

Project Report
SOFTDRINK DISPENSER

Subject Code- ECE-316

Bachelor of Engineering

(Division of Electronics and Communication Engineering)

2015-16

Under the Supervision of

Prof. Dhananjay Vasudeo Gadre

Prepared by:

Mukesh Kumar Mehto (107/EC/13)

Nitin Singh (112/EC/13)



NETAJI SUBHAS INSTITUTE OF TECHNOLOGY

Azad Hind Fauj Marg, Sector-3, Dwarka, New Delhi-110 078

Website: <http://www.nsit.ac.in>

ACKNOWLEDGEMENT

We are thankful to Prof. Dhananjay V. Gadre for providing us with the opportunity to make a project using microprocessor 8085 under subject code EC-316. The project helped us move over the theoretical content in the subject EC-311, and have a hands upon experience of understanding and implementing a microprocessor to use for a real life problem.

We are also thankful to all the supporting staff for providing us the facility of using the lab for fabrication of our project board.

Mukesh Kumar Mehto
Nitin singh

CONTENTS

1. Introduction	...4
I. Objective	...4
II. Role of 8085	...4
2. Block diagram	...5
3. Working	...6
4. Tools and softwares used	...7
5. Schematic	...8
I. Board 1	...8
II. Board 2	...9
6. Board file	...10
7. Components used	...11
8. Assembly code	...12
9. Gantt chart	...17

INTRODUCTION

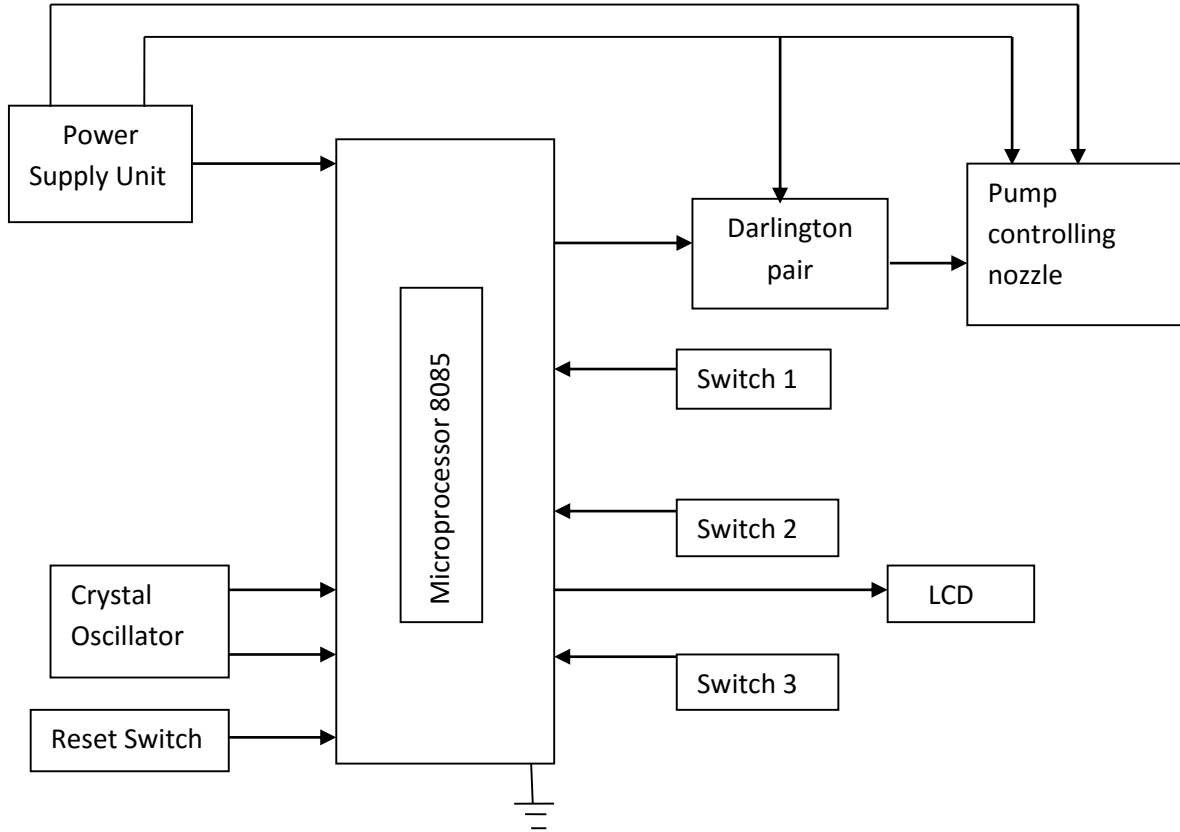
Objective:

The basic idea of the automated Softdrink dispenser is that it allows the user to select one out of three options (small, medium and large) for the amount of liquid as desired.

