

برای سوالاتی که از ان از این موتور استفاده میکنیم باید کتابخانه servo را باید فراخوانی کنیم و یک شی از servo میسازیم

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

```
Servo servo;
```

در ادامه انرا به پین 2 وصل میکنیم و در loop از دو حلقه برای تغییر مقدار pos از 0 تا 90 و بالعکس استفاده میکنیم و زاویه اعمالی را با دستور servo.write روی موتور اعمال میکنیم

```
void setup(){
```

```
    servo.attach(2);
```

```
}
```

```
void loop(){
```

```
    for (pos = 0; pos <= 90; pos += 1) {
```

```
        servo.write(pos);
```

```
        delay(10);
```

```
    }
```

```
    delay(100);
```

```
    for (pos = 90; pos >= 0; pos -= 1) {
```

```
        servo.write(pos);
```

```
        delay(10);
```

```
    }
```

```
    delay(100);
```

```
}
```

سوال 2)

ابتدا کتابخانه های مربوط به کی پد و موتور را فراخوانی میکنیم و سپس گیپد را تنظیم میکنیم و همچنین متغیر هایی که لازم اند را با توجه به کامنت های گذاشته شده تعریف میکنیم

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

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

```
Servo servo;
```

```
const byte ROWS = 4;
```

```
const byte COLS = 4;
```

```
// ست کردن کی پد
```

```
char hexaKeys[ROWS][COLS] = {
```

```
    {'7', '8', '9', '/'},
```

```
    {'4', '5', '6', '*'},
```

```
    {'1', '2', '3', '-'},
```

```
    {'C', '0', '=', '+'}
```

```
};
```

```
byte rowPins[ROWS] = {23, 25, 27, 29}; // وصل کردن سطر به کی پد
```

```
byte colPins[COLS] = {31, 33, 35, 37}; // وصل کردن ستون به کی پد
```

```
Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);
```

```
char Operator = NULL; // نگهداری عملیات
```

```
int len; // نگهداری سائز عدد
```

```
long int num1; // عدد اول برای عملیات
```

```
long int num2; // عدد دوم برای عملیات
```

```
char nums[10]; // نگهداری کاراکتر های وارد شده
```

سپس موتور را به پین 9 وصل میکنیم

```
void setup() {  
  Serial.begin(9600);  
  servo.attach(9);  
  len = 0;  
}
```

در `loop` با گرفتن کلید وارد شده بررسی میکنیم که اگر کلید برابر با `C` بود صفحه را پاک کرده و متغیرهای لازم را به 0 تغییر مقدار میدهیم و اگر = بود انگاه با تابع `convert` ارایه `nums` را به یک عدد تبدیل میکنیم و با توجه به خواست مساله ان را تبدیل کرده وبه موتور انتقال میدهیم زاویه بدست آمده را و اگر عدد باشد ان را در ارایه `nums` ریخته و نگهداری میکنیم

```
void loop() {  
  char Key = customKeypad.getKey();  
  if (Key) {  
    if(Key == 'C')  
      Serial.println();  
    else  
      Serial.print(Key);  
    if (Key == 'C') {  
      len = 0;  
      num1 = 0;  
    }  
    else if (Key == '=') {  
      num1 = converting(nums); //تبدیل کاراکتر به عدد دوم  
      len = 0;  
  
      //حل معادله با توجه به اعداد و اپراتور های وارد شده//  
      if(num1 < 180){  
        num2 = 1500 + ((num1 / 180.0) * 500);  
        servo.writeMicroseconds(num2);  
      }  
    }  
  }  
}
```

```

    }
    else{
        num2 = 1500 - (((360 - num1) / 180.0) * 500);
        servo.writeMicroseconds(num2);
    }
}
else {
    nums[len] = Key ;
    len++;
}
}
}
}
//تابع تبدیل کاراکتر به عدد
int converting(char* nums) {
    long int num = 0;
    for (int i = len; i < 10 ; i++ ) {
        nums[i] = 0;
    }
    num=atoi(nums);
    for (int i = 0 ; i < 10 ; i++ ) {
        nums[i] = 0;
    }
    return num;
}

