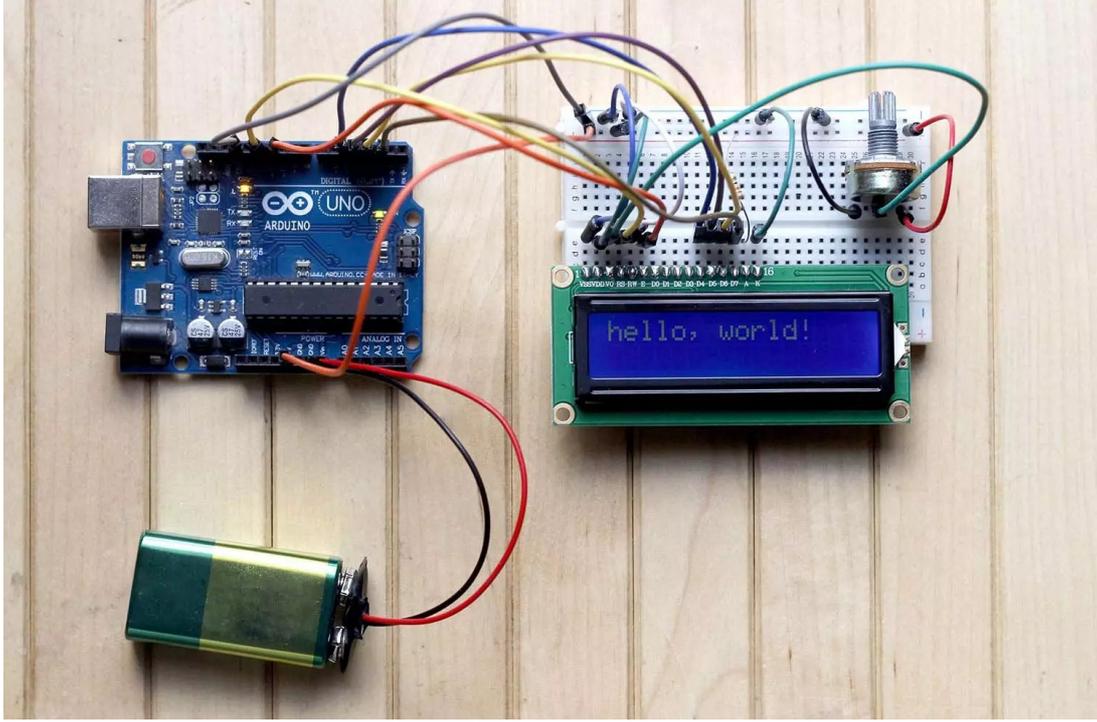


# Arduino Dersleri 10: 16x2 LCD Ekran (Kurulum ve Programlama Rehberi)

Yazar **B. Tuğçe Küçüködük** - 19 Kasım 2018

Merhaba arkadaşlar,

Bu yazıda, Arduino ve LCD bağlantılarının nasıl yapılacağını anlatacağım ve size programlayabileceğiniz bütün farklı yolları göstereceğim. Metni nasıl yazdıracağınızı, metni nasıl kaydıracağınızı, özel karakterleri nasıl yazacağınızı, metni nasıl kıracağınızı ve metni nasıl konumlandıracağınızı göstereceğim. Bu bilgiler output alacağınız herhangi bir proje için harika olacak ve bu sayede projenizi daha interaktif ve ilginç bir hale getirebileceksiniz.



Kullanacağımız ekran çok uygun bir fiyata alabileceğiniz 16x2 LCD ekran. Ekranın 16x2 olarak adlandırılması LCD'nin 2 satıra sahip olduğu ve satır başına 16 karakter görüntüleyebileceği anlamına geliyor. Yani ekran aynı anda 32 adet karakter görüntüleyebiliyor. Bunun dışında kaydırma yaparak 32 karakterden fazlasını da görüntüleyebilirsiniz.

Bu yazıdaki kodlar, standart Hitachi HD44780 sürücüsünü kullanan LCD'ler için yazılmıştır. LCD'nizin 16 iğnesi varsa, muhtemelen Hitachi HD44780 sürücüsüne sahiptir.

