APPLIANCE CONTROL USING TELEPHONE LINE

Project Report

Submitted By

Aditya Chandran

Arul Sam Dilip. J

Ravi Kumar. S

Suresh C.P

Under the guidance of

Mrs. Allin Christie, M.E

2000 - 2001

In partial fulfillment of the requirements
for the award of the degree of
Bachelor of Engineering in
Electronics and Communication Engineering
of Bharathiar University, Coimbatore

Department of Electronics and Communication Engineering

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE – 641 006

Kumaraguru College of Technology

Coimbatore – 641 006

Department of Electronics and Communication Engineering

Certificate

This is to certify that this project entitled

Appliance Control Using Telephone Line

has been submitted by

Mr. ______________________________

In partial fulfillment of the requirements for the award of degree of

Bachelor of Engineering in the Electronics and Communication Engineering Branch

of Bharathiar University, Coimbatore – 641 046 during the

Academic year 2000 – 2001

____
(Guide)

____
(Head of the Department)

Certified that the candidate was examined by us in the Project Work

Viva-Voce Examination held on [illegible]

University Register Number ______________________

____
(Internal Examiner)

12.03.2001
(External Examiner)

AL SYSTEMS

1st Floor, 105,
3rd Street,
Tatabad,
Coimbatore – 641 012.

Ph.: 493960, 494248
Fax: 0422-498158
email: also@iname.com

TO WHOMSOEVER IT MAY CONCERN

The following students doing final year BE (ECE) at Kumaraguru College of Technology had undergone their project titled "Appliance Control Using Telephone Line" at our factory.

The students name are:

1. Aditya Chandran
2. J. Arul Sam Dilip
3. S. Ravikumar
4. C.P. Suresh.

Their conduct and attendance were found to be satisfactory during their project period. We wish them all a successful future.

For AL SYSTEMS

J. Narayan Kumar.

(J. NARAYAN KUMAR)
Managing Partner.

Dedicated

to

Our Beloved Parents

CONTENTS

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

We thank our Principal **Dr.K.K.PADMANABAN** for his patronage and encouragement.

We express our sincere thanks to the Head of the Department, **Prof. M. RAMASAMY, M.E., M.I.S.T.E, M.I.E, C.ENG.(i),M.B.M.E.S.I, M.I.E.E.E (U.S.A), Head Of the Department of Electronics and Communication Engineering** for his motivation and suggestions during the course of our project work.

We are indebted to **Mrs. ALLIN CHRISTIE M.E,** for her guidance and enthusiasm which helped a long way in the successful completion of this project.

We also thank **Mr. J. NARAYAN KUMAR, M.D., (AL SYSTEMS)** for his keen interest and untiring efforts towards the completion of the project.

SYNOPSIS

SYNOPSIS

Control of appliances through telephone lines has proved to be a very reliable and efficient method of automation that finds application largely for domestic purposes and in industry to a certain extent. The system is essentially micro controller based and uses '**ring detector'** and '**DTMF decoder'** circuits as auxiliaries. The 'ON-OFF' conditions for the various appliances are relay controlled.

An interesting option for PC communication has also been provided. Depressing a specific digit in the transmitting telephone keypad sends a single data to the PC via it's serial port. On reception of this data (character) the **SHUTDOWN** program is automatically called. The RS-232 max IC has been used for this application.

Asynchronous mode of communication has been used. No external UART/USART is required for serial port communication. The entire operation is password protected, the mistyping of which, automatically cuts off the user phone link.

INTRODUCTION

INTRODUCTION

The use of telephone lines for various applications has a wide and extensive scope – obvious in multifarious circuit realizations. This project deals exclusively with the control of home (and industrial) appliances via the telephone line, highlighted by different numeric extensions controlling the 'ON-OFF' state of individual devices.

The provision for PC communication is decidedly an added advantage. Much importance has been given to the design and performance of the data communication system.

The project is aimed to develop an effective appliance control system, which may be later widened to serve a larger network. The use of several sophisticated ICs highlights the design.

CHAPTER : 1

OVERVIEW

OF THE PROJECT

OVERVIEW OF THE PROJECT

The details of the important parts of the system are defined breifly in this section.These parts will be considered in more detail in the forthcoming sections.

MICROCONTROLLER:

Our Project consists of a 89C51 Microcontroller which controls the overall operations of the system. It is an 8-bit microcontroller which has 128*8 bit internal RAM. It is the heart of the project which takes care of all the manipulations which takes place in the system. The list of operations which has to be done are programmed inside the microcontroller. A seven-segment display is added to highlight the key pressed.

RING DETECTOR:

To set the number of rings the user has to make in order to invoke the system, we go in for a ring detector circuit.The major components in the ring detector circuit are a transistor and a 555 timer which is the precise device for getting the exact time delay or oscillation. The number of rings which has to be made can be modified using the software part.

DTMF DECODER:

In our project we use a 8870 DTMF Decoder. Each digit in the telephone keypad is assigned two frequencies- high and low-corresponding to row- column arrangement. The received DTMF frequencies are decoded and a corresponding binary output is obtained which is sent to the micro controller.

The M-8870 decoder uses a digital counting technique to determine the frequencies of the limited tones and to verify that they correspond to standard DTMF frequencies.The DTMF Decoder gets it's input from the telephone line and the output is fed onto the port of the microcontroller.

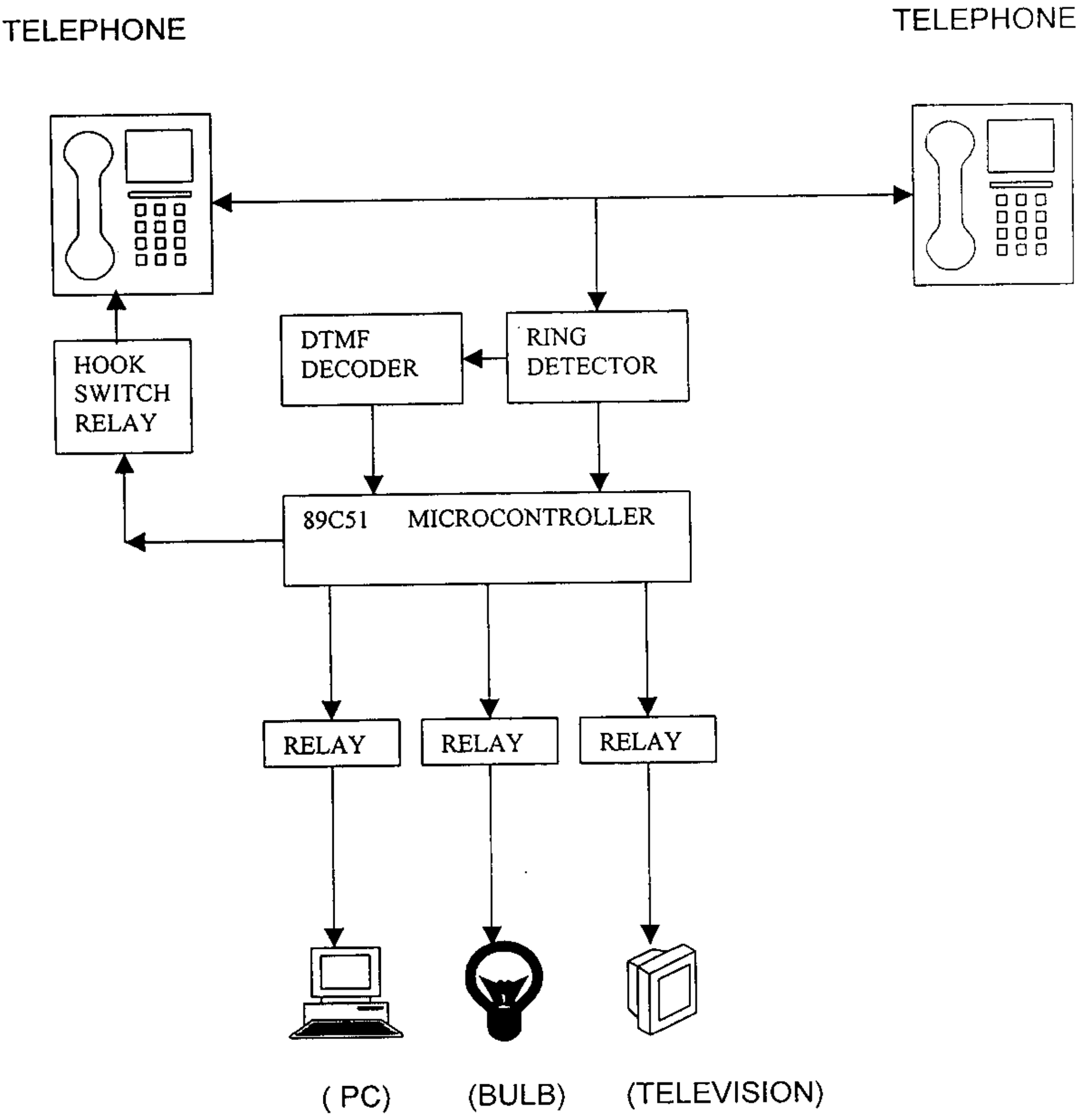
HOOKSWITCH RELAY:

Electromechanical relay is used for hook switch release.This relay consists of a core and a coil is wound around the core. When the coil is energized the core becomes maganitized and the armature placed near the core is attracted to it and the contact is made. When the electricity is removed, demagnetization of the core occurs and the contact is removed.The other relays which are used for controlling the operation of the different devices are also of electromechanical type.

SERIAL PORT:

In order to access the computer when it is to be turned off or on we go in for an RS-232 Cable, by which we communicate serially with the computer. In order to make out the voltage conversions we use a MAX IC. The data sending details is made through the software where we send data using the assembly language program and subsequently the receiving and the PC control is done by using the programming languages C and Visual Basic.

OVERALL BLOCK DIAGRAM



(Fig 1.1)

CHAPTER : 2

THE MICROCONTROLLER

THE MICROCONTROLLER

INTRODUCTION :

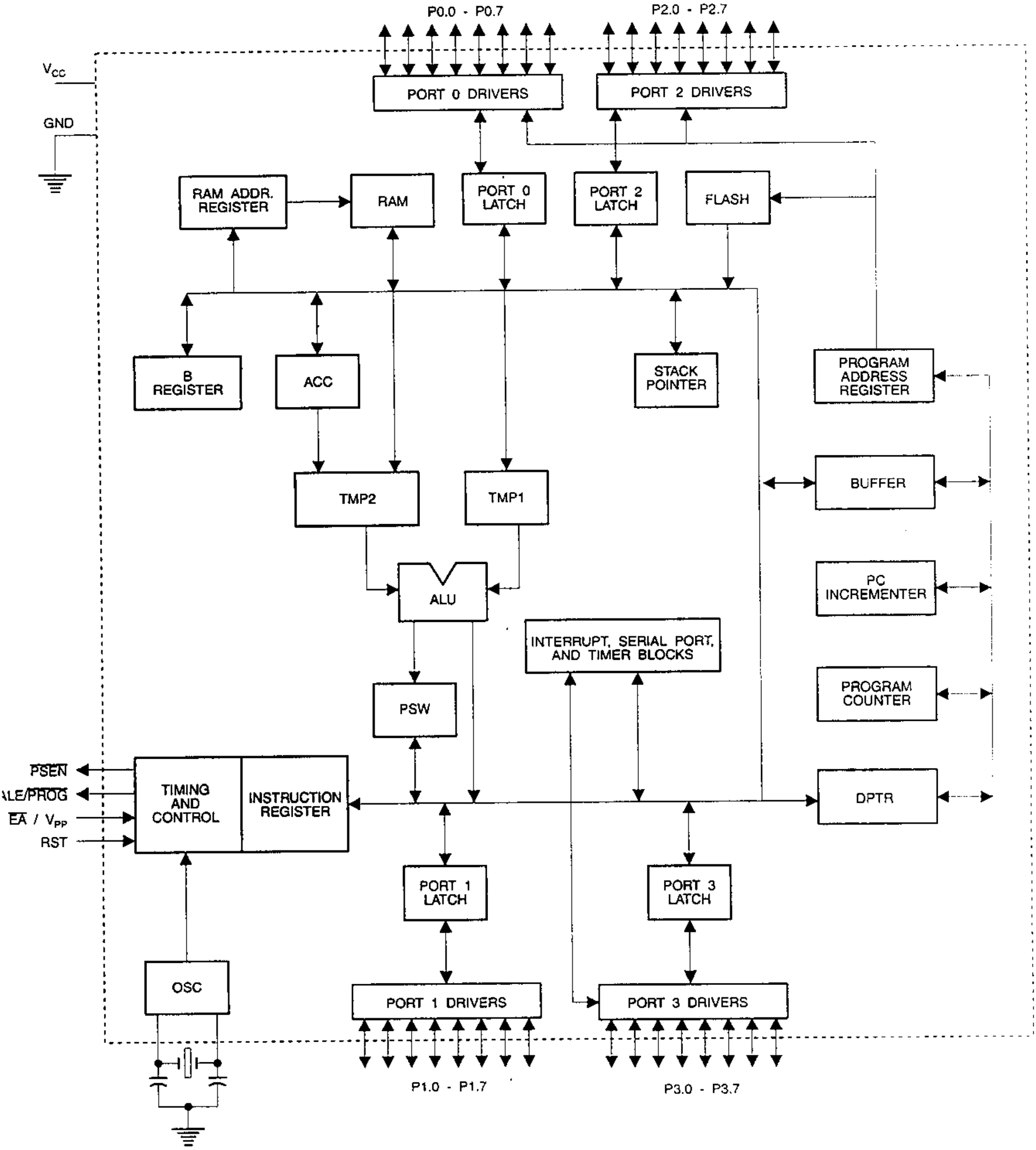
Microcontrollers can be described as microprocessors with RAM,ROM and various I/O facilities on a single chip. The distinguishing characteristic of a microcontroller is the inclusion ,on one chip ,of all the resources which permit it to serve as a controller in a device or an instrument. these microcontrollers are generally used in real-time control applications.

