



Number: 2015-6

FEN VE TEKNOLOJİ BİLGİ PAYLAŞIMI

SCIENCE AND TECHNOLOGY INFORMATION SHARING

Article Web Page: www.ibrahimcayiroglu.com



STEP MOTOR KONTROLÜ

(STEP MOTOR CONTROL)

Burak EKİM*, İbrahim ÇAYIROĞLU**

*Karabük Üniversitesi, Mühendislik Fakültesi, Mekatronik Mühendisliği, 78050, Karabük, eekimburak@hotmail.com

**Karabük Üniversitesi, Mühendislik Fakültesi, Mekatronik Mühendisliği, 78050, Karabük, icayiroglu@yahoo.com

Anahtar Kelimeler:

Step Motor, Program ile Kontrol

Özet: Bu makalede Step Motorun genel bir tanıtımı yapıldıktan sonra uygulamalı bir örneği verilmiştir. Step Motor elektrik enerjisini mekanik enerjiye dönüştüren elektromekanik cihazdır. Bir başka deyişle açılmal konumu adımlar halinde değiştiren çok hassas sinyallerle sürülen motorlara denir. Verilen örnek direkt uygulanıp denenebilir. Kavramların daha iyi anlaşılması için hem Türkçe alternatif anlatımları hem de İngilizce karşılıkları verilmiştir.

Keywords:

Stepper engine, control with program

Abstract: In this article, we will have introduced stepper engine and then we have given practice example. Stepper engine is an electromagnetic device which changes electric energy to mechanics energy. On the other hand, we call to engine which leads to very sensitive signal because of it changes to angular position likely stepper. The examples given are applied directly. Alternative explanations for better understanding of concepts in both Turkish and English equivalents are given.

©2015 ibrahimcayiroglu.com, All rights reserved. Bu makale hakem kontrolünden geçmeden bilgi paylaşımı amacıyla yayımlanan bir dökümandır. Olabilecek hata ve yanlışlıklardan dolayı sorumluluk kabul edilmez. Makaledeki bilgiler referans gösterilip yayınlanabilir. [These articles are published documents for the purpose of information sharing without checked by the referee. Not accepted responsibility for errors or inaccuracies that may occur. The information in the article can be published by referred.]

1. Giriş

Step Motor nedir? Step motor, elektrik enerjisini dönme hareketine çeviren elektromekanik bir cihazdır. Elektrik enerjisi alındığında rotor ve buna bağlı shaft, sabit açılmal birimlerde dönmeye başlar. Step motorlar, çok yüksek hızlı anahtarlama özelliğine sahip bir sürücüye bağlıdır (step motor sürücüsü). Bu sürücü, bir encoder veya PLC'den giriş palsları alır. Alınan her giriş palsında, motor bir adım ilerler. Step motorları, bir motor turundaki adım sayısı ile anılır. Örnek olarak 400 adımlık bir step motor, bir tam tur dönüşünde 400 adım yapar. Bu durumda bir adımın açısı $360/400 = 0.9$ derecedir. Bu değer, step motorun hassasiyetinin bir göstergesidir. Bir devirdeki adım sayısı yükseldikçe step motor hassasiyeti ve dolayısıyla maliyeti artar.

Step motorlar, yarım adım modunda çalıştıklarında hassasiyetleri daha da artar. Örnek olarak 400 adım/tur değerindeki bir step motor, yarım adım modunda tur başında 800 adım yapar. Bu da 0.9 dereceye oranla hassas olan 0.45 derecelik bir adım açısı anlamına gelir. Bazı step motorlarda mikro step tekniği ile adım açılarının daha da azaltılması

söz konusudur. Ancak tork kayıpları nedeni ile bu kullanım şekli pek uygun değildir.

Malzemeler

- a) Arduino UNO (www.f1depo.com)
- b) ULN2003 Step Motor Sürücü Devresi (www.f1depo.com)
- c) 1 Adet 28BYJ-48 Step Motor (www.f1depo.com)
- d) Breadboard (www.f1depo.com)
- e) Jumper kablo (www.f1depo.com)

2. Step Motor

Step motorun tipik yapısını inceleyecek olursak karşımıza 4 adet eleman çıkar.

