**Arduino ile Nokia 5110 LCD Ekran Kullanımı – Resim ve Şekil Görüntüleme**

Merhaba arkadaşlar, bu yazımızda arduino ile nokia 5110 lcd ekran kullanımı ile resim yazdırma konusunu inceleyeceğiz.

Cep telefonlarının düğmelerinin olduğu iPhone öncesi günleri hatırlıyor musunuz? O dönemlerde Nokia bu küçük LCD’leri 3310 ve 5110 cep telefonlarında kullandı.

Teknoloji değiştikçe, bu ekranlar sonunda DIY dünyasında yeni yerlerini buldu. Nokia 5110 ekran küçük, ucuz, kullanımı kolay olması, oldukça düşük güç tüketimi ve bitmap’lerin yanı sıra metin görüntüleyebildiğinden zamanla hobi uzmanları arasında popülerleştiler.

### Satın Alınabilir Ürünler;

* [**Arduino Uno R3**](https://robocombo.com/Arduino-UNO-R3-CH340)
* [**Nokia 5110 LCD Ekran Modülü**](https://robocombo.com/nokia-5110-lcd-modul)
* [**Jumper Kablo**](https://robocombo.com/40li-Erkek-Erkek-Jumper-Kablo-20cm,PR-1150.html)
* [**Mini Breadbord**](https://robocombo.com/Arama?1&kelime=mini%20breadboard)
* [**1k Direnç**](https://robocombo.com/1K-14W-Direnc-10-Adet,PR-275.html)
* [**10k Direnç**](https://robocombo.com/10K-14W-Direnc-10-Adet,PR-282.html)
* [**330R Direnç**](https://robocombo.com/330R-14W-Direnc-10-Adet,PR-772.html)

## Nokia 5110 LCD Ekran Sürücüsü – PCD8544

5110 ekran modülünün kalbinde, Philips – **PCD8544** tek yongalı düşük güçlü CMOS LCD sürücü denetleyicisi **bulunmaktadır**.

Çip, **84 × 48** piksel grafik ekran sürmek için tasarlanmıştır. SPI’ye benzer bir seri veri yolu arabirimi aracılığıyla mikrodenetleyicilere arabirim sağlar.

PCD8544 kontrol cihazının çok yönlülüğü, yonga üzerinde LCD kaynağı ve bias voltajları üreterek düşük güç tüketimiyle güce duyarlı uygulamalar için uygundur. Normalde, LCD yalnızca 6 ila 7mA kadar düşük tüketir.

Veri sayfasına göre, bu çip **2.7 ila 3.3 V** aralığında çalışır ve 3v iletişim seviyelerine sahiptir. Bu nedenle, Arduino gibi herhangi bir 5V mikrodenetleyicisi için bir tür logic seviye dönüştürücü gereklidir (aksi takdirde ekran zarar görebilir).

## 5110 Arka Işık

5110 lcd ekran ayrıca farklı renklerde bir arka plan ışığıyla birlikte gelir. Kırmızı, yeşil, mavi ve beyaz.

Arka ışık, ekranın kenarlarında dört LED’den başka bir şey değildir.

LCD’nin arka ışığını değiştirmek isterseniz, arka tarafındaki metal klipsleri iterek LCD’yi karttan çıkarın. Ekran kapandığında, ekranın kenarlarına lehimlenmiş dört LED’i göreceksiniz. LED’leri istediğiniz renkli LED’lerle değiştirin.

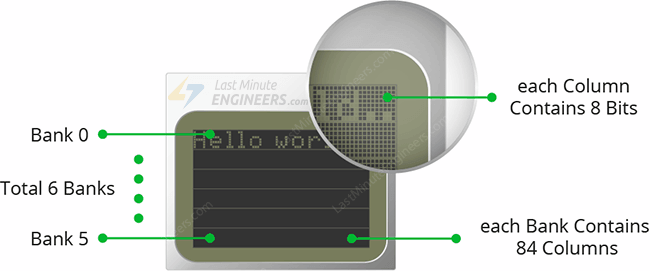
|  |
| --- |
| **Uyarı:** Bu LCD ekranların herhangi bir akım sınırlama direnci ile birlikte gelmeyen birçok sürümü vardır. Bu, güç kaynağını bağlarken dikkatli olmanız gerektiği anlamına gelir. Önlem olarak, ‘Arka Işık’ pimiyle seri olarak [330Ω akım sınırlama direnci](https://robocombo.com/330R-14W-Direnc-10-Adet,PR-772.html) yerleştirebilirsiniz. |

