A REPORT ON INTELLIGENT HUMIDISTAT

Submitted as a design project for the course CS F241

Microprocessor Programming and Interfacing

By

BHOOMI SAWANT	2017A7PS0001P
BACHHAV PRATIK MANOHAR	2017A7PS0003P
SUBHAM KUMAR DASH	2017A7PS0004P
PIKLU PAUL	2017A7PS0006P

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE PILANI, RAJASTHAN

25 April, 2019

ACKNOWLEDGEMENT

The successful completion of this report required a lot of guidance and support that we received throughout this project. We would like to extend our deepest gratitude to the Instructor-in-Charge (IC) of this course, Prof. Nitin Chaturvedi for giving us this opportunity to work on such an interesting assignment. The lectures given by Prof. J P Mishra were also helpful in clearing our basics.

We would also like to thank our tutorial professors Dr. Vinay Chamola, Devesh Samaiya for their excellent guidance. Last but not the least we would like to extend our gratitude to all the lab instructors who have helped us to get command over assembly language programs.

We would also like to thank our seniors for guiding us all along till the completion of this report.

CONTENTS

PROBLEM STATEMENT

A humidistat is supposed to be reset according to the outside temperature- as the outside temperature falls, the humidity level inside the house should be set lower. The purpose of this project is to develop a humidistat which senses the outside temperature and adjusts the humidity accordingly. Two sensors are required: outside temperature and inside humidity. Output is provided via a simple relay with the humidifier (presumably on the furnace) being on or off. Also readings from the humidity and temperature sensors must be displayed on an LCD display.

ASSUMPTIONS

- ALP is stored in the ROM in executable format.
- Outside temperature varies from -35 to +65 degree centigrade.
- Temperature and relative humidity (RH) are already stored in the DS and one degree change in temperature corresponds to one unit change in RH.
- When the humidifier is switched on, it decreases the humidity.

HARDWARE DEVICES

COMPONENT	NO. OF CHIPS	CHIP	USE
8086	1	Microprocessor	Central Processing Unit
6116	2	RAM-2K	Random access memory which contains DS, SS
2732	2	ROM-4K	Read only memory which contains entire code CS
74LS373	3	Octal Latch	To latch address bus
74LS245	4	8- bit buffer	To buffer data bus (bi-directional)
8255	2	PPI	Used for I/O
8253	1	Programmable Interval Timer	To generate the clock signal for ADC
ADC 0808	1	Analog to Digital Converter	Converts analog temperature From temperature sensor to digital form
LM020L	1	LCD Display	Use for displaying the temperature and humidity
74LS244	1	Unidirectional buffer	Buffering the control Lines
LM35	1	Temperature sensor	To detect the temperature of the atmosphere
DHT22	1	Humidity sensor	To detect the humidity of the atmosphere
74LS138	2	3:8 Decoder	Used for select signals
Logic gates	-	OR,NOR,NOT	Used for building decoding logic for memory interfacing and I/O interfacing

MEMORY ORGANIZATION

The system uses 8kb of ROM and 4kb of RAM. Both consist of 2 chips of 4kb and 2kb size respectively. They are organized into odd and even banks to facilitate both byte and word size data transfers.

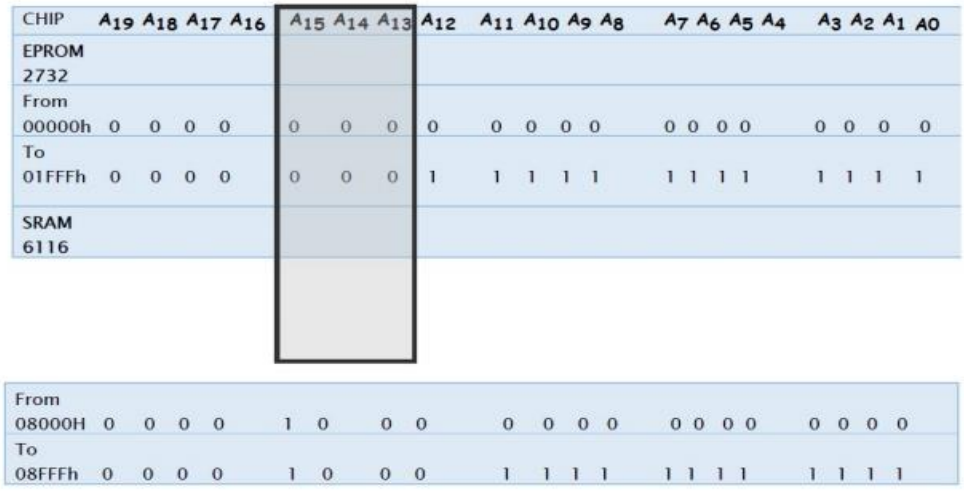
Random Access Memory (6116):

- Starting Address: 08000H
- Ending Address: 08FFFH

Read Only Memory (2732):

- Starting Address: 00000H
- Ending Address: 01FFFH

The code resides in the ROM and begins at address 00000H. The address loaded as soon as the system is switched on is FFFF0H.



