A Report On

SMART AC SYSTEMS

Submitted by

Group number 17

Abhinav Mishra (2014B3A7625P)
Amitojdeep Singh (2014B3A7615P)
Megha Shishodia (2014B3A7605P)
Rajas Pandey (2014B3A7582P)

For the course

CS/EEE/INSTR F241 Microprocessor

Programming and Interfacing

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE

PILANI, RAJASTHAN-333031

25th April 2017

CONTENTS

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 ^{0}C. 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 ^{0} C.

Single push button to vary temperature between 16^{0}C-35^{0}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 the ROM in executable form.

2) The temperatures of all parts of the room are independent of each other, as the room is assumed to be big.

3) After system startup, the temperature of each part of the room varies between 16 - 35°C only.

4) When all AC vents are completely open room temperature will be 16°C and when all are completely closed, the room temperature will be 35°C.

5) Rotation of motor by 90 degree opens/closes the AC vent.

6) When the AC is switched off, all the vents are completely closed.

7) The first address of the processor ends out when it is switched on is 01000h. At this location, there is a jump instruction which takes the program control to the beginning of the code.

8) There exists a mechanism which controls the flaps in such a way that it rotates motor just 90 degrees at once, then to 180 for closing, again to 270 for opening & then to 360 for closing the flap and so on.

9) Either of the push button needs to be held for at least one second to get the desired change.

SYSTEM DESCRIPTION

1) Intel 8086 microprocessor.

2) **INPUT DEVICE:**

   (i) 4 temperature sensors.

   (ii) 2 push buttons

3) **OUTPUT DEVICES:**

   (i) LCD to display temperature.

   (ii) 4 motors to open/close AC vents following assumption 8.

4) Two 8255 (Programmable Peripheral Interface) chips interfaced to 8086.

   (i) **8255-A(PORTS_LCD):** Port-A is interfaced to the 8 data lines of LCD driver HD244780. PB0 and PB7 are connected to the RS and R/W of LCD driver, respectively. PC0 is used to vary the mode temperature/timer; PC1 is used for setting the temperature.

   (ii) **8255-A(ADC_PORT):** Port-A takes input from ADC0808 which is interfaced with the 4 temperature sensors LM35. Port-C is used to select the input channel on ADC.

5) 8284 clock is used to generate 2.5 MHz clock signal for 8086.

6) 8253 is used to generate stepped down time signals for the given problem statement making use of the 2.5MHz clock signal from 8284.

7) The motors are operated by Darlington pair and controlled in a mechanism as in assumption 8.

HARDWARE DEVICES

CHIP NUMBER(No. of chips)	CHIP	USE
8086	Microprocessor	Central Processing Unit(C.P.U)
6116(2)	RAM-2K	Random Access Memory containing DS and SS segments.
2732(2)	ROM-4K	Read only Memory which contains entire code.
74LS373(3)	8-BIT LATCH	To Latch Address Bus.
74LS245(4)	8-BIT BUFFER	To Buffer Data Bus (BIDIRECTIONAL)
8255(2)	PROGRAMMABLE PERIPHERAL INTERFACE	Connected to Various Input/ Output Devices.
ADC0808(1)	ANALOG TO DIGITAL CONVERTER	CONVERTS ANALOG VOLTAGE SIGNAL V_{ce} TO DIGITAL FORM
8253(1)	CLOCK TIMER	TO KEEP THE TRACK OF TIME FOR WHICH THE PROCESSOR WILL WORK
LM 020	LCD DISPLAY	FOR OPENING/CLOSING AC VENTS TO DISPLAY TEMPERATURE/ TIMER
LM 35(4)	TEMPERATURE SENSOR	TO PRODUCE ANALOG SIGNAL FOR THE TEMPERATURE IN ROOM

74154	4:16 DECODER	TO PRODUCE THE CHIP SELECT SIGNALS FOR IO DEVICES
74138	3:8 DECODER	TO PRODUCE CHIP SELECT SIGNALS FOR ROM AND RAM
DC MOTOR(4)	12V MOTOR	CONNECTED TO DARLINGTON PAIR ARRAY
ULN2003A	DARLINTON PAIR ARRAY	TO SIMULATE THE OPENING AND CLOSING OF VENTS IN PROTEUS BY CONNECTING IT TO THE MOTOR

MEMORY INTERFACING

This system uses 4KB of RAM (as 2x2KB chips for even and odd banks respectively) and 4KB 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 while memory is byte organised.

