P-1602

REMOTE FAULT MONITORING SYSTEM

A PROJECT REPORT

Submitted by

SARANYA P. 71202104036
SUGANYA P. 71202104045
UDHAYA S. 71202104054

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE-06

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2006

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"REMOTE FAULT MONITORING SYSTEM"** is the bonafide work of "P.SARANYA, P.SUGANYA, and S.UDHAYA" who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
(Dr.S.Thangasamy)	(Mr.G.S.Nandakumar)
DEAN	**SUPERVISOR**
	Senior Lecturer
Department of	Department of
Computer Science and Engineering,	Computer Science and Engineering,
Kumaraguru College of Technology,	Kumaraguru College of Technology,
Coimbatore – 641006.	Coimbatore – 641006

The candidates with University Register Nos. 71202104036, 71202104045 and 71202104054 were examined by us in the project viva-voce examination held on 02.05.2006

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ANNA UNIVERSITY: CHENNAI 600 025

EVALUATION CERTIFICATE

College : KUMARAGURU COLLEGE OF TECHNOLOGY
Branch : COMPUTER SCIENCE AND ENGINEERING
Semester: EIGHT (08)

S.NO	Name of the students	Title of the project	Name of Supervisor
1.	P.Saranya	REMOTE FAULT	
2.	P.Suganya	MONITORING	Mr.G.S.Nandakumar
3.	S.Udhaya	SYSTEM	Sr. Lecturer

The report of the project work submitted by the above students in partial fulfillment for the award of BACHELOR OF ENGINEERING degree in COMPUTER SCIENCE AND ENGINEERING of Anna University were evaluated and confirmed to the report of the work done by the above students and then evaluated.

INTERNAL EXAMINER EXTERNAL EXAMINER

DECLARATION

We hereby declare that the project entitled "REMOTE FAULT MONITORING SYSTEM" is done by us and to the best of our knowledge a similar to the Anna University or any other Institution, for fulfillment of the requirement of the course study.

This report is submitted on the partial fulfillment of the requirements for all awards of the Degree of Bachelor of Computer Science & Engineering of Anna University, Chennai.

Place : Coimbatore
Date : 02.05.2006

P.Saranya

P.Suganya

S.Udhaya

ACKNOWLEGEMENT

We are extremely grateful to **Dr.K.K.Padmanabhan**, B.Sc (Engg.)., M.Tech., Ph.D., Principal, Kumaraguru College of Technology for having given us a golden opportunity to embark on this project.

We are deeply obliged to **Dr.S.Thangasamy,** Ph.D., Dean, Department of Computer Science and Engineering for his valuable guidance and useful suggestions during the course of this project.

We extend our heartfelt thanks to our course coordinator **Mrs.P.Devaki,** M.S., Assistant Professor, Department of Computer Science and Engineering for her valuable support given to us throughout this project.

We are extremely thankful to our project guide **Mr.G.S.Nandakumar,** B.E, Senior Lecturer, Department of Computer Science and Engineering, for his helpful guidance and support for completing this project.

We thank our class advisor **Mrs.Amudha Venkatessh,** M.E., Senior Lecturer, Department of Computer Science and Engineering, for providing us the moral support for completing this project.

We thank the **teaching and non-teaching staff** of our department for providing us the technical support in the duration of our project.

We also thank our **family and friends** for giving us the moral support and abundant blessing in all our activities to complete this project successfully.

ABSTRACT

Remote monitoring has become a watchword in current day computing technology. Keeping this perspective in mind the project entitled "Remote Fault Monitoring System" is developed to assist users in controlling a remote machine without having to be in direct physical contact, with the system to be monitored. This system is developed with a view towards minimizing the task of the remote accessing to identify the faults.

More and more companies are moving towards remote access in their manufacturing processes. The general operation of the project is very simple. This proposed system scans for abnormal conditions in the machinery being monitored. Sensors mounted at the bearing locations of the machinery send information to the manufacturer system via the internet.

If the system detects any abnormal condition it immediately generates a voice alert and sends this message to manufacturer system as an email message via the Internet. Once the message is received, any individual or consultant can diagnose and report on the cause of the condition from the monitoring system.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE
	ACKNOWLEDGEMENT	**iii**
	ABSTRACT	**iv**
	LIST OF FIGURES	**ix**
	LIST OF TABLES	**x**
	LIST OF ABBREVIATIONS	**xi**
1	**INTRODUCTION**	**1**
	1.1 Existing System	1
	1.2 Proposed System	1
	1.3 System Overview	1
	1.3.1 Block Diagram	2
	1.4 System Requirements	4
	1.4.1 Hardware Requirements	4
	1.4.2 Software Requirements	4
2	**UPS**	**5**
	2.1 UPS Overview	**5**
	2.2 Types of UPS System	7
	2.2.1 Offline UPS	7
	2.2.2 Online UPS	8
	2.3 Parts of the Online UPS	10
	2.3.1 Battery Charger	10
	2.3.2 Battery	11
	2.3.3 Inverter	12

	2.3.3.1 NE555 Timer	13
	2.3.3.2 4013 Flip-Flop	15
	2.3.3.3 IRF244 MOSFET	15
	2.3.3.4 Step up Transformer	16
3	**SENSORS**	**17**
	3.1 Introduction	17
	3.2 Voltage Sensors	17
	3.3 Temperature Sensor	18
4	**MICROCONTROLLER (AT89C51)**	**20**
	4.1 Introduction	20
	4.2 Features	21
	4.3 Block Diagram	21
	4.4 Description	22
	4.5 Oscillator Characteristics	22
	4.6 Memory organization	23
	4.6.1 Flash ROM	23
	4.6.2 RAM	24
	4.6.3 Program Memory	24
	4.6.4 Data Memory	25
	4.7 Addressing Modes	27
	4.7.1 Direct Addressing	27
	4.7.2 Indirect Addressing	27
	4.7.3 Indexed Addressing	27
	4.8 Power Modes of AT89C51	28
	4.8.1 Idle Mode	28
	4.8.2 Power down Mode	28
	4.8.3 Power on Reset	28

	4.9 Interrupt	29
	4.10 Timer/Counters	30
	4.11 Input/Output Ports	30
5	**RS232 SERIAL INTERFACE**	**33**
	5.1 RS232 Serial Interfacing	34
	5.1.1 RS232C	34
	5.1.2 RS232C Representations	35
	5.2 MAX232	36
	5.2.1 Baud Vs Bits per second	37
	5.2.2 Cables and Gender Changes	38
	5.2.3 Null Modems Cables and Adapter	38
	5.2.4 Functional Characteristics	39
	5.2.5 Maximum Data Rate	39
6	**HARDWARE CIRCUITS**	**40**
	6.1 Power Supply	40
	6.1.1 Transformer	40
	6.1.2 Rectifier	41
	6.1.3 Filter	41
	6.1.4 IC Voltage Regulator	42
	6.1.5Three Terminal Voltage Regulator	42
	6.2 Interfacing IC MAX232	43
	6.2.1 MAX232 Description	43
	6.2.2 Interfacing Circuit	45
	6.3 Modes of Communication	46
	6.4 Computer Ports	47

7	**SOFTWARE DESCRIPTION**	**48**
	7.1 Overview of the .NET	48
	7.2 What is Microsoft .NET?	49
	7.3 Microsoft .NET Programming Model	49
	7.4 The Improving Technology	50
	7.5 Microsoft .NET Framework Architecture	51
	7.6 Features of the .NET Framework	52
	7.7 Advantages of using .NET framework	53
	7.8 Namespaces in .NET	56
	7.9 Visual Basic .NET	57
	7.10 Conclusion	**59**
8	**CONCLUSION**	**60**
9	**APPENDICES**	61
	9.1 89C51 Microcontroller Data Sheet	61
	9.2 RS232 Drivers DF Data Sheet	69
	9.3 NE555 Timer Data Sheet	74
	9.4 Sample screen	80
	9.5 Sample codes	81
10	**REFERENCES**	**84**

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
1.1	Basic Block Diagram	2
2.1	Basic Block Diagram of a UPS	6
2.2	Basic Block Diagram of a Online UPS	9
2.3	Basic Block Diagram of Battery Charger	10
2.4	NE555 Timer Functional diagram	13
4.1	89C51 Architecture	21
4.2	External Clock Drive Configurations	22
4.3	Oscillator Connections	22
4.4	Flash ROM	23
4.5	Program Memory	24
4.6	Data Memory	26
5.1	RS232 DB9 Pin out	34
5.2	MAX232 Pin Diagram	37
6.1	Basic Block Diagram of Power Supply	40
6.2	Circuit Diagram of the Power Supply	41
6.3	IC Voltage Regulators	42
6.4	MAX232 Pin Out	44
6.5	MAX232 Interfacing to RS232 Connector	45

LIST OF TABLES

TABLE NO	TABLE NAME	PAGE NO
4.1	Port 3 – Alternate functions	32
5.1	Pin Description of Serial Port RS232	35
6.1	MAX232/232A Comparison	43

LIST OF ABBREVIATIONS

S.No	ABBREVIATION	EXPANSION
1)	AC	Alternating Current
2)	API	Application Programming Interface
3)	BCL	Base Class Library
4)	COM	Component Object Model
5)	CLR	Common Language Runtime
6)	CLS	Common Language Specifications
7)	DC	Direct Current
8)	DLL	Dynamic Link Library
9)	MSIL	Microsoft Intermediate Language
10)	RMS	Root Mean Square
11)	UPS	Uninterrupted Power Supply

CHAPTER 1
INTRODUCTION

1.1 EXISTING SYSTEM

The existing system is a manual process of identifying the fault and to fix the problems. The drawbacks of the system are

- Time consuming
- Requires more manual labour
- Remote accessing is not possible
- Administrator lacks complete control

1.2 PROPOSED SYSTEM

Owing to the above mentioned drawbacks in the existing system, an automated system is proposed. The proposed system aims to eliminate these drawbacks. It can be viewed as user friendly, efficient easing the work of the administrator.

The benefits of the proposed system are

- Less time consuming
- Monitoring the fault in the remote area are made easier
- Administrator will be able to know the current status of the remote machine

1.3 SYSTEM OVERVIEW

Remote monitoring has become a watchword in current day computing technology. Keeping this perspective in mind the project entitled "Remote Fault Monitoring System" is developed to assist users in controlling a remote machine without having to be in direct physical contact, with the system to be monitored.

This system is developed with a view towards minimizing the task of the remote accessing to identify the faults. Any variations from the normal functioning of the machines will produce drastic change in the output of the industry. Hence these variations should be detected at the earliest with most accuracy and is to be informed to the concerned authority. This process is being automated in our project. General block diagram of the project is shown in figure 1.1

1.1 BASIC BLOCK DIAGRAM

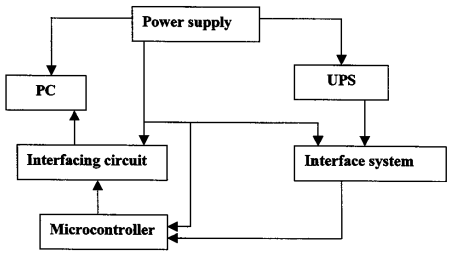


Figure 1.1 Basic Block Diagram

Figure 1.1 on the previous page provides an overview of the system architecture. The general operation of the project is very simple. Today the Internet plays a vital role in information management for industry. More and more companies are moving towards remote access in their manufacturing processes. Sensors mounted at the bearing locations of your machinery send information to the manufacturer system via the internet.

The system scans for abnormal levels in the machinery being monitored. If the system detects any abnormal condition it immediately generates a voice alert and sends this message to manufacturer system as an email message via the Internet. Once the message is received, any individual or consultant can diagnose and report on the cause of the condition from the monitoring system.

The requirement specified in our system is to find the fault that occurs in the UPS. Uninterrupted Power Supply (UPS) is the simulation model for our project. The sensors present in the UPS recognize the faults and indicates it to the microcontroller. Microcontroller used here is ATMEL 89C51. All the signals from the sensors are given to the microcontroller. The microcontroller is programmed in such a way that depending upon the signal received, the fault is identified or the parameter that is monitored is obtained.

The recognized faults in the UPS are battery fault, the inverter fault and excess heat due to overload condition. The identified fault is sent to the monitoring PC through interfacing circuit (RS232 serial interface). In order to make the microcontroller compatible with that of the PC, we use MAX232 interfacing. This converts TTL voltage levels into RS232 voltage levels. The RS232 is a voltage converter which converts the microcontroller voltage levels to the PC understandable form.

