Character LCD System Monitor

Created by Phillip Burgess

https://learn.adafruit.com/character-lcd-system-monitor
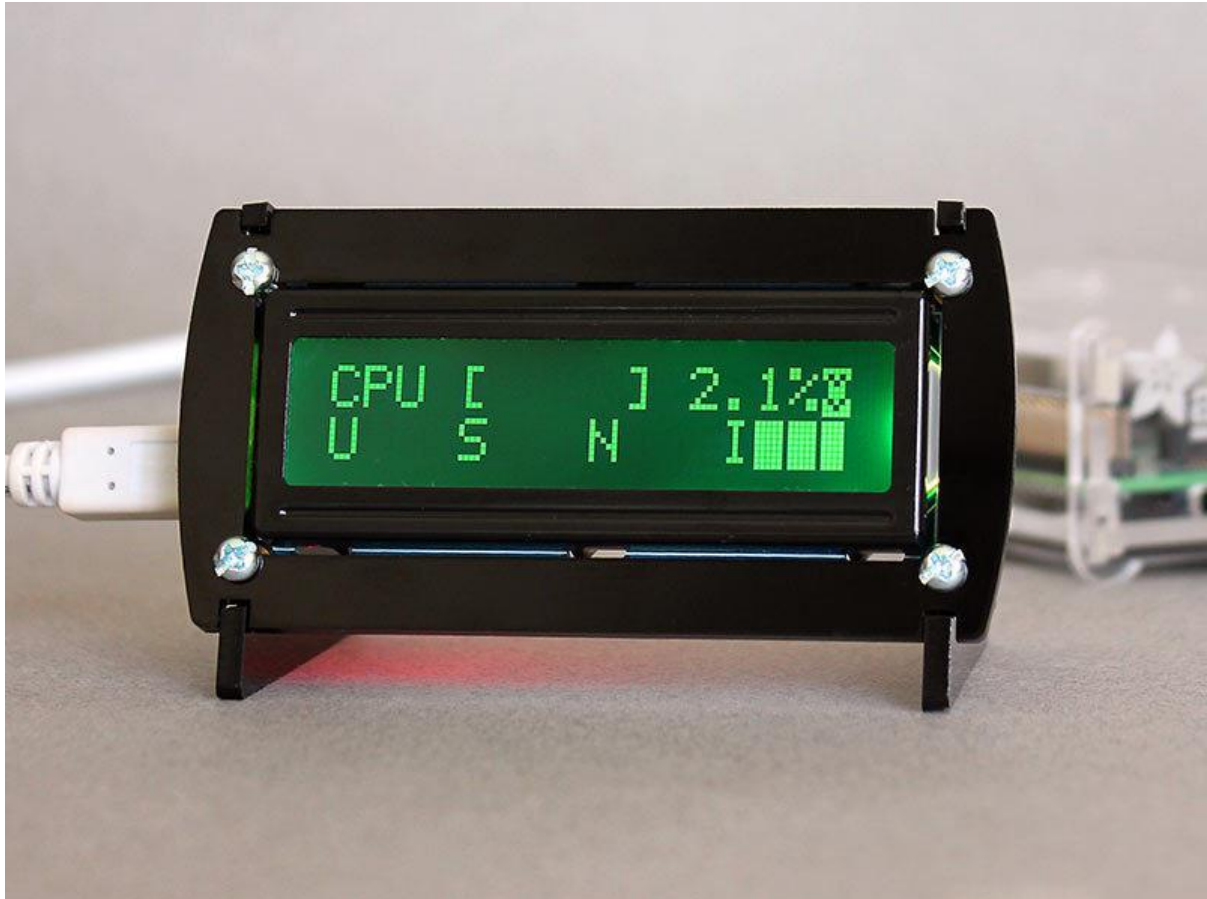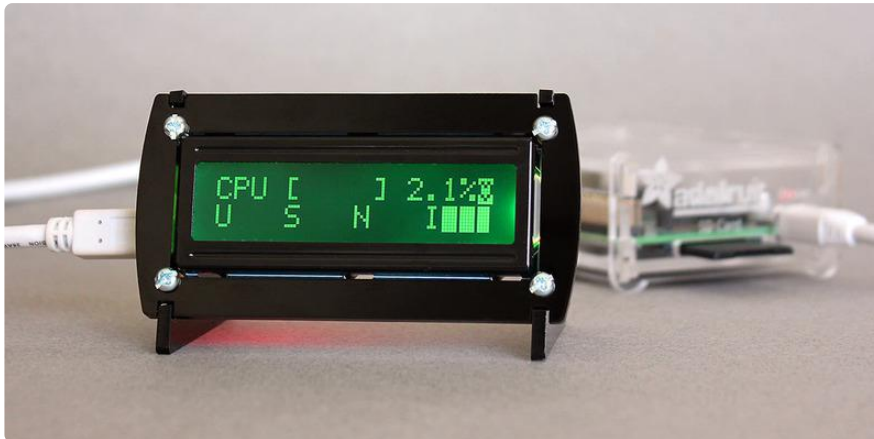
Last updated on 2023-08-29 02:20:01 PM EDT

Table of Contents

Overview

This guide walks through the process of assembling and configuring our LCD displays with USB/serial backpack and stand as a realtime system monitor.

