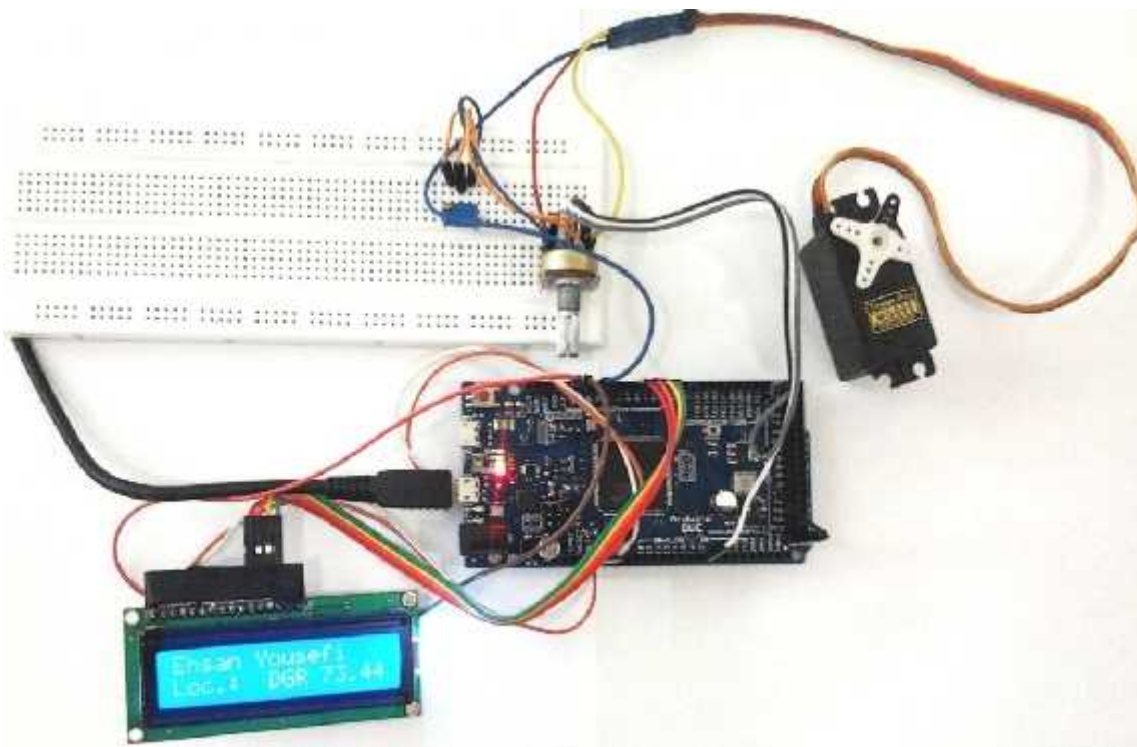


هدف از این پروژه کنترل زاویه یک سرو موتور آزمایشگاهی با توجه به اختلاف پتانسیل حاصل از دوران یک کفومت متغیر (پتانسیومتر) به صورت خطی و با دقت ۰/۱ درجه می باشد.

این مدار باید قابلیت نمایش زاویه تنظیم نقطه صفر و ریست شدن سخت افزاری نیز داشته باشد.

راه حل

استفاده از برد Arduino DUE که یک برد میکروکنترل بر پایه Atmel SAM3X8E ARM Cortex-m3 CPU با زبان برنامه نویسی جاوا ( از مشتقات زبان ++C ) و نمایش گر کارکتری ۲×۱۶ زمینه آبی و استفاده از مقاومت متغیر 5kΩ کربنی و یک عدد سرو موتور سری mg995 آزمایشگاهی .



## کنترل سرو موتور به وسیله Arduino

برای انجام این پروژه به قطعات زیر نیازمندیم

| تعداد | قطعات           |
|-------|-----------------|
| ۱     | برد برد         |
| ۱     | برد Arduino DUE |

|  |                                 |                   |
|--|---------------------------------|-------------------|
|     | <p>سیم برد بردی</p>             | <p>تعداد لازم</p> |
|     | <p>سرو موتور<br/>آزمایشگاهی</p> |                   |
|    | <p>پتانسیومتر</p>               | <p>۲</p>          |
|  | <p>LCD 2×16</p>                 | <p>۱</p>          |

موتور سرو دارای سه سر سیم است که در موتورهای مختلف رنگ های سیم ها متفاوت است اما معمولاً سیم قرمز به ۵ ولت و سیم سیاه یا قهوه ای به زمین و سیم دیگر به میکرو کنترلر باری گرفتن سیگنال های کنترل متصل میشود ، که در این پروژه سر سیم کنترل به پین شماره ۹ خروجی PWM برد کنترل متصل می شود ( مطابق با تعریفی که در برنامه شده است ).

