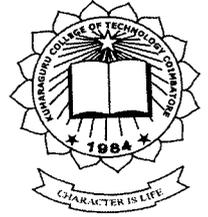


P-2536



**PNEUMATIC VALVE CONTROL AUTOMATION IN
THERMAL POWER PLANT**



By

R.ARUN PRASAATH

Reg. No : 0720105001

of

**KUMARAGURU COLLEGE OF TECHNOLOGY
(AUTONOMOUS)
COIMBATORE – 641 006**

A PROJECT REPORT



Submitted to the

FACULTY OF ELECTRICAL AND ELECTRONICS ENGINEERING

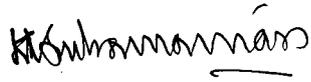
In partial fulfillment of the requirements
for the award of the degree
Of

**MASTER OF ENGINEERING
IN
POWER ELECTRONICS AND DRIVES**

MAY, 2009

BONAFIDE CERTIFICATE

Certified that this project report titled "PNEUMATIC VALVE CONTROL AUTOMATION IN THERMAL POWER PLANT" is the bonafide work of Mr.R.ARUN PRASAATH (0720105001) who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report of dissertation on the basis of which a degree or ward was conferred on an earlier occasion on this or any other candidate.



HEAD OF THE DEPARTMENT

(Prof. K.REGUPATHY SUBRAMANIAM)



GUIDE

(Prof.K.T.VARADARAJAN)

The candidate with University Register No. 0720105001 was examined by us in Project Viva-Voce examination held on 06.05.09



INTERNAL EXAMINER



EXTERNAL EXAMINER

VELALAR COLLEGE OF ENGINEERING AND TECHNOLOGY

THINDAL, ERODE-638012



DEPARTMENT OF EEE

FIRST

NATIONAL CONFERENCE ON

"APPLICATION OF EMERGING TECHNOLOGIES IN ELECTRICAL SCIENCE"



NAETES-09

CERTIFICATE

Awarded to

R.Arun Prasaath

M.E (Power Electronics & Drives)

Kumaraguru College of Technology, Coimbatore

For presenting a paper titled " PNEUMATIC VALVE CONTROL AUTOMATION IN THERMAL POWER PLANT" in the First National conference on " Application of Emerging Technologies in Electrical Science " during 09th April 2009 at Velalar college of Engineering and Technology, Erode.


COORDINATOR
M.SATHIVEL


CONVENOR
Dr.M.SRINIVASAN


SESSION CHAIR


PRINCIPAL
Dr.K.PALANIISWAMY


ADMINISTRATIVE DIRECTOR
Dr.P.SABAPATHI

TAMIL NADU ELECTRICITY BOARD:

FROM

**Er.M.RAMACHANDRAN. B.E.,
Executive Engineer,
Training & Environment,
M.T.P.S. / METTUR DAM –6.**

TO

**THE PRINCIPAL,
Kumaraguru College of Technology,
Coimbatore.**

Lr.No:- EE/ Trg. & Env./ MTPS / Trg. / F. 19 / D. 60 / 2009 dt. 30.04.2009.

Sir,

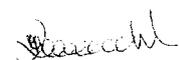
**Sub:- MTPS – Project Work – Student undergone Project Work at
MTPS / Attendance Certificate – Regarding.**

**Ref:- 1. Lr.No:- 03529/GM/HRD/N21/N212/08, dt.26.08.2008.
2. Lr.No:- EE/T&E/MTPS/Trg./F.19/D.757/09, dt.01.12.2008.**

**The following student from your college have done Project Work at
Mettur Thermal Power Station as detailed below:**

Sl. No.	Name of the Student	Branch	Period	Topic
1.	R.ARUNPRASAATH.	Final Year M.E. (Power Electronics & Drives)	01.12.2008. TO 30.04.2009.	Closed Loop Control System used in Plant Control Automation.

Yours faithfully,



**EXECUTIVE ENGINEER,
TRAINING/ MTPS.**

**Station: MTPS.
Date:30.04.2009.**

ABSTRACT

The Project deals with the design and fabrication of a prototype model for automating the Pneumatic Valve regulation systems at Mettur Thermal Power Station, Mettur, Tamil Nadu. Such valve regulating systems are used for controlling the physical parameters of steam, viz., pressure, flow rate and temperature that are controlling the generation and other auxiliary processes.

Based on a feedback from the steam output from a valve, the valve is either opened or closed, until the desired value of either pressure, temperature or flow rate is achieved. The Microcontroller (AT89C51) in our project is programmed so that it continuously sends signal input to the valve, maintaining the desired steam output. If any deviation is encountered in the output, the circuit changes the signal input to the valve, until the preset value is reached at the output. The circuit has the two modes of operation that is forced mode and automatic mode which depends on the range of deviation from the normal values. The automatic mode is the normal default operation status. In case of wide deviations from normal values the system switches to forced manual mode, requiring manual intervention. This transition is indicated by alarm and indication circuits. The system still tries to bring the parameter under control by continuously controlling the input and correcting the valve opening.

ACKNOWLEDGEMENT

I thank Prof **K.T.VARADARAJAN.**, for the support of encouragement and guidance given in completing the project. I thank for the support, encouragement and ideas.

I would like to convey my honest thanks to all **Teaching** staff members and **Non Teaching** staffs of the department for their support. I would like to thank all my classmates who gave me a proper light moments and study breaks apart from extending some technical support whenever I needed them most.

I dedicate this project work to my **Parents** for no reasons but feeling from bottom of my heart, without their love this work wouldn't possible.

LIST OF TABLES

TABLE No.	TITLE	PAGE NO.
3.1	Port Connections	10
3.2	Regulators ICs	21

LIST OF FIGURES

FIGURE No.	DESCRIPTION	PAGE NO.
1.1	Overall Block Diagram	2
2.1	Valve System in Main Steam Line	3
2.2	Temperature Control Methodology	4
3.1	Overall Circuit Diagram	6
3.2	Block Diagram of Microcontroller Connections	7
3.3	Master Control Circuit	8
3.4	Timing and Reset Circuit	11
3.5	Input Keypad	12
3.6	LCD Display	13
3.7	Alarm and Indication Circuits	13
3.8	Digital to Analog Conversion Circuitry	16
3.9	DAC IC	16
3.10	Amplifier Section	17
3.11	LED-LDR Couple	18
3.12	Analog to Digital Conversion Circuitry	19
3.13	Block Diagram of Power Supply	20
3.14	Power Supply Circuit	20
4.1	Motor Control Circuit	24
5.1	Complete Flowchart of the Project	25

LIST OF ABBREVIATIONS

MTPS	Mettur Thermal Power Station
ADC	Analog to Digital Converter
DAC	Digital to Analog Converter
DESH	De - Superheater
BEP	Boiler Feed Pump
LED	Light Emitting Diode
LDR	Light Dependent Resistor
LCD	Liquid Crystal Display
TTL	Transistor – Transistor Logic
RAM	Random Access Memory
CMOS	Complementary Metal Oxide Semiconductor

TABLE OF CONTENTS

Chapter No.	Title	Page No.
	BONAFIDE CERTIFICATE	ii
	PROOF OF PUBLISHING A PAPER	iii
	ABSTRACT	v
	LIST OF TABLES	vii
	LIST OF FIGURES	viii
	LIST OF ABBREVIATIONS	ix
	CONTENTS	
1	INTRODUCTION	1
	1.1 The Preamble	1
	1.2 Need for Automation	1
	1.3 Project Preview	2
2	SYSTEM OVERVIEW	3
	2.1 General Overview	3
	2.2 Controlling Methodology	4
	2.2.1 Pressure Control	4
	2.2.2 Temperature Control	4
	2.2.3 Flow Rate Control	4
3	HARDWARE MODULE	5
	3.1 General Circuit Description	5
	3.2 Master Control Circuit	6
	3.2.1 Micro-controller & Timing Circuit (2)	8
	3.2.1.1 Features of 89C51	8
	3.2.1.2 General Description 89C51	9
	3.2.1.3 Port Description	9
	3.2.1.4 Timing and Reset Circuit	10
	3.2.2 Input Keypad	11

3.2.3	Liquid Crystal Display	12
3.2.4	Alarm and Indication Circuit	13
3.3	Digital to Analog Conversion Circuitry	15
3.3.1	DAC Circuit	16
3.3.2	Amplifiers	17
3.3.3	LED – LDR Couple	18
3.4	Analog to Digital Conversion Circuitry	18
3.5	Power Supply Circuit	20
3.5.1.	Principle of Operation	20
3.5.2	Voltage Regulators	21
4	PROTOTYPE MODEL	22
4.1	Disturbance simulation	22
4.2	Actual Parameter Measurement	23
5	SOFTWARE MODULE	25
5.1	Flow Chart	25
5.2	Description of the Flow Chart	26
6	CONCLUSION AND FUTURE SCOPE	28
6.1	Conclusion	28
6.2	Future Scope	28
	APPENDIX - 1	29
	APPENDIX - 2	31
	APPENDIX - 3	34
	REFERENCES	55

CHAPTER 1

INTRODUCTION

1.1 THE PREAMBLE

The valve regulating systems are one of the most crucial ones in the Thermal Power Stations. Such valves are present universally in the power station. They are used primarily to control steam parameters like pressure, temperature and flow rate, apart from other functions like regulating flu gas, etc. The project mainly deals with the regulation of steam.

The steam directly obtained from the boiler has very high temperature and pressure. But for actual use, these values have to be reduced to the desired preset values. The controlling is achieved by regulating the extent of opening / closing of the valves. These valves are present throughout the steam line of the power plant. These valves are current – controlled pneumatic valves, which are identical in rating and functioning. The pneumatic valves work on a current input range of 4 to 20 mA. An input of 4mA completely closes the valve and a current input of 20 mA, fully opens the valve.

1.2 NEED FOR AUTOMATION

The regulation of steam requires continuous monitoring, since the steam condition directly affects several processes, including the main generation. But the presence of numerous valves in the system, results in a tedious and laborious task of monitoring. Thus in order to ensure reliability of operation and accuracy of output, inner closed loop system for each valve is necessary. Hence the automation of valve control mechanism has been taken up.

1.3 PROJECT PREVIEW

The project is an embedded system – based closed loop control system for controlling the valve operations. The overall scheme of the project is shown in fig 1.1

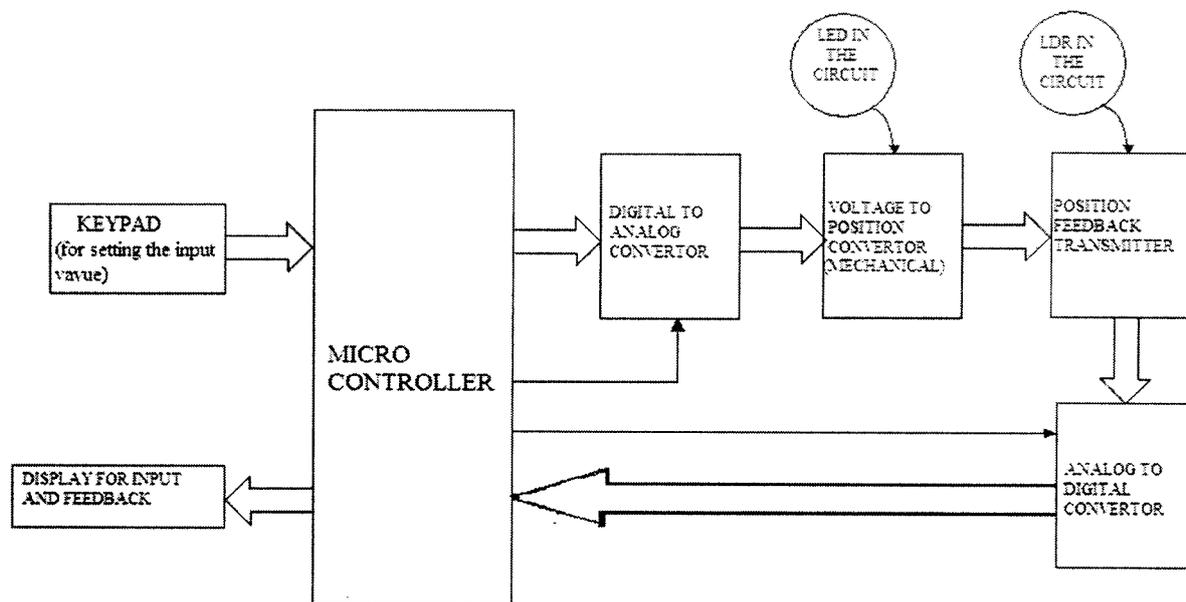


Fig 1.1 Overall Block Diagram

The Microcontroller controls the input current to the valve depending upon the signal received from the sensor. The sensor could be a pressure switch or a thermistor, depending on a parameter under regulation. For interfacing the analog signals from the system and the digital signals from the control circuit, Analog to Digital Converters and Digital to Analog Converters are used.

CHAPTER 2

SYSTEM OVERVIEW

2.1 GENERAL OVERVIEW

The project is designed for controlling the valve mechanism of a pneumatic valve that controls the pressure, temperature of flow rate of steam in the main steam line shown in Fig 2.1. Such identical valves are used to control the above mentioned parameters.

