

ARDUİNO VE UYGULAMALAR

1.UYGULAMA LED YAKMA

Bu programı yazmak için, pin 9'un yönünü ayarlamak için pinMode () komutunu kullanın ve çıkışı yüksek (5V) yapmak için digitalWrite () kullanın.

```
LED yakan program—led.ino
const int LED=9; // pin 9 değerini tanımla
void setup()
{
pinMode (LED, OUTPUT); // Pini çıkış olarak tanıma
digitalWrite(LED, HIGH); // Pin 9'a 5V ver High=5V
}
void loop()
{
// Döngü
}
```

ARDUİNO VE UYGULAMALAR

For Döngüsü Kullanma ve Gecikme

- Bir programın parametrelerini ayarlamak için değişken değerleri değişen döngüler kullanmak sıklıkla gereklidir.
- Yeni yazdığınız program söz konusu olduğunda, farklı yanıp sönme hızlarının sisteminizin çalışmasını nasıl etkilediğini görmek için bir for döngüsü uygulayabilirsiniz.
- Çeşitli hızlar arasında geçiş yapmak için bir for döngüsü kullanarak farklı yanıp sönme hızlarını görselleştirebilirsiniz.

```
const int LED=9; // pin 9 değerini tanımla
void setup()
{
  pinMode (LED, OUTPUT); // S
}
void loop()
{
  for (int i=100; i<=1000; i=i+100) //Döngü 100'den 1000'e
  {
    digitalWrite(LED, HIGH); //Çıkış Aktif
    delay(i); //mili saniye cinsinden gecikme
    digitalWrite(LED, LOW); //Çıkış Pasif
    delay(i);
  }
}
```

ARDUİNO VE UYGULAMALAR

For Döngüsü Kullanma ve Gecikme

- **For loop bildirimini her zaman üç noktalı virgülle ayrılmış giriş içerir:**
- **For döngüsünün çalışma mantığı**
- İlk girdi döngü için dizin değişkenini ayarlar. Bu durumda, dizin değişkeni
- i ve 100 değerinde başlayacak şekilde ayarlanmış.
- İkinci girdi, döngünün ne zaman durması gerektiğini belirtir. Döngünün içeriği
- bu koşul geçerliyken tekrar tekrar yürütülür. <= Operatörü
- "eşit veya daha küçük" anlamına gelir. Yani, bu döngü için, içerik devam edecek
- i değişkeninin bugünkü değeri hala ya da daha düşük olduğu sürece yürütmek için
- 1000'e eşittir.
- Son giriş, sonunda endeks değişkenine ne olması gerektiğini belirtir
- Bu durumda, geçerli değerine artı 100 olarak ayarlanır.

ARDUİNO VE UYGULAMALAR

For Döngüsü Kullanma ve Gecikme

- `for (int i=100; i<=1000; i=i+100) //Döngü>100 değerinden 1000 değerine`

1. i'nin değeri 100'e eşittir.
2. 100 değeri 1000 değerinden küçük veya ona eşit olduğundan döngü içeriği yürütülür.
3. LED yüksek olarak ayarlanmış ve 100ms için yüksek kalıyor, i'nin mevcut değeri.
4. LED düşük olarak ayarlanmış ve 100ms için düşük kalıyor, i'nin mevcut değeri.
5. Döngünün sonunda, i 100 arttırılır, bu yüzden şimdi 200'dür.
6. 200 değeri 1000'den küçük veya ona eşittir, bu nedenle döngü tekrar eder.
7. LED yüksek olarak ayarlanmış ve 200ms için yüksek kalıyor, i'nin mevcut değeri.
8. LED düşük olarak ayarlanmıştır ve i'nin mevcut değeri 200ms için düşük kalır.
9. Döngünün sonunda, i 100 arttırılır, bu yüzden şimdi 300'dür.
10. Bu işlem 1000 i aşana ve dış döngü işlevi tekrarlanana kadar tekrarlanır,
11. i değerini 100'e geri getirme ve işlemi yeniden başlatma

Düşük=0V

Yüksek=5V

ARDUİNO VE UYGULAMALAR

analogWrite() ile Darbe Geniřlięi Modlasyonu (PWM)

Pinlerin digital kontroln (0V veya 5V) geręekleřtirdik.

Bu, yanıp snen LED'ler, kontrol rleleri ve eęirme motorları iin sabit bir hızda mkemmeldir. Peki ya 0V veya 5V dıřında bir voltaj ıkarmak isterseniz?

