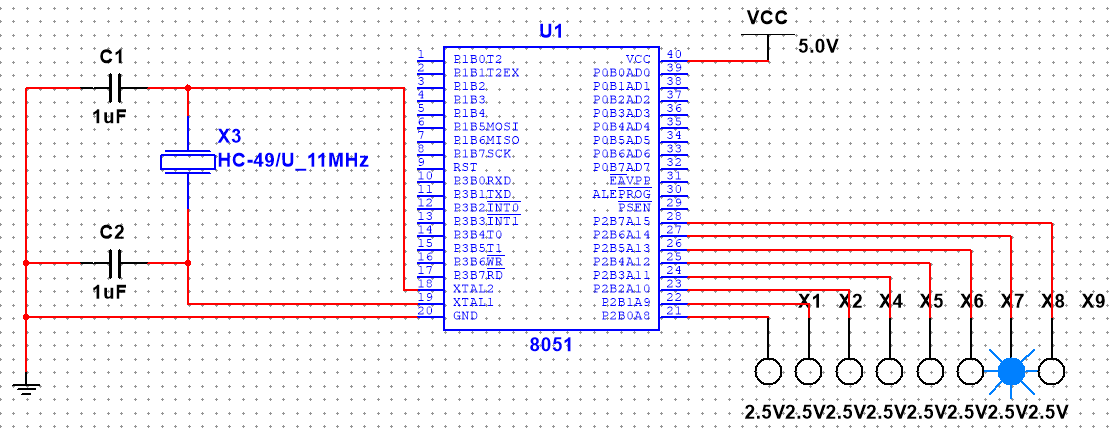
|  |  |
| --- | --- |
| **1** | **LED** |
| **2** | **SGD** |
| **3** | **LCD** |
| **4** | **Keypad** |
| **5** | **PIR done** |
| **6** | **Variable Resistor** |
| **7** | **RGB LED** |
| **8** | **Ultrasonic** |
| **9** | **Soil Moisture** |
| **10** | **Servo** |
| **11** | **DC motor** |
| **12** | **LM35** |
| **13** | **LDR** |
| **14** |  |
| **15** | **12C** |

**Perform rotation operation:**



A diagram of a circuit board

Description automatically generated

**Assembly code:**

$MOD51

ORG 0000h

MOV A, #001h

start:

MOV P2, A

ACALL delay

RR A

ACALL delay

SJMP start

delay:

MOV R0, #010h

MOV R1, #0ffh

back:

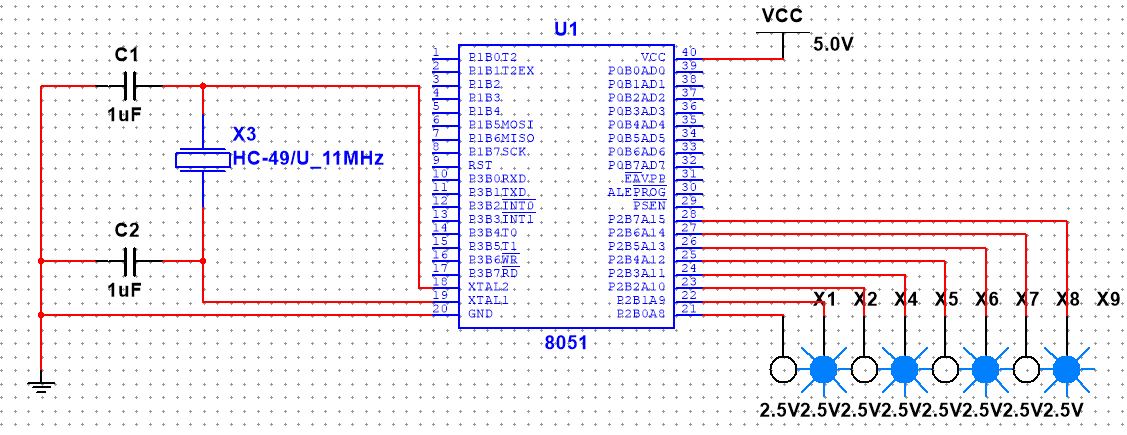
DJNZ R1, back

DJNZ R0, back

ret

END

**Perform four on and off**



A diagram of a circuit board

Description automatically generated

**Assembly Code:**

$MOD51 ; This includes 8051 definitions for the Metalink assembler

; Please insert your code here.

