A REPORT ON

SMART AC SYSTEMS

HARDWARE DESIGN AND SPECIFICATIONS

CS F241 Microprocessor Programming and Interfacing

Submitted by Group No. 33

Jitvan Shukla 2016A7PS0083P

Vaibhav Maheshwari 2016A7PS0081P

Swarup N 2016A7PS0080P

Arpan Parikh 2016A7PS0082P



BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE PILANI, RAJASTHAN-333031

25th April 2018

## CONTENTS

|  |  |  |
| --- | --- | --- |
| **SERIAL NO.** | **TITLE** | **PAGE NO.** |
| 1 | Problem Statement | 3 |
| 2 | Assumptions | 3 |
| 3 | System Description | 4 |
| 4 | Hardware Devices | 5 |
| 5 | Memory and I/O mapping | 6 |
| 6 | Algorithm | 8 |
| 7 | Flow Chart | 9 |
| 8 | ASM Code | 10 |
| 9 | Circuit Diagram | 26 |
| 10 | References | 26 |

**PROBLEM STATEMENT**

**Problem No. 17: SMART AC SYSTEM**

##### **Description:** This system opens/closes four AC vents based upon the current temperature in the Room. The temperature is maintained at a range of 16–35 0C. The AC vents can be gradually opened / closed. This is done in accordance with the temperature in the room. The room is a fairly large sized room so 4 temperature sensors are placed at different points of the room.

Each sensor and AC vent is associated with part of the room. You can assume that the room is broken up into 4 sub-areas each with its own sensor and ac vent.

**User Interface:** LCD displaying Temperature in 0C.

##### Single push button to vary temperature between 160C – 350 C.

The duration for which the system is ON can be set by the user in minutes ranging from 30 min. to 6 hours with a granularity of 30 min. Once the defined time has elapsed, the vents are closed.

**ASSUMPTIONS**

1. ALP is already stored in ROM in executable form.
2. The room is very big, so that the temperature of one part of room has no influence over the temperature of the other parts.
3. After system startup, the temperature of each part of the room varies between 160C – 350 C only.
4. When all AC vents are completely open, room temperature will be 160C and when all are completely closed, the room temperature will be 350C.
5. When the AC is switched off, all the vents are completely closed.
6. The 4 AC vents are modelled by 4 LEDs. Each LED is switched ON when that particular vent is supposed to be open.
7. When the microprocessor is switched on, its address is initialized to the starting address of the ROM.

## SYSTEM DESCRIPTION

1. Intel 8086 microprocessor
2. **INPUT DEVICE:**
   1. 4 temperature sensors
   2. 1 push button
   3. 1 switch
3. OUTPUT DEVICES:
   1. LCD to display temperature
   2. 4 motors to open/close AC vents (simulated via LEDs)
4. Two 8255 (Programmable Peripheral Interface) chips interfaced to 8086
   1. **8255-1:** Port-A is interfaced to the 8 data lines of LCD driverHD244780. Port-B is used for interfacing 8086 with the control lines of the LCD. PC0 is used to vary the mode of input (temperature/timer); PC1 is used for setting the values of time and temperature.
   2. **8255-2:** Port-A takes input from ADC0808 which is interfaced with the 4 temperature sensors LM35.Port-B is used as an output port connected to the motors (LEDs). Port-C is used to select a particular sensor from which input is taken.

5. 8284 clock is used to generate clock signal for 8086. (8284 clock generator not used in proteus design.)

6. 8253 is used to generate stepped down time signals for the given problem statement making use of the clock signal from 8284.

