		8086		
GND ←	1		40	← V_{CC}
AD_{14} ↔	2		39	↔ AD_{15}
AD_{13} ↔	3		38	→ AD_{16} / S_3
AD_{12} ↔	4		37	→ AD_{17} / S_4
AD_{11} ↔	5		36	→ AD_{18} / S_5
AD_{10} ↔	6		35	→ AD_{19} / S_6
AD_9 ↔	7		34	→ $\overline{BHE} / S_7$
AD_8 ↔	8		33	← $MN / \overline{MX}$
AD_7 ↔	9		32	→ $\overline{RD}$
AD_6 ↔	10		31	↔ HOLD $(\overline{RQ} / GT_0)$
AD_5 ↔	11		30	↔ HLDA $(\overline{RQ} / GT_1)$
AD			29	→ $\overline{WR}$ (LOCK)
			28	→ $M / \overline{IO}$ $(\overline{S_2})$
			7	→ $DT / \overline{R}$ $(\overline{S_1})$
				→ $\overline{DEN}$ $(\overline{S_0})$
				→ ALE (QS_0)
				→ $\overline{INTA}$ (QS_1)
				— $\overline{TEST}$
				— READY
				← RESET

SMART AC SYSTEMS

CS/EEE/ECE/INSTR F241 – MICROPROCESSOR PROGRAMMING AND INTERFACING

A REPORT

ON

"SMART AC SYSTEMS"

PREPARED FOR

Prof. K. R. Anupama
Department of Electrical and Electronics Engineering

By

2018A3PS0277G   NILAY SANJAY NAIK
2018A8PS0064G   SUMUKH DATTARAM PINGE
2018A8PS0019G   PRATIK PATIL
2018A3PS0550G   YASH KHANNA
2018A8PS0442G   VATSAL AGARWAL

In partial fulfilment of the requirements of
EEE/INSTR F241, Technical Report Writing

April 2020

Table of Contents

1. PROBLEM STATEMENT

Question 12-

Description

1. This system opens/closes six AC vents based upon the current temperature in the Room.

2. The temperature is maintained at a pleasant 20 –25-degree C. The AC vents can be gradually opened / closed. This is done in accordance with the temperature in the room.

3. The room is a fairly large sized room so 6 temperature sensors are placed at different points of the room.

4. Each sensor and AC vent are associated with part of the room.

5. You can assume that the room is broken up into 6 sub-areas each with its own sensor and ac vent.

User Interface:

1. Air-conditioner starts when user presses 'Start' button. User can also set the required temperature by using a keypad interface.

2. This temperature value should be displayed on a 7-segment display.

3. After setting temperature initially, user should be able to change the temperature setting by an up and down switch.

4. Each press on this arrow button increases/ decreases the temperature by one degree Celsius.

5. Min temp value is 20 deg, whereas maximum temp value is 25 deg Celsius. Pressing 'UP' button after reaching to 25 deg C, should not change the display value or setting of AC. Same is true for lower bound.

6. Air-conditioner can be stopped by pressing 'Stop' button.

7. User can also set the mode of AC as 'Bio-Sleep' mode besides a 'Regular mode' setting.

8. In Bio-sleep mode, user should be able to enter the value of time in terms of hours after which the AC has to be switched off automatically. (For example, if value entered is 2, then the AC should switch off after 2 hours from the time this setting is applied).

2. SYSTEM DESCRIPTION

1) Intel 8086 microprocessor.

2) Input Device:
   (i)   6 temperature sensors.
   (ii)   A 4x4 Matrix keypad.

3) Output Devices:
   (i)   6 stepper motors to open/close AC vents.
   (ii)   2 seven segment displays to display temperature and timer.

4) Clock is used to generate 5MHz clock signal for 8086.

5) 8253 timer is used.

   (i)   Clock 2 takes 5MHz input and generates 1MHz square wave output for ADC Clock and for input of Clock 1
   (ii) Clock 1 takes 1MHz input and gives 1KHz square wave output
   (iii) Clock 0 takes 1KHz input and gives approximately 2Hz output. This is the rate generator used for sensor polling

6) Five 8255 (Programmable Peripheral Interface) chips interfaced to 8086.

   (i)   **8255-A:** PortA is interfaced with the data lines of Seven Segment Display which shows LSB. PortB is interfaced to Seven Segment Display showing MSB. Port C is interfaced to 4X4 Matrix Keypad Input

   (ii)   **8255-B:** PortA takes input from ADC0808 which is interfaced with the 6 temperature sensors LM35. PortB is used to select the input channel on ADC. PC4 is used for detecting end of conversion. PC0 is used to give start of conversion.

   (iii)   **8255-C:** Port A, Port B, Port C are interfaced to the 1st-3rd stepper motors.