The monitoring system generates an e-mail automatically and sends it to the manufacturer site. The PC in the remote site generates a voice alert after the mail is generated. The fault information is stored in the database at the remote site. The consultant can then monitor and diagnose problems from his/her office. This eliminates the need for an on site visit.

1.4 SYSTEM REQUIREMENTS

1.4.1 HARDWARE REQUIREMENTS

Processor	:	Pentium-IV
Keyboard	:	102 keys keyboard
Mouse	:	Scrollbar mouse
Monitor	:	15'' SVGA monitor
Hard disk	:	10GB
RAM	:	128 MB RAM

1.4.2 SOFTWARE REQUIREMENTS

Operating System	:	Windows XP
Software Used	:	Microsoft Visual Studio .NET 2003

CHAPTER 2
UPS

2.1 UPS (Uninterruptible Power Supply) OVERVIEW

A UPS is basically an inverter designed to supply power without any interruption to the equipment connected to it. The fundamental purpose of a UPS is to provide an uninterruptible source of power for the equipment it protects. It is also supposed to generate and supply power to the equipment in case of AC mains power failure. When the AC mains is present the UPS takes this power and stores some energy in its batteries for the backup (battery charging), and then provides the required AC output to the end equipment. In case of input AC mains failure its inverter block takes the energy from the batteries and converts the battery voltage to the required AC output.

UPSes are designed so that there is one source of power that is normally used, called the primary power source, and another source that kicks in if the primary is disrupted, called the secondary power source. The power from the wall is always one of these sources, and the battery contained within the UPS is the other. A switch is used to control which of these sources powers the equipment at any given time. The switch changes from the primary source to the secondary when it detects that the primary power has gone out.

It switches back from the secondary power source to the primary when it detects that the primary power source has returned. Of course, the power that comes from the wall is AC, and your PC uses AC power as well. All batteries, however, provide DC power. Therefore, circuitry is provided within all UPSes to convert AC power to DC to charge the battery.

A device called an inverter is also provided to change the battery's stored DC electricity to AC to run your equipment.

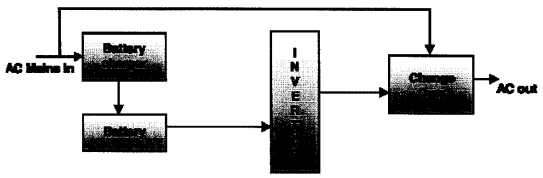


Figure 2.1: Basic block diagram of a UPS

The difference between an inverter and a UPS

The basic technology of both an inverter and a UPS is the same. At the time of automatic changeover from AC mains to battery backup, and battery backup to AC mains, there is always a time delay depending upon the type of changeover system used. In the inverter this delay is more, as cheaper components are used, in comparison to a UPS. In a UPS this delay is always less than the maximum delay allowed to protect the end equipment from getting reset. Another difference is in the quality of output AC. The output of an inverter is usually, but not necessarily, a square wave but the output of a UPS is supposed to be pure sine wave, or as close to it as possible. A UPS can be used in place of an inverter but not vice versa.

2.2 TYPES OF UPS SYSTEMS

There are two types of UPS system configurations:

- Offline UPS
- Online UPS

2.2.1 OFFLINE UPS

In offline UPS systems the inverter portion is only powered up when the input mains has some problem. In offline configurations a small hole or defect can be present at the output as well, but this is generally not corrected as many types of equipment can withstand it. This also permits designers to take liberties with the shape of the output waveform and enables the system makers to offer a nonsinusoidal, modified square wave (In other terms, quasi sine wave).

This simplifies the design and makes these systems cost-effective. Some offline UPS systems offer more advanced features, such as the ability to compensate for limited variations of input mains. But an offline UPS is generally more efficient when a mains power source is present. The main drawbacks of this system are

- The output mains is almost exactly equal to the input, until some major defect is detected, as no correction is offered (or is required to be offered) in the AC mains. Any glitches and transients are passed on to the end equipment.
- The output is not pure sinusoidal.
- The changeover system is prone to stress.
- The battery and inverter cannot be tested till the mains fails

To overcome the last problem some strategies can be implemented, such as periodical battery testing and power cycling the inverter. But nothing can be done in case the mains are not perfectly sinusoidal and the load requires it. A complete conversion of power (as in an online UPS) also preferred because the load tends to produce harmonics and disturbances, and injects these back to the input mains. In this case, especially when high power loads and inductive loads are involved, online UPS systems can be very effective in separating loads and mains.

2.2.2 ONLINE UPS

Ideally, only an online UPS should be called a UPS because it has all the features that can deliver an uninterrupted power output. Its principle of operation is different from that of an inverter or offline UPS. The online UPS sometimes called a true UPS. In the online UPS the primary power source is the UPS's battery, and utility power is the secondary power source. In an ideal online UPS there is no changeover delay and it delivers distortion-free pure sine wave regulated output.

The online UPS also protects end equipment from any surge and bad quality of input AC mains. In an online UPS the load is supplied continuously by a power transformed inside the UPS, with the battery connected as a buffer. This way, any problem in the input mains can never cause loss of power to the load as long as the battery can supply the needed energy to the output inverter. If the input mains are absent for a long time, at a certain point (depending on the load and battery capacity) the battery discharges below the minimum limit for the inverter to function properly and the load must be disconnected.

In online UPS systems the input stage must be sized so that it can supply the output power plus the energy for charging the battery. The output stage is basically a switch normally composed of solid-state components, so that a commutation between input mains and inverter (for maintenance or other reasons) could be performed with no significant hole at the output. The inverter runs directly from the battery voltage at all times or a boost stage can optionally be connected between these in case low voltage batteries are used.

There is no doubt that an online UPS is always better than an offline one. The online UPS delivers more reliable and controlled regulated output and protects the end equipment from the bad quality input AC mains. The major drawback of an online UPS is its complicated design and high cost.

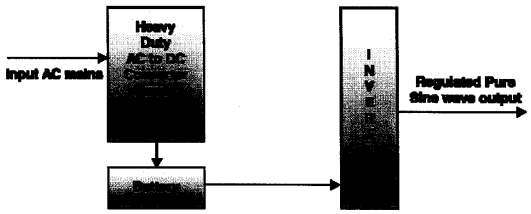


Figure 2.2: Basic block diagram of an online UPS

2.3 PARTS OF THE ONLINE UPS

2.3.1 BATTERY CHARGER

All UPSes include core circuitry that manipulates electricity, converting it from the AC power produced by your utility company to DC power stored in the battery, and back again for use by your equipment. Of course, the power that comes from the wall is AC, and your PC uses AC power as well. All batteries, however, provide DC power.

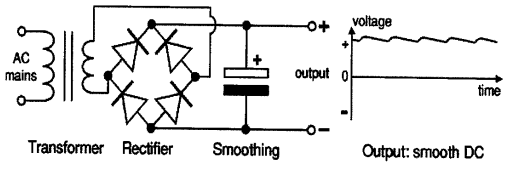


Figure 2.3 Block Diagram of the Battery Charger

TRANSFORMER

A transformer is a static (or stationary) piece of which electric power in one circuit is transformed 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 with the principle of mutual induction. A step-down transformer, which steps down the 230volt AC supply to 12-0-12V AC supply its current rating, is 1Amp. The transformer is of the core type in order to convert AC RMS voltage in to DC voltage, a rectifier is required.

RECTIFIER

A bridge rectifier can be made using four individual diodes, but it is also available in special packages containing the four diodes required. It is called a full-wave rectifier because it uses the entire AC wave (both positive and negative sections). 1.4V is used up in the bridge rectifier because each diode uses 0.7V when conducting and there are always two diodes conducting. During the positive half cycle conduct and during the negative half cycle remaining two were conducts. So we get the full wave for the sinusoidal signal. Bridge rectifiers are rated by the maximum current they can pass and the maximum reverse voltage they can withstand.

SMOOTHING

Smoothing is performed by a large value electrolytic capacitor connected across the DC supply to act as a reservoir, supplying current to the output when the varying DC voltage from the rectifier is falling. The diagram shows the unsmoothed varying DC (dotted line) and the smoothed DC (solid line). The capacitor charges quickly near the peak of the varying DC, and then discharges as it supplies current to the output. Therefore, circuitry is provided within all UPSes to convert AC power to DC to charge the battery.

2.3.2 BATTERY

The other main component is the battery, which of course holds the energy that is used by the UPS to run your equipment. The size of the UPS is primarily dictated by the size of the battery; the larger the battery, the more time your equipment can run on battery power before shutting down. The batteries used in most UPSes are lead-acid, 12V batteries.

Major advantages of the lead-acid system, in addition to cost are good service life, high-rate discharge capability, and high reliability. The construction of a lead-acid cell uses a positive plate of Lead Dioxide, a negative plate of Spongy Lead, and an electrolyte of Sulfuric Acid. The advent of the sealed lead-acid battery has, with reasonable care in charging, provided significant enhancements to both reliability and ease of use. By using very pure lead in the plates and a special separator which recombines the oxygen created during overcharge, the cell does not dry out and lifetimes of 8-10 years are claimed. As such, these batteries are certainly the optimum choice today for UPS and other backup power systems. Charging a lead acid

battery can be done in a variety of ways but it is important to closely monitor the voltage to prevent extended overcharge. The best chargers are multi-state machines which will initially provide a high current to replace some 80% of the energy as quickly as possible. When a definable over voltage is reached, the charger will hold that voltage for either a programmable time or until the rising battery voltage causes the current to diminish to a low threshold.

2.3.3 INVERTER

A device called an inverter is also provided to change the battery's stored DC electricity to AC to run your equipment. The components used in the inverter are,

- 555 Timers
- 4013 Flip-Flop
- IRF244 MOSFET
- Step up transformer

2.3.3.1 NE555 TIMER

The 555 monolithic timing circuits is a highly stable controller capable of producing accurate time delays or oscillation. In the time delay mode of operation, the time is precisely controlled by one external resistor and capacitor. For a stable operation as an oscillator, the free running frequency and the duty cycle are both accurately controlled with two external resistors and one capacitor. The circuit may be triggered and reset on falling waveforms, and the output structure can source or sink up to 200mA. Functional diagram of NE555 timer is shown in figure 2.4

Description of the Functional Diagram

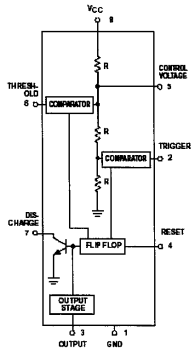


Figure: 2.4 NE 555 Timer Functional Diagram

The internal resistors present act as voltage dividers providing bias voltage of (2/3) Vcc to the upper comparator and (1/3) Vcc to the lower comparator, where Vcc is the supply voltage.

Since these two voltages fix the necessary comparator threshold voltages, they also aid in determining the time interval. It is possible to vary time electronically too, by applying a modulation voltage to the control voltage input terminal (pin 5). In the standby state, the output of the control flip flop is HIGH. This makes the output LOW because of the power amplifier which is basically an inverter. A negative going trigger pulse is applied to pin 2 and should have its DC level greater than the threshold of the lower comparator (i.e. Vcc/3).

At the negative going edge of the trigger, as the trigger passes through (Vcc/3), the output of the lower comparator goes high and sets the flip flop(Q = 1). During the positive excursion, when the threshold voltage at pin 6 passes through (2/3) Vcc, the output of the upper comparator goes high and resets the flip flop (Q =0). The reset input pin 4 provides the mechanism to reset the flip flop in a manner which overrides the effect of any instruction coming to the FF from the lower comparator. This overriding reset is effective when the reset input is less than about 0.4V. When this reset is not used it is returned to Vcc.

Features

- Turn-off time less than 2ms
- Max. Operating frequency greater than 500 kHz
- Timing from microseconds to hours
- Operates in both astable and monostable modes
- High output current and Adjustable duty cycle
- TTL compatible

2.3.3.2 4013 FLIP-FLOP

The CD4013 is the CMOS DUAL 'D' type flip-flop which consists of two identical independent data type flip-flops. Each flip-flop has independent data, set, reset, and clock inputs and Q and its inverse outputs. These devices can be used for shift register applications and, by connecting Q' output to the data input, for counter and toggle applications.

The logical level present at the D input is transferred to the Q output during the positive-going transition of the clock pulse. Setting or resetting is independent of the clock and is accomplished by a high level on the set or reset line. The features are

- Set-Reset capability
- Static Flip-Flop operations – retains state indefinitely with clock level either high or low
- Medium speed operation
- Standardized symmetrical output characteristics

2.3.3.3 IRF244 MOSFET

These are N-Channel enhancement mode silicon gate power field effect transistors. They are advanced power MOSFETs designed, tested, and guaranteed to withstand specified level of energy in the breakdown avalanche mode of operation. All of these power MOSFETs are designed for applications such as switching regulators, switching converters, motor drivers, relay drivers, and drivers for high power bipolar switching transistors requiring high speed and low gate drive power. These types can be operated directly from integrated circuits.