These displays are great for monitoring the health and status of "headless" systems such as servers, small Raspberry Pi installations, or as an auxiliary information display on your regular computer.

You can get one of our cute acrylic stands in the Adafruit shop (http://adafru.it/1365)

Assembly

To assemble the LCD backpack and stand, you will need:

- Soldering accoutrements (iron, solder, etc.)
- A small Phillips screwdriver
- Some steps may require wire cutters; see notes below.
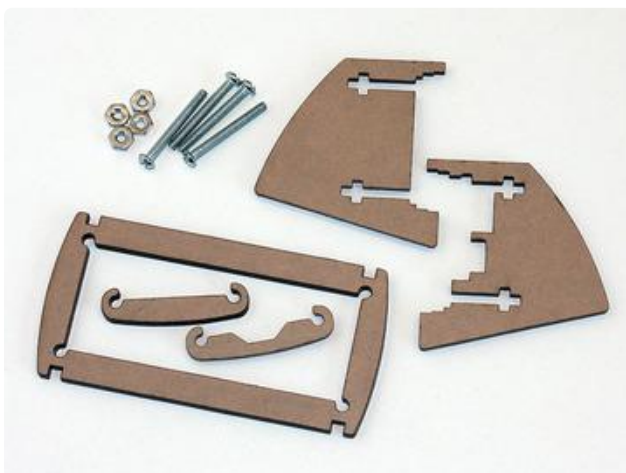
In addition to the electronics, the kit contains five laser-cut pieces and four #4-40 x 1" screws and nuts.

Peel the paper backing from both sides of the acrylic stand pieces. Some people use a razor blade as a scraper to get this started, but a fingernail at a corner is usually sufficient and won't gouge the plastic.

After peeling, the parts may have a little bit of paper soot on them from laser cutting. They can be washed with soap and water, but be absolutely certain that everything is completely dry before assembly! We can work on the electronics in the interim…

For reference, here's what a completed LCD+Backpack will look like, both front and back.
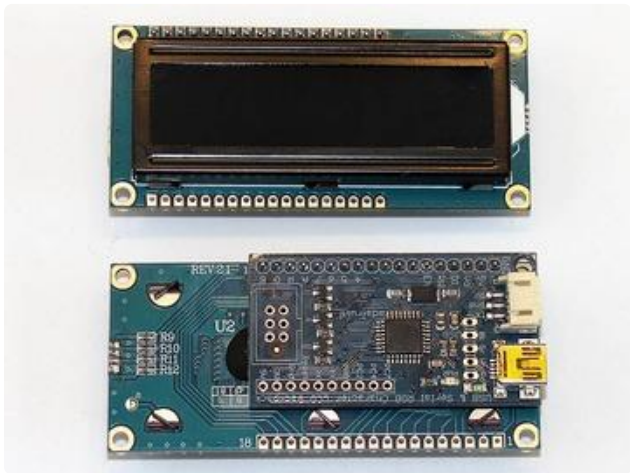
Looking at the face of the LCD, with the backlight LED protruding from the right, we'll be soldering to the header row across the top.

From the back, the USB and serial connectors should be along the outside edge, and the text on the backpack board is upside-down with respect to the LCD.

Usually the included pin strip is the same length as this header, but in some cases you may need to trim a few off. Use pliers or wire snips for this; doing it by hand sometimes snaps at the wrong spot.
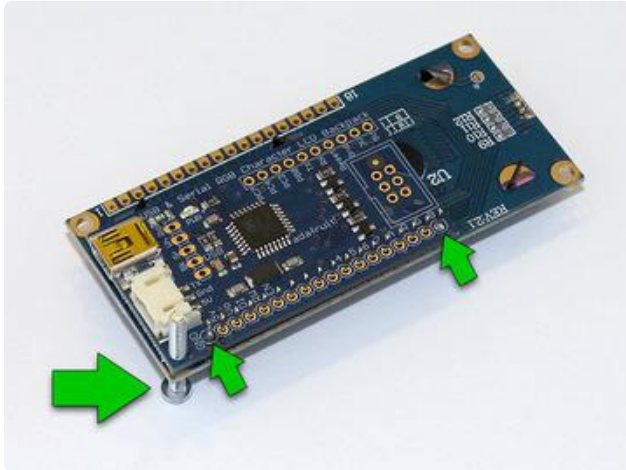
The pin strip has a long side and a short side. The long side of the pins should point back, to the header on the backpack board.

To help with alignment, temporarily insert one of the screws through the corner mounting hole on both boards.

Solder just the first and last pins on both front and back. Make sure both boards are aligned and parallel, and that the screw is pointed straight back. You can re-heat those four solder joints as needed while getting this alignment right.