Dijital-analog dnřtrc (DAC) entegre devre veya yerleřik DAC'ye sahip bir Arduino kullanmadıęınız srece yapamazsınız. (Arduino DUE)

Ancak, darbe geniřlik modlasyonu (**PWM**) adı verilen bir numara kullanarak analog ıkıř deęerleri oluřturmaya olduka yaklařabilirsiniz.

Her Arduino'daki pinler, belirli evre birimleriyle kullanıldıęında saf bir analog sinyali taklit edebilen PWM sinyalleri oluřturmak iin **analogWrite ()** komutunu kullanabilir. Bu pinler

Arduino zerinde ~ ile iřaretlenmiřtir.

Arduino Uno'da pin **3, 5, 6, 9, 10 ve 11** PWM pimleridir.

ARDUİNO VE UYGULAMALAR

analogWrite() ile Darbe Geniřlięi Modülasyonu (PWM)

PWM ıkıřı 8 bitlik bir deęerdir. Bařka bir deyiřle, 0 ila 2^8-1 veya 0 ila 255 arasında deęerler yazabilirsiniz. LED devrenizde, ıkıřı 255 olarak ayarlamak tam parlaklıęa neden olur ve 0, LED'in sönmesine neden olur. parlaklık bu iki deęer arasında deęiřir.



ARDUİNO VE UYGULAMALAR

analogWrite() ile Darbe Genişliği Modülasyonu (PWM)

```
const int LED=9; // Define LED for Pin 9
void setup()
{
  pinMode (LED, OUTPUT); // Set the LED pin as an output
}
void loop()
{
  for (int i=0; i<256; i++) //0'dan 255 değerine
  {
    analogWrite(LED, i); //PWM
    delay(10);
  }
  for (int i=255; i>=0; i--) //255'den 0 değerine
  {
    analogWrite(LED, i); //PWM
    delay(10); //10 mili saniye gecikme
  }
}
```

ARDUİNO VE UYGULAMALAR

analogWrite() ile Darbe Geniřliđi Modülasyonu (PWM)

PWM kontrolü, saf analog kontrolü taklit etmek için birçok durumda kullanılabilir, ancak aslında bir analog sinyale ihtiyacınız olduđunda her zaman kullanılamaz.

Örneđin, PWM deđişken akımlarda dođru akım (DC) motorları çalıştırmak için çok kullanışlıdır.

PWM bir kare dalganın görev döngüsünü modüle ederek çalışır (açılıp kapanan bir sinyal). Görev döngüsü (duty cycle), kare dalganın yüksek ile alçak arasındaki süresinin yüzdesini ifade eder. Yüzde 50'lik bir görev döngüsüne sahip kare dalgada zamanın yüksek yarısı ve zamanın düşük yarısı aynıdır (bu **analogWrite(9, 127)** çalıştırılarak gerçekleştirilir).

ARDUİNO VE UYGULAMALAR

analogWrite() ile Darbe Geniřliđi Modülasyonu (PWM)

AnalogWrite () komutu, kendisine ilettiđiniz deđere bađlı olarak kare dalganın görev dōngüsünü ayarlar:

- AnalogWrite () ile 0 deđeri yazmak, görev dōngüsü yüzde 0 olan (her zaman dūřük) bir kare dalgayı gösterir.
- 255 deđerinin yazılması, görev dōngüsü yüzde 100 (daima yüksek) olan kare bir dalgayı gösterir.
- 127 deđeri yazmak, görev dōngüsü yüzde 50 olan bir kare dalgayı gösterir (zamanın yüksek yarısı, zamanın dūřük yarısı).

ARDUİNO VE UYGULAMALAR

analogWrite() ile Darbe Geniřlięi Modülasyonu (PWM)