org 0000h;

mov A,#0F5h

mov A,#055h

start:

mov p2,A

acall delay

cpl A

sjmp start

delay:

mov r0,#010h

mov r1,#0ffh

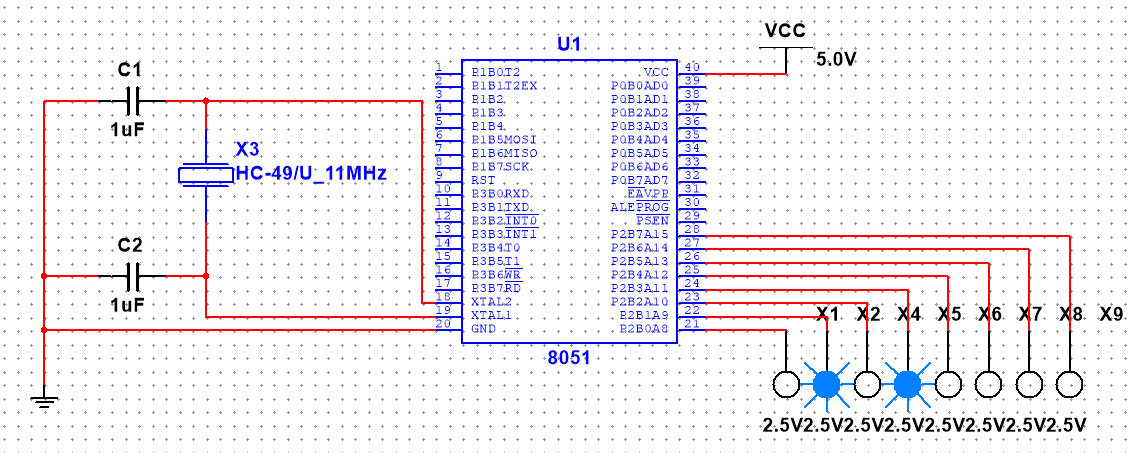
back: djnz r1,back

djnz r0, back

ret

END

**Two on and six off:**



**Assembly Code**

$MOD51 ; This includes 8051 definitions for the Metalink assembler

; Please insert your code here.

org 0000h;

mov A,#0F5h

mov B,#055h

start:

mov p2,A

acall delay

cpl A

sjmp start

delay:

mov r0,#010h

mov r1,#0ffh

back: djnz r1,back

djnz r0, back

ret

END

**C Code:**

**Perform rotation operation:**

#include <8051.h>

void main(){

while(1){

P2 = #0AAH;

Delay();

P2 = #055H;}

}

void Delay(){

for(int i=0;i<255;i++);

}

**Perform four on and off**

#include <8051.h>

void delay() {

unsigned int i, j;

for(i = 0; i < 500; i++)

for(j = 0; j < 1000; j++);

}

void main() {

unsigned char pattern = 0x0F;

while(1) {

P1 = ~pattern;

P2 = ~pattern;

P3 = ~pattern;

delay();

P1 = 0xFF;

P2 = 0xFF;

P3 = 0xFF;

delay();

pattern = ~pattern; // Toggle pattern (four on to four off)

}

}

**Two on and six off:**

#include <8051.h>

void delay() {

unsigned int i, j;

for(i = 0; i < 500; i++)

for(j = 0; j < 1000; j++);

}

void main() {

unsigned char pattern = 0x03; // Initial pattern (two LEDs on, six off)

while(1) {

P1 = ~pattern;

P2 = ~pattern;

P3 = ~pattern;

delay();

P1 = 0xFF; // Turn off LEDs on P1

P2 = 0xFF;

P3 = 0xFF;

delay();

}

}

**Program of LED through switch**

#include<reg51.h>

sbit sw=P1^0;

void main()

{ while(1)

{

if( sw == 0)

{

P2 = 0xFF;

}

else

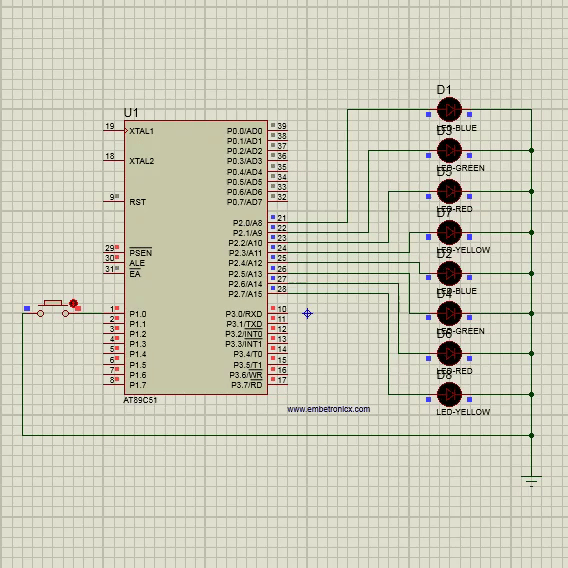
{

P2 = 0x00;