Role of 8085:

The 8085 microprocessor constantly monitors the switches and looks for the moment when any of the switches is pressed. Then the microprocessor inputs the data and detects which switch is pressed. After it knows which switch is pressed, it just has to put the SOD (Serial Output Data) pin on high state for the corresponding time. The SOD pin is connected to a darlington pair which is then used to control the pump at the output.

Block Diagram



WORKING

First, the microprocessor is to constantly monitor the input at port A of 8255. The input port with chip select set at 00H. The input read is stored in the register B and is then checked for the values 06H, 05H and 03H. 06H input corresponds to switch 1 being pressed, 05H input corresponds to switch 2 being pressed and the 03H input corresponds to switch 3.

As our aim is to generate a one shot pulse of different time period on SOD pin so we need to decide which switch is pressed. At this stage we know which switch was pressed and we use it to jump to different locations in our program, i.e. to LOW for 06H, MED for 05H and HIGH for 03H. In the LOW section we need to display the current volume count on the two port (port B and port C) and generate 'Logic High' on SOD pin for 10 seconds. Similarly in MED section volume count is displayed and the SOD is '1' for 20 seconds and for HIGH section volume count is displayed and the SOD remains '1' for 30 seconds.

TOOLS AND SOFTWARES USED

Hardware tools:

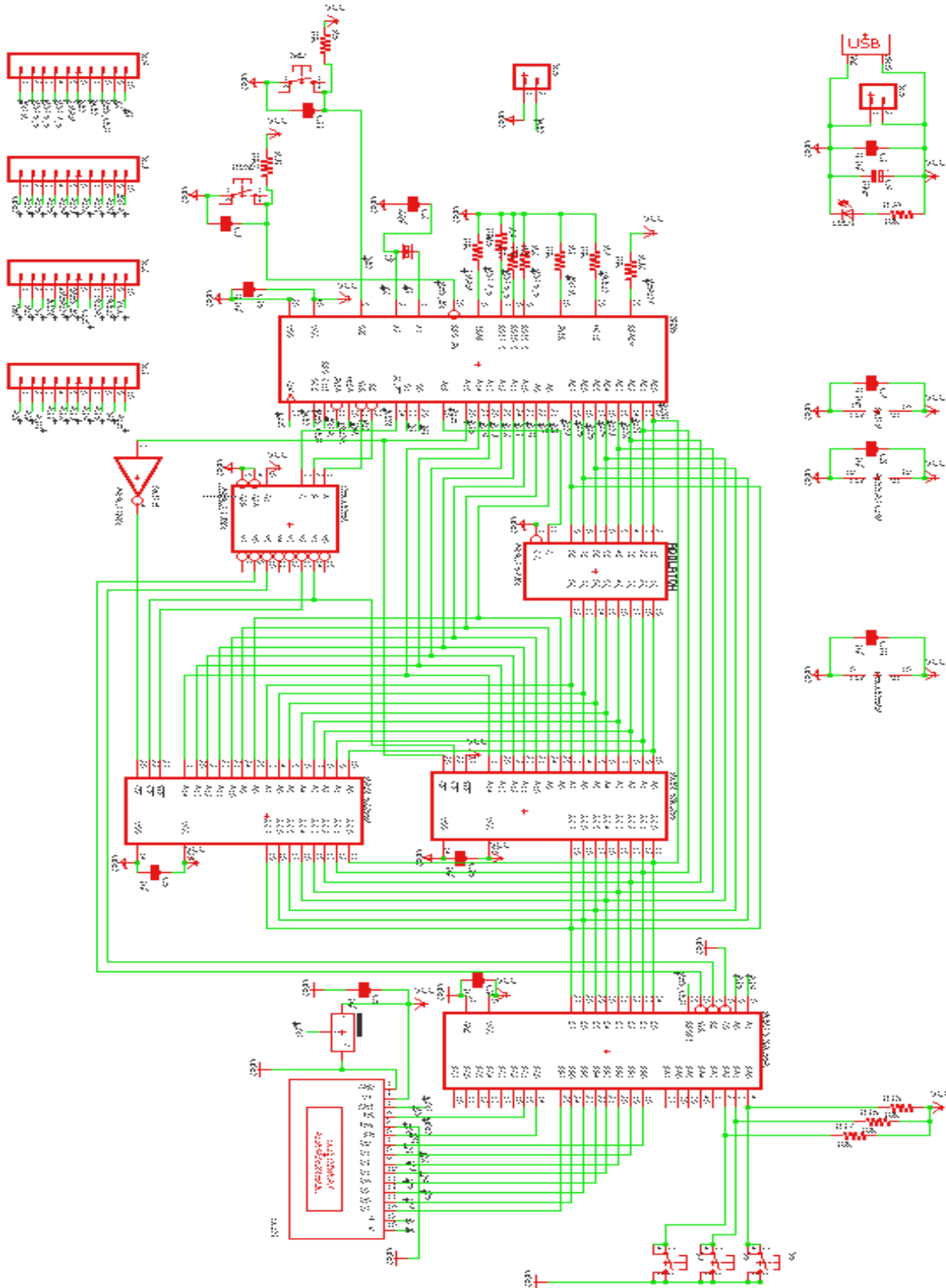
1. Soldering iron
2. Solder
3. EEPROM programmer
4. DC power supply

