plugs into the hands-free adaptor socket on the telephone handset and operates by receiving DTMF tone from the phone audio output. [5]

Home automation systems as multiple agent systems (MAS). In the paper home automation system has been proposed that includes home appliances and devices that are controlled and maintained for home management. The major tasks to improve performance. [6]

Design and implementation of SMS based control for monitoring systems. The paper has three modules involving sensing unit for monitoring the complex applications. A processing unit, that is microcontroller and a communication module that uses GPRS modem or cell phone via serial port RS-232. The SMS is used for status reporting such as power failure. The problems with the implementation of home automation systems. Furthermore, the possible solutions are devised through various network technologies. Several issues affecting home automation systems such as lack of robustness, compatibility issue and acceptability among the old and disabled people are discussed. remote monitoring through mobile phone involving the use of spoken commands.

The spoken commands are generated and sent in the form of text SMS to the control system and then the microcontroller on the basis of SMS takes a decision of a particular task. primary health-care management for the rural population. A solution proposes the use of the mobile web-technologies providing the PHC services to the rural population. The system involves the use of SMS and cell phone technology for information management, transactional exchange and personal communication. The use of speech to interact remotely with the home appliances to perform a particular action on behalf of the user.

The approach is inclined for people with disability to perform real-life operations at home by directing appliances through speech. Voice separation strategy is selected to take appropriate decision by speech recognition. The development of an Internet-based system to allow monitoring of important process variables from a distributed control system (DCS). This paper proposes hardware and software design considerations which enable the user to access the process variables on the DCS, remotely and effectively, using only a commonly available web browser. In their paper, Liang, Fu, and Wu (2002) propose a software architecture that makes home automation (HA) system based on multi-agent concept. The architecture is composed of five main components respectively named space agents, function agents, personal preference agents, environment variables server, and resource access right control kernel. There is need to construct a more powerful HA both in the hardware and software aspects

CHAPTER THREE

TOOLS AND COMPONENTS USED

This chapter will give description of the tools used to develop software applications and hardware codes, and major components used in Hardware.

Remote control Home Automation Through Internet



Figure 3. 1 Figure 3.1 sections

3.1 Tools

Software tools used in the project are:

- Visual Studio 2008
- PCWHD IDE Compiler for Microchip PIC
- Proteus

3.1.1 Visual Studio

Visual Studio .NET is Microsoft's integrated development environment (IDE) for creating, documenting, running and debugging programs written in a variety of .NET programming languages. Visual Studio is a complete set of development tools for building ASP.NET Web applications, XML Web Services, desktop applications, and mobile applications.

Visual Basic, Visual C#, and Visual C++ all use the same integrated development environment (IDE), which enables tool sharing and eases the creation of mixed-language solutions. It is used

to develop console and graphical user interface applications along with Windows Forms applications, web sites, web applications, and web services in both native codes together with managed code for all platforms supported by Microsoft Windows, Windows Mobile, Windows CE, .NET Framework and Microsoft Silverlight.

3.1.1.1 features of visual studio

a. Code editor

Visual Studio, like any other IDE, includes a code editor that supports syntax highlighting and code completion using IntelliSense for not only variables, functions and methods but also language constructs like loops and queries. The code editor is used for all supported languages.

b. Debugger

Visual Studio includes a debugger that works both as a source-level debugger and as a machine-level debugger. It works with both managed code as well as native code and can be used for debugging applications written in any language supported by Visual Studio. In addition, it can also attach to running processes and monitor and debug those processes. If source code for the running process is available, it displays the code as it is being run.

c. Designer

Visual Studio includes a host of visual designers to aid in the development of applications.

These tools include:

- Windows Forms Designer
- WPF Designer
- Web designer/development
- Class designer
- Data designer
- Mapping designer

3.1.2 PCHWD IDE Compiler for Microchip PIC

CCS provides a complete, integrated tool suite for developing and debugging embedded applications running on Microchip PIC®MCUs and dsPIC® DSCs. This suite includes an IDE for project management, a context sensitive C aware editor, build tools and real time debugger...helping developers create, analyze, debug and document project code. The CCS

IDE allows developers to manage every aspect of their embedded software development, from code creation through device programming. External programs can be invoked from the IDE, simplifying integration with other programmers and debuggers.

The PCWHD compiler supports the Microchip PIC® PIC10, PIC12, PIC16, PIC18, PIC24 MCU families and dsPIC® DSCs. PCWHD Compiler includes: Standard C pre-processors, operators and statements, interrupt handlers, discrete I/O, multiple compilation units, constants in ROM, float, ICSP and in-circuit debugging capabilities. For more detailed information on the features and functions of the PCWHD compiler.

PCD is a C Compiler for Microchip's 24bit opcode family of microcontrollers, which include the dsPIC30, dsPIC33 and PIC24 families. The compiler is specifically designed to meet the unique needs of the dsPIC® microcontroller. This allows developers to quickly design applications software in a more readable, high-level language. The compiler can efficiently implement normal C constructs, input/output operations, and bit twiddling operations. All normal C data types are supported along with special built in functions to perform common functions in the MPU with ease.

3.2 Components Used in Hardware

Table 3. 1 shows components used in Hardware

| | |
|---|------------------------|
| 1 | Power Supply |
| 2 | Serial Cable |
| 3 | LM7805 |
| 4 | MAX232S |
| 5 | Switches |
| 6 | Micro-controller (PIC) |
| 7 | Oscillator 4 MHz |

| | |
|----|-----------|
| | |
| 8 | LEDs |
| 9 | Capacitor |
| 10 | Resistors |

3.2.1 Power Supply

To meet the power requirements of microcontroller and other component used in circuit a 5V regulated power is supplied using LM7805 IC, as shown in Figure 2.3. The capacitors are used for filtration

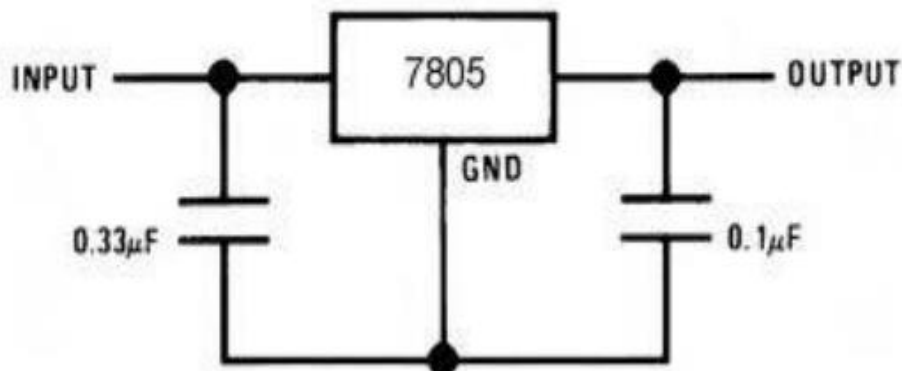


Figure 3. 2 Power Supply

The output voltage and current is +5V and 100mA respectively, the current can be increased to 1A by providing proper cooling to LM7805 IC. The capacitors used at input and output terminal should have voltage rating compatible with the input and output voltage. If a voltage level of more than +5V is required, the circuit can be modified by replacing the 7805 chips with another regulator with different output voltage from regulator 78xx chip family. The last numbers in the chip code tells the output voltage. To keep the regulator working properly the input voltage level must be +3V higher than the required output voltage.

3.2.2 Serial Port

The serial port is an I/O (Input/output) device. An I/O device is just a way to get data into and out of a computer. There are many types of I/O devices such as serial ports, parallel ports, disk drive controllers, Ethernet boards, universal serial buses, etc. Most PC's have one or two serial ports.

Each has a 9-pin connector (sometimes 25-pin) on the back of the computer.

Computer programs can send data (bytes) to the transmit pin (output) and receive bytes from the receive pin (input). The other pins are for control purposes and ground. Serial ports, also called communication (COM) ports, are bi-directional. Bi-directional communication allows each device to receive data as well as transmit it. Serial devices use different pins to receive and transmit data -- using the same pins would limit communication to half-duplex, meaning that information could only travel in one direction at a time. Using different pins allows for full-duplex communication, in which information can travel in both directions at once.

