WIRELESS CONTROL OF MULTIPLE DEVICES USING INTERNET

PROJECT REPORT

SUBMITTED BY

D. GIRI PRASSAD

M. MUKILAN EZHINI

C. PERUMAL KARTHIK

N. VIJAYAN

GUIDED BY

Ms. A. VASUKI M.E., M.I.S.T.E.

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF
BACHELOR OF ENGINEERING IN
ELECTRONICS & COMMUNICATION ENGINEERING
OF BHARATHIAR UNIVERSITY

2000 - 2001

Department of Electronics & Communication Engineering

Kumaraguru College of Technology

Coimbatore - 641 006.

CONTENTS

ACKNOWLEDGEMENT

It is our foremost duty to thank our Principal Dr. K.K. Padmanaban B.Sc.(Eng.), M.Tech., Ph.D., for his kind patronage. We express our hearty gratitude to our Professor and Head of the Department Prof. M. Ramasamy M.E., MISTE, MIEEE(USA), MIE(Engg), for his support, advice and for providing us the resources and time to complete this project.

We thank our project guide Ms. A. Vasuki M.E., MISTE for devoting her time and effort to guide us. We would always be grateful for her kind cooperation, constructive criticism and alacrity.

We would fail from our duty if we do not thank our beloved class advisor Mrs. R. Latha Sakthivel B.E., MISTE., who has always been generous in permitting us to spend more time on our project even during hectic working days. Last but not the least, we thank our loving parents, family members and our dear friends for their love and constant encouragement.

SYNOPSIS

SYNOPSIS

The information age has shrunk the world to a global village. The emergence of wireless and the Internet has transformed the concept of physical presence to more sublime and out of the world virtual presence. Devices like pagers, cell phones and PDAs have made people on the move accessible anywhere anytime. With people at the helm of affairs constantly on the move, the control of devices and machines from remote locations has become a necessity. The most efficient and cost effective way of remote access is the all-pervasive Internet.

Our project aims at controlling multiple devices present at a remote location through the Internet. A user from any location on earth with an Internet connection can access a web page from a web server. Through this page the user can see the status of the devices and change the state of any device. This information is updated in the server, which is reflected in a host computer, to which the device control hardware is connected.

The host computer signals the hardware to change the status of the designated device. Since the device may be located at a distant location from the host computer, this transfer has been implemented using a wireless link. The transmitted data is received at the device end, decoded and is given to a microcontroller, which controls the device.

The main advantage of our project is that it is simple and cost effective. The whole system is maintenance-free and requires no supervision. The system can be easily modified to suit the specific needs of any industry. Above all, it saves a lot of time, money and energy by allowing users to control devices from anywhere on the globe.

INTRODUCTION

1. INTRODUCTION

The communication field is in the midst of a boom with the advent of Internet and a host of new networking technologies. In this wired world, the ability to control devices from remote locations is fast becoming a reality. With the Internet becoming an indispensable part of a technocrat's life, the choice of Internet as a medium of data exchange is practical and cost effective.

Practically, all the devices cannot be located near the host computer. Moreover, assigning a host system for every device makes the solution expensive and impractical. Hence the system of wireless transfer has been devised to control multiple devices from a single host computer. The number of devices can be increased for the system to be tailored to suit specific purposes.

Our project can be categorized into four major blocks, namely:

1. ISA Add-On Card
2. Internet Interface Block
3. RF Transmission Block
4. RF Reception Block

The overall block diagram of the project is given in **Fig. 1.1**.

1.1 ISA ADD-ON CARD

The personal computer has many expansion slots, which serve to add additional functionality to the PC. Our project includes an add-on card designed for the ISA slot. This card has an 8255A PPI that is used to provide a parallel port to send data from the PC to external circuits. The ISA card has an address buffer, a data buffer and a decoder. A comparator is present to recognize the address sent and enable the address and data buffers. The decoder is used to select the ports of 8255A.

1.2. INTERNET INTERFACE BLOCK

The Internet interface block is used to provide connectivity to users from all over the world. The coding for this interface is done using Active Server Pages and VBScript. The back-end database in the web server is a MS Access database. The database stores information about the devices and their current status.

A user can access the web page residing in the web server by keying in a login name and a password. After authenticating the user name and password, the user is given access to the control page. The control page contains the device names and their status. The user can modify the status of any device by clicking on the check box provided for this purpose. After changing the status of the devices, the user clicks the 'Submit' button, which will store these changes in the server-side database.

A host computer is kept connected to the Internet. The device control hardware is connected to this computer using an ISA add-on card. The database residing in the web server is accessed through another web page that is coded using ASP and VBScript. This page is refreshed at a periodic interval of 10 seconds. The data from this web page are written to a disk-based text file. This text file is accessed by a C program, which sends appropriate data to the transmitter block through the ISA add-on card.

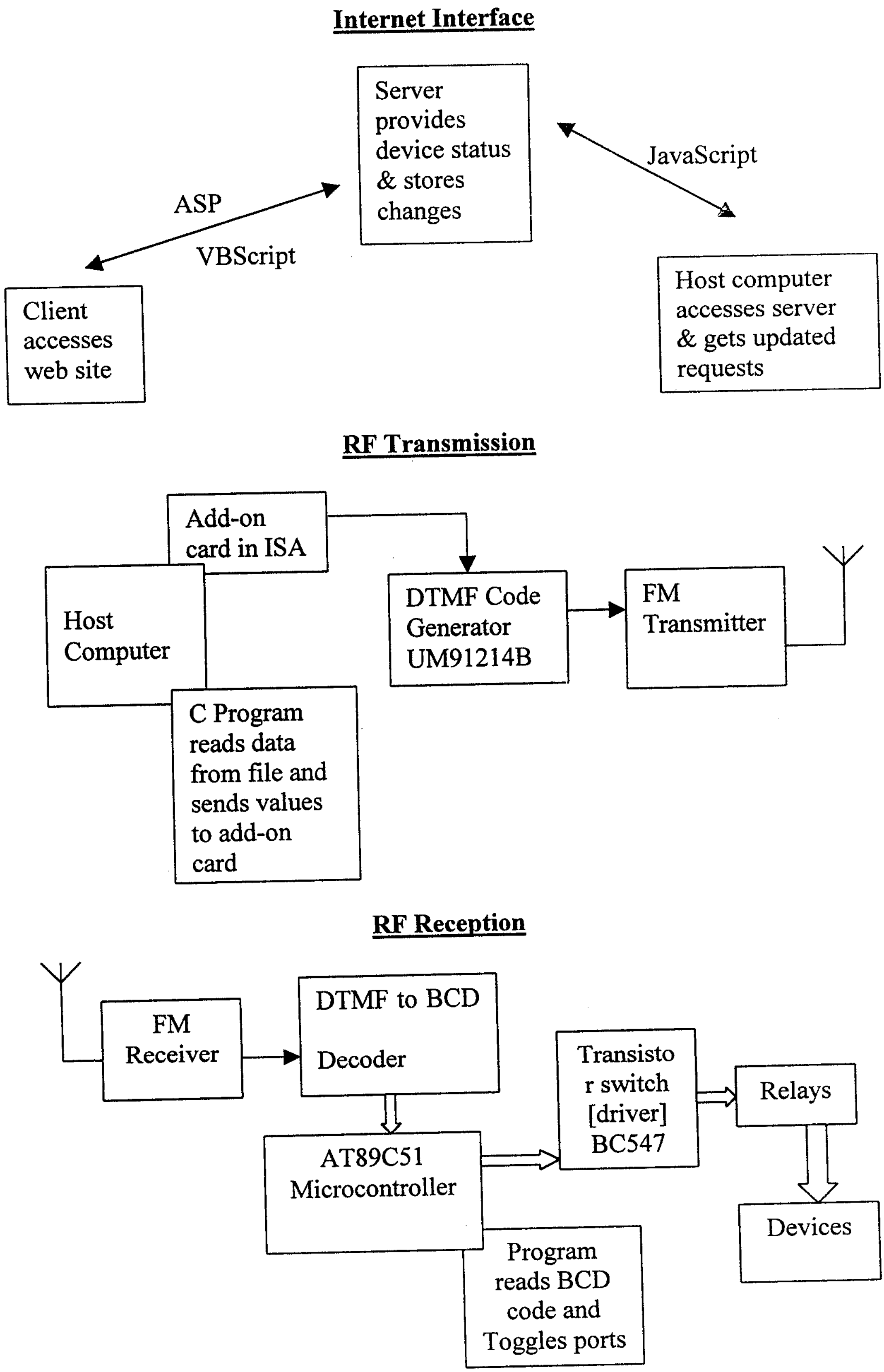
1.3. RF TRANSMISSION BLOCK

The host computer is typically located in an industry premises. The devices to be controlled may be present at remote locations within the industry. To control these devices in a hassle-free manner, a wireless approach is employed. The ISA add-on card is connected to the external circuit using a BC547 transistor driver circuit. The external circuit consists of a DTMF code generator UM91214B. This IC generates DTMF signals according to the input code from the ISA card. A FM transmitter is used to transmit this DTMF signal to a remote location.

1.4. RF RECEPTION BLOCK

An FM receiver (CXA1019S) is employed to receive the information from the FM transmitter. The received DTMF code is sent to DTMF decoder M-8870, which decodes the DTMF code into BCD code. This BCD data is given to an AT89C51 microcontroller. The microcontroller interprets this data and toggles the state of the corresponding ports. The ports are connected to transistor drivers that drive the relays. The relays are used to switch multiple devices, which may operate at different voltages.

Fig. 1.1: Project Block Diagram



Internet Interface

Server provides device status & stores changes
ASP
VBScript
JavaScript
Client accesses web site
Host computer accesses server & gets updated requests

RF Transmission

Add-on card in ISA
Host Computer
C Program reads data from file and sends values to add-on card
DTMF Code Generator UM91214B
FM Transmitter

RF Reception

FM Receiver
DTMF to BCD Decoder
AT89C51 Microcontroller
Program reads BCD code and Toggles ports
Transisto r switch [driver] BC547
Relays
Devices

ISA ADD ON CARD

2. ISA ADD-ON CARD

All personal computers have expansion slots to allow the use of add on cards. These facilitate the addition of new functionality and ports. The different types of slots available on PCs are ISA, EISA and PCI. By using an ISA slot for sending the data to the transmitter, the parallel and serial ports are available for other purposes. Since ISA slots have been replaced by EISA slots in most PCs, our add-on card has been designed for the EISA slot.

The block diagram of the ISA card is shown in **Fig. 2.1**. To design the add-on card with selection facility an eight-bit comparator (74688) is used. One input of the comparator is given to the address bus of the PC. Another one is given to DIP switches, with which the address can be selected. Instead of the DIP switches the pins have been shorted to fix a specific address. After the comparator is selected with the desired address, the comparator output will enable a three-to-eight decoder (74138). After enabling this decoder the three inputs from the address bus are given to the decoder. Depending on the decoder inputs A, B & C one of the eight outputs Y0 to Y7 will be selected. These Y0 to Y7 outputs are connected to the chip enable (CE) pin of the 8255. The address is held in the 74244 buffer and the data is held in the 74245 bi-directional buffer. The 74688 comparator enables both these buffers.

Since the pins for dip switches are shorted, the address of the card is from 300h to 303h. 300h is port A, 301h is port B, 302h is port C and 303h is the control register.

2.1. EXPANSION SLOT SIGNALS

The ISA slot has many signals that are needed for the interfacing of the ISA add-on card with the CPU. These signals are described below.

2.1.1. Address Bus (A0-A10):

The address bus signals are available at these pins. These signals are unidirectional coming from the motherboard to the add-on card. These signals are generated either by the CPU or by a DMA controller.

2.1.2. Data Bus (D0-D17):

The data bus signals of the motherboard are available at these pins. These signals are bi-directional. The CPU for reading or writing data from memory or I/O ports uses the data bus.

2.1.3. Control Bus:

MEMW:

These signals indicate the process of memory operation. The CPU or the DMA controller generates this signal. Memory uses these signals for storing the data available in the bus in the location specified by the address available in the address bus.

MEMR:

These signals indicate the process of memory read operation. This signal is generated either by CPU or by DMA controller. When this signal is low, memory is on the address bus.

I/O Read:

The signal indicates the selected input port to send data to the data bus. The signal is generated either by the CPU or by the DMA controller.

I/O Write:

This signal indicates the selected output port to take data from the data bus. This signal is generated either by the CPU or by the DMA controller.

Address Latch Enable (ALE):

This signal is generated by in the motherboard to latch the address from CPU. This signal is provided at the I/O port to indicate the CPU address bus cycle.

2.1.4. Clock signals:

Oscillator:

A 14.318 MHz signal is available at this pin. The signal is produced in the CPU card. The duty cycle of this signal is 50%.

Clock:

The 4.77 MHz signal generated is available at this pin. The CPU uses this signal as its clock input. The duty cycle of the signal is 33%.

2.1.5. Interrupt Signals:

IRQ 2:

This signal is not used in PC's.

IRQ 3:

The first serial port COM 1 raises this signal. This signal should be pulsing during COM 1 operation.

IRQ 4:

This signal is the interrupt raised by second serial port COM 2. The signal should be pulsing during COM 2 operation.

IRQ 5:

This signal is the interrupt raised by hard disk controller. This signal should be pulsing for hard disk operation.

IRQ 6:

This signal is the interrupt raised by the floppy disk controller. The signal should be pulsing for floppy disk controller.

IRQ 7:

This signal is the interrupt raised by the pointer controller as a request for data transfer. This signal should be pulsing during printer operation.

2.1.6. DMA signals:

DACK0 (DMA acknowledge):

This is the DMA acknowledgement signal for DRQ 0 request for DRAM refresh.

DREQ1 (DMA request):

This signal is not used in PC.

DACK (DMA acknowledge):

This signal is not used in PC.

DRQ 2:

Floppy disk controller generates this signal for every byte of data transferred. This signal should be pulsing for floppy disk operation. This is related to IRQ 6 signal.

DACK 2:

This is an acknowledge signal from the DMA controller in response to the request made by the floppy disk controller. This signal should be pulsing for normal floppy disk operation.

DRQ 3:

This signal is generated by the hard disk controller for every byte transfer to request the DMA whenever a data transfer is required. This signal should be pulsing during hard disk operation. This signal is related to IRQ 5 signal.

DACK 3:

This is an acknowledge signal from the DMA controller in response to the request made by hard disk controller. This signal should be pulsing during hard disk operation.

AEN:

This signal is generated by the motherboard when HOLD signal is given by controller. When AEN is high it means that DMA controller has been given control of the bus. At this time, DMA controller generates the address, date and control signals.

Terminal Count:

At the end of every DMA operation an EOP signal is generated by DMA controller. T/c signal is generated in the motherboard using EOP signal. This pulse indicated the DMA operation for one of the active DMA channels is completed, i.e. count had become zero.

Power:

GND,+5V,-5V,+12V & -12V are obtained from SMPS. These voltages are used by add-on card connected in the I/O slot.

2.1.7.Other Signals:

Reset DRV:

This signal is generated by clock generator during manual reset or power on reset. This is used by all add-on cards connected in the I/O slot to rest or initialize their internal circuit when the system is reset.

I/O Channel Check:

This signal is made low when RAM parity error is detected by a memory board in the I/O slot. The signal is used to generate NMI interrupt to the CPU to indicate a parity error.

I/O Channel Ready:

This signal is made low by the slow I/O device to make the CPU to introduce WAIT state. Whenever the signal is low, the CPU will not perform any bus operation.

2.1.8. ISA Slot Configuration:

A2-A9	-	Data line
A11	-	AEN
A28	-	A3
A29	-	A2
A30	-	A0
A31	-	A1
B31	-	GND
B2	-	RST
B3 & 29	-	Vcc
B13	-	WR
B14	-	RD
B22-B27	-	A9-A4

2.1.9. Buffer:

The buffer is a logic circuit that amplifies the current or power. It has one input line and one output line. The logic level of output is same as that of input. The buffer is primarily used to increase the driving capacity of the logic circuit.

A tristate logic device has 3 logic states: logic 0, logic 1 and high impedance state. The term tristate is a trademark of National Semiconductor and used to represent three logic states. A tristate device has a third line called enable. When this is activated, the tristate device function the same way as the logic device. When the third line is disabled, the logic device goes into high impedance state.

2.1.10. Hardware Operations:

Base Address Switching:

The base address is set by jumpers or determined by shorted pins available in the add-on card.

Address Comparison:

The base address is compared with DC addressing using comparator. If they are equal, a drive signal is given to the decoder.

Chip Selection:

The chip select signal from decoder selects any one of the 8255A as per software and connects to the bi-directional buffer control signal from the PC bus is given to the unidirectional buffer.

