P-3394

AUTOMATION OF PETROL PUMP USING RFID TECHNOLOGY

by

S.DHIVYA	0710107024
M.GOWTHAMY	0710107035
J.B.MUTHUSENBAGAM	0710107058
K.SRI BATHRAVARDHINI	0710107307

of

KUMARAGURU COLLEGE OF TECHNOLOGY,

COIMBATORE - 641 049.

(An Autonomous Institution affiliated to Anna University of Technology, Coimbatore)

A PROJECT REPORT

Submitted to the

FACULTY OF ELECTRONICS AND COMMUNICATION ENGINEERING

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

of

BACHELOR OF ENGINEERING
in
ELECTRONICS AND COMMUNICATION ENGINEERING

APRIL 2011

AUTONOMOUS INSTITUTION AFFILIATED TO ANNA UNIVERSITY OF TECHNOLOGY, COIMBATORE.

BONAFIDE CERTIFICATE

Certified that this project report **"AUTOMATION OF PETROL PUMP USING RFID TECHNOLOGY"** is the bonafide work of "S.DHIVYA, M.GOWTHAMY, J.B.MUTHUSENBAGAM, K.SRI BATHRAVARDHINI" who carried out the project work under my supervision.

SIGNATURE
Dr.Mrs.RAJESHWARI MARIAPPAN

HEAD OF THE DEPARTMENT

Department of Electronics and
Communication Engineering,
Kumaraguru College of Technology,
Coimbatore- 641049.

SIGNATURE
Ms.R.DHIVYA PRABA

SUPERVISOR
ASSISTANT PROFESSOR
Department of Electronics and
Communication Engineering,
Kumaraguru College of Technology,
Coimbatore- 641049.

The candidates with university register numbers 0710107024, 0710107035, 0710107058, 0710107307 were examined by us in the project viva voce examination held on

19.04.2011

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

Owing deeply to the supreme, we extend our sincere thanks to GOD almighty who has made all things possible.

I express my profound gratitude to our beloved Director **Dr.J.Shanmugam,** Kumaraguru College of Technology for his kind support and necessary facilities to carry out the work.

We are grateful to **Prof.Dr.S.Ramachandran,** Principal, Kumaraguru College of Technology for the facilities provided to complete this project work.

We wish to extend our gratitude to **Dr. Rajeswari Mariappan**, Head of the Department of Electronics and Communication Engineering, for her constant support and encouragement.

It is our duty to thank our project coordinator **Mrs.V.Jeyasri Arokiamary,** Associate Professor, Department of Electronics & Communication and our project guide **MS. R.Dhivya Praba**, Assitant Professor, Department of Electronics & Communication for their helpful guidance and valuable support given to us throughout this project.

Our thanks also to all the **Teaching** and **Non-Teaching Staffs** of the department of Electronics and Communication Engineering for providing us the technical support for our project.

We also thank our **Friends** and **Family** who helped us to complete this project fruitfully.

ABSTRACT

Today in this digitalized world, if the fuel indicator in the automobiles is also made digital it will help us to know the percentage amount of fuel available in the fuel tank. By using this digital fuel level indicator the amount of fuel available in the tank at all positions can be predicted.

We have also planned to use RFID payment which enables easy transaction. The petrol pump is equipped with a RFID reader and vehicle unit is equipped with RFID card. Once the fuel is filled the microcontroller sends the level information to the RFID card. The microcontroller acts as the heart of the circuit controlling the operation of both petrol pump unit and vehicle unit.The RFID reader reads the amount of petrol filled, from the RFID card. Then the corresponding amount is calculated & deducted from the card.

The bill is generated according to the amount of fuel filled in tank of the car via RS232 serial communication through the printer. Thus we can avoid theft, and it is time saving job and we don't want to carry money with us as it has an easy transaction through RFID payment.

TABLE OF CONTENTS

APPENDIX 1: RS 232 DATASHEET

APPENDIX 2: MICROCONTROLLER 89S51 DATASHEET

APPENDIX 3: ANALOG TO DIGITAL CONVERTER DATASHEET

LIST OF TABLES

LIST OF FIGURES

CHAPTER - 1

1.INTRODUCTION

1.1 RADIO FREQUENCY IDENTIFICATION SYSTEM

The Radio frequency identification (RFID) technology is a type of non-contact automatic identification technology realized through radio frequency communication. The RFID tag features small volume, great capacity, and long term of service life, repeatable use, and supports fast reading, positioning and long-term tracking management. RFID technology can be used together with Internet and communication technologies, realize article tracking and information sharing within global scope.

The RFID can substantially improve the management and operation efficiency when used in industries of logistics, manufacture and public information service. With constant improvement and upgrading of the relevant technologies, the RFID technology will become an emerging hi-tech industry group as well as new growth point of the national economy. Therefore, the study on RFID technology and development of RFID industry can have far-reaching impact in upgrading the social information level, promoting sustainable development of economy, improving the quality of people's life and enchancing public security and national defense security, and thus of strategic and important significance.

1.2 HOW DOES AN RFID SYSTEM WORKS?

An RFID system consists of a tag, which is made up of a microchip with an antenna, and an interrogator or reader with an antenna. The reader sends out electromagnetic waves. The tag antenna is tuned to receive these waves. A passive RFID tag draws power from field created by the reader and uses it to power the microchip's circuits. The chip then modulates the waves that the tag sends back to the reader and the reader converts the new waves into digital data.

RFID is not necessarily "better" than bar codes. The two are different technologies and have different applications, which sometimes overlap.

The big difference between the two is bar codes are line-of-sight technology. That is, a scanner has to "see" the bar code to read it, which means people usually have to orient the bar code towards a scanner for it to be read. Radio frequency identification, by contrast, doesn't require line of sight. RFID tags can be read as long as they are within range of a reader. Bar codes have other shortcomings as well. If a label is ripped, soiled or falls off, there is no way to scan the item. And standard bar codes identify only the manufacturer and product, not the unique item. The bar code on one milk carton is the same as every other, making it impossible to identify which one might pass its expiration date first.

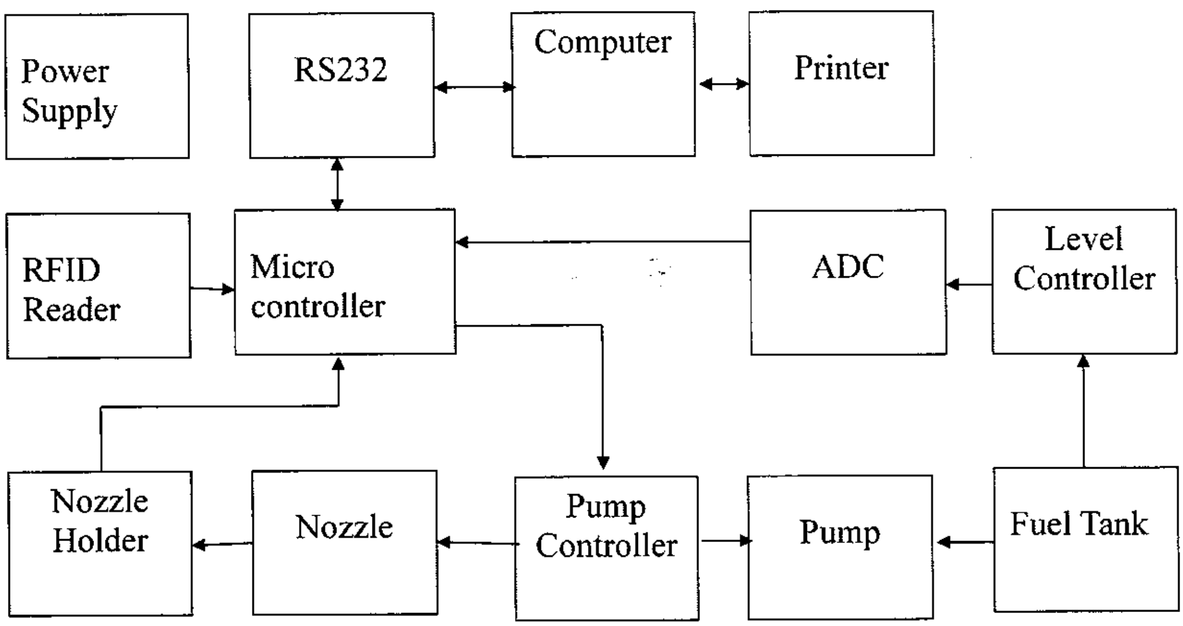
Active RFID tags have a battery, which is used to run the microchip's circuitry and to broadcast a signal to a reader (the way a cell phone transmits signals to a base station). Passive tags have no battery. Instead, they draw power from the reader, which sends out electromagnetic waves that induce a current in the tag's antenna. Semi-passive tags use a battery to run the chip's circuitry, but communicate by drawing power from the reader. Active and semi-passive tags are useful for tracking high-value goods that need to be scanned over long ranges, such as railway cars on a track, but they cost a dollar or more, making them too expensive to put on low-cost items. Companies are focusing on passive UHF tags, which cost under a 50 cents today in volumes of 1 million tags or more. Their read range isn't as far typically less than 20 feet vs. 100 feet or more for active tags but they are far less expensive than active tags and can be disposed of with the product packaging.

CHAPTER - 2

2. HARDWARE DETAILS

2.1 BLOCK DIAGRAM

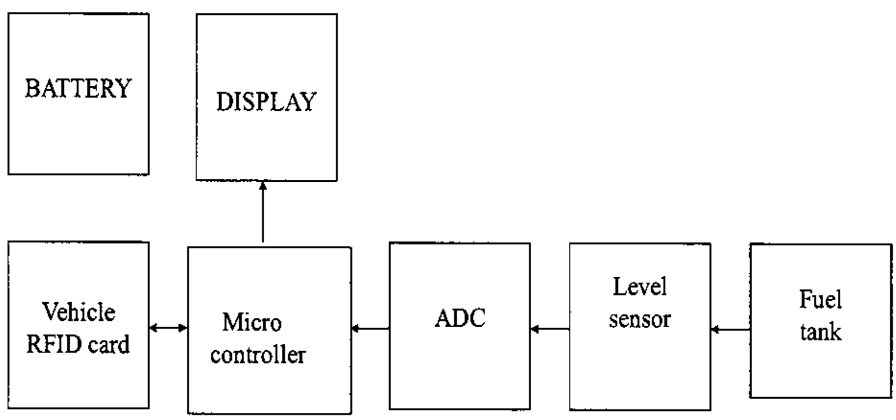
PETROL PUMP SECTION



DESCRIPTION

The required amount of petrol is entered in the computer which is given to the microcontroller through RS-232 serial interface. Now the microcontroller activates the pump and starts filling the petrol in the fuel tank Once the level is reached, the level controller sends the digital level of fuel information which deactivates the pump respectively.The RFID reader read the amount of petrol present the fuel tank from the RFID card of the vehicle section. The amount is deducted and bill will be generated automatically

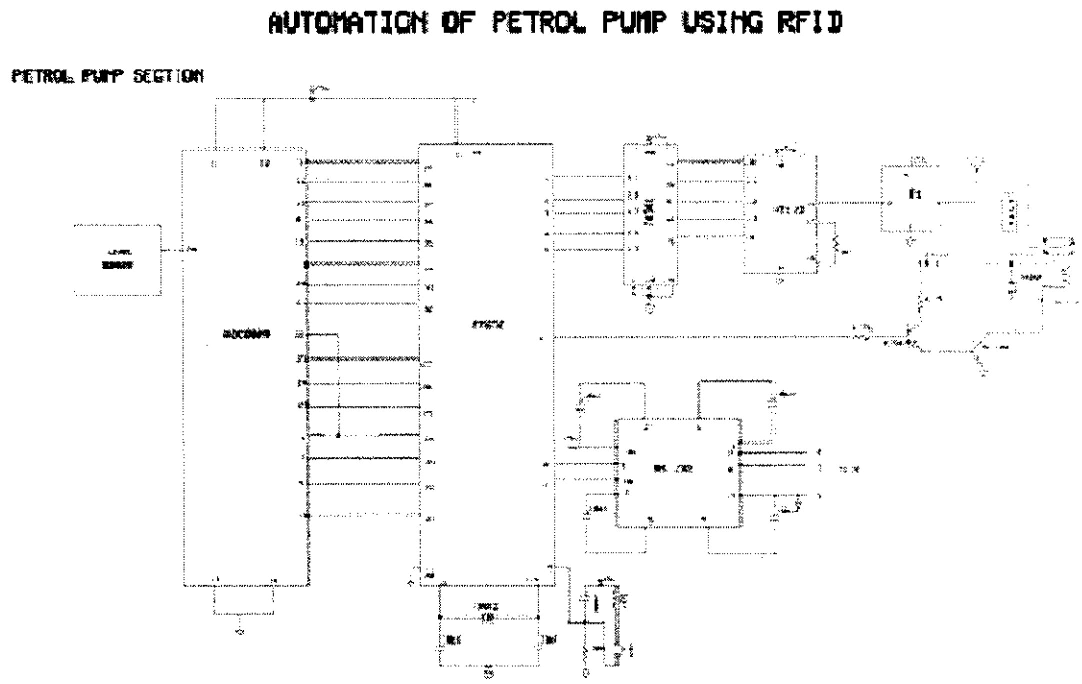
- VEHICLE UNIT

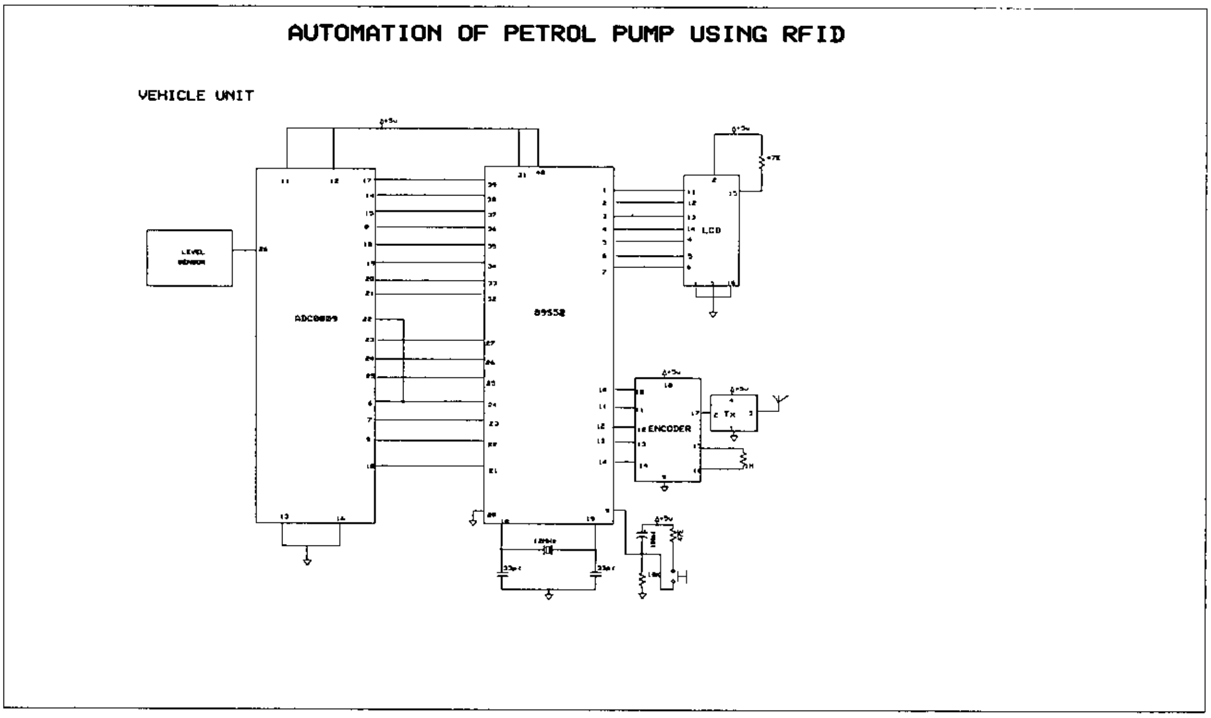


DESCRIPTION

The level sensor senses the amount of petrol in the fuel tank which is converted digitally with the help of ADC and it is given to the microcontroller. The microcontroller stores the fuel level information in the RFID card and also digitally displayed with the help of LCD.

