BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

Spring, 2019

Course No: CS F241

Course Title: Microprocessors and Interfacing

DESIGN ASSIGNMENT- Smart AC System (P2)

Group 88

2017A7PS0113P SOOREJ S NAIR
2017A7PS0114P YASHDEEP GUPTA
2017A7PS0116P AYUSH SINGHAL
2017A7PS0117P SATVIK GOLECHHA

PROBLEM STATEMENT (P2)

Smart AC System

- 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 degrees Celsius.

- 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.

- It is assumed that the room is broken up into 4 sub-areas each with its own sensor and AC vent.

- **User Interface:** LCD displaying Temperature and a single push button to vary temperature between 16-35 degrees.

- 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.

SYSTEM DESCRIPTION

1) Intel 8086 microprocessor.
2) INPUT:
   (i) 4 temperature sensors.
   (ii) 3 push buttons.
3) OUTPUT DEVICES:
   (i) 2 LCDs to display temperature.
   (ii) 4 motors to open/close AC vents.

4) Three 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, PB1 and PB2 are connected to the RS, R/W and E of LCD driver, respectively.

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

(iii) 8255-A(COUNTER_PORT): Port-A is used to initialize the counter 74LS169. Port-B is used to take the output from the counter. Port-C is used to toggle the push buttons used for varying the temperature and time.

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.

HARDWARE DEVICES

CHIP NUMBER(no of chips)	CHIP	USE
8086	MICROPROCESSOR	CPU
6116(2)	RAM-2k	Random Access Memory which contains DS,SS
2732(2)	ROM-4K	READ ONLY MEMORY WHICH CONTAINS THE ENTIRE CODE
74LS373(3)	8-BIT LATCH	TO LATCH ADDRESS BUS
74LS245(4)	8-BIT BUFFER	TO BUFFER DATA BUS (BIDIRECTIONAL)
8255(3)	PROGRAMMABLE PERIPHERAL INTERFACE	CONNECTED TO VARIOUS INPUT OUTPUT DEVICES
ADC0808(1)	ANALOG TO DIGITAL CONVERTER	CONVERTS ANALOG VOLTAGE SIGNAL V_{ce} TO DIGITAL FORM
8284(1)	CLOCK TIMER	TO PRODUCE THE STABLE FREQUENCY CLOCK SIGNAL WHICH STEPS THE 8086 EXECUTION OF ITS INSTRUCTIONS IN AN ORDERLY MANNER

Stepper motor(4) LM 020 HD244780(1)	 LCD DISPLAY LCD DRIVER	FOR OPENING/CLOSING AC VENTS TO DISPLAY TEMPERATURE DRIVER FOR LCD
LM35(4) --- NPN DARLINGTON AMPLIFIER PAIRS	TEMPERATURE SENSOR --- AMPLIFIERS	TO PRODUCE ANALOG SIGNAL --- TO AMPLIFY THE SIGNAL PROVIDED TO THE MOTORS
74LS138(3)	3-to-8 DECODER	To decide which IC to select for the next operation.
74LS169(1)	4-BIT SYNCHRONOUS BINARY UP-DOWN COUNTER	To count the number of 30 min periods passed
DC MOTOR(4)	12V MOTOR	CONNECTED TO DARLINGTON PAIR ARRAY
ULN2003A	DARLINGTON PAIR ARRAY	TO SIMULATE THE OPENING AND CLOSING OF VENTS IN PROTEUS BY CONNECTING IT TO THE MOTOR

ADDRESS MAPPING

MEMORY ORGANISATION:

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 while memory is byte organised.

ROM-2732(4k / chip)
ROM1 and ROM2 (Even + Odd): 0x20000 to 0x21FFF
RAM-6116(2k / chip)
RAM1 and RAM2 (Even + Odd): 0x00000 to 0x00FFF

I/O MAPPING

- 8255-A
  - PORT A- 10H
  - PORT B – 12H
  - PORT C – 14H
  - CR - 16H
- 8255-B
  - PORT A- 20H
  - PORT B – 22H
  - PORT C – 24H
  - CR - 26H
- 8255-C
  - PORT A- 40H
  - PORT B – 42H
  - PORT C – 44H
  - CR - 46H
- TIMER
  - COUNTER 1 - 30H
  - COUNTER 2 - 32H
  - COUNTER 3 - 34H
  - CR - 36H

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) 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.

8) The time of the system is set at the beginning and cannot be changed in the middle of the operation.

9) The error in temperature detection is +/- 1 degrees.

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 3 push buttons. One to set the timer,one to set the temperature and one to increase it.

3) By pressing the first push button, 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. By pressing the second push button, user confirms the time entered, and the AC starts working.