The pneumatic valves are used for regulating function and thus the opening / closing is smooth and continuous. The valve can be adjusted for any quantum of output, between the maximum and minimum valves. Each pneumatic valve is accompanied by a motorized valve and a hand valve, which are in series. This setup is common for any valve system in the plant. The pneumatic valves are current – controlled and operated within a range of 4 to 20 mA. A current input of 4 mA completely closes the valve and a current input of 20 mA fully opens the valve. By adjusting the current input to the valve, the extent of opening or closing of the valve can be changed.

Hand valve and motorized valves are ON / OFF valves that are used for isolation. In case of a system shutdown or any abnormality, the steam line may have to be isolated. This is done by completely closing motorized valve. Hand valve is used as a standby that can be used if and when the motorized valve fails.

This steam is used in the power plant for auxiliary purpose like Static Ejectors. Main Ejectors, Gland filling, de-aerator heating pumps, oil atomization and other heating purposes.

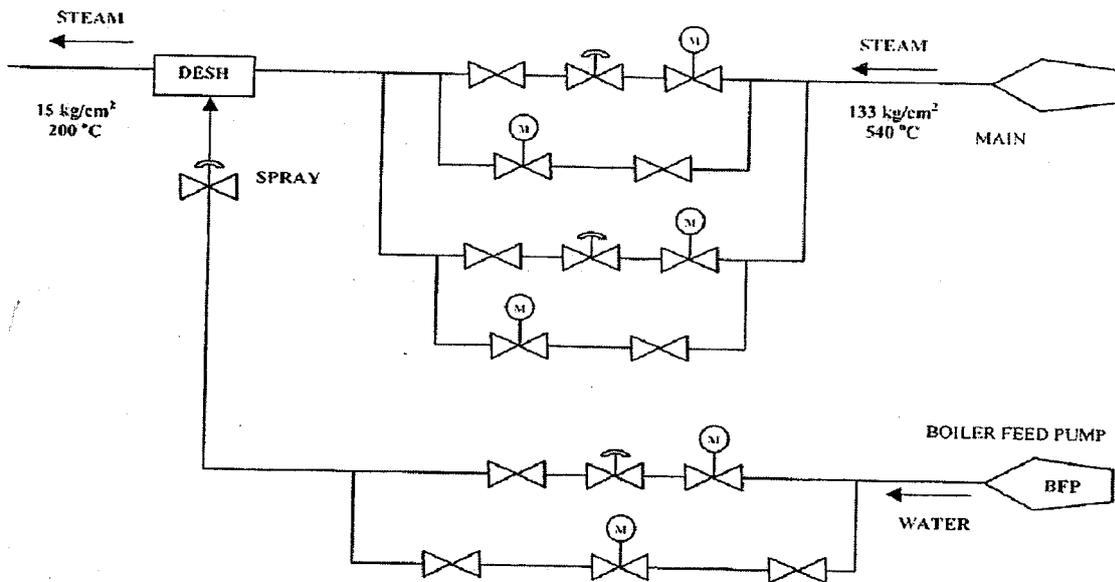


Fig 2.1 valve system in the main steam line

VALVE SPECIFICATIONS:

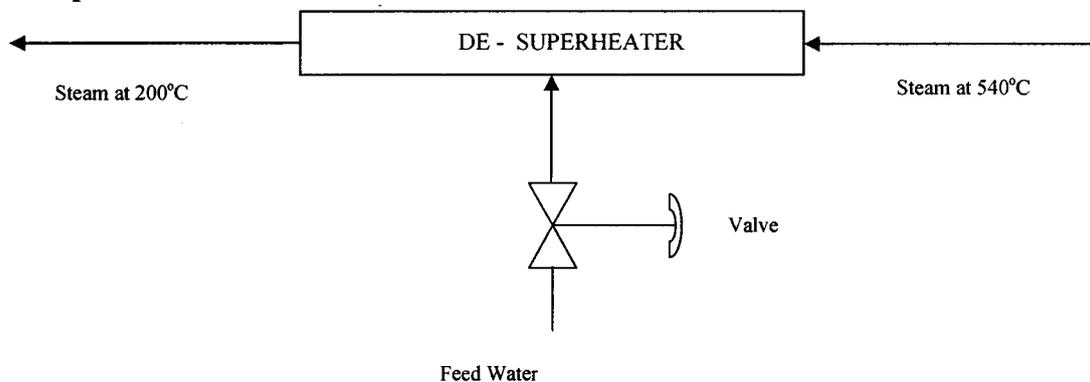
Make	24020 Gorle Italy
Size	200 mm
Signal	0.2 – 1 bar
Max. Supply	7 bar

2.2 CONTROLLING METHODOLOGY

2.2.1 Pressure Control

The pressure of steam is controlled by regulating the pressure valve. The steam from the main steam line is at a pressure of 133kg/cm^2 . This pressure has to be reduced to a value of 15 kg/cm^2 . In order to reduce the pressure, the valve is partly opened. This reduces the pressure in the output steam line. To increase the pressure level, the valve is opened still further. Similarly, for reducing the pressure, the valve is closed further. Thus by varying the amount of current supplied to the valve, the extent of opening/closing of valve can be controlled.

2.2.2 Temperature Control



Temperature of steam is reduced in a chamber called the Desuperheater as shown in fig 2.2. The steam temperature from the boiler is around 540°C . In order to reduce the temperature of steam, water from the boiler feed line is sprayed into the steam in the Desuperheater chamber. By controlling the amount of water sprayed, the temperature of steam can be controlled.

2.2.3 Flow Rate Control

The flow rate is controlled by controlling the degree of opening of the pneumatic valve. When the valve is opened further, the flow rate is increased, due to an increase in the output flow of steam. When the valve is closed further, the flow rate is reduced. Such flow rate control mechanisms are used at the utilization ends of the steam line. The desired valve of the steam flow rate depends upon the process, for which the steam is to be utilized.

CHAPTER 3

HARDWARE MODULE

3.1 GENERAL CIRCUIT DESCRIPTION

The overall circuit is based on ATME89C51 microcontroller. As shown in the block diagram (Fig 1), the various circuits like analog to digital conversion, interfaced to the microcontroller. Generally the overall circuit can be divided into the following parts.

1. Master Control Circuit
2. Digital to Analog Conversion Circuitry
3. Analog to Digital Conversion Circuitry
4. Power Supply Circuit

The Master Control Circuit Controls all the sub-circuits in the project. The digital to analog conversion circuitry is used to convert the digital command signals from microcontroller into analog voltage, which is used for exciting the LED, whose light intensity is being controlled. Additionally, the circuit also consists of the LED-LDR setup, which is used to demonstrate the closed loop operation of the project. The Analog to Digital Conversion Circuit converts the analog output from the LED and converts it into digital from the processing of the signal by the microcontroller. Based on the output voltage required, suitable voltage regulators are required for obtaining constant voltages.

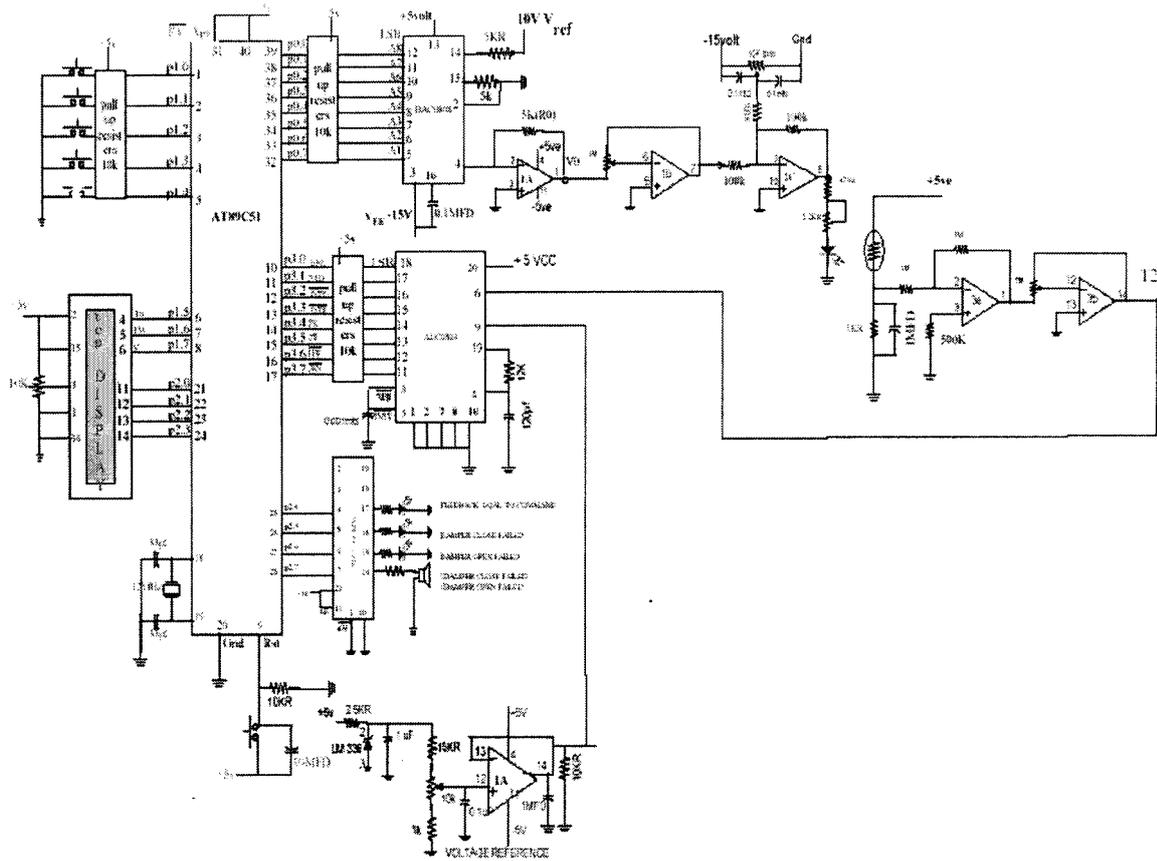


Fig 3.1 Prototype Model of the Circuit Diagram

3.2 MASTER CONTROL CIRCUIT

The master control circuit consists of the following sub-circuits, as shown in Fig 3.2

- Microcontroller and Timing Circuit
- Input keypad
- Liquid Crystal Display
- Alarm and Indication Circuit

The master control circuit is based on the ATMEL 89C51 microcontroller and the accompanying timing circuit. The input setting to the microcontroller is given through the key pad. The value of the parameter under control is displayed by the LCD. The alarm and indication circuit consists of any deviation or abnormality. The master control circuit diagram is shown in fig 3.3