2.2 CIRCUIT DIAGRAM




AUTOMATION OF PETROL PUMP USING RFID
VEHICLE UNIT
ADC0809
89S52
LCD
ENCODER
Tx

2.3.POWER SUPPLY

2.3.1 INTRODUCTION

Since our circuits work only with low D.C. voltage we need a power supply unit to provide the appropriate voltage supply.The required DC supply is obtained from the available AC supply after rectification, filtration and regulation.This unit consists of transformer, rectifier, filter and regulator. We require both the 5V and 12V DC supply where the former is used in microcontroller, LCD, Analog to digital converter, encoder and decoder and the latter is used in relay respectively.

2.3.2CIRCUIT DIAGRAM:

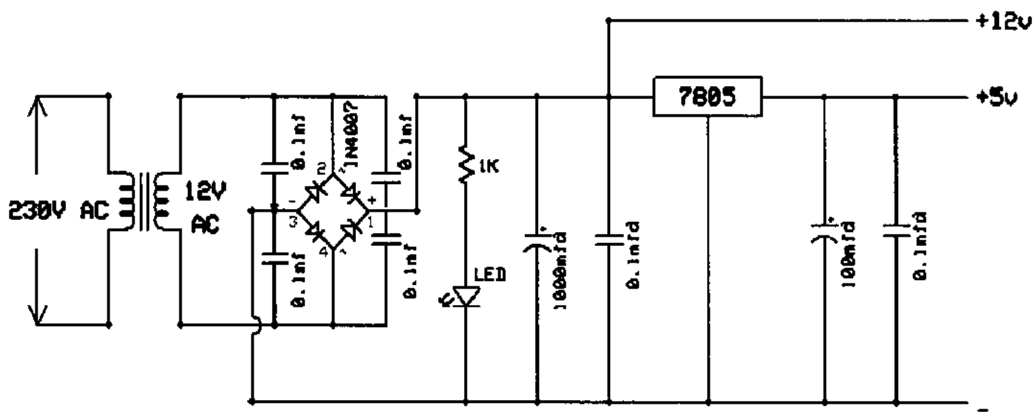


Figure 2.3.2 POWER SUPPLY CIRCUIT DIAGRAM

The main components used in the power supply unit are Transformer, Rectifier, Filter, and Regulator. The 230V ac supply is converted into 12V ac supply through the transformer. The output of the transformer has the same frequency as in the input ac power. This ac power is converted into dc power through the diodes. Here the bridge diode is used to convert the ac supply to the dc power supply.

This converted dc power supply has the ripple content and for the normal operation of the circuit, the ripple content of the dc power supply should be as low as possible. Because the ripple content of the power supply will reduce the life of circuit.

So to reduce the ripple content of the dc power supply, the filter is used. The filter is nothing but the large value capacitance. The output waveform of the filter capacitance will almost be the straight line.

This filtered output will not be the regulated voltage. For the normal operation of the circuit it should have the regulated output. Specifically for the microcontroller IC regulated constant 5V output voltage should be given.

For this purpose 78xx regulator should be used in the circuit. In that number of IC, the 8 represents the positive voltage and if it is 9, it will represent the negative voltage. The xx represents the voltage. If it is 7805, it represent 5V regulator, and if it is 7812, it represent 12V regulator. Thus the regulated constant output can be obtained.

2.4 RFID

RFID Tag is equipped in the vehicle section which can be read by the RFID reader present in the petrol pump section. RFID transmitter transmits the fuel level information to the RFID receiver based on which the amount is deducted and the bill will be generated. The Radio frequency used is 433 MHz which is a readymade module.

2.4.1 RFID READER

An RFID system consists of a reader and one or more tags. The reader's antenna is used to transmit radio frequency. The RFID reader module is controlled with a single TTL-level active-low/ ENABLE pin. When the ENABLE is in pulled LOW, the module will enter its active stage and enable the antenna to interrogate for tags. The current construction of the module will increase dramatically when the module is active. Visual indication of the state of the RFID reader module is given with the on-board LED. When the module is successfully powered-up and is in an ideal state, the LED will be GREEN. When the module is in an active state and the antenna is transmitting, the LED will be RED.

2.4.2 RFID TAGS

Tag is an memory device, usually EPROM, programmed with a series of bits. Its shapes vary from beads, nails, labels that can be incorporated and even special inks on, for eg: paper.

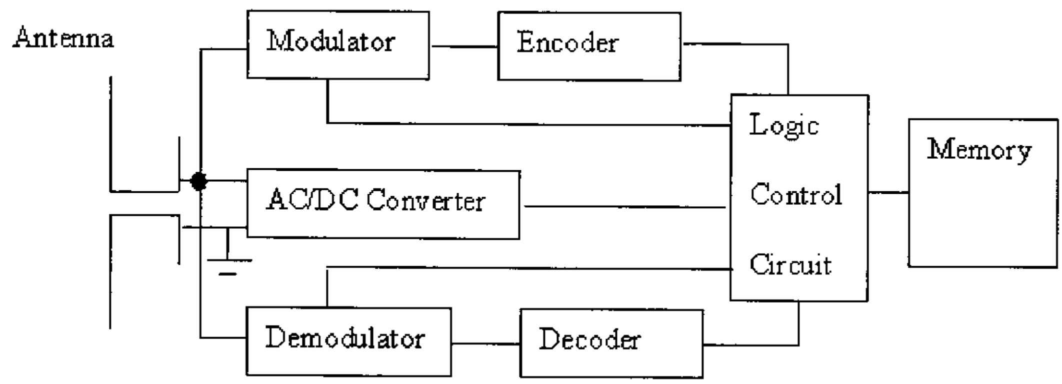


Figure 2.4.2 OPERATION OF RFID TAG

The operation of the RFID tag is described below:

Handshaking with the Reader (interrogator):

1. The reader continuously emits RF carrier signals, and keeps observing the received RF signals for data.
2. The presence of a tag (for our discussion, we consider only passive tag) modulates the RF field, and the same is detected by the reader.
3. The passive tag absorbs a small portion of the energy emitted by the reader, and starts sending modulated information when sufficient energy is acquired from the RF field generated by the reader. Note that the data modulation (modulation for 0s and 1s) is accomplished by either direct modulation or FSK or Phase modulation.
4. The reader demodulates the signals received from the tag antenna, and decodes the same for further processing.

2.4.3 RFID STANDARDS

Standards are critical in RFID. Be it payment systems or tracking goods in open supply chains. A great deal of work has been going on to develop standards for different RFID frequencies and applications.

EPC standards for tags are the class 0 and class 1 tags:

Class 1: a simple, passive, read-only backscatter tag with one-time, field-programmable non-volatile memory.

Class 0: read-only tag that was programmed at the time the microchip was made.

2.4.4 RFID FREQUENCIES

There are several frequencies that are used for RFID. These include LF, HF, UHF, and Microwave frequencies. The exact frequencies may vary depending on the country where it is used.

Table 2.4.4 RFID FREQUENCIES

Frequency Range	Description	Typical Applications
<135KHz	Low Frequency, Inductive coupling	Access Control & Security Widgets identification through manufacturing processes Ranch animal identification OEM applications
13.56 MHz	High Frequency, Inductive coupling	Access Control Library books Laundry identification
868 to 870 MHz 902 to 928 MHz	Ultra High Frequencies (UHF), Backscatter coupling	Supply chain tracking

The seven parts are:

1. 18000–1: Generic parameters for air interfaces for globally accepted frequencies
2. 18000–2: Air interface for 135 KHz
3. 18000–3: Air interface for 13.56 MHz
4. 18000–4: Air interface for 2.45 GHz
5. 18000–5: Air interface for 5.8 GHz
6. 18000–6: Air interface for 860 MHz to 930 MHz
7. 18000–7: Air interface at 433.92 MHz

2.4.5 ADVANTAGES OF RFID:

The Advantages of RFID vs. Barcode technology:

- No line of sight requirement.
- The tag can stand a harsh environment.
- Long read range.
- Portable database
- Multiple tag read/write.
- Tracking people, items, and equipment in realtime.

REAL TIME APPLICATIONS OF RFID:

a. Employee Identification and Access Control.
b. Airline baggage Identification.
c. Wafer Identification during manufacturing process.
d. Livestock Identification.
e. Parts Identification.
f. Identification and Tracking of Vehicles.
g. Identification of widgets through manufacturing process.
h. Supply Chain Automation.

2.5 SERIAL INTERFACE – RS 232:

2.5.1 INTRODUCTION:

RS232 is the most widely used serial I/O interfacing standard. In our circuit we have interfaced microcontroller and the personal computer using RS232 which is used to provide serial communication.Since the RS232 standard was set long before the advent of the TTL logic family, its input and output voltage levels are not TTL compatible.

we need a line driver (voltage convertor) such as MAX232 to convert the RS232's signals to the TTL voltage levels. One advantage of the MAX232 chip is that it uses a +5 V power source which is the same as the source voltage for the microcontroller.

VOLTAGE LEVELS IN RS 232:

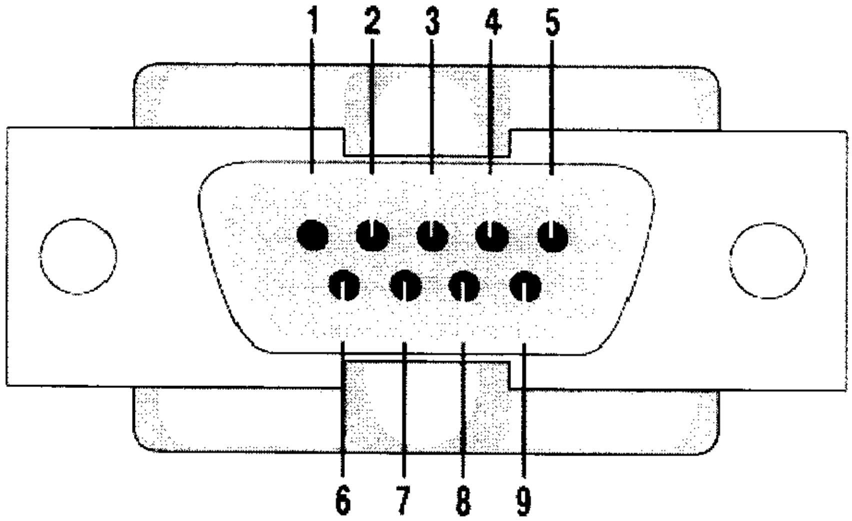
1. Logic high (1) represented as -3 to -25
2. Logic low (0) represented as +3 to +25
3. -3 to +3V not defined

2.5.2 RS 232 INTERFACE:

1. RS 232 was introduced in 1960, and is currently the most widely used communication protocol. It is simple, inexpensive to implement, and though relatively slow it is more than adequate for most simple serial communication devices such as keyboards and mice.
2. 232 is a single- ended data transmission system, which means that it uses a single wire for data transmission.
3. Since useful communication is generally two way, a two wire system is employed, one to transmit and one to receive.

2.5.3 RS 232 PINS:

The table provides the pins and the labels for RS 232 cable, commonly referred to as DB-25 connector.



Pin	Signal	Pin	Signal
1	Data Carrier Detect	6	Data Set Ready
2	Received Data	7	Request to Send
3	Transmitted Data	8	Clear to Send
4	Data Terminal Ready	9	Ring Indicator
5	Signal Ground		

Figure 2.5.3 RS 232 PIN DIAGRAM

The standards for RS 232 and similar interfaces usually restrict RS 232 to 20Kbps or less and line length of 15m or less. RS 232 is fair more robust than the traditional limits of 20kbps over a 15m line would imply RS 232 as perfectly adequate at speed upto 200kbps.

FEATURES OF RS 232:

The essential features of RS 232 is the signals are carried as signal voltages referred to a common earth and print data to a common earth pin, data is transmitted and received on pins 2 &3 respectively. Data Set Ready (DSR) is an indication for the data set i.e., the modem DSU/CSU i.e., it is on. Similarly, DTR indicates to the data set that the DTE is on. Data carrier (DCD) indicates that the carrier for data transmit is on. Pins 4 & 5 carry the RTS and CTS constantly go throughout the communication session. However when the DTE is connected turn carrier on the modem ON & OFF. On a multipoint linear, it is imperative that only one station is transmitting at a line when a station wants to transmit, it raises RTS the modem turns on carrier.

The truth table for RS 232

Signal >+3V=0

Signal <-3V=1

The output signal level usually swings between +12V and -12V. The dead area between +3V and -3V is designated to absorb line noise.

2.5.4 MAX 232:

MAX 232 is primary used for people building electronics with an RS 232 interface. Serial RS-232communications works with voltages (-15V ... -3V for high) and (+3V ... +15V for low) which are not compatible with normal computer logic voltages. To receive serial data from an RS-232 interface the voltages has to be reduced, and low and high voltage level inverted. In the other direction (sending data from some logic over RS-232) the logic voltage has to be "bumped up", and a negative voltages has to be generated, too.

The MAX232 has two sets of the line drivers for transferring and receiving data. The line drivers used for TxD are called T1 and T2, while the line drivers for RxD designated as R1 and R2. in many applications only one of each is used. For example T1 and R1 are used together for TxD and RxD of the PIC and second set is left unused. Notice in MAX232 that the T1 line driver has a designated to T1 in a T1 out on pin numbers 11 and 14 respectively.

The T1 in pin is the TTL side is connected to TxD of the microcontroller, while T1 out is the RS232 side that is connected to the RxD pin of the RS232 DB connector. The R1 line driver has a designation of R1 in R1 out pin numbers 13 and 12 respectively.

The R1 in pin 13 is the RS232 DB connected to the TxD pin of the RS232 DB connector and R1 out(pin 12) in the TTL side that is connected MAX232 requires 4 capacitors ranging from 1 to 22micro farad. The most widely used value for this capacitor is 22microfarad.

Table 2.5.4 MAX 232 Pin Details:

No	Name	Purpose	Signal Voltage
1	C1+	+connector for capacitor C1	Capacitor should stand at least 16V
2	V+	Output of voltage pump	+10V
3	C1-	-connector for capacitor C1	Capacitor should stand at least 16V
4	C2+	+connector for capacitor C2	Capacitor should stand at least 16V
5	C2-	-connector for capacitor C2	
6	V-	Output of voltage pump/inverter	-10V
7	T2out	Driver 2 output	RS232
8	R2in	Receiver 2 input	RS232
9	R2out	Receiver 2 output	TTL
10	T2in	Driver 2 output	TTL
11	T1in	Driver 1 input	TTL
12	R1out	Receiver 1 output	TTL
13	R1in	Receiver 1 input	RS-232
14	T1out	Driver 1 output	RS-232
15	GND	Ground	0V
16	Vcc	Power supply	+5V

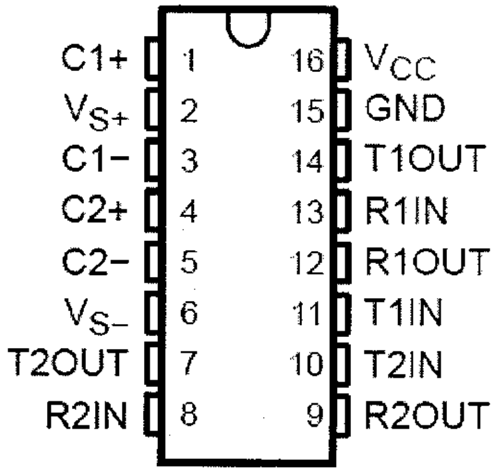


Figure 2.5.4 MAX232 PIN CONFIGURATION

FEATURES:

- Operate from single +5V power supply
- Low-power receive mode in shut down
- Multiple drivers and receivers
- 3-state drivers and Receiver outputs
- Open line detection

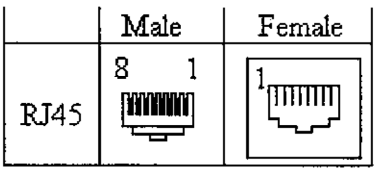
2.5.5 MAX 232 INTERFACING:

The data from microcontroller is made available to the serial port. The encoded RS232 level signal from pin 3 is given to pin 13 of MAX232 IC. The TTL level signal is obtained across pin 12 of the IC and is directly given to the microcontroller. Thus with this signal level conversion, the data is now ready to be transmitted.

