Title:- Alarm System using 8051 Microcontroller

Abstract:-

With the help of an 8051 microcontroller, this project aims to help detect and notify when the temperature reaches a certain value. In this case, the temperature to be detected is set as 100. On detection, the microcontroller will turn the LED, connected to one of its ports, on (or set to a high state). The microcontroller is connected to the sensor via an analog-to-digital converter (ADC), which helps convert the value from analog form to digital form.

Methodology:

Required Components:

- 8051 Microcontroller(P89V51RD2)

- ADC 0804

- LED

- LM35 Temperature Sensor

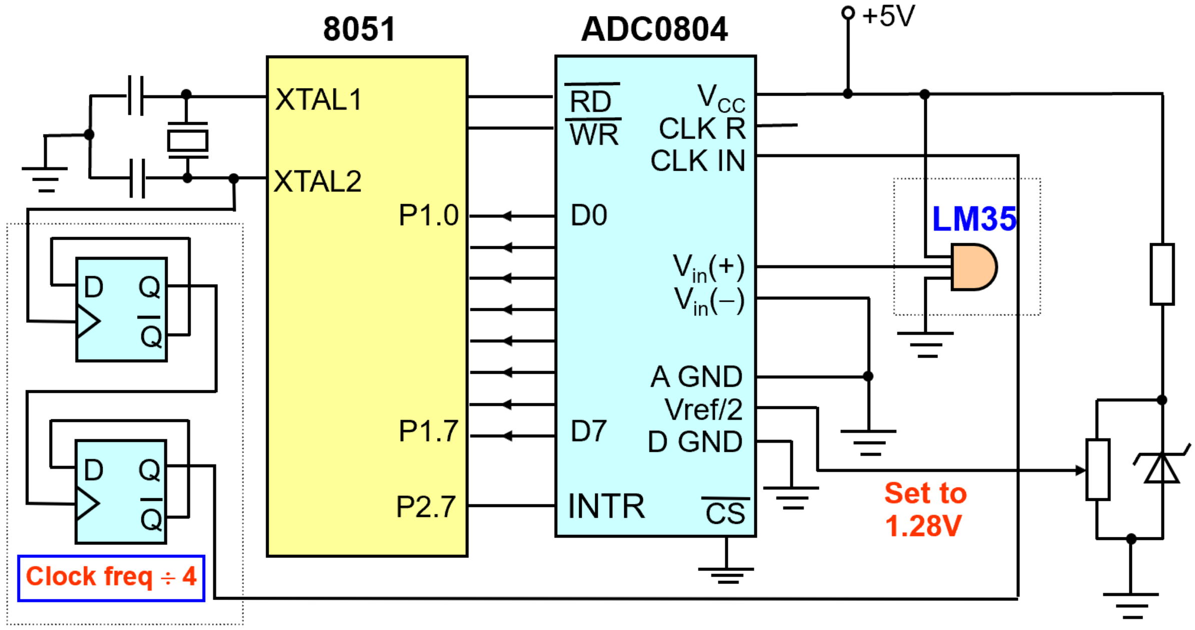
Software Used:-

- Kiel uVision

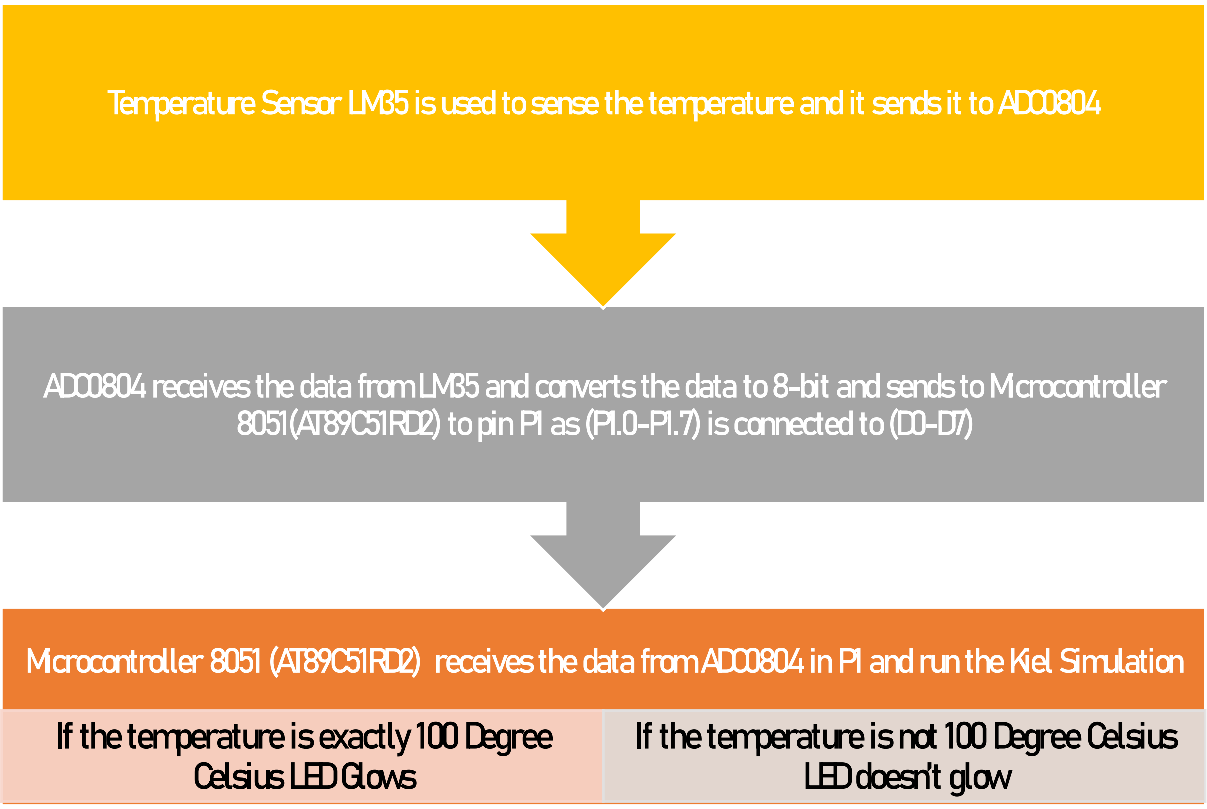
- Proteus

The LED is connected to the microcontroller via port pin P3.0. The ADC is connected to the microcontroller via port P2 and P1. The temperature data is continuously sent as input to the ADC. The ADC is further connected to the microcontroller. 