MICROCONTROLLER AT89C51 DETAILS: -

DESCRIPTION OF AT89C51 :

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash programmable and erasable read only memory (PEROM). The deviceis manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 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 Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

Block Diagram



LED DISPLAY INTERFACE

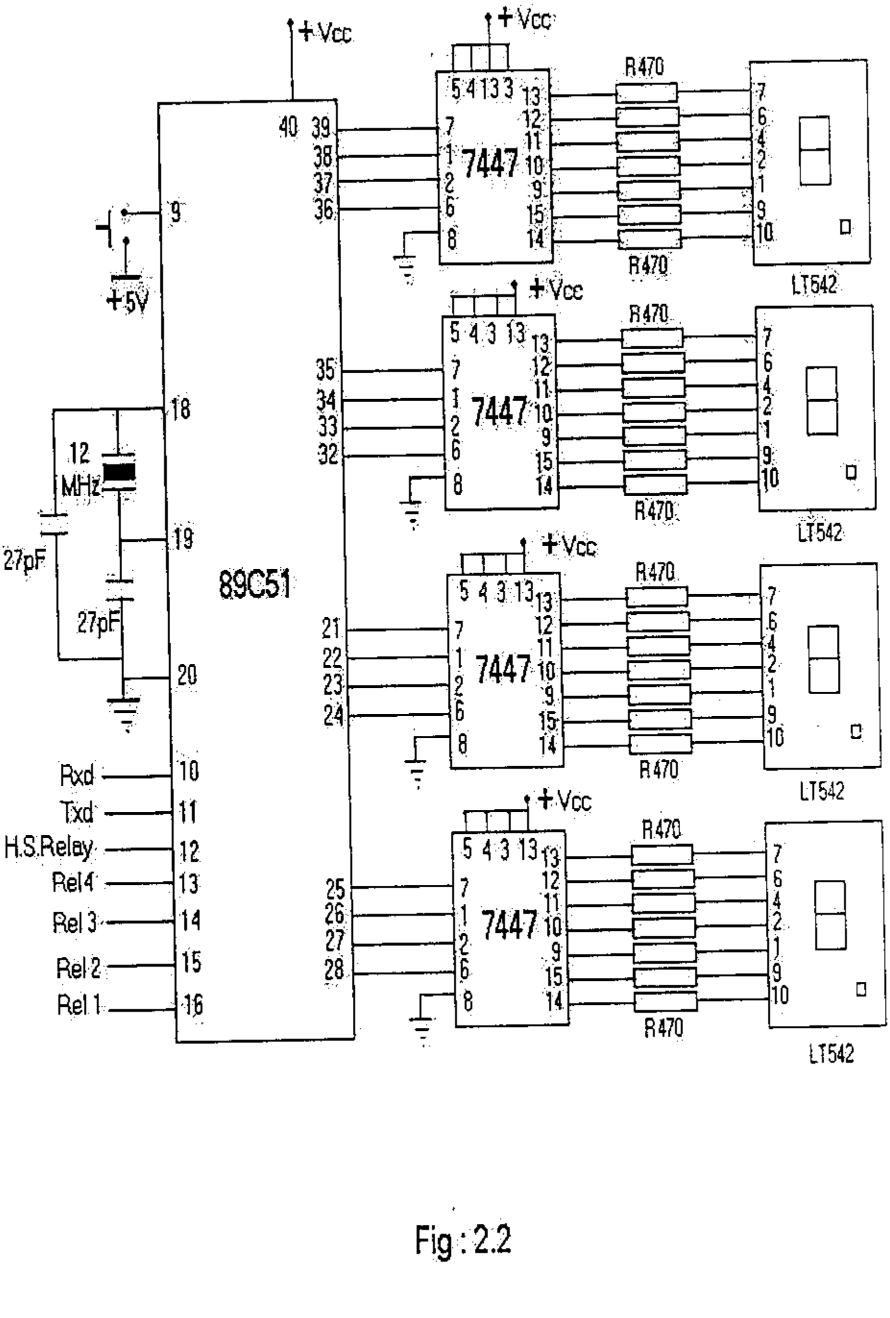
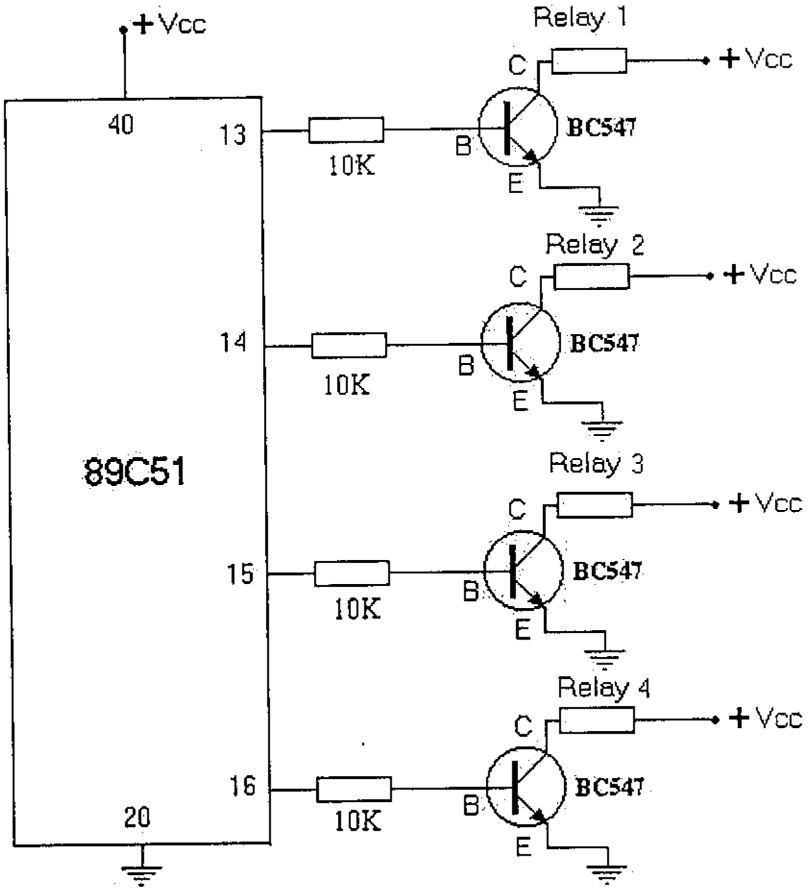


Fig : 2.2

Fig Interfacing Relays with the MicroController



FEATURES :

* Compatible with MCS-51 products

* 4K Bytes of In-System reprogrammable Flash Memory

* Endurance: 1,000 Write/Erase Cycles

* Fully Static Operation: 0 Hz to 24 MHz

* Three-level Program Memory Lock

* 128 x 8-bit Internal RAM

* 32 Programmable I/O Lines

* Two 16-bit Timer/Counters

* Six Interrupt Sources

* Programmable Serial Channel

* Low-power Idle and Power-down Modes

* 40-pin DIP package with Vcc= +5 v.

* Number of ports available ------3.

* Ports 0 to 3 .

* All ports are 8-bit bi-directional.

* The pins of port 3 have alternate functions.

*** VCC**

Supply voltage.

***GND**

Ground.

PORT DESCRIPTION

PORT 0:

Port 0 is an 8-bit open-drain bi-directional I/O port. As anoutput 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 may also be configured to be the multiplexed low-order address/data bus during accesses to external programand 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 bi-directional I/O Port with internal pull-ups.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. Port 1 also receives the low-order address bytes during Flash programming and verification.

PORT 2 :

Port 2 is an 8-bit bi-directional I/O port with internal pull-ups. 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, it uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the 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 bi-directional. 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 bythe 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 AT89C51 as listed below:

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

AVAILABLE MODES: -

IDLE MODE:

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this

mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset. It should be noted that when idle is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

POWER-DOWN MODE:

In the power-down mode, the oscillator is stopped, and the instruction that invokes power-down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power-down mode is terminated. The only exit from power-down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V CC is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

STATUS OF EXTERNAL PINS DURING IDLE AND POWER-DOWN MODES

Mode	Program Memory	ALE	PSEN	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Addr	Data
PD	Internal	0	0	Data	Data	Data	Data
PD	External	0	0	Float	Data	Data	Data

CHAPTER : 3

DTMF DECODER

DTMF

INTRODUCTION:

DTMF: Dual Tone Multi Frequency. Two tones are used to generate a DTMF digit. One tone is chosen out of four row tones, and the other is chosen out of four column tones(Ref Table 3.1).Two of eight tone can be combined so as to generate sixteen different DTMF digits. Of these sixteen keys shown in the Figure 3.1, twelve are the familiar keys of a touch tone keypad and four are reserved for future uses.

	Col 1	Col 2	Col 3	Col 4
Row 1	1	2	3	A
Row 2	4	5	6	B
Row 3	7	8	9	C
Row 4	*	0	#	D

Fig 3.1 KEYPAD

Dual tone multi frequency (DTMF) signaling is quickly replacing dial-pulse signaling in telephone banking or electronic mail systems, in which the user can select options from a menu by sending signals from a telephone.

DTMF STANDARDS:

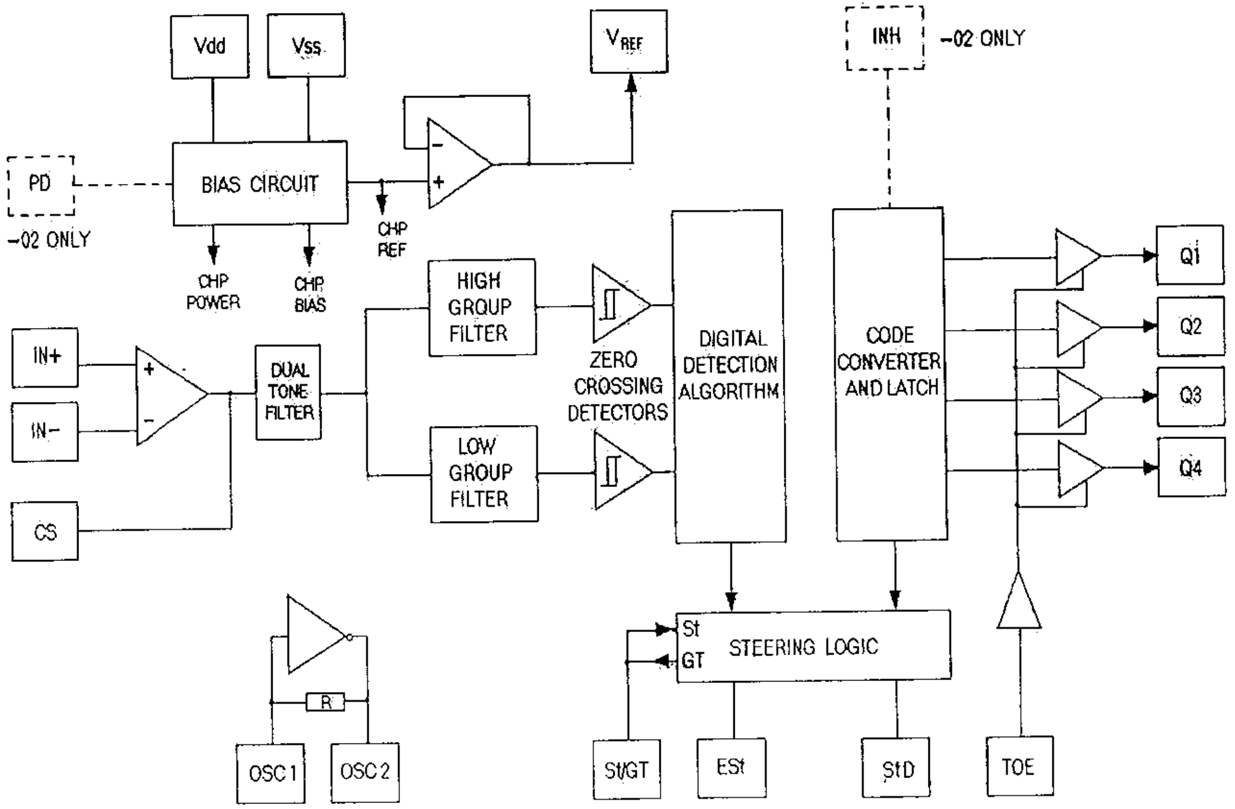
The DTMF tone-signalling standard is also known as touch tone or MFPB (Multi-frequency push button). Touch-tone was developed by bell labs for use by AT&T in the dial-pulse-signalling standard. Each administration has defined its own DTMF specifications. They are all very similar to the CCITT standard, varying by small amounts in the guard bands allowed in frequency, power twist and talk-off.