}

}

}



**Seven Segment Display:**

1. **Draw the Block diagram of seven segment display on Multisim/Proteous and write its code in C and Assembly**



**C-Code:**

#include <htc.h>

#define Port\_1 P2

void delay(int ms) {

int i, j;

for (i = 0; i < ms; i++) {

for (j = 0; j < 100; j++);

}

}

void display(int a){

switch(a){

case 0:

Port\_1=0x40;

break;

case 1:

Port\_1=0x79;

break;

case 2:

Port\_1=36;

break;

case 3:

Port\_1=48;

break;

case 4:

Port\_1=0x19;

break;

case 5:

Port\_1=0x12;

break;

case 6:

Port\_1=0x02;

break;

case 7:

Port\_1=0x78;

break;

case 8:

Port\_1=0x00;

break;

case 9:

Port\_1=0x10;

break;

}

}

**Draw block diagram of two seven segment display. Write a program in Assembly and C to display a digit from 0 to 9 in both seven segment display**

****

**C Code:**

#include <htc.h>

#define Port\_0 P0

#define Port\_1 P2

void delay(int ms) {

int i, j;

for (i = 0; i < ms; i++) {

for (j = 0; j < 100; j++);

}

}

void display(int a){

switch(a){

case 0:

Port\_0=0x40;

Port\_1=0x10;

break;

case 1:

Port\_0=0x79;

Port\_1=0x00;

break;

case 2:

Port\_0=36;

Port\_1=0x78;

break;

case 3:

Port\_0=48;

Port\_1=0x02;

break;

case 4:

Port\_0=0x19;

Port\_1=0x12;

break;

case 5:

Port\_0=0x12;

Port\_1=0x19;

break;

case 6:

Port\_0=0x02;

Port\_1=48;

break;

case 7:

Port\_0=0x78;

Port\_1=36;

break;

case 8:

Port\_0=0x00;

Port\_1=0x79;

break;

case 9:

Port\_0=0x10;

Port\_1=0x40;

break;

}

}

**LCD**

**C-Code:**

#include <LiquidCrystal.h>

LiquidCrystal lcd(10,9,8,7,6,5);

void setup() {

  lcd.begin(16, 2);

}

void loop() {

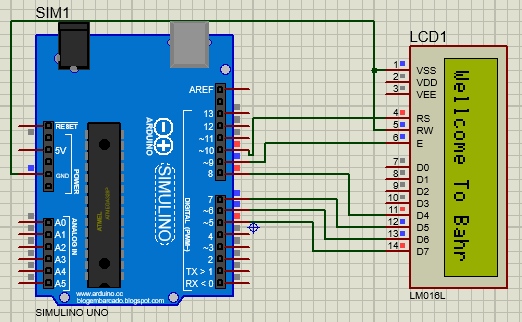
  lcd.setCursor(0, 1);

  lcd.print("Wellcome To Bahria");

}

**Output:**

**CIRCUIT:**



**LCD WITH KEYPAD**

**Diagram:**

**Variable Resistor/ Potentiometer**

#include <LiquidCrystal.h>

// Initialize the LCD with the specified pins

LiquidCrystal lcd(13, 12, 5, 4, 3, 2);

void setup() {

  lcd.begin(16, 2);  // Initialize the LCD with 16 columns and 2 rows

  lcd.print("Voltage: ");

}

void loop() {

  int sensorValue = analogRead(A0); // Read analog value from the potentiometer

  float voltage = sensorValue \* (3.75 / 1023.0); // Convert analog value to voltage

  lcd.setCursor(9, 0);

  lcd.print(voltage, 2); // Display voltage

  float error = 3.75 - voltage; // Calculate error

  lcd.setCursor(0, 1);

