**Nokia 5110 Ekran Arduino Kodu – Temel Çizimler**

Bu örnekte, bazı temel çizimleri deneyeceğiz. Bu çizim, dikdörtgenler, yuvarlak dikdörtgenler, daireler ve üçgenler dahil olmak üzere birçok çizim işlevini gösterir. Önce deneyin ve sonra biraz 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);

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();

 // Draw Rectangle

 display.drawRect(0, 0, 60, 40, BLACK);

 display.display();

 delay(2000);

 display.clearDisplay();

 //Draw Filled Rectangle

 display.fillRect(0, 0, 60, 40, BLACK);

 display.display();

 delay(2000);

 display.clearDisplay();

 //Draw Round Rectangle

 display.drawRoundRect(0, 0, 60, 40, 8, BLACK);

 display.display();

 delay(2000);

 display.clearDisplay();

 //Draw Filled Round Rectangle

 display.fillRoundRect(0, 0, 60, 40, 8, BLACK);

 display.display();

 delay(2000);

 display.clearDisplay();

 //Draw Circle

 display.drawCircle(20, 20, 20, BLACK);

 display.display();

 delay(2000);

 display.clearDisplay();

 //Draw Filled Circle

 display.fillCircle(20, 20, 20, BLACK);

 display.display();

 delay(2000);

 display.clearDisplay();

 //Draw Triangle

 display.drawTriangle(20, 0, 0, 40, 40, 40, BLACK);

 display.display();

 delay(2000);

 display.clearDisplay();

 //Draw Filled Triangle

 display.fillTriangle(20, 0, 0, 40, 40, 40, BLACK);

 display.display();

 delay(2000);

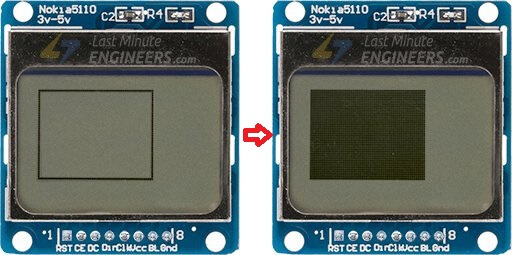
 display.clearDisplay();

}

void loop() {}

Temel çizimleri çizmek için aşağıdaki kod snippet’leri dışında, kodun çoğu (Kütüphaneler dahil ve ekranı başlatma) yukarıdaki kod örneğiyle aynıdır.

**Dikdörtgen Çizimi**

****

*Nokia Ekran İçi Boş ve Dolu Dikdörtgen Çizimi*

// Draw Rectangle

display.drawRect(0, 0, 60, 40, BLACK);

display.display();

delay(2000);

display.clearDisplay();

//Draw Filled Rectangle

display.fillRect(0, 0, 60, 40, BLACK);

display.display();

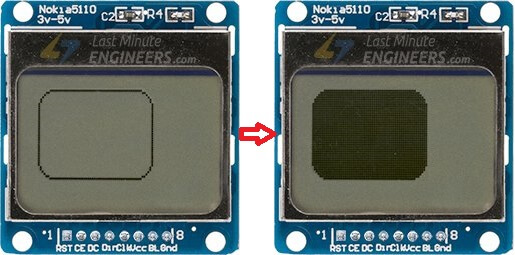
delay(2000);

display.clearDisplay();

Kopyala

drawRect() fonksiyonu kullanarak ekranda dikdörtgen çizebilirsiniz. Fonksiyon beş parametre viz alır. X koordinatı, Y koordinatı, Genişlik, Yükseklik ve renk. Aslında bu işlev, 1 piksel kenarlığı olan içi boş dikdörtgen çizer. fillRect() fonksiyonu kullanarak doldurulmuş dikdörtgen çizebilirsiniz.

**Yuvarlak Kenarlı Dikdörtgen Çizim**

****

*Nokia Ekran İçi Boş ve Dolu Yuvarlak Kenarlı Dikdörtgen Çizimi*

//Draw Round Rectangle

display.drawRoundRect(0, 0, 60, 40, 8, BLACK);

display.display();

delay(2000);

display.clearDisplay();

//Draw Filled Round Rectangle

display.fillRoundRect(0, 0, 60, 40, 8, BLACK);

display.display();