   (iv)   **8255-D:** Port A, Port B, Port C are interfaced to the 4th-6th stepper motors.

   (v)   **8255-E:** Port A0 used to detect output of rate generator, Port B0 controls gate of rate generator.

7) 74138 3-8 decoder is used to select which I/O device to use.

3. HARDWARE DEVICES

CHIP NUMBER	CHIP	PURPOSE
8086	MICROPROCESSOR	CENTRAL PROCESSING UNIT
6116(2)	RAM-2k	RANDOM ACCESS MEMORY-CONTAINS DS, SS
2732(2)	ROM-4K	READ ONLY MEMORY - CONTAINS ENTIRE CODE
74LS373(3)	8-BIT LATCH	TO LATCH ADDRESS BUS
74LS245(2)	8-BIT BUFFER	TO BUFFER DATA BUS (BIDIRECTIONAL)
8255(5)	PROGRAMMABLE PERIPHERAL INTERFACE	CONNECTED TO VARIOUS I/O DEVICES
ADC0808(1)	ANALOG TO DIGITAL CONVERTER	CONVERTS ANALOG VOLTAGE SIGNAL TO DIGITAL FORM
CLOCK GENERATOR	CLOCK	CLOCK INPUT FOR 8086 AND CLOCK2 of 8253
8253(1)	TIMER	PRODUCES CLOCK FOR RATE GENERATOR AND ADC
STEPPER MOTOR(6)		FOR OPENING/CLOSING AC VENTS
SEVEN SEGMENT DISPLAY(2)		TO DISPLAY DESIRED TEMPERATURE
LM35(6)	TEMPERATURE SENSOR	TO MEASURE TEMPERATURE IN VARIOUS PARTS OF THE ROOM
74138	3-8 DECODER	USED TO SELECT I/O DEVICE

4. I/O MAPPING

8255-A
- PORT A- 00H
- PORT B – 02H
- PORT C – 04H
- CR - 06H

8255-B
- PORT A - 08H
- PORT B – 0AH
- PORT C – 0CH
- CR - 0EH

8255-C
- PORT A- 10H
- PORT B – 12H
- PORT C – 14H
- CR - 16H

8255-D
- PORT A- 18H
- PORT B – 1AH
- PORT C – 1CH
- CR - 1EH

8253
- PORT A- 20H
- PORT B – 22H
- PORT C – 24H
- CR - 26H

8255-E
- PORT A- 28H
- PORT B – 2AH
- PORT C – 2CH
- CR - 2EH

5. ADDRESS MAPPING

MEMORY ORGANISATION:

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

Read Only Memory:
Starting Address: 00000h Ending Address: 00FFFh

Random Access Memory:
Starting Address: 01000h Ending Address: 03FFFh

MEMORY MAPPING

CHIP	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
ROM1(4K)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
To	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
RAM1(4K)	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
To	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

6. ASSUMPTIONS

1. ALP is already stored in the ROM in executable form.

2. The room is broken down into 6 parts. One sensor in placed in each part

3. The temperature of one part of room has no influence over the temperature of the other parts.

4. The temperature of each part of the room varies between 20-25∘C only.

5. If the temperature entered is greater than 25∘C it defaults to 25∘C. If lesser than 20∘C it defaults to 20∘C .

6. When the user presses the ON button, this starts the timer for polling the sensors.

7. When the temperature sensed by LM35 is less then Input Temp, the vent is closed.

8. The Vent is assumed to be closed at 0° reading.

9. Rotation of motor by 9 degree opens/closes the AC vent by one degree centigrade.

10. The motor reading (vent) is 36 degrees for fully open vent. When the difference in temperature sensed by the LM35 is >= 4°, The vent will be completely open.

7. WORKING

1) After pressing "ON" all vents are initially closed and the desired temperature defaults to 23∘C

2) The user specifies the initial temperature to be maintained by entering 2 digit temperature on the keypad. The user has to press the "SET TEMP" button if he has completed setting the temperature.

3) The temperature as sensed by the sensor is updated after every 0.5 seconds (2hz). This temperature is compared with the temperature required to be set. If the two are not same, the AC valve is opened or closed accordingly.

4) Whenever the user presses the T UP button or T DWN button required temperature is increased or decreased by 1∘C respectively.

6)When the user presses the BIOSLEEP button, this starts a software delay equal to the number of hours currently on the display. If greater than 09, defaults to 09. After this delay AC switches off. Please note, this step should only come after we set the initial temperature.

5) When the user presses the OFF button, this closes all the vents and stops the timer.