**HARDWARE DEVICES**

|  |  |  |
| --- | --- | --- |
| **CHIP NUMBER**  **(No. of chips)** | **CHIP** | **USE** |
| 8086 (1) | Microprocessor | Central Processing Unit (C.P.U) |
| 6116(2) | RAM-2K (Total-4K) | Random Access Memory containing  DS and SS segments. |
| 2732(2) | ROM-4K (Total-8K) | Read only Memory which contains the code. |
| 74LS373(3) | 8-BIT LATCH | To Latch the Address Bus. |
| 74LS245(4) | 8-BIT BUFFER (BIDIRECTIONAL) | To Buffer Data Bus |
| 8255(2) | PROGRAMMABLE PERIPHERAL INTERFACE | Connected to Various Input/ Output Devices. |
| ADC0808(1) | ANALOG TO DIGITAL CONVERTER | CONVERTS ANALOG VOLTAGE SIGNAL Vcc TO DIGITAL FORM |
| 8253(1) | CLOCK TIMER | TO KEEP THE TRACK OF TIME FOR WHICH THE PROCESSOR WILL WORK |
| LM020L (1) | LCD DISPLAY | FOR DISPLAYING TEMPERATURE AND TIME VALUES |
| LM 35(4) | TEMPERATURE SENSOR | TO PRODUCE ANALOG SIGNAL FOR THE TEMPERATURE IN ROOM |
| 74154 (1) | 4:16 DECODER | TO PRODUCE THE CHIP SELECT SIGNALS FOR I/O DEVICES |
| 74138 (1) | 3:8 DECODER | TO PRODUCE CHIP SELECT SIGNALS FOR ROM AND RAM |
|  | LED-RED | TO SIMULATE THE OPENING AND CLOSING OF VENTS IN PROTEUS |

**MEMORY INTERFACING**

This system uses 4KB of RAM (as 2x2KB chips for even and odd banks respectively) and 8KB of ROM (as 2x4KB chips for even and odd banks respectively). The memory is divided into even and odd banks because 8086 has a 16-bit data bus whereas memory is byte organized.

Random Access Memory (RAM) –6116

RAM1: 02000H - 02FFFH

Even Bank: 02000H - 02FFEH Odd Bank: 02001H – 02FFFH

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A19 | A18 | A17 | A16 | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |
| Start | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| End | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Read Only Memory (ROM) – 2732 ROM1: 00000H - 01FFFH

Even Bank: 00000H - 01FFEH Odd Bank: 01001H - 01FFFH

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A19 | A18 | A17 | A16 | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |
| Start | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| End | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**I/O MAPPING**

1. **8255-A:**

|  |  |  |
| --- | --- | --- |
| **Serial No.** | **Port Name** | **Starting Address** |
| 1 | Port A | 10H |
| 2 | Port B | 12H |
| 3 | Port C | 14H |
| 4 | CR(Control Register) | 16H |

