Interactive Voice Response System based College Automation

Submitted in partial fulfillment of the Degree of

Bachelor of Technology

Enrollment No. - 081032,081080,081093

Name of Students - Karan Puri ,Sourabh Gaba ,Shubham

Name of Supervisor - Mr. Munish Sood

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY,

WAKNAGHAT

CERTIFICATE

This is to certify that the work titled "**Interactive Voice Response System based College Automation**" submitted by "**Karan Puri ,Sourabh Gaba and Shubham**" in partial fulfillment for the award of degree of B. Tech, of Jaypee University of Information Technology, Waknaghat has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of Supervisor ....

Name of Supervisor ....Munish Sood....

Designation ....Lecturer..........

Date ....2/6/2012......

ACKNOWLEDGEMENT

We wish to express my deep sense of gratitude to our Project Guide, **Mr. Munish Sood,** for his able guidance and useful suggestions, which helped us in completing the project work to such extent.

We would also like to thank our **Project Pannel,** whose suggestions have been of great usefulness in the conduct of our project work.

ABSTRACT

In today's competitive world any business must build flexible systems that adapt easily to evolving requirements of the critical business processes. IVRS is one such system that transforms the traditional business model into customer centric model. IVRS is historically interactive speech memory driven that walk the caller through a series of prompts where they respond to questions by pressing the combination of one or more buttons of the phone keypad.

The decision tree associated with the prompts and the responses will route the caller to information they desire. These IVRS systems are typically utilized to check bank account balance, buy and sell stocks, check the show times for your favorite movie.

In telephony, Intelligent Voice Response, or IVR, is a phone technology that allows computer to detect voice and touch tones using a normal phone call. The IVR system can respond with pre-recorded or dynamically generated audio to further direct callers on how to proceed. IVR systems can be used to control almost any function where the interface can be broken down into a series of simple menu choices. Once constructed IVR systems generally scale well to handle large call volumes.

CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 BASICS

What can be done using an Inter voice System?

When connecting an Inter-voice system into telephone lines (either analog lines or digitalT1/E1 trunks), the applications can handle either incoming or outgoing calls and then perform the following voice processing features:

- DTMF or pulse tone input
- Provides unlimited pre- recorded voice messages
- Live recording of customer messages
- Accesses or stores information to and from the back-end host, database or the Internet
- Uses leading speech recognition technology to process either spoken words .

Who should use Inter voice?

Using Inter voice hardware and software, we have developed voice automation applications, which can:

- Transfer the customer calls to the right people to handle
- Provide the most updated product or service information
- Record customer messages for follow up later
- Perform automated transaction processing without human interven

What are interactive voice response (IVR) systems?

Interactive Voice Response (IVR) systems allow callers to interact with thecommunications system over the telephone. IVR is used to enable the caller to retrieveinformation from a database, enter information into a database, or both. IVR systemsallow the user to efficiently exchange information, reducing clerical processing.

How It Works?

An IVR system talks to callers following a recorded script. It prompts a response to thecaller and asks him to respond either verbally or by pressing a touchtone key, andsupplies the caller with information based on responses made.

What are important features of IVR systems?

IVR system should store responses made by callers.

Should be able to provide different responses to callers based on time of day called.
Should be able to capture either touch-tone or voice responses by callers.

1.2AIM OF THE PROJECT

The project aims at developing an Interactive Voice Response System based College Automation through which students can access their academic information,as on the webkiosk ,through a mobile phone

Interactive voice response refers to technology supporting the interaction of customer with the service provider generally over the telephone lines.

When a person wants to access any of the services of the Interactive Voice Response System, he presses a number through his telephone keypad , and the appropriate action assigned to key is performed

CHAPTER 2
DETAILS OF TOPIC AND ANALYSIS

2.1 PRINCIPLE OF IVRS

Interactive voice response refers to technology supporting the interaction of customer with the service provider generally over the telephone lines.

When a person wants to access any of the services of the Interactive Voice Response System, he presses a number through his telephone keypad. The pressed number appears across the line and the ring detector circuit senses this ring. After a specified number of rings the relay is activated through the microcontroller, which in turn connects the line to DTMF decoder.

The activation of relay causes the number pressed to appear across the DTMF decoder.The decoder decodes the number pressed and then the decoder output is passed throughthe microcontroller to the computer.
Now, when the caller presses a number, the number pressed is decoded by the DTMF decoder and passed to the computer through the microcontroller using MAX232.
The computer recognizes the number and accesses the particular file from the database to output the voice message.

The output voice is passed through the voice card where the digitized serial data isconverted into analog voice form and passed to the line. The caller gets the informationthrough the line.

2.2 SEQUENCE FOLLOWED IN THE IVRS SERVICE

- Caller dials the IVRS service number.
- The computer waits for a specified number of ringing tones at the end of which, the connection is established.
- The connection is established by lifting the handset of telephone base from ONHOOK condition.
- Now, a pre-recorded voice greets the caller conforming that the number dialed corresponding to the particular service.
- Next, the menu is presented to the caller again in the voice form, giving him the various options to choose from.
- If the information to be relayed back is confidential, then the system may even ask the dialer, to feed in a password number.

- The database is accordingly referenced and the necessary information is obtained.
- Next, the same information is put across to the user in voice.
- The caller generally given the option to :

a.Repeat whatever information was voiced to him.

b.Repeat the choices.

c.Break the call by restarting ON-HOOK condition

2.3 GENERAL DESCRIPTION ABOUT TELEPHONY

Any telephone set will always be in any of the conditions mentioned below:

2.3.1 ON-HOOK

It is the state whenever telephone handset is placed on the cradle. During this state, thetelephone line is open circuit with the exchange and the voltage of –48 V is available oneach telephone line from the exchange.

2.3.2 OFF-HOOK

This is the state whenever telephone handset is displaced from the cradle. During thisstate the voltage level is between± 5V to± 12 V. The telephone OFF – HOOK resistanceis typically 600Ω .

2.3.3 SIGNALING TONES

- **Dial tone:**

This tone indicates that the exchange is ready to accept dialed digits from the subscriber. The subscriber should start dialing only after hearing the dial tone. Otherwise, initial dialed pulse may be missed by the exchange that may result in the call landing on thewrong number. The dialed tone is 33 Hz or 50 Hz or 400 Hz continuous tones.

- **Ring tone:**

When the called party is obtained, the exchange sense out the ringing current to the telephone set of the called party. This ringing current has the familiar double ring pattern. Simultaneously, the exchange sends out the ringing tone to the calling subscriber, which has the pattern similar to that of ringing current, the two rings in the double ring pattern are separated by a time gap of 0.2s and two double rings patterns by a time gap of 2s.Theburst has duration of 0.4s. The frequency of the ringing tone is 133 Hz or 400 Hz.

- **Busy tone:**

Busy tone is bursty 400 Hz signal with silence period in between. The burst and silence duration has the same value of 0.75s. A busy tone is sent out to the calling subscriber whenever the switching equipment or junction line is not available to put through the call or called subscriber line is engaged.

- **Number unobtainable tone:**

The number unobtainable tone is a continuous 400 Hz signal. This tone may be sent to the calling subscriber due to a variety of reasons. In some exchanges this tone is 400 Hz intermittent with 2.5s ON period and 0.5s OFF period.

- **Routing tone:**

The routing tone or call – in – progress tone is 400 Hz or 800 Hz intermittent patterns. In an electromechanical system it is usually 800Hz with 50% duty ratio and 0.5s ON-OFF period. In analog electronic exchange it is 400 Hz pattern with 0.5s ON period and 0.5sOFF period. In digital exchange it has 0.1s ON-OFF period at 400 Hz

- **TOUCH –TONE KEY PAD**

Touching a button generates a ' tone', which is a combination of two frequencies, onefrom lower band and other from upper band. For e.g. pressing push button '7' transmits852 and 1209 Hz.

2.4 TELEPHONE INTERFACE SECTION

It consists of following subsections:

2.4.1 Ring Detector Section

Ring detector circuit does the function of detecting the ring activating signals and then counts the number of rings.