2.3.3.4 STEP UP TRANSFORMER

Step up transformer is defined as a transformer which receives electrical energy at one voltage level and delivers it at a higher voltage level (i.e. the output voltage is more than the input voltage). The transformer is used in power supply stage of television receiver (230/300- 300V, AC).Thus the inverter is provided to change the battery's stored DC electricity to AC to run your equipment.

CHAPTER 3
SENSORS

3.1 INTRODUCTION

A sensor is a device that detects a change in physical stimulus and turns it to a signal which can be measured or recorded. Sensors are used to detect changes in the operation of the system and measure parameters like temperature, voltage, etc.

3.2 VOLTAGE SENSORS

Voltage sensors measure AC and/or DC voltage levels. They receive voltage inputs and provide outputs as analog voltage signals, analog current levels, switches, or audible signals. They can also provide frequency and modulated frequency outputs. The MCT2E are phototransistors optocouplers which has optically coupled isolators consisting of a gallium arsenide infrared emitting diode and an NPN silicon phototransistor mounted in a standard 6-pin dual-in-line package.

The sensor combines an infrared LED light source and a phototransistor light detector into a single package. The LED and the detector point out of the package, almost parallel to each other. A phototransistor is a normal transistor in which the envelope enclosing the junction is transparent to allow light to fall on the base-emitter junction. At any PN junction hole-electron pairs are generated when light falls on the junction, so that any light falling on the base- emitter junction produces a current which is amplified by transistor action, making the device very sensitive.

APPLICATIONS

- Power supply regulators
- Digital logic inputs
- Microprocessor inputs

3.3 TEMPERATURE SENSOR

Thermistors, or thermal resistors, are semiconductor devices that behave as resistors with a high, usually negative, temperature coefficient of resistance. In some cases, the resistance of a thermistor at room temperature may decrease as much as 6 percent for each 1°C rise in temperature. This high sensitivity to temperature change makes the thermistor extremely well suited to precision temperature measurement, control, and compensation. Thermistors are therefore widely used in such applications; especially in the lower temperature range of -100°C to 300°C.Thermistor is widely used in temperature sensor. A thermistor is a type of resistor used to measure temperature changes, relying on the change in its resistance with changing temperature.

If we assume that the relationship between resistance and temperature is linear (i.e. we make a first-order approximation), then we can say that

$R = kT$

Where

R = change in resistance

T = change in temperature

k = first-order temperature coefficient of resistance

Thermistors can be classified into two types depending on the sign of k. If k is positive, the resistance increases with increasing temperature, and the device is called a positive temperature coefficient (PTC) thermistor, posistor or sensistor. If k is negative, the resistance decreases with increasing temperature, and the device is called a negative temperature coefficient (NTC) thermistor. Resistors that are not thermistors are designed to have the smallest possible k, so that their resistance remains almost constant over a wide temperature range.

Short-circuiting a battery by connecting its two terminals directly together with no load, can cause it to overheat and explode. The sensors are placed in the inverter and the signals are sent to the microcontroller.

CHAPTER 4
MICROCONTROLLER (AT89C51)

4.1 INTRODUCTION

The AT89C51, an 8 bit single chip microcontroller has got a powerful CPU optimized for control applications, 64K program memory address space, 64K data memory address space, 128 bytes of on-chip RAM(read/write memory), for 8 bit bi-directional parallel ports one full duplex serial ports, two 16 bit timers/counters and an extensive interrupt structure. There are 32 pins needed by the four 8 bit bi-directional ports. Eight additional pins are providing power, allow you to connect a crystal clock and provide a few timing and control signals. All these devices are connected to the AT89C51 internal 8-bit data bus. Each I/O port is also connected to the 8-bit internal data bus through a series of registers. These registers hold data during I/O transfers and control the I/O ports.

4.2 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
- Five Interrupt Sources
- Programmable Serial Channel
- Low-power Idle and Power-down Modes

4.3 BLOCK DIAGRAM

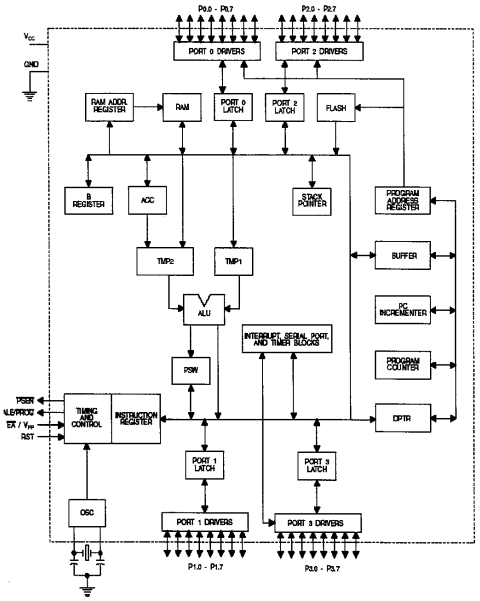


Figure 4.1: 89C51 Architecture

4.4 DESCRIPTION

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4Kbytes 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 pin out. 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.

4.5 OSCILLATOR CHARACTERISTICS

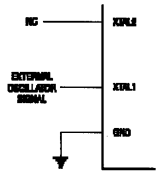


Figure: 4 .2 external clock drive configuration

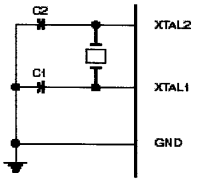


Figure: 4.3 oscillator connections

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 3.3. 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 4.2. 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.

4.6 MEMORY ORGANISATION

4.6.1 FLASH ROM

Flash ROM can be well explained with block diagram as shown in the following diagram. 4-kilo byte ROM is available in the microcontroller. It can be erased and reprogrammed. The Configuration of flash ROM is shown in figure 3.4. If the available memory is not enough for your program, you can interface the external ROM with this IC, it has 16 address lines, so the maximum of (2^16) i.e. 64 bytes of ROM can be interfaced with microcontroller. Both internal and external ROM cannot be used simultaneously.

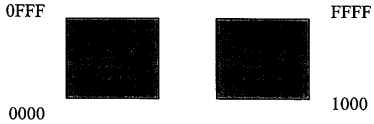


Figure: 4.4 Flash ROM

For external accessing of ROM, a pin is provided in the microcontroller itself is i.e. pin no.31 EA should be high to use internal ROM, and low to use external ROM.

4.6.2 RAM

Internal 256 bytes of RAM are available for the user. These 256 bytes of RAM can be used along with the external RAM. Externally you can connect 64-kilo bytes of RAM with microcontroller. In internal RAM first 128 bytes of RAM is available for the user and the remaining 128 bytes are used as special function registers (SFR). These SFR's are used as control registers for timer, serial ports etc.

4.6.3 PROGRAM MEMORY

The AT89C Microcontroller has separate address spaces for program memory and data memory. The program memory can be up to 64K bytes long. Its configuration is shown in figure 4.5.

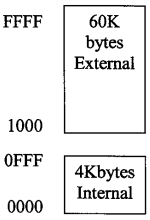


Figure: 4.5 Program Memory

Each interrupt is assigned a fixed location in program memory. The interrupt causes the CPU to jump to that location, where it executes the service routine. External interrupt 0 for example is assigned to location 0003h. If the interrupt is not used, its service location is available as general-purpose program memory.

The interrupt service locations are spaced at 8 byte intervals. 0003h for external interrupt 0, 000Bh for timer 0, and 0013h for external interrupt 1, 001Bh for timer 1 and so on. If an interrupt service routine is short enough (as is often the case in control applications) it can reside entirely within that 8-byte interval. Longer service routines can use a jump instruction to skip over subsequent interrupt locations.

The lower addresses may reside on-chip flash or in external memory. To make this selection, strap the External Access (EA) pin to either Vcc or GND. In AT89C51 with 4K bytes of on-chip Flash, if the EA pin is strapped to Vcc, program fetches to addresses 0000h through 0FFFh are directed to internal Flash. Program fetches to addresses 1000h through FFFFh are directed to external memory

4.6.4 DATA MEMORY

The AT89C can directly address up to 64K bytes of data memory external to the chip. The MOVX instruction accesses the external data memory. The configuration of data memory is shown in figure 3.6. The AT89C51 has 128 bytes of on-chip RAM plus a number of Special Function Registers (SFRs). The lower 128 bytes of RAM can be accessed either by direct addressing (MOV data addr) or by indirect addressing (MOV @Ri).

The internal data memory is divided into three blocks namely,

- The lower 128 bytes of internal RAM
- The upper 128 bytes of internal RAM
- Special Function Registers

Internal Data memory addresses are always 1 byte wide, which implies an address space of only 256 bytes. However, the addressing modes for internal RAM can in fact accommodate 384 bytes. Direct addresses higher than 7Fh access one memory space and indirect addresses higher than 7Fh access a different memory space.

The lowest 32 bytes are grouped into 4 banks of 8 registers. Program instructions call out these registers as R0 through R7. Two bits in the Program Status word (PSW) select, which register bank, are in use. This architecture allows more efficient use of code space, since register instructions are shorter than instructions that use direct addressing.

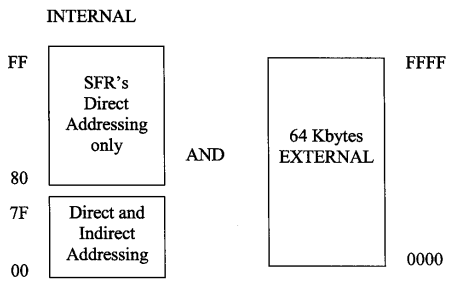


Figure: 4.6 Data Memory

The next 16 bytes above the register banks form a block of bit addressable memory space. The microcontroller instruction set includes a wide selection of single bit instructions and this instruction

can directly address the 128bytes in this area. These bit addresses are 00h through 7Fh.

4.7 ADDRESSING MODES

4.7.1 DIRECT ADDRESSING

In direct addressing, the operand specified by an 8-bit address field in the instruction. Only internal data RAM and SFR's can be directly addressed.

4.7.2 INDIRECT ADDRESSING

In Indirect addressing, the instruction specifies a register that contains the address of the operand. Both internal and external RAM can indirectly address.

The address register for 8-bit addresses can be either the Stack Pointer or R0 or R1 of the selected register Bank. The address register for 16-bit addresses can be only the 16-bit data pointer register, DPTR.

4.7.3 INDEXED ADDRESSING

Program memory can only be accessed via indexed addressing this addressing mode is intended for reading look-up tables in program memory. A 16 bit base register (Either DPTR or the Program Counter) points to the base of the table, and the accumulator is set up with the table entry number. Adding the Accumulator data to the base pointer forms the address of the table entry in program memory.

Another type of indexed addressing is used in the "case jump" instructions. In this case the destination address of a jump instruction is computed as the sum of the base pointer and the Accumulator data.

4.8 POWER MODES OF ATMEL89C51

To exploit the power savings available in CMOS circuitry, Atmel's Flash microcontrollers have two software- invited reduced power modes.

4.8.1 IDLE MODE

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset. In this mode current draw is reduced to about 15 percent of the current drawn when the device is fully active.

4.8.2 POWER DOWN MODE

In the power-down mode, the oscillator is stopped, and the instruction that invokes power-down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power-down mode is terminated. The only exit from power-down is a hardware reset. Reset redefines the SFR's but does not change the on-chip RAM. In this mode, the device typically draws less than 15μA and can be low as 0.6μA.

4.8.3 POWER ON RESET

When power is turned on, the circuit holds the RST pin high for an amount of time that depends on the capacitor value and the rate at which it charges. To ensure a valid reset, the RST pin must be high

long enough to allow the oscillator to start up plus two machine cycles.

4.9 INTERRUPTS

89C51 provides five interrupts. Three of these are generated automatically by internal operations: timer flag 0, timer flag 1 and serial port interrupt (RI or TI). Two interrupts are triggered by external signals provided by circuitry that is connected to pins INT1 and INT0.All interrupt functions are under the control of the program. The programmer is able to alter control bits in the Interrupt Enable register (IE); the Interrupt Priority register (IP) and Timer Control registers (TCON).

Timer Flag Interrupt

When a timer/counter overflows, corresponding Timer flag, TF0 or TF1, is set to 1.The flag is cleared to 0 when the resulting interrupt generates a program call to appropriate timer subroutine in memory.

