avdweb (/)

Home (/) **Solar bike** **Arduino (http://www.avdweb.nl/#)** **Photos**

Tech tips **Composites (http://www.avdweb.nl/#)** **Diversen**

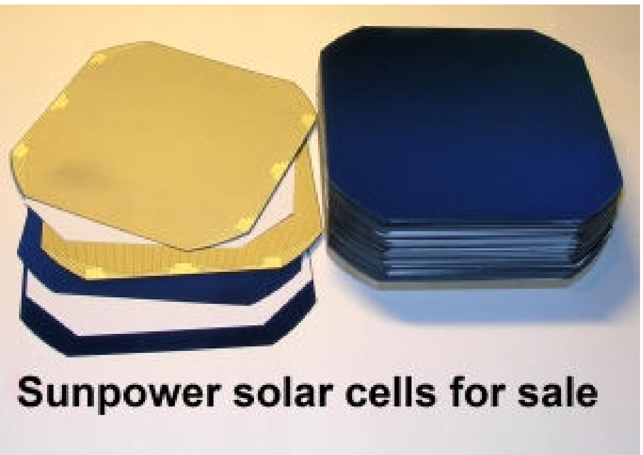
Computer **Programming** **Life (http://www.avdweb.nl/#)** **GPS**

Columns **Site map (/xmap.html)**

Search...

Most recent articles

- PV panel mismatch loss versus solar cell tolerance (/solar-bike/pv-panels/pv-mismatch-loss.html)
- Foto’s Zuid Limburg en omgeving (/holiday-photos/zuid-limburg.html)
- Carnaval foto's (/holiday-photos/carnaval.html)
- Samenvatting c++ (/programming/opfriscursus-programmeertaal-cpp-nl.html)
- Hersenen versus computer (/life/hersenen-vs-computer.html)
- Sunpower single monocrystalline solar cells for sale (/solar-bike/pv-panels/offer-solar-cells.html)



Nokia 5110 LCD

How to use the Nokia 5110 LCD Module at Arduino (/arduino/hardware-interfacing/nokia-5110-lcd.html)

Contents[Hide]

1. Hardware

The Nokia 5110 LCD Module uses a Philips PCD8544 LCD driver, which is designed for mobile phones.

(http://www.avdweb.nl/Article_files/Arduino/Nokia-5110/Nokia-5110-LCD.jpg)

Nokia 5110 LCD Module 84X48 with backlight - Nokia 5110 LCD Module 84X48 with backlight

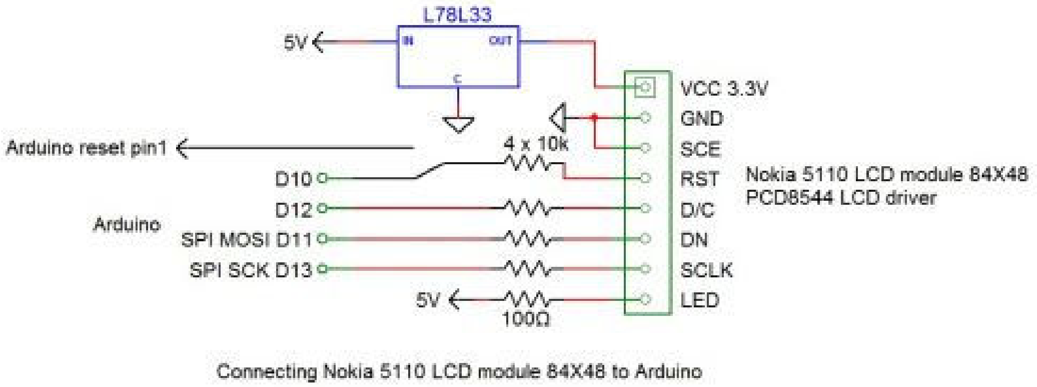
1.1. Operation at 3.3V

Many devices that can be used with an Arduino, require a power supply of 3.3V. This is also the case with the Nokia 5110. The best way to deal with 3.3V devices is to take an Arduino Pro, which can run on 3.3V.