2.4.2 Ring activating signals

This is send by telephone exchange to the subscriber. This signal causes an audio tone in the subscriber's telephone set. This ring tone is an alarming signal, which diverts the attention of the subscriber towards the instrument. The ring signal produced at the central office is composed of a 10v ac, 400Hz signal that is always present on the telephone line with the handset in ON-HOOK position.

The ring-activating signal is |ON for 0.2 sec and the subscriber can hear the sound of ring in that duration of time. For next 0.4 sec the ring-activating signal goes OFF. Now the subscriber can't hear the sound. Again this repeats for six times with the pause of 2 sec. Thus the subscriber hears six rings.

CHAPTER 3
BLOCK DIAGRAM AND CIRCUIT DIAGRAM OF IVRS

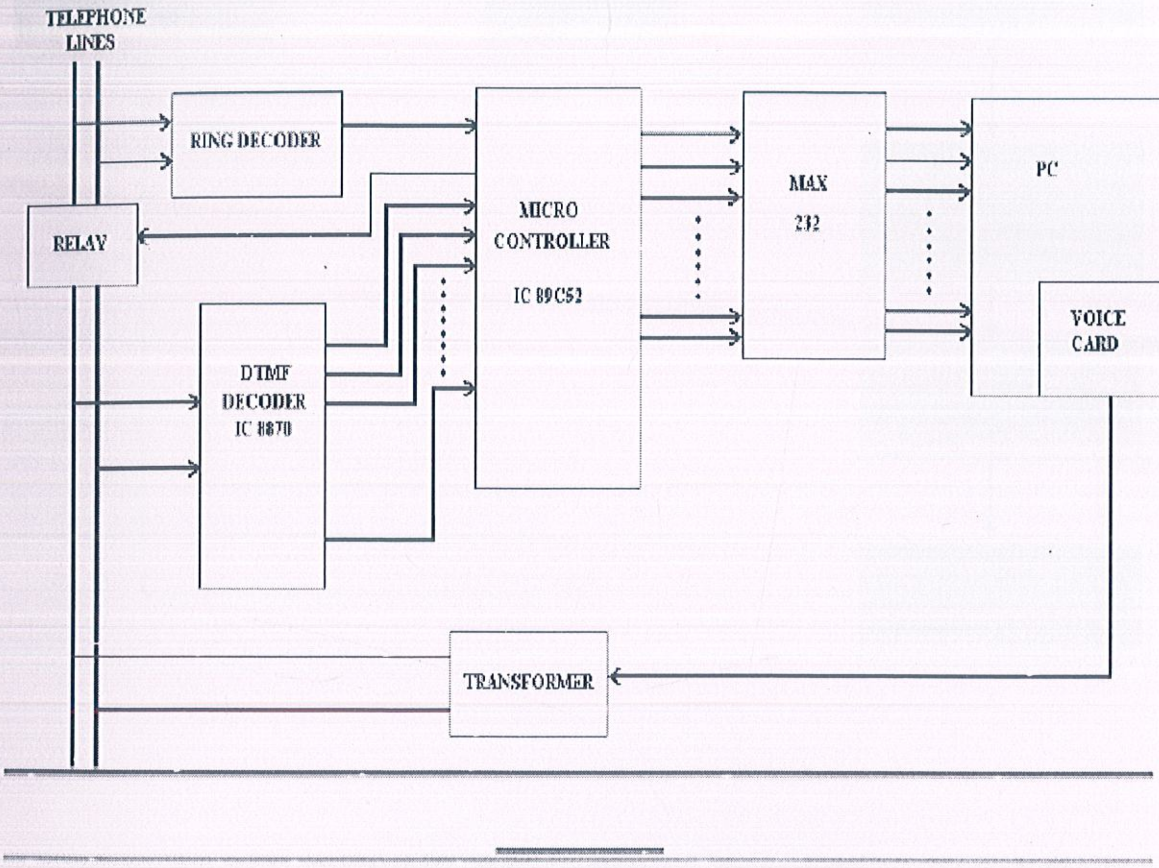
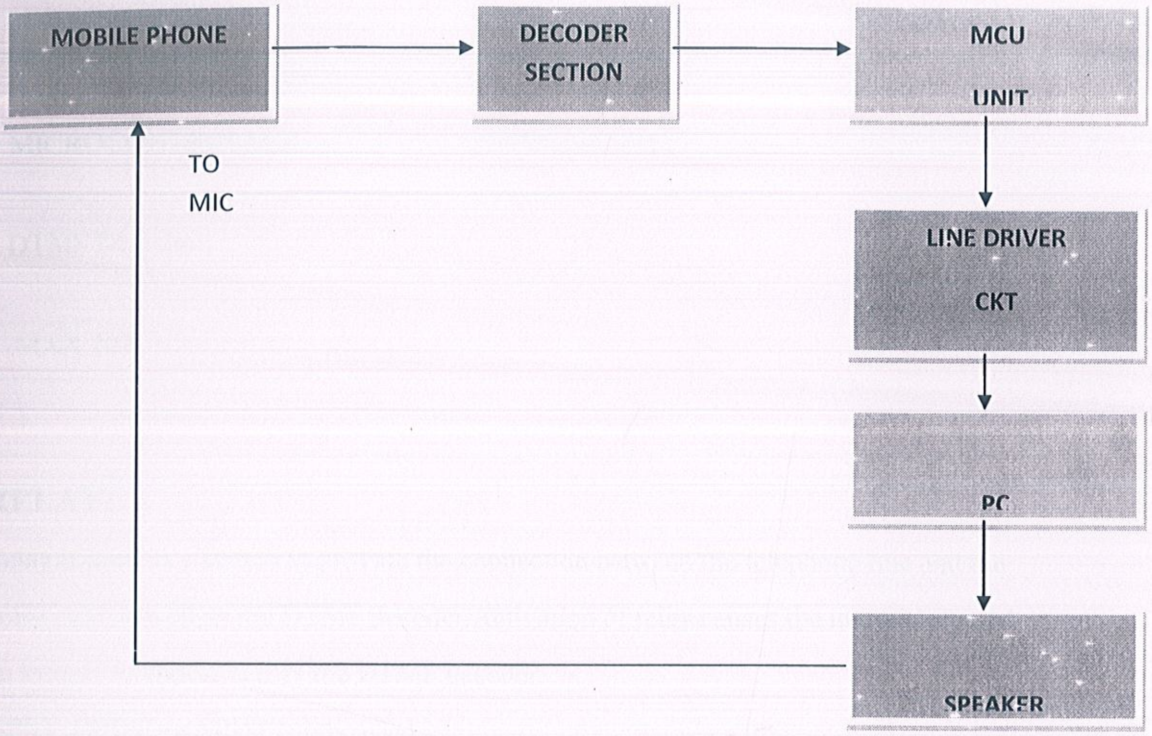


Fig. 3.1Block Diagram of IVR System

3.1 BLOCK DIAGRAM OF IVRS



3.2 BLOCK DIAGRAM EXPLANATION

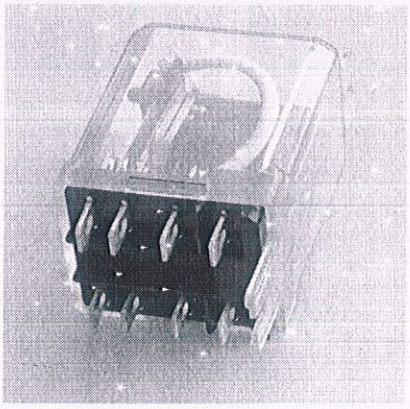
The main blocks of my system are as follows:

- Relay
- MICROCONTROLLER
- DTMF Decoder
- MAX 232

RELAY:

Relay is used as a switch to provide the connection between the telephone line and the voice card as well as the DTMF decoder.Activation of relay causes the number pressed on keypad to appear across the DTMF decoder.

