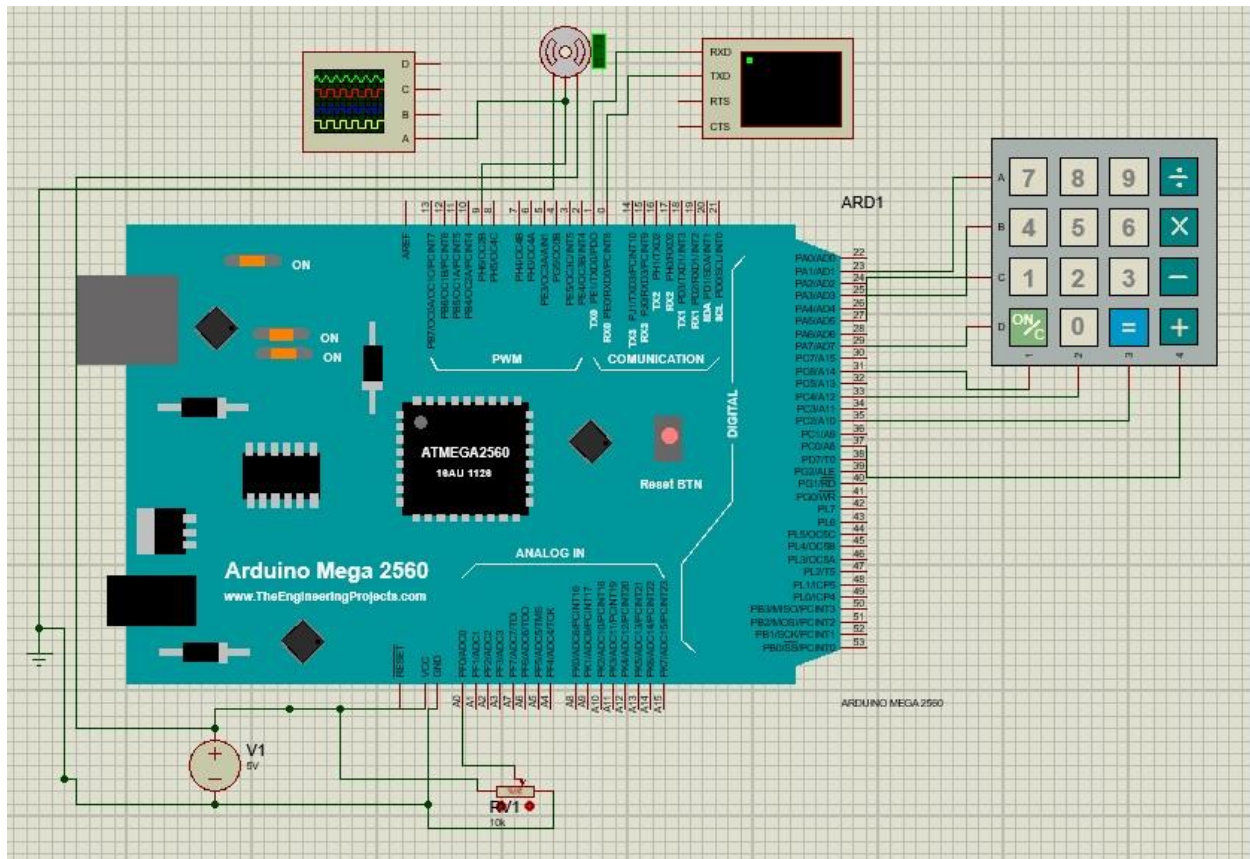


## تمرین ها:

### مدار



-1

کد:

```
#include <Servo.h>

Servo myservo;
int degree;
int steps = 1;

void setup() {
    // put your setup code here, to run once:
    myservo.attach(9); // attaches the servo on pin 9 to the servo object
    degree = 0; //set the degree 0
}

void loop() {
    //
    myservo.write(degree);
    delay(15);
    //we have to loop in range of 0 to 90 degree and then 90 to 0 degree
    //so when the value of variable(degree) become 90 the
    //steps initialize to -1 because we want to decrease and vice versa.
    //value of degree to zero .
    if(degree == 90)
        steps = -1;
    if(degree == 0)
        steps = 1;

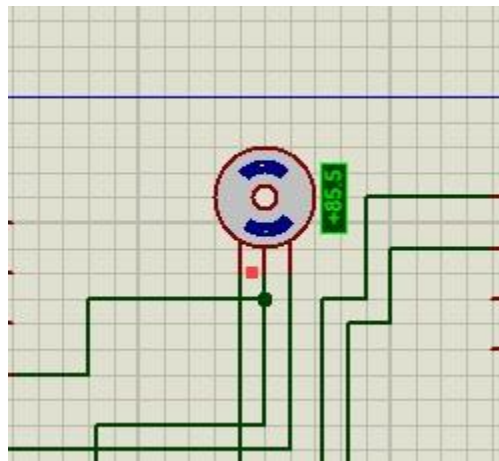
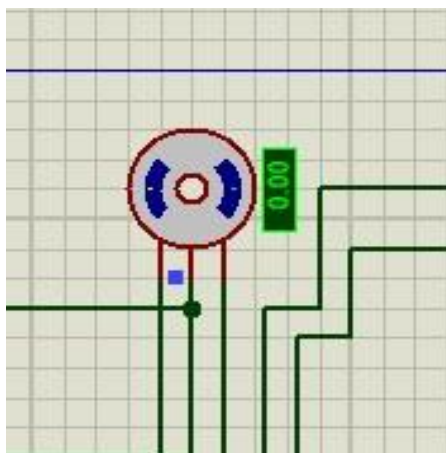
    //sum the value
    degree += steps ;
}
```

ابتدا step را یک در نظر می گیریم

ابتدا مقدار degree را صفر می کنیم.

همواره مقدار degree یکی زیاد می شود تا زمانی که به 90 برسد. هنگامی که به 90 رسید در این مرحله step برابر 1- شده و این بار مقدار degree یکی کم می شود تا به صفر برسد و این روند ادامه دارد(در اصل در یک لوپ قرار دارد)

## خروجی



```

#include <Servo.h>
#include <Keypad.h>

//Setting Keypad configs
const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns
char keys[ROWS][COLS] = {
  {'7','8','9','/'},
  {'4','5','6','*'},
  {'1','2','3','-'},
