

گزارش تمرین ۵

فهرست

۲ فهرست

۳ بخش ۱:

۴ بخش ۲:

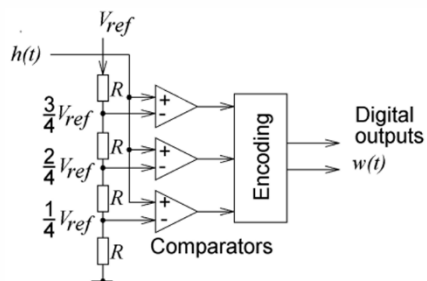
۵ مدل سازی

۷ شبیه سازی

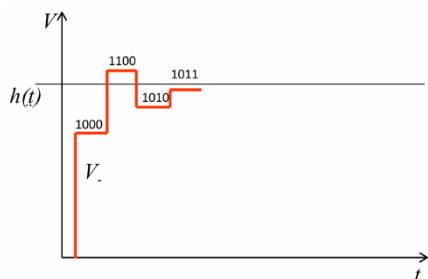
Error! Bookmark not defined. سروو موتور

بخش ۱

طبق فرض مسئله باید n بازه‌ی ولتاژ را به یک عدد نگاشت کنیم. پس مبدل آنالوگ به دیجیتال باید حداقل $\log_2 n$ بیتی باشد.



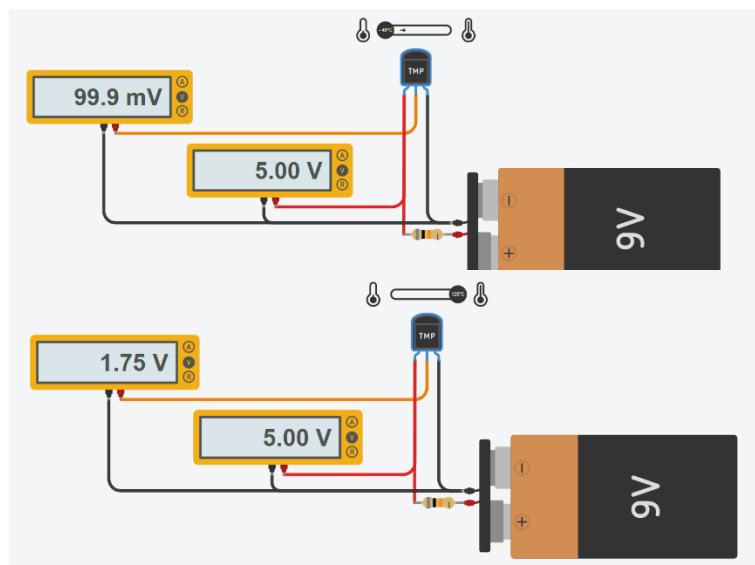
در مبدل فلش، ولتاژ ورودی به مبدل به صورت پلکانی از مقاوت‌هایی عبور می‌کند. در هر طبقه از مقاوت‌ها، ولتاژ کاهش می‌یابد و وارد یک مقایسه‌کننده می‌شود. واضح است که عبور جریان از مقاومت و کاهش ولتاژ در مدارهای الکتریکی با کمترین تاخیر انجام می‌شود. پس پیچیدگی زمانی آن $O(1)$ است. از آنجایی که در این مبدل به تعداد بازه‌ها، یک بیت تولید شده برای مقایسه داریم، پیچیدگی حافظه $O(n)$ است.



در مبدل successive، مشابه الگوریتم دودویی عدد متناظر با ولتاژ ورودی پیدا می‌شود. با شروع از بیت پر ارزش یک جستجوی دودویی انجام می‌شود و بر حسب اینکه عدد حدس زده کمتر از ولتاژ است یا خیر، یکی از بیت‌ها ۱ یا ۰ می‌شود. در این مبدل چون تمام بیت‌ها باید بررسی شوند، پیچیدگی زمانی آن $O(\log_2 n)$ است. از آنجایی که در این مبدل در هر لحظه فقط یک عدد $\log_2 n$ نگه می‌داریم، پیچیدگی حافظه $O(\log_2 n)$ است.