DTMF DECODER:

The DTMF signals transmitted over the telephone lines can be received and decoded outputs can be suitably used along with certain additional circuitry to design a DTMF code detection unit.

The DTMF digits transmitted over the telephone lines would have a nominal width of 50ms followed by a pause of similar duration between consecutive digits would be transmitted in one second. The on- cradle and off-cradle status of handset can be detected, based on the line voltage state, before the start of ringing. The voltage drops to 10 to 12V DC on lifting of the handset from the cradle. The ringing status can be detected with the use of either a capacitively coupled rectifier bridge or an AC opto-coupler or even a DC opto coupler with an external diode shunted in anti-parallel across the internal diode of the opto-coupler together with a current limiting series resistor.

Fig 3.2: Block Diagram of DTMF Receiver



IC M-8870:

This integrated DTMF decoder type M-8870 decodes the tone dialing codes received via the telephone line.

FEATURES

- Low power consumption
- Adjustable acquisition and release times
- Central office quality and performance
- Power-down and inhibit modes
- Single 5-volt power supply
- Dial tone suppression.

FUNCTIONAL DESCRIPTION OF M-8870:

It is a full DTMF receiver that integrates both band split filter and decoder functions into a single 18-pin DIP or SOIC package. Block diagram of M-8870 is shown in Fig 3.2. Manufactured using state of the art CMOS process technology, the M-8870 offers low power consumption and precise data handling. Its filter section uses switched capacitor technology for both the low and high group filters and for dial tone rejection. Its decoder uses digital counting techniques to detect and decode all 16 DTMF tone pairs into 4 bit code. External component count is minimized by the provision of a on-chip differential input amplifier, clock generator, and latched tri-state interface bus. Only a Minimal number of external components are required, which include a low cost 3.579545 MHz color burst crystal, a timing resistor, and timing capacitor.

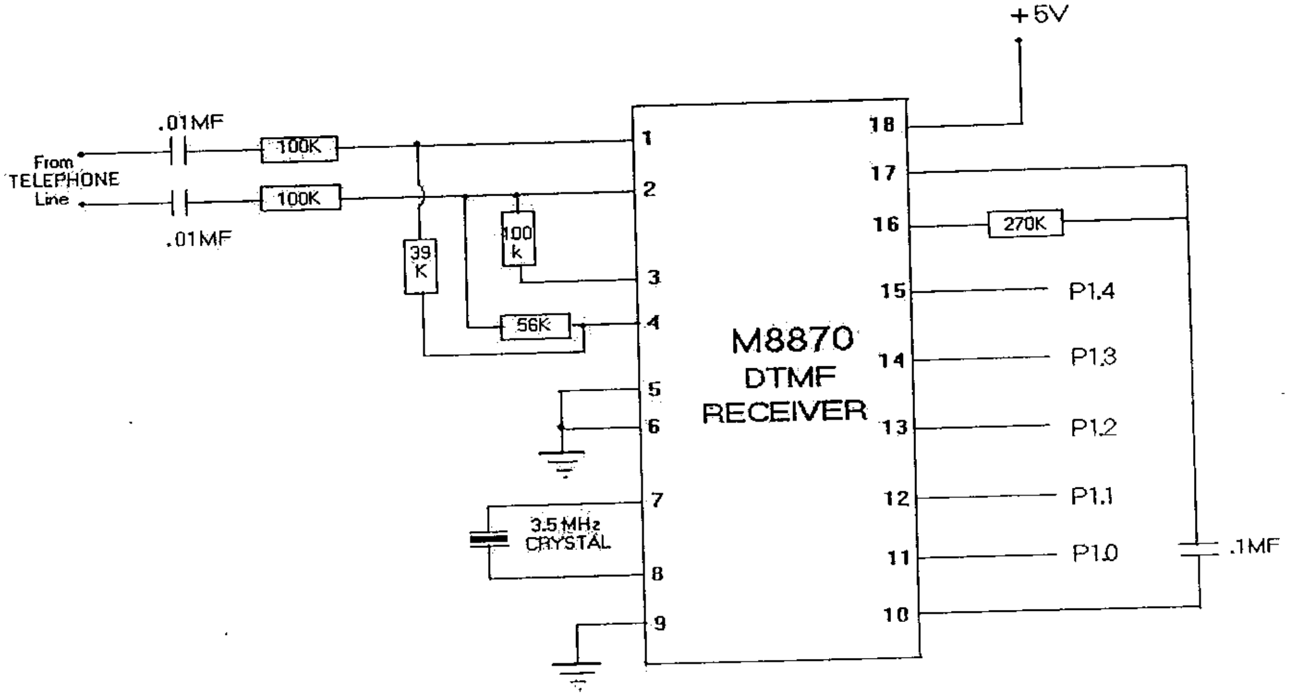


Fig 3.3 : DTMF Decoder

The other version of M-8870 ,M-8870-02 provides a “power down” option which, when enabled, drops consumption too less than .5 mw. The 02 version can also inhibit the decoding of the fourth column digits.

The M-8870 operating functions include a band split filter that separates the high and low tones of the received pair, and a digital decoder that verifies both the frequency and duration of the received tones before passing the resulting 4-bit code to the output bus.

FILTER:

The low and high group tones are separated by applying the dual tone signal to the inputs of two 9th order switched capacitor band pass filters with bandwidths that corresponds to the bands enclosing the low and high group tones. The filter also incorporates notches at 350 and 440Hz, providing excellent dial tone rejection.Each filter output is followed by a single order switched capacitor section that smoothens the signals prior to limiting. Signal limiting is performed by high gain comparators provided with hysteresis to prevent detection of unwanted low-level signals and noise. The comparator outputs provide full rail logic swings at the frequencies of the incoming tones.

DECODER:

The M-8870 decoder uses a digital counting technique to determine the frequencies of the limited tones and to verify that they correspond to standard DTMF frequencies. Complex averaging algorithm is used to protect against tone simulation by extraneous signals while tolerating small frequency variations. The algorithm ensures an optimal combination of immunity to talk off and tolerance to interfacing signals and noise. When the detector recognizes the simultaneous presence of two valid tones(known as "signal condition), it raises the early steering flag (EST). Any subsequent loss of signal condition will cause EST to fall.

STEERING CIRCUIT:

Before a decoded tone pair is registered, the receiver checks for valid signal duration. This check is performed by an external RC time constant driven by ESt. A logic high on ESt causes Vc(see Fig 3.4) to raise as capacitor discharges. Provided that the signal condition is maintained for the validation period Vc reaches the threshold (V_{TSt}) of the steering logic to register the tone pair thus latching its corresponding 4-bit code into the output latch.

At this point, the GT output is activated and drives Vc to V_{DD}. GT continues to drive high as long as ESt remains high. Finally, after a short delay to allow the output latch to settle, the delayed steering output flag (StD) goes high, signaling that received tone pair has been registered. The contents of the output latch are made available on the four-bit output bus by raising the three-state control input (OE) to logic high.

The steering circuit works in reverse to validate the inter digit pause between signals. Thus as well as rejecting signals too short to be considered valid, the receiver will tolerate signal interruptions too short to be considered a valid pause. This capability, together with the ability to select the steering time constants externally, allows the designer to tailor performance to meet a wide variety of system requirements.

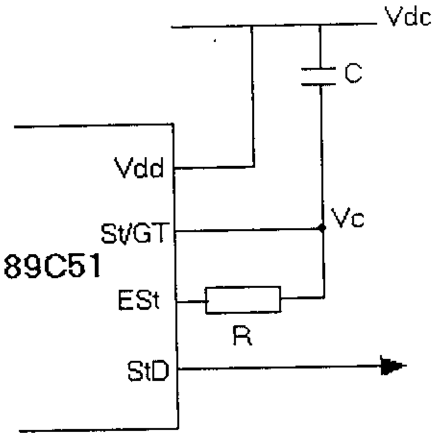


Fig 3.4 Basic Steering Circuit.

GUARD TIME ADJUSTMENT:

Where independent selections of receive and pauses are not required the simple steering circuit of (Fig 3.4) is applicable. Component values are chosen according to the formula:

tREC=tDP+tGTP

tGTP=0.67RC

The value of tDP is a parameter of the device and tREC is the minimum signal duration to be recognized by the receiver. A value for C of 0.1μF is recommended for most applications, leaving R to be selected by the designer. For example suitable value of R for a tREC of 40ms would be 300k ohm. The timing requirements for most telecommunications are satisfied with this arrangement.

Different steering arrangements may be used to select independently the guard times for tone present (tGTP) and tone absent (tGTA). This may be necessary to meet system specifications that place both accept and reject limits on both tone and interdigit pause.

Guard time adjustments also allows the designer to tailor system parameters such as talk off and noise immunity. Increasing to tailor system parameters such as talk off and noise immunity. Increasing tREC improves talk off performance, since it reduces the probability that tones simulated by speech will maintain signal condition long enough to be registered. On the other hand a relatively short tREC with a long to do would be appropriate for extremely noisy environment where fast acquisition time and immunity to dropouts would be required.

INPUT CONFIGURATION:

The input arrangement of the M-8870 provides a differential input operational amplifier as well as bias source to bias the inputs at mid rail. Provision is made for connection of a feedback resistor to the op-amp output (GS) for gain adjustments.

In a single ended configuration, the input pins are connected with the op-amp connected for unity gain and Vref biasing the input at ½ V_{DD}. Adjustable gain configuration is possible with the help of the feedback resistor.

DTMF CLOCK CIRCUIT:

The internal clock circuit is completed with the addition of a standard 3.579545MHz color burst crystal. The crystal can be connected to a single M-8870 or to a series of M-8870's by coupling the oscillator output of each M-8870 through a 30pF capacitor to the oscillator input of the next M-8870.

PIN FUNCTIONS OF M-8870:

IN+: Non-Inverting input.Connections to the front end differential amplifier.

IN-: Inverting input. Connections to the front end differential amplifier.

GS: Gain Select.Gives access to output of front ednd amplifier for connection of feedback resistor.

Vref: Reference voltage output(nominally Vdd/2).May be used to bias the inputs at mid-rail.

INH*: Inhibits detection of tones representing keys A,B,C,and D.

PD* : Power down. Logic high powers down the device and inhibits the oscillator.

OSC1,OSC2: Clock input. 3.579545MHZ crystal connected between these pins completes the internal oscillator.

Vss : Negative power supply (normally connected to 0v).

OE: Three state output enable (input). Logic high enables the outputs Q1-Q4. Internal pull up.

Q1, Q2 ,Q3,Q4: Three state outputs. When enabled by OE, provides the code corresponding to the last valid tone pair received.

StD : Delayed steering output. Presents a logic high when a received tone pair has been registered and the output latch is updated. Returns to logic low when the voltage on St/GT falls below V_{TSt}.

ESt: Early steering output presents a logic high immediately when the digital algorithm detects a recognizable tone pair. Any momemtary loss of signal condition will cause ESt to return to a logic low.

St/GT: Steering input/guard time output. A voltage greater than V_{TSt} detected at St causes the device to register the detected tone pair and update the output latch. A voltage less than V_{TSt} frees the device to accept a new tone pair. The GT output acts to reet the external steering time constant, and its state is a function of ESt and the voltage on St.

V_{DD} : Positive power supply.

Tabular Column showing the various Combinations of Frequencies for the keys in the keypad.

