PC BASED DATA ACQUISITION SYSTEM

SUBMITTED BY:-

SANJEEV KUAMR (10607002)

SAYED MOHAMMAD (10607010)

UNDER GUIDANCE OF:-

Prof. U.C.PATI

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGG.

NATIONAL INSTITUTE OF TECHNOLOGY ROURKELA-769008

Electronics & Communication Department
National Institute of Technology, Rourkela

CERTIFICATE

This is to certify that the thesis entitled, "**PC BASED DATA ACQUISITION SYSTEM "**by **SANJEEV KUMAR** and **SAYED MOHAMMAD** in partial fulfillment of the requirements for the award of Bachelor of Technology Degree in Electronics & Instrumentation Enggering at the National Institute of Technology, Rourkela (Deemed University) is an authentic operate carried out by them under my supervision.

And to the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University/Institute for the award of any Degree or Diploma.

Prof.U .C PATI

Date: Electronics & Communications Engineering

National Institute of Technology, Rourkela
India

Acknowledgements

My first and foremost regards for Prof. U.C Pati , without whose constant supervision and guidance this project would never have been a reality. He has been a guide par excellence whose academic mentorship and unflinching moral support kept us motivated and driven, even at the most difficult times when giving up seemed the best choice. The confidence and faith he reposed on us has indeed been instrumental in improving ourselves at each and every step of the project and has made it a significant learning experience. We are indebted to Prof. U.C Pati for all his invaluable aegis.

It wouldn't have been ever possible to accomplish a project of this scale without the references and ideas taken from others. They also deserve due credit and veneration.

We also thank the DYNALOG Company Personnels Mr. Sachin V. Koravi, Mr. Hemant Sharma, without whose help this project would have been a distant dream.

Sayed Sahazuddin Mohammad

Sanjeev Kumar

Dept. of Electronics & Communication Engineering

National Institute of Technology, Rourkela

Contents

ABSTRACT

DAQ (Data AcQuisition) is simply the process of bringing a real-world signal, such as avoltage, into the computer, for processing, analysis, storage or other data manipulation. A Physical phenomena represents the real-world signal you are trying to measure. In order to optimize the characteristics of a system in terms of performance, handling

capacity and cost, the relevant subsystem can be combined together. Analog data is generally acquired and transformed into the digital form for the purpose of processing, transmission and display. Rapid advances in Personal Computer (PC) hardware and software technologies have resulted in easy and efficient adoption of PCs in various precise measurement and complex control applications. A PC based measurement or control application requires conversion of real world analog signal into digital format and transfer of digitized data into the PC. A data acquisition system that performs conversion of analog signal to digital data and the digital data to analog signal is interfaced to a pc to implement the functions of a measurement and control instrumentation applications. In this project we have used the electromagnetic sensor to acquire the data of a magnetic disk angular velocity, which we have got in mili volts range. This has been further transformed approximately into the range of 5 volt by using an operational amplifier of suitable gain (~30)and then rectified .We then transformed the analog voltage into digital by using ADC 0804 and the processing part is done by using ATMEL89c51.In the second phase we have used the data acquisition card PCL-208 and 207 to interface the amplified output to PC by the help TURBO C (C compiler).

CHAPTER . 1

INTRODUCTION TO DATA ACQUISITION

1.1 :DATA ACQUISITION SYSTEM

DAQ (Data AcQuisition) is defined as the process of taking a real-world signal as input, such as avoltage or current any electrical input, into the computer, for processing, analysis, storage or other data manipulation or conditioning.A Physical phenomena represents the real-world signal we are trying to measure. Today, most scientists and engineers are using personal computers with ISA, EISA, PCI or PCMCIA bus for data acquisition in laboratory, research, test and measurement, and industrial automation applications. Many applications use plug-in boards to acquire data and transfer it directly to computer memory. Others use DAQ hardware remote from the PC that is coupled via parallel port, serial port, GPIB-Bus or Netoperate. Typically, DAQ plug-in boards are general-purpose data acquisition device that are well suited for measuring voltage signals. However, many real-world sensors and transducers output signals that must be conditioned before a DAQ board or device can effectively and correctly acquire the signal. This front-end preprocessing, which is generally referred to as signal conditioning, includes functions such as signal amplification, filtering, electrical isolation,and multiplexing. Afterall, many transducers require excitation currents or voltages, bridge completion, linearization, or high amplification for proper and accurate operation.

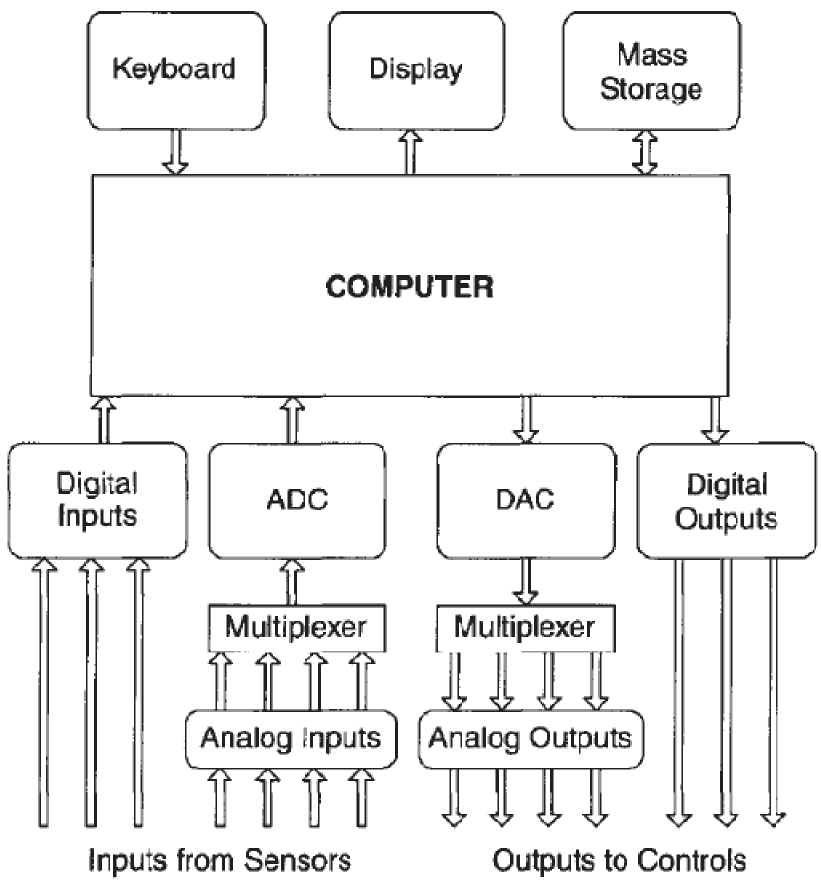


Fig 1.1 :Simplified Block Diagram of a DATA ACQUISITION SYSTEM

1.2 :The PC as a platform for data acquisition

The field of data acquisition encompasses a very wide range of activities. At its simplest level, it involves reading electrical signals into a computer from some form of sensor. These signals may represent the state of a physical process, such as the position and orientation of machine tools, the temperature of a furnace or the size and shape of a manufactured component. The acquired data may have to be stored, printed or displayed. Often the data have to be analyzed or processed in some way in order to generate further signals for controlling external equipment or for interfacing to other computers. This may involve manipulating only static readings, but it is also frequently necessary to deal with time-varying signals as well. Some systems may involve data to be gathered slowly, over time spans of many days or weeks. Other will necessitate short bursts of very high speed data acquisition – perhaps at rates of up to several thousand readings per second. DAQ *is* used widely for laboratory automation, industrial monitoring and control, as well as in a variety of other time-critical applications. The most central reason for using the PC for data acquisition and control is that there is now a large and expanding pool of programmers, engineers and scientists who are familiar with the PC.Indeed it is quite likely that many of these personnel will have learnt how to program on an IBM PC or PC clone.

1.3 :Input/output ports

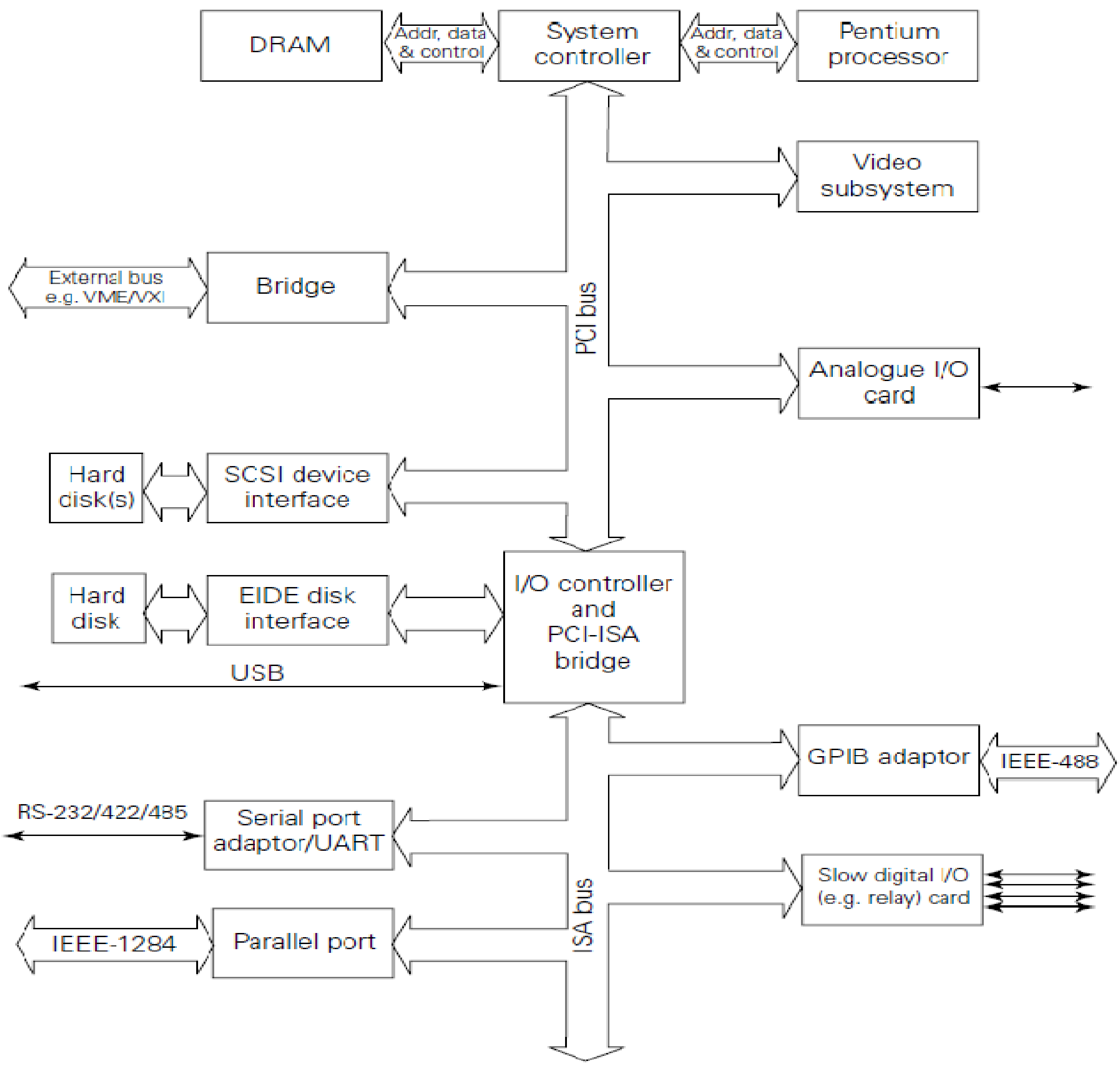
In addition to its memory, the PC has another entirely separate address space. This is dedicated to transferring data to or from marginal devices and is known as Input/Output space (or simply I/O space). Just as the PC's memory space is divided into separate byte locations, the I/O space consists of many byte-sized I/O ports. Each port is addressable in much the same way as memory, although an additional control line is used within the PC to differentiate between memory and I/O port accesses. I/O space consists of a contiguous series of I/O addresses. Unlike memory space, the I/O address space is not segmented and cannot be paged. In fact, the processor references I/O ports by means of a 16-bit address and this means that no more than 65 536 I/O ports can be supported by the PC. In practice, this is added partial by the I/O address decoding scheme used on the PC and its adaptor cards.The I/O ports provide a means of sending data to, and receiving data from, devices such as the video adaptor, the disk subsystem, or analogue-to-digital transformers (ADCs) on plug-in data-acquisition cards. Software can use the assembly language IN or OUT instructions,or their high level language counterparts, to communicate with hardware devices via the I/O ports. These are discussed in more detail in Chapter 6, but for the moment we will consider a simple example. Suppose that a plug-in 8-bit ADC card possesses control and data registers that are each mapped to one of the PC's I/O ports. The software starts the analogue-to-digital exchange process by writing a bit pattern to the I/O port that maps to the ADC card's control register. When the ADC has finished the conversion it might set a bit (known as the End of Conversion, or EOC, bit) in another register to indicate that digitized data is now available.In this way, the software is able to detect the EOC bit by reading the corresponding I/O port.