3.2.2.1 pin configuration of serial port

Figure 2.6 and Table 2.1 shows pin configuration of serial port

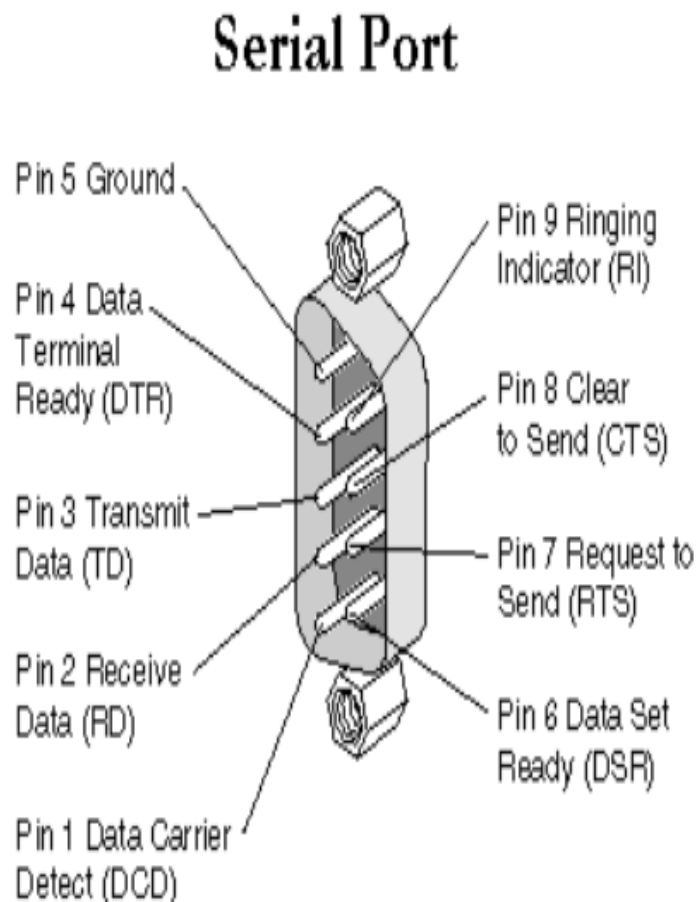


Figure 3.3 pin configurations of 9 pin serial port

Table 3. 2 Pin description of 9 pin serial port

| Pin# | Pin Name | Pin Description |
|------|---------------------|---|
| 1 | Carrier detect | It is used by modem to inform pc that it has detected carrier on phone line |
| 2 | Receive Data | Serial data is received on this line by pc |
| 3 | Transmit Data | Serial data is transmitted on this pin by pc |
| 4 | Data terminal ready | Terminal (computer) power up it assert DIR high |
| 5 | Ground | Pin is grounded |
| 6 | Data set ready | When modem powers up asserts DSR high |
| 7 | Request To send | Request to send is sent from (DTE) terminal (pc) to modem (DCE) to inform it that pc wants to send some data to modem |
| 8 | Clear to send | Up on received RTS from (pc) the modem (DCE) asserts CTS high whenever it is ready to receive data |
| 9 | Clear | It is set by modem to indicate the pc that a ringing signal has been detected on line. |

3.2.2.2 Flow Control

Flow control means the ability to slow down the flow of bytes in a wire. An important aspect of serial communications is the concept of **flow control**. This is the ability of one device to tell another device to stop sending data for a while. The commands Request to Send (RTS), Clear to Send (CTS), Data Terminal Ready (DTR) and Data Set Ready (DSR) are used to enable flow control.

3.2.2.3 Baud Rate

What baud really refers to is modulation rate or the number of times per second that a line changes state. This is not always the same as bits per second (BPS). If you connect two serial devices together using direct cables then baud and BPS are in fact the same.

Serial communication using RS-232 requires that is specified by four parameters: the **baud rate** of the transmission, the number of **data bits** encoding a character, the sense of the optional **parity bit**, and the number of **stop bits**.

In our project we have used 9600 as the baud rate since the mobile set we are using communicates at this baud rate. We have introduced 8 bits as the data and one bit as the stop. Now the total time for one bit is $(1/9600)/(8+1) = 0.1157\mu\text{s}$. [3]

3.2.3 MAX232- Level Converter

The MAX232 is an integrated circuit that converts signals from an RS-232 serial port to signals suitable for use in TTL compatible digital logic circuits. The MAX232 is a dual driver/receiver and typically converts the RX, TX, CTS and RTS signals.

The drivers provide RS-232 voltage level outputs (approx. ± 7.5 V) from a single + 5 V supply via on-chip charge pumps and external capacitors. This makes it useful for implementing RS-232 in devices that otherwise do not need any voltages outside the 0 V to + 5 V range, as power supply design does not need to be made more complicated just for driving the RS-232 in this case.

The receivers reduce RS-232 inputs (which may be as high as ± 25 V), to standard 5 V TTL levels. These receivers have a typical threshold of 1.3 V, and a typical hysteresis of 0.5 V. It is helpful to understand what occurs to the voltage levels. When a MAX232 IC receives a TTL level to convert, it changes a TTL Logic 0 to between +3 and +15 V, and changes TTL Logic 1 to between -3 to -15 V, and vice versa for converting from RS232 to TTL. This can be confusing when you realize that the RS232 Data Transmission voltages at a certain logic state are opposite from the RS232 Control Line voltages at the same logic state.

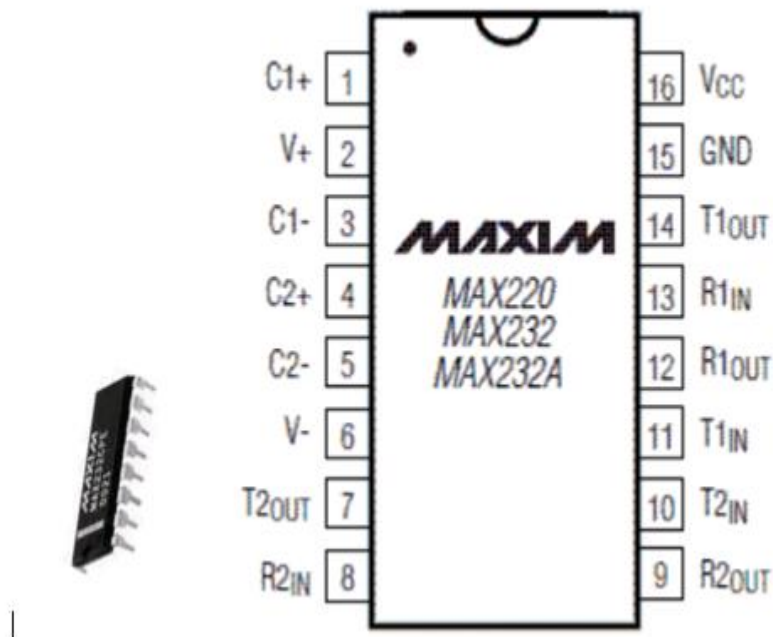


Figure 3. 4 MAX 232 top view

It is helpful to understand what occurs to the voltage levels. When a MAX232 IC receives a TTL level to convert, it changes a TTL Logic 0 to between +3 and +15 V, and changes TTL Logic 1 to between -3 to -15 V, and vice versa for converting from RS232 to TTL. This can be confusing when you realize that the RS232 Data Transmission voltages at a certain logic state are opposite from the RS232 Control Line voltages at the same logic state.

3.2.4 Switches

To turn on/off an electrical device upon the signal of micro controller an electrically operated switch is necessary. Only one or combination of two or more switches can be used according to the voltage and current requirements of the device.

Some of them which we used are

- Relay
- Transistor

3.2.4.1 Relay

A relay is an electrically operated switch. Current flowing through the coil of the relay

creates a magnetic field which attracts a lever and changes the switch contacts. The coil current can be on or off so relays have two switch positions and most have double throw (changeover) switch contacts. Relays allow one circuit to switch a second circuit which can be completely separate from the first. The coil of a relay passes a relatively large current, typically 30mA for a 12V relay, but it can be as much as 100mA for relays designed to operate from lower voltages. Most ICs (chips) cannot provide this current and a transistor is usually used to amplify the small IC current to the larger value required for the relay coil. the relay's switch connections are usually labeled COM, NC and NO:

- **COM** = Common, always connect to this; it is the moving part of the switch.
- **NC** = Normally Closed, COM is connected to this when the relay coil is off.
- **NO** = Normally Open, COM is connected to this when the relay coil is on.

3.2.4.2 Transistors

Like relays, transistors can be used as an electrically operated switch. For switching small DC currents (< 1A) at low voltage they are usually a better choice than a relay. The essential usefulness of a transistor comes from its ability to use a small signal applied between one pair of its terminals to control a much larger signal at another pair of terminals. This property is called gain. A transistor can control its output in proportion to the input signal; that is, it can act as an amplifier. Alternatively, the transistor can be used to turn current on or off in a circuit as an electrically controlled switch, where the amount of current is determined by other circuit elements.