Table 3.1

F LOW	F HIGH	KEY (ref.)	OE	Q4	Q3	Q2	Q1
697	1209	1	H	0	0	0	1
697	1336	2	H	0	0	1	0
697	1477	3	H	0	0	1	1
770	1209	4	H	0	1	0	0
770	1336	5	H	0	1	0	1
770	1477	6	H	0	1	1	0
852	1209	7	H	0	1	1	1
852	1336	8	H	1	0	0	0
852	1477	9	H	1	0	0	1
941	1336	0	H	1	0	1	0
941	1209	.	H	1	0	1	1
941	1477	#	H	1	1	0	0
697	1633	A	H	1	1	0	1
770	1633	B	H	1	1	1	0
852	1633	C	H	1	1	1	1
941	1633	D	H	0	0	0	0
ANY	ANY	ANY	L	Z	Z	Z	Z

APPLICATIONS:

- Telephone switch equipment
- Mobile radio
- Remote control
- Remote data entry.

CHAPTER :4

RING DETECTOR

RING DETECTOR

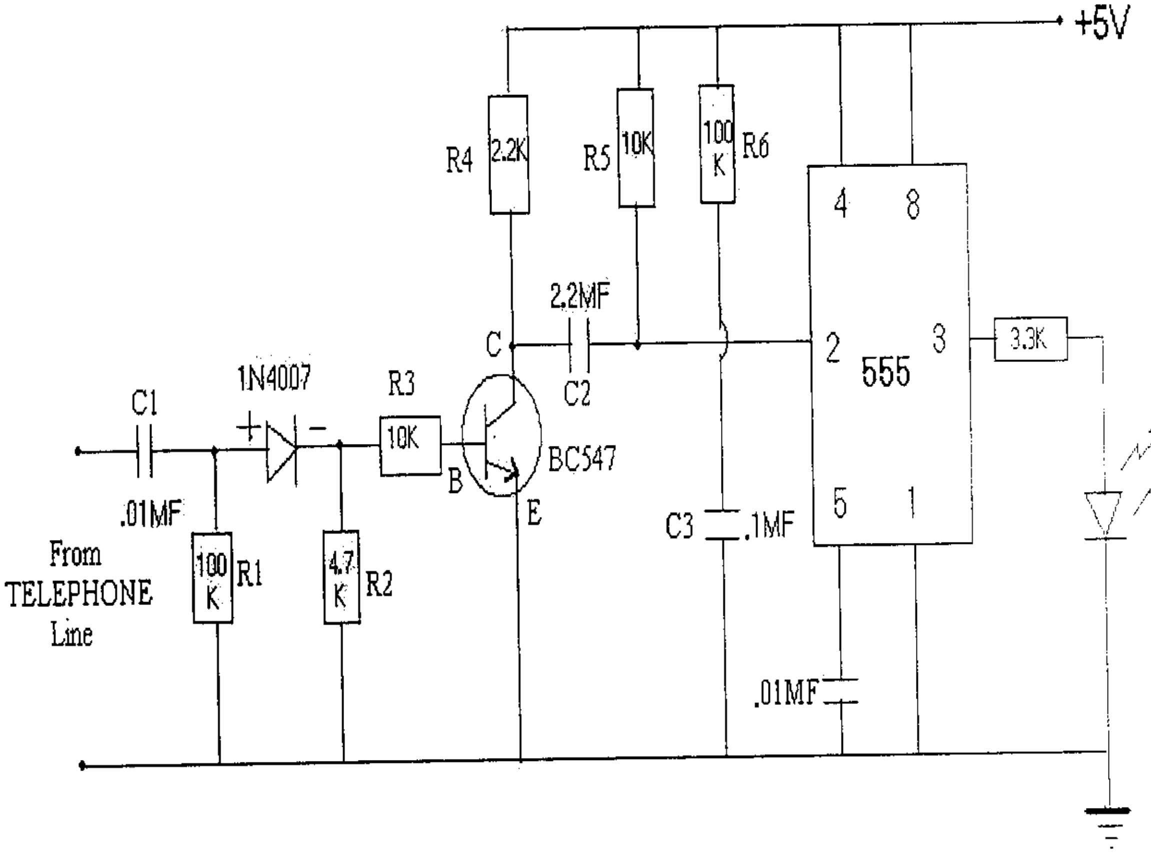
The circuit for the ring detector is as shown in the Figure 4.1. The capacitor C1 acts as a filter for the incoming AC signal. This AC signal is fed to the diode 1N4007, which acts as a rectifier. Resistors R1 and R2 are used for current limiting purposes. The base resistor R3 prevents the high current from reaching the transistor BC547 i.e. it prevents the transistor from getting damaged. The resistor R5 is used for signal boosting.

Initially the capacitor C2 is in the charged condition. When the ring signal is given as the input the transistor becomes ON and the capacitor C2 discharges, which in turn helps in triggering the timer 555. Depending upon R6 and C3 the timer is oscillated and a pulse is generated at the output. Thus the ring detector detects the number of ring pulses. The number of ring pulses can be set according to our need using software and an electromechanical relay is used for hook switch release.

This relay consists of a core and a coil is wound around the core. When the coil is energized the core becomes magnetized and the armature placed near the core is attracted to it and the contact is made. When the electricity is removed, demagnetization of the core occurs and the contact is removed.

RING DETECTOR

Fig 4.1 : Ring detector



555 TIMER:

555 TIMER is used in the ring detector circuit. It is a highly stable device for getting accurate time delay or oscillations. A single 555 TIMER can provide time delay ranging from microseconds to hours.

The 555 TIMER can be used with supply voltage in the range of +5V to +18V and can drive load upto 200mA. It is compatible with both TTL and CMOS logic circuits. Because of the wide range of supply voltage the 555 TIMER are versatile and are easy to use in various applications. The various applications include oscillator, pulse generator, ramp and square-wave generator, mono-shot multivibrator, burglar alarm, traffic light control and voltage monitor etc. The 555 timer is used in Monostable Multivibrator mode for Ring Detector.

CHAPTER : 5

POWER SUPPLY

CIRCUIT DESCRIPTION

The power supply to the main unit is obtained from the power section which consist of of a transformer, bridge rectifier and a voltage regulator.The transformer we have used is a step-down transformer.This is 230 to 15V AC step-down transformer.The 230V AC supply is directly given to the transformer,and the output of the step down transformer is a 15V AC supply. This 15V AC output signal from the transformer is given to the bridge rectifier.The rectifier is a device, which converts the ac signal in to pulsating DC signal and one more reason for choosing this bridge rectifier is that its efficiency is more compare to full wave rectifier.

The rectified 15V DC signal from the bridge rectifier is given to the voltage regulator (7805) .This voltage regulator is a three terminal device. This voltage watchdog gives a fixed voltage level of 5V DC, which is used to drive the main control unit .ie microcontroller unit.

The power supply to the relays is obtained from the another step-down transformer which converts the 220V AC to 15V AC .The output of this transformer is given to the bridge rectifier.Output of this bridge rectifier is 15V DC.The rectified 15V DC is given to the voltage regulator (7812). This voltage regulator gives a fixed voltage level of 12V DC, which is used to drive the relays.

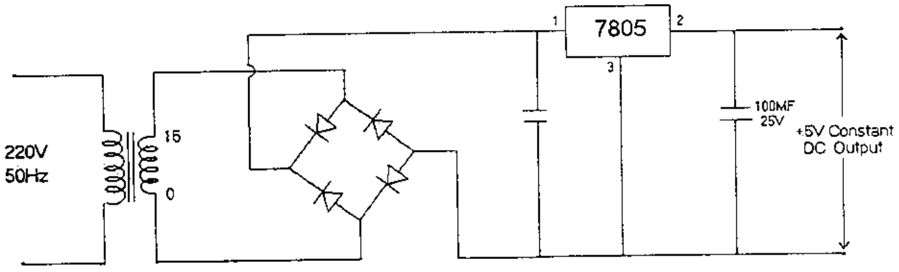
BRIDGE RECTIFIER

It is possible to design a full wave rectifier circuit in which the circuit can be fed directly from the line and also does not require a large peak inverse voltage rating for the diodes.Such a circuit is called bridge rectifier,since diodes are arranged in the form of bridge.During positive half cycle diode D2 and D4 conducts, developing the voltage across RL . D3 and D4 are reverse biased. During negative half cycle diodes D1and D3conducts while D2 and D4 are open resulting in a full wave rectified output.

ADVANTAGES :

- HIGH EFFICIENCY (81.2%)
- LESS PEAK INVERSE VOLTAGE PER DIODE
- SUITABLE FOR HIGH VOLTAGE APPLICATIONS

POWER SUPPLY DIAGRAMS



#.All Diodes are 1N4007

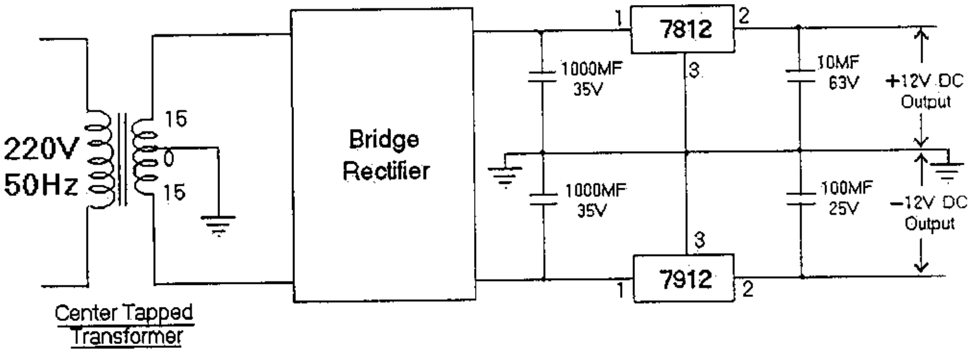


Fig : 5.1

VOLTAGE REGULATORS

IC voltage regulators offers low cost ,high reliability,small size and good performance.We are using 78XX and 79XX voltage regulators.The 78XX IC gives regulated positive output voltage and 79XX gives regulated negative output voltage.They are three terminal devices.

The standard block diagram is presented in Fig 5.1. The capacitor Cin is inserted between input and ground. It cancels the inductive effects due to long distribution leads .The transient response is highly improved by the output capacitor Cout.

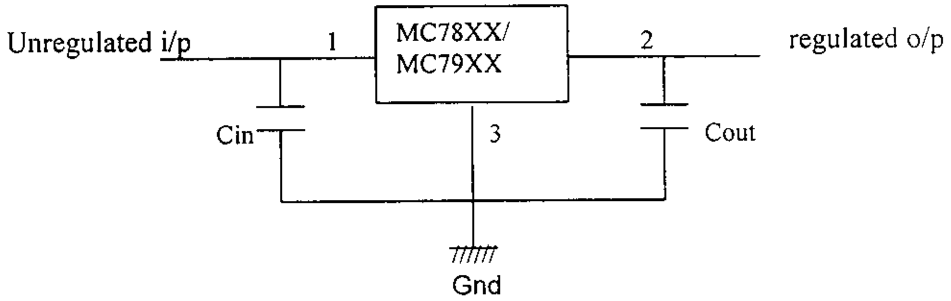


Fig 5.1 : General Regulator Circuitry

REGULATOR CHARACTERISTICS

The four import characteristics of the three terminal IC regulators are presented below.

- **Maximum load current (Io max)**

  The load may range from 0 to maximum output current. The IC is available with a heat sink in order to operated at the rated maximum output current.

- **Input voltage**

  The unregulated input voltage should be at least 2V greater than the regulated output.

- **Output voltage**

  The last two digits of 78XX series indicates the output voltage

- **Thermal shutdown**

  The IC consists of built in sensors. It turns off the IC when it gets too hot untill the IC has cooled suffciently. The range is usually 125C to 150C.

RELAY

The relay used here is an électromagnetic relay. This relay has a coil and a pair of contacts normally open.Once if the relay has to be operated,the coil is being energized.When the coil is energized ,it produces a magnetic field ,which magnetizes the magnet connected to it, and hence therefore the contact, which is a metal piece, gets attracted towards the magnet and the contact is made which is normally close (NC) condition.

Once if the coil is de-energized,the magnetic field gets vanished and contacts is released and hence it comes back to normal open (NO) condition.Thus the two statuses normally open (NO) and normal close (NC) are referred to OFF and ON states,which are used in driving the external load,given to the relay.

CHAPTER :6

SERIAL COMMUNICATION

SERIAL COMMUNICATION

Serial Communication, like any data transfer, requires coordination between the sender and receiver. For example, when to start the transmission and when to end it, when one particular bit or byte ends and another begins, when the receiver's capacity has been exceeded, and so on. A **protocol** defines the specific methods of coordinating transmission between a sender and receiver.

The scope of serial data transmission protocols is large and complex, encompassing everything from electrical connections to data encoding methods. This section summarizes the asynchronous protocol and standards related to using the Serial Driver.