2.2. 8255A – PROGRAMMABLE PERIPHERAL INTERFACE

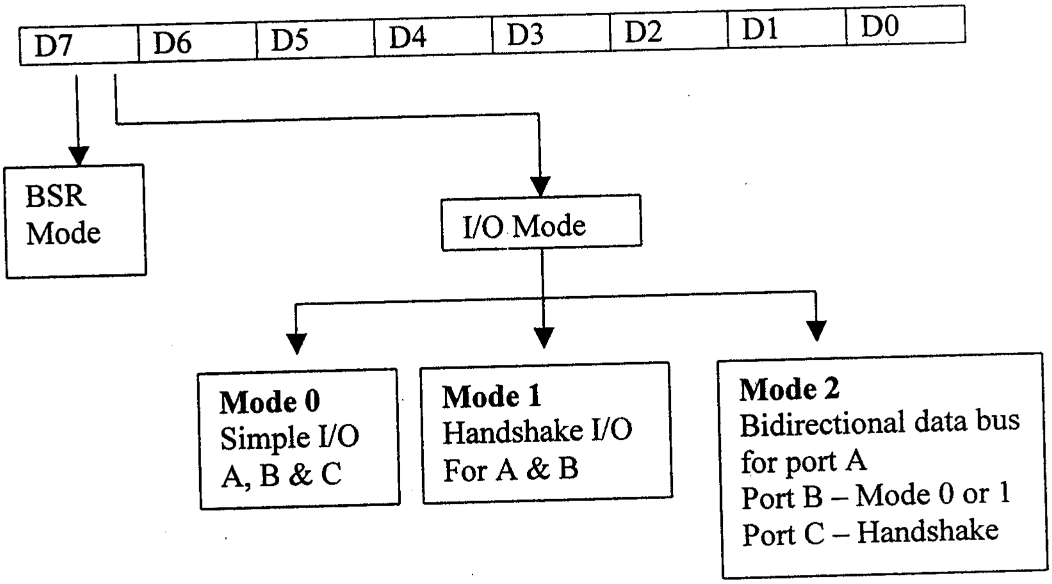
The 8255A is a widely used, programmable, parallel I/O device. It can be programmed to transfer data under various conditions, from simple I/O to interrupt I/O. It is flexible, versatile and economical, but somewhat complex. It is an important general-purpose I/O device that can be used with almost any microprocessor.

The 8255A has 24 I/O pins that can be grouped primarily into two 8-bit parallel ports A & B, with the remaining eight bits as port C. The eight bits of port C can be used as individual bits or can be grouped in two 4-bit ports, C_{UPPER} and C_{LOWER}. The functions of these ports are defined by writing a control word in the control register.

2.2.1. 8255A Modes

Fig. 2.2 shows the functions of 8255A, classified according to two modes: The Bit Set/Reset (BSR) mode and the I/O mode. The BSR mode is used to set/reset the bits in port C. The I/O mode is divided into modes 0,1 and 2, which are explained in detail in the following chapters.

Fig. 2.2: 8255A Modes



2.2.2. Mode 0: Simple Input or Output

In this mode, Ports A and B are used as two simple 8-bit I/O ports and Port C as two 4-bit ports. Each port can be programmed to function simply as an input or output port. The Input/Output features in Mode 0 are as follows:

1. Outputs are latched.
2. Inputs are not latched.
3. Ports do not have handshake or interrupt capability.

2.2.3. Mode 1: Input or output with Handshake

In Mode 1,handshake signals are exchanged between the MPU and peripherals prior to data transfer. The features of this mode are as follows:

1. Two ports (A and B) function as 8-bit I/O ports. They can be configured either as input or output ports.
2. Each port uses three lines from port C as handshake signals. The remaining or two lines of port C can be used for simple I/O functions.
3. Input and output data are latched.
4. Interrupt logic is supported.

In 8255A,the specific lines from port C used for handshake signals vary according to the I/O function of a port.

2.2.4. Mode2: Bidirectional Data Transfer

This mode is used primarily in applications such as data transfer between two computers or floppy disk controller interface. In this mode, Port A can be configured as the bi-directional port and port B either in Mode 0 or Mode 1.Port A uses five signals from Port C as handshake signals for data transfer. The remaining three from Port C can be used either as simple I/O or as handshake for port B.

2.2.5. Block Diagram of 8255A

The block diagram in **Fig. 2.3** shows two 8-bit ports (A & B), two 4-bit ports (C_U & C_L), the data bus buffer and control logic. The ports A & C_{UPPER} come under Group A, while ports B & C_{LOWER} come under Group B. The address select lines A0 & A1 are used to select the control register and the ports for reading and writing.

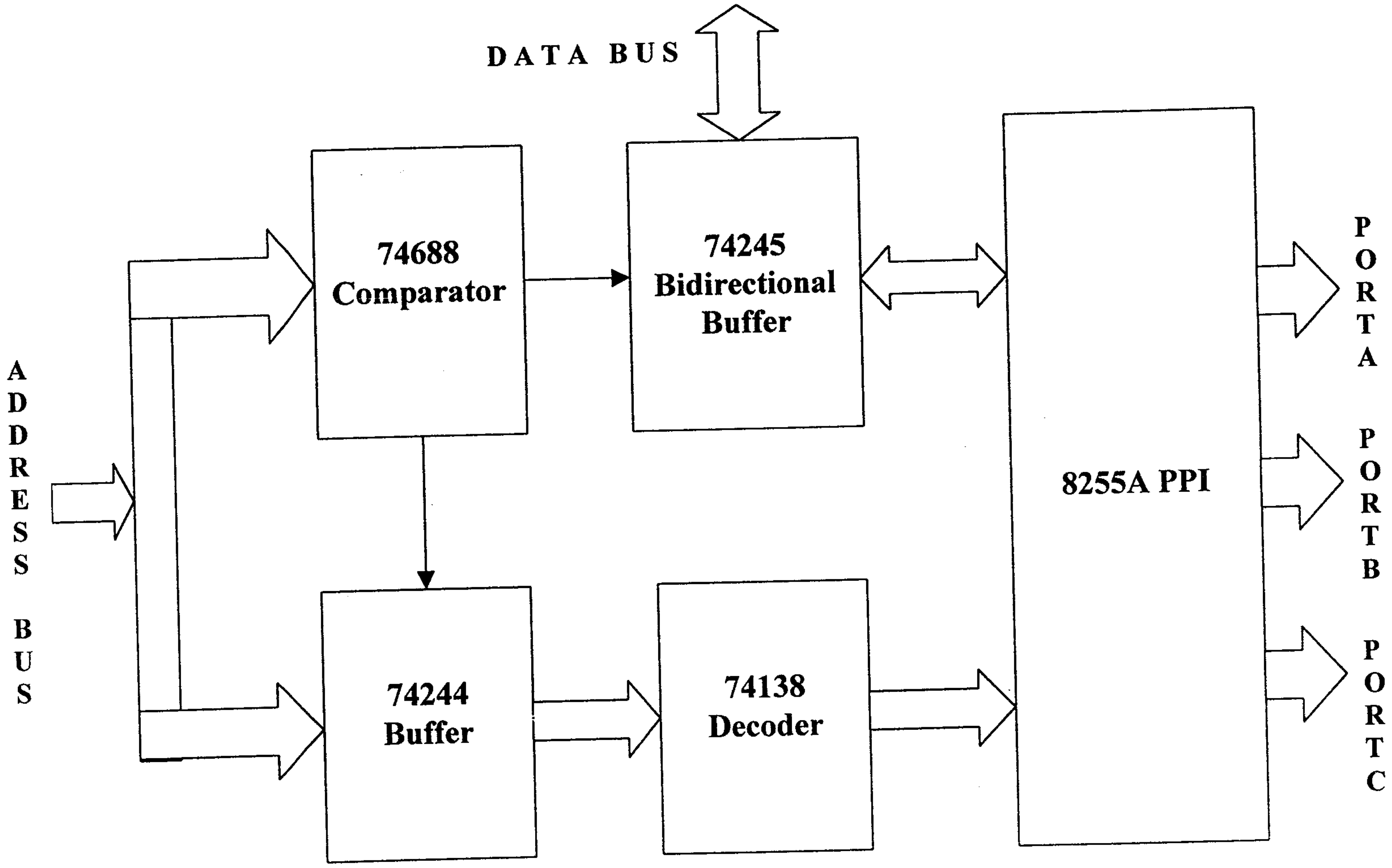


Fig. 2.1: ISA Add-On Card Block Diagram

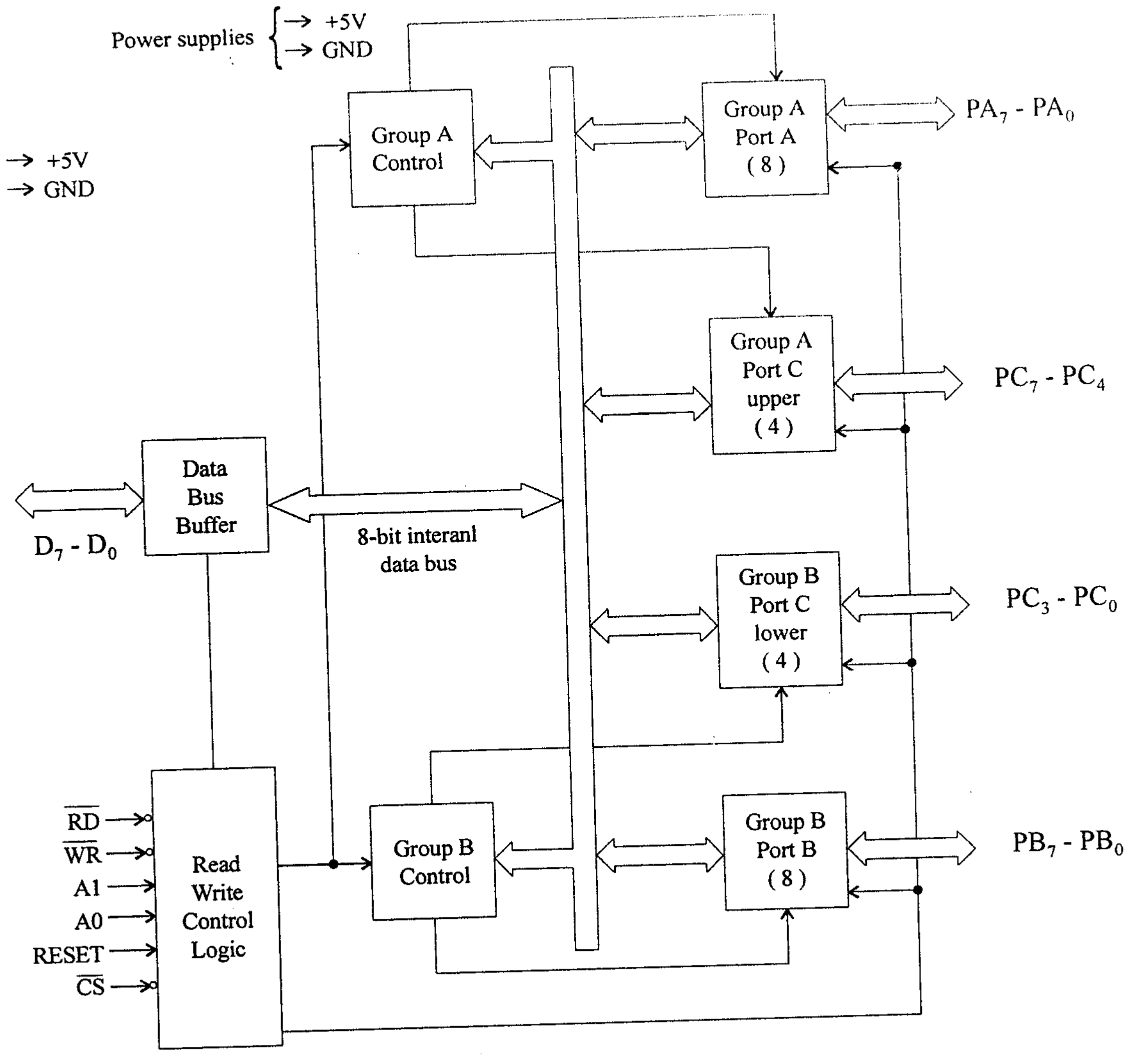


Fig. 2.3: Block Diagram of 8255A

RF TRANSMISSION BLOCK

3. TRANSMISSION BLOCK

The RF Transmission block consists of the following circuitry:

- Transistor Driver
- DTMF Code generator
- FM Transmitter
- 5V Power Supply

The blocks are shown in **Fig. 3.1**. Each of these blocks is described in detail below.

3.1. TRANSISTOR DRIVER

The driver circuit is shown in **Fig.3.2**. The TTL compatible signals from the ISA add-on card should not be directly connected to another circuit. This is because of the fact that the devices in that circuit may load the card and cause damage to the ISA bus system. Hence a transistor driver circuit is used to interface with the external circuitry. The transistor BC547 is used as a switch in this driver circuit.

3.2. DTMF CODE GENERATOR

The number sent from the ISA card should be converted to a format that can be mixed with a carrier and transmitted using frequency modulation. A simple coding method is implemented using a dual tone multi frequency (DTMF) code generator. For this purpose, a standard DTMF code generator UM91215B is used. It generates DTMF codes corresponding to the inputs given to pins 12 through 15.

3.3. FM TRANSMITTER

The DTMF code generator circuit and FM transmitter circuit are shown in **Fig. 3.3**. The signals are passed on from the DTMF code generator to the base of the transistor Q1 (BF494) through the capacitor C1 (0.1). Q1 is a NPN transistor and its collector is given positive supply through the coil L1. Its emitter is connected to the ground through the resistance R3 (100E) while its base is given the forward supply through the resistance R2 (68k). These signals are given to the base of the transistor Q1 (BF494) through the capacitor C1 (0.1). The coil L1 connected at its collector can be made by giving 5 turns of 24 SWG wire on a base of 0.5cm diameter. A capacitor C3 (12pf) has also been connected between the collector and the emitter of the transistor Q1. This capacitor triggers the oscillations. As soon as the audio signal is received at the base then the transistor starts to oscillate and it generates FM frequency, which is given to the antenna through the capacitor C4 and transmitted. The range of this transmitter lies between 100 meters to 500 meters.

3.4. +5V POWER SUPPLY

Since all electronic circuits work only with low D.C. voltage we need a power supply unit to provide the appropriate voltage supply. This unit consists of transformer, rectifier, filter and regulator. Typically 230V rms AC is connected to a transformer, which steps that AC voltage down to the level of the desired AC voltage. A diode rectifier then provides a full-wave rectified voltage that is initially filtered by a simple capacitor filter to produce a DC voltage. This resulting DC voltage usually has some ripple or AC voltage variations. A regulator circuit can use this DC input to provide DC voltage that not only has much lesser ripple voltage but also remains the same DC value even when the DC voltage varies slightly, or the load connected to the output DC voltages changes.

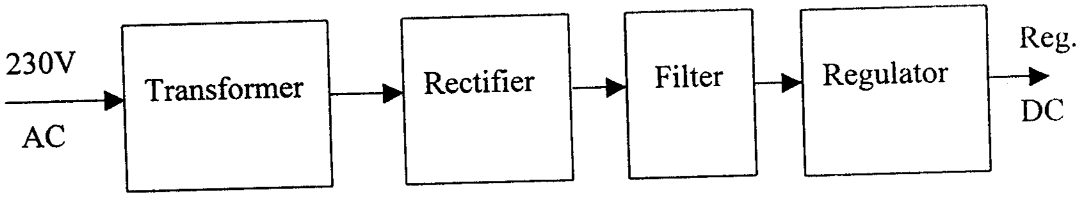


Fig. 3.4: Regulated Supply Block Diagram

3.4.1. Transformer:

A transformer is a device that is transforms the input power into electric power of the same frequency in another circuit. It can raise or lower the voltage in a circuit but with a corresponding decrease or increase in current. It works according to principle of mutual induction. A step down transformer is used for providing the necessary supply for the electronic circuits.

3.4.2. Rectifier:

A rectifier is a device that is used to convert alternating current to direct current. There are two types of rectifiers.

1. Half wave rectifier
2. Full wave rectifier

The full wave rectifier is far more efficient than the half wave rectifier due to the fact that it converts both the positive and negative half cycles of the AC signal are converted to DC. One example of a full wave rectifier is the bridge rectifier that uses four diodes. From the basic bridge configuration we see that two diodes (say D2 & D3) are conducting while the other two diodes (D1 & D4) are in "off" state during the period t =0 to T/2. Similarly, for the negative half cycle of the input the conducting diodes are D1 & D4. Thus the polarity across the load is the same.

3.4.3. Filter:

The filter circuit used here is a simple capacitor filter where a capacitor is connected at the rectifier output and a DC is obtained across it. The filtered waveform is essentially a DC voltage with negligible ripples, which is ultimately fed to the load.

3.4.4. Regulator:

The output voltage from the capacitor is further smoothened and regulated using a regulator. The voltage regulator is a device, which maintains the output voltage constant irrespective of the change in supply variations, load variation and temperature changes. Three fixed voltage regulators namely LM 7812, LM 7805 and LM7912 are used. The IC 7812 is a +12V regulator IC 7912 is a -12V regulator and IC 7805 is a +5V regulator.

3.4.5. Three-Terminal Voltage Regulators:

Fig.3.5 shows the basic connection of a three-terminal voltage regulator IC to a load. The fixed voltage regulator has an unregulated dc input voltage, Vi, applied to one input terminal, a regulated output dc voltage, Vo, from a second terminal, with the third terminal connected to ground. For a selected regulator, IC device specifications list a voltage range over which the input voltage can vary to maintain a regulated output voltage over a range of load current. The specifications also list the amount of output voltage change resulting from a change in load current (load regulation) or in input voltage (line regulation).