8. ALOGORITM / FLOWCHART
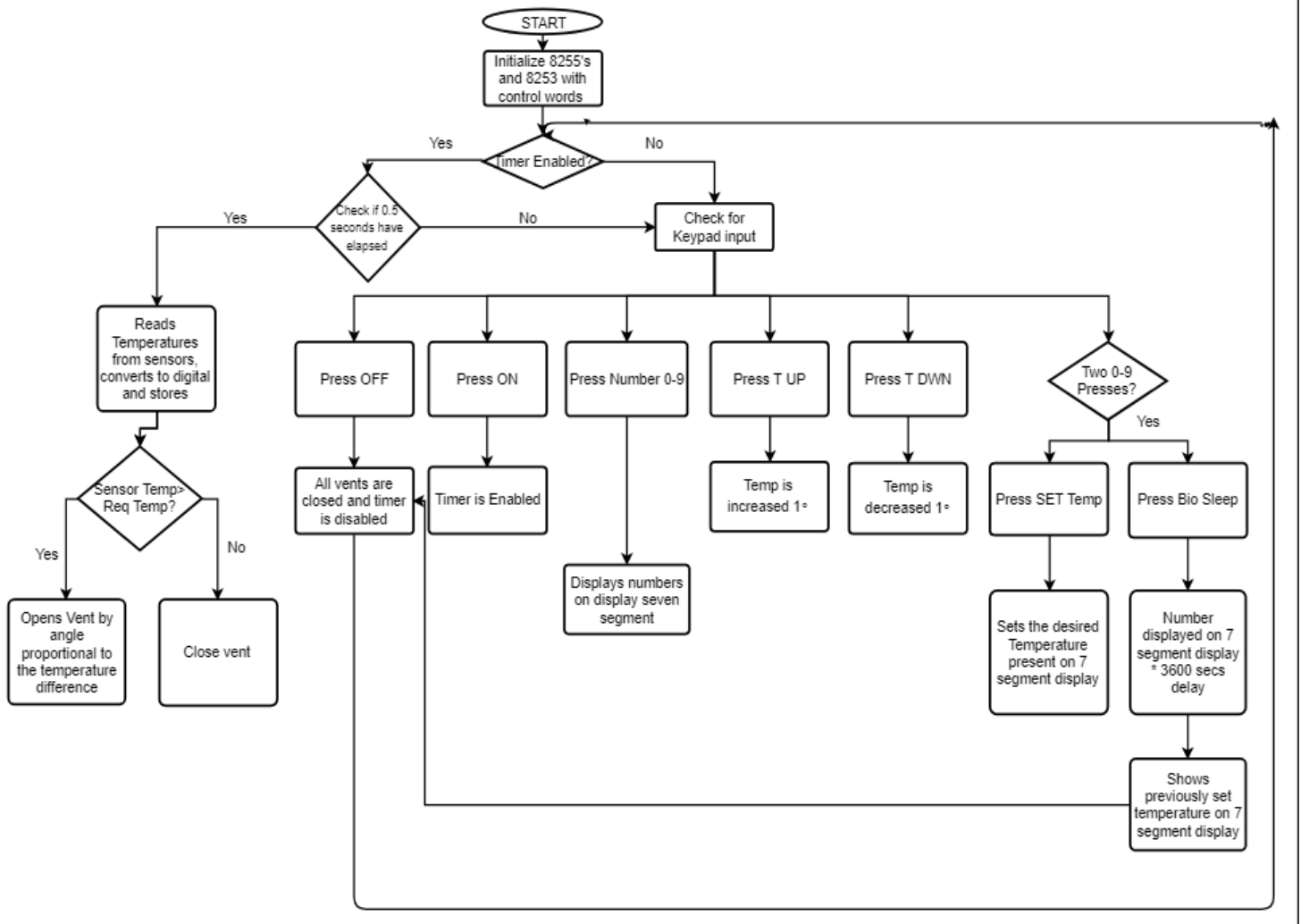
ISR FLOW CHART
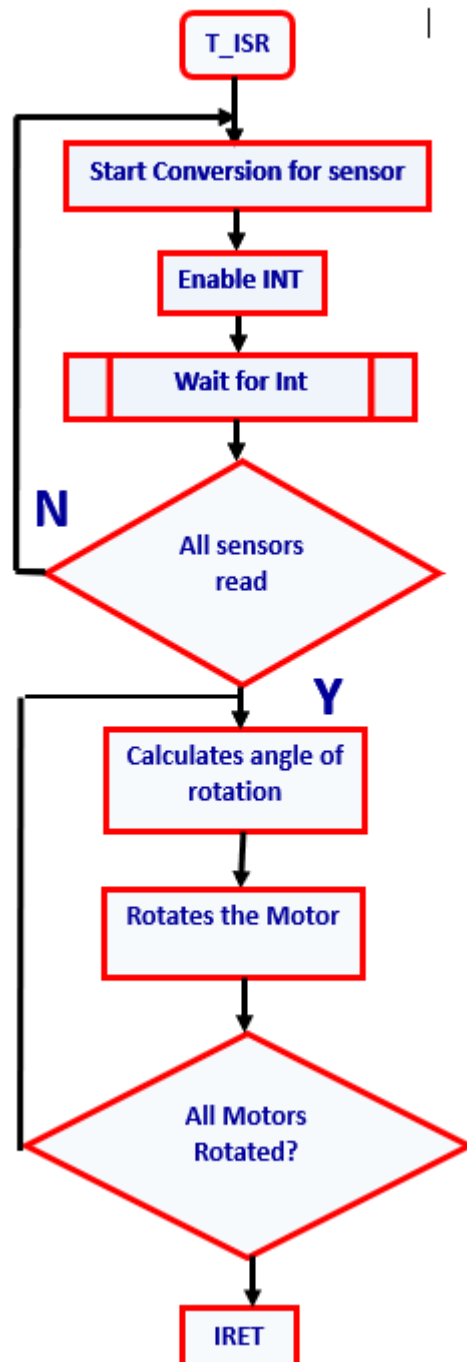