8051 constantly compares the value of the temperature to the required value, and turns on the LED (set to high state) when the temperature reaches the desired value. A delay is called to ensure the status of the LED is visible to the user. The LED is off (or set to low state) during the remaining time. This processes is run infinitely. The assembly level code is written on Keil MicroVision, converted to a hex file and further implemented on the software, Proteus, for a better representation.

• Block diagram (Hardware)



- **Flow chart (software)**



Temperature Sensor LM35 is used to sense the temperature and it sends it to ADC0804

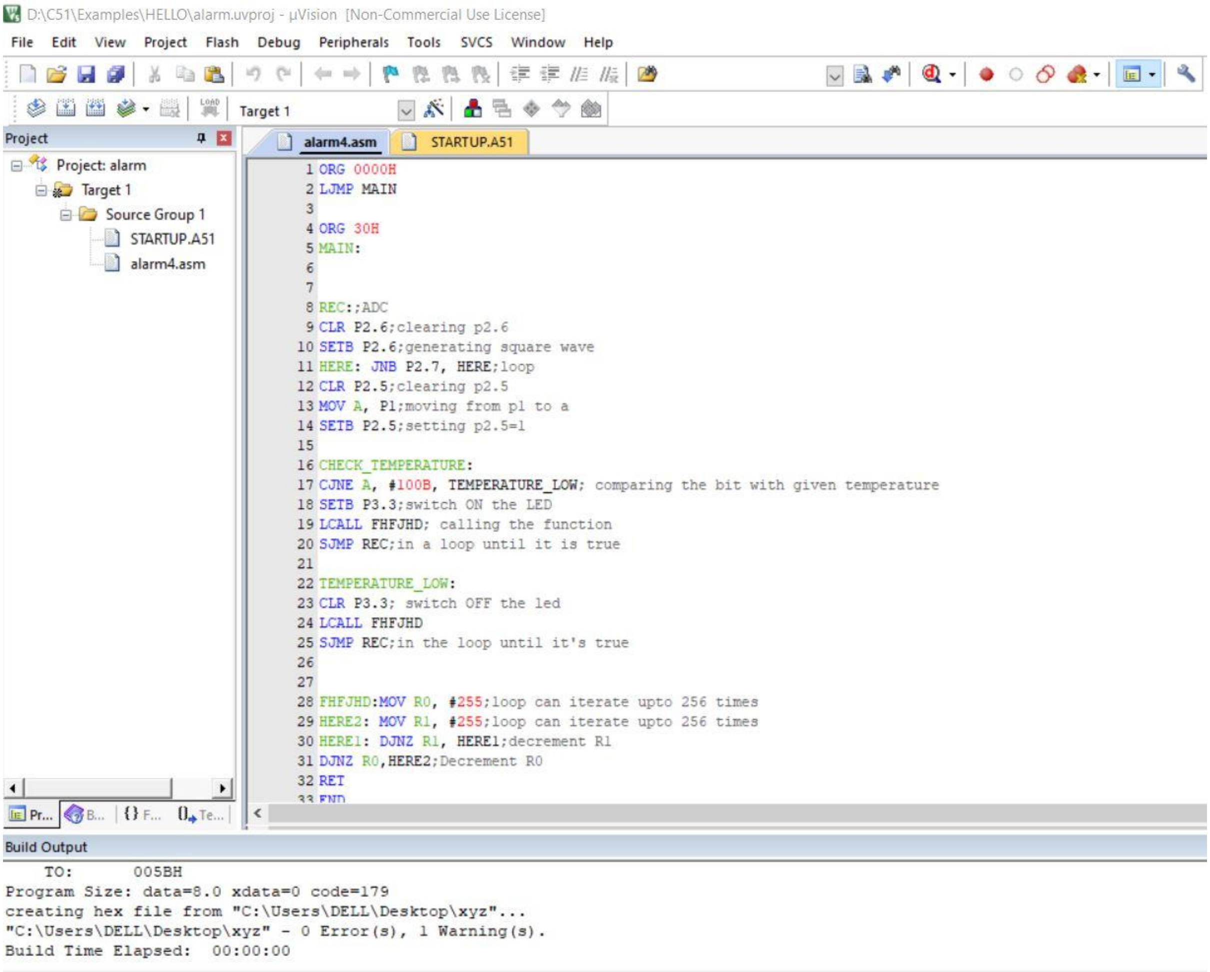
ADC0804 receives the data from LM35 and converts the data to 8-bit and sends to Microcontroller 8051(AT89C51RD2) to pin P1 as (P1.0-P1.7) is connected to (D0-D7)