1.4 :I/O port allocation

Hardware devices map their registers to specific I/O ports simply by decoding the PC's address bus and control lines. In this way, a specific mixture of address and control lines is needed to cause data to be transferred from the register to the PC's data bus or vice versa. Some I/O ports can only be read or written, while others are capable of bidirectional data transfer. Whether ports are read-only (R/O), write-only (W/O) or read-write (R/W) is determined by how the hardware decodes the address and control lines.

1.5 : Buses and adaptor card slots

Passing data to and from a DA&C card via an I/O port actually involves transferring the data over one or more system buses. A typical PC may not contain all of the buses shown, although the PCI and ISA buses are present in most systems. Other types of bus (many of them proprietary systems) can be interfaced by means of special adaptors or bridges to the PC. The IEEE-488 bus and the VXI bus, for example, are used in specialized instrumentation applications. Of primary concern here though are the PC's native buses – i.e. the ones that are an integral part of the PC's own architecture. The type of bus used within the PC not only has a bearing on the type of interface card that can be connected, it may also have a profound effect on the throughput of the system as a whole.



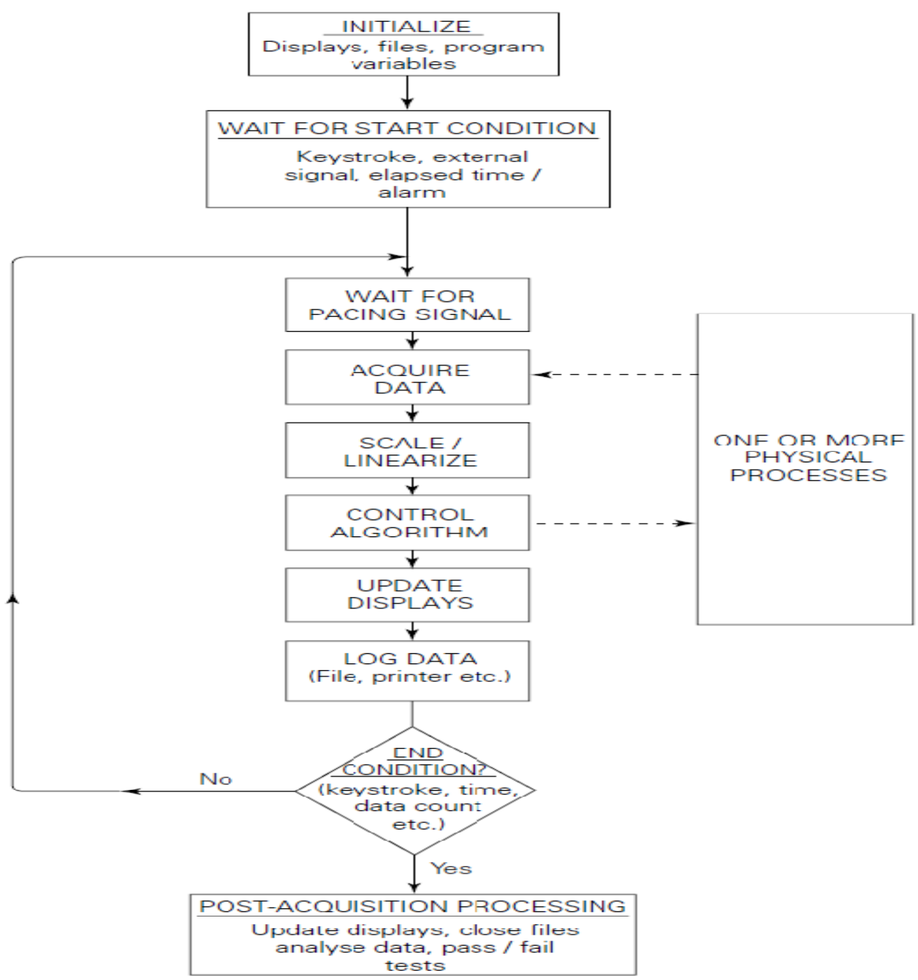
bus connections and interfaces on a PC used for data acquisition. Note that not all devices and buses shown will be present on every system, and some systems will incorporate additional devices

1.6 :Software Considerations

A typical DA&C program may contain the following modules and facilities:

- program configuration routines
- diagnostics modules
- system maintenance and calibration modules
- run-time modules
- device drivers
  data analysis modules.

TYPICAL DAQ PROGRAMMING ALGORITHM



CHAPTER .2

TRANSDUCERS

2.**Sensor**

Sensors are the primary input element involved in reading physical quantities (such as temperature, force or position) into a DA&C system. They are generally used to measure analogue signals although the term ‘sensor’ does in fact encompass some digital devices such as proximity switches. In this section we will deal only with sensing analogue signals.Analogue signals can be measured with sensors that generate
either analogue or digital representations of the quantity to be measured (the measurand).

2.1 :Digital tachometers

A tachometer (also called a revolution-counter, rev-counter, or RPM gauge) is an instrument that measures the rotation speed of a shaft or disk, as in a motor or other machine. The device usually displays the revolutions per minute(RPM) on a calibrated analogue dial, but digital displays are ever more common. Measurement resolution is governed by the number of marks around the circumference. Various types of sensor are used, such as optical, inductive and magnetic ones. As every mark is sensed, a pulse is generated and input to an electronic pulse ounter. Usually, velocity is calculated in terms of the pulse count in unit time, which of course only yields information about the mean velocity. If the velocity is changing, immediate velocity can be designed at each instant of time that an output pulse occurs, using the scheme shown in Figure .In this circuit, the pulses from the transducer gate the train of pulses from a 1MHz clock into a counter. Control logic resets the counter and updates the digital output value after receipt of each pulse from the transducer. The measurement accuracy of this system is highest when the speed of rotation is low.

2.2 :Optical sensor

Optical Tachometer also called hand held stroboscope is used for the study of rotating, reciprocating, oscillating or vibrating machine parts and vibrating strings. Optical pulses can be generated by one of the two alternatives photoelectric techniques illustrated in Figure, the pulses are produced as the windows in a slotted disc pass in sequence between a light source and a detector. The alternative form has both light source and detector mounted on the identical side of a reflective disc which has black sectors painted onto it at regular angular intervals. Light sources
are normally either lasers or LEDs, with photodiodes and phototransistors being used as detectors.Optical tachometers yield improved accuracy than other forms of digital tachometer but are not as reliable because dust and dirt can block light paths.

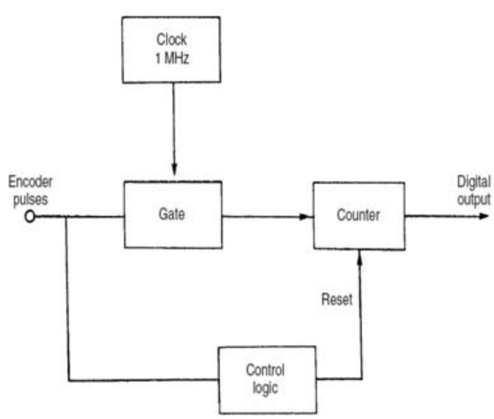


Fig 2.1 Optical sensor block diagram

2.4 :Inductive sensor

An inductive sensor variable reluctance sensor (VRS) is used to specify position and speed of metal components that are in motion . This VRS sensor consists of a permanent magnet, a ferromagnetic pole piece, a pickup coil, and a turning toothed wheel. As the gear teeth of the rotating wheel (or other target features) pass by the face of the magnet, the quantity of magnetic flux passing through the magnet and as a resultof which the coil varies. When the gear tooth is close to the sensor, the flux is at a highest value. When the tooth is more away, the flux quantity decreases off. The distressing target results in a time varying flux that induces a comparative voltage in the coil. Subsequent electronics are then used to route this signal to get a digital waveform that can be more readily counted and timed. A permanent magnet is an object made from a stuff that is magnetized and creates its own persistent magnetic field. An everyday example is a refrigerator magnet used to hold notes on a refrigerator door.In its most basic form, a VRS sensor consists of a coil of wire wrapped around a magnet.As gear teeth (or other target features) pass by the face of the magnet, they cause the quantity of magnetic flux passing through the magnet, and consequently the coil, to vary.When a target feature (such as a gear tooth) is moved close to the sensor, the flux is at a maximum. When the goal is further away, the flux drops off. As the target moves, this results in a time-varying flux that induces a proportional voltage in the coil. Subsequent electronics are then used to massage this signal to get a digital waveform that can be more readily counted and timed. One area in which VR sensors excel, however, is in high-temperature applications.The device operational temperature is restricted by the character of the materials used in the device, with proper construction VRS sensors can be made to operate at temperatures in excess of 300°C. An example of such an extreme application is sensing the turbine speed of a jet engine. VR sensor interface circuits VR sensors need waveform shaping for their output to be digitally readable. The normal output of a VR sensor is an analog signal, shaped much like a sine wave. The frequency and amplitude of the analog signal is proportional to the target's velocity.