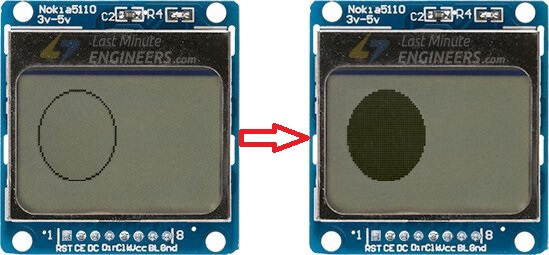
delay(2000);

display.clearDisplay();

Kopyala

drawRoundRect() fonksiyonunu kullanarak ekranda yuvarlak dikdörtgen çizebilirsiniz. Bu fonksiyon, drawRect()bir ek parametre dışında köşe ile aynı parametreleri alır (Köşe yuvarlama yarıçapı). Aslında bu işlev, 1 piksel kenarlığı olan içi boş yuvarlak dikdörtgen çizer. fillRoundRect() fonksiyonunu kullanarak da dolu yuvarlak dikdörtgen çizebilirsiniz.

**Çember, Daire, Yuvarlak**

****

*Nokia Ekran İçi Boş ve Dolu Yuvarlak, Daire, Çember*

//Draw Circle

display.drawCircle(20, 20, 20, BLACK);

display.display();

delay(2000);

display.clearDisplay();

//Draw Filled Circle

display.fillCircle(20, 20, 20, BLACK);

display.display();

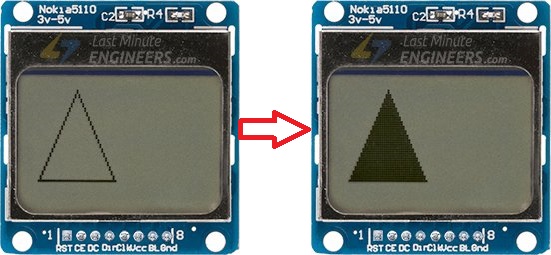
delay(2000);

display.clearDisplay();

Kopyala

drawCircle() fonksiyonu kullanarak ekranda daire çizebilirsiniz. Fonksiyon dört parametre viz alır. Merkezin X koordinatı, merkezin Y koordinatı, yarıçap ve renk. Bu işlev, 1 piksel kenarlığı olan içi boş daire çizer. fillCircle() fonksiyonu kullanarak dolu daire çizebilirsiniz.

**Üçgen Çizimi**

****

*Nokia Ekran İçi Boş ve Dolu Üçgen*

//Draw Triangle

display.drawTriangle(20, 0, 0, 40, 40, 40, BLACK);

display.display();

delay(2000);

display.clearDisplay();

//Draw Filled Triangle

display.fillTriangle(20, 0, 0, 40, 40, 40, BLACK);

display.display();

delay(2000);

display.clearDisplay();

Kopyala

drawTriangle() fonksiyonunu kullanarak ekranda üçgen çizebilirsiniz. Fonksiyon yedi parametre viz alır. Üçgen ve renk köşelerinin 3 X ve Y koordinatı (x0, y0, x1, y1, x2 ve y2). (X0, y0) üst tepe noktasını, (x1, y1) sol tepe noktasını ve (x2, y2) sağ tepe noktasını temsil eder.

Bu işlev, 1 piksel kenarlığı olan içi boş üçgen çizer. fillTriangle() fonksiyonu kullanarak dolu üçgen çizebilirsiniz.

**Arduino Kodu – Bitmap’i Görüntüleme**

Bu son örnek, Nokia 5110 LCD Ekrana bitmap görüntülerin nasıl çizileceğini gösterir. Bu, şirket logolarının açılış ekranlarını oluşturmak, sprite yapmak veya sadece bilgileri görüntülemek için eğlenceli grafikler oluşturmak için kullanışlıdır. Aşağıdaki kodu kopyalayın, Arduino IDE’ye yapıştırın ve yükle’yi tıklayın.

#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);

// 'Marilyn Monroe 84x48', 84x48px

const unsigned char MarilynMonroe [] PROGMEM = {

 0x00, 0x00, 0x00, 0x7f, 0x00, 0x02, 0xfe, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xbe, 0x00,

 0x00, 0x1f, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x20, 0x00, 0x00, 0x3f, 0x80, 0x00, 0x00,

 0x00, 0x00, 0x00, 0x00, 0xf0, 0x00, 0x00, 0x1f, 0xe1, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0xc0,

 0x00, 0x00, 0x0f, 0xf1, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0e, 0xd8, 0xe0,

 0x00, 0x00, 0x00, 0x00, 0x00, 0x1f, 0x80, 0x00, 0x07, 0xe0, 0x70, 0x00, 0x00, 0x00, 0x00, 0x03,