SPST, DPDT, or SPDT can be used



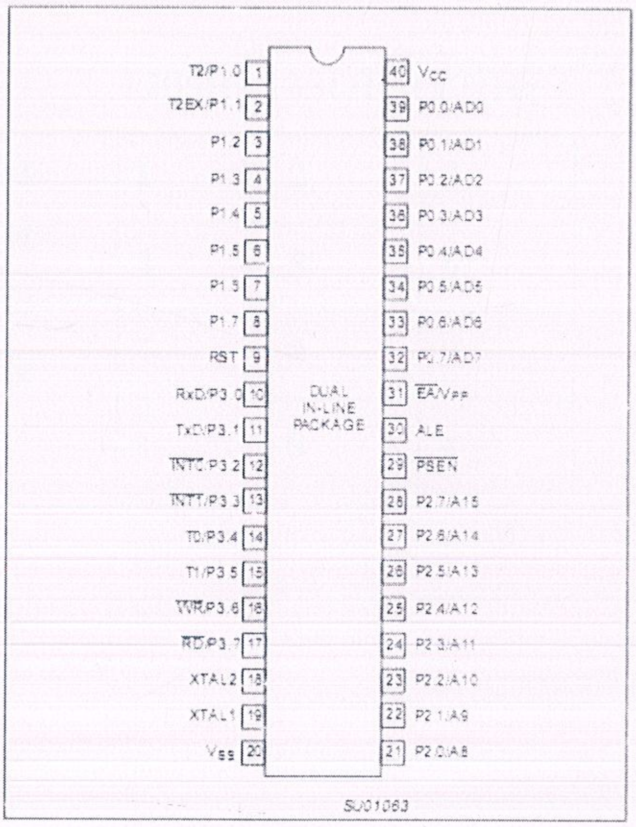
89C52 Microcontroller:

The function assigned to the particular key (dialled by the user) ,decoded by the DTMF decoder, is programmed into the microcontroller and the assigned function is completed using the data stored in PC.

After a specified number off rings the relay is activated through the microcontroller, which in turn connects the line to DTMF decoder.

- It is the central controller of the whole project.
- It scans all channels continuously.
- It transfers the logical values serially to the PC.

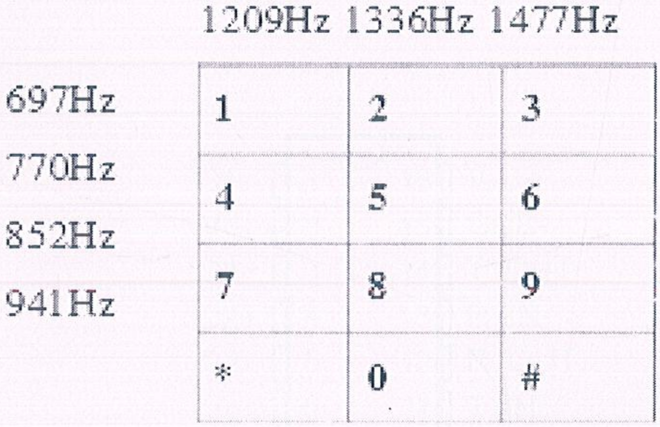
89C52 can be used.



DTMF decoder (8870):

The number pressed on keypad is decoded by the DTMF decoder and passed to the computer through the microcontroller .In DTMF signaling, two frequencies are allocated to each digit in the push button keypad.

The main function of the DTMF decoder is to detect the two frequencies and until and unless these two frequencies allocated for a particular digit are obtained, that particular digit will not be recognized by the DTMF decoder. The decoder uses digital counting technique to detect and decode all 16 DTMF tone pairs into 4 bit code.



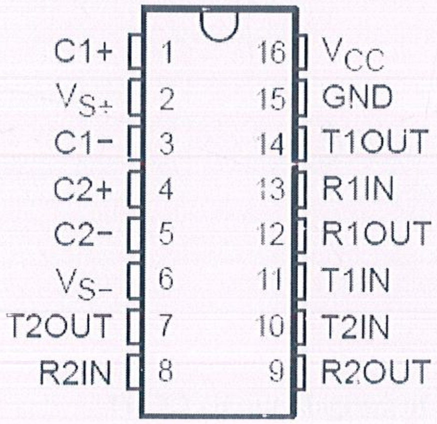
	1209Hz	1336Hz	1477Hz
697Hz	1	2	3
770Hz	4	5	6
852Hz	7	8	9
941Hz	*	0	#

MAX232:

The **MAX232** is an electronic circuit that converts signals from a serial port to signals suitable for usage in e.g. microprocessor circuits.

When communicating with various micro processors one needs to convert the RS232 levels down to lower levels, typically 3.3 or 5.0 Volts. Serial RS-232 communication works with voltages -15V to +15V for high and low. On the other hand, TTL logic operates between 0V and +5V . Modern low power consumption logic operates in the range of 0V and +3.3V or even lower.

Thus the RS-232 signal levels are far too high TTL electronics, and the negative RS-232 voltage for high can't be handled at all by computer logic. To receive serial data from an RS-232 interface the voltage has to be reduced. Also the low and high voltage level has to be inverted. The level converter uses a Max232 and five capacitors.



3.3 CIRCUIT DIAGRAM OF IVRS

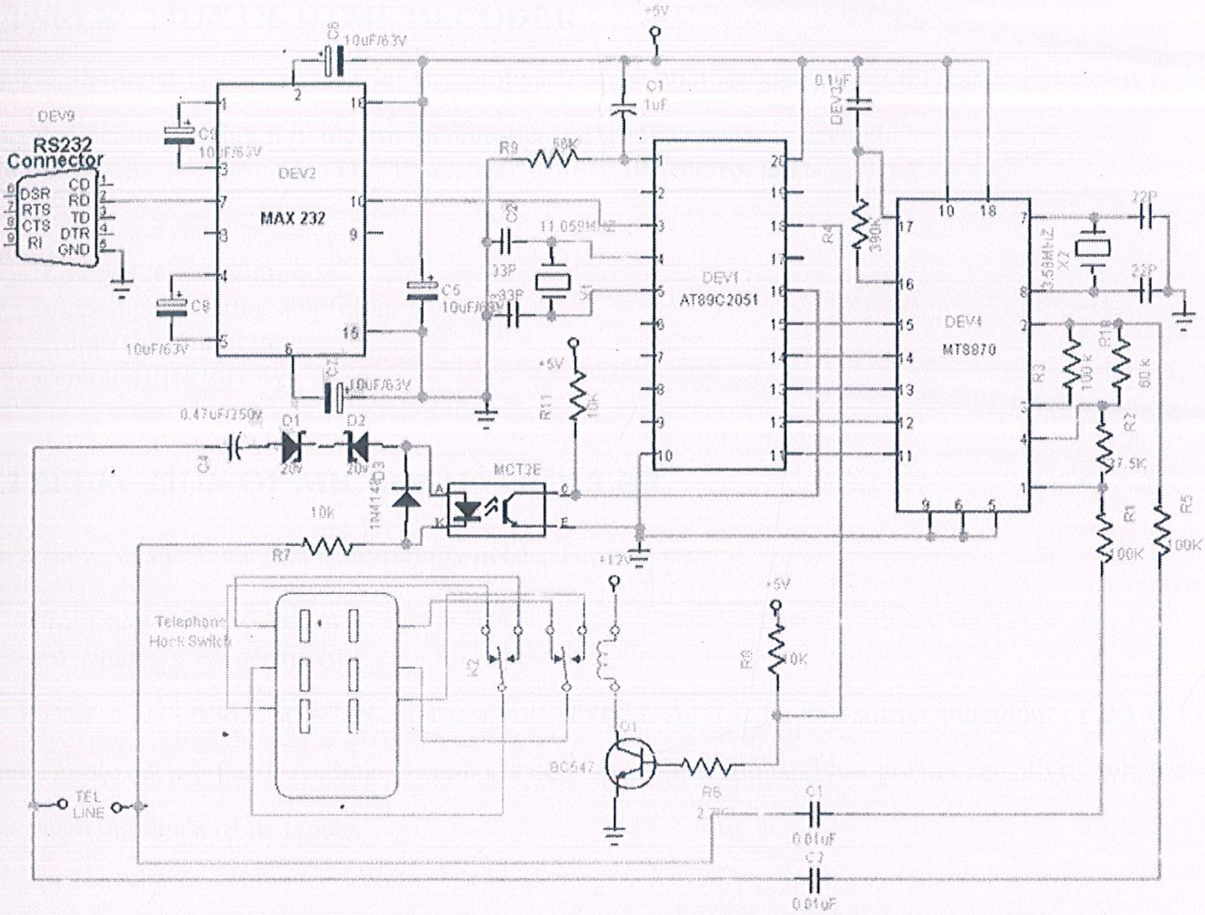


Fig 3.2 circuit diagram of ivrs

CHAPTER 4
SYSTEM DESIGN

4.1 SELECTION OF DTMF DECODER

This is the most important task for my project. As the number pressed by the caller is needed to be decoded before sending it to the microcontroller, a DTMF decoder is needed.
For our project, we have used DTMF decoder IC8870.Its features are :-

- Complete DTMF Receiver
- Low power consumption
- Internal gain setting amplifier
- Adjustable guard time
- Central office quality

4.2 SELECTION OF MICROCONTROLLER

To achieve this task we first analyzed my needs. Those were..

