SERVO MOTOR KÜTÜPHANESİ

Hazırlayan : Görkem Aktaş

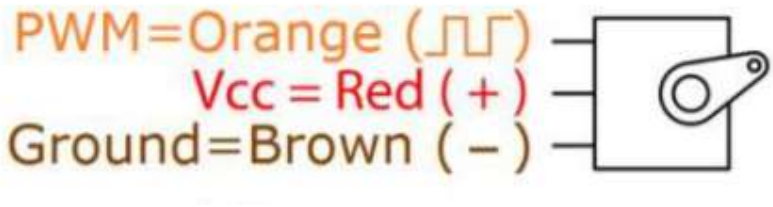
GİRİŞ

Gömülü sistemler ve otomasyon alanına yeni giren herkes az çok servo motorlar ile aşinalık kazanmıştır. Genellikle elimize bir Arduino aldığımızda en çok kullandığımı komponentlerden bir tanesi şüphesiz ki servo motorlardır. Bu yazımızın içeriğinde servo motorlardan, ARM tabanlı mimariye sahip STM32F103C8 mikrokontrolöründen ve ardından ve Liek Software Team olarak STM32F10x serisi mikrokontrolörler için hazırlamış olduğumuz servo motor kütüphanesinden bahsedilecektir.

SERVO MOTORLAR

Servo, mekanizmalardaki açısal-doğrusal pozisyon, hız ve ivme kontrolünü hatasız bir şekilde yapan tahrik sistemi olarak tanımlanır. Yani hareket kontrolü yapılan bir düzenektir. Servo motorlar, robot teknolojilerinde en çok kullanılan motor çeşidi olmakla birlikte, RC (Radio Control) uygulamalarda da kullanılmaktadırlar. RC Servo Motorlar ilk olarak uzaktan kumandalı model araçlarda kullanılmışlardır. Servolar, istenilen pozisyonu alması ve yeni bir komut gelmediği sürece bulunduğu pozisyonu değiştirmemesi amacıyla tasarlanmıştır.[1]

Basit anlamda servo motorlar üç pine sahiptirler. Bunlar VCC, GND ve PWM olarak da adlandırabiliriz. Piyasada çok farklı servo motorlar bulunmasına karşılık en çok kullanılanlardan bir tanesi SG90 modelidir. Bu servo motorun üç pini şu şekildedir,



Şekil 1 SG90 Pin Şeması[2]

Şekil1'de görüleceği üzere üç renkte ve bahsettiğimiz isimlendirmeler bulunmaktadır. Şimdi bunların neler olduğunu açıklayalım,

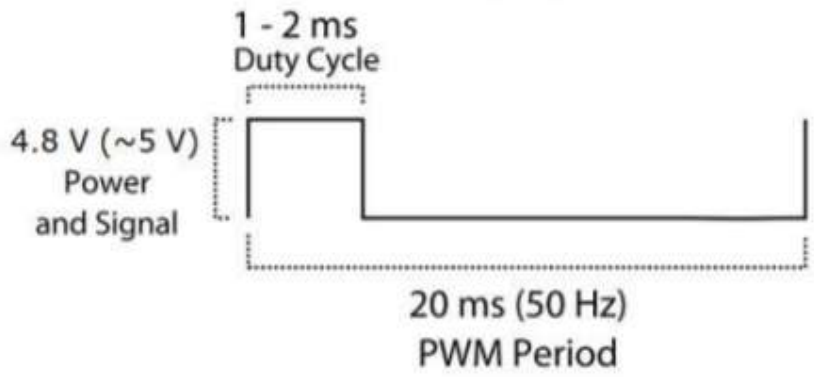
VCC : Besleme olarak da geçer. Servo motorumuzun esas enerji hattı buradan sağlanmaktadır. Datasheet'te belirtildiği üzere 4.8V-6V aralığında çalışabilmektedir.

GND : Toprak hattı motor devremizin, devreyi tamamlaması için gerekli referans noktasını işaret eder.