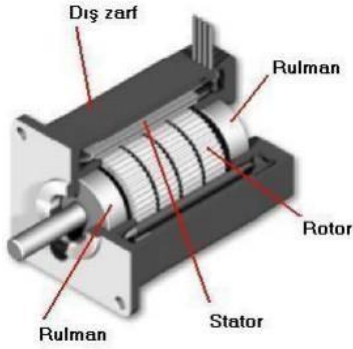
Stator: Step motorun hareketsiz olan ve sargılardan oluşan kısmıdır.

Rotor: Motorun hareketli, N ve S kutbundan oluşan ve tek parça sabit mıknatıs olan kısmıdır.

Dış Zarf: Motorun soğumasına da katkıda bulunan bu yapı, motoru dış etkilere koruyan bir yapıdır.

Rulman: Adım motorlara verilmesi gerek hareketin

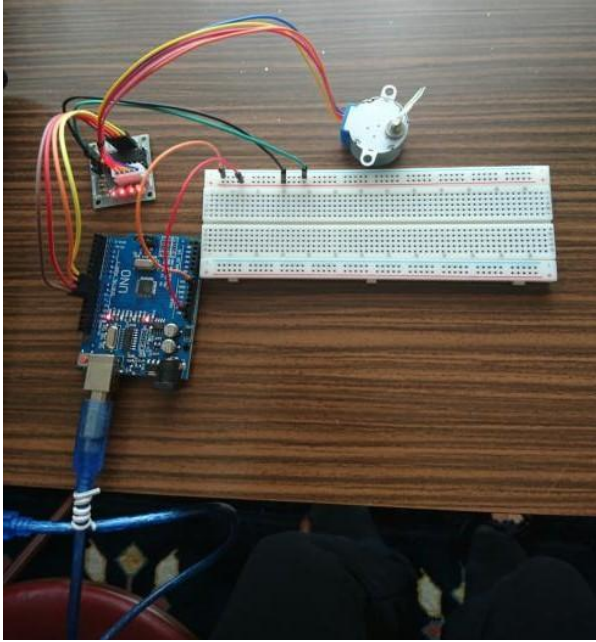
mümkün olan en az sürtünmeyle yani güçten tasarruf edilerek iletimini sağlamak için olan kısımdır.



Şekil 1. Step motorun yapısı

2.1. Step Motor Çalışma Prensibi

Step motora giriş palsi uygulandığı zaman, belli bir miktar döner ve durur. Bu dönme miktarı, motorun yapısına göre belli bir açı ile sınırlandırılmıştır. Adım açısı motorun yapısına bağlı olarak 90°, 45°, 18°, 7.5°, 1.8° veya daha değişik açılarda olabilir. Step motorda rotorun dönmesi, girişe uygulanan pals adedine bağlı olarak değişir. Girişe verilen pals adedi kadar rotor hareket eder.

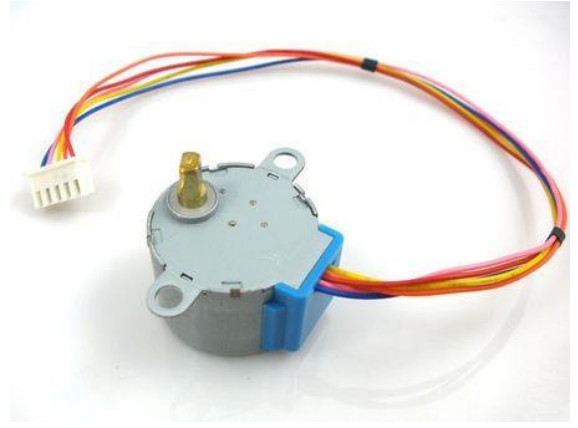


Şekil 2. Step motorun hareketi için gerekli olan bağlantılar.

2.2. 28BYJ-48 Step Motor

Bu çalışmada kullanılan 28BYJ-48 tipi step motor Şekil 3'de gösterilmiştir.

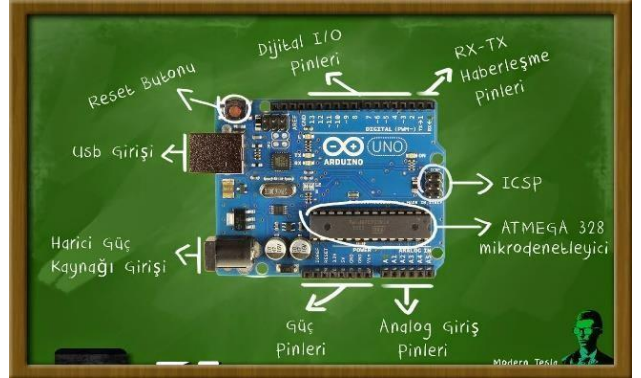
Anma voltajı 5V DC olup 4 faza sahiptir. Hız değişim oranı 1/64'tür. Adım açısı 5.625°/64'tür. Frekansı 100Hz'dir.