3.2.5 Microcontroller PIC

Microcontroller is used as Character Decoder. When a key from remote computer keyboard is depressed, a code is generated and transmitted to one PC to another. The PIC microcontroller is used as a decoder which enables or disables one of the 6 output lines

which are used to control external devices. We have used PIC16F877A microcontroller in our project. Its serial communication programming is easy to use PICs are popular with both industrial developers and hobbyists alike due to their low cost, wide availability, large user base, extensive collection of application notes, availability of low cost or free development tools, and serial programming (and reprogramming with flash memory) capability.

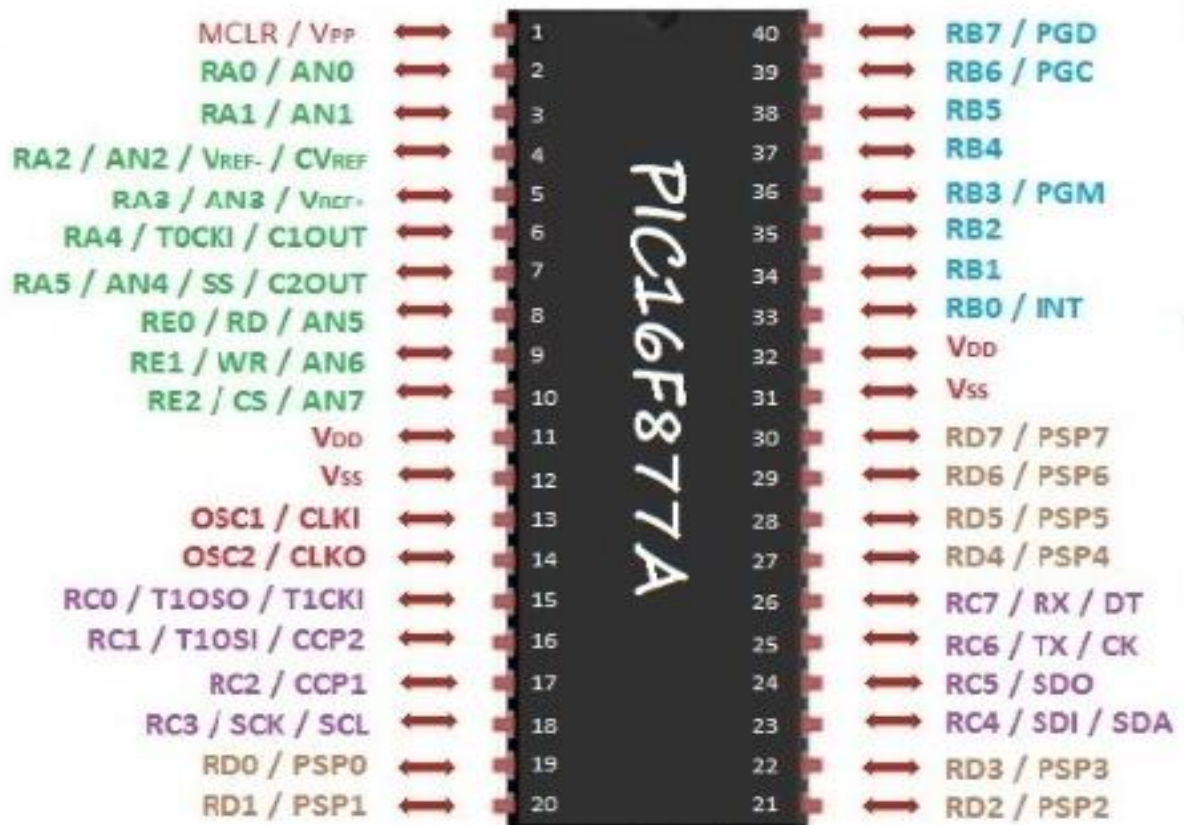


Figure 3.5 Top View of PIC 16F877A

CHAPTER FOUR

SYSTEM DESIGN AND ANALYSIS

As the previous chapter gave details of all the essential components used, this chapter will give details of two parts of our project i.e. hardware and software's working

4.1 Sections

Our project has two main sections:

- Software
- Hardware

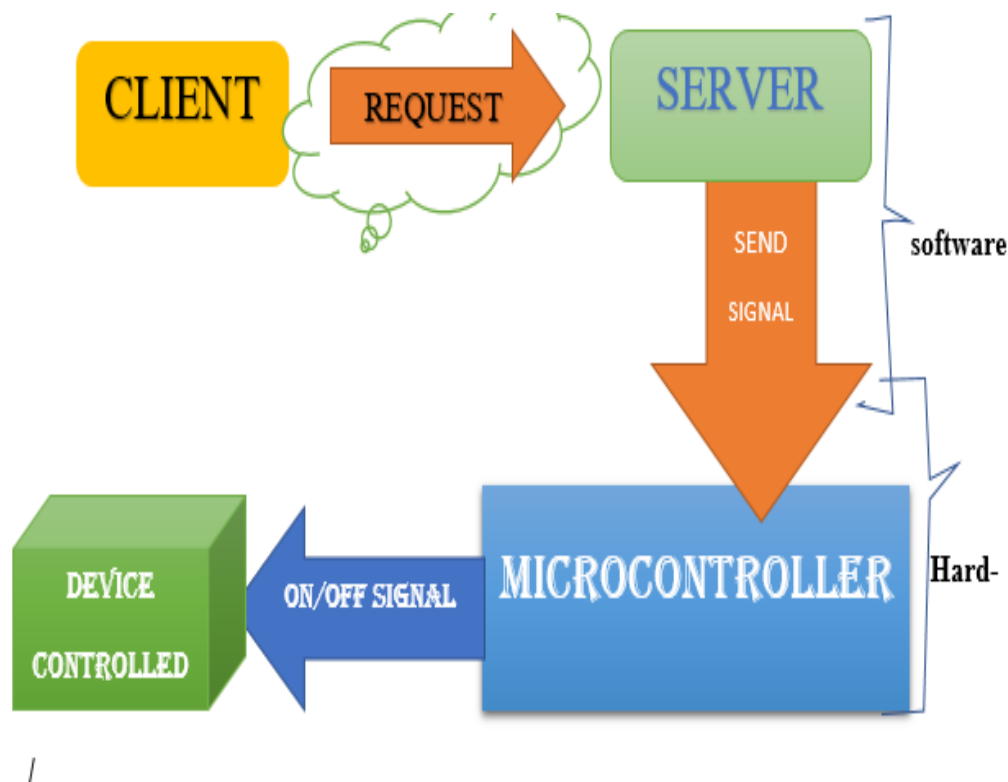


Figure 4. 1 Block Diagram of Project

As shown in figure 4.1 first of all server client will run application on their system. Client will send request to server, in response server will generate a corresponding signal which is sent to microcontroller, in response to that signal microcontroller will on /off the requested device

4.2 Windows Application

Now we will discuss the application which is developed in visual studio 2008. The application consists of two parts

- Server
- Client

Both of the them are basically windows application and developed in C sharp language. Firstly, we will discuss the SERVER application in detail.

4.2.1 Server Application

The graphical user interface is shown in the following Figure 4.2. Then we will run the application of the server the following window will appear. Then ever we want to establish the connection with the client, server should be in the ON state.