## Nokia 5110 LCD Bellek Haritası

PCD8544 LCD sürücüsünde, görüntülenecek bit desenini tutan ekran için yerleşik bir 504 bayt **Grafik Görüntü Veri RAM’i** (GDDRAM) bulunur. Bu hafıza alanı 6 bölüm olarak (0’dan 5’e kadar) düzenlenmiştir. Her bölüm 84 sütun / segment (0’dan 83’e kadar) içerir. Ve her sütun 8 bit veri saklayabilir (0’dan 7’ye).

**6 banka x 84 segment x 8 bit veri = 4032 bit = 504 bayt**

Bankalar(GDDRAM bölümleri), segmentler ve veriler içeren tüm bellek haritası aşağıda vurgulanmıştır.

****

*Nokia 5110 LCD Ekran Pixelleri*

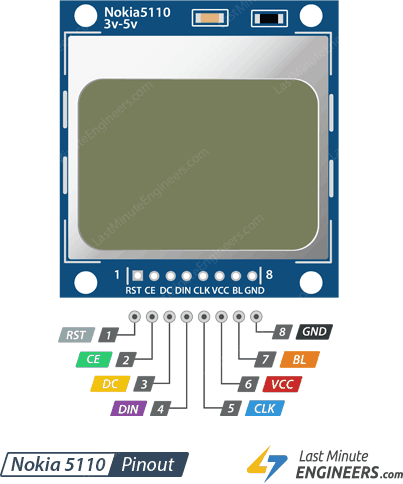
Her bit, ekrandaki programlı olarak **AÇIK** veya **KAPALI** hale getirilebilen belirli bir pikseli temsil eder.

### Nokia 5110 Ekran Özellikleri:

|  |  |
| --- | --- |
| Görüntü Teknolojisi | Nokta Vuruşlu LCD |
| MCU Arayüzü | SPI |
| Ekran boyutu | 1.5 inç |
| çözüm | 84 × 48 piksel |
| Çalışma gerilimi | 2.7V – 3.3V |
| Çalışma Akımı | Maks. 50mA |
| Görüş açısı | 180 ° |

## Nokia 5110 LCD Ekran Modülü Pinout

Bağlantı ve örnek kodlara dalmadan önce Pinout’a bir göz atalım.

****

*Nokia 5110 LCD Ekran Pinout*

RST pimi ekranı sıfırlar. Ayrıca bu pimi Arduino sıfırlamaya bağlayarak ekranı otomatik olarak sıfırlar.

CE (Chip Enable) pimi, aynı SPI veriyolunu paylaşan birçok bağlı cihazdan birini seçmek için kullanılır.

D/C (Veri / Komut) pimi, ekrana aldığı verilerin bir komut veya görüntülenebilir veri olup olmadığını gösterir.

DIN SPI arayüzü için seri veri pinidir.

CLK SPI arayüzü için bir seri saat pinidir.

VCC pin, Arduino’daki 3.3V volt pinine bağladığımız LCD için güç sağlayan pindir.

BL (Arka ışık) pimi ekranın arka ışığını kontrol eder. Parlaklığını kontrol etmek için bir potansiyometre ekleyebilir veya bu pimi PWM özellikli Arduino pinlerine bağlayabilirsiniz.

GND Arduino topraklarına bağlanmalıdır.

## Nokia 5110 LCD Ekran Modülünü Arduino Uno’ya Bağlama – Kablolama

Kod yüklemeye ve ekrana veri göndermeye başlamadan önce, ekranı Arduino’ya bağlayalım.