- To achieve fast operation
- To achieve great sensitivity

So for these we decided to select microcontroller**89C52.** As it is having sufficientamount of RAM for such simple operations of reading. It supports high speed operationand has greater sensitivity when put into scanning mode of its inputs.

4.3 SELECTION OF MAX 232

Actually MAX 232 is used because we wanted to have serial communication between microcontroller and computer. This supports short distance serial communication.

We are using MAX 232 in IC format due to ready made availability of this.Some capacitors have to arrange along peripherals pins of this IC which we haveconnected in our actual circuit.

Need of these capacitors is due to fact that MAX 232datasheet shows need of such capacitors in order to pull data from one point to another point.

CHAPTER 5

WORKING OF THE PROJECT

When the telephone is in the idle condition, the voltage will be -48V.

- When the ringing occurs, it will be 125V peak to peak AC signal superimposed on -48V.

- The opto isolator is used to isolate the microcontroller from high voltage AC signals and it consists of GaAs infrared emitting diode optically coupled to a monolithic silicon phototransistor.

- The microcontroller will detect the ring through the port 1.5 and it will count the number of rings.

- After a fixed number of rings, the microcontroller will send a signal to the relay and then the automatic off-hooking of the telephone takes place.At the same time, microcontroller will transmit '#' to the computer which is an indication to play the 'Welcome' message.

- The relay used is DPDT type and after automatic off-hooking takes place, the relay connects the telephone lines to the decoder IC 8870 and isolation transformer.

- The transformer used is a line transformer used to isolate voice card from high voltages.

- As the telephone lines are connected to the voice card, the caller gets to hear the stored messages and asks the caller to enter the roll number of the student whose result is to be known. After the caller dials the roll number from the touch tone keypad of his telephone, that number will be decoded by the decoder IC 8870 and the decoded information will be sent to the computer via the microcontroller.

- Computer on receiving the decoded information will check the database to access the result of the student whose roll number is entered.

- Then the computer will send the desired information to the voice card and the caller will get to hear the result of the student on his telephone through the voice card

CHAPTER 6

Description of microcontroller AT89S52

6.1 Features

- Compatible with MCS-51® Products
- 8K Bytes of In-System Programmable (ISP) Flash Memory

– Endurance: 1000 Write/Erase Cycles

- 4.0V to 5.5V Operating Range
- Fully Static Operation: 0 Hz to 33 MHz
- Three-level Program Memory Lock
- 256 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Three 16-bit Timer/Counters
- Eight Interrupt Sources
- Full Duplex UART Serial Channel
- Low-power Idle and Power-down Modes
- Interrupt Recovery from Power-down Mode
- Watchdog Timer
- Dual Data Pointer
- Power-off Flag

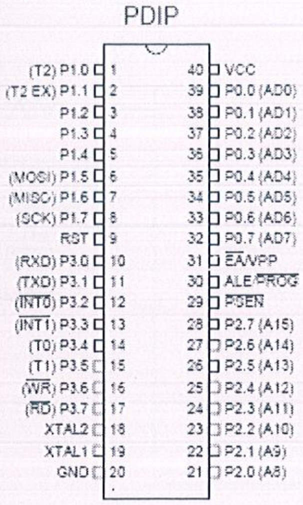
6.2 Description

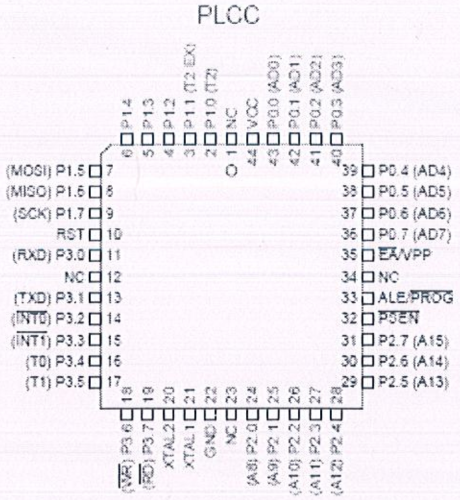
The AT89S52 is a low-power, high-performance CMOS 8-bit microcontroller with 8K bytes of in-system programmable Flash memory. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with in-system programmable Flash on a monolithic chip, the Atmel AT89S52 is a powerful microcontroller which provides a highly-flexible and cost-effective solution to many embedded control applications.

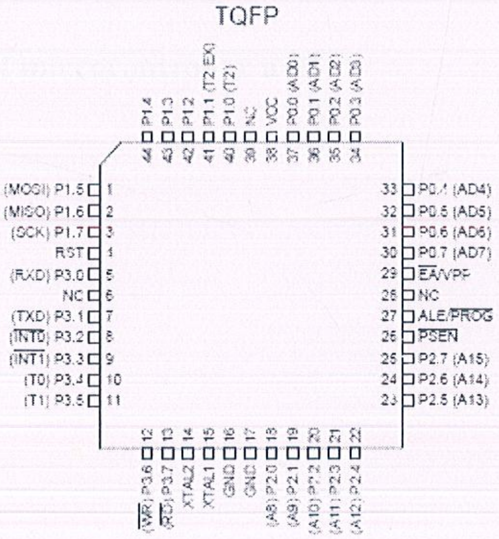
The AT89S52 provides the following standard features: 8K bytes of Flash, 256 bytes of RAM, 32 I/O lines, Watchdog timer, two data pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S52 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes.

