**4/11/2014 Nokia 5110 LCD Display - Direct Connect to Uno**

1. Project Reference Number / Title:

-20140411 **Nokia 5110 LCD Display - Direct Connect to Uno**

2. Link to Hardware Order:

-

3. Other Documentation:

-

4. Link to Source Documentation/ Actual Documentation:

-[**Nokia 5110 LCD Display - Direct Connect to Uno**](http://electronicavm.wordpress.com/2012/02/13/nokia-lcd-3310-arduino/)

5. Wiring Documentation:

-

6. ---------------- Project Code: (text format) START -------------------

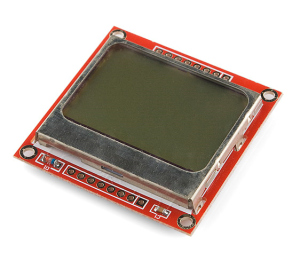
7. --------------- Project Code: (text format) END -----------------------

[13/02/2012](http://electronicavm.wordpress.com/2012/02/13/nokia-lcd-3310-arduino/)OF [IPADNANO](http://electronicavm.wordpress.com/author/ipadnano/)

# **Nokia 3310 LCD + Arduino**

In my view, there is no element that gives a better look to our projects and interaction DIY screen. We can choose between many different screens: LCD Character LCD color graphics, monochrome, TFT, TFT Touch, etc.

Today we will see how to connect and use with Arduino, **a graphical LCD of 84 × 48 pixels monochrome, very cheap and very easy to use, the LCD Nokia 3310/5110 that used the old phones from the Finnish company.**

****

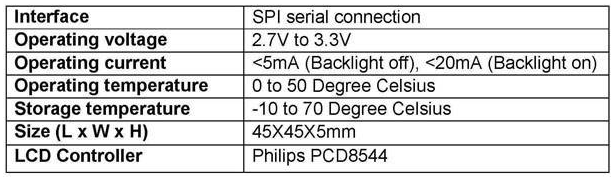
we get these small screens or Well removing a Nokia 3310/5110 phone or purchase online at many electronics stores.

In my case I bought on eBay, but we buy it for a few euros in almost all stores list this previous post: [Where to buy electronics?](http://electronicavm.wordpress.com/2012/01/23/donde-comprar-electronica/)

This small screen works with the **CMOS controller** [**PCD8544**](http://www.sparkfun.com/datasheets/LCD/Monochrome/Nokia5110.pdf) designed for handling graphics LCD 48 rows x 84 columns. **'s very low power and how to connect with μControladores is done via the SPI bus.**

Works with only **3.3 V** and has LED backlighting.

In one of the online stores where they sell the screen, I found this table very concise and brief specifications that explains the most important features of this screen:

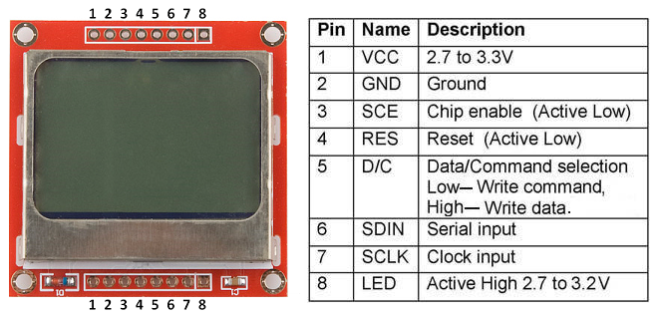


As you can see, it is really very low consumption, which makes it ideal for projects batteries, battery powered display.

With this brief introduction, let's move to action: **How do you connect?**

**Read on!**

Before connecting the screen, we will have to see what their pins:

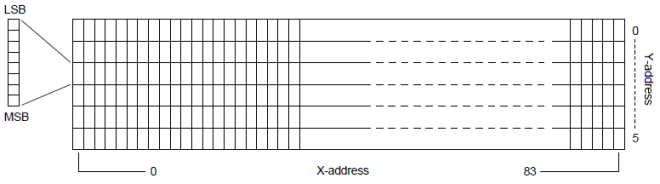


The display communicates via the SPI bus, an Arduino UNO consists of pins 10, 11, 12 and 13 if we used the original library of Arduino, but in this case, use an external library to Arduino, so the selection of pins is connected to our choice as we can configure through software.

Let's see how it works a little screen, in order to understand the functions of the library we use to control the screen.

We know that the screen is 84 pixels wide by 48 pixels high, or what is the same in the time to make it work, 84 bits wide by 48 bits high. The bits of the high are grouped into bytes, so that **the addressing of the display is performed as follows: 84 bits wide by 6 bytes high** , therefore, will in the following way: **X-axis = 0 through 83 / . axis = 0-5**

As you can see in the picture below:



