Palestine Polytechnic University

College of Engineering & Technology
Electrical & Computer Engineering

Graduation Project

IP Telephony Device

Project Team
Raed Al-Shammas
Abdullateef Abualfeilat

Project Supervisor
Eng. Radwan Tahboob

Hebron _ Palestine

July 2004

جامعة بوليتكنك فلسطين

الخليل_ فلسطين

كلية الهندسة و التكنولوجيا

دائرة الهندسة الكهربائية و الحاسوب

اسم المشروع:

IP Telephony Device

أسماء الطلبة

رائد الشماس   عبد الطيف أبو الفيلات

بناء على نظام كلية الهندسة و التكنولوجيا و إشراف و متابعة المشرف المباشر على المشروع و موافقة أعضاء اللجنة الممتحنة تم تقديم هذا المشروع إلى دائرة الهندسة الكهربائية و الحاسوب و ذلك للوفاء بمتطلبات درجة البكالوريوس في الهندسة تخصص الحاسوب.

توقيع المشرف

....

توقيع اللجنة الممتحنة

....

توقيع رئيس الدائرة

....

Abstract

Voice over IP is being developed to integrate voice and data transmission. Over the past few years it has moved from being a computer toy to a major force in today's telecommunications systems. It has the potential to provide a wide range of enhanced applications at a lower operational cost with greater flexibility than a tradition phone system.

The goals of this project were to understand both VoIP and the existing phone network in order to gain an insight on how the two will be combined. Also to outline the concepts concerned with voice and data networking and the sending of voice over IP. Then to use this knowledge to design and implement a software phone that operates within the Session Initiation Protocol specifications.

At the end of this project a program that communicated with other Session Initiation Protocol User Agents and Proxy Servers had been completed.

Dedication

To our parents, brothers, friends, to all peoples who support us in this project, to all who loves the technology and want to create new inventions to this world, to all who made any thing in VoIP (Voice Over IP).

Raed
Abdullateef

Acknowledgment

We would like to seize this opportunity to extend best wishes and gratitude to well-deserved supervisor Eng. Radwan Tahboob and all who have been so forthcoming and instrumental in helping us reach our goals.

Contents

List of Figures

List of tables

Chapter one

Chapter one

Introduction

1.1 Overview

When Alexander Graham Bell made the first phone call in 1876 you needed a physical connection to the phone you wanted to call, however it quickly became clear that this was not practical as a huge wire mesh would be required to have a connection to every other phone. So the idea of using a switch to connect calls was used, originally by using human operators then electronic switches were developed. Since then a well developed language has been designed for the switches to communicate with each other.

Why Switch to Voice over IP?

Increased savings are obtained by integrating the voice with a data network. The reduction to one network means a reduction of administration and support staff can be achieved. The actual cost of calls is reduced as charges switch from time to the amount of data, the user could possible chose to use a lower quality compression method, reducing the amount of data per second, and so make a call cheaper.

It also allows new companies to enter the telecommunications market. InternetService Providers (ISP) could now also provide a phone service. If a home is hooked up to a broadband permanent connection then the ISP could provide an IP phone removing the need for a conventional PSTN phone in the house. This would also eliminate the common problem of phones being engaged for hours while some one is connect to the internet as a voice call could still come through the connection. For dial-in connections a service called V2L has been designed which allows your phone calls to go through your ISP while you are connected to your computer allowing you to still take incoming calls even though your physical phone line is in use.

The most common case study used to highlight some of the benefits of switching to VoIP is that of a call centre. A tradition call centre would set up desks each with a computer terminal, connected to the support web pages for the particular company, and a phone, connected to the public phone systems through several expensive PBXs (Private Branch Exchange). The company has to pay for the office space, the phone system as well as people to maintain the internal phones. If it was to switch to a VoIP set-up the operators could work from home, with the company providing a internet connection and possibly a computer. The operator is now using the computer for both the source of information and to receive the phone call. The company saves money by not having to rent out office space and can negotiate more flexible working patterns with it operators since they are now working from home. All the functions provided by the PABX(Private Branch Automatic Exchange, call transfer, on hold etc, are still available and new function can be added by simply updating the software as apposed to having to add hardware or upgrade the PABX. Adjusting the number of operators is now very flexible where as with the old system a new desk had to be fitted or even new premises found.

1.2 Benefits

- Integration of Voice and Data

The integration of voice and data traffic will be demanded by multi application software. The inevitable evolution will be web servers capable of interacting with voice, data and images.

- Simplification

An integrated infra structure that supports all forms of communication allows more standardization and lesser equipment management. The result is a fault tolerant design.

- Network Efficiency

The integration of voice and data effectively fills up the data communication channels efficiently, thus providing bandwidth consolidation. The idea is to move away from the TDM(Time-Division Multiplexing) scheme wherein the user is given bandwidth when he is not talking. Data networks do not do this. It is a big saving when one considers the statistics that 50% of a conversation is silence. The network efficiency can be further boosted, by removing the redundancy in certain speech patterns.

- Cost reduction

The Public Switched Telephone Networks' toll services can be bypassed using the Internet backbone, which means slash in prices of the long distance calls. However these reductions may slightly decrease when the Federal communications Commission (FCC) removes the Enhanced Service Provider (ESP) status granted to Internet service providers (ISPs) by which they do not have to pay the local access fees to use the telephone company (TELCO) local access facilities. Access fees form a significant part of all long distance calls. But in spite of this, the circuit switched telephony would be expensive because of lack of bandwidth consolidation and speech compression techniques.

1.3 System Design Options

There are many branches of designing the project; some of these branches related to a software design and other related to a hardware design such as:

- PC to Phone Applications:

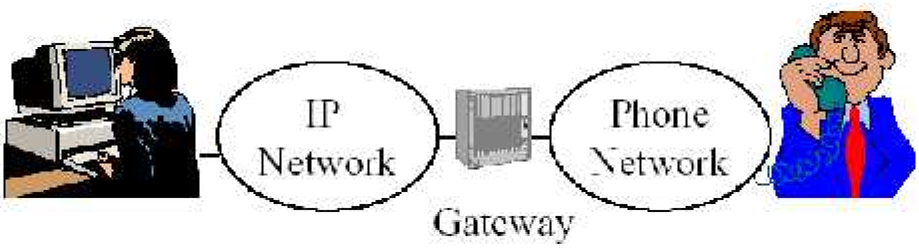


Figure (1.1): PC to Phone application
>Need a Gateway that connects the IP network to phone network.

- Phone to Phone Applications:

Through the PSTN(Public Switched Telephone Network) network.

Using network phone device that is connected directly to the network (without connect it to PC)

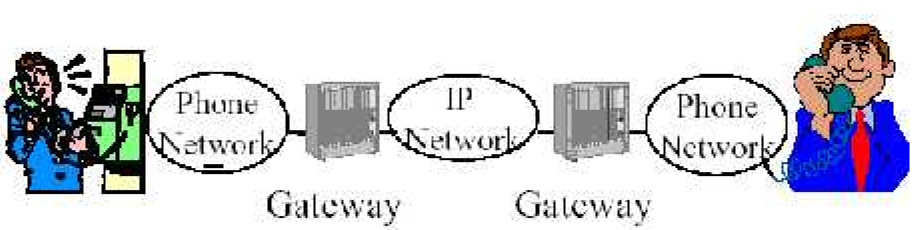


Figure (1.2): Phone to Phone application

This project choose another type which is similar to Phone to Phone type but the terminal connected to PC rather than Gateway.

1.4 System Requirements

1.4.1 Software Requirements

- Windows Operating System 2000.
- TCP/IP protocol. with H323 protocol..
- Visual C++ programming language.
- SDK of VoIP processor Tiger 560 B.

1.4.2 Hardware Requirements

- Two phone devices with RG11 ports.
- Two free RG45 ports on the Hub or Switch.
- Two USB connection on two computer.

1.5 System cost

The cost of our project is divided into three types of cost: component cost, software cost and effort.

Table 1.1 Actual cost of the project component

Component	Cost
Two Tjnet VoIP development kits	200$
Three DTMF Decoders (CM8870)	15$
TTL IC's	50$
Low power Relays	12$
Two phone devices	20$
Others(resistors,diodes,transistors…)	10$

Table 1.2 Software cost

Windows XP	100$
Visual C++6	500$
IP phone SDK	50$

Table 1.3 Efforts and services

Data collection (Internet, library…)	150$
Transportations and communications	100$
Shipment	45$
Working hours	5$/hour

1.6 Time Schedule

The project accomplished and implemented within 32 weeks. It divided into phases and some phases take placed in parallel.

Table 1.4 Time Estimation

Weeks	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32
Design																
Determine the suitable chips																
Searching and collecting information																
Implementation																
Documentation																

1.7Documentation Preview

This documentation report was written in the software engineering techniques and methods, and the design follows the object oriented programming approach. It was organized as follows:

The first chapter is Introduction. It outlines Overview, what is VoIP and its types, Design Options, Time Schedule and finally Documentation Preview.

The second chapter is the Theoretical Background. This chapter contains a theoretical background for both the hardware and the software parts.

The third chapter is Hardware Design, which discusses the general block diagram for the design, project unit.

The fourth chapter is Software System Design, which shows out the system software flow chats and subroutines.

The fifth chapter is Implementations and Testing, discuss the steps we followed to test out our design.

The sixth chapter is Conclusion and Future work, provides many suggestion to expand this project and summarize the project conclusion.

Chapter Two

Chapter Two

Theoretical Background

In this chapter we will talk about some component and devices used in the implementation and it present the theoretical background of the system

2.1 VoIP Technology

2.1.1 VoIP concept

VoIP stands for 'V'oice 'o'ver 'I'nternet 'P'rotocol. As the term says VoIP tries to let go voice (mainly human) through IP packets and, in definitive through Internet.

VoIP is the ability to make telephone calls and send faxes over IP-based data networks with a suitable quality of service (QoS) and superior cost benefit. In general, this means sending voice information in digital form in discrete packets rather than in the traditional circuit-committed protocols of the public switched telephone network. Voice over IP (VoIP) allows the human voice and fax information to travel over a packet data network concurrently with traditional data packets.

2.1.2 VoIP advantages

A major advantage of VoIP and Internet telephony is that it avoids the tolls charged by ordinary telephone service.

VoIP, now used somewhat generally, derives from the VoIP Forum, an effort by major equipment providers, including Cisco, VocalTec, 3Com, and Netspeak to promote the use of ITU-T H.323, the standard for sending voice (audio) and video using IP on the public Internet and within an intranet. The Forum also promotes the user of directory service standards so that users can locate other users and the use of touch-tone signals for automatic call distribution and voice mail. In addition to IP, VoIP uses the real-time protocol (RTP) to help ensure that packets get delivered in a timely way. Using public networks, it is currently difficult to guarantee Quality of Service .Better service is possible with private networks managed by an enterprise or by an Internet telephony service provider (ITSP). VoIP can use accelerating hardware to achieve this purpose and can also be used in a PC environment. VoIP provides rich benefits for all levels of users, from networking equipment manufacturers who are providing next generation equipment to service providers, who can now market new services to business and home users.

At the very minimum, these benefits include more cost effective traditional voice and fax service. However, users will also benefit from revolutionary, new VoIP-based services. VoIP will fundamentally change the way communication occurs.

2.1.3 VoIP Standards

VoIP technology is rapidly evolving, but still there are ongoing debates about a de facto standard that will bridge traditional circuit-switched networks and packet switched networks. Four protocol standards govern VoIP technology: H.323, SIP, MGCP and Megaco/H.248.

2.2 Telephone

A telephone is one of the simplest devices you have in your house. It is so simple because the telephone connection to your house has not changed in nearly a century. If you have an antique phone from the 1920s, you could connect it to the wall jack in your house and it would work fine!

The very simplest working telephone would look like this inside:

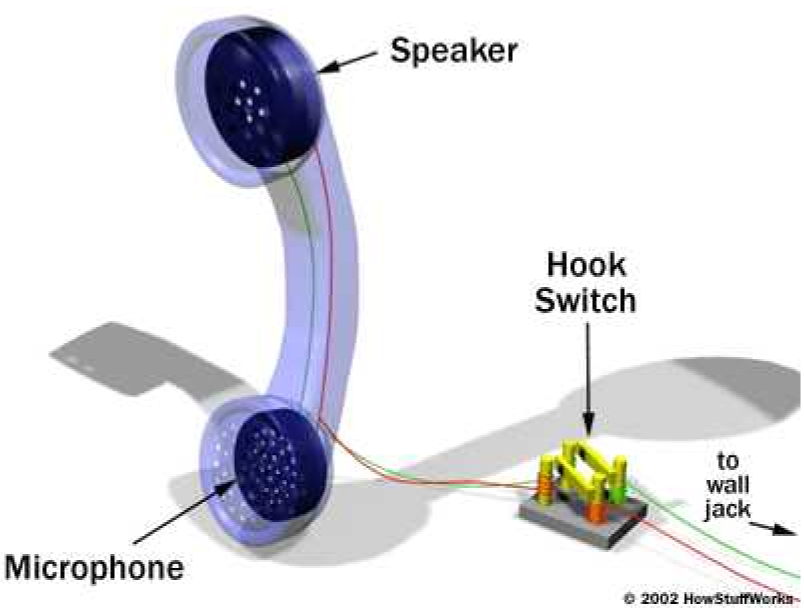


Figure (2.1): Old phone construction

As you can see, it only contains three parts and they are all simple:

- A switch to connect and disconnect the phone from the network - This switch is generally called the hook switch. It connects when you lift the handset.
- A speaker - This is generally a little 50-cent, 8-ohm speaker of some sort.
- A microphone - In the past, telephone microphones have been as simple as carbon granules compressed between two thin metal plates. Sound waves from your voice compress and decompress the granules, changing the resistance of the granules and modulating the current flowing through the microphone.