The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset

6.3 PIN CONFIGURATION AND BLOCK DIAGRAM







BLOCK DIAGRAM:-

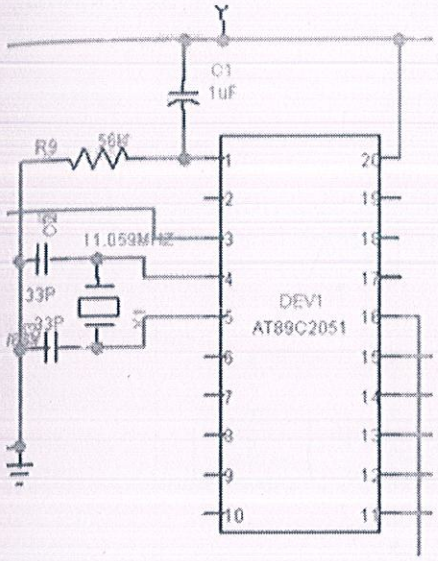
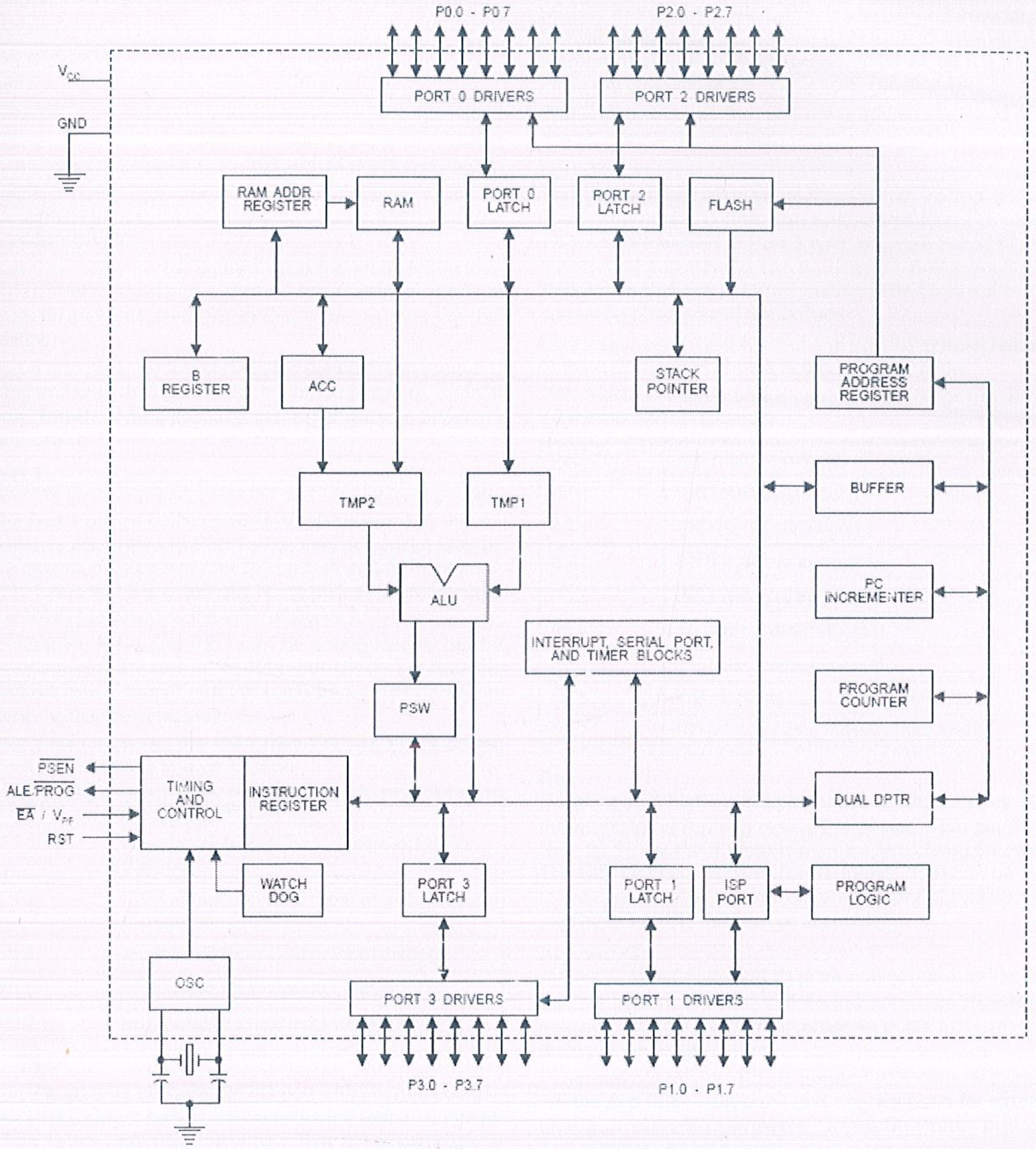


Fig 6.1 block diagram of microcontroller unit

6.4 Internal Block Diagram of 89s52

Block Diagram



6.5

Pin Description

VCC

Supply voltage.

GND

Ground.

Port 0

Port 0 is an 8-bit open drain bidirectional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1

Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

In addition, P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively, as shown in the following table.

Port 1 also receives the low-order address bytes during Flash programming and verification.

Port Pin	Alternate Functions
P1.0	T2 (external count input to Timer/Counter 2), clock-out
P1.1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)
P1.5	MOSI (used for In-System Programming)
P1.6	MISO (used for In-System Programming)
P1.7	SCK (used for In-System Programming)

Port 2

Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3

Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89S52, as shown in the following table.

Port 3 also receives some control signals for Flash programming and verification.

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt 0)
P3.3	$\overline{\text{INT1}}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{\text{WR}}$ (external data memory write strobe)
P3.7	$\overline{\text{RD}}$ (external data memory read strobe)

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. This pin drives High for 96 oscillator periods after the Watchdog times out. The DISRTO bit in SFR AUXR (address 8EH) can be used to disable this feature. In the default state of bit DISRTO, the RESET HIGH out feature is enabled.

ALE/$\overline{\text{PROG}}$

Address Latch Enable (ALE) is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ($\overline{\text{PROG}}$) during Flash programming.

In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

$\overline{\text{PSEN}}$

Program Store Enable ($\overline{\text{PSEN}}$) is the read strobe to external program memory.

When the AT89S52 is executing code from external program memory, $\overline{\text{PSEN}}$ is activated twice each machine cycle, except that two $\overline{\text{PSEN}}$ activations are skipped during each access to external data memory.

$\overline{\text{EA}}$/VPP

External Access Enable..$\overline{\text{EA}}$ must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, $\overline{\text{EA}}$ will be internally latched on reset.

$\overline{\text{EA}}$ should be strapped to V_{CC} for internal program executions.

This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming.

XTAL1

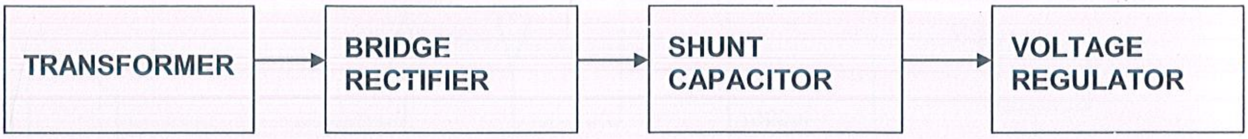
Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the inverting oscillator amplifier.

CHAPTER 7
POWER SUPPLY

7.1 Block Diagram



7.2 Working Of Power Supply

The power supply circuit comprises of four basic parts:

The transformer steps down the 220 V a/c. into 12 V a/c. The transformer work on the principle of magnetic induction, where two coils: primary and secondary are wound around an iron core. The two coils are physically insulated from each other in such a way that passing an a/c. current through the primary coil creates a changing voltage in the primary coil and a changing magnetic field in the core. This in turn induces a varying a/c. voltage in the secondary coil.

The a/c. voltage is then fed to the bridge rectifier. The rectifier circuit is used in most electronic power supplies is the single-phase bridge rectifier with capacitor filtering, usually followed by a linear voltage regulator. A rectifier circuit is necessary to convert a signal having zero average value into a non-zero average value. A rectifier transforms alternating current into direct current by limiting or regulating the direction of flow of current. The output resulting from a rectifier is a pulsating D.C. voltage. This voltage is not appropriate for the components that are going to work through it.

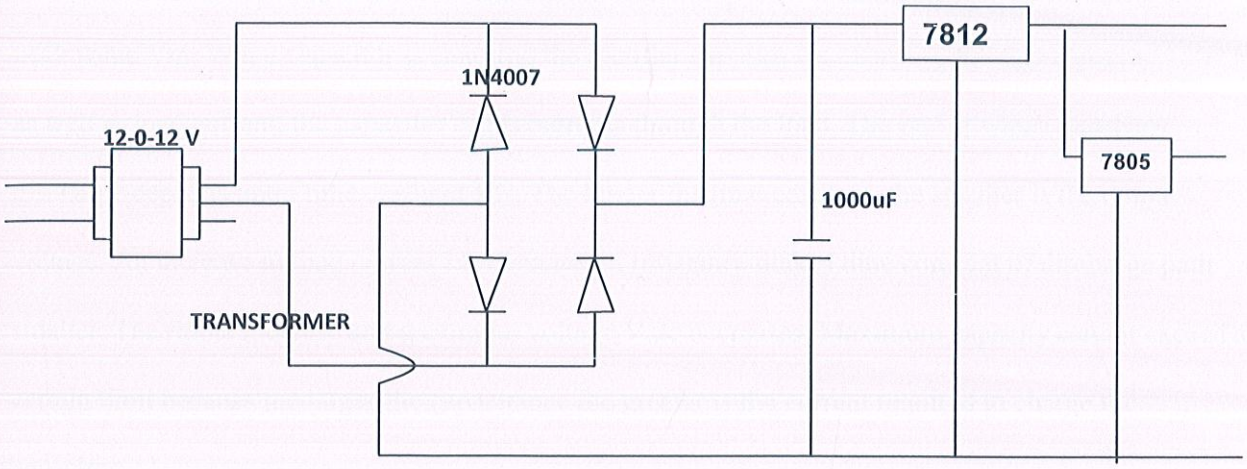


fig 7.1 circuit diagram of power supply unit

The ripple of the D.C. voltage is smoothened using a filter capacitor of 1000 microF 25V. The filter capacitor stores electrical charge. If it is large enough the capacitor will store charge as the voltage rises and give up the charge as the voltage falls. This has the effect of smoothing out the waveform and provides steadier voltage output. A filter capacitor is connected at the rectifier output and the d.c voltage is obtained across the capacitor. When this capacitor is used in this project, it should be twice the supply voltage. When the filter is used, the RC charge time of the filter capacitor must be short and the RC discharge time must be long to eliminate ripple action. In other words the capacitor must charge up fast, preferably with no discharge.