I/O MAPPING

The input and output devices such as temperature & humidity sensor and LCD are connected using 8255.

8255 (A)

PORT TYPE	PORT ADDRESS	TYPE
A	00H	Output
B	02H	Output
C (LOWER)	04H	Output
C (UPPER)	04H	Output
Control Register	06H	

8255 (B)

PORT TYPE	PORT ADDRESS	TYPE
A	10H	Input
B	12H	Input
C (LOWER)	14H	Output
C (UPPER)	14H	Input
Control Register	16H	

Both 8255 are used in i/o mode.

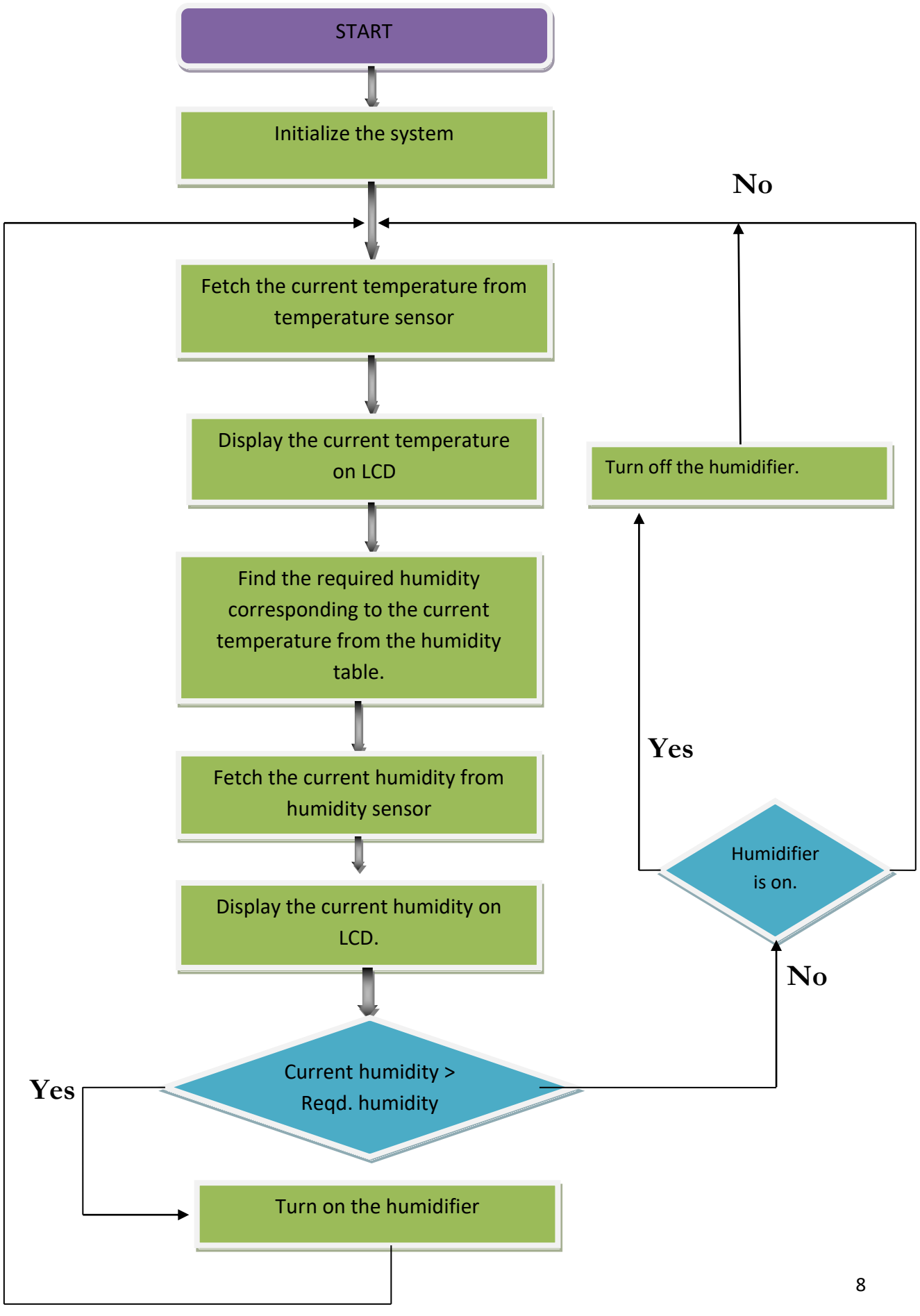
Control Word for 8255 (1): **10000000b**

- Port A is used for generating control signal of LCD
- Port B is used for giving input to the LCD
- PC7 is used to turn on the humidifier.