Random Access Memory (RAM) –6116

RAM 01E (Even Bank) Start: 00000h End: 00FFEh	RAM01: Starting Address:00000h Ending Address:00FFFh
RAM 01O (Odd Bank) Start: 00001h End: 00FFFh	

	A1	A18	A17	A16	A15	A14	A13	A12	A11	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
SA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
EA	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

Read Only Memory (ROM) –2732

ROM 01E (Even Bank) Start: 01000h End: 03FFEh	ROM01: Starting Address:01000h Ending Address:03FFFh
ROM 01O (Odd Bank) Start: 01001h End: 03FFFh	

	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
SA	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
EA	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

I/O MAPPING

8255- 1:

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

8255-2 :

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

TIMER :

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

WORKING

1) All vents are completely opened upon starting, and the room temperature is 25°C. The duration for which the system is ON has a granularity of 30 minutes.

2) There are 2 push buttons. One to set the timer, and one to set the temperature.

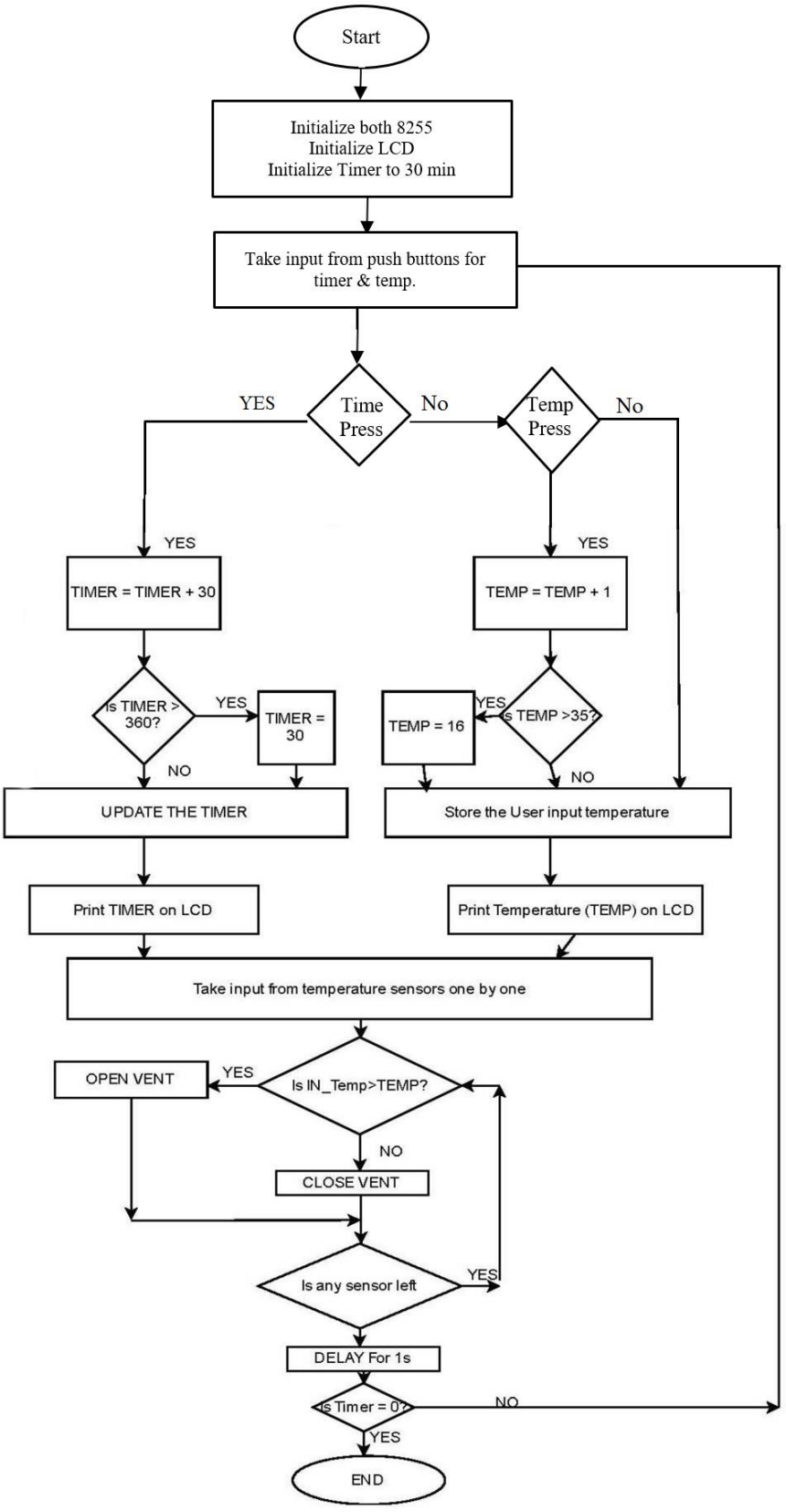
3) By setting the timer, the user can set the duration for which the AC system is ON, ranging from 30 minutes to 6 hours, one push increasing the timer by 30 minutes. If the time to be set goes beyond 360 min., it is reset to 30 minutes.

