1. Blinking of LED using 8051 microcontroller Using Proteus.

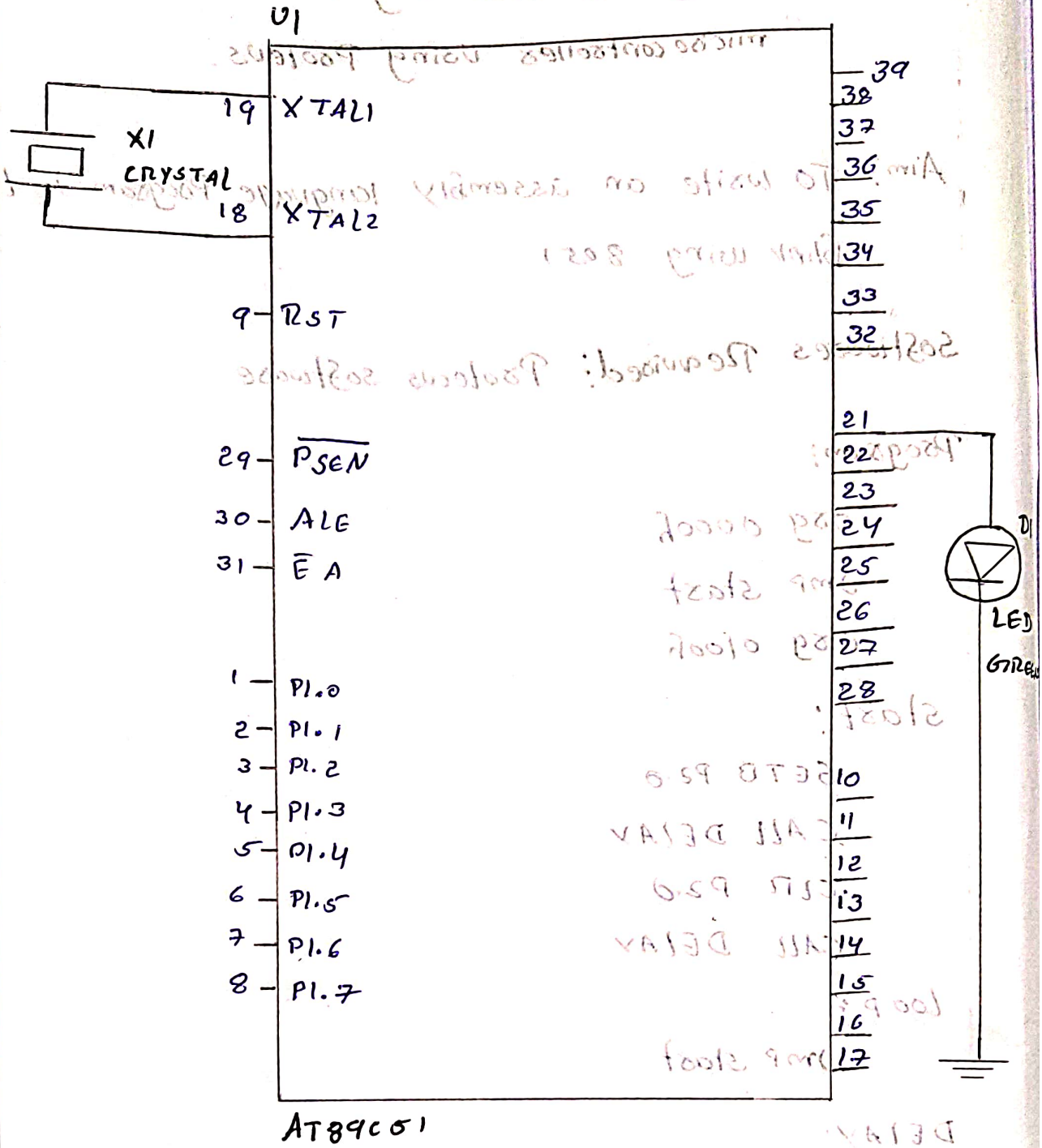
Aim: To write an assembly language Program to LED blink using 8051

Softwares Required: Proteus software.

Program:

```
    org 0000h
    jmp start
    org 0100h
start:
    SETB P2.0
    CALL DELAY
    CLR P2.0
    CALL DELAY
loop:
    jmp start

DELAY:
    mov R1, #255
L2:
    mov R2, #255
L1:
    DJNZ R2, L1
    DJNZ R1, L2
    RET
    END.
```



Procedure

1. Create a New Project

   Open Protecus, go to Files New Project name list it, and

   Open the Sematic editor

2. Add components.

Click P and search for:

* ATC89C51, LED, Crystal Oscillator, Power (VCC, GND)
* Drag and Place them on the workspace.

3. Connect the circuit.
4. Load the Program.
5. Run simulation.

Result:

Thus the Program has been successfully verified and executed.

2. LED switching using 8051 using Proteus.

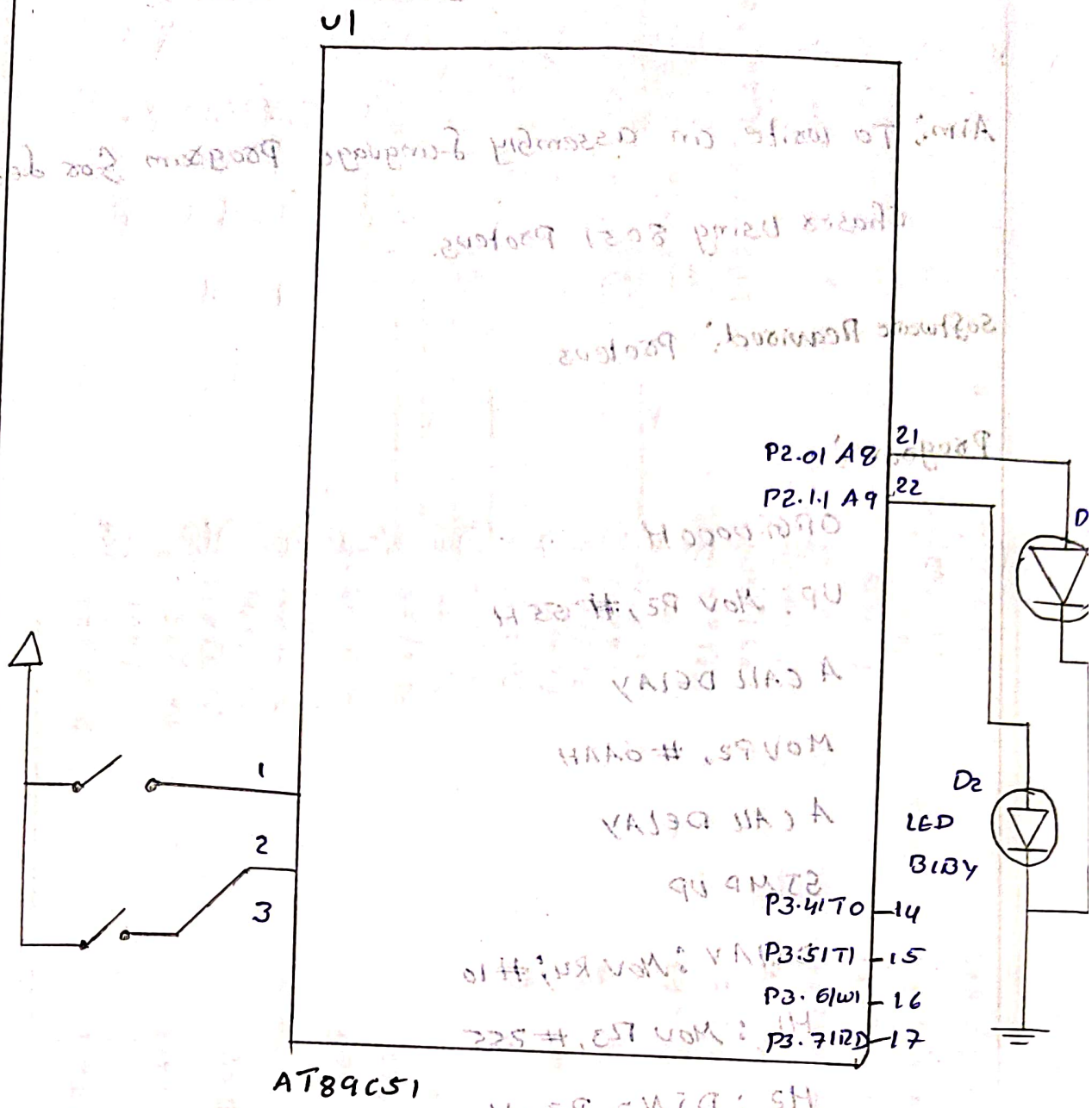
Aim: To write an assembly Language program to led switching using 8051.

Software Required: Proteus Required.

Program:

```
ORG 0000H
UP : MOV P2, #55H
ACALL DELAY
MOV P2, #0AAH
ACALL DELAY
SJMP UP

DELAY : MOV R4, #10
H1 : MOV R3, #255
H2 : DJNZ R3, H2
DJNZ R4, H1
RET
END.
```



Result.

Thus the program has been successfully verified and executed.

3. LED chaser using 8051 Proteus.

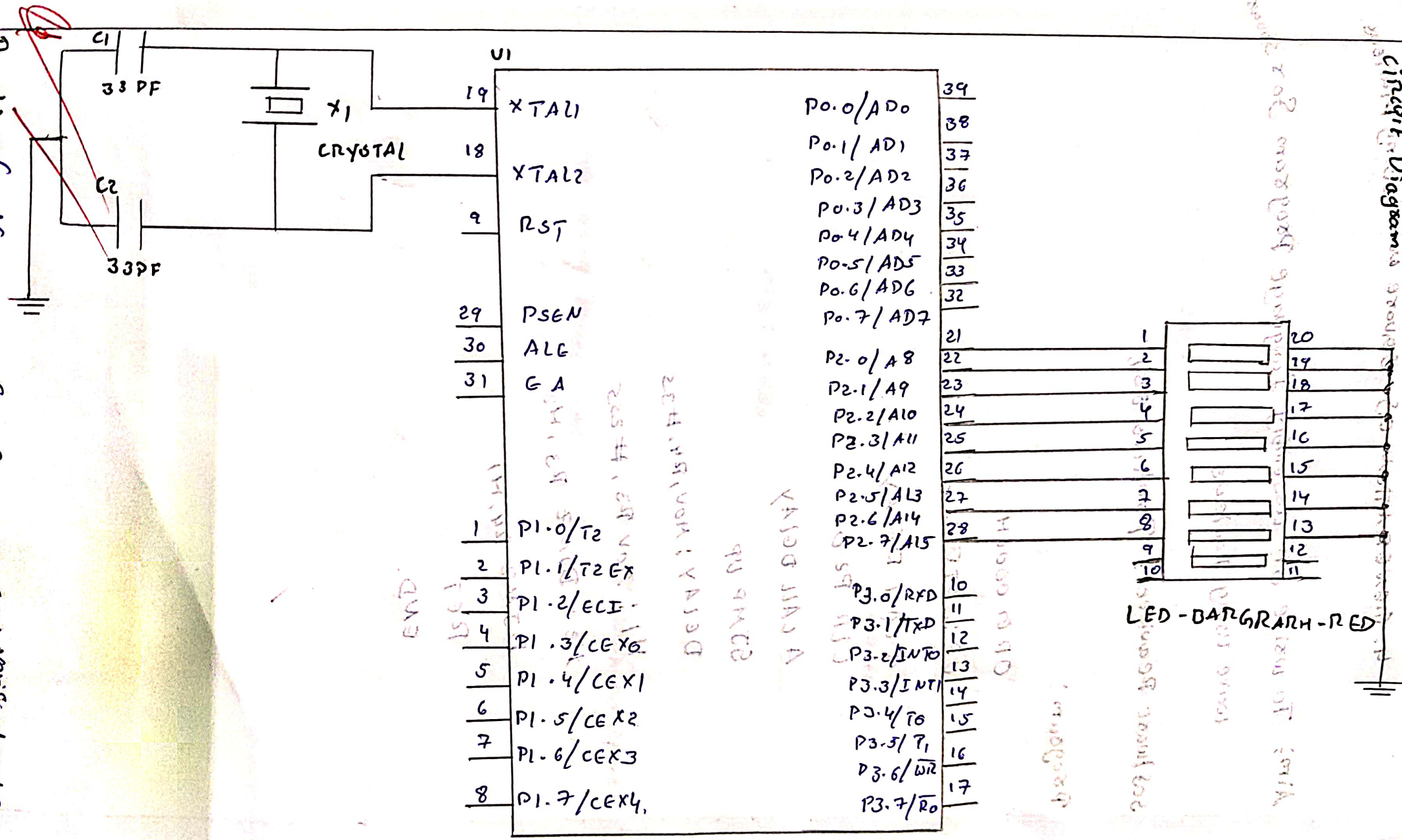
Aim: To write an assembly language program for Led Chaser using 8051 Proteus.