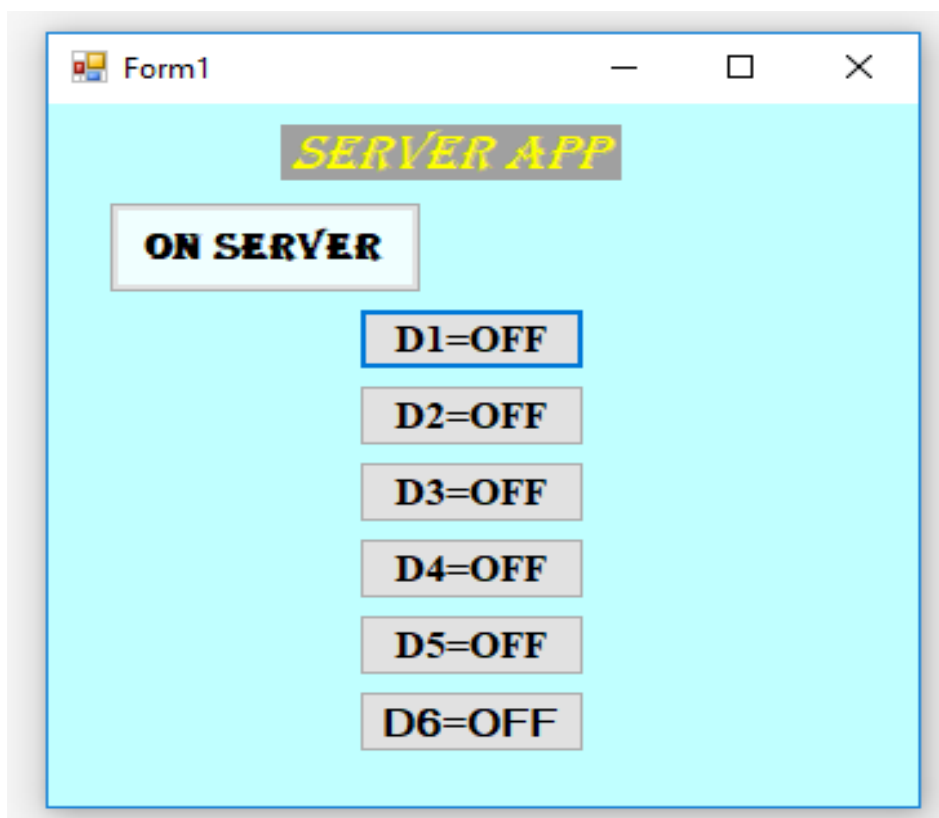


Figure 4. 2 Server GUI

Once we ON the server, it will generate a message “waiting for client”. We have basically controlled six devices in our application as shown in figure 4.2. Server application only

shows the status of each device. This status will be set on the bases of signal sent by the client to server application. The flowchart for SERVER application code is as follows:

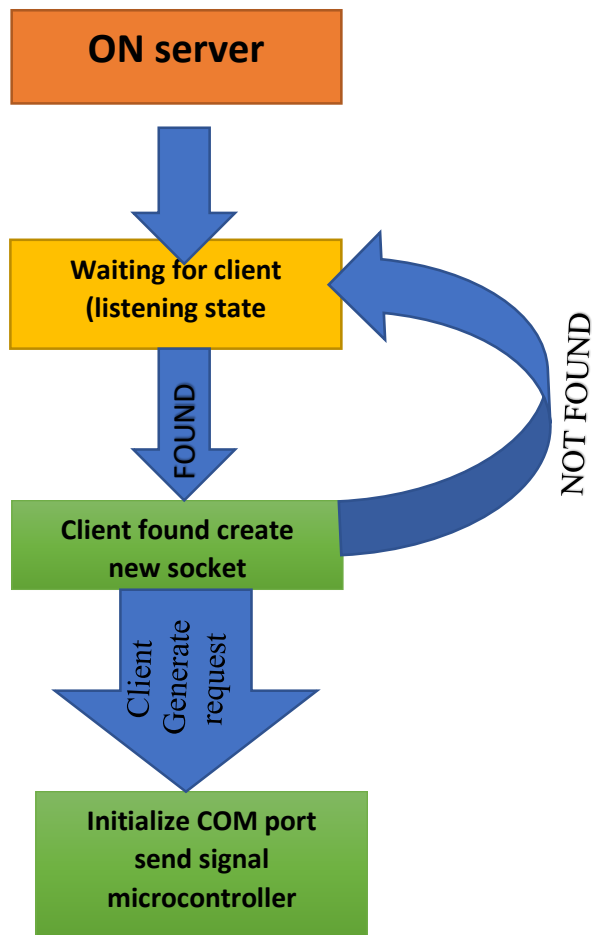


Figure 4. 3 Flow Chart of Server Application

Once the client is found, server will generate a message that “connected to 192.168.0.1, 1092”. Here the 192.168.0.1 (can be any number) is internet protocol and 1092 (can be any number) is port no. Now connection is established between server and client. Now whenever client sends a request to on/off a device, a new socket is created, server will then initialize the serial port “COM6” (for example), and server will send signal to microcontroller which will on/off the requested device.

4.2.2 Client Application

When we will run the application of the Client the following Figure 4.4 GUI will be seen.