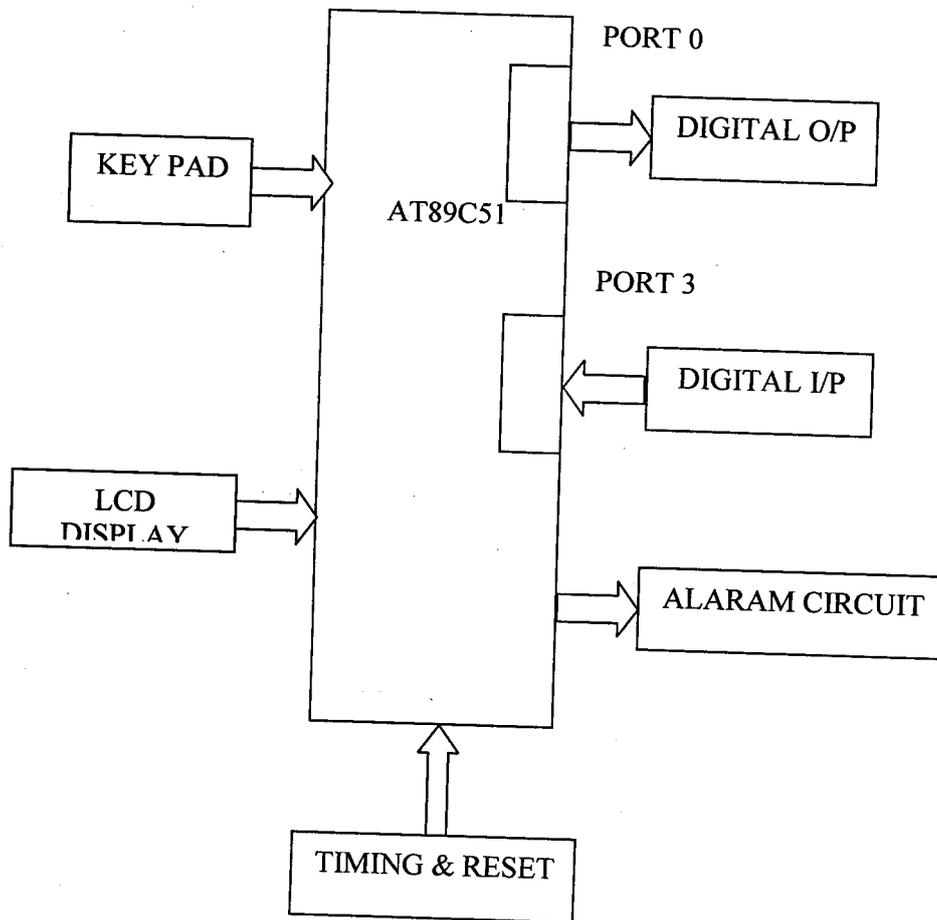


Fig 3.2 Block Diagram Microcontroller Connections

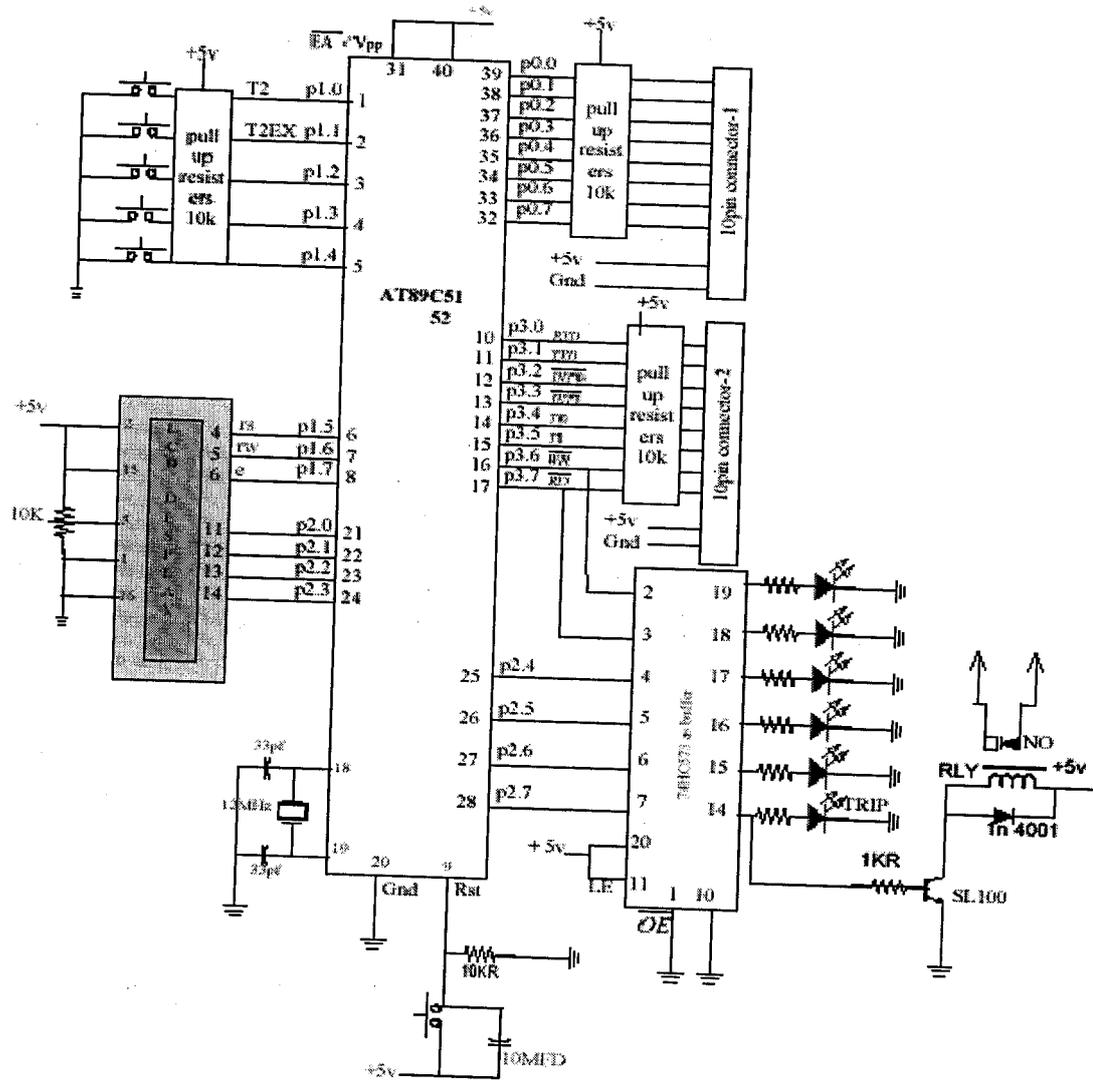


Fig 3.3 Master Control Circuit

3.2.1 Microcontroller and Timing Circuit (2)

3.2.1.1. Features of 89C51

- Compatible with MCS-510 Products
- 4 Kbytes of In-System Reprogrammable Flash Memory
Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation : 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 128 X 8- Bit Internal RAM
- 32 Programmable I/O Lines
- Two 16- Bit Timer/Counters

- Six Interrupt Sources
- Programmable Serial Channel
- Low Power Idle and Power Down Modes

3.2.1.2 General Description of 89C51

The AT89C51 is a low power, high-performance CMOS 8-bit microcomputer with 4 Kbytes of Flash Programmable and Erasable Read Only Memory (PEROM). The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard MCS-510 instruction set and pinout. The on-chip Flash allows the program memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

The AT89C51 provides the following standard features: 4 Kbytes of Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit timer/counters, a five Vector two-level interrupt architecture, a full duplex serial port. On -chip oscillator and clock circuitry. In addition, the AT89C51 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/ counters, serial port and interrupt system to continue functioning. The Power down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset. The maximum absolute rating of the IC is as follows.

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

3.2.1.3 Port Description

PORT 0:

Port 0 is an 8-bit open drain bidirectional I/O port. As an output port each pin can sink eight TTL inputs. When 1s are written to port 0 pins can be used as high-impedance inputs. Port 0 may also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode P0 has internal 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 bidirectional 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.

PORT 2:

Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. The Port2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (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 strong internal pull-ups when emitting 1s. During accesses to external data memory that uses 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register. Port 2 also receives the high-order address bits and some control signals during Flash programming verification.

PORT 3:

Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 3 Output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins they are pulled high by the internal Port 3 also receives some control signals for programming and programming verification.

Table 3.1 Port Connections

Port	Function
Port 0	Digital Output to DAC
Port 1	Keypad, LCD
Port 2	LCD Display, Alarm Circuit
Port 3	Digital Input from ADC

3.2.1.4 Timing and Result Circuit

The timing circuit is used to provide the clock pulses to the microcontroller. Here a 12MHz crystal oscillator is connected across the pins XTAL1 and XTAL2, as shown fig 3.4. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by two flip-flop.

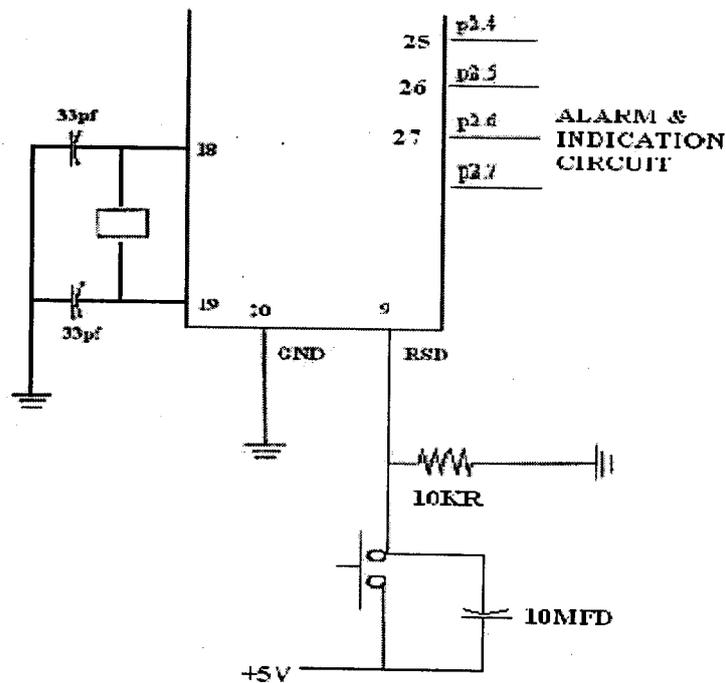


Fig 3.4 Timing and Reset circuit

The reset circuit is used for resetting the microcontroller. The reset button is connected to the keypad panel. When the reset button is pressed, a hardware reset occurs. This is used when manual intervention is required during the forced manual mode.

3.2.2 Input keypad

The input to the microcontroller is given through switches. Six switches from the input keypad. The switches are connected to port 1 as shown in fig 3.5.

Switch 1:

Switch 1 is the CLOSED LOOP ENABLE Switch. It enables the closed loop operation of the system.

Switch 2:

Switch 2 is the INCERASE Switch. It is used to increase the value of the feedback that is to be set as the desired value. The desired value is given in terms of percentage (%) of the feed back from the LED. One press increases the value by 5%.

Switch 3:

Switch 3 is the DECREASE switch. It is used to decrease the value of the feedback. It decreases the value by 1%.

Switch 4:

Switch 4 is the RUN Switch. It is used for putting the system in the closed loop mode. In order to run the closed loop mode, switch 1 is pressed first and then the RUN key.

Switch 5:

Switch 5 is a dummy key which is used when any of the fails, by reprogramming.

Switch 6:

Switch 6 is the RESET key. This is used if any abnormality is encountered leading to forced manual mode.

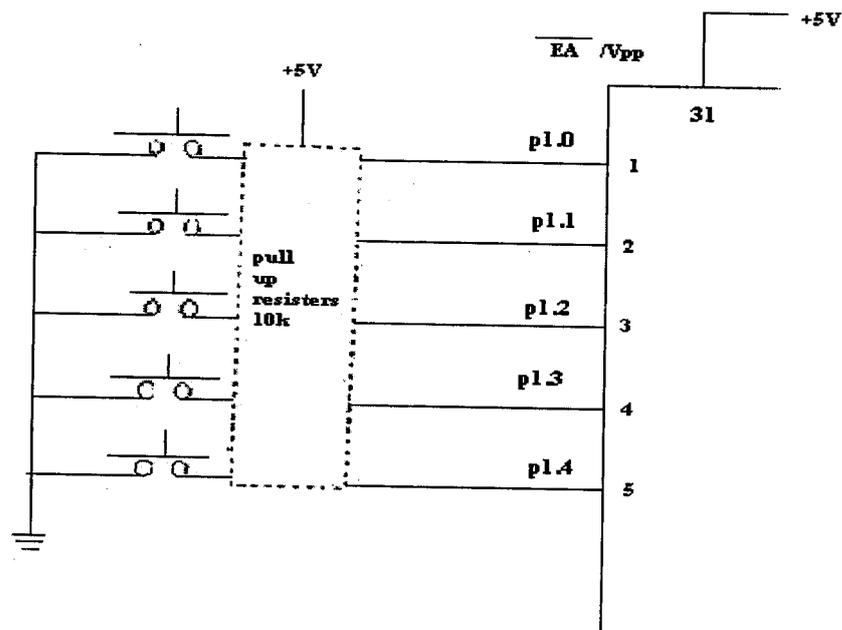


Fig 3.5. Input Keypad

3.2.3 Liquid Crystal Display

The LCD is used for displaying the actual value of the parameter under control and the desired value that is set. The LCD also displays the value of the parameter as and when changed by the operator. The LCD is connected to the microcontroller through the Hitachi 44780 interface as shown in

fig 3.6.

The LCD interface is a parallel bus, allowing simple and fast reading/writing of the data to and from the LCD. Here the LCD is operated in Four bit mode. It requires a minimum of six bits. To wire a microcontroller to an LCD in four bit mode, the top four bits (DB4-7)

are connected. The “R/S” bit is used to select whether data or an instruction is being transferred between the microcontroller and the LCD. If the Bit is set, then the byte at current LCD “Cursor” Position can be read or written. When the Bit is reset, either an instruction is being sent to the LCD or the execution status of the last instruction is read back (whether or not it has completed).

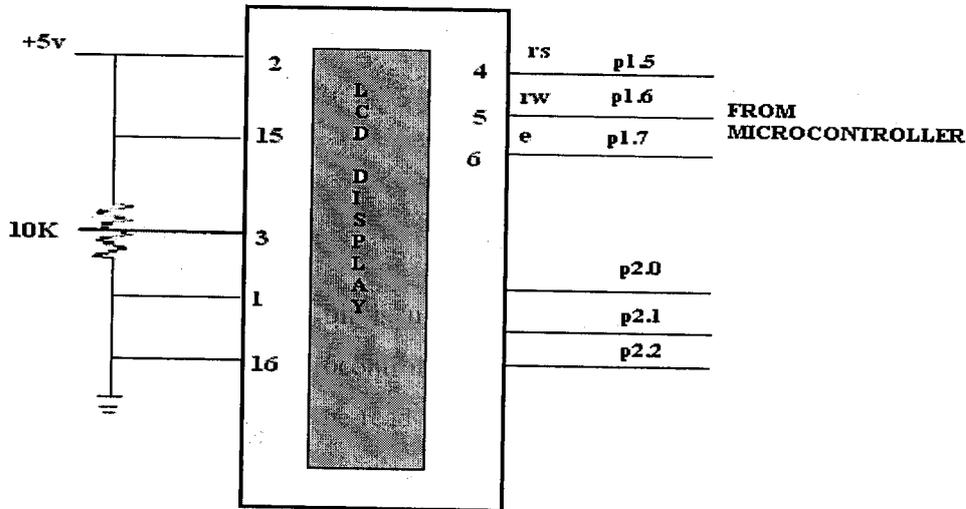


Fig 3.6 LCD Display

3.2.4 Alarm and Indication Circuit

The alarm and indication circuit is used for alerting the operator about certain crucial conditions, under which the system is operating. It consists of four LEDs and one buzzer as shown in fig 3.7. These devices are connected to the microcontroller through the output latch 74573.