Control Word for 8255 (2): **10011010b**

- Port A is used to take the digital output from ADC
- PC0 – PC3 are used to give control input to ADC
- PC5 is used to receive the signal of EOC.

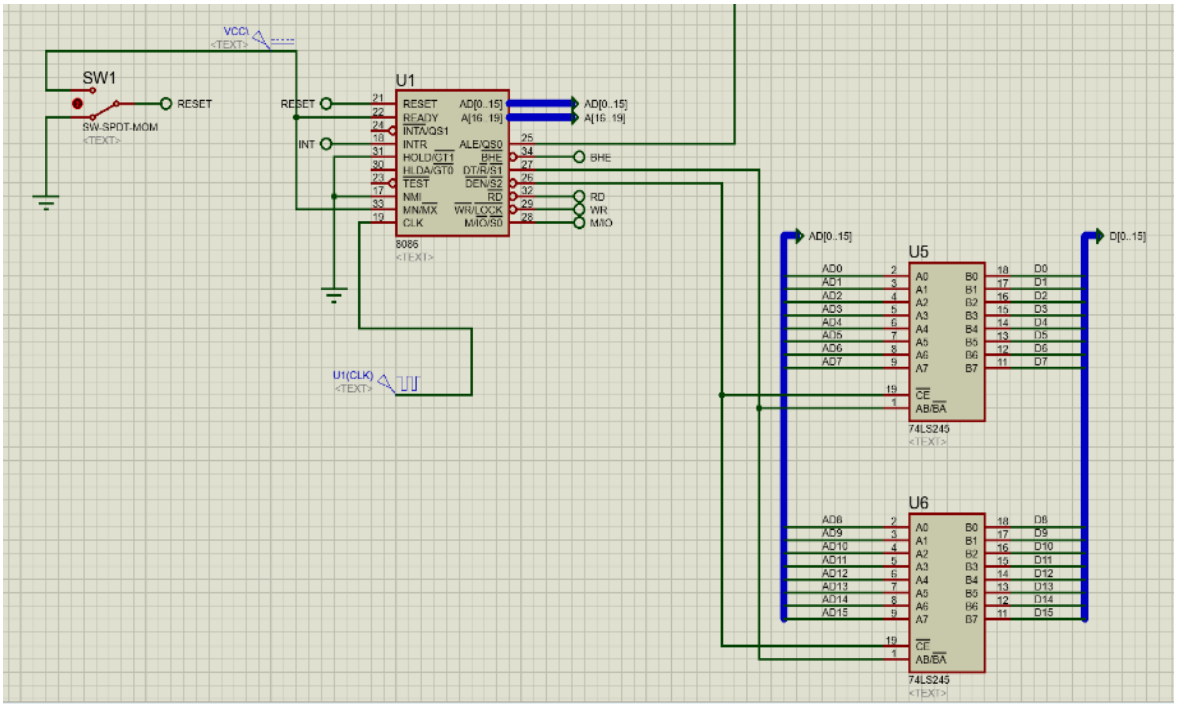
FLOWCHART

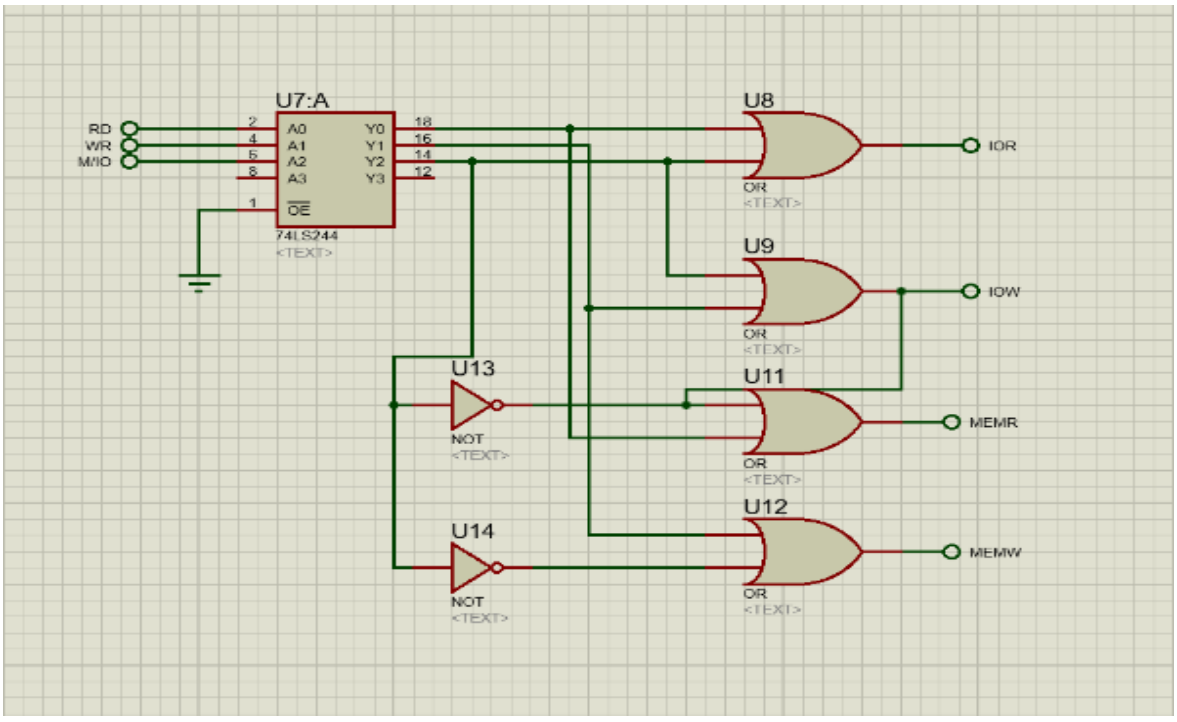

START
Initialize the system
No
Fetch the current temperature from temperature sensor
Display the current temperature on LCD
Turn off the humidifier.
Find the required humidity corresponding to the current temperature from the humidity table.
Yes
Fetch the current humidity from humidity sensor
Humidifier is on.
Display the current humidity on LCD.
No
Current humidity > Reqd. humidity
Yes
Turn on the humidifier