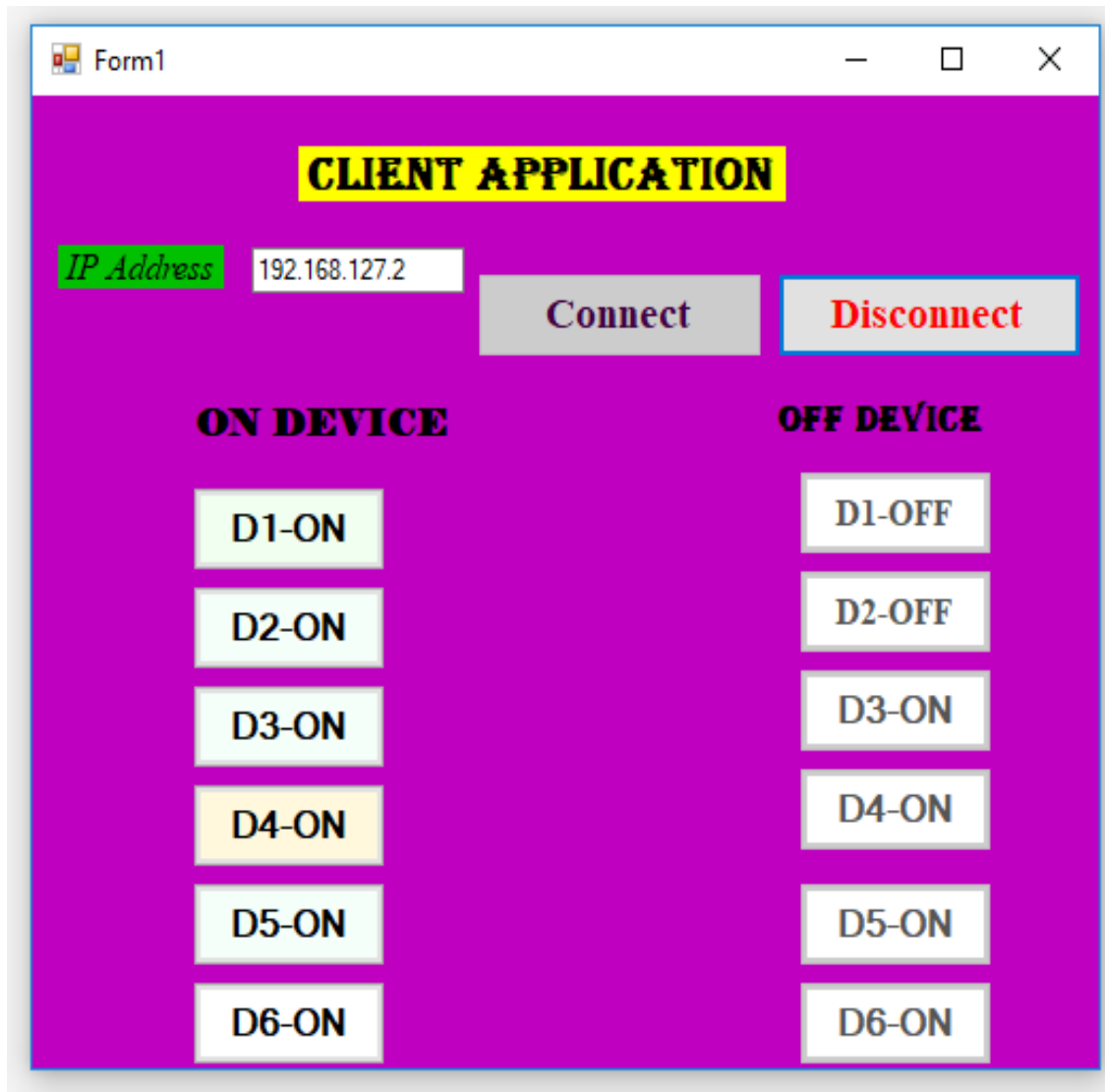


Figure 4.4 Client GUI

The client application code is illustrated in the following flowchart Figure 4.5

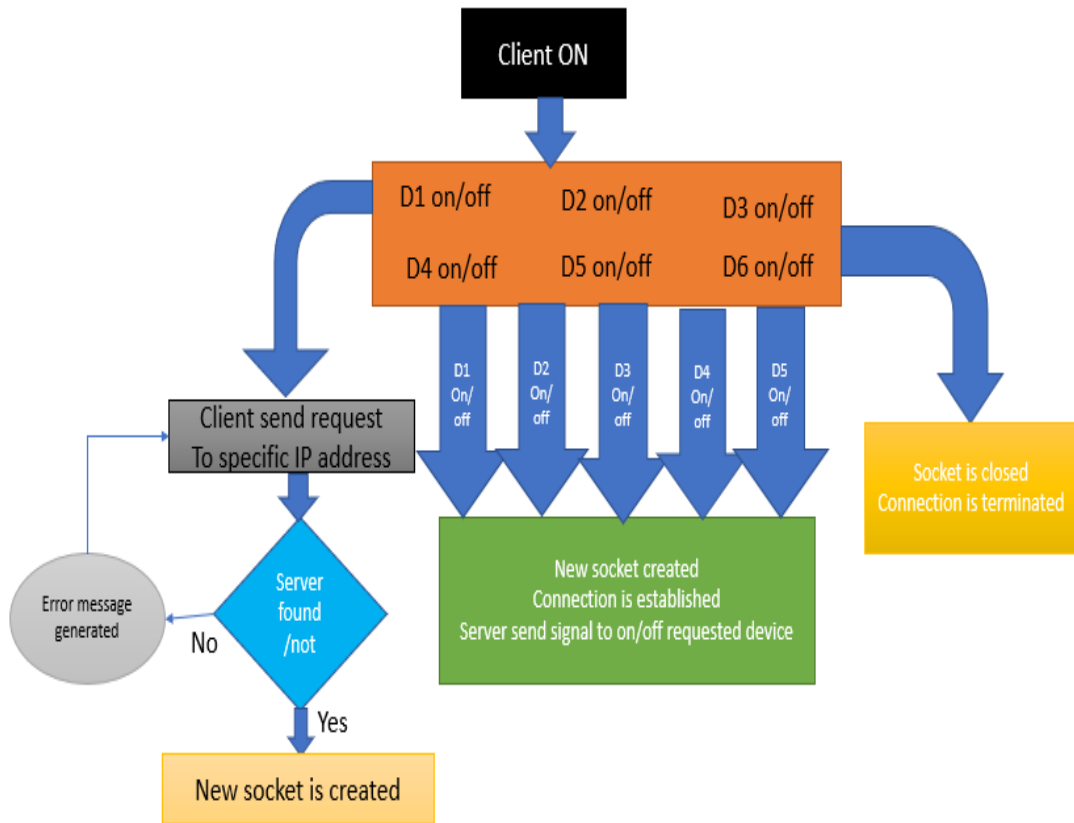


Figure 4.5 Flow Chart of Client Application

First of all, client will specify the IP address. And pushes the button CONNECT. If the server is in “ON” state and the requested IP address is correct, Connection will be established and the following message appear Connected to 192.168.0.1, 1011. Now the client has multiple options, like on/off any device or disconnect. If the client makes a request to ON/OFF the device, a new socket is created, connection is established and a specific signal is sent to the server.

Server will respond to that signal in the same i.e. it will initialize the COM port and send request to microcontroller. If client send the signal to “Disconnect “, connection will be terminated, socket is closed and hence no more data can be sent to the server. If client send the signal to “Disconnect “, connection will be terminated, socket is closed and hence no more data can be sent to the server. At any stage, if the server is turned off by the user or any other fault occur, then any request to server by the client will generate the error message as shown in Figure 4.6

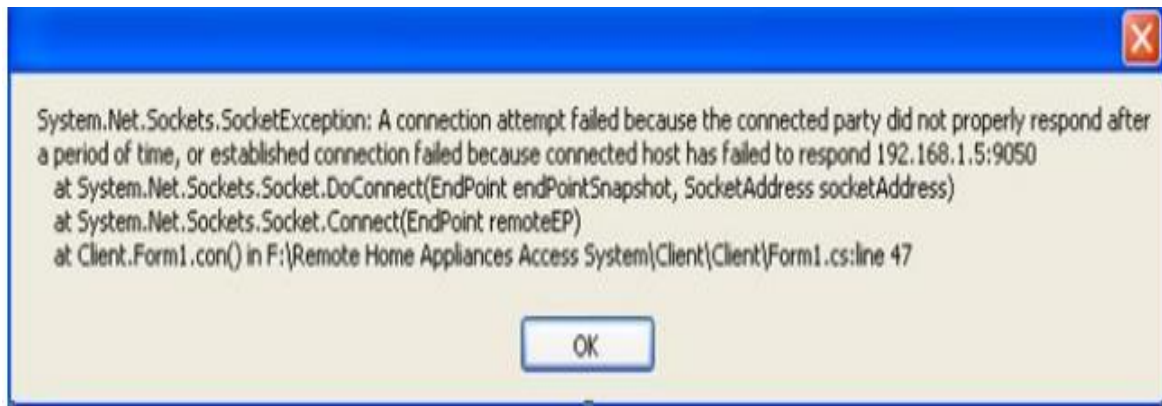


Figure 4. 6 Error in Connectivity

To establish a connection between Server and Client application we need to use a protocol. We used Transmission Control Protocol (TCP) is as a highly reliable host-to host protocol between hosts in packet-switched computer communication networks, and in interconnected systems of such networks. The functions to be performed by the Transmission Control Protocol, the program that implements it, and its interface to programs or users that require its services.

4.3 Structure of TCP/IP

TCP is based on concepts first described by Cerf and Kahn in. The TCP fits into a layered protocol architecture just above a basic Internet Protocol which provides a way for the TCP to send and receive variable-length segments of information enclosed in internet datagram "envelopes". The internet datagram provides a means for addressing source and destination TCPs in different networks. The internet protocol also deals with any fragmentation or reassembly of the TCP segments required to achieve transport and delivery through multiple networks and interconnecting gateways. The internet protocols also carry information on the precedence, security classification of the TCP segments, so this information can be communicated end-to-end across multiple networks.

Table 4. 1 Protocol Layer

| |
|-------------------|
| higher-level |
| TCP |
| internet protocol |
| internet protocol |

Some computer systems will be connected to networks via front-end computers which house the TCP and internet protocol layers, as well as network specific software. The TCP specification describes an interface to the higher-level protocols which appears to be implementable even for the front-end case, as long as a suitable host-to-front end protocol is implemented.

4.3.1 Main Features of TCP/IP

The following are main features of TCP/IP

4.3.1.1 Scope

The TCP is intended to provide a reliable process-to-process communication service in a multi network environment. The TCP is intended to be a host-to-host protocol in common use in multiple networks.

4.3.1.2 Interfaces

The TCP interfaces on one side to user or application processes and on the other side to a lower level protocol such as Internet Protocol. The interface between an application process and the TCP is illustrated in reasonable detail. This interface consists of a set of calls much like the calls an operating system provides to an application process for manipulating files. For example, there are calls to open and close connections and to send and receive data on established connections. It is also expected that the TCP can asynchronously communicate with application programs. Although considerable freedom is permitted to TCP implementers to design interfaces which are appropriate to a particular operating system environment, a minimum functionality is required at the TCP/user interface for any valid implementation. The interface between TCP and lower level protocol is essentially unspecified except that it is assumed there is a mechanism whereby the two levels can asynchronously pass information to each other. Typically, one expects the lower level protocol to specify this interface. TCP is designed to work in a very general environment of interconnected networks. The lower level protocol which is assumed throughout this document is the Internet Protocol.

4.3.1.3 Motivation

Computer communication systems are playing an increasingly important role in military, government, and civilian environments. This document focuses its attention primarily on military computer communication requirements, especially robustness in the presence of communication unreliability and availability in the presence of congestion, but many of these problems are found in the civilian and government sector as well.

4.3.1.4 Connection Oriented

TCP is a connection-oriented, end-to-end reliable protocol designed to fit into a layered hierarchy of protocols which support multi-network applications. The TCP provides for reliable inter-process communication between pairs of processes in host computers attached to distinct but interconnected computer communication networks. Very few assumptions are made as to the reliability of the communication protocols below the TCP layer. TCP assumes it can obtain a simple, potentially unreliable datagram service from the lower level protocols. In principle, the TCP should be able to operate above a wide spectrum of communication systems ranging from hard-wired connections to packet switched or circuit-switched networks.