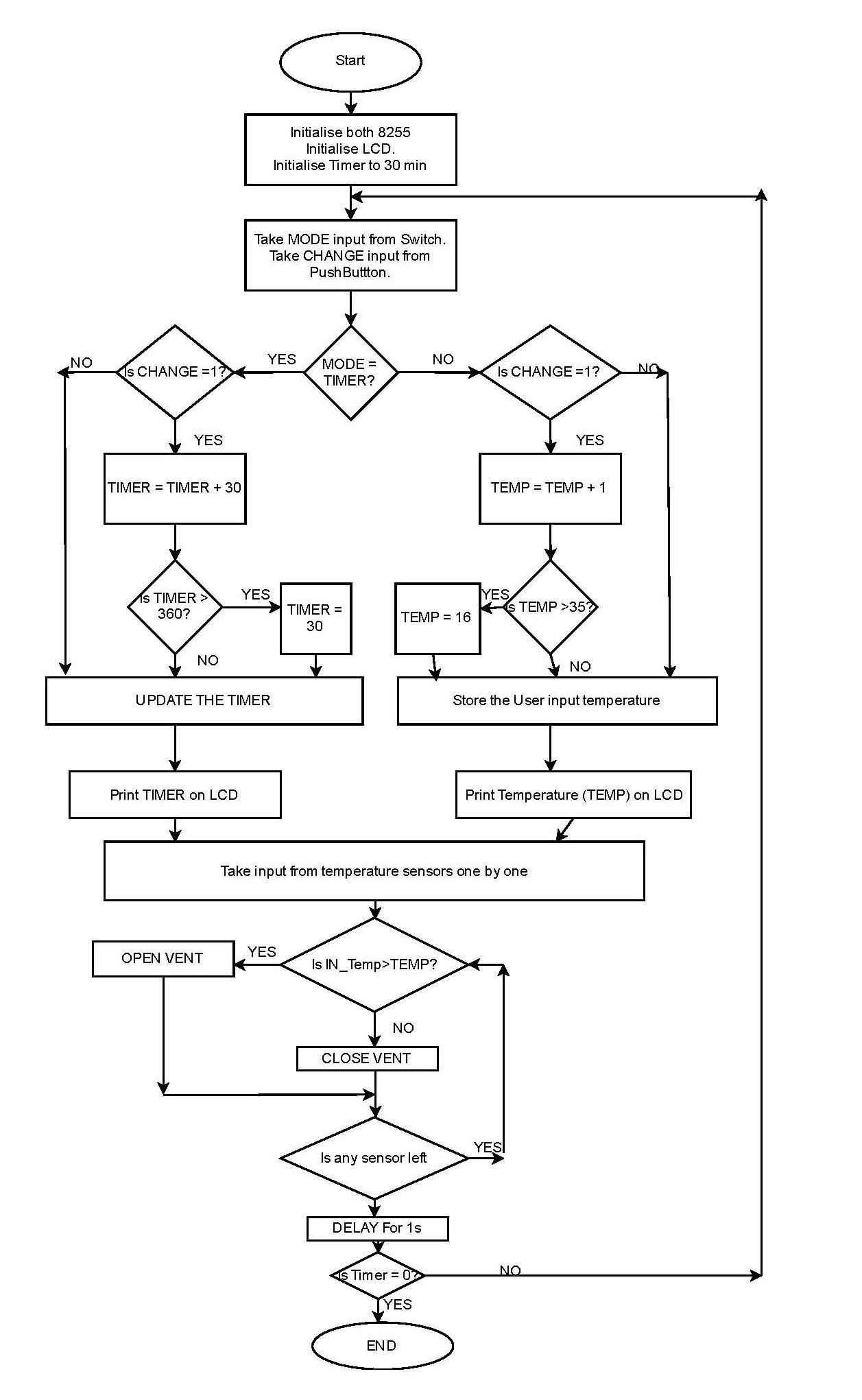
1. **8255-B:**

|  |  |  |
| --- | --- | --- |
| **Serial No.** | **Port Name** | **Starting Address** |
| 1 | Port A | 20H |
| 2 | Port B | 22H |
| 3 | Port C | 24H |
| 4 | CR(Control Register) | 26H |

**8253:**

|  |  |  |
| --- | --- | --- |
| **Serial No.** | **Port Name** | **Starting Address** |
| 1 | COUNTER 1 | 30H |
| 2 | COUNTER 2 | 32H |
| 3 | COUNTER 3 | 34H |
| 4 | CR(Control Register) | 36H |

**FLOWCHART**



**ALGORITHM**

1. After booting, all vents are opened and the reference room temperature becomes 250C. The duration for which the system is ON has a granularity of 30 minutes.
2. The input mode of the switch has two options, either to set the temperature of the AC system or to set the duration for which the system is ON.
3. a) Corresponding to the timer mode, the user can set the duration for which the AC system is ON, ranging from 30 minutes to 6 hours (360 minutes). If the time to be set goes beyond 360, it runs back to 30 minutes.

b) In the other mode, the temperature to be maintained is set, ranging from 160C to 350C by pressing the push button, one push increasing the temperature by 10C. If the temperature to be set goes beyond 350C, it runs back to 160C.

1. The temperature as sensed by the sensor is updated after certain interval (approximately 1sec). This temperature is compared with the temperature required to be set. If the two are not same, the AC valve is opened or closed depending on the temperature difference.
2. The LCD displays the temperature which is set as well as the time duration for which the AC is ON, depending on the mode of the switch.