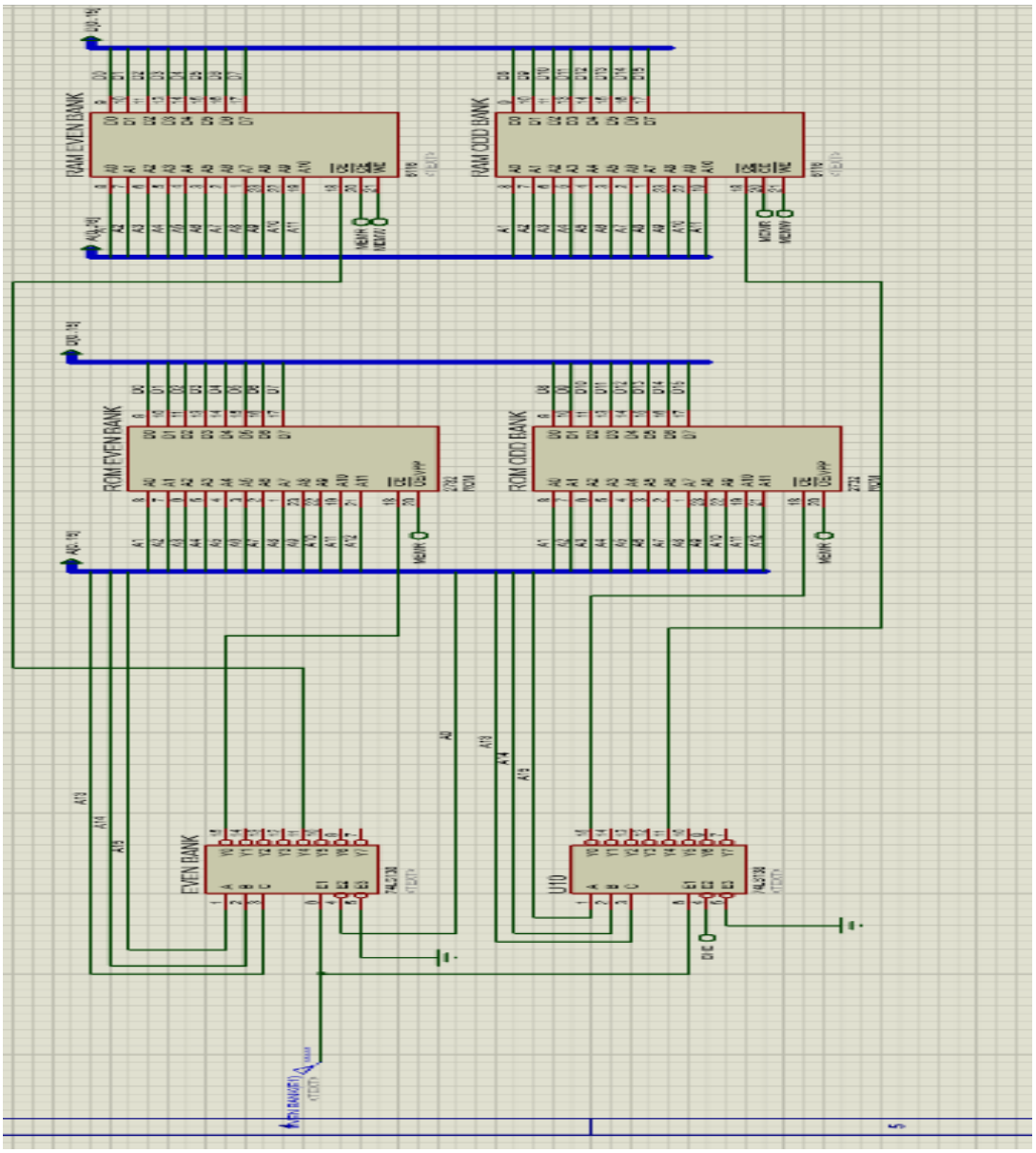
LIMITATIONS

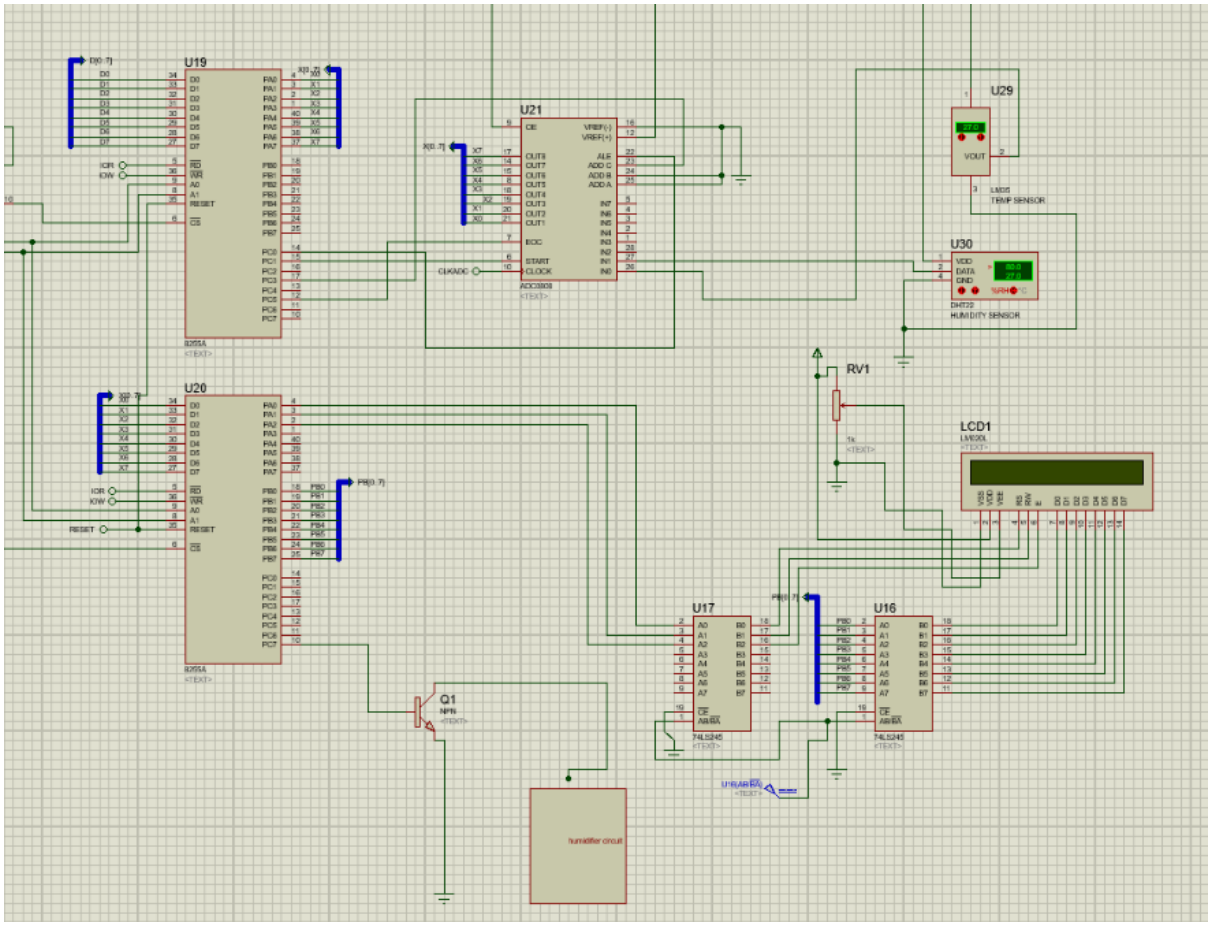
We have assumed that the relationship between temperature and humidity is linear, which may not be the case in real scenario. Hence, sometimes it may fail at high temperature.