When the rectifier output voltage is increasing, the capacitor charges to the peak voltage Vm. Just past the positive peak, the rectifier output voltage starts to fall but at this point the capacitor has +Vm voltage across it. Since the source voltage becomes slightly less than Vm, the capacitor will try to send current back through the diode of rectifier. This reverse biases the diode. The diode disconnects or separates the source the source form load. The capacitor starts to discharge through load. This prevents the load voltage from falling to zero. The capacitor continues to discharge until source voltage becomes more than capacitor voltage. The diode again starts conducting and the capacitor is again charged to peak value Vm. When capacitor is charging the rectifier supplies the charging through capacitor branch as well as load current, the capacitor sends currents through the load. The rate at which capacitor discharge depends upon time constant RC. The longer the time constant, the steadier is the output voltage. An increase in load current i.e. decrease in resistance makes time constant of discharge path smaller. The ripple increase and d.c output voltage V dc decreases. Maximum capacity cannot exceed a certain limit because the larger the capacitance the greater is the current required to charge the capacitor.

The voltage regulator regulates the supply if the supply if the line voltage increases or decreases. The series 78xx regulators provide fixed regulated voltages from 5 to 24 volts. An unregulated input voltage is applied at the IC Input pin i.e. pin 1 which is filtered by capacitor. The out terminal of the IC i.e. pin 3 provides a regular output. The third terminal is connected to ground. While the input voltage may vary over some permissible voltage range, and the output voltage remains constant within specified voltage variation limit. The 78xx IC's are positive voltage regulators whereas 79xx IC's are negative voltage regulators.

These voltage regulators are integrated circuits designed as fixed voltage regulators for a wide variety of applications. These regulators employ current limiting, thermal shutdown and safe area compensation. With adequate heat sinking they can deliver output currents in excess of 1 A. These regulators have internal thermal overload protection It uses output transistor safe area compensation and the output voltage offered is in 2% and 4% tolerance.

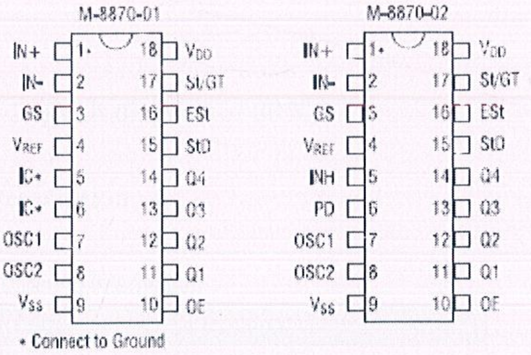
CHAPTER 8

DTMF Decoder IC 8870

8.1 Features

- Low Power Consumption
- Adjustable Acquisition and Release Times
- Central Office Quality and Performance
- Power-down and Inhibit Modes (-02 only)
- Inexpensive 3.58 MHz Time Base
- Single 5 Volt Power Supply
- Dial Tone Suppression

8.2 Pin Configuration



8.3 Circuit Diagram

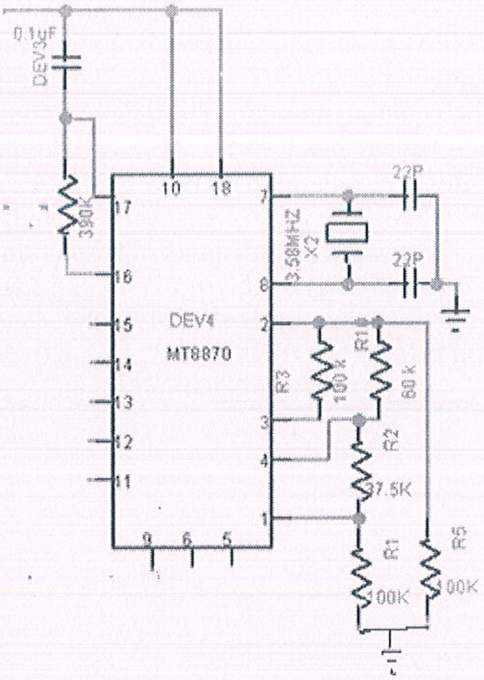
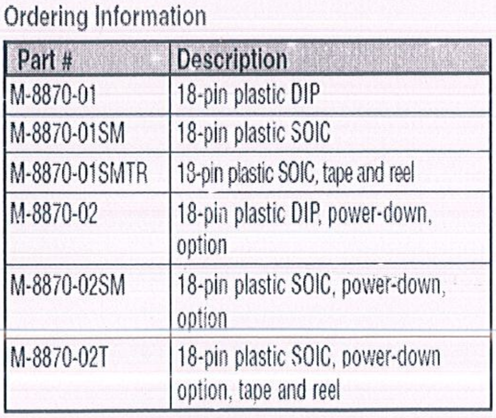


Fig 8.1 circuit diagram of 8870

8.4 Description



Ordering Information

Part #	Description
M-8870-01	18-pin plastic DIP
M-8870-01SM	18-pin plastic SOIC
M-8870-01SMTR	18-pin plastic SOIC, tape and reel
M-8870-02	18-pin plastic DIP, power-down, option
M-8870-02SM	18-pin plastic SOIC, power-down, option
M-8870-02T	18-pin plastic SOIC, power-down option, tape and reel

18-pin DIP or SOIC package. Manufactured using CMOS process technology, the M-8870 offers low power consumption (35 mW max) and precise data handling. Its filter section uses switched capacitor technology for both the high and low group filters and for dial tone rejection. Its decoder uses digital counting techniques to detect and decode all 16 DTMF tone pairs into a 4-bit code. External component count is minimized by provision of an on-chip differential input amplifier, clock generator, and latched tri-state interface bus. Minimal external components required include a low-cost 3.579545 MHz color burst crystal, a timing resistor, and a timing capacitor.

The M-8870-02 provides a "power-down" option which, when enabled, drops consumption to less than 0.5 mW. The M-8870-02 can also inhibit the decoding of fourth column digits

CHAPTER 9
MAX 232

9.1 Description

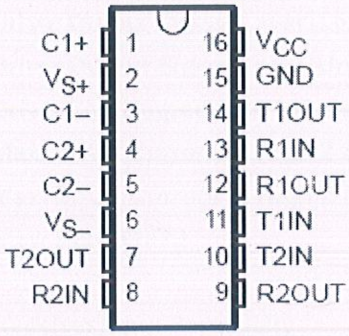
The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply EIA-232 voltage levels from a single 5-V supply.

Each receiver converts EIA-232 inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V and a typical hysteresis of 0.5 V, and can accept ±30-V inputs.

Each driver converts TTL/CMOS input levels into EIA-232 levels.

The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC□ library.