That's it! You can dial this simple phone by rapidly tapping the hook switch -- all telephone switches still recognize "pulse dialing." If you pick the phone up and rapidly tap the switch hook four times, the phone company's switch will understand that you have dialed a "4."

The only problem with the phone shown on the previous page is that when you talk, you will hear your voice through the speaker. Most people find that annoying, so any "real" phone contains a device called a duplex coil or something functionally equivalent to block the sound of your own voice from reaching your ear. A modern telephone also includes a bell so it can ring and a touch-tone keypad and frequency generator. A "real" phone looks like this:

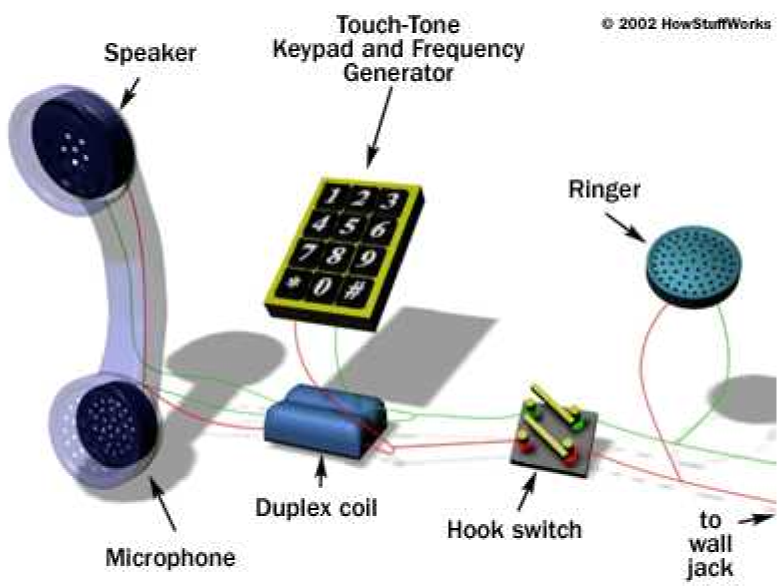


Figure (2.2):New phone construction

Still, it's pretty simple. In a modern phone there is an electronic microphone, amplifier and circuit to replace the carbon granules and loading coil. The mechanical bell is often replaced by a speaker and a circuit to generate a pleasant ringing tone.

2.3 USB Port

2.3.1 USB Port Concept

The Universal Serial Bus gives you a single, standardized, easy-to-use way to connect up to 127 devices to a computer.

Just about every peripheral made now comes in a USB version. A sample list of USB devices that you can buy today includes:

Printers, Scanners, Mice, Joysticks, Flight yokes, Digital cameras, Web Port cams, Scientific data acquisition devices (Modems, Speakers, Telephones, Video phones), Storage devices such as Zip drives, Network connections

2.3.2 USB Connections

Connecting a USB device to a computer is simple -- you find the USB connector on the back of your machine and plug the USB connector into it.

Figure (2.3): USB port

The rectangular socket is a typical USB socket on the back of a PC.

Figure (2.4):USB connector

A typical USB connector, called an "A" connection

If it is a new device, the operating system auto-detects it and asks for the driver disk. If the device has already been installed, the computer activates it and starts talking to it. USB devices can be connected and disconnected at any time.

Many USB devices come with their own built-in cable, and the cable has an "A" connection on it. If not, then the device has a socket on it that accepts a USB "B" connector.

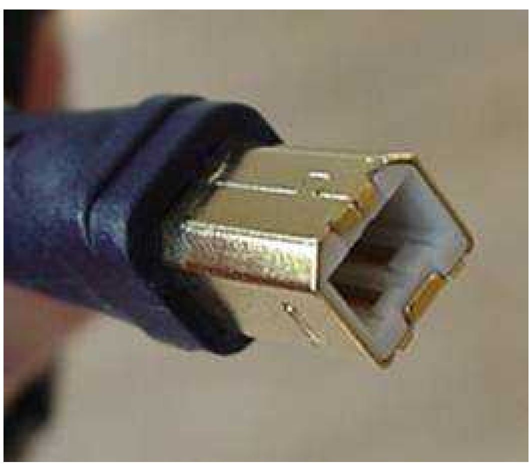


Figure (2. 5):USB connector type B

The USB standard uses "A" and "B" connectors to avoid confusion:

- "A" connectors head "upstream" toward the computer.
- "B" connectors head "downstream" and connect to individual devices.

By using different connectors on the upstream and downstream end, it is impossible to ever get confused -- if you connect any USB cable's "B" connector into a device, you know that it will work. Similarly, you can plug any "A" connector into any "A" socket and know that it will work.

The Universal Serial Bus has the following features:

- The computer acts as the host. Up to 127 devices can connect to the host, either directly or by way of USB hubs.
- Individual USB cables can run as long as 5 meters; with hubs, devices can be up to 30 meters (six cables' worth) away from the host. With USB 2, the bus has a maximum data rate of 480 megabits per second.

A USB cable has two wires for power (+5 volts and ground) and a twisted pair of wires to carry the data.

On the power wires, the computer can supply up to 500 milliamps of power at 5 volts.

Low-power devices (such as mice) can draw their power directly from the bus. High-power devices (such as printers) have their own power supplies and draw minimal power from the bus. Hubs can have their own power supplies to provide power to devices connected to the hub. USB devices are hot-swappable, meaning you can plug them into the bus and unplug them any time. Many USB devices can be put to sleep by the host computer when the computer enters a power-saving mode. The devices connected to a USB port rely on the USB cable to carry power and data.

Figure (2.6):USB cable

Inside a USB cable: There are two wires for power -- +5 volts (red) and ground (brown) -- and a twisted pair (yellow and blue) of wires to carry the data. The cable is also shielded.

2.3.3 The USB Process

When the host powers up, it queries all of the devices connected to the bus and assigns each one an address. This process is called enumeration -- devices are also enumerated when they connect to the bus. The host also finds out from each device what type of data transfer it wishes to perform:

Interrupt - A device like a mouse or a keyboard, which will be sending very little data, would choose the interrupt mode.

Bulk - A device like a printer, which receives data in one big packet, uses the bulk transfer mode. A block of data is sent to the printer (in 64-byte chunks) and verified to make sure it is correct.

Isochronous - A streaming device (such as speakers) uses the isochronous mode. Data streams between the device and the host in real-time, and there is no error correction. The host can also send commands or query parameters with control packets.

As devices are enumerated, the host is keeping track of the total bandwidth that all of the isochronous and interrupt devices are requesting. They can consume up to 90 percent of the 480 Mbps of bandwidth that is available. After 90 percent is used up, the host denies access to any other isochronous or interrupt devices. Control packets and packets for bulk transfers use any bandwidth left over (at least 10 percent).

The Universal Serial Bus divides the available bandwidth into frames, and the host controls the frames. Frames contain 1,500 bytes, and a new frame starts every millisecond. During a frame, isochronous and interrupt devices get a slot so they are guaranteed the bandwidth they need. Bulk and control transfers use whatever space is left. The technical links at the end of the article contain lots of detail if you would like to learn more.

2.4 VoIP processor (TIGER560B)

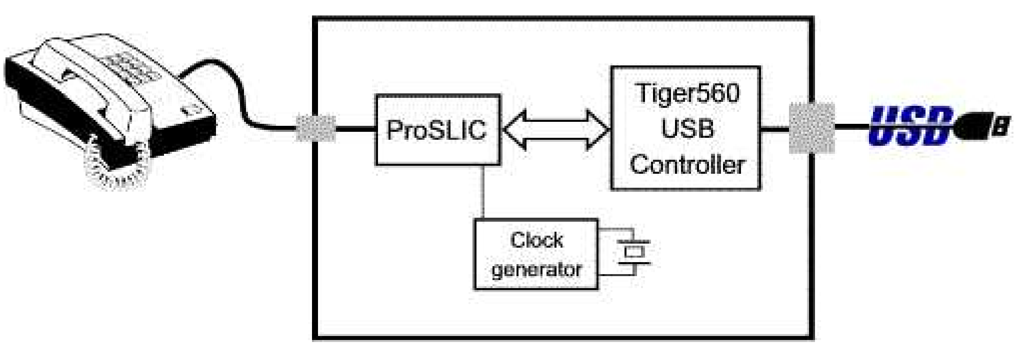


Figure (2.7):VoIP processor block diagram with a ProSLIC

VoIP processor with USB for Low Cost / High Quality Internet Telephony solutions.

It has the following features:

- Built in glue less support for: (Keypad, LCD, Phone interface (SLIC), Ringer / Buzzer)
- H.323 standard supported
- SIP standard supported
- No drivers required, all drivers are embedded in Windows. No hassle for installation or upgrade.
- Implements all of the required VoIP functions and USB interfacing for Internet telephony.
- Fully integrated with IP Phone Center application, provides easy to use functions: PC to PC calls , PC to gateway calls

2.5 SLIP (Subscriber Line Interface Chip)

The Si3210 ProSLIC provides a complete analog telephone interface ideal for customer premise equipment (CPE) applications. The Si3210 integrates subscriber line interface circuit (SLIC), codec and battery generation functionality into a single low-voltage CMOS integrated circuit.

The SLIC (Subscriber Line Interface Chip) is connected to the Tiger560B. The SLIC is controlled by the Tiger560B.

2.6 Relays

A relay is a simple electromechanical switch made up of an electromagnet and a set of contacts. Relays are found hidden in all sorts of devices. In fact, some of the first computers ever built used relays to implement Boolean gates.

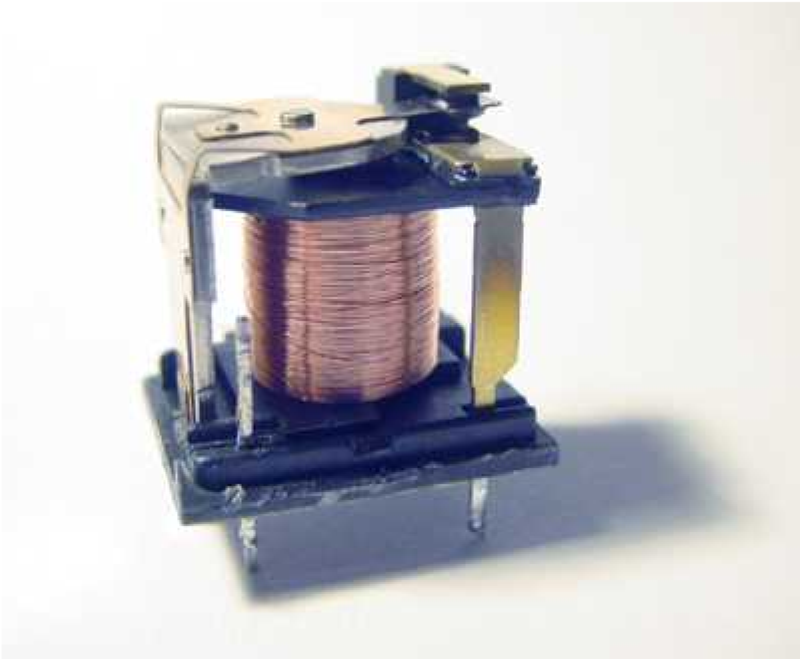


Figure (2.8):Relay

In this article, we will look at how relays work and a few of their applications. Relay Construction. Relays are amazingly simple devices. There are four parts in every relay:

- Electromagnet
- Armature that can be attracted by the electromagnet
- Spring
- Set of electrical contacts

A relay consists of two separate and completely independent circuits. The first is at the bottom and drives the electromagnet. In this circuit, a switch is controlling power to the electromagnet. When the switch is on, the electromagnet is on, and it attracts the armature (blue). The armature is acting as a switch in the second circuit. When the electromagnet is energized, the armature completes the second circuit and the light is on. When the electromagnet is not energized, the spring pulls the armature away and the circuit is not complete. In that case, the light is dark.