Şekil 3. Çalışmada kullanılan 28BYJ-48 Step Motor

3. Arduino UNO

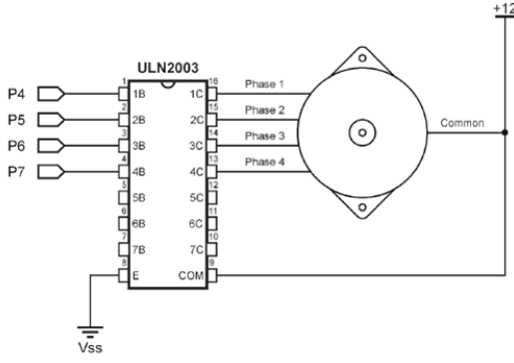
Temel olarak yapılan işlem bilgisayar üzerinden hazır kart olarak satın alınan Arduino kartın üzerindeki mikrodenetleyiciyi programlamak, giriş çıkış pinlerini kullanarak istediğimiz devreyi tasarlamaktan ibarettir. Arduino üzerinde Atmega 328 mikrodenetleyicisi bulunur. Arduino kart analog ve dijital verileri okuyup işleyebilir, çıktı üretebilir. Ancak karmaşık ve ağır işlemleri yapamaz. Arduino'yu Usb girişinden bilgisayara bağlayarak çalıştırabilir ve programlayabilirsiniz. Eğer mobil bir projede kullanılacaksa harici güç girişinden devreyi besleyebilirsiniz. (En fazla 12V a kadar güç kaynaklarının kullanılması önerilir). Arduino size 5V ve 3.3V çıkışlarını verir, çoğu elektronik elemanı çalıştırabilirsiniz ancak fazla akım ve gerilim gerektiren elemanları kullanmak için harici güç kaynağına ihtiyaç vardır.



Şekil 4. Arduino UNO kartın giriş ve çıkışları

3.1. ULN2003 Step Motor Sürücü Devresi

Aşağıda Şekil 5 verilen step motor sürücü devresinde ULN2003 entegre kullanılmıştır. Bu devrede dört giriş bulunur. Her giriş bir bobini kontrol eder.



Şekil 5. ULN2003 Step Motor Sürücü Devresinin bağlantı şeması

Şekil 6'da ise sürücü devre ve üzerinde ki elemanlar tanıtılmıştır.

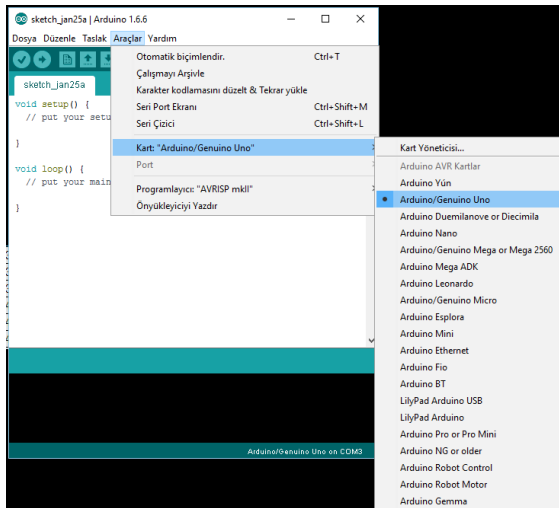


Şekil 6. ULN2003 Step Motor Sürücü Devresi.