Serial Port Interrupt

If a data byte is received, an interrupt bit, RI, is set to 1 in the SCON register. When a data byte has been transmitted an interrupt bit, TI, is set in SCON.These are ORed together to provide a single interrupt to the processor: the serial port interrupt. These bits are not cleared when the interrupt-generated program call is made by the processor. The program that handles serial data communication must reset RI or TI to 0 to enable the next data communication operation.

External Interrupts

The external interrupts INT0 and INT1 can each other be either level- activated or transition-activated, depending on bits IT0 and IT1 in Register TCON.The flags that actually generate these interrupts are the IE0 and IE1 bits in TCON.When the service routine is vectored to,

hardware clears the flag that generated an external interrupt only if the interrupt was transition activated.

If the interrupt was level-activated, then the external requesting source (rather than the on-chip hardware) controls the request flag. Each interrupt source causes the program to do a hardware call to one of the dedicated addresses in program memory. The interrupt saves the PC of the program, which is running at the time the interrupt is serviced on the stack in internal RAM.A RETI at the end of the routine restores the PC to its place in the interrupted program and resets the interrupt logic so that another interrupt can be serviced.

4.10 TIMER/COUNTERS

The AT89C51 has two 16-bit Timer/Counter registers: timer 0 and timer 1.All two can be configured to operate either as Timers or event counters. As Timer, the register is incremented every machine cycle. Thus, the register counts machine cycles. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

As a Counter, the register is incremented in response to a 1 to 0 transition at its corresponding external input pin, T0 or T1.the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since 2 machine cycles (24 oscillator periods) are required to recognize a 1-to-0 transition; the maximum count rate is 1/24 of the external input signal, but it should be held for at least one full machine cycle to ensure that a given level is sampled at least once before it changes.

In addition to the Timer or Counter functions, Timer 0 and Timer 1 have four operating modes (13 bit timer, 16 bit timer,8 bit auto-reload, split time.

4.11 INPUT/OUTPUT PORTS

There are four I/O ports available in Atmel89C51. All these ports are eight bit ports. All these ports can be controlled as eight bit ports or it can be individually controlled. All these port lines are available with internal pull-ups expect port0. If we want to use port 0 as I/O port we have to use pull-up resistors.

Port 0 is an 8-bit open-drain bi-directional I/O port. Port 0 may also be configured to be the multiplexed low order address/data bus during accesses to external program and data memory. Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification.

Port1 is an 8 bit bi-directional I/O port with internal pull-ups. Port1 pins have no dual functions. They receive the low-order address bytes during flash programming.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that uses 16-bit addresses. Port 2 also receives the high-order address bits and some control signals during Flash programming.

Port 3 serves the functions of various special features of the AT89C51 as listed in table 4.1.

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)

Table: 4.1 Port 3-Alternate Functions

CHAPTER 5
RS-232 SERIAL INTERFACE

5.1 RS-232 SERIAL INTERFACING

The data from PC (+12v and -12v) and converts into TTL logic then this given to Micro PC communicates with peripherals through serial ports com1 or com2 which communicates the data in terms of pulse form as follows.

Data		Pulse
0	⟶	+12v
1	⟶	-12v

The above logic is considered as PC Logic.

Microcontroller communicates with peripherals through serial pins RXD or TXD, which communicates the data in terms of pulse form as follows.

Data		Pulse
0	⟶	0v
1	⟶	+5v

The above logic is considered as TTL Logic.

RS232 is the interface between PC and microcontroller, which converts PC logic into TTL logic and vice versa.

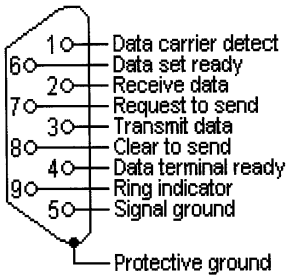


Figure 5.1 RS232 DB9 PINOUT

5.1.1 RS232C

RS232 stands for recommended standard number 232 and C is the latest version of the standard. It can transmit data reliable about 50 feet at its maximum rate of 20,000Bd. The serial ports on most computers use a subset of the RS232C standard.

PIN NO	SIGNAL
1	Carrier Detect (CD) (from DCE) Incoming signal from a modem
2	Received Data (RD) Incoming Data from a DCE
3	Transmitted Data (TD) outgoing Data to a DCE
4	Data Terminal Ready (DTR) Outgoing handshaking signal

5	Signal Ground Common reference voltage
6	Data Set Ready (DSR) Incoming handshaking signal
7	Request To Send (RTS) Outgoing flow control signal
8	Clear To Send (CTS) Incoming flow control signal
9	Ring Indicator (RI) (from DCE) Incoming signal from a modem

TABLE 5.1 PIN DESCRIPTION OF SERIAL PORT FOR RS232

5.1.2 RS232C REPRESENTATIONS

The RS232 signals are represented by voltage levels with respect to a system common power or logic ground. The idle state (mark) has the negative signal level. The active state (space) has the positive level signal. RS232 has numerous handshaking signals (primarily used with MODEMS), and also specifies a communication protocol.

The RS232 interface presupposes a common ground between the DTE and DCE. This is a reasonable assumption when a short cable connects the DTE to the DCE, but with longer lines and connection between the devices that may be on different electrical busses with different ground, this may not be true.

RS232 data is bipolar. +3 to +12 V indicates ON (SPACE) condition while a -3 to -12 V indicates an OFF (MARK) condition. Modern computer equipment ignores the negative level and accepts a zero level as the OFF state. In fact the ON state may be achieved with lesser positive potential.

This means circuits powered by 5 V dc are capable of driving RS232 circuits directly; however the over all range that the RS232 signal may be transmitted or received may be dramatically reduced.

The output signal level usually swings in between +12 V and -12 V. The dead area between +3V to -3V is designed to absorb line noise. In the various RS232 like definitions this dead area may vary. Data is transmitted and received on pin 2 and 3 respectively. Data Set Ready (DSR) is an indication from the Data Set that it is on. Similarly, DTR indicates to the Data Set that the DTE is on. Data Carrier Detect indicates that a good carrier is being received from the remote modem.

Pin4 RTS (Request to send from the transmitting computer) and 5 CTS (Clear to send from the data set) are used to control. In most asynchronous situation, RTS and CTS are constantly on throughout the communication session. However, the DTE is connected to a multipoint line. RTS is used to turn carrier on the modem on and off. On a multi-point line, it's imperative that only one station is transmitting at a time (because they share the return phone pair). When a station wants to transmit, it raises CTS. The DTE transmits when it sees CTS up. When the station has finished its transmission, it drops RTS and the modem drops CTS and carrier together.

5.2 MAX 232

RS232 does not use the conventional 0 and 5v implemented in TTL and CMOS designs. Drivers have to supply +5 to +15v for logic 0 and -5 to -15v for logica1; this is performed by IC MAX232. This means that extra power supplies are needed to drive the RS232 voltage levels. Typically a +12 and a -12v power supply are used to drive the RS232 outputs.

The first level charge pump essentially doubles the standard +5v power supply to provide the voltage level necessary for driving logic0. A second charge pump inverts this voltage and provides the voltage level necessary for driving logic1. These two charge pumps allow the RS232 interface products to operate from a single +5v supply.

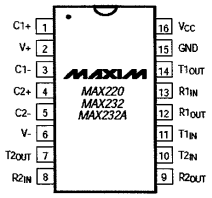


Figure 5.2 MAX232 PIN DIAGRAM

5.2.1 BAUD VS BITS PER SECOND

Baud refers to modulation rate or the number of times per second that a line changes state. This is not always the same as bits per second (BPS).

If two serial devices are connected together using direct cables then baud BPS are in fact same. Thus running at 19200BPS causes the line to change state at 19200 times per second. But when considering modems, this is not the case.

Since modem transfer signals over a telephone line; the baud rate is actually limited to a maximum of 2400 baud. This is a physical restriction of the lines provided by the phone company.

The increased data throughput achieved with 9600 or higher baud modems is accomplished by using sophisticated phase modulation and data compression techniques.

5.2.2 CABLES AND GENDER CHANGES

In a perfect world, all serial ports on every computer would be DTE devices with 25 pin male "D" connectors. All other devices would be DCE devices with 25 pin female connectors. This would allow using a cable in which each pin on one end of the cable is connected to the same pin on the other end. Serial ports use both 9 and 25 pins, many devices can be configured as either DTE or DCE, and as in the case of many data collection devices may use completely non-standard or proprietary pin outs. Due to this lack of standardization special cables called null cables are often required.

5.2.3 NULL MODEM CABLES AND ADAPTERS

If two DTE devices are connected using a straight RS232 cable then the transmit line on each device will be connected to the transmit line on the other device and the receive lines will likewise be connected to each other.

A null modem cable or null modem adapter simply crosses the receiver and transmitter lines so that transmit on one end is connected to receive on the other end and vice versa. In addition to transmit and receive, DTR and DSR, as well as RTS and CTS are also crossed in a Null Modem connection.

5.2.4 FUNCTIONAL CHARACTERISTICS

RS232 has defined the function of the different signals that are used in the interface. These signals are divided into four different categories: Common, data, control and timing. The standard provides an abundance of control signals and supports a primary and secondary communications channel. For example, only eight signals are used for a typical modem. Some simple applications may require only four signals (two for data and two for handshaking) while others may require only data signals with no handshaking.

5.2.5MAXIMUM DATA RATE

Another limitation in the RS232 standard is the maximum data rate. The standard defines a maximum data rate of 20kbps. While providing a communication rate at this frequency, the devices still maintain maximum 30V/ms maximum, slew rate to reduce the likelihood of cross-talk between adjacent signals.

CHAPTER 6
HARDWARE CIRCUITS

This chapter presents detailed description of the hardware circuits which includes power supply, sensors and interfacing circuits.

6.1 POWER SUPPLY

A power supply is a circuit, which converts alternating current in to direct current and provides regulated and unregulated supply. The power supply circuit consists of following blocks.

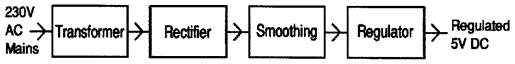


Figure 6.1: Block Diagram of the Power Supply

6.1.1 TRANSFORMER

A step-down transformer, which steps down the 230volt AC supply to 12-0-12V AC supply its current rating, is 1Amp. The transformer is of the core type in order to convert AC RMS voltage in to DC voltage, a rectifier is required.

The ratio of the number of turns on each coil, called the turns ratio, determines the ratio of the voltages. A step-down transformer has a large number of turns on its primary (input) coil which is connected to the high voltage mains supply, and a small number of turns on its secondary (output) coil to give a low output voltage.

Turns ratio = Vp/ Vs = Np/ Ns

Power out = power in (Vs × Is = Vp × Ip)

Vp = primary (input) voltage

Np = number of turns on primary coil

Ip = primary (input) current

Vs = secondary (output) voltage

Ns = number of turns on secondary coil

Is = secondary (output) current

6.1.2 RECTIFIER

There is a full wave bridge rectifier which contains 4 diodes. During the positive half cycle conduct and during the negative half cycle remaining two were conducts. So we get the full wave for the sinusoidal signal. Voltage drop across the diode is 0.6V. The rectified DC voltage consists of some AC components, it should be filtered.

6.1.3 FILTER

The output DC voltage from the rectifier may contain AC components. We can filter or smooth out the AC variation from the rectifier voltage by connecting a shunt capacitor as a filter. A large capacitor shorted with the load resistor causes only a small part of AC to pass through the load, producing a very small voltage.

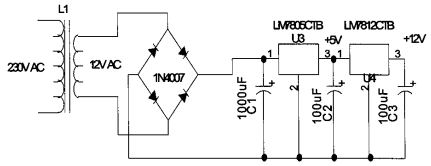


Figure 6.2 Circuit Diagram of the Power Supply

6.1.4 IC VOLTAGE REGULATORS

IC voltage regulators (78XX series) contain circuit for reference source, comparator, amplifier, control device and overload protection all in a single chip. The regulators are selected based on desired output DC voltage and operating load current. The current rating of the IC can be from hundreds of mill amperes to tens of amperes, corresponding to power rating from mill watts to tens of watts.

6.1.5 THREE TERMINAL VOLTAGE REGULATORS

Fig 6.2 shows the basic connection of a three terminal voltage regulator IC to a load. The regulator has an unregulated DC input voltage, Vi, applied to one input terminal, a regulated output voltage, Vo, obtained from third terminal, the second terminal is grounded.

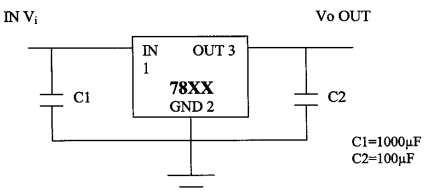


Figure 6.3 IC VOLTAGE REGULATORS

The series 78 regulator provides a fixed regulated voltage from 5V to 24V. The unregulated input voltage Vi, should be filtered by a capacitor C1. The output voltage is filtered by capacitor C2 to avoid high frequency noise. The input terminal may vary to a permissible range and the output voltage can also vary over a certain range, the output voltage remains constant.

