PROJECT REPORT

**“Automatic Rain Gauge”**



by

KARAN THACKER (EC-38)

Under guidance of

**Prof. Rachit K. Dana**

**Department of Electronics &Communication**

**Faculty of Technology**

**Dharamsinh Desai University, Nadiad**

**March 2016**

**CERTIFICATE**



This is to certify that the Project entitled **Microcontroller Based Rain Gauge** is the bonafide work of Mr. **Karan Thacker** Roll No.: **EC-038** Identity No.:**13ECUOS091** of B.Tech. Semester **VI** in the branch of **Electronics and Communication** during the academic year **2015-2016**.

Prof.Rachit Dana Dr. Nikhil J. Kothari

Project Supervisor Head of Department

Date: Date:

**Acknowledgement**

My sincere thanks to Prof. Rachit K. Dana for giving his help and guidance in making my project possible to achieve and complete . I am also thankful to Prof. Manish K Patel for the understanding and enlightenment received related to programming the controller from his book and to Vasim Vora Sir to clear my queries regarding the project.

My sincere thanks to all individuals who directly or indirectly supported in making this project from good to better. Thank you.

**ABSTRACT**

The report is based on the making and implementation of an automatic rain gauge. Now most people know that rain gauge is a very simplistic device which is just a vessel with markings on it which gathers water from rain so that the amount of rainfall can be measured during the day. The automatic rain gauge is based on a micro controller and a sensing circuit consisting of logic gates to check the level of water and display the output on LCD. This particular gauge that I have made is better in always than the common rain gauges as this has better accuracy which rules out the possibility of parallax or any other anomalies which might be occurring while taking the readings of waterfall by human eye as well as it constantly notifies the current status and immediate changes in the readings.