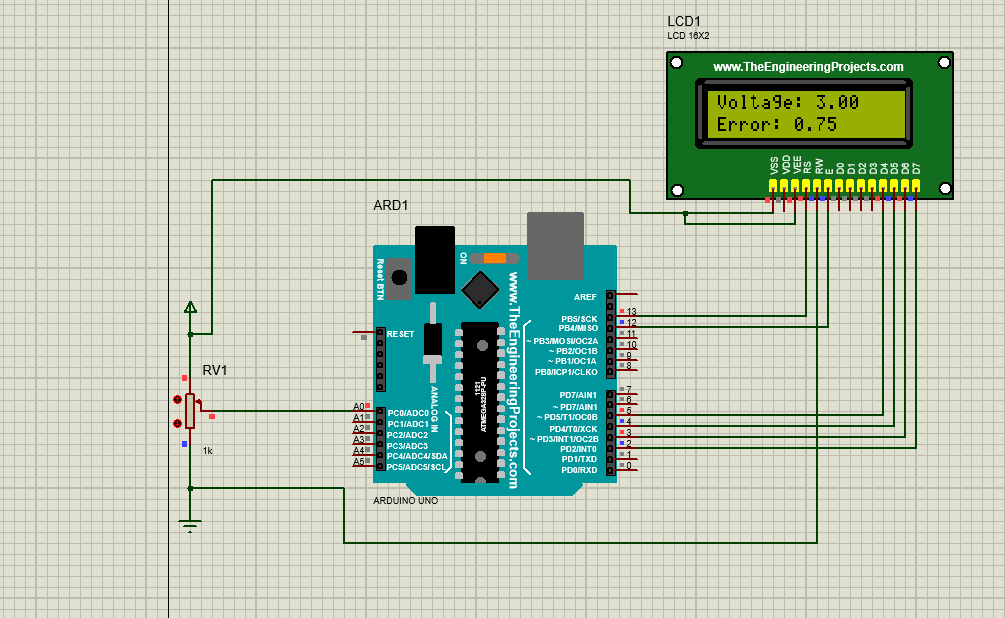
  lcd.print("Error: ");

  lcd.print(error, 2); // Display error

  delay(1000); // Wait for a second

}

**OUTPUT:**



**A computer screen shot of a circuit board

Description automatically generated**

<LiquidCrystal.h>

#include <Keypad.h>

LiquidCrystal lcd(13, 12, 11, 10, 9, 8);

const byte ROW\_NUM    = 4;

const byte COLUMN\_NUM = 4;

char keys[ROW\_NUM][COLUMN\_NUM] = {

  {'1','2','3'},

  {'4','5','6'},

  {'7','8','9'},

  {'\*','0','#'}

};

byte pin\_rows[ROW\_NUM] = {4, 3, 2, 1};

byte pin\_column[COLUMN\_NUM] = {7, 6, 5};

Keypad keypad = Keypad(makeKeymap(keys), pin\_rows, pin\_column, ROW\_NUM, COLUMN\_NUM);

void setup() {

  lcd.begin(16, 2);

  lcd.setCursor(0,0);

  lcd.print("LCD is");

  lcd.setCursor(0,1);

  lcd.print("running");

}

void loop() {

  char key = keypad.getKey();

  if (key == '0') {

    displayStudentInfo();

    delay(1000);

  }

}

void displayStudentInfo() {

  lcd.clear();

  lcd.setCursor(0, 0);

  lcd.print("0 is");

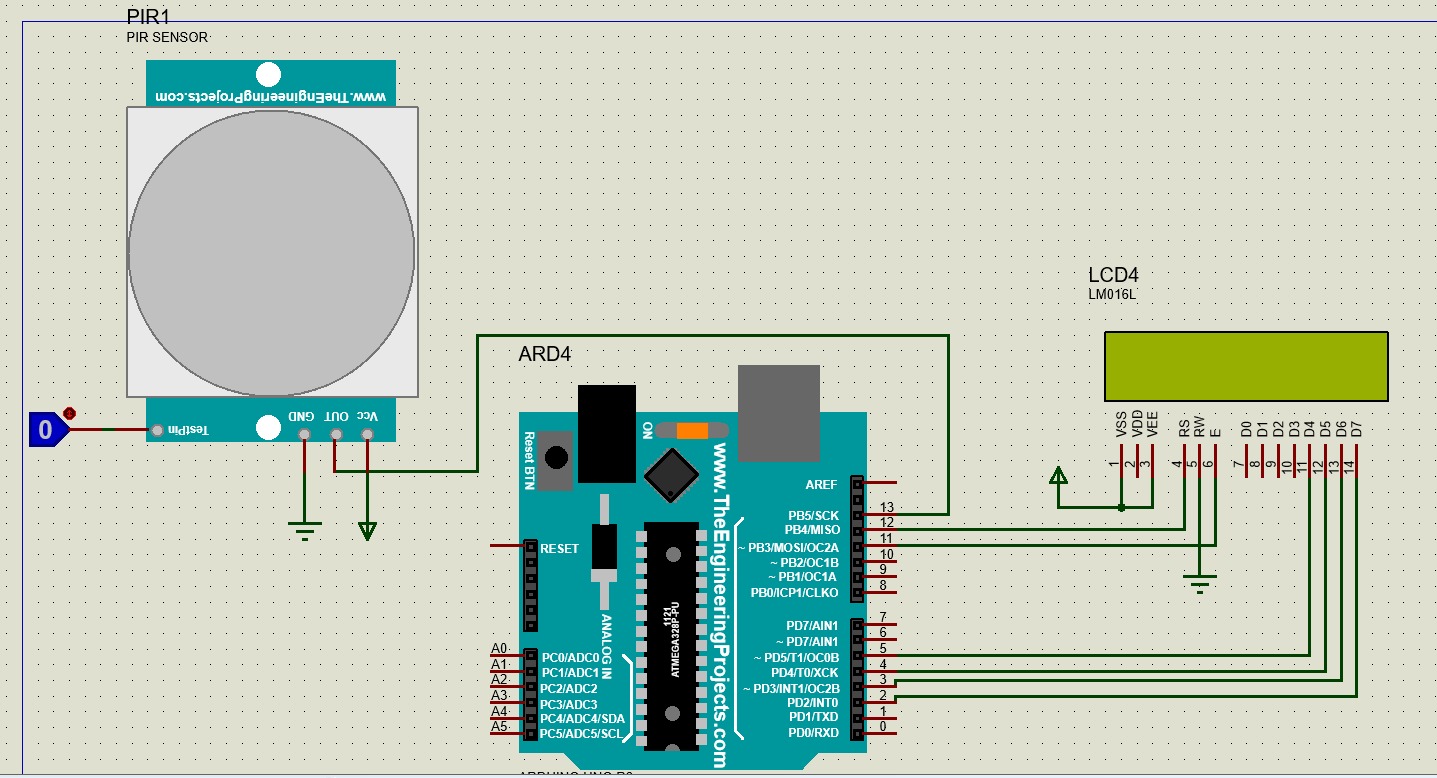
  lcd.setCursor(0, 1);