9. DESIGN
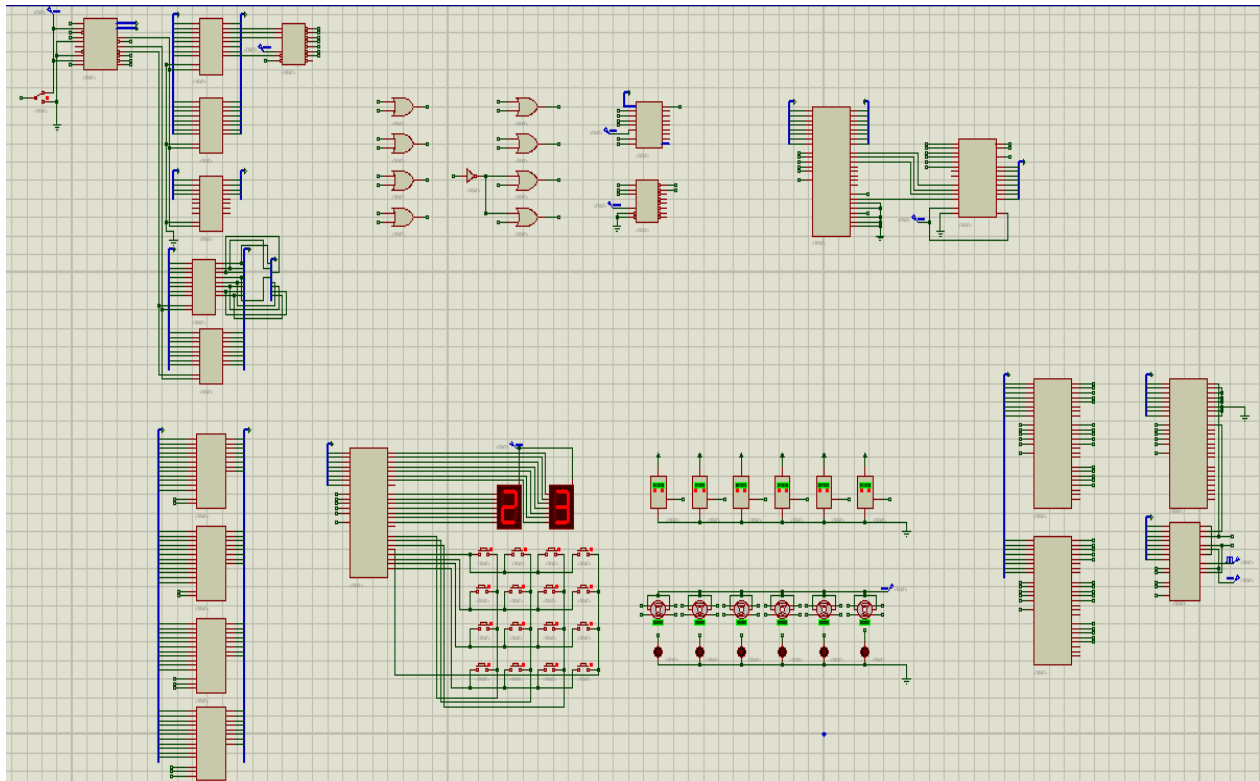
Kindly refer to the Proteus file for clearer view