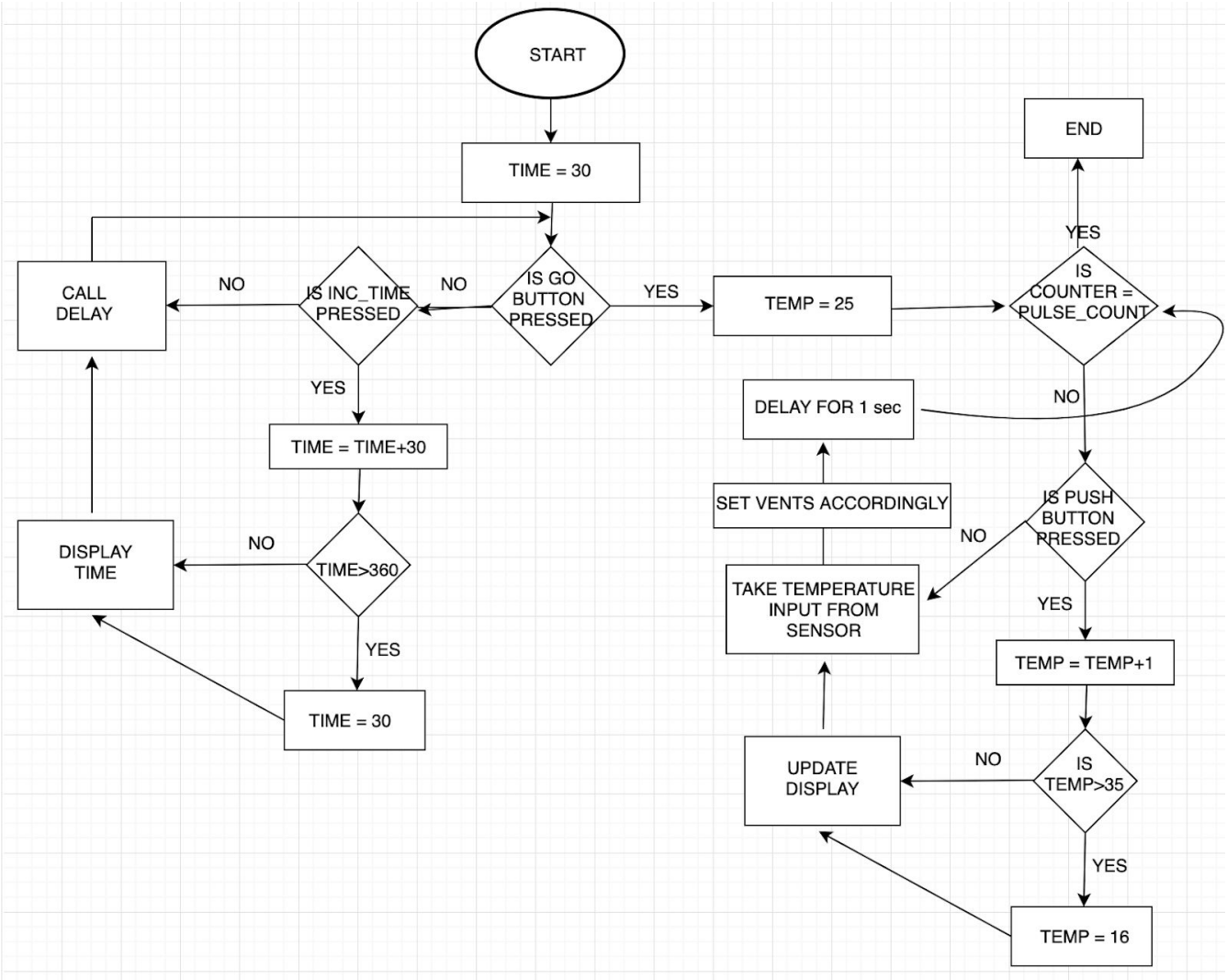
4) The temperature to be maintained is set ranging from 16°C to 35°C by pressing the third push button, one push increasing the temperature by 1°C. If temperature exceeds 35°C, it is reset to 16°C.

4) The temperature as sensed by the sensor is updated after certain interval (approximately 1 sec) 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) The new temperature is displayed whenever the third push button is pressed.

6) Updation of either time or temperature takes approximately 0.1 sec.

FLOWCHART


START
TIME = 30
CALL DELAY
NO
IS INC_TIME PRESSED
NO
IS GO BUTTON PRESSED
YES
TEMP = 25
END
YES
IS COUNTER = PULSE_COUNT
NO
YES
TIME = TIME+30
DELAY FOR 1 sec
SET VENTS ACCORDINGLY
IS PUSH BUTTON PRESSED
NO
DISPLAY TIME
NO
TIME>360
TAKE TEMPERATURE INPUT FROM SENSOR
YES
TEMP = TEMP+1
YES
TIME = 30
UPDATE DISPLAY
NO
IS TEMP>35
YES
TEMP = 16