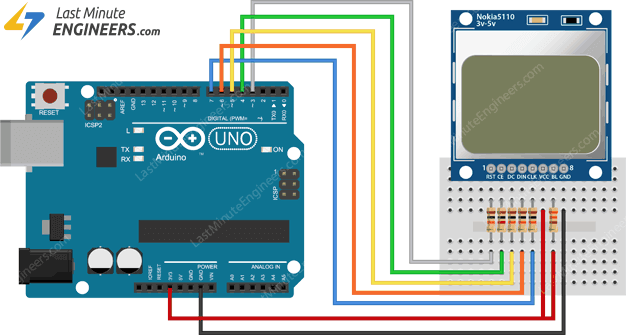
Bağlantılar oldukça basittir. Yazılım SPI’sini uyguladığımız için esnek pin seçeneklerine sahibiz. Veri iletim pinlerini herhangi bir dijital I / O pinine bağlayabilirsiniz. Seri saat (CLK), seri veri (DIN), veri / komut (D/C), çip etkinleştirme (CE) ve sıfırlama (RST) pinleri, Arduino’daki pin 3’e kadar bağlanır.

Ancak ne yazık ki, LCD 3v ile çalışır. Bu yüzden bu pinleri Arduino’ya doğrudan bağlayamıyoruz. Biraz korumaya ihtiyacımız var. Seviyeleri değiştirerek yapılabilir.

Seviyeleri kaydırmanın en ucuz ve en kolay yollarından biri, her veri iletim pimi ile birlikte dirençler eklemektir. Sadece CLK, DIN, D/C ve RST pimleri arasına [10kΩ direnç](https://robocombo.com/10K-14W-Direnc-10-Adet,PR-282.html) ve CE arasına [1kΩ direnç](https://robocombo.com/1K-14W-Direnc-10-Adet,PR-275.html) ekleyin.

Son olarak, arka ışık (BL) pimi [330Ω akım sınırlama direnci](https://robocombo.com/330R-14W-Direnc-10-Adet,PR-772.html) ile 3.3V’a bağlanır. Parlaklığını kontrol etmek istiyorsanız, bir [potansiyometre](https://robocombo.com/potansiyomet-ve-trimpot-74) ekleyebilir veya bu pimi PWM özellikli herhangi bir Arduino pimine bağlayabilirsiniz.

Aşağıdaki şemada her şeyin nasıl kablolanacağı gösterilmektedir.

**[](https://hayaletveyap.com/arduino-ile-nokia-5110-lcd-ekran-kullanimi-resim-yazdirma/nokia-5110-lcd-ekran-modul-kullanimi-arduino-baglantisi-kablolama/)**

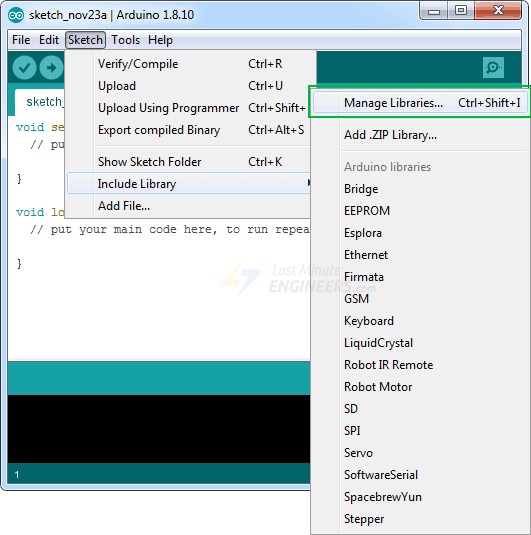
*Nokia 5110 LCD Ekran Modülünü Arduino UNO ile Kablolama*

Bununla birlikte, şimdi bir kod yüklemeye ve ekran baskısını almaya hazırsınız.