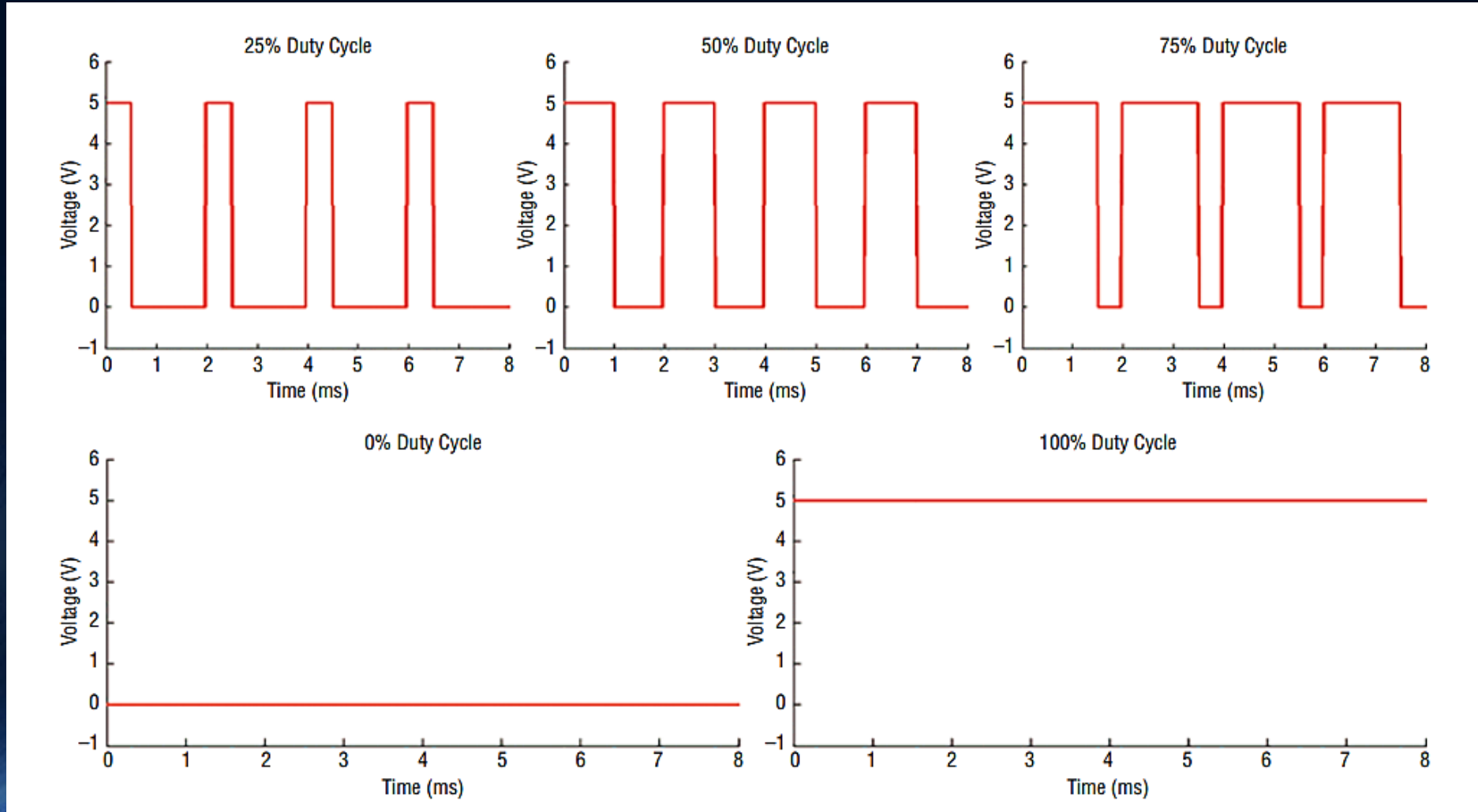
Döngüsü yüzde 25 olan bir sinyal için, zamanın yüzde 25'inin yüksek ve zamanın yüzde 75'inin düşük olduğunu bilinmektedir. Bu kare dalganın frekansı, Arduino Uno durumunda, yaklaşık 490 Hz'dir. Başka bir deyişle, sinyal her saniye yaklaşık 490 kez yüksek (5V) ve düşük (0V) arasında deęişir.

FREKANS VE PERİYOT

“Periyot” sıklıkla frekans yerine alternatif bir sinyali tanımlamak için de kullanılır. Bu sinyalin "periyodu", her döngüyü tamamlama zamanıdır. Periyot, frekansı 1 saniyeye bölerek kolayca hesaplanabilir. $1/490 \text{ Hz} = .002 \text{ saniye} = \text{döngü başına } 2 \text{ milisaniye}$

ARDUİNO VE UYGULAMALAR

analogWrite() ile Darbe Genişliği Modülasyonu (PWM)



ARDUİNO VE UYGULAMALAR

Dijital Girişlerin Okunması

Bir sonraki adım, gerçek zamanlı olarak etkileşim kurabilmeniz için anahtarlar ve düğmeler gibi dijital girişleri okumaktır.