COMPLETE ALP

```
#make_bin#

; BIN is plain binary format similar to .com format, but not limited to 1
segment;
; All values between # are directives, these values are saved into a
separate .binf file.
; Before loading .bin file emulator reads .binf file with the same file
name.

; All directives are optional, if you don't need them, delete them.

; set loading address, .bin file will be loaded to this address:
#LOAD_SEGMENT=0500h#
#LOAD_OFFSET=0000h#

; set entry point:
#CS=0500h# ; same as loading segment
#IP=0000h# ; same as loading offset

; set segment registers
#DS=0500h# ; same as loading segment
#ES=0500h# ; same as loading segment

; set stack
#SS=0500h# ; same as loading segment
#SP=FFFEh# ; set to top of loading segment

; set general registers (optional)
#AX=0000h#
#BX=0000h#
#CX=0000h#
#DX=0000h#
```

```
#SI=0000h#
#DI=0000h#
#BP=0000h#

; add your code here
JMP  _CODE
;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
; 8255-3 ADDRESS
PA3 EQU 40H
PB3 EQU 42H
PC3 EQU 44H
CA3 EQU 46H
; USER DATA
VENTSTATE DB 0
TEMPDISPLAY DB 25
TIMEDISPLAY DB 03H ; STORES FIRST TWO DIGITS OF TIME

_CODE:
MOV AL, 80H
OUT CA1, AL
;INITIALISE 8255-2
MOV AL, 90H
OUT CA2, AL
;INTIALISE 8255-3
MOV AL, 83H
OUT CA3, AL
```

```
MOV AL,6
CALL RESET_PUSH
;INIT LCD
CALL LCD_INIT
MOV TIMEDISPLAY,3
MOV TEMPDISPLAY,25
MOV AL,01100000B
OUT PC3,AL
CALL TIME_WRITE
      LABEL1:CALL DELAY
            IN AL,PC3
            AND AL,00001010B
            CMP AL,0
            JE LABEL1

            MOV BL,AL
            AND BL,00000010B
            CMP BL,0
            JE CHECK2
            MOV BL,TIMEDISPLAY
            ADD BL,3
            MOV TIMEDISPLAY,BL
            CMP BL,36
            JLE CHECK3
            MOV BL,3
            MOV TIMEDISPLAY,BL

            CHECK3:CALL TIME_WRITE
            CHECK2:
            MOV BL,AL
            AND BL,08H
            CMP BL,0
            CALL RESET_PUSH
            JNE LABEL2

      JMP LABEL1
      LABEL2:CALL RESET_PUSH
      CALL TEMP_WRITE
      CALL TIMER_INIT
      CALL COUNTER_INIT
```

```
LABEL3:
      IN AL,PB3;CHECK IF OPERATION TIME HAS ELAPSED
      AND AL,0FH
      MOV AH,0
      MOV DL,3
      MUL DL
      CMP AL,TIMEDISPLAY
      JGE EXIT

      IN AL,PC3;CHECK IF TEMPERATURE PUSH BUTTON WAS PRESSED
      AND AL,00000100B
      CMP AL,0
      JE NORMAL;CONTINUE NORMAL FUNCTIONING
      ;ELSE INCREASE TEMPERATURE AND DISPLAY IT
      ADD TEMPDISPLAY,1
      CMP TEMPDISPLAY,35
      JLE DISPLAY
      MOV TEMPDISPLAY, 16
      DISPLAY:
      CALL TEMP_WRITE
      CALL RESET_PUSH
      CALL DELAY_SHORT

      NORMAL:;GETS VENTSTATE IN INPUT
            MOV AL,0
            RE:
                  CALL GET_TEMP;OUTPUTS TEMPERATURE IN CL
                  MOV CH,0

                  PUSH AX
                        MOV AX,CX
                        MOV CX,100
                        MUL CX
                        MOV DX,0
                        MOV BX,256
                        DIV BX
                        OUT PC1,AL
                        CALL DELAY
                        CALL DELAY
```

```
                              CMP AL,TEMPDISPLAY
                              JG OPEN
                         POP AX
                         MOV DL,0FEH
                         MOV CL,AL
                         ROL DL,CL
                         AND VENTSTATE,DL
                         JMP VENTDONE
                         OPEN:POP AX
                         MOV DL,01H
                         MOV CL,AL
                         ROL DL,CL
                         OR VENTSTATE,DL
                         VENTDONE:
                         CALL DELAY
                         INC AL
                         CMP AL,4
                    JNZ RE
               MOV AL,VENTSTATE
               OUT PB2,AL
               CALL DELAY;CALL DELAY OF 20 MILISECOND
     JMP LABEL3

     EXIT:;;;;;CLOSE ALL VENTS ,YOU HAVENT DONE THIS YET
     MOV AL,0
     OUT PB2,AL
_STOP: JMP _STOP

GET_TEMP PROC NEAR
     ;ASSUMING AL HAS THE SENSOR TO BE SELECTED
     PUSH AX
     PUSH BX
     MOV BL,AL;STORE COPY OF AL
     out PC2, al

     OR al,00100000b;ALE
     out PC2,al

     CALL DELAY_SHORT
```

```
    OR al,00110000b;SOC
    out PC2,al

    CALL DELAY_SHORT

    AND al,11011111b;ALE 0
    out PC2,al

    CALL DELAY_SHORT

    AND al,11001111b;SOC 0
    out PC2,al

    RE1:;WAIT FOR EOC
    CALL DELAY_SHORT
    IN AL, PC3
    AND AL, 01H
    CMP AL,0
    JZ RE1

    CALL DELAY_SHORT

    MOV AL,BL;SET OE HIGH
    OR AL,00001000B
    OUT PC2, AL

    CALL DELAY_SHORT

    IN AL, PA2;TAKE TEMPERATURE AS INPUT
    MOV CL,AL

    CALL DELAY_SHORT

    MOV AL,BL;SET OE LOW
    AND AL,00000111B
    OUT PC2,AL

    CALL DELAY_SHORT

    POP BX
```

```
      POP AX
      RET
GET_TEMP ENDP

LCD_INIT PROC NEAR
      PUSH AX
      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
      CALL CLS
      MOV AL, 06 ;COMMAND FOR SHIFTING CURSOR RIGHT
      CALL COMNDWRT
      CALL DELAY
      POP AX
      RET
LCD_INIT ENDP

TIMER_INIT PROC NEAR ;STARTS TIMER SO IT GENERATES PULSES AT EVERY 30
MINUTES
      PUSH AX; INITIALIZE 1ST COUNTER
      MOV AL,34H
      OUT CRG,AL
      MOV AL,0E8H
      OUT CT0,AL
      MOV AL,03H
      OUT CT0,AL
      ; INIT 2ND COUNTER
      MOV AL,74H
      OUT CRG,AL
      MOV AL,0E8H
      OUT CT1,AL
      MOV AL,03H
      OUT CT1,AL
      ;INIT 3RD COUNTER
      MOV AL,0B4H
      OUT CRG,AL
      MOV AL,94H
```