10. CODE

```
#LOAD_SEGMENT=FFFFH
#LOAD_OFFSET=0000H

#AX=0000H
#BX=0000H
#CX=0000H
#DX=0000H
#SI=0000H
#DI=0000H
#BP=0000H

#CS=0000H
#IP=0000H

#DS=0000H
#ES=0000H
#SS=0000H
#SP=FFFEH


                              JMP     ST1
                              DB      509 DUP(0)
                              DW      T_ISR
                              DW      0000
                              DB      508 DUP(0)

; KEYPAD TABLE
KEYPAD_TABLE DB
0EEH,0EDH,0EBH,0E7H,0DEH,0DDH,0DBH,0D7H,0BEH,0BDH,0BBH,0B7H,07EH,07DH,07BH,077H

;DISPLAY TABLE
DISPLAY_TABLE DB 3FH,06H,5BH,4FH,66H, 6DH,7DH,27H,7FH,6FH

;STEPPER MOTOR TABLE
MOTOR_TABLE DB 19H,13H,16H,1CH,19H,13H
```

```
;ALL VARIABLES
KEY0 DB ?
KEY1 DB ?
TEMP0 DB ?
TEMP1 DB ?
TEMP  DB ?
T_SENSOR1 DB ?
T_SENSOR2 DB ?
T_SENSOR3 DB ?
T_SENSOR4 DB ?
T_SENSOR5 DB ?
T_SENSOR6 DB ?

HOURS DB ?
CURRENT_PORT DB ?
CURRENT_TEMP DB ?

DISP DB ?
DB                              465 DUP(0)

;PROGRAM START
ST1:    CLI

; INTIALIZE DS, ES,SS TO START THE RAM
MOV     AX,0200H
MOV     DS,AX
MOV     ES,AX
MOV     SS,AX
MOV     SP,0FFFEH


;INITIALIZE VARIABLES
MOV KEY0,00H
MOV KEY1,00H

MOV TEMP1,02H
MOV TEMP0,03H
MOV TEMP,23D
```

```
MOV T_SENSOR1,25D
MOV T_SENSOR2,25D
MOV T_SENSOR3,25D
MOV T_SENSOR4,25D
MOV T_SENSOR5,25D
MOV T_SENSOR6,25D

MOV HOURS,01H

MOV CURRENT_PORT,10H
MOV CURRENT_TEMP,17H

MOV DISP,01H
; 8255-A (STARTING 00H)
; INTIALISE KEYPAD
; UPPER PORT C AS INPUT ,PORT B ,PORT A AND LOWER PORT C AS OUTPUT
                              MOV    AL, 10001000B
                              OUT    06H,AL

; 8255-B (STARTING 08H)
; INITIALIZE SENSORS
; PORT A AND UPPER PORT C AS INPUT, PORT B AND LOWER PORT C AS (GIVES) OUTPUT,
                              MOV          AL, 10011000B
                              OUT          0EH, AL

; 8255-C (STARTING 10H)
;INITIALIZE MOTORS 1,2,3
;PORT A,B,C AS OUTPUT
                              MOV          AL, 10000000B
                              OUT          16H, AL

; 8255-D (STARTING 18H)
;INITIALIZE MOTORS 4,5,6;
;PORT A,B,C AS OUTPUT
                              MOV          AL, 10000000B
                              OUT          1EH, AL
```

```
;INITIALIZE TIMER 8253. (STARTING 20H)
;CLOCK 0 IN MODE 2 WITH 1KHZ INPUT
;CLOCK 1 IN MODE 3 WITH 1MHZ INPUT
;CLOCK 2 IN MODE 3 WITH 5MHZ INPUT
                                    MOV            AL, 00110100B
                                    OUT            26H, AL
                                    MOV    AL, 01110110B
                                    OUT    26H, AL
                                    MOV    AL, 10110110B
                                    OUT    26H, AL

;SEND COUNT OF 01F4H = 500D TO CLOCK 0
;SEND COUNT OF 03E8H = 1000D TO CLOCK 1
;SEND COUNT OF 0005H = 5D TO CLOCK 2
                                    MOV    AL,0F4H
                                    OUT    20H,AL
                                    MOV    AL,01H
                                    OUT    20H,AL

                                    MOV    AL,0E8H
                                    OUT    22H,AL
                                    MOV    AL,03H
                                    OUT    22H,AL

                                    MOV    AL,05H
                                    OUT    24H,AL
                                    MOV    AL,00H
                                    OUT    24H,AL
;OUT CLOCK 0 = 2HZ.
;OUT CLOCK 1 = 1KHZ.
;OUT CLOCK 2 = 1MHZ.

;INITIALIZE TIMER 8255-E
;PORT A INPUT,PORT B PORT C OUTPUT 28H
                                    MOV            AL, 10010000B
                                    OUT    2EH, AL
                                    MOV    AL,00H
                                    OUT    2AH, AL
```

```
;8259 -                        ENABLE IRO ALONE USE AEOI
                               MOV   AL,00010011B
                               OUT   30H,AL
                               MOV   AL,80H
                               OUT   32H,AL
                               MOV   AL,03H
                               OUT   32H,AL
                               MOV   AL,0FEH
                               OUT   32H,AL
                               STI

;INITIALIZE DISPLAY WITH 23 DEGREES.
                               MOV AL,4FH
                               NOT AL
                               OUT 00H,AL

                               MOV AL,5BH
                               NOT AL
                               OUT 02H,AL

;CHECK FOR KEY RELEASE -DEBOUNCE DELAY:
X0:
                               MOV      DH,00H
                               MOV      AL,00H
                               OUT      04H,AL
X1:
                               IN       AL,04H
                               AND      AL,0F0H
                               CMP      AL,0F0H
                               JNZ      X1

                               CALL     D20MS

;===========================================================================
```