  lcd.print("Pressed!");

}

**PIR Sensor**

**Circuit Diagram:**



**Code:**

#include <LiquidCrystal.h>

// const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;

LiquidCrystal lcd(12,11,5,4,3,2);

const int pirPin = 13;

void setup() {

  lcd.begin(16, 2);

  pinMode(pirPin, INPUT);

}

void loop() {

  int motionStatus = digitalRead(pirPin);

  lcd.clear();

  lcd.setCursor(0, 0);

  if (motionStatus == HIGH) {

    lcd.print("Motion Detected");

  } else {

    lcd.print("Motionless");

  }

  delay(1000);

}

**RGB LED (Increase Intensity)**

const byte ledPin = 3; // red RGB LED on Experimenter

const byte FADE\_MAX = 255; // max value for setting duty cycle

const byte FADE\_INC = 5; // increment for changing duty cycle

void setup()

{

pinMode(ledPin, OUTPUT);

}

void loop()

{

int fadeValue; // PWM value

// fade in from min to max in increments of 5 points:

for(fadeValue = 0 ; fadeValue <= FADE\_MAX; fadeValue +=FADE\_INC)

{

analogWrite(ledPin, fadeValue); // sets the value (range from 0 to 255):

}

// fade out from max to min in increments of 5 points:

for(fadeValue = FADE\_MAX; fadeValue >= 0; fadeValue -=FADE\_INC)

{

analogWrite(ledPin, fadeValue); // sets the value (range from 0 to 255):

}

}

**OR**

int redPin= 6;

int greenPin = 5;

int bluePin = 4;

void **setup**() {

pinMode(redPin, OUTPUT);

pinMode(greenPin, OUTPUT);

pinMode(bluePin, OUTPUT);

}

void **loop**() {

setColor(255, 0, 0); *// Red Color*

delay(1000);

setColor(0, 255, 0); *// Green Color*

delay(1000);

setColor(0, 0, 255); *// Blue Color*

delay(1000);

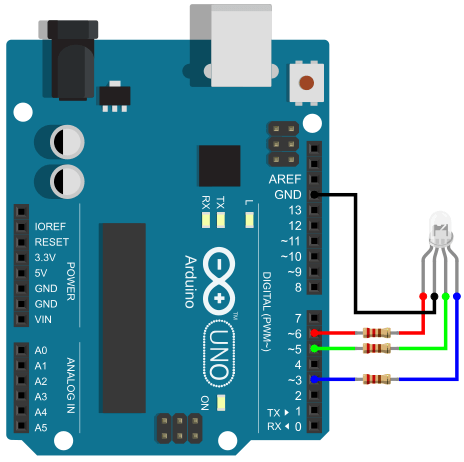
setColor(255, 255, 255); *// White Color*

delay(1000);

setColor(170, 0, 255); *// Purple Color* delay(1000); } void **setColor**(int redValue, int greenValue, int blueValue) { analogWrite(redPin, redValue);

analogWrite(greenPin, greenValue);

analogWrite(bluePin, blueValue); }

****

**CODE: Moisture Sensor**

#include <LiquidCrystal.h>

const int sensorPin = A1;