Once everything is square and level, solder the rest of the header pins on both sides.
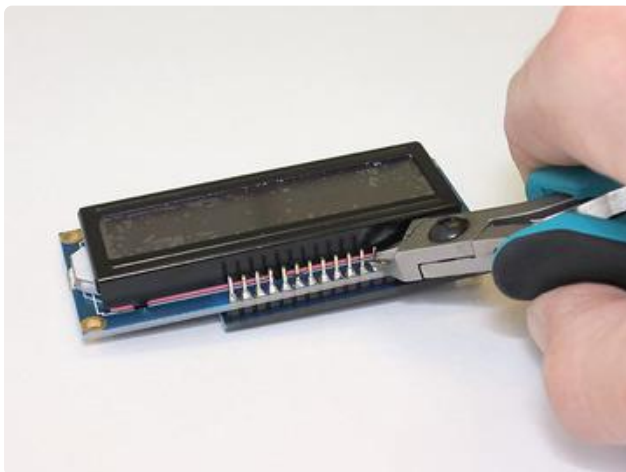
If you accidentally installed the pin header the wrong way (long side facing forward), no biggie…you'll just need to clip off the pins before fitting in the stand.
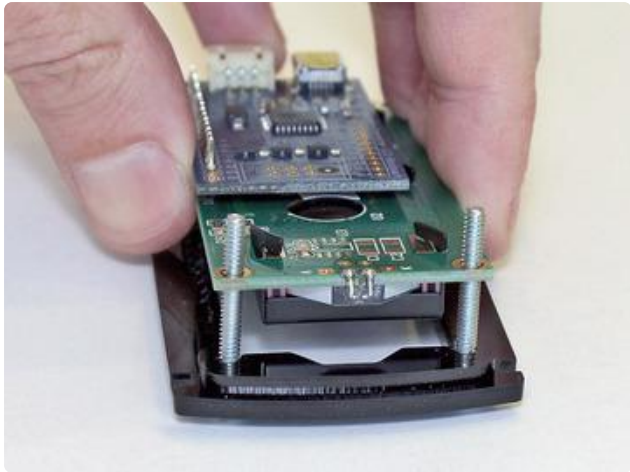
Insert 2 screws through one end of the front bezel, then slide one of the small spacer pieces (they resemble tiny handlebar mustaches) over the screws, and finally the LCD.

Let's start with the end where the LCD backlight protrudes, it's a little easier. The spacer for this end has an extra bite to fit around the backlight.
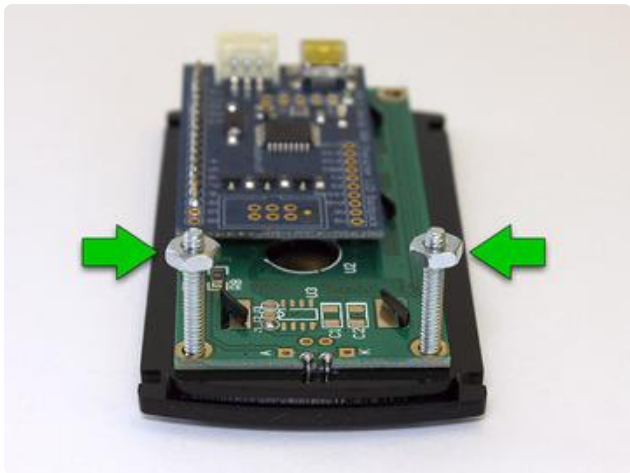
Add nuts to these two screws. Position them about 1/10" (2mm) down the screw.

Install the first side; the longer straight edge is the bottom. Align the front notches with the piece held at an angle, then pivot up so the screws/nuts fit into the cutouts . You may need to tweak the nut positions a little and try again.
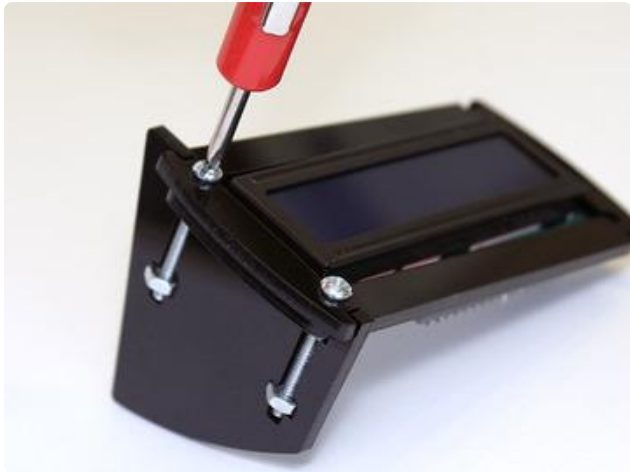
Give the screws a couple turns...just "finger tight" for now.
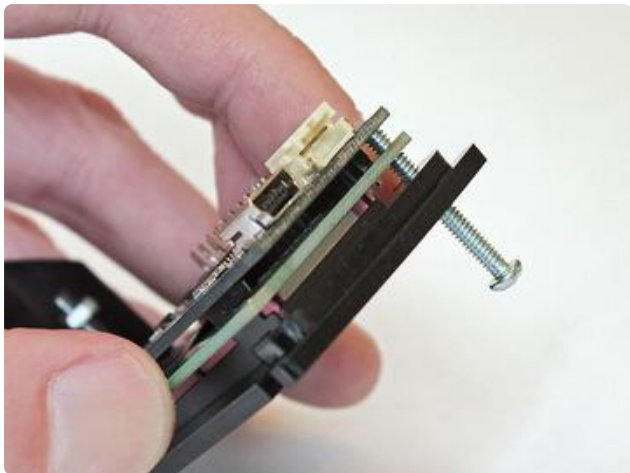
The spacer/screw arrangement is similar for the second side. With the first side already installed, we won't be lowering the board and spacer into place...instead the pieces are aligned and the screws inserted front to back.

