

به نام خدا

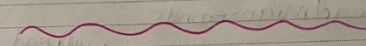
آزمایش ۴

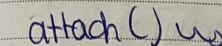
تاریخ: ۱۴۰۰/۸/۱۲

شماره دانشجویی: ۹۸۳۱۱۱۸

استاد: سرکارخانم علیزاده نویسنده: روژینا کاشفی

پیش گزارش

* PWM به معنای پهنای پالس است
pwm = pulse-width modulation
این روش برای کاهش میانس، توان میانس است به نسبت به روش
تغییر به سیکل ۱/۵ میسر است توان را کاهش می دهد و عریضی
و ۷VCC میسر
کابریهای آن: در حالت عادی می توانی به اتصال به پورت ED
و هر مودرها
در هر یک برای تبدیل می توانی آنرا به دیجیتال به کار می رود.

* فیلتر سرو موتور (تغییر دقت) برای میسر است
و از کابریهای آن
تغییر فرکانس میسر است، تغیر میسر است،
تغییر دقت است
* در هر ۱۰۰۰ میسر است ۱۰ تا ۲۰۰۰ میسر است
از آن به دیجیتال دارد
مقایسه دردی با مایک (۱۰۲۴) نمایش دارد میسر است
و با استفاده از دستور analog Read می توان این دیتا را
از ورودی بخوانیم

*  attach() میسر است
که متصل است که آنرا میسر است
این را میسر است و هر یک میسر است
write() میسر است
این میسر است میسر است
read() میسر است
مانند write است میسر است
لحاظ آن میسر است ۱۰۰ تا ۲۰۰۰ میسر است
مانند read است
لحاظ آن میسر است ۱۰۰ تا ۲۰۰۰ میسر است

توضیحات کد ها در کامنت آورده شده است.

سوال (۱)

مقدار سروو موتور بین ۱۰۰۰ تا ۲۰۰۰ است که ما تنظیمات آن را طوری تغییر دادیم که ۱۰۰۰ برابر ۱۸۰- شود و ۲۰۰۰ برابر ۱۸۰۰ شود.

مطابق این مقدار ۹۰ برابر ۱۷۵۰ و مقدار صفر برابر ۱۵۰۰ است.

```
#include <Servo.h>

Servo servo;
void setup() {
  //pin of servo attached to bord
  servo.attach(2);
}

void loop() {
  //1000 means -180 degree
  //2000 means 180 degree
  //switches between 0 and 90
  //1500 means 0 degree
  //1750 means 90 degree
  servo.writeMicroseconds(1500);
  delay(500);
  servo.writeMicroseconds(1750);
  delay(500);
}
```

سوال ۲)

اگر عدد بزرگتر از ۱۸۰ بود به معنی آنکه منفی است و اگر عدد بین ۰ تا ۱۸۰ بود به معنای آنکه مثبت است برداشت میشود و چون تابع writeMicroseconds عدد بین ۱۰۰۰ تا ۲۰۰۰ می پذیرد. عدد بدست آمده را با ۱۵۰۰ جمع میکنیم که همان صفر است.

زاویه بین ۰ تا ۱۸۰ بین ۱۵۰۰ تا ۲۰۰۰ قرار میگیرد.

زاویه باین -۱۸۰ تا ۰ بین ۱۰۰۰ تا ۱۵۰۰ قرار میگیرد.

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

//our keypad has 4 row and 4 col
//write value of each key in matrix

const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] = {
  {'7', '8', '9', '*'},
  {'4', '5', '6', '*'},
  {'1', '2', '3', '*'},
  {'0', '0', '*', '*'}
};

//row and col pins
byte rowPins[ROWS] = {31, 33, 35, 37};
byte colPins[COLS] = {23, 25, 27, 29};
//initialize keypad
//initialize servo
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
Servo servo;
String degree;
//attach pin of servo
void setup() {
  servo.attach(2);
}

