Experiment - 01

Study of Proteus and Keil MICRO VISION

Aim:-

To Study the working procedures of Proteus and Keil MICRO VISION softwares.

Keil MICRO VISION:-

Is a free software which solves many of the main points for an embedded program developer. This software is an Integrated development environment (IDE), which integrated a text editor to write programs, a compiler and it will convert your source code to hex files too. μ vision4 introduces a flexible window management system, enabling us to drag and drop individual windows anywhere on the visual surface including support for multiple Monitors.

Keil procedure:-

* Open the software, click on project and open new vision project.
* create a new project file.
* Enter AT89C51
* click NO.

* click [ctrl + N] and type the code.
* open project and click build target
* open Build target and open source file and ADD, CLOSE
* click build target.
* Next debug start and stop
* open peripherals and select port 2
* Now run the program in Debug
* open project and click optional properties and in that Gives output as hex file.
* create hex file.

Proteus procedure:-

* open proteus by clicking run as administrator.
* open new project and Enter the file name.
* click next, next, next and finish.
* click P symbol and search keyword and place the required components.
* Now connect the components as required.

* Give input to AT89C51 as hex file.
* start the simulation process.

Result :-

Thus the proteus and Keil MICRO VISION Softwares were studied.

Experiment - 02

Blinking of led using 8051 Micro-Controller using proteus.

Aim:-

To write an assembly language program to LED blink using 8051.

Softwares Required:-

Proteus software.

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 program has been successfully verified and executed.

Experiment-03

Generation of Square wave using Proteus.

Aim:-

Write an assembly language to Generate square wave using 8051.

Software Required:-

proteus 8 software.

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 program has been Successfully verified and executed.

Experiment - 04

Fade in fade out of led using 8051 using proteus.

Aim :-

write an assembly language program for fade in fade out of led using 8051 using Keil and proteus.

Software required :-

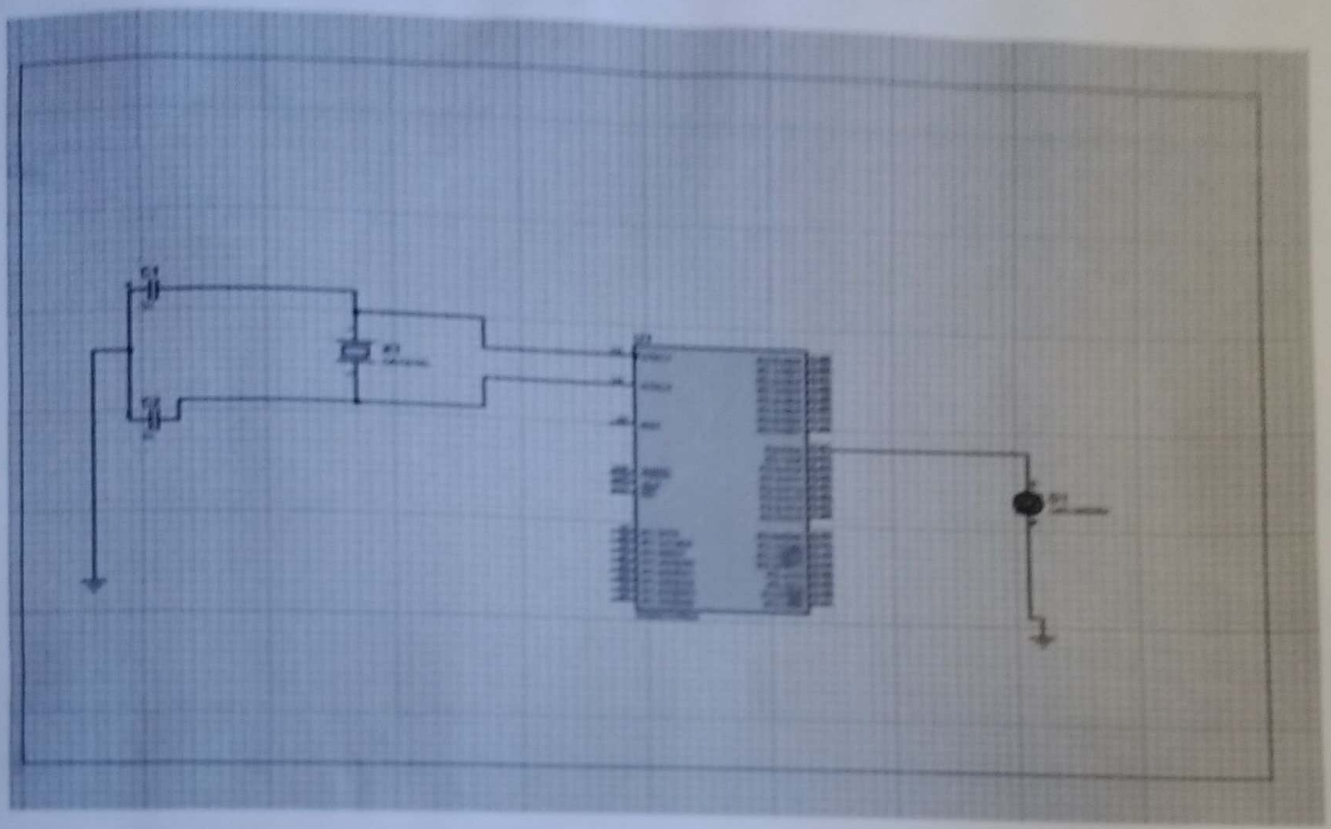
proteus 8 software.

program :-

```
        ORG 0000H
        SETB P1.0
LOOP:   ACALL DELAY
        CLR P1.0
        ACALL DELAY
        SETB P1.0
        SJMP LOOP
DELAY:
        MOV R3,#0FFH
DLOOP1:
        MOV R2,#0FFH
DLOOP2:
        MOV R1,#0FFH
DLOOP3:
        DJNZ R1,DLOOP3
```

```
        DJNZ R2, DLOOP2
        DJNZ R3, DLOOP1
        RET
END
```

FADE IN FADE OUT



Result :-

Thus the program has been successfully verified and executed.

Experiment - 05

Clockwise stepper motor using 8051 using proteus.

Aim:-

write an assembly language program for stepper motor using 8051 using keil and proteus.