9.2 Pin Diagram



9.3 Circuit Diagram Of Max 232 Unit

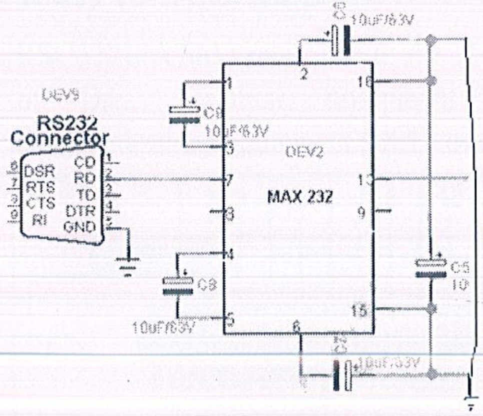


Fig 9.1

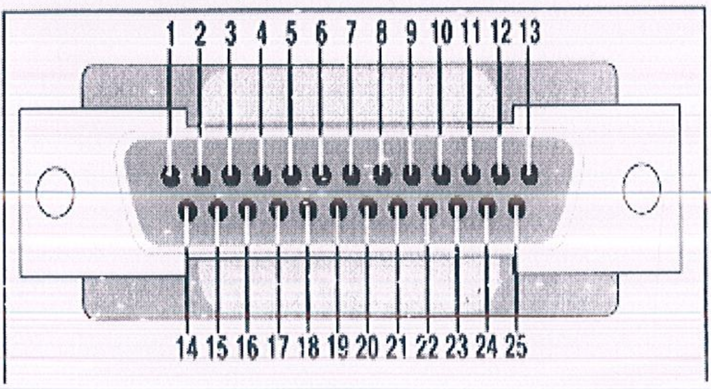
CHAPTER 10
RS232 PORT AND CONNECTOR

RS 232 PORT

Mode of Operation		SINGLE -ENDED
Total Number of Drivers and Receivers on One Line		1 DRIVER 1 RECVR
Maximum Cable Length		50 FT.
Maximum Data Rate		20kb/s
Maximum Driver Output Voltage		+/-25V
Driver Output Signal Level (Loaded Min.)	Loaded	+/-5V to +/-15V
Driver Output Signal Level (Unloaded Max)	Unloaded	+/-25V
Driver Load Impedance (Ohms)		3k to 7k
Max. Driver Current in High Z State	Power On	N/A
Max. Driver Current in High Z State	Power Off	+/-6mA @ +/-2v
Slew Rate (Max.)		30V/uS
		+/-15V
Receiver Input Sensitivity		+/-3V
Receiver Input Resistance (Ohms)		3k to 7k

RS-232 Interface

RS-232 (EIA Std.) applicable to the 25 pin interconnection of Data Terminal Equipment (DTE) and Data Communications Equipment (DCE) using serial binary data



CHAPTER 11
SOFTWARE REQUIREMENTS AND CODING USED

11.1 Coding:-

```
using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Speech.Recognition;//this will appear when you will add the reference
of system.speech

using System.Speech.Synthesis;

using System.Data.SqlClient;


namespace newivrs1

{

    public partial class Form1 : Form

    {

        SpeechRecognizer speechreco = new SpeechRecognizer();

        SpeechSynthesizer sp_sz = new SpeechSynthesizer();

        int flag;

        int flag2 = 1;

        int flag_open;

        int ff = 0;
        string read = "";

        String[] roll = new String[100];//array which  can accept 100 numbers

        String bt_read;//to read from the serial port

        int id1;//to store the roll

        int i = 0;

        public Form1()

        {

            InitializeComponent();




        }





        //SqlConnection con = new
SqlConnection("server=eeast;database=studentsivrs;uid=sa;pwd=123");

      //  SqlConnection con = new SqlConnection("server=saurabh-
pc;database=ivrs;uid=sa;pwd=123");


        SqlConnection con = new
SqlConnection("server=localhost;database=ivrs;Integrated Security=true");

        private void button1_Click(object sender, EventArgs e)

        {

            try

            {

                serialPort1.Open();

                sp_sz.SpeakAsync("you are successfully connected now");

                sp_sz.SpeakAsync("welcome to JUIT SOLAN");

                sp_sz.SpeakAsync("Press # or hash to connect Electronics and
communication DEPARTMENT ");

                flag_open = 1;
```

```
    }
    catch
    {
        MessageBox.Show("port busy");
    }

    //MessageBox.Show("you are connected now");


}

private void Form1_Load(object sender, EventArgs e)
{


}

private void button2_Click(object sender, EventArgs e)
{
    serialPort1.Close();
    //MessageBox.Show("you have closed the connection");
    sp_sz.SpeakAsync("you have closed the connection");
    timer1.Enabled = false;
    flag_open = 0;
}

private void timer1_Tick(object sender, EventArgs e)
{
```

```
            if (serialPort1.IsOpen)

            {

            //serialPort1.Open();

                if (serialPort1.BytesToRead > 0)

                {

                    bt_read = (serialPort1.ReadExisting());


                    serialPort1.DiscardInBuffer();


                  richTextBox1.Text += bt_read;

                    //started here

                    //first bracket

                    if (bt_read == "#" && flag2 == 1)//this is in place of #

                    {timer1 .Enabled =true ;

                        flag = 1;

                        flag2 = 0;

                        sp_sz.SpeakAsync("welcome to IET Bhaddal  ");

                        sp_sz.SpeakAsync("Electronics and communication
DEPARTMENT");

                        sp_sz.SpeakAsync("ENTER YOUR ROLL AND PRESS * or star TO
KNOW YOUR ATTENDENCE AND MARKS");


                    }

                    else if (bt_read == "*" && flag == 1)//this is inplace of *

                    {

                        richTextBox1.Text = "";

                        sp_sz.SpeakAsync("Press  zero hash To know about Attendence
and double zero hash to know about result ");

                      //richTextBox1.Text;
```

```
                          ff=1;

                     }

                        //

                     else

                         if(ff==1 && bt_read=="#")

                         {

                             string str = richTextBox1.Text;

                         flag = 0;


                         if (con.State == ConnectionState.Closed)

                         {

                             con.Open();

                         }

                         String qry;

                         //qry = "select stmrk from student where stid=@id;select
stdatd from student where stid=@id";

                         qry = "select * from student where stid=@id";

                         SqlCommand cmd = new SqlCommand(qry, con);

                         cmd.Parameters.Add("@id", SqlDbType.Int).Value = id1;

                         SqlDataReader dr = cmd.ExecuteReader();

                         while (dr.Read())

                         {

                             if (str == "0#" )

                             {

                                 sp_sz.SpeakAsync("Welcome   " +
dr["stname"].ToString());


                                 sp_sz.SpeakAsync("your marks is " +
dr["stmrk"].ToString());

                                 sp_sz.SpeakAsync("out of" +
dr["stmrkeng"].ToString());
```

```
                            }
                            else
                                if (str == "00#")
                                {
                                    sp_sz.SpeakAsync("Welcome  " +
dr["stname"].ToString());

                            sp_sz.SpeakAsync("and your attendence is " +
dr["stdatd"].ToString());
                             //here we can add the coloumns and we can even take
the output
                                }


                                }
                        //dr.Close();
                        //cmd.Dispose();
                        //qry1 = "select stdatd from student where stid=@id";
                        //SqlCommand cmd1 = new SqlCommand(qry1, con);
                        //cmd.Parameters.Add("@id", SqlDbType.Int).Value = id1;
                        //SqlDataReader dr1 = cmd1.ExecuteReader();
                        //while (dr1.Read())
                        //{
                          //  sp_sz.SpeakAsync("the attendence is " +
dr1["stdatd"].ToString() + "percent");
                        //}
                        dr.Close();
                        cmd.Dispose();
                          richTextBox1.Text="";
                        i = 0;
                        roll[0] = null;//this is done to make the array empty
```

```
            roll[1] = null;//this is done to make the array empty

            roll[2] = null;//this is done to make the array empty

            roll[3] = null;//this is done to make the array empty

            roll[4] = null;//this is done to make the array empty

            flag2 = 1;//this flag is true,hence the repeat is going on



            }




            else if (bt_read == "1" || bt_read == "2" || bt_read == "3"
|| bt_read == "4" || bt_read == "5" || bt_read == "6" || bt_read == "7" || bt_read
== "8" || bt_read == "9")

            {

                roll[i] = (bt_read);

                i++;

                if (i == 3)

                {

                    id1 = Convert.ToInt32(Convert.ToInt32(roll[0]) *
100 + Convert.ToInt32(roll[1]) * 10 + Convert.ToInt32(roll[2]));

                }

                else if (i == 2)

                {

                    id1 = Convert.ToInt32((Convert.ToInt32(roll[0]) *
10) + Convert.ToInt32(roll[1]));

                }

                else if (i == 1)

                {

                    id1 = Convert.ToInt32(roll[0]);

                }
```

```
                    }

            }

        }
    }
    }
}
```

11.2 SOFTWARE REQUIREMENTS:-

1.keil compiler:-

It is used to burn the coding into microcontroller.