When you purchase relays, you generally have control over several variables:

- The voltage and current that is needed to activate the armature

- The maximum voltage and current that can run through the armature and the armature contacts
- The number of armatures (generally one or two)
- The number of contacts for the armature (generally one or two -- the relay shown here has two, one of which is unused)
- Whether the contact (if only one contact is provided) is normally open (NO) or normally closed (NC)

Chapter Three

Chapter Three

Design Concepts

This chapter shows the project objectives and its general block diagram. It divides the schematic diagram into deferent functional units and discuss how these units will interacts together.

3.1 Project Objectives

In this project we are going to implement the following objectives:

- Build an interface between the RG 11 and the USB port.
- Make a phone calls depend on VoIP technology rather than PSTN.
- Connect more than one telephone on the same IP address with deferent extension code.

There are other objectives like:

- Provide user privacy so no one can hire his call from other terminals.
- Provide very cheap long distance calls using the TCP/IP networks rather than PSTN.

3.2 Block Diagrams

3.2.1 General Block Diagram

The figure 3.1 shows the general block diagram for the system and its main parts:
VoIP Processor: responsible for VoIP arranging and formatting the voice calls and makes the interactions between the PC and the SLIC.

USB interface: used as a way to deal the voice data from PC to the VoIP processor, the USB has lot of benefits compared with the other ports like Parallel or Rs232 port, for example its high speed in the data transferring and Plug and play feature.

SLIC Interface: used for interfacing between the VoIP processor and the PSTN standards.

PC: used as an interface between the TCP/IP network and the IP telephony device.

Intranet: used as a media for the voice calls

VoIP processor: contains the parts for formatting the voice and interface with the SLIC and the USB port.

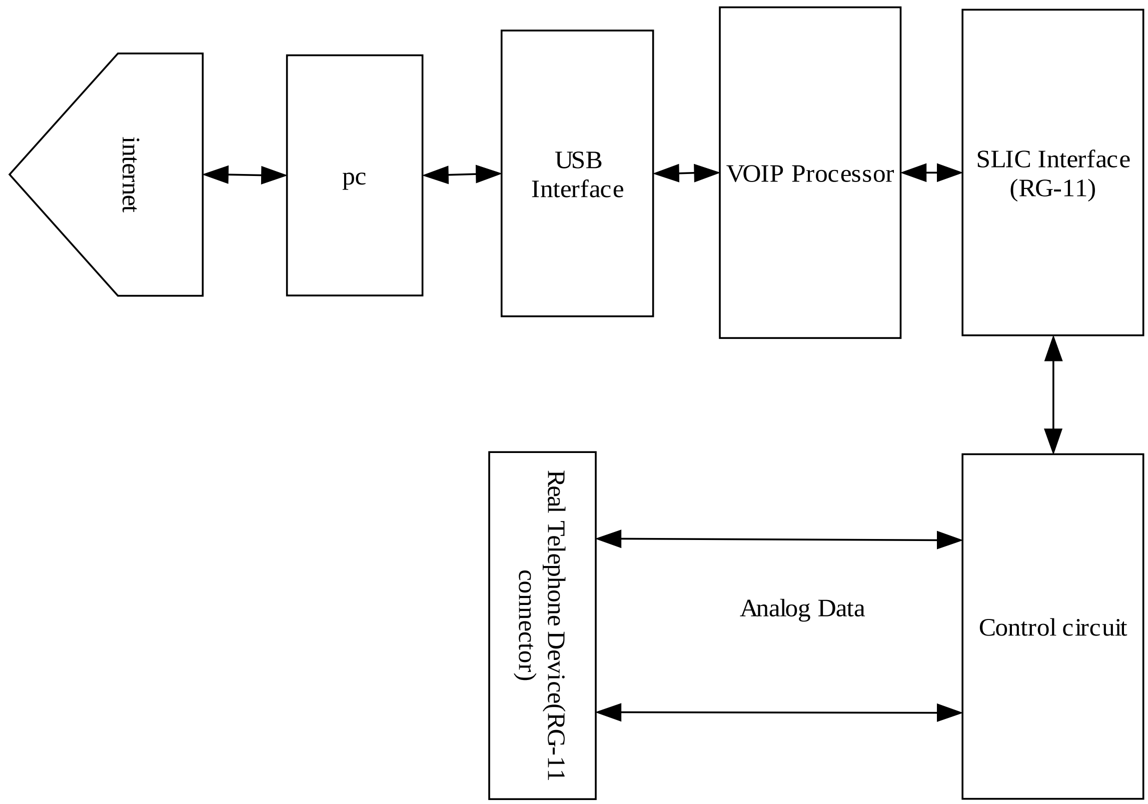


Figure (3.1): General Block Diagram of the System

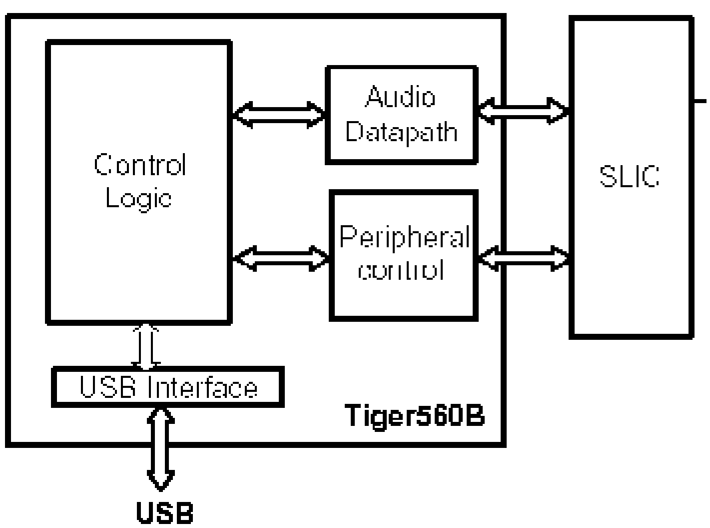


Figure (3.2): Tiger VoIP processor Block Diagram

3.2.2 Detailed Block Diagram

The detailed block diagram shows the components of units and how these components work together is shown in a figure 4.2 and 4.3, in these figures we have following parts:

- DTMF Receiver: responsible for decoding the DTMF frequencies that comes from the SLIC and convert it to decimal values.
- Decoding Circuit: Give the active phone line due to the received tone frequency.
- Switch circuit and Isolation: this circuit gives the priority to the phone that receive a call or the phone that request to make a call, and isolate the control circuit from the PSTN line.
- Open line Detector Circuit: Gives a signal to indicate which telephone that are now in service.

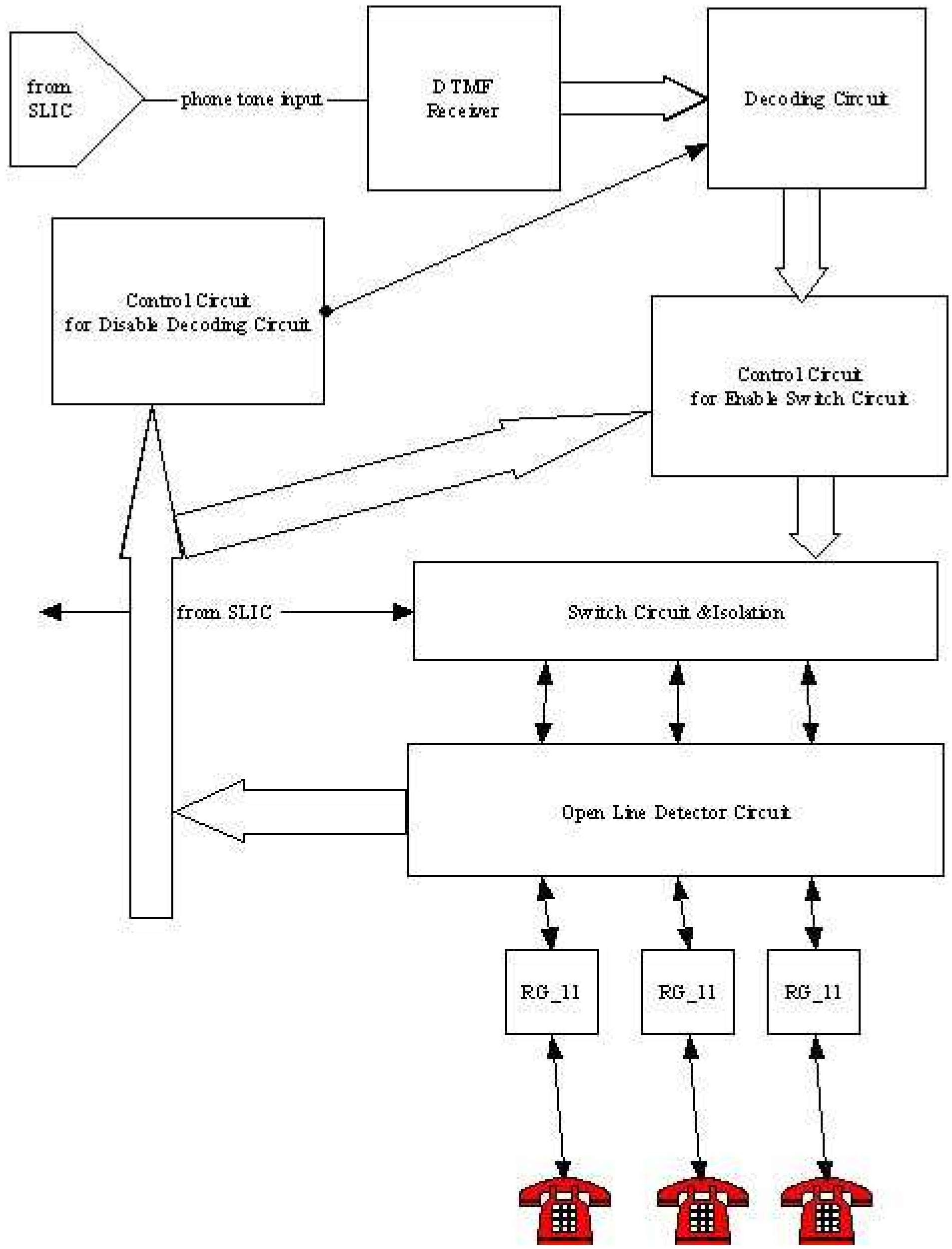


Figure (3.3): Detailed Block Diagram of Control Circuit

3.3 System Operation:

The initial setting of the system, we have two computers every computer with an IP, and the software of the system install on it, on the beginning all the telephone lines are connected to the network.

If any telephone need to make call and there is no other telephone in service (making call or receiving call), the line chose to this telephone line, then there is service waiting for the number interred from the keypad of the telephone to establish the connection to other terminal.

In the other side if telephone answer the call first the line chose to this telephone line then the ring signal sent to.

After the call end the system go to the first status.

Chapter Four

Chapter Four

Hardware System Design

In this chapter we will discuss the different design option to construct our system; also we will explain the detailed description of each part of the project.

4.1 Design Options

There are tow main option to use VoIP calls: the hard ware solution and the software solution. Both of these options are discussed in the following subsection:

4.1.1 The Hardware Option

In this option most of the function that needed in order to generating and completing a VoIP call using the normal telephone devices implementing by hardware components which is more efficient and effective.

4.1.2 The Software Option

In this option the complete function of generating VoIP calls lays on the software. This method decrease the cost as no additional chips needed to connected with the computer. However, the necessary function be implemented as software which be a load on the microprocessor in the computer, and it suffer if there is any application running on this computer.

4.2 Detailed description of system parts

The system contains the following components:

- VoIP processor with USB (TIGER560B)
- CMOS Integrated DTMF Receiver (CM8870)
- Decoding circuit (4_Bit BCD to a Decimal decoder 7442)
- Encoding circuit (10 to 4 encoder 74147)
- Current detecting circuit (Coil of Relay)
- Switching & Isolation circuit (Relays & Transistors to operate the Relay)

4.2.1 VoIP processor with USB (TIGER560B)

The TIGER560B has 100 pins as shown in the appendix.

This VoIP processor will used because:

- It is implement all the required VoIP functions for internet telephony and efficient .
- It is compatible with the used component (USB & SLIC).
- It is accommodate with the requirements.
- Low cost.

We use from this processor as functional line as in the following diagram:

- Line (92'D-', 93'D+', 94'V3', 98'USB_VCC') for USB Port.
- Line (81'DCIK', 84'FCS', 85'Din', 86'Dout', 78'SUSPEND')for SLIC to interfacing the RJ_11.