Software tools:

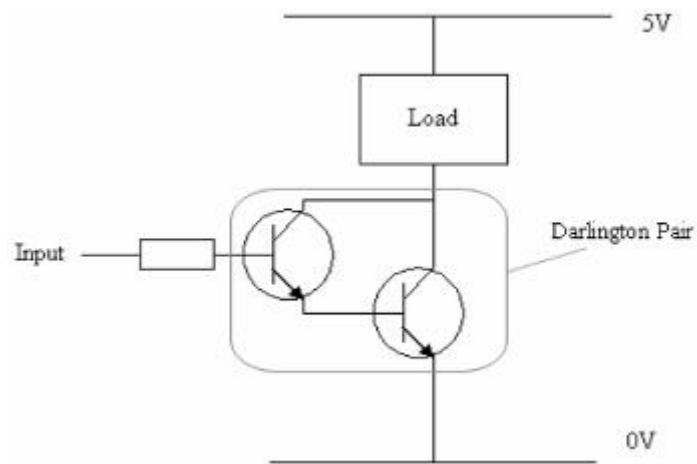
1. EAGLE CAD for schematic and board layout
2. OSHON Soft assembler
3. EEPROM software for uploading the code on ROM

SCHEMATIC

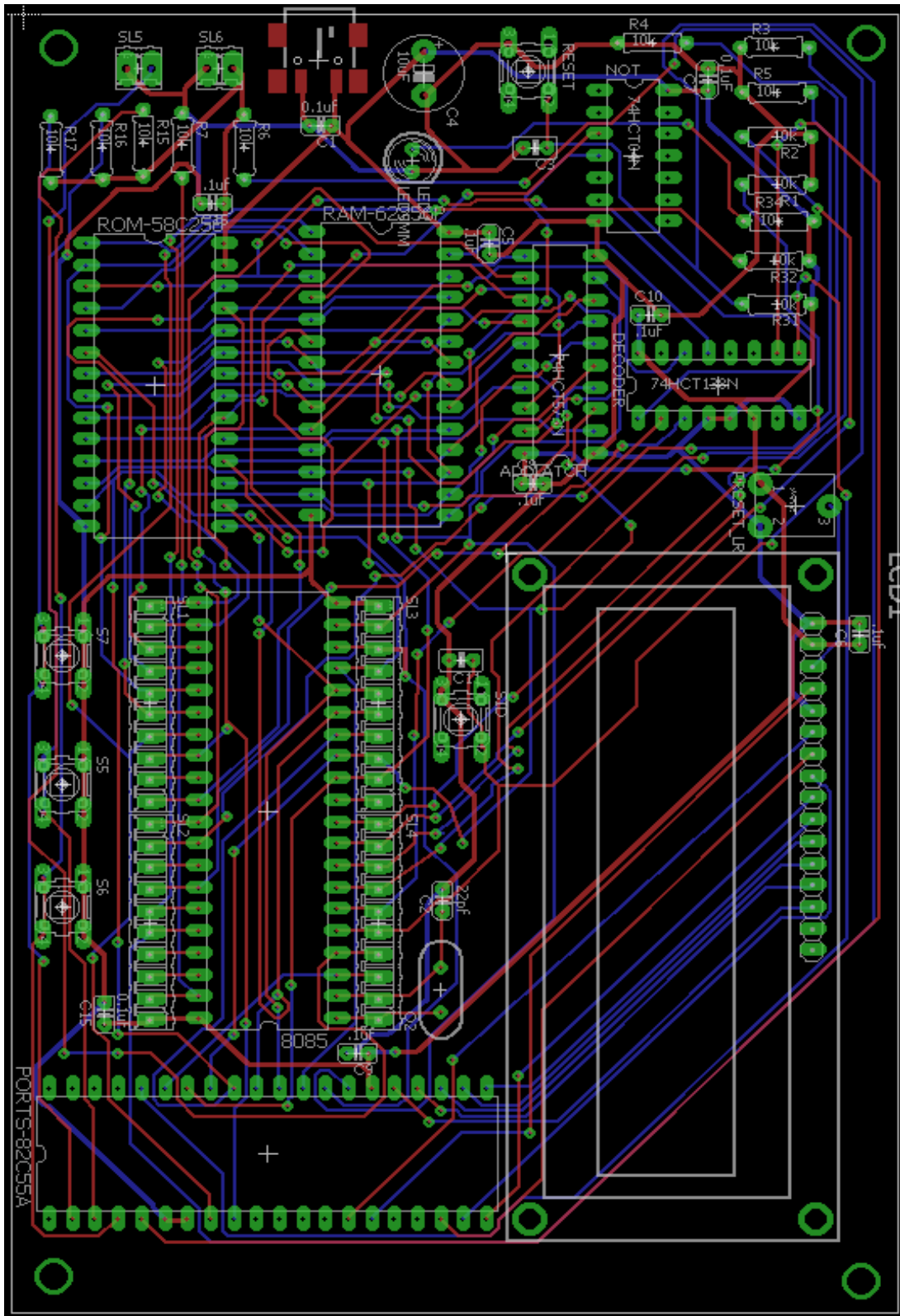
Board1:



Circuit for 2nd board:



Board file:



COMPONENTS USED

NAME	COMPONENTS	QUANTITY
MICROPROCESSOR	8085	1
RAM	62256P	1
EEPROM	27256	1
INPUT/OUTPUT PORT	8255	1
LATCH	74HCT573N	1
DECODER	74HCT138N	1
PUSH BUTTONS		5
LCD DISPLAY	16*2 LCD	1
NOT GATE	74HCT04N	1
USB POWER		1
CRYSTAL	4 MHz	1
CONNECTORS	M02	2
RESISTORS	10K, 220 ohm	
CAPACITORS	0.01uF, 22pF, 10uF	
PRESET		1

CALL DATA

MVI A,52H

CALL DATA

MVI A,45H

CALL DATA

MVI A,53H

CALL DATA

MVI A,53H

CALL DATA

MVI A,20H

CALL DATA

MVI A,54H

CALL DATA

MVI A,48H

CALL DATA

MVI A,45H

CALL DATA

MVI A,20H

CALL DATA

MVI A,42H

CALL DATA

MVI A,55H

CALL DATA

MVI A,54H
CALL DATA
MVI A,54H
CALL DATA
MVI A,4FH
CALL DATA
MVI A,4EH
CALL DATA

MVI A,098H /initialize control register

OUT 0C0H

SWITCH: MVI A,040H /set sod pin to low

SIM

IN 00H /input

MOV B,A /store input in register B

ANI 07H /Masking of port A

XRI 06H /Check for switch 1

JZ LOW

MOV A,B

XRI 05H /check for switch 2

JZ MEDIUM

MOV A,B

XRI 03H /check for switch 3

JZ HIGH

JMP SWITCH

LOW: /delay loop for switch 1

MVI A, 0C0H /set sod pin high

SIM

LOOP:

MVI D, 010H

MVI E, 040H

AGAIN1:

MVI B,00H

AGAIN:

DCR B

JNZ AGAIN

DCX D

MOV A,E

ORA D

JNZ AGAIN1

JMP SWITCH

MEDIUM: /delay loop for switch 2

MVI A, C0H /set sod pin high

SIM

MVI D, 020H

MVI E, 07FH

AGAIN3:

MVI B, 00H

AGAIN2:

DCR B

JNZ AGAIN2

DCX D

MOV A, E

ORA D

JNZ AGAIN3

HIGH: /delay loop for switch 3

MVI A, C0H /set sod pin high

SIM

MVI D, 030H

MVI E, 0C0H

AGAIN5:

MVI B, 00H

AGAIN4:

DCR B

JNZ AGAIN4

DCX D


```
MOV A, E
ORA D
JNZ AGAIN5
```

```
CMD: OUT 80H
```

```
    MVI A,01H    ;RS=0,E=1
    OUT 40H
    CALL DELAY
    MVI A,00H    ;RS=0,E=0
    OUT 40H
    CALL DELAY
    RET
```

```
DATA: OUT 80H
```

```
    MVI A,03H    ;RS=1,E=1
    OUT 40H
    CALL DELAY
    MVI A,02H    ;RS=1,E=0
    OUT 40H
    CALL DELAY
    RET
```

```
DELAY:
```

MVI C,8CH

LOOP:

DCR C

JNZ LOOP

RET

.END

SOME PICTURES OF PROJECT :