بخش ۲

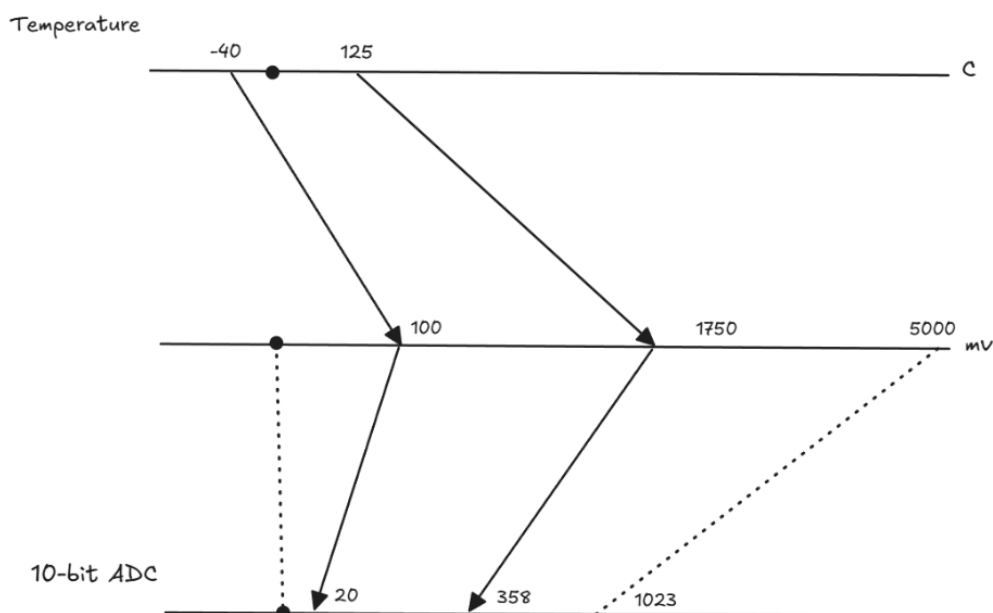
در این شبیه‌سازی یک سنسور دما از نوع TMP36 استفاده شده است. به دو سر این سنسور ولتاژ ۵ ولت اعمال شده است. پایه سوم این سنسور کمیت فیزیکی دما را به ولتاژ تبدیل کرده و خروجی می‌دهد. با استفاده از یک Multimeter هر دو ولتاژ به نمایش در آمده‌اند.



در تصویر مقابل دمای -40 درجه سانتی‌گراد اندازه‌گیری شده است و خروجی 99.9 میلی‌ولت خروجی داده است.

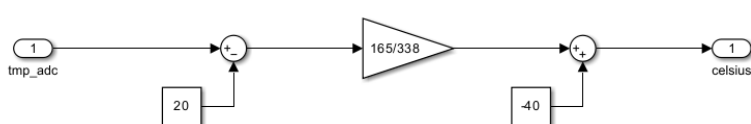
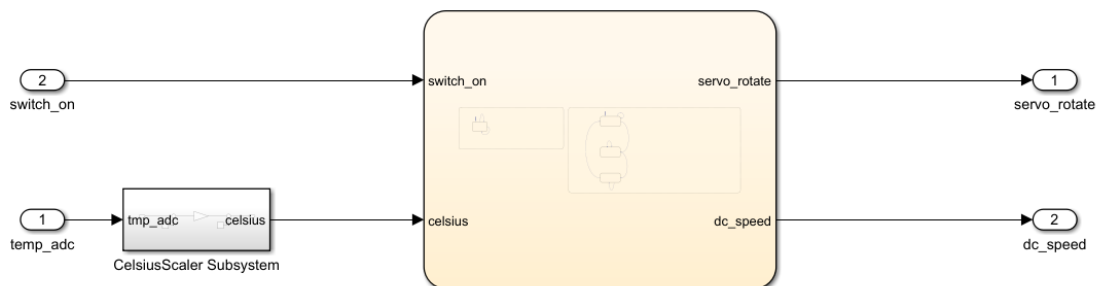
در این تصویر دمای 125 درجه سانتی‌گراد اندازه‌گیری شده است و ولتاژ 1.75 ولت خروجی داده است.