Microcontroller 8051 (AT89C51RD2) receives the data from ADC0804 in P1 and run the Kiel Simulation

If the temperature is exactly 100 Degree Celsius LED Glows

If the temperature is not 100 Degree Celsius LED doesn't glow

Program with comments

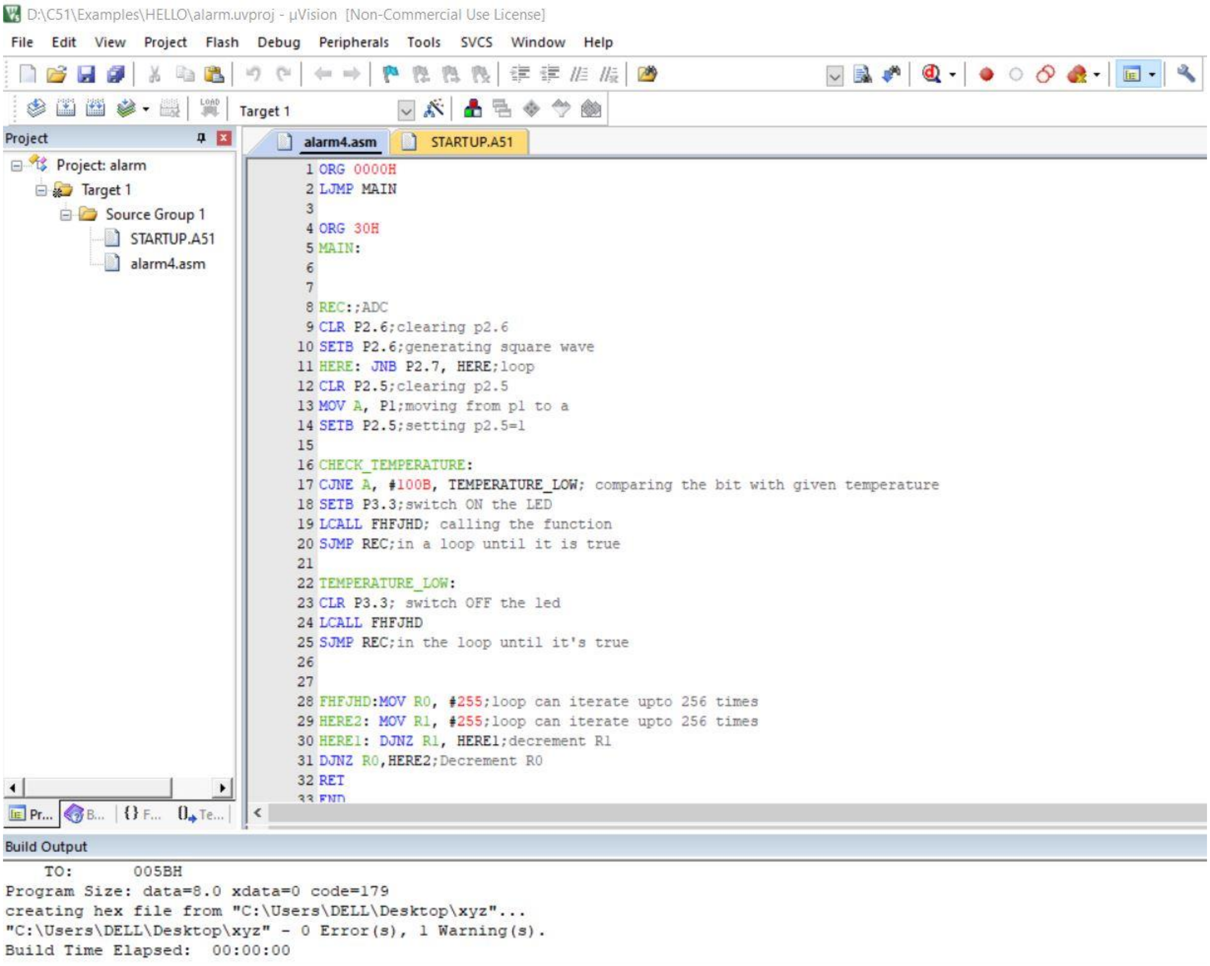

D:\C51\Examples\HELLO\alarm.uvproj - µVision [Non-Commercial Use License]
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Target 1
Project
Project: alarm
Target 1
Source Group 1
STARTUP.A51
alarm4.asm
alarm4.asm
STARTUP.A51
Build Output

```
ORG 0000H
LJMP MAIN

ORG 30H
MAIN:


REC:;ADC
CLR P2.6;clearing p2.6
SETB P2.6;generating square wave
HERE: JNB P2.7, HERE;loop
CLR P2.5;clearing p2.5
MOV A, P1;moving from p1 to a
SETB P2.5;setting p2.5=1

CHECK_TEMPERATURE:
CJNE A, #100B, TEMPERATURE_LOW; comparing the bit with given temperature
SETB P3.3;switch ON the LED
LCALL FHFJHD; calling the function
SJMP REC;in a loop until it is true

TEMPERATURE_LOW:
CLR P3.3; switch OFF the led
LCALL FHFJHD
SJMP REC;in the loop until it's true


FHFJHD:MOV R0, #255;loop can iterate upto 256 times
HERE2: MOV R1, #255;loop can iterate upto 256 times
HERE1: DJNZ R1, HERE1;decrement R1
DJNZ R0,HERE2;Decrement R0
RET
END
```

```
    TO:       005BH
Program Size: data=8.0 xdata=0 code=179
creating hex file from "C:\Users\DELL\Desktop\xyz"...
"C:\Users\DELL\Desktop\xyz" - 0 Error(s), 1 Warning(s).
Build Time Elapsed:  00:00:00
```

Results

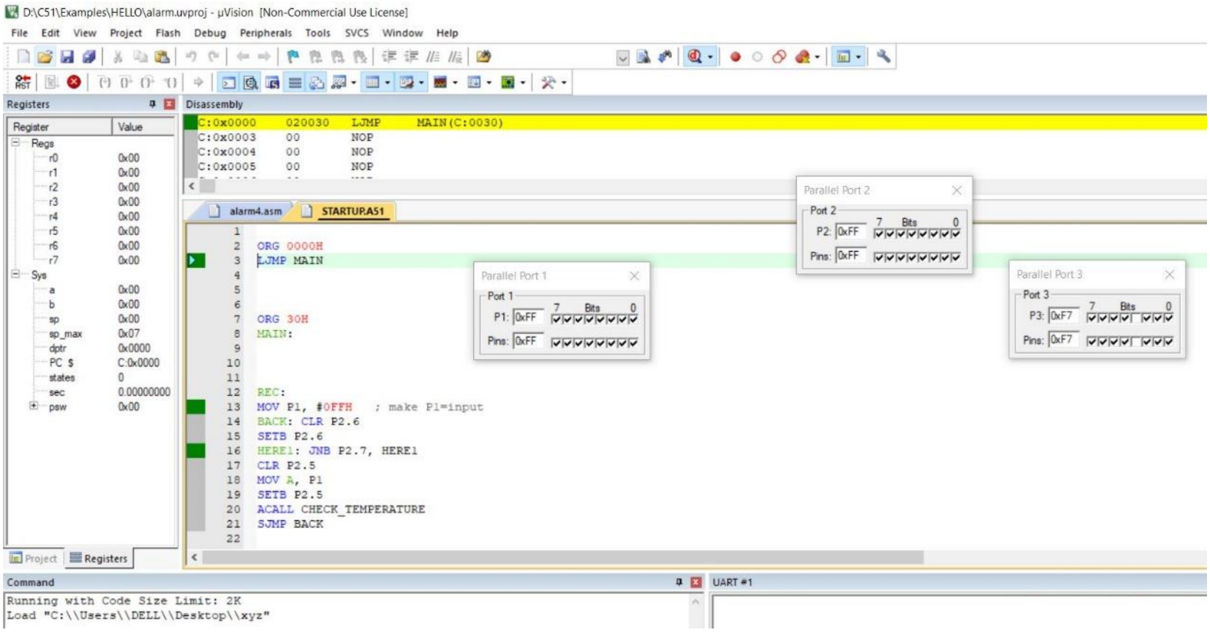
- **Screen shot showing program with zero syntax error(KEIL)**