PWM : Pulse Width Modulation, Türkçesi ile genlik genişlik modülasyonu olarak da geçer. Servo motorlar DC motorlar gibi direkt birim basamak girişinin sürekli fonksiyonu

cinsinden çalışmazlar. Yani sürekli sabit bir beslemeleri yoktur. Onun yerine belirli frekans aralıklarında darbeler ile çalışarak ortalama değer üzerinden çalışırlar. Böylece açı kontrolü yapmamıza olanak tanırlar. Bu pini mikrokontrolördeki PWM özelliği bulunan bir pine bağlamanız gerekmektedir.

Şimdi de PWM olayını biraz daha detaylı inceleyelim ve SG90 servo motorlarında nasıl çalıştıklarına bakalım.



Şekil 2 SG90 PWM Şeması[2]

Şekil2'de görüleceği üzere servo motorumuzun ana periyodu 20ms(milisaniye)'dir. Bu şu anlama geliyor, servo motorumuza gidecek olan PWM'in zamanlayıcı ayarlamasını yaparken periyodumuzu 20ms olacak şekilde ayarlamalıyız. Ardından duty cycle yani görev süresi denilen bir terim karşımıza çıkmaktadır. Bu da servo motorumuza gelecek sinyalin özelliğini belirtmektedir. Yani istediğimiz açı değerleri için 1-2ms aralığında bir değer seçmemiz gerekiyor. Datasheet'te belirtildiği üzere −90° için 1ms'lik bir sinyal, 0° için 1.5ms'lik bir sinyal ve 90° için de 2ms'lik bir sinyal göndermemiz gerekiyor. Buradan anlaşılacak sonuç şudur ki servo motorumuz %5 ve %10'luk duty cycle ile çalışmaktadır. O sebeple ayarlamalar ona göre yapılmalı ki açı konusunda kontrole hakim olunabilsin.