Software Reavised: Proteus.

Program:

```
        ORG 0000H
UP:     MOV P2, #55H
        ACALL DELAY
        MOV P2, #0AAH
        ACALL DELAY
        SJMP UP
DELAY:  MOV R4; #10
H1:     MOV R3, #255
H2:     DJNZ R3, H2
        DJNZ R4, H1
        RET
        END
```

Circuit Diagram



Result: Thus the program has been successfully verified and executed.

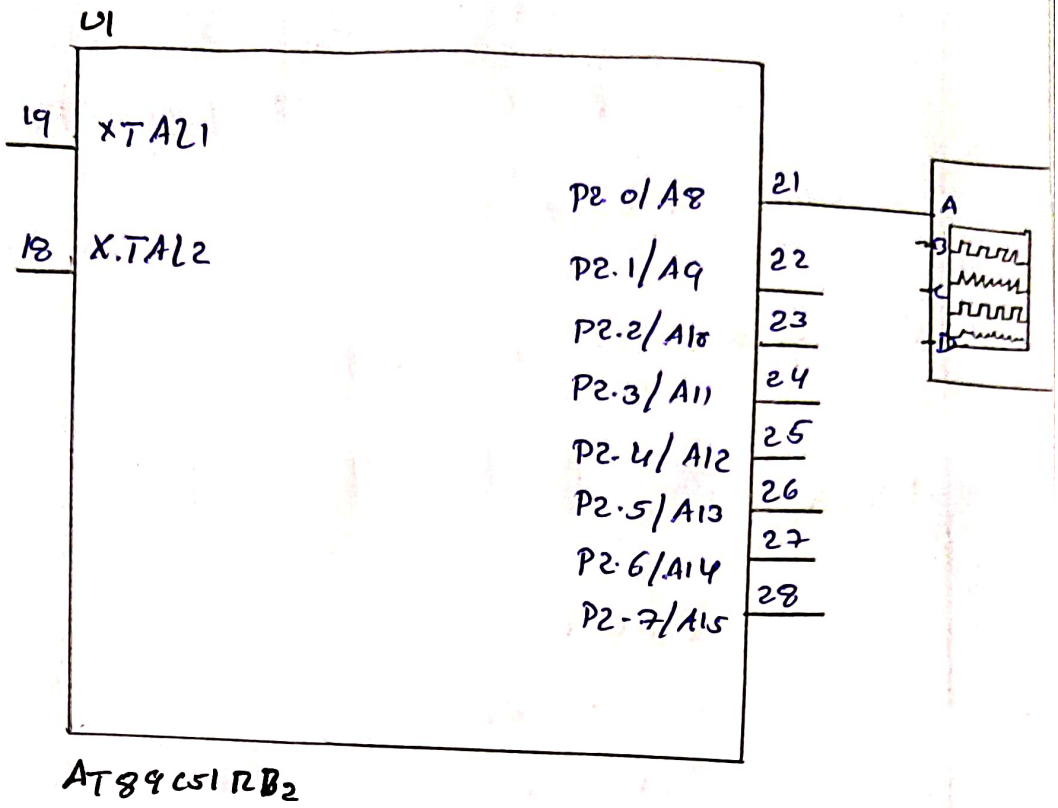
4. GENERATION OF SQUARE WAVE USING PROTEUS.

Aim: To write an assembly Language Program for square wave using Proteus.

Software Required: Proteus 8051.

Program:

```
ORG 0000H
UP: SETB P2.0
ACALL DELAY
CLR P2.0
ACALL DELAY
SJMP UP
DELAY: MOV R4, #35
H1: MOV R3, #255
H2: DJNZ R3, H2
DJNZ R4, H1
RET
END.
```



Result:

Thus, the code is executed successfully with no errors for sawtooth wave using Proteus 8051.

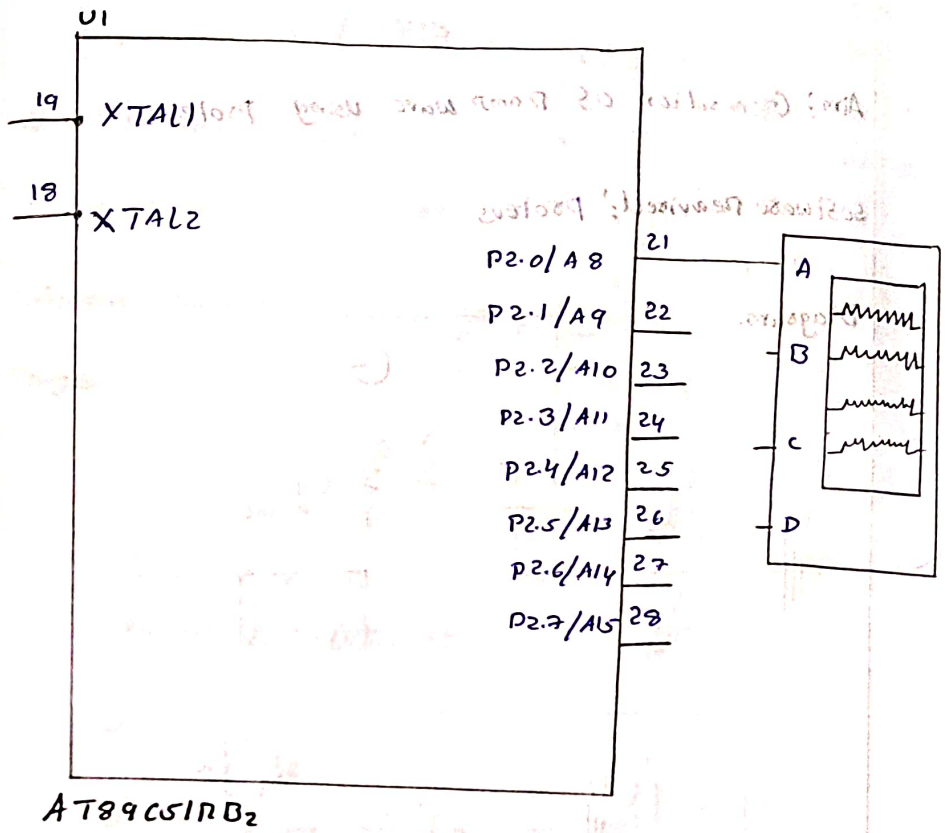
5. Generation of Triangular Wave using PROTEUS.

Aim: To write an assembly language program for square wave using Proteus.

Software required: Proteus 8051

Program.

```
ORG 00H
MOV P1,#00H
MOV A, #00H
MOV R0, 00H
INC A
MOV P1, A
A CALL DELAY
CJNE A, #0EFH
DEC A
MOV P1, A
A CALL DELAY
CJNE A, #00FFH UPWARD
END.
```



Result:

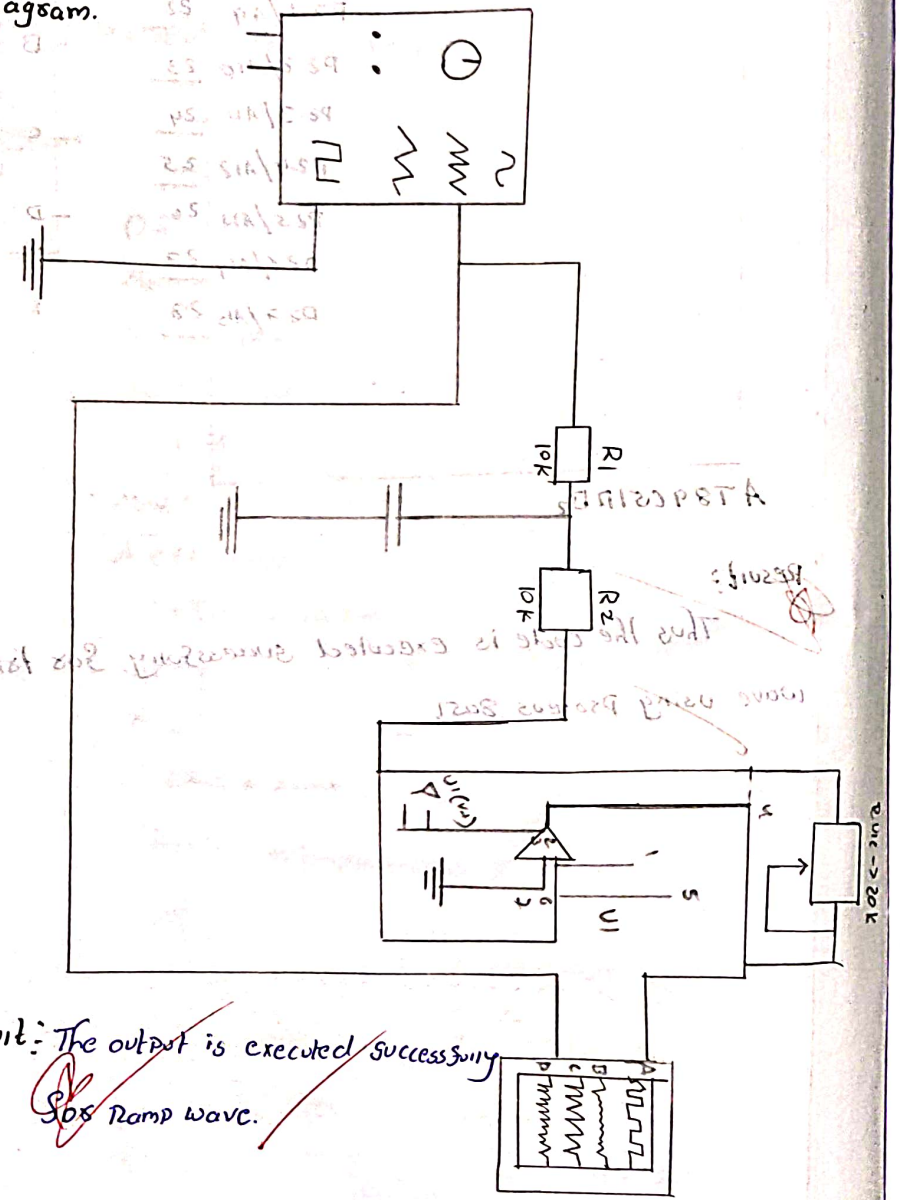
Thus the code is executed successfully for triangular wave using Proteus 8051

6. Generation of Ramp wave using PROTEUS

Aim: Generation of Ramp wave using Proteus.

Software Reavised: Proteus

Diagram.



Result: The output is executed successfully for Ramp wave.

7. Fade In Fade out of LED using 8051, Proteus.

Aim: To write an assembly Language Program for Fade in Fade out using LED.

Software Reavised: Arduino uno & proteus

Program:

```
Int LEDS [] = {13,12,11};
  Void setup () {
for (int i = 0; i<5; i+=1) {
Pinmode (LEDS [i], output);
}
}
Void loop (){
for (int i=0; i<6; i+=1){
digital write (13, High);
delay (100);
digital write (13, low);
delay (100);
}
for (int i=0; i<=3; i+=1){
digitial write (12, HIGH);
delay (100);
digital write (11, low);
delay (100);
}
for (int i=0; i<=1; i+=1) {
```

delay(100)

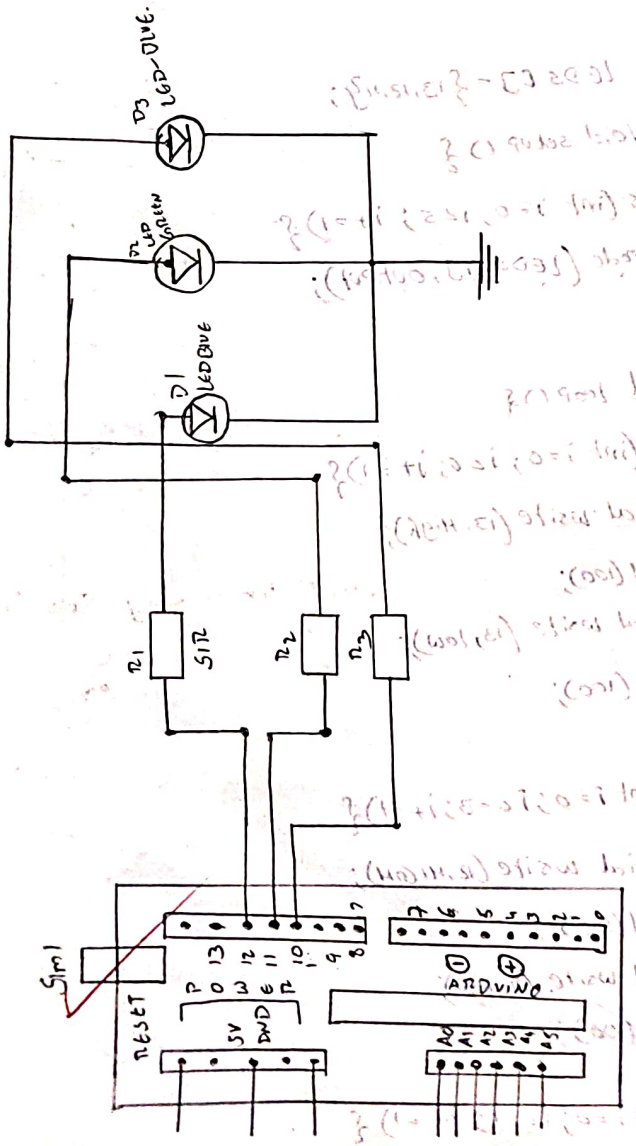
digital write(11, Low)

Delay (100);

}

}

Circuit Diagram



Procedure:

* Place an LED & Resistors on the sereies
* Connect the resistor with Arduino Uno & Proteus workspace
* Connect the other END of LED To Ground (GND).
* Connect the 5V & GND pins to Arduino uno.
* Click on run simulation and observe the led brightness gradually increasing & Decreasing.