Note: we printed this circuit (fig 4.1) on board in Tiger Jet Network Inc

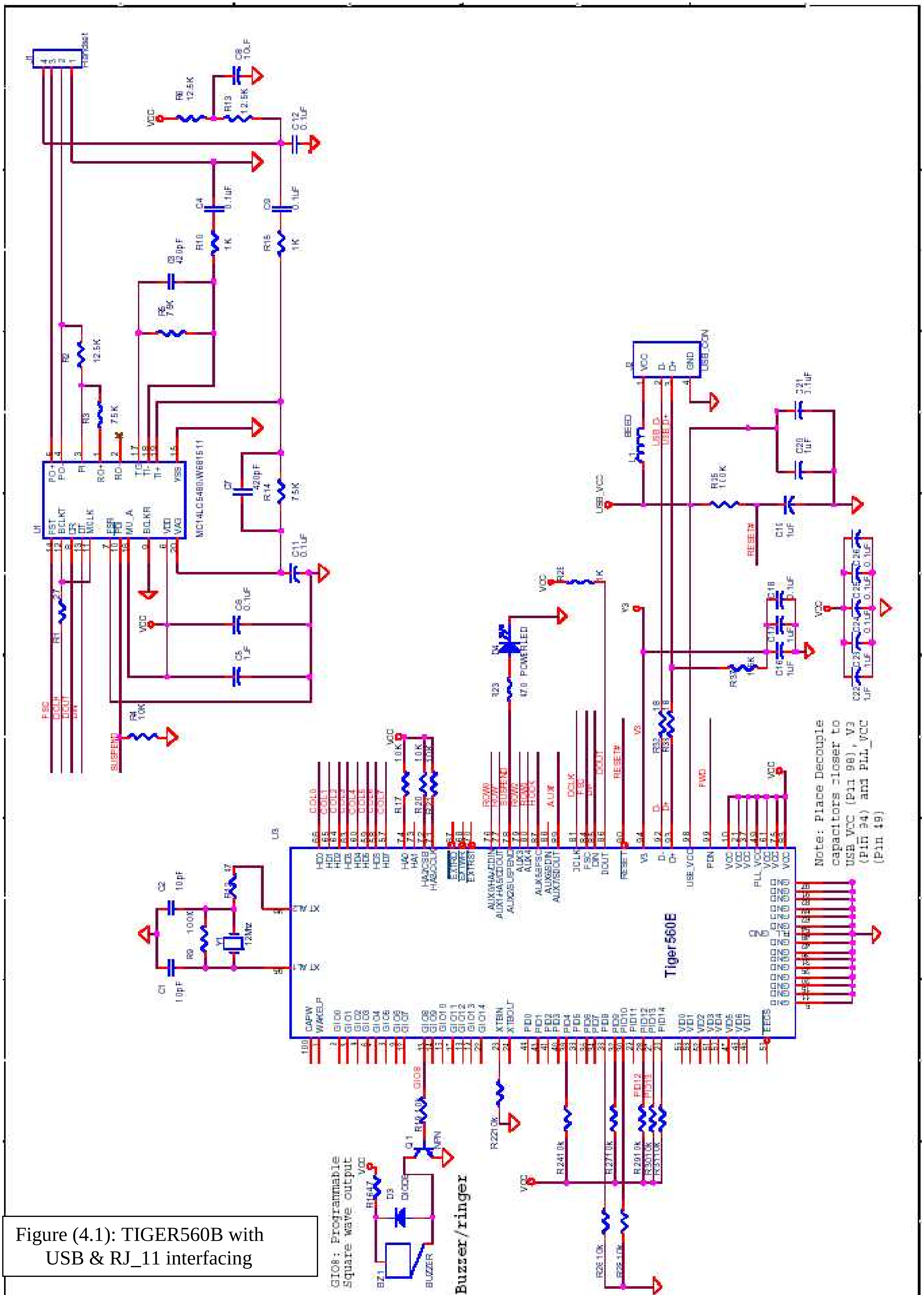


Figure (4.1): TIGER560B with USB & RJ_11 interfacing

4.2.2 CMOS Integrated DTMF Receiver (CM8870)

The CAMD CM8870 provides full DTMF receiver capability by integrating both the band split filter and digital decoder function into a single 18_DIP fig 4.2.

We use this IC in order to receive the number of the telephone that will be receive the incoming call fig 4.2.

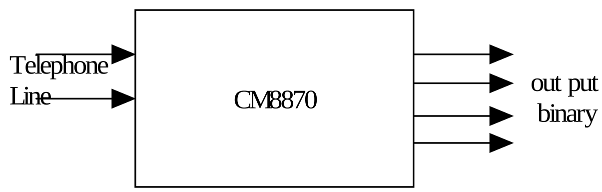


Figure (4.2): Block diagram of CM 8870

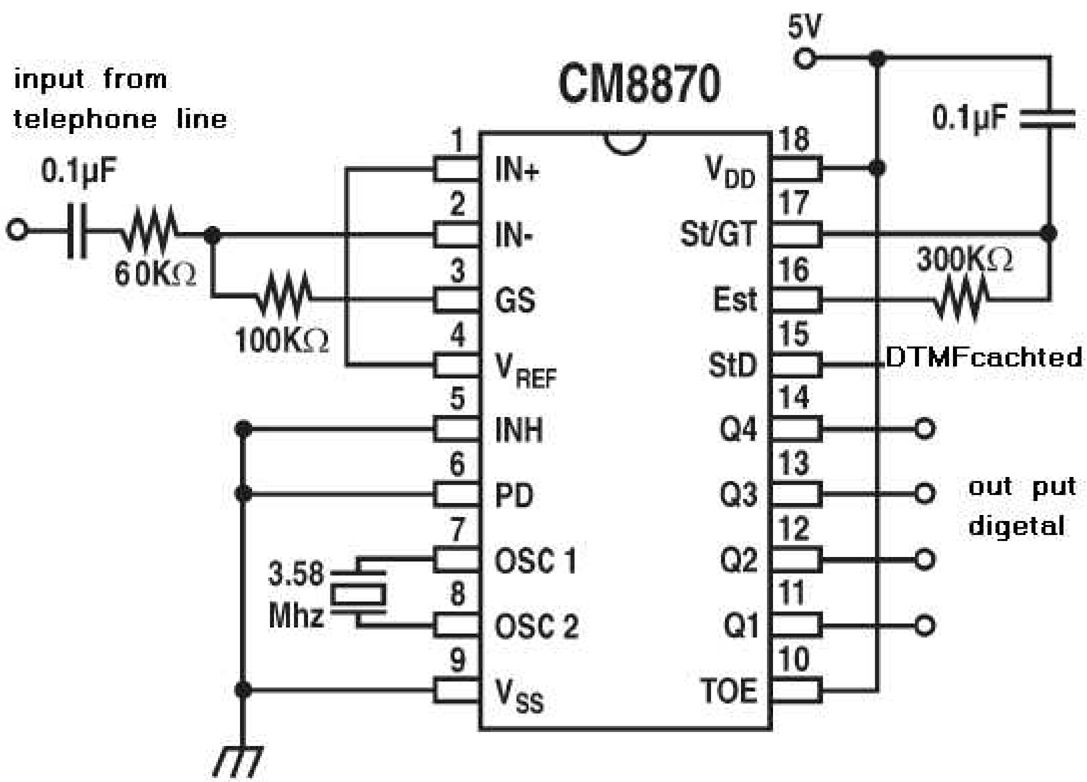


Figure (4.3): circuit of CM8870

.

4.2.3 Encoding circuit (10 to 4 encoder 74147)

Phone Lines decoding circuit

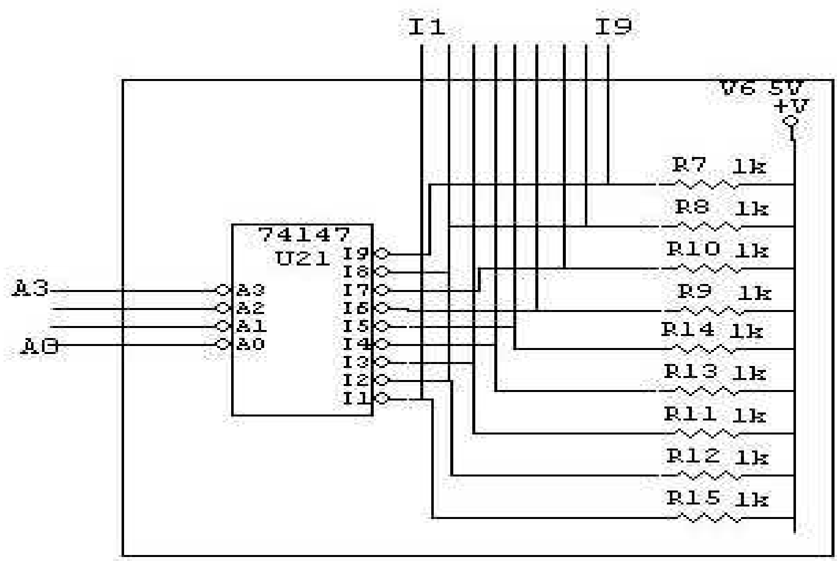


Figure (4.4): Phone Lines decoding circuit

This system uses this circuit to know which phone device is currently hooked up. Its input is supplied from the phone line detection circuit. A four bit generated code is provided as an output of the circuit to indicate that opened phone line.

The 74147 TTL Integrated circuit used as the decoder IC.

4.2.4 Decoding circuit

(4_Bit BCD to a Decimal decoder 7442)Phone Line Determination Circuit

This circuit accepts inputs either from the Phone Line Decoding Circuit or CIMOS Integrated DTMF Receiver. Highest priority is assigned to Phone Line Decoding Circuit. When no inputs supplied, this circuit activates all the phones. When CIMOS Integrated DTMF Receiver receives a call it redirects the request to this circuit. It generates a 4-bit binary code that identifies the required phone.

When a phone is handed up, this circuit cancels all coming requests from CIMOS Integrated DTMF Receiver and provides the services to that phone.

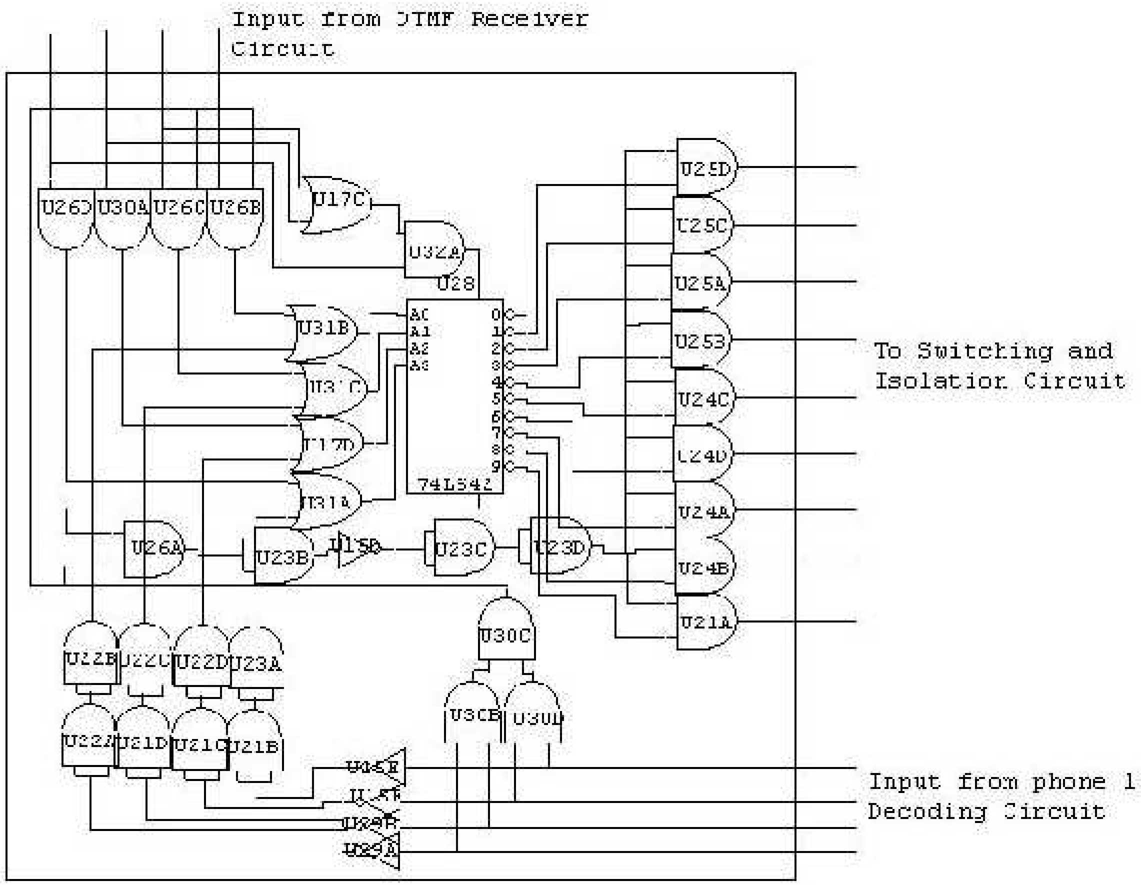


Figure (4.5): Phone Line Determination Circuit

4.2.5 Switching & Isolation circuit

(Relays & Transistors to operate the Relay)

This circuit recognizes the isolation of the phone devices from other components of the system. This circuit is required due to the high voltage (10 – 45 Volts) that the phone device deals with. It protects the phone devices from the unwanted signals that may be absorbed from other components. Also, it protects the remaining circuits components from signal that comes from the phone Lines.