The Electronics Industries Association (EIA) standard RS-232-C[1] as of 1969 defines:

- Electrical signal characteristics such as voltage levels, signaling rate, timing and slew-rate of signals, voltage withstand level, short-circuit behavior, and maximum load capacitance.
- Interface mechanical characteristics, pluggable connectors and pin identification.
- Functions of each circuit in the interface connector.
- Standard subsets of interface circuits for selected telecom applications.

Table 2.5.5 MAX 232 PIN DETAILS



Pin No.	Signal Description	Abbr.	DTE	DCE
1	DCE Ready, Ring Indicator	DSR/RI	←	→
2	Received Line Signal Detector	DCD	←	→
3	DTE Ready	DTR	→	←
4	Signal Ground	SG		
5	Received Data	RxD	←	→
6	Transmitted Data	TxD	→	←
7	Clear To Send	CTS	←	→
8	Request To Send	RTS	→	←

2.6.LCD DISPLAY

2.6.1 INTRODUCTION

LCD's also are used as numerical indicators,likewise in our project we are using this to display the fuel level digitally.LCD needs much smaller current than LED displays (microamperes compared with mill amperes) prolong battery life. Liquid crystals are organic (carbon) compounds, which exhibit both solid and liquid properties. A 'cell' with transparent metallic conductors, called electrodes, on opposite daces, containing a liquid crystal, and on which light falls, goes 'dark' when a voltage is applied across the electrodes. The effect is due to molecular rearrangement within the liquid crystal. The LCD display used in our project consists of 2 rows. Each row consists of maximum 16 characters. So using maximum of 32 characters can be display.

LCDs are of two types:

I. Dynamic scattering type

II. Field effect type

The construction of a dynamic scattering liquid crystal cell. The liquid crystal material may be one of the several components, which exhibit optical properties of a crystal though they remain in liquid form. Liquid crystal is layered between glass sheets with transparent electrodes deposited on the inside faces.

When a potential is applied across the cell, charge carriers flowing through the liquid disrupt the molecular alignment and produce turbulence. When the liquid is not activated, it is transparent. When the liquid is activated the molecular turbulence causes light to be scattered in all directions and the cell appears to be bright.

This phenomenon is called dynamic scattering.

The construction of a field effect liquid crystal display is similar to that of the dynamic scattering type, with the exception that two thin polarizing optical filters are placed at the inside of each glass sheet. The liquid crystal material in the field effect cell is also of different type from employed in the dynamic scattering cell.

Liquid crystal cells are of two types:

i. Transmittive type

ii. Reflective type

In the transmittive type cell, both glass sheets are transparent, so that light from a rear source is scattered in the forward direction when the cell is activated.

The liquid crystals are light reflectors are transmitters and therefore they consume small amounts of energy (unlike light generators).

The seven segment display, the current is about 25micro Amps for dynamic scattering cells and 300micro amps for field effect cells. Unlike LEDs which can work on d.c. the LCDs require a.c. voltage supply. A typical voltage supply to dynamic scattering LCD is 30v peak to peak with 50 Hz

2.6.2 PIN DESCRIPTION

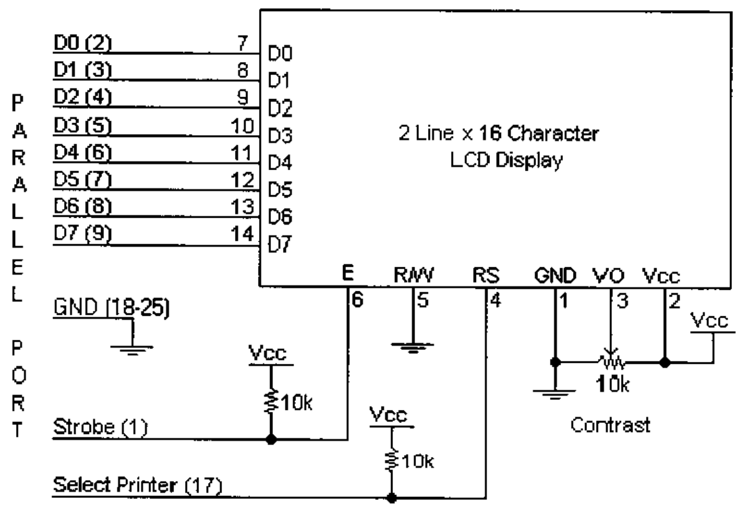


Figure 2.6.2 LCD PIN DESCRIPTION

Table 2.6.2 LCD PIN DETAILS

PIN NO	SYMBOL	FUNCTION
1	Vss	Ground terminal of Module
2	Vdd	Supply terminal of Module, +5v
3	Vo	Power supply for liquid crystal drive
4	RS	Register select, RS=0...Instruction register RS=1...Data register
5	R/W	Read/Write, R/W=1...Read, R/W=0...Write
6	E(EI)	Enable
7-14	DB0-DB7	Bi-directional Data Bus. Data Transfer is performed once ,thru DB0-DB7,incase of interface data length is 8-bits;and twice, thru DB4-DB7 in the case of interface data length is 4-bits. Upper four bits first then lower four bits.
15	LAMP-(L-)	LED or EL lamp power supply terminals
16	LAMP+(L+) (E2)	Enable

2.6.3 ADVANTAGES:

1. Consume much lesser energy (i.e. low power) when compared to LEDs.
2. Utilizes the light available outside and no generation of light.
3. Since very thin layer of liquid crystal is used, more suitable to act as display elements (in digital watches, pocket calculators, ect.)
4. Since reflectivity is highly sensitive to temperature, used as temperature measuring sensor.
5. Very cheap.

DISADVANTAGES:

1. Angle of viewing is very limited.
2. External light is a must for display.
3. Since not generating its own light and makes use of external light for display, contrast is poor.
4. Cannot be used under wide range of temperature.

APPLICATIONS:

1. Watches
2. Fax & Copy machines & Calculators.

CHAPTER - 3

3.MICROCONTROLLER ATMEL 89S51:

SERIES: 89S51 Family, **TECHNOLOGY** :CMOS

The microcontroller acts as the heart of the circuit. Its main function is to control all the operations of the circuit.

3.1 FEATURES:

The major Features of 8-bit Micro controller **ATMEL 89S51**:

- 8 Bit CPU optimized for control applications
- Extensive Boolean processing (Single - bit Logic) Capabilities.
- On - Chip Flash Program Memory
- On - Chip Data RAM
- Bi-directional and Individually Addressable I/O Lines
- Multiple 16-Bit Timer/Counters
- Full Duplex UART
- On - Chip Oscillator and Clock circuitry.
- On - Chip EEPROM
- SPI Serial Bus Interface
- Watch Dog Timer

3.2 MEMORY ORGANIZATION:

* Logical Separation of Program and Data Memory *

All Atmel Flash micro controllers have separate address spaces for program and data memory as shown in Fig 1.The logical separation of program and data memory allows the data memory to be accessed by 8 bit addresses. This can be more quickly stored and manipulated by an 8 bit CPU Nevertheless 16 Bit data memory addresses can also be generated through the DPTR register.

Program memory can only be read. There can be up to 64K bytes of directly addressable program memory. The read strobe for external program memory is the Program Store Enable Signal (PSEN) Data memory occupies a separate address space from program memory. Up to 64K bytes of external memory can be directly addressed in the external data memory space.

3.2.1 PROGRAM MEMORY:

After reset, the CPU begins execution from location 0000h. 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 external Interrupt 0 is used, its service routine must begin at location 0003h. If the interrupt in 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, 0013h for External interrupt 1,001Bh for Timer1, 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. If other interrupts are in use. The lowest addresses of program memory can be either in the on-chip Flash or in an external memory. To make this selection, strap the External Access (EA) pin to either Vcc or GND. For example, in the AT89S51 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.

3.2.2 DATA MEMORY:

The Internal Data memory is dived into three blocks namely, the lower 128 Bytes of Internal RAM. The Upper 128 Bytes of Internal RAM. Special Function Register. 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, is in use. This architecture allows more efficient use of code space, since register instructions are shorter than instructions that use direct addressing.The next 16-bytes above the register banks form a block of bit addressable memory space. The micro controller instruction set includes a wide selection of single - bit instructions and this instruction can directly address the 128 bytes in this area.

These bit addresses are 00h through 7Fh. either direct or indirect addressing can access all of the bytes in lower 128 bytes. Indirect addressing can only access the upper 128. The upper 128 bytes of RAM are only in the devices with 256 bytes of RAM.

3.2.3 FLASH ROM:

4-kilo byte ROM is available in the Micro controller. It can be erased and reprogrammed. 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 maximum of (2^16) i.e. 64 bytes of ROM can be interfaced with this Micro controller. Both internal and external ROM cannot be used simultaneously.

For external accessing of ROM, A pin is provided in Micro controller itself is i.e. pin no.31 EA should be high to use internal ROM, low to use external ROM

3.2.4 RAM:

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

3.3 OSCILLATOR AND CLOCK CIRCUIT:

XTAL1 and XTAL2 are the input and output respectively of an inverting amplifier which is intended for use as a crystal oscillator in the pierce configuration, in the frequency range of 1.2 Mhz to 12 Mhz. XTAL2 also the input to the internal clock generator.

The clock generator divides the oscillator frequency by 12 and provides a low phase clock signal to the chip. The phase 1 signal is active during the first half to each clock period and the phase 2 signals are active during the second half of each clock period.

Block Diagram

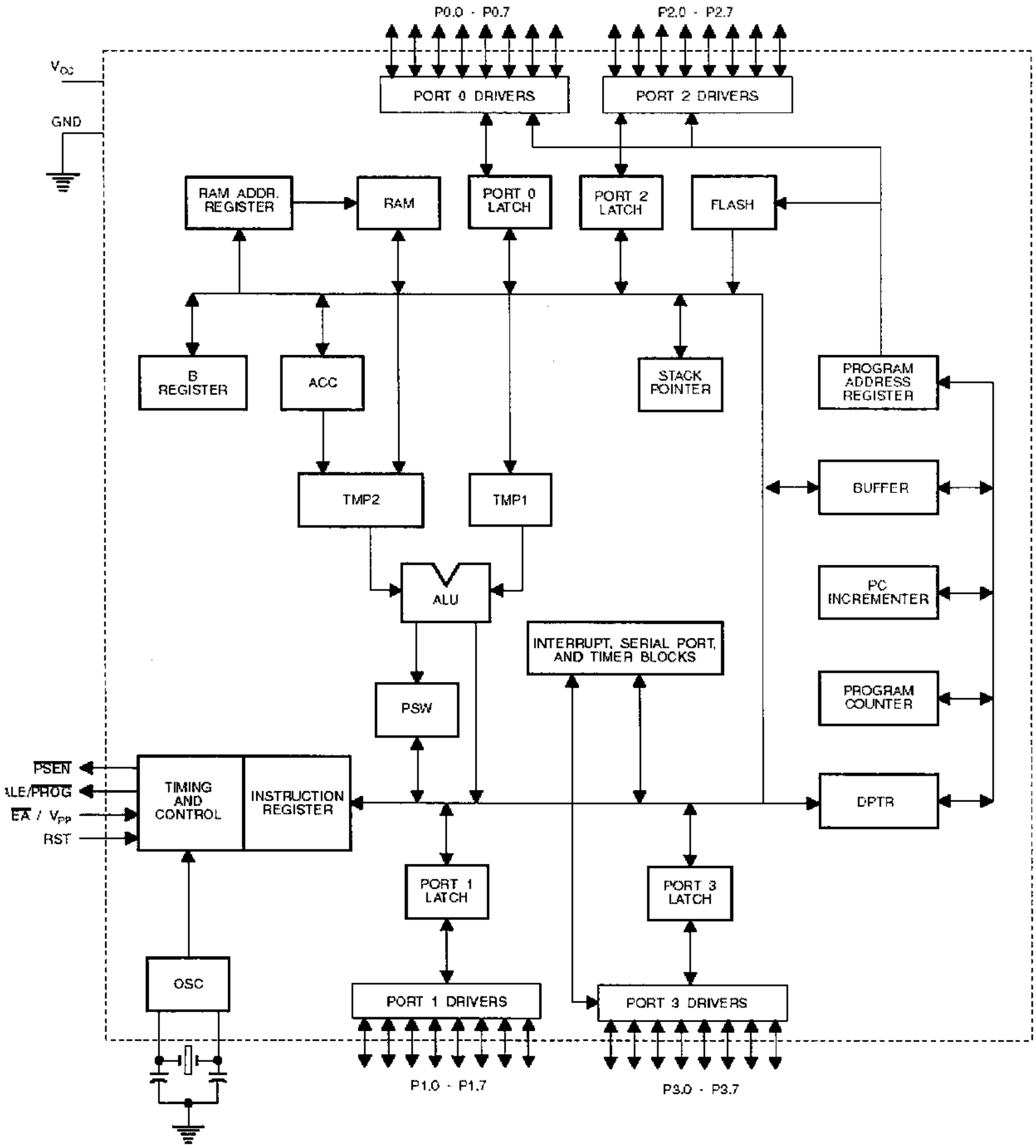


Figure 3.3 MICROCONTROLLER BLOCK DIAGRAM

3.3.1 CPU TIMING:

A machine cycle consists of 6 states. Each stare is divided into a phase / half, during which the phase 1 clock is active and phase 2 half. Arithmetic and Logical operations take place during phase1 and internal register - to register transfer take place during phase 2.

3.4 COMPARISION BETWEEN MICROCONTROLLER AND MICROPROCESSOR:

The difference between Microprocessor and Micro controller is Microprocessor can only process with the data, Micro controller can control external device. That is if you want switch "ON" or "OFF" a device, you need peripheral ICs to do this work with Micro controller you can directly control the device.

Like Microprocessor, Micro controller is available with different features. It is available with inbuilt memory, I/O lines, timer and ADC. The micro controller, which we are going to use, is 89S51 it is manufactured by Atmel, MC, USA. This is advanced version of 8031. This Micro controller have inbuilt 4K bytes of flash ROM, 256 bytes of RAM, 32 I/O lines (4 bit ports) and 6 vectored interrupts

3.5 TRENDS AND DEVELOPMENT IN MICROCONTROLLER:

The manner in which the use of micro controllers is shaping our lives is breathtaking. Today, this versatile device can be found in a variety of control applications. **CVTs, VCRs, CD** players, microwave ovens, and automotive engine systems are some of these.

A micro controller unit (MCU) uses the microprocessor as its central processing unit (CPU) and incorporates memory, timing reference, **I/O** peripherals, etc on the same chip. Limited computational capabilities and enhanced **I/O** are special features.