Result:

Thus, the Program is executed successfully by fade in and fade out of lights.

8.

8. Clockwise Rotation of stepper motor using 8051 using Proteus.

Aim: To write an assembly Language Program for clockwise Rotation of stepper motor

Software required: Proteus 8.5.01.

Procedure:

* Create a new project, open Proteus, goto files new project, name it, and open the schematic editor.
* Add components to the schematic editor Now click P and
  - AT89C51, Battery, Button, motor Steeper and ULN 2003A
* Connect the stepper motor to the battery and stepper motor to the ULN2003A
* connect the button to the pin P3.1 INT0.

Input:

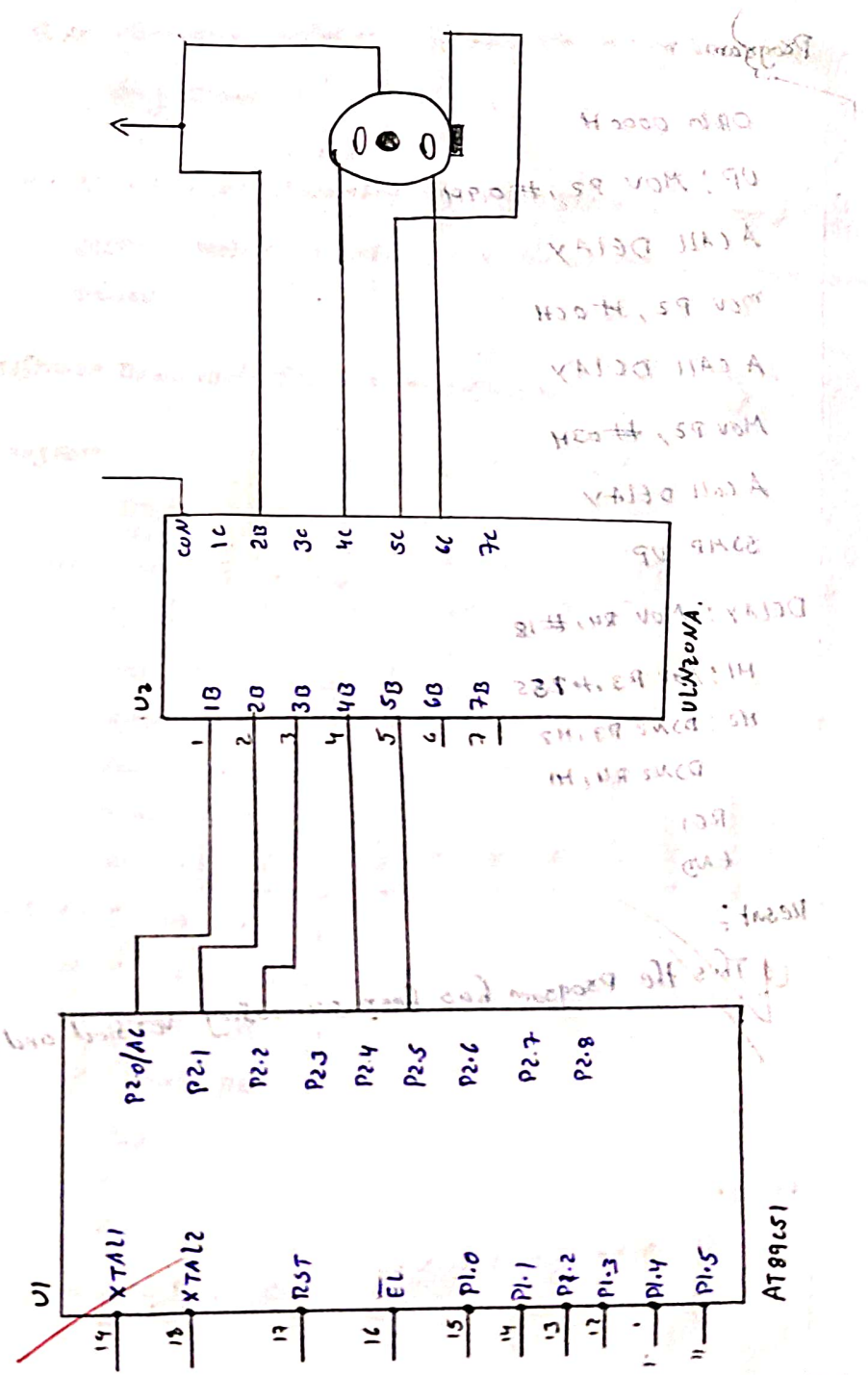
ULN2003A - P2.0, P2.1, P2.2, P2.3

Stepper motor - 1C, 15, 14, 13

Button - P3.0.

Output:

Stepper motor connected to the battery



Program

```
ORG 0000H
UP: MOV P2, #09H
ACALL DELAY
MOV P2, #0CH
ACALL DELAY
MOV P2, #03H
ACall DELAY
SJMP UP

DELAY: MOV R4, #18
H1: MOV R3, #255
H2: DJNZ R3, H2
    DJNZ R4, H1
    RET
    END.
```

Result:

Thus the program has been successfully verified and exec

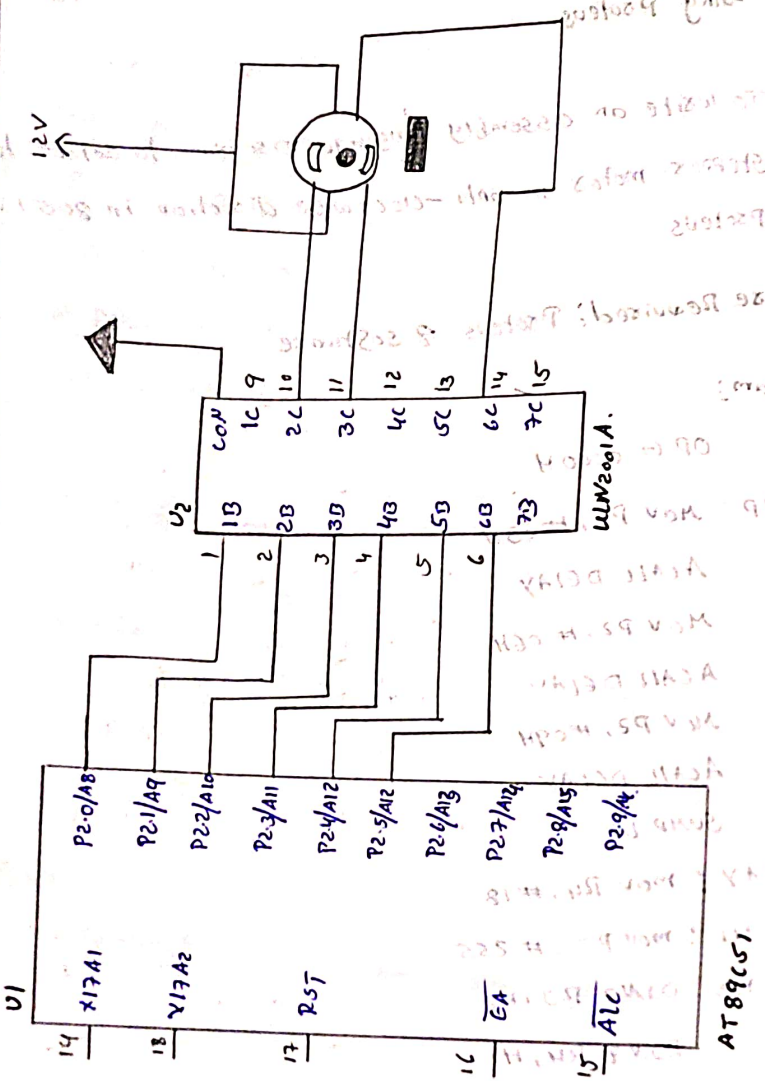
9. Anticlock wise Rotation of stepper motor using 8051 using Proteus.

Aim: To write an assembly language program to rotate the stepper motor in anti-clockwise direction in 8051 using Proteus.

Software Required: Proteus & software

Program:

```
        ORG 0000H
UP:     MOV P2, #03H
        ACALL DELAY
        MOV P2, #06H
        ACALL DELAY
        MOV P2, #09H
        ACALL DELAY
        SJMP UP
DELAY:  MOV R4, #18
H1:     MOV R3, #255
H2:     DJNZ R3, H2
        DJNZ R4, H1
        RET
        END.
```



Output:

The stepper motor is rotating in anticlockwise direction.

Result:

Thus the program has been successfully verified and executed.

10. Rotating stepper motor for 45 Angle in 8051 using Proteus.

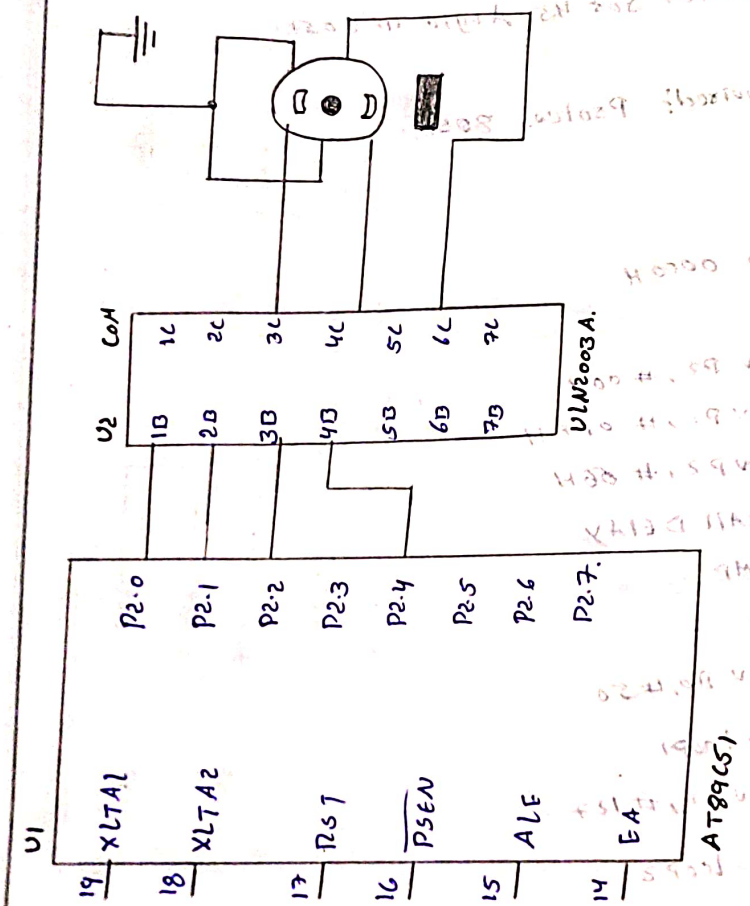
Aim: To write an assembly language program rotating the stepper motor for 45 Angle in 8051.

Software Required: Proteus 8051

Program:

```
        ORG 0000H
MAIN:
        MOV P2, #00H
        MOV P1, #0FFH
        MOV P2, #66H
        ACALL DELAY
        SJMP
DELAY:
        MOV R0, #50
DELAY - LOOP1
        MOV R1, #127
DELAY - LOOP2
        DJNZ R1, DELAY - LOOP2
        DJNZ R0, DELAY - LOOP1
END
```

Circuit Diagram



Result:

Thus the program is executed successfully for 45° angle in rotating stepper motor.