4) The temperature to be maintained is set ranging from 16°C to 35°C by pressing the second push button, one push increasing the temperature by 1°C. If temperature exceeds 35°C, it is reset to 16°C.

4) 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 there is a difference, the AC valve is opened or closed depending on whether it is higher or lower than the input temperature. (In Proteus, a motor is used to simulate that behavior.)

5) Depending on the push button pressed, the LCD displays the temperature set or the Time duration set.

FLOWCHART:


Start
Initialize both 8255
Initialize LCD
Initialize Timer to 30 min
Take input from push buttons for timer & temp.
Time Press
YES
No
Temp Press
No
YES
TIMER = TIMER + 30
YES
TEMP = TEMP + 1
Is TIMER > 360?
YES
TIMER = 30
NO
TEMP = 16
YES
Is TEMP >35?
NO
UPDATE THE TIMER
Store the User input temperature
Print TIMER on LCD
Print Temperature (TEMP) on LCD
Take input from temperature sensors one by one
OPEN VENT
YES
Is IN_Temp>TEMP?
NO
CLOSE VENT
Is any sensor left
YES
DELAY For 1s
Is Timer = 0?
NO
YES
END

CODE:

```
.MODEL tiny
.DATA

;TIMER-1 ADDRESS
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

    ;INITIALISE 8255-1
    MOV AL, 10001001B
    OUT CA1, AL

    ;INITIALISE 8255-2
```

```
        MOV AL, 90h
        OUT CA2, AL

        CALL LCD_INIT; INITIALIZES THE LCD

        ; 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

        ;PUT TIME IN 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
        JNC X1
        MOV AL, UTMP
        INC AL
```

```
        CMP AL, 35
        JLE X2
        MOV AL, 16
X2:
        MOV UTMP, AL
X1:
        ; SHOW TEMP OUTPUT IN LCD
        CALL TEMP_WRITE
        JMP E1

;TIMER MODE STARTS
TM1:

        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
        ;PUT TIME IN MAIN TIMER
        MOV AL, 10010100B
        OUT CRG, AL
        MOV AL, TVAL  ;MOVE THE MINUTES
        OUT CT2, AL
X3:
        ; SHOW TIMER VALUE IN LCD
        CALL TIME_WRITE

E1: ;AFTER TIMER MODE

        ; CHECK FOR TEMPERATURES SENSORS
```

```
        ;IN FROM ROOM TEMP TO AL
        MOV AL, 00
        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:

        ; TEMP 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:
        ; TEMP 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:
        ; TEMP 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

        INT 3H
```

```
.EXIT

DELAYX PROC
        ; START DELAY
        MOV SI, 43690
        MOV BP, 43690
DELAY2:
        DEC BP
        NOP
        JNZ DELAY2
        DEC SI
        CMP SI,0
        JNZ DELAY2
        ; END DELAY
        RET
DELAYX ENDP

GET_TEMP PROC

                ;ASSUMING AL HAS THE 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
```