## Nokia 5110 LCD Ekran Modülü Kullanımı için Arduino Kitaplığı Yükleme

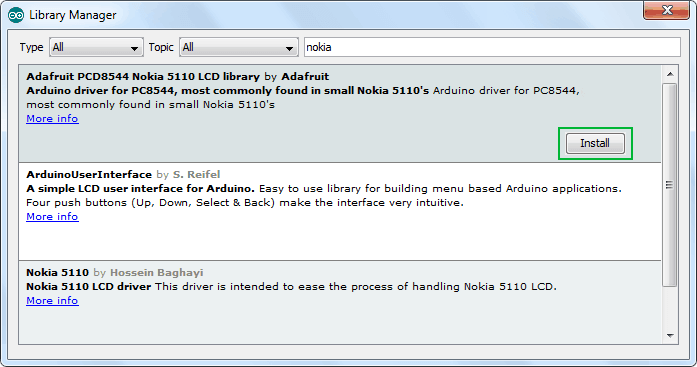
PCD8544 LCD denetleyicide esnek ancak karmaşık sürücüler vardır. PCD8544 kontrol cihazını kullanmak için bellek adresleme konusunda geniş bilgi gereklidir. Neyse ki, [Adafruit PCD8544 Nokia 5110 LCD kütüphanesi](https://github.com/adafruit/Adafruit-PCD8544-Nokia-5110-LCD-library), tüm karmaşıklıkları gizlemek için yazılmıştır, böylece ekranı kontrol etmek için basit komutlar verebiliriz.

Kitaplığı yüklemek için **Taslak> Kitaplığı Dahil Et> Kitaplıkları Yönet…** ‘e gidin. Kitaplık Yöneticisi’nin kitaplıklar dizinini indirmesini ve yüklü kitaplıklar listesini güncellemesini bekleyin.

****

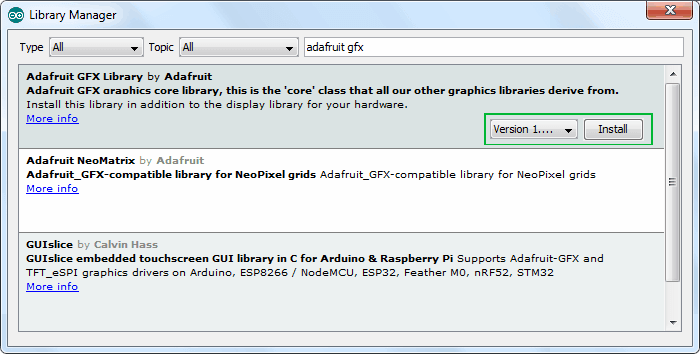
*Nokia 5110 LCD Ekran Modülü Kullanımı için Arduino Kitaplığı Yükleme*

‘**Nokia**‘ yazarak aramanızı filtreleyin. Birkaç sonuç çıkacaktır. **Adafruit PCD8544 Nokia 5110 LCD** kütüphanesini bulun. Bu girişi tıklayın ve ardından **Yükle**‘yi seçin.

****

*Nokia 5110 LCD Ekran Modülü Kullanımı için Arduino Kitaplığı Arama*

Bu kitaplık, alt düzey işlevleri işleyen donanıma özgü bir kitaplıktır. Noktalar, çizgiler, daireler, dikdörtgenler gibi grafikleri görüntülemek için Adafruit GFX Kütüphanesi ile eşleştirilmesi gerekir.

****

*Nokia 5110 LCD Ekran Modülü Kullanımı için Adafruit GFX Kitaplığı Arama*

\*\*\*Artı Olarak “Adafruit\_BusIO” kütüphaneside yüklenmesi gerekiyor\*\*\*

**PCD8544**, ekran için yerleşik bir **GDDRAM**‘a sahip olmasına rağmen, içeriğini okuyamıyoruz. Matematiksel işlemleri gerçekleştirmek için ekran arabelleğini değiştirmek mümkün değildir.

Alternatif olarak, kütüphane ATmega328P’den tampon olarak 504 bayt bellek ayırır. Böylece, ekran arabelleğini değiştirebilir ve daha sonra ATmega328P’nin belleğinden PCD8544 denetleyicisinin dahili belleğine toplu bir aktarım gerçekleştirebiliriz.

## Arduino Kodu – Metin Görüntüleme

Şimdi ilginç şeyler geliyor!

Aşağıdaki test kodu ile ‘Merhaba Dünya!’ mesajı görüntülenir. Ayrıca kod, şunları da içerir;

* Ters çevrilmiş metni görüntüleme
* Numaraları Görüntüleme
* Bazlı Numaraları Görüntüleme (Ondalık Sayılar)
* ASCII sembollerini görüntüleme
* Metin Döndürme

Bu bilgiler Nokia 5110 LCD ekranın nasıl kullanılacağı hakkında tam bir anlayış sağlayacaktır ve daha pratik deneyler ve projeler için temel oluşturabilir. Kodu deneyin ve aşağıda ayrıntılı olarak parçalara ayıracağız.