 0x3f, 0xe0, 0x00, 0x07, 0xf0, 0x78, 0x00, 0x00, 0x00, 0x00, 0x01, 0xe0, 0x70, 0x00, 0x0f, 0xee,

 0x7c, 0x00, 0x00, 0x00, 0x00, 0x03, 0xc0, 0x00, 0x00, 0x0f, 0xf7, 0x1c, 0x00, 0x00, 0x00, 0x00,

 0x07, 0x80, 0x00, 0x0f, 0xc7, 0xf3, 0x1e, 0x00, 0x00, 0x00, 0x00, 0x07, 0xc0, 0x00, 0x0f, 0xf3,

 0xdf, 0x7f, 0x80, 0x00, 0x00, 0x00, 0x07, 0xfe, 0x00, 0x08, 0x7d, 0xef, 0xff, 0xc0, 0x00, 0x00,

 0x00, 0x7f, 0xff, 0x80, 0x30, 0x0f, 0xfc, 0xe0, 0xc0, 0x00, 0x00, 0x01, 0x9e, 0x73, 0xc0, 0xe0,

 0x07, 0xf8, 0xc1, 0xc0, 0x00, 0x00, 0x03, 0xfc, 0x00, 0x01, 0xc0, 0x0f, 0xfd, 0xe1, 0x80, 0x00,

 0x00, 0x03, 0xf8, 0x00, 0x01, 0x9c, 0x0f, 0xff, 0xc1, 0xc0, 0x00, 0x00, 0x02, 0xc0, 0x00, 0x01,

 0x9f, 0xbf, 0xfe, 0x01, 0x40, 0x00, 0x00, 0x02, 0x60, 0x00, 0x03, 0x07, 0xef, 0xff, 0x01, 0x40,

 0x00, 0x00, 0x00, 0x60, 0x00, 0x07, 0x01, 0xf7, 0xff, 0x80, 0xc0, 0x00, 0x00, 0x00, 0x50, 0x01,

 0xdf, 0x00, 0x7f, 0xff, 0x1c, 0x80, 0x00, 0x00, 0x00, 0x40, 0x01, 0xff, 0x00, 0x1f, 0xff, 0x1e,

 0xe0, 0x00, 0x00, 0x02, 0x08, 0x00, 0x3f, 0x80, 0x07, 0xef, 0x03, 0xe0, 0x00, 0x00, 0x06, 0x08,

 0x00, 0x03, 0xc0, 0x07, 0xdf, 0x07, 0xc0, 0x00, 0x00, 0x06, 0x08, 0x0f, 0x81, 0x80, 0x1f, 0xdf,

 0x1f, 0x80, 0x00, 0x00, 0x03, 0x08, 0x1f, 0x98, 0x00, 0x3f, 0xfe, 0x19, 0x80, 0x00, 0x00, 0x18,

 0x08, 0x3f, 0xfe, 0x00, 0x7f, 0xfe, 0x3f, 0x00, 0x00, 0x00, 0x08, 0x08, 0x30, 0x3f, 0x00, 0xff,

 0xff, 0x3f, 0x00, 0x00, 0x00, 0x01, 0xe0, 0x76, 0x0f, 0x89, 0xff, 0xff, 0x9f, 0x00, 0x00, 0x00,

 0x03, 0xe0, 0x7f, 0xc3, 0x81, 0xff, 0xfe, 0x9f, 0x80, 0x00, 0x00, 0x03, 0xf0, 0x7f, 0xf3, 0xc3,

 0xff, 0xfe, 0x1f, 0x00, 0x00, 0x00, 0x03, 0xf0, 0x7f, 0xfd, 0xc3, 0xff, 0xfe, 0x5e, 0x00, 0x00,

 0x00, 0x03, 0xf0, 0x7f, 0xff, 0xc3, 0xff, 0xf3, 0x1e, 0x00, 0x00, 0x00, 0x03, 0xf0, 0x71, 0xff,

 0x87, 0xff, 0xe3, 0xff, 0x00, 0x00, 0x00, 0x07, 0xf0, 0x7c, 0x3f, 0x87, 0xff, 0xe3, 0xfe, 0x00,

 0x00, 0x00, 0x0f, 0xf0, 0x3c, 0xff, 0x05, 0xff, 0xf3, 0xfc, 0x00, 0x00, 0x00, 0x0f, 0xf0, 0x0f,