11. Rotating stepper motor for 90° Angle in 8051 using Proteus.

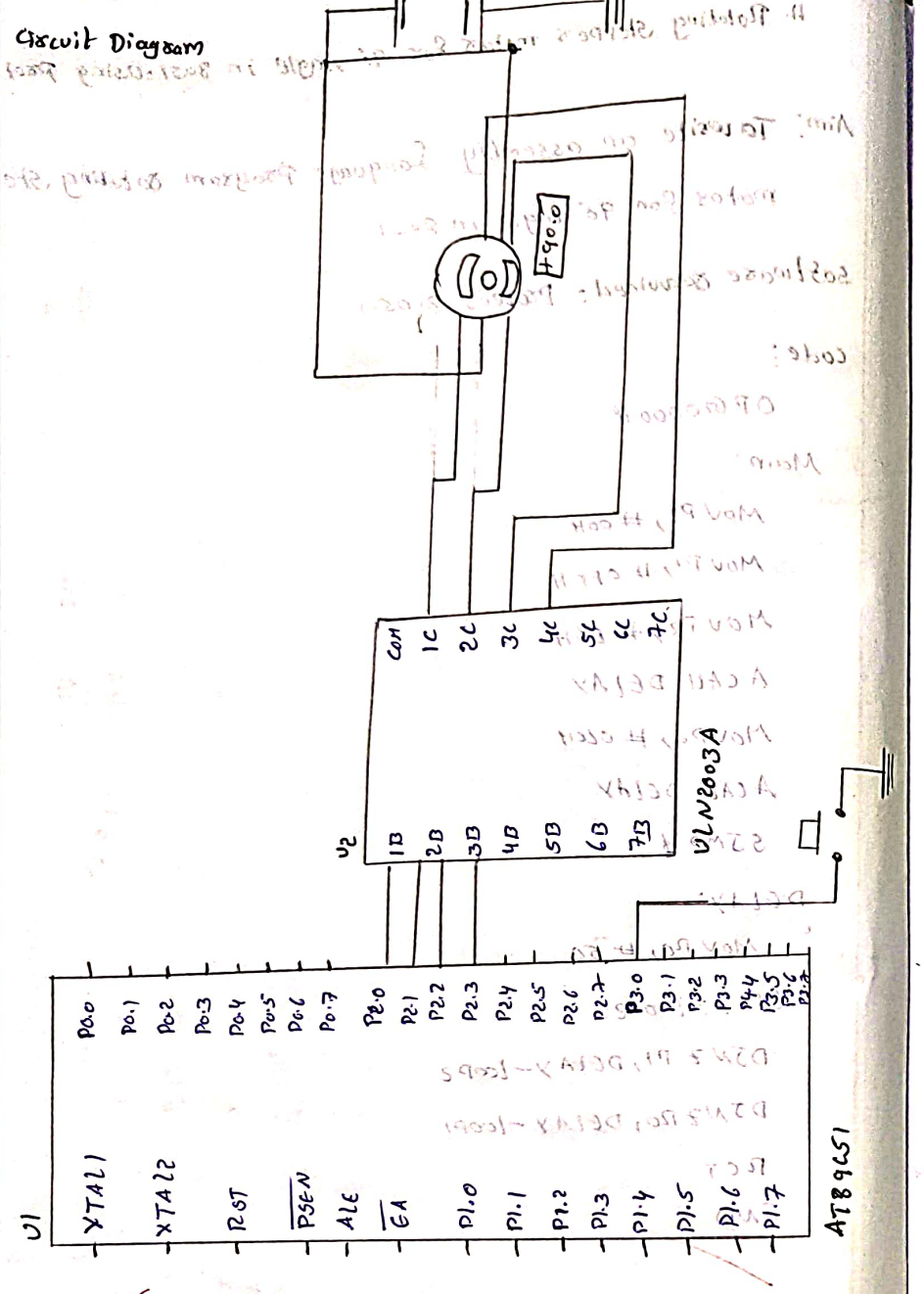
Aim: To write an assembly Language program rotating stepper motor for 90° angle in 8051

Software required: Proteus 8.05.1

code:

```
ORG 0000H
Main:
    MOV P2, #00H
    MOV P1, #0FFH
    MOV P2, #66H
    ACALL DELAY
    MOV P2, #01CH
    ACALL DELAY
    SJMP 3
DELAY:
    MOV R0, #50
DELAY: LOOP2:
    DJNZ R1, DELAY-LOOP2
    DJNZ R0, DELAY-LOOP1
    RET
END
```

Circuit Diagram



Result: Thus the program is executed successfully for 90 angle is rotating stepper motor.

12. Interfacing of Relay & Bulb with 8051 using Proteus.

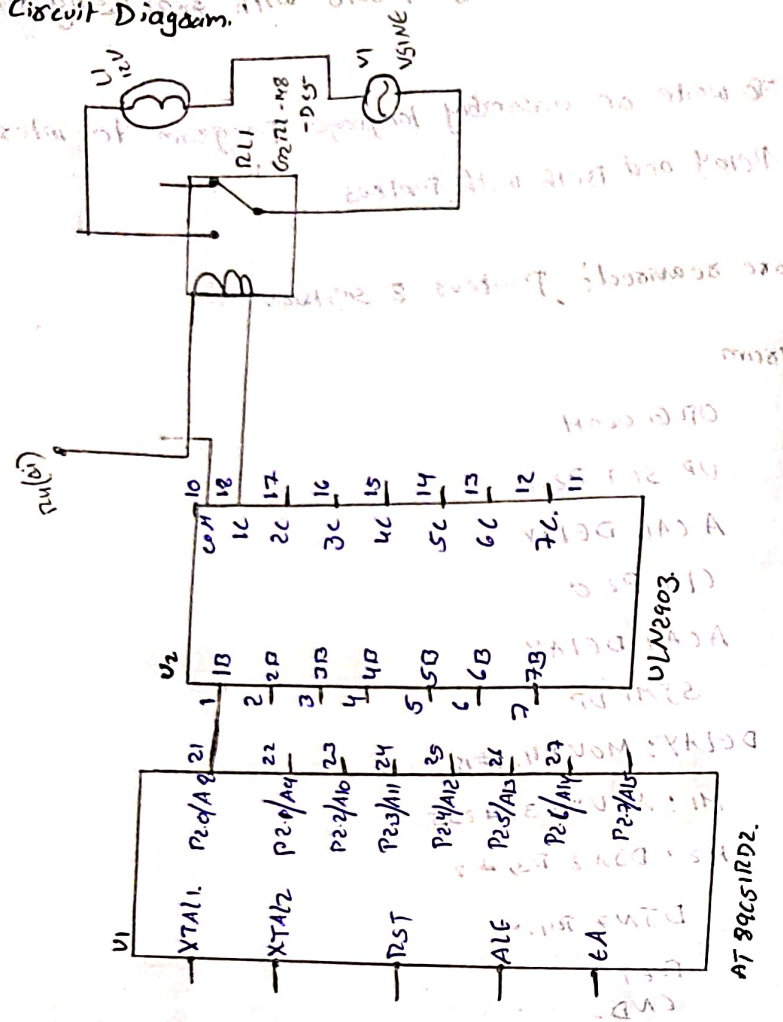
Aim: To write an assembly language program to interface Relay and Bulb with Proteus.

Software required: Proteus & software.

Program.

```
ORG 000H
UP: SET P2.0
ACALL DELAY
CLR P2.0
ACALL DELAY
SJMP UP
DELAY: MOV R4, #18
M1: MOV R3, #255
M2: DJNZ R3, #2
DJNZ R4, M1
RET
END.
```

Circuit Diagram:



output:

The bulb connected to the relay will glow at regular interv
with an and off duration.

Result:

Thus, the program has been successfully verified and executed

13. Interfacing of Relay and LED with 8051 using Proteus

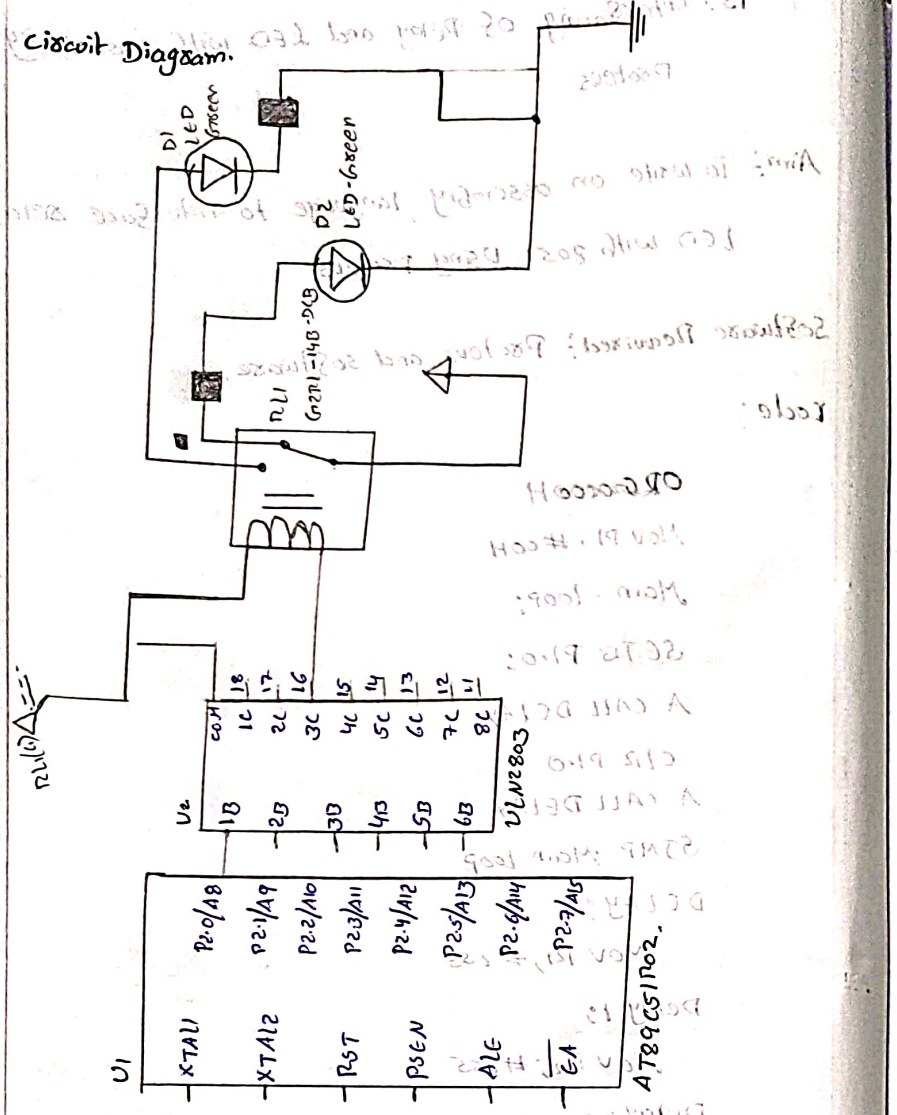
Aim: To write an assembly language to interface relay LED with 8051 using Proteus.

Software Required: Proteus and software.

Code:

```
ORG 0000H
MOV P1, #00H
Main - loop:
SETB P1.0:
A CALL DELAY
CLR P1.0
A CALL DELAY
SJMP Main Loop
DELAY:
   MOV R1, #255
Delay 1:
   MOV R2, #255
Delay 2:
   DJN2 R2, Delay 2
   DJN2 R1, Delay 1
   RET
END
```

Circuit Diagram.



Output:

The LED connected through the relay will blink with a controlled ON and OFF duration

Result:

Thus the program has been successfully verified and executed

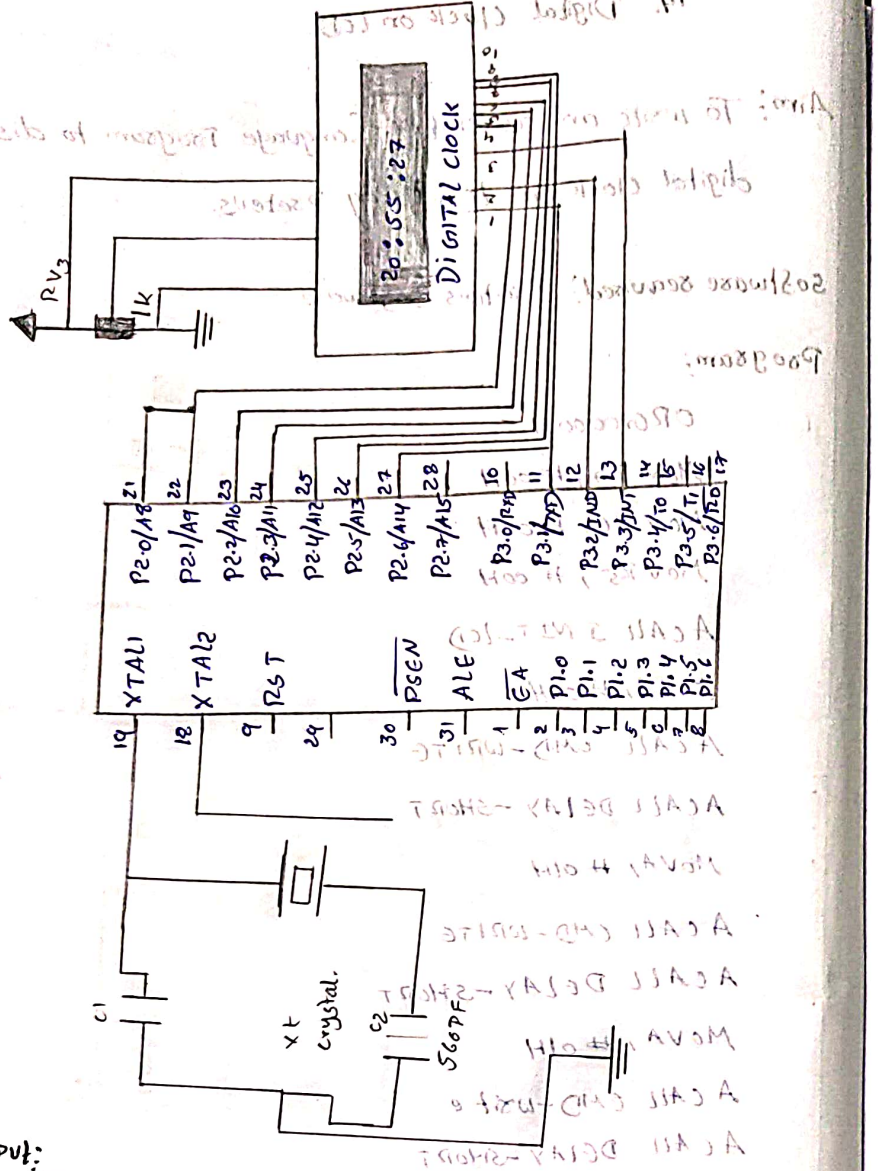
14. Digital clock on LCD

Aim: To write an assembly Language program to display digital clock on LCD using Proteus.

