

به نام خدا

گزارش کار آزمایش چهارم آزمایشگاه ریزپردازنده

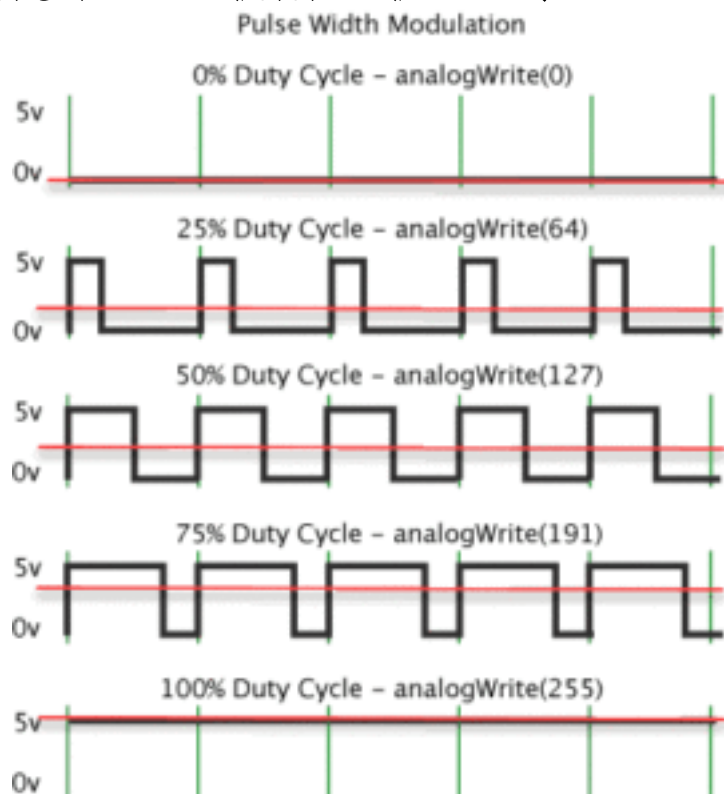
امیرپارسا سلمان خواه

۹۸۳۱۰۳۴

## مفهوم PWM و استفاده های آن:

PWM روشی است برای این که بتوانیم یک سیگنال آنالوگ مد نظر را با کمک یک سیگنال دیجیتال شبیه سازی کنیم و نتیجه ای مشابه با دادن همان سیگنال آنالوگ به مدار از آن بگیریم. به بیان دیگر، از آن جا که خروجی های میکروکنترلر ما دیجیتال هستند و ورودی های برخی ماژول ها و قطعات مانند موتور ها به یک سیگنال آنالوگ برای ورودی نیازمند هستند، PWM این امکان را به ما می دهد که با کمک یک پین خروجی دیجیتال، یک سیگنال آنالوگ را شبیه سازی کنیم.

این روش به این شکل عمل می کند که در یک دوره ی تناوب خیلی کوتاه که آن را Fundamental Period می نامند، بخشی از سیگنال را به شکل ۱ منطقی و بخش دیگر آن را به شکل ۰ منطقی می فرستند. حال به نسبت مدت زمانی که سیگنال ۱ بوده است یک خروجی آنالوگ دریافت می کنیم. به عنوان مثال اگر در ۵۰ درصد دوره ی تناوب سیگنال یک باشد و ماکسیمم ولتاژ ۵ ولت باشد، معادل این است که یک سیگنال آنالوگ ۲.۵ ولت در این بازه ی زمانی کوتاه در خروجی تولید کرده ایم. به نسبت زمانی که سیگنال ۱ است به کل مدت دوره ی تناوب، Duty Cycle گفته می شود. مثلاً Duty Cycle ۲۰ درصد یعنی این که در ۲۰ درصد دوره ی تناوب مقدار سیگنال برابر یک و در مابقی برابر با ۰ است.



## کاربرد های سروو موتور:

از سروو موتور ها در موارد زیادی استفاده می شود.  
در نیروگاه های خورشیدی برای تغییر زاویه ی پنل های خورشیدی به منظور دریافت هر چه بیشتر نور خورشید از سروو موتور ها استفاده می شود.  
در کار های رباتیک، از سروو موتور ها برای چرخاندن چرخ ربات استفاده می شود.  
همچنین جالب است که در دوربین های دیجیتال امروزی نیز از سروو موتور های بسیار دقیقی برای فوکوس بهتر روی اشیا استفاده می شود.  
در کارخانه ها و برای دستگاه های خط تولید، برای قرار دادن دقیق کالا های تولیدی در زیر دستگاه ها از سروو موتور استفاده می شود.

## توضیح در مورد ورودی آنالوگ و تحلیل آن در آردوینو و تابع مورد استفاده این آزمایش:

درون آردوینو یک مبدل آنالوگ به دیجیتال ۱۰ بیت تعبیه شده، که در صورت ورود سیگنال های بین ۰ تا ۵ ولت یا ۳.۳ ولت این مقدار را به عددی بین ۰ تا ۱۰۲۳ تبدیل می کند. مثلاً اگر ولتاژ ورودی ۲.۵ ولت و ولتاژ عملیاتی ۵ ولت باشد، مقدار ۵۱۱ توسط آردوینو برگردانده می شود. برای دریافت سیگنال های آنالوگ در آردوینو UNO از پین های A0 تا A5 استفاده می شود و برای این کار لازم است تا از تابع `analogRead` استفاده کنیم. به این شکل که نام پین را در ورودی تابع ذکر می کنیم و خروجی تابع نیز یک مقدار بین ۰ تا ۱۰۲۳ است که همان مقدار آنالوگ ورودی است.