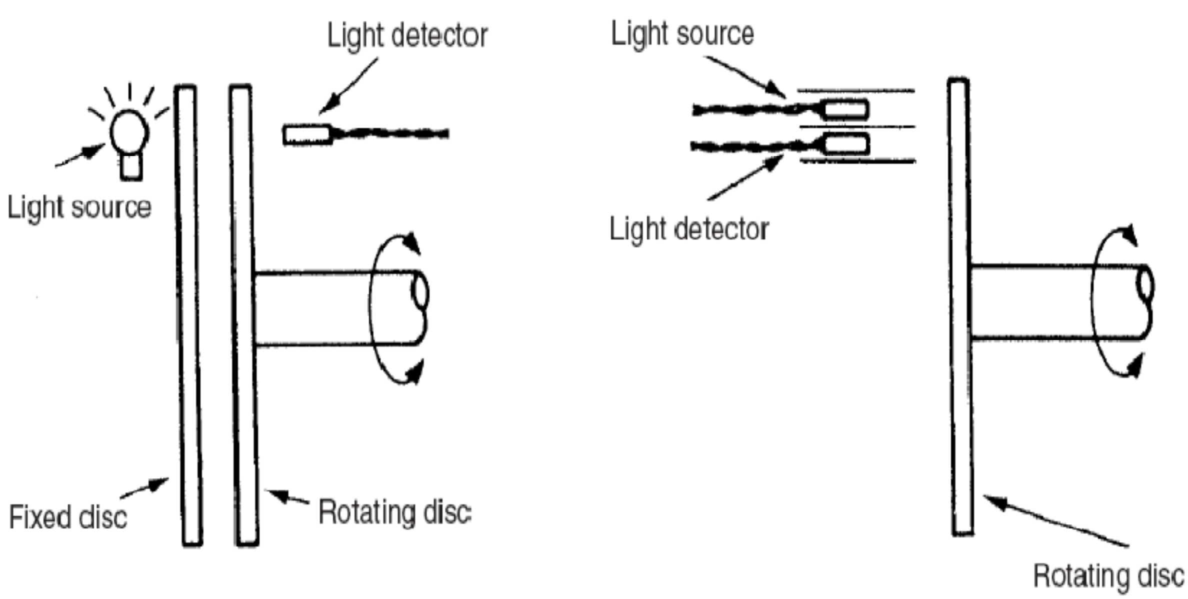
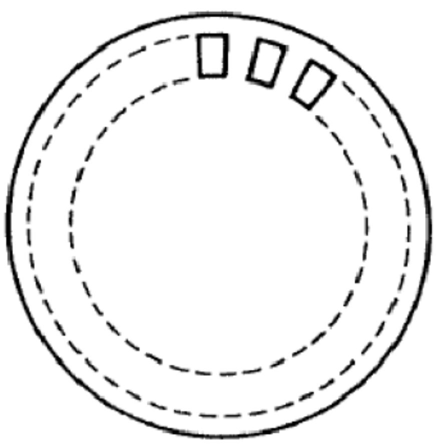


Fig 2.2 Optical sensor



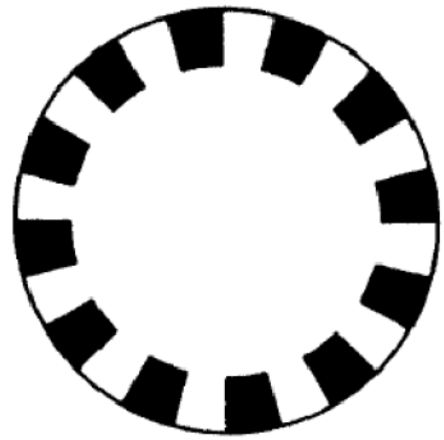


Fig 2.3 electromagnetic sensors

2.5 :Output of the tachometer

E=bm*wr* sin (m*wr*t)

b- amplitude of flux variation

m- number of teeth in the rotating wheel

wr -angular velocity of the turning wheel

Thus sinusoidal signal amplitude E= bm*wr* and frequency f=m*wr*/2π are proportional to angular velocity of wheel.

However this output voltage comes in the range of mili volt . But the necessity of Data acquisition card is that input should be in the range of +/- 5 volt. So it need to be amplified preceding to inputting it into Data acquisition card.

CHAPTER . 3

SIGNAL CONDITIONING

3.1 :Amplifier

The output voltage from the sensor would be fed to the amplifying circuit input .The output of the sensor is in mill volt so we need to amplify it to fetch it to the input range of Pcl card 208.After the voltage getting amplified it would get noise in the process for which it has to pass through signal conditioning element .After that the voltage would be fed into input of Pcl card.

As well as resistors and capacitors, **Operational Amplifiers**, or **Op-amps** as they are more commonly called, are one of the basic building blocks of Analogue Electronic Circuits. It is a linear device that has all the properties required for nearly ideal DC amplification and is used extensively in signal conditioning, filtering or to perform mathematical operations such as add, subtract, integration and differentiation. An ideal **Operational Amplifier** is basically a 3-terminal device that consists of two high impedance inputs, one an **Inverting input** marked with a negative sign, ("-") and the other a **Non-inverting input** marked with a positive plus sign ("+"). The amplified output signal of an Operational Amplifier is the difference between the two signals being applied to the two inputs.

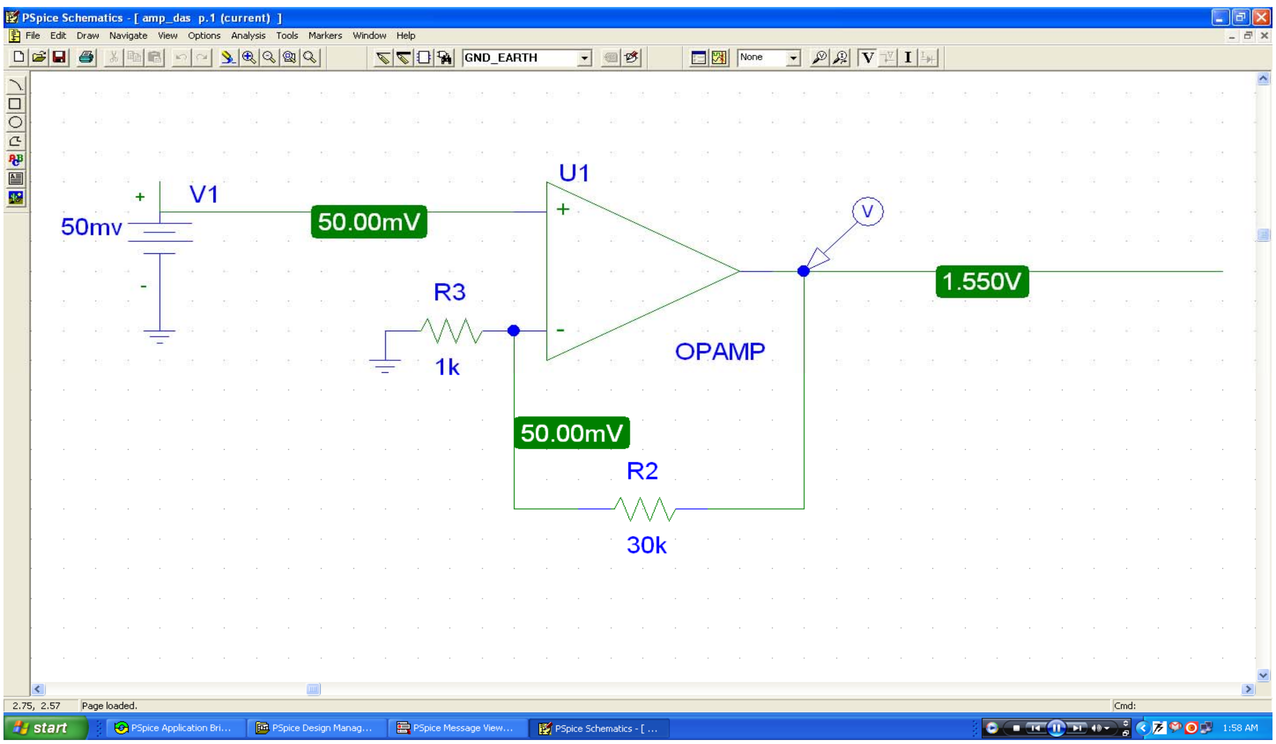


Fig 3.1 :Circuit diagram of amplifying circuit

3.3 :Calculation of Amplifier Gain:

Input range: 53 mv to 238 mv

Output range: 0 to 10 v

Required gain : 30

R2=1k

R3=30k

GAIN= Vout/ Vin

= 1+R3/R2

= 1+30/1

=31

Table 3.1

SL No.	RPM	Tachometer output (mv)	Amplifier output (v)
1	53	53	1.03
2	109	71	2.13
3	134	84	2.62
4	150	109	2.94
5	162	123	3.17
6	178	134	3.49
7	190	150	3.72
8	202	162	3.96
9	223	178	4.37
10	232	190	4.54
11	245	200	4.8
12	251	213	5