The micro controller is the most essential **IC** for continuous process- based applications in industries like chemical, refinery, pharmaceutical automobile, steel, and electrical, employing programmable logic systems (DCS). **PLC** and **DCS** thrive on the programmability of an **MCU.**

There are many **MCU** manufacturers. To understand and apply general concepts, it is necessary to study one type in detail.

Micro controller devices have many similarities. When you look at the differences, they are not so great either. Most common and popular **MCU** are considered to be mature and well-established products, which have their individual adherents and devotees.

The **MCU** is designed to operate on application-oriented sensor data-for example, temperature and pressure of a blast furnace in an industrial process that is fed through its serial or operated on under the control of software and stored in **ROM**. Incoming data can be from a variety of external devices sensors. Priority for receiving and operating on the data can be established using the interrupt-control circuitry. The microcontroller operates with external circuitry to perform control-oriented tasks, using a control program in **ROM**. The instruction set is simpler than that of a microprocessor, since most of its instructions will move code and data from internal memory to **ALU** (arithmetic logic unit).

The micro controller is designed to allow programming for single-task operations, which is ideal for applications requiring large product volumes. For instance, microcontroller-based automatic washing machines monitor and control external parameters (such as water level and temperature) according to the requirements of the wash cycle. Such **RAM** to store transient data, and a large **ROM** to accommodate the control program. The microcontroller includes registers (not only used by its **CPU** but by timer), interrupt control units, port latches, etc.

3.6 PIN CONFIGURATION:

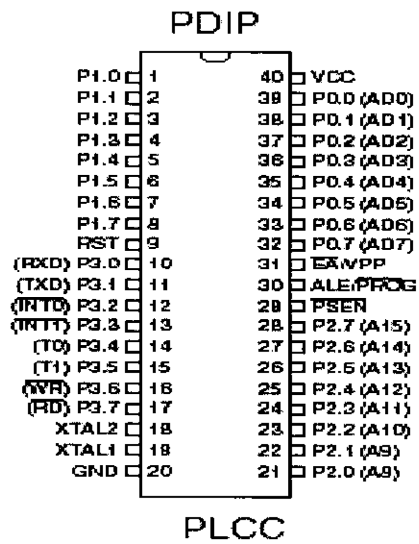


Figure 3.6 MICROCONTROLLER PIN CONFIGURATION

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 1 s ear written to port 0 pins, the pins can be used as high impedance inputs. Port 0 can also be configured to be the multiplexed lower order address/data bus during accesses to external program and data memory. In this mode, p0 has internal pull-ups. Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pull-ups are required during program verification.

PORT 1:

Port 1 is an 8-bit bi-directional I/O port with internal pull-ups. The port 1 output buffers can sink/source four TTL inputs. When 1s are written to port 1 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 1 pins that are externally being pulled low will source current(IIL) because of the internal pull-ups. In addition, P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively, as shown in the following table. Port 1 also receives the low-order address bytes during Flash programming and verification.

PORT 2:

Port 2 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 2 buffers can sink/source four TTL inputs. When 1 s are written to Port 2 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 2 pins that are externally being pulled low will source current(IIL) because of the internal pull-ups. Port 2 emits the high order address byte during fetches from external program memory and during accesses to external data memory that uses 16-bit addresses. In this application, Port 2 uses strong internal pull-ups when emitting 1s. during accesses to external data memory that uses 8-bit addresses, Port2 emits the contents of the P2 special function register .

PORT 3:

Port 3 is an 8-bit bi-directional I/O port with internal pull-ups. The Port3 output buffers buffers can sink/source four TTL inputs. When 1 s are written to Port 2 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 2 pins that are externally being pulled low will source current(IIL) because of the internal pull-ups. 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 is an output pulse for latching the low byte of the address during accesses to external memory. 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 weekly pulled high. Setting the ALE-disable bit has no effect if the micro controller is in external execution mode.

PSEN:

Program store enable is the read strobe to external program memory. When the AT89S51 is executing code from external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory.

EA/VPP:

External Access Enable. EA must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000h up to FFFFH. Note, however, that if lock bit 1 is programmed, EA will be internally latched on reset. EA should be strapped to VCC for internal program executions. This pin also receives the 12V programming enable voltage(VPP) during Flash programming when 12V programming is selected.

3.7.RELAYS

3.7.1 INTRODUCTION

The relay is connected to the microcontroller where the fuel level information can be shared without any interfacing. By using this the microcontroller can be connected to the external load directly. Relays are electromagnetic switches, which provide contact between two mechanical elements. Relays have a coil, which works on 12 V dc power supply and provides DPDT action as an output. In general, relays provide potential free contacts which can be used for universal function like DC, AC voltage switching and to control bigger electrical switchgears.

The electromechanical relays are based on comparison between operating torque/force and restraining torque/force. The characteristics of these relays have some limitations.

Each relay can perform only one protective function. Such relays are used for simple and less costly protection purpose. For important and costly equipment installation static relays are preferred.

Relays are components, which allow a low-power circuit to switch a relatively high current on and off, or to control signals that must be electrically isolated from the controlling circuit itself.

3.7.2 CIRCUIT DIAGRAM

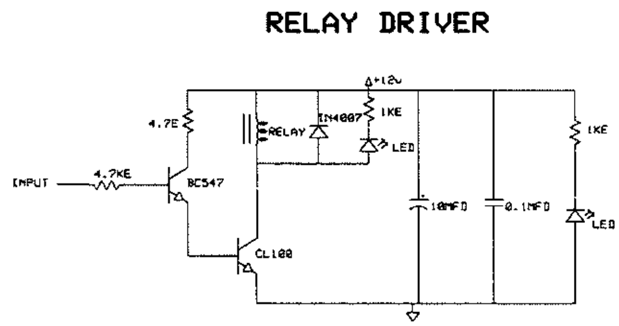


Figure 3.7.2 CIRCUIT DIAGRAM

DESCRIPTION

Finally in these circuits, it's good to fit the supply line of the relay/driver stage with reasonably high value of bypass capacitor, to absorb the current transient when the relay turns on and off. This will ensure more reliable operation, and help prevent interference with the operation of your control circuitry.

Protective relaying is necessary for most every electrical plant and no part of the power system is left unprotected. The choice of protection depends upon several aspects such as,

- Type and rating of the protected equipment
- Location and cost
- Probable abnormal conditions between generators and final points.

There are several electrical equipments and machines of various ratings. Each needs certain adequate protection. The protection relaying senses the abnormal conditions in the part of the power system. The relays used in this project are compact, self contained devices, which respond to abnormal conditions.

These contacts can be either normally open(NO) or normally close(NC), or change over contacts. Normally open contacts connect the circuit when the relay is activated; the circuit is disconnected when the relay is inactive. It is also called form A contact or "make" contact. Form A contact is ideal for application that require to switch a high current power source from a remote device.

Normally closed contacts disconnect the circuit when the relay is activated; the circuit is connected when the relay is inactive. It is also called form B contact or "break" contact. Form B contact is ideal for application that require the circuit to remain closed until the relay is activated.

Change over contacts control two circuits. Normally open contact and one normally closed contact. It is also called form C contacts.

3.7.3 OPERATION

All relay operates using the same basic principle .The four pin relay is used. A control circuit has a small control coil while the load circuit has a switch. The coil controls the operation of the switch. When no voltage is applied to pin 1,there is no current flow through the coil. Due to which the magnetic field will not be developed. And the switch will be open. When voltage is supplied to pin 1, current flow through the coil which creates the magnetic field needed to close the switch allowing continuity between pin 2 and 4.

TYPES OF RELAY

- Latching relay
- Reed relay
- Mercury wetted relay
- Machine tool relay

3.7.4 APPLICATIONS

Relays are used to control a high voltage circuit with a low voltage signal, as in some types of modem.

- To control a high current circuit with a low current signal, as in starter solenoid of
- an automobile.
- To perform time delay functions.
- To perform logic functions.
- To detect and isolate faults on transmission and distribution lines by opening
- and closing circuit breakers.
- To generate sound.
- As oscillators, also called vibrators

3.8.ANALOG TO DIGITAL CONVERTER

3.8.1 INTRODUCTION

The analog information of the fuel level is converted digitally here. ADC 0809 analog to digital converter is a successive approximation type analog to digital converter. The successive approximation technique uses a very efficient code search strategy to complete n-bit conversion in just n-clock periods. The circuit uses a successive approximation register to find the required value by trial and error. The ADC0809 data acquisition component is a monolithic CMOS device with an 8-bit analog-to-digital converter, 8-channel multiplexer and microprocessor compatible control logic. The converter features a high impedance chopper stabilized comparator, a 256R voltage divider with analog switch tree and a successive approximation register.

3.8.2 OPERATION OF CIRCUIT:

With the arrival of START command, the SAR sets the MSB d1=1 with all other bits to zero so that the trial code is 10000000. The output of the ADC is now compared with analog input.

If input is greater than the DAC output then 10000000 is less than the correct digital representation. The MSB is left at '1' and the next lower significant bit is made '1' and further tested. However, if input is less than the DAC output, then 10000000 is greater than the correct digital representation. So reset MSB to '0' and go on to the next lower significant bit.

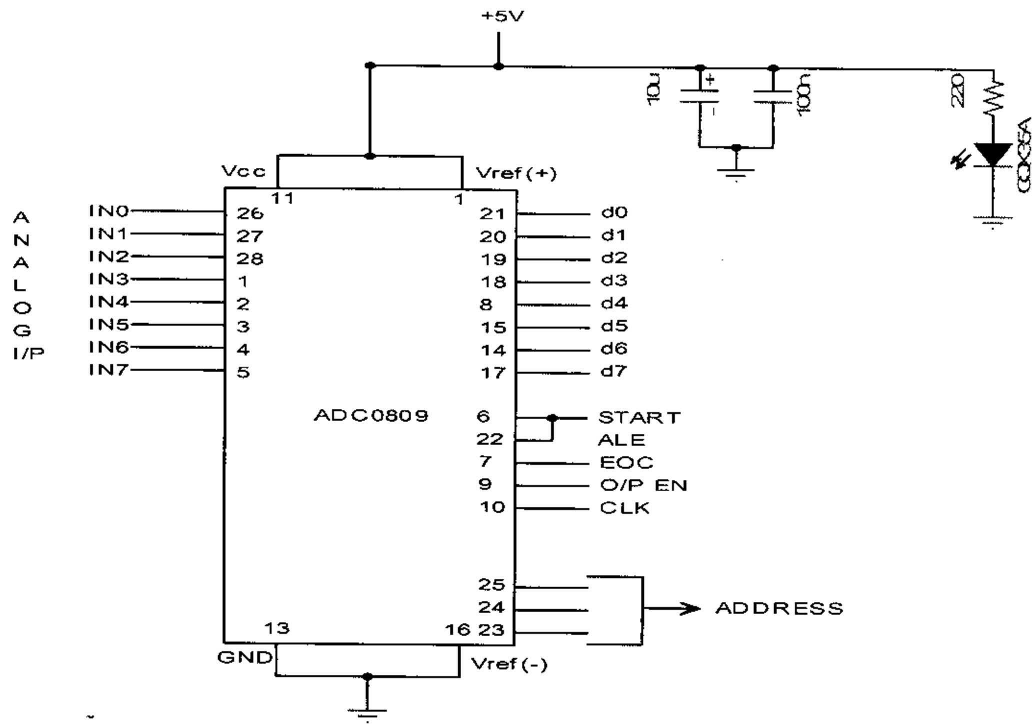


Figure 3.8.2 ADC PIN CONFIGURATION

3.8.3 CIRCUIT DESCRIPTION:

The output from the filter is given to pin 26 of ADC 0809 shown in fig .The address channels A, B, C are grounded so that channel 1 is enabled. The digitized output from the converter is given to port 0 of micro controller. The filter capacitors in the circuit remove the low and high frequency noises. The control signals from the ADC are given to port 2 of the Microcontroller. This circuit follows the principle of successive approximation method and when the start of conversion goes high, it marks the beginning of the process and high end of conversion marks the end of it.

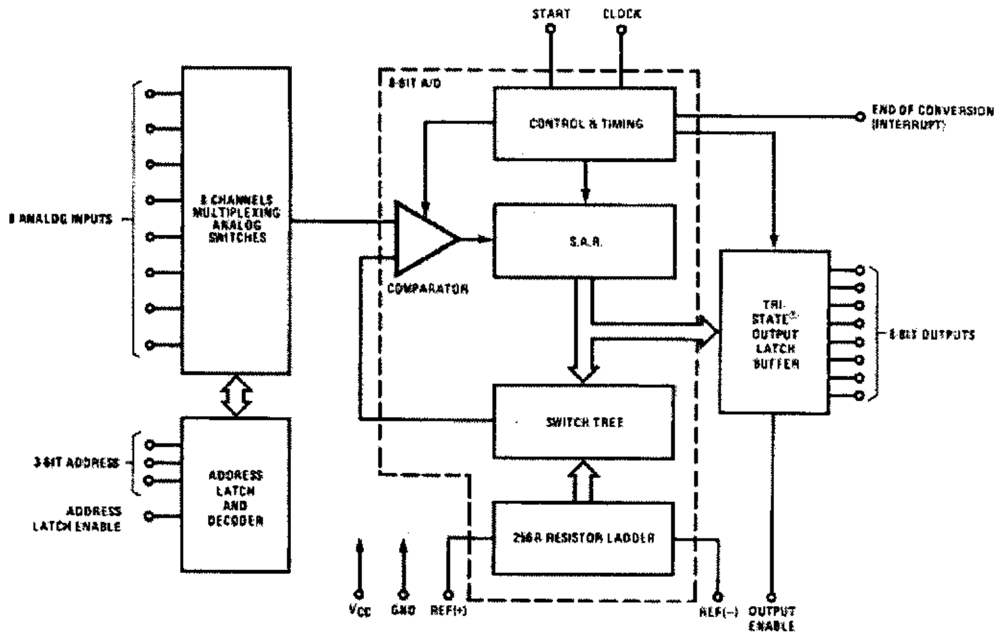


Figure 3.8.3 ADC BLOCK DIAGRAM

3.8.4 FEATURES:

- Easy interface to all microprocessors
- Operates ratiometrically or with 5 VDC or analog spa adjusted voltage reference
- No zero or full-scale adjust required
- 8-channel multiplexer with address logic
- 0V to 5V input range with single 5V power supply
- Outputs meet TTL voltage level specifications
- Standard hermetic or molded 28-pin DIP package
- 28-pin molded chip carrier package
- ADC0808 equivalent to MM74C949
- ADC0809 equivalent to MM74C949-1

CHAPTER - 4

4. SOFTWARE DETAILS

4.1 INTRODUCTION TO VISUAL BASIC

Visual Basic (VB) is the third-generation event-driven programming language and integrated development environment (IDE) from Microsoft for its COM programming model. VB is also considered a relatively easy to learn and use programming language, because of its graphical development features and BASIC heritage.

Visual Basic was derived from BASIC and enables the rapid application development (RAD) of graphical user interface (GUI) applications, access to databases using Data Access Objects, Remote Data Objects, or ActiveX Data Objects, and creation of ActiveX controls and objects. Scripting languages such as VBA and VBScript are syntactically similar to Visual Basic, but perform differently.

A programmer can put together an application using the components provided with Visual Basic itself. Programs written in Visual Basic can also use the Windows API, but doing so requires external function declarations.The final release was version 6 in 1998. Microsoft's extended support ended in March 2008 and the designated successor was Visual Basic .NET (now known simply as Visual Basic).