Software required: Proteus software.

Program:

```
        ORG 0000H
        MOV R7, #00H
        MOV R6, #00H
        MOV R5, #00H
        ACALL INIT_LCD
        MOV A, #0CH
        ACALL CMD_WRITE
        ACALL DELAY_SHORT
        MOV A, #01H
        ACALL CMD_WRITE
        ACALL DELAY_SHORT
        MOV A, #01H
        ACALL CMD_WRITE
        ACALL DELAY_SHORT
        RET
DELAY_1_SEC:
        MOV R3, #50
DELAY_LOOP
        MOV R4, #255
DELAY_LOOP_INNER
        DJNZ R4, DELAY_LOOP_INNER
        DJNZ R3, DELAY_LOOP
        END
```



Output:

When the program is running the LCD will display the time in the given format.

Every second the display will update to increment the seconds value.

Result:

Thus the assembly language program to display digital clock on LCD with using proteus was executed

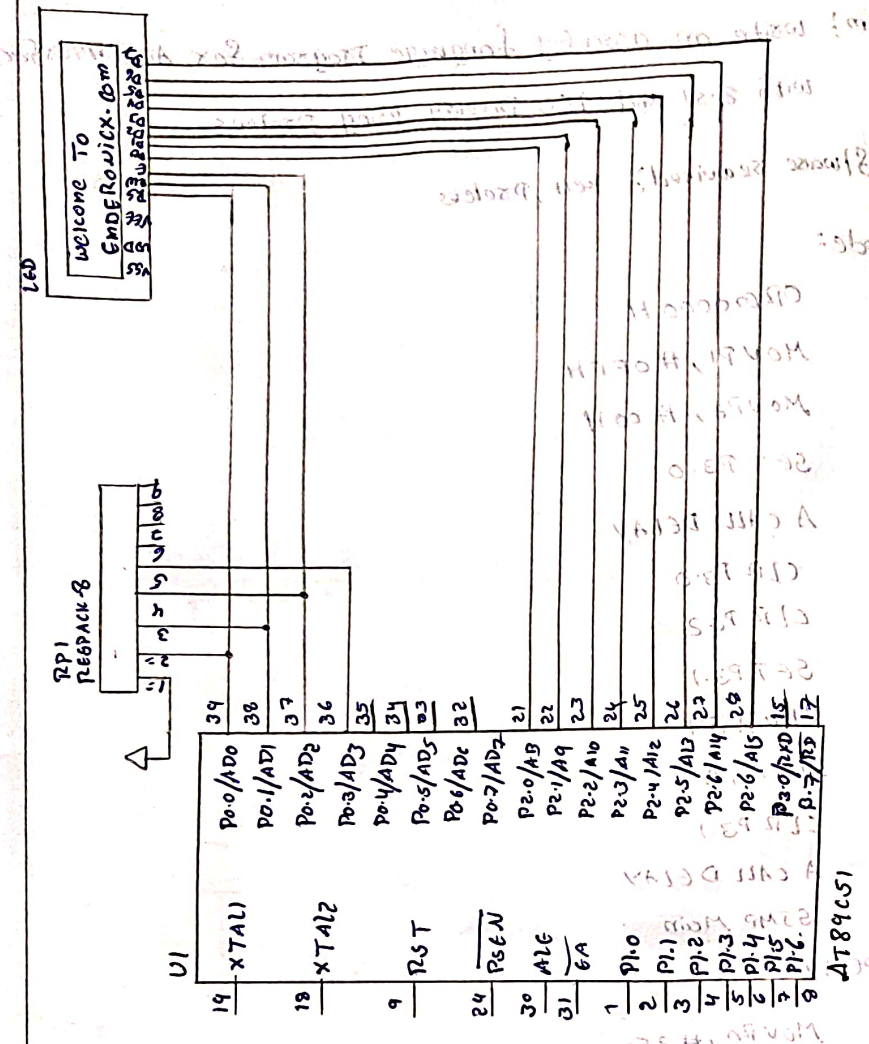
15. ADC interfacing with 8051 and LED Display using proteus.

Aim: write an assembly language program for ADC interfacing with 8051 and LED Display using proteus.

Software required: keil, proteus

code:

```
ORG 0000H
MOV P1, #0FFH
MOV P2, #00H
SET P3.0
ACALL DELAY
CLR P3.0
CLR P3.2
SET P3.1
MOV A, P1
MOV P2, A
CLR P3.1
ACALL DELAY
SJMP Main
DELAY: MOV R0, #250
MOV R1, #250
DJNZ R1, $
DJNZ R0, DELAY
RET
END.
```



Output:

When the analog input voltage is varied, the ADC converts into a digital value.

The converted digital value is displayed on LED.

Result:

Thus the assembly language program is executed and verified successfully.

16. ADC Interfacing with 8051 and LCD Display using Proteus

Aim: To write an assembly language program for ADC the Interfacing with 8051 and LCD Display using Proteus.

Software Required: Proteus 8 software.

program

```
        ORG 0000H
        MOV P1, #0FFH
        MOV P2, #00H
LOOP:
        SETB SC
        SETB ALE
        ACALL DELAY
        CLR ALE
        CLR SC
        SETB OE
        MOV A, P1
        CLR OE
DELAY:
        MOV R0, #250
        MOV R1, #250
DELAY_LOOP
        DJNZ R1, $
        DJNZ R0,
        RET
        END
```

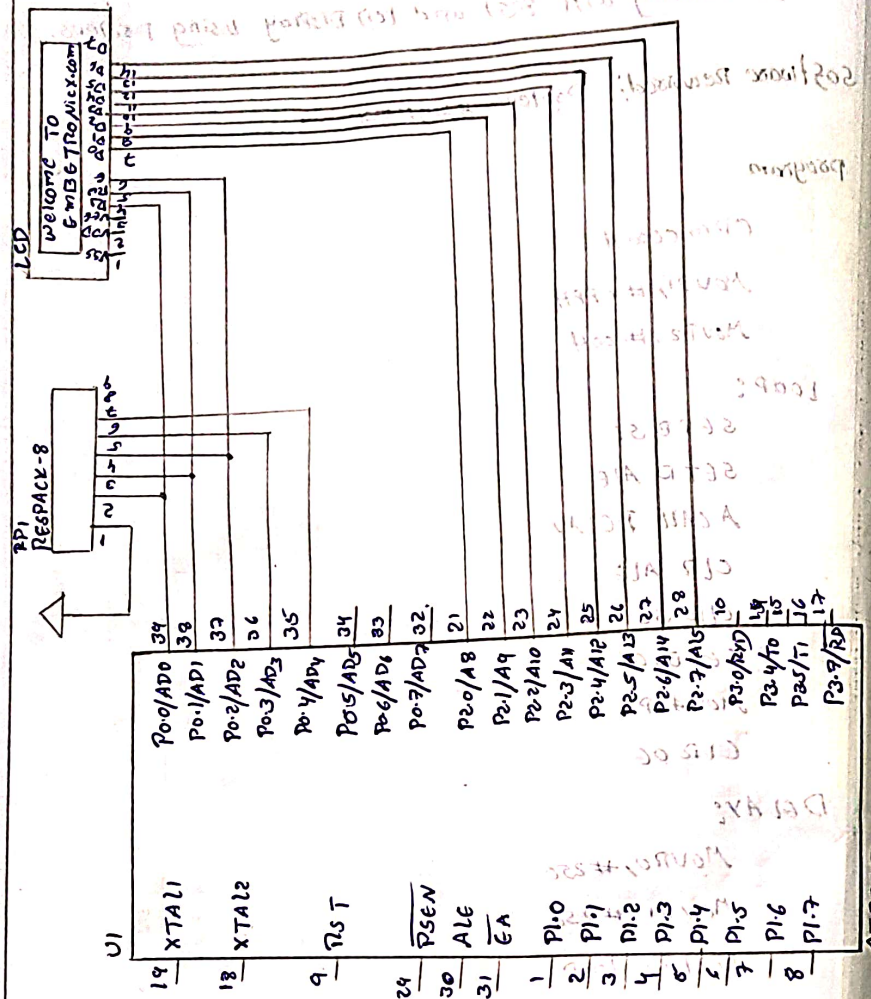
17. Temperature Monitoring and controlling system using 8051 Proteus.

Aim: To write an assembly Language code for Temperature monitoring and controlling system using 8051 Proteus.

Software Required: keil µvision, Proteus.

Program.

```
ORG 0000H
MOV P1, #0FFH
MOV P2, #00H
MOV P3, #00H
Display:
    MOV A,B
    MOV C, #100
    DIV AB
    ADD A
    ACALL LCD-DATA
DELAY:
    MOV R0, #250
    MOV R1, #250
DELAY-LOOP:
    DJNZ R1, $
    DJNZ R0, DELAY-LOOP
    RET
    END.
```

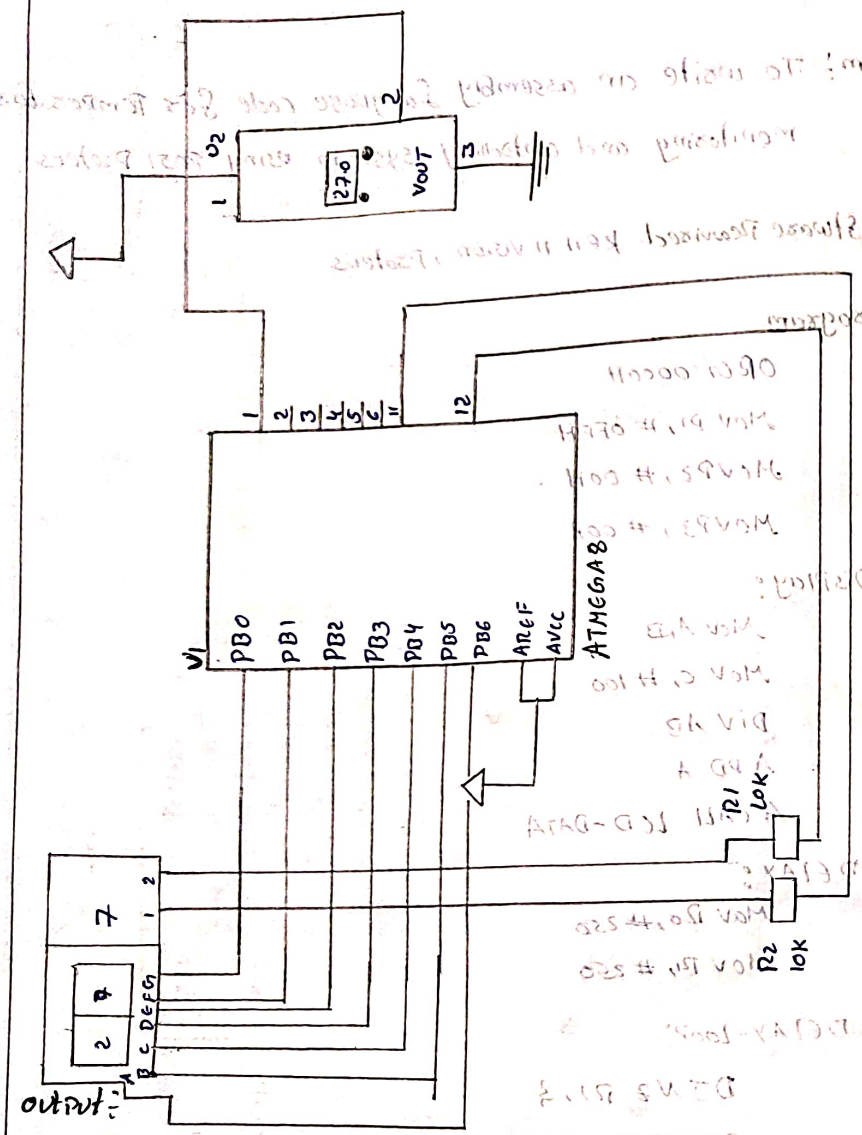


output:

The ADC converts the analog input into a digital value. The converted value is displayed on the LCD in decimal format.