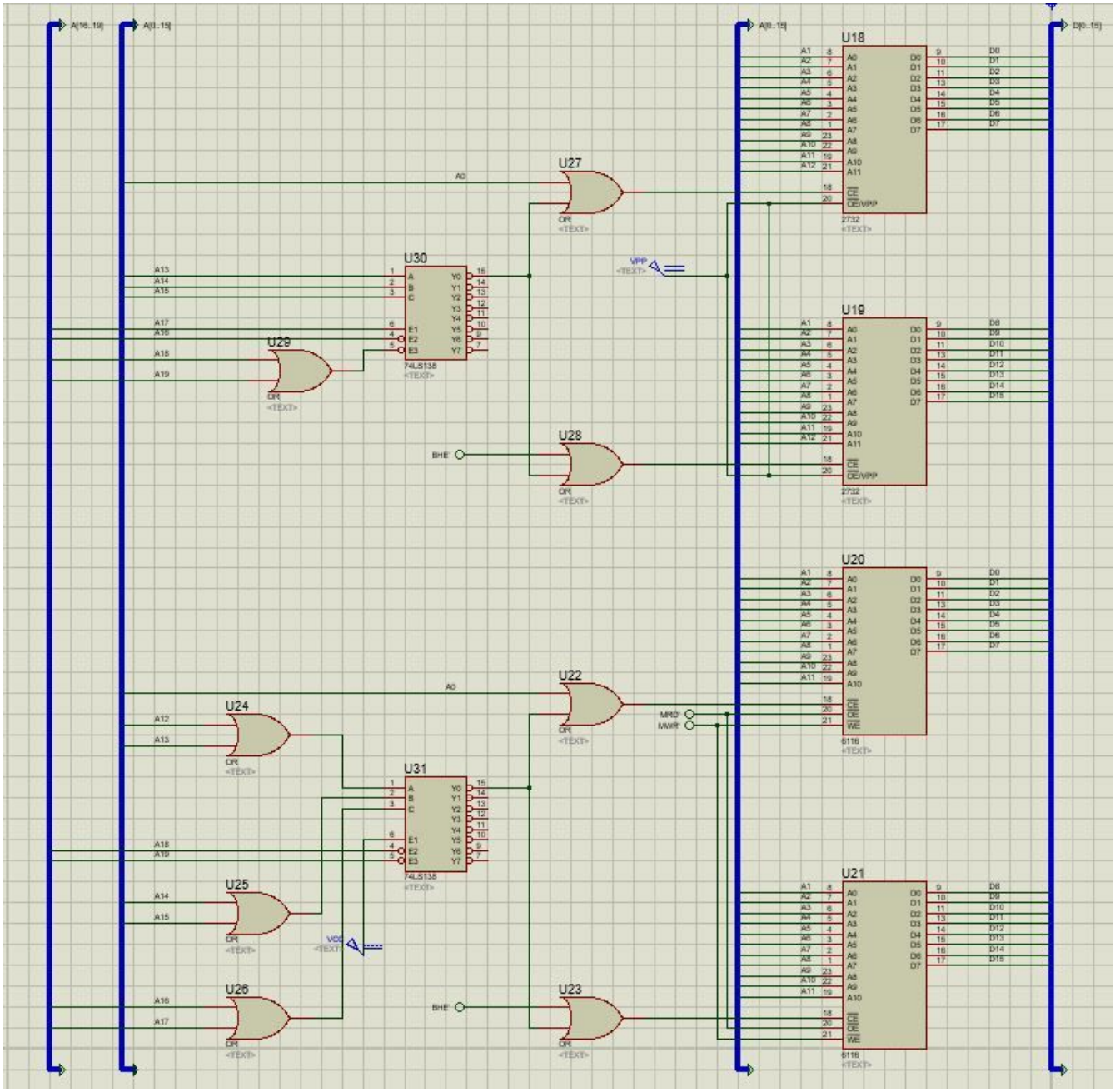
```
      OUT CT2,AL
      MOV AL,11H
      OUT CT2,AL
      POP AX
      RET
TIMER_INIT ENDP

COUNTER_INIT PROC NEAR
      PUSH AX
      MOV AL,01000000B
      OUT PC3,AL
      MOV AL,0
      OUT PA3,AL
      MOV AL,01100000B
      OUT PC3,AL
      POP AX
      RET
COUNTER_INIT ENDP

DATWRIT PROC NEAR
      PUSH DX ;save DX
      PUSH AX
      OUT PA1, AL ;issue the char to LCD
      MOV AL, 00000101B ;RS=1, R/W=0, E=1 for H-to-L pulse

      OUT PB1, AL ;make enable high
      nop
      nop
      MOV AL, 00000001B ;RS=1,R/W=0 and E=0 for H-to-L pulse
      OUT PB1, AL
      POP AX
      POP DX
      RET
DATWRIT ENDP ;writing on the lcd ends



COMNDWRT PROC NEAR;THIS PROCEDURE WRITES COMMANDS TO LCD
      PUSH AX
      OUT PA1, AL ;SEND THE CODE TO PORT A
```