4.3.1.4 Operation

As noted above, the primary purpose of the TCP is to provide reliable, securable logical circuit or connection service between pairs of processes. To provide this service on top of a less reliable internet communication system requires facilities in the following areas:

- Basic Data Transfer
- Reliability
- Flow Control
- Multiplexing
- Connections
- Precedence and Security

4.4 Socket Programming

C# simplifies the network programming through its namespaces like System.Net and System.Net.Sockets. A Socket is an End-Point off To and from (Bidirectional) communication link between two programs (Server Program and Client Program) running on the same network. We need two programs for communicating a socket application in C#. A Server Socket Program (Server) and a Client Socket Program (Client). C# Server Socket Program: A C# Server Socket Program running on a computer has a socket that bound to a Port Number on the same computer and listening to the client's incoming requests.

4.4.1 C# Server Socket Program

A C# Client Socket Program have to know the IP Address (Hostname) of the computer that the C# Server Socket Program resides and the Port Number assign for listening for client's request. Once the connection is established between Server and Client, they can communicate (read or write) through their own sockets.

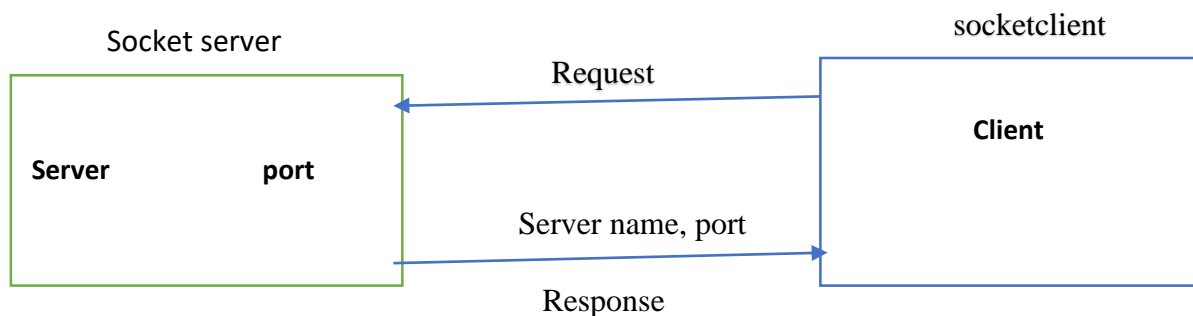


Figure 4. 7 Socket programming

The Server Socket Program act as a Server and listening to client's request. Here we assign a Port No. 9050 for the Server Socket, these are the commands which will create a port and server will be now in listening state;

```
IPEndPoint ipep = new IPEndPoint (IPAddress.Any, 9050);  
newsock = new Socket (AddressFamily.InterNetwork,  
SocketType.Stream, ProtocolType.Tcp);  
newsock. Bind(ipep);  
newsock. Listen (9050);
```

The next step is to create an infinite loop for monitoring the request from Client's side. When the Server Socket accept a request from the Client side, it reads the data from Network Stream and also it writes the response to **Network Stream**.

4.4.2 C# Client Socket Program

The C# Client Socket Program is the second part of the C# Server Socket Program. The C# Client Socket Program is a Windows based application. The Client is connected to the Port 9050 of the C# Server Socket Program, and the IP Address (Computer Name).

```
IPEndPoint ipep = new IPEndPoint (IPAddress.Parse(ServrIP),9050);  
//IPEndPoint ipepnewIPEndPoint (IPAddress.Parse("10.170.201.18"),9050);  
server = new Socket (AddressFamily.InterNetwork,  
SocketType.Stream, ProtocolType.Tcp);  
server. Connect(ipep);
```

When the C# Client program starts, it will connect to the C# Server Socket Program and start to reads data from Network Stream, and also write to the Network Stream. When you start the client program you will get a message from Server "client started". When press the button at the bottom of Client window, it will send a message to the Server and also receive response from the Server.

4.5 Hardware

The hardware in the project is connected Serially with the Server Computer. So, it is important to tell about serial programming of microcontroller

4.5.1 Serial Programming in PIC

There are two ways for serial communication in PIC one is through Software UART library and second is Hardware USART Library. Both are discussed briefly. But the one we have used is USART.

4.5.1.1 Software UART Library

Universal asynchronous receiver transmitter (UART) software library is used for RS232-based serial communication between two electronic devices. In serial communication, only two cables (plus a ground cable) are required to transfer data in either direction. Data is sent in serial format over the cable bit by bit. Normally, the receiving device is in idle mode with it transmit (TX) pin at logic 1, also known as MARK. Data transmission starts when this pin goes to logic 0, also known as SPACE. The first bit sent is the start bit at logic 0. Following this bit,

7 or 8 data bits are sent, followed by an optional parity bit. The last bit sent is the stop bit at logic 1. Serial data is usually sent as a 10-bit frame consisting of a start bit, 8 data bits, and a stop bit, and no parity bits.

4.5.1.2 Hardware USART Library

The universal synchronous asynchronous receiver transmitter (USART) hardware library contains a number of functions to transmit and receive serial data using the USART circuits built on the PIC microcontroller chips. Some PIC18F-series microcontrollers have only one USART (e.g., PIC18F452), while others have two USART circuits (e.g., PIC18F8520). Hardware USART has an advantage over software-implemented USART, in that higher baud rates are generally available and the microcontroller can perform other operations while data is sent to the USART. The hardware USART library provides the following functions

4.5.2 Hardware Description

Figure 4.8 shows the schematics of the hardware

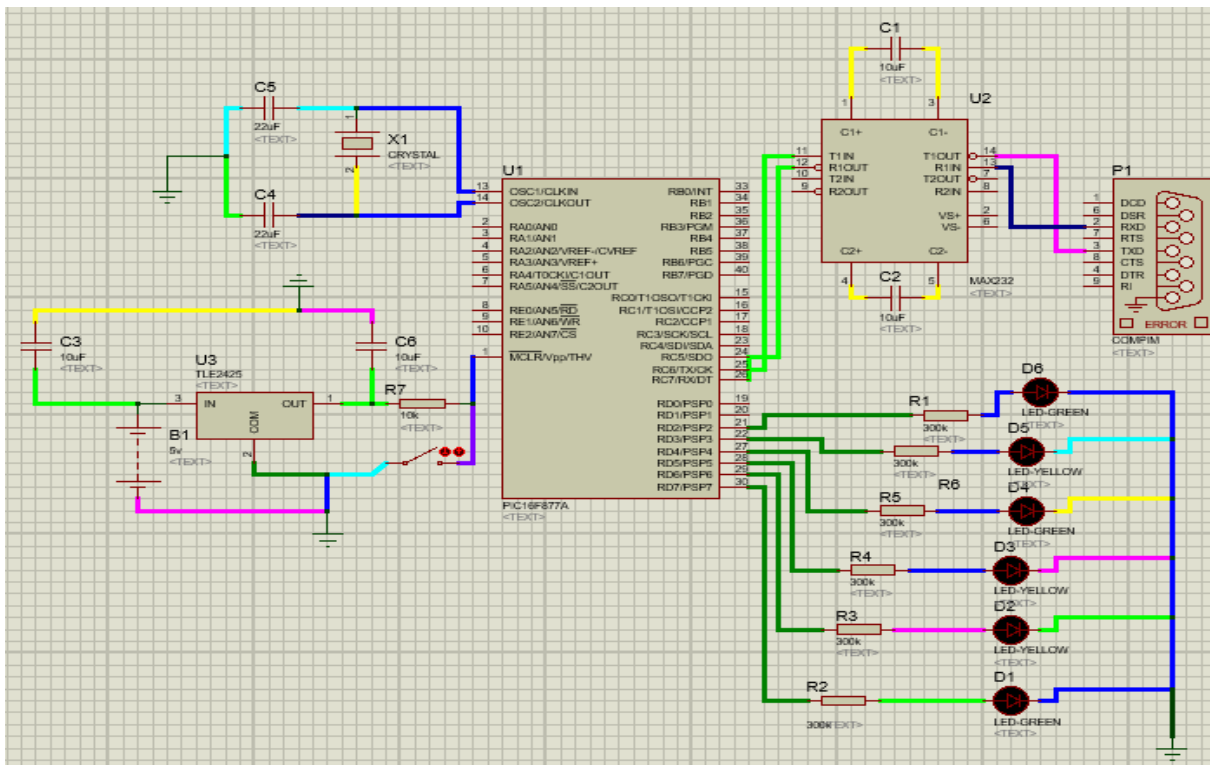


Figure 4. 8 Schematics of Hardware

Serial Port is connected with PIC16f877A through MAX232 to make the voltage levels compatible. And to test the circuit in we used LEDs in schematics Serial port is connected to Tx (26) and Rx (25) pin of controller via level converter the controller will wait for the data from serial port of Server will send the data to controller, which will be in a form of a string.