Result:

Thus the assembly code for LED display is executed successfully and verified.



The current temp is displayed on LCD

If temp exceeds more than 30c the fan is "on"

if temp falls below so, the fan turns "off"

Result:

Thus the assembly language code for temperature monitoring system is executed successfully

18. Interfacing of RELAY and Buzzer using Proteus 8051.

Aim: Write an assembly language program for interfacing of Relay and Buzzer using Proteus 8051.

Software Required: Proteus 8051.

Program:

```
        ORG 0000H
        MOV P1,#00H
        MOV P2,#00H
        RELAY EQU P2.0
        P2.0
        P2.1
Main:
        MOV A,P1
        JB ACC.0,TURN-ON:
        SJMP TURN-OFF:
TURN-ON:
        SETB RELAY
        SETB BUZZER
        SJMP Main
TURN-OFF
        CLR RELAY
        CLR BUZZER.
        SJMP Main
    END.
```

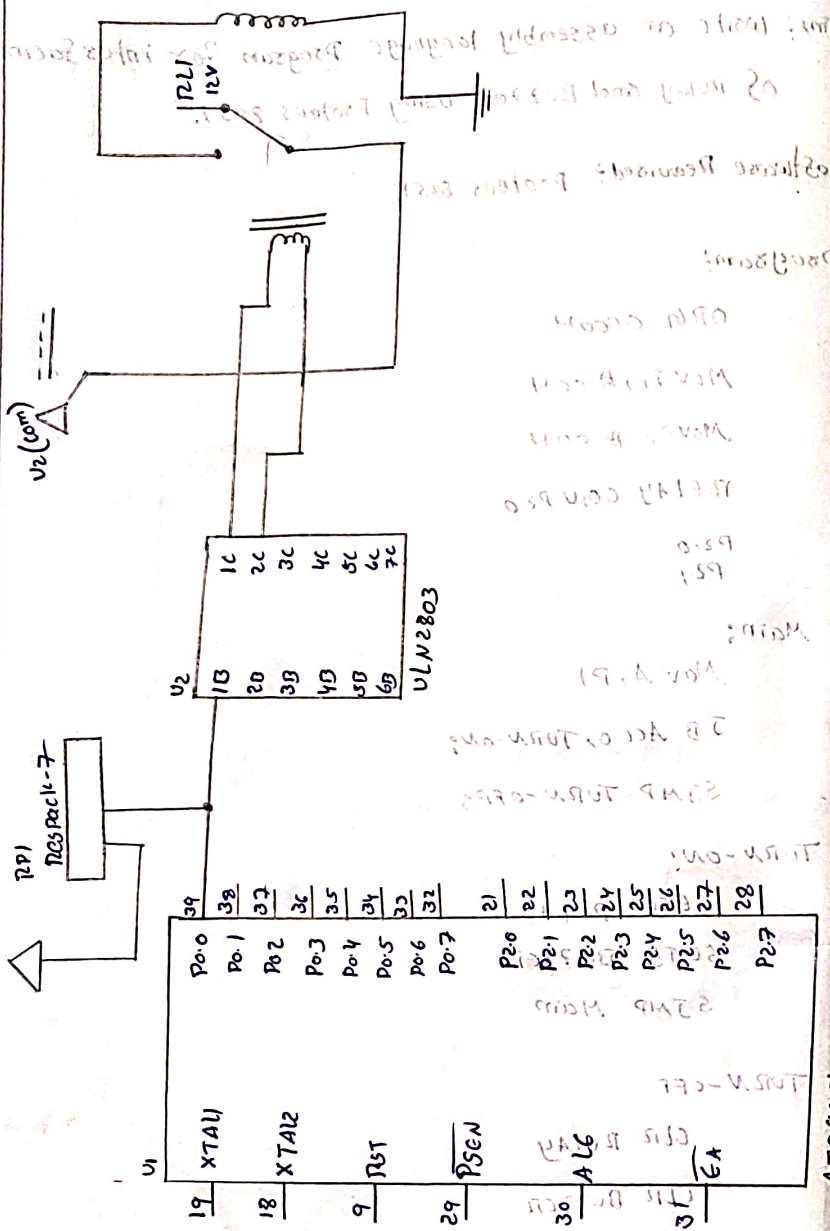
19. Interfacing of RELAY and DC Motor with Proteus 8051.

Aim: To write an assembly language program for interfacing of relay and DC motor with proteus 8051.

Software required: Keil µ version, Proteus 8051.

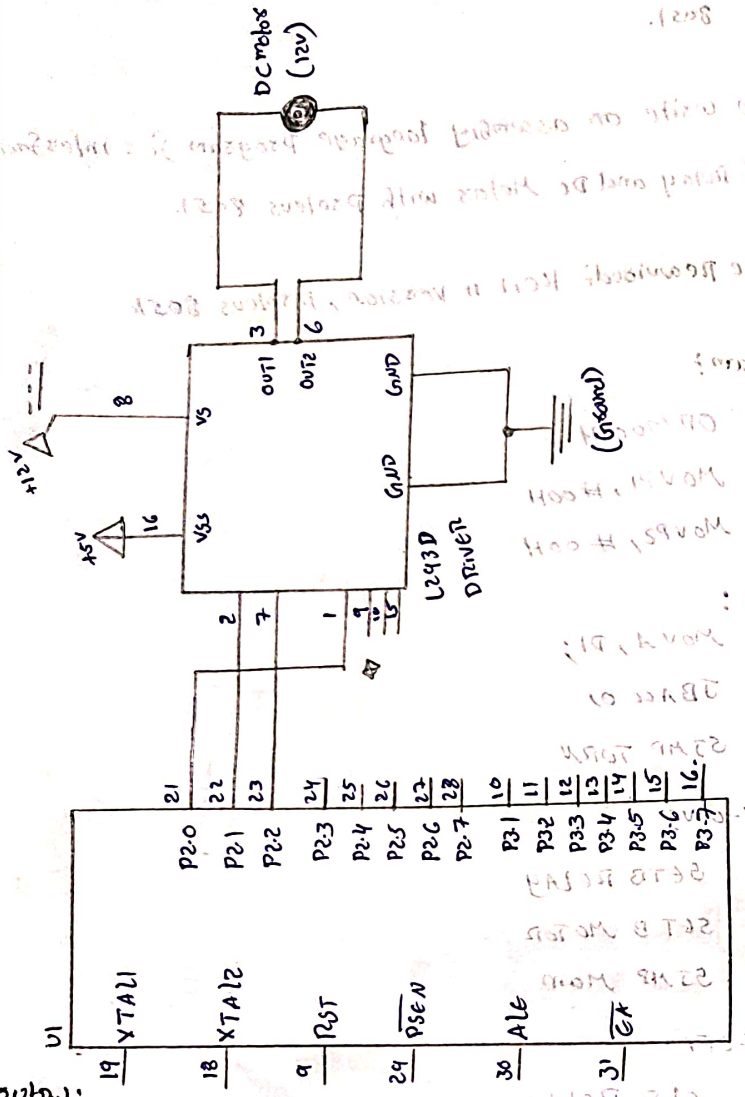
Program:

```
ORG 0000H
MOV P1, #00H
MOV P2, #00H
Main:
    MOV A, P1;
    JB Acc.0, TURN-ON
    SJMP TURN-OFF
TURN-ON:
    SETB RELAY
    SET B MOTOR
    SJMP Main
TURN-OFF:
    CLR RELAY
    CLR MOTOR
    SJMP Main
END.
```



Result:

Thus, the assembly language program for interfacing of relay and buzzer is executed successfully and verified.



Output:

* When the input switch is pressed, the relay is activated, and the DC motor starts running

* Releasing the switch turns off the relay and motor.

Result:

Thus the code is verified and executed successfully for DC motor.

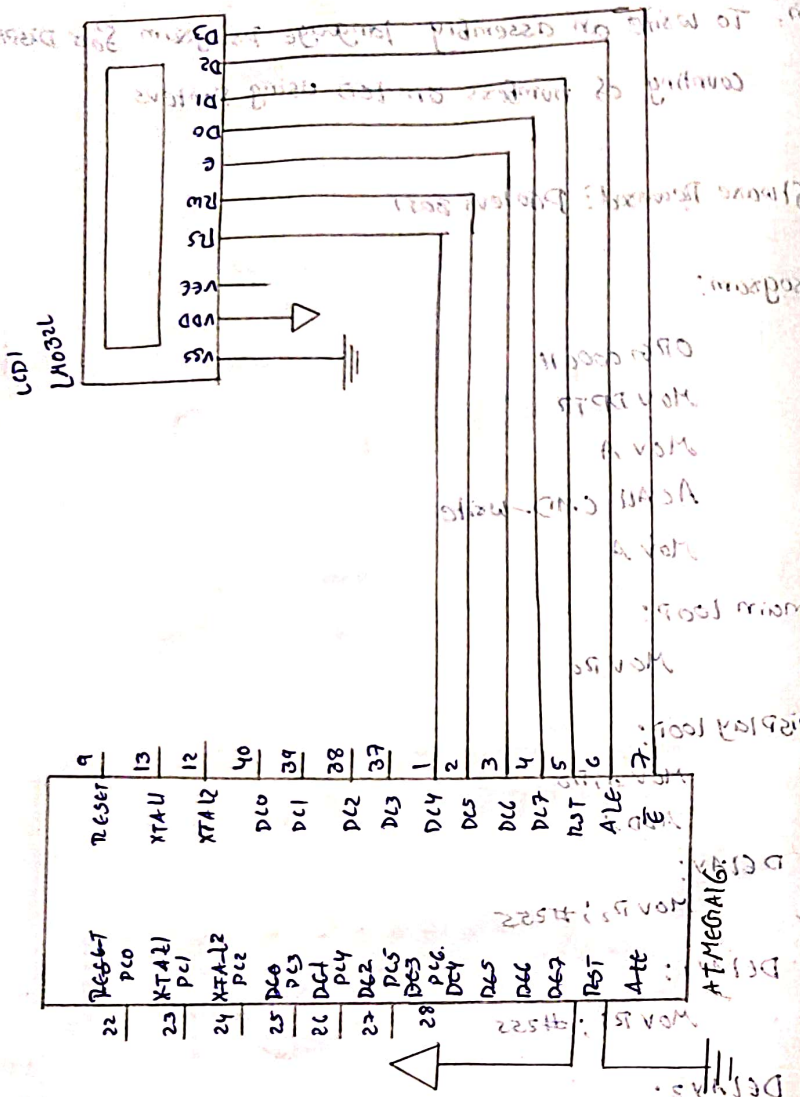
20. Display counting of numbers on LCD using Proteus.

Aim: To write an assembly language program for Display the counting of numbers on LCD using Proteus

Software Required: Proteus 8051

Program:

```
        ORG 0000H
        MOV DPTR
        MOV A
        ACALL CMD_write
        MOV A
main loop:
        MOV R0
Display loop:
        MOV A,R0
        ADD A
DELAY:
        MOV R2;#255
DELAY1:
        MOV R1;#255
DELAY2:
        DJNZ R1,DELAY2
        DJNZ R2,DELAY1
        RET
        END
```



output;

* The LCD will Display numbers from 0 to 9 at one time
* After reaching it will clear the display and restart the counting

Result;

Thus the output is verified and executed successfully

21. Display A Number in 7 segment Display with 8051 Proteus.

Aim: To write an assembly language program for Display a number in 7 segment display.