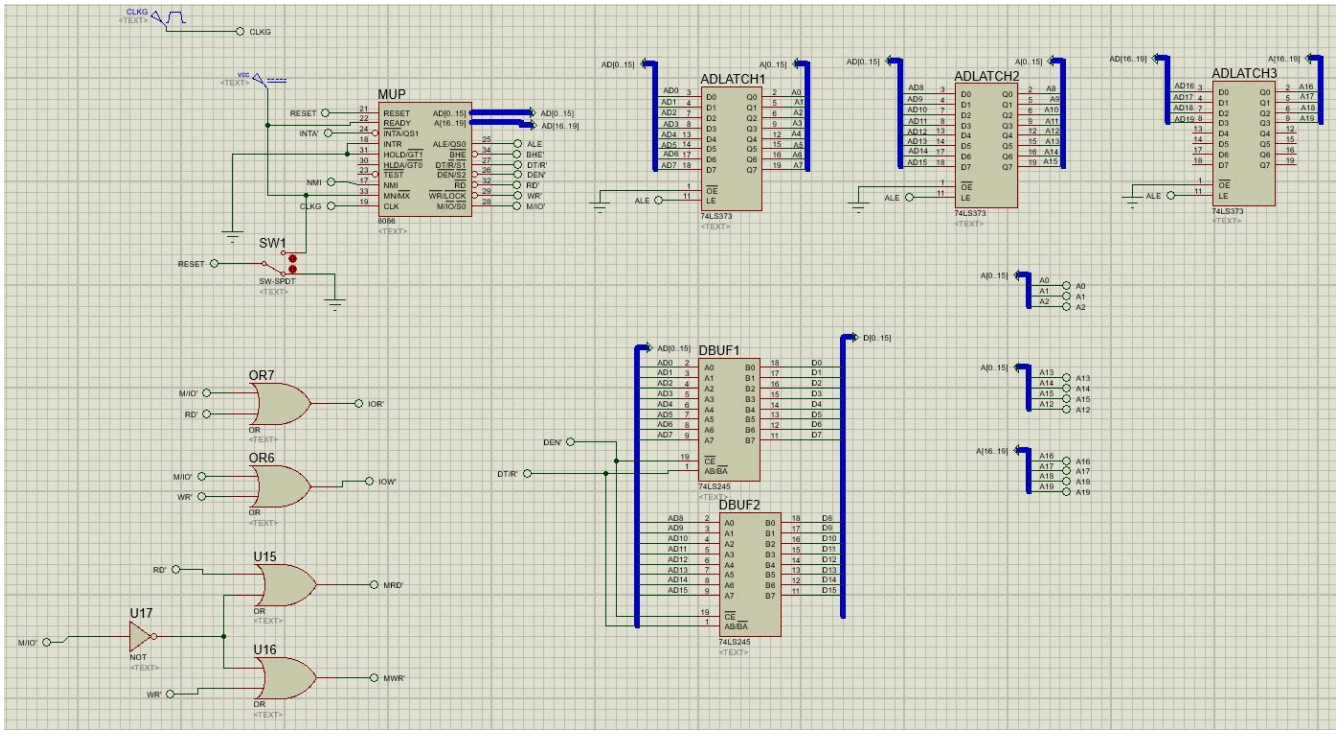
```
      MOV AL, 00000100B ;RS=0,R/W=0,E=1 FOR H-TO-L PULSE
      OUT PB1, AL
      NOP
      NOP
      MOV AL, 00000000B ;RS=0,R/W=0,E=0 FOR H-TO-L PULSE
      OUT PB1, AL
      POP AX
      RET
COMNDWRT ENDP


;DELAY IN THE CIRCUIT HERE THE DELAY OF 20 MILLISECOND IS
;PRODUCED
DELAY PROC NEAR
      PUSH CX
      MOV CX, 1325 ;1325*15.085 USEC = 20 MSEC
      W1:
      NOP
      NOP
      NOP
      NOP
      NOP
      LOOP W1
      POP CX
      RET
DELAY ENDP
DELAY_SHORT PROC NEAR
      PUSH CX
      MOV CX,10 ;10*15.085 USEC = 150 USEC
      Work:
      NOP
      NOP
      NOP
      NOP
      NOP
      LOOP Work
      POP CX
      RET
DELAY_SHORT ENDP
```