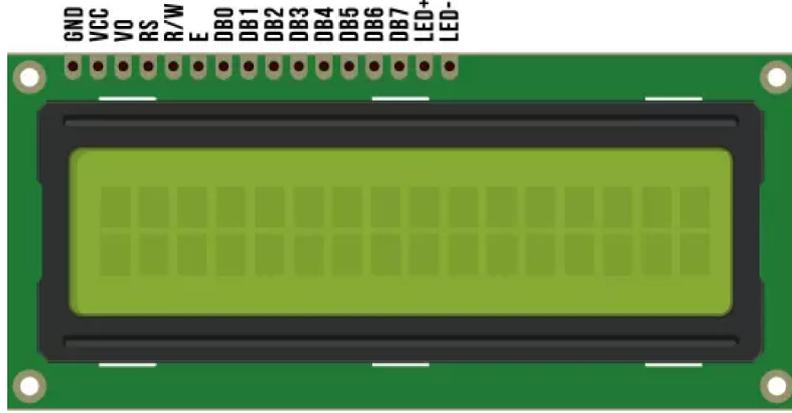
## Gerekli malzemeler:

- Arduino ([Arduino Nano](#), [Arduino Uno](#))
- [Breadboard](#)
- 16x2 LCD Ekran ([Yeşil Üzerine Siyah](#), [Mavi Üzerine Beyaz](#))
- [10 kΩ potansiyometre](#)
- [Erkek-erkek jumper kablo](#)

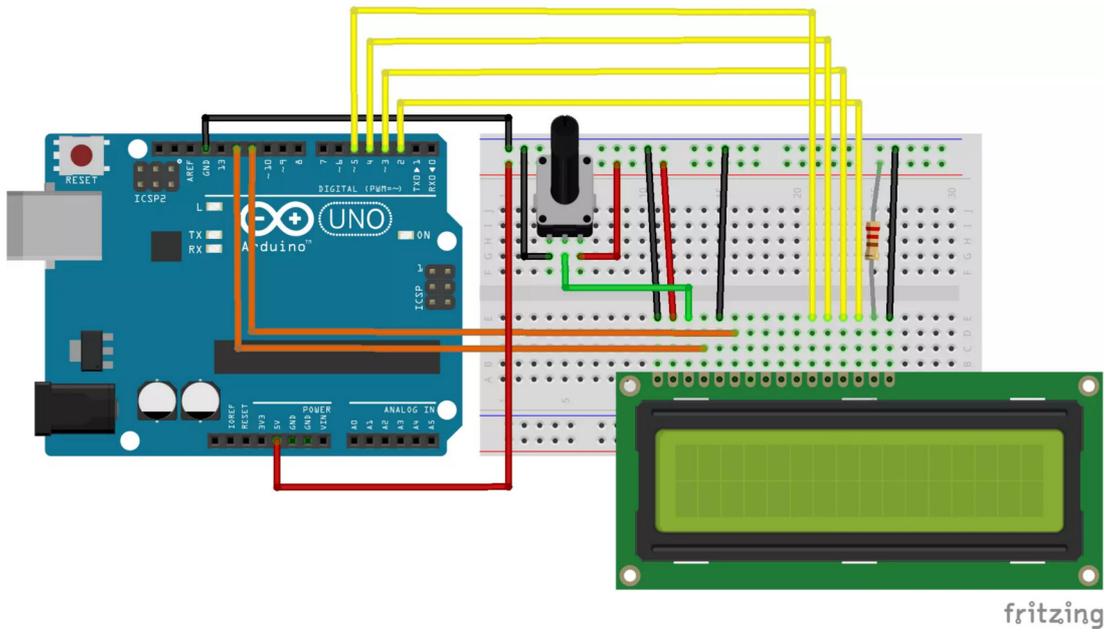
## LCD ve Arduino Bağlantıları

Kullandığım LCD'deki pinlerin diyagramını aşağıda bulacaksınız. LCD ekranımızda 16 adet pin var. Kullanacağımız ekrana göre pinler ekranın üst, alt veya her iki tarafında da yer alabilir. Çok

nadir olarak bazı ekranlarda ise arka aydınlatma ışığı bulunmadığından 14 adet pin yer almaktadır. 15 ve 16 numaralı pinler, ekran aydınlatması bulunan ekranlarda arka ışığı yakmak için kullanılır. Her bir pinden Arduino'ya olan bağlantılar aynı olacaktır, ancak pinleriniz LCD üzerinde farklı şekilde düzenlenebilir. Bunun için kullandığınız LCD'nin datasheet'ine bakabilirsiniz.