Her relays used as switch and isolation two circuit.

The circuit also accepts a trigger signal from the Phone Line Determination Circuit and activates only one line according the triggering line.

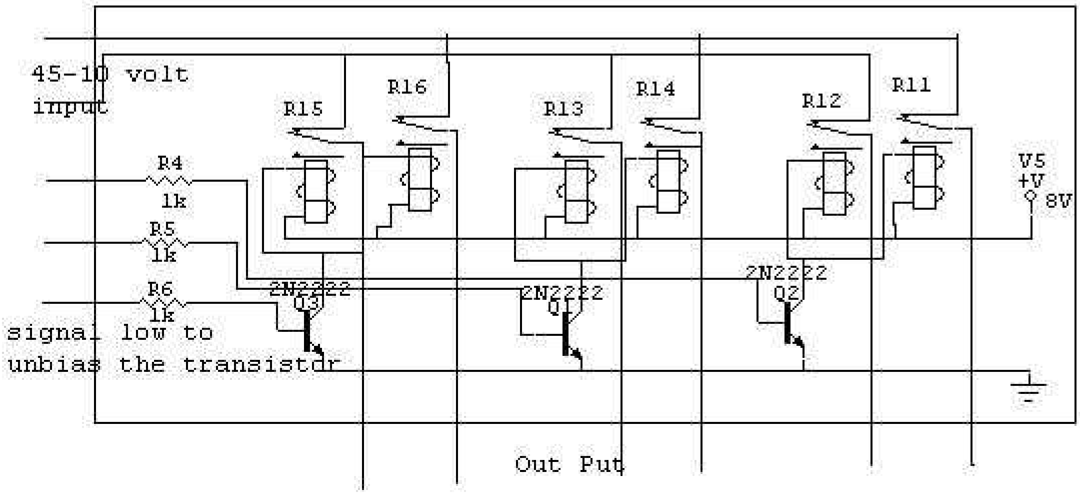


Figure (4.6): Switching & Isolation circuit

4.2.6 Current detecting circuit (Coil of Relay)

Phone device Detection Circuit

This circuit accomplishes two main jobs:

1. If a coming call is active for a certain phone, this circuit asserts that the related phone device responds or not.
2. The circuit detects which phone device needs a service. It ignores other devices at that case, and generate a signal for the decoding circuit.

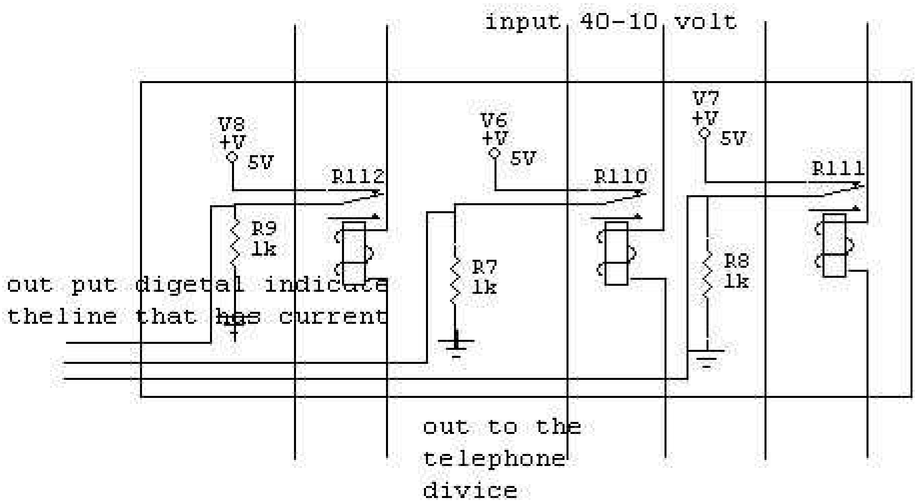


Figure (4.7): Phone device Detection Circuit

Chapter Five

Chapter Five

Software System Design

5.1. Overview

This project presents the software design that controls the system and makes an interconnection between its parts.

This chapter shows the flow charts for general system parts and their functionality.

The system software divides into three main parts:

- Detection part.
- Make call part.
- Receive call part.

5.2. Detection part

This part is responsible for initialize the program and detect whether the Tiger device is connected or not, and also detect if there is a call or not and detect when the call is ended or no one reply, so when the program is run and the Tiger device is not connected so a message is appeared that the device is not connected and then the program disabled. When a user establish a call and the destination IP is wrong or the call is not answered so this part will end the call dialog immediately.

5.3. Make call part

This part concern with the call making process, this process has some criteria's that govern it. When the user wants to establish a call so it will enter the destination IP address and the extension code from the phone hook in the following form: xxx*xxx*xxx*xxx*y# where the x represent the IP digit and y represent the extension code to determine which phone that receive the call.

So this part will detect the tone frequencies from the phone keypad, manage the IP address, replace the '*' by '.' and sub the IP from the extension.

When the user open the phone hook the program will open a call dialog that can receive the IP either from the phone keypad or from the dialog textbox. If the call is ended or the phone hook is closed then the make call part will send a signal to the reset circuit to return to its previous state.

5.4 Receive call part

This part concern with receiving the call. That need to give a message to the user to accept or decline the call and gives some information about the caller,

The receiving part gets the extension code and sends the code as tone frequency to the DTMF receiver that will let the call go to the correct destination phone. Then the receiving part send the command the VoIP processor to generate the ring signal.

When the user open the phone hook the program will send a message the make call part as an indication that the call is accepted. If the call is ended or not accepted then the receiving part will send a signal to the reset circuit to return to its previous state.

5.5 System Context Diagram

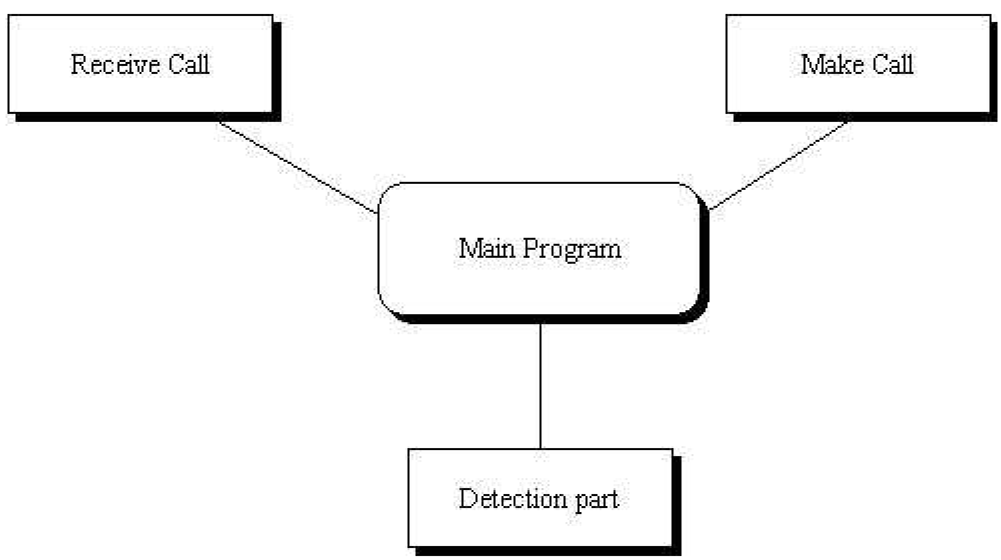


Figure (5.1): System context diagram.