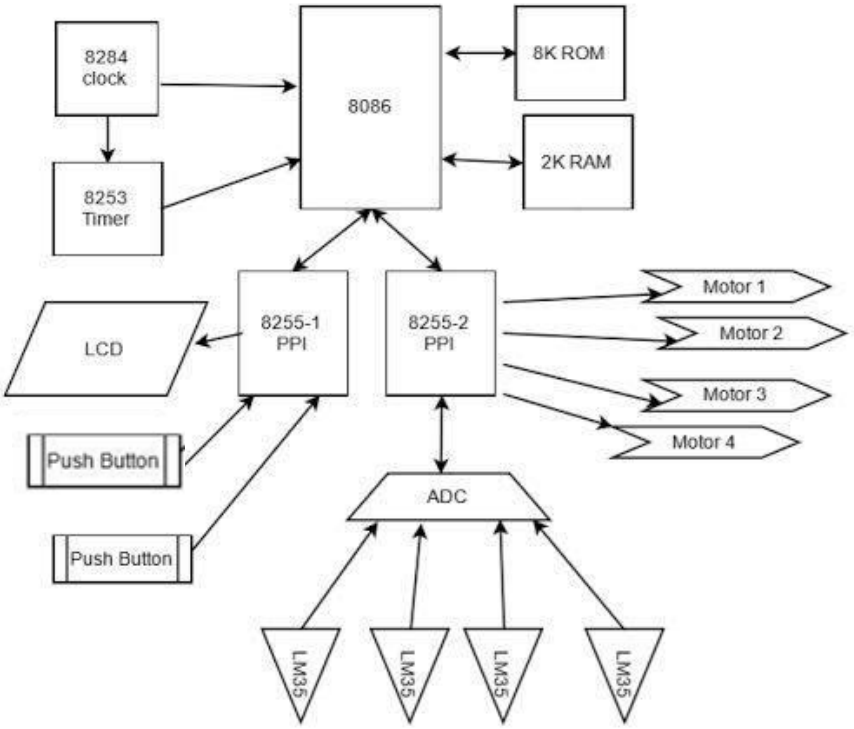
```
;make SOC 0

            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 LCD FOR 2 LINES & 5*7 MATRIX
    CALL COMNDWRT ;WRITE THE COMMAND TO LCD
    CALL DELAY ;WAIT BEFORE ISSUING THE NEXT COMMAND
    MOV AL, 0EH ;SEND COMMAND FOR LCD ON, CURSOR ON
    CALL COMNDWRT
    CALL DELAY
    MOV AL, 01  ;CLEAR 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 ;writing on the lcd ends

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 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

TEMP_WRITE PROC NEAR
        PUSH AX
        PUSH BX
        CALL CLS
        CALL DELAY ;WAIT BEFORE ISSUING THE NEXT CHARACTER

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

        ADD AL, '0' ;DISPLAY TENS OF TEMP
        CALL DATWRIT ;ISSUE IT TO LCD
        CALL DELAY ;WAIT BEFORE ISSUING THE NEXT CHARACTER

        MOV AL, AH
        ADD AL, '0' ;DISPLAY ONES OF TEMP
        CALL DATWRIT ;ISSUE IT TO LCD
        CALL DELAY ;WAIT BEFORE ISSUING 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 ISSUING 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 ;ISSUE IT TO LCD
        CALL DELAY ;WAIT BEFORE ISSUING THE NEXT CHARACTER

        MOV AL, AH
        ADD AL, '0' ;DISPLAY TENS OF TEMP
        CALL DATWRIT ;ISSUE IT TO LCD
        CALL DELAY ;WAIT BEFORE ISSUING THE NEXT CHARACTER

        MOV AL, BH
        ADD AL, '0' ;DISPLAY ONES OF TEMP
        CALL DATWRIT ;ISSUE IT TO LCD
        CALL DELAY ;WAIT BEFORE ISSUING THE NEXT CHARACTER

        POP BX
        POP AX
        RET
TIME_WRITE ENDP

CLS PROC
        MOV AL, 01  ;CLEAR LCD
        CALL COMNDWRT
        CALL DELAY
        RET
CLS ENDP

END
```

CIRCUIT DIAGRAM:



REFERENCES:

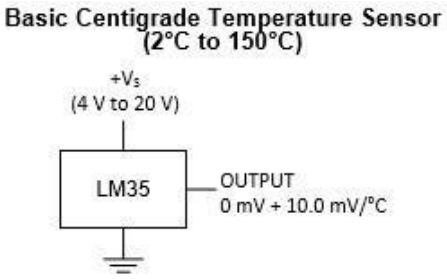
LM35 (Temperature sensor)

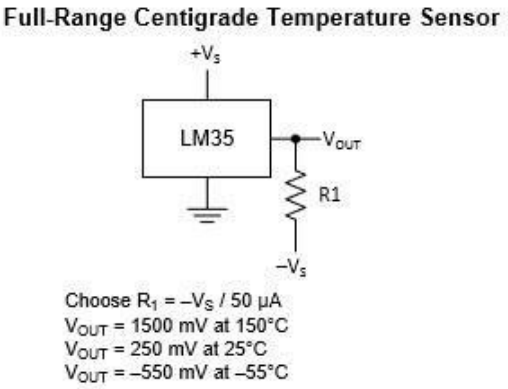
Range: -55 °C to 150 °C

Vin: 4V to 20V

For 0°C: OUTPUT = 0mV

Increment 10mV/°C



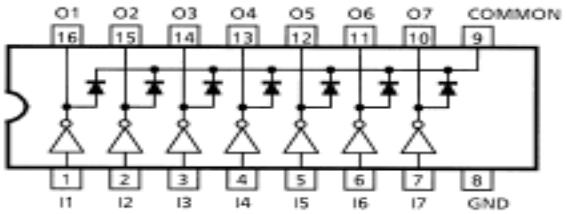


ULN2003A (Darlington pair array)

Vin = 30V
Vout = 50V

Source: Texas Instruments Datasheet
http://www.ti.com/lit/ds/slrs027o/slrs027o.pdf accessed on 24th April 2017.

Internal Schematic



Pin-out