3.4.6. Fixed Positive Voltage Regulators:

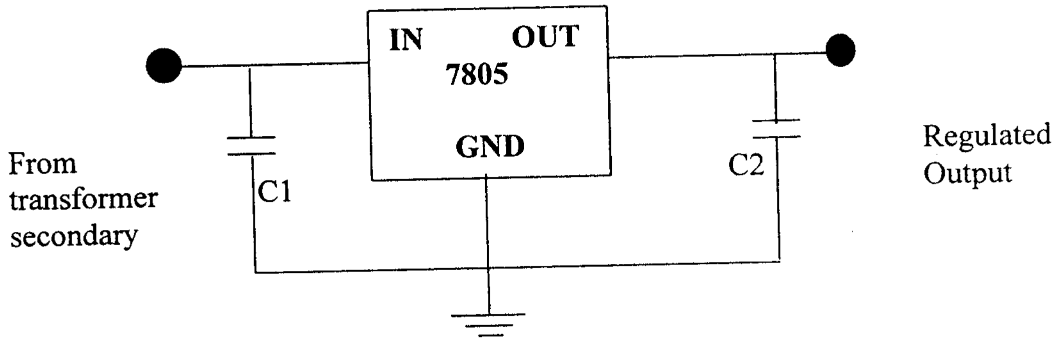


Fig. 3.5: 7805 Fixed Voltage Regulator Connection

The 78XX series regulators provide fixed regulated voltages from 5 to 24 V. **Fig.3.5** shows how one such IC, a 7805, is connected to provide voltage regulation with output from this unit of +5V dc. An unregulated input voltage Vi is filtered by capacitor C1 and connected to the IC's IN terminal. The IC's OUT terminal provides a regulated + 12V which is filtered by capacitor C2 (mostly for any high-frequency noise). The third IC terminal is connected to ground (GND). While the input voltage may vary over some permissible voltage range, and the load may vary over some acceptable range, the output voltage remains constant within specified voltage variation limits. These limitations are spelled out in the manufacturer's specification sheets. The complete circuit diagram is shown in Fig. 3.6. A table of positive voltage regulated ICs is given in **Table 3.1**

Table 3.1: Positive Voltage Regulators in 78XX series

IC Part	OutputVoltage (V)	Minimum Vi (V)
7805	+5	7.3
7806	+6	8.3
7808	+8	10.5
7810	+10	12.5
7812	+12	14.6
7815	+15	17.7
7818	+18	21.0
7824	+24	27.1

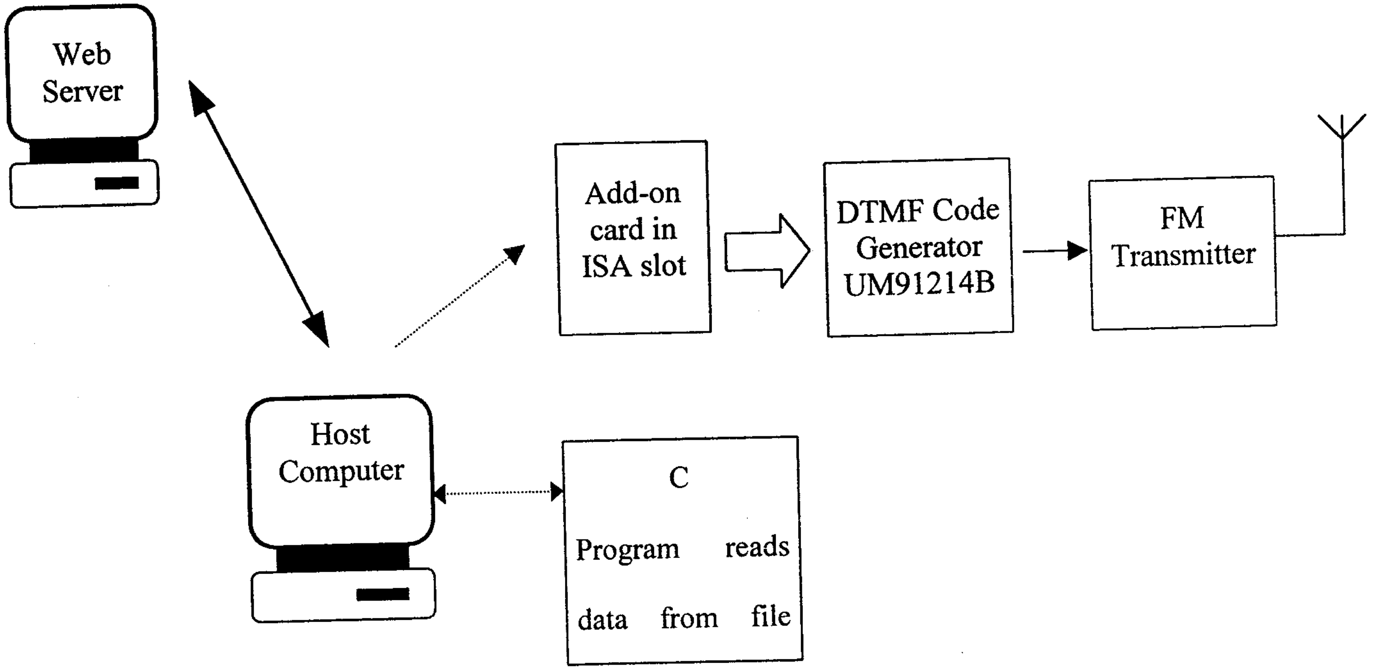


Fig. 3.1: RF Transmission Side Block Diagram

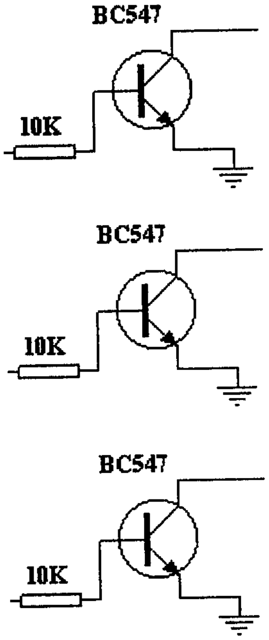


Fig. 3.2: BC547 Transistor Driver

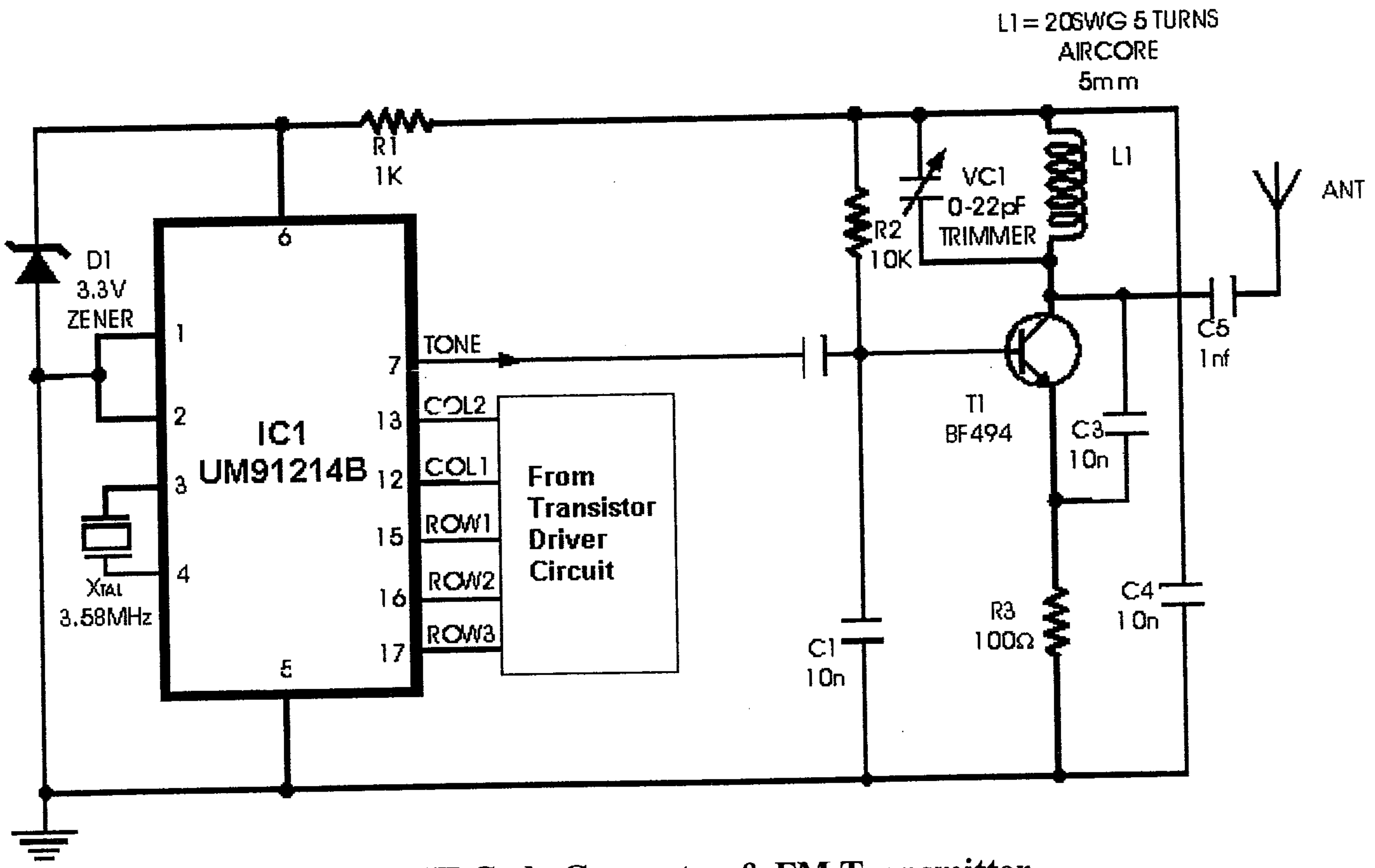


Fig. 3.3: DTMF Code Generator & FM Transmitter

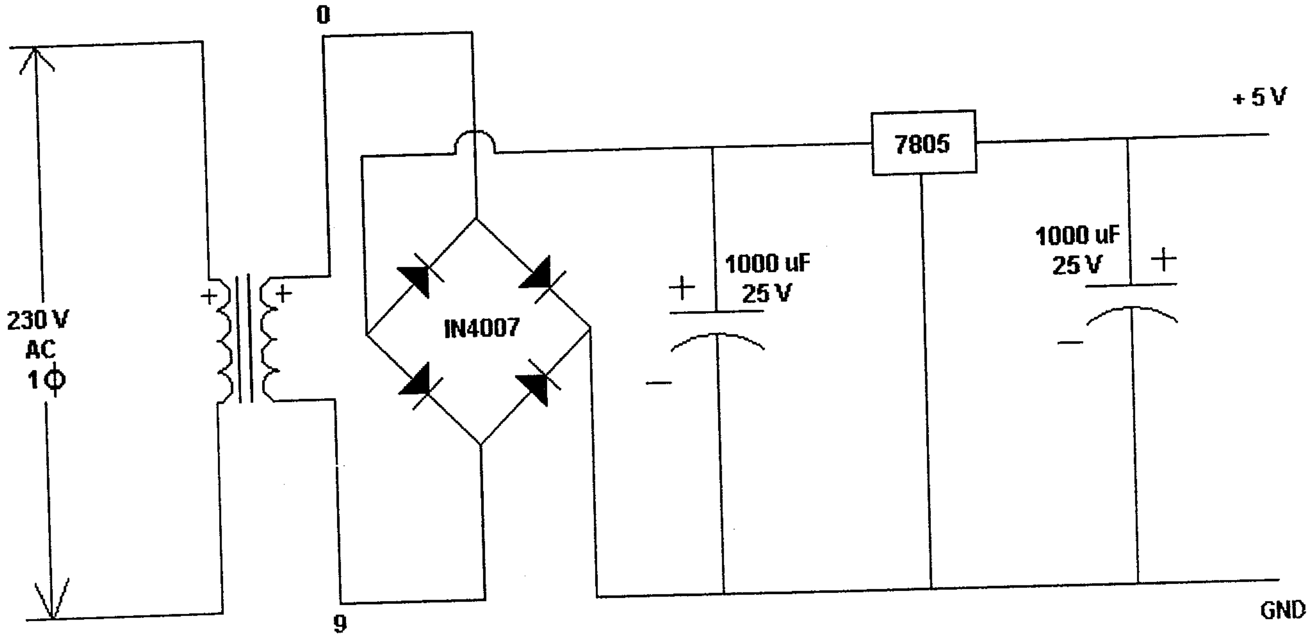


Fig. 3.6: +5V Regulated Power Supply Circuit Diagram

4. RF RECEPTION BLOCK

The RF reception block consists of the following blocks, which are explained below:

- FM Receiver
- DTMF Decoder
- Microcontroller
- LED Display Driver
- Transistor Driver
- Relay
- +/- 12V Power Supply

The block diagram of this block is shown in **Fig. 4.1**

4.1. FM RECEIVER

The circuit connections of the FM receiver are shown in **Fig. 4.2**. IC CXA1019S is a 30-pin DIL IC manufactured by Sony. Facility to connect a tuning indicator has also been provided in this IC apart from the various sections built within the IC. A 10.7MHz ceramic filter has been used in this circuit in place of the IFTs. This is a single chip AM/FM Radio IC. An audio output section is also built within the IC apart from all the necessary sections for the AM/FM radio. The audio output section of this IC has not been used in the given circuit. The signal received at the IC pin-24 is given to the pin-1 of the volume control and the pin 2 of the volume control is connected to the audio output section.

The IC pin no.27 is the positive supply pin, which is given +6V supply through a resistance R4 (56E). This supply is filtered by the capacitor C1 (1000MF/16V). IC pin no.7 & 8 are connected to the oscillator section and a coil L1 is connected at the pin no.8. A capacitor C3 (39PF) and a button trimmer B1 (15pf) are connected parallel to this coil L1. This trimmer selects the desired frequency. RF coil L2 is connected at the IC pin no.10 and a capacitor C7 (33pf) and a button trimmer B2 (15pf) are connected in parallel to this coil.

IC pin no.9 is the RF ground pin. RF coils, the oscillator and the related components are all connected to this pin. Antenna coil L3 is connected at the IC pin no.13 through a capacitor C9 and a capacitor C10 (27pf) is connected in parallel to this coil. A telescopic antenna is connected from the center point of the capacitor C9 and coil L3. The capacitor C8 has been connected to it for filtration. The IC pin no.1,2,3,4,5,12,14,19,21 and 30 are the dc ground pins. IC pin no.9 is the RF ground pin.

4.2. DTMF RECEIVER

The circuit used for receiving DTMF signals is shown in **Fig. 4.3**. This is a full DTMF receiver that integrates both band split filter and decoder functions into a single 18-pin DIP or SOIC package. 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 provision of a 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 timing capacitor.

The new M-8870-02 provides a "power down" option which, when enabled, drops consumption to less than .5 mw. The 02 versions can also inhibit the decoding of the fourth column digits.

DTMF is the generic name for pushbutton telephone signaling equivalent to the bell system's touch-tone. 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.

4.2.1. DTMF Standards:

The DTMF tone-signaling 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-signaling 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.

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. Two of eight tones can be combined so as to generate sixteen different DTMF digits.

4.2.2. DTMF Decoder:

The DTMF signals transmitted 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 would have a nominal width of 50ms followed by a pause of similar duration between consecutive digits would be transmitted in one second.

4.2.3. 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.

4.2.4. Applications

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

4.2.5. Functional Description

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. A single order switched capacitor section that smoothers the signals prior to limiting follows each filter output. High gain comparator provided with hysteresis to prevent detection of unwanted low-level signals and noise performs signal limiting. 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, 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 V_{CC} to rise as capacitor discharges. Provided that the signal condition is maintained for the validation period V_{CC} 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 VCC 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.

Guard Time Adjustment

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

$$T_{REC}=t_{dp}+T_{GTP}$$

$$T_{GTP}=0.67rc$$

The value of t_{dp} is a parameter of the device and T_{REC} 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 T_{REC} 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 (T_{GTP}) and tone absent (T_{GTA}). This may be necessary to meet system specifications that place both accept and reject limits on both tone and inter digit pause.

Guard time adjustments also allow 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 T_{REC} 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 T_{REC} 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 V_{REF} 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. a single crystal can be used to connect a series of M-8870 by coupling the oscillator output of each M-8870 through a 30pF capacitor to the oscillator input of the next M-8870.

Pin Functions

IN+ : Non-inverting input connected to the front-end of the differential amplifier.

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

GS : Gain select. It gives access to output of front-end amplifier for connection of feedback resistor.

VREF : Reference voltage output. 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 : 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 : Three state outputs. When enabled by OE,

Q3,Q4 : Provides the code corresponding to the last valid tone pair received.

StD : Delayed steering output. Presents 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_{tsi}.

Est : Early steering output presents a logic high immediately when the digital algorithm detects a recognizable tone pair. Any momentary loss of signal condition will cause E_{ST} to return to logic low.

St/GT : Steering input/guard time output voltage greater than V_{TSt} detected at St cause 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 reset the external steering time constant, and its state is a function of Est and the voltage on St.

V_{DD} : Positive power supply.

Table 4.1: Tone Decoding

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

H = High

L = Low

Z = High Impedance

4.3. AT89C51 MICROCONTROLLER

4.2.1. 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

4.2.2. 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.

The AT89C51 provides the following standard features: 4K bytes of Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator and clock circuitry. In addition, the AT89C51 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 hardware reset.

4.2.3. 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 programmed data memory. In this mode P0 has internal pull-ups. Port 0 also receives the code bytes during Flash programming, and outputs the code bytes during program verification. External pull-ups 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 pull-ups 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 pull-ups. 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 a sink/source four TTL inputs. When 1s are written to Port 2 pins they are pulled high by the internal pull-ups 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 pull-ups. 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 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 pull-ups. 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 pull-ups 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 pull-ups. Port 3 also serves the functions of various special features of the AT89C51 as listed below:

Table 4.2: Port 3 Alternate Functions

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)

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/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 (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.

PSEN:

Program Store Enable is the read strobe to external program memory. When the T89C51 is executing code from external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory.

EA/VPP:

External Access Enable. 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.

If lock bit 1 is programmed, EA will be internally latched on reset. 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.

