

DC MOTOR KÜTÜPHANESİ

Hazırlayan : İlayda Kuzucu

DC MOTORLAR

Teknolojinin gelişmesi ile birlikte elektronik ve motorlu sistemler yaşamın bir parçası haline gelmiş durumdadır. DC motor ise en çok kullanılan motor çeşitleri arasında yerini almaktadır. Bu yazımızın içeriğinde ise DC motor çeşitlerinden, ek olarak kullanılan komponentlerden ve IEEE ITU Robotics and Automation Society çatısı altında doğmuş ve Liek Software Team olarak adıyla devam eden takımımızın STM32F10x serisi mikrokontrolörler için hazırlamış olduğu DC motor kütüphanesinden bahsedilecektir.

GİRİŞ

DC motor, devresinden aldığı doğru akım elektrik enerjisini mekanik enerjiye dönüştüren bir sınıf döner elektrik motorudur. Motorun çalışma prensibi manyetik alan ile akım arasındaki etkileşimdir. Motorun içerisindeki sargılardan elektrik akım geçirilerek yine motorun içerisindeki sabit mıknatısların akıma zıt yönde oluşturduğu manyetik alanın etkisi ile hareket etme prensibine dayanır. Bu akım yönünün, sürekli olarak sabit mıknatısın akıma ters manyetik alan oluşturacağı şekilde değiştirilmesi gerekir . Bu değişimler fırçalı motorlarda fırça yardımıyla, fırçasız motorlarda ise akım yönü değiştirilerek yapılır.

DC motorlar temelde fırçalı motorlar, fırçasız motorlar, step motorlar ve servo motorlar olarak çeşitlenir. Her birinin karakteristik olarak birbirinden farklı özellikleri vardır. Projemiz fırçalı motor üzerine olduğundan diğerlerinden kısaca bahsedip fırçalı motorları inceleyelim.

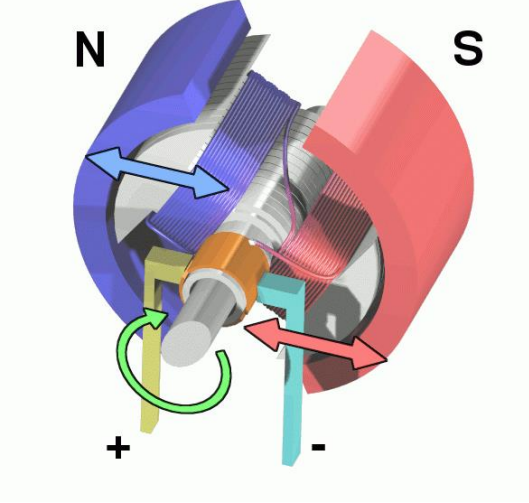
Step Motorlar: Konumlama hassasiyeti en yüksek motor çeşitidir. 2 ve 3 boyutlu yazıcılarda kullanılır. Çok fazla akım çekerler ve hareket hızı diğerlerine göre daha yavaştır.

Servo Motorlar: 0 ile 180 derece arasında dönme kabiliyetine sahiptir. Uzaktan kumandalı (RC) araçlarda manevra , kanat açısı, direksiyon gibi özelliklerin kontrol edilmesi için kullanılır.

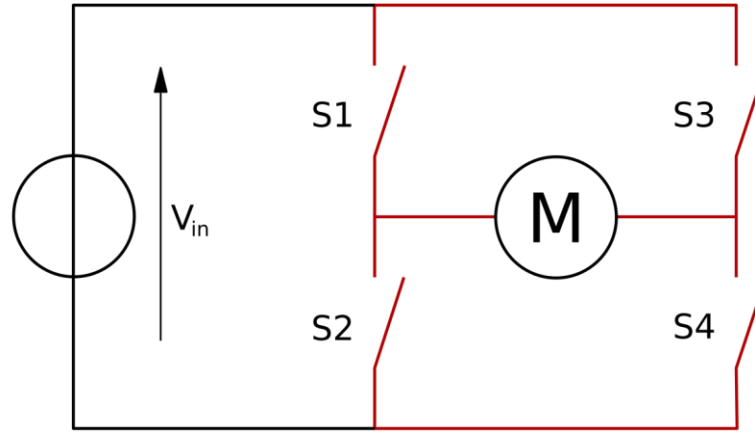
Fırçasız Motorlar: Fırçalı motorların geliştirilmesiyle üretilmiştir. ESC isimli özel sürücü devreleriyle kullanılırlar. Sürtünmenin minimum seviyede olması sayesinde verimlilikleri çok yüksektir. Yüksek performans gerektiren uygulamalarda kullanılırlar.