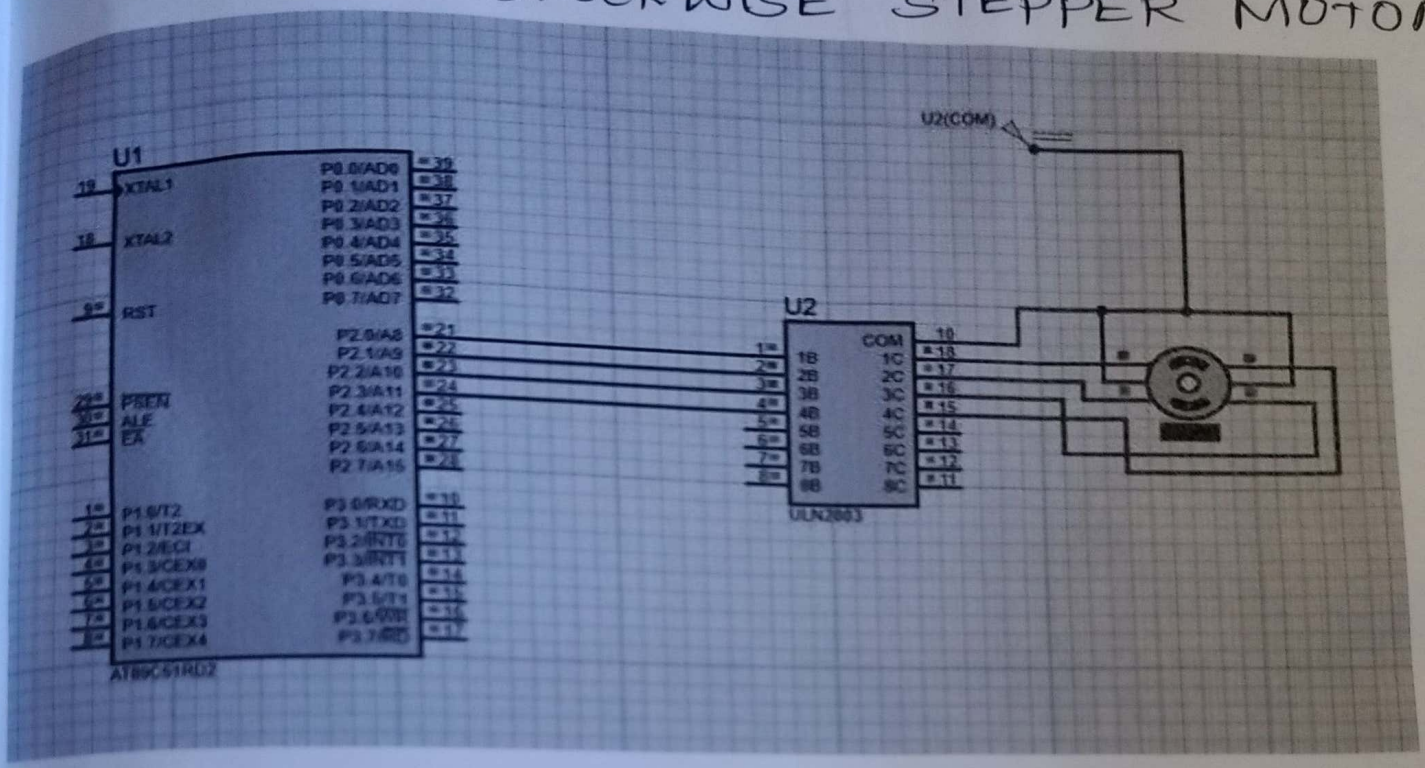
Software Required:-

proteus 8 software.

Program:-

```
ORG 0000H
UP: MOV P2,#09H
ACALL DELAY
MOV P2,#0CH
ACALL DELAY
MOV P2,#06H
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
```

CLOCKWISE STEPPER MOTOR



Result :-

Thus the program has been successfully verified and executed.

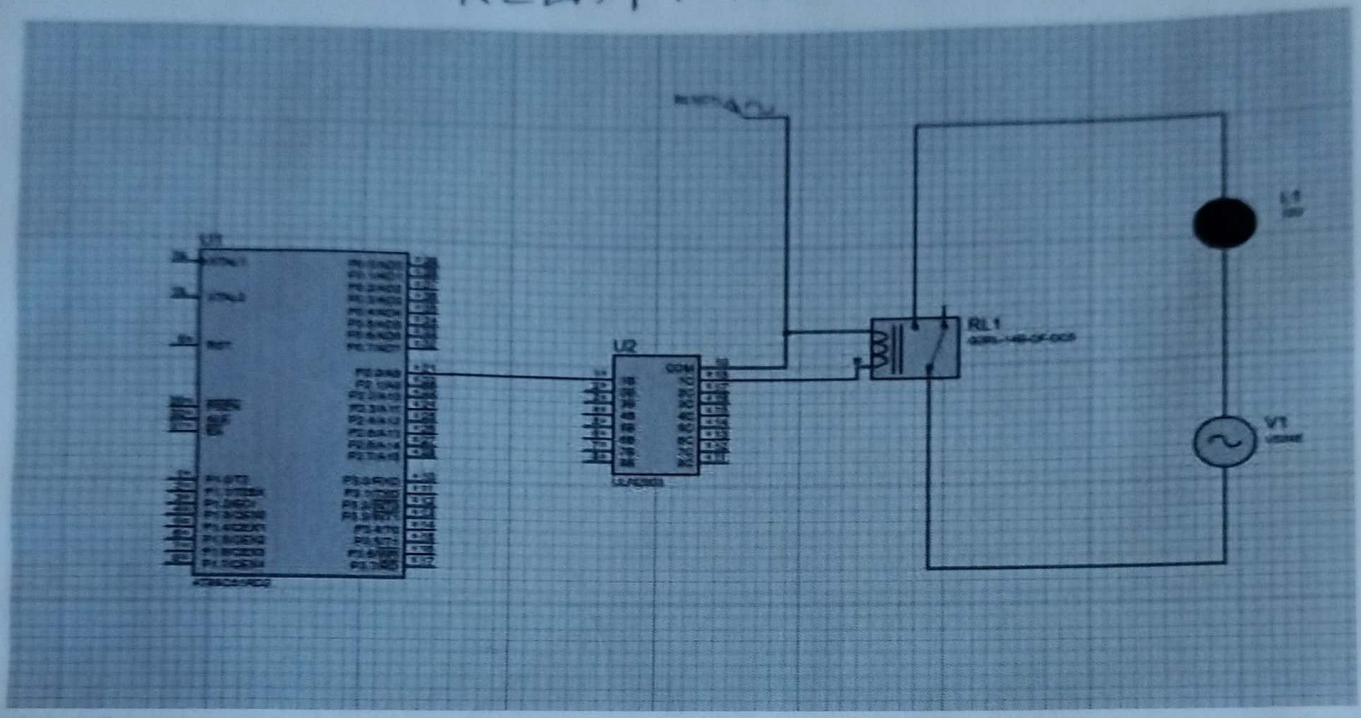
Experiment - 06

Interfacing of Relay and bulb using 8051 using proteus.

AIM :-

write an assembly language program for interfacing of Relay using 8051 using keil and proteus.

RELAY AND BULB.




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

Result :-

Thus the program has been successfully verified and executed.

Experiement - 07

Led Toggle using 8051 using Proteus.

Aim :-

Write an assembly language program for led toggle using 8051 using Keil and proteus.

Software Required :-

Proteus 8 software.