D:\C51\Examples\HELLO\alarm.uvproj - µVision [Non-Commercial Use License]
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Target 1
Project
Project: alarm
Target 1
Source Group 1
STARTUP.A51
alarm4.asm
alarm4.asm
STARTUP.A51
Pr... B... F... Te...
Build Output

```
ORG 0000H
LJMP MAIN

ORG 30H
MAIN:


REC:;ADC
CLR P2.6;clearing p2.6
SETB P2.6;generating square wave
HERE: JNB P2.7, HERE;loop
CLR P2.5;clearing p2.5
MOV A, P1;moving from p1 to a
SETB P2.5;setting p2.5=1

CHECK_TEMPERATURE:
CJNE A, #100B, TEMPERATURE_LOW; comparing the bit with given temperature
SETB P3.3;switch ON the LED
LCALL FHFJHD; calling the function
SJMP REC;in a loop until it is true

TEMPERATURE_LOW:
CLR P3.3; switch OFF the led
LCALL FHFJHD
SJMP REC;in the loop until it's true


FHFJHD:MOV R0, #255;loop can iterate upto 256 times
HERE2: MOV R1, #255;loop can iterate upto 256 times
HERE1: DJNZ R1, HERE1;decrement R1
DJNZ R0,HERE2;Decrement R0
RET
END
```

```
    TO:        005BH
Program Size: data=8.0 xdata=0 code=179
creating hex file from "C:\Users\DELL\Desktop\xyz"...
"C:\Users\DELL\Desktop\xyz" - 0 Error(s), 1 Warning(s).
Build Time Elapsed:  00:00:00
```

- **Screen shot of the result (KEIL)**



D:\C51\Examples\HELLO\alarm.uvproj - µVision [Non-Commercial Use License]

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x00
sp_max	0x07
dptr	0x0000
PC $	C:0x0000
states	0
sec	0.00000000
psw	0x00

```
C:0x0000    020030    LJMP     MAIN(C:0030)
C:0x0003    00        NOP
C:0x0004    00        NOP
C:0x0005    00        NOP
```

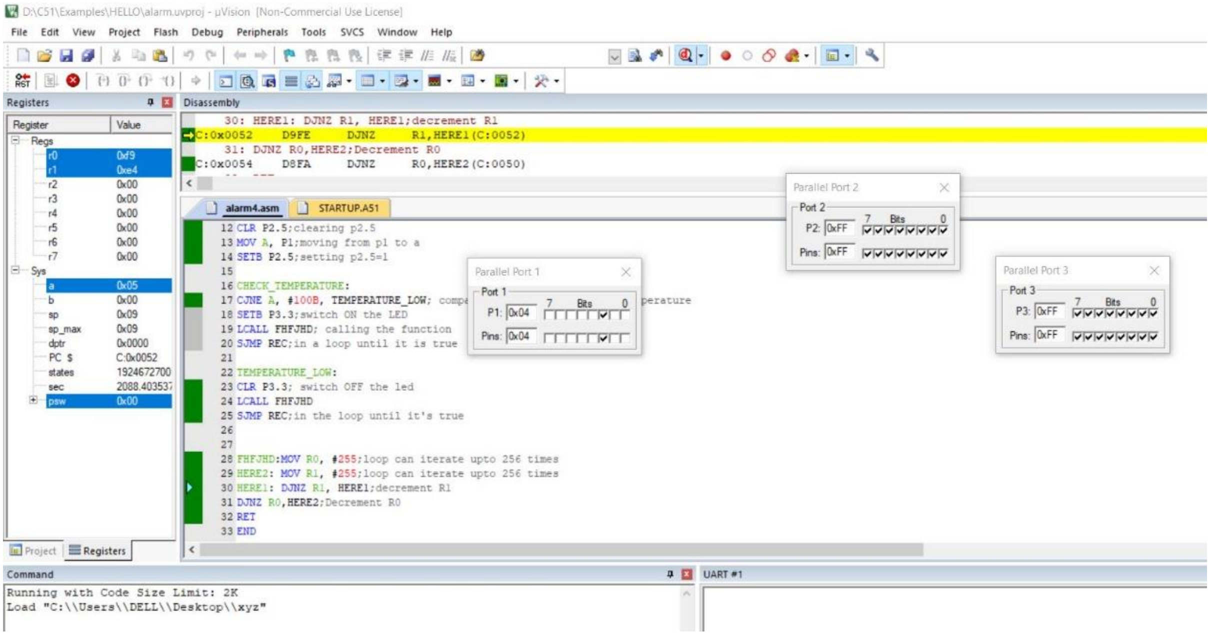
```
ORG 0000H
LJMP MAIN



ORG 30H
MAIN:



REC:
MOV P1, #0FFH   ; make P1=input
BACK: CLR P2.6
SETB P2.6
HERE1: JNB P2.7, HERE1
CLR P2.5
MOV A, P1
SETB P2.5
ACALL CHECK_TEMPERATURE
SJMP BACK
```

