****

T.C. İSTANBUL OKAN ÜNİVERSTESİ

MESLEK YÜKSEKOKULU

MEKATRONİK BÖLÜMÜ

HAZIRLAYAN VE SUNAN

İSMİYE ALMAY

18MY24010

DERS ÖGRETİM GÖREVLİSİ

TARIK ASLAN

Step Motorlar,

Step motorların bir diğer ismi adım motorlardır. Konumlarını adım veya adımlar halinde değiştiren ve adımlarını hassas sinyallerle değiştiren sabit mıknatıs kutuplu motorlardır. Adım açısı yapısına göre 90° , 45° , 18° , 7.5° ve 1.8 derecelik veya daha farklı açılarda eklenebilir. Motor hızının kontrolü için frekansa uygulanacak sinsal önemlidir. Step motorların dönüş yönü uygulanan sinyalin sırasına bağlı saat yönü ise (cw) ve saat yönün tersi ise (ccw) olabilir. Motor turundaki adım sayısı alınır örnek olarak 900 adımlık motor bir tur döndüğü (360°) zaman 900 adım yapar ve bir adım açısı 360/900=0,4° olur çıkan bu değer motorun hassasiyetini gösterir

Step Motorların Avantajları

* Geri beslemeye gereksinim duymazlar.
* Açık çevrim olarak kontrol edilir.
* Hareketlerinde konum hatasına rastlanmaz.
* Gönderilen sinyallere anında cevap verirler.
* Mekanik yapısının basit olması nedeniyle bakım gerektirmez.

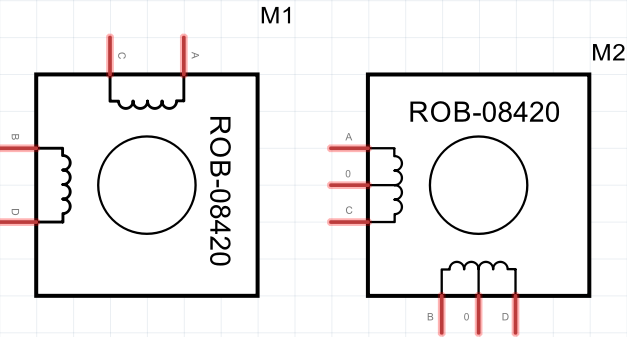
Step Motorların Dezavantajları

* Adım açıları sabittir sonradan değiştirilemez.
* Verimleri sürücülere oranla düşük olabilir.
* Çıkış momenti, güçü sınırlıdır.

Step Motorun Çeşitleri

* Sabit Mıknatıslı Step Motor
* Değişken Relüktanslı Step Motor
* Hibrit Step Motorlar

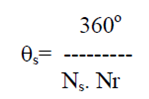
İki Fazlı Step Motorlar



Bipolar Step Motor / Unipolar Step motor

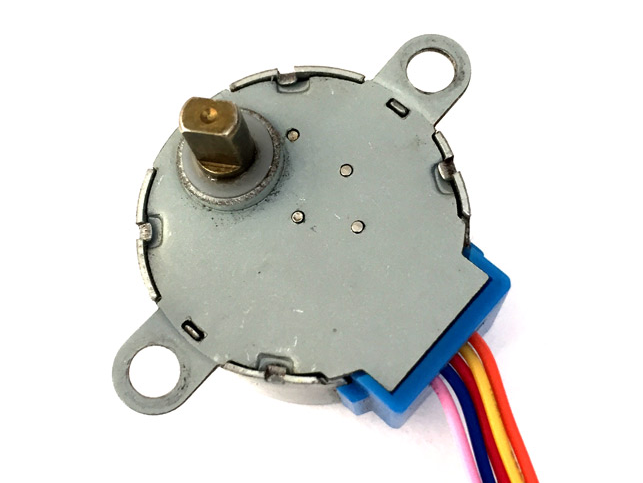
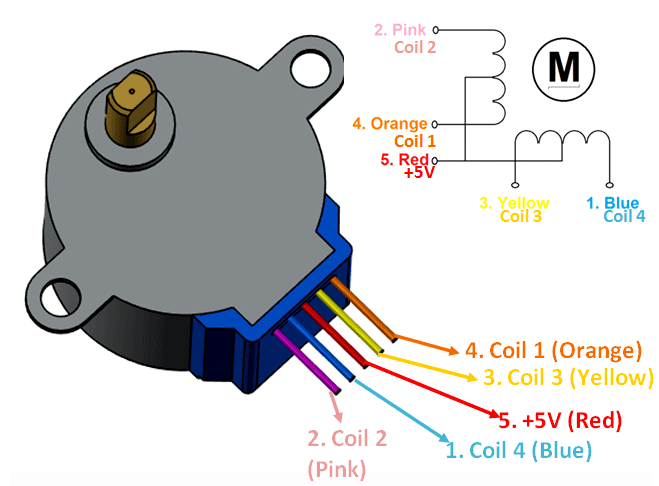
Bipolar step motorlar faz başına bir sargı bulunduran motorlardır. Unipolar step motolar ise faz başına iki sargı bulundurmatadır