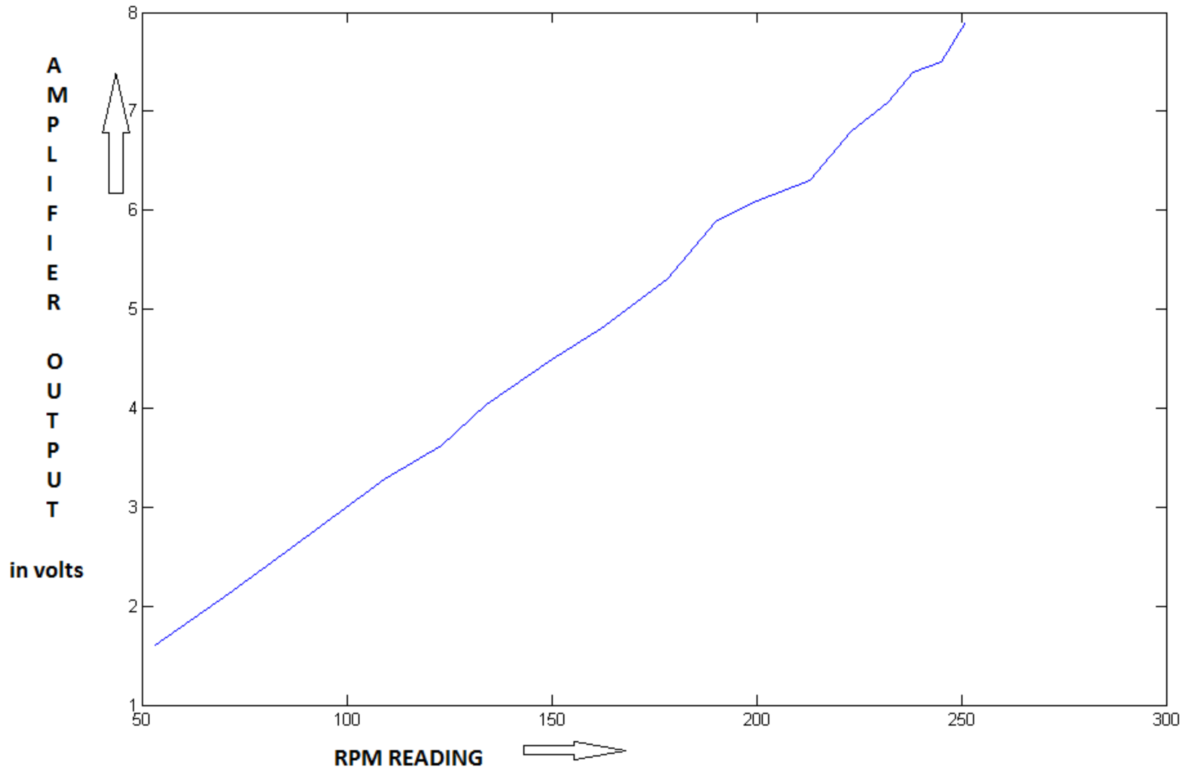


Fig 3.2 : amplifier output vs RPM reading graph

Output we get after amplification is sine wave(a.c) . PCL-208 accept dc values only , hence we need rectifiers to transform ac to dc

.

3.4 :Rectifier

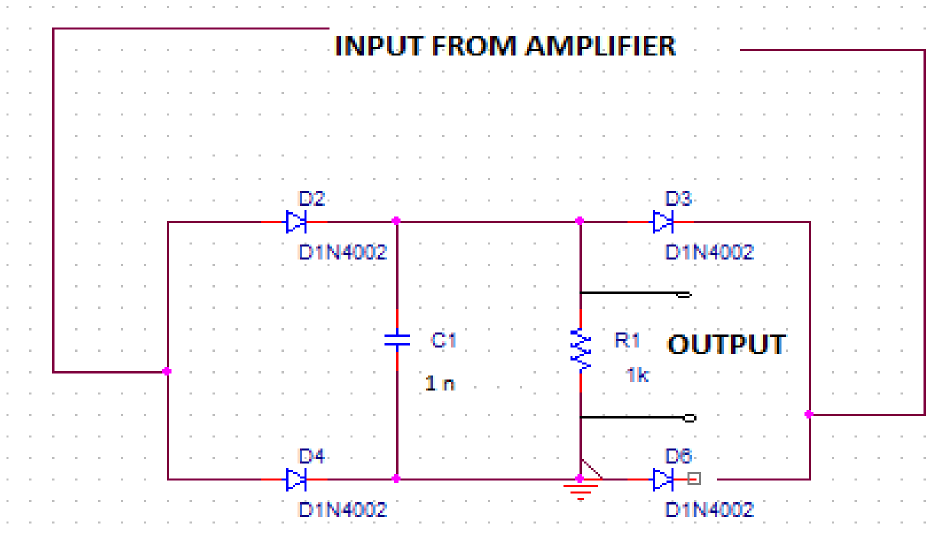


Fig 3.3 CIRCUIT DIAGRAM FOR RECTIFIER

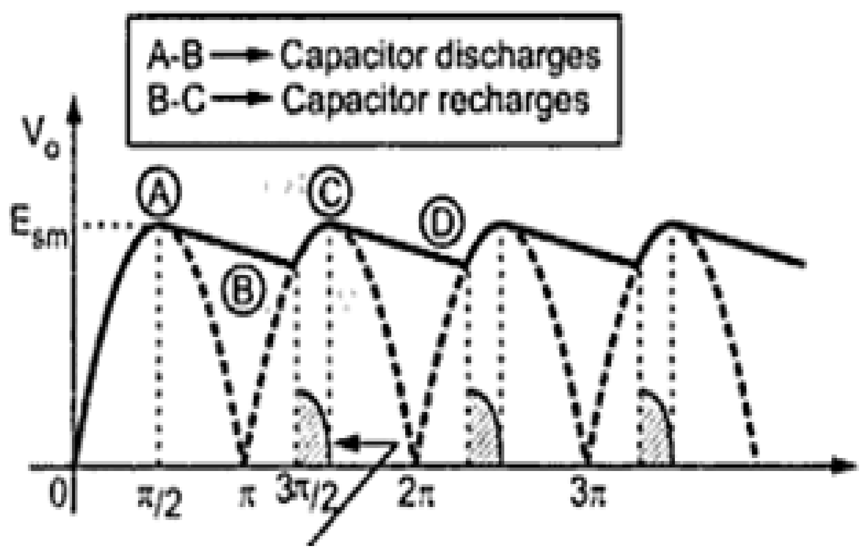


Fig 3.4 Output of the rectifier

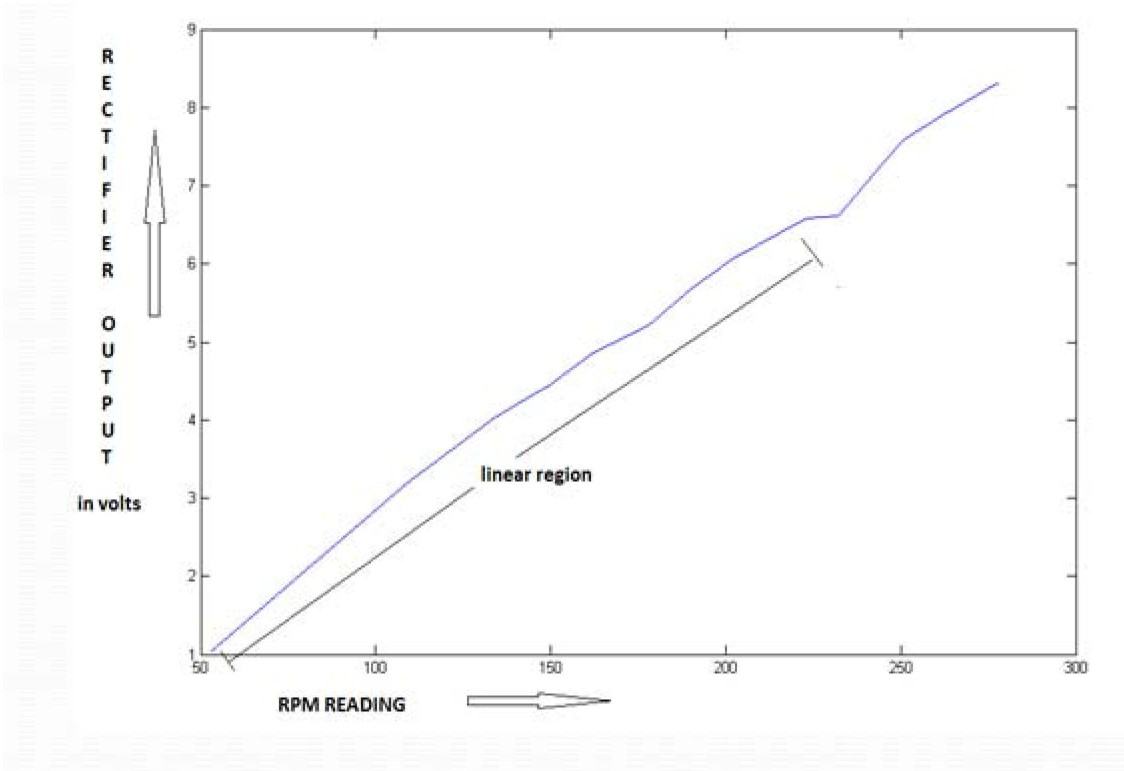
Time constant of the RC filter used shud be much lesser than the time period of input

finput << 1/RC

50 Hz << 10^6 Hz

Table 3.2

SL No.	RPM	Output of rectifier (v)
1	53	1.03
2	109	2.13
3	134	2.62
4	150	2.94
5	162	3.17
6	178	3.49
7	190	3.72
8	202	3.96
9	223	4.37
10	232	4.54
11	245	4.8
12	251	5



<u>Fig 3.5 rectifiers output vs RPM reading</u>

Graph is liner for RPM upto 250 so we program accordingly to merasure RPM value till that point

CHAPTER .4

PCL 208

4.1 :PCL 208 KEY FEATURES:

• It has16 single-ended analog input channels.
• An industrial standard 12-bit successive approximation transformer(HADC574Z) to transform analog inputs is used. The maximum A/D sampling rate is 30 KHz in DMA mode.
• Analog input ranges Software programmable.
• Bipolar voltage values available: +/- 5V, +/- 2.5 V, +/- 1.25V +/- 0.625 V +/- 0.3125 V
• Three A/D trigger modes:
  • Software trigger
  • Programmable pacer trigger
  • External pulse trigger
• It has the ability to transfer A/D transformed data by program control,interrupt handler routine or DMA transfer.
• An Intel 8253-5 Programmable Timer/Counter provides pacer output (trigger pulse) at the rate of 0.5 MHz to 35 minutes/pulse.The timer increment is 2 MHz. One 16-bit counter channel is reserved for user configuration applications.
• Two 12-bit monolithic multiplying D/A output channels. An output range from 0 to +5V or 0 to +10V can be created by usingthe on-board -5V or -10V reference. This precision reference is derived from the A/D transformer reference. External AC or DC references can also be used to generate other D/A output ranges.