**INDEX**

1.Introduction……………………………………………………..…………………………7

2.Block Diagram and Circuit……………………………………………………………8

3.Hardware description………………………………………………………………….9

3.1Microcontroller…………………………………………….............9

3.1.1 Description……………………………………….............9

3.1.2 Feature………………………………………….…………….9

3.1.3 Pin Diagram…………………………………................10

3.2 Liquid Crystal Display (LCD-2X16)…………………………..13

3.2.1 Description…………………………………………………13

3.2.2 Signals to LCD…………………………………………….13

3.2.3 Logic Status on Control Lines …………………….13

3.2.4 Writing Data to LCD……………………………………14

3.2.5 Pin Diagram……………………………………………….14

4. Working…………………………………………………………………………………….16

4.1 Sensing circuit……………………………………………….16

4.2 Interfacing of circuit with controller………………16

4.3 Interfacing of LCD with controller………………….16

4.4 Proteus simulation………………………………………..17

5 Practical Application……………………………………………………..……………18

6 Program Code.................................................................................19

7 Future Scope and Modification…………………………………………………..27

8 Conclusion……………………………………………………………..…………………..28

9 Bibliography………………………………………………………..……………………..29

List of Figures

Fig2.1 block diagram………………………………………………………………………..8

Fig2.2 circuit diagram……………………………………………………………………….8

Fig3.1 pin diagram………………………………………………………………………….10

Fig3.2 LCD picture…………………………………………………………………………..14

Fig4.1 picture of working setup………………………………………………………17

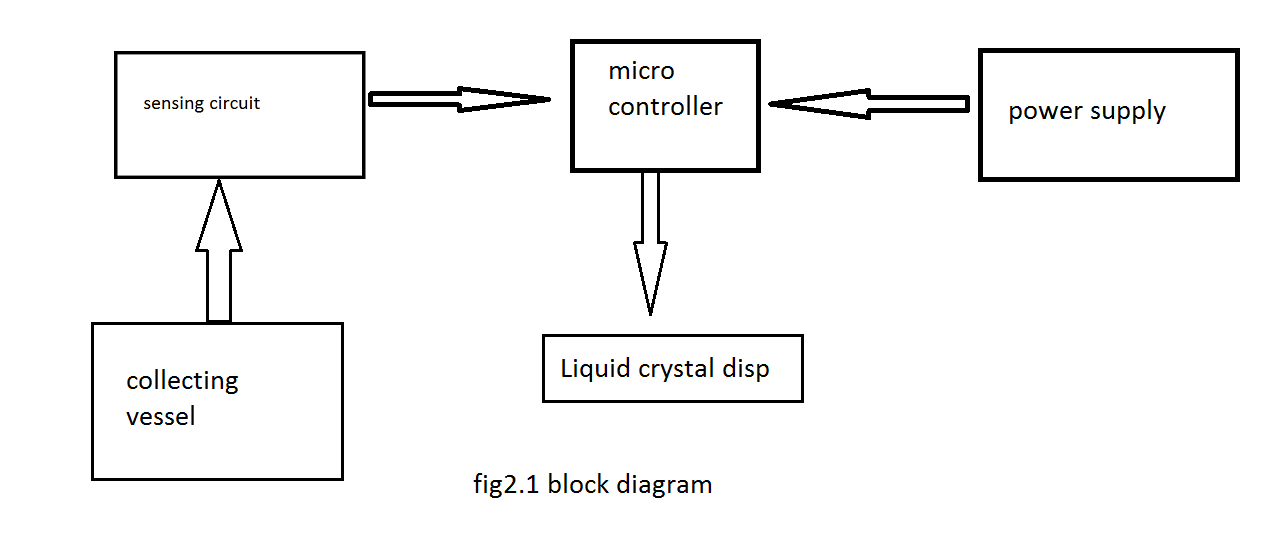
Fig4.2 proteus simulation……………………………………………………………….17

Fig7.1 modification…………………………………………………………………………26

1.Introduction

This report consists of all the details and information regarding to my term project named “automatic rain gauge”. Its starts off with the idea behind this project, then it reaches the part of choosing the hardware, the heart of the project which is the controller, the sensing circuit and finally how the whole project is implemented and all the modules are linked to give the final product. At the end of the report, future aspects of the project also have been discussed.

2.Block Diagram and Circuit



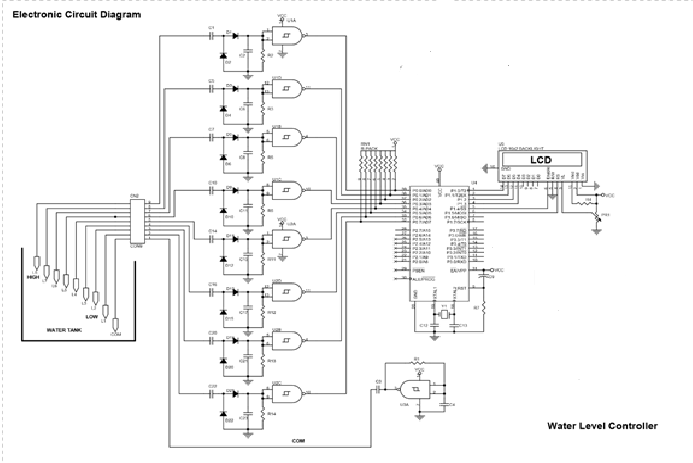


Fig2.2 circuit diagram

3.Hardware description

**3.1 Microcontroller:**

**3.1.1 Description:**

The AT89C51 is a low-power, high–performance CMOS 8-bit microcomputer with 4K bytes of Flash programmable and electrically erasable permanent read only memory (E2PROM). The device is manufactured using Atmel’s high density non-volatile memory technology and is compatible with the industry-standard MCS-51 instruction set and pin out. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional non-volatile 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 application.

**3.1.2 Features:**

* Compatible with MCS-51TM Products.
* 4K Bytes of In-system reprogrammable Flash Memory.
* Endurance: 1,000 Write/Erase cycles.
* Fully Static Operation: 0 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.