طبیعی است پس هر دمایی بین بازه‌ی کاری این سنسور دماسنج یعنی -40 تا 125 درجه به یک بازه ولتاژ بین 99.9 میلی‌ولت تا 1.75 ولت نگاشت می‌شود. مبدل آنالوگ به دیجیتال در برد Arduino Uno به صورت 10 بیتی در بازه صفر تا 5 ولت ولتاژ ورودی را به یک عدد نگاشت می‌کند. پس عدد متناظر با یک ولتاژ بین صفر تا 1023 خواهد بود.



مدل سازی

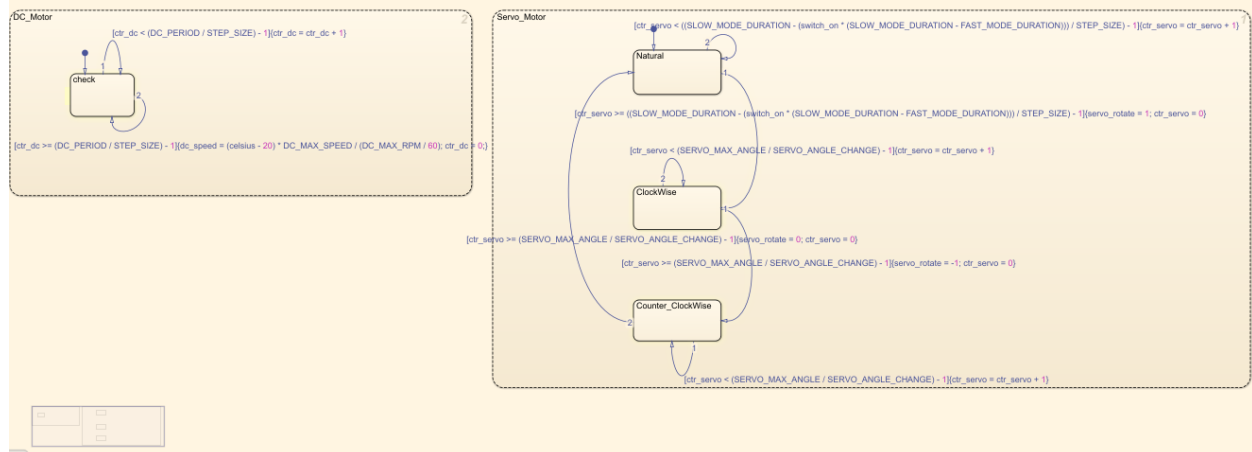
ورودی و خروجی ماشین حالت این مدار به صورت زیر است:



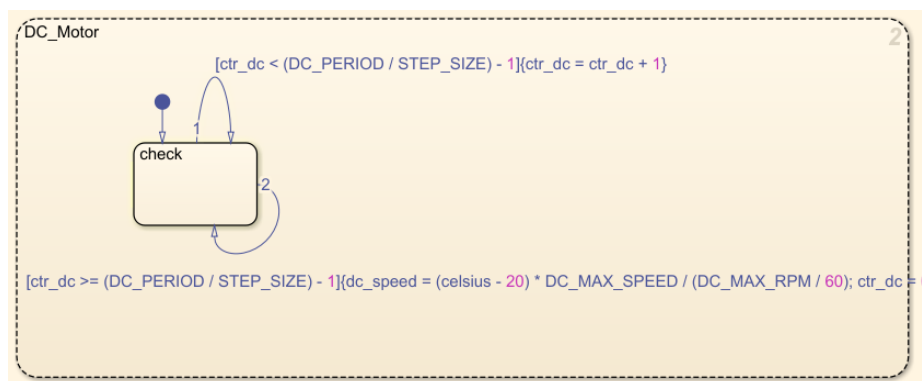
از آنجایی که تعداد دورهای موتور DC بر حسب دمای لحظه‌ای محیط است، به کمک ADC آردوینو باید مقدار سنسور خوانده شود و این مقدار خوانده شده تبدیل به واحد سانتی‌گراد شود.