- Switch SW2: I/O Port to I/O Port address space (I/O Port base address) to DIP Switch
- Jumper SW3:It determines either 16 single ended/8differential analog input is to be taken.
- Jumper SW2:It selects either Unipolar/Bipolar input is to be taken
- DIP SW1:This switch determines Input range selection either 0-1v to0-10v in unipolar and +/-0.5v to +/-10v in bipolar range.It has six switch.
- Jumper SW4:DMA data transfer capability LEVEL 1/3 is selected by this switch
- Jumper SW1:Clock input frequency 1/10Mhz is choosen.
- Connectors :1:analog input single ended channels
  2:analog input differential channels
  3:analog output
  4:digital output
  5:digital input
  6:counter

4.1.1 :Specification:

Analog Input (A/D Transformer)

Channels: 16 single-ended
Resolution: 12 bits
Input Range: Bipolar +/- 10V, +/- 5V, +/- 2.5 V, +/- 1.25 V, +/0.625 V, +/- 0.3125 V.
All input ranges are software programmable.
Overvoltage: Continuous +/- 30V max.
Conversion type: Successive approximation

Transformer:	HADC574Z (built-in sample and hold)
Conversion speed:	30 KHz max.
Accuracy:	0.015 % of reading +/- 1 bit
Linearity:	+/- 1 bit
Trigger mode:	Software trigger, on-board programmable timer or external trigger.
Data transfer:	Program control, Interrupt control or DMA
External trigger	TTL or compatible, load 0.4 mA max. at 0.5V (low) or 0.05 mA max. at 2.7V (high).

Analog Output (D/A Transformer)

Channels:	2
Resolution:	12 bits
Output range:	0 to + 5V or 0 to +10V with fixed -5V or -10V reference. Max. +10V or -10V with DC or AC reference.
Reference voltage	
Internal:	-5V (+/- 0.1V), -10V (+/- 0.2 V)
External:	DC or AC, +/- 10 V max
Conversion type:	12-bit monolithic multiplying
Analog devices:	AD7541AKN or equivalent
Linearity:	+/- 0.5 bit
Output drive:	+/- 5 mA max
Settling time:	30 microseconds

Digital Input

Channel:	16 bits
Level:	TTL compatible
Input voltage	
Low:	0.8 V max
High:	2.0V min.
Input load	
Low:	0.4 mA max. at 0.5 V
High:	0.05 mA max. at 2.7 V

Digital Output

Channel:	16 bits
Level:	TTL compatible
Output voltage	
Low:	Sink 8 mA at 0.5 V max.
High:	Source -0.4 mA at 2.4V min.

Programmable Timer/Counter

Device:	Intel 8253

Counters:	3 channels, 16-bit, 2 channels permanently connected to 2 MHz clock as programmable pacer, 1 channel free for user application
Input, gate:	TTL/DTL/CMOS compatible
Time base:	2 MHz
Pacer output:	35 minutes/pulse to 0.5 MHz

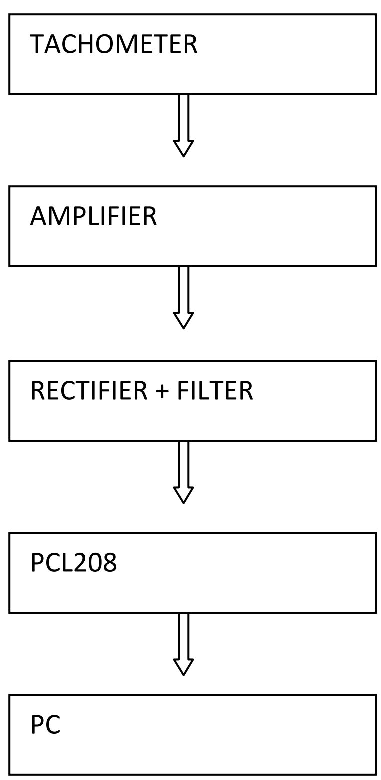
Interrupt Channel

Level:	IRQ 2 to 7, 10, 11, 12, 14, 15 jumper selectable
Enable:	VIA S0, S1 and S2 of CONTROL register

DMA Channel

Level:	1 or 3, jumper selectable
Enable:	Via S0, S1 and S2 of CONTROL register

4.2 : BLOCK DIAGRAM OF COMPLETE PROCESS



4.3 :REGISTER STRUCTURE AND FORMAT:
I/O Port address space Registers

LOCATION	READ	WRITE
BASE+0	A/D LOW BYTE AND CHNL	SOFTWARE A/D TRIGGER
BASE+1	A/D HIGH BYTE	N/A
BASE+2	MUX SCAN CHANNEL	MUX SCAN CHANNEL
BASE+3	D/I LOW BYTE	D/O LOW BYTE
BASE+4	N/A	D/A 0 LOW BYTE
BASE+5	N/A	D/A 0 HIGHBYTE
BASE+6	N/A	D/A 1 LOW BYTE
BASE+7	N/A	D/A 1 HIGHBYTE
BASE+8	PCL 208 STATUS	CLEAR INTERRUPT RQST
BASE+9	PCL 208 CONTROL	PCL 208 CONTROL
BASE+10	N/A	COUNTER ENABLE
BASE+11	D/I HIGH BYTE	D/O HIGH BYTE
BASE+12	COUNTER0	COUNTER0
BASE+13	COUNTER1	COUNTER1
BASE+14	COUNTER2	COUNTER2
BASE+15	N/A	COUNTER CONTROL

OTHER REGISTERS ARE:

- A/D DATA REGISTER
- MUX SCAN REGISTER
- DIGITAL I/O REGISTER
- D/A OUTPUT REGISTER
- A/D STATUS REGISTER
- PCL 208 CONTROL REGISTER
- TIMER COUNTER ENABLE REGISTER

4.4 : PROGRAMMING PCL 208:

For programming pcl 208 we need either BASIC or TURBO C.

4.4.1GW BASIC

The programming language GW-BASIC was a language of BASIC programming language developed by the giant software firm Microsoft from BASICA(programming language), originally developed for Compaq It is compatible with Microsoft and IBM BASICA, but were disk based and were not required the ROM BASIC. It was packaged with MS-DOS OS on IBM PC compatibles by Microsoft. Microsoft also sell a BASIC compiler, BASCOM, compatible with GW-BASIC, for those applications which require more speed. The programming language was appropriate for uncomplicated games, big business firm programs and the similar to corporations and uses. Since it was built-in with most versions of MS-DOS, it was also an reasonably priced way for many learning stage programmers to learn the rudiments of computer programming With the launch of MS-DOS 5.0, GW-BASIC's consign was eventually taken by QBasic, a cut-down version of the unconnectedly available Quick BASIC compiler.

Before programming for PCL 208 we need to load driver.

LOADING MACHINE DRIVER IMMEDIATELY AFTER BASIC WS

```
LOAD PCL208.BIN DRIVER TO BASIC AREA
CLEAR,57344! ‘SET BASIC OPERATE SPACE WITHIN 56K
DEF SEG=0
SG=256*PEEK(&H511)+PEEK(&H510)’GET BASIC SEGMENT
SG=SG+57344!/16
DEF SEG=SG
BLOAD” PCL208.BIN ”,0
‘END OF DRIVER LOADING
```

LOADING MACHINE LANGUAGE DRIVER INDEPENDENT OF BASIC WS AREA

```
LOAD PCL208.BIN DRIVER TO AN OUTSIDE AREA
DEF SEG=&H5000 ‘DEFINE OUTSIDE AREA
BLOAD” PCL208.BIN ”,0
‘END OF DRIVER LOADING
```

4.5 :PCL 208 DRIVER ROUTINES

FUNCTION	DESCRIPTION
0	INITIALIZE PCL 208 DRIVER
1	SET MUX SCAN RANGE
2	READ NEXT CONVERSION CHANNEL
3	PERFORM SFTWARE TRIGGERED SINGLE A/D CONVERSION
13	WRITE DIGITAL OUTPUT D/O
14	READ D/I
15	WRITE TO ONE OF TWO D/A
16	WRITE TO BOTH D/A CHANNELS

Accessing driver function

CALL PCL208(FUNC%,DAT%,ER%)

FUNCTION 0 : INITIALIZATION

- PORT% = &H300
- DAT%(0) = PORT%
- DAT%(1) = 3
- DAT%(2) = 1
- PCL208 = 0
- ER% = 0
- FUNC % = 0
- CALL PCL208(FUNC%,DAT%(0),ER%)
- IF ER%<>0 THEN PRINT “INSTALLATION FAILED !”:STOP

FUNCTION 1: SET MUX SCAN RANGE

- DAT%(0)=3
- DAT%(1)=5
- FUNC%=1
- CALL PCL208(FUNC%,DAT%(0), ER%)
- IF ER%<>0 THEN PRINT “SET SCAN CHANNEL FAILED !”:STOP

FUNCTION 2 : READ NEXT MUX CHANNEL

- FUNC%=2
- CALL PCL208(FUNC%,DAT%(0),ER%)
- IF ER%<>0 THEN PRINT “READ VALUE FAILED !”:STOP
- PRINT “NEXT SCAN CHANNEL=”;DAT%(0)
- PRINT “START SCAN CHANNEL=”;DAT%(1)
- PRINT “STOP SCAN CHANNEL=”;DAT%(2)

FUNCTION 3 : PERFORM SINGLE A/D CONVERSION

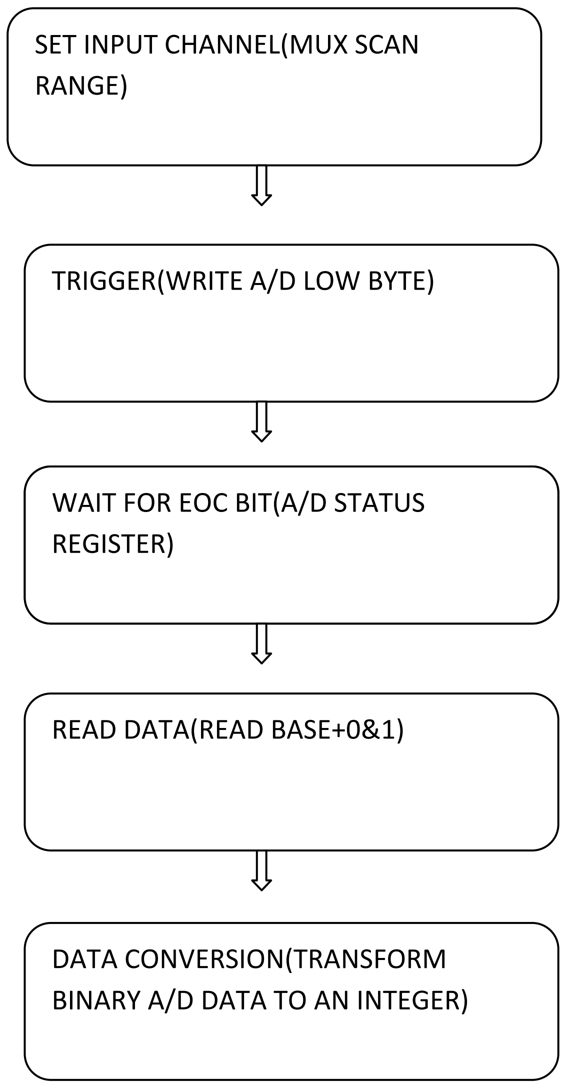
- FUNC%=3
- CALL PCL208(FUNC%,DAT%(0),ER%)
- IF ER%<>0 THEN PRINT “SOFTWARE TRIGGER FAILED !”:STOP
- PRINT USING“SCAN CHANNEL=## READING=####”;DAT%(1), DAT%(0)