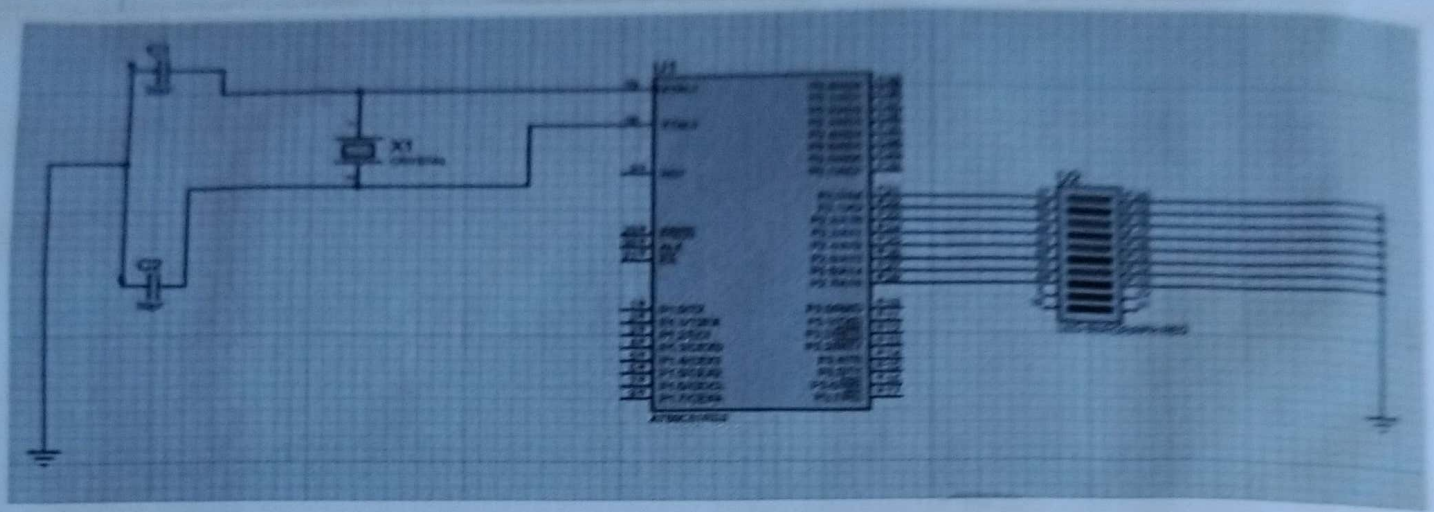
program :-

```
        ORG1 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.

LED TOGGLE



Experiment - 08.

7 Segment Display using 8051 Using Proteus.

AIM:-

Write an assembly language program for 7 Segment Display using 8051 using Keil and proteus.

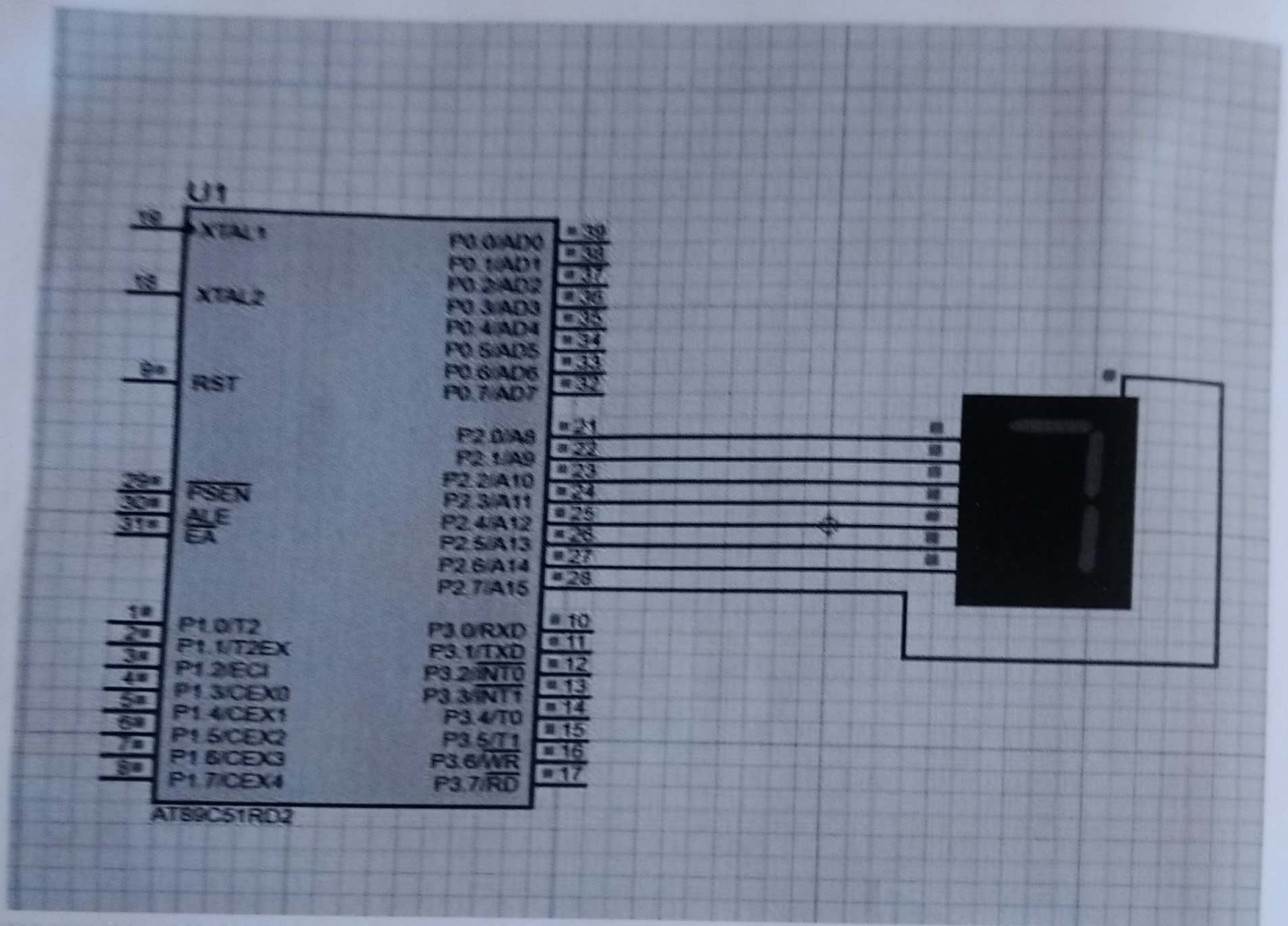
Software Required :-

Proteus 8 software.

Program :-

```
        ORG 0000H
UP:     MOV P2, #0C0H
        ACALL DELAY
        MOV P2, #0F9H
        ACALL DELAY
        MOV P2, #0B0H
        ACALL DELAY
        MOV P2, #99H
        ACALL DELAY
        MOV P2, #92H
        ACALL DELAY
        MOV P2, #82H
        ACALL DELAY
        MOV P2, #0F8H
        ACALL DELAY
        MOV P2, #80H
```

7 Segment Display

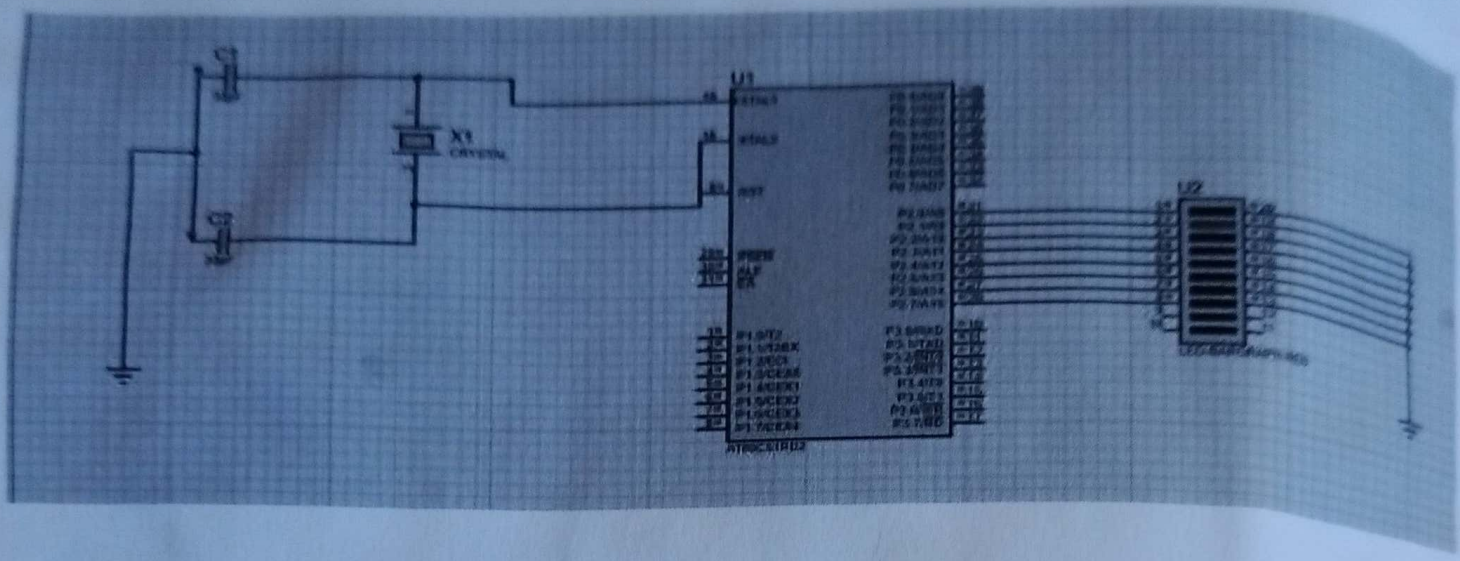