4.3. LED DISPLAY DRIVER

The output of the microcontroller can be monitored using the following arrangement. The device number that is currently accessed is displayed using four 7-segment LED displays. These LED displays are driven by four 7447 BCD to 7-Segment Decoder/Driver ICs. The salient features of 7447 are:

- Consists of NAND gates, input buffers and seven AND-OR-IN-VERT gates.
- Offers active LOW, high sink current outputs for driving indicators directly.
- Provides lamp test, blanking input/ripple-blanking output and ripple-blanking input.
- Accepts 4-bit binary-coded-decimal (BCD)
- Depending on the state of the auxiliary inputs, decodes this data to drive a 7-segment display indicator. Output configurations are designed to withstand the relatively high voltages required for 7-segment indicators.

4.4. TRANSISTOR DRIVER

A transistor driver circuit is used to interface the microcontroller unit with the relay circuit. This switching circuit is similar to the one used in the transmitter side. It is shown in **Fig.3.2**

4.5. RELAY CIRCUIT

A relay is an electromechanical switching device. It is used to switch devices that operate at higher voltages when compared to the control voltage. Relays are of two types:

1. Normally Open (NO)

2. Normally Closed (NC)

The NO type relays are normally open when no voltage is applied to the electro magnets. The NC type relays are normally closed when no voltage is applied to the electro magnets. When a voltage is applied to the relay coil, the electro magnet is energized, which makes or breaks a contact. When no voltage is applied, the electro magnet is de-energized and releases the contact. Thus a device working any voltage can be switched using a relay.

4.6. ±12V AND +5V POWER SUPPLIES

As discussed earlier in Chap. 3.4, a regulated power supply is essential for the operation of any electronic circuit. The ±12V power supply required for the reception block is built using two regulators 7812 (+12V) and 7912 (-12V). The +5V power supply is similar to the one used in the transmission block. The circuit diagram of the +5V supply is shown in **Fig. 3.6**. The circuit connections of the +/- 12V power supply are shown in **Fig. 4.4.**

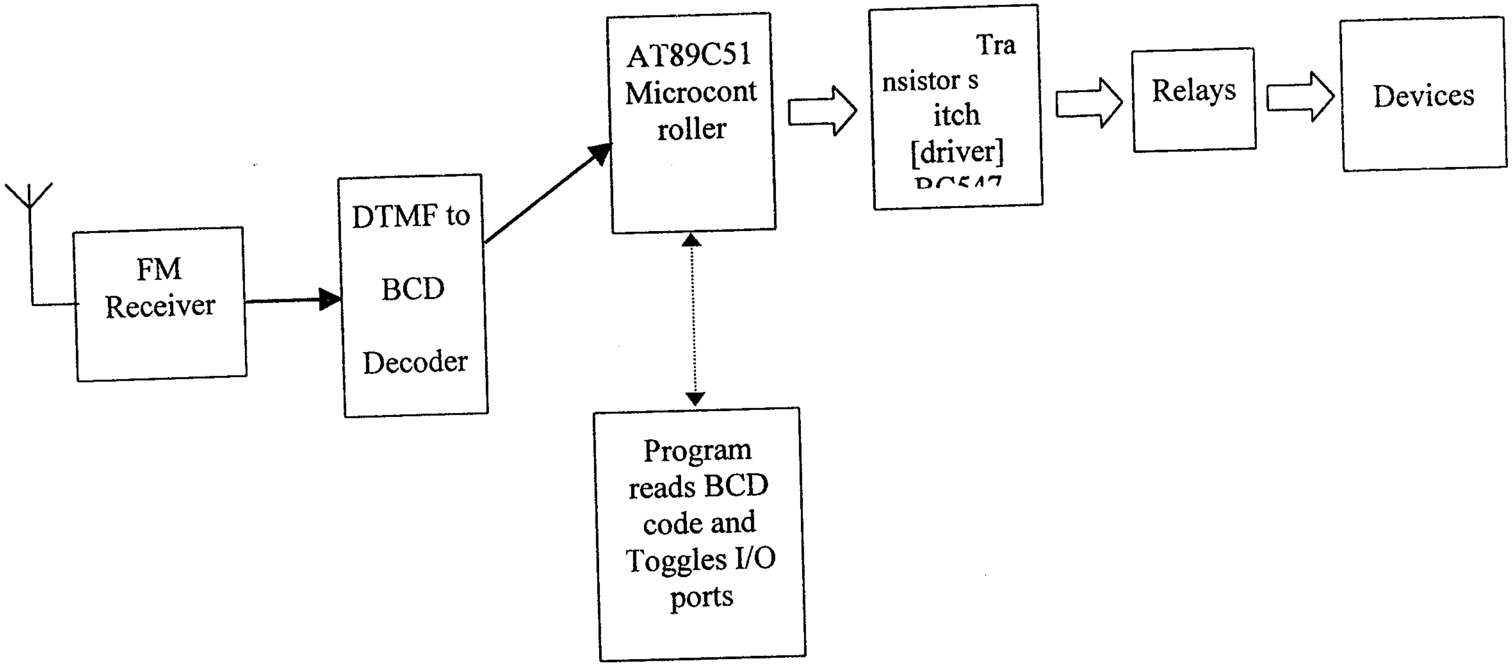
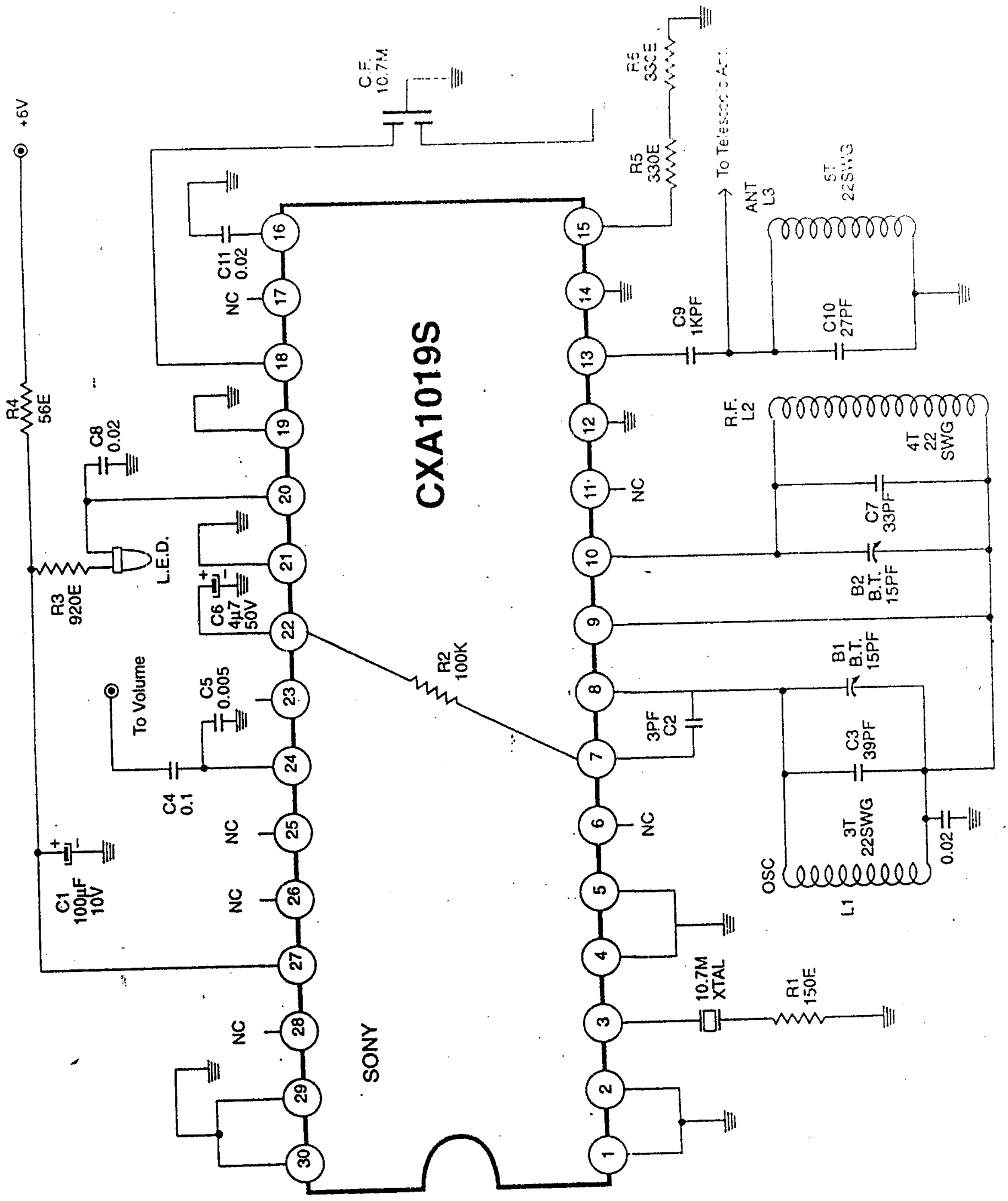


Fig. 4.1: RF Reception Side Block Diagram



DATA
L1 Oscillator Coil = 3 Turns 22 SWG Air Core Coil of Diameter 0.5cm
L2 R.F. Coil = 4 Turns 22 SWG Air Core Coil of Diameter 0.5cm
L3 Antenna Coil = 5 Turns 22 SWG Air Core Coil of Diameter 0.5cm

Fig. 4.2: FM Receiver Circuit Diagram

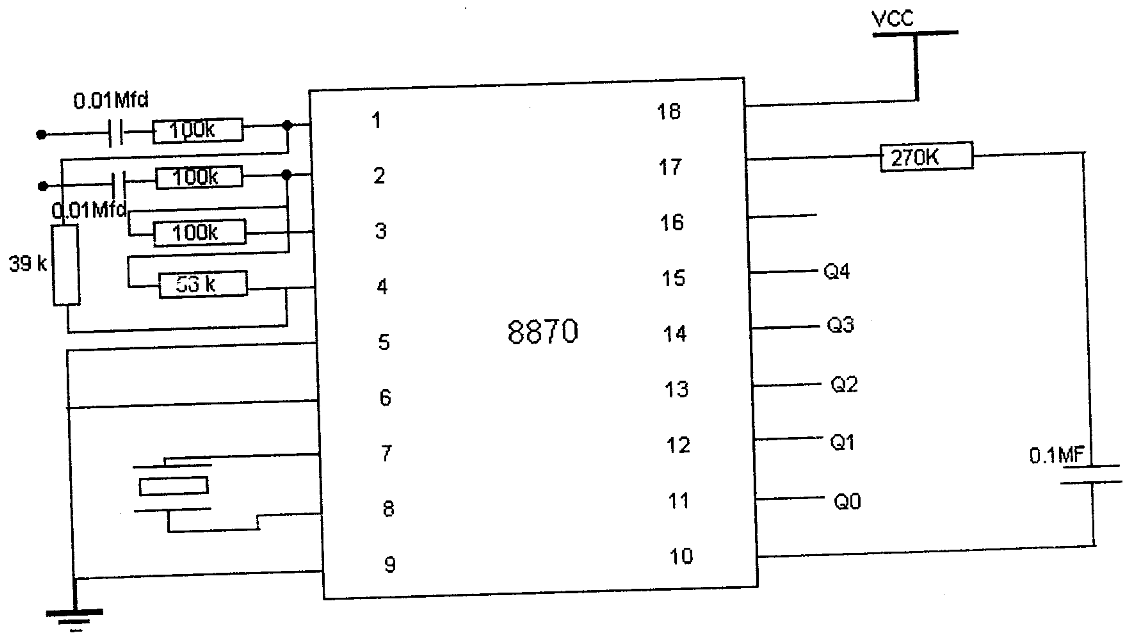


Fig.. 4.3.: DTMF Receiver Circuit Diagram

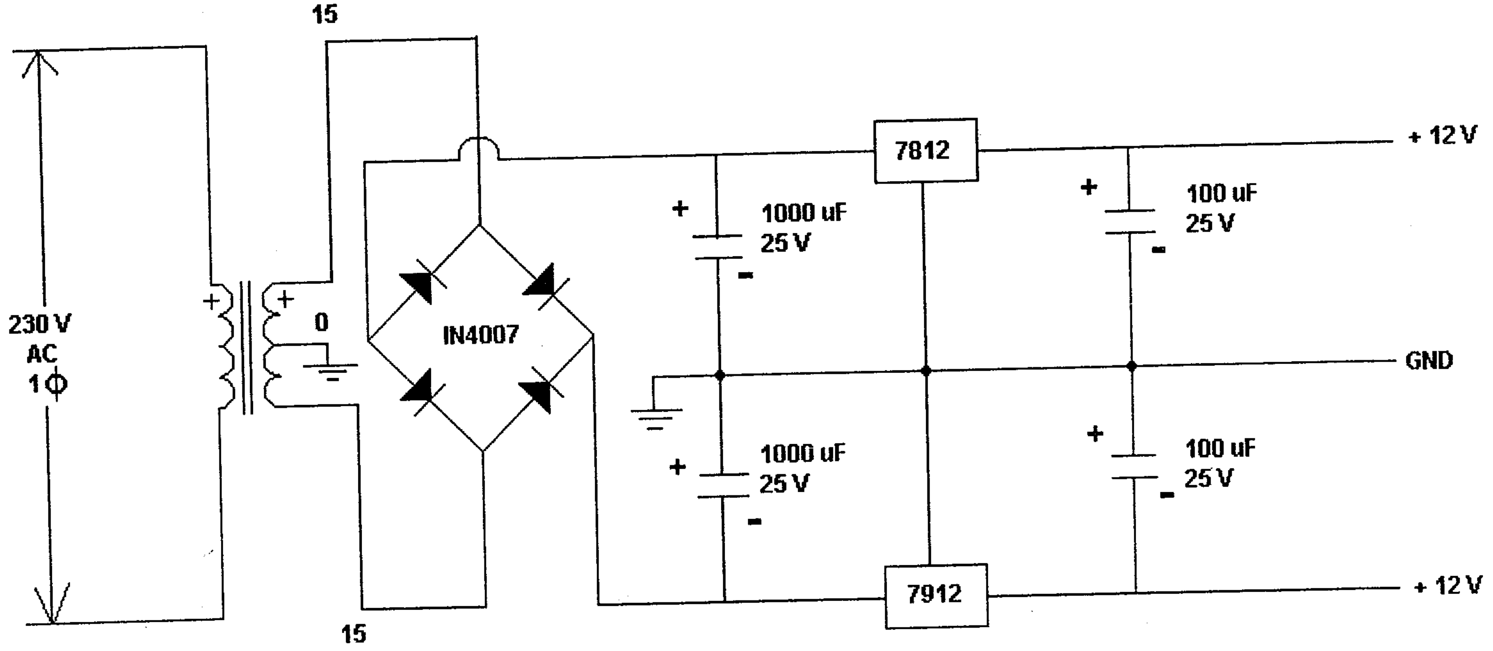


Fig. 4.4: ±12V Regulated Power Supply Circuit Diagram

5. INTERNET INTERFACE BLOCK

5.1. ACTIVE SERVER PAGES

5.1.1. HTML Pages

An HTML file is one that is used by a web browser to generate a web page. HTML files differ from ordinary text files in that they include special codes called HTML tags. All HTML expressions are surrounded by angle brackets. Examples of HTML tags are <BR>, <BODY>,<FONT>. Most of the HTML tags need to be ended by a matching tag. For instance, a <FONT> tag should be ended by a </FONT> tag. Using these tags, the colour, text font and many more properties of a web page can be changed.

As HTML has evolved, other features have been added, such as forms, frames, tables, and so on. However, even with all the new features, HTML basically deals with Web content by:

- Formatting and displaying the content of a page.
- Waiting for the user to click something in the page.
- Depending on what the user clicks on, fetching something else (a new page, for example) and repeating the process.

Although this provides a wealth of possibilities for content manipulation, it doesn't allow for more advanced things like accessing a database, ordering catalog items online, or making animated graphics within a web page. JavaScript and VBScript are some of the scripting languages used to achieve these capabilities along with Active Server Pages (ASP) or Common Gateway Interface (CGI).

5.1.2. Active Server Pages - Overview

An Active Server Page is a standard HTML file that has been extended with additional features. The most important application of an ASP file is that it can be used for creating dynamic HTML content.

The three important features of ASP are:

- An ASP can contain server-side scripts. By including server-side scripts, web pages with dynamic content can be created
- An ASP provides a number of built-in objects. These objects allow retrieval of information from and transmission of information to the browser.
- ASP provides easy connectivity to back-end databases like MS SQL Server and MS Access.

5.1.3. Advantages of ASP:

- Compile free
- Faster than Common Gateway Interface (CGI)
- Supports existing investments
- Supports all scripting languages (ActiveX scripting)
- Database Connectivity

5.1.4. ASP – Steps of operation:

1. A user enters the URL of an ASP file into address bar of web browser
2. The browser sends a request for the ASP to web server
3. The web server receives the request and recognizes that the request is for an ASP file

4. The web server retrieves the proper page from disk or memory
5. The web server sends the page to a special program called asp.dll
6. The ASP is processed from top to bottom and any encountered commands are executed. The result of this is a standard HTML file
7. The html file is sent back to the browser
8. The client's web browser interprets the html file and the results are displayed in the browser window

5.2. DATABASE DESIGN

The database used as back-end at the server-side is MS Access. The database contain the device number, device name and device status. This is used to show the information to the user. After the user makes changes to the device states, the new data is updated in the database. The database design is shown in Table 4.1

Table 5.1: Database Structure

S.No.	Field Name	Field Type	Field Size
1.	DevNum	Number	Auto
2.	DevName	Text	20
3.	DevStatus	Yes/No	Auto

5.3. VBSCRIPT

Microsoft Visual Basic Scripting Edition, the newet member of the Visual Basic family of programming languages, brings active scripting to a wide variety of environments, including web client scripting in Microsoft Internet Explorer and web server scripting in Microsoft Internet Information Server

5.3.1. ActiveX Scripting

VBScript talks to host applications using ActiveX scripting. With ActiveX scripting, browsers and other host applications don’t require special ingration code for each scripting component. ActiveX enables a host to compile scripts, obtain and call entry points, and manage the namespace available to the developer. With ActiveX scripting, language vendors can create standard language run times for scripting. Microsoft will provide run time support for VBScript. ActiveX scripting is used in Microsoft Internet Explorer and Microsoft Internet Information Server.