For accessing the PC we use a RS-232 cable and a line driver IC for alternating the voltage levels. Here we do an asynchronous serial data communication .

ASYNCHRONOUS SERIAL COMMUNICATION PROTOCOL :

This section provides an overview of the protocol that governs the lowest level of data transmission .How serialized bits are sent over a single electrical line.

When a sender is connected to a receiver over an electrical connecting line, there is an initial state in which communication has not yet begun, called the **idle** or **Mark** state. Because older electromechanical devices operate more reliably with current continually passing through them, the **Mark** state employs a positive voltage level. Changing the state of the line by shifting the voltage to a negative value is called a **Space.** Once this change has occurred, the receiver interprets a negative voltage level as a 0 bit, and a positive voltage level as a 1 bit. These transitions are shown in Figure 6.1

The change from **Mark** to **Space** is known as the Start bit ,and this triggers the synchronization necessary for asynchronous serial transmission. The start bit delineates the beginning of the transmission unit defined as a character frame. The receiver then samples the voltage level at periodic intervals known as the bit time, to determine whether a 0-bit or a 1-bit is present on the line.

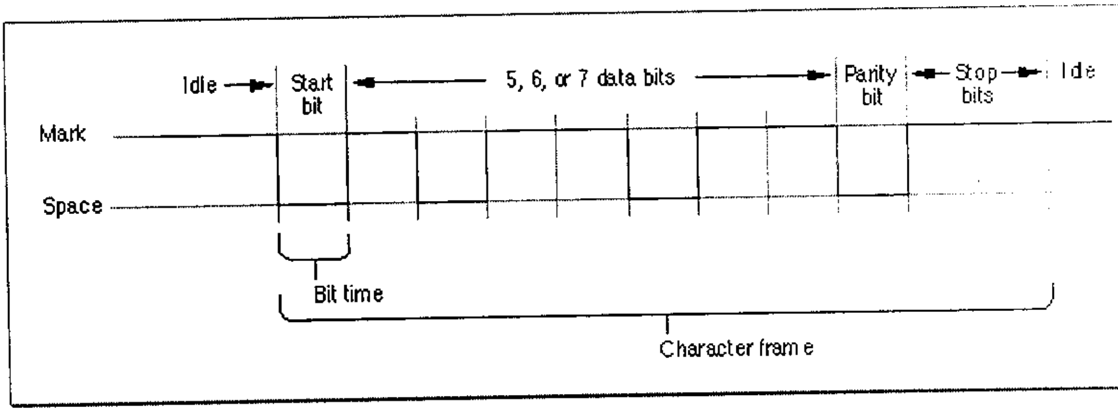


Fig 6.1

STANDARDS IN SERIAL COMMUNICATION:

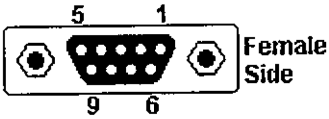
When data are transmitted as voltage, the commonly used standard is known as **RS-232C** .It is defined in reference to Data Terminal Equipment(DTE) and Data Communication Equipment(DCE). However,it's voltage levels are not compatible with TTL logic levels and so we use the line driver for data transmission into the PC.The rate of data transmission in RS-232C is restricted to a maximum of 20 Kbaud and a distance of 50 feet.

The voltage levels occupied are +3 V to +15 V as logic 0 and -3V to -15V as logic 1.The line driver converts logic 1 into approximately -9V and logic 0 into +9V.

BIT RATE:

Another important part of every asynchronous serial signal, is the bit rate at which the the data is transmitted. The rates at which the data is sent is based on the minimum speed of 300 bps (bits per second), you may find some slower speeds of 50, 100 and 150 bps, but these are not used in todays technologies. Faster speeds are all based on the 300 bps rate, you merely double the the preceding rate, so the rates are as follows, 600, 1200, 2400, 4800, 9600, 19200 and 38400 which is the fastest speed supported by today's BIOS's.In our project we are using a bit rate of 1200 bps for transmitting data to the PC.

<u>**PIN DETAILS:**</u>



DB- 9 PIN	SIGNAL NAME
1	CARRIER DETECT(CD)
2	RECEIVED DATA(RXD)
3	TRANSMITTED DATA(TXD)
4	DATA TERMINAL READY(DTR)
5	SIGNAL GROUND (GND)
6	DATA SET READY(DSR)
7	REQUEST TO SEND(RTS)
8	CLEAR TO SEND(CTS)
9	RING INDICATOR

CHAPTER : 7

ASSEMBLY

LANGUAGE PROGRAMMING

ASSEMBLY LANGUAGE PROGRAMMING DESCRIPTION

Various operations of the entire project is controlled by means of the assembly language programming done using the 89C51 instructions written in an assembler and then downloaded into the ATMEL IC.

There are as many as 111 instructions found in this 89C51 Assembler. This is the core of the project as it connects the different links in the hardware circuits and then performing the required operations.

The programming part can be divided into different modules.The modules are initializing the different ports,ring detection part, password detection part,password checking part and controlling the different equipment parts.

In the initializing part, we clear all the relays ,hook switch and we initialize a ring counter.It is done in the software part by using th clear instruction.

The Ring detection part ensures that the connection is being established between the two terminals after the requisite number of rings have been made.In our project we have set the number of rings to be two.This value can be changed to any number using the instructions. So after two rings connection is established between the caller and the terminal.

To prevent unauthorized users from accessing , a password is necessary.So we have built the password using the 89C51 instructions.In our project we have made the password to be equal to 7654 .This password is hardware programmed and it can't be changed very easily with an exception that it has to be again hardware programmed.

The password entering is done as four modules where one has to get four digits from the caller. If the password matches with the programmed number then the user is in a position to control the various devices.If the password entered is incorrect then the connection is lost and the user has to dial again along with the password.

We have different extensions for controlling the different devices. One is used to switch on the first device ,two is to switch off the first device, three to switch on the second device , four to switch off the second device and so on.

EXTENSIONS	DEVICE 1	DEVICE 2	DEVICE 3	DEVICE 4
1	**ON**	OFF	OFF	OFF
2	**OFF**	OFF	OFF	OFF
3	OFF	**ON** .	OFF	OFF
4	OFF	**OFF**	OFF	OFF
5	OFF	OFF	**ON**	OFF
6	OFF	OFF	**OFF**	OFF
7	OFF	OFF	OFF	**ON**
8	OFF	OFF	OFF	**OFF**

Multiple devices can also be controlled at the same time.That is first and second device can be switched on at the same time by pressing 1 and 3.All the four devices can be switched on at the same time by pressing the buttons 1,3,5 and 7.

This can be illustrated in the table as,

EXTENSIONS	DEVICE 1	DEVICE 2	DEVICE 3	DEVICE 4
1,5	ON	OFF	ON	OFF
1,7,5	ON	OFF	ON	ON
1,3,5,7	ON	ON	ON	ON

In order to maintain the devices that have been switched to remain in that same state after the end of the call, we use the number 0 in order to make the devices remain in that state.When pressing the number 0 the connection will be automatically cut off.

The PC is controlled by means of the extension numbers 7,8 and 9.The number 7 is used to switch on the computer .Before switching off the computer the shutdown subroutine has to be called. Number nine initializes the serial port to start and enable the bits.In the ser_out module which is called in the 9th module ,it transfers a data ‘S’ or ’T’ which is being transferred to the serial port of the PC.If the 9th pin is not pressed then the alphabet ‘T’ gets transferred. If the 9th key is pressed the alphabet ‘S’ gets transferred.If the alphabet ‘S’ is transferred then the shutdown program will be called which is written using C and Visual basic which will be discussed in the succeeding chapters.

A delay of about 50 seconds is given by means of a delay loop when the 9th pin is being pressed.No other device can be activated during this time. After the delay is over, then we have to press the 8th pin so that the computer power is being switched off.

The Algorithm and the flowchart of the assembly language program is given in the forthcoming section.

ALGORITHM:

STEPS:

1) Clear all the device port settings and initialize the ring counter.

2) Detect the number of rings and set the hook switch after the ring detection is over.

3) Check the number entered with the built-in password.

4) If correct then proceed ,else goto step 1

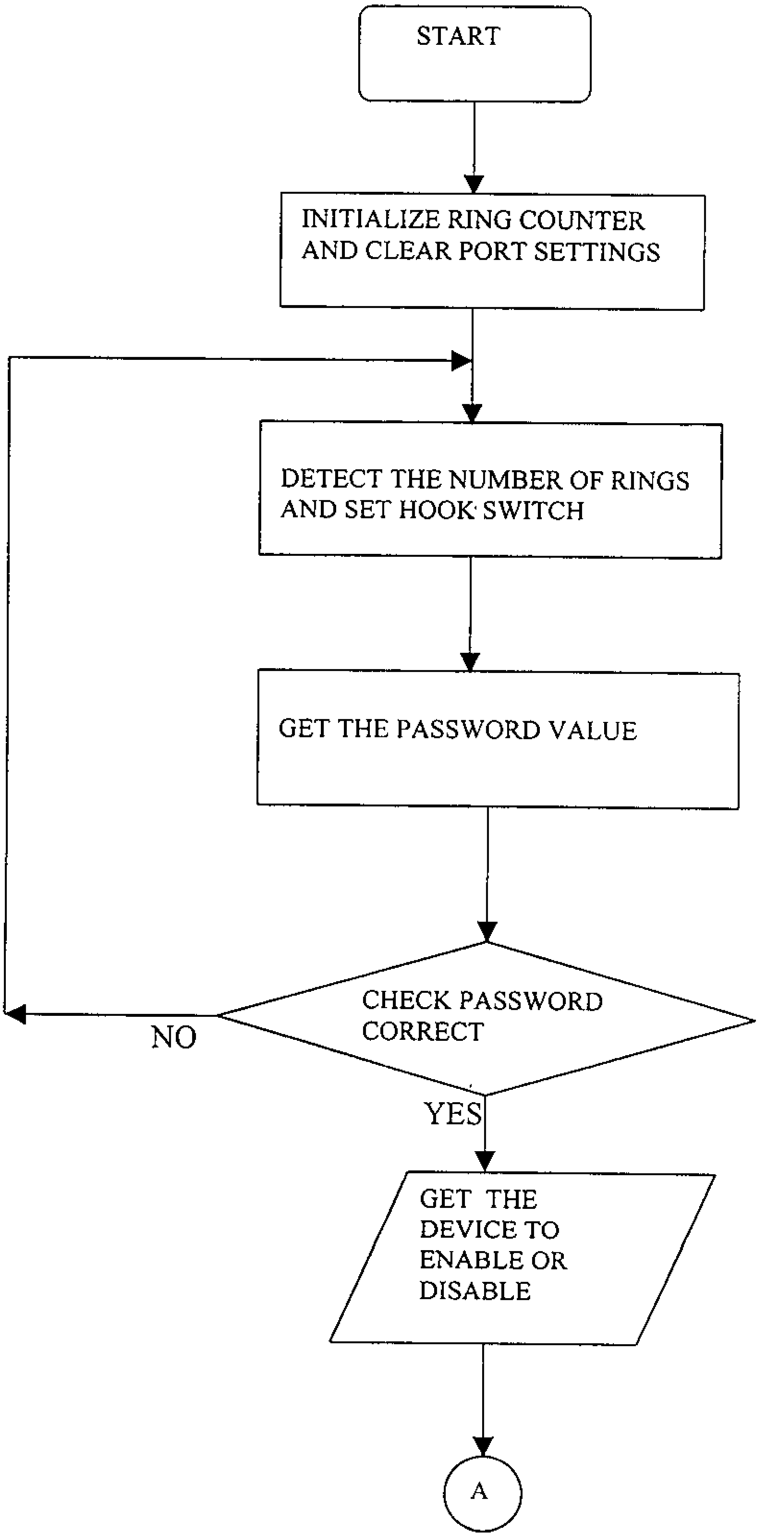
5) Get the device which has to be enabled or disabled

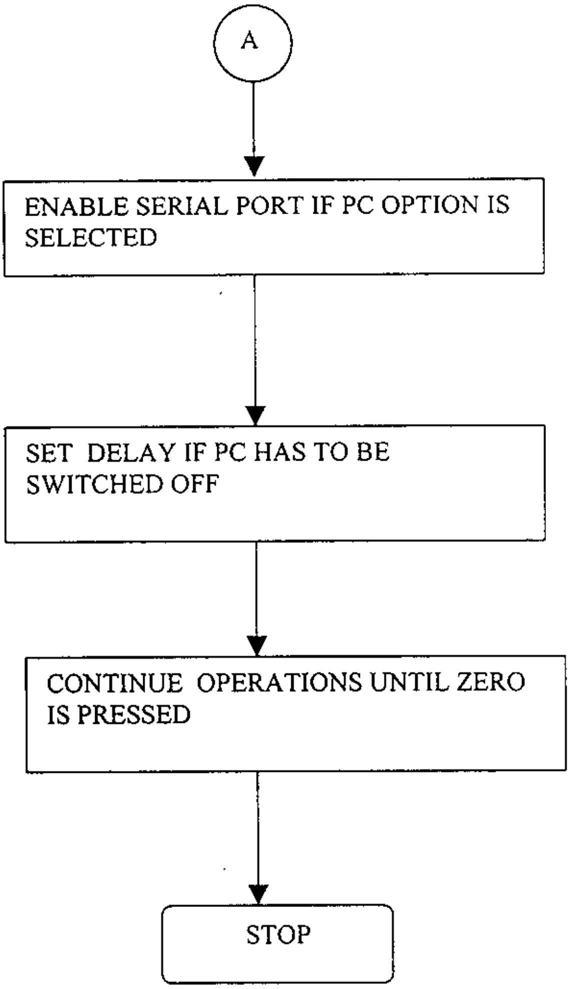
6) Generate a delay if the 9th pin is pressed.