Install nuts as before, and the second side then similarly pivots into place. This one's a little fussier due to the port cutouts.

When tightening the screws, give a gentle squeeze so there's good contact between the side and front pieces; we don't want the side piece to splay outward.

If everything looks good and straight, you can give all the screws an extra half turn or so to hold more firmly...but not so tight as to crack the plastic!

Taa-daah! Ready to plug in.

Windows

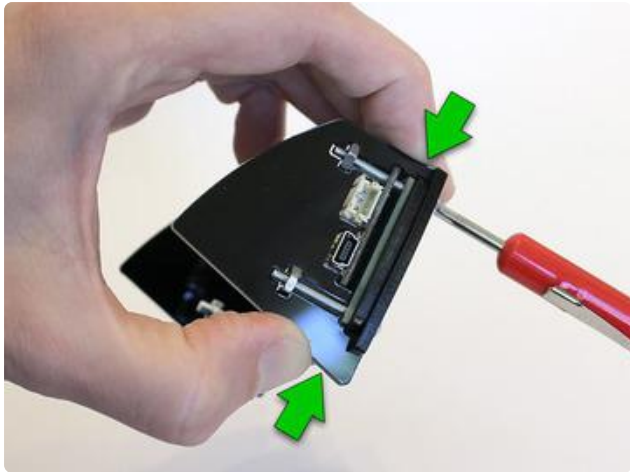To use the LCD+Backpack on Windows systems, you'll first need to download this .INF driver file (). This makes the device appear as a COM port.

Then you'll want to download LCD Smartie (), which provides all the system status and monitoring information.

With a clean system and the .INF file, you might be able to skip ahead to the "browse" step below (step 6). But it's pretty common to excitedly plug in the new device without the driver file ready, ending up with a botched and non-working device. If this happens, download the .INF and proceed through these steps...

0-devices

From the Start menu, select "Devices and Printers."
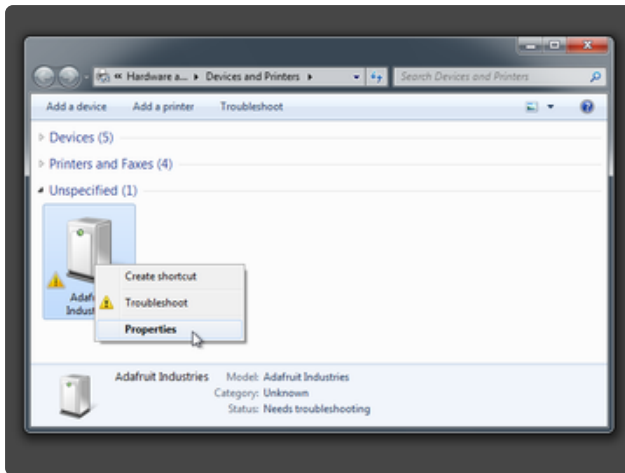
This panel lists devices (printers, etc.) connected to the system. At the bottom will be an unspecified device labeled "Adafruit Industries."