Fırçalı Motorlar: En eski ve en sık kullanılan çeşittir. Şarjlı el matkaplarından oyuncak arabalara kadar birçok farklı alette kullanılır. Kolay şekilde sürülebilirler. Fırçaları aşınma sebebiyle belli periyotlarda değiştirilmelidir. Devir hızını azaltıp yüksek tork istenirse redüktörlü (dişli kutusu), yüksek devir gerektiren cihazlarda ise redüktörsüz motor tercih edilir. Motora uygulanan gerilimin büyüklüğü ve yönü değiştirilerek çok kolay bir biçimde hız ve yön kontrolü yapılabilir. Mikrokontrollör kullanarak sürmek için sürücü entegrelerine ihtiyaç duyulur.

Çalışma prensibi basitçe şu şekildedir: motorun ana milinin üzerinde bobinler bulunur. Motorun ana gövdesinin iç kısmında ise güçlü mıknatıslar yer alır. Şaft üzerinde yer alan bobinlere fırçalar vasıtasıyla elektrik akımı uygulanır. Uygulanan elektrik akımıyla bobinlerde manyetik alan oluşur. Bu manyetik alan mıknatısların manyetik alanıyla sürekli çakışacak şekilde bir etki gösterir ve bu sayede mil hareket etmiş olur.

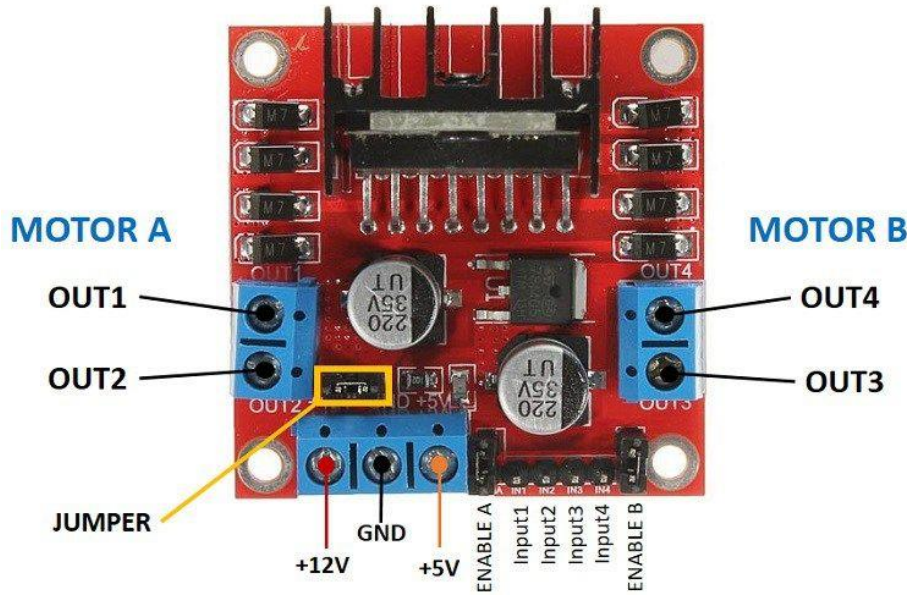


Oldukça basit bir çalışma prensibine sahip olan fırçalı DC motorlarının hızını değiştirmek için uygulanan voltaj değiştirilerek motorun hız kontrolü sağlanabilir. Motorun çalışma yönünü değiştirmek istenirse iletkenlerden geçen akımın yönünü değiştirmemiz gerekir. Bu işlem için ise **H-köprüsü** ismi verilen devrelere ihtiyacımız vardır. H-bridge devresi, anahtarlama elemanlarının açılıp kapanma kombinasyonları ile motora yön verme işlemidir. Bu işlem için L298N, L293D gibi sürücü entegreleri kullanılır.