7) Continue with step 5 until the 0th pin being pressed which disables the hook switch

8) End

FLOWCHART:


START
INITIALIZE RING COUNTER AND CLEAR PORT SETTINGS
DETECT THE NUMBER OF RINGS AND SET HOOK SWITCH
GET THE PASSWORD VALUE
CHECK PASSWORD CORRECT
NO
YES
GET THE DEVICE TO ENABLE OR DISABLE
A


A
ENABLE SERIAL PORT IF PC OPTION IS SELECTED
SET DELAY IF PC HAS TO BE SWITCHED OFF
CONTINUE OPERATIONS UNTIL ZERO IS PRESSED
STOP

ASSEMBLY LANGUAGE PROGRAMMING

CODING

The assembly language coding is given in this section.

```
0000                    org    0000h
0000   7f 00            mov    r7,#00h      ;initialize ring counter
0002   75 90 ff         mov    p1,#0ffh
0005   c2 b2            clr    p3.2         ;hook switch
0007   c2 b3            clr    p3.3         ;first relay
0009   c2 b4            clr    p3.4         ;second relay
000b   c2 b5            clr    p3.5         ;third relay
000d   c2 b6            clr    p3.6         ;fourth relay
000f   c2 b7            clr    p3.7
```

RING DETECTION

```
0011   7f 00       rings:    mov    r7,#00h
0013   30 95 fd    ring:     jnb    p1.5,ring
0016   20 95 fd    ring1:    jb     p1.5,ring1
0019   0f                    inc    r7
001a   ef                    mov    a,r7
001b   94 02                 subb   a,#02h
001d   40 f4                 jc     ring
001f   7f 00                 mov    r7,#00h
0021   d2 b2                 setb   p3.2      ;hook switch setting
```

FIRST PASS

```
0023   20 94 fd    back1:    jb     p1.4,back1   ;get new data
0026   30 94 fd    back:     jnb    p1.4,back
0029   e5 90       feed:     mov    a,p1
002b   54 0f                 anl    a,#0fh
002d   b4 01 0b              cjne   a,#01h,power1   ;one
0030   f5 30                 mov    30h,a
```

```
0032  75 80 01               mov    p0,#01h
0035  75 a0 00               mov    p2,#00h

0038  02 00 bc               ljmp   back2
003b  b4 02 0b      power1: cjne    a,#02h,power2    ;two
003e  f5 30                  mov    30h,a
0040  75 80 02               mov    p0,#02h
0043  75 a0 00               mov    p2,#00h
0046  02 00 bc               ljmp   back2
0049  b4 03 0b      power2: cjne    a,#03h,power3    ;three
004c  f5 30                  mov    30h,a
004e  75 80 03               mov    p0,#03h
0051  75 a0 00               mov    p2,#00h

0054  02 00 bc               ljmp   back2
0057  b4 04 0b      power3: cjne    a,#4h,power4
005a  f5 30                  mov    30h,a            ;four
005c  75 80 04               mov    p0,#04h
005f  75 a0 00               mov    p2,#00h

0062  02 00 bc               ljmp   back2
0065  b4 05 0b      power4:cjne     a,#5h,power5
0068  f5 30                  mov    30h,a            ;five
```

```
006a   75 80 05               mov    p0,#05h
006d   75 a0 00               mov    p2,#00h


0070   02 00 bc               ljmp   back2
0073   b4 06 0b     power5:cjne    a,#6h,power6
0076   f5 30                  mov    30h,a          ;six


0078   75 80 06               mov    p0,#06h
007b   75 a0 00               mov    p2,#00h


007e   02 00 bc               jmp    back2
0081   b4 07 0b     power6: cjne   a,#7h,power7
0084   f5 30                  mov    30h,a          ;seven
0086   75 80 07               mov    p0,#07h
0089   75 a0 00               mov    p2,#00h


008c   02 00 bc               ljmp   back2
008f   b4 08 0b     power7:  cjne  a,#8h,power8
0092   f5 30                  mov    30h,a          ;eight
0094   75 80 08               mov    p0,#08h
0097   75 a0 00               mov    p2,#00h


009a   02 00 bc               ljmp   back2
```

```
009d  b4 09 0b    power8: cjne   a,#9h,power9
00a0  f5 30               mov    30h,a          ;nine
00a2  75 80 09            mov    p0,#09h
00a5  75 a0 00            mov    p2,#00h

00a8  02 00 bc            ljmp   back2
00ab  b4 0a 0b    power9: cjne   a,#0ah,power0
00ae  f5 30               mov    30h,a
00b0  75 80 00            mov    p0,#00h        ;zero
00b3  75 a0 00            mov    p2,#00h

00b6  02 00 bc            ljmp   back2
00b9  02 00 23    power0: ljmp   back1
```

PASS 2

```
00bc  20 94 fd    back2:     jb    p1.4,back2
00bf  30 94 fd    back_2:    jnb   P1.4,back_2
```

```
00c2  e5 90                mov   a,p1
00c4  54 0f                anl   a,#0fh
00c6  b4 01 0b             cjne  a,#01h,power12
00c9  f5 31                mov   31h,a
00cb  75 80 01             mov   p0,#01h
00ce  75 a0 00             mov   p2,#00h


00d1  02 01 55             ljmp  back3
00d4  b4 02 0b   power12:  cjne  a,#02h,power22
00d7  f5 31                mov   31h,a
00d9  75 80 02             mov   p0,#02h
00dc  75 a0 00             mov   p2,#00h

00df  02 01 55             ljmp  back3
00e2  b4 03 0b   power22:  cjne  a,#03h,power32
00e5  f5 31                mov   31h,a
00e7  75 80 03             mov   p0,#03h
00ea  75 a0 00             mov   p2,#00h

00ed  02 01 55             ljmp  back3
00f0  b4 04 0b   power32:  cjne  a,#04h,power42
00f3  f5 31                mov   31h,a
```

```
00f5   75 80 04                 mov    p0,#04h
00f8   75 a0 00                 mov    p2,#00h

00fb   02 01 55                 ljmp   back3
00fe   b4 05 0b     power42:    cjne   a,#05h,power52
0101   f5 31                    mov    31h,a

0103   75 80 05                 mov    p0,#05h
0106   75 a0 00                 mov    p2,#00h

0109   02 01 55                 ljmp   back3
010c   b4 06 0b     power52:    cjne   a,#06h,power62
010f   f5 31                    mov    31h,a
0111   75 80 06                 mov    p0,#06h
0114   75 a0 00                 mov    p2,#00h

0117   02 01 55                 ljmp   back3
011a   b4 07 0b     power62:    cjne   a,#07h,power72
011d   f5 31                    mov    31h,a
011f   75 80 07                 mov    p0,#07h
0122   75 a0 00                 mov    p2,#00h

0125   02 01 55                 ljmp   back3
```

```
0128  b4 08 0b    power72: cjne   a,#08h,power82

012b  f5 31                mov     31h,a

012d  75 80 08             mov     p0,#08h

0130  75 a0 00             mov     p2,#00h


0133  02 01 55             ljmp    back3

0136  b4 09 0b    power82: cjne   a,#09h,power92

0139  f5 31                mov     31h,a

013b  75 80 09             mov     p0,#09h

013e  75 a0 00             mov     p2,#00h


0141  02 01 55             ljmp    back3

0144  b4 0a 0b    power92: cjne   a,#0ah,power02

0147  f5 31                mov     31h,a

0149  75 80 00             mov     p0,#00h

014c  75 a0 00             mov     p2,#00h


014f  02 01 55             ljmp    back3

0152  02 00 bc    power02  ljmp    back2
```

PASS 3

```
0155   20 94 fd    back3:      jb     p1.4,back3
0158   30 94 fd    back_3:     jnb    P1.4,back_3

015b   e5 90                   mov    a,p1
015d   54 0f                   anl    a,#0fh
015f   b4 01 0b                cjne   a,#01h,power13
0162   f5 32                   mov    32h,a
0164   75 80 01                mov    p0,#01h
0167   75 a0 00                mov    p2,#00h

016a   02 01 ee                ljmp   back4
016d   b4 02 0b    power13: cjne   a,#02h,power23
0170   f5 32                   mov    32h,a
0172   75 80 02                mov    p0,#02h
0175   75 a0 00                mov    p2,#00h

0178   02 01 ee                ljmp   back4
017b   b4 03 0b    power23:cjne    a,#03h,power33
017e   f5 32                   mov    32h,a
```

```
0180   75 80 03                  mov     p0,#03h
0183   75 a0 00                  mov     p2,#00h

0186   02 01 ee                  ljmp    back4
0189   b4 04 0b    power33: cjne    a,#04h,power43
018c   f5 32                     mov     32h,a


018e   75 80 04                  mov     p0,#04h
0191   75 a0 00                  mov     p2,#00h

0194   02 01 ee                  ljmp    back4
0197   b4 05 0b    power43: cjne    a,#05h,power53
019a   f5 32                     mov     32h,a
019c   75 80 05                  mov     p0,#05h
019f   75 a0 00                  mov     p2,#00h

01a2   02 01 ee                  ljmp    back4
01a5   b4 06 0b    power53:  cjne   a,#06h,power63
01a8   f5 32                     mov     32h,a
01aa   75 80 06                  mov     p0,#06h
01ad   75 a0 00                  mov     p2,#00h
```

```
01b0   02 01 ee                 ljmp   back4
01b3   b4 07 0b    power63:     cjne   a,#07h,power73
01b6   f5 32                    mov    32h,a
01b8   75 80 07                 mov    p0,#07h
01bb   75 a0 00                 mov    p2,#00h



01be   02 01 ee                 ljmp   back4
01c1   b4 08 0b    power73:     cjne   a,#08h,power83
01c4   f5 32                    mov    32h,a
01c6   75 80 08                 mov    p0,#08h
01c9   75 a0 00                 mov    p2,#00h
01cc   02 01 ee                 ljmp   back4

01cf   b4 09 0b    power83:     cjne   a,#09h,power93
01d2   f5 32                    mov    32h,a
01d4   75 80 09                 mov    p0,#09h
01d7   75 a0 00                 mov    p2,#00h

01da   02 01 ee                 ljmp   back4
01dd   b4 0a 0b    power93:     cjne   a,#0ah,power03
01e0   f5 32                    mov    32h,a
```

```
01e2   75 80 00                  mov     p0,#00h
01e5   75 a0 00                  mov     p2,#00h

01e8   02 01 ee                  ljmp    back4
01eb   02 01 55     power03:     ljmp    back3
```