## تعریف مختصر توابع مورد نیاز از کتابخانه ی `servo.h`:

`attach()`

این تابع ۲ پیاده سازی مختلف دارد.  
در پیاده سازی اول صرفاً پین متصل به سروو موتور تعریف می شود تا از آن برای ارسال داده استفاده شود.

در پیاده سازی دوم علاوه بر پین ذکر شده، باید حداقل و حداکثر عرض سیگنال نیز تعیین شود که این می تواند نکته ی مهمی برای عملکرد بهتر موتورمان باشد و این کتابخانه در تابع `write` از این مقادیر استفاده می کند تا میزان درجه ی داده شده را به میکروثانیه تبدیل کند و چرخه ی کار PWM را تشخیص دهد.

`write()`

این تابع یک زاویه دریافت می کند و سروو را روی آن تنظیم می کند.

read()

زاویه ی فعلی سروو را باز می گرداند.

writeMicroseconds()

با دریافت یک عدد در میکروثانیه، عرض پالس ارسالی را تنظیم می کند. مثلا اگر مینیمم و ماکسیمم عرض سیگنال موتورمان ۱۰۰۰ و ۲۰۰۰ باشد، با ارسال ۱۵۰۰ به این تابع، موتور در زاویه ی ۹۰ درجه قرار می گیرد.

readMicroseconds()

عرض پالس فعلی سروو را باز می گرداند.

کد آزمایش اول:

```
// Include the Servo library
#include <Servo.h>
// Declare the Servo pin
int servoPin = 9;
// Create a servo object
Servo Servo1;
void setup() {
    // We need to attach the servo to the used pin number
    Servo1.attach(servoPin);
}
void loop(){
    // Make servo go to 0 degrees
    Servo1.write(0);
    delay(1000);
    // Make servo go to 90 degrees
    Servo1.write(90);
    delay(1000);
}
```

## کد آزمایش دوم:

```
// Include the Servo library
#include <Servo.h>
#include <Keypad.h>

const byte ROWS = 4;
const byte COLS = 4;

char keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};

#define KEYPAD1 A5
#define KEYPAD2 A4
#define KEYPAD3 A3
#define KEYPAD4 A2
#define KEYPAD5 A1
#define KEYPAD6 A0
#define KEYPAD7 2
#define KEYPAD8 3

byte rowPins[ROWS] = {KEYPAD1, KEYPAD2, KEYPAD3, KEYPAD4};
//connect to the row pinouts of the keypad
byte colPins[COLS] = {KEYPAD5, KEYPAD6, KEYPAD7, KEYPAD8};
//connect to the column pinouts of the keypad
```

```
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);
```

```
int degree = 0;
```

```
// Declare the Servo pin
```

```
int servoPin = 9;
```

```
// Create a servo object
```

```
Servo Servo1;
```

```
void setup() {
```

```
    // We need to attach the servo to the used pin number
```

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

```
    Servo1.attach(servoPin);
```

```
}
```

```
void loop(){
```

```
    Serial.print("Please enter the degree: ");
```

```
    char key = keypad.waitForKey();
```

```
    // Make servo go to 0 degrees
```

```
    if (key >= '0' && key <= '9'){
```

```
        degree = degree * 10 + key - '0';
```

```
        Serial.println(degree);
```

```
    }
```

```
    else if (key == '*'){
```

```
        degree = 0;
```

```
    }
```

```
    else if (key == '#'){
```

```
        Serial.println("Done");
```

```
        if (degree <= 180){
```

```

        degree = 180 - degree;
        Servo1.write(0);
    }
    else{
        degree = 360 - degree;
        Servo1.write(180);
    }
    delay(1000);
    Servo1.write(degree);
    degree = 0;
}
}

```

کد آزمایش سوم:

```

// Include the Servo library
#include <Servo.h>

int current_degree = 180;
// Declare the Servo pin
int servoPin = 9;
// Create a servo object
Servo Servo1;
void setup() {
    // We need to attach the servo to the used pin number
    Serial.begin(9600);
    Servo1.attach(servoPin);
    Servo1.write(180);
}

```

```

void loop() {
    // put your main code here, to run repeatedly:
    Serial.print("Enter the number: ");
    while(Serial.available() == 0) {
    }
    String input = Serial.readString();
    long number = input.toInt();
    Serial.println(number);
    int diff = -number;
    int final_degree = current_degree + diff;
    if (final_degree < 0 || final_degree > 180){
        Serial.println("Can not rotate");
    }
    else{
        current_degree += diff;
        Serial.print("Moved to ");
        Serial.println(current_degree);
        Servo1.write(current_degree);
    }
}

```

کد آزمایش چهارم:

```

// Include the Servo library
#include <Servo.h>

// Declare the Servo pin
int servoPin = 9;
// Create a servo object
Servo Servo1;
void setup() {

```



```
// We need to attach the servo to the used pin number
Serial.begin(9600);
Servo1.attach(servoPin);
Servo1.write(0);
}
```

```
void loop() {
  // put your main code here, to run repeatedly:
  int input = analogRead(A5);
  input = map(input, 0, 1023, 0, 180);
  Servo1.write(input);
  Serial.println(input);
}
```