```
        ACALL DELAY
        MOV P2 ,#90H
        ACALL DELAY
DELAY : MOV  R5, #10
  H1  : MOV  R4 , #180
  H2  : MOV  R3 , #255
  H3  : DJNZ R3 , H3
        DJNZ R4 , H2
        DJNZ R5 , H1
        RET
        END
```

Result :-

Thus the program has been successfully verified and executed.

LED CHASER



Experiement - 09

LED chaser using 8051 using proteus

Aim :-

Write an assembly language program for LED chaser using 8051 using Keil and proteus.

Software Required :-

Proteus 8 software.

Program :-

```
ORG 0000H
UP: MOV P2, #01H
ACALL DELAY
MOV P2, #02H
ACALL DELAY
MOV P2, #04H
ACALL DELAY
MOV P2, #08H
ACALL DELAY
MOV P2, #10H
ACALL DELAY
MOV P2, #20H
ACALL DELAY
MOV P2, #40H
ACALL DELAY
MOV P2, #80H
ACALL DELAY
SJMP UP
DELAY: MOV R4, #25
H1: DJNZ R4, H1
RET
END
```

Result :-

Thus, the program has been Successfully verified and executed.

Experiment - 10

Generation of triangular wave using proteus.

AIM:-

To write an assembly language program to generate triangular wave using 8051.

Software required:-

proteus 8 software.

program:-

```
        ORG 00H
        MOV P2.0, #00H
        MOV A, #00H
        MOV R0, #00H
UPWARD: INC A
        MOV P1, A
        ACALL DELAY
        CJNE A, #0FFH, UPWARD;
DOWNWARD:
        DEC A
        MOV P1, A
        ACALL DELAY
        CJNE A, #00H, DOWNWARD;
        SJMP UPWARD
DELAY:
        MOV R1, #255
DELAY-LOOP1:
```

```
        MOV  R2, #255
DELAY-LOOP2:
        DJNZ R2, DELAY_LOOP2
        DJNZ R1, DELAY_LOOP1
        RET
END.
```

Result :-

Thus, the program has been Successfully Verified and Executed.

Experiment - 11

Anti Clockwise 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.

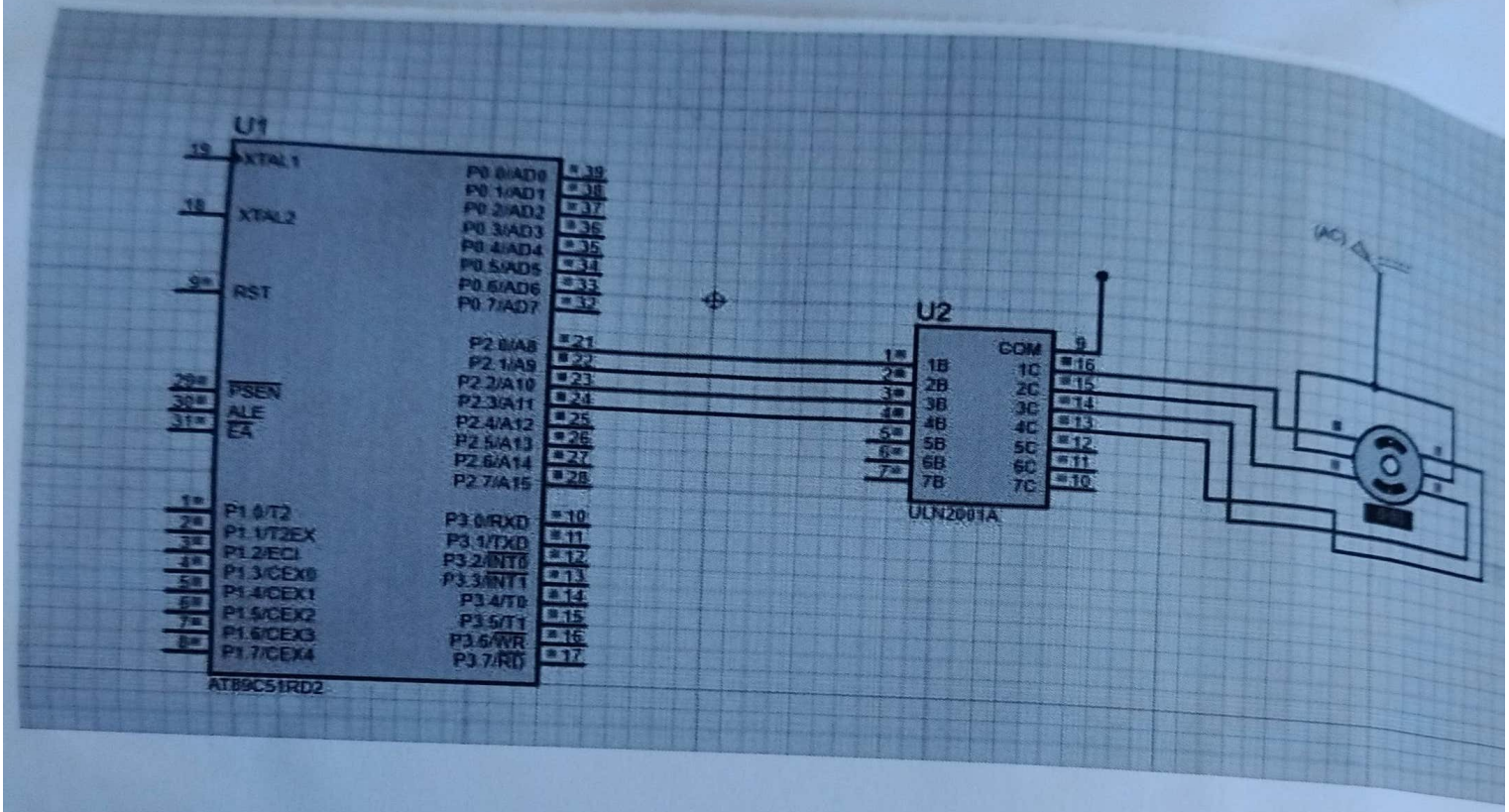
Software Required :-

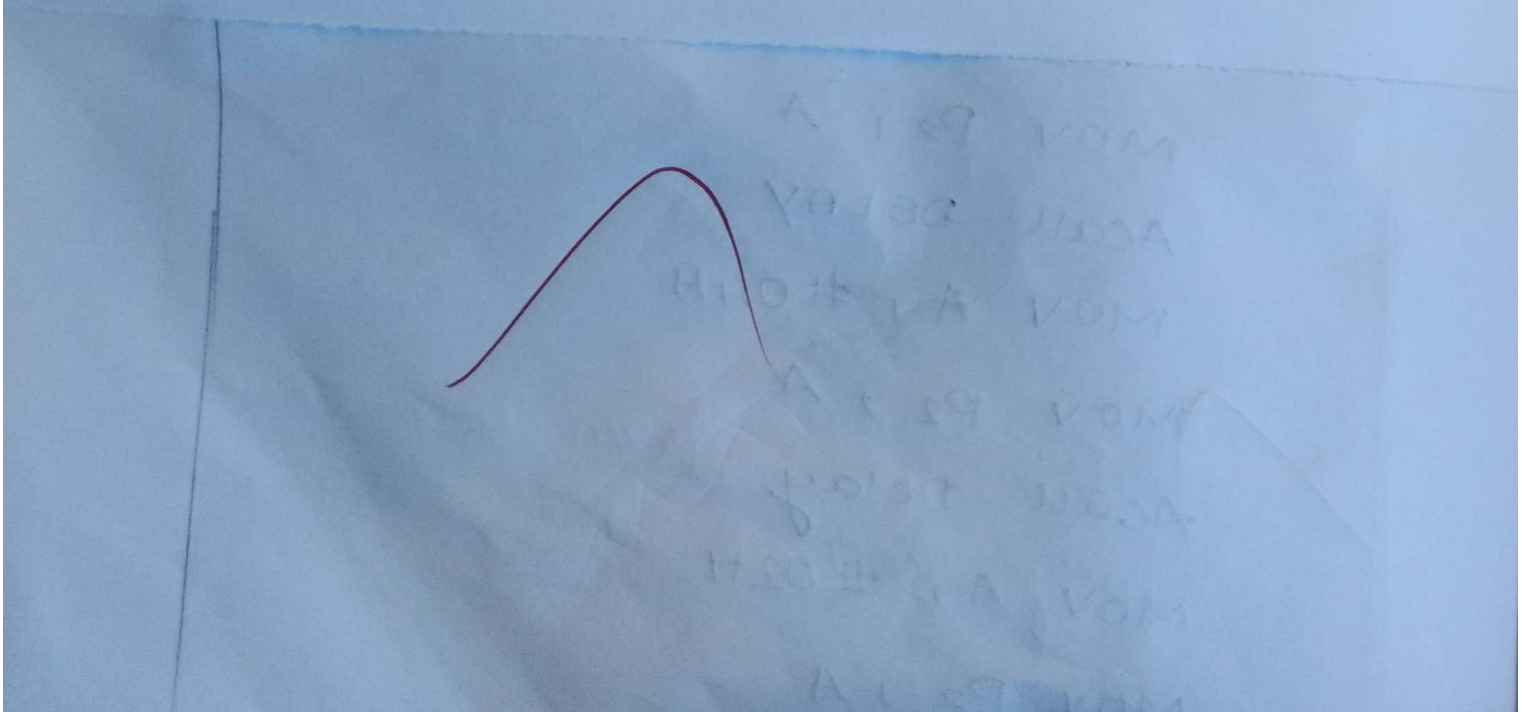
Proteus 8 software

program :-