به وسیله نرم افزار Arduino برنامه زیر که به زبان جاوا نوشته شده است را بر روی برد بار گذاری می کنیم

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

```
#include <LiquidCrystal.h>
```

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

```
Servo myservo;
```

```
int potpin = 7;
```

```
int val ;
```

```
void setup()
```

```
{
```

```
  lcd.begin(16, 2);
```

```
  lcd.print("Ehsan Yousefi");
```

```
  myservo.attach(9);
```

```
}
```

```
void loop()
```

```
{
```

```
val = analogRead(potpin);  
  
val = map(val, 0, 1023, 0, 155);  
  
myservo.write(val);  
  
delay(15);  
  
  
lcd.setCursor(0, 1);  
  
lcd.print("Loc.: DGR");  
  
lcd.setCursor(11, 1);  
  
lcd.print(val / 1.28);  
  
}
```

توضیح خط به خط برنامه

```
#include<servo.h>
```

تابه کتابخانه ای سرو برای استفاده از دستورات سرو در برنامه

```
#include<liquid Crystal.h>
```

تابع کتابخانه ای LCD برای استفاده از دستورات LCD در برنامه

```
liquidCristal lcd(12,11,5,4,3,2)
```

در برنامه تعریف می کنیم که ترتیب پایه های D7 , D6 , D5 , D4 , Enable , Rs ال سی دی به پایه های 2 , 3 , 4 , 5 , 11 , 12 خروجی Digital برد Arduino متصل شوند.

```
Servo myservo
```

یک متغیر جدید از نوع servo با نام myservo تعریف میکنیم ، با این کار ما میتوانیم servo را در برنامه نشان دهیم ما میتوانیم به صورت هم زمان هشت سرو را تعریف کنیم مانند myservo1 ,my servo2, ... .

```
Int potpin=7
```

Pin 7 را به عنوان potpin (هم زمان به عنوان integer) تعریف و سر وسط پتانسومتر را به ورودی 7 آنالوگ متصل می کنیم.

```
Int val;
```

Val را به عنوان یک متغیر تعریف می کنیم .

```
Void setup()
```

```
{
```

```
Lcd.begin(16.2);
```

با این دستور ستون و ردیف LCD که می خواهیم در آن نمایش دهیم مشخص می شود.

```
Lcd.print(" Ehsan Yousefi");
```

با این دستور کارکتر داخل پرانتز را بر روی LCD نمایش داده می شود .

```
Myservo.attach(9)
```

با این دستور مشخص می کنیم که سرو از طریق پین 9 سیگنال کنترلی را دریافت کند. باید توجه داشته باشیم که کتاب خانه servo.h تنها از دو پین 9 و 10 پشتیبانی می کند.

```
Void loop()
```

```
{
```

```
Val=analogRead(potpin)
```

خواندن مقدار از پین آنالوگ potpin که همان ورودی آنالوگ 7 است. یعنی مقدار پتانسیو متر را می خواند که بین ۰ تا ۱۰۲۳ میباشد.

```
Val=map (val,0,1023,0,155 );
```

ساختار دستور map به این صورت است ،

```
map(value,fromLow.fromHigh,toLow,toHigh)
```

به طوری که مقدار value که بین مقدار های fromLow و fromHigh تعریف شده اند را به نگاشتی در فاصله toLow و toHigh ر می آورد. به عنوان مثال

```
Y=map(x,1,50,50,1)
```

در واقع با این دستور مقیاس ۰ تا ۱۰۲۳ را بین ۰ تا ۱۸۰ تغییر میدهیم.

```
Myservo.write(val);
```

با استفاده از این دستور مقعیت سرو را باتوجه به مقدار val تغییر میکند

```
delay(15)
```

۱۵ میلی ثانیه تاخیر تعریف میکنیم ، این مدت زمان لازم است تا سرو به مکان تعیین شده برسد

```
Lcd.setCursor(0.1)
```

کتاب خانه liquidCrystal به ما اجازه میدهد تا صفحه نمایش هایی که با درایو هیتاچی HD44780

سازگار هستند را کنترل کنیم. این دستور با این ساختار (lcd.setCursor(col, row) جایی را که می خواهیم کرکتر در آن جا تمایش داده شود مشخص می کنیم.

به عنوان مثال برای یک صفحه نمایش ۱۶×۲

```
Lcd.setCursor(0.0);//top left
```

```
Lcd.setCursor(15.0);//top right
```

```
Lcd.setCursor(0.1);//down left
```

```
Lcd.setCursor(15.1);//down right
```

```
Lcd.print("LOC. : DGR");
```

عبارت داخل پرانتز در مکانی که در خط قبل تعریف کردیم نمایش داده می شود.

```
Lcd.setCursor(11,1);
```

این دستور مکان نما را به ستون ۱۱ ردیف پایین انتقال می دهد.

```
Lcd.print(val/1.28 )
```

مقدار داخل پرانتز را در مکان مشخص شده در سطر قبل نمایش میدهد.

### چگونگی کنترل سرو با استفاده از PWM

در این پروژه مقعیت شفت سرو با استفاده از عرض پالس کنترل میشود ، سرو تقریباً در هر ۲۰ میلی ثانیه پالسی را دریافت می کند. حال در صورتی که عرضپالس دریافتی ۱ میلی ثانیه باشد زاویه سرو صفر خواهد شد و اگر ۱/۵ میلی ثانیه باشد زاویه شفت سرو ۹۰ درجه خواهد شد و اگر عرض پالس ۲ میلی ثانیه شود زاویه شفت سرو ۱۸۰ درجه ( منظور حد اکثر زاویه چرخش شفت سرو است که در موتور های مختلف بسته به طراحی متفاوت است ، بسیاری از موتورها حدود ۱۷۰ درجه می توانند تغییر مقعیت دهند ولی سرو های continuous می توانند تا ۳۶۰ درجه بچرخد اما این زاویه برای موتور استفاده شده در این پروژه ۱۲۰ درجه است زاویه های کمتر هم با همین نسبت تغیر می کنند ، مثلاً زاویه موتور در عرض پالس ۱/۵ میلی ثانیه ، ۶۰ درجه است ) خواهد شد ، و ما میتوانیم با استفاده از پتانسیومتر عرض پالس را تغیر دهیم به طوری که زاویه شفت از ۰ تا ۱۸۰ درجه به وسیله تغیرات مقاومت پتانسیومتر قابل تغیر باشد.