Ayrıca breadboard'a bağlamadan önce LCD'nize 16 pinlik bir başlık lehimlemeniz gerekebilir. LCD'yi Arduino'nuzla bağlamak için aşağıdaki şemayı takip edin:



Arduino ve LCD Ekran Devre Bağlantıları

Yukarıdaki diyagramdaki direnç, arka ışık parlaklığını ayarlar. Tipik bir değer olan 220 Ohm direnç kullandım ama diğer değerler de işe yarayacaktır. Daha küçük dirençler arka ışığı daha parlak hale getirir.

Potansiyometre ekran kontrastını ayarlamak için kullanılır. Ben genellikle 10K Ohm potansiyometre kullanıyorum ancak diğer değerler de çalışır.

Ekranla ilgili tüm teknik bilgileri içeren 16x2 LCD'nin datasheet'ini incelemek isterseniz [buradan](#) indirebilirsiniz.

## Arduino Programlama

Aşağıdaki kodların tamamı Arduino IDE ile önceden yüklenmiş olarak gelen LiquidCrystal kütüphanesini kullanır. Bir kütüphanenin kullanılabilmesi için programa dahil edilmesi gerekir. Aşağıdaki kodlardaki 1. Satır olan #include <LiquidCrystal.h> komutu bu işe yarar. Bir programda bir kütüphane eklediğinizde, kütüphanedeki tüm kod, programınızın koduyla birlikte Arduino'ya yüklenir. Şimdi bağlantılarınızı tamamladıysanız programlamaya başlayabiliriz. Basit bir test programıyla başlayalım. Bu program ekrana "hello, world!" yazdıracaktır. Aşağıdaki kodu Arduino'ya yükleyelim:

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  lcd.begin(16, 2);
  lcd.print("hello, world!");
}

void loop() {

}
```

LCD Ekranınız şu şekilde gözükmelidir:



## LCD Görüntüleme Seçenekleri

LiquidCrystal kütüphanesinin kullanabileceğimiz 19 farklı fonksiyonu vardır. Bu fonksiyonların yazının konumunu değiştirme, ekran boyunca yazıyı kaydırma veya görüntüyü açıp kapatmak gibi işlevleri vardır. Şimdi bu fonksiyonları kısaca inceleyelim:

### LiquidCrystal ( )

İşlevi Arduino'nun LCD'ye bağlanmak için kullandığı pinleri ayarlar. LCD'yi kontrol etmek için Arduino'nun dijital pinlerinden herhangi birini kullanabilirsiniz. Arduino pin numaralarını bu sırayla parantez içine koyunuz: LiquidCrystal (RS, E, D4, D5, D6, D7). RS, E, D4, D5, D6, D7 LCD pinleridir.

Örneğin, LCD pini D7'nin Arduino pin 12'ye bağlanmasını istediğinizi varsayalım. D7'nin yerine şu şekilde "12" koyunuz: LiquidCrystal (RS, E, D4, D5, D6, 12). Bu işlev, programın void setup () bölümünün önüne yerleştirilmelidir.

### **lcd.begin ( )**

Bu işlev LCD'nin boyutlarını ayarlar. Programın void setup ( ) bölümünde herhangi bir LiquidCrystal işlevinin önüne yerleştirilmelidir. Satır ve sütunların sayısı lcd.begin (sütunlar, satırlar) olarak belirtilir. 16×2 LCD için, lcd.begin (16, 2) kullanmanız gerekiyor ve 20×4 LCD için lcd.begin (20, 4) kullanmanız gerekiyor.

### **lcd.clear ( )**

Bu işlev, LCD'de halihazırda görüntülenen herhangi bir metni veya veriyi siler. lcd.print ( ) ile lcd.clear ( ) ögesini ve void loop ( ) bölümünde delay ( ) işlevini kullanırsanız, basit bir yanıp sönen metin programı oluşturabilirsiniz:

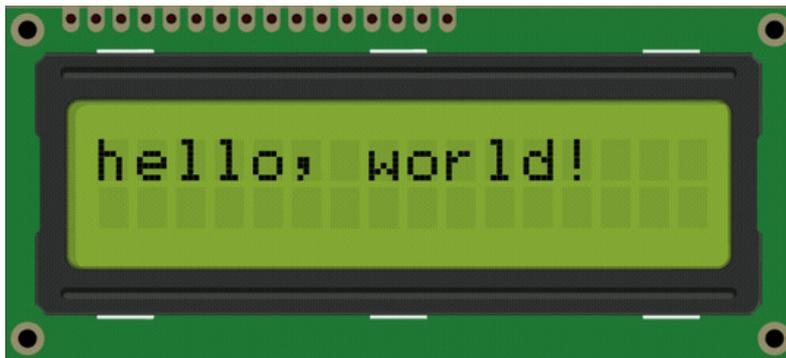
```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  lcd.begin(16, 2);
}