6.2 INTERFACING IC - MAX232

6.2.1 MAX232 DESCRIPTION

Since the RS232 is not compatible with today's microprocessor. We need a line driver (voltage converter) to convert the RS232 signals to TTL voltage level that will be acceptable by PIC16F877's TXD and RXD pins. One example of such converter is MAX232 from Maxim Corporation.

The MAX232 converts from RS232 voltage levels to TTL voltage levels and vice versa. One advantage of MAX232 chip is that it uses a +5V powered source which is the same as a source voltage for the PIC16F877, i.e. a single +5V power supply can power both the PIC16F877 and MAX232, with no need for dual power supplies that are common in many older systems.

PART NO.	POWER SUPPLY (V)	NO.OF RS232 DRIVERS/RX	NO. OF EXT. CAPS	NOMINAL CAP VALUE (μF)	DATA RATE (Kbps)	FEATURES
MAX232	+5	2/2	4	1.0	120	Industry standard
MAX232A	+5	2/2	4	0.1	200	Higher slew rate

Table 6.1 MAX232/232A COMPARISON

MAX232 requires four capacitors ranging from 1 to 22 μF. The most widely used values for this capacitor is 22 μF.

The inside typical MAX232 is shown in Figure 6.4.

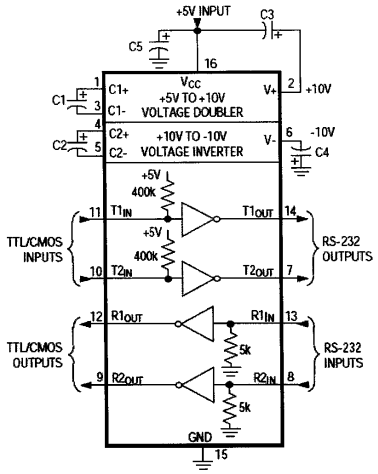


Figure 6.4 MAX232 PINOUT

The MAX232 contains four sections

- Dual charge-pump DC-DC voltage converters
- RS232 drivers
- RS232 receivers
- Receiver and transmitter enable control inputs

6.2.2 INTERFACING CIRCUIT

The MAX232 has two sets of line drives for transferring and receiving data, the line drivers used for TXD are called T1 and T2, while the line drivers for RXD are designated as R1 and R2.

The T1 line driver has a designation of T1in and T1out on the pin numbers 11 and 14 respectively. The T1in pin is the TTL side and is connected to the TXD of the microcontroller, while T1out is the RS232 side that is connected to the RXD pin of the RS232 DB connector.

The R1 line driver has a designation of R1in and R1out on the pin numbers 13 and 12 respectively. The R1in is the RS232 side and is connected to the TXD pin of RS232 DB connector, R1out is the TTL side that is connected to the RXD pin of the microcontroller.

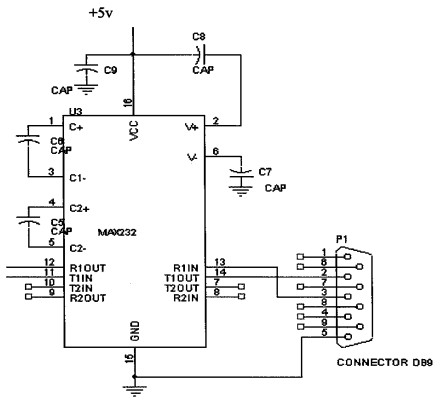


Figure 6.5 MAX232 INTERFACING TO RS232DB9 CONNECTOR

6.5 MODES OF COMMUNICATION

There are two modes of communication parallel mode and serial mode. In short distance communication parallel mode is the general choice. In long distance, serial is the suitable mode.

In parallel mode, the interconnection cable is a bus with separate lines for all the bits of the bytes, lines for handshaking signals, lines for error checking signals such as parity. Thus the interconnection cable is expensive.

In serial mode, the interconnection cable is a bus with two lines for receiving and transmitting the bit, lines for handshaking signals, and lines for error checking signals.

Two types of serial transfer are in existence, asynchronous and synchronous transmission. The asynchronous format is character oriented.

Each character carries the information of the start and stop bits. This is also known as framing. This format is generally used in high speed transmission.

Serial communication also can be classified according to the direction and simultaneity.

In simplex transmission data are transmitted in only one direction. In duplex transmission occurs in both directions. However if the transmission goes way at a time it is called half duplex.

The rate at which the bits are transmitted/seconds is called baud rate in serial transmission.

6.6 COMPUTER PORTS

This is an introduction to program the PC's communication ports. The basic way by which the computer communicates with the devices is through ports. On PC's it is basically an 8-bit doorway. The computer communicates with other devices either through serial ports or through parallel ports.

Each method has its own merits and demerits. A parallel port transfers entire byte at a time whereas the serial port transfers only one bit at a time. This is the basic difference between the ports. These serial ports take eight-fold time that of parallel port to transfer a byte. This makes us to come to a conclusion that parallel transfer is much faster than serial transfer. In spite of this we have selected serial port as the medium of transfer for this project.

CHAPTER 7
SOFTWARE DESCRIPTION

After receiving the signal i.e., the fault information from the serial port, the following functions are performed:

- Generating an e-mail automatically and sending it to the manufacturer system
- Generating an voice alert informing the fault occurrence
- Storing the fault information inside an database
- Displaying the list of faults stored in the database along with the date and time of occurrence

These functions are performed using Visual Studio .NET platform. The features of the .NET platform are discussed for the better understandings.

7.1 OVERVIEW OF THE .NET PLATFORM

.NET is Microsoft's platform for the next generation of software that connects our world of information, devices and a people in a unified, personalized way. The .NET platform enables the creation and use of various applications, processes and services that share and combine information and functionality with each other by design, on any platform or smart devices, to provide tailored solutions for organizations and individual people. .NET is a comprehensive family of products, built on industry and internet standards, that provide for each aspect of developing(tools), managing(servers), using(building block services and smart clients) and experiencing XML web services. .NET will become part of the Microsoft applications, tools, and servers you already use today- as well as new

products that extend XML web services capabilities to all of your business needs.

7.2 WHAT IS MICROSOFT .NET?

Microsoft is creating an advanced new generation of software that melds computing and communications in a revolutionary new way, offering every developer the tools they need to transform the web and every other aspect of the computing experience. This initiative is called Microsoft .NET, and for the first time it enables developers, businesses and consumers the ability to harness technology on their terms. Microsoft .NET will allow the creation of truly distributed web services that will integrate and collaborate a range of complementary services to serve customers. Microsoft .NET will drive the next generation internet. It will make information available any time, any place and any devices.

7.3 MICROSOFT .NET PROGRAMMING MODEL

The Microsoft .NET programming model gives developers the opportunity to focus fewer resources on where or how an application runs and focus more resources on what the application does- where they can add real value. Microsoft .NET addresses some of the biggest challenges facing developers, particularly the trade off between functionality and manageability. In addition, developers will be able to leverage and customize a range of core Microsoft .NET building block services in their own applications and services, reducing the effort required to create compelling products.

Visual studio .NET represents the complete development environment for building applications on the Microsoft .NET platform. With visual studio .NET, developers can build web

applications and XML web services that render in any browser and on virtually any devices. By using the language of their choice, developers can leverage their existing investments in skills and systems. The result is increased productivity, end-to-end web development and a short time delivery.

7.4 THE IMPROVING TECHNOLOGY

From a technology standpoint, .NET is the platform and the .net experiences built on the top of the platform. The platform includes:

- Tools - to built applications and XML web devices (.NET Framework and visual studio .NET)
- Servers - on which to built, host and deploy those applications and services (Windows 2000 Server and the .NET enterprise servers)
- Services – a core set of .NET building block services.
- Client Software – the software that powers smart devices allowing users to interact and experience the .NET platform.
- Experiences – the combination of the above components of the .NET platform allow for more personal, integrated user experiences.

Microsoft .NET supports high levels of integration with existing infrastructures an applications. It provides new technologies and evolves existing ones, enabling you to draw upon the existing experience of your staff. It also uses your existing hardware and supports scaling up and scaling out as your needs demand.

A key promise of Microsoft .NET is integrating applications. As a result managing integration is critical to its success. By enabling

customers to better monitor and control their computing resources, Microsoft is providing the basis for managing distributed computing as an integral part of all windows hosted environments.

Microsoft .NET addresses most of today's computing deficiencies to realize the vision of enabling access to all user's data and applications anywhere and from any device; allowing users to interact with their data through handwriting, speech and vision technologies.

Microsoft's vision for this new generation of distributing computing on the Internet is one where software is delivered as a service, accessible by any device, any time, any place and is fully programmable and personalized. To enable this vision, Microsoft is delivering the .NET platform and .NET experiences, built on public Internet standard and protocols, with tools and services that integrate computing and communications in productive new ways.

The Microsoft .NET platform is explicitly designed to enable the rapid development and integration of any group of XML Web services and applications into a single solution.

7.5 MICROSOFT .NET FRAMEWORK ARCHITECTURE

The core components of the .NET framework are described below.

- The Common Language Runtime (CLR): The CLR is the execution engine for .NET framework applications. It provides a number of services including the loading and execution of code, application memory isolation, memory management, exception handling, access to metadata (enhanced type information), and the conversion of MSIL (Microsoft Intermediate Language) to native platform code.

- The Base Class Library (BCL): The BCL provides an extensive set of classes, logically grouped into hierarchical namespaces that provide access to the underlying features of the operating system.
- ADO.NET: this is an evolutionary update to the ActiveX Data Objects (ADO) data access technology, with significant improvements aimed at the disconnected nature of the web.
- ASP.NET: This is an advanced version of Active Server Pages (ASP) for Web Application development (using web forms) and web service development.
- The Common Language Specification (CLS): This is responsible for making many of the aforementioned technologies available to all language that support the.NET framework. The CLS itself is not a technology, and there is no source code to it. It defines a set of rules providing a contract that governs the interoperability between language compilers and libraries.
- Win Forms: This programming model and control set provide a robust architecture for windows based application development.
- Visual Studio .NET: This delivers the tools allowing you to exploit the framework's features to create concrete applications.

7.6 FEATURES OF THE .NET FRAMEWORK

COMMON LANGUAGE RUNTIME

The Common Language Runtime (CLR) plays a role in both the development and execution of .NET applications. At runtime, the CLR is responsible for managing the execution environment of .NET (managed code), including memory allocation and thread management, as well as enforcing security policies.

MULTILANGUAGE AND EXTENSIBLE

The critical role of the runtime in the .NET framework (and what really sets it apart) is that it provides a unified environment across all programming languages. The base classes of the .NET framework provide a unified, object-oriented, and hierarchical, extensible set of class libraries for developers that we need to work with today.

CROSS LANGUAGE INTEROPERABILITY

Among the many strengths of .NET is language transparency. Whichever managed language we choose, we have equal access to the platform. The .NET framework enables cross language inheritance, error handling, and debugging. As a developer, we can take advantage of the .NET platform using your preferred language.

MULTI-PLATFORM DESIGN

.NET represents a new way of developing software for windows and potentially other platforms. .NET is not inextricably tied to the Windows operating system. For example, the intermediate code (MSIL) generated by .NET language compilers, is not processor specific and the BCL classes are designed to work on any operating system.

7.7 ADVANTAGES OF USING .NET FRAMEWORK

Here are some of the advantages provided by the .NET frameworks are

- Consistent Programming model. All application services are accessed through a common object oriented programming model,

unlike today where some operating system facilities are accessed via DLL functions and other facilities are accessed via COM objects
- Run Once, Run Always. All developers are familiar with DLL Hell. Since installing components for a new application can overwrite components of an old application, the old application can exhibit strange behavior or stop functioning altogether. The .NET architecture now separates application components so that an application always loads the components with which it was built and tested. If the application runs after installation, then the application should always run.
- Execute on many platforms. Once written and compiled to MSIL, a managed .NET application can execute on any platform that supports the .NET CLR. You immediately appreciate the value of this broad execution model when you need to support multiple computing hardware configuration or operating systems.
- Language Integration. While COM allows different programming languages to interoperate with one another, .NET allows languages to be integrated with one another. For example, it is possible to create a class in C# that derives from a class implemented in Visual Basic .NET. .NET can enable this because it defines and provides a Common Type System (CTS) common to all .NET languages. The Microsoft Common Language Specification (CLS) describes what compilers vendors must do in order for their languages to integrate with other languages.
- Code Reuse. Using mechanisms just described, you can create your own classes that offer services to third-party applications. This is of course, makes it extremely simple to reuse code and broadens the market for component vendors.