FUNCTION 4 :PERFORM A/D CONVERSION ,SAVE TO ARRAY VARIABLE

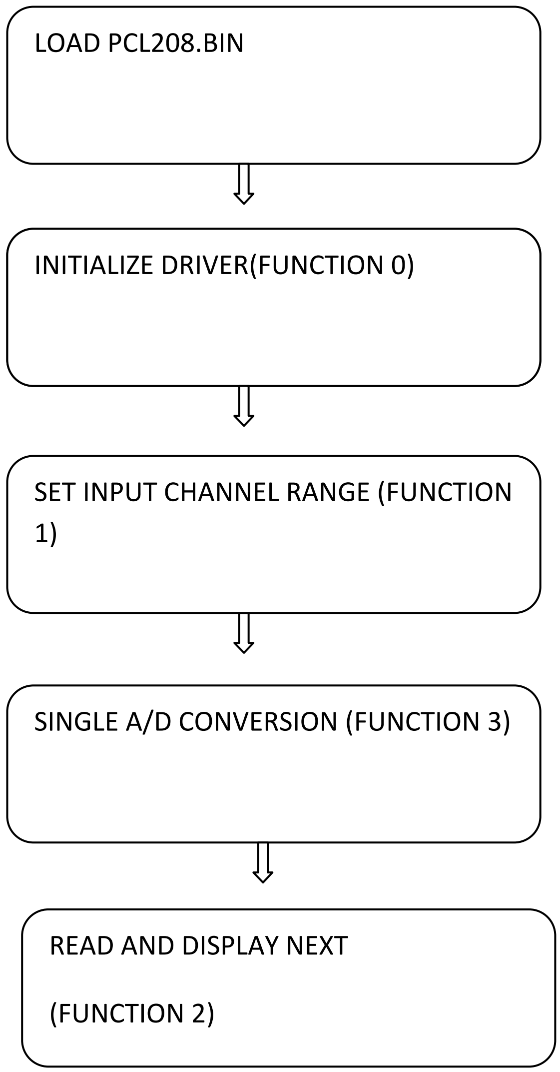
- DIM ARY%(599)

- DAT%(0)=600
- DAT%(1)=VARPTR(ARY%(0))
- DAT%(2)=1/0
- FUNC%=4
- CALL PCL208(FUNC%,DAT%(0),ER%)
- FOR I=0 TO 599
- PRINT ARY%(I)
- NEXT I

4.6 : A/D CONVERSION WITHOUT THE PCL 208 DRIVER:



4.7 :A/D CONVERSION WITH THE PCL 208 DRIVER



4.8 :ADC USING PCL207

Procedure for connection:-

1) Connect the PCL card to any of the ISA slot .Before connecting check the switch setting it base address should be set to 220.
2) Give the input voltage to **channel 1** as 0 to 5 volt.
3) Compile and run the program using Turbo C and the result on the console screen as the voltage is varied.

C CODE:

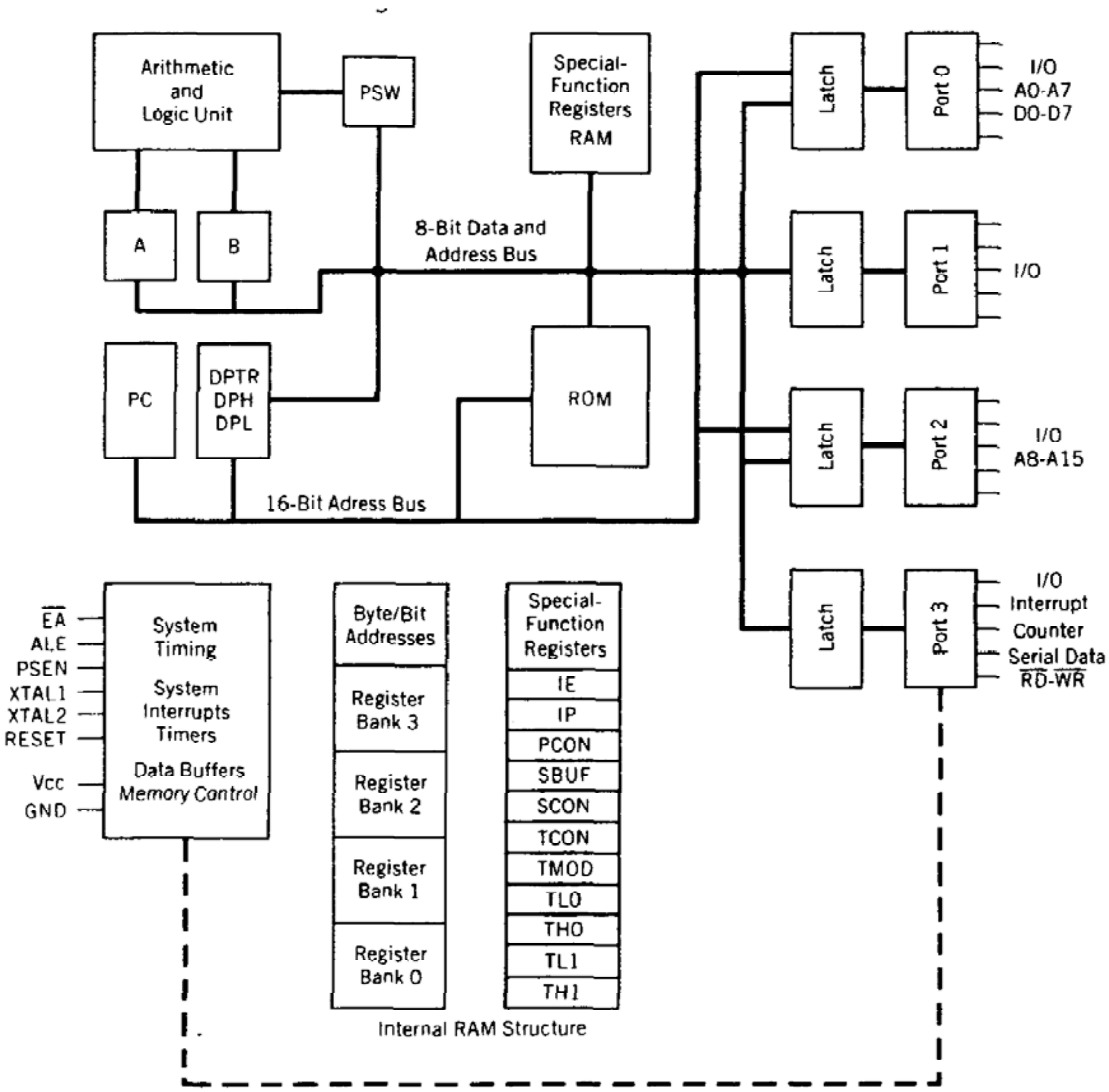
```
#include<stdio.h>

main()

{       int ch,ba=0x220,hb,lb,e;

        float volts,count;

        clrscr();
up:

        ch=0;           //ch1=start channel no.
outp(ba+10,ch);        //mux scan channel
outp(ba+12,0);          //software ad trigger
chk:
e=inp(ba+5);
if(e>=16)              //check for DRDY low
goto chk;
else
hb=(e & 15);
lb=inp(ba+4);
count=(hb*256+lb)-2048;
volts=(count*10)/4096;
printf("\n     CHANNEL %d   VOLTS %10.4f COUNT %10.0f\n " ch,volts,count);
goto up;

 }
```

CHAPTER .5

DATA ACQUISITON USING MICRO CONTROLLER

5.1 :Data acquisition using AT80C51 & ADC0804LCN

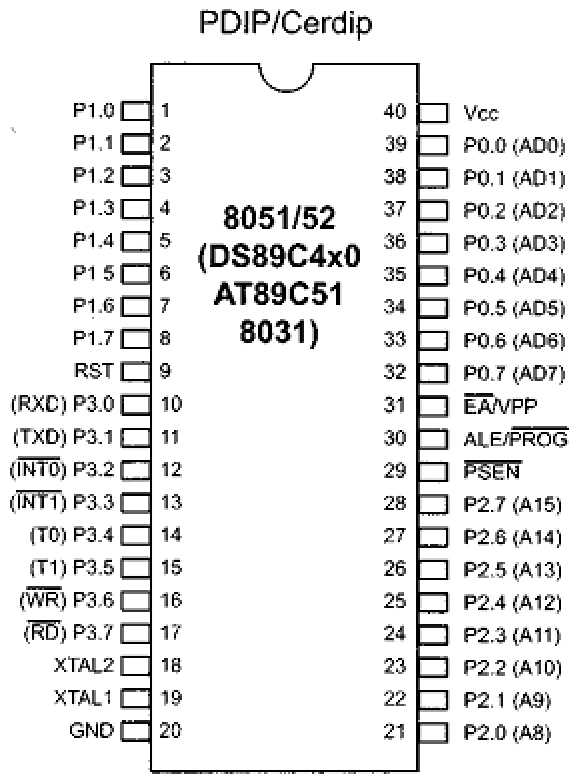
AT80C51 Architecture:



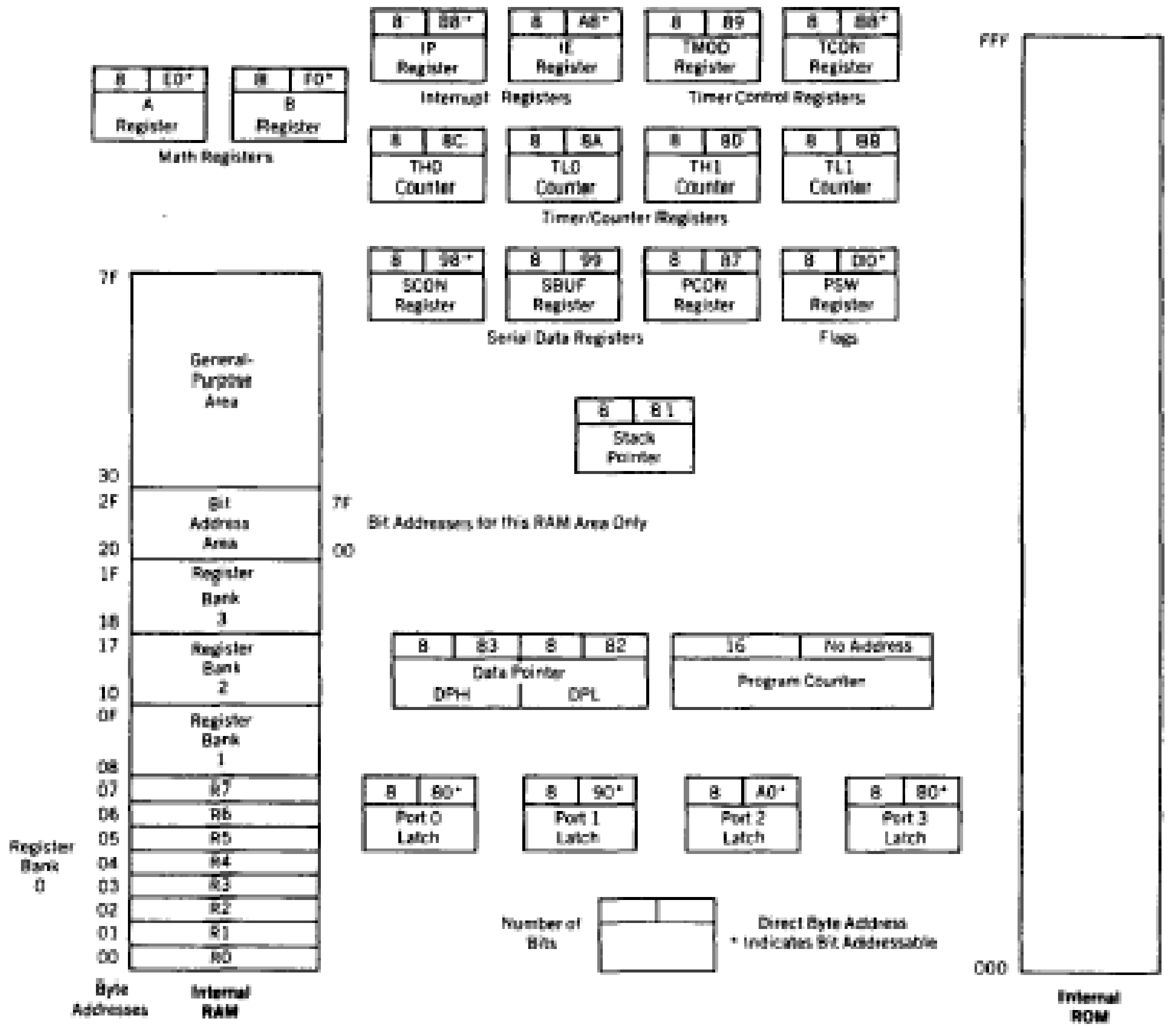
5.2 :MEMORY ORGANIZATION:

The logical separation of program and data memory allows the data memory to be accessed by 8-bit addresses, which can be rapidly stored and manipulated by an 8-bit CPU.
Program memory (ROM, EPROM) can only be read, not written to.There can be up to 64k bytes of program memory. In the 80C51, the lowest 4k bytes of program are on-chip. In the ROMless version, all program memory is external. The read strobe for external program memory is the PSEN (program store enable).Data Memory (RAM) occupies a separate address space from Program Memory. In the 80C51, the lowly 128 bytes of data memory are on-chip. Up to 64k bytes of external RAM can be addressed in the external Data Memory space. In the ROMless version, the lowest 128 bytes are on-chip. The CPU generates read and write signals, RD and WR, as needed for the period of outside Data Memory accesses.External Program Memory and external Data Memory may be combined if desired by applying the RD and PSEN signals to the inputs of an AND gate and using the output of the gate as the read strobe to the external Program/Data memory.

5.3 :ATMEL AT89C51 Pin Diagram



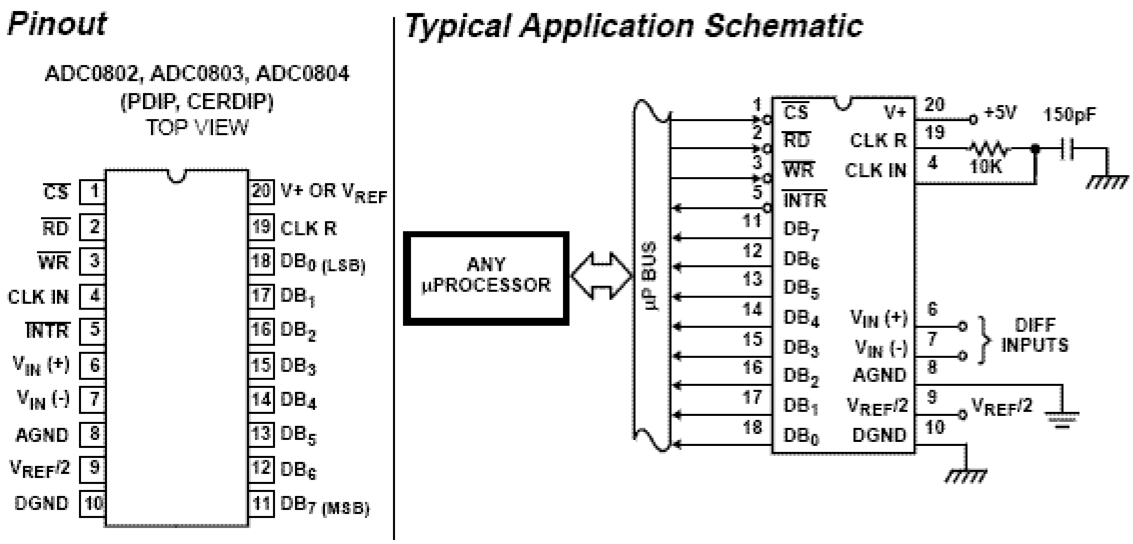
:



<u>Fig 5.1 </u>ATMEL AT89C51 Programming Model

5.4 :ADC 0804 PIN DIAGRAM:

The peripheral used in processors ADC0804LCN is an analog to digital converter peripheral interfaced with all kinds of microcontroller. Of all pins 11 pins of ADC0804LCN are used for the application, eight for data pins and 3 for control pins.



The shown timing diagrams on the next page are from ADC0804LCN datasheet. The first diagram (FIGURE 5.2) shows how to start a conversion. Also you can see which signals are to be asserted and at what time to start a conversion. So looking into the timing diagram FIGURE 5.2. We make a note of the steps or say the array in which signals are to be asserted to begin a conversion of ADC. As we have determined to make Chip select(CS) pin as low so we don't need to bother about the CS signal in the timing diagram. Steps are displayed below to start an ADC conversion using ADC0804LCN.

1. Make chip select (CS) signal low.
2. Make write (WR) signal low.
3. Make chip select signal(CS) high.
4. Wait for INTR pin to go low (when conversion ends).

   After the conversion is over, the data is obtainable in the output latch of the ADC. Data of the new conversion is only available for reading after ADC0804LCN made INTR pin low or when the conversion is over. Steps are displayed below to read output from the ADC0804LCN.

1. Make chip select (CS)signal pin low.
2. Make read (RD) signal low.
3. Read the data from port where ADC0804LCN is connected.
4. Make read (RD) signal high.
5. Make chip select (CS)signal high.