Software Required: Proteus 8051

Code:

```
        ORG 0000H
        MOV P1, #92H
HERE
        SJMP HERE ; infinite loop to keep the display active
DELAY:
        MOV R2,
        MOV R1,
        DJNZ R1,
        DJNZ R2
        RET
END
```

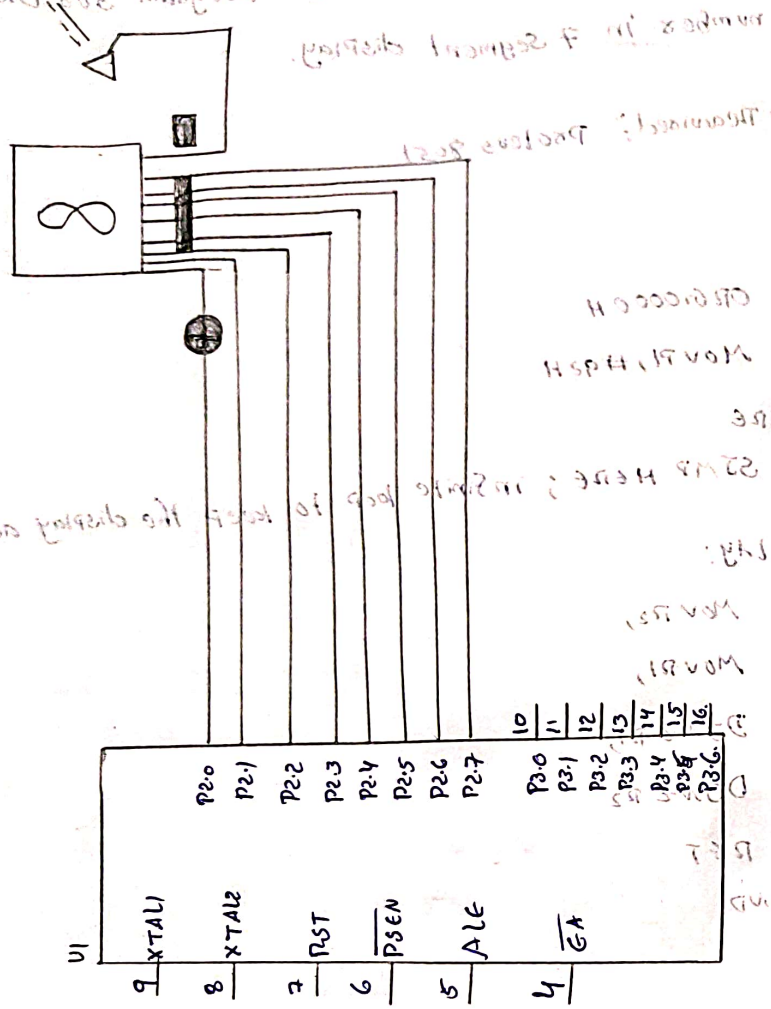
22. counting up the numbers in 7 segment Display with 8051 using Proteus

Aim: To write an assembly language Program Sor counting Up the numbers in 7 segment Display.

Software Reauired: Proteus 8051

Program:

```
ORG 0000H
  MOV R0,
  MOV A,
A CALL Display
Delay subroutine
Delay:
   MOV R2, #255
Delay1:
   MOV R1, #255
Delay2:
   DJNZ R1, DELAY2
   DJNZ R2, Delay1
   RET
END.
```



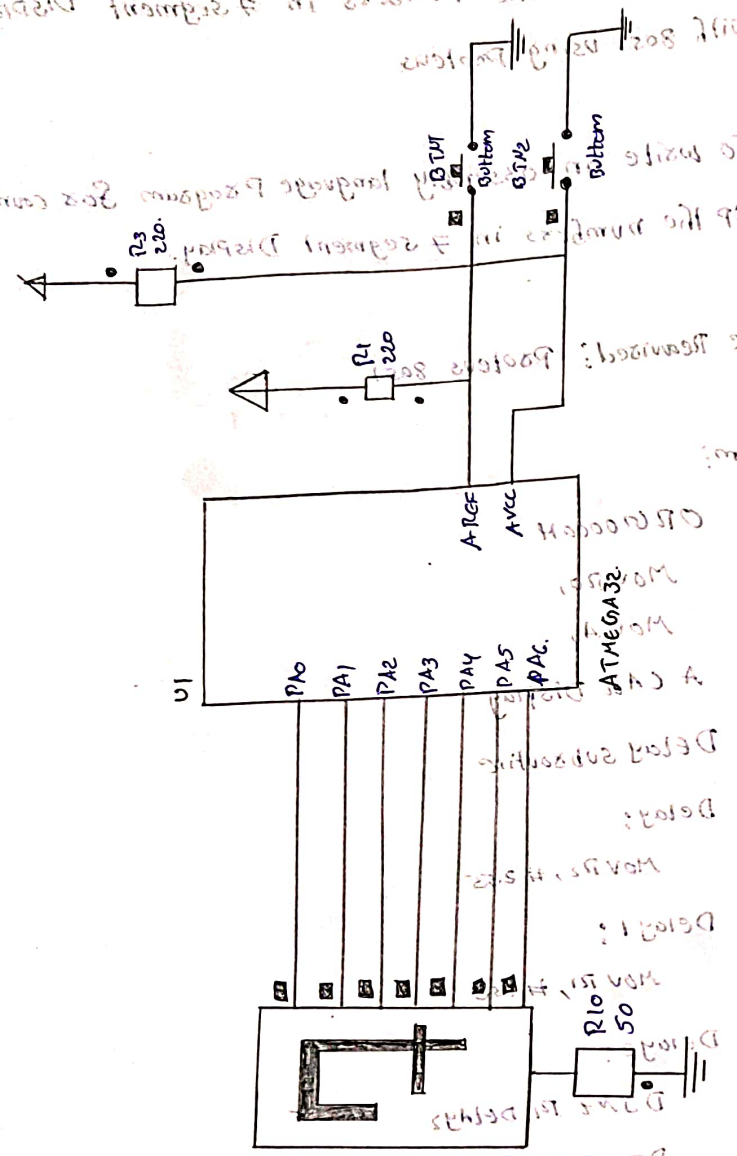
Output:

This corresponds to the digits.

load the Hexafile and output will be displayed.

Result:

Thus, the Program is executed



Output:

* The 7-segment display will seaventially show numbers from 0 to 9, one at a time.

* The seaverence will repeat in an infinite loop, with display numbers

Result:

Thus the output is executed successfully

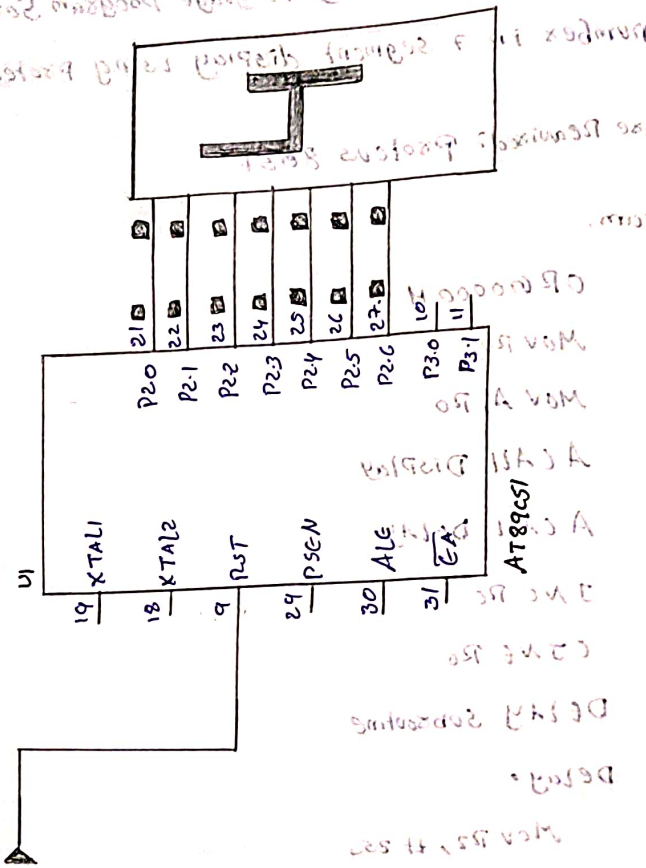
23. Display a Hexadecimal number in 7 segment display using 8051 Proteus.

Aim: To write an assembly language program for Hexadecimal number in 7 segment display using Proteus.

Software Reauired: Proteus 8051.

program.

```
ORG 0000H
MOV R0,
MOV A, R0
ACALL DISPLAY
ACALL DELAY
INC R0
CJNE R0
DELAY SUBROUTINE
Delay:
    MOV R2, #255
Delay1:
    MOV R1, #255
Delay2:
    DJNZ R1,
    DJNZ R2,
    RET
END.
```



Output:

The 7-segment display will seaventially show hexadecimal number : 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F.

Result:

Thus the output is verified and executed successfuly.

24. Study of PCB Printing using EagleCAD.

Aim: To write an assembly language Program for study of PCB Printing using eagle cad.

Software Reauired: Eagle.CAD, KiCAD

Procedure.

1, Schematic Creation:

* open Eagle CAD and create a new Project.
* add components from the Library, connect then using wires, and assign values.

2 PCB layout:

* Switch to PCB layout View
* Arrange components and create traces between them use Auto-Rotate for faster results.

3, Export PCB for Printing

* Go to File > CAM Processor, select the Gerber format, and generate the file for PCB fabrication.

output:

1. The schematic is converted into a PCB layout with the component Placement and routed traces.
2. A ready to fabricate Gerber file is generated for manufacturing

Result:

Thus the output is verified and executed successfuly.

25. Blinking of LED using Arduino uno

Aim:- To blink an Led connected to an arduino uno board at regular intervals.

Apparatus:-

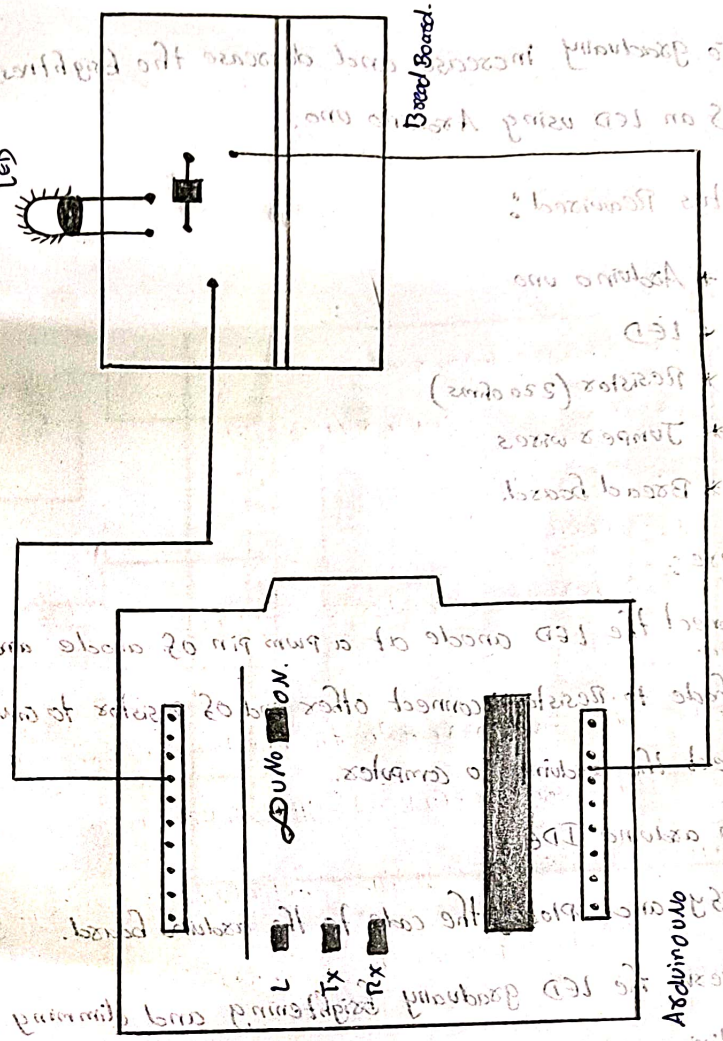
* Arduino uno
* LED
* Resistor
* Jumper wires
* Bread board.

Procedure:

1. Connect the LED's longer Led (Anode) to digital pin and the shorter leg to resistor.
2. connect the arduino to a computer using a USB cable.
3. Open the Arduino IDE
4. Verify and upload the code to the Arduino board.
5. Observe the LED blinking at 1-second intervals.

output:

* The LED will turn ON for 1 second
* The LED will turn OFF for 1 second
* The cycle will repeat indefinetly.



Result:-

We successfully completed the executed the blinking of LED.

26. Fading of an LED using Arduino UNO.

Aim; To gradually increase and decrease the brightness of an LED using Arduino uno.

Apparatus Required:

* Arduino uno
* LED
* Resistor (220 ohms)
* Jumper wires
* Bread board.