- Automatic resource management through garbage collection. Programming requires great skill and discipline. This is especially true when it comes to managing resources such as files, memory, network connections, database resources, and so on. One of the most common types of bug occurs when an application neglects to free one of these resources, causing that application or another to fail. The .NET CLR automatically tracks resources usage, guaranteeing that your application never leaks resources. This is called garbage collection. There is no way to explicitly free memory resources, as it is automatically garbage collected when it is no longer required.
- Type Safety. The .NET CLR can verify that all our code is type safe. Type safety ensures that allocated objects are always accessed in compatible ways. Hence, if a method input parameter is declared as accepting a 4-byte value, the CLR detects and traps attempts to pass an 8-byte value using this parameter. Type safety also means that execution flow will only transfer to well known locations. There is no way to construct an arbitrary reference to a memory location and cause code at that location to begin execution. Type safety and a lack of direct pointer manipulation form the basis of the security model, and they also eliminate many common programming errors and classic system attacks such as exploitations of buffer overruns.
- Deployment. Today, Windows-based applications can be incredibly difficult to install and deploy. There are usually multiple files, registry setting, and short cuts that need to be created. In addition, completely uninstalling an application is nearly impossible. With Windows 2000, the Microsoft introduced the Windows installer, a new installation engine that helps with these issues, but it is still possible that a company authoring a new installer package may fail to do everything correctly. .NET seeks to completely overcome these

issues. .NET components are not referenced in the registry and installing most .NET-based application requires no more than copying the files to the directory. Uninstalling a .NET application is as easy as deleting those files. Also, the CLR allows side-by-side deployments where, for example, you can have two versions of the same applications running at the same time on the same machine while you migrate with the users.

7.8 NAMESPACES IN .NET

The base class library (BCL) is included in .NET as a base framework assembly that contains many class definitions, where each class exposes a feature of the underlying platform. Since Microsoft defines hundreds of classes in the BCL, the library is divided into namespaces that logically group related classes together. For example, the system namespace contains the base class, object, from which all other classes ultimately derive. In addition, the system namespace contains classes for exception handling, garbage collection, console I/O, as well as a variety of utility classes that perform type safe conversions, format data types, generate random numbers, and perform various math functions.

AUTOMATIC MEMORY MANAGEMENT

The .NET framework advances the features of COM+ resulting in much higher developer productivity. For example, the .NET framework introduces automatic memory management, regardless of the programming language. It also provides you with a single, unified programming framework that brings together the best features of the Windows Foundation Classes, the Visual Basic APIs, and augments them with other capabilities including advanced data access architecture.

The .NET framework also uses the same underlying component services that COM+ provides today, with features such as transactions, queuing, and object pooling. In feature versions, it will provide a feature called partitioning. This provides stronger process isolation and is designed for application service providers. COM+ is the highest-performance, most feature-rich set of application services available today.

7.9 VISUAL BASIC .NET

Based on the .NET type system, Visual Basic .NET offers a number of significant improvements over previous versions of Visual Basic. The main ones relate to its support for the object oriented notions of polymorphisms, inheritance, and method and operator overloading. Others come from its common foundation with other languages provided by the CLS.

We can freely pass data types to and from components developed in other languages and inherit from base classes developed in other languages. Visual Basic .NET supports ADO.NET for data access, the windows Forms programming model, multithreading and new improved wizards and designers.

Existing Visual Basic developers will benefit from learning the new language constructs and features, allowing them to develop more powerful and robust applications. However, we don't have to upgrade all of our code straight away, as Visual Basic 6 code and Visual Basic .NET code will happily run side by side. If we do choose to upgrade, COM interoperability services will greatly help with a phased transition, allowing you seamlessly use your existing components. Visual Basic .NET also provides an upgrade wizard, which steps you

through the upgrade process and performs the necessary language syntax changes when a visual basic 6 project is loaded.

The primary new features introduced by Visual Basic .NET include

- Inheritance: Through implementation inheritance, classes can extend (or inherit from) an existing base class. Visual Basic .NET (in common with all .NET languages) only supports single inheritance, meaning that we can only inherit from a single base class. The new class can obtain the functionality and behavior where appropriate. A derived class can also choose to provide new behaviors, of example by implementing new methods. To use implementation inheritance keyword to express which base class the new one derives from. You then use the overrides keyword to modify the base class functionality where appropriate. Note that a method that is capable of being overridden in a derived class must be marked as overridable in the base class.
- Polymorphism: This is the ability to treat objects of different class types in the same way. Traditionally, we could achieve this in Visual Basic using interfaces. While Visual Basic .NET still supports polymorphism through interfaces, it can now also be achieved using inheritance.
- Constructors: Constructors allow an object to be created and initialized to an initial state in a single operation. This alleviates the need to create an object, and then call a separate method to initialize it.

In Visual Basic .NET, object constructors allow you to pass values to an object as it is being created. This simplifies coding and helps reduce errors when working with objects.

- Initialzers: Initializers allow you to declare a variable and provide it with an initial value in a single line of code. This leads to smaller, simpler, more maintainable code.
- Structured Error Handling: The .NET CLR uses exceptions to handle error conditions in a consistent way, irrespective of the programming language you use. Exceptions are responses to failure conditions that can occur while an application is running. They can either be generated by the .NET run time or can be created programmatically. Exceptions can be propagated across language boundaries. For example, a component written in C# could generate an exception which could subsequently be caught and handled by a client program developed with Visual Basic .NET.

7.10 CONCLUSION

With .NET, the internet more fully becomes a platform for business. Microsoft .NET addresses most of today's computing deficiencies to realize the visions of enabling access to all user's data and applications anywhere and from any device; allowing users to interact with their data through handwriting, speech, and vision technologies. The Microsoft .NET platform is explicitly designed to enable the rapid development and integration.

CHAPTER 8

CONCLUSION

The complete system is more reliable than the existing manual methods in the industrial process. The integration of hardware and software provide efficient operation of the system. Moreover the developed system is quite simple and user friendly. The system has been working satisfactorily with the designed values and the system will fulfill the industrial requirement.

Also, with further enhancements more and more applications could be embedded in our model by changing few of its present components, so that it would provide support for the new applications. When such enhancements are done, our project could be used as model to build a robot in real life.

CHAPTER 9

APPENDICES

9.1 89C51 MICROCONTROLLER DATA SHEET

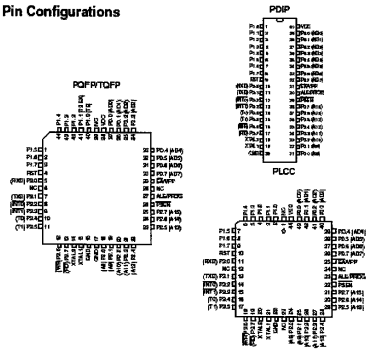
Features

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

Description

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

Pin Configurations



8-bit Microcontroller with 4K Bytes Flash

AT89C51

Rev. 0265G–02/98

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.

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	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (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 ($\overline{PROG}$) during Flash programming.

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

pulse is skipped during each access to external Data Memory.

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

$\overline{PSEN}$

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

When the AT89C51 is executing code from external program memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to external data memory.

EA/VPP

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

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

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

XTAL1

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

XTAL2

Output from the inverting oscillator amplifier.

Oscillator Characteristics

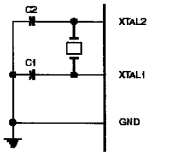
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected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

Idle Mode

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

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

Figure 1. Oscillator Connections

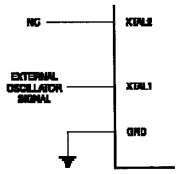


Note: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Status of External Pins During Idle and Power-down Modes

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

Figure 2. External Clock Drive Configuration



Power-down Mode

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

Program Memory Lock Bits

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

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

Lock Bit Protection Modes

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

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) = FFH	(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 $\overline{EA}/V_{PP}$ to 12V for the high-voltage programming mode.
5. Pulse ALE/$\overline{PROG}$ 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.

Data Polling: The AT89C51 features $\overline{Data}$ 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 P0.7. Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. $\overline{Data}$ Polling may begin any time after a write cycle has been initiated.

Ready/$\overline{Busy}$: The progress of byte programming can also be monitored by the RDY/$\overline{BSY}$ 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/$\overline{PROG}$ 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.

Flash Programming Modes

Mode		RST	PSEN	ALE/PROG	EA/V_{PP}	P2.6	P2.7	P3.6	P3.7
Write Code Data		H	L		H/12V	L	H	H	H
Read Code Data		H	L	H	H	L	L	H	H
Write Lock	Bit - 1	H	L		H/12V	H	H	H	H
	Bit - 2	H	L		H/12V	H	H	L	L
	Bit - 3	H	L		H/12V	H	L	H	L
Chip Erase		H	L	(1)	H/12V	H	L	L	L
Read Signature Byte		H	L	H	H	L	L	L	L

Note: 1. Chip Erase requires a 10 ms PROG pulse.

Figure 3. Programming the Flash

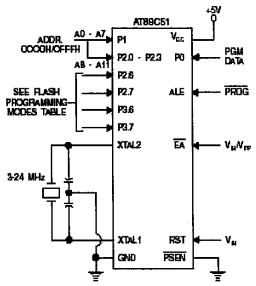
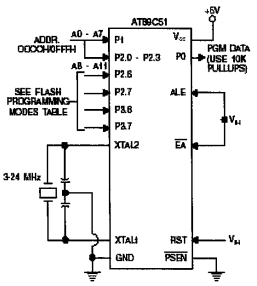
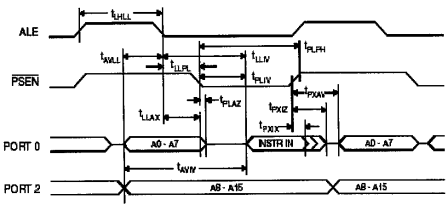


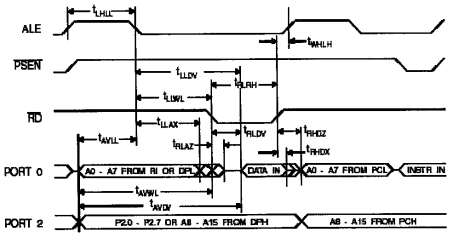
Figure 4. Verifying the Flash



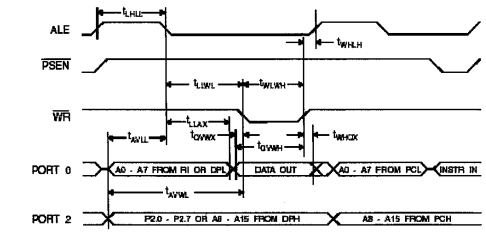
External Program Memory Read Cycle



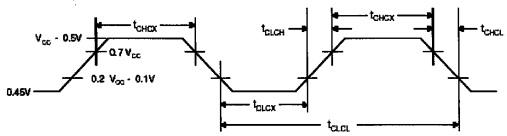
External Data Memory Read Cycle



External Data Memory Write Cycle



External Clock Drive Waveforms



External Clock Drive

Symbol	Parameter	Min	Max	Units
1/t	Oscillator Frequency	0	24	MHz
t	Clock Period	41.6		ns
t	High Time	15		ns
t	Low Time	15		ns
t	Rise Time		20	ns
t	Fall Time		20	ns

19-4323; Rev 9; 5/00

MAXIM

+5V-Powered, Multichannel RS-232 Drivers/Receivers

General Description

The MAX220–MAX249 family of line drivers/receivers is intended for all EIA/TIA-232E and V.28/V.24 communications interfaces, particularly applications where ±12V is not available.

These parts are especially useful in battery-powered systems, since their low-power shutdown mode reduces power dissipation to less than 5µW. The MAX225, MAX233, MAX235, and MAX245/MAX246/MAX247 use no external components and are recommended for applications where printed circuit board space is critical.