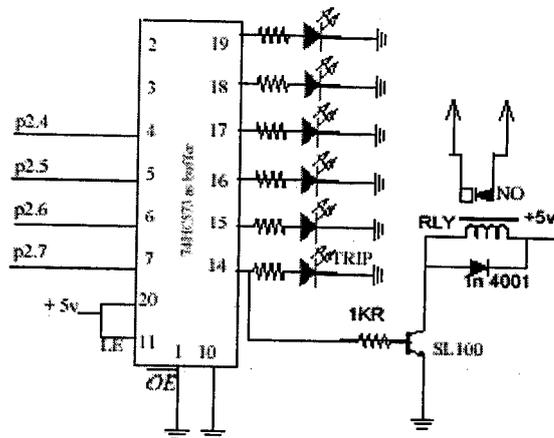


Fig 3.7 Alarm Indication Circuit

The LEDs are operated based on the following scheme of conditions.

LED 1: DEVIATION HIGH

This indicates that the actual value of the output has deviated from the preset value by a large value (>20%). In this case, the LED glows and dims alternatively. This LED also glows when damper opening or closing failed conditions are also reached.

LED 2: FEEDBACK EQUAL TO COMMAND

This LED indicates that the system is perfectly under control, i.e., the output matches the preset value of the parameter. The LED glows after the input and output values are compared every cycle.

LED 3: DAMPER CLOSED FAILED

This LED indicates that Damper closing has failed, i.e., the output cannot be reduced to the preset value even after the command signal to the main LED has reached zero. This condition is analogous to the condition, in which the actual steam parameter value is higher than the preset value, due to the failure of the valve to close completely, even after the minimum current is given.

LED 4: DAMPER OPEN FAILED

This LED indicates that damper opening has failed, i.e., the output cannot be increased to the preset value even after the command signal to the main LED has reached maximum. This condition is analogous to the condition in which the actual steam parameter value is very much lower than the preset value, due to the failure of the valve to open completely, even after the minimum current is given to the valve.

BUZZER:

The buzzer is switched on when the damper opening or closing failed condition is reached. This indicates an emergency in the actual operation requiring manual intervention, i.e., forced manual mode. The buzzer is switched off only if the system is brought to the forced manual mode, by resetting the microcontroller.

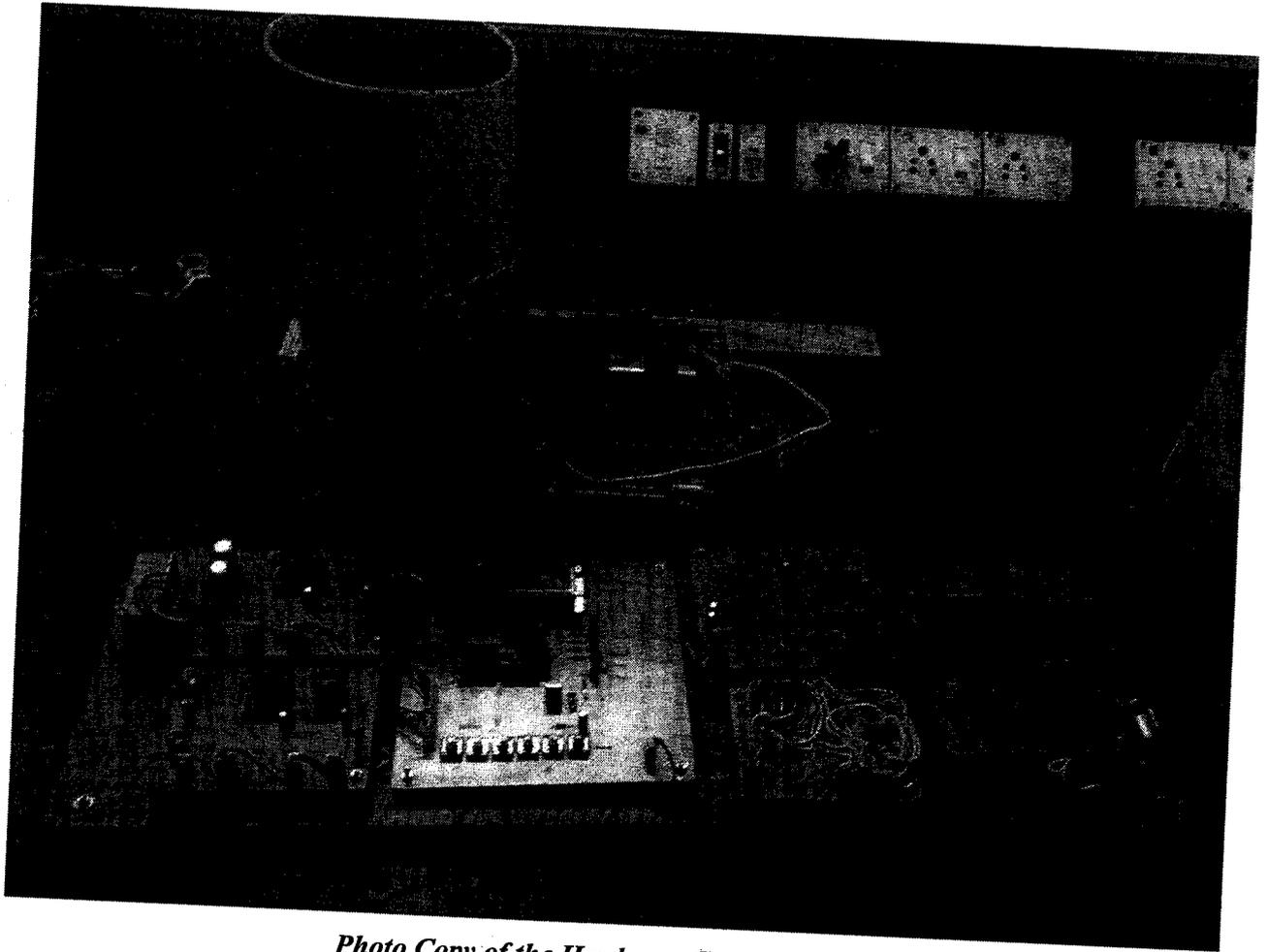


Photo Copy of the Hardware Circuit

3.3 DIGITAL TO ANALOG CONVERSION CIRCUITRY (3)

The Digital to Analog Conversion circuit is used for converting the digital command signal from the microcontroller to equivalent analog value which is fed on the LED. This signal is termed as the command signal. The signal arises from the port 0 of the microcontroller, which is configured as an output port. The digital to analog conversion circuit also contains the LED-LDR couple, where the LED receives the input signal from the microcontroller and the LDR sends the feedback signal. The following are the parts of the DAC circuitry as shown in Fig 3.8.

- DAC Circuit
- Amplifiers
- LED-LDR Couple

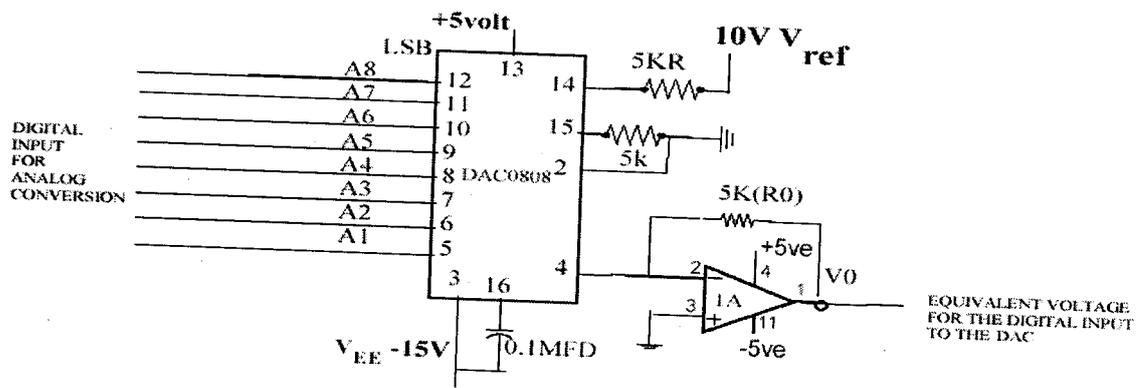


Fig 3.8 Digital to Analog Conversion Circuitry

3.3.1 DAC Circuit

The DAC circuit is built around DAC0808 IC as shown in fig 3.9. The DAC0808 is an 8-bit monolithic digital-to-analog converters provide very high high-speed performance coupled with low cost and outstanding applications flexibility. Advanced circuit design achieves 85 ns settling times with very low “glitch” energy and low power consumption. Monotonic multiplying performance is attained over a wide 20-to-1 reference current ranger. Matching to within 1 LSB between reference and full-scale currents eliminates the need for full-scale trimming in most applications. Direct interface to all popular logic families with full noise immunity is provided by the high swing, adjustable threshold logic input. High voltage compliance complementary current outputs are provided, increasing versatility and enabling differential operation to effectively double the peak-to-peak output swing.

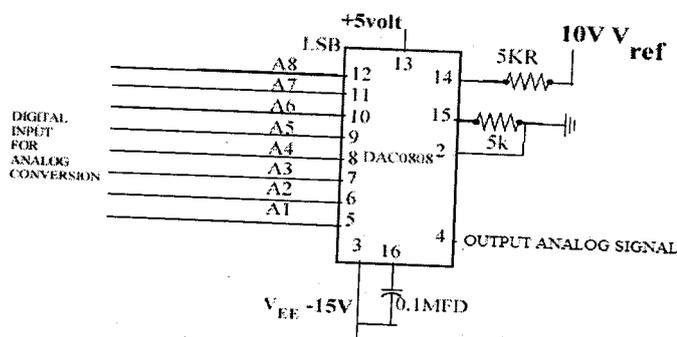


Fig 3.9 DAC IC

In many applications, the outputs can be directly converted to voltage without the need for external op amp. DAC0808 IC guarantees full 8-bit monotonicity and nonlinearities. Device performance is essentially unchanged over the 4.5 V to 18 V power supply range, with 33 mW power consumption attainable at 5 V supplies. The compact size and low power consumption make the DAC08 attractive for portable applications where low cost, high speed and complete input/output versatility are required.

The output voltage from the DAC is amplified to the required level and converted into equivalent voltage signal and applied across the LED. The intensity of the emitted by the LED, and therefore, the light incident on the LDR is directly proportional to the applied voltage across the LED. Thus the potential across the LDR is function of the voltage applied across the LED. This voltage across the LED is taken as the feedback signal, after amplification.

3.3.2 Amplifiers

Here a quad amplifier IC LM324 is used for fabricating the amplifiers as shown in Fig 3.10. LM324 IC consists of four independent high-gain frequency-compensated operation amplifiers that are designed specifically to operate from a single supply over a wide range of voltages. Operation from split supplies also is possible when the difference between the two supplies is 3 V to 30 V and VCC is at least 1.5 V more positive than the input common-mode voltage. The low supply-current drain is independent of the magnitude of the supply voltage.

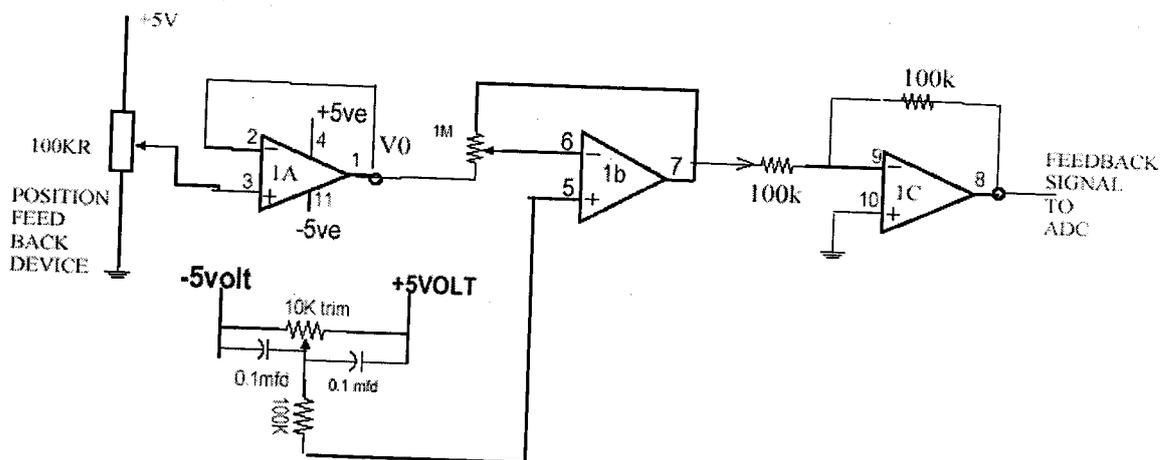


Fig 3.10 Amplifier Section

The op amp is used for obtaining an inverting amplifier (for amplifying the current signal), a buffer (for avoiding the loading effect on the buses) and I- to- V converter. The I- to- V converter converts the input current signal from the DAC into an equivalent voltage value.

