# ARDUINO AND NOKIA 3310 LCD, 84x84 pixels

Features

The Nokia 3310 LCD is a nice small graphical LCD, suitable for a lot of various projects.
The display is 38*35 mm, with an active display surface of 30*22 mm, and a 84*48 pixel resolution.
The display is easy to interface, using standard SPI communication. A 1-10 uF electrolytic capacitor from VOUT to GND, is the only external component needed.
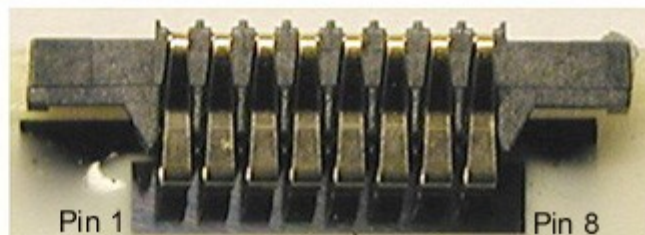
- Logic supply voltage range VDD to VSS : 2.7 to 3.3 V
- Low power consumption, suitable for battery operated systems
- Temperature range: -25 to +70 °C

Mechanical specification
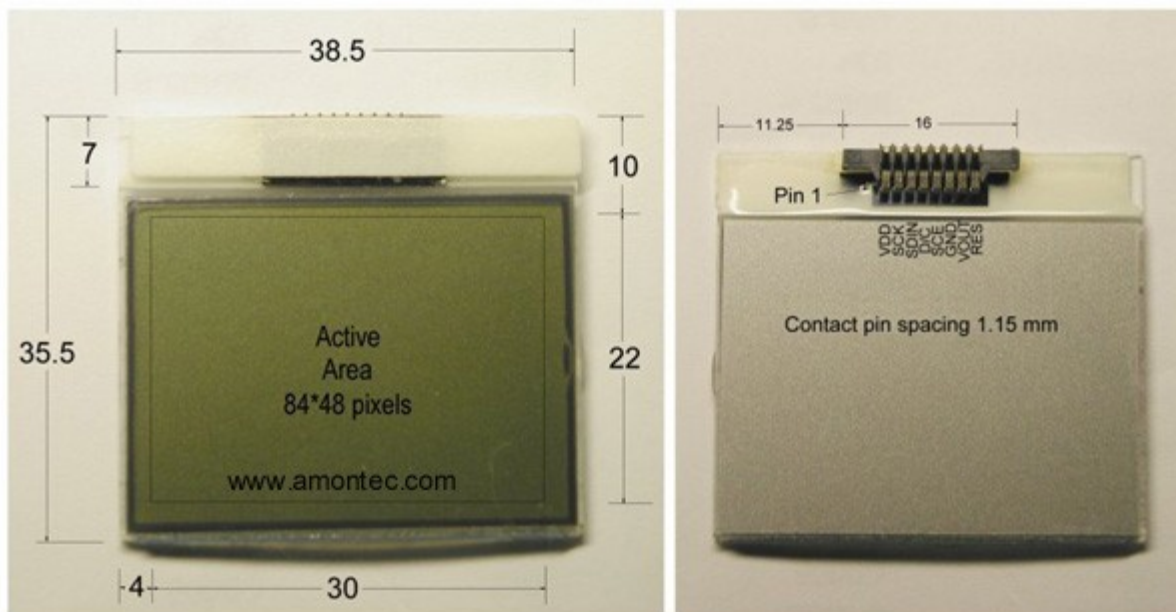
Electrical Interface specification

Pin	Signal	Description	Port
1	VDD	Power Input. Logic supply voltage range VDD to GND : 2.7 to 3.3 V	Power
2	SCLK	Serial clock. Input for the clock signal: 0.0 to 4.0 Mbits/s.	Input
3	SDIN	Serial data. Input for the data line.	Input
4	D/C	Mode Select. To select either command/address or data input.	Input
5	SCE	Chip enable input. The enable pin allows data to be clocked in. The signal is active LOW.	Input
6	GND	Ground	Power
7	VOUT	Ouptut voltage. **Add external 1-10 uF electrolytic capacitor from VOUT to GND**	Power
8	RES	External reset. This signal will reset the device and must be applied to properly initialize the chip. The signal is active LOW.	Input

If you got your lcd from an old phone you might be better off with the following tutorial: http://code.google.com/p/pcd8544/ . That is the code base I used at first and it has some information on how to link everything. It also links back to http://www.sparkfun.com/tutorials/65 with all the information you need to interface a 5v microcontroller with a 3v device.