5.3.2. A Simple VBScript page

```
<HTML>
<HEAD><TITLE>A Simple first page</TITLE>
<SCRIPT LANGUAGE=”VBScript”>
<! - -
Sub Button1_OnClick
        MsgBox “Welcome to KCT”
End Sub
- - >
</SCRIPT>
</HEAD>
<BODY>
<H3>A simple first page</H3><HR>
<FORM><INPUT NAME=”Button1” TYPE=”BUTTON” VALUE=”Click
Here”></FORM>
</BODY>
</HTML>
```

The result of this program is a dialog box that displays the phrase “Welcome to KCT”. However the operation is much more complex than what it seems to be. When Internet Explorer reads the page, it finds the <SCRIPT> tags, recognizes that there is a piece of VBScript code, and saves the code. When the button is clicked, IE makes the connection between the button and the code, and runs the event procedure named ‘Button1_OnClick’.

5.4. JAVASCRIPT

JavaScript is a scripting language developed by Netscape that allows creation of dynamic pages. JavaScript is not Java. JavaScript is a simple scripting language that is very similar to C. It is a scripting language that is parsed and then executed by the parser. The script can be anywhere inside a HTML code as long as it is enclosed within the <SCRIPT LANGUAGE="JavaScript"></SCRIPT> tags.

The salient features of JavaScript are:

- Variables are Case sensitive.
- There are no type-casted variables. Something that is a string can also be an integer.
- Knowledge of the object model is necessary. The hierarchy of the object model must be understood in order to manipulate objects in any HTML page.

JavaScript is a scripting language specifically designed to work with the World Wide Web. With JavaScript, the power of HTML and the World Wide Web can be extended in a variety of ways. A script is nothing more than a sequence of program instructions (called statements). The program steps through the statements one at a time and performs whatever the script tells it. This is exactly the same as 'programming', except that scripts tend to have simpler rules and require less learning time. Some examples of programs that provide scripting are dBASE, Paradox, and Microsoft Access (though there are many more). Some examples of stand-alone scripting languages are Perl and REXX.

Based on Java, JavaScript supports most of Java's expression constructs (another word for statements). However, JavaScript doesn't require a compiler or knowledge of programming to get it up and running. All that is needed is an understanding of HTML and a little logic.

Like Java, JavaScript is built on the concept of objects. Unlike Java, however, JavaScript provides a basic collection of objects to work with. While new objects and object types can be easily created, the ones provided give great deal of power.

- Scripts are evaluated after a page has finished loading, but before the page is displayed.
- Any functions defined in scripts are not automatically executed when the page loads. They are stored until called by something else in the page.
- Scripts loaded through the SRC= attribute (in other words, scripts that are kept in separate files) are evaluated before inline (or "in-page") scripts.

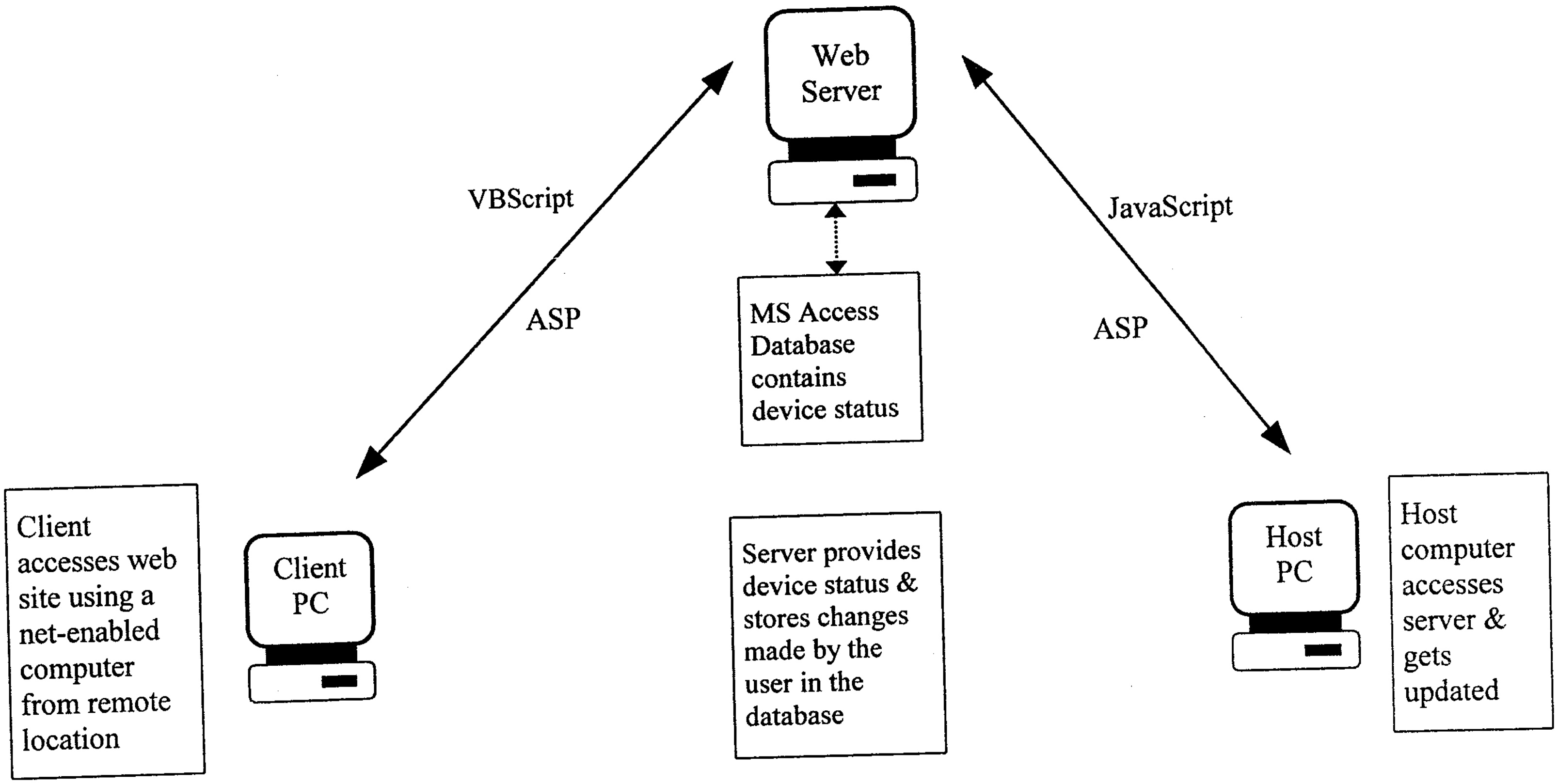

Web Server
VBScript
ASP
JavaScript
ASP
MS Access Database contains device status
Client accesses web site using a net-enabled computer from remote location
Client PC
Server provides device status & stores changes made by the user in the database
Host PC
Host computer accesses server & gets updated

Fig. 5.1: Internet Interface Block Diagram

6. SOFTWARE DESCRIPTION

6.1. Microcontroller Program – 8051 ALP

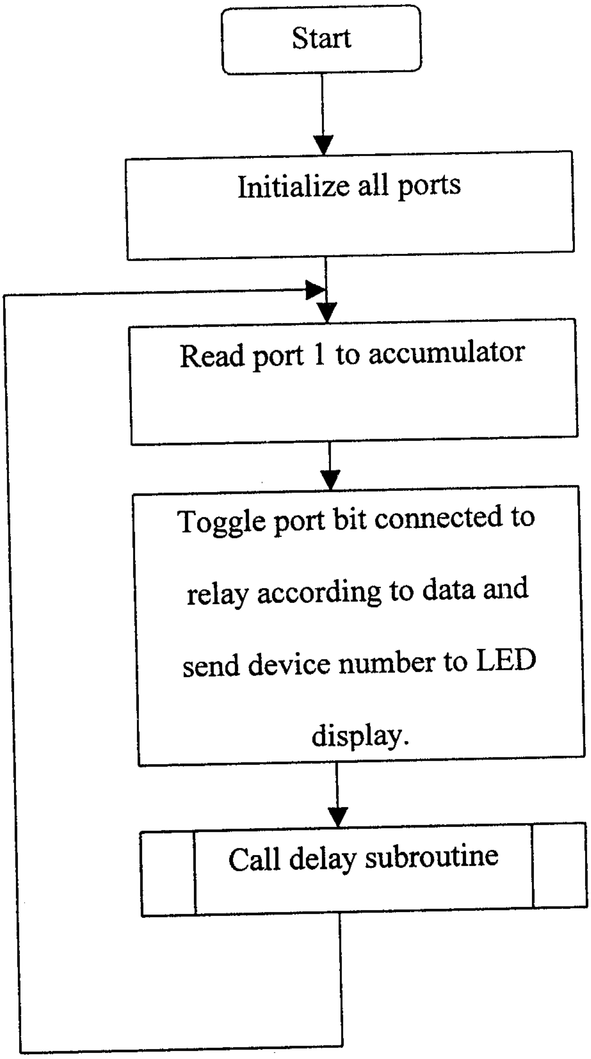

Start
Initialize all ports
Read port 1 to accumulator
Toggle port bit connected to relay according to data and send device number to LED display.
Call delay subroutine

Fig. 6.1: Microcontroller program flow chart

The microcontroller actually performs the operation of getting the code from the DTMF receiver and toggles the port corresponding to that code. The relay connected to the port is switched using the transistor driver. The program flow chart is shown in **Fig. 6.1**

6.2. C Program

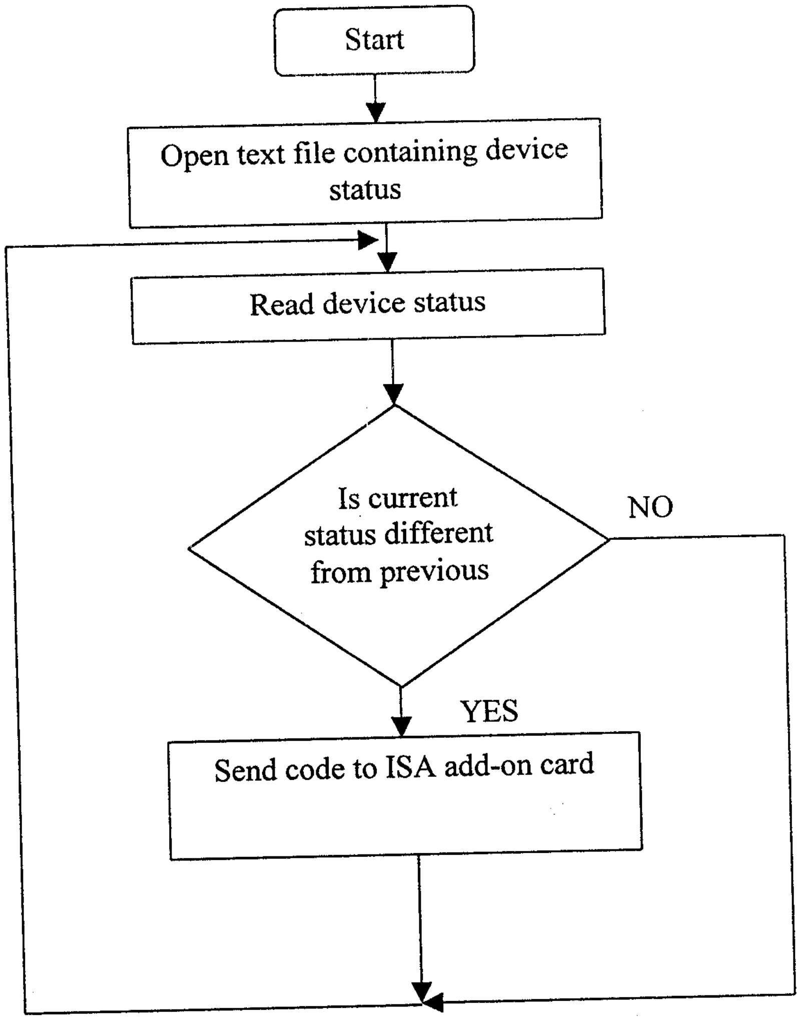

Start
Open text file containing device status
Read device status
Is current status different from previous
NO
YES
Send code to ISA add-on card

Fig. 6.2: C program flow chart

The C program gets the device status from the text file, which is obtained using JavaScript. The status of each device is compared with its previous state. If it has changed, the device code is sent to the ISA card.

6.3. INTERNET PROGRAM DESCRIPTION

The user is presented with the current device status when he visits the URL of the site. Before he can change the status of any device, the user has to enter a user name and a password. After authentication of this data, the user is given access to the above said page. The status is shown by reading the database residing in the web server. This is done using VBScript and ASP.

After the changing the device status and clicking the Submit button, the data is sent to the database in the web server using VBScript. This is done by deleting the existing information and then writing the new data entered in the web page by the user.

The host computer contacts the web server at constant time intervals and checks for the updated information. This information is written in a text file using JavaScript. The text file contains the device code, device name and device status. The ON state is indicated by a '1' and the OFF state is indicated by a '0'.

Please refer Appendix A for the coding.

7. FUTURE DEVELOPMENTS

The field of e-control is a fast developing field, which is now in the incipient stage. Hence our project has many avenues for further development and enhancements. We would like to suggest a few features that offer space for enhancements.

The wireless link used in our project is implemented using frequency modulation. Frequency modulation imposes restrictions on the maximum range offered by the system. Hence AM can be used to increase the range of the wireless link.

The essence of the project is to decrease the need for physical presence of an individual. This can be further enhanced by WAP enabling the system so that the website can be accessed from cellular phones and PDAs by people on the move. This reduces dependence on DSLs, telephone links and other forms of wired communication. This can be done by hosting the website in a WAP server and coding it in WML (Wireless Markup Language).

One of the most common problems faced by firms in a competitive environment is industrial espionage. Hence the data from website can be encrypted in order to increase security. This can make the use of the system in industries possible in real time situations.

This project envisions only switching of devices as control of a system. This can be further improved to incorporate features to transmit control information to the device in the form of multiple parameters as in speed control of motors and precision manufacturing. An important enhancement may be incorporating a feedback loop to the server so that the actual status of the device can be polled.

8. CONCLUSION

The emergence of e-control systems is a milestone in the evolution of computer-aided manufacturing (CAM). The objective of the project was to control six devices through remote access. For this purpose, a web site was put up. The main features of the web site are:

- User authentication
- Remote access to devices

The list of devices is published in a web page. The changes made to the device status are updated in a server-side database. The host machine accesses this database and the relevant device information is retrieved. This is processed by a C program and sent to the ISA add-on card connected to the PC. This is encoded and transmitted using frequency modulation. An FM receiver receives this data, which is then decoded and fed to a microcontroller that controls the devices.

The parameters envisaged during the start of the project were successfully achieved. We expect that our project will create ripples if not waves in the world of e-control systems.

BIBLIOGRAPHY

- McManus, Jefferey P., Database Access with Visual Basic 6 - The authoritative Solution, 1999 Macmillan Computer Publishing, USA.
- Ayala , Kenneth J., The 8051 Microcontroller, 1998 Penram International.
- Frentzen, Jeff & Sobokka, Henry, JavaScript Annotated Archives, 1998 Tata McGraw Hill.
- Rosch, Winn L., Hardware Bible Premier Edition, 1997 Techmedia.
- Farrar, Brian, Special Edition - Using ActiveX, 1996 Que corporation.
- Gaonkar, Ramesh S., 8085 Microprocessor Programming & Applications, 1995 Penram International.
- Martin, James, Telecommunication & the Computer, 1990 Prentice Hall India.
- Smith, Jack R., Modern Communication Circuits, 1998 McGraw Hill Book Company.
- Schoenbeck, Robert J., Electronic Communication - Modulation & Transmission 2/e, 1992 Prentice Hall Inc.
- Afergan, Michael, The Web Programming Desktop Reference 6-in-1, 1996 Prentice Hall India.