PASS 4

```
01ee   20 94 fd     back4:       jb      p1.4,back4
01f1   30 94 fd     back_4:      jnb     P1.4,back_4
01f4   e5 90                     mov     a,p1
01f6   54 0f                     anl     a,#0fh
01f8   b4 01 0b                  cjne    a,#01h,power14
01fb   f5 33                     mov     33h,a
01fd   75 80 01                  mov     p0,#01h
0200   75 a0 00                  mov     p2,#00h

0203   02 02 8c                  ljmp    back5
0206   b4 02 0b     power14:     cjne    a,#02h,power24
0209   f5 33                     mov     33h,a
```

```
020b   75 80 02                  mov    p0,#02h
020e   75 a0 00                  mov    p2,#00h

0211   02 02 8c                  ljmp   back5
0214   b4 03 0b      power24:    cjne   a,#03h,power34
0217   f5 33                     mov    33h,a


0219   75 80 03                  mov    p0,#03h
021c   75 a0 00                  mov    p2,#00h

021f   02 02 8c                  ljmp   back5
0222   b4 04 0b      power34:    cjne   a,#04h,power44
0225   f5 33                     mov    33h,a
0227   75 80 04                  mov    p0,#04h
022a   75 a0 00                  mov    p2,#00h

022d   02 02 8c                  ljmp   back5
0230   b4 05 0b      power44:    cjne   a,#05h,power54
0233   f5 33                     mov    33h,a
0235   75 80 05                  mov    p0,#05h
0238   75 a0 00                  mov    p2,#00h
```

```
023b   02 02 8c                  ljmp    back5
023e   b4 06 0b      power54: cjne    a,#06h,power64
0241   f5 33                     mov     33h,a
0243   75 80 06                  mov     p0,#06h
0246   75 a0 00                  mov     p2,#00h



0249   02 02 8c                  ljmp    back5
024c   b4 07 0b      power64: cjne    a,#07h,power74
024f   f5 33                     mov     33h,a
0251   75 80 07                  mov     p0,#07h
0254   75 a0 00                  mov     p2,#00h

0257   02 02 8c                  ljmp    back5
025a   b4 08 0b      power74: cjne    a,#08h,power84
025d   f5 33                     mov     33h,a
025f   75 80 08                  mov     p0,#08h
0262   75 a0 00                  mov     p2,#00h

0265   02 02 8c                  ljmp    back5
0268   b4 09 0b      power84  cjne    a,#09h,power94
026b   f5 33                     mov     33h,a
```

```
026d  75 80 09               mov   p0,#09h

0270  75 a0 00               mov   p2,#00h


0273  02 02 8c               ljmp  back5

0276  b4 0a 0b     power94:  cjne  a,#0ah,power04

0279  f5 33                  mov   33h,a



027b  75 80 00               mov   p0,#00h

027e  75 a0 00               mov   p2,#00h


0281  02 02 8c               ljmp  back5

0284  02 01 ee     power04:  ljmp  back4
```

CHECKING OF PASSWORD

```
0287   c2 b2       err:      clr    p3.2
0289   02 00 11              ljmp   rings
028c   e5 30       back5:    mov    a,30h
028e   b4 07 03              cjne   a,#07h,err1
0291   02 02 97              ljmp   corr
0294   02 02 87    err1:     ljmp   err
0297   e5 31       corr:     mov    a,31h
0299   b4 06 f8              cjne   a,#06h,err1
029c   e5 32                 mov    a,32h


029e   b4 05 f3              cjne   a,#05h,err1
02a1   e5 33                 mov    a,33h
02a3   b4 04 ee              cjne   a,#04h,err1
```

DATA ENTRY

```
02a6   20 94 fd     new:      jb      p1.4,new
02a9   30 94 fd     new1:     jnb     p1.4,new1
02ac   e5 90                  mov     a,p1
02ae   54 0f                  anl     a,#0fh
02b0   b4 01 0b               cjne    a,#01h,is_for
02b3   d2 b4                  setb    p3.4
02b5   75 80 01               mov     p0,#01h
02b8   75 a0 00               mov     p2,#00h
02bb   02 02 a6               ljmp    new
02be   b4 02 0b     is_for:   cjne    a,#02h,is_rev


02c1   c2 b4                  clr     p3.4
02c3   75 80 02               mov     p0,#02h
02c6   75 a0 00               mov     p2,#00h
02c9   02 02 a6               ljmp    new
02cc   b4 03 0b     is_rev:   cjne    a,#03h,is_left
02cf   d2 b3                  setb    p3.3
```

```
02d1   75 80 03                mov    p0,#03h
02d4   75 a0 00                mov    p2,#00h
02d7   02 02 a6                ljmp   new
02da   b4 04 0b     is_left:   cjne   a,#04h,is_right
02dd   c2 b3                   clr    p3.3
02df   75 80 04                mov    p0,#04h
02e2   75 a0 00                mov    p2,#00h
02e5   02 02 a6                ljmp   new
02e8   b4 05 0b     is_right:  cjne   a,#05h,is_out1
02eb   d2 b5                   setb   p3.5
02ed   75 80 05                mov    p0,#05h
02f0   75 a0 00                mov    p2,#00h
02f3   02 02 a6                ljmp   new
02f6   b4 06 0b     is_out1:   cjne   a,#06h,is_out2
02f9   c2 b5                   clr    p3.5


02fb   75 80 06                mov    p0,#06h
02fe   75 a0 00                mov    p2,#00h
0301   02 02 a6                ljmp   new
0304   b4 07 0b     is_out2:   cjne   a,#07h,is_out3
0307   d2 b6                   setb   p3.6
0309   75 80 07                mov    p0,#07h
```

```
030c  75 a0 00              mov    p2,#00h
030f  02 02 a6              ljmp   new
0312  b4 08 0b   is_out3:cjne   a,#08h,is_out4
0315  c2 b6                 clr    p3.6
0317  75 80 08              mov    p0,#08h
031a  75 a0 00              mov    p2,#00h
031d  02 02 a6              ljmp   new
0320  b4 09 18   is_out4: cjne  a,#09h,is_last
0323  12 03 49              lcall  init_ser
0326  12 03 59              lcall  ser_out
0329  75 80 09              mov    p0,#09h
032c  75 a0 00              mov    p2,#00h
032f  75 7f c0              mov    7fh,#0c0h
0332  12 03 7a   xxxx:      lcall  del
0335  d5 7f fa              djnz   7fh,xxxx


0338  02 02 a6              ljmp   new
033b  b4 0a 08   is_last:   cjne   a,#0ah,is_final
033e  c2 b2                 clr    p3.2
0340  75 80 00              mov    p0,#00h
0343  75 a0 00              mov    p2,#00h
0346  02 00 11   is_final:  ljmp   rings
```

INITIALIZATION OF SERIAL PORT

```
  0349   75 89 20     init_ser:       mov     tmod,#20h       ; auto reload mode
  034c   75 8d e6                     mov     th1,#0e6h      ; count
  034f   75 88 40                     mov     tcon,#40h      ; start
  0352   75 98 58                     mov     scon,#58h       ; receive enble and
start
  0355   12 03 7a                     lcall   del
  0358   22                 ret
```

DATA TRANSFER

```
  0359                  ser_out:
  0359   74 53          mov     a,#"S"
  035b   f5 99          mov     sbuf,a          ; output to sport
  035d   12 03 7a       lcall   del
  0360   12 03 7a       lcall   del
  0363   12 03 7a       lcall   del
```

```
0366   75 98 58             mov    scon,#58h      ; ready for nxt data
0369   74 54                mov    a,#"T"
036b   f5 99                mov    sbuf,a         ; output to sport
036d   12 03 7a             lcall  del
0370   12 03 7a             lcall  del
0373   12 03 7a             lcall  del
0376   75 98 58             mov    scon,#58h      ; ready for nxt data
0379   22                   ret
```

DELAY

```
037a   7f ff    del:        mov    r7,#0ffh
037c   7a ff    loops1:     mov    r2,#0ffh
037e   da fe    loopd2:     djnz   r2,loopd2
0380   df fa                djnz   r7,loops1
0382   22                   ret
```

CHAPTER :8

SOFTWARE

SOFTWARE

INTRODUCTION

The software written in C and visual basic are used in our project to control the PC access.To switch on the PC we have to activate the relay which is connected between the power input to the PC and main power supply.To turnoff the PC in windows environment we have to call the windows inbuilt shutdown function.The software written in C is used to read the com port and visual basic code to call the shutdown program in windows environment.Windows based visual basic software makes shutdown program calling process easier.

ALGORITHM

STEPS:-

1 Initialize the com port (baud rate 1200, no parity, 8bits)

2 Read the data from the com port using the C program

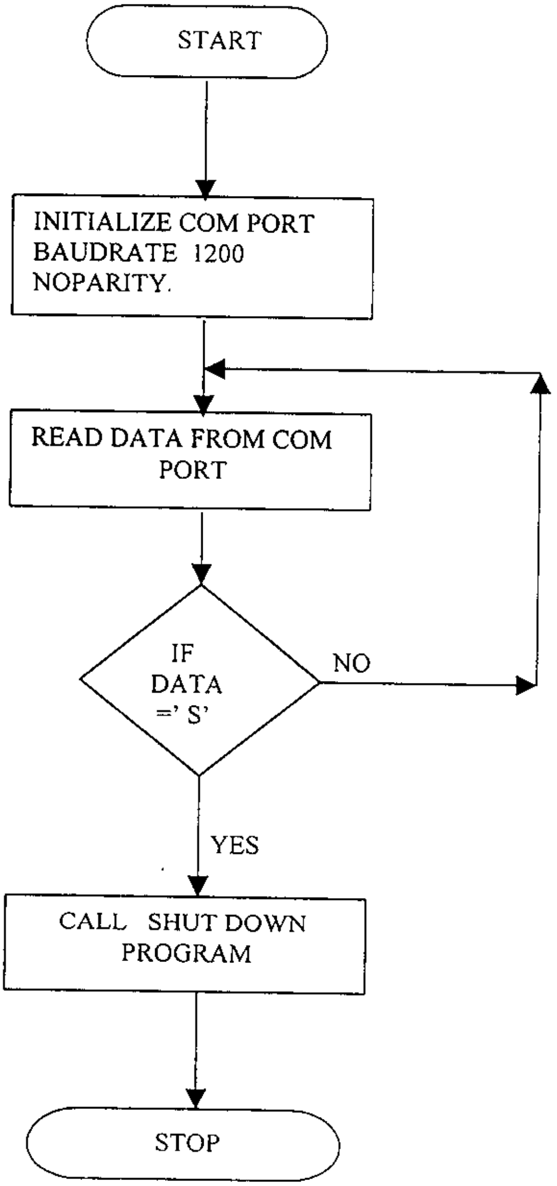
3 Open a file and write the data which is read from com port in to that file

4 If the data read from the port equal to character '**S**' call the shutdown program.

5 If the data which is read from the com port not equal to character '**S**' then go to the step 2 .

6 End

FLOWCHART


START
INITIALIZE COM PORT
BAUDRATE 1200
NOPARITY.
READ DATA FROM COM
PORT
IF
DATA
=' S'
NO
YES
CALL SHUT DOWN
PROGRAM
STOP

DESCRIPTION

To turn on the PC the microcontroller will send a signal to activate the relay.The relay is an electromechanical relay,which is connected between power input to the PC and main power supply. To turn off the PC microcontroller will send a character to the PC through com port. The C program, which is running inside PC, will read that data and it store the data in a file called data file.The visual basic program which is running inside the PC read that file .If the data inside the file equal to character 's', then shut-down program is called. The C and VB program are kept in autoexec path to run the programs on background .

The C and the VB programs are given in the following section.

C PROGRAM

```
#include<process.h>
#include<stdio.h>
#include<conio.h>
#include<dos.h>
#include<ctype.h>
main()
{
        FILE *fp;
        int i,b;
        char a;
        system("mode com1,1200,n,8,1");
        clrscr();
        for(;;)
        {
          aa:
          a=inportb(0x3f8);
          printf("%c",a);
          delay(200);
          fp=fopen("sample.dat","w");
        fprintf(fp,"%c",a);
        fcloseall();
        if (a!='S')
        {
        goto aa;
         }
         fp=fopen("sample.dat","w");
         fprintf(fp,"%c",a);
       fcloseall();
}
}
```

VB PROGRAM

```
Private Sub Form_Load()
    Dim s As String
    Open "c:\a.txt" For Input As #1
    Do Until EOF(1)
      Input #1, s$
    Loop
    If (s$ = "S") Then
       Shell ("c:\Windows\Rundll user,ExitWindows")
    End If
    Close #1
    Kill "c:\a.txt"
    End
End Sub
```

CONCLUSION

CONCLUSION

The telephone line approach is a very flexible circuit realization which can be widened to accommodate a much wider scope .Since the entire operation is microcontroller based any wanted changes or modifications can be achieved through software-the hardware remaining unaffected.

The PC communication aspect can be expanded to access any networking environment. Ample security measures provided ,definitely enhance the application.

Various other applications include mail access, switching equipment, industrial parameter control, data transmission and reception.

This circuit can be viewed as a multi objective application, which serves as a simple but effective appliance control system.

FUTURE IMPROVEMENTS

This application can be considered as the building block for various other advanced circuit realizations.

Theoritically 'n' number of passwords can be provided, the typing of which can provide access to different applications (industrial and domestic).

These may include temperature control, network security, humidity control and similar processes.

Data present in the hard disk of two individual PC's can be transferred using this application in its advanced stage.

BIBILIOGRAPHY

I) Ramesh S Gaonkar: Microprocessor Architecture Programming and applications, Penram International

2) Douglas V Hall: Microprocessors and Interfacing. Programming And Hardware, Tata McGraw Hill

3) I.P.Singh: Microprocessor and their applications-Indian Institute Of Technology,NewDelhi.