Symbols			
TYPE	NAME	VALUE	PORT
	SLOW_MODE_DURATION	10000	
	servo_rotate	0	1
	FAST_MODE_DURATION	5000	
	STEP_SIZE	100	
	switch_on		1
	ctr_dc	0	
	ctr_servo	0	
	SERVO_MAX_ANGLE	180	
	SERVO_ANGLE_CHANGE	10	
	DC_PERIOD	1000	
	dc_speed	0	2
	celsius		2
	DC_MAX_RPM	16530	
	DC_MAX_SPEED	255	

ورودی‌ها، خروجی‌ها و متغیر محلی مورد استفاده در ماشین حالت طراحی شده:

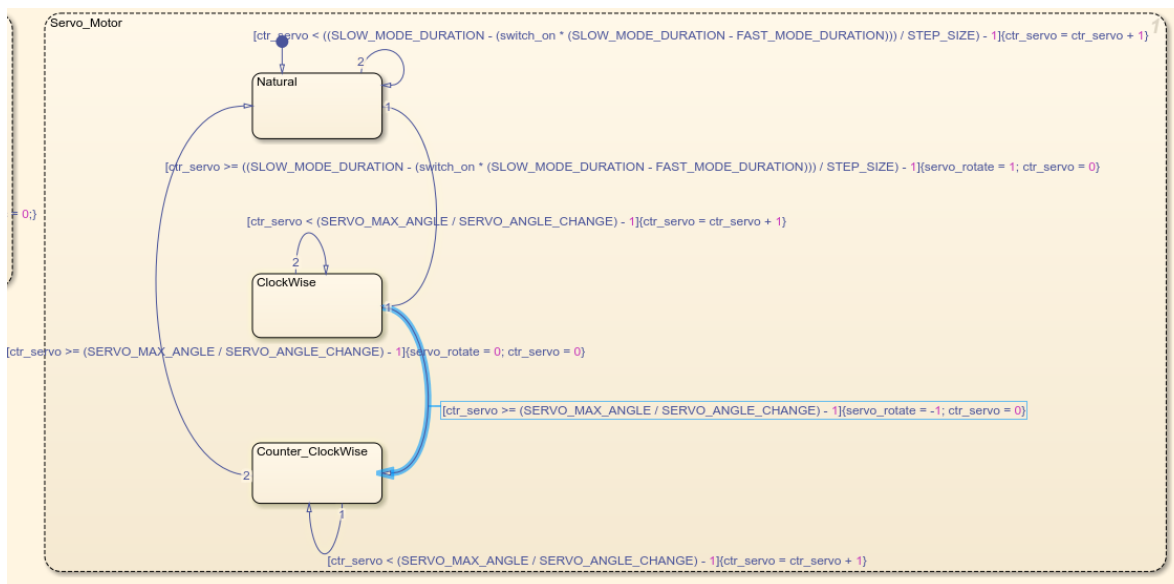


استیت ماشین موتور دی سی تنها شامل یک استیت و یک گذار است. این گذار پس از ۱ ثانیه سرعت موتور دی سی را آپدیت می‌کند.