**3.1.3 Pin diagram and description:**

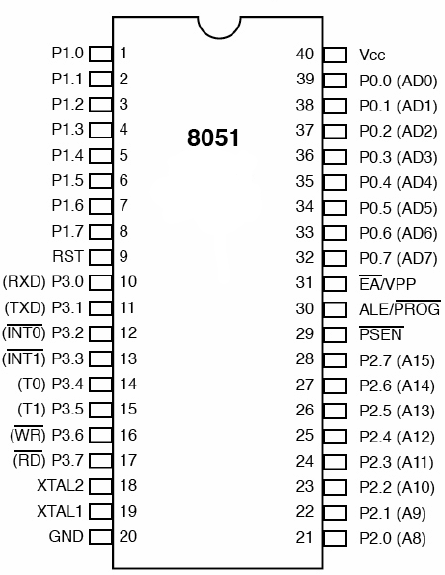


Fig 3.1 pin diagram

* **VCC:** Supply voltage.
* **GND:** Ground.
* **Port 0:** Port 0 is an 8-bit open drain bidirectional I/O port. As an outputport, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs. Port 0 can 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 1 also receives the low-order address bytes during Flash programming and verification.
* **Port 2:** Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (IIL) because of the internal pull-ups .Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses 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 and 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 pull-ups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (IIL) because of the pull-ups. Port 3 receives some control signals for Flash programming and verification. Port 3 also serves the functions of various special features of the AT89S51, as shown in the following table.
* **RST:** Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. This pin drives High for 98 oscillator periods after the Watchdog times out. The DISRTO bit in SFR AUXR (address 8EH) can be used to disable this feature. In the default state of bit DISRTO, the RESET HIGH out feature is enabled.
* **ALE/PROG:** Address Latch Enable (ALE) is an 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 maybe used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory. If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.
* **PSEN:** Program Store Enable (PSEN) is the read strobe to external program memory. When the AT89S51 is executing code from external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory.
* **EA/VPP:** External Access Enable. EA must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, EA will be internally latched on reset. EA should be strapped to VCC for internal program executions. This pin also receives the 12-volt programming enable voltage during Flash programming.
* **XTAL1:** Input to the inverting oscillator amplifier and input to the internal clock operating circuit.
* **XTAL2:** Output from the inverting oscillator amplifier.

**3.2 Liquid Crystal Display (LCD-2X16):**

**3.2.1 Description:**

A liquid crystal display (LCD) is a thin, flat device made up of any number of colour or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other the liquid crystals twist the polarization of light entering one filter to allow it to pass through the other. Many microcontroller devices use ‘smart LCD’ displays to output visual information.

**3.2.2 Signals to the LCD:**

The LCD also requires 3 control lines from the microcontroller:

* **Enable (E):** This line allows access to the display through R/W and RS lines. When this line is low, the LCD is disabled and ignores signals from R/W and RS. When (E) line is high, the LCD checks the state of the two control lines and responds accordingly.
* **Read/Write(R/W):** This line determines the direction of data between the LCD and microcontroller. When it is low, data is written to the LCD. When it is high, data is read from the LCD.
* **Register Select (RS):** With the help of this line, the LCD interprets the type data on data lines. When it is low, an instruction is being written to the LCD. When it is high, a character is being written to the LCD.

**3.2.3 Logic Status on Control Lines:**

* **E:** 0 (Access to LCD disabled)

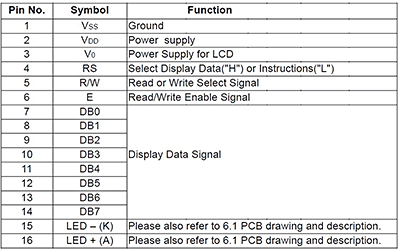
1 (Access when LCD enabled)

* **R/W:** 0 (writing data to LCD)

1 (reading data from LCD)

* **R:**  0 (Instruction)

1 (Character)