5.6 Make Call Flow Chart

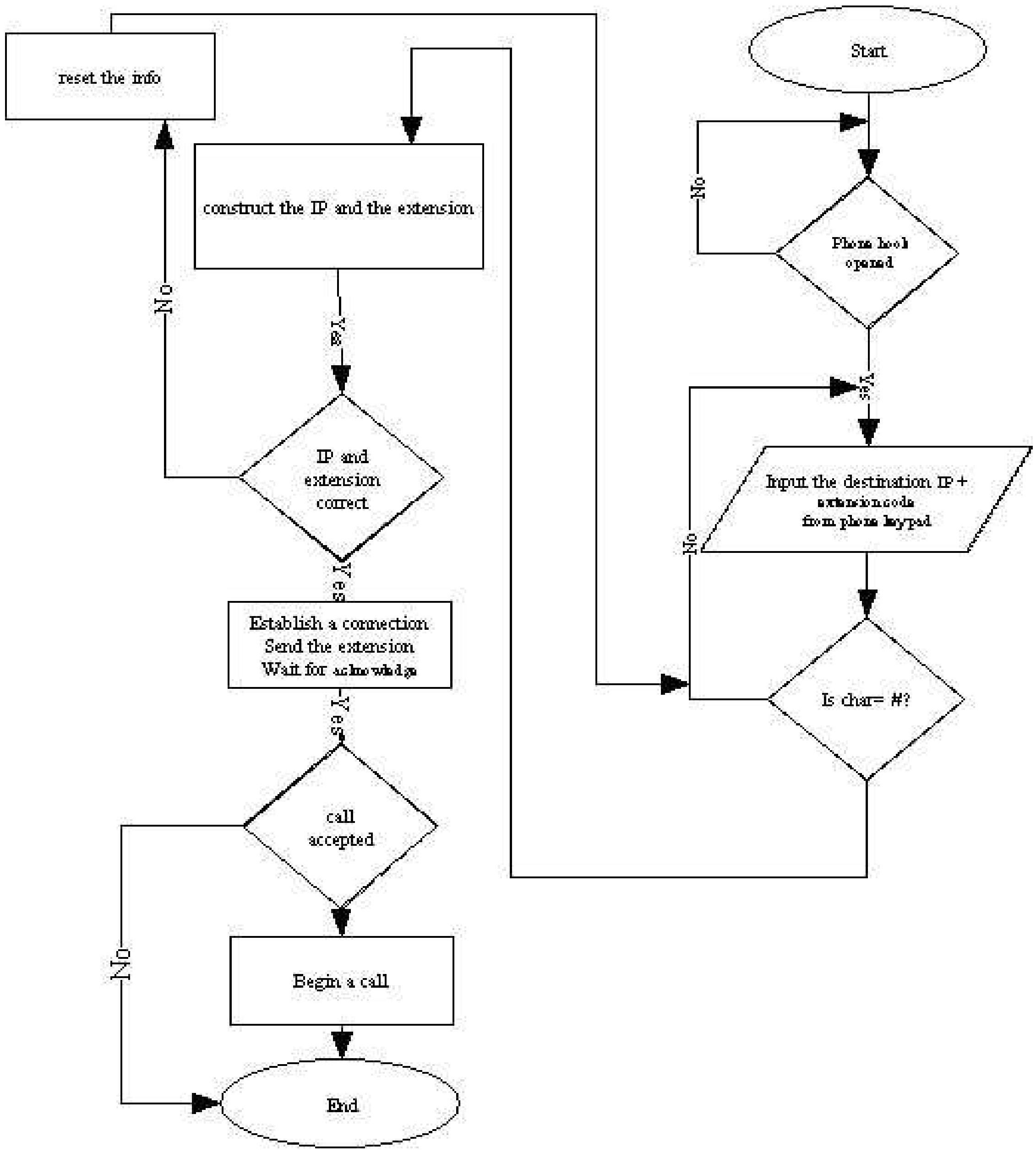


Figure (5.2): make Call Flow Chart

5.7. Receive call Flow Chart

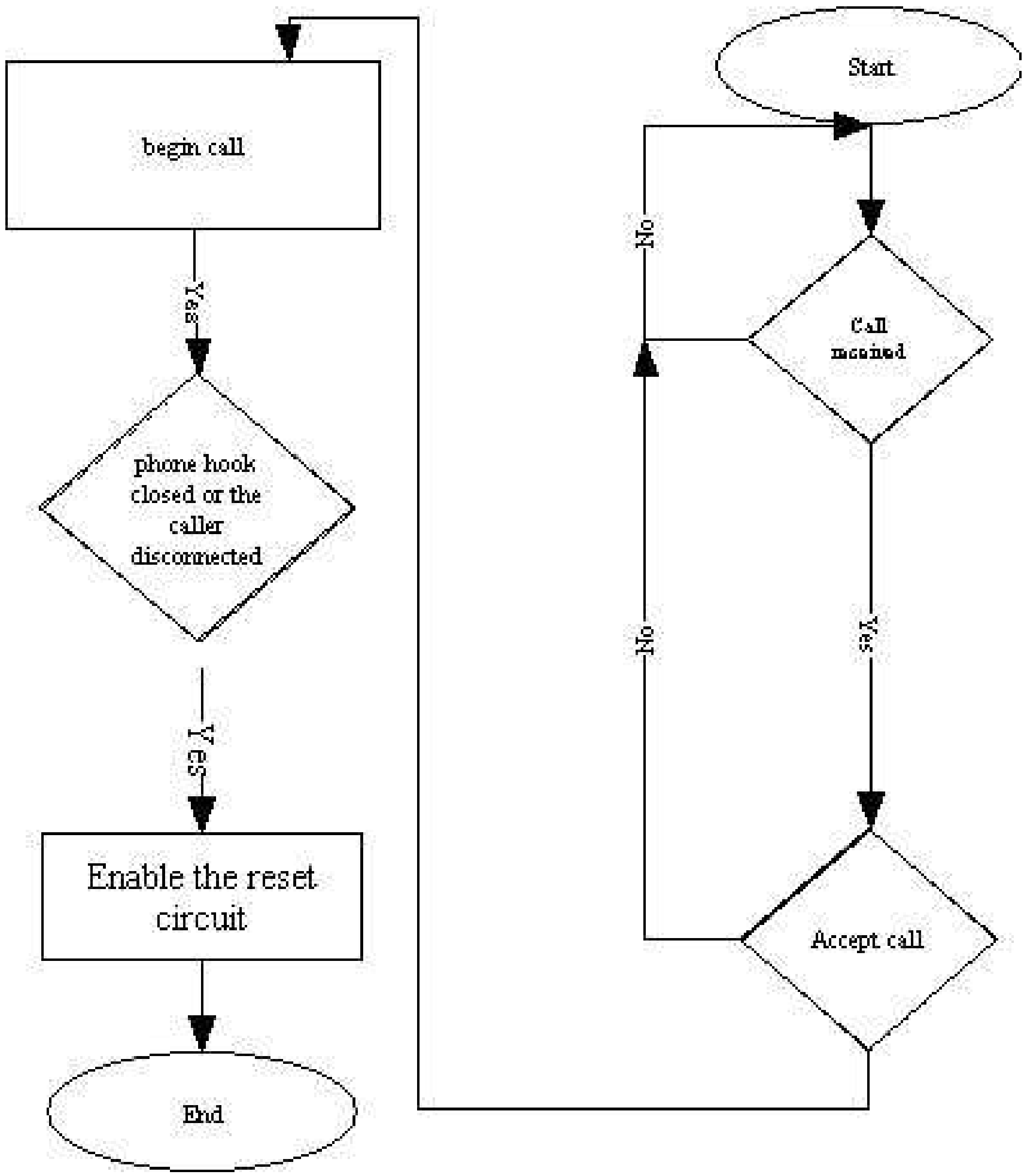


Figure (5.3):Receive Call Flow Chart

5.8 Detector Flow Chart

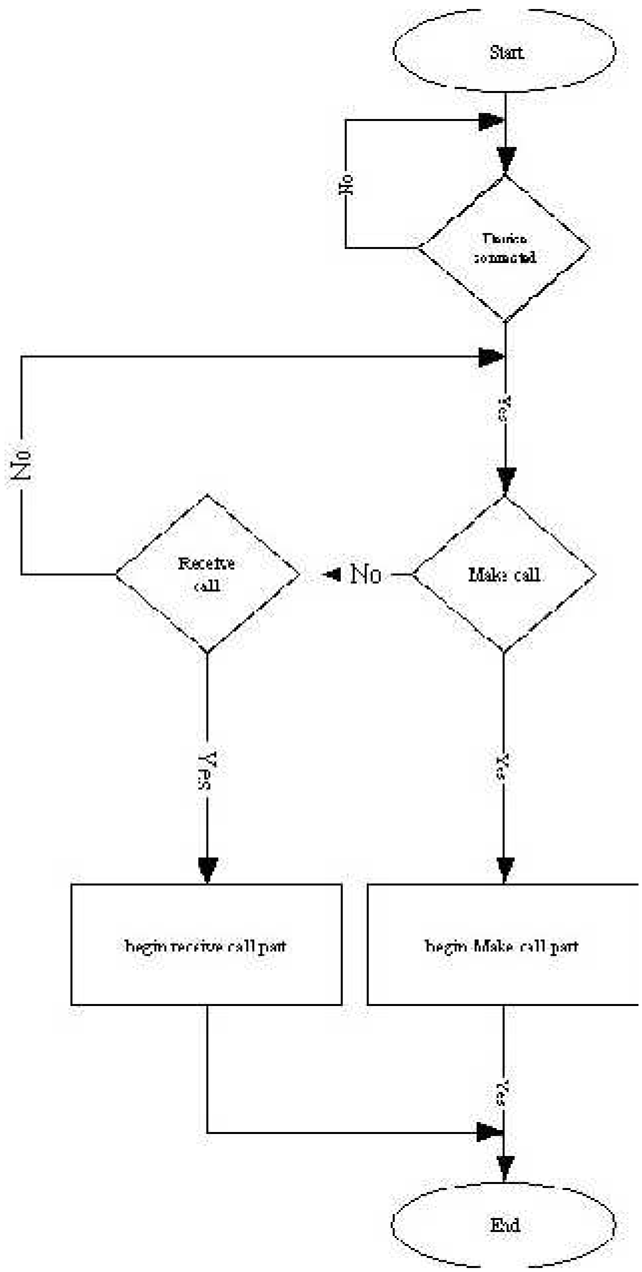


Figure (5.4): Detector Flow Chart

Chapter Six

Chapter Six

System Implementation and Testing

This chapter will be discussing the system implementation and the testing process. The testing process of the system complemented during developing this project, and consists of three stages: hardware testing, software testing, and integration are testing.

In the hard ware unit testing, individual components are tested to ensure that they operate correctly. Each component is tested independently, without other system components. The integration testing on the on the other hand, the overall system is tested. The hardware and the software of this system will be tested; the testing process will be discussed in more details in the next section of this chapter.

6.1 System implementation

The design of the system is implemented as described in the previous chapters. First, independent parts of this project implemented, then these parts are assembled to construct the overall the system implementation. Most of the specifications are met in the implementation, but there was a problem in synchronizing the signals between hardware units because some signals receive the correct signal needed. This problem solved by adding a delay level like adding an OR gate or AND gate.

6.2 Testing

Testing process can be divided into three parts: hardware testing, software testing and integration testing. This will be discussed in details in the next few sections.

6.2.1 Hardware Testing

- Wires connection testing.
- DTMF Receiver testing.
- VoIP processor with USB (TIGER560B)
- Decoding circuit (4_Bit BCD to a Decimal decoder 7442)
- Encoding circuit (10 to 4 encoder 74147)
- Current detecting circuit (Coil of Relay)
- Switching & Isolation circuit (Relays & Transistors to operate the Relay)

6.2.1.1 Wires connection testing.

All IC's connection was tested to ensure that they are connected correctly and ensure that there is no short circuit.

6.2.1.2 DTMF Receiver (CM8870)

It is the first circuit that builds and testing, in the following circuit Figure (6.1) shows how the DTMF Receiver circuit is operated and tested.

The led are use in order to display the code of the tone frequency that inter from the keypad and frequency generator, the keypad and frequency generator obtained from telephone device. It act the input line to this circuit.

The clock circuit of this IC by connecting the end of crystal to CM8870,then connecting them with the oscilloscope to note the clock state and calculating the frequency of the clock , it give 3.57965MHz.

All actual result matching the theoretical result, for more information go to appendix

6.2 .1.3 VoIP processor with USB Interface (TIGER560B)

Before connecting to USB of the computer, serves was installed on this computer in order to examine the plug of this circuit and configuration it.

The voltage level on the out put port of SLIC is 45 voltages that mean the circuit is ready to make a call or receive a call.

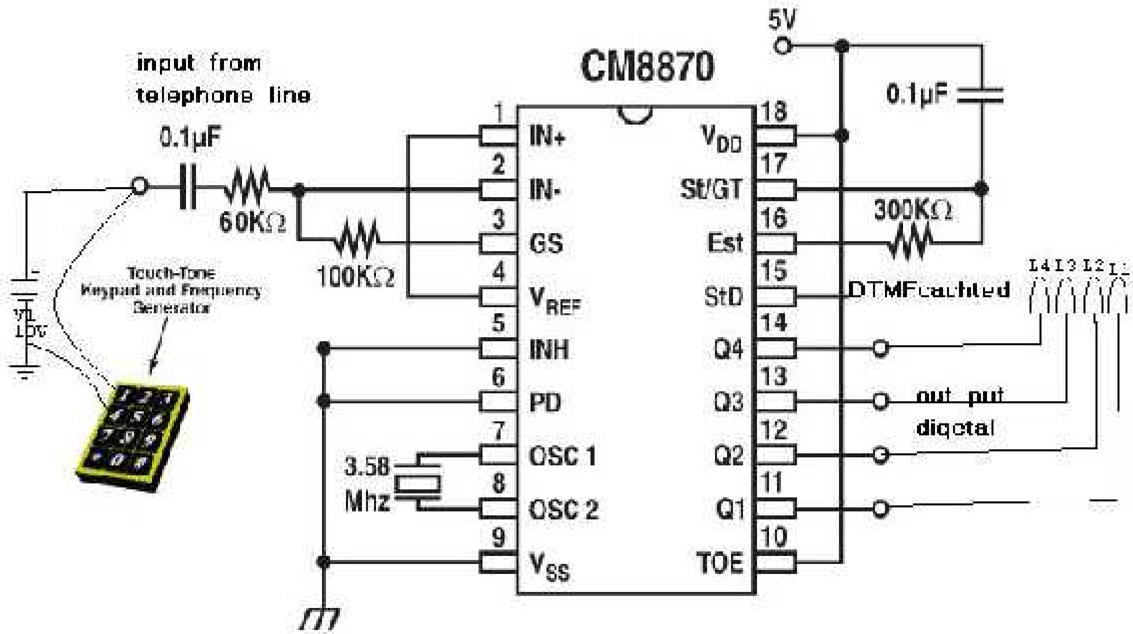


Figure (6. 1): The CM8870 Operating circuit

6.2.1.4 Decoding circuit

(4_Bit BCD to a Decimal decoder 7442) Figure (6.2)

This circuit has two inputs:

- Input from DTMF Receiver circuit, this input is active high, and the valid range of input from 0000 – 1001.
- Input from Phone line decoding circuit, this input is active low.

The outputs of the circuit are ten line active low.

If there is input in the correct rang one line of outputs is active, else all line are active. Figure (6.3).