After receiving it controller will use the code and decode the received data to give an appropriate output to PORTD to turn on/off the desired device. Then the controller will send back an acknowledge to the sever that the desired device has been turned on/off by writing back on. All this will be done using Hardware USART Library commands as stated above.

Figure 4.9 shows the working of hardware in Flow Chart

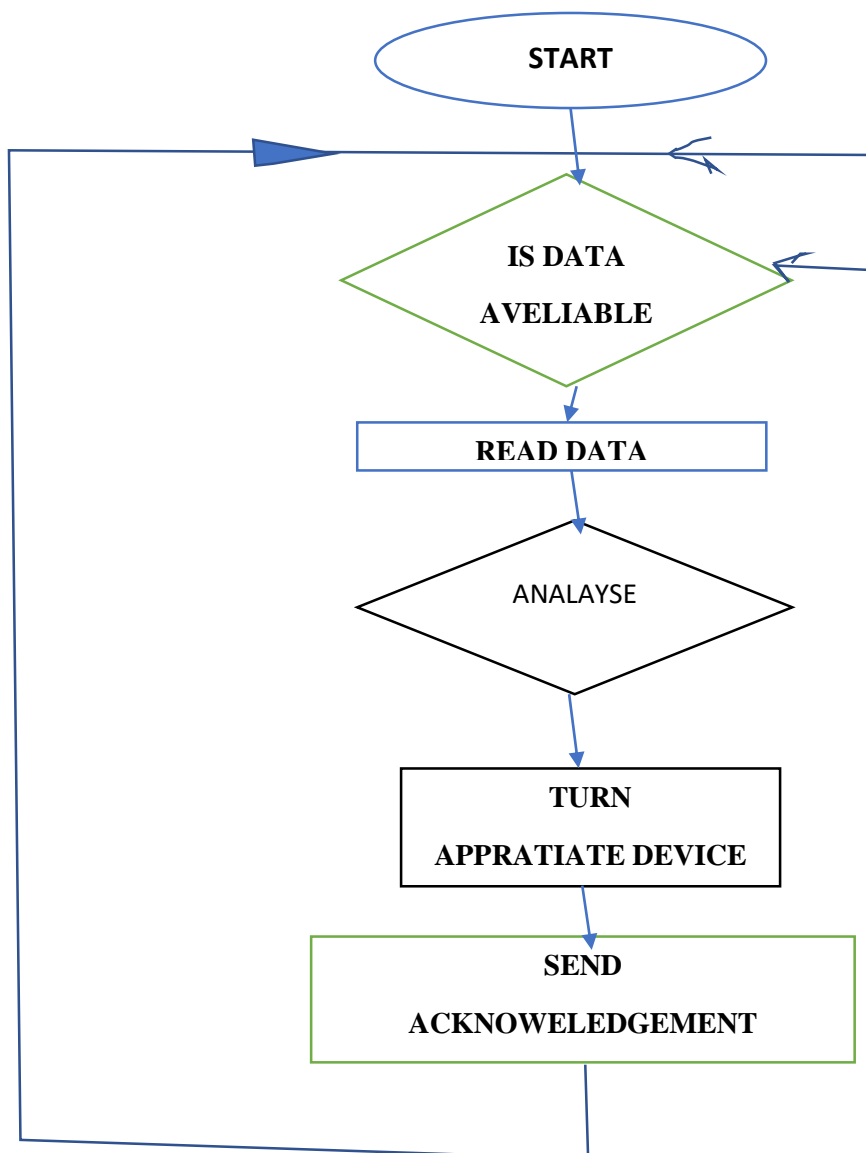


Figure 4.9 shows the working of hardware in Flow Chart

4.5 Simulation Result

Proteus and visual studio software have been used for the simulation of the circuit discussed above. The microcontroller is controlled the home appliance by using server and client application.

The simulation circuit is shown in the figure below.

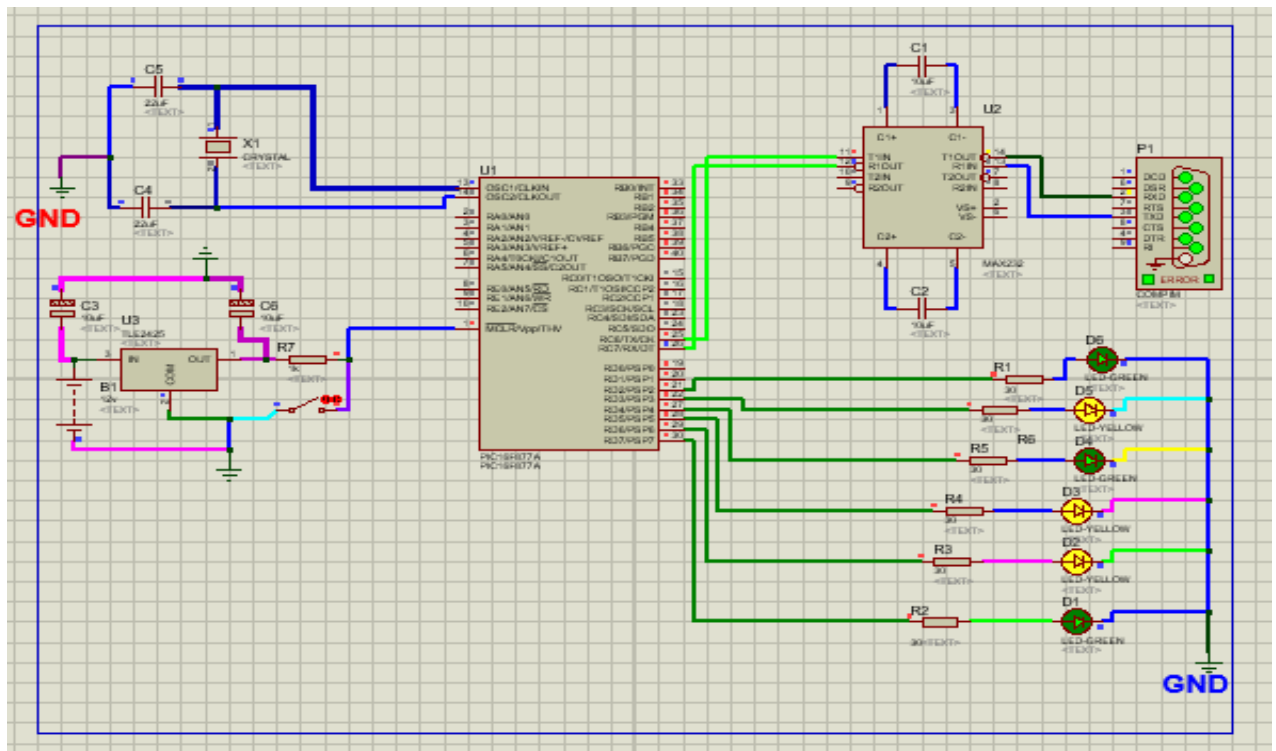
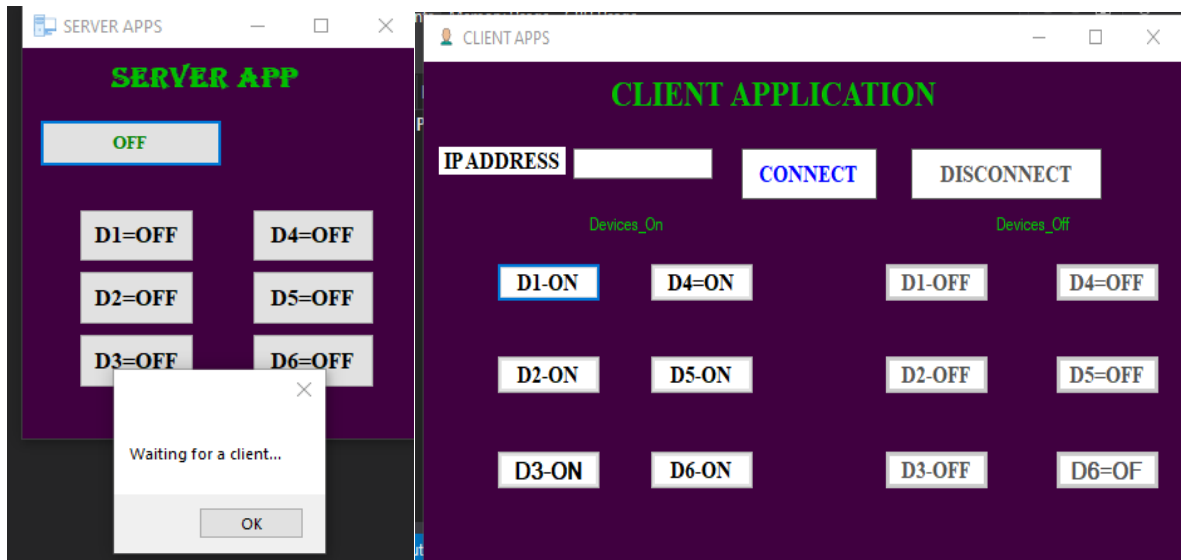


Figure 5. 1 Simulation Result of project

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 CONCLUSION

We developed a system to automate home appliances. This system has many special features controlling the electricity of the building is one of the features which enables the user to control the electronic appliances of his/her building from within the building boundary or anywhere in the world using web-based system.