```
        ORG 00H
MAIN:   MOV P2, #0F0H
        ACALL counterclockwise
        ACall Delay
        SJMP MAIN
counterclockwise:
        MOV A, #08H
        MOV P2, A
        ACall DELAY
        MOV A, #04H
        MOV P2, A
        ACall Delay
        MOV A, #02H
        MOV P2, A
```

ANTICLOCK WISE STEPPER MOTOR





```
    Acall Delay
    MOV A, #01H
    MOV P2, A
    Acall Delay.
    RET.

Delay:
    MOV R1, #0FFH
    Delay-Loop1:
        MOV R2, #0FFH.
    Delay-loop2:
        DJNZ R2, Delay-loop2
        DJNZ R1, Delay-Loop1
    RET
END
```

Result :-

Thus, the program has been Successfully verified and executed.

Experiment - 12

Interfacing of Relay and LED with 8051 using proteus.

AIM :-

to write an assembly language program to interface relay and LED with 8051 using proteus.

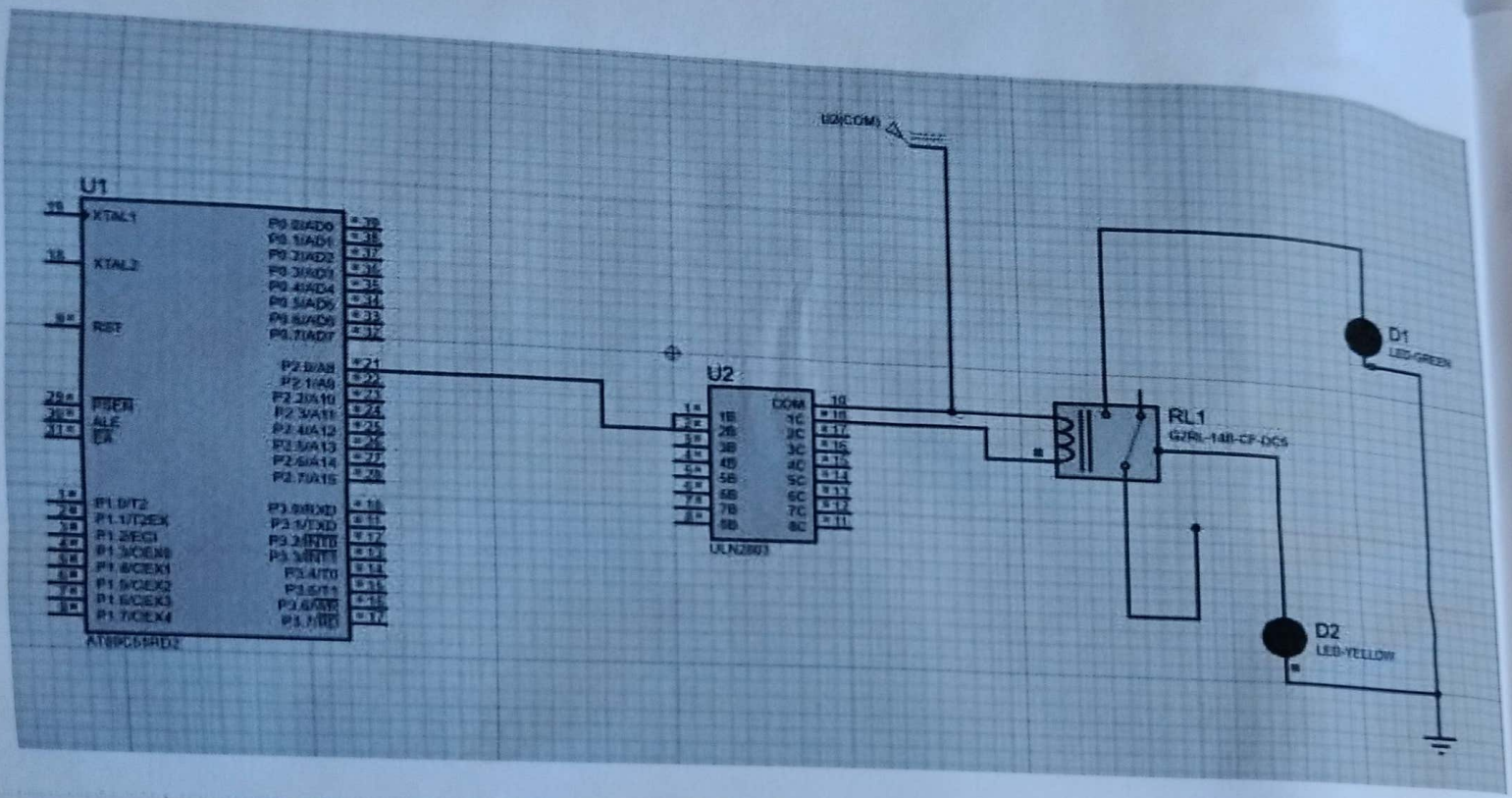
Software Required :-

Proteus 8 Software.

program :-