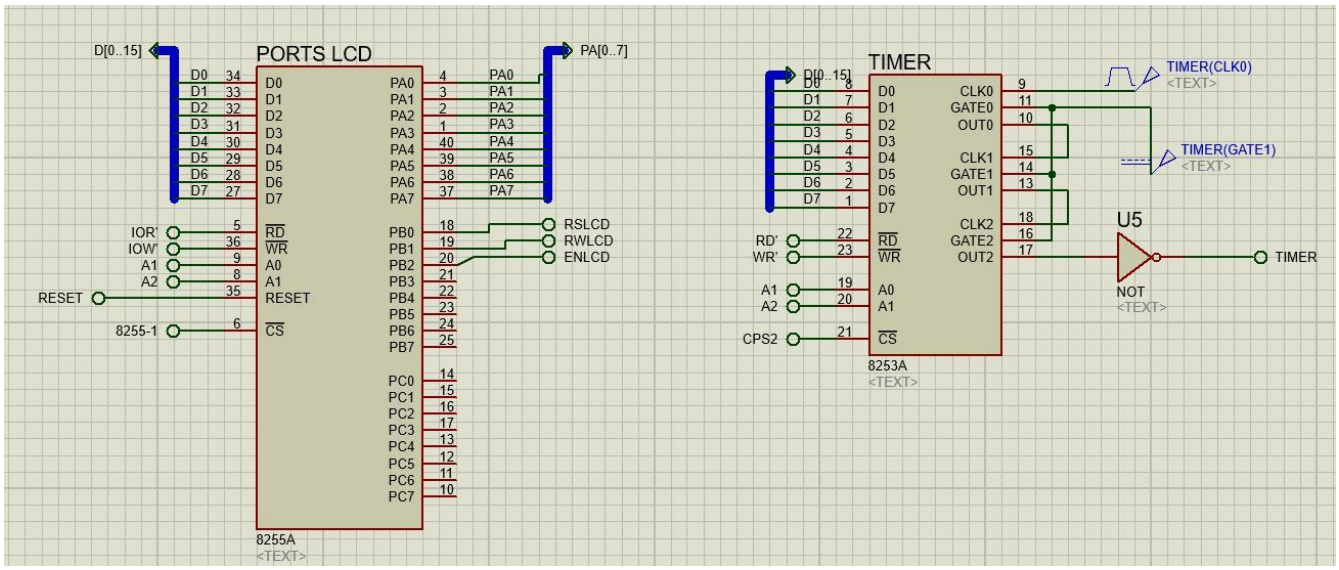
```
RESET_PUSH PROC NEAR
      PUSH AX
      MOV AL,00100000B
      OUT PC3,AL
      NOP
      NOP
      MOV AL,01100000B
      OUT PC3,AL
      POP AX
      RET
RESET_PUSH ENDP

TEMP_WRITE PROC NEAR
      PUSH AX
      PUSH BX
      CALL CLS
      CALL LCD_INIT
      CALL DELAY ;WAIT BEFORE ISSUING THE NEXT CHARACTER
      MOV BL, 10
      MOV AL, TEMPDISPLAY
      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
      MOV AL,0
      OUT PB1,AL
      CALL DELAY
```

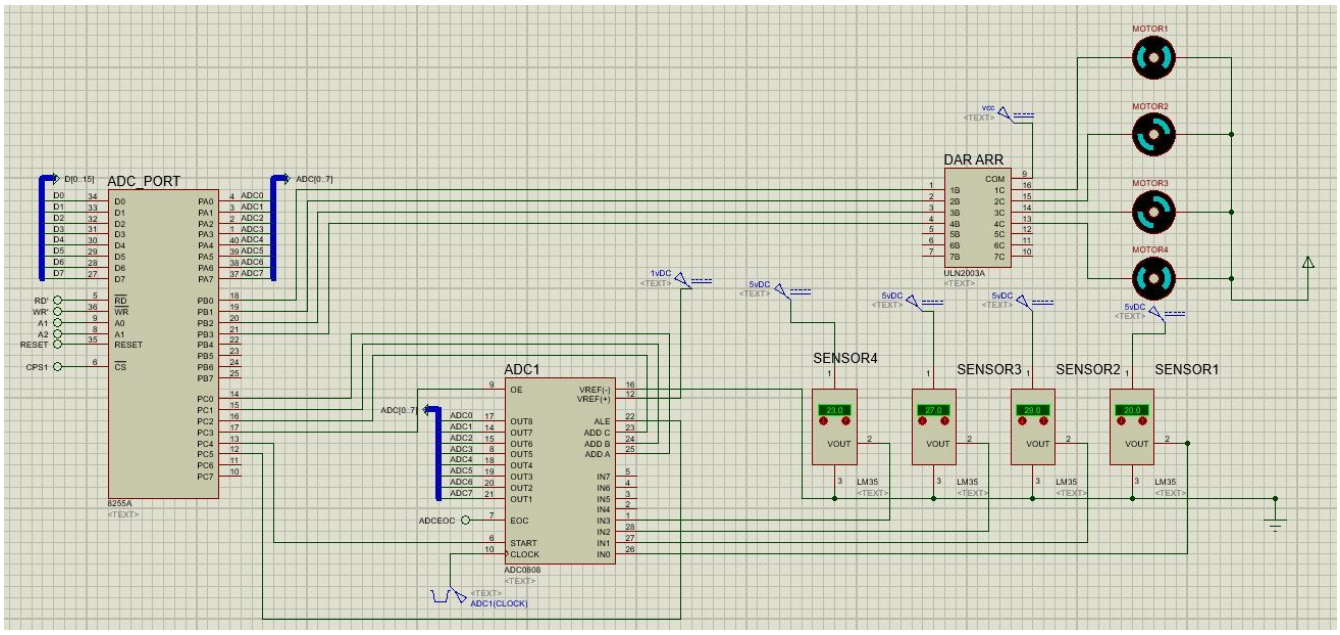
```
      CALL CLS
      CALL LCD_INIT
      MOV BL, 10
      MOV AL, TIMEDISPLAY
      MOV AH,00
      DIV BL
      ADD AL, '0' ;DISPLAY HUNDREDS OF TIME
      CALL DATWRIT ;ISSUE IT TO LCD
      CALL DELAY ;WAIT BEFORE ISSUING THE NEXT CHARACTER
      MOV AL, AH
      ADD AL, '0' ;DISPLAY TENS OF TIME
      CALL DATWRIT ;ISSUE IT TO LCD
      CALL DELAY ;WAIT BEFORE ISSUING THE NEXT CHARACTER
      MOV 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 NEAR
      PUSH AX
      MOV AL, 01 ;CLEAR LCD
      CALL COMNDWRT
      CALL DELAY
      POP AX
      RET
CLS ENDP

HLT           ; halt!
```