4) ATMEL Data Sheets For 89C51 Microcontroller.

5) D Roy Choudary and Shail Jain: Linear Integrated Circuits.

6) Programming in C, Schaum's Series, Tata McGraw Hill

7) Let Us C, Yashvant Kanetkar, BPB Publications

8) Programming in Visual Basic, Gary Korner, Tata McGraw Hill

APPENDIX

Features

- Compatible with MCS-51™ Products
- 4K Bytes of In-System Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial Channel
- Low-power Idle and Power-down Modes

Description

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 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 Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

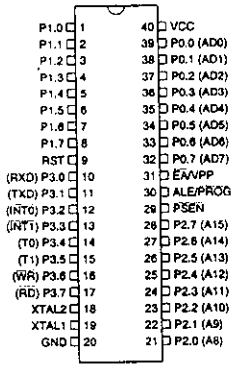
8-bit Microcontroller with 4K Bytes Flash

AT89C51

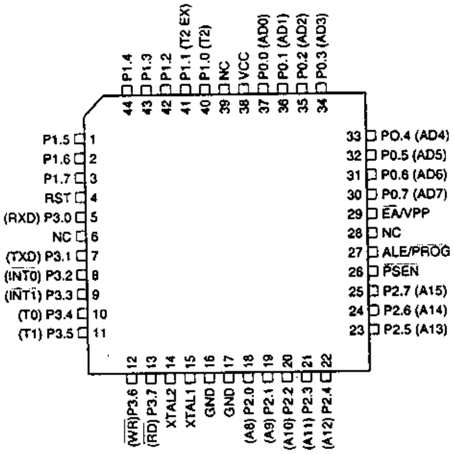
Rev. 0265G–02/0

Pin Configurations

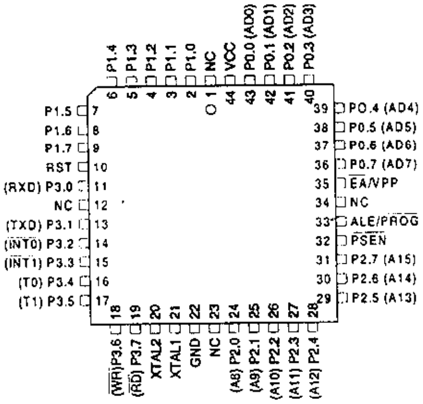
PDIP



PQFP/TQFP



PLCC



he AT89C51 provides the following standard features: 4K ytes of Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit mer/counters, a five vector two-level interrupt architecture, full duplex serial port, on-chip oscillator and clock ciruitry. In addition, the AT89C51 is designed with static logic or operation down to zero frequency and supports two oftware selectable power saving modes. The Idle Mode tops the CPU while allowing the RAM, timer/counters, erial port and interrupt system to continue functioning. The ower-down Mode saves the RAM contents but freezes he oscillator disabling all other chip functions until the next ardware reset.

Pin Description

VCC

Supply voltage.

GND

Ground.

Port 0

Port 0 is an 8-bit open-drain bi-directional 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 may 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 bi-directional 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.

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

Port 2

Port 2 is an 8-bit bi-directional 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, it 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 bi-directional 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 AT89C51 as listed below:

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)

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

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

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

Address Latch Enable 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 is the read strobe to external program memory.

When the AT89C51 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, for parts that require 12-volt V_{PP}.

XTAL1

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

XTAL2

Output from the inverting oscillator amplifier.

Oscillator Characteristics

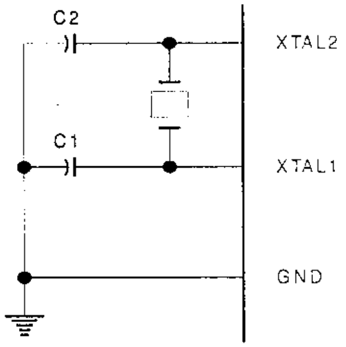
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

It should be noted that when idle is terminated by a hard ware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

Figure 1. Oscillator Connections

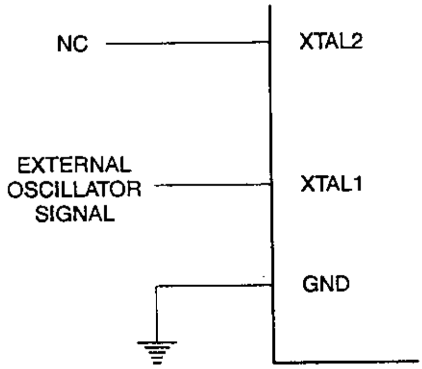


Note: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Status of External Pins During Idle and Power-down Modes

Mode	Program Memory	ALE	$\overline{\text{PSEN}}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

Figure 2. External Clock Drive Configuration



Power-down Mode

In the power-down mode, the oscillator is stopped, and the instruction that invokes power-down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power-down mode is terminated. The only exit from power-down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

Program Memory Lock Bits

On the chip are three lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below.

When lock bit 1 is programmed, the logic level at the $\overline{EA}$ pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value, and holds that value until reset is activated. It is necessary that the latched value of $\overline{EA}$ be in agreement with the current logic level at that pin in order for the device to function properly.

Lock Bit Protection Modes

Program Lock Bits				
	LB1	LB2	LB3	Protection Type
1	U	U	U	No program lock features
2	P	U	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on reset, and further programming of the Flash is disabled
3	P	P	U	Same as mode 2, also verify is disabled
4	P	P	P	Same as mode 3, also external execution is disabled

Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0V to +7.0V
Maximum Operating Voltage	6.6V
DC Output Current	15.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

T_A = -40°C to 85°C, V_{CC} = 5.0V ± 20% (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low-voltage	(Except $\overline{EA}$)	-0.5	0.2 V_{CC} - 0.1	V
V_{IL1}	Input Low-voltage ($\overline{EA}$)		-0.5	0.2 V_{CC} - 0.3	V
V_{IH}	Input High-voltage	(Except XTAL1, RST)	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V
V_{IH1}	Input High-voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC} + 0.5	V
V_{OL}	Output Low-voltage[1] (Ports 1,2,3)	I_{OL} = 1.6 mA		0.45	V
V_{OL1}	Output Low-voltage[1] (Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA		0.45	V
V_{OH}	Output High-voltage (Ports 1,2,3, ALE, $\overline{PSEN}$)	I_{OH} = -60 μA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -25 μA	0.75 V_{CC}		V
		I_{OH} = -10 μA	0.9 V_{CC}		V
V_{OH1}	Output High-voltage (Port 0 in External Bus Mode)	I_{OH} = -800 μA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -300 μA	0.75 V_{CC}		V
		I_{OH} = -80 μA	0.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	V_{IN} = 0.45V		-50	μA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	V_{IN} = 2V, VCC = 5V ± 10%		-650	μA
I_{LI}	Input Leakage Current (Port 0, $\overline{EA}$)	0.45 < V_{IN} < V_{CC}		±10	μA
RRST	Reset Pull-down Resistor		50	300	KΩ
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		20	mA
		Idle Mode, 12 MHz		5	mA
	Power-down Mode[2]	V_{CC} = 6V		100	μA
		V_{CC} = 3V		40	μA

Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
Maximum I_{OL} per port pin: 10 mA
Maximum I_{OL} per 8-bit port: Port 0: 26 mA
Ports 1, 2, 3: 15 mA
Maximum total I_{OL} for all output pins: 71 mA
If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

2. Minimum V_{CC} for Power-down is 2V.

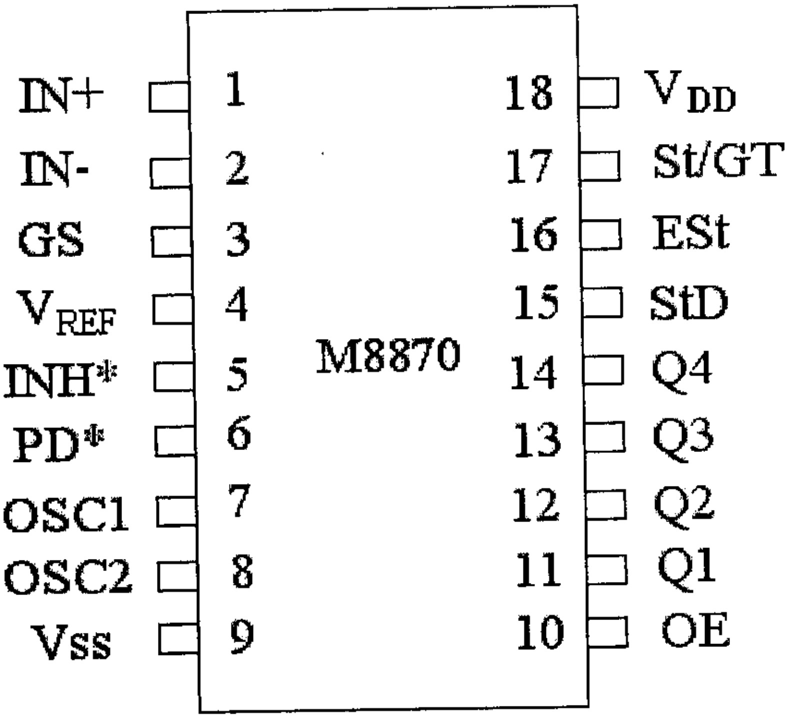
AC Characteristics

Under operating conditions, load capacitance for Port 0, ALE/$\overline{PROG}$, and $\overline{PSEN}$ = 100 pF; load capacitance for all other outputs = 80 pF.

External Program and Data Memory Characteristics

		12 MHz Oscillator		16 to 24 MHz Oscillator		
Symbol	Parameter	Min	Max	Min	Max	Units
$1/t_{CLCL}$	Oscillator Frequency			0	24	MHz
t_{LHLL}	ALE Pulse Width	127		$2t_{CLCL}$-40		ns
t_{AVLL}	Address Valid to ALE Low	43		t_{CLCL}-13		ns
t_{LLAX}	Address Hold After ALE Low	48		t_{CLCL}-20		ns
t_{LLIV}	ALE Low to Valid Instruction In		233		$4t_{CLCL}$-65	ns
t_{LLPL}	ALE Low to $\overline{PSEN}$ Low	43		t_{CLCL}-13		ns
t_{PLPH}	$\overline{PSEN}$ Pulse Width	205		$3t_{CLCL}$-20		ns
t_{PLIV}	$\overline{PSEN}$ Low to Valid Instruction In		145		$3t_{CLCL}$-45	ns
t_{PXIX}	Input Instruction Hold After $\overline{PSEN}$	0		0		ns
t_{PXIZ}	Input Instruction Float After $\overline{PSEN}$		59		t_{CLCL}-10	ns
t_{PXAV}	$\overline{PSEN}$ to Address Valid	75		t_{CLCL}-8		ns
t_{AVIV}	Address to Valid Instruction In		312		$5t_{CLCL}$-55	ns
t_{PLAZ}	$\overline{PSEN}$ Low to Address Float		10		10	ns
t_{RLRH}	$\overline{RD}$ Pulse Width	400		$6t_{CLCL}$-100		ns
t_{WLWH}	$\overline{WR}$ Pulse Width	400		$6t_{CLCL}$-100		ns
t_{RLDV}	$\overline{RD}$ Low to Valid Data In		252		$5t_{CLCL}$-90	ns
t_{RHDX}	Data Hold After $\overline{RD}$	0		0		ns
t_{RHDZ}	Data Float After $\overline{RD}$		97		$2t_{CLCL}$-28	ns
t_{LLDV}	ALE Low to Valid Data In		517		$8t_{CLCL}$-150	ns
t_{AVDV}	Address to Valid Data In		585		$9t_{CLCL}$-165	ns
t_{LLWL}	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	$3t_{CLCL}$-50	$3t_{CLCL}$+50	ns
t_{AVWL}	Address to $\overline{RD}$ or $\overline{WR}$ Low	203		$4t_{CLCL}$-75		ns
t_{QVWX}	Data Valid to $\overline{WR}$ Transition	23		t_{CLCL}-20		ns
t_{QVWH}	Data Valid to $\overline{WR}$ High	433		$7t_{CLCL}$-120		ns
t_{WHQX}	Data Hold After $\overline{WR}$	33		t_{CLCL}-20		ns
t_{RLAZ}	$\overline{RD}$ Low to Address Float		0		0	ns
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	t_{CLCL}-20	t_{CLCL}+25	ns

PIN CONFIGURATION OF M-8870


M8870
IN+ 1
IN- 2
GS 3
V_REF 4
INH* 5
PD* 6
OSC1 7
OSC2 8
Vss 9
18 V_DD
17 St/GT
16 ESt
15 StD
14 Q4
13 Q3
12 Q2
11 Q1
10 OE