```

سوال(3)

ابتدا کار های اولیه یعنی فراخوانی کتابخانه های لازم و تعریف متغیر های لازم را انجام می‌دهیم و در تابع setup پین موتور را برابر 9 قرار داده و با استفاده از servo.writeMicroseconds مقدار اولیه آن را تعیین می‌کنیم

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

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

```
int val;
```

```
Servo servo;
```

```
int num = 0;
```

```
int num2 = 1500;
```

```
int temp = 0;
```

```
String entered = "";
```

```
void setup(){
```

```
    Serial.begin(9600);
```

```
    servo.attach(9);
```

```
    servo.writeMicroseconds(num2);
```

```
}
```

سپس در `loop` با بررسی اینک `serial.available()` است یا نه شروع میکنیم و مقدار وارد شده را تا `n` میگیریم و با تابع ای که خودمان نوشتیم به نام `converting` مقدار آن را به عدد تبدیل میکنیم

در خط بعد با بررسی اینکه این عدد قابل قبول است یا خیر در تابع `valid` که در پایین تر آمده است مقدار مورد نظر را روی ترمینال پرینت کرده و مقدار آن را قرینه کرده و به موتور میدهیم با استفاده از `writeMicrosecond` و با پایان هر مرحله آن رشته را پاک میکنیم با قرار دادن `entered=""`

```
void loop(){
```

```
    if (Serial.available() > 0) {
```

```
        entered = Serial.readStringUntil('\n');
```

```
        num = converting(entered);
```

```
        if(num != 0 && valid(num)){
```

```
            Serial.println(num);
```

```

        temp = num/180.0 * 500;

        num2 -= temp;

        servo.writeMicroseconds(num2);
    }

    entered = "";
}

}

int converting(String a) {
    int flag = 0;
    int i , num1 = 0;
    if(a[0] == '-'){
        flag = 1;
        i = 1;
    }
    else
        i = 0;
    int k = pow(10,a.length() - flag - 1);
    if(a.length()- flag == 0)
        return 0;
    for(; i < a.length() ;i++){
        num1 += (a[i] - '0') * k;
        k /= 10;
    }
    if(flag == 1)
        num1 *= -1;
    return num1;
}

int valid(int a){

```

```

if(a > 0){
    if(num2 - a < 999)
        return 0;
}
else
    if(num2 + a > 2001)
        return 0;
return 1;
}

```

سوال 4)

جریان را از پتانسیومتر رد کردیم و سپس به سرو موتور وصل کردیم تا بتوانیم با استفاده از پتانسیومتر سرو موتور را تغییر بدهیم
ابتدا کتابخانه servo.h را فراخوانی میکنیم سپس یک شی از آن میسازیم از servo
سپس موتور را به پین 9 وصل میکنیم

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

```
Servo servo;
```

```
int val;
```

```

void setup() {
    servo.attach(9);
}

```

سپس مقدار analogRead(A0) را در متغیر val میریزیم و با استفاده از تابع map آن را به مقداری بین 1000 تا 2000 تبدیل میکنیم
که بتوان از آن در servo.writeMicroseconds استفاده کرد

```

void loop() {
    val = analogRead(A0);
    val = map(val, 0, 1023, 1000, 2000);
}

```

```
servo.writeMicroseconds(val);  
  
delay(15);  
  
}
```

سوال 5)

با توجه به نمایش سیگنال هایی که در اسیلوسکوپ آمده است میتوان فهمید دوره تناوب سیگنال ها همان fundamental period و نسبت زمانی که در هر دوره سیگنال 1 بوده به کل دوره برابر با duty cycle است که در هر لحظه با توجه به زاویه موتور در حال تغییر است