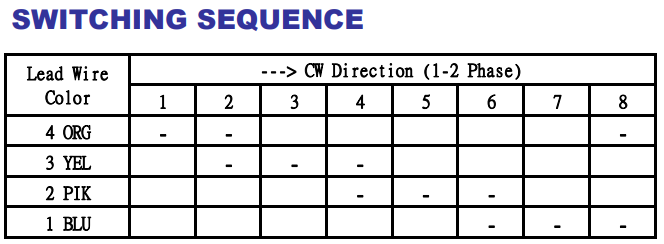
Step Motorun Adım Açısının Hesaplanması



**Formüldeki;**

**Qs=** Step motorun adım açısını  
**Ns=** Step motor faz sayısını  
**Nr=** Rotorun çıkıntılı kutup sayısını ifade etmektedir.





PROJE

# Raspberry Pi 3 İle Step Motor Kontrolü

# PROJENİN HEDEFİ

Hassas hareket gerektiren uygulamalarda kullanılan step motorun Raspberry Pi kullanarak kontrolü gerçekleştirmek ve projemizi geliştirmek**.**

# YAZILIM DİLİ

****

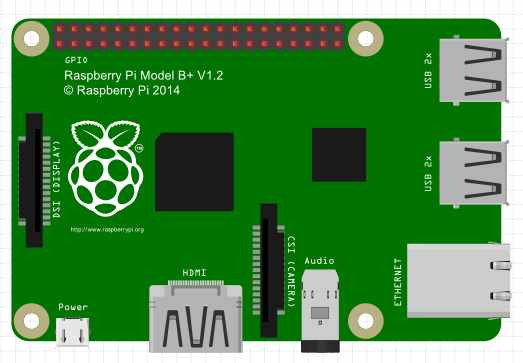
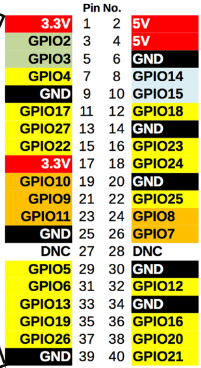
Yazılım dilimiz Python. Kısaca Python yorumsaldır nesne yönelimi yapan modülerliği olan ve okunabilen script dilidir. C ve C++ gibi dillerin aksine derlenmeden çalıştırılabilir. Diğer dillere oranla sade ve hızlıdır ve zaman kaybı yaratmaz.

# DEVRE TASARMINI İÇİN KULLANDIGIM PLATFORM

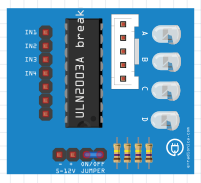
Fritzing istediği devre elemanını rahatça bulabilcegim yapacağımız devrelerde bağlantımızı paylaşacağa bileceğimiz ve hatalarımızı kontrol edeceğimiz devre modelleme programıdır.

Kullandığım Malzemeler:

* Raspberry Pi
* Step Motor
* 5V güç kaynağı
* Step motor ve sürücü kartı
* Jumper kablo

**Raspberry Pi** özellikle küçük yaştaki çocukların programlama öğrenmesi ve ilgi duyması için tasarlanmış kredi kartı büyüklüğünde olan gerçek bir bilgisayardır. Raspberry Pi tam anlamıyla bir bilgisayarın yapabileceği herşeyi yapabilmektedir. Uygun fiyatı ve küçük boyutundan dolayı herkes tarafından tercih edilmiştir.



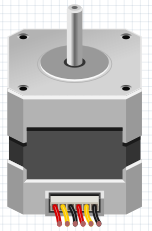
[**Step motor ve sürücü kartı**](https://www.robotistan.com/28-byj-48-reduktorlu-step-motor-ve-uln2003a-step-motor-surucu-karti)

Uln2003a redüktörlü Motor Sürücü Kartı üzerinde 4 led ve 4 kontrol pini bulunmaktadır. Bu pinler step motorun sürülmesini, Raspberry veya herhangi bir mikrodenetleyeci ile gerçekleştirebilirsiniz.



**Jumper Kablo**

Kısaca Bağlantı kablosudur. Breadboard ve Raspberry arasında bağlantı kurmak için oldukça iyidir. üç çeşit jumper bulunmaktadır.