4. Projenin Gerçekleştirilmesi

4.1. Arduino'nun Kurulumu ve Bilgisayara Bağlanması

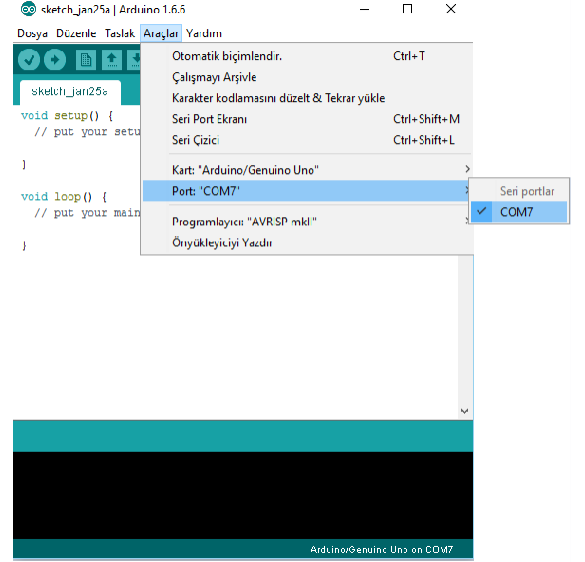
Önce <http://www.arduino.cc> sitesine girip arduino yazılımını indirerek bilgisayarımıza kuruyoruz. Daha sonra yazılımı başlattıktan sonra Şekil 7'te görüldüğü gibi Araçlar menüsüne girip kullanacağımız Arduino kart modelimizi seçiyoruz.



Şekil 7. Arduino'nun Bilgisayara Tanıtılması

4.2. Arduino ile Haberleşme

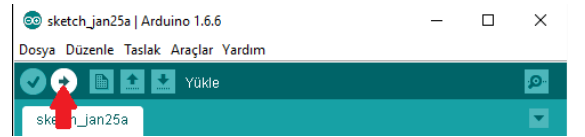
Arduino Uno bir bilgisayar ile başka bir Arduino ile ya da diğer mikrodnetleyiciler ile haberleşme için çeşitli imkanlar sunar. ATmega328 mikrodnetleyici RX ve TX pinlerinden erişilebilen UART TTL (5V) seri haberleşmeyi destekler. Kart üzerindeki bir ATmega16U2 seri haberleşmeyi USB üzerinden kanallı eder ve bilgisayardaki yazılıma sanal bir COM portu olarak görünür.



Şekil 8. Arduino ile Bilgisayarın Haberleşmesi

4.3 Programın Karta Yüklenmesi

Arduino kartlarımıza program yüklemek için önce kartımızı usb kablo yardımı ile bilgisayarımıza bağlamalıyız. Daha sonra 4.1 ve 4.2 de anlatılmış olan bağlantı ve haberleştirmeleri yapmalıyız. Daha sonra yazmış olduğumuz Arduino programımızı Şekil 9'da gösterildiği gibi ilgili düğmeye tıklayarak kartımıza yükleyebiliriz.



Şekil 9. Arduino'ya Programın Yüklenmesi

5. Uygulama

Uygulama 1: Step motorun saat yönünde çevrilmesi

Arduino UNO kart ile step motorun saat yönünde çevrilmesi için aşağıdaki kodlar kullanılmıştır.

Kodlar

```
int motorPin1 = 8; // 1. Bobinin bağlı olduğu pin
int motorPin2 = 9; // 2. Bobinin bağlı olduğu pin
int motorPin3 = 10; // 3. Bobinin bağlı olduğu pin
int motorPin4 = 11; // 4. Bobinin bağlı olduğu pin
```

```

int delayTime = 5; // Bekleme süresi

void setup() {
  pinMode(motorPin1, OUTPUT); // Bobinleri çıkış
  olarak tanımlıyoruz.
  pinMode(motorPin2, OUTPUT); // Bobinleri çıkış
  olarak tanımlıyoruz.
  pinMode(motorPin3, OUTPUT); // Bobinleri çıkış
  olarak tanımlıyoruz.
  pinMode(motorPin4, OUTPUT); // Bobinleri çıkış
  olarak tanımlıyoruz.
}

void loop() // Loop döngüsü
{
  digitalWrite(motorPin1, HIGH); // 1. Bobini 1
  konumuna getiriyoruz.
  digitalWrite(motorPin2, LOW); // 2. Bobini 0
  konumuna getiriyoruz.
  digitalWrite(motorPin3, LOW); // 3. Bobini 0
  konumuna getiriyoruz.
  digitalWrite(motorPin4, LOW); // 4. Bobini 0
  konumuna getiriyoruz.
  delay(delayTime); // Bekleme süresi
  digitalWrite(motorPin1, LOW); // 1. Bobini 0
  konumuna getiriyoruz.
  digitalWrite(motorPin2, HIGH); // 2. Bobini 1
  konumuna getiriyoruz.
  digitalWrite(motorPin3, LOW); // 3. Bobini 0
  konumuna getiriyoruz.
  digitalWrite(motorPin4, LOW); // 4. Bobini 0
  konumuna getiriyoruz.
  delay(delayTime); // Bekleme süresi
  digitalWrite(motorPin1, LOW); // 1. Bobini 0
  konumuna getiriyoruz.
  digitalWrite(motorPin2, LOW); // 2. Bobini 0
  konumuna getiriyoruz.
  digitalWrite(motorPin3, HIGH); // 3. Bobini 1
  konumuna getiriyoruz.
  digitalWrite(motorPin4, LOW); // 4. Bobini 0
  konumuna getiriyoruz.
  delay(delayTime); // Bekleme süresi
  digitalWrite(motorPin1, LOW); // 1. Bobini 0
  konumuna getiriyoruz.
  digitalWrite(motorPin2, LOW); // 2. Bobini 0
  konumuna getiriyoruz.
  digitalWrite(motorPin3, LOW); // 3. Bobini 0
  konumuna getiriyoruz.
  digitalWrite(motorPin4, HIGH); // 4. Bobini 1
  konumuna getiriyoruz.
  delay(delayTime); // Bekleme süresi