H-Köprüsü devresi

H-köprüsü konfigürasyonu için biz projemizde L298N sürücü kartı kullandık. Kısaca incelemek gerekirse; S1 ve S4 anahtarları aynı anda açılarak motorun üzerinden geçen akım, S2 ve S3 anahtarları açıldığında motorun üzerinden tam tersi yöne dönecek, bu sayede motorun dönüş yönünü kolayca kontrol etmemize olanak sağlayacaktır. Entegrenin blok diyagramından da görebileceğimiz üzere L298 sürücünün üzerinde **In1**, **In2**, **In3**, **In4**, **EnA** ve **EnB** girişleri mevcuttur. L298 sürücü iki adet motor sürebilme yeteneğine sahip olduğundan In1, In2 ve EnA girişlerini bir kanal (Motor A), In3, In4 ve EnB girişlerini ise diğer kanalı (Motor B) kontrol etmek için kullanabiliriz.



DC Motor Kütüphanesi

DC motor kütüphanemizin STM32F10X serisi için olduğundan bahsetmiştik. Kütüphanemizin ilk versiyonlarında bu serideki mikrokontrolörler için tanımlı Peripheral Library kullanarak çalışıyordu. Sonraki versiyonlarımızda kütüphanemizin daha optimize olması ve sonraki hedef serimiz olan ARM Cortex M4 serisi kartlar için de geliştirilebilir, dinamik bir hale gelmesi için tüm kodlarımızı C dili ve register kullanarak yeniden düzenleyip “**DC Motor Library V2.4**” versiyonumuzu Liek Software Team olarak hazırladık. Kütüphane kodlarını anlatmaktan ziyade bu kısımda kütüphaneyi nasıl kullanabileceğinizden bahsedeceğim.

```
#include "dc_motor_library.h"
```

```
Motor Motor1;
```

```
int main () {
```

```
    Motor1.pwm_port=GPIOA;
```

```
//pwm kullanılan pin bilgileri
```

```
    Motor1.pwm_pin=GPIO_Pin_3;
```

```
    Motor1.backward_port=GPIOA;
```

```
//geri yönde dönüş sağlayan pin bilgileri
```

```
    Motor1.backward_pin= GPIO_Pin_4;
```

```
    Motor1.forward_port=GPIOA;
```

```
//ileri yönde dönüş sağlayan pin bilgieri
```

```
    Motor1.forward_pin=GPIO_Pin_5;
```

```
    MotorConfig(Motor1);
```

```
    while(1){
```

```
        MotorRun(Motor1, 100, 'f');
```

```
//ileri yönde
```

```
        Motor_Delay(3000);
```

```
        Motor_Stop(Motor1);
```

```
        Motor_Delay(500);
```

```
        MotorRun(Motor1, 80, 'b');
```

```
//geri yönde
```

```
        Motor_Delay(2000);
```

```
        Motor_Stop(Motor1);
```

```
        Motor_Delay(500);
```

```
    }
```

```
}
```

Yukarıda gördüğünüz örnek kod, motorumuzu L298N entegresiyle hız kontrolü yaparak kullandığımız bir uygulamamız aittir. Unutmayalım ki hız kontrolü yapacağımız pin için Pwm çıkışı olan pinlerden birini kullanmamız gerekiyor. Projenizi Keil veya STMCubeIDE üzerinden oluşturduğunuzu varsayalım. Ardından da kütüphaneyi ekledikten sonra yapacağınız işlem “**dc_motor_libray.h**” kütüphanesini “**main.c**” yani projenizin ana derlenme kaynak koduna dahil etmek olacaktır.

DC motor kodları **struct** yapısı kullanarak tanımlanmıştır. Bu nedenle bu yapıyı kullanarak motor nesnenizi üretmek için örnek koddaki gibi “**Motor Motor1;**” gibi bir dizin yazıp nesnenize ad vermeniz gerekmektedir. Derleyiciniz kütüphane içerisinden “Motor” yapısını bulacak ve “Motor1” adlı nesnenize aynen kopyalayacaktır. Ardından ana fonksiyonun içerisine motorunuzu bağladığınız pinlerin bilgilerini girmelisiniz. Burada ufak bir açıklama yapalım, STM32F10x serisi mikrokontrolörleri içeren Discovery veya Nucleo kartlarında pin numaraları PA2, PB3 vb şeklinde tanımlıdır. Örneğin PA2 demek A portunun 2 numaralı pini demektir. Bu sebeple Motor1 için pin ve port tanımlamaları aslında bize bu motorun A3, A4, A5 pinine bağlandığını göstermektedir. Seçtiğimiz A3 pini, PWM kullanarak hız kontrolü ve ölçümü yapacağımız pindir. Diğer iki pini ise dönüş yönünü belirlemek için kullanıyoruz. Sonrasında içerik ayarlamaları için çağırdığımız **MotorConfig()** fonksiyonu, içerisine tanımladığımız nesnenin bilgilerini alarak arka planda gerekli GPIO ve PWM ayarlamalarını sizin için otomatik olarak yapmaktadır. Ardından motorumuzu çalıştıracak olan **MotorRun()** fonksiyonumuzu çağırıyoruz. Bu fonksiyon 3 adet input almaktadır. İlki nesne, ikincisi yüzdelik olarak hız, üçüncüsü de yön bilgisidir. Motorunuzu durdurmak isterseniz de **Motor_Stop()** fonksiyonuna nesnenizi girerek bağlantılarınızı resetleyebilirsiniz.