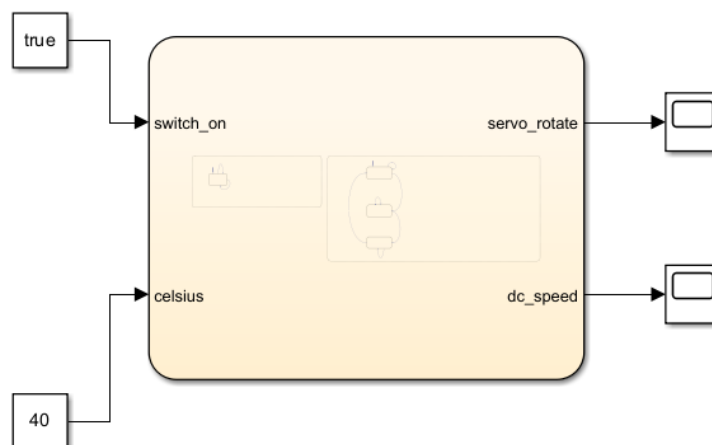
استیت ماشین سروو موتور شامل ۳ استیت است.

- استیت Natural زمانی است که $servo_rotate = 0$ و موتور حرکت نمی‌کند.
- استیت ClockWise همانطور که از اسم آن پیداست، زمانی است که $servo_rotate = 1$ و موتور در جهت عقربه‌های ساعت حرکت می‌کند.
- استیت Counter_ClockWise همانطور که از اسم آن پیداست، زمانی است که $servo_rotate = -1$ و موتور در خلاف جهت عقربه‌های ساعت حرکت می‌کند.
-

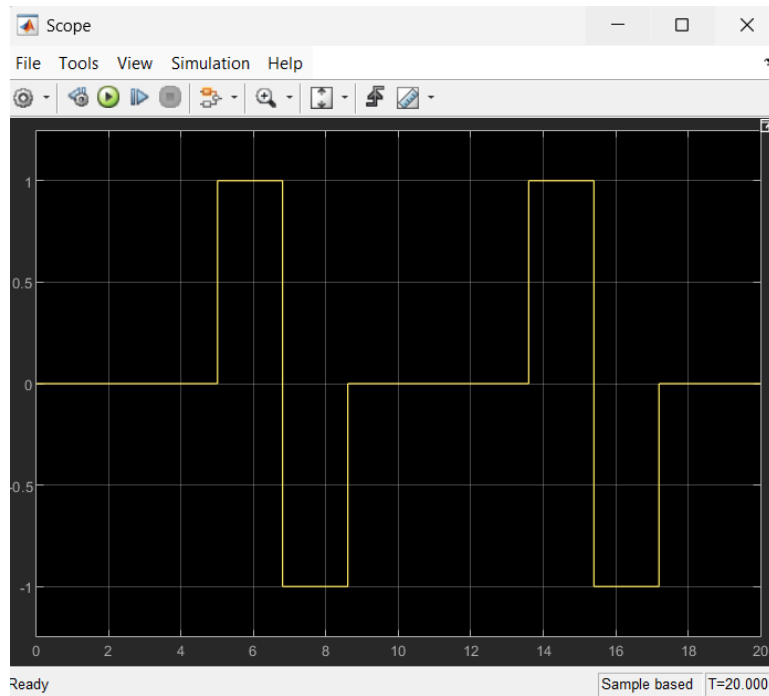


شبیه‌سازی

به منظور شبیه‌سازی استیت چارت، ورودی سویچ را روشن و مقدار دما به سلسیوس را ۴۰ قرار می‌دهیم.