LiquidCrystal lcd(13, 12, 5, 4, 3, 2);

int sensorValue;

double moisturePercentage;

void setup() {

  lcd.begin(16, 2);

  pinMode(sensorPin, INPUT);}

void loop() {

  sensorValue = analogRead(sensorPin);

  moisturePercentage = map(sensorValue, 0, 1023, 0, 100);

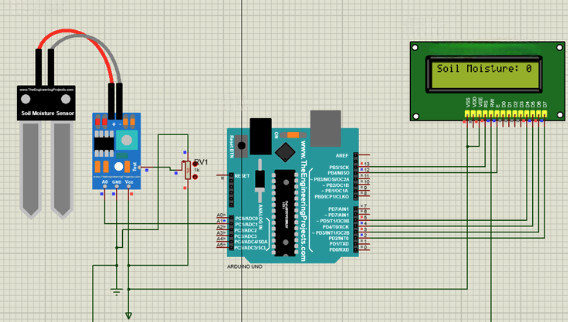
  lcd.clear();

  lcd.print("Soil Moisture: ");

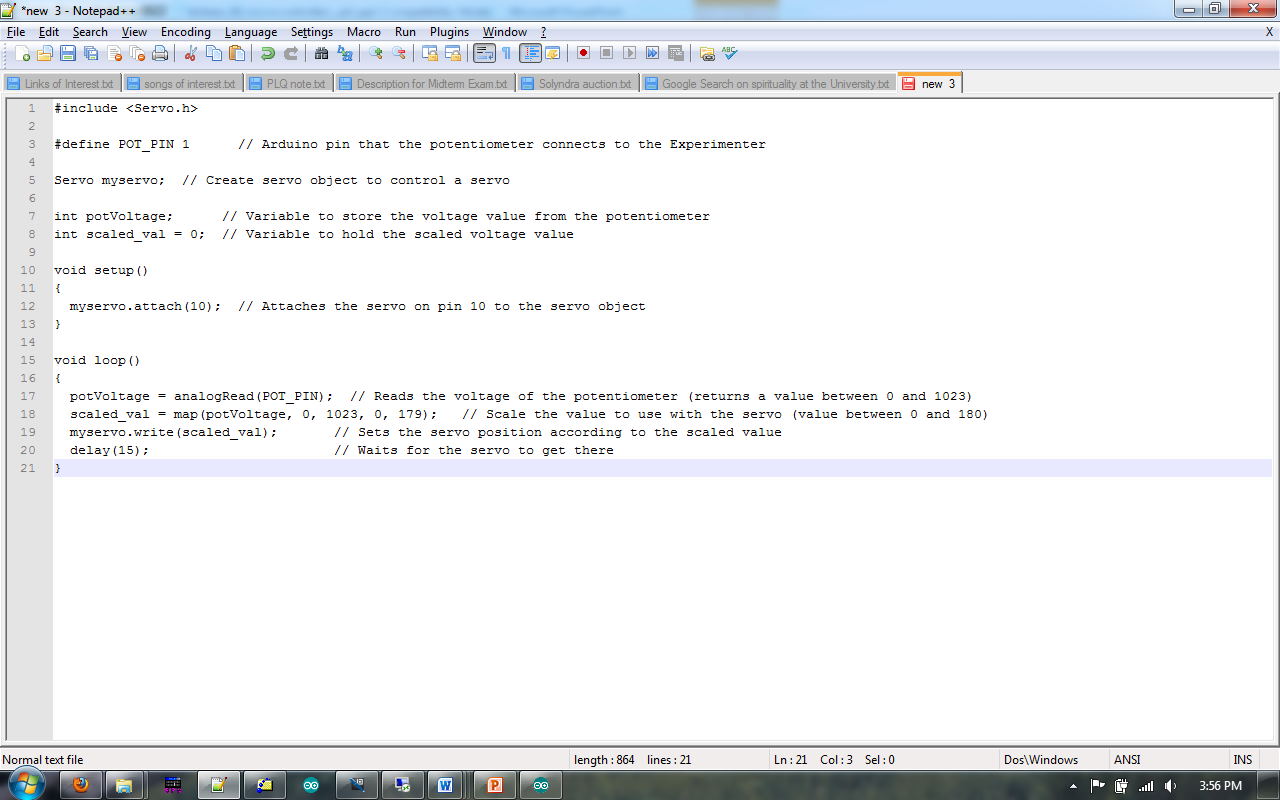
  lcd.print("%");