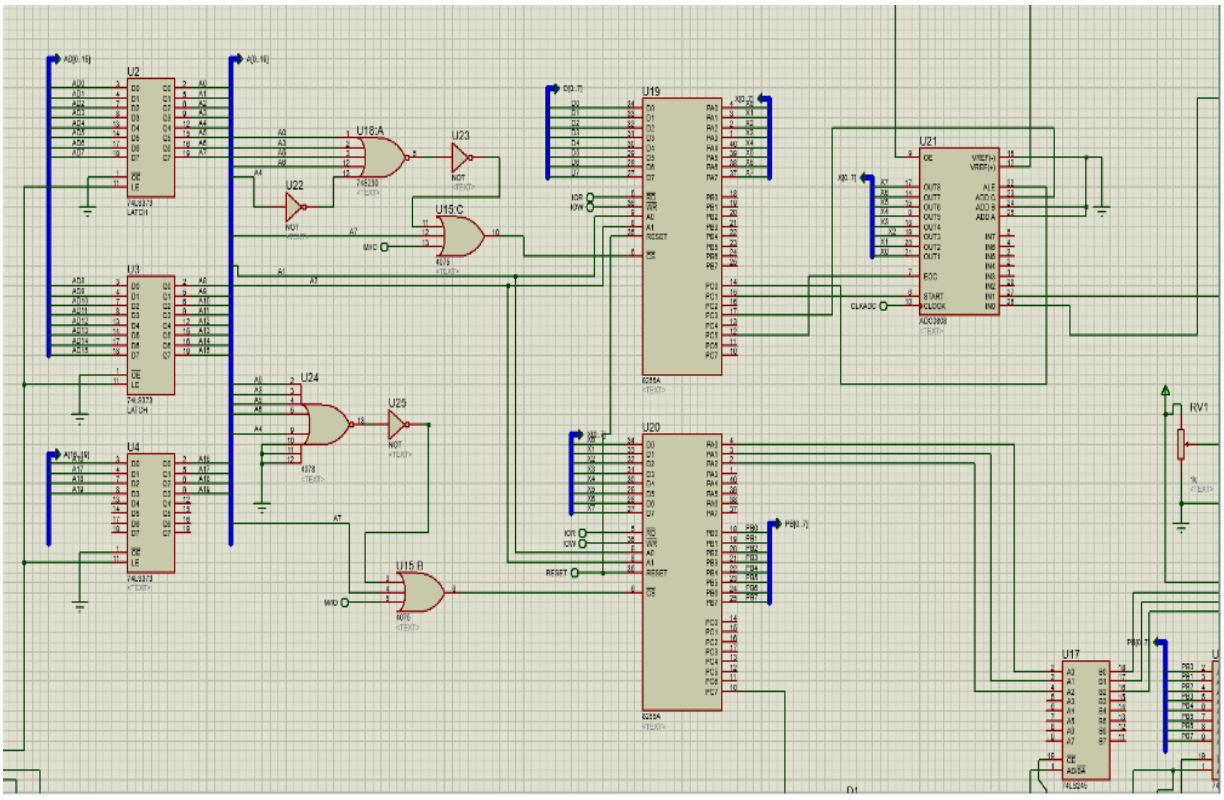
CIRCUIT DIAGRAM






RAM EVEN BANK
RAM ODD BANK
ROM EVEN BANK
ROM ODD BANK
EVEN BANK
U10
A0
A19
A14
A15
A13


U19
U21
U29
LM35
TEMP SENSOR
U30
DHT22
HUMIDITY SENSOR
RV1
LCD1
U20
U17
U16
Q1
humidifier circuit