  {'0','0','=','+'}
};

byte rowPins[ROWS] = {23, 25, 27, 29}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {31,33, 35,37}; //connect to the column pinouts of the keypad

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
//Servo
Servo servo; // create servo object to control a servo

void setup() {
  servo.attach(9,1000,2000); // attaches the servo on pin 9 to the servo object
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  static String degree="";
  char key = keypad.getKey(); //the key that we get from keypad
  if(key) {
    //if we press the "=" it means that the input is ready
    //we have to subtract the degree with 180 because the input degree
    //is 180 degree less than output degree.
    //I divided the value into 2 because servo motor can get the value between 0 to 180
    //and our input is in range of 360.
    if(key == '='){
      int value = degree.toInt() ;
      Serial.println();
      Serial.print("Result: ");
      Serial.println(degree.toInt() - 180);
      value =value/2;
      servo.write(value);
      delay(100);
      degree="";
    }
    if(key >= '0' && key <= '9'){ //if the key input is in range of 0 to 9
      //we have to concat it with previous degree value and print it into the terminal
      Serial.print(key);
      degree += key;
    }
  }
}

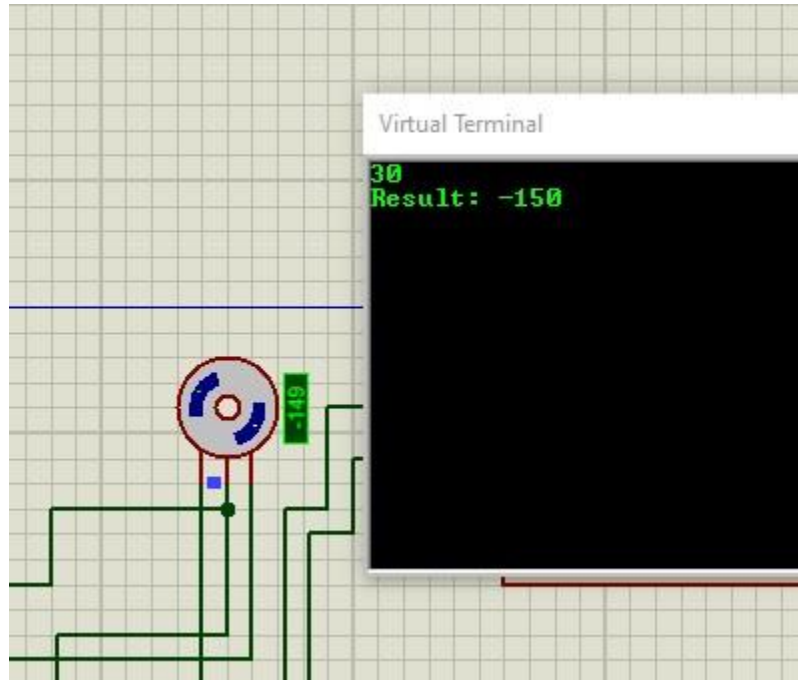
```