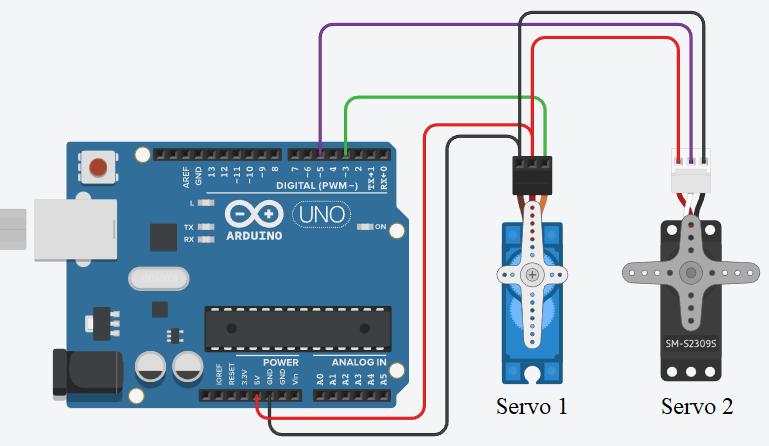
  delay(500);}

**OUTPUT:**

****

**Servo Motor**

****

****

**#include <Servo.h>**

**Servo servo1;  // Create object for Servo motor 1**

**Servo servo2;  // Create object for Servo motor 2**

**int position = 0;     // Variable to store the position**

**void setup()**

**{**

**servo1.attach(3);   // Set PWM pin 3 for Servo motor 1**

**servo2.attach(5);   // Set PWM pin 5 for Servo motor 2**

**}**

**void loop()**

**{**

**// Rotating Servo motor 1 in Anti clockwise from 0 degree to 180 degree**

**for (position = 0; position <= 180; position++)**

**{**

**servo1.write(position);  // Set position of Servo motor 1**

**delay(10);**

**}**

**// Rotating Servo motor 1 in clockwise from 180 degree to 0 degree**

**for (position = 180; position >= 0; position--)**

**{**

**servo1.write(position);  // Set position of Servo motor 1**

**delay(15);               // Short delay to control the speed**

**}**

**// Rotating Servo motor 2 in clockwise from 0 degree to 180 degree**

**for (position = 0; position <= 180; position++)**

**{**

**servo2.write(position);  // Set position of Servo motor 2**

**delay(10);               // Short delay to control the speed**

**}**

**// Rotating Servo motor 2 in Anti clockwise from 180 degree to 0 degree**

**for (position = 180; position >= 0; position--)**

**{**

**servo2.write(position);  // Set position of Servo motor 2**

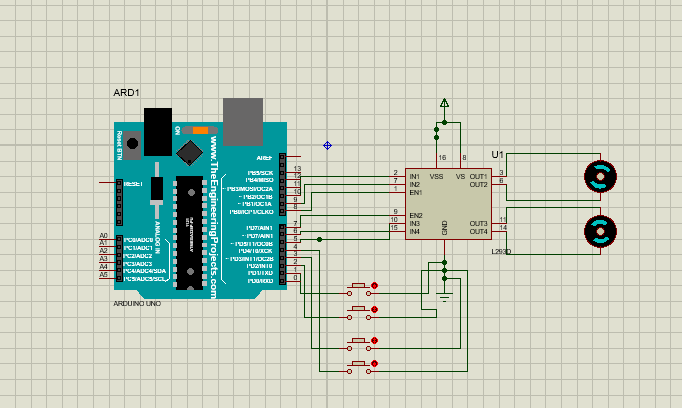
**delay(15);               // Short delay to control the speed**

**} }**

**Simulation**

**A computer circuit board with wires and wires

Description automatically generated with medium confidence**

****

**Code for one DC Motor**

int in1 = 10;

int in2 = 9;

int en = 8;

void setup() {

  pinMode(in1, OUTPUT);

  pinMode(in2, OUTPUT);

  pinMode(en, OUTPUT);

}

void loop() {

  digitalWrite(en, HIGH);

  digitalWrite(in1, HIGH);

  digitalWrite(in2, LOW);

  delay(5000);

  digitalWrite(en, LOW);

  delay(1000);

  digitalWrite(en, HIGH);

  digitalWrite(in2, HIGH);

  digitalWrite(in1, LOW);

  delay(5000);

  digitalWrite(en, LOW);

  delay(1000);

}

**Code 2 DC motor**

// Define motor control pins for left motor

int leftMotorIn1 = 10;

int leftMotorIn2 = 9;

int leftMotorEn = 8;

// Define motor control pins for right motor

int rightMotorIn1 = 6;

int rightMotorIn2 = 5;

int rightMotorEn = 7;