Right-click this item and select "Properties." (Or just double-click the icon.)
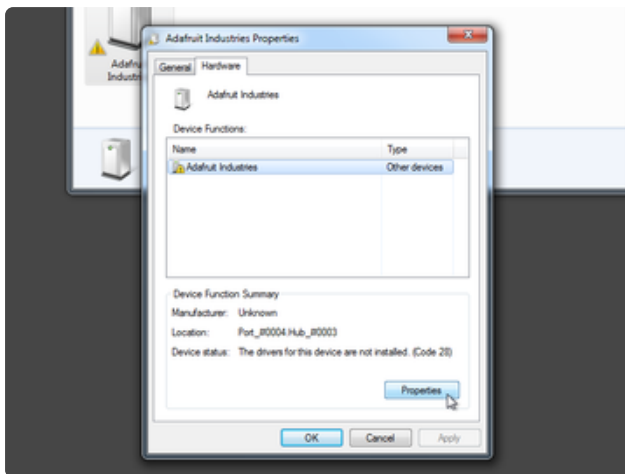
From the "Hardware" tab of the Properties panel, click the "Properties" button. It's properties all the way down!
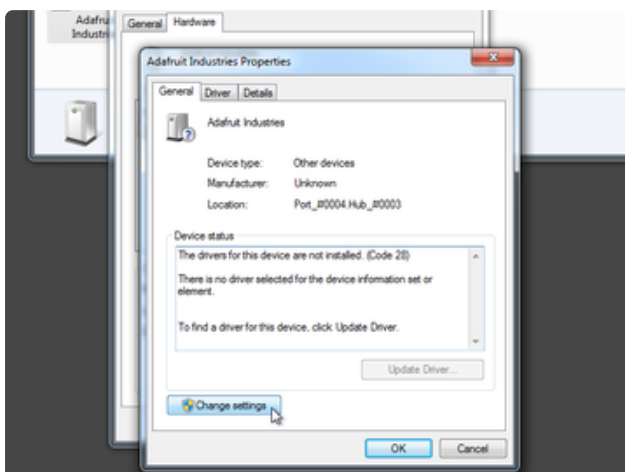
From the "General" tab, click "Change settings."
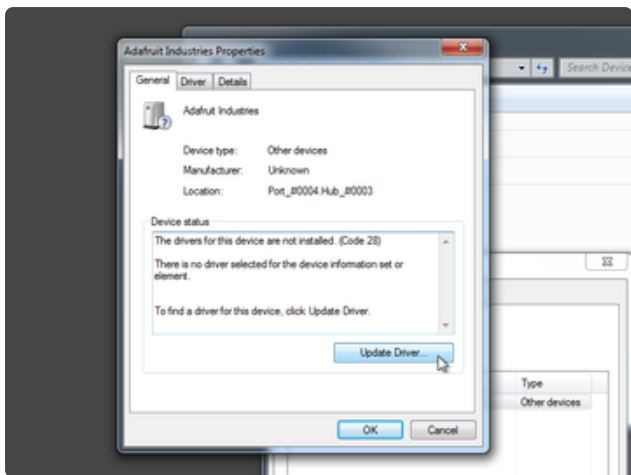
This should enable the "Update Driver…" button. Click…
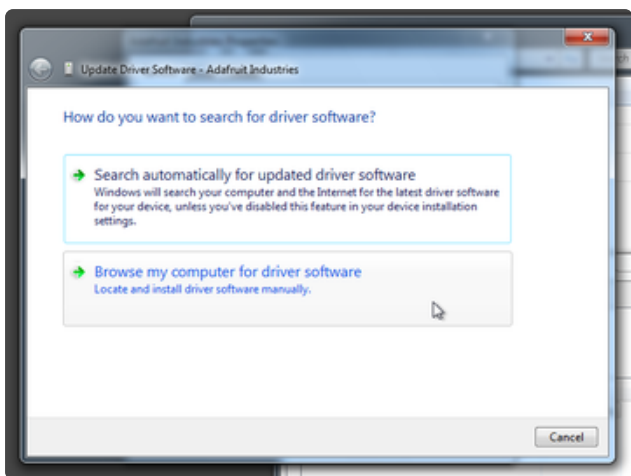
When prompted, select "Browse my computer for driver software" (do not use the automatic search).

You'll be presented with a file selection dialog. Navigate to the location of the previously-downloaded .INF file. It's probably in the Downloads folder.
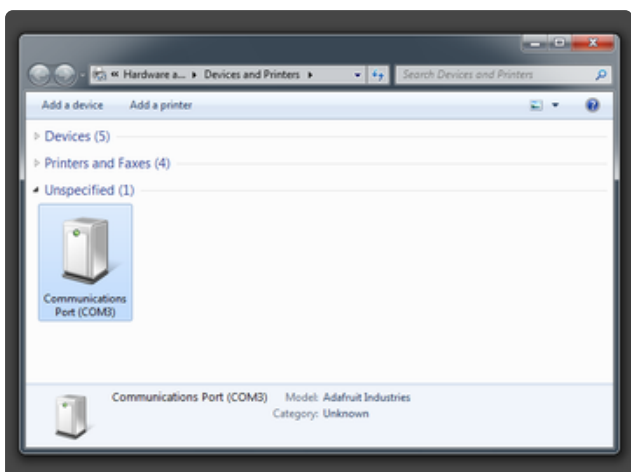
You'll see a message about an unsigned driver. That's okay, tell it to go ahead an install.

In a moment, the LCD+Backpack should show in the Devices panel as a "Communications Port," along with a COM port number. Keep track of that port number, you'll need it in a moment.

If you haven't extracted and installed the LCD Smartie software, go ahead and do that now.

LCD Smartie is a "classic" piece of software…the last official release was in 2007…but the good news is that it works well with current versions of Windows, though the user interface is a bit dated. This is a complex package with a ton of features, more than we can cover here. Some instructions are available on the LCD Smartie web site, or you can explore the GUI and work out most of these options.

First we need to set it up for our LCD+Backpack.