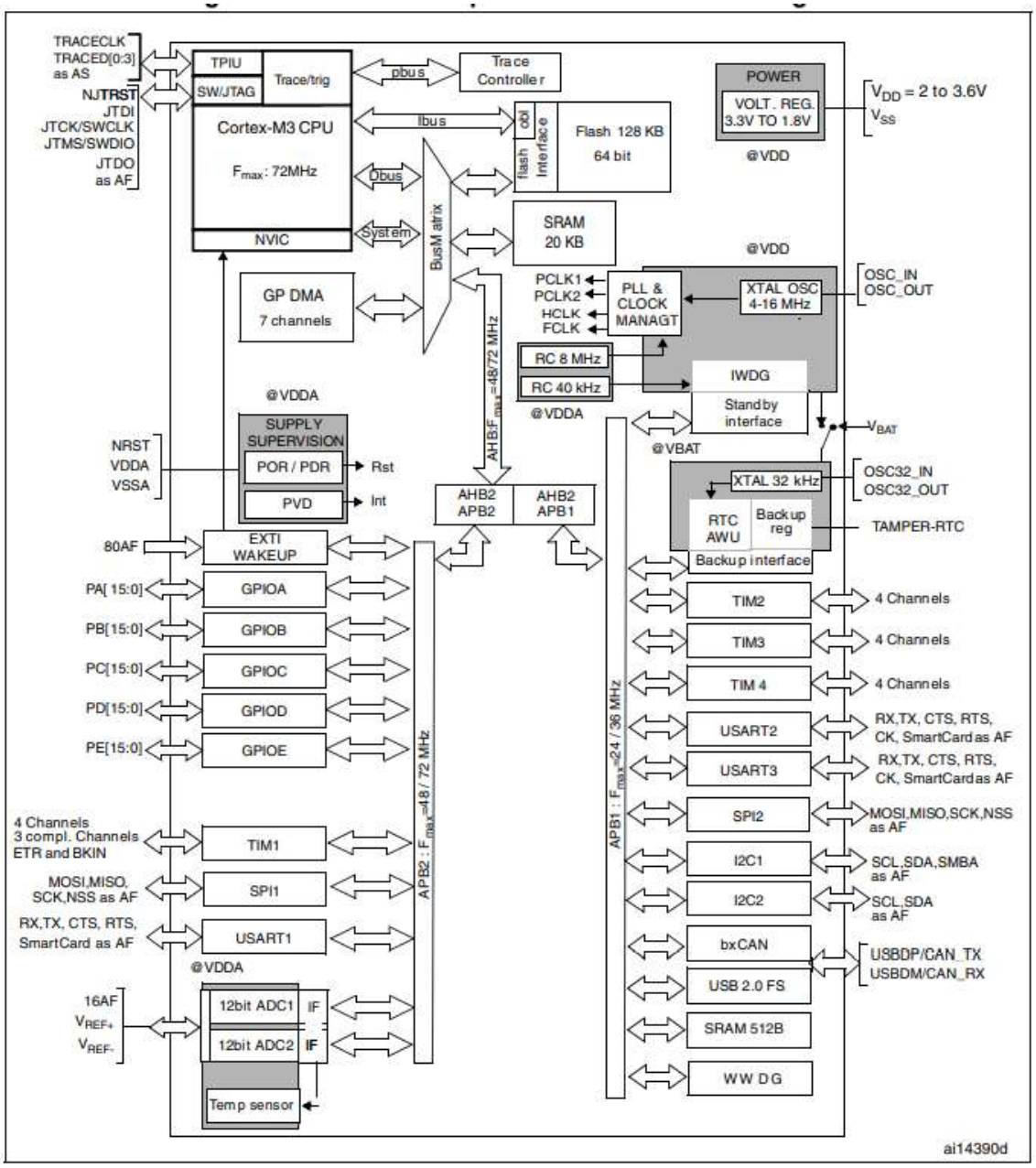
ARM TABANLI MİKROKONTROLÖRLER

Servo motorları anlattıktan sonra şimdi de bu servo motoru süreceğimiz mikrokontrolör yapısını anlatmamız gerekiyor. Öncelikle ARM nedir derseniz bunun cevabına Arduino ile farkından yola çıkarak başlayabiliriz. Arduino 8-bitlik data bus özelliğine sahip bir AVR mimari mikrokontrolördür. Bu cümlenin anlamı şudur, Arduino anlık örnekleme zamanı içerisinde 8bitlik verileri işleyebilir ve mimarisinin farklı oluşu ona farklı bir Assembly yazım tekniği de sağlamaktadır. ARM'ın en önemli farkı ise 32-bitlik bir data bus özelliğine sahip olmasıdır. Bu da şüphesiz ARM'ı Arduinodan güçlü kılan yegane yanı. Şimdi yüzeysel olarak ARM'dan bahsedelim.

Öncelikle projemizde kullanılan STM32F10x serisi mikrokontrolörler ARM Cortex M3 model mikrokontrolörlerdir. Bu mikrokontrolörler akrabaları olan ARM7, ARM9,

ARM11 gibi mikrokontrolöre nazaran zayıf olsalar da ufak otomasyon projeleri için oldukça ekonomik çözümlerdir. Yukarıda bahsettiğimiz ARM modelleri ise bugün kullandığımız akıllı telefonlar veya bilgisayarlar gibi daha üst segment ürünlerde tercih edilmektedir.

Şimdi kullanılabilecek bir mikrokontrolör üzerinden yapıyı anlatalım.



Şekil 3 STM32F10x Blok Diyagram [3]

Şekil3'te görülen blok diyagrama kısaca göz atacak olursak, elimizdeki mikrokontrolörün General Purpose Input Output (GPIO) yani genel amaçlı kullanabilecek giriş çıkış pinlerinin A,B,C,D,E olmak üzere 5 port şeklinde tanımlandığı görülebilmektedir. Burada bizim için önemli olan kısım servo motorlar alakalı olan kısımlardır. Öncelikle bir

servo motorun PWM ihtiyacı olduğunu biliyoruz. Peki PWM nasıl oluşturabiliriz ? PWM oluşturabilmek için bir zamanlayıcıya yani Timer'a ihtiyaç vardır. Şekil3'e bakacak olursanız TIM1, TIM2, TIM3 ve TIM4 olmak üzere 4 adet zamanlayıcı bulunmaktadır. Dikkat edilmesi gereken nokta TIM1, APB2 data bus hattına bağlı iken TIM2, TIM3 ve TIM4 ise APB1 data bus hattına bağlıdırlar. Eğer bu mikrokontrolör ile bunları yönetmek isterseniz öncelikle miktrokontrolörün CPU'su aracılığı ile bu haberleşme hatlarını aktif etmeniz gerekmektedir. Bu aktifleştirme işlemleri RCC adı altında tanımlanmış registerlar ile mümkündür. Ardından PWM özelliği içeren bir pini seçerek ilgili GPIO portundan aktif hale getirip onu bir zamanlayıcı ile entegre ederek servo motorumuzu sürebiliriz. Oldukça karışık değil mi ? İşte tüm bunlarla uğraşmamak için Liek Software Team olarak Servo Motor Library projesini yaptık. Şimdi de kütüphanemizden bahsedelim.