#include <**SPI**.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_PCD8544.h>

// Declare LCD object for software SPI

// Adafruit\_PCD8544(CLK,DIN,D/C,CE,RST);

Adafruit\_PCD8544 display = Adafruit\_PCD8544(7, 6, 5, 4, 3);

int rotatetext = 1;

void setup()   {

**Serial**.begin(9600);

 //Initialize Display

 display.begin();

 // you can change the contrast around to adapt the display for the best viewing!

 display.setContrast(57);

 // Clear the buffer.

 display.clearDisplay();

 // Display Text

 display.setTextSize(1);

 display.setTextColor(BLACK);

 display.setCursor(0,0);

 display.println("Hello world!");

 display.display();

 delay(2000);

 display.clearDisplay();

 // Display Inverted Text

 display.setTextColor(WHITE, BLACK); // 'inverted' text

 display.setCursor(0,0);

 display.println("Hello world!");

 display.display();

 delay(2000);

 display.clearDisplay();

 // Scaling Font Size

 display.setTextColor(BLACK);

 display.setCursor(0,0);

 display.setTextSize(2);

 display.println("Hello!");

 display.display();

 delay(2000);

 display.clearDisplay();

 // Display Numbers

 display.setTextSize(1);

 display.setCursor(0,0);

 display.println(123456789);

 display.display();

 delay(2000);

 display.clearDisplay();

 // Specifying Base For Numbers

 display.setCursor(0,0);

 display.print("0x"); display.print(0xFF, HEX);

 display.print("(HEX) = ");

 display.print(0xFF, DEC);

 display.println("(DEC)");

 display.display();

 delay(2000);

 display.clearDisplay();

 // Display ASCII Characters

 display.setCursor(0,0);

 display.setTextSize(2);

 display.write(3);

 display.display();

 delay(2000);

 display.clearDisplay();

 // Text Rotation

 while(1)

 {

 display.clearDisplay();

 display.setRotation(rotatetext);  // rotate 90 degrees counter clockwise, can also use values of 2 and 3 to go further.

 display.setTextSize(1);

 display.setTextColor(BLACK);

 display.setCursor(0,0);

 display.println("Text Rotation");

 display.display();

 delay(1000);

 display.clearDisplay();

 rotatetext++;

 }

}

void loop() {}

### Kodlarımızı inceleyelim

Kod, üç kütüphane vizini ekleyerek başlar. **SPI.h**, **Adafruit\_GFX.h** ve **Adafruit\_PCD8544.h**. Ardından, bir LCD nesnesi oluşturmamız gerekiyor. Bu nesne 5 parametre alır ve LCD’nin CLK, DIN, D/C, CE ve RST  pinine hangi Arduino pinlerinin bağlandığını belirtir. rotatetext ile anlamlı olacak değişkenleri de tanımladık.

#include <**SPI**.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_PCD8544.h>

// Initialize LCD object for software SPI

// Adafruit\_PCD8544(CLK,DIN,D/C,CE,RST);

Adafruit\_PCD8544 display = Adafruit\_PCD8544(7, 6, 5, 4, 3);

int rotatetext = 1;

Kurulum işlevinde: LCD nesnesini begin() işlevini kullanarak başlatmamız gerekir. Ayrıca setContrast(value) 0-100 arasında herhangi bir yerde olabilen değer ile fonksiyon kullanarak ekranın kontrastını ayarlamamız gerekir. Bununla birlikte, 50-60 arasındaki değer harika sonuçlar verir.

Ardından, ilk mesajımızı ekrana yazdırmadan önce arabelleği temizleriz.

//Initialize Display

display.begin();

// you can change the contrast around to adapt the display for the best viewing!

display.setContrast(57);

// Clear the buffer.

display.clearDisplay();

### Basit Metin Görüntüleme (Merhaba Dünya)

****

*Arduino ile Nokia 5110 LCD Ekran Kullanımı – Yazı Yazma*

// Display Text

display.setTextSize(1);

display.setTextColor(BLACK);

display.setCursor(0,0);

display.println("Hello world!");

display.display();

delay(2000);

display.clearDisplay();

Ekranda metin görüntülemek için yazı tipi boyutunu ayarlamamız gerekir. Bu, setTextSize() yazı tipi boyutunu (1’den başlayarak) parametre olarak çağırarak ve geçirerek yapılabilir.