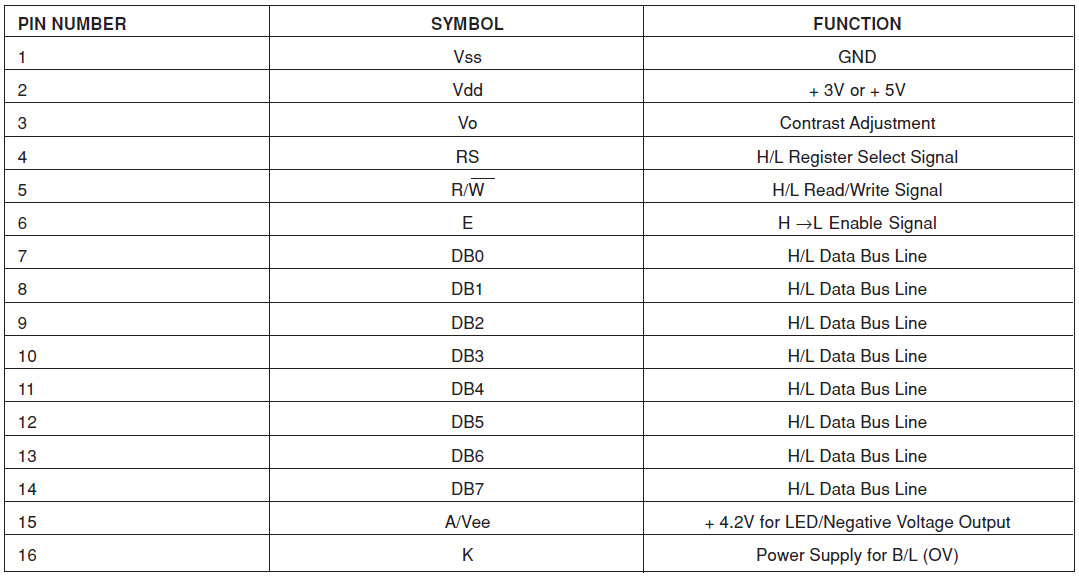


**3.2.4 Writing Data to LCD:**

* Set R/W bit to the low.
* Set RS bit to logic 0 or 1 (Instruction or character).
* Set data to data lines (If it is writing).
* Set E line to HIGH.
* Set E line to LOW (After some delay).

**3.2.5 Pin diagram:**

Fig 3.2 LCD picture

**3.2.6 pin description**

4.Working

The working of the project is divided into three parts based on the three major modules of the system (controller, LCD, sensory circuit).

1. Sensing circuit
2. Interfacing of sensing circuit with the controller
3. Interfacing of LCD with the controller

**4.1 Sensing circuit**

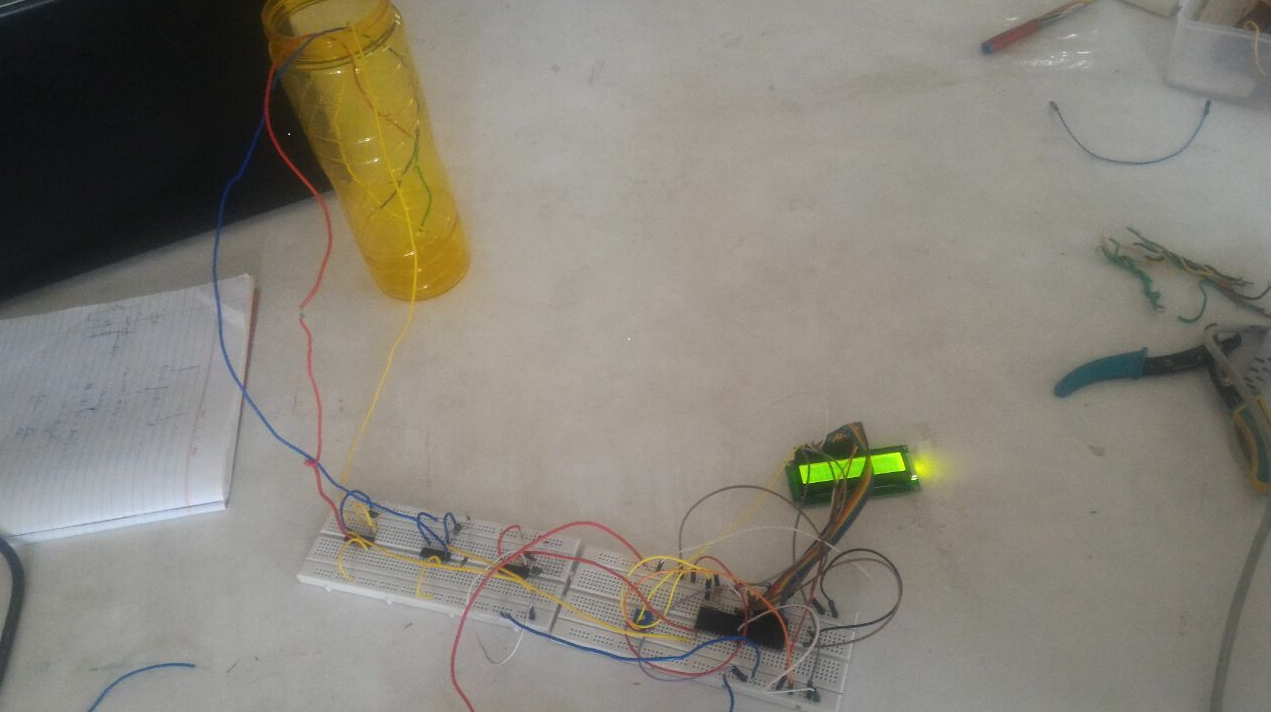
The sensing circuit mainly relying on NAND IC (7400) . The input of the gate are shorted which makes it a NOT gate. The input terminal is placed inside the measuring vessel.