 0xfe, 0x09, 0xff, 0xf7, 0xfc, 0x00, 0x00, 0x00, 0x08, 0xf8, 0x01, 0xfc, 0x19, 0xff, 0xff, 0xf8,

 0x00, 0x00, 0x00, 0x0c, 0x78, 0x00, 0x00, 0x13, 0xff, 0xff, 0xf8, 0x00, 0x00, 0x00, 0x0e, 0x78,

 0x00, 0x00, 0x23, 0xff, 0xff, 0xf0, 0x00, 0x00, 0x00, 0x0e, 0xf8, 0x00, 0x00, 0x47, 0xff, 0xff,

 0xf0, 0x00, 0x00, 0x00, 0x0c, 0xfa, 0x00, 0x01, 0x8f, 0xff, 0xff, 0xe0, 0x00, 0x00, 0x00, 0x08,

 0x7b, 0x00, 0x03, 0x3f, 0xff, 0xff, 0xe0, 0x00, 0x00, 0x00, 0x0c, 0x3b, 0xf8, 0x0f, 0xff, 0xff,

 0xff, 0xe0, 0x00, 0x00, 0x00, 0x0f, 0xbb, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x00, 0x00,

 0x07, 0xfb, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x71, 0xff, 0xff, 0xff,

 0xff, 0xff, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x41, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe0, 0x00, 0x00

};

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 bitmap

 display.drawBitmap(0, 0,  MarilynMonroe, 84, 48, BLACK);

 display.display();

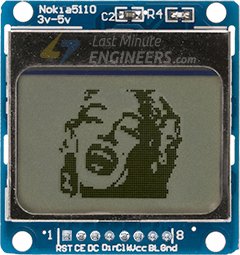
 // Invert Display

 //display.invertDisplay(1);

}

void loop() {}

Böyle görünür.

****

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

Nokia 5110 LCD ekranında bitmap görüntüsünü göstermek için drawBitmap() işlevi çağırmamız gerekir. Altı parametre viz alır. Sol üst köşe X koordinatı, sol üst köşe Y koordinatı, tek renkli bitmapin bayt dizisi, piksel cinsinden bitmap genişliği, piksel cinsinden bitmap yüksekliği ve Renk.

Örneğimizde, bitmap görüntüsü 84×48 boyutundadır. Böylece, genişlik ve yükseklik 84 ve 48 olarak ayarlanırken X ve Y koordinatları 0 olarak ayarlanır.

// Display bitmap

display.drawBitmap(0, 0,  MarilynMonroe, 84, 48, BLACK);

display.display();

Kopyala

Ancak, drawBitmap() işlevi çağırmadan önce, çizmek için bir görüntüye ihtiyacımız var. Unutmayın, Nokia 5110 LCD ekranın ekran çözünürlüğü 84 × 48 pikseldir, bu nedenle bundan daha büyük görüntüler düzgün görüntülenmez. Doğru boyutta bir görüntü elde etmek için tuval boyutunu 84 × 48 piksel olarak ayarlayarak Inkscape, Photoshop, Paint, vb. Gibi favori çizim programlarını kullanabilirsiniz.

Gülen Marilyn Monroe görüntüsünü örnek olarak aldık ve Paint kullanarak 84 × 48 piksele dönüştürdük ve .bmp olarak kaydettik.

Bir bitmap’e sahip olduğunuzda, PCD8544 denetleyicisinin anlayabileceği bir diziye dönüştürme zamanı gelmiştir. Bu iki yolla yapılabilir: image2cpp kullanan çevrimiçi yöntem ve LCD Assistant kullanan Çevrimdışı yöntem.

**Çevrimiçi Bitmap Dizisi Oluşturucu – image2cpp**

Resminizi bir diziye dönüştürebilen image2cpp – <http://javl.github.io/image2cpp/> adlı çevrimiçi bir uygulama var. Image2cpp, LCD Asistan’dan daha yeni ve çok daha güçlüdür.