Ardından, setTextColor() işlevini çağırarak yazı tipi rengini ayarlamamız gerekir. Koyu arka plan için **BLACK** parametresini ve parlak arka plan için **WHITE** yazın. Şimdi mesajı yazdırmadan önce işlevi çağırarak imleç konumunu ayarlamamız gerekiyor, bu da setCursor(X,Y) ile olur.

Ekrandaki pikseller yatay (**X**) ve dikey (**Y**) koordinatlarıyla adreslenir. Koordinat sistemi başlangıç ​​noktasını (0,0) sol üst köşeye yerleştirir, **pozitif X** sağa ve **pozitif Y** aşağı doğru artar.

Seri monitörde veri yazdırdığımız gibi mesajı ekrana yazdırmak için basit print(” “) veya println(” “) işlev kullanabiliriz . Unutmayın, println()imleci yeni satıra götürür.

Kütüphanenin ekran arabelleğinde (saniyede 100 kareden fazla) son derece hızlı matematiksel işlemler yapabilmesi için, yazdırma işlevlerine yapılan çağrılar ekran arabelleğinin içeriğini hemen PCD8544 denetleyicisine aktarmaz. Kütüphaneye, ATmega328P’deki ekran arabelleğinden PCD8544 denetleyicisinin dahili belleğine toplu aktarım gerçekleştirmesi talimatını vermek için bir komut gereklidir, bu da display() işlevidir. Bellek aktarılır aktarılmaz, LCD ekranda ekran arabelleğine karşılık gelen pikseller görünecektir.

### Ters Çevrilmiş Metni Görüntüleme

****

*Arduino ile Nokia 5110 LCD Ekran Kullanımı – Ters Yazı Yazma*

// Display Inverted Text

display.setTextColor(WHITE, BLACK);

display.setCursor(0,0);

display.println("Hello world!");

display.display();

delay(2000);

display.clearDisplay();

Ters çevrilmiş metni görüntülemek için setTextColor(FontColor,BackgroundColor) işlevini çeğıracağız. Dikkat ettiyseniz, bu işlevde daha önce yalnızca bir parametre geçtiğimizi görebilirsiniz. Ancak şimdi iki parametre geçiriyoruz. Bu, [Function overloading](https://en.wikipedia.org/wiki/Function_overloading) adı verilen bir şey nedeniyle yapılmaktadır. İşlev aşırı yüklenmesi olarak bilinen bu olay, aynı isimde ancak farklı parametre kümesiyle birden çok işlev oluşturma yeteneğidir. Aşırı yüklenmiş bir işleve yapılan çağrılar, geçirilen parametrelere bağlı olarak bu işlevin belirli bir uygulamasını çalıştırır.

Bizde geçen setTextColor(BLACK, WHITE) kodu, dolu arka plan üzerinde siyah metin oluşturacaktır.

### Yazı Tipi Boyutunu Ölçeklendirme

****

*Nokia 5110 Ekran Modülünde Yazı Tipi Boyutunu Değiştirme*

// Scaling Font Size

display.setTextColor(BLACK);

display.setCursor(0,0);

display.setTextSize(2);

display.println("Hello!");

display.display();

delay(2000);

display.clearDisplay();

Bu öğreticinin başlarında, setTextSize() yazı tipi boyutunu ayarlamak için işlevini çağırdık ve parametre olarak 1’i geçtik. Negatif olmayan bir tam sayı ileterek yazı tipini ölçeklendirmek için bu işlevi kullanabilirsiniz.