From that page I read:

On arduino output pins:

10K resistors will decrease the amount of current flowing into the 3.3V device. The internal clamping diodes within the 3.3V device will attempt to clamp the incoming signal to 3.3V thus protecting the rest of the 3.3V device. Clamping diodes are normally found on the input lines of low voltage devices.

Note that according to the code in the library all pins are configured as output pins on the arduino.

On arduino input pins:

You should not need any resistors since the arduino will take a signal starting from 1V as a high value so 3.3V will work just fine.

On CS pin:

A 1K resistor on the CS line will form a voltage divider circuit with the 10K resistor.

Pictures

Some pictures of everything connected. You can clearly see the 10k resistors and the 10mF capacitor I used. Soldering to the tiny connector is really nerve wrecking. I used a piece of IDE ribbon cable like you can find in older computers. On the other end I attached some headers with normal breadboard spacing so I could use it on the mini breadboard on the picture.
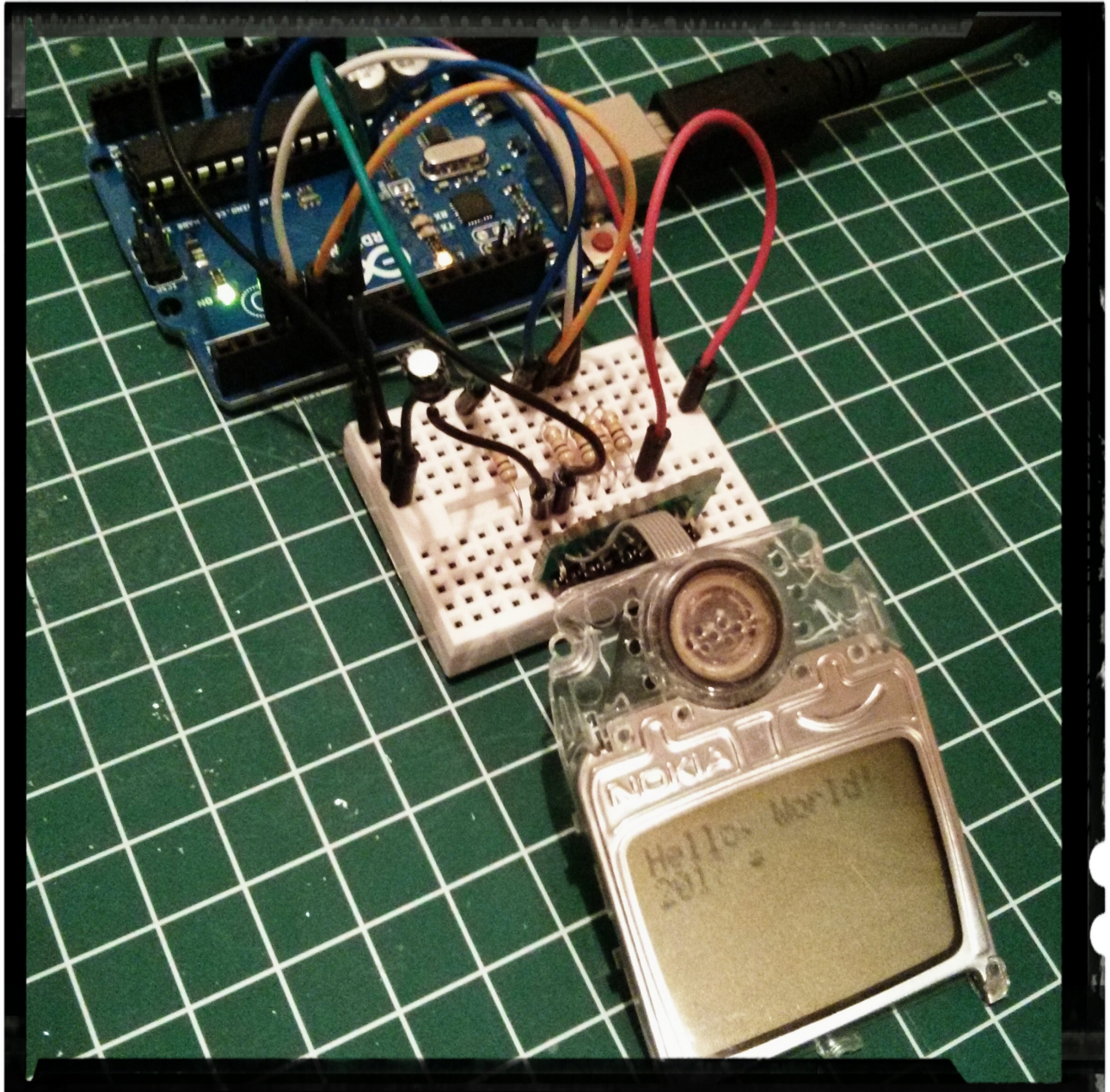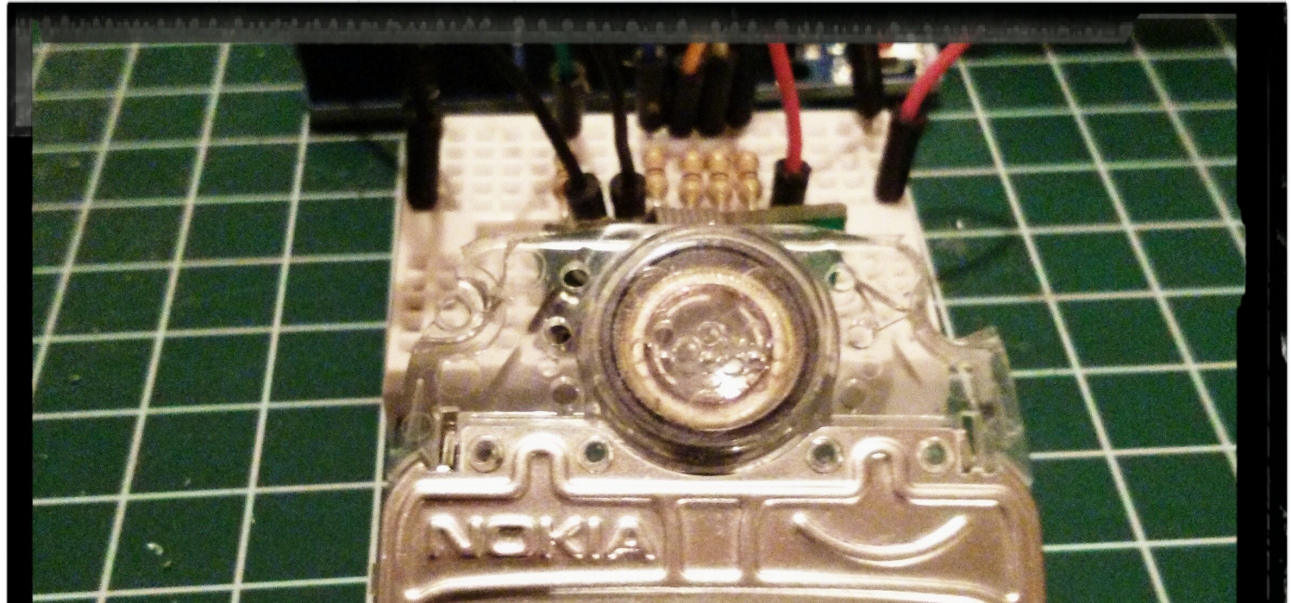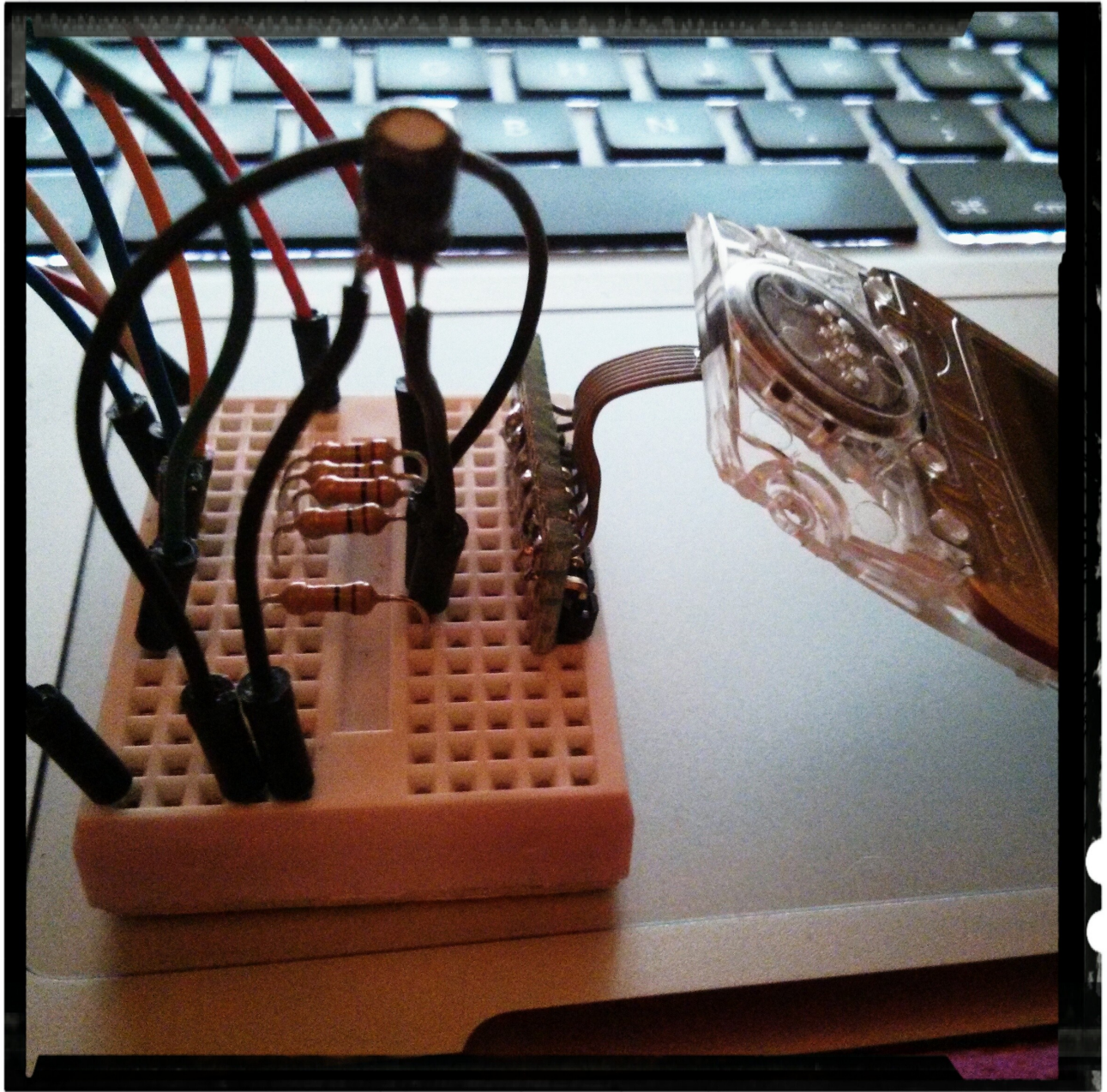
Arduino Libraries