FEATURES

Like the BASIC programming language, Visual Basic was designed to be easily learned and used by beginner programmers. The language not only allows programmers to create simple GUI applications, but can also develop complexapplications. Programming in VB is a combination of visually arranging components or controls on a form, specifying attributes and actions of those components, and writing additional lines of code for more functionality. Since default attributes and actions are defined for the components, a simple program can be created without the programmer having to write many lines of code. Performance problems were experienced by earlier versions, but with faster computers and native code compilation this has become less of an issue.

Although programs can be compiled into native code executables from version 5 onwards, they still require the presence of runtime libraries of approximately 1 MB in size. This runtime is included by default in Windows 2000 and later, but for earlier versions of Windows like 95/98/NT it must be distributed together with the executable.

Forms are created using drag-and-drop techniques. A tool is used to place controls (e.g., text boxes, buttons, etc.) on the form (window). Controls have attributes and event handlers associated with them. Default values are provided when the control is created, but may be changed by the programmer. Many attribute values can be modified during run time based on user actions or changes in the environment, providing a dynamic application. For example, code can be inserted into the form resize event handler to reposition a control so that it remains centered on the form, expands to fill up the form, etc.

By inserting code into the event handler for a keypress in a text box, the program can automatically translate the case of the text being entered, or even prevent certain characters from being inserted.

Visual Basic can create executables (EXE files), ActiveX controls, or DLL files, but is primarily used to develop Windows applications and to interface database systems. Dialog boxes with less functionality can be used to provide pop-up capabilities. Controls provide the basic functionality of the application, while programmers can insert additional logic within the appropriate event handlers. For example, a drop-down combination box will automatically display its list and allow the user to select any element. An event handler is called when an item is selected, which can then execute additional code created by the programmer to perform some action based on which element was selected, such as populating a related list.

Alternatively, a Visual Basic component can have no user interface, and instead provide ActiveX objects to other programs via Component Object Model (COM). This allows for server-side processing or an add-in modu.The language is garbage collected using reference counting, has a large library of utility objects, and has basic object oriented support. Since the more common components are included in the default project template, the programmer seldom needs to specify additional libraries.

Unlike many other programming languages, Visual Basic is generally not case sensitive, although it will transform keywords into a standard case configuration and force the case of variable names to conform to the case of the entry within the symbol table. String comparisons are case sensitive by default, but can be made case insensitive if so desired.

The Visual Basic compiler is shared with other Visual Studio languages (C, C++), but restrictions in the IDE do not allow the creation of some targets (Windows model DLLs) and threading models.

CHARACTERISTICS

Visual Basic has the following traits which differ from C-derived languages:

- Multiple assignment available in C language is not possible. A = B = C does not imply that the values of A, B and C are equal. The boolean result of "Is B = C?" is stored in A. The result stored in A would therefore be either false or true.

- Boolean constant `True` has numeric value −1. This is because the Boolean data type is stored as a 16-bit signed integer. In this construct −1 evaluates to 16 binary 1s (the Boolean value `True`), and 0 as 16 0s (the Boolean value `False`). This is apparent when performing a `Not` operation on a 16 bit signed integer value 0 which will return the integer value −1, in other words `True = Not False`. This inherent functionality becomes especially useful when performing logical operations on the individual bits of an integer such as `And`, `Or`, `Xor` and `Not`. This definition of `True` is also consistent with BASIC since the early 1970s Microsoft BASIC implementation and is also related to the characteristics of CPU instructions at the time.

- Logical and bitwise operators are unified. This is unlike some C-derived languages (such as Perl), which have separate logical and bitwise operators. This again is a traditional feature of BASIC.

- Variable array base. Arrays are declared by specifying the upper and lower bounds in a way similar to Pascal and Fortran. It is also possible to use the Option Base statement to set the default lower bound. Use of the Option Base statement can lead to confusion when reading Visual Basic code and is best avoided by always explicitly specifying the lower bound of the array.

- This lower bound is not limited to 0 or 1, because it can also be set by declaration. In this way, both the lower and upper bounds are programmable. In more subscript-limited languages, the lower bound of the array is not variable. This uncommon trait does exist in Visual Basic .NET but not in VBScript.

- Relatively strong integration with the Windows operating system and the Component Object Model. The native types for strings and arrays are the dedicated COM types, BSTR and SAFEARRAY.

- Banker's rounding as the default behavior when converting real numbers to integers with the `Round` function.[5] `? Round(2.5, 0)` gives 2, `? Round(3.5, 0)` gives 4.

- By default, if a variable has not been declared or if no type declaration character is specified, the variable is of type Variant. However this can be changed with Deftype statements such as `DefInt`, `DefBool`, `DefVar`, `DefObj`, `DefStr`. There are 12 `Deftype` statements in total offered by Visual Basic 6.0. The default type may be overridden for a specific declaration by using a special suffix character on the variable name (`#` for Double, `!` for Single, `&` for Long, `%` for Integer, `$` for String, and `@` for Currency) or using the key phrase `As (type)`. VB can also be set in a mode that only explicitly declared variables can be used with the command `Option Explicit`.

4.2 VISUAL BASIC PROGRAM

```
Dim vtank, ptank, nstatus, temp, amount, famount, dvalue As Integer
Dim ttemp As String
Dim commdata As String
Private Sub Command1_Click()
If Val(Text2.Text) > 100 Then
If Text7(0) = "ON" Then
amount = Val(Text2.Text)
MSComm1.Output = "1"
Text7(2).Text = "ON"
dvalue = 0
```

```
Timer2.Enabled = True
End If
End If
End Sub
Private Sub Command3_Click()
If Text7(2) = "ON" Then
MSComm1.Output = "0"
Text7(2).Text = "OFF"
Calculate
End If
End Sub
Private Sub Form_Load()
Text2.Text = 5000
amount = Val(Text2.Text)
Text2.Text = amount
MSComm1.PortOpen = True
Text7(0).Text = "OFF"
Text7(2).Text = "OFF"
End Sub
Private Sub Form_Unload(Cancel As Integer)
MSComm1.PortOpen = False
End Sub
Private Sub Label6_Click(Index As Integer)
End Sub
Private Sub MSComm1_OnComm()
Text5.Text = MSComm1.Input
If Text5.Text = "N" Then commdata = ""
commdata = commdata & Text5.Text
Text4.Text = commdata
If Len(commdata) = 8 Then
ttemp = Mid(Text4.Text, 1, 1)
If ttemp = "N" Then
nstatus = Val(Mid(Text4.Text, 2, 3))
ptank = Val(Mid(Text4.Text, 5, 3))
```

```
vtank = Val(Mid(Text4.Text, 8, 1))
commdata = ""
Text4.Text = commdata
Text6.Text = ptank
'Text1.Text = vtank
End If
End If
If nstatus = 1 Then Text7(0).Text = "ON" Else Text7(0).Text = "OFF"
If vtank = 1 Then Text1.Text = "FULL" Else Text1.Text = "LOW"
If Text7(2).Text = "ON" Then
If vtank = 1 Then Text7(2).Text = "OFF": MSComm1.Output = "0": calculate
End If
End Sub
Private Sub calculate()
Timer2.Enabled = False
amount = amount - famount
Text2.Text = amount
Form2.Label2(0).Caption = Text3.Text
Form2.Label2(1).Caption = Text8.Text
Form2.Label2(2).Caption = Text2.Text
Form2.Label2(3).Caption = "XYZ"
Form2.Show
End Sub
Private Sub Timer1_Timer()
MSComm1.Output = "2"
End Sub
Private Sub Timer2_Timer()
dvalue = dvalue + 1
famount = dvalue * 63
Text3.Text = dvalue
Text8.Text = famount
If Val(Text2.Text) < famount Then Text7(2).Text = "OFF": MSComm1.Output = "0": calculate
End Sub
```

4.3 ASSEMBLY LANGUAGE

RAD51 is an acronym for Rapid ApplicationDevelopment for 8051s. RAD51 is an Integrated Development Environment (IDE) with a comfortable text editor, and an 8051 cross-assembler. It's a true 32-bit Windows application and requires Windows 9X or NT. RAD51 takes a 'project' view of your code and retains basic information in a PRJ file. Shared code such as I/O routines can be kept in a 'parent' folder. Individual projects are stored in 'child' folders which can #include the shared code. We include several samples which illustrate this.

RAD51 FEATURES

- 32-bit Windows application. Takes advantage of unlimited memory and "flat" memory model in the 32-bit environment. Supports long filenames.
- Extensible-able to accept tool wizards and plug-ins.
- Processors defined in an ASCII controller.cfg file which can be easily expanded to accommodate new microcontrollers.
- Tools to help you write the assembly code programs. For example, RAD51 generates a symbol table in one of the project window panes. Click on any symbol and the editor takes you to the place of that symbol's definition.
- Built-in text editor with syntax highlighting.
- Project management features. For example, RAD51 supports an environment in which a base code folder holds frequently-reused code such as I/O libraries. Individual projects are subfolders below this folder. The projects include whichever of the common libraries they need. In this manner you have only one place to keep and maintain the common code, and a logically structured layout for each project.

4.4 MICROCONTROLLER PROGRAM

VEHICLE SIDE ASSEMBLY LANGUAGE PROGRAM

```
txd0            equ 90h
txd1            equ 91h
txd2            equ 92h
txd3            equ 93h
te              equ 94h         ;transmit enable


prs      equ 0b0h               ;register select
prw       equ 0b1h              ;read/write
pen      equ 0b2h               ;enable

lcd_d4          equ 0b3h        ;databit4
lcd_d5          equ 0b4h        ;databit5
lcd_d6          equ 0b5h        ;databit6
lcd_d7          equ 0b6h        ;databit7

data      equ 70h               ;data storage to command LCD and displ


ale             equ 0a0h         ;6,22
eoc             equ 0a4h         ;7

oe              equ 0a5h         ;9
clk             equ 0a6h         ;10

addra           equ 0a3h         ;25 address bit a
addrb           equ 0a2h         ;24 address bit b
addrc           equ 0a1h         ;23 address bit c


adc             equ p0          ;data bits

adc_value       equ 30h
set_value       equ 31h
incrb           equ 95h
decrb           equ 96h

  org 0000h

       mov p2,#00h
       mov p3,#00h
```

```
mov p1,#0ffh
mov adc,#0ffh

 setb incrb
 setb decrb

 mov set_value,#50d

 clr addra
 clr addrb
 clr addrc

                 mov 50h,#00h      ;first byte
                 mov 51h,#00h      ;second byte
                 mov 52h,#00h      ;third byte
                 mov 53h,#00h      ;fourth byte
                 mov 54h,#00h      ;fifth byte

                 mov 43h,50h     ;first byte
                 mov 42h,51h     ;second byte
                 mov 41h,52h     ;third byte
                 mov 40h,53h     ;fourth byte



                 acall initialise_lcd

                 mov dptr,#wel
                 acall cont

                 acall delay
                 acall delay
                 acall delay
                 acall delay

                 mov dptr,#condition
                 acall cont


  start
                 mov a,#89h
                 acall command

      acall read_adc

      mov adc_value,a

      mov 53h,a
```

```
acall hex_to_decimal

mov a,53h
anl a,#0fh
acall convert
acall display

mov a,54h
swap a
anl a,#0fh
acall convert
acall display

mov a,54h
anl a,#0fh
acall convert
acall display

mov a,#"%"
acall display


mov a,#0c9h
            acall command

            mov a,set_value

mov 53h,a
acall hex_to_decimal

mov a,53h
anl a,#0fh
acall convert
acall display

mov a,54h
swap a
anl a,#0fh
acall convert
acall display

mov a,54h
anl a,#0fh
acall convert
acall display

mov a,#"%"
acall display
```

```
                jb incrb,s1
                acall delay
                jb incrb,s1
                acall increment


s1              jb decrb,s2
                acall delay
                jb decrb,s2
                acall decrement

s2

                mov a,adc_value

                cjne a,set_value,x1
                sjmp lowvalue


x1      jnc highvalue                                  ;when value of a is less than 98 will set
        sjmp lowvalue


lowvalue   mov a,#01h                                  ;empty
                acall sebd_data
                        acall delay
                ajmp start



highvalue  mov a,#02h                                  ;full
                acall sebd_data
                        acall delay
                ajmp start


                                          ;transmit data wirelessly
                mov a,#0fh
                acall sebd_data

                          mov a,53h
                anl a,#0fh
                acall sebd_data

                mov a,54h
```

```
            swap a
            anl a,#0fh
            acall sebd_data

            mov a,54h
            anl a,#0fh
            acall sebd_data

            ;acall delay

                        ajmp start


delay           mov r0,#01h
in2             mov r1,#00h
in1             mov r2,#00h
wait1           djnz r2,wait1
        djnz r1,in1
        djnz r0,in2
        ret



increment       inc set_value
                        ret


decrement   dec set_value
                        ret



        sebd_data       ;ret
                        mov c,acc.0
                        mov txd0,c
                        mov c,acc.1
                        mov txd1,c
                        mov c,acc.2
                        mov txd2,c
                        mov c,acc.3
                        mov txd3,c

                        clr te
                        acall delay
                        setb te
                        acall delay

                        setb incrb
```

```
setb decrb

hex_to_decimal

        mov 43h,50h     ;first byte
        mov 42h,51h     ;second byte
        mov 41h,52h     ;third byte
        mov 40h,53h     ;fourth byte

        mov 50h,#00h    ;first byte
        mov 51h,#00h    ;second byte
        mov 52h,#00h    ;third byte
        mov 53h,#00h    ;fourth byte
        mov 54h,#00h    ;fifth byte


        mov r0,#20h
nxtbit
        mov a,40h
        add a,40h
        mov 40h,a

        mov a,41h
        addc a,41h
        mov 41h,a

        mov a,42h
        addc a,42h
        mov 42h,a

        mov a,43h
        addc a,43h
        mov 43h,a

        mov a,54h
        addc a,54h
        da a
        mov 54h,a

        mov a,53h
        addc a,53h
        da a
        mov 53h,a

        mov a,52h
        addc a,52h
        da a
```

```
            mov 52h,a

            mov a,51h
            addc a,51h
            da a
            mov 51h,a


            mov a,50h
            addc a,50h
            da a
            mov 50h,a

            djnz r0,nxtbit
            mov r1,50h      ;first byte
            mov r2,51h      ;second byte
            mov r3,52h      ;third byte
        mov r4,53h      ;fourth byte
        mov r5,54h      ;fifth byte

            ret


 read_adc
            setb ale
            clr oe
            setb eoc

            clr clk
            nop
            setb clk
            nop

            clr ale

adc1
            clr clk
            nop
            setb clk
            nop

            jnb eoc,adc1

            setb oe

            clr clk
            nop
            setb clk
```

```
                nop

                clr clk
                nop
                setb clk
                nop

                clr clk
                nop
                setb clk
                nop

                mov a,adc

                ret


initialise_lcd
                mov a,#2ch        ;4bits/char,2 rows,5x10 dots/char
                acall command
                mov a,#0ch                ;screen on,cursor off and no blink
                acall command
                mov a,#06h                ;shift cursor right
                acall command
                mov a,#01h                        ;clear memory and home cursor
                acall command
                mov a,#80h        ;move cursor to space(0) 0,line 1(8)
                acall command

                ret



cont            mov a,#00h
        movc a,@a+dptr
                inc dptr

                cjne a,#80h,d1        ;move cursor to space(0) 0,line 1(8)
                acall command
                        sjmp cont

d1                      cjne a,#0c0h,d2       ;move cursor to space(0) 0,line 2(c)
                acall command
                        sjmp cont

d2                      cjne a,#0ffh,d3       ;displayed
                ret
```

```
d3              acall display
                sjmp cont

command         acall ready
                clr prs
                clr prw
                acall dout_lcd
                ret

display         acall ready
                setb prs
                clr prw
                acall dout_lcd
                ret


ready           acall lcddelay
                ret


lcddelay        mov 34h,#00h
lcdwait         djnz 34h,lcdwait
                ret

dout_lcd        clr prw

                mov data,a

                mov a,data          ;first send upper nibble
                swap a

                mov c,acc.0
                mov lcd_d4,c

                mov c,acc.1
                mov lcd_d5,c

                mov c,acc.2
                mov lcd_d6,c

                mov c,acc.3
                mov lcd_d7,c

                setb pen
                clr pen
                mov a,data          ;second lower nibble
```

```
            mov c,acc.0
            mov lcd_d4,c
            mov c,acc.1
            mov lcd_d5,c
            mov c,acc.2
            mov lcd_d6,c

            mov c,acc.3
            mov lcd_d7,c

            setb pen
            clr pen

            ret



convert     inc a
            movc a,@a+pc
            ret

            db 30h
            db 31h
            db 32h
            db 33h
            db 34h
            db 35h
            db 36h
            db 37h
            db 38h
            db 39h

            db 41h
            db 42h
            db 43h
            db 44h
            db 45h
            db 46h


wel         db 80h, " AUTOMATION OF  "
            db 0c0h,"   PETROL PUMP   "
            db 0ffh


condition   db 80h, "C. LEVEL:       "
            db 0c0h,"S. LEVEL:       "
            db 0ffh
```