**Şunları Sağlar:**

* Aynı anda birden fazla görüntüyü dönüştürme.
* Görüntü dosyanızı ölçeklendirme – Sığdırmak için Sığdır / Ölçeklendir / Orijinal
* Parlaklık eşiğini siyah ve beyaz arasında ayarlama.
* Görüntüyü dikey ve / veya yatay olarak yeniden ortalama.
* Ters görüntü renkleri

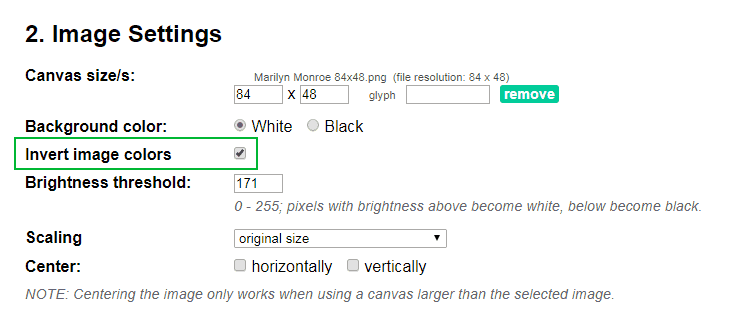
Başlamak için tarayıcınızda [image2cpp’yi](http://javl.github.io/image2cpp/) açın ve ekranda görüntülemek istediğiniz herhangi bir resmi seçin.

****

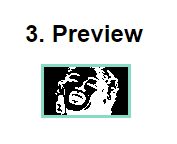
Görüntünüzün boyutları, Görüntü ayarları altındaki Tuval boyutu seçeneğinde bulunur. 84 × 48’den daha büyük bir görüntü seçtiyseniz, görüntüyü 84 × 48 olarak değiştirin ve uygun Ölçeklendirme seçeneğini belirleyin. Çıktıyı Önizleme bölümünde görebilirsiniz.

Gerekirse Arka plan rengini veya Tersine görüntü renklerini değiştirebilirsiniz.

Son olarak, en önemli seçeneği değiştirin (Parlaklık Eşiği). Eşiğin ayarlanması bu seviyenin üstündeki pikselleri beyaz ve siyahın altındaki pikselleri yapar. Biz güzel detayları almak için 171’e ayarladık.

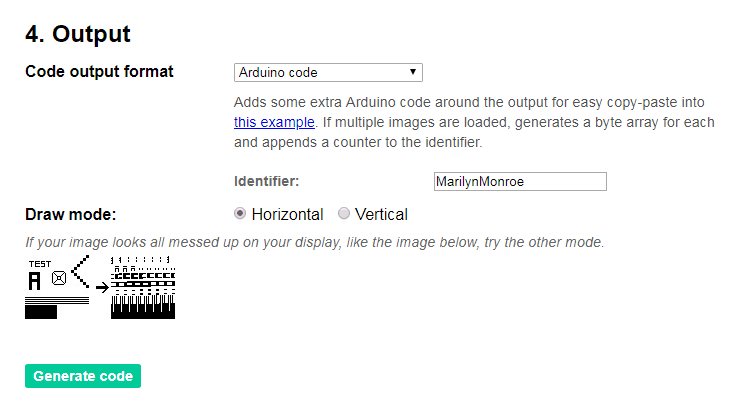
****

Bu küçük önizleme, ayarlarınızda yaptığınız değişiklikleri yansıtır. Ayarları göz önünde tutarak değiştirebilirsiniz.

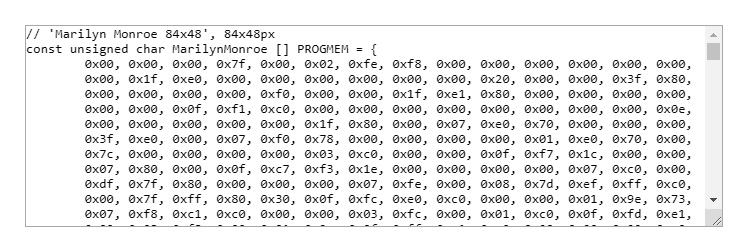
****

Sonuçtan memnun kaldığınızda, veri dizisini oluşturmaya devam edebilirsiniz. Arduino Kodu olarak Kod çıkış formatını seçin ve Kod oluştur düğmesine tıklayın.

Yalnızca bilgileriniz için Çizim modu adı verilen bir seçenek vardır. Aslında ekranın tarama desenine göre görüntü oluşturur. Resminiz ekranınızda karışık görünüyorsa modu değiştirmeyi deneyin.

****

Bu kadar. Bitmap’inizin bayt dizisi oluşturulur. Çıktıyı doğrudan örnek kodumuzla kullanabilirsiniz. Sadece uygun şekilde adlandırdığınızdan emin olun. Ardından diziyi drawBitmap() işlevi içinde çağırın.