2.Microsoft dotnet

3.Sql server 2005

11.3 CODING OF MICROCONTROLLER:-

```
#define DATA P1
#define RS P35
#define RW P36
#define E P37

#include<lcdrout.h>
#define dport P0
#include<serial.h>
bit kk=1;
#define std P27
#define buzz P24

unsigned char k;
void main()
{
lcd_init();
lcd_puts("Cellular Based");
lcd_cmd1(0xc0);
lcd_puts("     IVRS      ");

secdelay(3);

init_serial(9600);

std=1;
lcd_cmd1(0x01);


while(1)
        {
                lcd_cmd1(0x80);
                lcd_puts("Received Data:-");

                while(std==1);
                buzz=0;
                ms_delay(15);
                k=dport ;
                lcd_cmd1(0xc0);
                displaypval(k);


                if(k<10)
                {
                        send(48+k);

                }
                else
                {
                if(k==10)
                  send(48);
                if(k==11)
                  send('*');
                if(k==12)
                {
                  send('#');
                }
                }
                P0=255;
                std=1;
                buzz=1;
                secdelay(1);
                lcd_cmd1(0xc0);
                lcd_puts("                ");

                ms_delay(250);
        }
}
```

CHAPTER 11
APPLICATIONS

APPLICATIONS

- IVR systems are typically used to service high call volumes, reduce cost and improve the customer experience. Examples of typical IVR applications are: telephone banking, tele- voting, and credit card transactions. Large companies use IVR services to extend the business hours of operation.
- Call centers use IVR systems to identify and segment callers. The ability to identify customers allows the ability to tailor services according to the customer profile. It also allows the option of choosing automated services. Information can be fed to the caller allowing choices such as: wait in the queue, choose an automated service, or request a callback (at a suitable time and telephone number). The use of computer telephony integration (CTI) will allow the IVR system to look up the caller line identification (CLI) on a network database and identify the caller. This is currently accurate for about 80% of inbound calls. In the cases where CLI is withheld or unavailable, the caller can be asked to identify themselves by other methods such as a PIN or password. The use of DNIS will ensure that the correct application and language is executed by the IVR system.
- CTI allows a contact center or organization to gather information about the caller as a means of directing their inquiry to an appropriate agent. CTI can also extract important or relevant information about the individual customer from the database, making for a more effective and efficient service.
- The use of IVR and voice automation enables a company to improve its customer service and lower its costs, due to the fact that callers' queries can be resolved without the cost of a live agent who, in turn, can be directed to deal with specific areas of the service. If the caller does not find the information they need, or require further assistance, the call is then transferred to an agent who can deal with them directly through CTI integration. This makes for a more efficient system in which agents have more time to deal with complex interactions, for example, customer retention, up selling, cross selling and issue resolution. This way, the customer is more likely to be satisfied with a personalized service and the interaction is likely to be more

fulfilling and rewarding for the agent, as opposed to dealing with basic enquiries that require yes/no responses, such as obtaining customer details. Employee satisfaction is important in the telecommunications industry due to the fast turnover of staff, IVR is therefore one way of retaining a workforce and allowing them to do a more effective job.

- IVR also enables customer prioritization. In a system whereby individual customers may have a different status, for example, a bronze, gold or platinum card holder, the service will automatically prioritize the individuals call and, in the case of a platinum card holder, move them to the front of the calling queue.

Voice-Activated Dialers

(VAD) Voice-activated IVR systems are now used to replace the switchboard or PABX (Private Automatic Branch Exchange) operators and are used in many hospitals and large businesses to reduce the caller waiting time. An additional function is the ability to allow external callers to page hospital staff and transfer the inbound call to the paged person.

Entertainment and Information

The largest installed IVR platforms are used for applications such as tele-voting on TV game shows such as *Pop Idol* and *Big Brother* which can generate enormous call spikes. Often the network provider will have to deploy *Call gapping* in the Public network to prevent Network overload.

The following are some of the more common uses of an IVR:

- Mobile (Pay as you go Top up)
- Telephone Banking (Balance, payments, and transfers)
- Mobile Purchases (particularly for mobile content, such as ringtones and logos)
- Caller identification and routing
- Order Placements (Credit Card Payments)
- Airline (Ticket booking, Flight arrivals, Flight departures, Check in)
- Adult entertainment (Dating, Chat line etc)
- Weather forecasts

Anonymous Access

IVR systems also allow callers to obtain data relatively anonymously. Hospitals and Clinics have used IVR systems to allow callers to receive anonymous access to test results. This is information that could easily be handled by a person but the IVR system is used to preserve privacy and avoid potential embarrassment of sensitive information or test results. Users are given a pass code to access their results.

Clinical Trials

IVR systems are used by pharmaceutical companies and contract research organizations to conduct clinical trials and manage the large volumes of data generated. The caller will respond to questions in their preferred language and their responses will be logged into a database and possibly recorded at the same time to confirm authenticity. Applications include patient

Outbound Calling

IVR systems can be used for outbound calls, as IVR systems are more intelligent than Dialer systems, they can recognize different line conditions.

- RNA Ring No Answer
- Answered by Voicemail or Answering machine (In this circumstances they can leave a message)
- Fax Tone (IVR can leave a Fax Message based upon a TIFF Image)
- Answer (IVR can tell the customer who is calling and ask them to wait for an agent)
- Recognize Divert messages and abandon call.

IVR uses Call Progress Detection to monitor Line conditions, and report to the IVR Database.

ADVANTAGES AND DISADVANTAGES OF USING IVR SYSTEMS

6.1 ADVANTAGES

The biggest advantage of IVR for small and large organizations is to save time and money. Answering phone calls takes a lot of time, and not every phone call deserves the attention of a trained employee. IVR systems can take care of most of the frequently asked questions that an organization receives (office hours, directions, phone directory, common tech support questions, et cetera) and allow customer service reps, salesmen and tech support specialists to concentrate on the harder stuff. If a large company is able to shave even a second off the average length of each phone call with a live operator, it can save them hundreds of thousands or even millions of dollars a year [source: Human Factors International]. IVR systems have the advantage of making callers and customers feel like they're being attended to, even if it's just by a machine. If you have a simple question, it's better to get a quick answer from a computerized operator than to wait ten minutes on hold before talking to a human being.

Another advantage is that IVR systems don't sleep. They don't take lunch breaks. They don't go on vacations to the Bahamas. An IVR system can be available 24 hours a day to field questions and help customers with simple tasks. An IVR system can make a small company look bigger. Some IVR hosting plans even set you up with an 800 number to look more official. Subscription IVR hosting plans make it easier for businesses and organizations to use these automated phone services. This is a big advantage of days past, when only large companies with big telecommunications and computing budgets could afford the hardware, software and staff to run in-house IVR systems.

6.2 DISADVANTAGES

The greatest disadvantage of IVR systems is that many people simply dislike talking to machines. Older adults may have a hard time following telephone menus and lengthy instructions. And younger callers get frustrated with the slowness of multiple phone menus

CONCLUSION

The system designed will be intelligent for interaction and will suitably provide a good response to the caller who will access it. It will be truly a responsible system for students. We will make it better than the present scenario system. It will be digitally accessed and will have a strong data base and can be operated easily and of low cost.

BIBLIOGRAPHY

Books:

1.ThiagarajanVishwanathan/Telecommunication Switching System & Networks/India PRI Pvt.ltd/Second Edition

2.Kenneth J.Ayala/The 8051 Microcontroller Architecture ,Programming and Applications/India/PRI Pvt.ltd/Second Edition

3. Douglas V.Hall/Microcontroller and Interfacing/New York/TMH Publishing Company PvtLtd/Second Edition