void loop() {
  lcd.print("hello, world!");
  delay(500);
  lcd.clear();
  delay(500);
}
```

Bu kodu Arduino'larınıza yüklediğinizde LCD ekranınız aşağıdaki gibi gözükecektir:



### **lcd.home ( )**

Bu işlev, imleci ekranın sol üst köşesine yerleştirir ve sonraki metni bu konuma yazdırır. Örneğin, aşağıdaki kod "hello, world!" metninin ilk üç harfini X ile değiştirir:

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  lcd.begin(16, 2);
```

```
lcd.print("hello, world!");  
}  
  
void loop() {  
  lcd.home();  
  lcd.print("XXX");  
}
```

Ekranınız şu şekilde gözükecektir:



### **lcd.setCursor ( )**

Benzer, ancak lcd.home ( ) 'dan daha kullanışlıdır. Bu işlem, imleci (ve herhangi bir yazılı metni) ekranın herhangi bir yerine yerleştirir. Programınızın void setup ( ) veya void loop ( ) bölümünde kullanılabilir.

İmleç konumu, lcd.setCursor (sütun, satır) ile tanımlanır. Sütun ve satır koordinatları sıfırdan başlar (sırasıyla 0-15 ve 0-1). Örneğin, yukarıdaki "hello, world!" programının void setup ( ) bölümünde lcd.setCursor (2, 1) kullanılarak, "hello, world!" alt satıra yazdırılır ve sağa iki alan kaydırılır.

```
#include <LiquidCrystal.h>  
  
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);  
  
void setup() {  
  lcd.begin(16, 2);  
  lcd.setCursor(2, 1);  
  lcd.print("hello, world!");  
}  
  
void loop() {  
  
}
```

Yukarıdaki kodu Arduino'ya yüklediğinizde ekranınız aşağıdaki gibi gözükecektir:



### **lcd.write ( )**

LCD'ye farklı tipte veri yazmak için bu işlevi kullanabilirsiniz, örneğin bir sıcaklık sensöründen okuma veya bir GPS modülünden gelen koordinatlar. Kendiniz oluşturduğunuz özel karakterleri yazdırmak için de kullanabilirsiniz. Programınızın void setup ( ) veya void loop ( ) bölümünde lcd.write ( ) ögesini kullanabilirsiniz.

### **lcd.print ( )**

Bu işlev, LCD'ye metin yazdırmak için kullanılır. Programın void setup ( ) bölümünde veya void loop ( ) bölümünde kullanılabilir.

Harfleri ve sözcükleri yazdırmak için, metnin etrafına tırnak işaretleri ( " ") yerleştirin. Örneğin; hello, world!ü yazdırmak için yukarıda lcd.print ("hello,world!") kullanmıştık.

Sayıları yazdırmak için tırnak işaretleri gerekmez. Örneğin; 123456789'u yazdırmak için lcd.print (123456789) yazabilirsiniz.

lcd.print ( ); sayıları ondalık, ikili, onaltılık ve sekizlik bazlarda yazdırabilir. Örneğin:

- lcd.print (100, DEC) "100" yazar.
- lcd.print (100, BIN) "1100100" yazar.
- lcd.print (100, HEX) "64" yazar.
- lcd.print (100, OCT) "144" yazar.