Click "Setup" in the small LCD Smartie window, which brings up the massive configuration dialog. From the "Screens" tab, in the "Display settings" section at the top right, select the "Plugin" tab. Select matrix.dll from the Display Plugin menu, and change the Startup Parameters to reflect the correct COM port for the device. You will need to quit and restart LCD Smartie after changing the COM port setting.
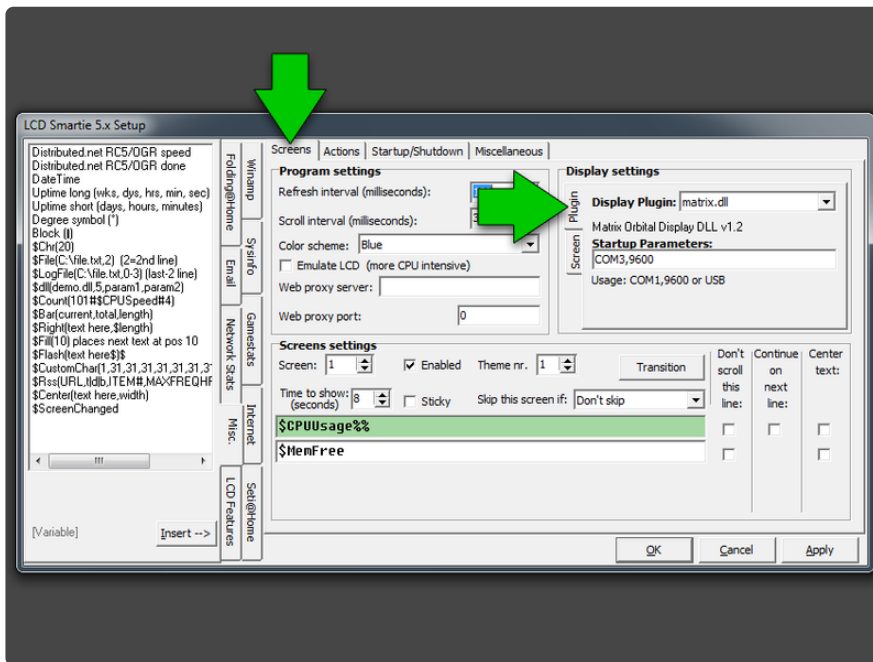
In the "Startup/Shutdown" tab, you'll see there are options to run the software automatically at system startup, so you don't need to go through this process every time.

Linux (incl. Raspberry Pi)

Installation (the Easy Way)

Many Linux distributions (Ubuntu, Raspbian "Wheezy," Occidentalis, etc.) offer LCDproc as a package, which greatly simplifies basic installation (though you'll still need to work through the configuration steps). From a command line, simply type:

```
sudo apt-get install lcdproc
```

You'll be prompted whether you want to install all the prerequisite packages. Answer "Y" and the software should all be download and installed in a minute or so.

Installation (the Hard Way)

It's not really that hard, just involves a few extra steps.

First, download the latest version of the source code, using this link () from a web browser, or from the command line type (as a single contiguous line):

```
wget http://sourceforge.net/projects/lcdproc/files/lcdproc/0.5.6/
lcdproc-0.5.6.tar.gz/download -O lcdproc.tar.gz
```

Then extract the archive. If using a GUI desktop, this can usually be done by double-clicking the file, or right-clicking and selecting "Extract…" or similar. Or from the command line, type:

```
tar xzf lcdproc.tar.gz
```

To build and install the software, you'll then need to enter these commands in a terminal window:

```
cd lcdproc-0.5.6
./configure
make
sudo make install
```

(You may need to change the first command to match the current version number.)

This should take just a few minutes to complete. If it stops due to a missing library, you'll need to track down and install the missing components (perhaps using apt-get).

USB Connection

Connect the LCD backpack to a USB port, then in a terminal type the following:

```
dmesg
```

Toward the bottom of the output you'll see references to a USB device. One of these lines will include a tty name, e.g.:

```
[ 3710.377212] cdc_acm 1-1.2:1.0: ttyACM0: USB ACM device
```

Remember that name "ttyACM0" (or whatever yours has assigned), you'll need it for configuring the software.

TTY Serial Connection

With a Raspberry Pi, we can skip the USB port and connect the LCD backpack to the Pi's GPIO header...especially useful on the Model A board.

Normally the serial connection on the header is used to log in with a terminal. This needs to be turned off in order to allow use with the LCD (the terminal function is then unavailable — you need to use a USB keyboard or a wireless network connection to log in). Two files need to be edited:

```
sudo nano /boot/cmdline.txt
```

Change this line:

```
dwc_otg.lpm_enable=0 console=ttyAMA0,115200 kgdboc=ttyAMA0,115200 console=tty1
root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline rootwait
```

to:

```
dwc_otg.lpm_enable=0 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4
elevator=deadline rootwait
```

(Basically, delete the references to ttyAMA0)

And:

```
sudo nano /etc/inittab
```

Comment out or delete the last line. i.e. change this:

```
T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```

to:

```
# T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```

Or simply delete that line.

Remember the name "ttyAMA0", you'll need it for configuring the software.

The system must be rebooted for this to take effect. You can either do that now, or after setting up the LCD configuration files.

Configuring LCDd, the LCD daemon

The location of the configuration file LCDd.conf can vary depending on the installation method used. If installed using the package manager (apt-get), then it's most likely in: /etc
If installed from source, it's most likely in: /user/local/etc
If it doesn't appear to be in either of those locations, use the find command:

```
sudo find / -name LCDd.conf -print
```

This file must be edited as root, e.g.:

```
sudo nano /etc/LCDd.conf
```

The default configuration file that ships with LCDproc is huge, with support for many devices and every imaginable option. Rather than try to guide you through this massive file, it's much easier to replace the entire file's contents with this much-simplified version, limited to our LCD backpack. Then only a couple lines need editing. (You may want to back up the original file first.)

```
# Pared-down LCDd configuration file; contains only those
# settings relevant to the Adafruit USB+Serial Backpacks.

# Global settings apply to all driver types ------------
[server]

# Path to driver modules.  IMPORTANT: EDIT THIS TO MATCH
# YOUR SETUP.  Last character should ALWAYS be slash.
DriverPath=/usr/lib/lcdproc/
# Select Matrix Orbital driver
Driver=MtxOrb

# Interface, port and user under which daemon runs
Bind=127.0.0.1
Port=13666
User=nobody

# If 'yes' server runs in foreground (handy during setup)
#Foreground=yes

# Settings specific to Matrix Orbital driver ------------
[MtxOrb]

# Output device.  IMPORTANT: EDIT THIS FOR YOUR SETUP!
Device=/dev/ttyACM0

# Speed applies only to serial pins; ignored w/USB.
Speed=9600

# Display size, type, contrast and brightness range.
Size=16x2
Type=lkd
Contrast=750
hasAdjustableBacklight=yes
```

```
Brightness=1000
OffBrightness=0
```