APPENDIX A – CODING

8051 Assembly Language Program

```
            org    000h
            mov    p0,#00h
            mov    p2,#00h
            mov    p1,#0ffh
            mov    p3,#00h
back1:      jb     p1.4,back1
back:       jnb    P1.4,back
feed:       mov    a,p1
            anl    a,#0fh
            cjne   a,#07h,power1
            cpl    p3.0
            mov    p0,#01h
            mov    p2,#00h
            lcall  delay
            ljmp   back1
 power1:    cjne   a,#04h,power2
            cpl    p3.1
            mov    p0,#02h
            mov    p2,#00h
            lcall  delay
            ljmp   back1
 power2:    cjne   a,#01h,power3
            cpl    p3.2
            mov    p0,#03h
            mov    p2,#00h
            lcall  delay
            ljmp   back1
 power3:    cjne   a,#08h,power4
            cpl    p3.3
            mov    p0,#04h
            mov    p2,#00h
            lcall  delay
            ljmp   back1
 power4:    cjne   a,#05h,power5
            cpl    p3.4
            mov    p0,#05h
            mov    p2,#00h
            lcall  delay
            ljmp   back1
 power5:    cjne   a,#02h,power6
            cpl    p3.5
            mov    p0,#06h
            mov    p2,#00h
            lcall  delay
```

```
power6: ljmp   back1

delay:   mov    r0,#0ffh
del_1:   mov    r1,#0ffh
del:     djnz   r1,del
         djnz   r0,del_1
         ret
```

C Program

```
#include<stdio.h>
#include<conio.h>
#include<dos.h>

void main(void)
{
        int codeno[6],oldstatus[6],status,devno,i=0;
        char devname[30];
        FILE *fp;

        codeno[0]=0x01;
        codeno[1]=0x02;
        codeno[2]=0x04;
        codeno[3]=0x08;
        codeno[4]=0x10;
        codeno[5]=0x20;

        for(i=0;i<6;i++)
                oldstatus[i]=0;

        i=0;

        fp = fopen("input.txt","r");
        if(!fp) printf("ddd");

        while(!feof(fp))
        {
                fscanf(fp,"%d%s%d",&devno,devname,&status);
                printf("%d--%s--%d\n",devno,devname,status);
                if(oldstatus[i] != status)
                {
                        outportb(0x300,codeno[i]);
                        oldstatus[i]=status;
                }
                sleep(2);
        }
        fclose(fp);
}
```

Internet Programs

Input.asp

```
<%@ Language=VBScript %>
<%Response.Buffer = true%>
<%
if Request.QueryString("code")="add" then
        'Response.Write Request.Form("hide1")
        Set condev = Server.CreateObject("ADODB.Connection")
        condev.Open "dsn=device;uid=;pwd=;database="
        sql="delete from dev"
        condev.execute sql
        condev.close
        'sending datas to text file for accessing by external prg

        set fs=server.CreateObject("Scripting.FileSystemObject")
        set a=fs.CreateTextFile("c:\input.txt")

        Set condev1 = Server.CreateObject("ADODB.Connection")
        condev1.Open "dsn=device;uid=;pwd=;database="
        set rs1 = server.CreateObject("adodb.recordset")
        rs1.CursorType =1
        rs1.CursorLocation =2
        rs1.LockType =3
        rs1.ActiveConnection =condev1
        rs1.open "select * from dev"
        'Response.Write Request.Form ("hide1")
        for i=0 to Request.Form("hide1")-1
                rs1.AddNew
                rs1("DEVICENAME")=Request.form("dname"&i)
                rs1("DEVICENO")=Request.form("dno"&i)
                if Request.Form("checkbox"&i)=checked then
                rs1("STATUS")=false
                else
                rs1("STATUS")=true
                end if
                rs1.Update
        next
        Response.Redirect("confirm.asp")

  else
```

```
        Set condev = Server.CreateObject("ADODB.Connection")
        condev.Open "dsn=device;uid=;pwd=;database="

        set rs = server.CreateObject("adodb.recordset")

        rs.CursorType =1
        rs.CursorLocation =2
        rs.LockType =3
        rs.ActiveConnection =condev
        rs.open "select * from dev"
'Response.Write rs.recordcount
end if
%>

<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
</HEAD>
<BODY>
<form name=inputfrm method=post action="input.asp?code=add">
<P>
<TABLE cellSpacing=1 cellPadding=1 width="75%" border=1>

  <TR>
   <TD><STRONG>DEVICENUMBER</STRONG></TD>
   <TD><STRONG>DEVICE NAME</STRONG></TD>
   <TD><STRONG>STATUS</STRONG></TD>
  </TR>

   <%for i=0 to rs.recordcount-1%>

  <TR>
        <td><input type=text name=dno<%=i%> id=dno<%=i%>
 value=<%=rs("DEVICENO")%>></td>
        <td><input type=text name=dname<%=i%> id=dname<%=i%>
 value=<%=rs("DEVICENAME") %>></td>

         <%if rs("STATUS")=True then%>

                <TD><INPUT id=checkbox<%=i%> type=checkbox
 name=checkbox<%=i%> checked>On/Off</TD>
   </TR>

         <%else%>
         <TD><INPUT id=checkbox<%=i%> type=checkbox
 name=checkbox<%=i%>>On/Off</TD>
   </TR>

         <%end if%>
   <%rs.movenext%>
   <%next%>


  <TR>
   <TD><INPUT id=submit1 type=submit value=Submit name=submit1></TD>
  </TR>
  </TABLE></P>
  <input type=hidden name=hide1 id=hide1 value=<%=rs.recordcount%>>
</form>
</BODY>
</HTML>
```

Confirm.asp

```
<%@ Language=VBScript %>
<%
Response.Write Request.QueryString ("code")
%>
<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
</HEAD>
<BODY>
<form name=frmconfirm method=post action="input.asp">
<P>
<b>Data accepted</b> </P>
<P><INPUT id=submit1 type=submit value=Continue name=submit1></P>
</form>
</BODY>
</HTML>
```

GetText.asp

```
<%@ Language=VBScript %>
<%
        Set condev1 = Server.CreateObject("ADODB.Connection")
        condev1.Open "dsn=device;uid=;pwd=;database="

        set fs=server.CreateObject("Scripting.FileSystemObject")
        set a=fs.CreateTextFile("c:\input.txt")
        set rs1 = server.CreateObject("adodb.recordset")

        rs1.CursorType =1
        rs1.CursorLocation =2
        rs1.LockType =3
        rs1.ActiveConnection =condev1
        rs1.open "select * from dev"
        for i=0 to rs1.RecordCount-1
        number=rs1("DEVICENO")
        status=rs1("STATUS")
        a.writeline(number)
        a.writeline(status)
        rs1.MoveNext
        next
        a.close
%>

<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
</HEAD>
<BODY>

<%response.write "Data written to file"%>

</BODY>
</HTML>
```

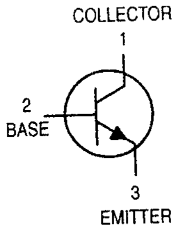
MOTOROLA
SEMICONDUCTOR TECHNICAL DATA

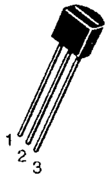
Order this document
by BC546/D

Amplifier Transistors

NPN Silicon

BC546, B
BC547, A, B, C
BC548, A, B, C





CASE 29–04, STYLE 17
TO–92 (TO–226AA)

MAXIMUM RATINGS

Rating	Symbol	BC 546	BC 547	BC 548	Unit
Collector–Emitter Voltage	V_{CEO}	65	45	30	Vdc
Collector–Base Voltage	V_{CBO}	80	50	30	Vdc
Emitter–Base Voltage	V_{EBO}	6.0			Vdc
Collector Current — Continuous	I_C	100			mAdc
Total Device Dissipation @ T_A = 25°C Derate above 25°C	P_D	625 5.0			mW mW/°C
Total Device Dissipation @ T_C = 25°C Derate above 25°C	P_D	1.5 12			Watt mW/°C
Operating and Storage Junction Temperature Range	T_J, T_{stg}	–55 to +150			°C

THERMAL CHARACTERISTICS

Characteristic	Symbol	Max	Unit
Thermal Resistance, Junction to Ambient	$R_{\theta JA}$	200	°C/W
Thermal Resistance, Junction to Case	$R_{\theta JC}$	83.3	°C/W

ELECTRICAL CHARACTERISTICS (T_A = 25°C unless otherwise noted)

Characteristic		Symbol	Min	Typ	Max	Unit
OFF CHARACTERISTICS						
Collector–Emitter Breakdown Voltage (I_C = 1.0 mA, I_B = 0)	BC546	$V_{(BR)CEO}$	65	—	—	V
	BC547		45	—	—	
	BC548		30	—	—	
Collector–Base Breakdown Voltage (I_C = 100 µAdc)	BC546	$V_{(BR)CBO}$	80	—	—	V
	BC547		50	—	—	
	BC548		30	—	—	
Emitter–Base Breakdown Voltage (I_E = 10 µA, I_C = 0)	BC546	$V_{(BR)EBO}$	6.0	—	—	V
	BC547		6.0	—	—	
	BC548		6.0	—	—	
Collector Cutoff Current		I_{CES}				
(V_{CE} = 70 V, V_{BE} = 0)	BC546		—	0.2	15	nA
(V_{CE} = 50 V, V_{BE} = 0)	BC547		—	0.2	15	
(V_{CE} = 35 V, V_{BE} = 0)	BC548		—	0.2	15	
(V_{CE} = 30 V, T_A = 125°C)	BC546/547/548		—	—	4.0	µA

REV 1

ELECTRICAL CHARACTERISTICS (T_A = 25°C unless otherwise noted) (Continued)

Characteristic		Symbol	Min	Typ	Max	Unit
ON CHARACTERISTICS						
DC Current Gain (I_C = 10 µA, V_{CE} = 5.0 V)	BC547A/548A	h_{FE}	—	90	—	—
	BC546B/547B/548B		—	150	—	
	BC548C		—	270		
(I_C = 2.0 mA, V_{CE} = 5.0 V)	BC546		110	—	450	
	BC547		110	—	800	
	BC548		110	—	800	
	BC547A/548A		110	180	220	
	BC546B/547B/548B		200	290	450	
	BC547C/BC548C		420	520	800	
(I_C = 100 mA, V_{CE} = 5.0 V)	BC547A/548A		—	120	—	
	BC546B/547B/548B		—	180	—	
	BC548C		—	300	—	
Collector–Emitter Saturation Voltage (I_C = 10 mA, I_B = 0.5 mA)		$V_{CE(sat)}$	—	0.09	0.25	V
(I_C = 100 mA, I_B = 5.0 mA)			—	0.2	0.6	
(I_C = 10 mA, I_B = See Note 1)			—	0.3	0.6	
Base–Emitter Saturation Voltage (I_C = 10 mA, I_B = 0.5 mA)		$V_{BE(sat)}$	—	0.7	—	V
Base–Emitter On Voltage (I_C = 2.0 mA, V_{CE} = 5.0 V)		$V_{BE(on)}$	0.55	—	0.7	V
(I_C = 10 mA, V_{CE} = 5.0 V)			—	—	0.77	
SMALL–SIGNAL CHARACTERISTICS						
Current–Gain — Bandwidth Product (I_C = 10 mA, V_{CE} = 5.0 V, f = 100 MHz)	BC546	f_T	150	300	—	MHz
	BC547		150	300	—	
	BC548		150	300	—	
Output Capacitance (V_{CB} = 10 V, I_C = 0, f = 1.0 MHz)		C_{obo}	—	1.7	4.5	pF
Input Capacitance (V_{EB} = 0.5 V, I_C = 0, f = 1.0 MHz)		C_{ibo}	—	10	—	pF
Small–Signal Current Gain (I_C = 2.0 mA, V_{CE} = 5.0 V, f = 1.0 kHz)	BC546	h_{fe}	125	—	500	—
	BC547/548		125	—	900	
	BC547A/548A		125	220	260	
	BC546B/547B/548B		240	330	500	
	BC547C/548C		450	600	900	
Noise Figure (I_C = 0.2 mA, V_{CE} = 5.0 V, R_S = 2 kΩ, f = 1.0 kHz, Δf = 200 Hz)	BC546	NF	—	2.0	10	dB
	BC547		—	2.0	10	
	BC548		—	2.0	10	

Note 1: I_B is value for which I_C = 11 mA at V_{CE} = 1.0 V.