Örnek kodumuzda motorumuz tam hızla(%100) ileri yönde 3 saniye dönüyor, ardından yarım saniye bekleyip %80 hızla geri yönde 2 saniye dönüyor. Tekrar yarım saniye daha bekleyerek bu döngüyü tekrar başlatıyor.

```
#include "dc_motor_library.h"
```

```
Motor Motor2;
```

```
int main(){
```

```
    Motor2.backward_port=GPIOA;           //geri yönde dönüş sağlayan pin bilgileri  
    Motor2.backward_pin= GPIO_Pin_6;
```

```
    Motor2.forward_port=GPIOB;            //ileri yönde dönüş sağlayan pin bilgileri  
    Motor2.forward_pin=GPIO_Pin_7;
```

```
    Basic_MotorConfig(Motor2);
```

```
    while(1){
```

```
        Basic_MotorRun(Motor2, 'f');      //ileri yönde dönüş  
        Motor_Delay(2000);
```

```
        Motor_Stop(Motor2);  
        Motor_Delay(800);
```

```
        Basic_MotorRun(Motor2, 'b');      //geri yönde dönüş  
        Motor_Delay(3000);
```

```
        Motor_Stop(Motor2);  
        Motor_Delay(800);
```

```
    }
```

```
}
```


Yukarıda gördüğünüz bir diğer örnek kod ise motorunuzu hız kontrolü yapmadan da kullanabildiğiniz uygulamamıza aittir. Önceki uygulamamızdaki gibi “**dc_motor_libray.h**” kütüphanesini “**main.c**” sayfasına dahil etmelisiniz.

İlk olarak motor nesnesine “**Motor2**” gibi bir ad vermelisiniz. Ardından ana fonksiyonunuzun içerisine motorunuzu bağladığınız pin bilgilerini girmelisiniz. Hız kontrolü yapılmayacağı için istediğiniz herhangi bir pini seçebilirsiniz. Örnek kodda motorun pinleri A6 ve B7 pinlerine bağlıdır. Sonrasında içerik ayarlamaları için **Basic_MotorConfig()** fonksiyonunu çağırıp içerisine nesnenizi tanımlamalısınız. Motorumuzu çalıştıracak olan **Basic_MotorRun()** fonksiyonumuzu çağırıyoruz. Bu fonksiyon 2 adet input almaktadır. İlki nesne, ikincisi de yön bilgisidir. Motorunuzu durdurmak isterseniz de **Motor_Stop()** kodunu kullanmanız gerekmektedir. Bu örneğimizde ise motorumuz ileri yönde 2 saniye dönüyor, ardından 0.8 saniye bekleyip geri yönde 3 saniye daha dönüyor. 0.8 saniye daha bekleyip döngüyü tekrar başlatıyor.

Kütüphanemizi incelemek ve kullanmak isterseniz şu linkten elde edebilirsiniz:

<https://github.com/Liek-Software-Team/Liek-DC-Motor-Library-STM32F10x>

KAYNAKÇA

- [1] <https://maker.robotistan.com/dc-motor-cesitleri-nelerdir/>
- [2] <https://www.elektrikrehberiniz.com/elektrik-motorlari/fircali-dc-motor-3848/>
- [3] <https://www.elektrikport.com/teknik-kutuphane/h-koprusu-nedir/17187#ad-image-2>
- [4] <http://pdf.datasheetcatalog.com/datasheet/stmicroelectronics/1773.pdf>