```
        ORG  0000H
        MOV  P1, #00H
        Main_loop:
          SETB  P1.0.
          ACall Delay.
          CLR  P1.0
          Acall Delay
          SJMP main_loop
      Delay:
          MOV R1, #255
          Delay1:
              MOV R2, #255
          Delay2:
              DJNZ  R2, Delay2
              DJNZ  R1, Delay1
          RET
      END
```

Interfacing of Relay and LED



Output :-

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

The relay act as a switch controlled by the 8051 microcontroller, turning the LED on where P1.0 is high and off when P1.0 is low.

Result :-

Thus, the program has been successfully verified and executed.

Experiment – 13

Digital Thermometer using Keil Software.

Aim :-

To write an assembly language program for Digital Thermometer using 8051 using Keil and proteus.

Software Required :-

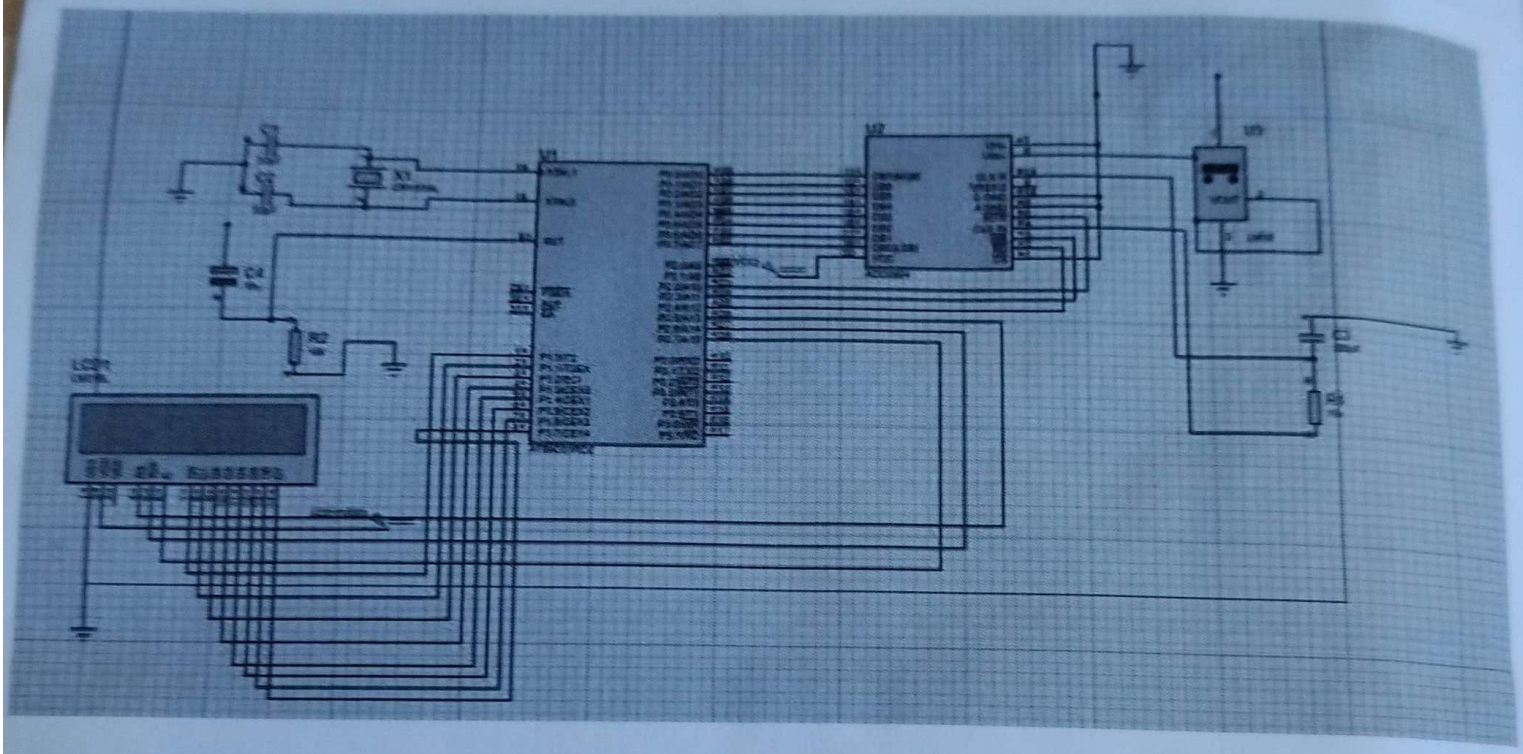
Proteus 8 software.
Keil Software.

Program :-

```
        ORG  0000H
LCD_DATA  EQU  P1
Input     EQU  P0
RD        BIT  P2.4
WR        BIT  P2.3.
INTR      BIT  P2.2
RS        BIT  P2.5
RW        BIT  P2.6
E         BIT  P2.7

DELAY:  MOV  R2, #250.
DELAY_LOOP1: MOV  R1, #250
DELAY_LOOP2: DJNZ R1, DELAY_L2
        DJNZ  R2, DELAY_L1
        RET
```

Digital Thermometer.



```
LCD_CMD:
        MOV   LCD_DATA, A
        CLR   RS
        CLR   RW
        SETB  E
        ACALL DELAY
        CLR   E
        RET
; Subroutine.
LCD_DATA_write;
        MOV   LCD_DATA, A
        SETB  RS
        CLR   RW
        SETB  E
        ACALL DELAY
        CLR   E
        RET
ADC_READ:
        SETB  RD
        CLR   WR
        ACALL DELAY
        SETB  WR
WAIT_INTR:
        JB INTR, wait_intr
        CLR RD
        MOV A, input
        SET B INTR
        RET
END
```

Result :-

Thus the Program has been successfully verified and executed.

Experiment - 14

Digital clock ON LCD

AIM :-

To write an assembly language program to display digital clock on LCD using proteus.

Software Required :-

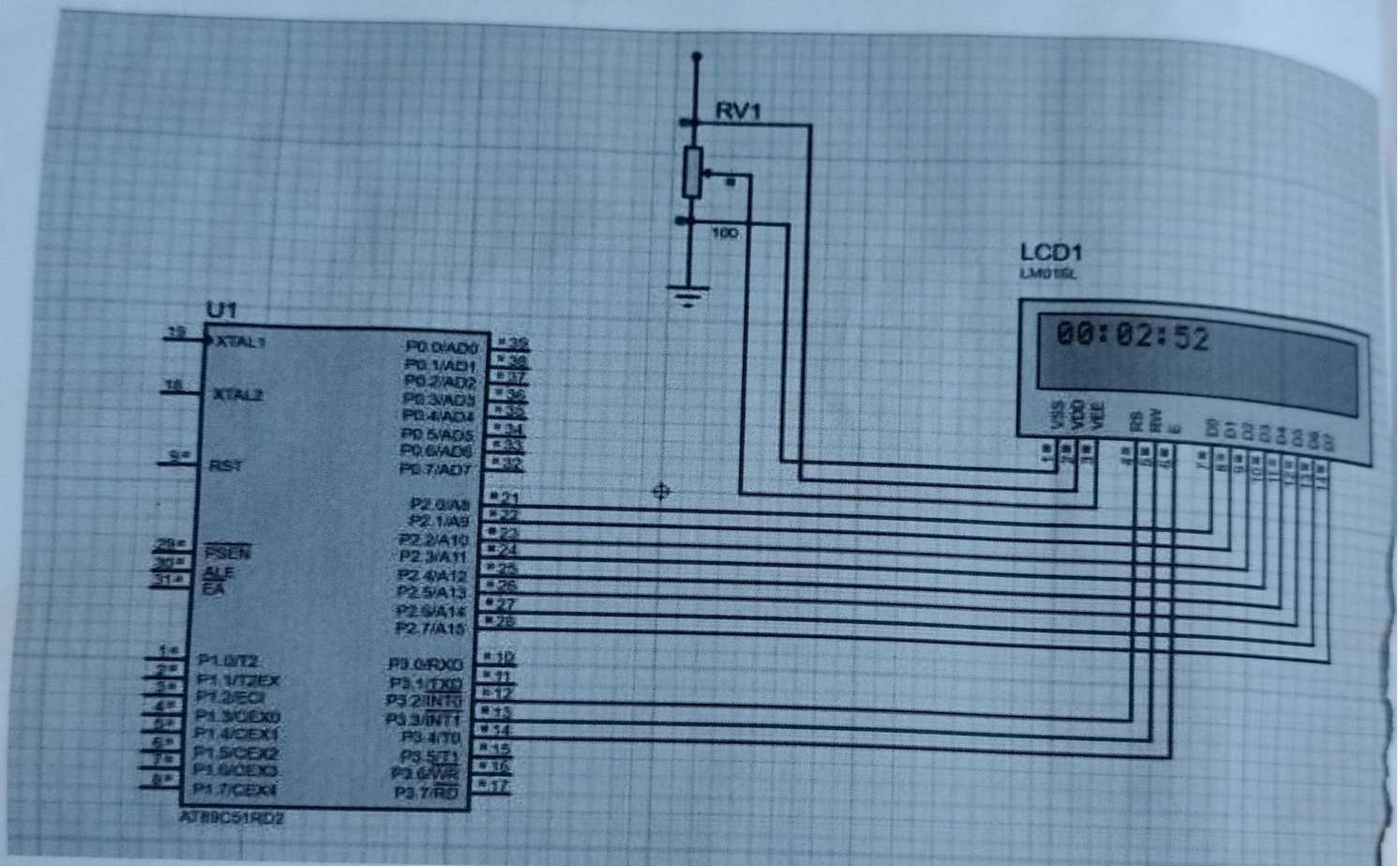
proteus software.

program :-