```

5.2 Uygulama 2: Step motorun saat yönünün tersine çevrilmesi

Arduino UNO kartı ile step motoru saat yönünün tersine çevirme işleminde aşağıdaki kodlar kullanılmıştır.

Kodlar

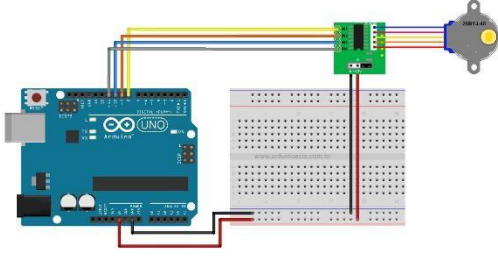
```

int motorPin1 = 8; // 1. Bobinin bağlı olduğu
pin
int motorPin2 = 9; // 2. Bobinin bağlı olduğu
pin
int motorPin3 = 10; // 3. Bobinin bağlı olduğu
pin
int motorPin4 = 11; // 4. Bobinin bağlı olduğu
pin
int delayTime = 5; // Bekleme süresi

void setup() {
  pinMode(motorPin1, OUTPUT); // Bobinleri çıkış
  olarak tanımlıyoruz.
  pinMode(motorPin2, OUTPUT); // Bobinleri çıkış
  olarak tanımlıyoruz.
  pinMode(motorPin3, OUTPUT); // Bobinleri çıkış
  olarak tanımlıyoruz.
  pinMode(motorPin4, OUTPUT); // Bobinleri çıkış
  olarak tanımlıyoruz.
}

void loop() {
  digitalWrite(motorPin1, LOW); // 1. Bobini 0
  konumuna getiriyoruz.
  digitalWrite(motorPin2, LOW); // 2. Bobini 0
  konumuna getiriyoruz.
  digitalWrite(motorPin3, LOW); // 3. Bobini 0
  konumuna getiriyoruz.
  digitalWrite(motorPin4, HIGH); // 4. Bobini 1
  konumuna getiriyoruz.
  delay(delayTime); // Bekleme süresi
  digitalWrite(motorPin1, LOW); // 1. Bobini 0
  konumuna getiriyoruz.
  digitalWrite(motorPin2, LOW); // 2. Bobini 0
  konumuna getiriyoruz.
  digitalWrite(motorPin3, HIGH); // 3. Bobini 1
  konumuna getiriyoruz.
  digitalWrite(motorPin4, LOW); // 4. Bobini 0
  konumuna getiriyoruz.
  delay(delayTime); // Bekleme süresi
  digitalWrite(motorPin1, LOW); // 1. Bobini 0
  konumuna getiriyoruz.
  digitalWrite(motorPin2, HIGH); // 2. Bobini 1
  konumuna getiriyoruz.
  digitalWrite(motorPin3, LOW); // 3. Bobini 0
  konumuna getiriyoruz.
  digitalWrite(motorPin4, LOW); // 4. Bobini 0
  konumuna getiriyoruz.
  delay(delayTime); // Bekleme süresi
  digitalWrite(motorPin1, HIGH); // 1. Bobini 1
  konumuna getiriyoruz.
  digitalWrite(motorPin2, LOW); // 2. Bobini 0
  konumuna getiriyoruz.
  digitalWrite(motorPin3, LOW); // 3. Bobini 0
  konumuna getiriyoruz.
  digitalWrite(motorPin4, LOW); // 4. Bobini 0
  konumuna getiriyoruz.
  delay(delayTime); // Bekleme süresi
}