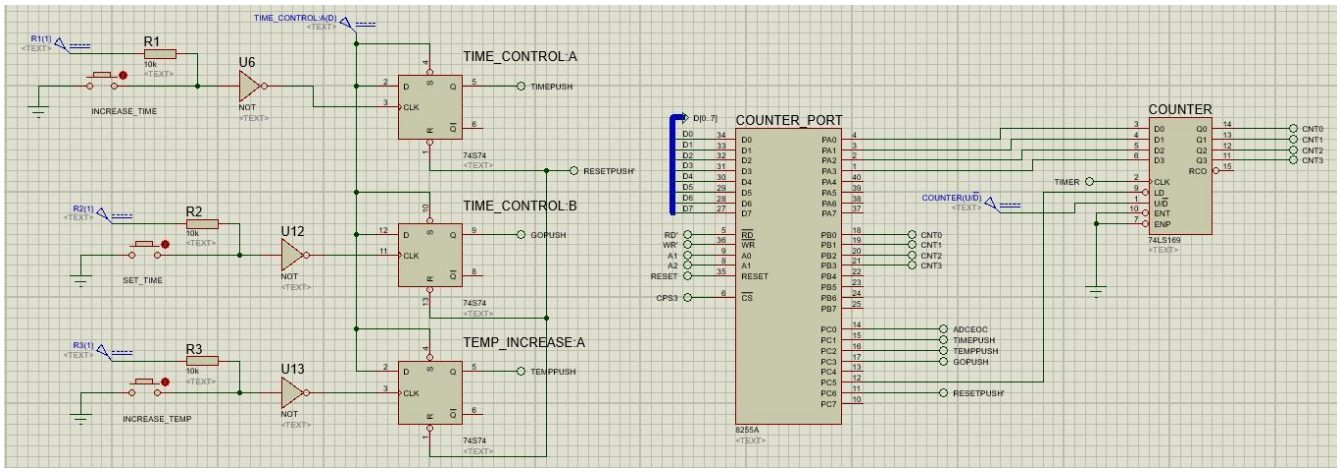
CIRCUIT DIAGRAMS

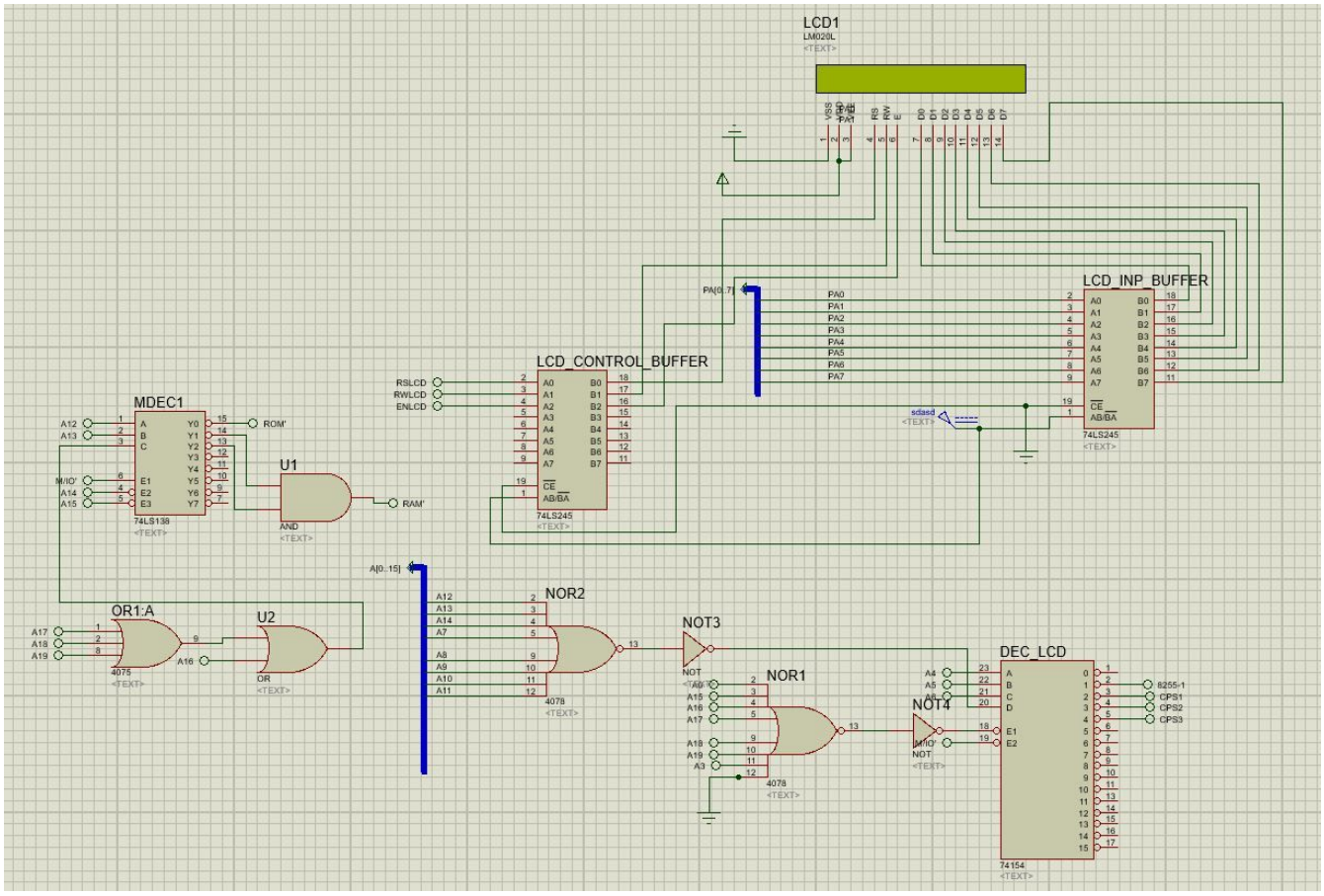



CLKG
MUP
8086
ADLATCH1
ADLATCH2
ADLATCH3
74LS373
SW1
SW-SPDT
RESET
OR7
OR6
U15
U16
U17
NOT
IOR'
IOW'
MRD'
MWR'
M/IO'
RD'
WR'
DBUF1
DBUF2
74LS245
DEN'
DT/R'


PORTS LCD
8255A
D[0..15]
PA[0..7]
RSLCD
RWLCD
ENLCD
IOR'
IOW'
A1
A2
RESET
8255-1
TIMER
8253A
TIMER(CLK0)
TIMER(GATE1)
U5
NOT
CPS2


MOTOR1
MOTOR2
MOTOR3
MOTOR4
vcc
DAR ARR
ULN2003A
ADC_PORT
8255A
ADC1
ADC0808
ADC1(CLOCK)
1vDC
5vDC
SENSOR4
SENSOR3
SENSOR2
SENSOR1
VOUT
LM35


TIME_CONTROL:A(D)
R1(1)
R1
10k
INCREASE_TIME
U6
NOT
TIME_CONTROL:A
TIMEPUSH
74S74
RESETPUSH'
R2(1)
R2
SET_TIME
U12
TIME_CONTROL:B
GOPUSH
R3(1)
R3
INCREASE_TEMP
U13
TEMP_INCREASE:A
TEMPPUSH
D[0..7]
COUNTER_PORT