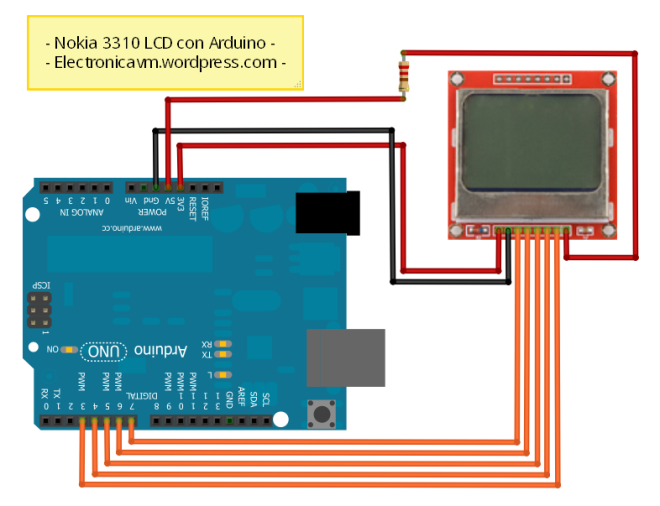
This thus seen not help us much if our idea is to simply add the screen to your project and use it, but you need to know, since **the characters when printing a text screen, each occupy a space of 5 bits wide by 8 high** , so only have a maximum of 6 lines to write to the LCD.

Knowing how it works the screen, in my case I chose the first sketch of this [page Arduino](http://www.arduino.cc/playground/Code/PCD8544) and became Seller to use the screen and have a clean and easy to read and understand sketches.

**We can download the library from here: [Bookseller NokiaLCD](http://electronicavm.comuv.com/Nokia.rar)**

With this library we can handle the LCD comfortably using the various functions available, such as **configuring the connector pins, initialize the LCD, clear the LCD, position the cursor, enter characters or strings,**  **read the serial port for charging an image bitmap, draw on the screen from an array of bytes, etc..**

Then, proceed to connect the LCD and test the library, in my case, I use pins 3, 4, 5, 6 and 7 for connecting digital screen, as follows:

Load the example that comes with the library and analyze a bit:

| 1  2  3  4  5  6  7  8  9  10  11  12  13 | # Include <NokiaLCD.h>    NokiaLCD NokiaLCD (3,4,5,6,7); / / (SCK, MOSI, DC, RST, CS)    void setup () {  NokiaLCD.init (); / / Init screen.  NokiaLCD.clear (); / / Clear screen.  }    void loop () {  NokiaLCD.setCursor (1,1);  NokiaLCD.print ("Hello World!");  } |
| --- | --- |

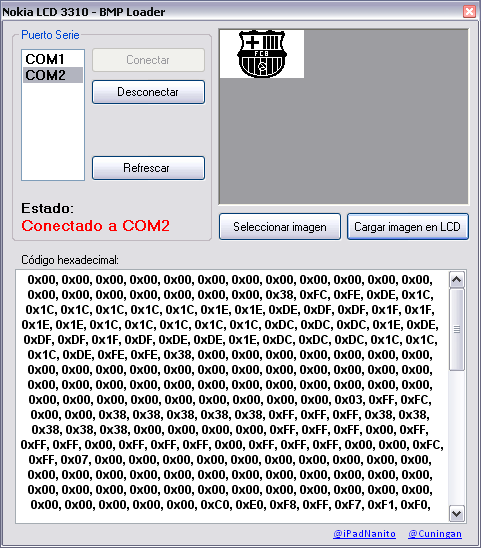
Instantiate and assign an LCD connector pins, initialize the LCD and Full erase, write and place the cursor on the screen. Management emphasizes simplicity :)

The library contains the following functions:

**- NokiaLCD NokiaLCD (SCK, MOSI, DC, RST, CS),** / / Instance LCD and assign the pins. **- NokiaLCD.init ()** / / Initialize the LCD. **- NokiaLCD.clear ()** / / Clears the screen and places the cursor at the position (0,0). **- NokiaLCD.setCursor (x, y)** / / Move the cursor to the position indicated. **- NokiaLCD.print ("string")** / / Write a string on screen character. **- NokiaLCD.character ('A')** / / Write a character on screen. **- NokiaLCD.bitmap (array)** / / Draw on the screen an array of bytes of a BMP image. **- NokiaLCD.sBitmap ();** / / Listen to the serial port of an array of bytes of a BMP image and draws it on the screen.

Most interesting of the screen, is that being a graphic display, **we can draw on it with NokiaLCD.bitmap function (array)** that receives an array of bytes and draws it on the screen, so it was necessary to get this array of bytes from a bitmap.

This, along with **Dani (** [**@ Cuningan**](http://www.twitter.com/Cuningan) **-** [**Simplify 2.0**](http://cuningan.wordpress.com/) **)** , we developed a software in C # to load BMP images on the screen from the PC, either by sending them through the serial port or convert array of bytes and store that array in memory the μControlador or EEPROM.



As we can see in the above image, the software is simple to use, simply **choose the serial port** that is connected to our Arduino, connect, **and proceed to choose an image file** .

Once the image is loaded into the program, **by clicking on the "Load image in LCD" on the serial port send**  **the byte array of the image, and** at the same time, **we will see in the box, and copy it to paste it into the code of our Arduino sketch.**

If we are going to load an image from the PC via the serial port, our Arduino sketch, you should call the NokiaLCD.sBitmap () function; in the loop "void loop ()", so you're constantly getting through the serial port, waiting to get the image from PC.

The software is installable, but only for Windows.

**Here the download link: [Nokia BMP Loader Software](http://electronicavm.comuv.com/Nokia%20BMP%20Loader.rar)**

And that's it!

It really is a screen to consider for small projects, inexpensive, simple to use, and low power consumption.

**Let's see some pictures and a video performance! :)**

**** 