**CODE**

.MODEL tiny

.DATA ;DATA DECLARATIONS

;TIMER-1 ADDRESS MAIN TIMER

CT0 EQU 30H CT1 EQU 32H CT2 EQU 34H CRG EQU 36H

; 8255-1 ADDRESS

PA1 EQU 10H PB1 EQU 12H PC1 EQU 14H CA1 EQU 16H

; 8255-2 ADDRESS

PA2 EQU 20H PB2 EQU 22H PC2 EQU 24H CA2 EQU 26H

; USER DATA

UTMP DB 25

TVAL DB 15

OPV DB 00

.CODE

.STARTUP

;INITIALIZING 8255-1

MOV AL, 10001001B OUT CA1, AL

;INITIALIZING 8255-2

MOV AL, 90h OUT CA2, AL

CALL LCD\_INIT ; INITIALIZING THE LCD

; INITIALIZING THE MAIN TIMER

MOV AL, 00110110B OUT CRG, AL

MOV AL, 01110110B OUT CRG, AL

MOV AL, 88H OUT CT0,AL MOV AL, 13H OUT CT0,AL

MOV AL, 60H OUT CT1, AL MOV AL, 0EAH OUT CT1, AL

;ENTERING TIME IN THE MAIN TIMER

MOV AL, 10010100B OUT CRG, AL

MOV AL, 15 OUT CT2, AL

RPT1:

IN AL, PC1 AND AL, 03H ROR AL, 1

JC TM1 ; JUMP TO TIMER MODE ROR AL, 1

JC X1

MOV AL, UTMP INC AL

CMP AL, 35 JLE X2

MOV AL, 16

X2:

MOV UTMP, AL

X1:

; SHOW TEMPERATURE OUTPUT IN LCD

CALL TEMP\_WRITE JMP E1

;TIMER MODE BEGINS FROM HERE

TM1:

ROR AL, 1 JC X3

MOV AL, TVAL MOV AH, 00

MOV BL, 15 DIV BL

MOV AH, 00 MUL BL ADD AL, 15

CMP AL, 195

JNZ X4 MOV AL, 15

X4:

MOV TVAL, AL

;ENTER TIME IN THE MAIN TIMER

MOV AL, 10010100B OUT CRG, AL

MOV AL, TVAL ;MOVE THE MINUTES OUT CT2, AL

X3:

;DISPLAY THE TIMER VALUE ON LCD

CALL TIME\_WRITE

E1: ;AFTER TIMER MODE

;INPUT OF ROOM TEMPERATURE FROM TEMPERATURES SENSORS

;AND TRANSFER IT TO AL

MOV AL, 00 ;TEMPERATURE SENSOR 1 CALL GET\_TEMP

CMP AL, UTMP JL TOK1

MOV AL, OPV OR AL, 01

MOV OPV, AL JMP TOK2

TOK1:

MOV AL, OPV AND AL, 0FEH MOV OPV,AL

TOK2:

; TEMPERATURE SENSOR 2

MOV AL, 01 CALL GET\_TEMP CMP AL, UTMP JL TOK3

MOV AL, OPV OR AL, 02 MOV OPV, AL JMP TOK4

TOK3:

MOV AL, OPV AND AL, 0FDH MOV OPV,AL

TOK4:

; TEMPERATURE SENSOR 3

MOV AL, 02

CALL GET\_TEMP CMP AL, UTMP JL TOK5

MOV AL, OPV OR AL, 04 MOV OPV, AL JMP TOK6

TOK5:

MOV AL, OPV AND AL, 0FBH MOV OPV,AL

TOK6:

; TEMPERATURE SENSOR 4

MOV AL, 03 CALL GET\_TEMP CMP AL, UTMP JL TOK7

MOV AL, OPV OR AL, 08 MOV OPV, AL JMP TOK8

TOK7:

MOV AL, OPV AND AL, 0F7H

MOV OPV,AL;

TOK8:

MOV AL, OPV OUT PB2, AL

;CALL DELAYX

; GET TIMER VAL

;IF ZERO REPEAT

;MOV AL,80H

;OUT CR2,AL

;IN AL, CT2

;CMP AL, 00

;JNZ RPT1

JMP RPT1 ;GO BACK AND TAKE NEXT INPUT

;FROM USER

INT 3H

.EXIT

DELAYX PROC

MOV SI, 43690

MOV BP, 43690 DELAY2:

DEC BP NOP

JNZ DELAY2 DEC SI

CMP SI,0 JNZ DELAY2

RET DELAYX ENDP

GET\_TEMP PROC

;ASSUMING AL HAS THE ADDRESS OF SENSOR TO BE SELECTED

out PC2, al

;give ale

OR al,00100000b

out PC2,al

;give soc

OR al,00110000b

out PC2,al

nop nop nop nop

;make ALE 0

AND al,11011111b

out PC2,al

;make SOC 0

AND al,11001111b

out PC2,al

RE1:

IN AL, PC1 AND AL, 04H JZ RE1

OR al,00001000b

out PC2, al

in al, PA2

RET

GET\_TEMP ENDP

LCD\_INIT PROC NEAR

MOV AL, 38H ;INITIALIZE THE LCD FOR 2 LINES AND ALSO 5\*7 MATRIX CALL COMNDWRT ;WRITE THE COMMAND TO LCD

CALL DELAY ;WAIT BEFORE SENDING THE NEXT COMMAND

MOV AL, 0EH ;SEND COMMAND FOR LCD ON, CURSOR ON CALL COMNDWRT

CALL DELAY

MOV AL, 01 ;CLEAR THE LCD

CALL COMNDWRT CALL DELAY

MOV AL, 06 ;COMMAND FOR SHIFTING CURSOR RIGHT CALL COMNDWRT

CALL DELAY RET

LCD\_INIT ENDP

DATWRIT PROC

PUSH DX ;save DX

MOV DX,PA1 ;DX=port A 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, PB1 ;port B 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

COMNDWRT PROC ;THIS PROCEDURE WRITES COMMANDS TO LCD MOV DX, PA1

OUT DX, AL ;SEND THE CODE TO PORT A MOV DX, PB1

MOV AL, 00000100B ;RS=0,R/W=0,E=1 FOR H-TO-L PULSE OUT DX, AL

NOP NOP

MOV AL, 00000000B ;RS=0,R/W=0,E=0 FOR H-TO-L PULSE OUT DX, AL

RET COMNDWRT ENDP

DELAY PROC

MOV CX, 1325 ;15.085\*1325 USEC = 20 MSEC W1:

NOP NOP NOP NOP NOP

LOOP W1 RET

DELAY ENDP

TEMP\_WRITE PROC NEAR PUSH AX

PUSH BX CALL CLS

CALL DELAY ;WAIT BEFORE SENDING THE NEXT CHARACTER

MOV BL, 10 MOV AL, UTMP MOV AH,00 DIV BL

ADD AL, '0' ;DISPLAY TENS OF TEMP

CALL DATWRIT ;SEND IT TO LCD

CALL DELAY ;WAIT BEFORE SENDING THE NEXT CHARACTER

MOV AL, AH

ADD AL, '0' ;DISPLAY ONES OF TEMP

CALL DATWRIT ;SEND IT TO LCD

CALL DELAY ;WAIT BEFORE SENDING THE NEXT CHARACTER

POP BX POP AX RET

TEMP\_WRITE ENDP

TIME\_WRITE PROC NEAR PUSH AX

PUSH BX CALL CLS

CALL DELAY ;WAIT BEFORE SENDING THE NEXT CHARACTER

MOV BL, 10 MOV AL, TVAL

MOV AH,00 ADD AX, AX

DIV BL

MOV BH, AH MOV AH,00 DIV BL

ADD AL, '0' ;DISPLAY HUNDREDS OF TEMP

CALL DATWRIT ;SEND IT TO LCD

CALL DELAY ;WAIT BEFORE SENDING THE NEXT CHARACTER

MOV AL, AH

ADD AL, '0' ;DISPLAY TENS OF TEMP

CALL DATWRIT ;SEND IT TO LCD

CALL DELAY ;WAIT BEFORE SENDING THE NEXT CHARACTER

MOV AL, BH

ADD AL, '0' ;DISPLAY ONES OF TEMP

CALL DATWRIT ;SEND IT TO LCD

CALL DELAY ;WAIT BEFORE SENDING THE NEXT CHARACTER

POP BX POP AX RET

TIME\_WRITE ENDP

CLS PROC

MOV AL, 01 ;CLEAR THE LCD

CALL COMNDWRT CALL DELAY

RET CLS ENDP

END

**CIRCUIT DIAGRAM :** Included in the zip folder in pdf format.

# REFERENCES

LM35 (Temperature sensor) (Device specifications) Range: -55oC to 150oC

Vin: 4V to 20V

For 0oC: OUTPUT = 0mV

Increment 10mV/oC

(2.55 V/255)