Procedure:

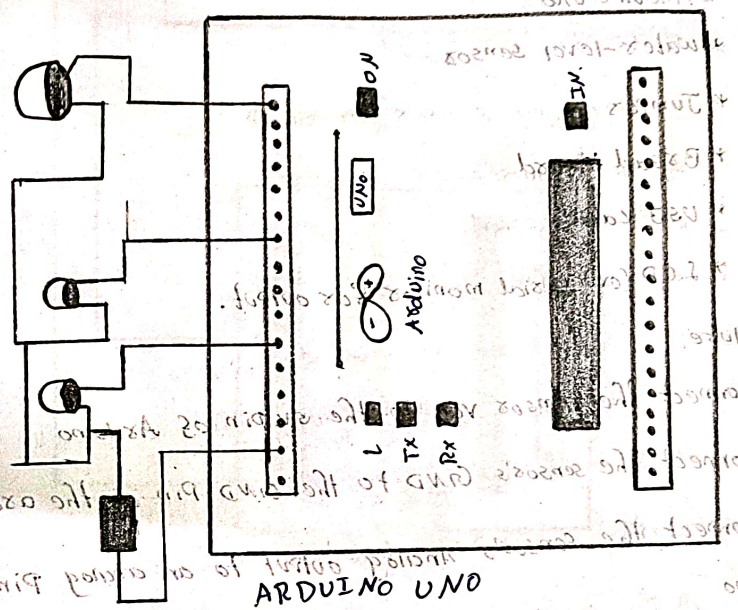
1. connect the LED anode at a pwm pin of anode anode and it's cathode to Resistor connect other end of Resistor to GND.
2. connect the arduino to computer.
3. open arduino IDE.
4. Verify and upload the code to the arduino board.
5. Observe the LED gradually brightening and dimming in a continious loop.

output.

The LED brightness gradually increases.

The LED brightness gradually decreases.

The fading process repeats continiously.



Result:

Thus, the above program is executed successfully fading of LED.

27. Interfacing a water-level sensor with Arduino UNO.

Aim: To detect and display water levels using a water level sensor connected to Arduino UNO.

Apparatus Required.

* Arduino UNO
* water-level sensor
* Jumpers
* Bread board
* USB cable
* LCD (or) serial monitor for output.

Procedure.

Connect the sensor VCC to the 5V pin of Arduino

Connect the sensor's GND to the GND Pin of the arduino

Connect the sensor's Analog output to an analog Pin on Arduino

Open the serial monitor in the Arduino IDE to observe real-time water level-sensor.

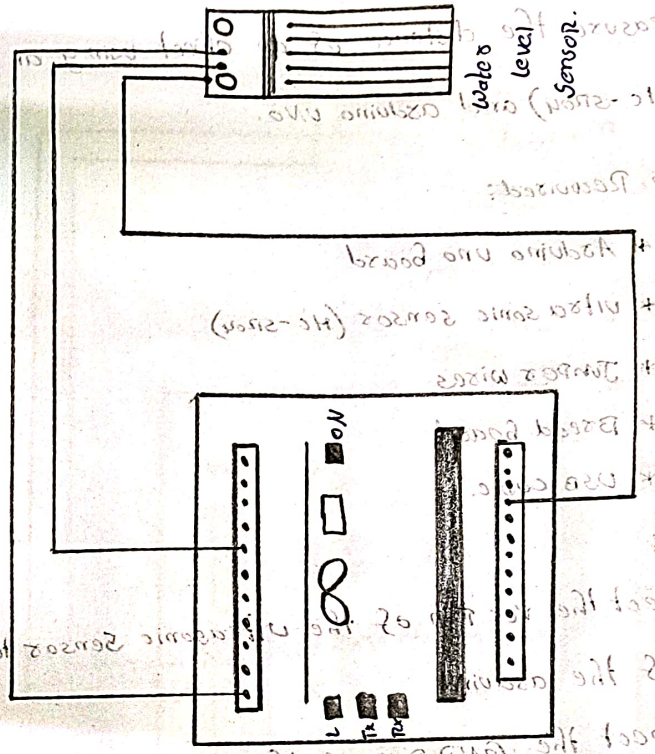
Set thresholds to trigger alerts for high (or) low water levels.

Output:

The sensor detects water levels based on its immersion depth.

Real-time water-level readings are displayed on the serial monitor.

Changes in sensor reading reflects variations in water levels



Result:

Thus the above program is executed successfully water level sensors.

28. Interfacing an ultrasonic sensor with Arduino uno.

Aim:

To measure the distance of an object using an ultrasonic sensor (Hc-sr04) and arduino uno.

Apparatus Required:

* Arduino uno board
* ultrasonic sensor (Hc-sr04)
* Jumper wires
* Bread board
* USB cable.

Procedure;

1, Connect the vcc pin of the ultrasonic sensor to the 5V pin of the arduino.

2, Connect the GND pin of the sensor to GND of Arduino.

3, Connect the trig pin of the sensor to pin 9 of the Arduino.

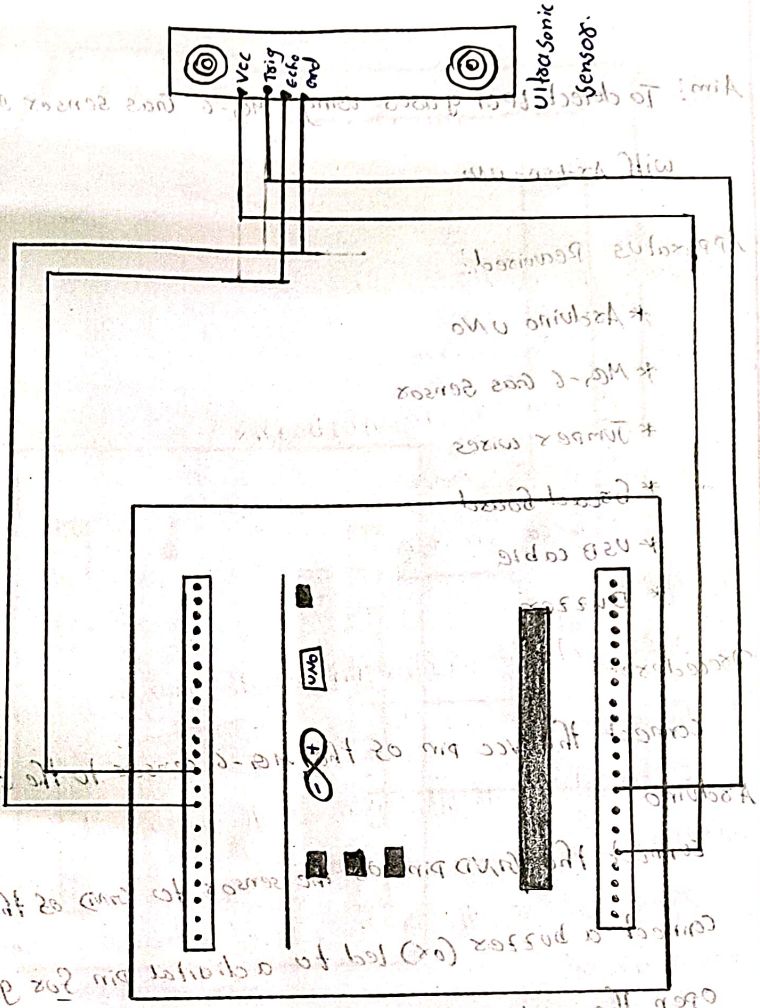
4, Open the arduino.

5, Open the serial monitor in the Arduino and observe distance readings.

6, Display distance reading on a LCD.

output

The ultrasonic sensor emits ultrasonic waves and measures the time taken for them return.



Result:

Thus the above program is executed successfully Ultrasonic sensor.

29. MQ-6 gas sensor Interfacing with arduino uno.

Aim: To detect LPG gases using MQ-6 Gas sensor interfaced with Arduino uno.

Apparatus Reavired:

* Arduino uNo
* MQ-6 Gas sensor
* Jumper wires
* Bread board
* USB cable
* Buzzer.

Procedure;

Connect the vcc pin of the MQ-6 sensor to the 5V pin of Arduino

Connect the GND pin of the sensor to GND of the Arduino.

Connect a buzzer (or) led to a digital pin for gas-level

Open the arduino IDE

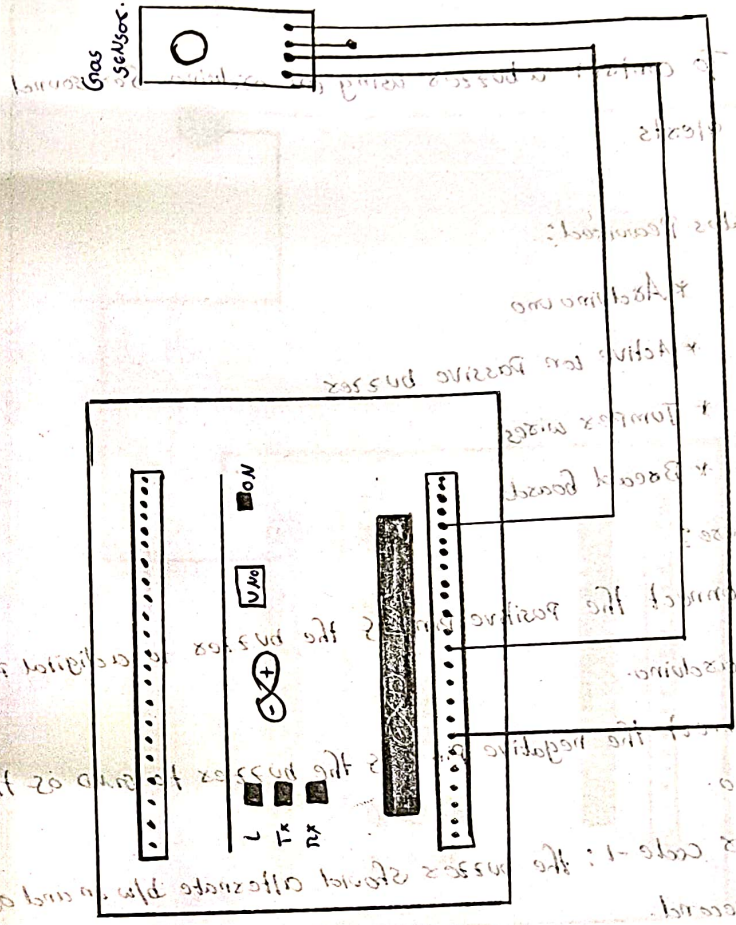
Open the serial monitor to observe gas-level readings.

Output:

The sensor detects gas levels and outputs analog reading

Real time gas levels are displayed.

An alert is triggered when gas levels exceed the set threshold



Result:

Thus the above Program is executed successfully by agas sensor detection.

30. Interfacing a buzzer with arduino UNO.

Aim: To control a buzzer using an arduino for sound based alerts.

Apparatus Required:

* Arduino uno
* Active (or) Passive buzzer
* Jumper wires
* Bread boards

Procedure:

1) connect the positive pin of the buzzer to a digital pin on the arduino.

2) Connect the negative pin of the buzzer to GND of the arduino.

3) For code-1: the buzzer should alternate b/w on and off every second.

4) For code-2: the buzzer will generate a periodic tone.

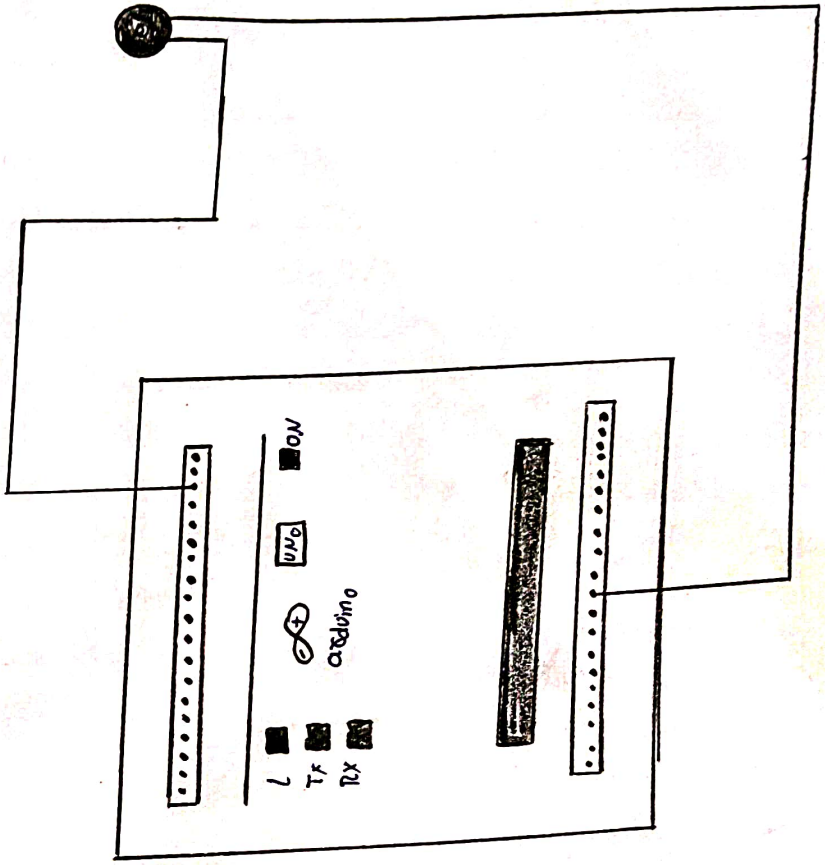
5) Adjust the tone frequency for different sound patterns.

output:

The buzzer emits a continious (or) periodic sound.

The sound varies based on code.

The buzzer saves as an audible alert mechanism.



ARDUINO UNO.

Result:

Thus the above program is executed successfully by interfacing a buzzer.