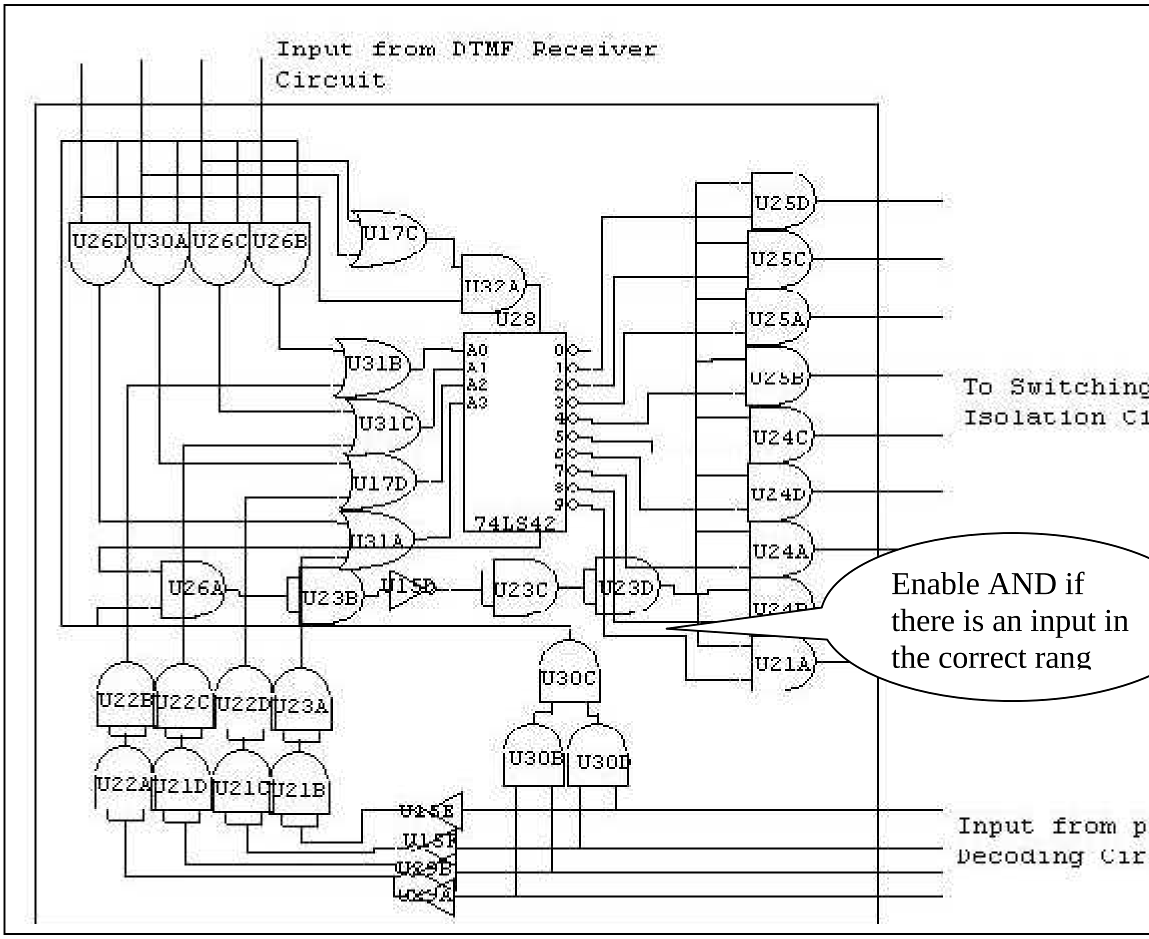


Figure (6.2): Decoding circuit

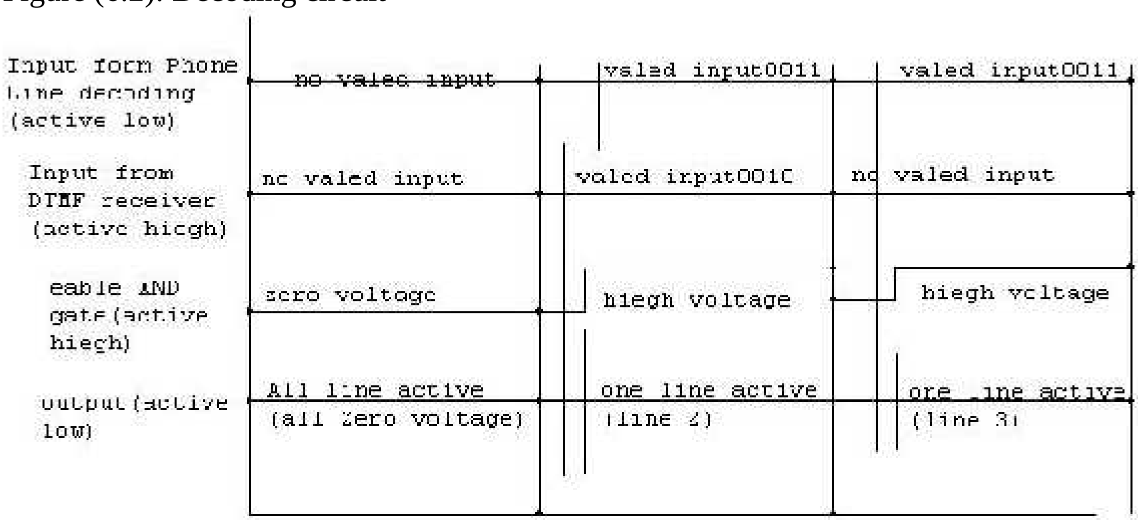


Figure (6.3): Decoding circuit operation

6.2.1.5 Encoding circuit (10 to 4 encoder 74147)

Also the encoding circuit tested to ensure that it works probably, the circuit is built, digital signal applied on the input pins of the encoder through deep switch, then the output get on led ,this make sure that the circuit is work correctly.

6.2.1.6 Current detecting circuit (Coil of Relay)

A telephone line is applied on the input of this circuit, on the out put connect a telephone device ,and on the digital out put applied a led diode.

6.2.1.6 Switching & Isolation circuit

(Relays & Transistors to operate the Relay)

The circuit is built, and then makes a bias to transistor in order to operate the relay, heir the relay used as switch and full isolation.

6.2.2 Software testing

On the service that installed on computer for TIGER560B subroutine is added in order to read the pressed button from the keypad of the telephone device, then this subroutine is testing and it work in the approved manner.

Other subroutines added to get the IPand the number of phone that wonted, and other function to transmit between two computers.

6.2.3 Integration testing

the overall system installed and the hard ware components connected to gather ,in this moment one of the processor is damaged because the electricity become high from the source, but the sub system testing and it works probably.

Chapter seven

Chapter seven

Conclusions and Future Work

7.1 Overview:

For every project there must be many things to be concluded from overall work, these conclusions are derived by the project developers from their intensive study of the project. Also there are many different modification and enhancements that can be added to the project later, these are called future work.

In this chapter we will discuss these two aspects.

7.2 Future work

For this project there are many different enhancements that can be added to the project. These enhancements can be archived by expanding the system interface through some of the following ideas:

The major enhancement can be done by connecting a PSTN central instead of telephone device.

Instead of connecting and programming the system on a PC, using MicrocoWeb Server to do the function of PC.

Enhancement the system to display the caller ID (IP + Extinction).

7.3 Conclusions

VoIP technology is very important in the communication part of our life because of widely use in many application like messenger, net to phone calls, and this calls is very cheep for along distance.

Team project obtained many skills and benefits after completing this project: building a digital circuit as the function needed, how testing the circuit, and how to use hardware with software in order to accomplish the objective.

References

http://www.tjnet.com

http://www.agere.com/client/voip_solutions.html

CISCO, http://www.cisco.com

MICOM, http://www.micom.com

Lucent Technolgies, http://www.lucent.com

Nortel Networks, http://www.nortel.com

VocalTec, http://www.vocaltec.com

Nuera, http://www.nuera.com

Ericsson, http://www.ericsson.com

Qwest, http://www.qwest.com

ITXC, http://www.itxc.com

Motorola, http://www.motorola.com

Delta Threeehttp://www.telephonyworld.com/service/

[Marcus 98] Marcus Gonclaves, "Voice over IP Networks", 1998

[Black 99] Uyless Black, "Voice over IP", 1999

[DP 99] Jonathan Davidson, Jim Peters, "Voice Over IP Fundamentals," Macmillan, November 1999

[KG 99] Matthew Kolon, Walter J. Goralski, "IP Telephony," McGraw Hill, September 1999

[Minoli 98] D. Minoli and E. Minoli, "Delivering Voice over IP Networks," John Wiley, 1998

[Douskalis 99] Bill Douskalis, , "IP Telephony: The Integration of Robust VolP Services," Prentice Hall, 1999