RD'
WR'
A1
A2
RESET
CPS3
8255A
CNT0
CNT1
CNT2
CNT3
ADCEOC
COUNTER
TIMER
COUNTER(U/D')
74LS169


LCD1
LM020L
PA[0..7]
LCD_INP_BUFFER
LCD_CONTROL_BUFFER
RSLCD
RWLCD
ENLCD
74LS245
sdasd
MDEC1
74LS138
ROM'
U1
AND
RAM'
A[0..15]
OR1:A
4075
U2
OR
A16
NOR2
4078
NOT3
NOR1
NOT4
DEC_LCD
74154
8255-1
CPS1
CPS2
CPS3

REFERENCES

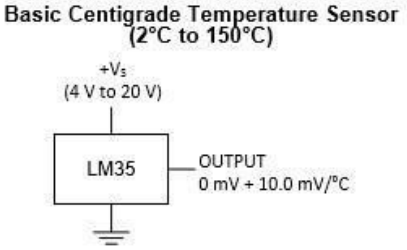
LM35 (Temperature sensor)

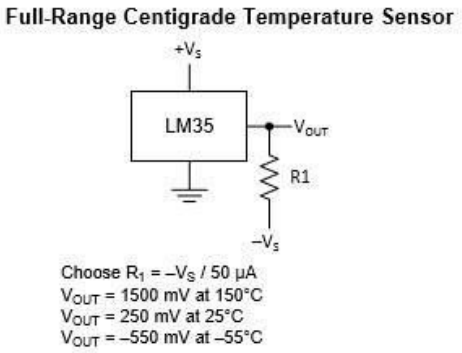
Range: -55 ºC to 150 ºC

Vin: 4V to 20V

For 0ºCelsius: OUTPUT = 0mV

Increment 10mV/ºC





ULN2003A (Darlington pair array)

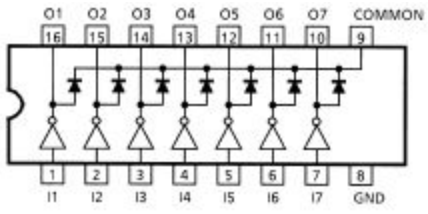
Vin = 30V

Vout = 50V

Source: Texas Instruments Datasheet

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

Internal Schematic



Pin-out