Finally, we checked working of this system by software simulation. If we get material, we can do hardware implementation.

5.2 RECOMMENDATION

This project is a small implication of our concept in automating and monitoring a system. The practical applications of this project are immense and can have vast level of implementation. This small concept can be used in fields home automation, and many other related fields where continuous monitoring and regulation is needed. So, this is not the end of the project but rather is a step towards exploring other possibilities that it brings with it.

We feel very happy to work in such a challenging project which has tremendous application and possibilities.

We recommend our brothers and sisters to work in such field, which actually gives a lot of satisfaction while working. The project work in the fact gives a lot of confidence to fight out in this challenging world.

As one proceeds one cannot believe how much knowledge he/she gains and the teamwork, which the project work teaches, really will have a new experience.

References

- [1]. A. Alheraish, “Design and implementation of home automation system,” IEEE Transactions on Consumer Electronics, vol. 50, pp.1087-1092, Nov. 2004.
- [2] A. R. Al-Ali and M. Al-Rousan, “Java-based home automation system,” IEEE Transactions on Consumer Electronics, vol. 50, no. 2, pp.498-504, May 2004.
- [3] Richa Shukla, Rituraj, shriya mehotra, shobhit chatuedi, vishal kany rai
SMSBASEDDEVICESCONTROLSYSYEMUSINGGSM.
- [4] <https://circuitdigest.com/microcontroller-projects/gsm-based-home-automation-using-arduino>
- [5] https://en.wikipedia.org/wiki/Home_automation
- [6] K. Thamaraiselvi¹, G. Mariselvam², A. Dinesh Kumar³, R. Lavanya⁴, J. Sumaiya Parveen⁵ Department of CSE, K.S.R college of Engineering, Tiruchengode.” **Home Automation System Using GSM Technology**” IJSART - Volume 2 Issue 5, PAGE 6–MAY 2016
- [7] Conte, G., & Scaradozzi, D. (2003). Viewing home automation systems as multiple agents systems. RoboCUP2003, Padova, Italy. Retrieved from http://www.robosiri.it/ROBOCUP_2003/ROBOCUPSITOSIRI/articles/pdf/Conte.pdf
- [8] 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>
- [9] Delgado, A. R., Picking, R., & Grout, V. (2006) Remote-controlled home automation systems with different network 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>
- [10] 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.acadmypublisher.com/jnw/vol03/no02/jnw03025863.pdf>

[11] Murthy, M. V. R. (2008). Mobile based primary health care system for rural India. W3C workshop on Role of Mobile Technologies in Fostering Social Development, Jun 2008

[12] 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. 21972200, Geneva, Switzerland, Sept. 1-4, 2003. Retrieved from

<http://www.wcl.ee.upatras.gr/ai/papers/potamitis14.pdf>

http://www.w3.org/2008/02/MS4D_WS/ retrieved on 4th November 2010

<http://www.smarthome.com/homeautomation.html> retrieved on 4th November 2010

APPENDIX-A

A-1 Source Code of Server Application

Form1.c

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO.Ports;
using System.Linq;
using System.Net;
using System.Net.Sockets;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace NEW_SERVER
{
    public partial class SERVER : Form
    {
        private int recv;
        private byte[] data;
        private Socket newsock;
        private Socket client;
        private IPEndPoint clientep;
        private readonly SerialPort sp1;

        public SERVER()
        {
            InitializeComponent();
            sp1 = new SerialPort();
        }

        private void SERVER_Load(object sender, EventArgs e)
        {
            sp1.PortName = "COM2";
            sp1.BaudRate = 9600;
            sp1.DataBits = 8;
            sp1.Parity = Parity.None;
            sp1.StopBits = StopBits.One;
            sp1.Open();
        }

        public void CON()
        {
            try
            {
                data = new byte[1024];
                IPEndPoint ipep = new IPEndPoint(IPAddress.Any, 9050);
                newsock = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
                newsock.Bind(ipep);
                newsock.Listen(0);
                MessageBox.Show("Waiting for a client...");
                client = newsock.Accept();
                clientep = (IPEndPoint)client.RemoteEndPoint;
                MessageBox.Show("Connected with " + clientep.Address + " , " + clientep.Port);
                string welcome = "connected...";
                data = Encoding.ASCII.GetBytes(welcome);
            }
        }
    }
}
```

```
        client.Send(data, data.Length, SocketFlags.None);  
    }  
    catch (Exception ex)  
    {  
        MessageBox.Show(ex.ToString());  
    }  
}
```

A-2 Source Code of Client Application

Form1.c

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Net;
using System.Net.Sockets;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace NEW_CLIENT
{
    public partial class Client : Form
    {
        private byte[] data;
        private string input;
        private string stringData;
        private Socket server;
        private String ServerIP;

        public Client()
        {
            InitializeComponent();
        }
        private void Client_Load(object sender, EventArgs e)
        {
            Button12.Enabled = false;
            Button11.Enabled = false;
            Button10.Enabled = false;
            Button9.Enabled = false;
            Button8.Enabled = false;
            Button7.Enabled = false;
            Button14.Enabled = false;
            TextBox1.Text = "127.0.0.1";
            ServerIP = TextBox1.Text;
        }
        private void CON()
        {
            ServerIP.Trim();
            try
            {
                ServerIP = TextBox1.Text;
                data = new byte[1024];
                IPEndPoint ipep = new IPEndPoint(IPAddress.Parse(ServerIP), 9050);
                //IPEndPoint ipep = new IPEndPoint(IPAddress.Any, 9050);
                server = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
                server.Connect(ipep);
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.ToString());
            }
        }
    }
}
```

A-3 Code of Microcontroller

```
#include <16F877A.h>

#fuses XT, NOLVP, PUT

#use delay (clock = 4000000)

#use rs232(baud=9600, parity=N, xmit=PIN_C6, rcv=PIN_C7, bits=8)

#byte PORTA = 0x05

#byte PORTB = 0x06

#byte PORTC = 0x07

#byte PORTD = 0x08

#byte PORTE = 0x09

#byte TRISA = 0x85

#byte TRISB = 0x86

#byte TRISC = 0x87

#byte TRISD = 0x88

#byte TRISE = 0x89

#bit pd_7=PORTD.7

#bit pd_1=PORTD.1

#bit pd_2=PORTD.2

#bit pd_3=PORTD.3

#bit pd_6=PORTD.6

#bit pd_4=PORTD.4

#bit pd_5=PORTD.5

void main () {

unsigned char ch='0';

PORTD=255;
```

```
PORTB=255;

TRISD=0;

TRISB=0;

while (1) {

delays (1000);

ch=getc ();

switch(ch){

case '1':

PORTD=PORTD & 0x7F;           // pd_7=0;

break;

case '2':

PORTD=PORTD & 0xBF;           // pd_6=0;

break;

case '3':

PORTD=PORTD & 0xDF;

break;

case '4':

PORTD=PORTD & 0xEF;

break;

case '5':

PORTD=PORTD & 0xF7;

break;

case '6':

PORTD=PORTD & 0xFBbreak;
```

```
PORTD=PORTD | 0x80;    // pd_7=1;
```

```
break;
```

```
case 'b':
```

```
PORTD=PORTD | 0x40;
```

```
break;
```

```
case 'c':
```

```
PORTD=PORTD | 0x20;
```

```
break;
```

```
case 'd':
```

```
PORTD=PORTD | 0x10;
```

```
break;
```

```
case 'e':
```

```
PORTD=PORTD | 0x08;
```

```
break;
```

```
case 'f':
```

```
PORTD=PORTD | 0x04;
```

```
break;}}
```