Parallel Port 1 — Port 1 — P1: 0xFF, Bits 7 ☑☑☑☑☑☑☑☑ 0; Pins: 0xFF ☑☑☑☑☑☑☑☑

Parallel Port 2 — Port 2 — P2: 0xFF, Bits 7 ☑☑☑☑☑☑☑☑ 0; Pins: 0xFF ☑☑☑☑☑☑☑☑

Parallel Port 3 — Port 3 — P3: 0xF7, Bits 7 ☑☑☑☑□☑☑☑ 0; Pins: 0xF7 ☑☑☑☑□☑☑☑

```
Running with Code Size Limit: 2K
Load "C:\\Users\\DELL\\Desktop\\xyz"
```



D:\C51\Examples\HELLO\alarm.uvproj - µVision [Non-Commercial Use License]

Register	Value
Regs	
r0	0xf9
r1	0xe4
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x05
b	0x00
sp	0x09
sp_max	0x09
dptr	0x0000
PC $	C:0x0052
states	1924672700
sec	2088.403537
psw	0x00

```
    30: HERE1: DJNZ R1, HERE1;decrement R1
C:0x0052    D9FE      DJNZ     R1,HERE1(C:0052)
    31: DJNZ R0,HERE2;Decrement R0
C:0x0054    D8FA      DJNZ     R0,HERE2(C:0050)
```

```
CLR P2.5;clearing p2.5
MOV A, P1;moving from p1 to a
SETB P2.5;setting p2.5=1

CHECK_TEMPERATURE:
CJNE A, #100B, TEMPERATURE_LOW; compa                              perature
SETB P3.3;switch ON the LED
LCALL FHFJHD; calling the function
SJMP REC;in a loop until it is true

TEMPERATURE_LOW:
CLR P3.3; switch OFF the led
LCALL FHFJHD
SJMP REC;in the loop until it's true


FHFJHD:MOV R0, #255;loop can iterate upto 256 times
HERE2: MOV R1, #255;loop can iterate upto 256 times
HERE1: DJNZ R1, HERE1;decrement R1
DJNZ R0,HERE2;Decrement R0
RET
END
```

Parallel Port 1 — Port 1 — P1: 0x04, Bits 7 □□□□□☑□□ 0; Pins: 0x04 □□□□□☑□□

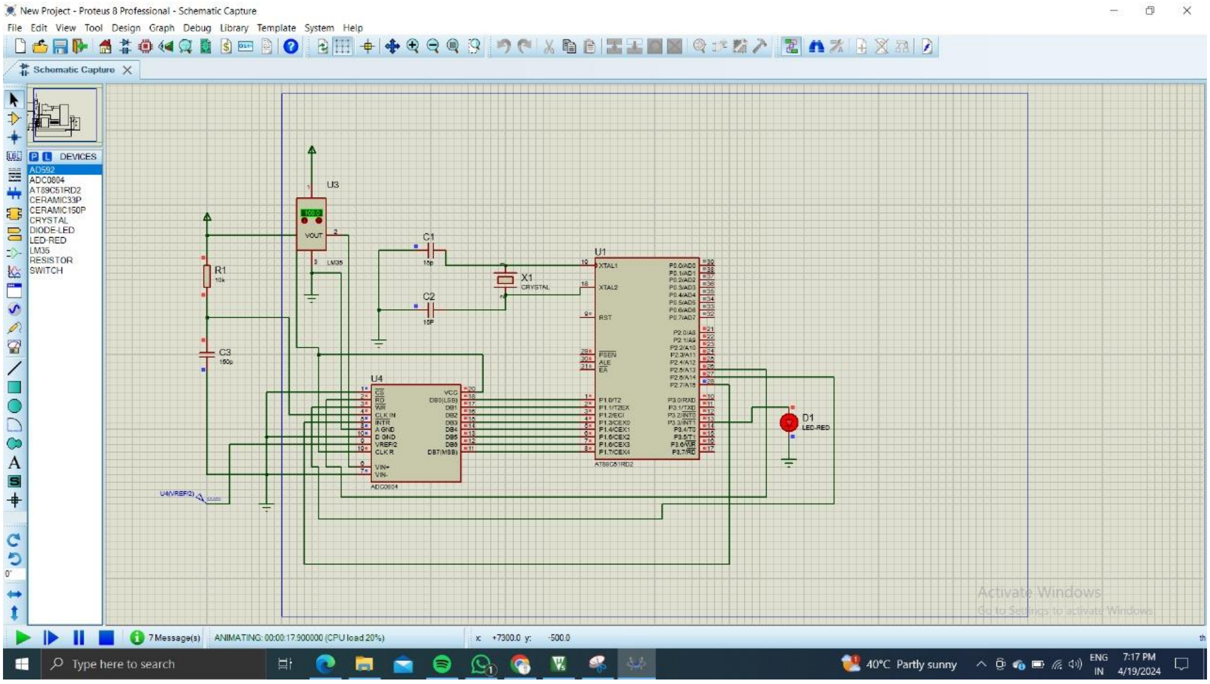
Parallel Port 2 — Port 2 — P2: 0xFF, Bits 7 ☑☑☑☑☑☑☑☑ 0; Pins: 0xFF ☑☑☑☑☑☑☑☑