Thanks to the internal clamp of the PCD8544 we can use a very simple level shifter. Four current limiting resistors of 10kΩ can do the job. When an LCD control line is high, the current through the 10kΩ resistor is just 40uA, so this is harmless. Note that we can't read back from the LCD with this circuit.

1.2. Operation at 5V

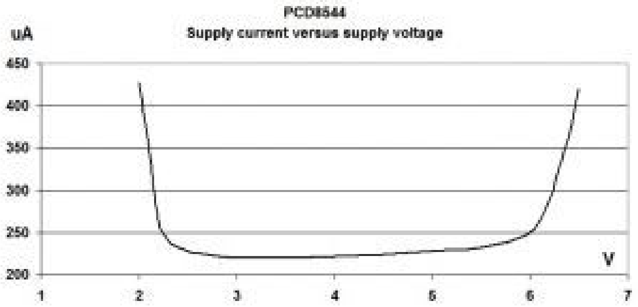
Because VDD max = 7V, the PCD8544 controller can handle 5V, but the Nokia 5110 LCD works best at 3.3V. The four resistors of 10kΩ avoid streaks on the LCD display.



(http://www.avdweb.nl/Article_files/Arduino/Nokia-5110/Nokia-5110-LCD-Arduino-3-wire-control.jpg)
Connecting Nokia 5110 LCD to Arduino with 3 wires - Connecting Nokia 5110 LCD to Arduino with 3 wires

Note that the Arduino SPI pins 11 and 13 are used to allow hardware SPI.

1.3. PCD8544 supply current versus supply voltage



(http://www.avdweb.nl/Article_files/Arduino/Nokia-5110/PCD8544-supply-current-versus-supply-voltage.jpg)
PCD8544 supply current versus supply voltage - PCD8544 supply current versus supply voltage

1.4. Using the control lines

The Nokia 5110 LCD has 5 control lines; the interface is of the type SPI. Mostly, when no other SPI devices are used, the chip select can be connected to the GND, so 4 control lines stay over. To save Arduino pins, there is an option to use only 3 control lines, while the Nokia reset pin is connected to the Arduino reset. This works only when the serial monitor is used, not with a standalone Arduino.

1.5. Auto reset circuit

An option is to generate a reset pulse on power up automatically, with a RC combination, to save one control line to the Arduino. However, I found out that this solution is poorly reproducible; the proper functioning depends for instance on the capacitor across the 3.3V supply. Also the Arduino can't be used with the serial monitor anymore.

1.6. Nokia 5110 LCD with different connector wiring

Not all Nokia 5110 LCDs are the same. Although they look identical, the connector wiring can be completely different.

2. Software

2.1. Improved PCD8544 library

Limor Fried from Adafruit has created a library which can be download at github (https://github.com/adafruit/Adafruit-PCD8544-Nokia-5110-LCD-library). I have changed the library a little bit, you can download it here (/Article_files/Arduino/Nokia-5110/PCD8544.zip).

Changes:

- Solved problems with bad LCD contacts.
- Added clear() and init(), for compatibility with older library versions.

Unzip the library and place it into the Arduino library folder like this: \libraries\Adafruit_PCD8544. For troubleshooting see HERE (/arduino/troubleshooting.html).
Also install the Adafruit_GFX (https://github.com/adafruit/Adafruit-GFX-Library) library in the same way.

2.2. Solving problems with bad LCD contacts

The LCD is connected to the PCB by means of a conductive rubber strip; this sometimes causes contact problems and a black screen. Restoring the connection doesn't help; the LCD must be re-initialized by manually restarting the Arduino. This is inconvenient. The LCD application can be made more robust by initializing the LCD repeatedly with an interval of about 2s. This allows disconnecting and reconnecting the LCD during operation. I have changed the Adafruit-PCD8544-Nokia-5110-LCD-library to add this functionality.