;CHECKS FOR BIOSLEEP TIMER THEN POLLING THEN CHECKS FOR KEY PRESS:

```
                        MOV       AL,00H
                        OUT       04H ,AL

X2:                     IN        AL, 04H
                        AND       AL,0F0H
                        CMP       AL,0F0H
                        JZ        X2

                        CALL      D20MS

                        MOV       AL,00H
                        OUT       04H ,AL
                        IN        AL, 04H
                        AND       AL,0F0H
                        CMP       AL,0F0H
                        JZ        X2
```
;==

;DECODES KEY MATRIX
;CHECK COLUMN 0
```
                        MOV       AL, 0EH
                        MOV       BL,AL
                        OUT       04H,AL
                        IN        AL,04H
                        AND       AL,0F0H
                        CMP       AL,0F0H
                        JNZ       X3
```

;CHECK COLUMN 1
```
                        MOV       AL, 0DH
                        MOV       BL,AL
                        OUT       04H ,AL
                        IN        AL,04H
                        AND       AL,0F0H
                        CMP       AL,0F0H
                        JNZ       X3
```

;CHECK COLUMN 2

```
MOV     AL, 0BH
MOV      BL,AL
OUT     04H,AL
IN      AL,04H
AND      AL,0F0H
CMP      AL,0F0H
JNZ      X3
```

;CHECK COLUMN 3

```
MOV     AL, 07H
MOV      BL,AL
OUT     04H,AL
IN      AL,04H
AND      AL,0F0H
CMP      AL,0F0H
JZ      X2
```

;DECODE THE KEY
X3:

```
OR      AL,BL
MOV      CX,0FH
MOV      DI,00H
```

X4:

```
CMP      AL,CS:KEYPAD_TABLE[DI]
JZ      X5
INC     DI
INC     DH
LOOP    X4
```

;DISPLAY THE KEY
X5:

```
                              CMP       DH,09H
                              JG        BUTTON
                              LEA       BX, DISPLAY_TABLE
                              MOV       AL, CS:[BX+DI]
                              NOT       AL
                              MOV       DL,DISP
                              CMP       DL,00H
                              JNE       X6
                              OUT       00H,AL
                              XOR       DL,01H
                              MOV       DISP,DL
                              MOV       KEY0,DH
                              JMP       X0
```

X6:

```
                              OUT       02H,AL
                              XOR       DL,01H
                              MOV       DISP,DL
                              MOV       KEY1,DH
                              JMP       X0
```

;IF BUTTON PRESSED IS NOT A NUMBER, THIS PROCEDURE CHECKS WHICH BUTTON AND JUMPS
TO THE PROCEDURE
BUTTON:

```
                              CMP DH,0AH
                              JE T_UP
                              CMP DH,0BH
                              JE T_DWN
                              CMP DH,0CH
                              JE ON
                              CMP DH,0DH
                              JE BIOSLEEP
                              CMP DH,0EH
                              JE OFF
                              CMP DH,0FH
                              JE DEFAULT
                              JMP X0
```

```
;INCREASE TEMPERATURE BY 1 DEGREE(MAX 25 DEGREES)
T_UP:
                              MOV CL,TEMP0
                              CMP CL,05H
                              JE X0
                              INC CL
                              MOV TEMP0,CL
                              MOV AL,TEMP
                              ADD AL,1
                              MOV TEMP,AL
                              CALL DISPLAY0
                              JMP X0




;DECREASE TEMPERATURE BY 1 DEGREE(MIN 20 DEGREES)
T_DWN:
                              MOV CL,TEMP0
                              CMP CL,00H
                              JE X0
                              DEC CL
                              MOV TEMP0,CL
                              MOV AL,TEMP
                              SUB AL,1
                              MOV TEMP,AL
                              CALL DISPLAY0
                              JMP X0




;STARTS RATE GENERATOR FOR TAKING INPUT FROM SENSORS PERIODICALLY
ON:
                              ;STARTS TIMER
                              MOV   AL,01H
                              OUT   2AH, AL
                              JMP X0
```

```
;SETS SLEEP TIMER(UPTO 9 HOURS),AFTER TAKING INPUT RESETS DISPLAY TO TEMPERATURE
BIOSLEEP:

    MOV AL,KEY1
                                MOV BL,KEY0
                                MOV AH,00
                                MOV CL,0AH
                                MUL CL
                                ADD AL,BL
                                CMP AX,0009H
                                JG HOURH
                                MOV HOURS,AL
                                CALL DISPLAY0
                                CALL DISPLAY1
                                JMP TIMER_START
HOURH:

                                MOV HOURS,09H
                                CALL DISPLAY0
                                CALL DISPLAY1
                                JMP TIMER_START



;DISABLES RATE GENERATOR, SHUTS ALL AC VENTS
OFF:
                                MOV AL,06H
                                OUT 10H,AL
                                OUT 12H,AL
                                OUT 14H,AL
                                OUT 18H,AL
                                OUT 1AH,AL
                                OUT 1CH,AL
                                MOV   AL,00H
                                OUT   2AH, AL
                                JMP X0
```