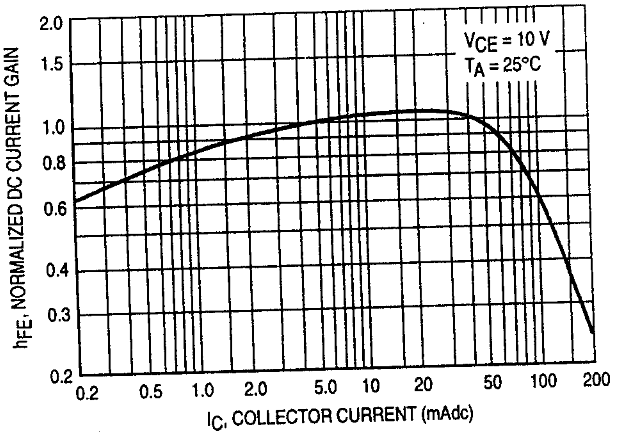


Figure 1. Normalized DC Current Gain

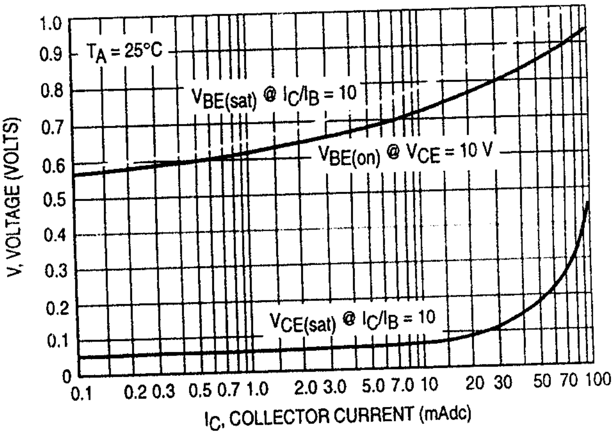


Figure 2. "Saturation" and "On" Voltages

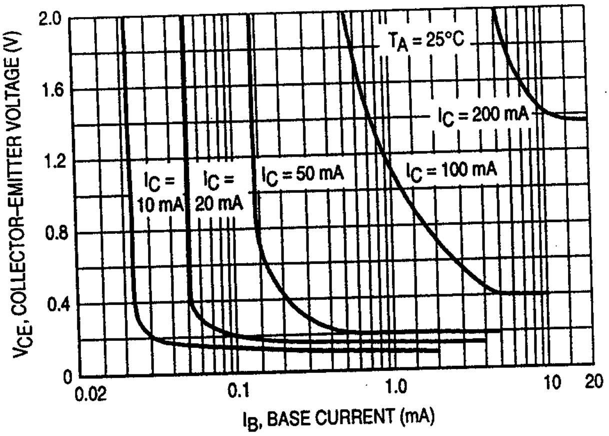


Figure 3. Collector Saturation Region

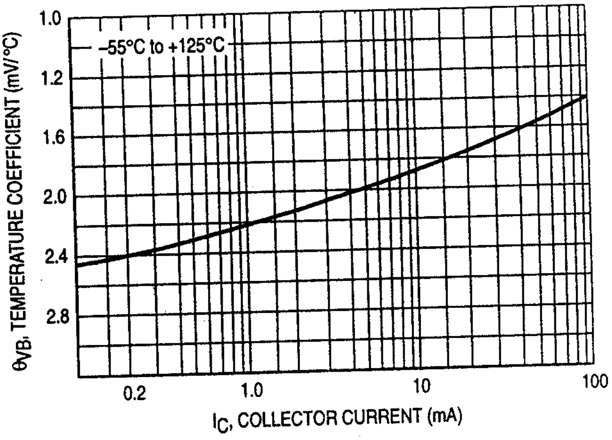


Figure 4. Base–Emitter Temperature Coefficient

BC547/BC548

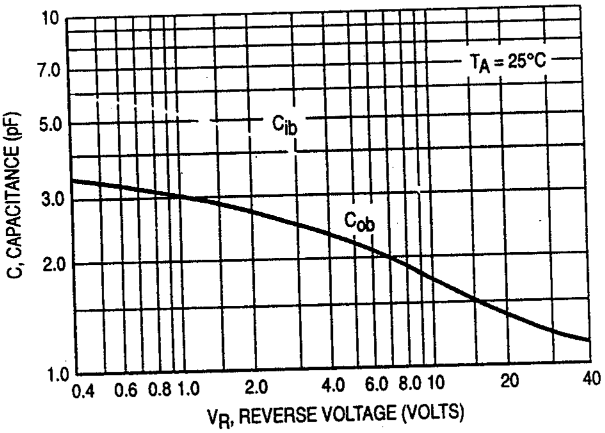


Figure 5. Capacitances

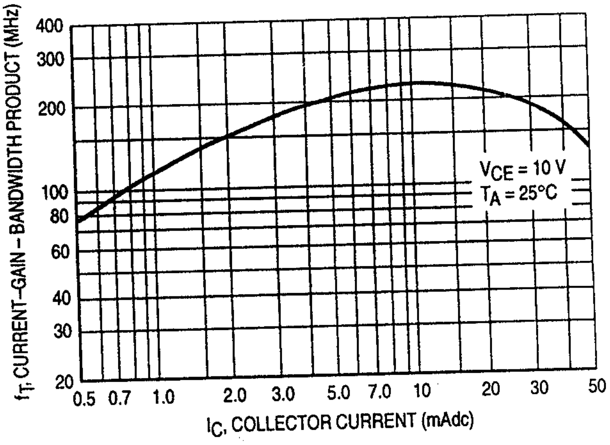


Figure 6. Current–Gain – Bandwidth Product

BC547/BC548

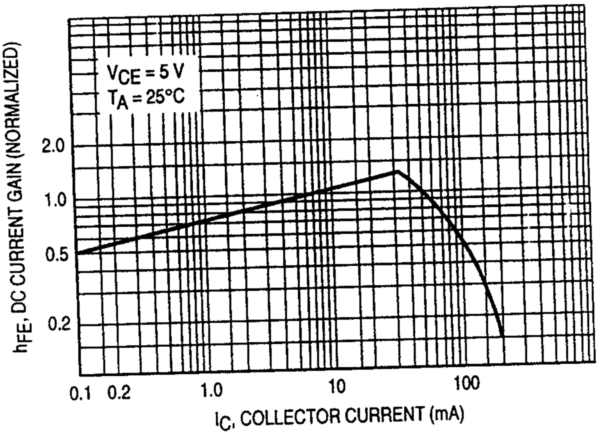


Figure 7. DC Current Gain

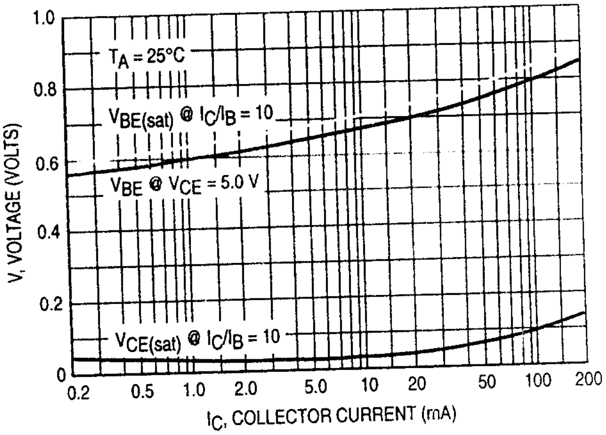


Figure 8. "On" Voltage

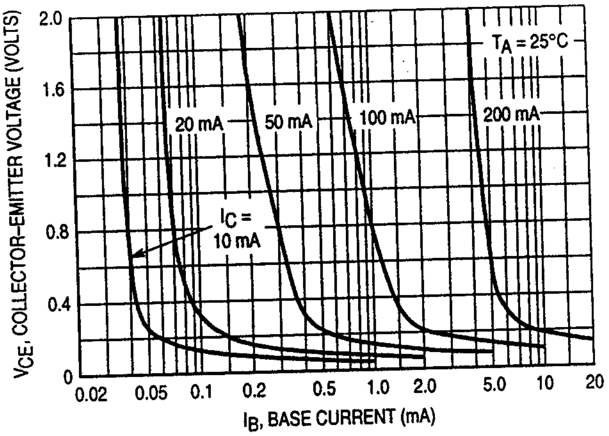


Figure 9. Collector Saturation Region

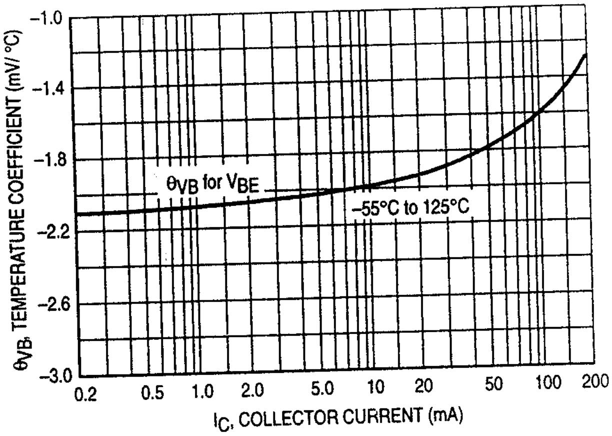


Figure 10. Base–Emitter Temperature Coefficient

BC546

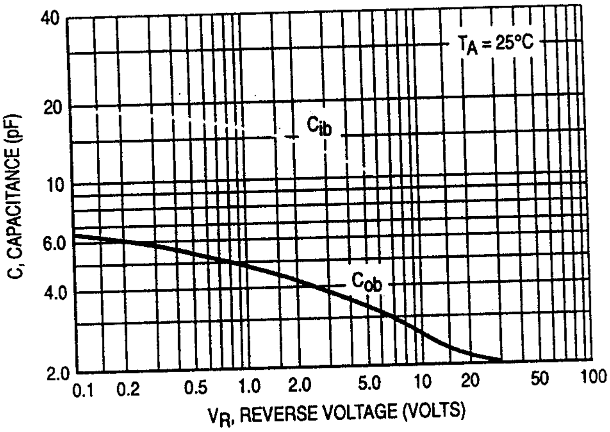


Figure 11. Capacitance

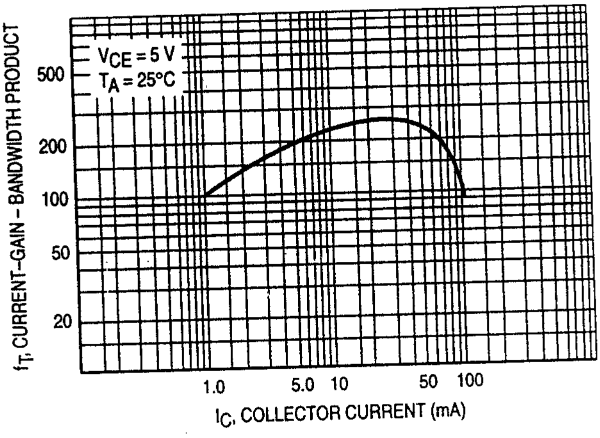


Figure 12. Current–Gain – Bandwidth Product

8-bit Microcontroller with 4K Bytes Flash

AT89C51

eatures

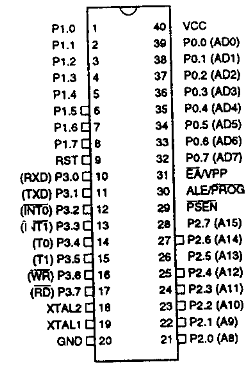
- Compatible with MCS-51™ Products
- 4K Bytes of In-System Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt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

escription

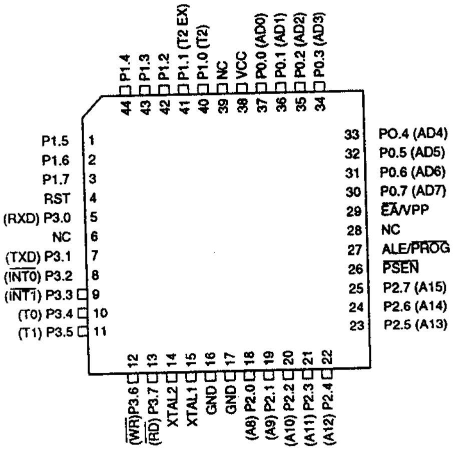
The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K ytes of Flash programmable and erasable read only memory (PEROM). The device s manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 instruction set and pinout. 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.

Pin Configurations

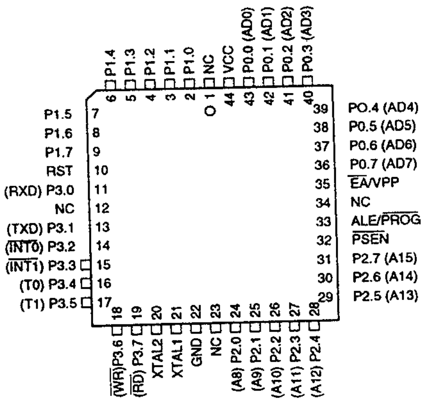
PDIP



PQFP/TQFP

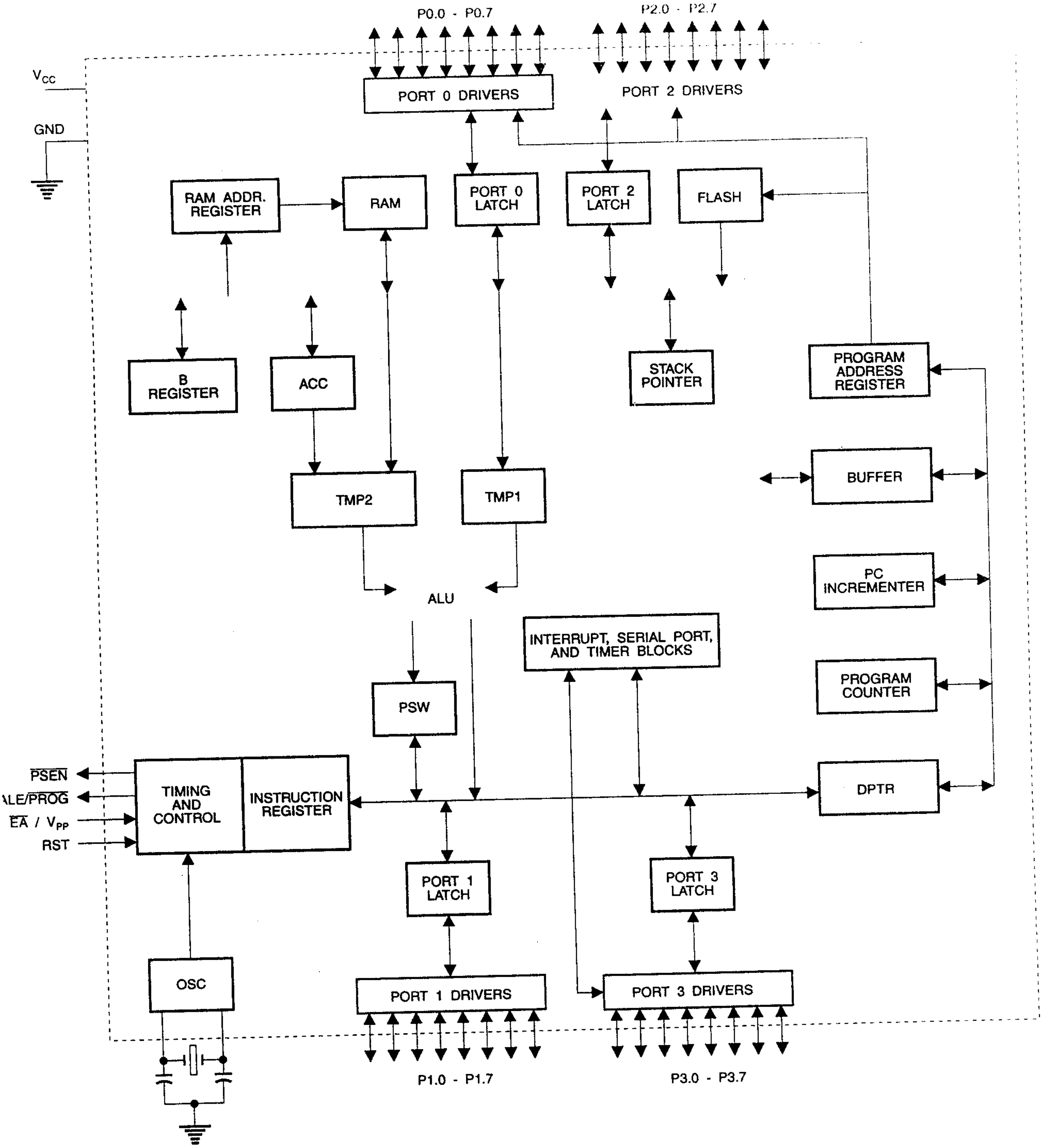


PLCC



Rev. 0265G–02/00

lock Diagram



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 Power-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 ulse is skipped during each access to external Data lemory.

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

$\overline{\text{SEN}}$

Program Store Enable is the read strobe to external proram memory.

Vhen the AT89C51 is executing code from external proram memory, $\overline{\text{PSEN}}$ is activated twice each machine ycle, 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 proram memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, $\overline{\text{EA}}$ will be nternally 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

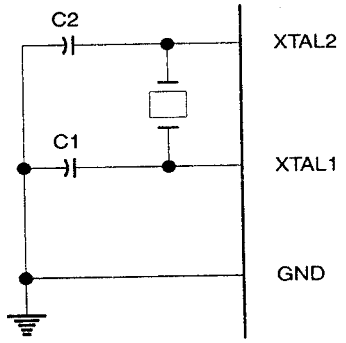
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected 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 onchip 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

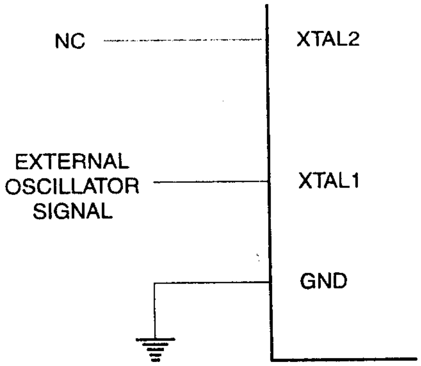


Note: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

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

igure 2. External Clock Drive Configuration



Power-down Mode

In the power-down mode, the oscillator is stopped, and the nstruction 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

Programming the Flash

The AT89C51 is normally shipped with the on-chip Flash memory array in the erased state (that is, contents = FFH) and ready to be programmed. The programming interface accepts either a high-voltage (12-volt) or a low-voltage (V_{CC}) program enable signal. The low-voltage programming mode provides a convenient way to program the AT89C51 inside the user's system, while the high-voltage programming mode is compatible with conventional third-party Flash or EPROM programmers.

The AT89C51 is shipped with either the high-voltage or low-voltage programming mode enabled. The respective top-side marking and device signature codes are listed in the following table.

	V_{PP} = 12V	V_{PP} = 5V
Top-Side Mark	AT89C51 xxxx yyww	AT89C51 xxxx-5 yyww
Signature	(030H) = 1EH (031H) = 51H (032H) =F FH	(030H) = 1EH (031H) = 51H (032H) = 05H

The AT89C51 code memory array is programmed byte-by-byte in either programming mode. *To program any non-blank byte in the on-chip Flash Memory, the entire memory must be erased using the Chip Erase Mode.*

Programming Algorithm: Before programming the AT89C51, the address, data and control signals should be set up according to the Flash programming mode table and Figure 3 and Figure 4. To program the AT89C51, take the following steps.

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise $\overline{\text{EA}}$/V_{PP} to 12V for the high-voltage programming mode.
5. Pulse ALE/$\overline{\text{PROG}}$ once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 1.5 ms. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

$\overline{\text{Data}}$ Polling: The AT89C51 features $\overline{\text{Data}}$ Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written datum on PO.7. Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. $\overline{\text{Data}}$ Polling may begin any time after a write cycle has been initiated.

Ready/$\overline{\text{Busy}}$: The progress of byte programming can also be monitored by the RDY/$\overline{\text{BSY}}$ output signal. P3.4 is pulled low after ALE goes high during programming to indicate BUSY. P3.4 is pulled high again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

Chip Erase: The entire Flash array is erased electrically by using the proper combination of control signals and by holding ALE/$\overline{\text{PROG}}$ low for 10 ms. The code array is written with all "1"s. The chip erase operation must be executed before the code memory can be re-programmed.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 030H, 031H, and 032H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

(030H) = 1EH indicates manufactured by Atmel
(031H) = 51H indicates 89C51
(032H) = FFH indicates 12V programming
(032H) = 05H indicates 5V programming

Programming Interface

Every code byte in the Flash array can be written and the entire array can be erased by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

MOTOROLA

SN54/74LS47

BCD TO 7-SEGMENT DECODER/DRIVER

The SN54/74LS47 are Low Power Schottky BCD to 7-Segment Decoder/Drivers consisting of NAND gates, input buffers and seven AND-OR-INVERT gates. They offer active LOW, high sink current outputs for driving indicators directly. Seven NAND gates and one driver are connected in pairs to make BCD data and its complement available to the seven decoding AND-OR-INVERT gates. The remaining NAND gate and three input buffers provide lamp test, blanking input/ripple-blanking output and ripple-blanking input.

The circuits accept 4-bit binary-coded-decimal (BCD) and, depending on the state of the auxiliary inputs, decodes this data to drive a 7-segment display indicator. The relative positive-logic output levels, as well as conditions required at the auxiliary inputs, are shown in the truth tables. Output configurations of the SN54/74LS47 are designed to withstand the relatively high voltages required for 7-segment indicators.

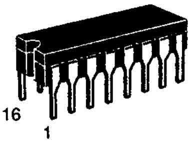
These outputs will withstand 15 V with a maximum reverse current of 250 µA. Indicator segments requiring up to 24 mA of current may be driven directly from the SN74LS47 high performance output transistors. Display patterns for BCD input counts above nine are unique symbols to authenticate input conditions.

The SN54/74LS47 incorporates automatic leading and/or trailing-edge zero-blanking control (RBI and RBO). Lamp test (LT) may be performed at any time which the BI/RBO node is a HIGH level. This device also contains an overriding blanking input (BI) which can be used to control the lamp intensity by varying the frequency and duty cycle of the BI input signal or to inhibit the outputs.

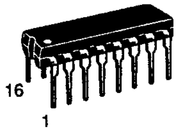
- Lamp Intensity Modulation Capability (BI/RBO)
- Open Collector Outputs
- Lamp Test Provision
- Leading/Trailing Zero Suppression
- Input Clamp Diodes Limit High-Speed Termination Effects

BCD TO 7-SEGMENT DECODER/DRIVER

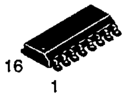
LOW POWER SCHOTTKY



J SUFFIX
CERAMIC
CASE 620-09



N SUFFIX
PLASTIC
CASE 648-08

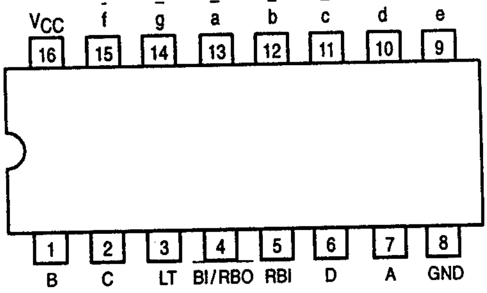


D SUFFIX
SOIC
CASE 751B-03

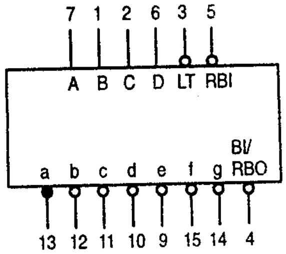
ORDERING INFORMATION

SN54LSXXJ	Ceramic
SN74LSXXN	Plastic
SN74LSXXD	SOIC

CONNECTION DIAGRAM DIP (TOP VIEW)



LOGIC SYMBOL



VCC = PIN 16
GND = PIN 8

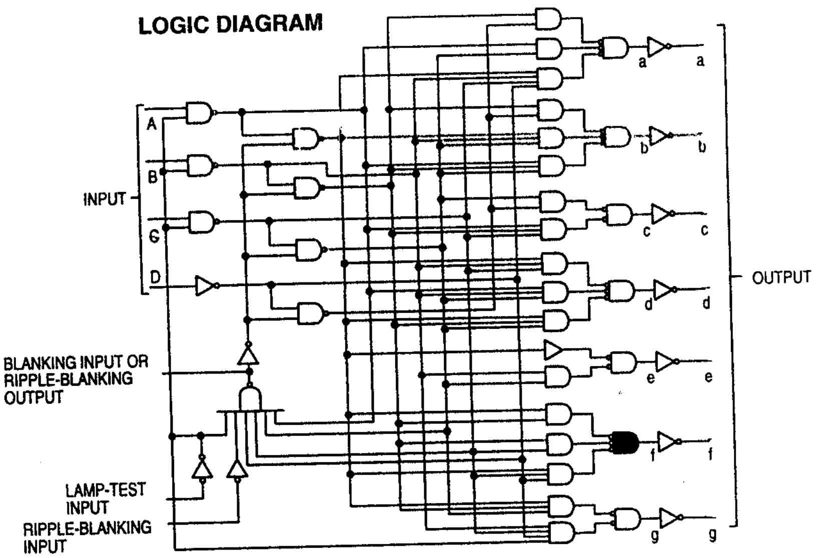
PIN NAMES

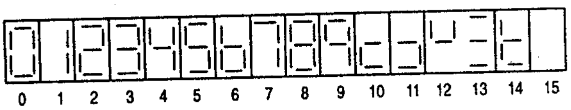
		LOADING (Note a)	
		HIGH	LOW
A, B, C, D	BCD Inputs	0.5 U.L.	0.25 U.L.
RBI	Ripple-Blanking Input	0.5 U.L.	0.25 U.L.
LT	Lamp-Test Input	0.5 U.L.	0.25 U.L.
BI/RBO	Blanking Input or	0.5 U.L.	0.75 U.L.
	Ripple-Blanking Output	1.2 U.L.	2.0 U.L.
a, to g	Outputs	Open-Collector	15 (7.5) U.L.