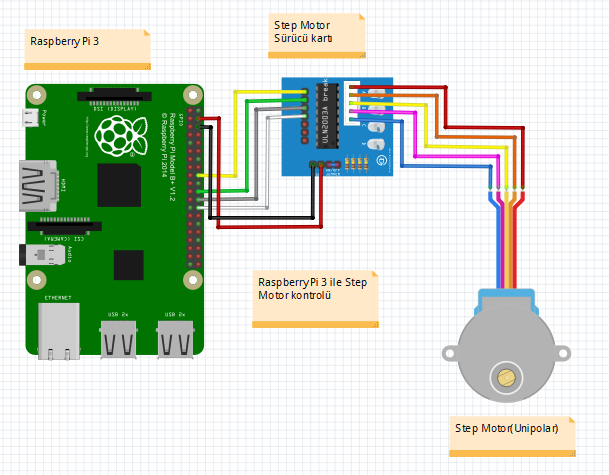
**Step motor**

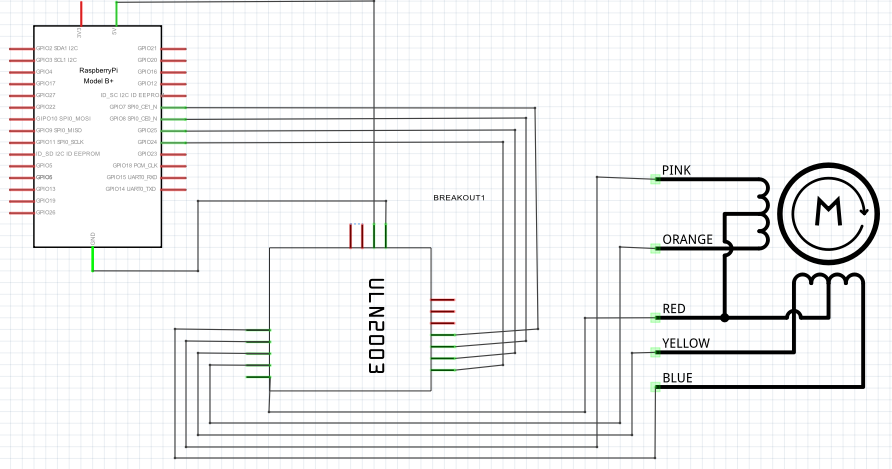


[**5V güç kaynağı (adaptör veya regülatör)**](https://www.robotistan.com/gepro-um-55)

Projemizde kurduğumuz devreye ihtiyacı olan enerjiyi karşılayan araçtır.

BAĞLANTI ŞEMALARI





KOD

import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BCM) #pin numaralandırma sistemini BCM numaralandırma sistemi olarak ayarladık

#sürücünün motoru çalıştırması için gerekli olan enable pinini 18 numaralı gpıo bağladık

enable\_pin = 18

coil\_A\_1\_pin = 24 #sarı

coil\_A\_2\_pin = 25 #yeşil

coil\_B\_1\_pin = 8 #gri

coil\_B\_2\_pin = 7 #beyaz

#bu kısımda tanımlarını yaptığımız pinleri çıkış olarak tanımlıyoruz bu kısımda

GPIO.setup(enable\_pin, GPIO.OUT)

GPIO.setup(coil\_A\_1\_pin, GPIO.OUT)

GPIO.setup(coil\_A\_2\_pin, GPIO.OUT)

GPIO.setup(coil\_B\_1\_pin, GPIO.OUT)

GPIO.setup(coil\_B\_2\_pin, GPIO.OUT)

#bu kısımda motorumuzu caşıltırıyoruz

GPIO.output(enable\_pin, 1)

#çıkış pinlerini aktif etmek için setStep fonksiyonunu oluşturuyoruz

# setStep sarım numaralarının hangilerinin aktif edileceğini belirtmekte

def setStep(w1, w2, w3, w4):

    GPIO.output(coil\_A\_1\_pin, w1)

    GPIO.output(coil\_A\_2\_pin, w2)

    GPIO.output(coil\_B\_1\_pin, w3)

    GPIO.output(coil\_B\_2\_pin, w4)

# ileri yöne doğru hareket etmesi için forward fonksiyonu tanımlarız bu fonksiyon bekleme süresini ve kaç adım gidilecegini tanımlamakta

def forward(delay, steps):

for i in range(0, steps):

setStep(1, 0, 1, 0)

time.sleep(delay)

setStep(0, 1, 1, 0)

time.sleep(delay)

setStep(0, 1, 0, 1)

time.sleep(delay)

setStep(1, 0, 0, 1)

time.sleep(delay)

#motorun geri hareketi için yani saat yönü tersi hareketi için bu kısımda bir önceki forward fonksiyonunun pin sıraları tam tersi şekline yazılır

def forward(delay, steps):

for i in range(0, steps):

setStep(0, 1, 0, 1)

time.sleep(delay)

setStep(1, 0, 0, 1)

time.sleep(delay)

setStep(1, 0, 1, 0)

time.sleep(delay)

setStep(0, 1, 1, 0)

time.sleep(delay)

#bu bölümde bize adım arası bekleme süresini ve ileriye kaç adım atılması sorulmakta

while True:

        delay = raw\_input("Time Delay (ms)?")

        steps = raw\_input("How many steps forward? ")

        forward(int(delay) / 1000.0, int(steps))

        steps = raw\_input("How many steps backwards? ")

        backwards(int(delay) / 1000.0, int(steps))

PROJENİN SONUNDA ULAŞTIGINIMIZ SONUCLAR

Projeme başlangıç olarak Step motor üzerinde araştırmalarla başladım avantajları dezavantajlarını inceledim hangi alanlarda ağırlıkla tercih edildiğine baktım ve benzer örnek çalışmalara göz atıım. sonrasında Fritzing kullandım devreyi çizmeye başladım hemen ardından devremi oluşturdum sonra yaptığım araştırmalar ve örneklerden yararlanarak kodlarımı tamamladım.