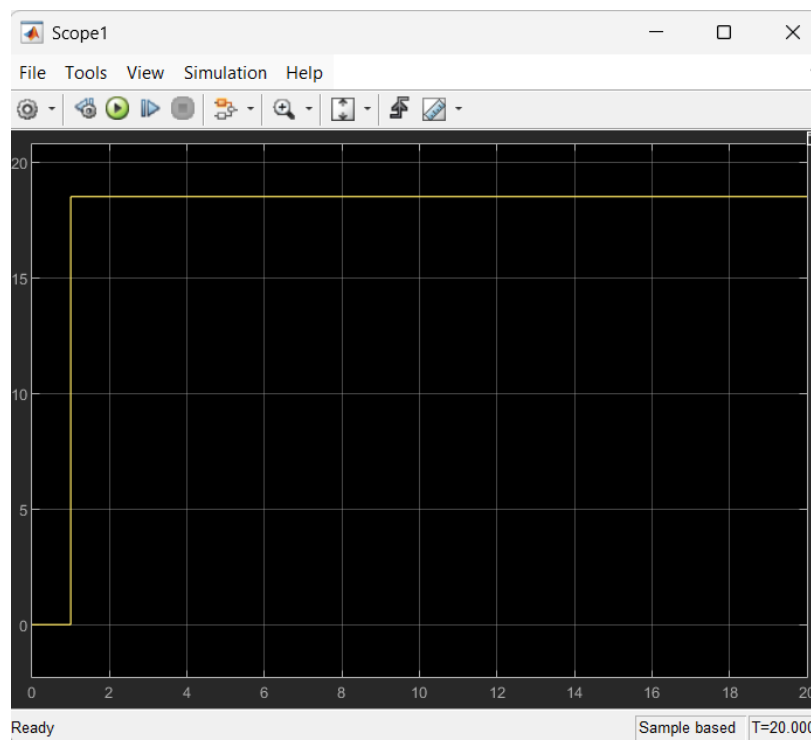


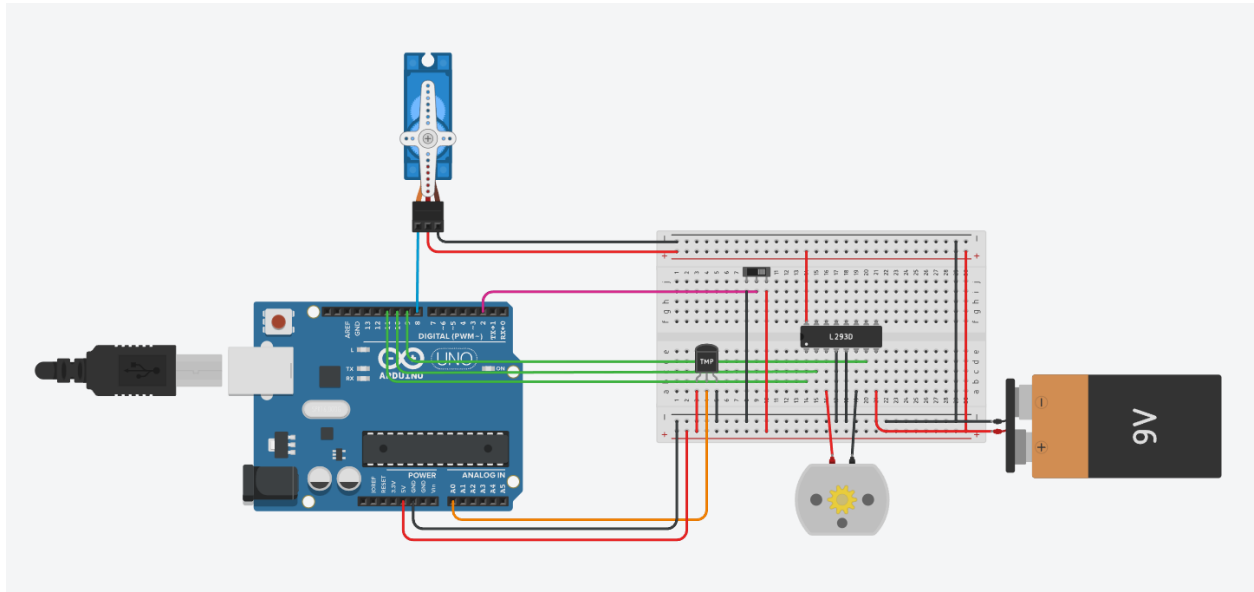
مشاهده می‌شود که سروو موتور، به مدت ۱.۸ ثانیه، ساعت‌گرد و ۱.۸ ثانیه پاد ساعت‌گرد می‌چرخد و سپس با روشن بودن کلید در حالت سریع قرار می‌گیرد و ۵ ثانیه تا حرکت بعدی فاصله دارد.