NOTES:
a) 1 Unit Load (U.L.) = 40 µA HIGH, 1.6 mA LOW.
b) Output current measured at V_{OUT} = 0.5 V
The Output LOW drive factor is 7.5 U.L. for Military (54) and 15 U.L. for Commercial (74) Temperature Ranges.

SN54/74LS47

LOGIC DIAGRAM





NUMERICAL DESIGNATIONS — RESULTANT DISPLAYS

TRUTH TABLE

	INPUTS							OUTPUTS							
DECIMAL OR FUNCTION	$\overline{LT}$	$\overline{RBI}$	D	C	B	A	$\overline{BI/RBO}$	$\bar{a}$	$\bar{b}$	$\bar{c}$	$\bar{d}$	$\bar{e}$	$\bar{f}$	$\bar{g}$	NOTE
0	H	H	L	L	L	L	H	L	L	L	L	L	L	H	A
1	H	X	L	L	L	H	H	H	L	L	H	H	H	H	A
2	H	X	L	L	H	L	H	L	L	H	L	L	H	L	
3	H	X	L	L	H	H	H	L	L	L	L	H	H	L	
4	H	X	L	H	L	L	H	H	L	L	H	H	L	L	
5	H	X	L	H	L	H	H	L	H	L	L	H	L	L	
6	H	X	L	H	H	L	H	H	H	L	L	L	L	L	
7	H	X	L	H	H	H	H	L	L	L	H	H	H	H	
8	H	X	H	L	L	L	H	L	L	L	L	L	L	L	
9	H	X	H	L	L	H	H	L	L	L	H	H	L	L	
10	H	X	H	L	H	L	H	H	H	H	L	L	H	L	
11	H	X	H	L	H	H	H	H	H	L	L	H	H	L	
12	H	X	H	H	L	L	H	H	L	H	H	H	L	L	
13	H	X	H	H	L	H	H	L	H	H	L	H	L	L	
14	H	X	H	H	H	L	H	H	H	H	L	L	L	L	
15	H	X	H	H	H	H	H	H	H	H	H	H	H	H	
BI	X	X	X	X	X	X	L	H	H	H	H	H	H	H	B
RBI	H	L	L	L	L	L	L	H	H	H	H	H	H	H	C
LT	L	X	X	X	X	X	H	L	L	L	L	L	L	L	D

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial

NOTES:

(A) BI/RBO is wire-AND logic serving as blanking Input (BI) and/or ripple-blanking output (RBO). The blanking out (BI) must be open or held at a HIGH level when output functions 0 through 15 are desired, and ripple-blanking input (RBI) must be open or at a HIGH level if blanking of a decimal 0 is not desired. X = input may be HIGH or LOW.

(B) When a LOW level is applied to the blanking input (forced condition) all segment outputs go to a LOW level regardless of the state of any other input condition.

(C) When ripple-blanking input (RBI) and inputs A, B, C, and D are at LOW level, with the lamp test input at HIGH level, all segment outputs go to a HIGH level and the ripple-blanking output (RBO) goes to a LOW level (response condition).

(D) When the blanking input/ripple-blanking output (BI/RBO) is open or held at a HIGH level, and a LOW level is applied to lamp test input, all segment outputs go to a LOW level.

GUARANTEED OPERATING RANGES

Symbol	Parameter		Min	Typ	Max	Unit
V_{CC}	Supply Voltage	54 74	4.5 4.75	5.0 5.0	5.5 5.25	V
T_A	Operating Ambient Temperature Range	54 74	−55 0	25 25	125 70	°C
I_{OH}	Output Current — High BI/RBO	54, 74			−50	µA
I_{OL}	Output Current — Low BI/RBO BI/RBO	54 74			1.6 3.2	mA
$V_{O\ (off)}$	Off-State Output Voltage $\bar{a}$ to $\bar{g}$	54, 74			15	V
$I_{O\ (on)}$	On-State Output Current $\bar{a}$ to $\bar{g}$ $\bar{a}$ to $\bar{g}$	54 74			12 24	mA

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

Symbol	Parameter		Limits			Unit	Test Conditions	
			Min	Typ	Max			
V_{IH}	Input HIGH Voltage		2.0			V	Guaranteed Input HIGH Theshold Voltage for All Inputs	
V_{IL}	Input LOW Voltage	54			0.7	V	Guaranteed Input LOW Threshold Voltage for All Inputs	
		74			0.8			
V_{IK}	Input Clamp Diode Voltage			−0.65	−1.5	V	V_{CC} = MIN, I_{IN} = −18 mA	
V_{OH}	Output HIGH Voltage, $\overline{BI}/\overline{RBO}$		2.4	4.2		V	V_{CC} = MIN, I_{OH} = −50 µA, V_{IN} = V_{IN} or V_{IL} per Truth Table	
V_{OL}	Output LOW Voltage BI/RBO	54, 74		0.25	0.4	V	I_{OL} = 1.6 mA	V_{CC} = MIN, V_{IN} = V_{IN} or V_{IL} per Truth Table
		74		0.35	0.5	V	I_{OL} = 3.2 mA	
$I_{O\ (off)}$	Off-State Output Current a thru g				250	µA	V_{CC} = MAX, V_{IN} = V_{IN} or V_{IL} per Truth Table, $V_{O\ (off)}$ = 15 V	
$V_{O\ (on)}$	On-State Output Voltage a thru g	54, 74		0.25	0.4	V	$I_{O\ (on)}$ = 12 mA	V_{CC} = MAX, V_{IN} = V_{IH} or V_{IL} per Truth Table
		74		0.35	0.5	V	$I_{O\ (on)}$ = 24 mA	
I_{IH}	Input HIGH Current				20	µA	V_{CC} = MAX, V_{IN} = 2.7 V	
					0.1	mA	V_{CC} = MAX, V_{IN} = 7.0 V	
I_{IL}	Input LOW Current BI/RBO Any Input except BI/RBO				−1.2 −0.4	mA	V_{CC} = MAX, V_{IN} = 0.4 V	
I_{OS} BI/RBO	Output Short Circuit Current (Note 1)		−0.3		−2.0	mA	V_{CC} = MAX, V_{OUT} = 0 V	
I_{CC}	Power Supply Current			7.0	13	mA	V_{CC} = MAX	

Note 1: Not more than one output should be shorted at a time, nor for more than 1 second.

AC CHARACTERISTICS (T_A = 25°C)

Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
t_{PHL} t_{PLH}	Propagation Delay, Address Input to Segment Output			100 100	ns ns	V_{CC} = 5.0 V C_L = 15 pF
t_{PHL} t_{PLH}	Propagation Delay, RBI Input To Segment Output			100 100	ns ns	

AC WAVEFORMS

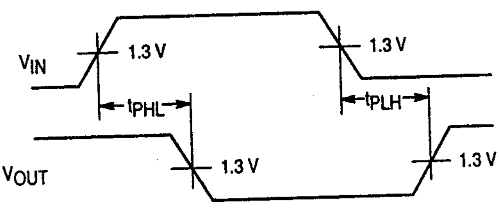


Figure 1

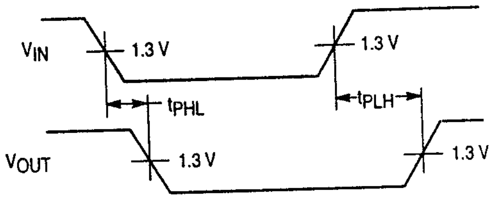
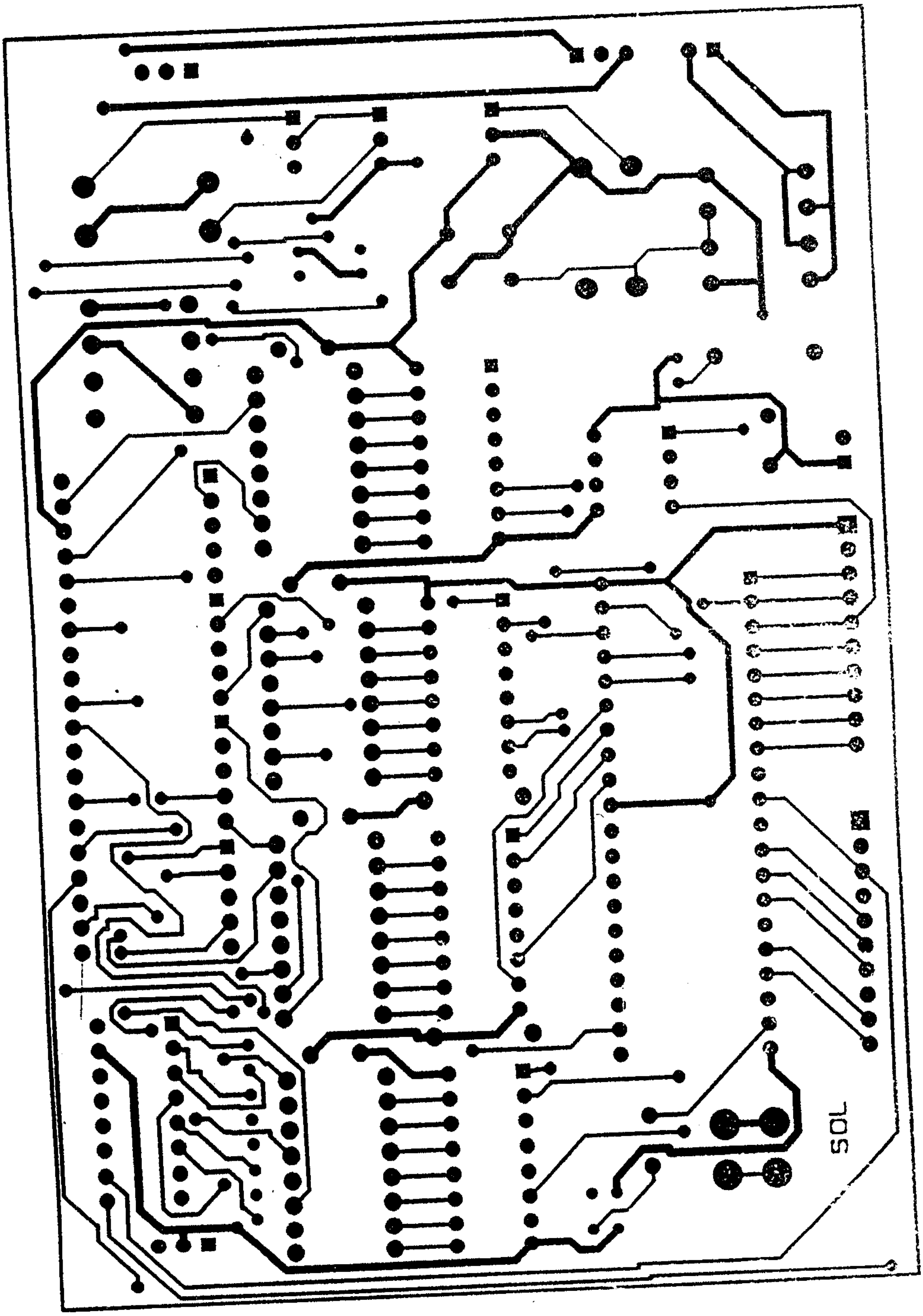
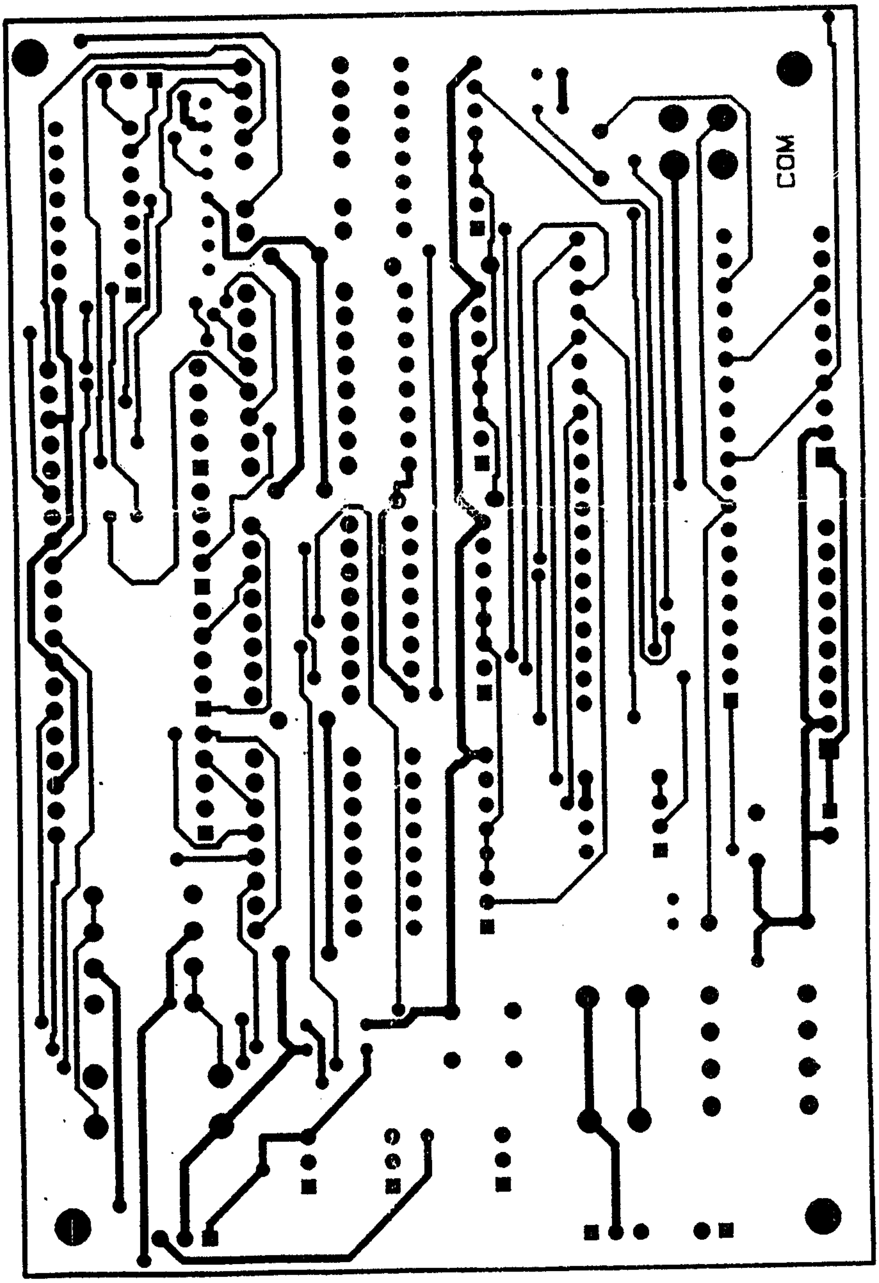


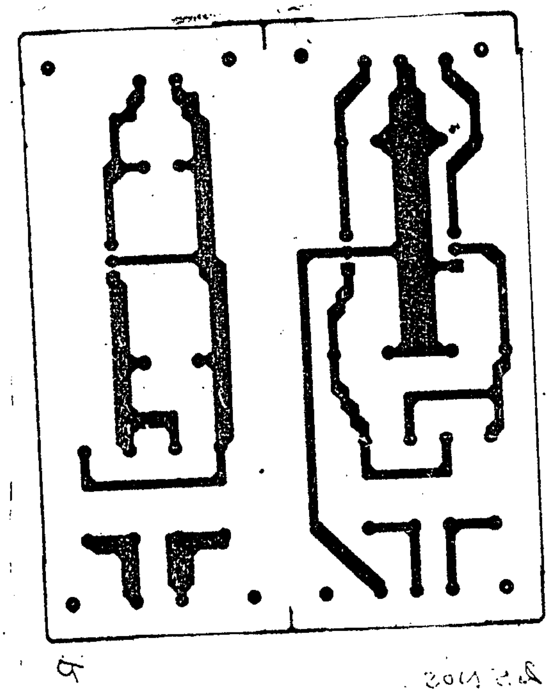
Figure 2



ISA ADD-ON CARD



MICROCONTROLLER & LED DISPLAY BOARD



POWER SUPPLY