### **lcd.Cursor( )**

Bu işlev görünür bir imleç oluşturur. İmleç, LCD'ye yazdırılacak bir sonraki karakterin altına yerleştirilmiş yatay bir çizgidir.

lcd.noCursor ( ) işlevi imleci kapatır. lcd.cursor ( ) ve lcd.noCursor ( ) birçok metin giriş alanında gördüğünüze benzer yanıp sönen imleci yazdırmak için void loop ( ) bölümünde birlikte kullanılabilir.

```
#include <LiquidCrystal.h>
```

```
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

```
void setup() {  
  lcd.begin(16, 2);  
  lcd.print("hello, world!");  
}
```

```
void loop() {  
  lcd.cursor();  
  delay(500);  
  lcd.noCursor();  
  delay(500);  
}
```

Yukarıdaki kodu yüklediğinizde ekranınız şu şekilde gözükecektir:



İmleçler, `lcd.setCursor ( )` işleviyle ekranın herhangi bir yerine yerleştirilebilir. Bu kod "hello, world!" deki ünlem işaretinin hemen altına yanıp sönen bir imleç yerleştirir:

```
#include <LiquidCrystal.h>  
  
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);  
  
void setup() {  
  lcd.begin(16, 2);  
  lcd.print("hello, world!");  
}  
  
void loop() {  
  lcd.setCursor(12, 1);  
  lcd.cursor();  
  delay(500);  
  lcd.setCursor(12, 1);  
  lcd.noCursor();  
  delay(500);  
}
```



### **lcd.blink ( )**

Bu işlev, döngü başına yaklaşık 500 milisaniyede bir yanıp sönen blok stili bir imleç oluşturur. Bunu void loop ( ) bölümünde kullanın. lcd.noBlink ( ) işlevi yanıp sönen blok imlecini devre dışı bırakır.

### **lcd.display ( )**

Bu işlev, LCD ekrana yazdırılmış herhangi bir metin veya imleci açar. lcd.noDisplay ( ) işlevi, LCD'nin belleğinden silmeden LCD'ye yazdırılan herhangi bir metni veya imleci kapatır.

Bu iki işlev, yanıp sönen bir metin efekti oluşturmak için void loop ( ) bölümünde birlikte kullanılabilir. Bu kod "hello, world!" metnini açıp kapatacak:

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  lcd.begin(16, 2);
  lcd.print("hello, world!");
}

void loop() {
  lcd.display();
  delay(500);
  lcd.noDisplay();
  delay(500);
}
```

### **lcd.scrollDisplayLeft ( )**

Bu işlev, LCD'ye yazdırılan her şeyi alır ve sola taşır. Bunu takiben bir gecikme komutu ile void loop ( ) bölümünde kullanılmalıdır. İşlev, ilk karaktere geri dönmeden önce metni 40 alan sola taşır. Bu kod, "hello, world!" metnini, karakter başına bir saniyede olmak üzere sola taşır:

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  lcd.begin(16, 2);
  lcd.print("hello, world!");
}

void loop() {
  lcd.scrollDisplayLeft();
  delay(1000);
}
```



### **lcd.scrollDisplayRight( )**

Bu işlev, lcd.scrollDisplayLeft ( ) gibi davranır ancak metni sağa taşır.

### **lcd.autoscroll( )**

Bu işlev bir metin dizisi alır ve dizinin karakter sayısının artışlarıyla sağdan sola kaydırır.

Örneğin; 3 karakter uzunluğunda bir metin diziniz varsa, her adımda metni 3 alan sola kaydırır:

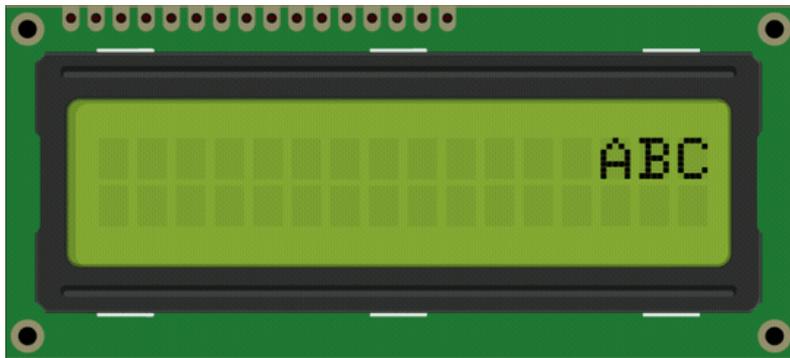
```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  lcd.begin(16, 2);
}