همچنین سرعت موتور دی سی، برابر با عبارت زیر می شود:

$$dc_{speed} = (40 - 20) \cdot \frac{255}{\frac{16530}{60}} = 18.51$$





```

#define switch_pin 2
#define temp_pin A0
#define dc_IN2_pin 9
#define dc_IN1_pin 10
#define dc_ENA_pin 11
#define servo_pin 8

#define servo_angle_change 5

int angle = 0;
Servo servo;

void setup() {
    // put your setup code here, to run once:
    pinMode(switch_pin, INPUT);
    pinMode(temp_pin, INPUT);

    pinMode(dc_IN2_pin, OUTPUT);
    pinMode(dc_IN1_pin, OUTPUT);
    pinMode(dc_ENA_pin, OUTPUT);

    digitalWrite(dc_IN2_pin, 0);
    digitalWrite(dc_IN1_pin, 1);
    analogWrite(dc_ENA_pin, 0);

    servo.attach(servo_pin);
    servo.write(0);
}

void loop() {
    // put your main code here, to run repeatedly:
    ex5_StateChart1_U.temp_adc = analogRead(temp_pin);
    ex5_StateChart1_U.switch_on = digitalRead(switch_pin);

    ex5_StateChart1_step();

    int servo_rotate = ex5_StateChart1_Y.servo_rotate;

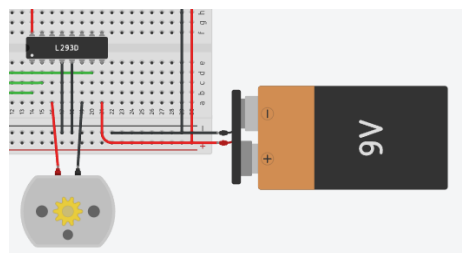
    if (servo_rotate != 0)
    {
        angle += servo_rotate * servo_angle_change;
        servo.write(angle);
    }
    analogWrite(dc_ENA_pin, ex5_StateChart1_Y.dc_speed);
}

```

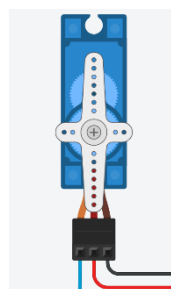
```

delay(100);
}

```



به منظور کنترل سرعت موتور دی سی از درایور L293D استفاده شده است. این درایور شامل سه پین کنترلی با نام ENA, IN2, IN1 است. برای قرار دادن موتور در حالت آماده‌باش $IN1 = 1$, $IN2 = 0$ را مقداردهی می‌کنیم. حال برای کنترل سرعت موتور کافی‌ست عددی‌ست میان ۰ تا ۲۵۵ را به پین اختصاص می‌دهیم.



سروو موتور شامل سه پایه است. VCC, GND و ورودی سیگنال PWM که توسط میکروکنترلر آردونیو تولید می‌شود. این موتور یک موج پالس مربعی با دوره تناوب ۲۰ میلی‌ثانیه ورودی می‌گیرد. طول پالس سیگنال مشخص‌کننده‌ی زاویه‌ی چرخش این موتور می‌باشد. اگر یک موج پالس مربعی با طول پالس ۱ میلی‌ثانیه داده شود، موتور می‌چرخد و در زاویه صفر درجه قرار می‌گیرد و اگر با طول پالس ۲ میلی‌ثانیه داده شود، موتور می‌چرخد و در زاویه ۱۸۰ درجه قرار می‌گیرد.

به کمک کتابخانه‌ی Servo می‌توان از توابع سطح بالاتر برای تعیین زاویه قرارگیری موتور استفاده کرد. مانند کدی که برای این پروژه زدیم، در این قسمت از کد، بر حسب خروجی دستور جهت چرخش موتور از کد تولیدی state-chart، زاویه جدیدی به موتور برای چرخش اعمال می‌شود.

```

if (servo_rotate != 0)
{
    angle += servo_rotate * servo_angle_change;
    servo.write(angle);
}

```