Applications

Portable Computers
Low-Power Modems
Interface Translation
Battery-Powered RS-232 Systems
Multidrop RS-232 Networks

Features

Superior to Bipolar

- Operate from Single +5V Power Supply (+5V and +12V—MAX231/MAX239)
- Low-Power Receive Mode in Shutdown (MAX223/MAX242)
- Meet All EIA/TIA-232E and V.28 Specifications
- Multiple Drivers and Receivers
- 3-State Driver and Receiver Outputs
- Open-Line Detection (MAX243)

Ordering Information

PART	TEMP. RANGE	PIN-PACKAGE
MAX220CPE	0°C to +70°C	16 Plastic DIP
MAX220CSE	0°C to +70°C	16 Narrow SO
MAX220CWE	0°C to +70°C	16 Wide SO
MAX220C/D	0°C to +70°C	Dice*
MAX220EPE	-40°C to +85°C	16 Plastic DIP
MAX220ESE	-40°C to +85°C	16 Narrow SO
MAX220EWE	-40°C to +85°C	16 Wide SO
MAX220EJE	-40°C to +85°C	16 CERDIP
MAX220MJE	-55°C to +125°C	16 CERDIP

*Ordering Information continued at end of data sheet.
*Contact factory for dice specifications.

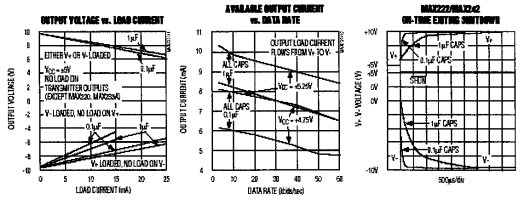
Selection Table

Part Number	Power Supply (V)	No. of RS-232 Drivers/Rx	No. of Ext. Caps	Nominal Cap. Value (µF)	SHDN & Three-State	Rx Active in SHDN	Data Rate (kbps)	Features
MAX220	+5	2/2	4	0.1	No	—	120	Ultra-low-power, industry-standard pinout
MAX222	+5	2/2	4	0.1	Yes	—	200	Low-power shutdown
MAX223 (MAX213)	+5	4/5	4	1.0 (0.1)	Yes	✔	120	MAX241 and receivers active in shutdown
MAX225	+5	5/5	0	—	Yes	✔	120	Available in SO
MAX230 (MAX200)	+5	5/0	4	1.0 (0.1)	Yes	—	120	5 drivers with shutdown
MAX231 (MAX201)	+5 and +7.5 to +13.2	2/2	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; same functions as MAX232
MAX232 (MAX202)	+5	2/2	4	1.0 (0.1)	No	—	120 (64)	Industry standard
MAX232A	+5	2/2	4	0.1	No	—	200	Higher slew rate, small caps
MAX233 (MAX208)	+5	2/2	0	—	No	—	120	No external caps
MAX233A	+5	2/2	0	—	No	—	200	No external caps, high slew rate
MAX234 (MAX204)	+5	4/0	4	1.0 (0.1)	No	—	120	Replaces 1488
MAX235 (MAX208)	+5	5/5	0	—	Yes	—	120	No external caps
MAX236 (MAX206)	+5	4/3	4	1.0 (0.1)	Yes	—	120	Shutdown, three-state
MAX237 (MAX207)	+5	5/3	4	1.0 (0.1)	No	—	120	Complements IBM PC serial port
MAX238 (MAX208)	+5	4/4	4	1.0 (0.1)	No	—	120	Replaces 1488 and 1489
MAX239 (MAX206)	+5 and +7.5 to +13.2	3/5	4	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; single-package solution for IBM PC serial port
MAX240	+5	5/3	4	1.0	Yes	—	120	DIP or flatpack package
MAX241 (MAX211)	+5	4/5	4	1.0 (0.1)	Yes	—	120	Complete IBM PC serial port
MAX242	+5	2/2	4	0.1	Yes	✔	200	Separate shutdown and enable
MAX243	+5	2/2	4	0.1	No	✔	200	Open-line detection simplifies cabling
MAX244	+5	8/10	4	1.0	No	—	120	High slew rate
MAX245	+5	8/10	0	—	Yes	✔	120	High slew rate, int. caps, two shutdown modes
MAX246	+5	8/10	0	—	Yes	✔	120	High slew rate, int. caps, three shutdown modes
MAX247	+5	8/9	0	—	Yes	✔	120	High slew rate, int. caps, nine operating modes
MAX248	+5	8/8	4	1.0	Yes	✔	120	High slew rate, selective half-chip enables
MAX249	+5	8/10	4	1.0	Yes	✔	120	Available in quad flatpack package

+5V-Powered, Multichannel RS-232 Drivers/Receivers

Typical Operating Characteristics

MAX220/MAX222/MAX232A/MAX233A/MAX242/MAX243



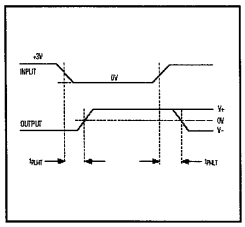


Figure 1. Transmitter Propagation-Delay Timing

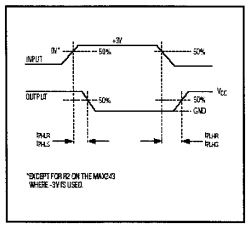


Figure 2. Receiver Propagation-Delay Timing

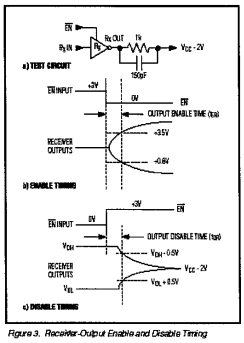


Figure 3. Receiver-Output Enable and Disable Timing

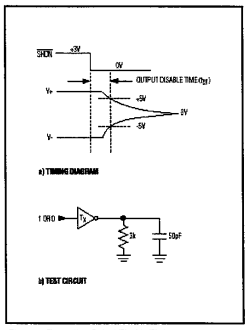


Figure 4. Transmitter-Output Disable Timing

Table 1a. MAX245 Control Pin Configurations

$\overline{ENT}$	$\overline{ENR}$	OPERATION STATUS	TRANSMITTERS	RECEIVERS
0	0	Normal Operation	All Active	All Active
0	1	Normal Operation	All Active	All 3-State
1	0	Shutdown	All 3-State	All Low-Power Receive Mode
1	1	Shutdown	All 3-State	All 3-State

Table 1b. MAX245 Control Pin Configurations

$\overline{ENT}$	$\overline{ENR}$	OPERATION STATUS	TRANSMITTERS		RECEIVERS	
			TA1–TA4	TB1–TB4	RA1–RA5	RB1–RB5
0	0	Normal Operation	All Active	All Active	All Active	All Active
0	1	Normal Operation	All Active	All Active	RA1–RA4 3-State, RA5 Active	RB1–RB4 3-State, RB5 Active
1	0	Shutdown	All 3-State	All 3-State	All Low-Power Receive Mode	All Low-Power Receive Mode
1	1	Shutdown	All 3-State	All 3-State	RA1–RA4 3-State, RA5 Low-Power Receive Mode	RB1–RB4 3-State, RB5 Low-Power Receive Mode

Table 1c. MAX246 Control Pin Configurations

$\overline{ENA}$	$\overline{ENB}$	OPERATION STATUS	TRANSMITTERS		RECEIVERS	
			TA1–TA4	TB1–TB4	RA1–RA5	RB1–RB5
0	0	Normal Operation	All Active	All Active	All Active	All Active
0	1	Normal Operation	All Active	All 3-State	All Active	RB1–RB4 3-State, RB5 Active
1	0	Shutdown	All 3-State	All Active	RA1–RA4 3-State, RA5 Active	All Active
1	1	Shutdown	All 3-State	All 3-State	RA1–RA4 3-State, RA5 Low-Power Receive Mode	RB1–RB4 3-State, RA5 Low-Power Receive Mode

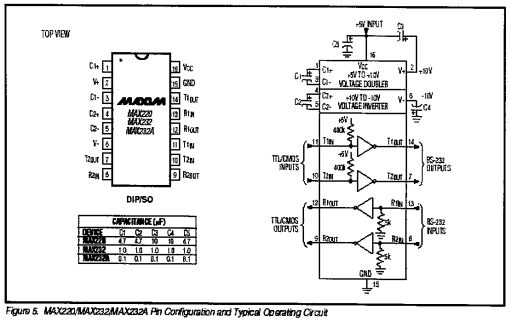


Figure 5. MAX220/MAX232/MAX232A Pin Configuration and Typical Operating Circuit

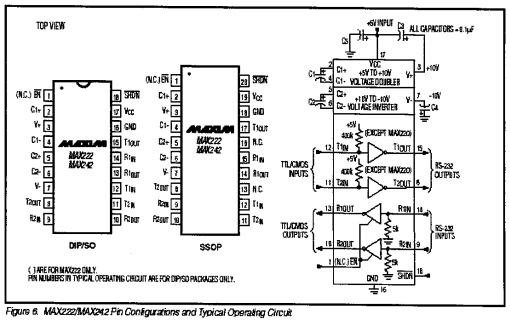


Figure 6. MAX222/MAX242 Pin Configurations and Typical Operating Circuit

9.3 NE555 TIMER DATA SHEET

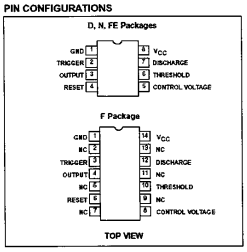
DESCRIPTION

The 555 monolithic timing circuit is a highly stable controller capable of producing accurate time delays, or oscillation. In the time delay mode of operation, the time is precisely controlled by one external resistor and capacitor. For a stable operation as an oscillator, the free running frequency and the duty cycle are both accurately controlled with two external resistors and one capacitor. The circuit may be triggered and reset on falling waveforms, and the output structure can source or sink up to 200mA.

FEATURES

- Turn-off time less than 2µs
- Max. operating frequency greater than 500kHz
- Timing from microseconds to hours
- Operates in both astable and monostable modes
- High output current
- Adjustable duty cycle
- TTL compatible
- Temperature stability of 0.005% per °C

PIN CONFIGURATIONS



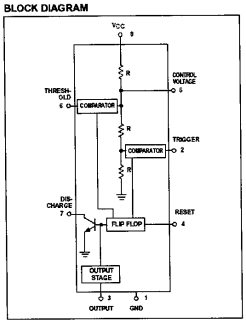
APPLICATIONS

- Precision timing
- Pulse generation
- Sequential timing
- Time delay generation
- Pulse width modulation

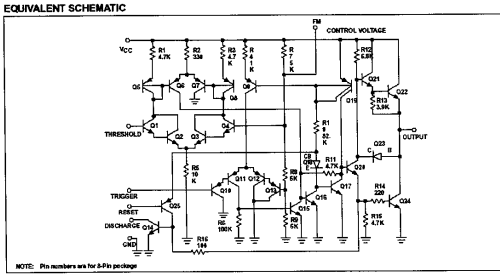
ORDERING INFORMATION

DESCRIPTION	TEMPERATURE RANGE	ORDER CODE	DWG #
8-Pin Plastic Small Outline (SO) Package	0 to +70°C	NE555D	0174C
8-Pin Plastic Dual In-Line Package (DIP)	0 to +70°C	NE555N	0404B
8-Pin Plastic Dual In-Line Package (DIP)	-40°C to +85°C	SA555N	0404B
8-Pin Plastic Small Outline (SO) Package	-40°C to +85°C	SA555D	0174C
8-Pin Hermetic Ceramic Dual In-Line Package (CERDIP)	-55°C to +125°C	SE555CFE	
8-Pin Plastic Dual In-Line Package (DIP)	-55°C to +125°C	SE555CN	0404B
14-Pin Plastic Dual In-Line Package (DIP)	-55°C to +125°C	SE555N	0405B
8-Pin Hermetic Cerdip	-55°C to +125°C	SE555FE	
14-Pin Ceramic Dual In-Line Package (CERDIP)	0 to +70°C	NE555F	0581B
14-Pin Ceramic Dual In-Line Package (CERDIP)	-55°C to +125°C	SE555F	0581B
14-Pin Ceramic Dual In-Line Package (CERDIP)	-55°C to +125°C	SE555CF	0581B

BLOCK DIAGRAM



EQUIVALENT SCHEMATIC



NOTE: Pin numbers are for 8-Pin package

ABSOLUTE MAXIMUM RATINGS

SYMBOL	PARAMETER	RATING	UNIT
V_{CC}	Supply voltage		
	SE555	+18	V
	NE555, SE555C, SA555	+16	V
P_D	Maximum allowable power dissipation[1]	600	mW
T_A	Operating ambient temperature range		
	NE555	0 to +70	°C
	SA555	-40 to +85	°C
	SE555, SE555C	-55 to +125	°C
T_{STG}	Storage temperature range	-65 to +150	°C
T_{SOLD}	Lead soldering temperature (10sec max)	+300	°C

NOTES:
1. The junction temperature must be kept below 125°C for the D package and below 150°C for the FE, N and F packages. At ambient temperatures above 25°C, where this limit would be derated by the following factors:
   D package 160°C/W
   FE package 150°C/W
   N package 100°C/W
   F package 105°C/W

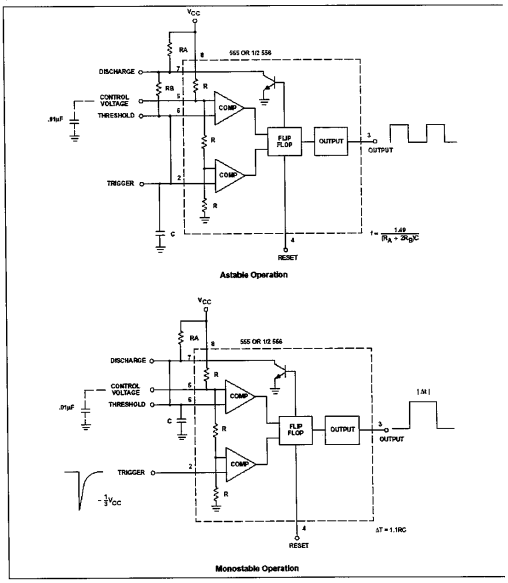
DC AND AC ELECTRICAL CHARACTERISTICS

T_A = 25°C, V_{CC} = +5V to +15 unless otherwise specified.