3.3.3 LED-LDR Couple

The voltage from the previous stage is applied across the mail LED. This is the actual command signal from the microcontroller to the LED. The intensity of the light incident on the LDR, and therefore the voltage across the LDR is the feedback signal that is to be bed to the microcontroller. The resistance of the LDR is a function of the intensity of falling on it. Thus the actual resistance of the LDR changes with the change in the intensity of the incident light. The resistance across the LDR for zero illumination is around 1 Mega Ohm and the resistance falls to around 500 Ohm. The change in intensity of light can be sensed by measuring the change in voltage drop across the LDR, caused as a result of change in resistance. These values are calibrated and the range of operation is selected such that the LED and LDR operate as much close to the linear region as possible.

COMMAND SIGNAL

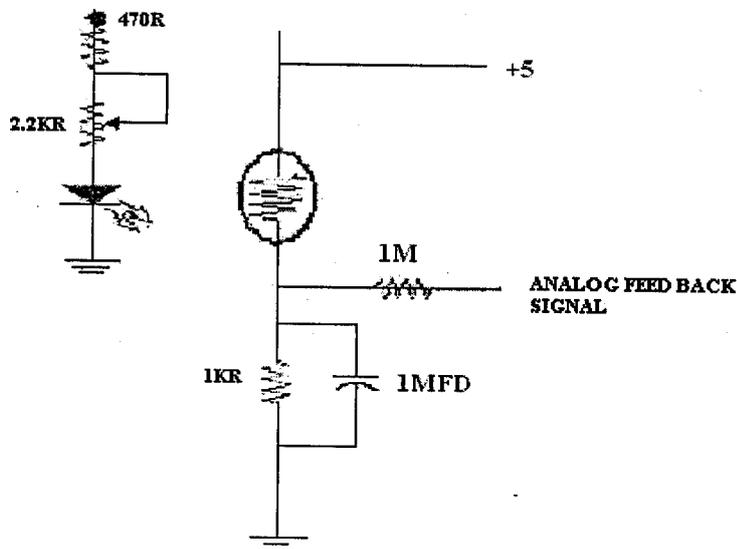


Fig 3.11 LED-LDR Couple

3.4 ANALOG TO DIGITAL CONVERSION CIRCUITRY (3)

The Analog to Digital Conversion circuitry, as shown in Fig 3.12, is used for converting the analog feedback signal from the LDR into digital form, which can be processed by the controller. Here the circuit is built around ADC0804 IC. The sub-circuit for setting the reference voltage for the ADC0804 is op amp-based.

The ADC0804 is CMOS 8-bit successive approximation A/D converter that uses a differential potentiometer ladder –similar to the 256R products. These converters are designed to allow operation with derivative control bus with TRI-STATE output latches directly driving the data bus. These A/Ds appear like memory locations or I/O ports to the microprocessor and no interfacing logic is needed. Differential analog voltage inputs allow increasing the common-mode rejection and offsetting the analog zero input

Voltage value. In addition, the voltage reference input can be adjusted to aloe encoding any smaller analog voltage span to the full 8 bits of resolution. The total conversion time is around 100ns. The total error is in the range of $\pm 1/4$ LSB, $\pm 1/2$ LSB and ± 1 LSB.

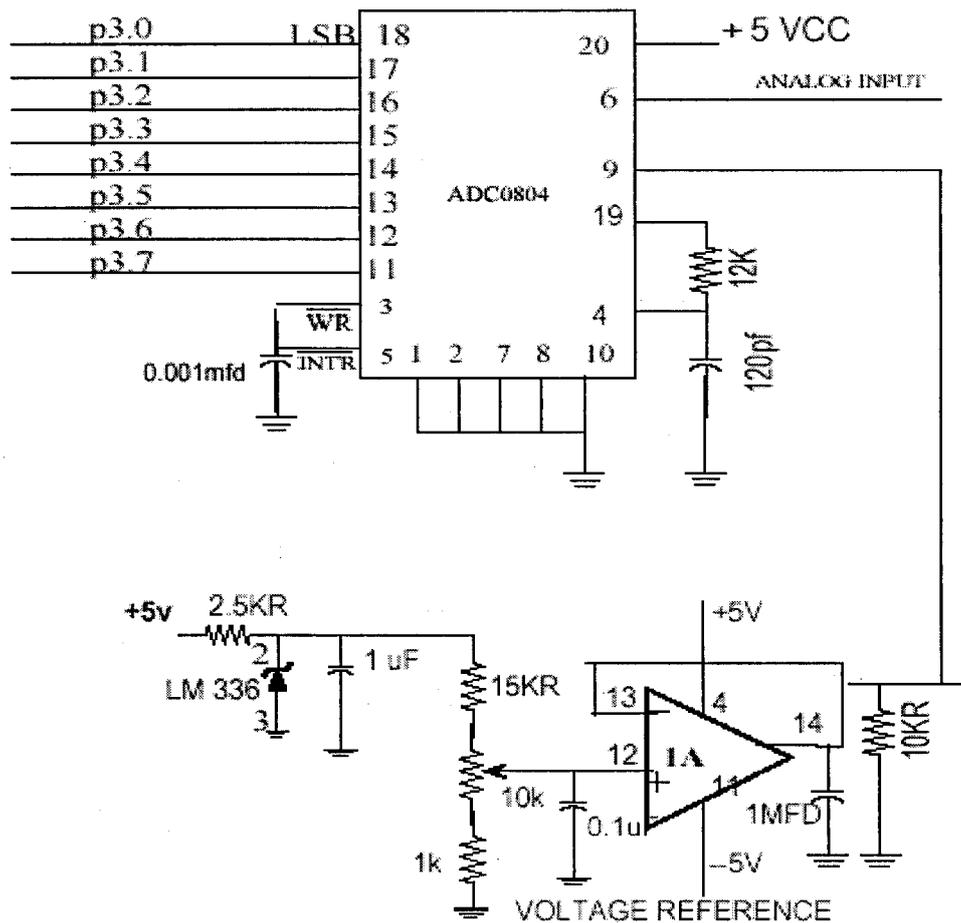


Fig 3.12 Analog to Digital Conversion Circuitry

The analog input from the LDR is converted into equivalent digital signals by the ADC0804 IC. The reference voltage is set for the IC using the circuit as shown above. The digital output from the ADC is fed to the Port 3 of the microcontroller with the help of pull-up resistors.

3.5 POWER SUPPLY CIRCUIT

The power supplies required the project are +5V, -5V, +10V and -15V. In order to obtain constant voltages from the power supply, voltage regulator ICs are used suitably. All the circuits draw power from the a single 18-0-18 signal transformer.

3.5.1 Principle of Operation

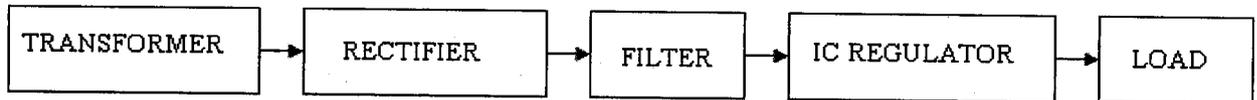


Fig 3.13 Block diagram of power supply

A DC voltage is required for powering all the ICs used in the project. This is achieved using the power supply circuit shown in fig 3.14. the input supply as voltage of 230V rms is stepped down to 18V using a centre-tapped transformer, 18-0-18V/1 A. then, using a W04 bridge rectifier, the ac voltage is converted into equivalent DC voltage. The W04 IC has a maximum peak inverse voltage of 400V and maximum forward current of 1 A. in order to reduce the ripple content of the output DC voltage and for smoothening the DC waveform, capacitors are used in parallel to the supply. Now the voltage available is 18V DC. To obtain the required voltages, suitable voltage regulators are used as shown in the actual circuit diagram.

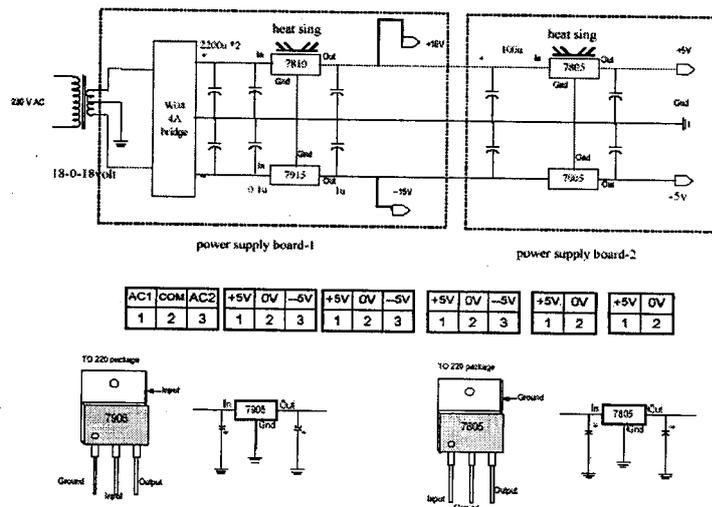


Fig 3.14 Power Supply Circuit

3.5.2 Voltage Regulators

The voltage regulators used here have an output current up to 1A and output voltage of +5, -5, +10 and -15V are required as tabulated in Table 3.2. the regulators are equipped with thermal Overload Protection and Short Circuit Protection. In addition, the regulators also possess output Transistor Safe Operating Area Protection. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and current.

Table 3.2 Regulator ICs

Supply Voltage	IC used	Purpose
-15V	LM7915	DAC(-V _{EE}), I-V Converter
-5V	LM7905	Opamp Supply Voltage - V _{cc}
+10V	LM7810	DAC reference voltage
+5V	LM7805	All other digital ICs

CHAPTER 4

PROTOTYPE MODEL

In the actual system, the parameters being measured are the temperature, pressure or flow rate of steam. Here in the demonstration circuit, the parameter under control is the intensity of light emitted by the main LED. The light intensity can be controlled by adjusted the excitation voltage across the LED. This adjustment is done after comparing the actual intensity with the preset required intensity. The feedback of intensity of light is taken using the Light Dependent Resistor (LDR). The LDR is a transducer whose value of resistance varies with the change in the intensity of the incident light. By measuring the change in the voltage drop across the LDR, the change in the intensity of light can be determined. This analog value of the voltage is converted into digital form and fed to the microcontroller.

The microcontroller scans the LDR voltage every 500ms. This value of voltage is calibrated in percentage. The microcontroller, after obtaining the current value of the feedback signal, compares with the preset value that the user enters while initializing the closed loop system. If the actual value is more than the preset value, the microcontroller reduces the value of the command voltage given to the LED. A corresponding change in the LDR voltage is observed. This value is again sent as a feedback to the microcontroller which again compares it with the preset value. Thus the pressure is iterated continuously until the feedback from the LDR equals the set value. The microcontroller acts in a similar manner, if the actual value is less than the preset value, finally attaining the set condition.

4.1 DISTURBANCE SIMULATION

The system may experience three types of abnormality in the actual working conditions, which are,

- High deviation
- Damper opening Failed
- Damper closing Failed

These conditions are simulated on the demonstration circuit and the system response to those abnormalities is also checked. The high deviation condition is an aberrant condition which is caused when the set value and the actual value differ by an abnormally large value. Here in our project, this value is set as 20%. The microcontroller compares the set value and actual feedback. If their difference is greater than 20%, the microcontroller triggers the

corresponding LED and it is announced to the operator through intermittent buzzer sounds. The microcontroller still tries to control the system so as to assist the operator. The operator can also bring the system under forced manual mode, if the situation demands.