void loop() {
  //get a key from keypad
  char key = keypad.getKey();
  //if there was a key
  if (key) {
    /* means to change servo
    if (key == '*')
    {
      //if degree bigger than 180 it means its between 0 till -180
      if (degree.toInt() > 180) {
        // first we reduce 180 degree
        //assume it is between 0 and 180
        //devide it to 360 to fits in 0 till 360 degree
        // * 1000 to fits between 1000 till 2000
        // * -1 to reduce it from 1500 and put it between begative range (-180 to 0 )
        //then we multiply it * -1
        servo.writeMicroseconds(-(float(degree.toInt() - 180) / 360 * 1000) + 1500);
      }
      //if smaller than 180 it means its between 0 till 180
      else {
        //we should devide it to 360 to fit it in scale of servo
        /* 1000 to give a number which we add with 1500 its between 1500 till 2000 ( positive numbers 0 till 180)
        servo.writeMicroseconds(float(degree.toInt()) / 360 * 1000 + 1500);
      }
      degree = "";
    }
    //if it was a number it means we are saying what degree we want to change
    else if (key >= '0' && key <= '9')
    {
      degree += key;
    }
  }
}
```

سوال ۳

کد اول علامت منفی نمی پذیرد و اعداد بزرگتر از ۱۸۰ را منفی در نظر میگیرد.

```
#include <Servo.h>

Servo servo; // create servo object to control a servo

int degree;

void setup() {
  Serial.begin(9600);
  // attaches the servo on pin 2 to the servo object
  servo.attach(2);
}

void loop() {
  //check if there is anything available on terminal
  if (Serial.available() > 0) {
    //convert to int
    degree = Serial.parseInt();
    // the number must be bigger than 0
    // negative numbers == numbers bigger than 180
    if (degree > 0) {
      // read the current value + the new degree
      // the value is negative the symmetry is positive
      if (degree > 180) {
        servo.writeMicroseconds(servo.readMicroseconds() + float(degree - 180) / 360 * 1000);
      }
    }
    else {
      // devide to 360 because to see the degree between 0 to 360
      // multiply with 1000 to put it in range of 1000 till 2000
      // the value is positive the symmetry is negative
      servo.writeMicroseconds(servo.readMicroseconds() - float(degree) / 360 * 1000);
    }
    Serial.println(degree);
    degree = 0;
  }
}
```

روش دوم منفی نیز می پذیرد و کاراکتر به کاراکتر میخواند و زمانی که s وارد شود خواندن تمام میشود.

```
#include <Servo.h>

Servo servo; // create servo object to control a servo

int degree;

void setup() {
  Serial.begin(9600);
  servo.attach(2); // attaches the servo on pin 2 to the servo object
}

//flag for when we write -
boolean flag = false;
// finish to say stop reading
boolean finish = false;
// the number we write
String number = "";
int num = 0;
void loop() {
  // while we are still reading
  if (! finish) {
    if (Serial.available() > 0) {
      char string = Serial.read();
      // if we have - make flag true
      // the number is negative
      if (string == '-') {
        flag = true;
        // if we enter s
        // means stop reading
      } else if (string == 's') {
        finish = true;
      } else {
        // add the char we are reading to number
        number = number + string;
      }
    }
    } else if (num == 0) {
      num = number.toInt();
      // convert our number to int
      // if we have -
      // make the number positive to negative
      if (! flag) {
        num = num * -1;
      }
      //sum the number with current value
      //devide to 360 to make it between 0 and 360
      servo.writeMicroseconds(servo.readMicroseconds() + float(num) / 360 * 1000);
      finish = false;
      flag = false;
      number = "";
      num = 0;
    }
  }
}
```

سوال ۴)

در این سوال مقدار پتانسیومتر با استفاده از analogRead خوانده میشود و به سروو موتور داده میشود.

```
#include <Servo.h>

Servo servo;
//PIN of potonsiometer on arduino
const int potpin = A0;
int val;

void setup() {
  Serial.begin(9600);
  servo.attach(2);
}

void loop() {
  //value of potonsiometer
  val = analogRead(potpin);
  Serial.println(val);
  //because the scale is KB we need to devide it to 1024 which is almost 1000
  //and convert it within number 1000 to 2000 (in range of servo)
  servo.writeMicroseconds(float(val) / 1024 * 1000 + 1000);
  delay(15);
}
```

سوال ۵)

دوره تناوب سیگنال در همه زاویه ها یکسان است و زمانی که زاویه را تغییر میدهیم تغییر مشاهده نمیکنیم.

در سایت آردوینو این دوره برای برد meag برابر ۴۹۰ هرتز است برای پین شماره ۲.

Duty cycle در هر زاویه مقدارش تغییر میکند و با تغییر آن زاویه موتور تنظیم میشود و کمترین مقدار را در صفر دارد و بیشترین مقدار را در ۱۸۰ دارد.

در ۱۸۰ درجه برابر است با ۱۲/۵ درصد است.