```
ORG 0000H
MOV R7 , #00H
MOV R6, #00H
MOV R5 , #00H
ACALL INIT_LCD
MAINLOOP :
ACALL UPDATE_LCD.
ACALL DELAY_1_SEC
ACALL INCREMENT_
                TIME
SJMP MAIN_LOOP

INIT_LCD :
MOV A, #38H
ACALL CMD_write
ACALL Delay_short
MOV A , #0CH
ACall CMD_write
ACall CMD_short
mov A , #01H.
ACall CMD_write
ACall Delay_short
RET
Increment_time
INC R5
CJNE R5 , #60,
MOV R5 , #00H
INC R6
CJNE R6 , #60.
MOV R7 , #00H
DONE_SEC :
RET
CMD_write :
MOV P2 , A
CLR P3.2
```

Digital clock.



```
CLR P3.3
SETB P3.4
NOP
CLR P3.4
ACALL DELAY_Short
RET
DELAY_Short:
    MOV R0, #250
Delay_Short_loop:
    DJNZ R0, Delay_Short_
                loop.
    RET
DELAY_1_SEC:
    MOV R3, #50.
DELAY_Loop:
    MOV R4, #255
DELAY_Loop_inner:
    DJNZ R4, DELAY_
        Loop_inner.
    DJNZ R3, DELAY_
            Loop.
    RET
    END.
```

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

Experiment-01

Write and Execute C-program to blink LED's using software delay routine in LPC 2148 kit.

Aim:-

To write and execute a C-program to blink LEDS using software delay routine in LPC 2148 kit.

Apparatus Required:-

Keil uVision5 Software
philips flash programmer

program:-

```
#include "lpc214x.h"
void delay (unsigned int k);
void main(void)
{
  IODIR0 = 0XFFFFFFFF;
  PINSEL0 = 0;
  while(1)
  {
    IOSET0 = 0x0000ff00;
    delay(1000);
    IOCLR0 = 0x0000ff00;
    delay(1000);
  }
}
void delay(unsigned int k)
{
  unsigned int i, j;
```

```
    for(j=0; j<k; j++)
        for(i=0; i<=800; i++);
}
```

Output :-

LEDS P0.15 - P0.8 are blinking.

Result :-

Thus the C-program to blink LEDS using software delay routine was written and executed in LPC 2148 Kit.

Experiment - 02

Write and execute c program to read the switch and display in the LEDS using LPC2148 kit

AIM:-

To write and execute c-program to read the switch and display in the LEDS using LPC2148 kit.

Apparatus Required:-

Keil uVision5 software
philips flash programmer
LPC 2148 kit.

program:-

```
#include "lpc214x.h"
int main(void)
{ unsigned int sw_sts;
  IODIR0 = 0x0000ff00;
  port 0
     PINSEL0=0;
  while(1)
  { sw_sts = IOPIN0;
    IOSET0 = 0x0000ff00;
    P0.8 to '1'
    IOCLR0 = sw_sts >> 8;
    P0.8 to '0'
  }
}
```

Output :-

LED's P0.15 - P0.08 displayed the bits entered in the switches.

Result :-

Thus C program was written read the switch and display in the LEDS using LPC 2148 kit.

Experiment - 03

Write and Execute C program to display a Number in Seven segment LED in LPC2148 Kit.

AIM:-

To write and execute C program to display a Number in Seven segment LED in LPC2148 Kit.

Apparatus Required:-

Keil uVision5 Software
Philips flash programmer
LPC 2148 kit.

Program:-

```
#include <LPC214X.H>
unsigned char const seg-dat[] =
{ 0x3F, 0x6, 0x5B, 0x4F, 0x66,
0x6D, 0x7D, 0x7, 0x7F, 0x67};
void delayms (int n)
{ int i, j;
  for (i = 0; i < n; i++)
  { for (j=0; j<5035; j++)
    { ; }
  }
}
int main (void)
{
```

```
unsigned char count;
PINSEL0 = 0;
PINSEL0 = 0;
IODIR0 = SEG-CODE | DS3 | DS4;
IOSET0 = SEG-CODE | DS3;
DS3 display
IOCLR0 = DS4;
count = 0;
IOCLR0 = SEG-CODE;
IOSET0 = seg-dat[count] <<16;
while(1)
{   delayms(1000);
    count ++;
    if (count >9) count=0;
    IOCLR0 = SEG-CODE;
    IOSET0 = seg-dat[count]<<16;
}
```

Output :-

7-segment display counting from 0 to 9.

Result :-

Thus C-program, was written and executed to display a Number in Seven segment LED in LPC2148 kit.

Experiment - 04

write and execute c-program for serial transmission and reception using on-chip UART in LPC2148 kit.

AIM:-
To write and execute c-program for serial transmission and reception using on-chip UART in LPC2148 kit.

Apparatus Required :-

Keil uVision5 software
philips flash programmer
LPC 2148 kit.

program :-

```
#include <lpc214x.h>
void UART0-init(void)
{ PLL0CON = 0;
  PLL0FEED = 0xAA;
  PLL0FEED = 0x55;
  VPBDIV = 1;
  PINSEL0 = 0x5;
  U0FCR = 0;
  U0LCR = 0x83;
  U0DLL = 0x27;
  U0DLM = 0;
  U0LCR = 3;
}
```

```
void sout (unsigned char dat)
{ while (!(U0LSR & 0x20));
  Buffer empty
  U0THR = dat;
}
int main (void)
{ int dat;
  UART0_Init();
  do {
      if (U0LSR & 1) {
   dat = U0RBR;
   Serial port
   sout (dat);
  }
 } while (1);
}
```

Output :-

Data was serially transmitted

Result :- Thus a C program was written and executed for serial transmission and reception.

Experiment – 05

Write and Execute C-program for accessing an internal ADC and display the binary output in LEDs in LPC2148 Kit.

Aim:-

To write and Execute c-program for accessing an Internal ADC and display the binary output in LEDs in LPC2148 Kit.

Apparatus Required:-

Keil UVISION 5 software.
Philips flash programmer
LPC 2148 Kit.

program:-

```
#include <LPC214x.H>
#define LEDS 0XFF<<8.
#define ADO-1 1<<24
#define CLK-DIV 1<<8
#define PDN 1<<21
#define SOC 1<<24
#define BURST 1<<16
#define DONE 1<<31
Void delay (unsigned int K)
{
  unsigned int i, j;
  for (j=0; j<K; j++)
```

```
for (i=0; i<=800; i++);
}
void adc_init()
{ unsigned long int ADC_CH;
    ADC_CH=0|1<<1;
    AD0.1
    ADOCR=80c|PDN|CLK_DIV|
ADC_CH|BURST;
}
unsigned int adc_read (unsigned
char channel).
{   unsigned int aval;
    unsigned long int val;
if (channel==1) val=ADODR1;
else if (channel==2) val=ADODR2;
else if (channel==3) val=ADODR3;
val=val>>6;
val=val & 0x3FF;
aval=val;
return (aval);
int main (void)
{ do{   tp1=adc_read(1);
        tp1=tp1>>2;
        IOSET0=LEDS;
```