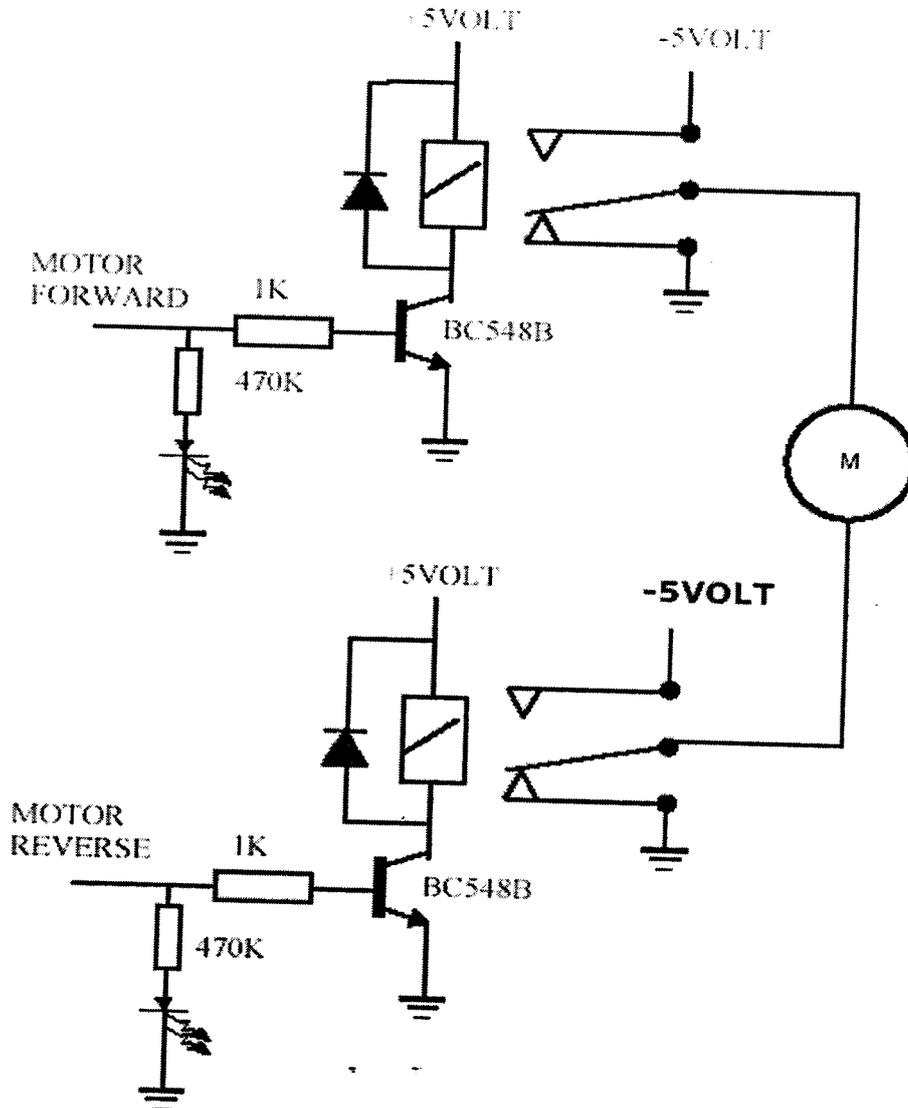
The Damper Opening Failed condition is achieved when the preset value is not reached ever after the value has been completely opened. This is an abnormality which has to be treated with great caution. Such conditions are very rare in actual working conditions, but if they occur, they indicate a serious fault in the steam line. This can be simulated by blocking the path of light between the LED and LDR with an opaque obstacle. Due to this, the actual feedback is very less. Now the microcontroller keeps increasing the command voltage. Due to the blockade, even full excitation of the LED will not yield the desired feedback. Now the microcontroller initiates the forced manual mode. This is indicated by an LED and a continuous buzzer sound. Now the operator has to acknowledge the emergency by resetting the system. On reset, the system reverts to the forced manual mode. Now the operator can vary the LED excitation voltage as required.

The damper Closing Failed condition is met when the feedback signal cannot be reduced to the preset value even after the LED excitation has been completely removed. This disturbance can be simulated by using light from an external light source to be incident on the LDR. The microcontroller initiates the alarm and indication units as described above.

4.2 ACTUAL PARAMETER MEASUREMENT

In the actual system, the pressure, temperature and flow rate can be measured by using the available transducers. These transducers are calibrated to give an output current in the range of 4-20mA. For the measurement of temperature, the MTPS employs K-type thermocouple. For measuring the pressure, pressure transmitters are used. The analog signals from these transducers are directly converted into digital form by the Analog to Digital Conversion Circuit. The set value in the microcontroller is modified so as to match with the requirement.

Fig 4.1 VALVE OPENING AND CLOSING USING MOTOR CONTROL CIRCUIT



The motor control circuit will be driven directly by the micro controller.

The motor control circuit mainly involves in two actions

1. Motor forward
2. Motor reverse

The supply voltage of +5v & -5v is supplied to the circuit.

The motor control circuit will works on the principal of relay action.

Depending on the value of command the motor will be driven by the controller circuit and the feedback will be drawn from the potentiometer.

CHAPTER 5 SOFTWARE MODULE

5.1 FLOW CHART

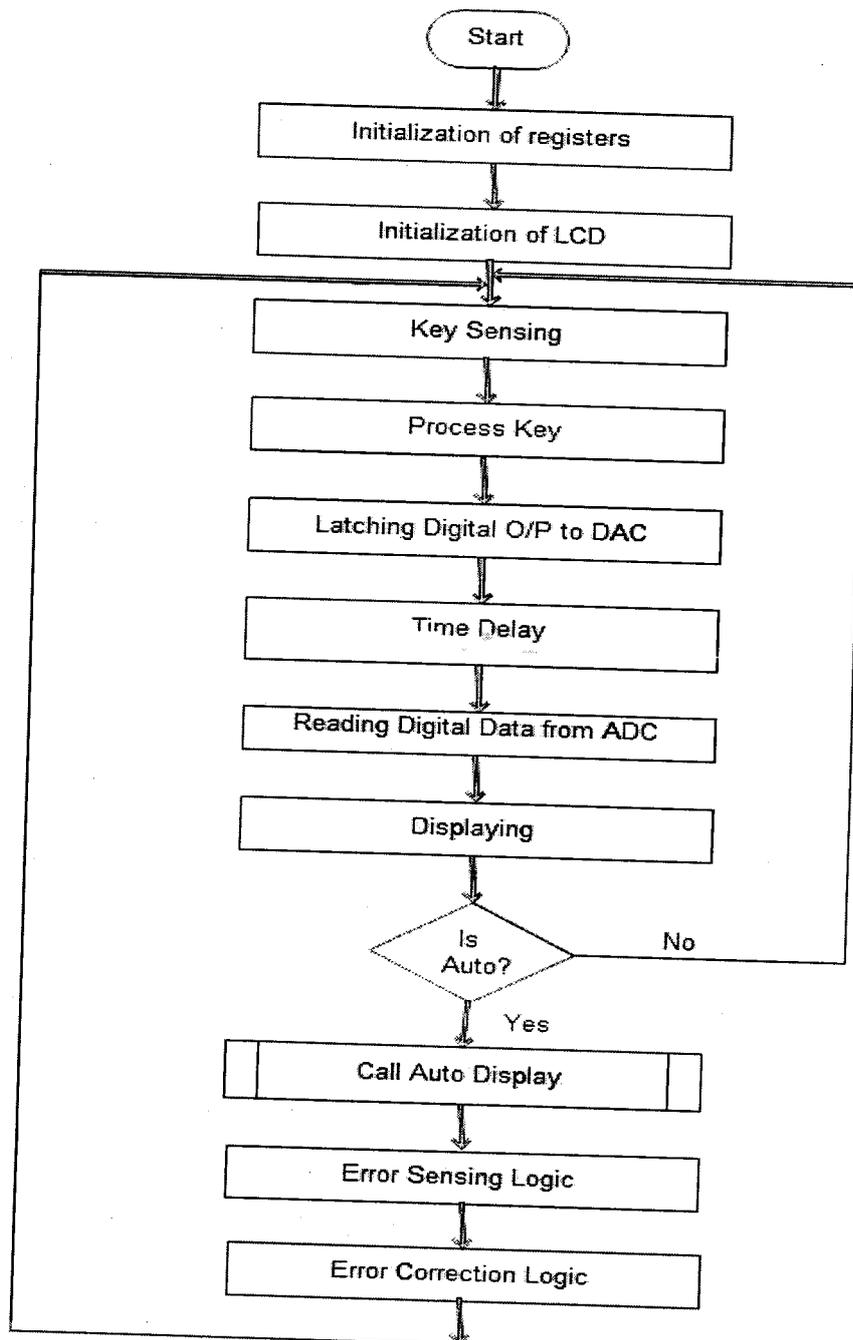


Fig 5.1 Complete Flowchart of the Project

5.2 DESCRIPTION OF THE FLOW CHART

Step 1: Start the program

Step 2: Initially all the registers are initialized. The registers are aliased such that the readability of the program is better. Registers are initialized for storing the input value of command signal from the user and for incrementing and decrementing the value of the command signal, when the operator presses the key pad. Bits are initialized as flags for storing the status of the LEDs and closed loop enabling, etc.

Step 3: Then the LCD display is initialized which displays the welcome screen on resetting or switching on.

Step 4: Then the key that is pressed is sensed using the polling logic. The polling logic is executed continuously with minimum time delay so as to ensure quick response to any key press.

Step 5: The sensed key is then processed. For example, if the DECREASE is pressed, the value of the command is reduced by a value of 1%. This is the key process logic. This is executed by accordingly changing the data stored in the corresponding registers.

Step 6: Then this new value of the register is placed on the command signal output port of the microcontroller. Similarly, the other keys are also processed, when pressed.

Step 7: Then a time delay is introduced. As mentioned earlier the delay is around 500ms. This time delay is introduced for two reasons namely,

- Mechanical components, characterized by high inertias, are involved in the actual system.
- The steam being handled is high is at high temperature and pressure. Any sudden change in the system may cause severe damage to the system.

Step 8: After the time delay, the feedback signal from the LDR is read. This is the input to the microcontroller.

Step 9: The value of the current output is immediately on the LCD screen.

Step 10: Now the mode of operation is checked.

- If the system is in the Auto Control mode, the microcontroller proceeds further.
- If the system is in Forced Manual Mode, the microcontroller goes to step 4, since the system has relinquished its control and has handed over the control to the operator.

Step 11: Auto Display Function is called. The Auto Display Function is used to automatically display the preset value, the current command value and the actual feedback value.

Step 12: The Error Sensing logic is called which compares the actual value and the set value and indicates if there is any error. It also indicates whether the command has to be increased or decreased.

Step 13: then the Error Correction Logic is called, which alters the value of the current value signal across the main LED.

Step 14: then again the microcontroller moves to step 4 sense and key press.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

By automating the valve regulation by this method, it is possible to make the valve regulation system as an automatic system, requiring least manual intervention. Since the valves operate to achieve the preset values of steam parameters, they are beyond the intervention of external variable like load, coal input, etc. Thus they become autonomous, requiring manual intervention only if any emergency is encountered.

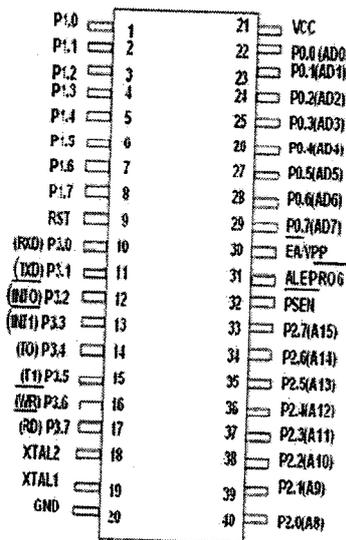
Since the valve systems are identical, the same project can be used for any type of regulation valve, provided the output parameter and the input current to be given to the valve are calibrated accordingly.

6.2 FUTURE SCOPE

In the system developed, the interrupt handling capability of the microcontroller has not been fully exploited. Using interrupt handling mechanism of the microcontroller several valves can be effectively controlled.

APPENDIX 1

Pin Diagram of ATME 89C51 microcontroller:



Pin Description:

VCC

Supply voltage

GND

Ground

Port 0

Port 0 is an 8-bit open drain bidirectional I/O port.

Port 1

Port 1 is an 8-bit bidirectional I/O port internal pullups.

Port 2

Port 2 is an 8-bit bidirectional I/O port internal pullups.

Port 3

Port 3 is an 8-bit bidirectional I/O port internal pullups. Port 3 also serves the functions of various special features of the AT89C5 as listed below:

Port Pin Alternate Functions

P3.0 RXD (serial input port)

P3.1 TXD (serial output port)

P3.2 INT0 (external interrupt 0)

P3.3 INT1 (external interrupt 1)

- P3.4 T0 (timer 0 external input)
- P3.5 T1 (timer 1 external input)
- P3.6 WR (external data memory write strobe)
- P3.7 RD (external data memory read strobe)

RST

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

ALE/PROG

Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (PROG) during Flash programming. In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory. If desired, ALE operation can be disabled setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

PSEN

Program Store Enable is the read strobe to external program memory. When the AT89C51 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 12-volt programming enable voltage (VPP)during Flash Programming, for parts that require 12-volt VPP.

XTAL1

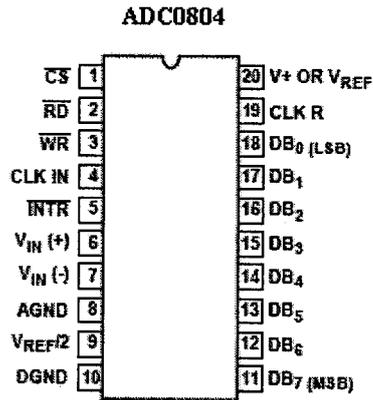
Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2

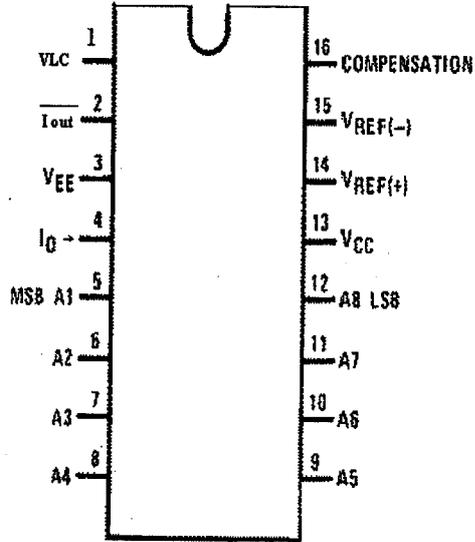
Output from the inverting oscillator amplifier.