The lines of interest include:

```
DriverPath=/usr/lib/lcdproc/
```

This is the location of the LCDproc driver files, and will usually be either /usr/lib/lcdproc/ or /usr/local/lib/lcdproc/
As with the location of the configuration file, you might need to use the "find" command. When editing this line, make sure the last character is a forward slash: /

```
Device=/dev/ttyACM0
```

This is the device name for the LCD backpack, which was discovered in an earlier step. On most Linux systems it will be /dev/ttyAMC0 for a USB-connected backpack, or /dev/ttyAMA0 for a serial connection to the Raspberry Pi GPIO header.

One more line should be edited for initial testing:

```
#Foreground=yes
```

This line is commented out by default. Remove the initial # character to enable it. This keeps the LCDd process running in the foreground, so it's easily stopped with Control+C rather than having to muck around with kill commands. Once we know everything works, the comment character can be added back in.

Test Run

Let's see if it works! Type:

```
sudo LCDd
```

If this throws up an error message, it's most likely that the wrong device name was specified on the Device line, the DriverPath line doesn't correctly point to the location of the driver files (or is missing the trailing slash), or the configuration file is not in the expected location.

If LCDd runs as expected, you should get a greeting message on the LCD. That's good news! Press Control+C to stop the program, then we'll do more configuration…

Set up lcdproc and Auto-Start

If LCDd is running acceptably, we can restore it to the background configuration. Edit LCDd.conf as before and Add the initial # back to the "Foreground" line:

```
#Foreground=yes
```

There are a few different clients that can work with LCDd, but the most useful among these is lcdproc, which cycles through several system status screens.

lcdproc.conf is the configuration file for lcdproc. It should be located in the same directory alongside LCDd.conf. This is another hefty file to slog through...you're welcome to root around and change it to your liking, but the default configuration is pretty useful and we'll keep it as-is for simplicity's sake.

We'll set up both LCDd and lcdproc to start up automatically when the system boots.

If you installed using the package manager (apt-get), LCDd should already be set up to auto-start. Otherwise, you can create this file in /etc/init.d:

```
cd /etc/init.d
sudo nano LCDd
```

Copy and paste the following into the file as a start; we'll tweak a couple lines afterward.

```
#! /bin/sh
#
# LCDd   initscript for LCDd
# by Jose Luis Tallon &lt;jltallon@adv-solutions.net&gt;
#
### BEGIN INIT INFO
# Provides:          lcdd LCDd
# Required-Start:    $remote_fs $syslog
# Required-Stop:     $remote_fs $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: LCD daemon
# Description:       Control LCD displays connected to the computer
### END INIT INFO
#
#       Written by Miquel van Smoorenburg &lt;miquels@cistron.nl&gt;.
#       Modified for Debian
#       by Ian Murdock &lt;imurdock@gnu.ai.mit.edu&gt;.
#       Modified for lcdproc by Jonathan Oxer &lt;jon@debian.org&gt;
#
#

PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
DAEMON=/usr/sbin/LCDd
NAME="LCDd"
DESC="LCDd"
```

```
DAEMON_OPTS="-s 1 -f -c /etc/LCDd.conf"

test -x $DAEMON || exit 0

set -e

case "$1" in
  start)
    echo -n "Starting $DESC: "
    start-stop-daemon --start --quiet --background \
        --exec $DAEMON -- $DAEMON_OPTS
    echo "$NAME."
    ;;
  stop)
    echo -n "Stopping $DESC: "
    start-stop-daemon --stop --oknodo --quiet \
        --exec $DAEMON
    echo "$NAME."
    ;;
  restart|force-reload)
    echo -n "Restarting $DESC: "
    $0 stop
    sleep 1
    $0 start
    ;;
  *)
    N=/etc/init.d/$NAME
    # echo "Usage: $N {start|stop|restart|reload|force-reload}" &gt;&amp;2
    echo "Usage: $N {start|stop|restart|force-reload}" &gt;&amp;2
    exit 1
    ;;
esac

exit 0
```

The lines of interest are DAEMON and DAEMON_OPTS. Edit these to reflect the actual locations of the LCDd program and configuration file, respectively.

Make sure the script is executable:

```
sudo chmod 755 LCDd
```

Then we'll set up lcdproc to start a little later in the bootup process.

```
sudo nano /etc/rc.local
```

Toward the end of this file (but before the final 'exit 0'), add this line:

```
/usr/bin/lcdproc
```

You might need to change the path to reflect the installation on your system (e.g. might be /usr/local/bin or similar).

Now reboot the system. The LCD should spring to life automatically, cycling through several screens of system information.