DEVICE SIDE ASSEMBLY LANGUAGE PROGRAM

```
ale             equ 0a0h        ;6,22
eoc             equ 0a4h        ;7

oe              equ     0a5h    ;9
clk             equ 0a6h        ;10

addra           equ 0a3h        ;25 address bit a
addrb           equ 0a2h        ;24 address bit b
addrc           equ 0a1h        ;23 address bit c

adc             equ p0          ;data bits

fuel_level      equ 30h

sense      equ 94h

baudnum                 equ 0f3h

pump            equ 95h
nozzle          equ 96h

org 0000h

        mov p0,#0ffh
                        mov p1,#0ffh
        mov p2,#00h
        mov p3,#0ffh

        clr pump
        setb nozzle

                anl pcon,#7fh           ;set SMOD bit to 0 for Buad*32 rate
                anl tmod,#30h           ;alter timer T1 configuration only
                orl tmod,#20h           ;set timer T1 as an 8-bit autoload

                mov th1,#baudnum        ;TH1 set for divide clock by 13d

                setb tr1        ;T1 running
                mov scon,#40h
```

```
                setb ren

                        mov a,#"c"
                        acall xmit
                        mov a,#"o"
                        acall xmit
                        mov a,#"n"
                        acall xmit
                        mov a,#"n"
                        acall xmit
                        mov a,#"e"
                        acall xmit
                        mov a,#"c"
                        acall xmit
                        mov a,#"t"
                        acall xmit
                        mov a,#"e"
                        acall xmit
                        mov a,#"d"
                        acall xmit


l1                      ;jnb sense,lx1

                        mov a,p1
                        anl a,#0fh
                        ;acall convert
                        ;acall xmit

                        cjne a,#01h,s1
                        mov fuel_level,#"0"

s1                      cjne a,#02h,s2
                        mov fuel_level,#"1"

s2


l2                      jb sense,l2
 sjmp l1


lx1                     jnb ri,l1
                        acall receive

                        cjne a,#"0",l3
                        clr pump
```

```
                sjmp l1

l3              cjne a,#"1",l4
                setb pump
                sjmp l1

l4              cjne a,#"2",l4

      jb nozzle,x1
      acall delay
      jb nozzle,x1

      mov a,#"N"
                acall xmit

                mov a,#"0"
                acall xmit

                mov a,#"0"
                acall xmit

                mov a,#"1"
                acall xmit

                sjmp x2

x1     mov a,#"N"
                acall xmit

                mov a,#"0"
                acall xmit

                mov a,#"0"
                acall xmit

                mov a,#"0"
                acall xmit


x2     acall read_adc

          mov 53h,a
          acall hex_to_decimal

          mov a,53h
          anl a,#0fh
          acall convert
          acall xmit
```

```
        mov a,54h
        swap a
        anl a,#0fh
        acall convert
        acall xmit

        mov a,54h
        anl a,#0fh
        acall convert
        acall xmit

        mov a,fuel_level                ;send fuel level
        acall xmit

        acall delay

    ajmp l1


hex_to_decimal

        mov 43h,50h    ;first byte
                mov 42h,51h    ;second byte
                mov 41h,52h    ;third byte
                mov 40h,53h    ;fourth byte

                mov 50h,#00h    ;first byte
                mov 51h,#00h    ;second byte
                mov 52h,#00h    ;third byte
                mov 53h,#00h    ;fourth byte
                mov 54h,#00h    ;fifth byte


                mov r0,#20h
nxtbit
                mov a,40h
                add a,40h
                mov 40h,a

                mov a,41h
                addc a,41h
                mov 41h,a

                mov a,42h
                addc a,42h
                mov 42h,a
```

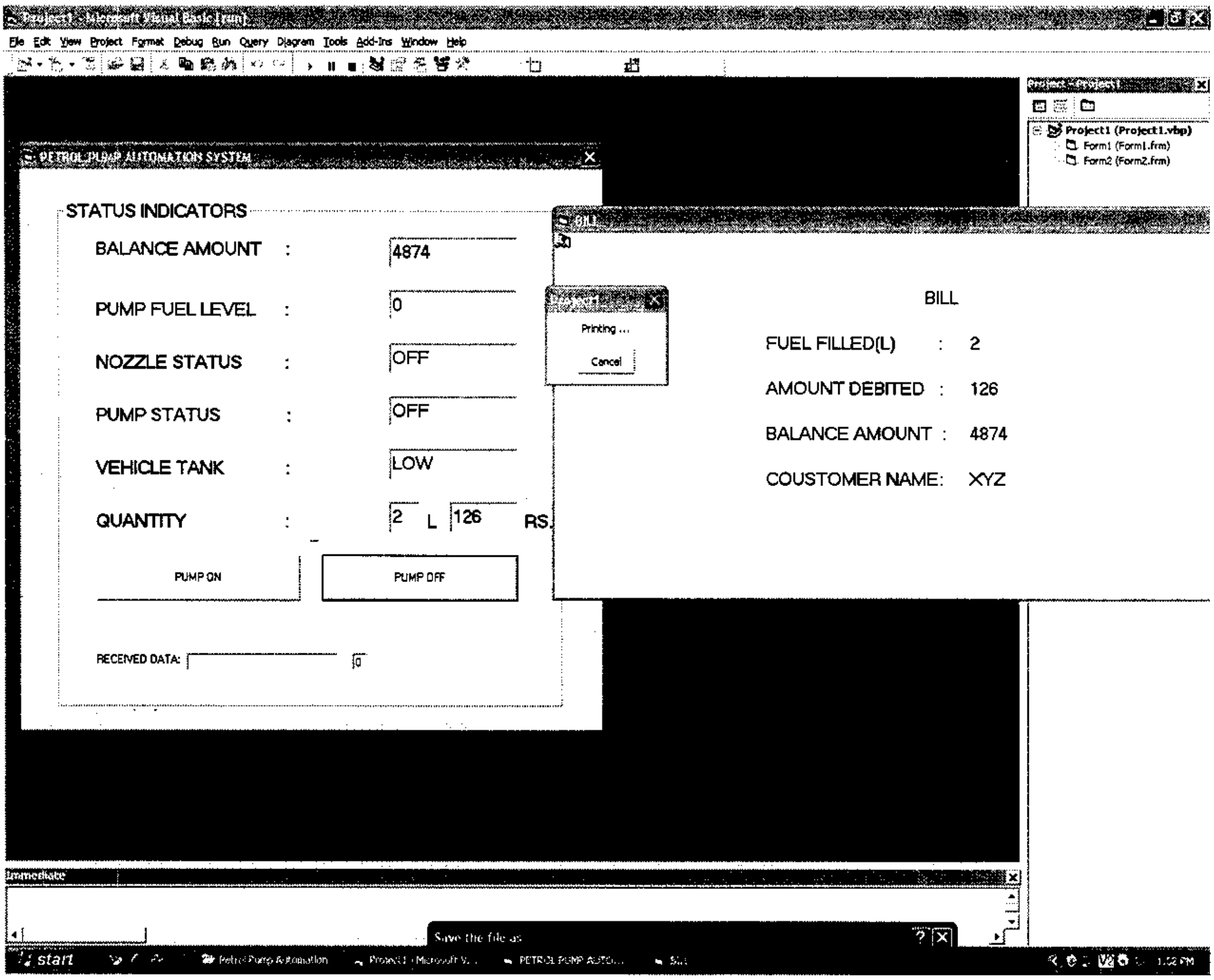
```
                mov a,43h
                addc a,43h
                mov 43h,a

                mov a,54h
                addc a,54h
                da a
                mov 54h,a

                mov a,53h
                addc a,53h
                da a
                mov 53h,a

                mov a,52h
                addc a,52h
                da a
                mov 52h,a

                mov a,51h
                addc a,51h
                da a
                mov 51h,a


                mov a,50h
                addc a,50h
                da a
                mov 50h,a

                djnz r0,nxtbit
                mov r1,50h      ;first byte
                mov r2,51h      ;second byte
                mov r3,52h      ;third byte
                mov r4,53h      ;fourth byte
                mov r5,54h      ;fifth byte

                ret

read_adc
                setb ale
                clr oe
                setb eoc

                clr clk
                nop
                setb clk
```

```
                nop

                clr ale

adc1
                clr clk
                nop
                setb clk
                nop

                jnb eoc,adc1

                setb oe

                clr clk
                nop
                setb clk
                nop

                clr clk
                nop
                setb clk
                nop

                clr clk
                nop
                setb clk
                nop

                mov a,adc

                ret


xmit        mov sbuf,a
xmit1
                jbc ti,treturn
                nop
                sjmp xmit1

treturn     ret



receive

                clr ri
                mov a,sbuf
```

```
                    ret

delay   mov r0,#01h
din2    mov r1,#00h
din1    mov r2,#00h
dwait   djnz r2,dwait
        djnz r1,din1
        djnz r0,din2

        ret

        convert         inc a
                        movc a,@a+pc
                        ret

                        db 30h
                        db 31h
                        db 32h
                        db 33h
                        db 34h
                        db 35h
                        db 36h
                        db 37h
                        db 38h
                        db 39h

                        db 41h
                        db 42h
                        db 43h
                        db 44h
                        db 45h
                        db 46h
```

CHAPTER - 5

5.RESULT



PETROL PUMP AUTOMATION SYSTEM
STATUS INDICATORS
BALANCE AMOUNT : 4874
PUMP FUEL LEVEL : 0
NOZZLE STATUS : OFF
PUMP STATUS : OFF
VEHICLE TANK : LOW
QUANTITY : 2 L 126 RS.
PUMP ON
PUMP OFF
RECEIVED DATA:
Printing ...
Cancel
BILL
FUEL FILLED(L) : 2
AMOUNT DEBITED : 126
BALANCE AMOUNT : 4874
COUSTOMER NAME: XYZ

CHAPTER - 6

6. CONCLUSION:

Hereby in this Petrol Pump Automation using RFID Technology, the data is transmitted from RFID tag kept in the vehicle section to the RFID reader in the petrol pump section. The required amount of fuel is set in the PC. The Microcontroller activates the pump and starts filling the required fuel automatically. Once the required level is reached, Microcontroller deactivates the pump. In the vehicle section, the filled fuel level is displayed in the LCD. According to the fuel filled the corresponding amount will be deducted from the card which is already credited. Finally the bill will be generated automatically. With the help of Microcontroller 89S51 automation of fuel filling, RFID transaction and finally bill generation with respect to the amount of fuel filled in the vehicle can be done successfully.

CHAPTER - 7

7. FUTURE ENHANCEMENT:

This implementation can be further extended by

- Showing the quantity of fuel in litres.
- Encorporating our system in the original fuel tank of automobiles with elimination of linearity problem.

CHAPTER – 8

8.REFERENCES:

1.Mazidi – "Microcontroller & Embedded System, Pearson Education.

2.Garycornell(1999), " Visual Basic 6.0 ", fourth edition Tata Mcgraw Hill.

3.Kenneth j.Ayala(1996), "8051 Microcontroller Architecture,Programming and Application", Tenran International Publishing (India) pvt.ltd.

4.http://www.rfid.tamu.edu.

5.http://www.maxim-ic.com

6.www.atmel.com

7.www.google.com

8.www.electronicsforyou.com

9.www.rs232.com

10.www.electroniccircuits.com

APPENDIX

APPENDIX 1: RS 232 DATASHEET

19-0065; Rev 5; 12/96

MAXIM

+5V RS-232 Transceivers with 0.1µF External Capacitors

General Description

The MAX200-MAX211/MAX213 transceivers are designed for RS-232 and V.28 communication interfaces where ±12V supplies are not available. On-board charge pumps convert the +5V input to the ±10V needed for RS-232 output levels. The MAX201 and MAX209 operate from +5V and +12V, and contain a +12V to -12V charge-pump voltage converter.

The MAX200-MAX211/MAX213 drivers and receivers meet all EIA/TIA-232E and CCITT V.28 specifications at a data rate of 20kbits/sec. The drivers maintain the ±5V EIA/TIA-232E output signal levels at data rates in excess of 120kbits/sec when loaded in accordance with the EIA/TIA-232E specification.

The 5µW shutdown mode of the MAX200, MAX205, MAX206, and MAX211 conserves energy in battery-powered systems. The MAX213 has an active-low shutdown and an active-high receiver enable control. Two receivers of the MAX213 are active, allowing ring indicator (RI) to be monitored easily using only 75µW power.

The MAX211 and MAX213 are available in a 28-pin wide small-outline (SO) package, and a 28-pin shrink small-outline package (SSOP), which occupies only 40% of the area of the SO. The MAX207 is now available in a 24-pin SO package and a 24-pin SSOP. The MAX203 and MAX205 use no external components, and are recommended for applications with limited circuit board space.

Applications

Computers
- Laptops, Palmtops, Notebooks

Battery-Powered Equipment

Features

Superior to Bipolar:
- **0.1µF to 10µF External Capacitors**
- **120kbits/sec Data Rate**
- **2 Receivers Active in Shutdown Mode (MAX213)**
- **Small 28-Pin SSOP Package - Uses 60% Less Space than SOIC**
- **Low-Power Shutdown Current: 1µA**
- **Designed for RS-232 and V.28 Applications**
- **Three-State TTL/CMOS Receiver Outputs**

Ordering Information

PART	TEMP. RANGE	PIN-PACKAGE
MAX200CPP	0°C to +70°C	20 Plastic DIP
MAX200CWP	0°C to +70°C	20 Wide SO
MAX200EPP	-40°C to +85°C	20 Plastic DIP
MAX200EWP	-40°C to +85°C	20 Wide SO
MAX201CPD	0°C to +70°C	14 Plastic DIP
MAX201CWE	0°C to +70°C	16 Wide SO
MAX201C/D	0°C to +70°C	Dice*
MAX201EPD	-40°C to +85°C	14 Plastic DIP
MAX201EWE	-40°C to +85°C	16 Wide SO

Ordering Information continued at end of data sheet.