```

Şekil 10. Uygulama 1 ve 2 için devre bağlantıları

5.3. Uygulama 3: Step motorun yön ve zaman kontrolü

Arduino UNO kartımız ile step motorumuzun yön ve zaman kontrolü yapılmaktadır. Bunu için aşağıdaki kodlar kullanılmıştır.

Kodlar

```
int motorPin1 = 8; // 1. Bobinin bağlı olduğu pin
int motorPin2 = 9; // 2. Bobinin bağlı olduğu pin
int motorPin3 = 10; // 3. Bobinin bağlı olduğu pin
int motorPin4 = 11; // 4. Bobinin bağlı olduğu pin
int delayTime = 0; // Bekleme süresi
int buton1=2; // 1. Butonun bağlı olduğu pin
int buton2=4; // 2. Butonun bağlı olduğu pin
int butondeger=0; // Butonun başlangıç konumu
int dly=0; // Hız kontrolü sırasında ki bekleme süresi

void setup() {
  pinMode(motorPin1, OUTPUT); // Bobinleri çıkış olarak tanımlıyoruz.
  pinMode(motorPin2, OUTPUT); // Bobinleri çıkış olarak tanımlıyoruz.
  pinMode(motorPin3, OUTPUT); // Bobinleri çıkış olarak tanımlıyoruz.
  pinMode(motorPin4, OUTPUT); // Bobinleri çıkış olarak tanımlıyoruz.
  pinMode(buton1, INPUT); // Butonları giriş olarak tanımlıyoruz.
  pinMode(buton2, INPUT); // Butonları giriş olarak tanımlıyoruz.
}

void loop()
{
  dly=digitalRead(buton2); // Buton 1'in basılı olup olmadığını kontrol eder.
  if(dly==HIGH) // Buton 1'e basılı ise bu döngüyü uygula
  {
    delayTime=2; // Bekleme süresi
  }
  else{ // Buton 1'e basılı değil ise bu döngüyü uygula
```

```
    delayTime=5; // Bekleme süresi
  }

  butondeger=digitalRead(buton1); // Buton 2'nin basılı olup olmadığını kontrol eder.

  if(butondeger==HIGH) // Buton 2'ye basılı ise bu döngüyü uygula
  {
    digitalWrite(motorPin1, HIGH); // 1. Bobini 1 konumuna getiriyoruz.
    digitalWrite(motorPin2, LOW); // 2. Bobini 0 konumuna getiriyoruz.
    digitalWrite(motorPin3, LOW); // 3. Bobini 0 konumuna getiriyoruz.
    digitalWrite(motorPin4, LOW); // 4. Bobini 0 konumuna getiriyoruz.
    delay(delayTime); // Bekleme süresi
    digitalWrite(motorPin1, LOW); // 1. Bobini 0 konumuna getiriyoruz.
    digitalWrite(motorPin2, HIGH); // 2. Bobini 1 konumuna getiriyoruz.
    digitalWrite(motorPin3, LOW); // 3. Bobini 0 konumuna getiriyoruz.
    digitalWrite(motorPin4, LOW); // 4. Bobini 0 konumuna getiriyoruz.
    delay(delayTime); // Bekleme süresi
    digitalWrite(motorPin1, LOW); // 1. Bobini 0 konumuna getiriyoruz.
    digitalWrite(motorPin2, LOW); // 2. Bobini 0 konumuna getiriyoruz.
    digitalWrite(motorPin3, HIGH); // 3. Bobini 1 konumuna getiriyoruz.
    digitalWrite(motorPin4, LOW); // 4. Bobini 0 konumuna getiriyoruz.
    delay(delayTime); // Bekleme süresi
    digitalWrite(motorPin1, LOW); // 1. Bobini 0 konumuna getiriyoruz.
    digitalWrite(motorPin2, LOW); // 2. Bobini 0 konumuna getiriyoruz.
    digitalWrite(motorPin3, LOW); // 3. Bobini 0 konumuna getiriyoruz.
    digitalWrite(motorPin4, HIGH); // 4. Bobini 1 konumuna getiriyoruz.
    delay(delayTime); // Bekleme süresi
  }
  else if(butondeger==LOW) // Buton 2'ye basılı ise bu döngüyü uygula
  {
    digitalWrite(motorPin1, LOW); // 1. Bobini 0 konumuna getiriyoruz.
    digitalWrite(motorPin2, LOW); // 2. Bobini 0 konumuna getiriyoruz.
    digitalWrite(motorPin3, LOW); // 3. Bobini 0 konumuna getiriyoruz.
    digitalWrite(motorPin4, HIGH); // 4. Bobini 1 konumuna getiriyoruz.
    delay(delayTime); // Bekleme süresi
    digitalWrite(motorPin1, LOW); // 1. Bobini 0 konumuna getiriyoruz.
    digitalWrite(motorPin2, LOW); // 2. Bobini 0
```