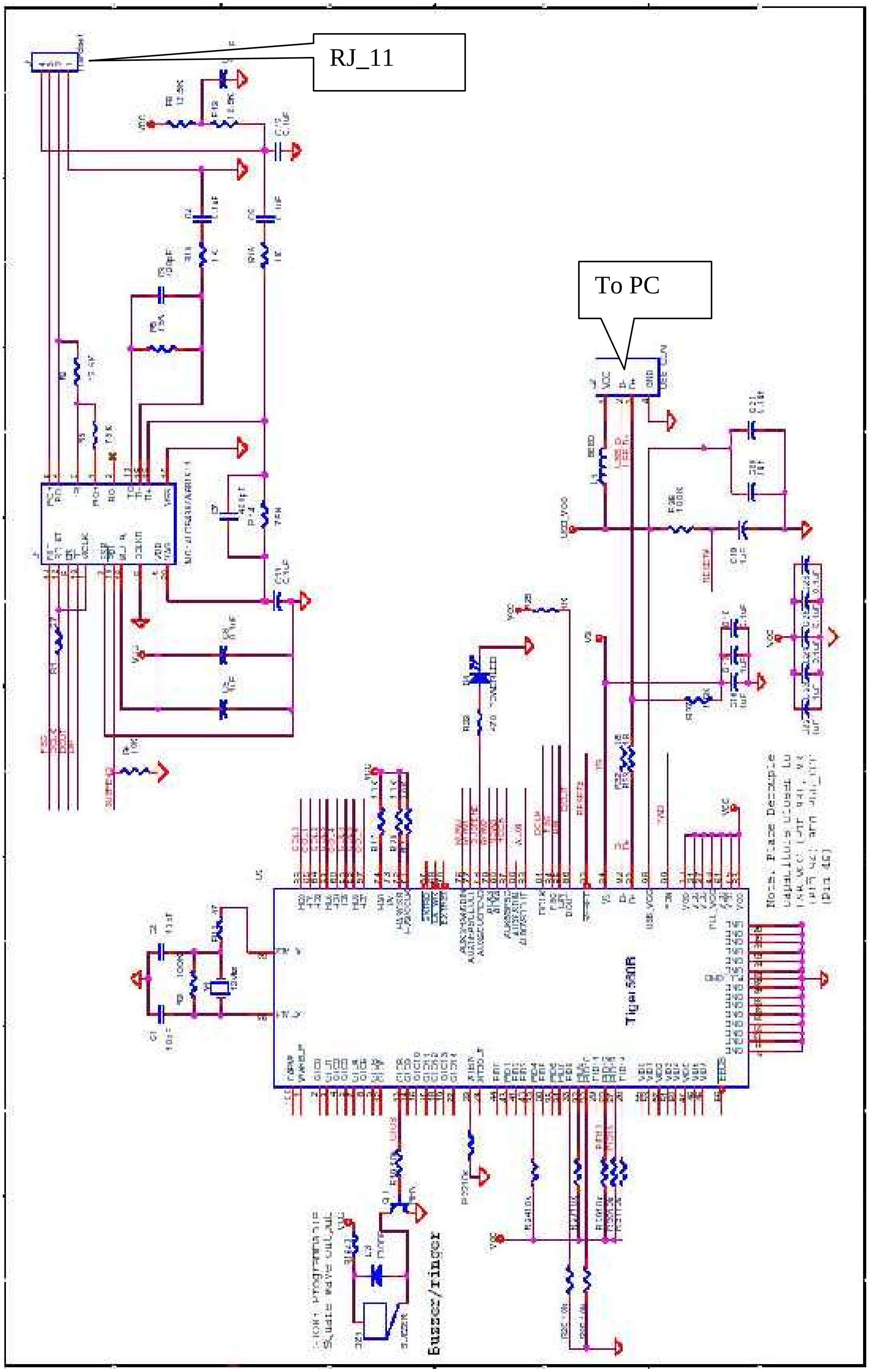
Appendixes

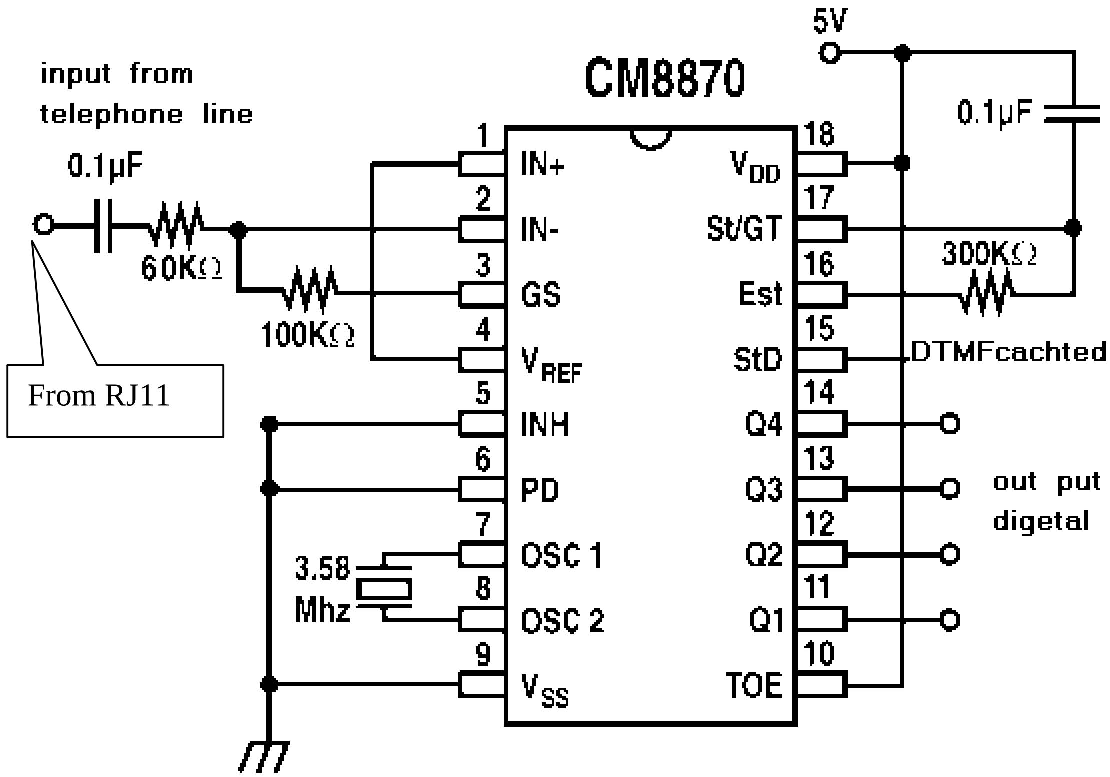
Appendix A

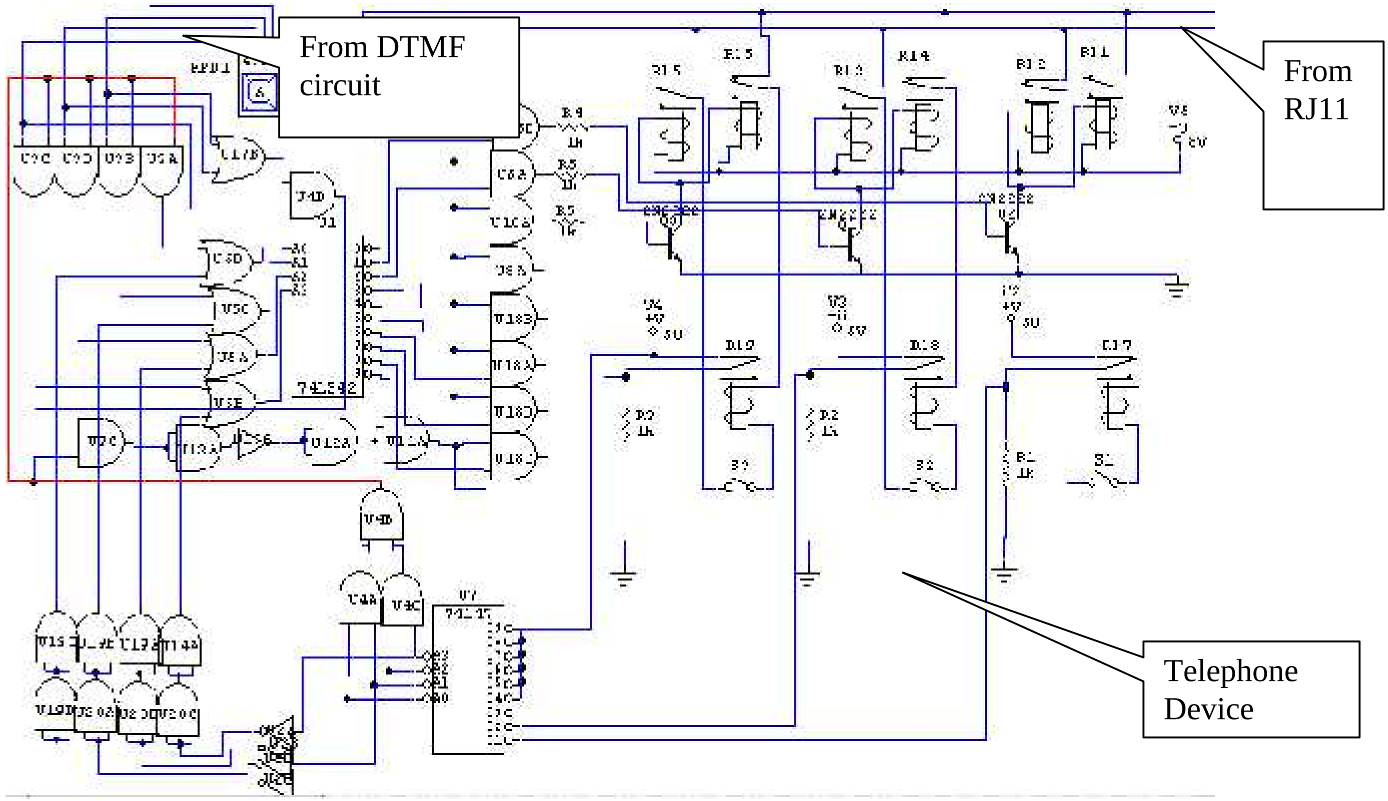
******

Schematic Design

******


RJ_11
To PC


5V
CM8870
input from
telephone line
0.1µF
60KΩ
100KΩ
From RJ11
1 IN+
2 IN-
3 GS
4 VREF
5 INH
6 PD
7 OSC 1
8 OSC 2
9 VSS
18 VDD
17 St/GT
16 Est
15 StD
14 Q4
13 Q3
12 Q2
11 Q1
10 TOE
0.1µF
300KΩ
DTMFcachted
out put
digetal
3.58
Mhz


From DTMF circuit
From RJ11
Telephone Device

Appendix B

*****

Software Program

*****

```
BOOL CTjIpDemoDlg::PrepareTjDevice()
}
//	TjIpProductID	uProductID = TJIP_NONE؛

	if (m_pTjDev(
		return TRUE;  // Device is already opened

	OpenTjIpDevice(m_hInstance, &m_uProductID);	// Open TigerJet560 device
with keypad first

	switch (m_uProductID(
	}
		case TJIP_TJ320PHONE:
			m_pTjDev = new CTj320Phone؛()
			break؛

		case TJIP_TJ560PHONE:
		case TJIP_TJ560PHONE_CUSTOM:
			m_pTjDev = new CTj560Phone؛()
			break؛

		case TJIP_TJ560PHONE_ECHO:
			m_pTjDev = new CTj560Phone(TRUE؛(
			break؛

		case TJIP_TJ560PHONE_SWITCH:
			m_pTjDev = new CTj560PhoneSwitch؛()
			break؛

		case TJIP_TJ560HANDSET_KEYPAD:
			switch (m_KeypadType(
			}
				case KEYPAD_TONG_YA:
					m_pTjDev = new CTj560HandsetKeypad_TY؛()
					break؛

				case KEYPAD_KINGTELT:
					m_pTjDev = new CTj560HandsetKeypad_KT؛()
					break؛

				default:
					m_pTjDev = new CTj560HandsetKeypad؛()
					break؛
			{

			break؛
```

```
		case TJIP_TJ560HANDSET_SWITCH:
			m_pTjDev = new CTj560HandsetSwitch؛()
			break؛

		case TJIP_TJ560HANDSET_ID:
			m_pTjDev = new CTj560HandsetID؛()
			break؛

		case TJIP_TJ560BPHONE_HID:
		case TJIP_TJ560BPHONE_CUSTOM_HID:
			m_pTjDev = new CTj560BPhone_Hid؛()
			break؛

		case TJIP_TJ560BPHONE_SWITCH_HID:
			m_pTjDev = new CTj560BPhoneSwitch_Hid؛()
			break؛

		case TJIP_TJ560BHANDSET_HID:
		case TJIP_TJ560BHANDSET_HID_OKI:
			m_pTjDev = new CTj560BHandset_Hid؛()
			break؛

		case TJIP_TJ560BHANDSET_KEYPAD_HID:
		case TJIP_TJ560BHANDSET_KEYPAD_HID_OKI:
			m_pTjDev = new CTj560BHandsetKeypad_Hid(m_nBuzzerFreq؛(
			break؛
	{

	switch (m_uProductID&0xfffc(
	}
		case TJIP_TJ560PHONE_ECHO:
			m_pTjDev = new CTj560Phone(TRUE؛(
			break؛

		case TJIP_TJ560BPHONE_ECHO_HID:
			m_pTjDev = new CTj560BPhone_Hid(TRUE؛(
			break؛
	{

	if (m_pTjDev(
	}
		m_pTjDev->InitTjHardware();		// Initialize hardware here if
necessary

		m_HookState = m_pTjDev->GetHookState؛()
```

```
		m_pTjDev->StartKeypadScanningTimer(this,
TIMER_CHECK_KEYPAD, TIMER_PERIOD_CHECK_KEYPAD); // Start a timer for
keypad scanning
		LaunchSwitchDlg؛()
	{

	return m_pTjDev ? TRUE : FALSE؛
{

void CTjIpDemoDlg::ProcessNewKey(char cNewKey(
}
	static WORD wDisplayType = 0؛
	CWnd *pFocus = GetFocus؛()
	CWnd *pFocusParent = NULL؛
	BOOL bBroadcast = FALSE؛

	if (pFocus(
		pFocusParent = pFocus->GetParent؛()

	if ((this != pFocus) && (this != pFocusParent((
		bBroadcast = TRUE;  // Broadcast keys only when we are not in focus

	switch((unsigned char)cNewKey(
	}
		case 0:
			break؛

		case VK_ALT_U:
			SendVirtualKey(VK_MENU, FALSE);	// Alt key is dowm
			SendVirtualKey('U', FALSE؛(
			SendVirtualKey(VK_MENU, TRUE);	// Alt key is up
			break؛

		case VK_ALT_TCD:
	// Launch PC2Phone
			OnPc2phone؛()
			Sleep(500؛(
			SendVirtualKey(VK_MENU, FALSE);	// Alt key is dowm
			SendVirtualKey('T', FALSE؛(
			SendVirtualKey('C', FALSE؛(
			SendVirtualKey('D', FALSE؛(
			SendVirtualKey(VK_MENU, TRUE);	// Alt key is up
			break؛

		case VK_CALLGATEWAY:
			OnGateway؛()
```

```
				break؛

			case VK_HANGUPGATEWAY:
				HangupCallPc2pc؛()
				if (!m_pTjDev->HasHardwareDTMF())	// ٥٦٠handset only
					SendMsgToBuddyList( ID_ESCAPE, 0, 0, TRUE ؛(
				break؛

			case VK_OFF_HOOK:		// Proslic only: answer the phone
				OnOffHook؛()
				SendMsgToBuddyList( ID_OPEN, 0, 0, TRUE؛(
				SendMsgToSpecialDialListener( ID_OFF_HOOK_KEY, NULL,
0, TRUE؛(
				break؛

			case VK_ON_HOOK:		// Proslic only: hangup the phone
				OnOnHook؛()
				SendMsgToBuddyList( ID_HIDE, 0, 0, TRUE؛(
				SendMsgToSpecialDialListener( ID_ON_HOOK_KEY, NULL, 0,
TRUE؛(
				break؛

			case VK_SHOWBUDDY:
//					Pc2pc(m_hWnd, FALSE؛(
//					OnCallIp؛()
				SendMsgToBuddyList( ID_SWITCH_DISPLAY, 0, 0, TRUE؛(
				break؛

			٧.٠٦ //
			case VK_ACCEPT_CALL:	// ٥٦٠handset only
				if (bBroadcast(
					SendVirtualKey(VK_RETURN, FALSE؛(
				SendMsgToPc2PcCall(ID_AccetpIncomingCall, NULL, 0,
TRUE؛(
				SendMsgToBuddyList(ID_RETURN, 0, 0, TRUE ؛(
				break؛

			case '3':		// new 7.06
			case '6:'
				if (!m_pTjDev->HasHardwareDTMF(()
				}
					SendMsgToPc2PcCall( ID_KEYCODE, (void
*)&cNewKey, sizeof (cNewKey), TRUE؛(

	SendMsgToBuddyList((cNewKey=='3')?ID_UP:ID_DOWN, 0, 0, TRUE ؛(
				{
```

```
				SendMsgToSpecialDialListener( ID_INPUT_KEY, (void
*)&cNewKey, sizeof (cNewKey), TRUE؛(
				if (bBroadcast(
					SendVirtualKey(cNewKey, FALSE؛(
				break؛

			case '5':		// ٧.٠٦
				if (!m_pTjDev->HasHardwareDTMF(()
				}
					SendMsgToPc2PcCall( ID_KEYCODE, (void
*)&cNewKey, sizeof (cNewKey), TRUE؛(
					SendMsgToBuddyList( ID_SETFOCUS, 0, 0, TRUE ؛(
				{
				SendMsgToSpecialDialListener( ID_INPUT_KEY, (void
*)&cNewKey, sizeof (cNewKey), TRUE؛(
				if (bBroadcast(
					SendVirtualKey(cNewKey, FALSE؛(
				break؛

			case '0':				// New 2.10
			case '1:'
			case '2:'
			case '4:'
			case '7:'
			case '8:'
			case '9:'
			case VK_MULTIPLY:
			case VK_POUND:
			}
				char		cKeySend؛

				cKeySend = cNewKey؛

				if ((unsigned char)cNewKey == VK_MULTIPLY(
					cKeySend؛'*' =

				if ((unsigned char)cNewKey == VK_POUND(
					cKeySend؛'#' =

				if (!m_pTjDev->HasHardwareDTMF(()
					SendMsgToPc2PcCall(ID_KEYCODE, (void
*)&cKeySend, sizeof (cKeySend), TRUE؛(

				SendMsgToSpecialDialListener( ID_INPUT_KEY, (void
*)&cKeySend, sizeof (cKeySend), TRUE؛(
```

```
            if (bBroadcast(
                SendVirtualKey(cNewKey, FALSE؛(
        {
        break؛

        default:
            if (bBroadcast(
                SendVirtualKey(cNewKey, FALSE؛(
            break؛
    {
{
```

Appendix C

*****

Data Sheets

*****