Parallel Port 3 — Port 3 — P3: 0xFF, Bits 7 ☑☑☑☑☑☑☑☑ 0; Pins: 0xFF ☑☑☑☑☑☑☑☑

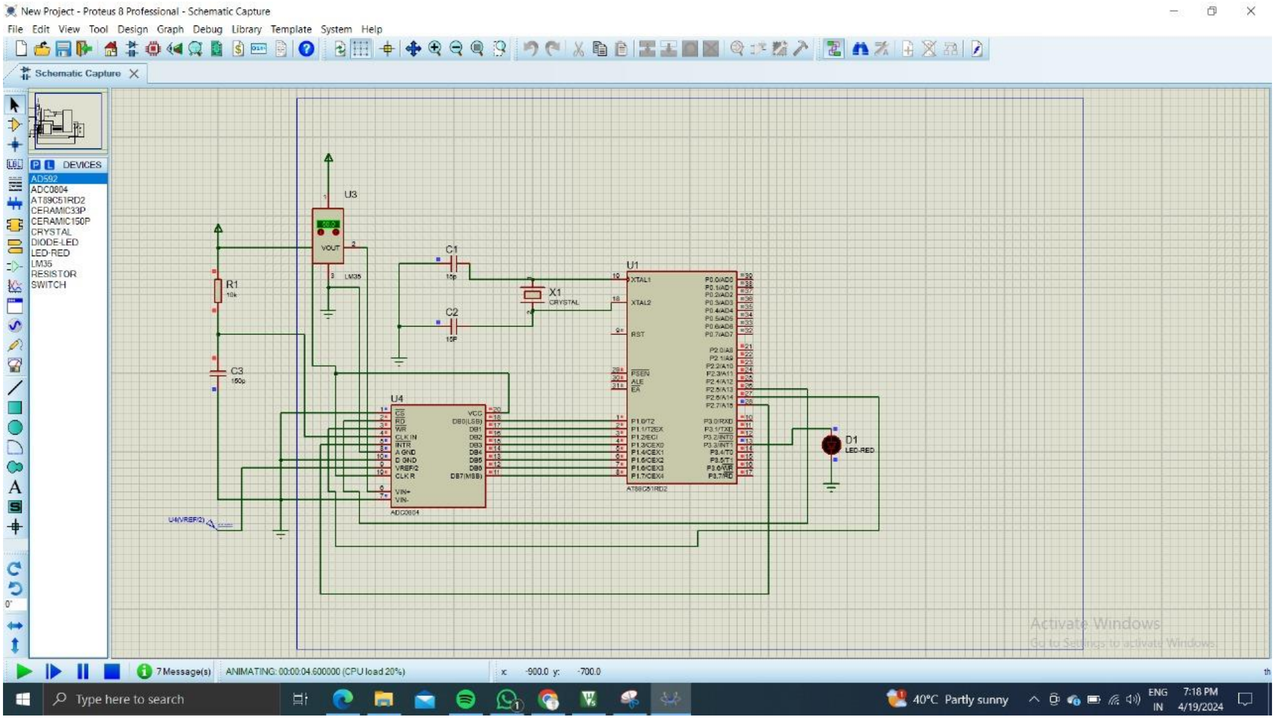
```
Running with Code Size Limit: 2K
Load "C:\\Users\\DELL\\Desktop\\xyz"
```