**4.2 interfacing of sensing circuit with controller**

The output of the NAND gate is connected to the output of the port P2 of the micro controller. The pins of the port (P2.0-P2.3) are configured high initially. When the input wires to the gates are grounded they in return ground the output from the controller. For example, if the water level has come above the first level the output of the port will be get changed from 11111111 to 11111110 (P2.7-P2.0).

**4.3 interfacing of LCD with controller**

Port P1 of the controller is configured to the data lines of the LCD and supply Vcc is given 5V and a contrast Vee joins to Vcc and Vss (ground) through a potentiometer of 10kohms. RS, R/W and Enable are connected to the port P3.0, P3.1 and P3.2 respectively. The LED(±) pins are connected to blow the background light only.



**Fig 4.1 picture of working setup**

**4.4 Proteus simulation**

Before implementing the project on bread board, to check the logic of code and circuitry does actually work I did a virtual run of the project on a simulation software “proteus” . To check the logic on which the device works , I grounded my port 2 pins one by one to see the display on the LCD of the measurement marks reached and it worked fine and so I decided to move on to actual implementation.

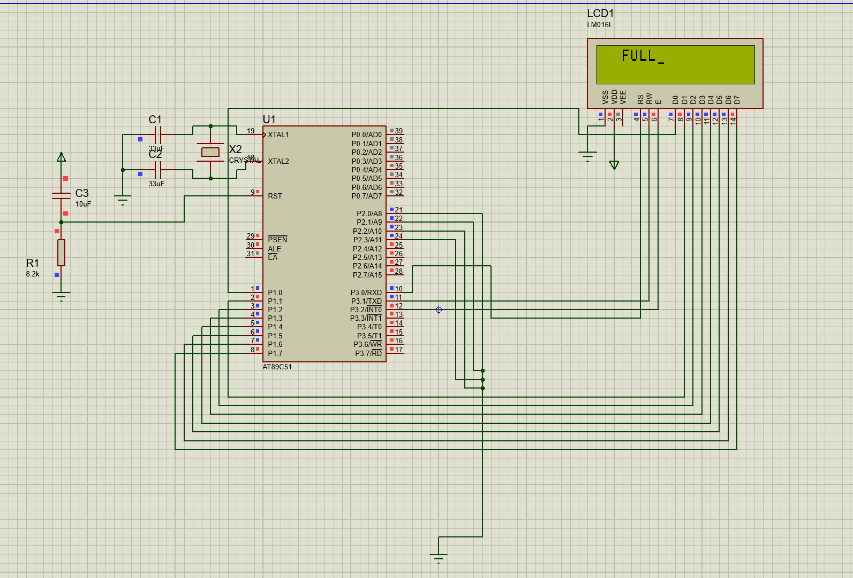


Fig4.2 proteus simulation

**5.Practical Application**

As the name suggests the project can be implemented as rain gauge as well it can be used in following ways:

* It can be used in overhead tanks to measure the level.
* It can be implemented in chemical factories and industries

**6.Program Code**

ORG 0000H

LCALL WAIT

LCALL WAIT

MOV A,#38H

LCALL COMMAND

LCALL WAIT

MOV A,#0EH

LCALL COMMAND

LCALL WAIT

MOV A,#06H

LCALL COMMAND

LCALL WAIT

MOV A,#82H

LCALL COMMAND

LCALL WAIT

MOV A,#0FFH

MOV P2,A // initiates P2 as sensor input

MOV A,#00000000B

MAIN:

MOV A,P2 // moves the current status of P2 to A

CJNE A,#11110000B,LABEL1 // checks whether tank is full

MOV A,#82H

LCALL COMMAND

LCALL WAIT

MOV A,#'F'

LCALL DISPLAY

LCALL WAIT

MOV A,#'U'

LCALL DISPLAY

LCALL WAIT

MOV A,#'L'

LCALL DISPLAY

LCALL WAIT

MOV A,#'L'