اگر دکمه ی = در کیبورد فشرده شود یعنی ورودی آماده است و چون موتور سروو بین 0 تا 180 می تواند دریافت کند و ورودی ما تا 360 هست بنابراین بعد از تبدیل رشته به عدد آن را تقسیم بر دو کرده تا در بازه ی ما قرار گیرد همچنین برای اینکه نتیجه نهایی را ببینیم آن را در ترمینال نوشته و با انجام شدن تبدیل زاویه ی خروجی روی موتور باید 180 درجه از زاویه ی ورودی کم تر باشد پس از 180 کم می کنیم.

در اینجا هم کلید ورودی کیبورد اگه عدد باشد

با متغیر ما کانکت می شود

## خروجی



```

#include <Servo.h>
#include <Keypad.h>
Servo myservo; // create servo object to control a servo
String serialInput = "";
int serialDegree = 0;

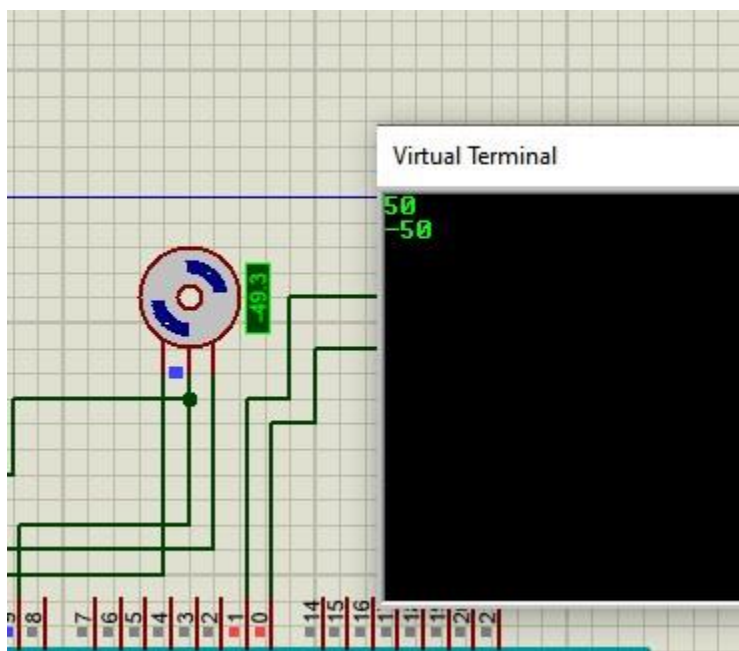
void setup() {
  myservo.attach(9,1000,2000);
  Serial.begin(9600);
}
void loop() {
  // put your main code here, to run repeatedly:

  //at first we check that the value is entered in to the terminal.
  //so after that it reads the string and converts it into the integer.
  // because if we enter the negative degree we will come out the range of 0 to 180
  // so we have to subtract the input with 180
  // then divided that into 2.
  //At last we print the result into the terminal
  if(Serial.available() > 0){
    serialInput = Serial.readString();
    Serial.println(serialInput.toInt());
    serialDegree = (180 - serialInput.toInt())/2;
    Serial.println(serialInput.toInt()* (-1));
  }
  myservo.write(serialDegree);
}