APPENDIX 2

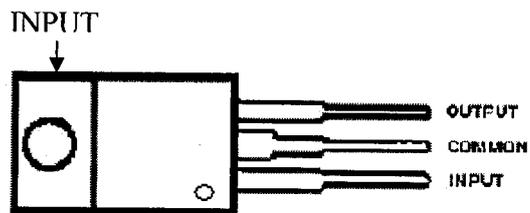
Pin diagram of ADC 0804



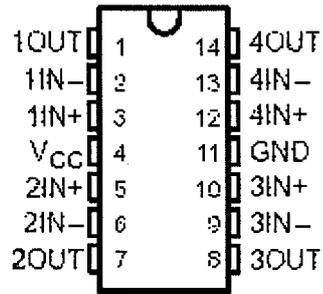
Pin diagram of DAC 0808:



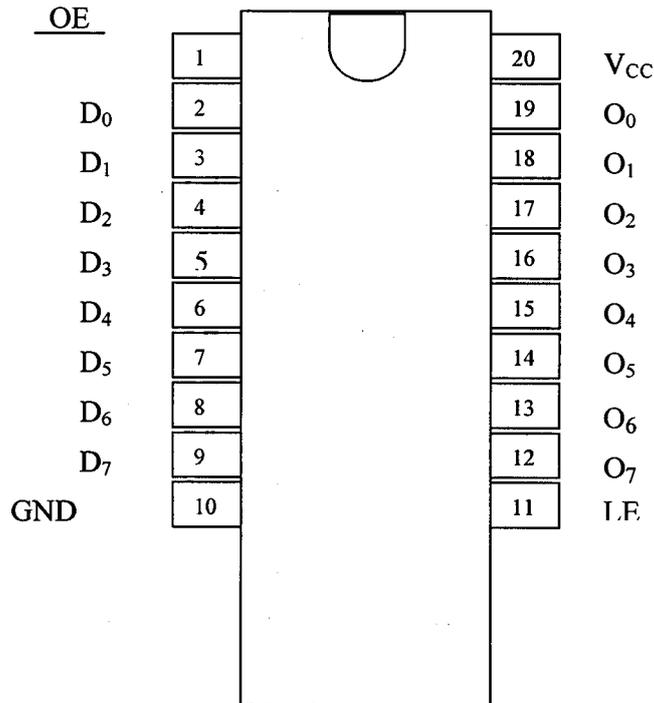
Pin Diagram of LM 7xxx voltage regulators



Pin Diagram of LM324 Quad amps



Pin Diagram of 74573 Latch



Pin Description of 74573

PIN NO	SYMBOL	NAME AND FUNCTION
2,3,4,5,6,7,8,9	D ₀ to D ₇	Data inputs
11	LE	Latch enable input (active HIGH)
1	OE	3-state output enable input (active LOW)
10	GND	Ground (0V)
19,18,17,16,15,14,13,12	Q ₀ to Q ₇	3-state latch outputs
20	V _{CC}	positive supply voltage

Pin Description of LCD Interface (44780):

Pins	Description
1	Ground
2	V _{CC}
3	Control Voltage
4	"R/S" – Instruction/Register Select
5	"R/W" – Read/Write LCD registers
6	"E" Clock
7-14	Data I/O Pins

APPENDIX 3

Microcontroller Program [1]

```

;=====
$MOD51
;-----
RS_PIN          BIT P1.5
RW_PIN          BIT P1.6
E_PIN           BIT P1.7
;
FB_EQ_CMD_BIT   BIT P2.4
DMPR_CLOSE_FAILED_BIT BIT P2.5
DMPR_OPEN_FAILED_BIT BIT P2.6
ALARM_BIT       BIT P2.7; IT IS ACTUATED WHEN
EITHER OF THE ABOVE 2 FAILED CONDITIONS OCCURS
FORCED_MANUAL_BIT EQU ALARM_BIT
;
COMMON_BUS      DATA P0
data_port       data P2
;
;-----
;REGISTERS AND FLAGS ASSIGNMENTS

RUN_FLAG        BIT 20H.0
CLSD_LOOP_FLAG  BIT 20H.1
;
DAMPER_CMD_REG  DATA 25H
DAMPER_FB_REG   DATA 26H
port_read_reg   DATA 27H
DAMPER_CMD_ADJUSTED_REG DATA 28H
deci_msb_output_reg DATA 29H
deci_lsb_output_reg DATA 2AH
hex_input_reg   DATA 2BH
hex_output_reg  DATA 2CH
higher_deci_input_reg DATA 2DH
lower_deci_input_reg DATA 2EH
;-----
ORG 00H
JMP MAIN

```

```

;-----
MAIN:   call initialize_lcd
        CALL LCD_TEST
        call q1second
        call q1second
        call welcome_display
        call q1second
        call q1second
        call reg_initialisation
;-----
CONTINUE:
        MOV C, DMPR_CLOSE_FAILED_BIT
        ORL C, DMPR_OPEN_FAILED_BIT
        MOV ALARM_BIT, C
;
CONTINUE11:
        CALL DATA_LATCH
        CALL READ_FEED_BACK
        MOV DPL, R6
        MOV DPH, R7
        INC DPTR
        MOV R6, DPL
        MOV R7, DPH
        MOV A, R6
        ORL A, R7
        JNZ CC;V
        CALL SET_MODE_DISPLAY
CC:     CALL KEY_SENSOR
        JNB RUN_FLAG, CONTINUE11
;
        JB CLSD_LOOP_FLAG, PRO1
        CALL DATA_LATCH
        CALL READ_FEED_BACK
        CALL RUN_MODE_DISPLAY
        SJMP CONTINUE
PRO1:  CALL DATA_LATCH_CLSD_LOOP
        CALL Q100MILLISECOND
        CALL Q100MILLISECOND
        CALL Q100MILLISECOND
        CALL Q100MILLISECOND

```



```

PRO11:;^PRO1
    CLR FB_EQ_CMD_BIT; DECLARES THAT FEEDBACK IS
NOT EQUAL TO AIMED OPENING
    JC PRO12;V
    MOV A, DAMPER_CMD_ADJUSTED_REG
    CJNE A, #63H, PRO111;V
    SETB DMPR_OPEN_FAILED_BIT
    JMP CONTINUE
;
PRO111:;^PRO11
    CLR DMPR_OPEN_FAILED_BIT
    INC DAMPER_CMD_ADJUSTED_REG
    CALL DATA_LATCH_CLSD_LOOP
    CALL READ_FEED_BACK
    CALL RUN_MODE_DISPLAY
    JMP CONTINUE;^
pro12:;^PRO11
    MOV A, DAMPER_CMD_ADJUSTED_REG
    CJNE A, #00, PRO121;V
    SETB DMPR_CLOSE_FAILED_BIT
    JMP CONTINUE
;
PRO121:
    CLR DMPR_CLOSE_FAILED_BIT
    DEC DAMPER_CMD_ADJUSTED_REG
    CALL DATA_LATCH_CLSD_LOOP
    CALL READ_FEED_BACK
    CALL RUN_MODE_DISPLAY
    JMP CONTINUE;^
;-----
DATA_LATCH_CLSD_LOOP:
    MOV p0, DAMPER_CMD_ADJUSTED_REG
    RET
;-----
DATA_LATCH:
    MOV p0, DAMPER_CMD_REG
    RET
;-----
DAMPER_LATCH_ROUTINE:
MOV p0, DAMPER_CMD_REG

```

```

RET
;
;-----
READ_FEED_BACK:
MOV A, p3;
CJNE A, #63H, RFB1
MOV DAMPER_FB_REG, A;
RET
RFB1: JC RFB2
CALL FB_HIGH_DISPLAY
RFB2: MOV DAMPER_FB_REG, A;
RET
;-----
FB_HIGH_DISPLAY:
CLR A
MOV DPTR, #FHD1
CALL DISPLAY_STRING
MOV A, #20H
MOV DPTR, #FHD1
CALL DISPLAY_STRING
call q1second
call q1second
RET
;
; '0123456789ABCDEF'
FHD1: DB ' FEEDBACK HIGH ',00
FHD2: DB ' ',00
;-----
reg_initialisation:
;
CLR FB_EQ_CMD_BIT
CLR DMPR_CLOSE_FAILED_BIT
CLR DMPR_OPEN_FAILED_BIT
CLR ALARM_BIT
;
CLR CLSD_LOOP_FLAG
MOV DAMPER_CMD_REG, #0H
CLR RUN_FLAG
mov port_read_reg, #0ffh
RET

```

```

;-----
KEY_SENSOR:
    orl p1, #00011111b; NOT TO DISTURB THE LCD
    mov a, p1
    jb run_flag, ks_1
    jnb acc.0,
CLSD_LOOP_KEY;process_value_pickup_key_a;
    JNB ACC.1, DOWN_KEYA
    jnb acc.2, up_key_a
ks_1:    jnb acc.4, run_keyA
    mov port_read_reg, #0ffh
key_return: ret
;
run_keyA: JMP run_key
DOWN_KEYA: JMP DOWN_KEY
up_key_a: jmp up_key
;-----]
CLSD_LOOP_KEY:; HERE BY ENABLING IT, CLIOSED LOOP
GAIN IS BEING ATTAINED
;
    SETB CLSD_LOOP_FLAG
    MOV DAMPER_CMD_ADJUSTED_REG, DAMPER_CMD_REG
    CALL CLSD_LOOP_INITIATED_DISPLAY
    call q1second
    call q1second
    RET
;-----
CLSD_LOOP_INITIATED_DISPLAY:
    MOV DPTR, #CLID1
    CLR A
    CALL DISPLAY_STRING
    MOV DPTR, #CLID2
    MOV A, #20H
    CALL DISPLAY_STRING
    RET
;
    '0123456789ABCDEF'
CLID1:    DB    'CLSD LOOP ENABLD', 00
CLID2:    DB    ' ', 00
;-----
down_key:

```

```

;
MOV A, DAMPER_CMD_REG; DATA IS BETWEEN 0 AND
99% IN DECIMAL
cjne A, #0, DK1
;dec A, #01H
;CJNE A, #64H, DK_1
;DEC A

DK11:;UK11: ;V
MOV DAMPER_CMD_REG, A
CALL SET_MODE_DISPLAY
CALL DAMPER_LATCH_ROUTINE; IT SENDS DATA
PROPORTIONAL TO THE % SET
RET
DK_1: JC DK11;V
CLR A
SJMP DK11;^

DK1: DEC A
JMP DK11
;-----
UK_RET: RET
up_key:
;
MOV A, DAMPER_CMD_REG; DATA IS BETWEEN 0 AND
99% IN DECIMAL
ADD A, #05H
CJNE A, #64H, UK_1
DEC A
UK11: ;V
MOV DAMPER_CMD_REG, A
CALL SET_MODE_DISPLAY
CALL DAMPER_LATCH_ROUTINE; IT SENDS DATA
PROPORTIONAL TO THE % SET
RET
UK_1: JC UK11;V
CLR A
SJMP UK11;^
;-----
run_key:

```

```

    cpl run_flag
    jb run_flag, rk_1
    CLR CLSD_LOOP_FLAG
    call set_mode_display
    ret
rk_1: call run_mode_display
rk_2:  ret
;=====DISPLAY
ROUTINES=====
LCD_TEST:
    mov dptr, #wel_1x
    clr a
    call display_string
;
    mov dptr, #wel_2x
    mov a, #20h
    call display_string
    ret
;
wel_1x: db
        0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0f
        fh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh, 00
wel_2x: db
        0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh,0f
        fh,0ffh,0ffh,0ffh,0ffh,0ffh,0ffh, 00
;-----
welcome_display:
;
    mov dptr, #wel_1
    clr a
    call display_string
;
    mov dptr, #wel_2
    mov a, #20h
    call display_string
    ret
wel_1:  db  '   POWER PLANT   ', 00
wel_2:  db  '   AUTOMATION   ', 00;
;-----
set_mode_display:

```

```

;
MOV DPTR, #SMD1
CLR A
CALL DISPLAY_STRING
MOV A, #20H
MOV DPTR, #SMD2
CALL DISPLAY_STRING
;
MOV A, #21H
CALL LOCATE_CURSOR
MOV HEX_INPUT_REG, DAMPER_CMD_REG
CALL ONE_BYTE_HEX_TO_DECI
MOV A, deci_lsb_output_reg
PUSH ACC
ANL A, #0F0H
SWAP A
ORL A, #30H
CALL SEND_DATA
POP ACC
ANL A, #0FH
ORL A, #30H
CALL SEND_DATA
MOV A, #2BH
CALL LOCATE_CURSOR
MOV A, DAMPER_FB_REG
MOV HEX_INPUT_REG, A
CALL ONE_BYTE_HEX_TO_DECI
MOV A, DECI_LSB_OUTPUT_REG
PUSH ACC
ANL A, #0F0H
SWAP A
ORL A, #30H
CALL SEND_DATA
POP ACC
ANL A, #0FH
ORL A, #30H
CALL SEND_DATA
;
MOV A, #24H
CALL LOCATE_CURSOR