**Contact factory for dice specifications.*

MAX200-MAX211/MAX213

+5V RS-232 Transceivers with 0.1µF External Capacitors

ABSOLUTE MAXIMUM RATINGS

V_{CC} .... -0.3V to +6V
V+ .... (V_{CC} - 0.3V) to +14V
V- .... +0.3V to -14V
Input Voltages
T_{IN} .... -0.3V to (V_{CC} + 0.3V)
R_{IN} .... ±30V
Output Voltages
T_{OUT} .... (V+ + 0.3V) to (V- - 0.3V)
R_{OUT} .... -0.3V to (V_{CC} + 0.3V)
Short-Circuit Duration
T_{OUT} .... Continuous
Continuous Power Dissipation (T_A = +70°C)
14-Pin Plastic DIP (derate 10.00mW/°C above +70°C) 800mW
16-Pin Plastic DIP (derate 10.53mW/°C above +70°C) 842mW
16-Pin SO (derate 8.70mW/°C above +70°C) 696mW
16-Pin Wide SO (derate 9.52mW/°C above +70°C) 762mW
16-Pin CERDIP (derate 10.00mW/°C above +70°C) 800mW
20-Pin Plastic DIP (derate 11.11mW/°C above +70°C) 889mW
20-Pin Wide SO (derate 10.00mW/°C above +70°C) 800mW
20-Pin CERDIP (derate 11.11mW/°C above +70°C) 889mW
24-Pin Narrow Plastic DIP (derate 13.33mW/°C above +70°C) 1067mW
24-Pin Wide Plastic DIP (derate 9.09mW/°C above +70°C) 727mW
24-Pin Wide SO (derate 11.76mW/°C above +70°C) 941mW
24-Pin SSOP (derate 8.00mW/°C above +70°C) 640mW
24-Pin CERDIP (derate 12.50mW/°C above +70°C) 1000mW
28-Pin Wide SO (derate 12.50mW/°C above +70°C) 1000mW
28-Pin SSOP (derate 9.52mW/°C above +70°C) 762mW
Operating Temperature Ranges:
MAX2_ _C_ _ .... 0°C to +70°C
MAX2_ _E_ _ .... -40°C to +85°C
MAX2_ _M_ _ .... -55°C to +125°C
Storage Temperature Range .... -65°C to +160°C
Lead Temperature (soldering, 10sec) .... +300°C

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS

(MAX202/204/206/208/211/213 V_{CC} = 5V ±10%, MAX200/203/205/207 V_{CC} = 5V ±5%, C1–C4 = 0.1µF, MAX201/MAX209 V_{CC} = 5V ±10%, V+ = 9.0V to 13.2V, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
Output Voltage Swing	All transmitter outputs loaded with 3kΩ to ground		±5	±8		V
V_{CC} Power-Supply Current	No load, T_A = +25°C	MAX202, MAX203		8	15	mA
		MAX200, MAX204-MAX208, MAX211, MAX213		11	20	
		MAX201, MAX209		0.4	1	
V+ Power-Supply Current	No load	MAX201		5	10	mA
		MAX209		7	15	
Shutdown Supply Current	Figure 1, T_A = +25°C	MAX200, MAX205, MAX206, MAX211		1	10	µA
		MAX213		15	50	
Input Logic Threshold Low	T_{IN}, $\overline{EN}$, SHDN, EN, $\overline{SHDN}$				0.8	V
Input Logic Threshold High	T_{IN}		2.0			V
	$\overline{EN}$, SHDN, EN, $\overline{SHDN}$		2.4			
Logic Pull-Up Current	T_{IN} = 0V			15	200	µA
RS-232 Input Voltage Operating Range			-30		+30	V
Receiver Input Threshold Low	V_{CC} = 5V, T_A = +25°C	Active mode	0.8	1.2		V
		Shutdown mode, MAX213, R4, R5	0.6	1.5		
Receiver Input Threshold High	V_{CC} = 5V, T_A = +25°C	Active mode		1.7	2.4	V
		Shutdown mode, MAX213, R4, R5		1.5	2.4	
RS-232 Input Hysteresis	V_{CC} = 5V, no hysteresis in shutdown		0.2	0.5	1.0	V
RS-232 Input Resistance	V_{CC} = 5V, T_A = +25°C		3	5	7	kΩ

ELECTRICAL CHARACTERISTICS (continued)

(MAX202/204/206/208/211/213 V_{CC} = 5V ±10%, MAX200/203/205/207 V_{CC} = 5V ±5%, C1-C4 = 0.1μF, MAX201/MAX209 V_{CC} = 5V ±10%, V+ = 9.0V to 13.2V, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
TTL/CMOS Output Voltage Low	I_{OUT} = 3.2mA (MAX201, MAX202, MAX203), I_{OUT} = 1.6mA (all others)				0.4	V
TTL/CMOS Output Voltage High	I_{OUT} = 1.0mA		3.5			V
TTL/CMOS Output Leakage Current	$\overline{EN}$ = V_{CC}, EN = 0V, 0V ≤ R_{OUT} ≤ V_{CC}			0.05	±10	μA
Output Enable Time (Figure 2)	MAX205, MAX206, MAX209, MAX211, MAX213			600		ns
Output Disable Time (Figure 2)	MAX205, MAX206, MAX209, MAX211, MAX213			200		ns
Receiver Propagation Delay	MAX213	$\overline{SHDN}$ = 0V, R4, R5		4	40	μs
		$\overline{SHDN}$ = V_{CC}		0.5	10	
	MAX200-MAX211			0.5	10	
Transmitter Output Resistance	V_{CC} = V+ = V- = 0V, V_{OUT} = ±2V		300			Ω
Transition Region Slew Rate	C_L = 50pF to 2500pF, R_L = 3kΩ to 7kΩ, V_{CC} = 5V, T_A = +25°C measured from +3V to -3V or -3V to +3V	MAX200, MAX202-MAX211, MAX213	3	5.5	30	V/μs
		MAX201		4	30	
RS-232 Output Short-Circuit Current				±10	±60	mA
Maximum Data Rate	R_L = 3kΩ to 7kΩ, C_L = 50pF to 1000pF, one transmitter		120			kbps

+5V RS-232 Transceivers with 0.1μF External Capacitors

Applications Information

Capacitor Selection

The type of capacitor used is not critical for proper operation. Ceramic capacitors are suggested. To ensure proper RS-232 signal levels over temperature when using 0.1μF capacitors, make sure the capacitance value does not degrade excessively as the temperature varies. If in doubt, use capacitors with a larger nominal value. Also observe the capacitors' ESR (effective series resistance) value over temperature, since it will influence the amount of ripple on V+ and V-. To reduce the output impedance at V+ and V-, use larger capacitors (up to 10μF). If polarized capacitors are used, obey the polarities shown in Figure 1 and the Pin Configuration diagrams.

Driving Multiple Receivers

Each transmitter is designed to drive a single receiver. Transmitters can be paralleled to drive multiple receivers.

Driver Outputs when Exiting Shutdown

Figure 5 shows two driver outputs exiting shutdown. As they become active, the two driver outputs are shown going to opposite RS-232 levels (one driver input is high, the other is low). Each driver is loaded with 3kΩ in parallel with 2500pF. The driver outputs display no ringing or undesirable transients as they come out of shutdown.

Power-Supply Decoupling

In applications that are sensitive to power-supply noise, decouple VCC to ground with a capacitor of the same value as the charge-pump capacitors.

V+ and V- as Power Supplies

A small amount of power can be drawn from V+ and V-, although this will reduce noise margins.

Power Supplies for MAX201/MAX209

If at power-up, the V+ supply rises after the VCC supply, place a diode (e.g. 1N914) in series with the V+ supply.

Table 2. Summary of EIA/TIA-232E, V.28 Specifications

PARAMETER	CONDITION	EIA/TIA-232E, V.28 SPECIFICATION
Driver Output Voltage 0 Level 1 Level Output Level, Max	 3kΩ to 7kΩ load 3kΩ to 7kΩ load No load	 +5.0V to +15V -5.0V to -15V ±25V
Data Rate	$3k\Omega \le R_L \le 7k\Omega$, $C_L \le 2500pF$	Up to 20kbits/sec
Receiver Input Voltage 0 Level 1 Level Input Level, Max		 +3.0V to +15V -3.0V to -15V ±25V
Instantaneous Slew Rate, Max	$3k\Omega \le R_L \le 7k\Omega$, $C_L \le 2500pF$	30V/μs
Driver Output Short-Circuit Current, Max		100mA
Transition Rate on Driver Output	V.28	1ms or 3% of the period
	EIA/TIA-232E	4% of the period
Driver Output Resistance	$-2V < V_{OUT} < +2V$	300Ω

APPENDIX 2: MICROCONTROLLER 89S51 DATASHEET

Programming the Flash – Parallel Mode

The AT89S51 is shipped with the on-chip Flash memory array ready to be programmed. The programming interface needs a high-voltage (12-volt) program enable signal and is compatible with conventional third-party Flash or EPROM programmers.

The AT89S51 code memory array is programmed byte-by-byte.

Programming Algorithm: Before programming the AT89S51, the address, data, and control signals should be set up according to the Flash Programming Modes table (Table 17-1) and Figure 17-1 and Figure 17-2. To program the AT89S51, 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.
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 50 µs. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

$\overline{\text{Data}}$ Polling: The AT89S51 features $\overline{\text{Data}}$ Polling to indicate the end of a byte write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written data on P0.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin. $\overline{\text{Data}}$ Polling may begin any time after a write cycle has been initiated.

Ready/$\overline{\text{Busy}}$: The progress of byte programming can also be monitored by the RDY/$\overline{\text{BSY}}$ output signal. P3.0 is pulled low after ALE goes high during programming to indicate $\overline{\text{BUSY}}$. P3.0 is pulled high again when programming is done to indicate READY.

Programming the Flash – Serial Mode

The Code memory array can be programmed using the serial ISP interface while RST is pulled to V_{CC}. The serial interface consists of pins SCK, MOSI (input) and MISO (output). After RST is set high, the Programming Enable instruction needs to be executed first before other operations can be executed. Before a reprogramming sequence can occur, a Chip Erase operation is required.

The Chip Erase operation turns the content of every memory location in the Code array into FFH.

Either an external system clock can be supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The maximum serial clock (SCK) frequency should be less than 1/16 of the crystal frequency. With a 33 MHz oscillator clock, the maximum SCK frequency is 2 MHz.

Serial Programming Algorithm

To program and verify the AT89S51 in the serial programming mode, the following sequence is recommended:

1. Power-up sequence:
   a. Apply power between VCC and GND pins.
   b. Set RST pin to "H".

If a crystal is not connected across pins XTAL1 and XTAL2, apply a 3 MHz to 33 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.

2. Enable serial programming by sending the Programming Enable serial instruction to pin MOSI/P1.5. The frequency of the shift clock supplied at pin SCK/P1.7 needs to be less than the CPU clock at XTAL1 divided by 16.
3. The Code array is programmed one byte at a time in either the Byte or Page mode. The write cycle is self-timed and typically takes less than 0.5 ms at 5V.
4. Any memory location can be verified by using the Read instruction that returns the content at the selected address at serial output MISO/P1.6.

Serial Programming Instruction Set

Instruction	Instruction Format				Operation
	Byte 1	Byte 2	Byte 3	Byte 4	
Programming Enable	1010 1100	0101 0011	xxxx xxxx	xxxx xxxx 0110 1001 (Output on MISO)	Enable Serial Programming while RST is high
Chip Erase	1010 1100	100x xxxx	xxxx xxxx	xxxx xxxx	Chip Erase Flash memory array
Read Program Memory (Byte Mode)	0010 0000	xxxx A11 A10 A9 A8	A7 A6 A5 A4 A3 A2 A1 A0	D7 D6 D5 D4 D3 D2 D1 D0	Read data from Program memory in the byte mode
Write Program Memory (Byte Mode)	0100 0000	xxxx A11 A10 A9 A8	A7 A6 A5 A4 A3 A2 A1 A0	D7 D6 D5 D4 D3 D2 D1 D0	Write data to Program memory in the byte mode
Write Lock Bits[1]	1010 1100	1110 0B1B2	xxxx xxxx	xxxx xxxx	Write Lock bits. See Note (1).
Read Lock Bits	0010 0100	xxxx xxxx	xxxx xxxx	xx LB3 LB2 LB1 xx	Read back current status of the lock bits (a programmed lock bit reads back as a "1")
Read Signature Bytes	0010 1000	xxxx A11 A10 A9 A8	A7 xxx xxx0	Signature Byte	Read Signature Byte
Read Program Memory (Page Mode)	0011 0000	xxxx A11 A10 A9 A8	Byte 0	Byte 1... Byte 255	Read data from Program memory in the Page Mode (256 bytes)
Write Program Memory (Page Mode)	0101 0000	xxxx A11 A10 A9 A8	Byte 0	Byte 1... Byte 255	Write data to Program memory in the Page Mode (256 bytes)

Note: 1. B1 = 0, B2 = 0 → Mode 1, no lock protection
B1 = 0, B2 = 1 → Mode 2, lock bit 1 activated
B1 = 1, B2 = 0 → Mode 3, lock bit 2 activated
} Each of the lock bit modes need to be activated sequentially before Mode 4 can be executed.

AC Characteristics

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

External Program and Data Memory Characteristics

Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency			0	33	MHz
t_{LHLL}	ALE Pulse Width	127		$2 t_{CLCL}$-40		ns
t_{AVLL}	Address Valid to ALE Low	43		t_{CLCL}-25		ns
t_{LLAX}	Address Hold After ALE Low	48		t_{CLCL}-25		ns
t_{LLIV}	ALE Low to Valid Instruction In		233		$4 t_{CLCL}$-65	ns
t_{LLPL}	ALE Low to $\overline{PSEN}$ Low	43		t_{CLCL}-25		ns
t_{PLPH}	$\overline{PSEN}$ Pulse Width	205		$3 t_{CLCL}$-45		ns
t_{PLIV}	$\overline{PSEN}$ Low to Valid Instruction In		145		$3 t_{CLCL}$-60	ns
t_{PXIX}	Input Instruction Hold After $\overline{PSEN}$	0		0		ns
t_{PXIZ}	Input Instruction Float After $\overline{PSEN}$		59		t_{CLCL}-25	ns
t_{PXAV}	$\overline{PSEN}$ to Address Valid	75		t_{CLCL}-8		ns
t_{AVIV}	Address to Valid Instruction In		312		$5 t_{CLCL}$-80	ns
t_{PLAZ}	$\overline{PSEN}$ Low to Address Float		10		10	ns
t_{RLRH}	$\overline{RD}$ Pulse Width	400		$6 t_{CLCL}$-100		ns
t_{WLWH}	$\overline{WR}$ Pulse Width	400		$6 t_{CLCL}$-100		ns
t_{RLDV}	$\overline{RD}$ Low to Valid Data In		252		$5 t_{CLCL}$-90	ns
t_{RHDX}	Data Hold After $\overline{RD}$	0		0		ns
t_{RHDZ}	Data Float After $\overline{RD}$		97		$2 t_{CLCL}$-28	ns
t_{LLDV}	ALE Low to Valid Data In		517		$8 t_{CLCL}$-150	ns
t_{AVDV}	Address to Valid Data In		585		$9 t_{CLCL}$-165	ns
t_{LLWL}	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	$3 t_{CLCL}$-50	$3 t_{CLCL}$+50	ns
t_{AVWL}	Address to $\overline{RD}$ or $\overline{WR}$ Low	203		$4 t_{CLCL}$-75		ns
t_{QVWX}	Data Valid to $\overline{WR}$ Transition	23		t_{CLCL}-30		ns
t_{QVWH}	Data Valid to $\overline{WR}$ High	433		$7 t_{CLCL}$-130		ns
t_{WHQX}	Data Hold After $\overline{WR}$	33		t_{CLCL}-25		ns
t_{RLAZ}	$\overline{RD}$ Low to Address Float		0		0	ns
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	t_{CLCL}-25	t_{CLCL}+25	ns

DC Characteristics

The values shown in this table are valid for T_A = -40°C to 85°C and V_{CC} = 4.0V to 5.5V, unless otherwise noted.