```

ابتدا چک می کنیم داده وارد ترمینال شده باشد سپس رشته ی خوانده شده را به عدد تبدیل می کنیم و چون اگر ما زاویه ی منفی وارد کنیم از محدوده ی 0 تا 180 خارج میشویم ورودی را از 180 کم کرده و بر 2 تقسیم می کنیم سپس ورودی و نتیجه را در ترمینال چاپ می کنیم. در آخر هم نتیجه را به موتور منتقل می کنیم.

## خروجی



```
#include <Servo.h>
#include <Keypad.h>

Servo myservo;
int val;// create servo object to control a servo

void setup() {
  // attaches the servo on pin 9 to the servo object
  myservo.attach(9,1000,2000);
  Serial.begin(9600);
}

void loop() {

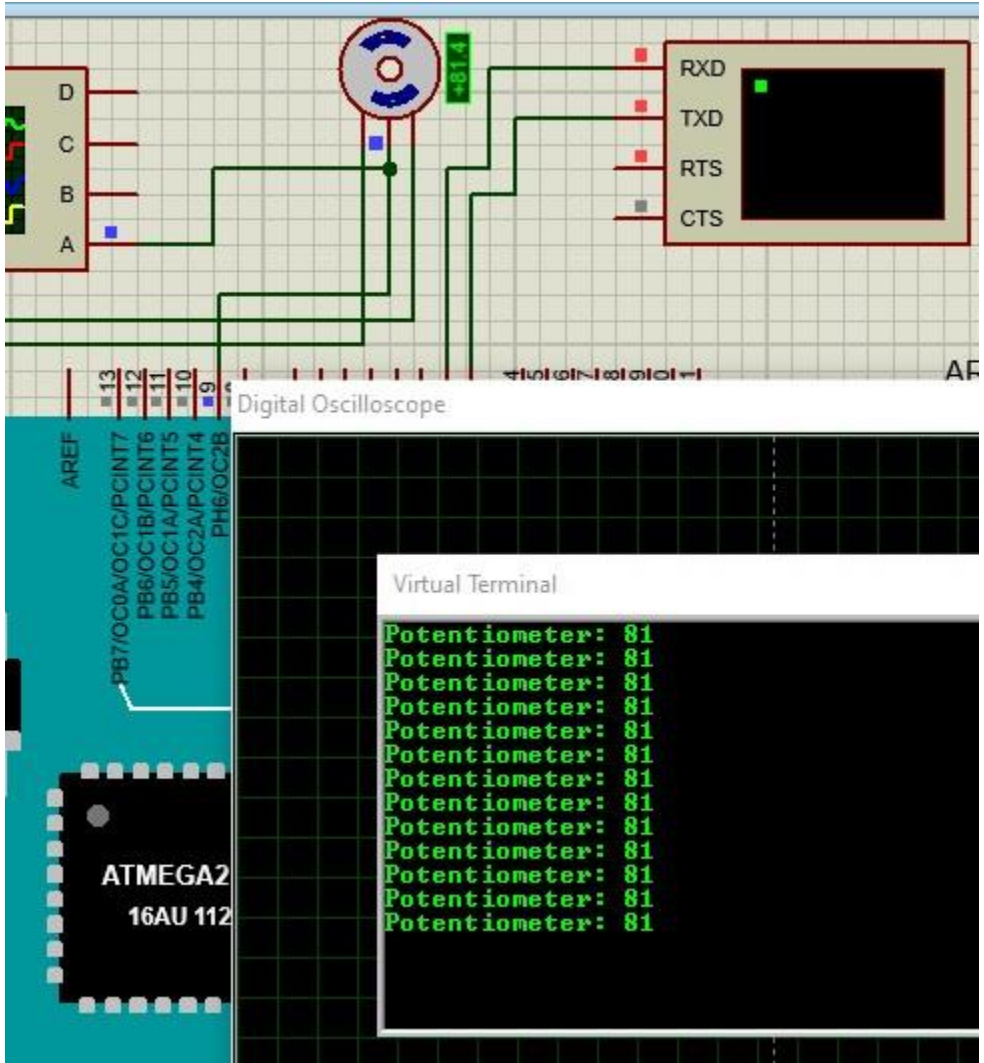
  Serial.print("Potentiometer: ");

  val = analogRead(A0);          // reads the value of the potentiometer(the value is between 0 and 1023)
  val = map(val, 0, 1023, 0, 180); // map( 0 to 1023) into 0 to 180 which is the servo motor get
  Serial.println(val);
  myservo.write(val);
  //the 50 delay means that our circuits after 0.05s checks that the value of pontasiometer
  delay(50);
}
```