// Define button pins

int forwardButton = 1;

int reverseButton = 2;

int leftButton = 3;

int rightButton = 4;

void setup() {

  // Set up left motor pins

  pinMode(leftMotorIn1, OUTPUT);

  pinMode(leftMotorIn2, OUTPUT);

  pinMode(leftMotorEn, OUTPUT);

  // Set up right motor pins

  pinMode(rightMotorIn1, OUTPUT);

  pinMode(rightMotorIn2, OUTPUT);

  pinMode(rightMotorEn, OUTPUT);

  // Set up button pins

  pinMode(forwardButton, INPUT\_PULLUP);

  pinMode(reverseButton, INPUT\_PULLUP);

  pinMode(leftButton, INPUT\_PULLUP);

  pinMode(rightButton, INPUT\_PULLUP);

}

void moveForward() {

  digitalWrite(leftMotorEn, HIGH);

  digitalWrite(leftMotorIn1, HIGH);

  digitalWrite(leftMotorIn2, LOW);

  digitalWrite(rightMotorEn, HIGH);

  digitalWrite(rightMotorIn1, HIGH);

  digitalWrite(rightMotorIn2, LOW);

}

void moveReverse() {

  digitalWrite(leftMotorEn, HIGH);

  digitalWrite(leftMotorIn1, LOW);

  digitalWrite(leftMotorIn2, HIGH);

  digitalWrite(rightMotorEn, HIGH);

  digitalWrite(rightMotorIn1, LOW);

  digitalWrite(rightMotorIn2, HIGH);

}

void turnLeft() {

  digitalWrite(leftMotorEn, HIGH);

  digitalWrite(leftMotorIn1, LOW);

  digitalWrite(leftMotorIn2, HIGH);

  digitalWrite(rightMotorEn, HIGH);

  digitalWrite(rightMotorIn1, HIGH);

  digitalWrite(rightMotorIn2, LOW);

}

void turnRight() {

  digitalWrite(leftMotorEn, HIGH);

  digitalWrite(leftMotorIn1, HIGH);

  digitalWrite(leftMotorIn2, LOW);

  digitalWrite(rightMotorEn, HIGH);

  digitalWrite(rightMotorIn1, LOW);

  digitalWrite(rightMotorIn2, HIGH);

}

void stopMotors() {

  digitalWrite(leftMotorEn, LOW);

  digitalWrite(rightMotorEn, LOW);

}

void loop() {

  // Check the status of each button

  if (digitalRead(forwardButton) == LOW) {

    moveForward();

  } else if (digitalRead(reverseButton) == LOW) {

    moveReverse();

  } else if (digitalRead(leftButton) == LOW) {

    turnLeft();

  } else if (digitalRead(rightButton) == LOW) {

    turnRight();

  } else {

    stopMotors();

  }

}

**LM35**

int tempPin=A0;

float temp;

#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup()

{

Icd.begin (16, 2);

lcd.print("Temperature=");

}

void loop()

{

temp = analogRead (tempPin);

temp = temp\*0.48828125;

delay (1000);

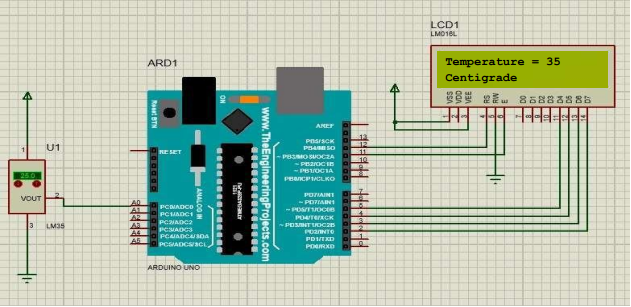
Icd.setCursor (0, 1);

lcd.print(temp);

lcd.print(" Centigrade");

}

**Simulation screenshot :**

****

**To Display Light intensity in Percentage(LDR)**

#include <LiquidCrystal.h>

// Initialize the LCD with the specified pins

LiquidCrystal lcd(13, 12, 5, 4, 3, 2);

const int ldrPin = A0; // LDR connected to analog pin A0

void setup() {

  lcd.begin(16, 2);  // Initialize the LCD with 16 columns and 2 rows

  lcd.print("Light Intensity:");

  pinMode(ldrPin, INPUT);

}

void loop() {

  int ldrValue = analogRead(ldrPin); // Read analog value from the LDR

  float intensity = map(ldrValue, 0, 1023, 0, 100); // Map the value to percentage (0-100%)

  lcd.setCursor(0, 1);

  lcd.print(intensity);

  lcd.print("%"); // Display light intensity percentage

  delay(1000); // Wait for a second

}

A computer screen shot of a circuit board

Description automatically generated