SERVO MOTOR KÜTÜPHANESİ

Servo motor kütüphanemizin STM32F10x serisi için olduğundan bahsetmiştik. Bu kütüphane ilk versiyonlarında bu serideki mikrokontrolörler için tanımlı Peripheral Library aracılığı ile çalışıyordu. Ardından yeni versiyonlar ile buradaki fonksiyonlardan vazgeçip **"Servo Motor Library V2.2"** ile beraber tüm kodlarımızı C dili ve register kullanarak hazırladık. Böylece kütüphane hem daha optimize oldu hem de bir sonraki hedefimiz olan ARM Cortex M4 serisi kartlar için de geliştirilebilir dinamik bir ortam hazırlamış olduk.

Kütüphane kodlarını anlatmaktan ziyade yazının bu bölümünde sizlere bu kütüphaneyi nasıl kullanabileceğinizi anlatacağım.

```c
#include "servo_lib.h"

Servo myServo;
Servo myServo2;

int main(){

    myServo.port='A';
    myServo.pin=2;
    myServo2.port='A';
    myServo2.pin=3;

    Servo_Config(myServo);
    Servo_Config(myServo2);

    while(1){

        Servo_Degree(myServo,0);
        Servo_Degree(myServo2,0);
        Servo_Delay(2000);

        Servo_Degree(myServo,80);
        Servo_Degree(myServo2,80);
        Servo_Delay(2000);

        Servo_Degree(myServo,150);
        Servo_Degree(myServo2,150);
        Servo_Delay(2000);

        Servo_Detach(myServo2);


    }
}
```

Yukarıda örnek bir uygulama kodu görmektesiniz. Projenizi Keil veya STMCubeIDE üzerinden oluşturduğunuzu varsayalım. Ardından da kütüphaneyi ekledikten sonra yapacağınız işlem **"servo_lib.h"** kütüphanesini **"main.c"** yani projenizin ana derlenme kaynak koduna dahil etmek olacaktır.

Servo motor kodları **struct** olarak tanımlanmışlardır. Bu sebeple bu yapıyı kullanarak servo üretmek için örnek koddaki gibi **"Servo myServo;"** benzeri bir kod yazmanız gerekmektedir. Derleyiciniz kütüphane içerisinden Servo yapısını bulacak ve myServo adıyla aynen kopyalayacaktır. Sonrasında yapmanız gereken ise ana fonksiyonunuz içerisinde servonuzu bağladığınız pin bilgilerini girmek. 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. O sebeple myServo için pin ve port tanımlamaları aslında bize bu servonun PA2 pinine bağlandığını göstermektedir. Sonrasında içerik kurulumları için Servo_Config() fonksiyonunu çağırıyoruz. Bu fonksiyon ile alınan verilerle arka planda gerekli GPIO ve PWM ayarlamaları sizin için otomatik olarak yapılmaktadır. Ardından servonuz için açı bilgisini Servo_Degree() fonksiyonu ile girebilirsiniz. Bu fonksiyon 2 input almaktadır. Bunlardan ilki servo motorun kendisi, ikincisi ise açı değeridir. Servonuzu bir şekilde

pininden bağımsız kılmak yani deaktif etmek isterseniz de Servo_Detach() fonksiyonu ile servo motorunuzun pin bağlantılarını resetleyebilirsiniz.

KÜTÜPHANE İNDİRME LİNKİ

https://github.com/Liek-Software-Team/STM32F10x_Servo_Library

KAYNAKÇA

[1] https://maker.robotistan.com/rc-servo-motor-nedir/

[2] http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf

[3] STM32F10x Series Datasheet