• Screen shot of Proteus simulation

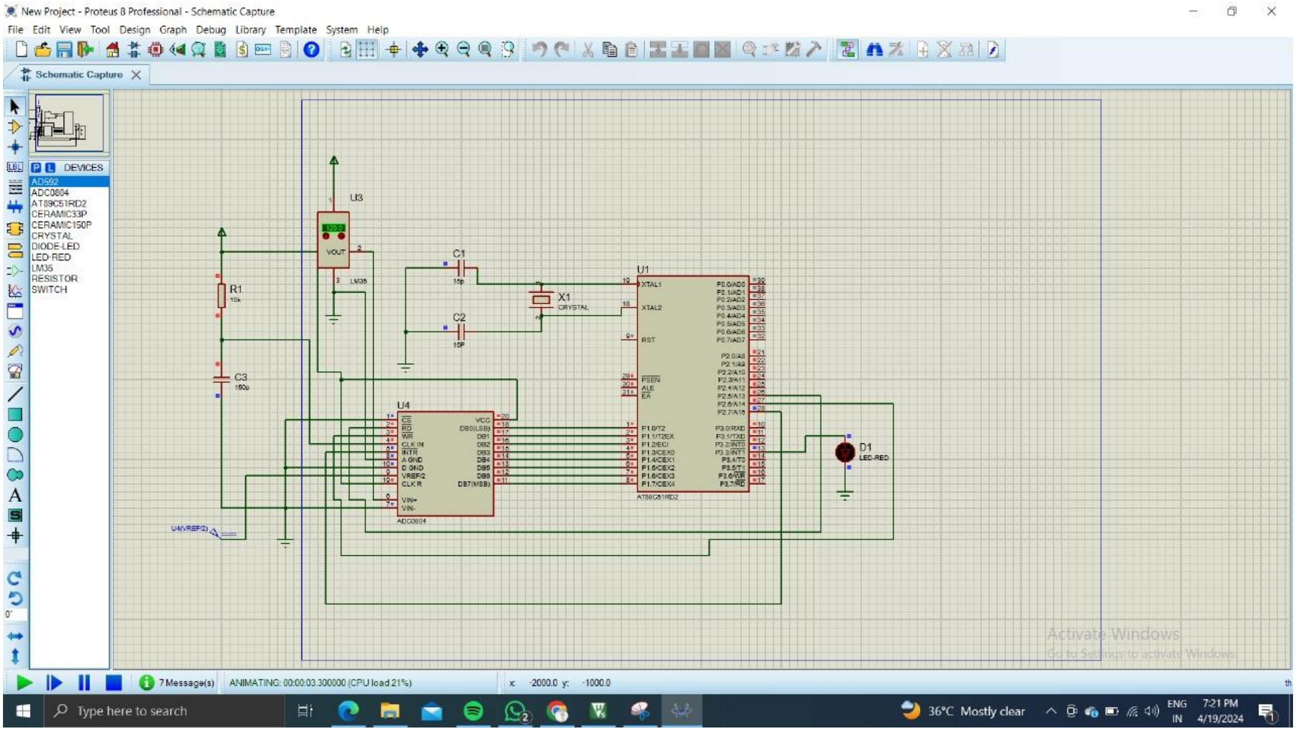
Temperature= 100^0 C



Temperature= 60^0 C



Temperature = 120^0 C



New Project - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
DEVICES
AD592
ADC0804
AT89C51RD2
CERAMIC33P
CERAMIC150P
CRYSTAL
DIODE-LED
LED-RED
LM35
RESISTOR
SWITCH
U3
120.0
VOUT
LM35
R1
10k
C3
150p
C1
15p
C2
15P
X1
CRYSTAL
U1
XTAL1
XTAL2
RST
PSEN
ALE
EA
P0.0/AD0
P0.1/AD1
P0.2/AD2
P0.3/AD3
P0.4/AD4
P0.5/AD5
P0.6/AD6
P0.7/AD7
P2.0/A8
P2.1/A9
P2.2/A10
P2.3/A11
P2.4/A12
P2.5/A13
P2.6/A14
P2.7/A15
P1.0/T2
P1.1/T2EX
P1.2/ECI
P1.3/CEX0
P1.4/CEX1
P1.5/CEX2
P1.6/CEX3
P1.7/CEX4
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
AT89C51RD2
U4
CS
RD
WR
CLK IN
INTR
A GND
D GND
VREF/2
CLK R
VIN+
VIN-
VCC
DB0(LSB)
DB1
DB2
DB3
DB4
DB5
DB6
DB7(MSB)
ADC0804
U4(VREF/2)
D1
LED-RED
7 Message(s)
ANIMATING: 00:00:03.300000 (CPU load 21%)
x -2000.0 y: -1000.0
Activate Windows
Go to Settings to activate Windows.
Type here to search
36°C Mostly clear
ENG
IN
7:21 PM
4/19/2024