LCALL DISPLAY

LCALL WAIT

setb p3.3

ljmp main

LABEL1:MOV A,P2

CJNE A,#11111000B,LABEL2 // checks whether tank is 3/4

MOV A,#82H

LCALL COMMAND

LCALL WAIT

MOV A,#'T'

LCALL DISPLAY

LCALL WAIT

MOV A,#'H'

LCALL DISPLAY

LCALL WAIT

MOV A,#'R'

LCALL DISPLAY

LCALL WAIT

MOV A,#'E'

LCALL DISPLAY

LCALL WAIT

MOV A,#'E'

LCALL DISPLAY

LCALL WAIT

MOV A,#'F'

LCALL DISPLAY

LCALL WAIT

MOV A,#'O'

LCALL DISPLAY

LCALL WAIT

MOV A,#'U'

LCALL DISPLAY

LCALL WAIT

MOV A,#'R'

LCALL DISPLAY

LCALL WAIT

MOV A,#'T'

LCALL DISPLAY

LCALL WAIT

MOV A,#'H'

LCALL DISPLAY

LCALL WAIT

clr p3.3

ljmp main

LABEL2:MOV A,P2

CJNE A,#11111100B,LABEL3 // checks whether tank is 1/2

MOV A,#82H

LCALL COMMAND

LCALL WAIT

MOV A,#'H'

LCALL DISPLAY

LCALL WAIT

MOV A,#'A'

LCALL DISPLAY

LCALL WAIT

MOV A,#'L'

LCALL DISPLAY

LCALL WAIT

MOV A,#'F'

LCALL DISPLAY

LCALL WAIT

clr p3.3

ljmp main

LABEL3:MOV A,P2

CJNE A,#11111110B,LABEL4 // checks whether tank is 1/4

MOV A,#82H

LCALL COMMAND

LCALL WAIT

MOV A,#'Q'

LCALL DISPLAY

LCALL WAIT

MOV A,#'U'

LCALL DISPLAY

LCALL WAIT

MOV A,#'A'

LCALL DISPLAY

LCALL WAIT

MOV A,#'R'

LCALL DISPLAY

LCALL WAIT

MOV A,#'T'

LCALL DISPLAY

LCALL WAIT

MOV A,#'E'

LCALL DISPLAY

LCALL WAIT

MOV A,#'R'

LCALL DISPLAY

LCALL WAIT

clr p3.3

ljmp main

LABEL4:MOV A,P2

CJNE A,#11111111B,MAIN2 // checks whether tank is empty

MOV A,#82H

LCALL COMMAND

LCALL WAIT

MOV A,#'E'

LCALL DISPLAY

LCALL WAIT

MOV A,#'M'

LCALL DISPLAY

LCALL WAIT

MOV A,#'P'

LCALL DISPLAY

LCALL WAIT

MOV A,#'T'

LCALL DISPLAY

LCALL WAIT

MOV A,#'Y'

LCALL DISPLAY

LCALL WAIT

clr p3.3

ljmp main

MAIN2:LJMP MAIN

COMMAND:

MOV P1,A

CLR P3.0

CLR P3.1

SETB P3.2

LCALL WAIT

CLR P3.2

RET

DISPLAY:

MOV P1,A

SETB P3.0

CLR P3.1

SETB P3.2

LCALL WAIT

CLR p3.2

RET

WAIT:

mov r3,#250

here: nop

nop

djnz r3,here

RET

END

**7.Future Scope and Modifications**

For further enhancement and development in project there can be a motor connected to one of the port pins such that it stops running as soon as the level reaches the maximum value at the values of port P2 become 11110000. But mostly the motor cannot run on 5V of supply so a relay can be which disconnects the path of the motor when the max value of measurement has been achieved.

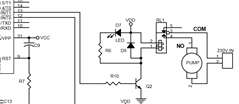


Fig . 7.1 modification

Another idea is to achieve a drainage mechanism which can be used in floods and access rainfall. There can be outlets designed to remove water which are opened by a relay when the water level reaches to a critical value.

8.Conclusion

In this project I successfully developed a rain gauge using micro controller and displayed the output by interfacing a LCD to it. During the making of this project I also developed skills of troubleshooting and analysing margins of errors which occur when dealing with ICs and electronic components.

9.Bibliography

Websites:

* Wikipedia.org
* Instrucables.com
* Engineersgarage.com
* Electrohub.org

Book references:

* Micro controller 8051 by Manish K Patel