konumuna getiriyoruz.

```
digitalWrite(motorPin3, HIGH); // 3. Bobini 1  
konumuna getiriyoruz.
```

```
digitalWrite(motorPin4, LOW); // 4. Bobini 0  
konumuna getiriyoruz.
```

```
delay(delayTime); // Bekleme süresi
```

```
digitalWrite(motorPin1, LOW); // 1. Bobini 0  
konumuna getiriyoruz.
```

```
digitalWrite(motorPin2, HIGH); // 2. Bobini 1  
konumuna getiriyoruz.
```

```
digitalWrite(motorPin3, LOW); // 3. Bobini 0  
konumuna getiriyoruz.
```

```
digitalWrite(motorPin4, LOW); // 4. Bobini 0  
konumuna getiriyoruz.
```

```
delay(delayTime); // Bekleme süresi
```

```
digitalWrite(motorPin1, HIGH); // 1. Bobini 1  
konumuna getiriyoruz.
```

```
digitalWrite(motorPin2, LOW); // 2. Bobini 0  
konumuna getiriyoruz.
```

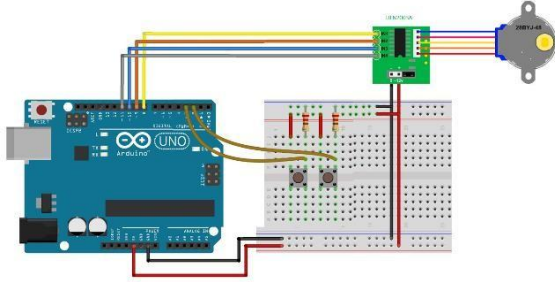
```
digitalWrite(motorPin3, LOW); // 3. Bobini 0  
konumuna getiriyoruz.
```

```
digitalWrite(motorPin4, LOW); // 4. Bobini 0  
konumuna getiriyoruz.
```

```
delay(delayTime); // Bekleme süresi
```

```
}
```

```
}
```



Şekil 11. Uygulama 3 için devre bağlantıları

4. SONUÇ

Yapılan projede bir mikrodenetleyici kart ve step motor sürücüsü ile step motor kontrolünün nasıl yapılacağını öğrendik. Bu projede edinilen bilgiler çeşitli projelerde kullanılıp daha kapsamlı projeler kolaylıkla yapılabilir.

5. KAYNAKLAR

1. hbogm.meb.gov.tr
2. www.teknobeyin.com
3. www.robotiksistem.com

The Authors



Burak Ekim is a student in Mechatronic Engineering at Karabuk University, Turkey. He is born in Merkez/MANISA. Autocad, Ansys, Visual Studio, Matlab, Plc, Arduino and is working on robotic systems..His research interests

include CAD-CAM,CNC,3D Printers and Mechatronics System .



Ibrahim Cayiroglu is an insructor in Mechatronic Engineering at Karabuk University, Turkey. He received his B.Sc. in Mechanical Engineering from Istanbul Technical University in 1991. He received his M.Sc. and Ph.D. in Computer Aided Design and Manufacturing from Kirikkale University, in 1996 and 2002, respectively. His research interests include CAD-CAM, Software and Mechatronic Systems.