U2
U3
U4
U18:A
U23
U22
U15:C
U24
U25
U15:B
U19
U21
U20
RV1
U17

.ASM CODE

```
.MODEL TINY
.DATA


;8253 USED TO GENERATE CLOCK FOR ADC
CNT0 EQU 20H
CREG EQU 26H


;8255(1)  INITIALISE
PORT1A EQU 00H        ;CONTROLLING THE LCD
PORT1B EQU 02H        ;INPUT TO LCD
PORT1C EQU 04H        ;UPPER - ROW
                      ;LOWER - COLUMN
CREG1 EQU 06H


;8255(2) USED FOR ADC

PORT2A EQU 10H        ;INPUT TO DI DEVICE
PORT2B EQU 12H        ;ADC
```

```
PORT2C EQU 14H          ;PC1 - SOC OF ADC

;PC3 - ADDC OF ADC (USED FOR SELECTING THE ;FIRST & SECOND INPUT CHANNEL OF ADC)

;PC5 - EOC OF ADC

CREG2 EQU 16H


TEMP_RANGE DB -35,-34,-33,-32,-31,-30,-29,-28,-27,-26,-25,-24,-23,-22,-21,-20,-19,-18,-17,-16,-15,-14,-13,-12,-11

                DB -10,-9,-8,-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25

                DB 26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59

                DB 60,61,62,63,64,65


HUMIDITY_RANGE DB 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36

                DB

37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70

                DB 71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100


CURR_TEMP db ?

CURR_HUMIDITY db ?

TEMP db ?
```

```
TABLE_K DB
0EEH,0EDH,0EBH,0E7H,0DEH,0DDH,0DBH,0D7H,0BEH,0BDH,0BBH,0B7
H,7EH,7DH,7BH,77H

DAT2 DB 3 DUP(" ");

T DB 30H,31H


.CODE

.STARTUP


;INITIALIZE DS,SS,ES TO START OF RAM

MOV AX,08000H

MOV DS,AX

MOV SS,AX

MOV ES,AX

MOV SP,08FFEH


;INITIALIZING 8253

MOV AL,00010110B

OUT CREG,AL

MOV AL,5

OUT CNT0,AL


;INITIALIZING 8255(1)

MOV AL,10000000B
```

```
OUT CREG1,AL
CALL DELAY_2MS


;INITIALIZING 8255(2)
MOV AL,10011010B
OUT CREG2,AL
CALL DELAY_2MS

X1:
CALL FETCH_TEMP
MOV AL,CURR_TEMP
CALL FUNC
LEA SI,TEMP_RANGE
DEC SI
MOV CX,100

CALL DELAY
CALL DELAY
CALL DELAY

AGAIN:
INC SI
CMP [SI],AL
LOOPNE AGAIN
```

```
SUB SI,OFFSET TEMP_RANGE
LEA DI,HUMIDITY_RANGE
ADD DI,SI
MOV BL,[DI]

CALL FETCH_HUMIDITY
MOV AL,CURR_HUMIDITY
CALL FUNC
CMP BL,CURR_HUMIDITY
JA X2
MOV AL,00001111B
OUT CREG1,AL      ;SWITCHES ON THE HUMIDIFIER(LED)

LOOP1:
CALL DELAY_2MS
CALL FETCH_HUMIDITY
CMP BL,CURR_HUMIDITY
JL LOOP1
MOV AL,00001110B
OUT CREG1,AL      ;SWITCHES ON THE HUMIDIFIER(LED)

X2:
CALL DELAY_2MS
JMP X1
```

```
.EXIT



;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;PROCEDURE TO FETCH CURRENT TEMPERATURE
FETCH_TEMP PROC NEAR

PUSH SI
;MOV AL,20H
;OUT PORT2A,AL

MOV AL,06H      ;GIVE ADC
OUT CREG2,AL

MOV AL,00H      ;GIVE ALE
OUT CREG2,AL

MOV AL,02H      ;GIVE SOC
OUT CREG2,AL

MOV AL,01H   ;SET ALE
OUT CREG2,AL
```

```
MOV AL,03H   ;SET SOC
OUT CREG2,AL

MOV AL,02H       ;GIVE SOC
OUT CREG2,AL

MOV AL,00H       ;GIVE ALE
OUT CREG2,AL

LOOP2:
IN AL,PORT2C
CALL DELAY_2MS
AND AL,20H    ;CHECK FOR EOC
CMP AL,20H
JNZ LOOP2
CALL DELAY_2MS

MOV AL,10011010B ;INITIALIZING 8255(2)
OUT CREG2,AL
IN AL,PORT2A ;AL HAS THE CURRENT TEMPERATURE

LEA SI,CURR_TEMP
MOV [SI],AL
POP SI
RET