SYMBOL	PARAMETER	TEST CONDITIONS	SE555			NE555/SE555C			UNIT
			Min	Typ	Max	Min	Typ	Max	
V_{CC}	Supply voltage		4.5		18	4.5		16	V
I_{CC}	Supply current (low state)[1]	V_{CC}=5V, R_L=∞		3	5		3	6	mA
		V_{CC}=15V, R_L=∞		10	12		10	15	mA
	Timing error (monostable)	R_A=2kΩ to 100kΩ							
t_A	Initial accuracy[2]	C=0.1µF		0.5	2.0		1.0	3.0	%
$\Delta t_A/\Delta T$	Drift with temperature			30	100		50	150	ppm/°C
$\Delta t_A/\Delta V_S$	Drift with supply voltage			0.05	0.2		0.1	0.5	%/V
	Timing error (astable)	R_A, R_B=1kΩ to 100kΩ							
t_A	Initial accuracy[2]	C=0.1µF		4	6		5	13	%
$\Delta t_A/\Delta T$	Drift with temperature	V_{CC}=15V			500			500	ppm/°C
$\Delta t_A/\Delta V_S$	Drift with supply voltage			0.15	0.6		0.3	1	%/V
V_C	Control voltage level	V_{CC}=15V	9.6	10.0	10.4	9.0	10.0	11.0	V
		V_{CC}=5V	2.9	3.33	3.8	2.6	3.33	4.0	V
V_{TH}	Threshold voltage	V_{CC}=15V	9.4	10.0	10.6	8.8	10.0	11.2	V
		V_{CC}=5V	2.7	3.33	4.0	2.4	3.33	4.2	V
I_{TH}	Threshold current[3]			0.1	0.25		0.1	0.25	µA
V_{TRIG}	Trigger voltage	V_{CC}=15V	4.8	5.0	5.2	4.5	5.0	5.6	V
		V_{CC}=5V	1.45	1.67	1.9	1.1	1.67	2.2	V
I_{TRIG}	Trigger current	V_{TRIG}=0V		0.5	0.9		0.5	2.0	µA
V_{RESET}	Reset voltage[4]	V_{CC}=15V, V_{TH}=10.5V	0.3		1.0	0.3		1.0	V
I_{RESET}	Reset current	V_{RESET}=0.4V		0.1	0.4		0.1	0.4	mA
	Reset current	V_{RESET}=0V		0.4	1.0		0.4	1.5	mA
V_{OL}	Output voltage (low)	V_{CC}=15V							
		I_{SINK}=10mA		0.1	0.15		0.1	0.25	V
		I_{SINK}=50mA		0.4	0.5		0.4	0.75	V
		I_{SINK}=100mA		2.0	2.2		2.0	2.5	V
		I_{SINK}=200mA		2.5			2.5		V
		V_{CC}=5V							
		I_{SINK}=8mA		0.1	0.25		0.3	0.4	V
		I_{SINK}=5mA		0.05	0.2		0.25	0.35	V
V_{OH}	Output voltage (high)	V_{CC}=15V							
		I_{SOURCE}=200mA		12.5			12.5		V
		I_{SOURCE}=100mA	13.0	13.3		12.75	13.3		V
		V_{CC}=5V							
		I_{SOURCE}=100mA	3.0	3.3		2.75	3.3		V
t_{OFF}	Turn-off time[5]	V_{RESET}=V_{CC}		0.5	2.0		0.5	2.0	µs
t_R	Rise time of output			100	200		100	300	ns
t_F	Fall time of output			100	200		100	300	ns
	Discharge leakage current			20	100		20	100	nA

NOTES:
1. Supply current when output high typically 1mA less.
2. Tested at V_{CC}=5V and V_{CC}=15V.
3. This will determine the max value of R_A+R_B, for 15V operation, the max total R=10MΩ, and for 5V operation, the max. total R=3.4MΩ.
4. Specified with trigger input high.
5. Time measured from a positive going input pulse from 0 to 0.8×V_{CC} into the threshold to the drop from high to low of the output. Trigger is tied to threshold.

TYPICAL APPLICATIONS



TYPICAL APPLICATIONS

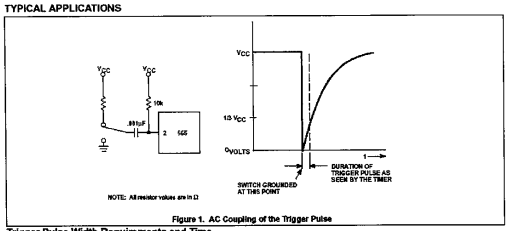


Figure 1. AC Coupling of the Trigger Pulse

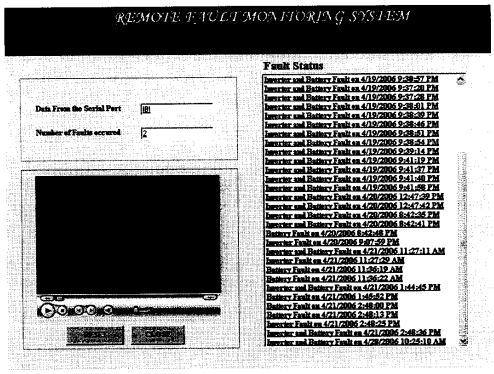
Trigger Pulse Width Requirements and Time Delays

Due to the nature of the trigger circuitry, the timer will trigger on the negative going edge of the input pulse. For the device to time out properly, it is necessary that the trigger voltage level be returned to some voltage greater than one third of the supply before the time out period. This can be achieved by making either the trigger pulse sufficiently short or by AC coupling into the trigger. By AC coupling the trigger, see Figure 1, a short negative going pulse is achieved when the trigger signal goes to ground. AC coupling is most frequently used in conjunction with a switch or a signal that goes to ground which initiates the timing cycle. Should the trigger be held low, without AC coupling, for a longer duration than the timing cycle the output will remain in a high state for the duration of the low trigger signal, without regard to the threshold comparator state. This is due to the predominance of Q_{15} on the base of Q_{16}, controlling the state of the bi-stable flip-flop. When the trigger signal then returns to a high level, the output will fall immediately. Thus, the output signal will follow the trigger signal in this case.

Another consideration is the "turn-off time". This is the measurement of the amount of time required after the threshold reaches 2/3 V_{CC} to turn the output low. To explain further, Q_1 at the threshold input turns on after reaching 2/3 V_{CC}, which then turns on Q_9, which turns on Q_8. Current from Q_8 turns on Q_{18} which turns Q_{17} off. This allows current from Q_{19} to turn on Q_{20} and Q_{24} to given an output low. These steps cause the 2µs max. delay as stated in the data sheet.

Also, a delay comparable to the turn-off time is the trigger release time. When the trigger is low, Q_{20} is on and turns on Q_{21} which turns on Q_{15}. Q_{15} turns off Q_{16} and allows Q_{17} to turn on. This turns off current to Q_{20} and Q_{24}, which results in output high. When the trigger is released, Q_{20} and Q_{21} shut off, Q_{15} turns off, Q_{16} turns on and the circuit then follows the same path and time delay explained as "turn off time". This trigger release time is very important in designing the trigger pulse width so as not to interfere with the output signal as explained previously.

9.4 SAMPLE SCREEN



9.5 SAMPLE CODE

```
Imports System.Data.OleDb
Public Class Form1
Inherits System.Windows.Forms.Form
#Region " Windows Form Designer generated code "
 Public Sub New()
 MyBase.New()
    InitializeComponent()
 End Sub

Dim m As New SocketMail("127.0.0.1", 25)
Dim Once As Boolean
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load
      AxMSComm1.PortOpen = True
End Sub

Private Sub AxMSComm1_OnComm(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
AxMSComm1.OnComm
      TextBox1.Text = AxMSComm1.Input()
      Dim ConnectionString As String
      Dim strsubject, strbody
      Dim dbcon As OleDbConnection
      Dim dbcom As OleDbCommand
        ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;data
        source =c:\faultdb.mdb"
       If Once = False Then
       m.To = "udhayaaffable@yahoo.com"
       m.From = "sugan_pan@yahoo.co.in"

         If TextBox1.Text = "BBB" Then
           TextBox2.Text = 1
           With m
             m.Subject = "Battery Fault"
             m.Message = "low quality, no voltage"
             AxWindowsMediaPlayer1.URL = "c:\battery.wav"
             Once = True
           End With
         End If
```



```
If TextBox1.Text = "III" Then
   TextBox2.Text = 1
   With m
     m.Subject = "Inverter Fault"
     m.Message = "No voltage, step up transformer not
      working, oscillator failed,driver IC not working,"
     AxWindowsMediaPlayer1.URL = "c:\inverter.wav"
     Once = True
   End With
End If
If TextBox1.Text = "HHH" Then
   TextBox2.Text = 1
   With m
     m.Subject = "Over Heat"
     m.Message = "excess heat due to over load"
     AxWindowsMediaPlayer1.URL = "c:\overheat.wav"
     Once = True
   End With
End If
If TextBox1.Text = "IIH" Or TextBox1.Text = "IHI" Or
 TextBox1.Text = "HII" Then
   TextBox2.Text = 2
   With m
     m.Subject = "Inverter and Over Heat Fault"
     m.Message = "No voltage, step up transformer not
 working, oscillator failed,driver IC not working,no voltage"
      AxWindowsMediaPlayer1.URL = "c:\HI.wav"
      Once = True
   End With
End If
End If
 If TextBox1.Text = "HIB" Or TextBox1.Text = "IBH" Or
 TextBox1.Text = "BIH" Then
   TextBox2.Text = 3
   With m
     m.Subject = "Inverter, Battery and Over Heat Fault"
           m.Message = "step up transformer not working,
 oscillator failed,driver IC not working,no voltage"
      AxWindowsMediaPlayer1.URL = "c:\IBH.wav"
      Once = True
   End With
End If
```



```
      If TextBox1.Text = "HHB" Or TextBox1.Text = "HBH" Or
       TextBox1.Text = "BHH" Then
         TextBox2.Text = 2
         With m
           m.Subject = "Inverter, Battery and Over Heat Fault"
           m.Message = "step up transformer not working, oscillator
            failed,driver IC not working,no voltage"
           AxWindowsMediaPlayer1.URL = "c:\HB.wav"
           Once = True
         End With
       End If

       m.Open()
       m.Send()
       m.Close()
       Dim strsql As String
       strsql = "insert into tblerr(errmsg) values('" & m.Subject & "')"
       dbcom = New OleDbCommand(strsql, dbcon)
       dbcom.ExecuteNonQuery()
  End If
  End Sub


 Private Sub Form1_Leave(ByVal sender As Object, ByVal e As
System.EventArgs) Handles MyBase.Leave
      AxMSComm1.PortOpen = False
End Sub
End Class
```

CHAPTER 10
REFERENCES

1. David.G, "The future of computer-telecommunications integration", IEEE transaction on communication, Vol.2, NO.3, pp.10-15, April 1996.
2. Kenneth J. Ayala (1996) 'The 8051 Microcontroller Architecture, Programming and Applications', Penram International, pp.54-86
3. Jan Axelson, "Serial Port Complete", 4th Edition, Penram International Publications, 1999.
4. Jeffrey R. Shapiro," The Complete Reference, Visual Basic .NET", fifth reprint, Tata McGraw-Hill Edition 2002
5. Stephen Walther, " ASP.NET Unleashed", Techmedia, First Indian Reprint 2002
6. www.microchip.com
7. www.pic.com
8. www.Discovercircuits.com
9. www.google.com
10. www.rs232.com