;SETS TEMPERATURE EQUAL TO NUMBER CURRENTLY ON DISPLAY, IF >25, AUTOMATICALLY SETS TO 25,SAME WITH <20

```
DEFAULT:
                                MOV AL,KEY1
                                MOV TEMP1,AL
                                MOV CL,0AH
                                MUL CL
                                MOV CL,KEY0
                                MOV TEMP0,CL
                                ADD AL,CL
                                MOV TEMP,AL
                                CMP AL,25D
                                JG DEFH
                                CMP AL,20D
                                JL DEFL
                                JMP X0


DEFH:
                                MOV TEMP1,02H
                                MOV TEMP0,05H
                                MOV AL,25D
                                MOV TEMP,AL
                                CALL DISPLAY0
                                CALL DISPLAY1
                                JMP X0


DEFL:
                                MOV TEMP1,02H
                                MOV TEMP0,00H
                                MOV AL,20D
                                MOV TEMP,AL
                                CALL DISPLAY0
                                CALL DISPLAY1
                                JMP X0
```

```
;MULTIPLIES USER INPUT WITH 3600 AND STARTS SLEEP TIMER
TIMER_START:
                                MOV    BL,HOURS
                                MOV    AX,3600d
                                MUL    BL
DHOUR:

                                CALL   D1S
                                DEC    AX
                                JNZ    DHOUR
                                JMP    OFF


JMP X0

;COMPARES TEMPERATURE ENTERED WITH POLLED TEMPERATURE AND MOVES VENT
ACCORDING THE CALCULATIONS
MOTOR:
                                MOV BL,TEMP
                                MOV AL,CURRENT_TEMP
                                SUB AL,BL
                                CMP AL,01H
                                JE P1
                                CMP AL,02H
                                JE P2
                                CMP AL,03H
                                JE P3
                                CMP AL,04H
                                JGE P4

                                JMP NO

NO:

                                MOV DL,CURRENT_PORT
                                MOV DH,00H
                                MOV AL,06H
                                OUT DX,AL
                                RET
```

```
P1:
                                MOV DL,CURRENT_PORT
                                MOV DH,00H
                                MOV AL,12H
                                OUT DX,AL
                                RET

P2:
                                MOV DL,CURRENT_PORT
                                MOV DH,00H
                                MOV AL,13H
                                OUT DX,AL
                                RET

P3:
                                MOV DL,CURRENT_PORT
                                MOV DH,00H
                                MOV AL,11H
                                OUT DX,AL
                                RET

P4:
                                MOV DL,CURRENT_PORT
                                MOV DH,00H
                                MOV AL,19H
                                OUT DX,AL
                                RET


;CHECKS FOR EOC
CONV:
                                IN AL,0CH
                                AND AL,0F0H
                                CMP AL,10H
                                JNE CONV
                                RET
```

;UPDATE LSB OF DISPLAY
DISPLAY0:

```
                              MOV CL,TEMP0
                              MOV CH,00H
                              MOV DI,CX
                              LEA BX, DISPLAY_TABLE
                              MOV AL,CS:[BX+DI]
                              NOT AL
                              OUT 00H,AL
                              RET
```

;UPDATE MSB OF DISPLAY
DISPLAY1:

```
                              MOV CL,TEMP1
                              MOV CH,00H
                              MOV DI,CX
                              LEA BX, DISPLAY_TABLE
                              MOV AL,CS:[BX+DI]
                              NOT AL
                              OUT 02H,AL
                              RET
```