Karakterler 5:7 oranında oluşturulur. Yani, yazı tipi boyutu 1 olursa, metni karakter başına 5 × 7 piksel oluşturacak; 2 geçmesi, metni karakter başına 10×14 pikselde görüntüler.

|  |
| --- |
| **Not:** Adafruit\_GFX kütüphanesi yazı tipini oluşturmaktan sorumludur. Varsayılan olarak tek aralıklı yazı tipi seçilidir. Ancak, Adafruit GFX kütüphanesinin daha yeni sürümleri alternatif yazı tiplerini kullanma olanağı sunar. Kütüphaneyle birlikte birkaç alternatif yazı tipi gelir, ayrıca yenilerini ekleme yeteneği vardır. |

### Rakamları Görüntüleme

****

*Arduino ile Nokia 5110 LCD Ekran Kullanımı – Rakamlar*

// Display Numbers

display.setTextSize(1);

display.setCursor(0,0);

display.println(123456789);

display.display();

delay(2000);

display.clearDisplay();

Numaralar LCD ekranda sadece çağrı print() veya println()işlevi ile görüntülenebilir. Bu işlevlerin aşırı yüklenmiş bir uygulaması 32 bit işaretsiz int’i kabul eder, bu nedenle yalnızca 0 ile 4,294,967,295 arasındaki sayıları görüntüleyebilirsiniz. Bu kadar rakam da gayet işinizi görecektir

### Sayılar için Taban Belirtme

****

*Sayılara Format Belirleme*

// Specifying Base For Numbers

display.setCursor(0,0);

display.print("0x"); display.print(0xFF, HEX);

display.print("(HEX) = ");

display.print(0xFF, DEC);

display.println("(DEC)");

display.display();

delay(2000);

display.clearDisplay();

print() ve println() işlevlerinde, kullanılacak tabanı (format) belirten isteğe bağlı ikinci parametre bulunur; izin verilen değerler BIN (ikili veya taban 2), OCT (sekizli veya taban 8), DEC (ondalık veya taban 10), HEX (onaltılık veya taban 16) ‘dır. Kayan nokta sayıları için, bu parametre kullanılacak ondalık basamak sayısını belirtir.

**Örneğin:**

* print (78, BIN) “1001110” verir
* print (78, OCT) “116” verir
* print (78, DEC) “78” verir
* print (78, HEX) “4E” verir
* println (1.23456, 0) “1” verir
* println (1.23456, 2) “1.23” verir
* println (1.23456, 4) “1.2346” verir

### ASCII Sembollerini Görüntüleme

****

*Nokia 5110 LCD Ekranda Sembol ve Şekiller*

// Display ASCII Characters

display.setCursor(0,0);

display.setTextSize(2);

display.write(3);

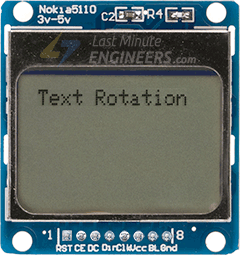
display.display();

delay(2000);

display.clearDisplay();

print() ve println() işlevleri ekrana insan tarafından okunabilen ASCII metni olarak veri gönderirken, write () işlevi ekrana ikili veri gönderir. Bu nedenle, ASCII sembollerini görüntülemek için bu işlevi kullanabilirsiniz. Örneğimizde, 3 numaralı gönderen kalp simgesi görüntülenecektir.

### Metin(Yazı) Döndürme

****

*Nokia 5110 Ekran Yazı Döndürme*

// Text Rotation

while(1)

{

 display.clearDisplay();

 display.setRotation(rotatetext);

 display.setTextSize(1);

 display.setTextColor(BLACK);

 display.setCursor(0,0);

 display.println("Text Rotation");

 display.display();

 delay(1000);

 display.clearDisplay();

 rotatetext++;

}

setRotation() Fonksiyonunu çağırarak ekrandaki görüntüyü döndürebilirsiniz. Ekranınızı portre modunda görüntülemenizi veya baş aşağı çevirmenizi sağlar.

Fonksiyon 4 kardinal rotasyona karşılık gelen sadece bir parametreyi kabul eder. Bu değer 0’dan başlayarak negatif olmayan bir tamsayı olabilir. Değeri her artırdığınızda, ekranın içeriği saat yönünün tersine 90 derece döndürülür.

**Örneğin:**

* 0 – Ekranı standart yatay yönde tutar.
* 1 – Ekranı 90 ° sağa döndürür.
* 2 – Ekranı ters çevirir.
* 3 – Ekranı 90 ° sola döndürür.