پورت مربوط به پنتاسیومتر را تعریف کرده و سپس مقدار آن را خوانده و  
مپ می کنیم و روی ترمینال نمایش می دهیم.



## خروجی

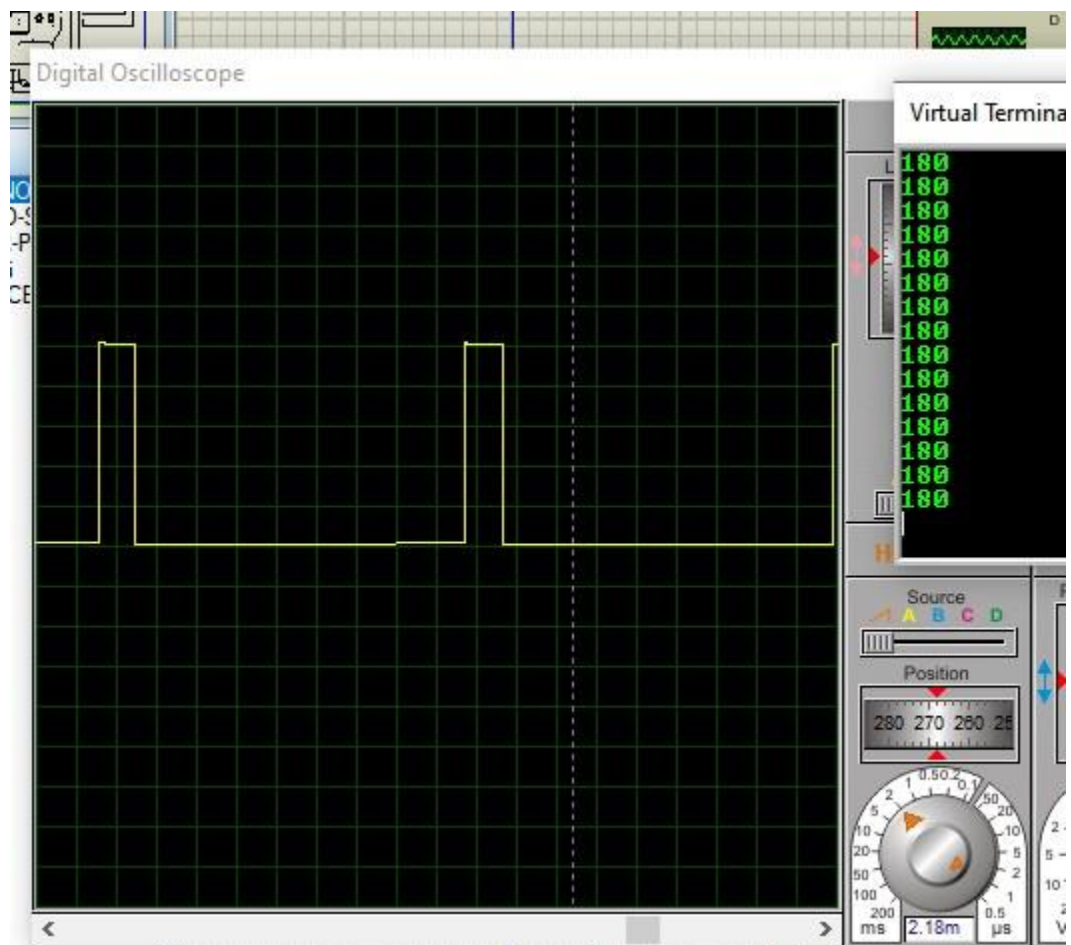


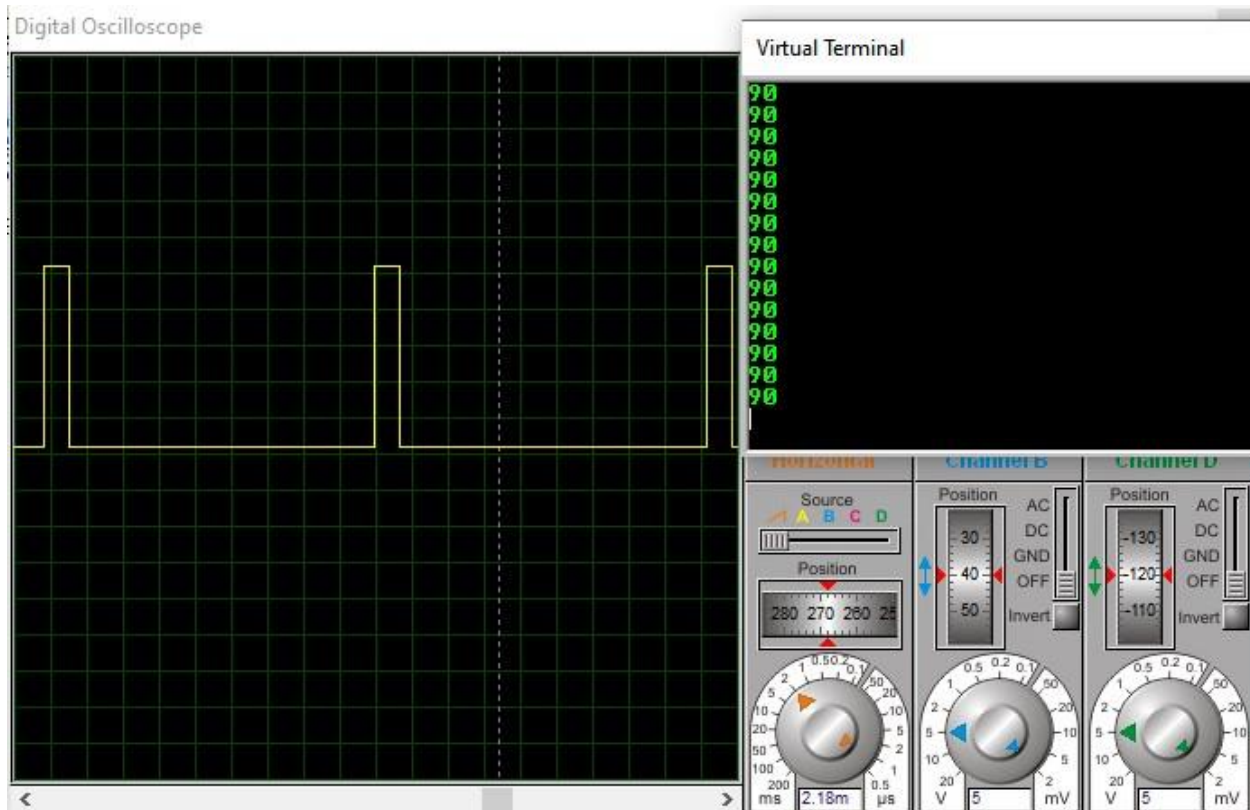
```
#include <Servo.h>
#include <Keypad.h>
int val;
Servo myservo; // create servo object to control a servo
void setup() {
  myservo.attach(9,1000,2000); // attaches the servo on pin 9 to the servo object
  Serial.begin(9600);
}

void loop() {
  //check the different val and print it into the terminal
  val=90;
  Serial.println(val);
  myservo.write(val);
  delay(50);
}
```

در کد مقادیر متفاوت را امتحان می کنیم.

## خروجی





همانطور که می بینیم با بیشتر شدن مقدار دوره ی کاری افزایش می یابد زیرا تاخیر ها یکسان بوده و مدت زمانی که طول می کشد تا پالس مربعی بعدی دیده شود یکسان می باشد.