;GENERATES DEBOUNCE DELAY
D20MS:

```
                              MOV        CX,20 ; DELAY GENERATED
XN:

                              LOOP       XN
                              RET
```

```
D1S:

                              MOV        CX,50000 ; DELAY GENERATED IS 1S
```

```
XS:

                              LOOP       XS
                              RET
```

```
#INTURREPT SERVICE ROUTINE
T_ISR:

   PUSH AX
                                        ;SELECT SENSOR 1
                                            MOV AL,00H
                                            OUT 0AH,AL

                                        ;HIGH TO LOW TRANSITION ON START AND ALE
                                            MOV AL,00H
                                            OUT 0CH,AL
                                            MOV AL,03H
                                            OUT 0CH,AL
                                            MOV AL,00H
                                            OUT 0CH,AL

                                        ;WAIT FOR CONVERSION
                                            CALL CONV

                                        ;STORE READ TEMPERATURE
                                            IN AL,08H
                                            MOV T_SENSOR1,AL

                                        ;SELECT SENSOR 2
                                            MOV AL,01H
                                            OUT 0AH,AL

                                        ;HIGH TO LOW TRANSITION
                                            MOV AL,00H
                                            OUT 0CH,AL
                                            MOV AL,03H
                                            OUT 0CH,AL
                                            MOV AL,00H
                                            OUT 0CH,AL

                                        ;WAIT FOR CONVERSION
                                            CALL CONV

                                        ;STORE READ TEMPERATURE
                                            IN AL,08H
                                            MOV T_SENSOR2,AL
```

```
                            ;SELECT SENSOR 3
                                MOV AL,02H
                                OUT 0AH,AL

                            ;HIGH TO LOW TRANSITION
                                MOV AL,00H
                                OUT 0CH,AL
                                MOV AL,03H
                                OUT 0CH,AL
                                MOV AL,00H
                                OUT 0CH,AL

                            ;WAIT FOR CONVERSION
                                CALL CONV

                            ;STORE READ TEMPERATURE
                                IN AL,08H
                                MOV T_SENSOR3,AL

                            ;SELECT SENSOR 4
                                MOV AL,03H
                                OUT 0AH,AL

                            ;HIGH TO LOW TRANSITION
                                MOV AL,00H
                                OUT 0CH,AL
                                MOV AL,03H
                                OUT 0CH,AL
                                MOV AL,00H
                                OUT 0CH,AL

                            ;WAIT FOR CONVERSION
                                CALL CONV

                            ;STORE READ TEMPERATURE
                                IN AL,08H
                                MOV T_SENSOR4,AL

                            ;SELECT SENSOR 5
                                MOV AL,04H
                                OUT 0AH,AL
```

```
                        ;HIGH TO LOW TRANSITION
                            MOV AL,00H
                            OUT 0CH,AL
                            MOV AL,03H
                            OUT 0CH,AL
                            MOV AL,00H
                            OUT 0CH,AL

                        ;WAIT FOR CONVERSION
                            CALL CONV

                        ;STORE READ TEMPERATURE
                            IN AL,08H
                            MOV T_SENSOR5,AL

                        ;SELECT SENSOR 6
                            MOV AL,05H
                            OUT 0AH,AL

                        ;HIGH TO LOW TRANSITION
                            MOV AL,00H
                            OUT 0CH,AL
                            MOV AL,03H
                            OUT 0CH,AL
                            MOV AL,00H
                            OUT 0CH,AL

                        ;WAIT FOR CONVERSION
                            CALL CONV

                        ;STORE READ TEMPERATURE
                            IN AL,08H
                            MOV T_SENSOR6,AL
;===========================================================================
```

; UPDATE MOTORS

```
                              MOV AL,T_SENSOR1
                              MOV CURRENT_TEMP,AL
                              MOV CURRENT_PORT,10H
                              CALL MOTOR

                              MOV AL,T_SENSOR2
                              MOV CURRENT_TEMP,AL
                              MOV CURRENT_PORT,12H
                              CALL MOTOR

                              MOV AL,T_SENSOR3
                              MOV CURRENT_TEMP,AL
                              MOV CURRENT_PORT,14H
                              CALL MOTOR

                              MOV AL,T_SENSOR4
                              MOV CURRENT_TEMP,AL
                              MOV CURRENT_PORT,18H
                              CALL MOTOR

                              MOV AL,T_SENSOR5
                              MOV CURRENT_TEMP,AL
                              MOV CURRENT_PORT,1AH
                              CALL MOTOR

                              MOV AL,T_SENSOR6
                              MOV CURRENT_TEMP,AL
                              MOV CURRENT_PORT,1CH
                              CALL MOTOR
                              POP AX
IRET
```

11.   LISTS OF ATTACHMENTS

1. Hardware.pdf.
2. Manuals
   a. ADC 0808
   b. LM35
   c. Stepper Motor
3. Proteus File – Smart_AC-Proteus.pdsprj
4. EMU8086 ASM File – Smart_AC.asm
5. Binary File after assembly – SmartAC.bin