Graphic LCD 84x48 - Nokia 5110

**DESCRIPTION**

- 45x45mm

**TUTORIALS**

**Graphical LCD Hookup Guide**

NOVEMBER 18, 2013

How to add some flashy graphics to your project with a 84x48 monochrome graphic LCD.

**REVIEWS**

4.3 out of 5

14 reviews

**COMMENTS**

213 comments
Based on 14 ratings:

- 5 star: 6
- 4 star: 7
- 3 star: 0
- 2 star: 1
- 1 star: 0

Currently viewing all customer reviews.

1 of 1 found this helpful:

**Easiest to use small graphical lcd**

about 3 years ago by Chiel

It is a small graphical lcd with backlight on a sturdy pcb with probably one of the easiest to use controller ever at a ludicrously cheap price! What else could you want. If you are new to the joy of electronics and want to try coding for graphical screens this is the best starting point i can think of!

1 of 1 found this helpful:

**Be Careful at 5V**

about 3 years ago by Member #670968

We ran it at 5V and I think it might have blown our LCD. We were having trouble getting it to display anything, and when it finally did, it was very dim. Then eventually it displayed nothing at all.

Tried another LCD (same model), running at 3.3 this time, and it worked great!

So, be careful running it at 5V.

Also, listen to the other members who recommended changing the contrast setting in initialization. You might need to play around with it, but you will get there eventually.

3 of 3 found this helpful:

**Very good display, I've used several**

about 3 years ago by RobotCamera

There are a few things you need be aware of here.

Dirty connections: I have found that one of my units would randomly stop working or flicker, or alternate between being too dim or being too dark, and this was clearly a physical problem (pressing on the board made the problem go away). The problem, as I discovered, was dirty contacts on the board and screen itself. I unclipped the board and found dust particles on the contacts. Cleaned both sets of contacts with cotton swabs and rubbing alcohol, and the problem was resolved.

Contrast setting: I think most of the difficulties people are having here stem from trying random hardcoded contrast settings. The proper way to do this is to allow for adjustable contrast settings by the end user. The acceptable range is between 0x80 (being a contrast value of 0) and 0xFF (being a contrast value of 127). The ideal contrast setting will change with ambient temperature. This is why I suggest making your contrast setting changeable on-the-fly, through your microcontroller software. Whether by a trimmer, or buttons, or whatever.

For typical room temperature, a good value is likely to be around 0xBF. That’s where mine is at. If it’s too dark, decrease by increments of 1 until you find the ideal value, if too light increase by increments of 1. I have buttons on my device that do this and I save the user-selected contrast setting to EEPROM. The code for changing the contrast on the fly is easy. Set the SCE pin to low to enable the serial interface. Set the D/C pin to low (which tells the LCD you are sending commands, not pixel data), then send byte 0x21 which enables the extended instruction set, then send the contrast byte, then send byte 0x20 which returns to the basic instruction set. You do not need to reset the panel or anything like that.

Hardware SPI: use it. It makes the display so much more responsive. For this, connect the MOSI line to the microcontroller’s MOSI pin, and the SCK line to the microcontroller’s SCK pin. Then, avoid using the MISO and SS lines of the microcontroller unless you know what you’re doing. On a ATmega328 chip, as found in Arduino UNO, Pro, and Pro mini, the MOSI and SCK lines are digital pins 11 and 13, respectively. Then you want your library to communicate with the LCD using SPI. Easy to do. The LCD’s speed is 4MHz, data is sent most significant bit first (MSBFIRST), the data clock is idle when low (Clock Polarity/CPOL = 0), and samples are taken on the rising/positive edge of clock pulses (Clock Phase/CPHA = 0). So for Arduino SPI, that means using SPI_MODE0. It’s all downhill from there and you can chain multiple devices to SPI. It’s wonderful.

I have a working library for this LCD that uses hardware SPI that I’m pretty happy with, and plan to release under MIT license on github sooner or later. If there is demand I can rush that process.

Definitely run this LCD on 3.3V. Don’t even think of 5V. Not even a little bit. You should be able to put a voltage dropping resistor on the Vin pin, though, if you don’t have a 3.3V regulated output.

Downloaded from Arrow.com.
I can get the backlight to work but nothing appears on the display itself. I don't know if I have the right library installed. I have checked my wiring several times and it appears to be wired correctly. Don’t know what could be wrong, any suggestions would be helpful, thanks in advance. Graig

Be sure to check out the hook up guide – https://learn.sparkfun.com/tutorials/graphic-lcd-hookup-guide

The weather here has been so good that I’ve spent time working in the garden rather than working with electronic.

It is fairly simple to do either graphics or text on by simply using/modifying the example code. There is example code on-line to do a larger 7-segment display if you need bigger numbers. All-in-all very handy to apply to a project.

I unfortunately was not as lucky as the other reviewers. The first screen I received was usable but defective (bad row). The great people at SparkFun send me a replacement no questions asked.

Unfortunately the second screen would not even turn on; The SparkFun Squad wanted to help me debug the issue, but I decided against it since it was pretty obvious that the screen was dead and I didn’t want to spend time fiddling around with it. The one with the bad row will do just fine for now.

Using it with hardware SPI and 3.3V. Had to play with some of the settings to get a good image. Can think of lots of cool projects to do with this little display.

Works as expected!

It’s SPI (2 wire no SDO from LCD). Works well with Microchip PIC16F1454.

This does what it says it does for a cheaper price than many other lcd’s. It is also very simple to use, so you can get started right away.
In 2003, CU student Nate Seidle blew a power supply in his dorm room and, in lieu of a way to order easy replacements, decided to start his own company. Since then, SparkFun has been committed to sustainably helping our world achieve electronics literacy from our headquarters in Boulder, Colorado.

No matter your vision, SparkFun's products and resources are designed to make the world of electronics more accessible. In addition to over 2,000 open source components and widgets, SparkFun offers curriculum, training and online tutorials designed to help demystify the wonderful world of embedded electronics. We’re here to help you start something.