Mac OS X

Setup for Mac OS X is similar to Linux, with a few extra hurdles:

- You'll need to [download and build from the source](https://) (); there's no packaged installer available (i.e. similar routine to "The Hard Way" in the Linux instructions).
- To do this, you'll need to have XCode installed (the currently-available version requires OS X 10.7 Lion or 10.8 Mountain Lion), as well as the supplementary Command Line Tools (Preferences→Downloads).
- Automated startup is more involved.

After downloading, double-click the file to extract the source code. Then open a Terminal window and "cd" into the newly-created folder:

```
cd lcdproc-0.5.6
```

Do not build the software using the exact Linux procedure — we need to make some small changes along the way. The first step is the same though. From the command line, in the source folder, type:

```
./configure
```

Then edit the source file for the Matrix Orbital driver:

```
nano server/drivers/MtxOrb.c
```

(Or you can use vi, emacs, or whatever the editor of your choice.)

Look for this line (around line 387):

```
p-&gt;fd = open(device, O_RDWR | O_NOCTTY);
```

Replace this with the following two lines:

```
p-&gt;fd = open(device, O_RDWR | O_NONBLOCK);
#undef HAVE_CFMAKERAW
```

The rest of the build process is then similar to Linux:

```
make
sudo make install
```

Configuration is then similar to the Linux directions, with the following changes:

- dmesg won't show the USB device name. Instead, you'll need to root around in the /dev directory to find the device name; e.g. start with:

```
ls /dev/tty.usb*
```

Compare the output with the device first disconnected, then connected, to help track down the correct device name, and edit the Device= line in LCDd.config to match.

- The LCD drivers are located in /usr/local/lib/lcdproc/ by default; edit the DriverP ath= line in LCDd.config to match (and remember to include the trailing slash).
- The LCDd daemon is located in /usr/local/sbin by default.
- The lcdproc client is in /usr/local/bin (and must be run as root — use "sudo" during initial configuration/testing).

The auto-start configuration is also notably different, given OS X's unique method of launching daemons.

First, edit the lcdproc configuration file so that it's respawn behavior doesn't interfere with the system's own facility:

```
sudo nano /usr/local/etc/lcdproc.conf
```

Look for this line, which is commented out by default:

```
#Foreground=true
```

Remove the initial # character to enable this line, then save the changes to the file.

It's slightly awkward...this line should be enabled in lcdproc.conf (which runs in the foreground), but disabled in LCDd.conf after testing (the background daemon).

One more change. Look for this line in lcdproc.conf (around line 77):

```
HighLoad=1.3
```

If you're on a multi-core system, increase this number to suit. For example, on a 4-core iMac, I'm using a 4X-ish larger value of 5.0.

This sets the processor load threshold at which the LCD backlight will blink (to indicate heavy work). At startup and login, a lot of processes are launched and may temporarily trigger this blinking if set too low.

Next, we need to create plist files for both programs. From the command line:

```
nano net.omnipotent.LCDd.plist
```

Copy and paste the following into the new file:

```
&lt;?xml version="1.0" encoding="UTF-8"?&gt;
&lt;!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/
PropertyList-1.0.dtd"&gt;
&lt;plist version="1.0"&gt;
 &lt;dict&gt;
  &lt;key&gt;Label&lt;/key&gt;
  &lt;string&gt;net.omnipotent.LCDd&lt;/string&gt;
  &lt;key&gt;ProgramArguments&lt;/key&gt;
   &lt;array&gt;
    &lt;string&gt;/usr/local/sbin/LCDd&lt;/string&gt;
   &lt;/array&gt;
  &lt;key&gt;KeepAlive&lt;/key&gt;
  &lt;true/&gt;
 &lt;/dict&gt;
&lt;/plist&gt;
```

Then a second file:

```
nano net.omnipotent.lcdproc.plist
```

Containing:

```
&lt;?xml version="1.0" encoding="UTF-8"?&gt;
&lt;!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/
PropertyList-1.0.dtd"&gt;
&lt;plist version="1.0"&gt;
 &lt;dict&gt;
  &lt;key&gt;Label&lt;/key&gt;
  &lt;string&gt;net.omnipotent.lcdproc&lt;/string&gt;
  &lt;key&gt;ProgramArguments&lt;/key&gt;
   &lt;array&gt;
    &lt;string&gt;/usr/local/bin/lcdproc&lt;/string&gt;
   &lt;/array&gt;
  &lt;key&gt;KeepAlive&lt;/key&gt;
  &lt;true/&gt;
 &lt;/dict&gt;
&lt;/plist&gt;
```

Set the ownership of both of these files to "root," then copy to the system daemon directory:

```
sudo chown root:wheel net.omnipotent.LCDd.plist
sudo chown root:wheel net.omnipotent.lcdproc.plist
sudo cp net.omnipotent.LCDd.plist /Library/LaunchDaemons
sudo cp net.omnipotent.lcdproc.plist /Library/LaunchDaemons
```

On the next reboot, the LCD should begin cycling status information per the lcdproc configuration.

If there's no response from the LCD, it may be the wrong device name in the LCDd configuration file. Double-check the value there against the contents of the /dev directory, and test by running LCDd from the command line.