```

```
FETCH_TEMP ENDP

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

; PROCEDURE TO GET THE CURRENT HUMIDITY

FETCH_HUMIDITY PROC NEAR

PUSH SI

MOV AL,07H        ;GIVE ADC
OUT CREG2,AL

MOV AL,00H        ;GIVE ALE
OUT CREG2,AL

MOV AL,02H        ;GIVE SOC
OUT CREG2,AL

MOV AL,01H   ;SET ALE
OUT CREG2,AL

MOV AL,03H   ;SET SOC
OUT CREG2,AL
```

```
MOV AL,02H       ;GIVE SOC
OUT CREG2,AL

MOV AL,00H       ;GIVE ALE
OUT CREG2,AL

LOOP2:
IN AL,PORT2C
CALL DELAY_2MS
AND AL,20H    ;CHECK FOR EOC
CMP AL,20H
JNZ LOOP2
CALL DELAY_2MS

MOV AL,10011010B ;INITIALIZING 8255(2)
OUT CREG2,AL
IN AL,PORT2A ;AL HAS THE CURRENT HUMIDITY

LEA SI,CURR_HUMIDITY
MOV [SI],AL
POP SI
RET

FETCH_HUMIDITY ENDP
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

FUNC PROC NEAR
PUSH AX
MOV AL,38H
CALL COMNDWRT
CALL DELAY
CALL DELAY
CALL DELAY
MOV AL,0EH
CALL COMNDWRT

MOV AL, 01  ;CLEAR LCD
CALL COMNDWRT
CALL DELAY
CALL DELAY

POP AX
PUSH AX
LEA DI,DAT2
MOV BX,100D
MOV DX,0
DIV BX
ADD AL,30H
CALL DATWRIT ;ISSUE IT TO LCD
```

```
CALL DELAY
CALL DELAY
MOV AX,DX
MOV BX,10D
MOV DX,0
DIV BX
ADD AL,30H
CALL DATWRIT
CALL DELAY
CALL DELAY
MOV AX,DX
MOV DX,0
ADD AL,30H
CALL DATWRIT
CALL DELAY
CALL DELAY
POP AX

RET
FUNC ENDP

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

COMNDWRT PROC ;THIS PROCEDURE WRITES COMMANDS TO LCD
OUT PORT1B, AL  ;SEND THE CODE TO PORT B
```

```
MOV AL, 00000100B ;RS=0,R/W=0,E=1 FOR H-TO-L PULSE
OUT PORT1A, AL
NOP
NOP
MOV AL, 00000000B ;RS=0,R/W=0,E=0 FOR H-TO-L PULSE
OUT PORT1A, AL
RET
COMNDWRT ENDP

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

DATWRIT PROC NEAR
        PUSH DX  ;SAVE DX
        MOV DX,PORT1B  ;DX=PORT B ADDRESS
        OUT DX, AL ;ISSUE THE CHAR TO LCD
        MOV AL, 00000101B ;RS=1, R/W=0, E=1 FOR H-TO-L PULSE
        MOV DX, PORT1A ;PORT A ADDRESS
        OUT DX, AL  ;MAKE ENABLE HIGH
        MOV AL, 00000001B ;RS=1,R/W=0 AND E=0 FOR H-TO-L PULSE
        OUT DX, AL
        POP DX
        RET
DATWRIT ENDP ;WRITING ON THE LCD ENDS

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
DELAY_2MS PROC NEAR

MOV CX,100

HER: NOP

 LOOP HER

RET

DELAY_2MS ENDP

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;DELAY IN THE CIRCUIT HERE THE DELAY OF 20 MILLISECOND IS
PRODUCED

DELAY PROC

        MOV CX, 1325 ;1325*15.085 USEC = 20 MSEC

        W1:

                NOP

                NOP

                NOP

                NOP

                NOP

        LOOP W1

        RET

DELAY ENDP


DELAY_2S PROC

        MOV CX, 33125D
```

```
W2:
        NOP
        NOP
        NOP
        NOP
        NOP
LOOP W2
        MOV CX, 33125D
W3:
        NOP
        NOP
        NOP
        NOP
        NOP
LOOP W3
        MOV CX, 33125D
W4:
        NOP
        NOP
        NOP
        NOP
        NOP
LOOP W4
        MOV CX, 33125D
W5:
```

```
        NOP
        NOP
        NOP
        NOP
        NOP
    LOOP W5
    RET
DELAY_2S ENDP

END
```