void loop() {
  lcd.setCursor(0, 0);
  lcd.autoscroll();
  lcd.print("ABC");
  delay(500);
}
```

Bu kodu çalıştırdığınızda ekranınız aşağıdaki gibi gözükecektir:



lcd.scrollDisplay ( ) işlevleri gibi; metin, tekrarlanmadan önce en fazla 40 karakter uzunluğunda olabilir. İlk bakışta, bu işlev kullanışsız gözükyor olabilir, ancak özel karakterlerle animasyon oluşturmak için çok yararlı olabilir.

### **lcd.noAutoscroll( )**

lcd.noAutoscroll ( ), lcd.autoscroll ( ) işlevini kapatır. Kayan metin veya animasyon dizileri oluşturmak için void loop ( ) bölümünde lcd.autoscroll ( ) işlevinden önce veya sonra bu işlevi kullanabilirsiniz.

### lcd.rightToLeft()

Bu işlev, metnin ekrana yazdırıldığı yönü ayarlar. Varsayılan mod, lcd.leftToRight ( ) komutunu kullanarak soldan sağa doğrudur, ancak metnin ters yönde yazılışının yararlı olduğu bazı durumlar bulabilirsiniz:

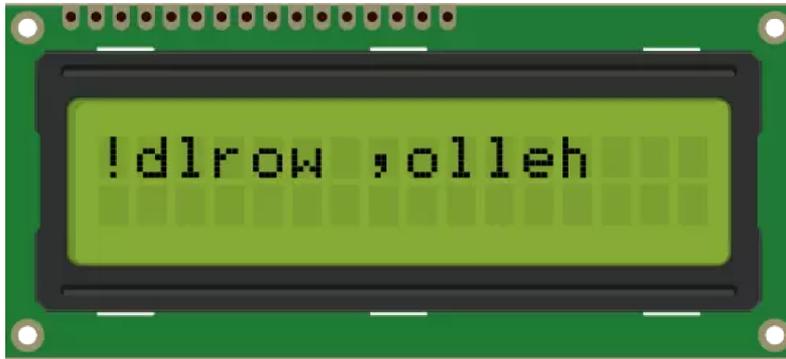
```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  lcd.begin(16, 2);
  lcd.setCursor(12, 0);
  lcd.rightToLeft();
  lcd.print("hello, world!");
}

void loop() {

}
```



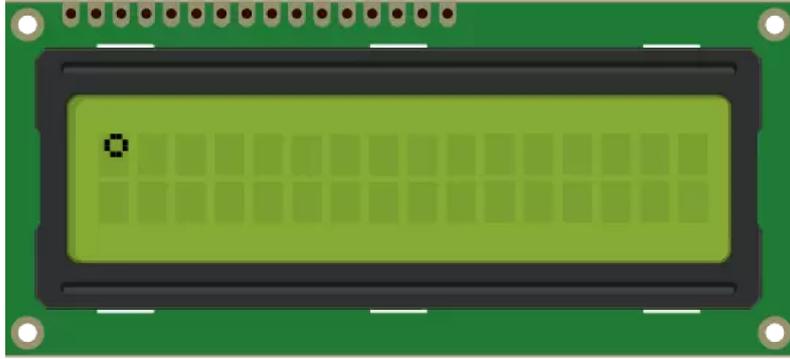
### lcd.createChar ( )

Bu komut kendi özel karakterlerinizi oluşturmanıza izin verir. 16x2 LCD'nin her bir karakterinin 5 piksel genişliği ve 8 piksel yüksekliği vardır. Tek bir programda 8 farklı özel karakter tanımlanabilir. Kendi karakterlerinizi tasarlamak için, özel karakterinizin ikili karakter matrisini **LCD karakter jeneratörü**nden yapmanız veya kendiniz haritalamanız gerekir. Aşağıdaki kod bir derece sembolü (°) oluşturur:

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
byte customChar[8] = {
  0b00110,
  0b01001,
```

```
0b01001,  
0b00110,  
0b00000,  
0b00000,  
0b00000,  
0b00000  
};  
  
void setup()  
{  
  lcd.createChar(0, customChar);  
  lcd.begin(16, 2);  
  lcd.write((uint8_t)0);  
}  
  
void loop() {  
}
```



16x2 LCD ile yapabileceğiniz çok güzel şeyler var. Bu işlevlerden bazılarını birleştirmeyi denemenizi öneriyorum. □ Sorularınızı sormaktan çekinmeyin lütfen. Görüşmek üzere.

---

---