2.3. LCD printing with streaming operator <<

With the Streaming library, we can use the streaming operator << for printing to the console:

`Serial << "hello!\n";` instead of **`Serial.println("hello!\n");`**

Because the PCD8544 parent class is Print we can also use the streaming operator **<<** for printing to the LCD. Here is an example:

```
#include <Adafruit_GFX.h>
#include <Adafruit_PCD8544.h>
#include <Streaming.h>
#include <Metro.h>

const byte NokiaSCLKpin = 13; // compatible with hardware ISP
const byte NokiaDNpin = 11; // compatible with hardware ISP
const byte NokiaDCpin = 12;
const byte NokiaRESETpin = 10;

const char LCDcontrast = 57;
int i;

Adafruit_PCD8544 nokia(NokiaSCLKpin, NokiaDNpin, NokiaDCpin, 0, NokiaRESETpin);
Metro LCDinitTimer(2000);

void setup(void)
{ Serial.begin(9600);
  nokia.init(LCDcontrast);
  nokia.clear();
  nokia << "Hello!\n"; // without using F
  nokia << F("Save RAM with the Flash library\n"); // use F to save RAM space
  nokia.display();
  findBestContrast(40, 60, 500);
  nokia.setContrast(LCDcontrast);
}

void loop(void)
{ nokia.clearDisplay();
  nokia << F("Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor
") << i++;
  nokia.display();
  if(LCDinitTimer.check()) nokia.init(); // allow disconnecting and connecting the LCD during
execution
}

void findBestContrast(int contrastMin, int contrastMax, int _delay)
{ for(int contrast=contrastMin; contrast<=contrastMax; contrast++)
```

```
  { nokia.clearDisplay();
    nokia.setContrast(contrast);
    nokia << F("12345678901234");
    nokia << F("12345678901234\n"); // Bug: "\n" after 14 characters creates a blank line
    nokia << F("bla bla bla bla bla bla\n");
    nokia << contrast;
    nokia.display();
    delay(_delay);
  }
}
```

2.4. Adjusting the contrast value

The contrast must be set separately for each LCD. With the function **`findBestContrast()`**, the correct contrast value can be determined.

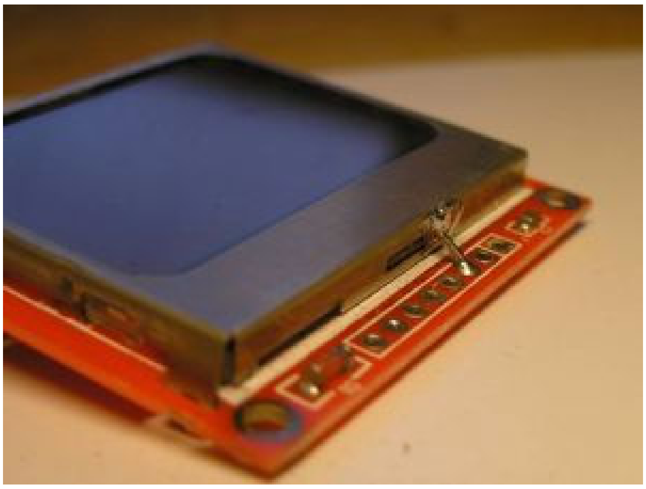
2.5. PCD8544 library with hardware SPI

The library Adafruit_PCD8544 uses communication with the LCD by **software SPI**; therefore the library is quite slow. In my example, writing the full LCD screen takes 31ms.

Snigelen has created a hardware SPI PCD8544 library, see here:
<a " href="https://github.com/snigelen/pcd8544">https://github.com/snigelen/pcd8544

3. LCD elastomeric Zebra connector problem

Because the thin PCB warps by the pressure of the LCD, the Zebra connector often makes a bad contact. Pressing the LCD to the PCB by soldering a wire between them solves the contact problems:



(http://www.avdweb.nl/Article_files/Arduino/Nokia-5110/Zebra-connector.jpg)

Solving Nokia 5110 LCD Zebra connector problem - Solving Nokia 5110 LCD Zebra connector problem

4. Notes

- Don't use "\n" after a 14 character line, this creates an extra blank line.
- Don't use endl, **`<< endl;`** takes 60 byte more program space than **`<< "\n";`**
- The PCD8544 LCD driver may be damaged if it is not properly reset.

Disclaimer (/disclaimer.html)
Contact (/contact.html)

Do you have any comments? Please let me know (http://www.avdweb.nl/contact.html).

Back to Top