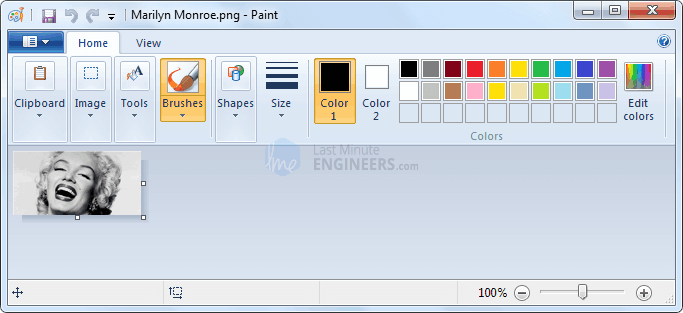
****

**Çevrimdışı Bitmap Dizisi Oluşturucu – Paint**

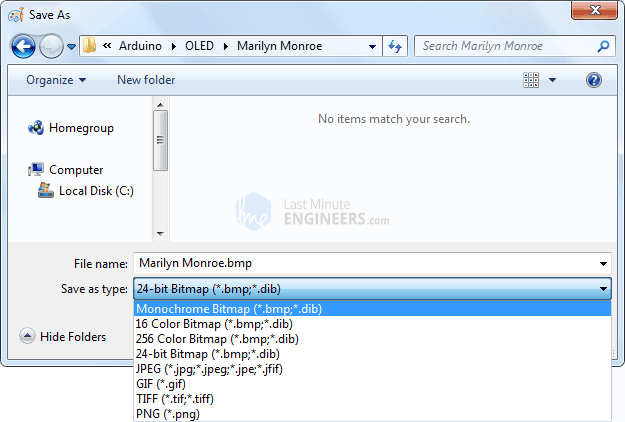
Bit eşlem görüntünüzü veri dizisine dönüştürebilen paint benzeri [BitMap Converter](http://en.radzio.dxp.pl/bitmap_converter/" \t "_blank) kullanabilirsiniz. İmage2cpp kadar güçlü değil ama yine de popülerdir.

Başlamak için, görüntüyü 84 × 48 1 bit tek renkli bitmap’e dönüştürmeniz gerekir.

Dosyanızı açın ve 84 × 48 olarak yeniden boyutlandırın.

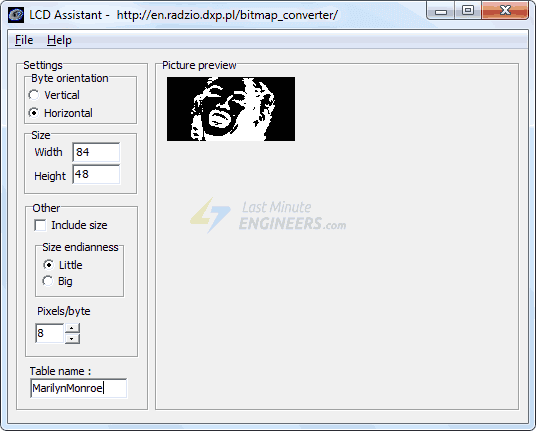
****

Şimdi dosyanızı bitmap olarak kaydedin. Dosyayı kaydederken Farklı kaydet tipi’ni seçin: Monochrome Bitmap (\*. Bmp; \*. Dib). Bu, her piksel için yalnızca iki olası değere sahip 1 bit / ikili bitmap görüntü oluşturur, yani 0 (siyah) veya 1 (beyaz).

****

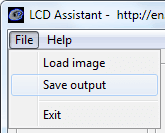
Buradaki tek dezavantaj, parlaklık eşik seviyesini ayarlayamamanızdır. Varsayılan olarak % 50 olarak ayarlanmıştır ve değiştirilemez.

LCD asistan programını indirin. Yürütülebilir dosyayı açın ve Dosya menüsünden bitmap’inizi yükleyin.

****

Bu araçla yapabileceğiniz çok şey yok. Bu nedenle, Dosya menüsüne gidin ve Çıktıyı kaydet seçeneğini tıklayın. Dosyayı metin dosyası olarak kaydedin.

Sadece bilgi için Bayt Yönlendirme adlı bir seçenek var. Aslında ekranın tarama desenine göre görüntü oluşturur. Resminiz ekranınızda karışık görünüyorsa modu değiştirmeyi deneyin.

****

Dizin oluşturulduğunda, kodunuza yapıştırın. Sadece uygun şekilde adlandırdığınızdan emin olun. Ardından diziyi drawBitmap() işlevi içinde çağırın.