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage	(Except $\overline{EA}$)	-0.5	0.2 V_{CC}-0.1	V
V_{IL1}	Input Low Voltage ($\overline{EA}$)		-0.5	0.2 V_{CC}-0.3	V
V_{IH}	Input High Voltage	(Except XTAL1, RST)	0.2 V_{CC}+0.9	V_{CC}+0.5	V
V_{IH1}	Input High Voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC}+0.5	V
V_{OL}	Output Low Voltage[1] (Ports 1,2,3)	I_{OL} = 1.6 mA		0.45	V
V_{OL1}	Output Low Voltage[1] (Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA		0.45	V
V_{OH}	Output High Voltage (Ports 1,2,3, ALE, $\overline{PSEN}$)	I_{OH} = -60 μA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -25 μA	0.75 V_{CC}		V
		I_{OH} = -10 μA	0.9 V_{CC}		V
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	I_{OH} = -800 μA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -300 μA	0.75 V_{CC}		V
		I_{OH} = -80 μA	0.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	V_{IN} = 0.45V		-50	μA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	V_{IN} = 2V, V_{CC} = 5V ± 10%		-300	μA
I_{LI}	Input Leakage Current (Port 0, $\overline{EA}$)	0.45 < V_{IN} < V_{CC}		±10	μA
RRST	Reset Pulldown Resistor		50	300	KΩ
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		25	mA
		Idle Mode, 12 MHz		6.5	mA
	Power-down Mode[2]	V_{CC} = 5.5V		50	μA

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

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

APPENDIX 3: ADC DATA SHEET

ADC0808/ADC0809
8-Bit µP Compatible A/D Converters with 8-Channel Multiplexer

General Description

The ADC0808, ADC0809 data acquisition component is a monolithic CMOS device with an 8-bit analog-to-digital converter, 8-channel multiplexer and microprocessor compatible control logic. The 8-bit A/D converter uses successive approximation as the conversion technique. The converter features a high impedance chopper stabilized comparator, a 256R voltage divider with analog switch tree and a successive approximation register. The 8-channel multiplexer can directly access any of 8-single-ended analog signals.

The device eliminates the need for external zero and full-scale adjustments. Easy interfacing to microprocessors is provided by the latched and decoded multiplexer address inputs and latched TTL TRI-STATE® outputs.

The design of the ADC0808, ADC0809 has been optimized by incorporating the most desirable aspects of several A/D conversion techniques. The ADC0808, ADC0809 offers high speed, high accuracy, minimal temperature dependence, excellent long-term accuracy and repeatability, and consumes minimal power. These features make this device ideally suited to applications from process and machine control to consumer and automotive applications. For 16-channel multiplexer with common output (sample/hold port) see ADC0816 data sheet. (See AN-247 for more information.)

Features

- Easy interface to all microprocessors
- Operates ratiometrically or with 5 V_{DC} or analog span adjusted voltage reference
- No zero or full-scale adjust required
- 8-channel multiplexer with address logic
- 0V to 5V input range with single 5V power supply
- Outputs meet TTL voltage level specifications
- Standard hermetic or molded 28-pin DIP package
- 28-pin molded chip carrier package
- ADC0808 equivalent to MM74C949
- ADC0809 equivalent to MM74C949-1

Key Specifications

- Resolution: 8 Bits
- Total Unadjusted Error: ±½ LSB and ±1 LSB
- Single Supply: 5 V_{DC}
- Low Power: 15 mW
- Conversion Time: 100 µs

Absolute Maximum Ratings (Notes 2, 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

Supply Voltage (V_{CC}) (Note 3)	6.5V
Voltage at Any Pin	−0.3V to (V_{CC}+0.3V)
Except Control Inputs	
Voltage at Control Inputs	−0.3V to +15V
(START, OE, CLOCK, ALE, ADD A, ADD B, ADD C)	
Storage Temperature Range	−65°C to +150°C
Package Dissipation at T_A=25°C	875 mW
Lead Temp. (Soldering, 10 seconds)	
Dual-In-Line Package (plastic)	260°C
Dual-In-Line Package (ceramic)	300°C
Molded Chip Carrier Package	
Vapor Phase (60 seconds)	215°C
Infrared (15 seconds)	220°C
ESD Susceptibility (Note 8)	400V

Operating Conditions (Notes 1, 2)

Temperature Range (Note 1)	T_{MIN}≤T_A≤T_{MAX}
ADC0808CCN,ADC0809CCN	−40°C≤T_A≤+85°C
ADC0808CCV, ADC0809CCV	−40°C ≤ T_A ≤ +85°C
Range of V_{CC} (Note 1)	4.5 V_{DC} to 6.0 V_{DC}

Electrical Characteristics

Converter Specifications: V_{CC}=5 V_{DC}=V_{REF+}, $V_{REF(-)}$=GND, T_{MIN}≤T_A≤T_{MAX} and f_{CLK}=640 kHz unless otherwise stated.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
	ADC0808					
	Total Unadjusted Error	25°C			±½	LSB
	(Note 5)	T_{MIN} to T_{MAX}			±¾	LSB
	ADC0809					
	Total Unadjusted Error	0°C to 70°C			±1	LSB
	(Note 5)	T_{MIN} to T_{MAX}			±1¼	LSB
	Input Resistance	From Ref(+) to Ref(−)	1.0	2.5		kΩ
	Analog Input Voltage Range	(Note 4) V(+) or V(−)	GND−0.10		V_{CC}+0.10	V_{DC}
$V_{REF(+)}$	Voltage, Top of Ladder	Measured at Ref(+)		V_{CC}	V_{CC}+0.1	V
$\frac{V_{REF(+)}+V_{REF(-)}}{2}$	Voltage, Center of Ladder		V_{CC}/2-0.1	V_{CC}/2	V_{CC}/2+0.1	V
$V_{REF(-)}$	Voltage, Bottom of Ladder	Measured at Ref(−)	−0.1	0		V
I_{IN}	Comparator Input Current	f_c=640 kHz, (Note 6)	−2	±0.5	2	μA

Electrical Characteristics

Digital Levels and DC Specifications: ADC0808CCN, ADC0808CCV, ADC0809CCN and ADC0809CCV, 4.75≤V_{CC}≤5.25V, −40°C≤T_A≤+85°C unless otherwise noted

Symbol	Parameter	Conditions	Min	Typ	Max	Units
ANALOG MULTIPLEXER						
$I_{OFF(+)}$	OFF Channel Leakage Current	V_{CC}=5V, V_{IN}=5V,				
		T_A=25°C		10	200	nA
		T_{MIN} to T_{MAX}			1.0	μA
$I_{OFF(-)}$	OFF Channel Leakage Current	V_{CC}=5V, V_{IN}=0,				
		T_A=25°C	−200	−10		nA
		T_{MIN} to T_{MAX}	−1.0			μA
CONTROL INPUTS						
$V_{IN(1)}$	Logical "1" Input Voltage		V_{CC}−1.5			V
$V_{IN(0)}$	Logical "0" Input Voltage				1.5	V
$I_{IN(1)}$	Logical "1" Input Current (The Control Inputs)	V_{IN}=15V			1.0	μA
$I_{IN(0)}$	Logical "0" Input Current (The Control Inputs)	V_{IN}=0	−1.0			μA
I_{CC}	Supply Current	f_{CLK}=640 kHz		0.3	3.0	mA

Electrical Characteristics (Continued)

Digital Levels and DC Specifications: ADC0808CCN, ADC0808CCV, ADC0809CCN and ADC0809CCV, $4.75 \le V_{CC} \le 5.25V$, $-40°C \le T_A \le +85°C$ unless otherwise noted

Symbol	Parameter	Conditions	Min	Typ	Max	Units
DATA OUTPUTS AND EOC (INTERRUPT)						
$V_{OUT(1)}$	Logical "1" Output Voltage	V_{CC} = 4.75V I_{OUT} = −360µA I_{OUT} = −10µA		 **2.4** **4.5**		 V(min) V(min)
$V_{OUT(0)}$	Logical "0" Output Voltage	I_O=1.6 mA			0.45	V
$V_{OUT(0)}$	Logical "0" Output Voltage EOC	I_O=1.2 mA			0.45	V
I_{OUT}	TRI-STATE Output Current	V_O=5V V_O=0	 −3		3	µA µA

Electrical Characteristics

Timing Specifications $V_{CC}=V_{REF(+)}$=5V, $V_{REF(-)}$=GND, $t_r=t_f$=20 ns and T_A=25°C unless otherwise noted.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
t_{WS}	Minimum Start Pulse Width	(*Figure 5*)		100	200	ns
t_{WALE}	Minimum ALE Pulse Width	(*Figure 5*)		100	200	ns
t_s	Minimum Address Set-Up Time	(*Figure 5*)		25	50	ns
t_H	Minimum Address Hold Time	(*Figure 5*)		25	50	ns
t_D	Analog MUX Delay Time From ALE	R_S=0Ω (*Figure 5*)		1	2.5	µs
t_{H1}, t_{H0}	OE Control to Q Logic State	C_L=50 pF, R_L=10k (*Figure 8*)		125	250	ns
t_{1H}, t_{0H}	OE Control to Hi-Z	C_L=10 pF, R_L=10k (*Figure 8*)		125	250	ns
t_c	Conversion Time	f_c=640 kHz, (*Figure 5*) (Note 7)	90	100	116	µs
f_c	Clock Frequency		10	640	1280	kHz
t_{EOC}	EOC Delay Time	(*Figure 5*)	0		8+2 µS	Clock Periods
C_{IN}	Input Capacitance	At Control Inputs		10	15	pF
C_{OUT}	TRI-STATE Output Capacitance	At TRI-STATE Outputs		10	15	pF

Functional Description

Multiplexer. The device contains an 8-channel single-ended analog signal multiplexer. A particular input channel is selected by using the address decoder. *Table 1* shows the input states for the address lines to select any channel. The address is latched into the decoder on the low-to-high transition of the address latch enable signal.

TABLE 1.

SELECTED ANALOG CHANNEL	ADDRESS LINE		
	C	B	A
IN0	L	L	L
IN1	L	L	H
IN2	L	H	L
IN3	L	H	H
IN4	H	L	L
IN5	H	L	H
IN6	H	H	L
IN7	H	H	H

CONVERTER CHARACTERISTICS

The Converter

The heart of this single chip data acquisition system is its 8-bit analog-to-digital converter. The converter is designed to give fast, accurate, and repeatable conversions over a wide range of temperatures. The converter is partitioned into 3 major sections: the 256R ladder network, the successive approximation register, and the comparator. The converter's digital outputs are positive true.

The 256R ladder network approach (*Figure 1*) was chosen over the conventional R/2R ladder because of its inherent monotonicity, which guarantees no missing digital codes. Monotonicity is particularly important in closed loop feedback control systems. A non-monotonic relationship can cause oscillations that will be catastrophic for the system. Additionally, the 256R network does not cause load variations on the reference voltage.

The bottom resistor and the top resistor of the ladder network in *Figure 1* are not the same value as the remainder of the network. The difference in these resistors causes the output characteristic to be symmetrical with the zero and full-scale points of the transfer curve. The first output transition occurs when the analog signal has reached +½ LSB and succeeding output transitions occur every 1 LSB later up to full-scale.

The successive approximation register (SAR) performs 8 iterations to approximate the input voltage. For any SAR type converter, n-iterations are required for an n-bit converter. *Figure 2* shows a typical example of a 3-bit converter. In the ADC0808, ADC0809, the approximation technique is extended to 8 bits using the 256R network.

The A/D converter's successive approximation register (SAR) is reset on the positive edge of the start conversion (SC) pulse. The conversion is begun on the falling edge of the start conversion pulse. A conversion in process will be interrupted by receipt of a new start conversion pulse. Continuous conversion may be accomplished by tying the end-of-conversion (EOC) output to the SC input. If used in this mode, an external start conversion pulse should be applied after power up. End-of-conversion will go low between 0 and 8 clock pulses after the rising edge of start conversion.

The most important section of the A/D converter is the comparator. It is this section which is responsible for the ultimate accuracy of the entire converter. It is also the comparator drift which has the greatest influence on the repeatability of the device. A chopper-stabilized comparator provides the most effective method of satisfying all the converter requirements.

The chopper-stabilized comparator converts the DC input signal into an AC signal. This signal is then fed through a high gain AC amplifier and has the DC level restored. This technique limits the drift component of the amplifier since the drift is a DC component which is not passed by the AC amplifier. This makes the entire A/D converter extremely insensitive to temperature, long term drift and input offset errors.

Figure 4 shows a typical error curve for the ADC0808 as measured using the procedures outlined in AN-179.

Functional Description (Continued)

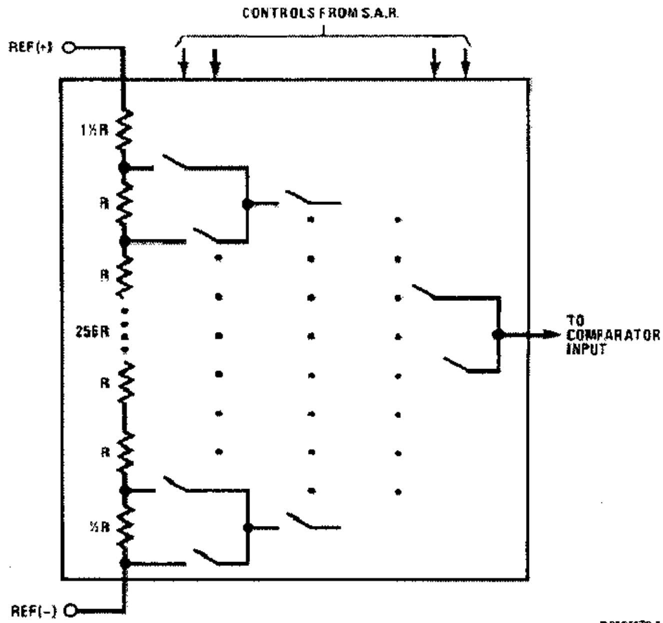


FIGURE 1. Resistor Ladder and Switch Tree

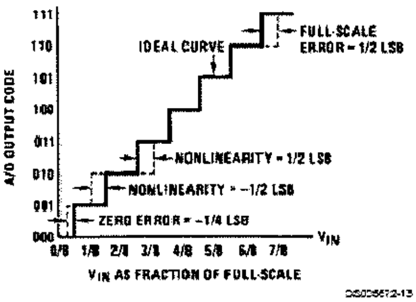


FIGURE 2. 3-Bit A/D Transfer Curve

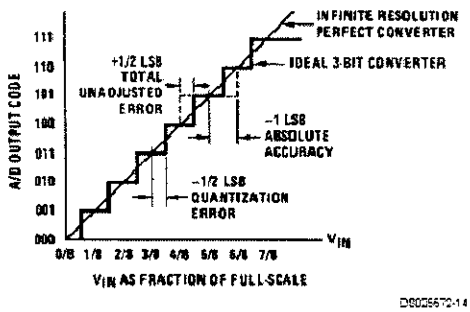


FIGURE 3. 3-Bit A/D Absolute Accuracy Curve

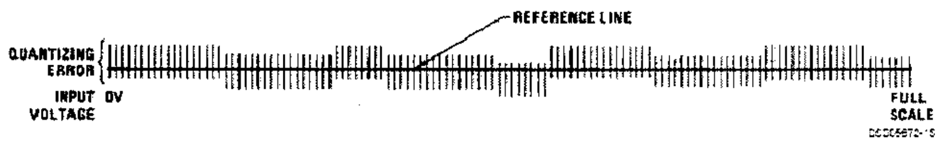


FIGURE 4. Typical Error Curve