Now for the Arduino code you also have the same 2 options (or even more). If you only need to print text and some glyphons to the screen you'll be fine with the smaller library from the non adafruit page. If you want graphics you'll need the adafruit library.

PCD8544 library on Google code

By far the smalles and easiest library. Just get it from http://code.google.com/p/pcd8544/ , check the connections to match the project page, add to libraries folder, restart Arduino IDE and load the sample project. The contrast worked out fine for me.

The only downside to this library is that you can only print characters. Either text or the glyphons.

I connected the following pins:

```
VDD => 3.3V
SCK => to pin 3 with a 10kO resistor
DIN => to pin 4 with a 10kO resistor
D/C => to pin 5 with a 10kO resistor
SCE => to pin 7 with a 10kO resistor
GND => GROUND
VOUT => with a 10mF capacitor to GROUND
RES => to pin 6 with a 10kO resistor
```

Adafruit PCD8544 library

For graphical applications you'll need this library from the adafruit tutorial at http://learn.adafruit.com/nokia-5110-3310-monochrome-lcd/overview . Same idea, only now you'll have to double check to match the pinout if you used the connections from the other project. They are kind of reversed. If you used all the steps from the adafruit tutorial you can skip this step. Otherwise, with the connections from the second tutorial modify the following line in the example code. I also changed the comments to show what pin is what.

```
// pin 3 - Serial clock out (SCLK) = SCK
// pin 4 - Serial data out (DIN) = SDIN
// pin 5 - Data/Command select (D/C)
// pin 7 - LCD chip select (CS) = SCE
// pin 6 - LCD reset (RST) = RES
Adafruit_PCD8544 display = Adafruit_PCD8544(3, 4, 5, 7, 6);
```

I wouldn't try with pins swapped around so double check. Once I ran this example project I noticed I had to turn up the contrast. Can be done at the following line. 60 instead of 50 was fine for me. The contrast in the smaller library was fine by default.

```
  // you can change the contrast around to adapt the display
  // for the best viewing!
  display.setContrast(60);
```

More information about the graphics library is available at http://learn.adafruit.com/adafruit-gfx-graphics-library .