```

```

RET
;
;          '0123456789ABCDEF'
SMD1: DB 'SET :MANUAL:FB ', 00
SMD2:  DB '      %:      :      %', 00
;-----
run_mode_display:
    JB CLSD_LOOP_FLAG, RMDIS1
    CLR A
    MOV DPTR, #RMD1
    CALL DISPLAY_STRING
    MOV HEX_INPUT_REG, DAMPER_CMD_REG
    CALL ONE_BYTE_HEX_TO_DECI
    MOV A, deci_lsb_output_reg
    PUSH ACC
    ANL A, #0F0H
    SWAP A
    ORL A, #30H
    CALL SEND_DATA
    POP ACC
    ANL A, #0FH
    ORL A, #30H
    CALL SEND_DATA
;2ND LINE
    MOV A, #20H
    MOV DPTR, #RMD2
    CALL DISPLAY_STRING
    MOV HEX_INPUT_REG, DAMPER_FB_REG
    CALL ONE_BYTE_HEX_TO_DECI
    MOV A, deci_lsb_output_reg
    PUSH ACC
    ANL A, #0F0H
    SWAP A
    ORL A, #30H
    CALL SEND_DATA
    POP ACC
    ANL A, #0FH
    ORL A, #30H
    CALL SEND_DATA
    MOV A, #30H

```

```

CALL LOCATE_CURSOR
RET
;          '0123456789ABCDEF'
RMD1:     DB  'DMPR OPN CMD %', 00
RMD2:     DB  'DMPR OPN FB  %', 00
;
RMDIS1:
  CLR A
  MOV DPTR, #RMD3A
  CALL DISPLAY_STRING
  MOV A, #09H
  CALL LOCATE_CURSOR
  MOV HEX_INPUT_REG, DAMPER_CMD_REG
  CALL ONE_BYTE_HEX_TO_DECI
  MOV A, deci_lsb_output_reg
  PUSH ACC
  ANL A, #0F0H
  SWAP A
  ORL A, #30H
  CALL SEND_DATA
  POP ACC
  ANL A, #0FH
  ORL A, #30H
  CALL SEND_DATA
  MOV A, #20H
  MOV DPTR, #RMD4B
  CALL DISPLAY_STRING
  MOV HEX_INPUT_REG, DAMPER_CMD_ADJUSTED_REG
  CALL ONE_BYTE_HEX_TO_DECI
  MOV A, #24H
  CALL LOCATE_CURSOR
  MOV A, deci_lsb_output_reg
  PUSH ACC
  ANL A, #0F0H
  SWAP A
  ORL A, #30H
  CALL SEND_DATA
  POP ACC
  ANL A, #0FH
  ORL A, #30H

```

```

CALL SEND_DATA
MOV A, #2DH
CALL LOCATE_CURSOR
;
MOV HEX_INPUT_REG, DAMPER_FB_REG
CALL ONE_BYTE_HEX_TO_DECI
MOV A, deci_lsb_output_reg
PUSH ACC
ANL A, #0F0H
SWAP A
ORL A, #30H
CALL SEND_DATA
POP ACC
ANL A, #0FH
ORL A, #30H
CALL SEND_DATA
MOV A, #30H
CALL LOCATE_CURSOR
RET

```

```

; '0123456789ABCDEF'
;RMD3: DB 'CLSD LOP CMD %', 00
;RMD4: DB 'CLSD LOP FB %', 00
RMD3A: DB 'AIMED FB %', 00
RMD4B: DB 'CMD % : FB %', 00

```

```

;=====CODE CONVERSION
ROUTINES=====

```

```

one_byte_DECI_TO_HEX:
;it converts decimal value into hex value
;registers used
;r0, r1 as temporary not refered an where
;lower_deci_input_reg : IT CAN HOLD VLAUE FORM
00 TO 99D
;higher_deci_input_reg: IT CAN HOLD VALUE FORM
00 TO 02D ONLY
;hex_output_reg |HEXADECIMAL VALUE
OUTPUT
;; USING THIS ROUTINE DECIMAL VALUE LESS THAN
OR EQUAL TO ONE BYTE HEX CAN ONLY BE CONVERTE

```

```

; FORM DECIMAL TO HEX
;
MOV A, lower_deci_input_reg
MOV R1, A
ANL A, #0F0H
MOV R0, A
MOV A, R1
ANL A, #0FH
MOV R1, A
CJNE R0, #00H, XXY
sjmp xxz
XXY:
DEC R0
MOV A, R0
MOV B, #0AH
MUL AB
INC R1
ADD A, R1
ADD A, #09H
mov r1, a
;
xxz:
mov b, #64h
mov a, higher_deci_input_reg
mul ab
add a,r1
mov hex_output_reg, a
ret

```

```

;-----
one_byte_hex_to_deci:
;
; CONVERTS ONE BYTE OF HEX VALUE IN TO DECIMAL
VALUE
; HEX_INPUT_REG: hex value is input here
; DECI_MSB_output_REG: msb output is placed here
as output, the value can be of atmost 02d
; DECI_LSB_output_REG: lsb output is place here
as output, the value can be from 00 to 99d
;
; r3, r4: temporary registers
mov deci_msb_output_reg, #0h;

```

```

    mov deci_lsb_output_reg, #00h;
    anl a, #0f0h
    mov r4, a
    jz sb_1;go to convert least most nibble
    mov deci_lsb_output_reg, #16h
sb_4:    djnz r4, sb_2
sb_1:   mov a, hex_input_reg;32h
        anl a, #0fh
        mov r4, a
        jz sb_5
sb_8:    mov a, deci_lsb_output_reg
        add a, #01
        da a
        mov deci_lsb_output_reg, a
        jnc sb_6
sb_7:    djnz r4, sb_8
        jmp sb_5

sb_6:   mov a, deci_msb_output_reg
        add a, #01
        mov deci_msb_output_reg, a
        jmp sb_7
sb_2:   mov a, deci_lsb_output_reg
        add a, #16h
        da a
        mov deci_lsb_output_reg, a
        jc sb_3
        jmp sb_4
sb_3:   mov a, deci_msb_output_reg
        add a, #01
        da a
        mov deci_msb_output_reg, a
        jmp sb_4
sb_5:   pop 04
        pop 03
        ret

;=====DELAY ROUTINES=====
;TIMING-DELAY ROUTINES STARTS
;
Q50MICROSECOND:

```

```

        PUSH 00
        MOV R0, #0a0H
STEP1:  DJNZ R0, STEP1
        POP 00
        RET
        ;
Q200MICROSECOND: CALL Q50MICROSECOND
                CALL Q50MICROSECOND
                CALL Q50MICROSECOND
                CALL Q50MICROSECOND
                RET
Q10MILLISECOND:  call q5millisecond
                call q5millisecond
                ret
        ;
q5millisecond:    PUSH 00
                PUSH 01
                MOV R0, #05
                MOV R1, #05
SWA:           DJNZ R0, SEA
                DJNZ R1, SSA
                pop 01
                pop 00
                RET
SSA:           MOV R0, #05
                JMP SWA
SEA:           CALL Q200MICROSECOND
                JMP SWA
        ;
Q20MILLISECOND: CALL Q10MILLISECOND
                CALL Q10MILLISECOND
                RET
        ;
Q30MILLISECOND: CALL Q10MILLISECOND
                CALL Q10MILLISECOND
                CALL Q10MILLISECOND
                RET
Q100MILLISECOND: PUSH 00
                PUSH 01
                MOV R0, #05

```

```

                MOV R1, #100
SW:             DJNZ R0, SE
                DJNZ R1, SS
                POP 01
                POP 00
                RET
SS:            MOV R0, #05
                JMP SW
SE:            CALL Q200MICROSECOND
                JMP SW
;
;
Q1SECOND:     PUSH 00                ;1388H = 5000D  FOR
MAKING 200 MICROSECOND ROUTINE TO 1SECOND
                PUSH 01
                MOV R0, #04H
                MOV R1, #011H        ;11 MAY BE CHANGED TO
88 IN ACTUAL
QWE:          DJNZ R1, QWEF
                DJNZ R0, QWE2
                POP 01
                POP 00
                RET
QWE2:         MOV R1, #088H
                JMP QWE
;
QWEF:         CALL Q200MICROSECOND
                JMP QWE
;-----
INITIALIZE_LCD:
                MOV A, #030H
                CALL SEND_COMMAND
                CALL Q10MILLISECOND
                MOV A, #030H
                CALL SEND_COMMAND
                CALL Q10MILLISECOND
                MOV A, #030H
                CALL SEND_COMMAND
                CALL Q10MILLISECOND
                MOV A, #02H

```

```

CALL SEND_COMMAND
;
MOV A, #11101000B
CALL SEND_COMMAND
MOV A, #00H
CALL SEND_COMMAND
MOV A, #01H
CALL SEND_COMMAND
MOV A, #10010100B
CALL SEND_COMMAND
MOV A, #10001111B
CALL SEND_COMMAND
RET

```

```

;-----
;Display String Pointed by DPTR
;Input:
;  A      - Starting position on LCD
;           for display
;           (      D7 D6 D5 - LINE NUMBER.
;           D4 D3 D2 D1 D0 - CHARACTER NUMBER.
;           Both line and character numbers
;           start from 0.)
;
;  DPTR - Starting address of the String
;-----

```

```

DISPLAY_STRING:

```

```

    ACALL LOCATE_CURSOR           ;Locate the
cursor at given Position

```

```

REPEAT_DISP_STRING:

```

```

    CLR A
    MOVCA,@A+DPTR                ;Get the character
from string

```

```

    INC DPTR
    JZ END_DISP_STRING           ;Check for end of
string

```

```

    LCALL SEND_DATA              ;Send the data to
LCD

```

```

    SJMP REPEAT_DISP_STRING

```

```

END_DISP_STRING:

```

```

    RET
;-----
;Clear Display
;-----
CLEAR_LCD:
    MOV     A,#01H           ;Command to clear
display
    ACALL  SEND_COMMAND     ;Send the command
    call  q10millisecond
    RET
;-----
;Locate the cursor on LCD
; (for 20 characters LCD)
;Input:
;  A      - Starting position on LCD
;           for display
;           (      D7 D6 D5 - LINE NUMBER.
;           D4 D3 D2 D1 D0 - CHARACTER NUMBER.
;           Both line and character numbers
;           start from 0.)
;-----
LOCATE_CURSOR:
    PUSHB
    MOV  B,A
    ANL A,#0E0H           ;Mask the character
number
    SWAPA
    RR  A                 ;Get the line number
    ADD A,#LOCATE_CURSOR_TABLE-LOC_CURSOR
                                ;Add with cursor locating
data table address
    MOVCA,@A+PC           ;Get the command byte
LOC_CURSOR:
    ANL B,#01FH
    ORL A,B               ;Add the character
position
    ACALL  SEND_COMMAND     ;Send the command to
locate the cursor
    POP  B
    RET

```

```

LOCATE_CURSOR_TABLE:
    DB 080H,0C0H,094H,0D4H
SEND_COMMAND:
    PUSH ACC
    ANL A,#0F0H
    SWAPA
    ACALL SEND_COMMAND_NIBBLE ;Send higher
nibble of the command
    POP ACC
    ANL A,#0FH
    ACALL SEND_COMMAND_NIBBLE ;Send lower
nibble of the command
    RET
SEND_COMMAND_NIBBLE:
    CLR E_PIN
    CLR RS_PIN ;RS_PIN=0 for
instruction
    CLR RW_PIN ;R/W_PIN=0 for write
operation
    ACALL SEND_NIBBLE ;Send the nibble
    CALL Q50MICROSECOND
    setb rs_pin
    setb rw_pin
    CALL Q200MICROSECOND
    RET
SEND_NIBBLE:
    ANL A,#0FH
    ANL DATA_PORT,#0F0H
    ORL DATA_PORT,A ;Send data to the
port
    NOP
    ;Give a transition pulse to Enable pin
    RET
;-----
SEND_DATA:
    PUSH ACC
    ANL A,#0F0H
    SWAPA
    ACALL SEND_DATA_FORNIBBLE ;Send higher
nibble

```

```

    POP ACC
    ANL A, #0FH
    ACALL SEND_DATA_FORNIBBLE ;Send lower
nibble
    RET
SEND_DATA_FORNIBBLE:
    CLR E_PIN
    SETB RS_PIN ;RS_PIN=1 for DATA
    CLR RW_PIN ;R/W_PIN=0 for write
operation
    ACALL SEND_NIBBLE ;Send a nibble
    CALL Q50MICROSECOND
    clr RS_PIN
    setb RW_PIN
    RET
;
;Set DDRAM address 0 in address counter and bring
;the shift position to original position.
;-----
RETURN_HOME_LCD:
    MOV A, #02H ;Command to return
home
    ACALL SEND_COMMAND ;Send the command
    ACALL q50microsecond ;Call a delay
    RET
;-----
;Cursor On/Off
;Input: Set F0 bit to switch On Cursor
; Clear F0 bit to switch Off Cursor
;-----
cursor_on_off:
    MOV A, #00FH ;Move command to on/off
cursor
    MOV C, 21h.7;F0
    MOV ACC.1, C
    ACALL SEND_COMMAND ;Send the command
    RET
;-----
end

```

REFERENCES

- [1] Myke Predko (2001), 'Programming and Customizing the 8051 Microcontroller', The McGraw-Hill Companies, Inc., Second Edition.
- [2] Rajkamal (2000), 'The Concepts and Features of Microcontrollers', Wheeler Publishing House, First Edition.
- [3] Roy Choudhury .D, Shail B.Jain (2005), 'Linear Integrated Circuits', New Age International Private Limited, Second Edition.
- [4] Richard Barnett (1995), 'Programming in ATMEL', Narosa Publishing House, New York.