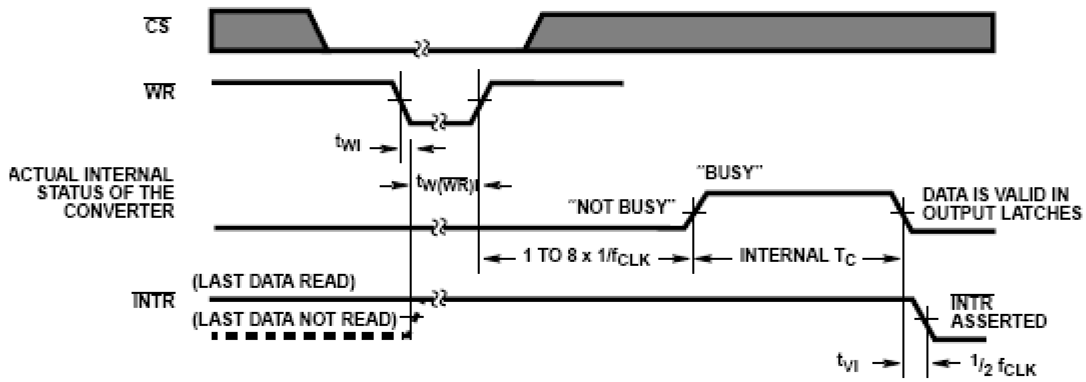


Fig 5.2 start conversion

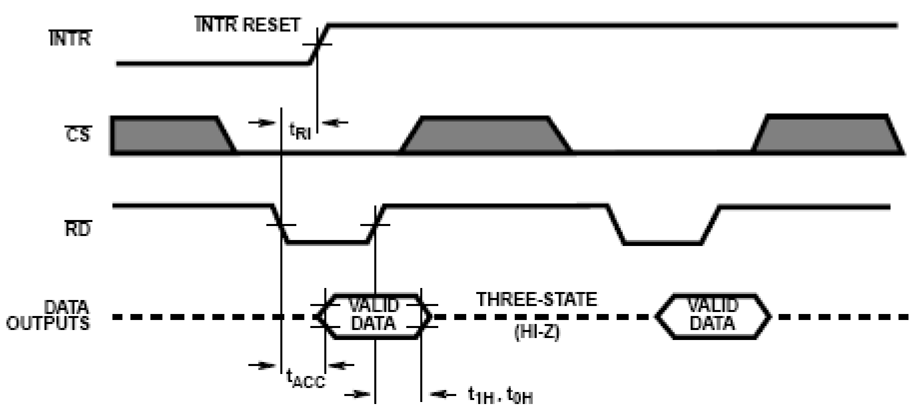
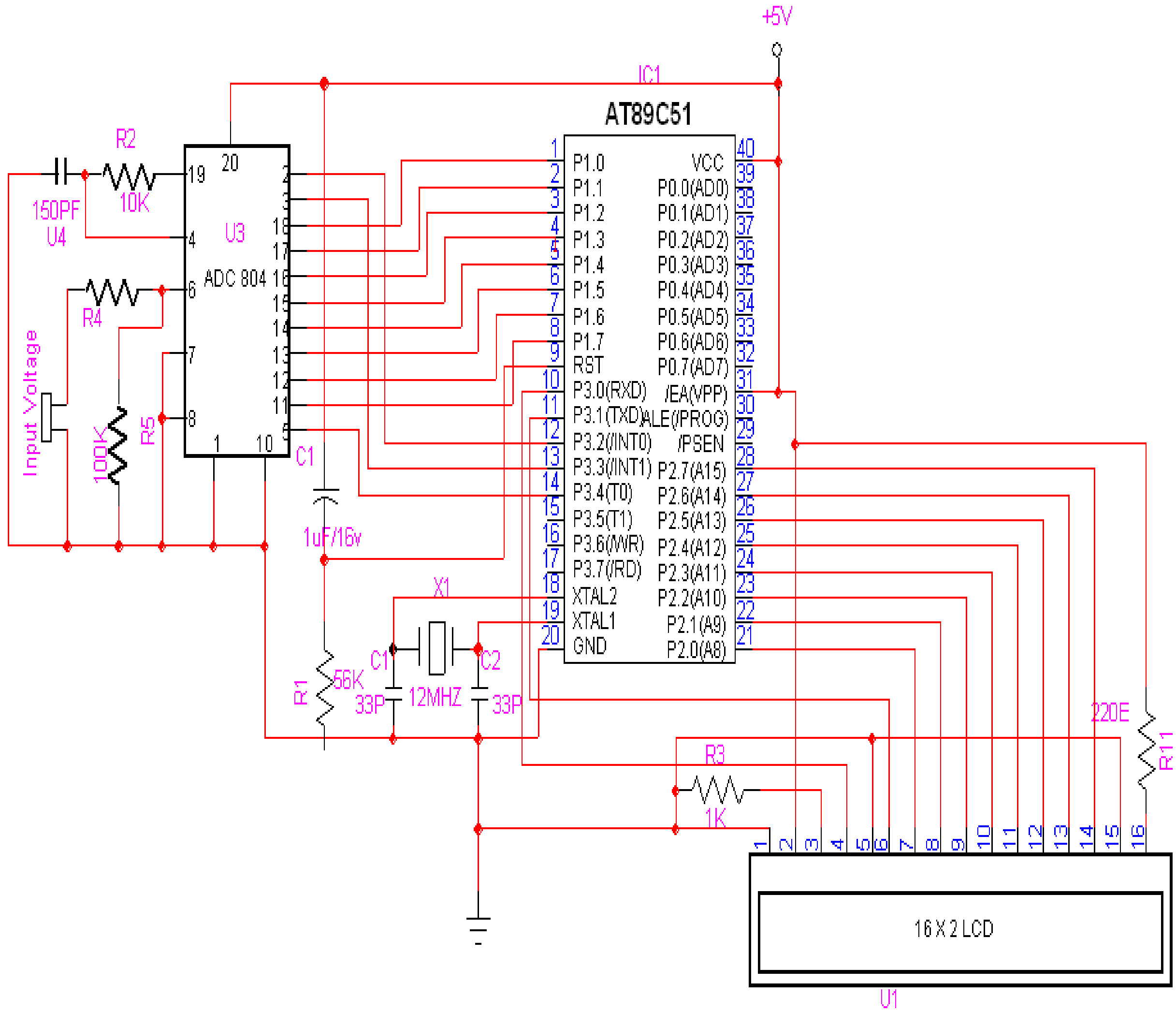


Fig 5.3 Out enable and reset INTR

To be capable to employ A/D conversion using the ADC0804LCN 8-bit A/D transformer. We modeled a circuit and programmed the chip so that when an analog signal is given as input, the equivalent digital voltage is shown on the LCD display. Thus, in effect, our circuit should operate as a simple voltmeter.

5.5 :Circuit Diagram of Data Acquisition using ADC0804LCN,AT89C51,LCD



5.6 :Description:

Signal processing applications deals with converting analog signals into digital values and vice-versa . The goal of an A/D transformer is to make a decision on the output digital word matching to an analog input signal. The A/D transformer operates on the successive approximation principle. Analog switches are closed consecutively by successive-approximation logic until the analog differential input voltage [Vin(+) - Vin(-)] matches a voltage derived from a tapped resistor string across the reference voltage. The normal process proceeds as follows. On the high-to-low transform of the WR input, the internal SAR latches and the shift-register stages are reset, and the INTR output will be set high. Only if the CS input and WR input stay low, the A/D will remain in a reset state. Conversion start from 1 to 8 clock periods after one of these inputs makes a low-to-high transition. After the necessary counts of clock pulses to complete the conversion, the INTR pin will make a high-to-low transition. This can be used to interrupt a processor, or or else signal the availability of a new conversion. A RD operation(with CS low) will clear the INTR line high again. The device may be operated in the free-running mode by connecting INTR to the WR input with CS=0.Since this is an 8-bit A/D transformer, a voltage from 0-5V. O will be repersented as 0000 0000 (0 in decimal) and 5V is represented as 1111 1111 (256 in decimal). To change a value X volts to decimal, formula: (X * 5.0)/256.To get a improved resolution, and display the vlaue as a floating point number, you can multiply the numerator by a factor of 100, 1000 etc. and then print the voltage in view of that.

Program

```
        ORG             0
        RD              BITP2.5         ;RD
        WR              BITP2.6         ;WR(start conversion)
        INTR            BITP2.7         ;end of conversion
        MYDATA           EQU P1         ;P1.0-P1.7=D0-D7 of the ADC0804LCN
        MOV             P1,#0FFH        ;make P1= input
        SETB            INTR
BACK:CLR                WR              ;WR=0
        SETB            WR              ;WR=1 L-to-H to start conversion
HERE:JB                 INTR,HERE       ;wait for end of conversion
        CLR             RD              ;conversion finished enable RD
        MOV             A,MYDATA        ;read the data
        ACALL           CONVERSION      ;hex to ASCII conversion
        ACALL           DATA_DISPLAY    ;display the data
        SETB            RD              ;make RD = 1for next round
        SJMP            BACK
CONVERSION:
        RAM_ADDR        EQU   40H
        ASCI_RESULT     EQU   50H
        COUNT           EQU   3
        ORG             100
        ACALL           BIN_DEC_CONVRT
        ACALL           DEC_ASCI_CONVRT
        SJMP            $
BIN_DEC_CONVRT
        MOV             R0,#RAM_ADDR
        MOV             A,P1
```

```
        MOV         B,#10
        DIV         AB
        MOV         @R0,B
        INC         R0
        MOV         B,#10
        DIV         AB
        MOV         @RO,B
        INC         R0
        MOV         @RO,A
        RET
DEC_ASCI_CONVRT
        MOV         R0,#RAM_ADDR
        MOV         R1,#ASCI_RESULT
        MOV         R2,#3
BACK:MOV            A,@R0
        ORL         A,#30H
        MOV         @R1,A
        INC         R0
        INC         R1
        DJNZ        R2,BACK
        RET
DATA_DISPLAY:
        ACALL       READY
        MOV         P1,A
        STB         P2.0
        CLR         P2.1
        SETB        P2.2
        ACALL       DELAY
        CLR         P2.2
        RET
READY:
        SETB        P1.7
        CLR         P2.0
        SETB        P2.1
BACK:CLR            P2.2
        ACALL       DELAY
        SETB        P2.2
        JB          P1.7,BACK
        RET
DELAY:
        MOV         R3,#50
HERE2:MOV           R4,#255
HERE:DJNZ           R4,HERE
        DJNZ        R3,HERE2
        RET
        END
```

5.7 LCD OUTPUT TABLE

SL NO.	ANALOG INPUT TO ADC 0804	LCD READING(RPM)
1	1.03	52
2	2.13	110
3	2.62	136
4	2.94	148
5	3.17	160
6	3.49	175
7	3.72	195
8	3.96	200
9	4.37	222
10	4.54	230
11	4.8	240
12	5	254

CONCLUSION

Here in our project we have designed a data acquisition system using ATMEL89c51 microcontroller and ADC0804LCN analog to digital transformer. The designed model was programmed in assembly level language for single analog input and was tested under proper condition. Also we have used PCL-208 as an interface to realize PC based data acquisition in partial fulfillment. Owing to shifting of most instrumentation system towards PC compatibility it offer more advantage compared to 89c51 based data acquisition system. Also it is more flexible as program can be changed according to requirement repeatedly.

Table :

SL No.	RPM	TACHOMETER OUTPUT (mv)	AMPLIFIER OUTPUT (v)	ANALOG INPUT TO ADC 0804 or OUTPUT OF RECTIFIER	LCD READING(RPM)
1	53	53	1.03	1.03	52
2	109	71	2.13	2.13	110
3	134	84	2.62	2.62	136
4	150	109	2.94	2.94	148
5	162	123	3.17	3.17	160
6	178	134	3.49	3.49	175
7	190	150	3.72	3.72	195
8	202	162	3.96	3.96	200
9	223	178	4.37	4.37	222
10	232	190	4.54	4.54	230
11	245	200	4.8	4.8	240
12	251	213	5	5	254

REFERENCES

1. PC-Based Instrumentation- by N.Mathivanan, PHI Publication.
2. PC-Based Instrumentation and control by Mike Tooley, Newnes Publication.
3. Opamps & Linear Integrated Circuits by R.A. Gayakward , PHI Publication.
4. Data acquisition technique using personal computers by Howard Austurlitz.
5. Bentley John P.2005 Princople of Measurement system 3rd ed. Pearson education
6. Helfrick Albet D. and Cooper William D.2007 Modern electronics Instrumentation and measurement system 2nd ed. Phi Private Limited.
7. ATMEL89c51 Datasheet
8. ADC0808ccn Datasheet

WEBSITES:

1.http://www.answers.com/topic/operationalamplifier
2.http://www.ATMEL AT89C51projects.info/expC13.asp
3.http://www.ATMEL AT89C51projects.net/adc-interfacing/ADC0804LCN-interfacing.php
4.http://en.wikipedia.org/wiki/Variable_reluctance_sensor
5.http://dbpedia.org/resource/Operational_amplifier
6.http://www2.electronicproducts.com/Hall_vs_VR_Which_speed_sensor_is_wrong_for_you--article-OCTCHE1-OCT2000.aspx