```
        IOCLRO = tp1<<8;
        delay(1000);
    }
    while(1);
}
```

Output :-

The potentiometer knob was adjusted to generate Analog input and Digital display is observed.

Result :-

Thus c program was written and executed for accessing an internal ADC and display the binary output in LEDS in LPC2148 kit.

Experiment - 06

Triangle waveform Generation using 10BIT DAC IN LPC2148

Aim:-

To write and execute a program to generate a triangle waveform with 50% delay during cycle using internal 10-bit DAC using LPC2148 ARM Microcontroller

Apparatus Required:-

Keil MVision 5 software
philips flash programmer
LPC 2148 Kit.

program:-

```
#include<LPC214x.h>
#define MSE 14
#define PSEL (1<<5)
#define AOUT (int n)
void delayms (int n)
{  int i, j;
   for(i=0; i<n; i++)
   {  for(j=0; j<5035; j++)
      {  ;
      }
   }
}
int main(void)
{ clock_select(); PINSEL1 = AOut;
  PIN SELO = 0;   while(1) {
                  wave( );
                }
}
```

Result:-

Thus the program has been successfully verified and executed.

Experiment - 07

Square waveform Generation using 10BIT DAC LPC2148.

Aim:- To write and execute c-program to generate a square wave form with 50% duty cycle using internal 10-bit LPC2148 kit.

Apparatus Required:-

Keil Mvision 5 software, Philips flash programmer, LPC2148 kit.

Program:-

```
#include <LPC214x.h>
#define MSE 14
#define PSE 1<<5
#define AOut 1<<19
Void clock-select (void)
{
 PLLOFG = PSELIMSEL;
 PLL OFEED = 0XAA;
 PLL OFEED = 0X55;
 PLLOCON = 3;
 PLLofeed = 0XAA;
 PLLofeed = 0X55;
 VPBDIV = 0;
}
while(1)
{ DACR = 0X3FF<<6;
  }
}
```

Result:-

Thus the program has been Successfully Verified and executed.

Experiment - 01

BLINKING OF AN LED WITH ARUINO.

AIM:- To interface an LED with Arduino and write a program to make the LED blink at defined intervals.

Components Required :-

* PC or Laptop
* Arduino UNO
* 220 Ω resistor
* LED
* Bread board
* connecting wires.

Software Required :-

Arduino IDE

Connections :-

→ Connect the longer leg of the LED, that is the anode, to the digital pin 13 in the Arduino, through a 220 Ω resistor.

→ Connect the shorter leg of the LED, that is the cathode, to the ground (GND) pin on the Arduino.

Procedure :-

→ Connect the components on a breadboard as a mentioned above and connect the Arduino to the PC / laptop.

→ open Aurdino IDE and open a New editor.

→ Enter the required code save it as a ".ino" file.

→ In the tools menu go to the "port" tab and select the appropriate port.

→ upload the program into the Arduino and verify the output.

Program :-

```
void setup() {
  pinmode (LED_BUILTIN, output);
}
void loop(){
digital write (led_builtin, high);
delay (1000);
digital write (led_builtin, low);
delay (1000);
}
// externally connect to LED Blink
int led = 13;
void setup() {
    pinmode (led, output);
}
```

```
void loop()
{ digitalwrite(led, high);
  delay(1000);
  digitalwrite(led, low);
  delay(1000);
}
```

Result:-

A program has been written to make the built-in and externally connected LED's blink and the output verified.

EXPERIMENT - 02

Interfacing a Buzzer with Arduino UNO

AIM:- To write and execute c program for interfacing a buzzer with Arduino UNO.

Apparatus Required:-

- Arduino UNO Board
- Buzzer
- USB Cable
- Arduino UNO software.

program:-

```
const int buzzerpin = 8;
void setup() {
  pinmode(buzzerpin, output); }
Void loop() {
  digitalwrite (buzzerpin, high);
  delay(500);
  digital write (buzzerpin, low);
  delay (500);
}
```

Output:-

The buzzer emits a continuous "beep" sound.

Result:-

Thus the program has been successfully verified and executed.

EXPERIMENT - 03

Fading of an LED using Arduino UNO

AIM :-

To write and execute a program for fading of an LED using Arduino UNO.

Apparatus Required :-

* Arduino UNO Software
* Arduino UNO Board
* Bread board
* Connecting wires
* LED
* 220 Ω Resistor

program :-

```
int led = 9;
int brightness =0;
int fadeamount= 5;
void setup() {
pinmode (led, output); }
void loop () {
analogwrite (led, brightness);
brightness = brightness + fadeamount;
if (brightness <= 0 || brightness >= 255)
        fadeamount = - fadeamount;
}       delay(30);
```

Output :-

LED is fading In and out

Result :-

Thus the program has been successfully verified and executed.

Experiment - 04

Interfacing of water level sensor with Arduino UNO :-

Aim :-

To write and Execute a program for interfacing of water level sensor with Arduino UNO.

Apparatus Required :-

Arduino uno software, water level sensor, Arduino uno Board, Bread Board, connecting wires.

program :-

```
int sensorpin = A3;
int sensorvalue = 0;
int value;
void setup() {
  serial.begin (9600);
  pinmode (sensor pin, Input);
}
void loop() {
  value = analogRead (sensor pin);
  if (value <=480) {
serial.println("water level : 0mm -
Empty!"); }
  else if (value > 480 && value <= 530)
    serial.println("water level : 16
mm to 5mm");
  else if (value > 530 && value <= 615)
```

```
Serial.println("water level : 5mm to 10mm");
else if (value > 615 && value <= 660)
  Serial.println("water level: 10mm to 15mm");
else if (value > 660 && value <= 680) {
  Serial.println("water level : 15mm to 20mm");
}
else if (value > 705) {
  Serial.println("water level : 35mm to
40mm");
}
delay (2000);
}
```

output :-

The work level is high and low based on the water reaching the sensor's detection point.

Result :-

Thus the program has been successfully verified and executed.

Experiment - 05

MQ-6 GAS SENSOR INTERFACING WITH ARDUINO UNO

AIM :-

To write and execute a program for MQ-6 Gas Sensor Interfacing with Arduino UNO.

Apparatus Required :-

Arduino UNO Software
Arduino UNO Board
MQ-6 GAS Sensor
Bread Board
connecting wires.

Program :-

```
const int gasSensorPin = A0;
void setup(){
    Serial.begin(9600); }

    void loop(){
int sensorValue = analogRead(gasSensorPin);
Serial.print("Gas: level: ");
Serial.print(sensorValue);
delay(1000);
}
```

Output :-

The Gas level depend on the concentration of gas in the Environment.

Result :-

Thus the program has been successfully verified and executed.

Experiement - 06.

Interfacing an ultrasonic sensor with Arduino UNO.

Aim:-

To write and execute a program for interfacing an ultrasonic sensor with Arduino UNO.

Apparatus Required:-

Arduino UNO, Board, ultrasonic sensor, Bread Board, connecting wires.

program:-

```
const int trigpin = 2;
const int echopin = 3;
const int buzzerpin = 8;
float length = 0;
const float thresholdDistance = 30.0;
void setup() {
  Serial.begin(9600);
  Pinmode (trigpin, output);
  pinmode (echopin, Input);
  pinmode (buzzerpin, output);
}
void loop() {
  digitalwrite (trigpin, low);
  delaymicroseconds (2);
  digitalwrite (trigpin, high);
  delaymicroseconds (10);
  digitalwrite (trigpin, low);
```

```
    length = duration * 0.0343/2;
    Serial.print("Distance: ");
    Serial.print(length);
    Serial.println(" cm");
  if (length < thresholdDistance) {
    tone(buzzerPin, 1000);
    delay(200);
    noTone(buzzerPin);
  }
  delay(500);
}
```

Output:-

The Distance values will change dynamically based on the distance of the object from the sensor.

Result:-

Thus the program has been successfully verified and executed.