LCARS-inspired Circuit Board Panel

Created by Anne Barela

https://learn.adafruit.com/lcars-inspired-circuit-board-panel

Last updated on 2023-10-18 01:51:00 PM EDT
# Table of Contents

**Overview**
- Project Level: Intermediate  

**Design**
- Realizing the Design

**Exporting for Gingerbread**

**Using Gingerbread**
- Open your SVG
- Ready to Export to KiCad

**Pasting the Gingerbread Data into the KiCad Footprint Creator**

**KiCad**

**Designing the LED Circuit: Schematic**
- Bill of Materials (BOM)

**Routing and Component Placement**

**Fabricating Your Design**
- Creating the Gerber and Drill Files

**Ordering**
- Parameters
- Checks

**Soldering**
- Debugging

**Use**
- Going Further
Overview

Making props from your favorite television and film franchises can be very tedious and expensive. There are whole forums dedicated to how others have sourced obscure parts and made replicas for hundreds of dollars.

Take Star Trek. There are beautiful graphics of the LCARS displays on Star Trek: The Next Generation and similar in other programs. These are often realized with two methods: A coated piece of plexiglas with the design and lit from behind, or a dedicated LCD panel. The first is not really touch reactive, the second is expensive to implement and hard to program. How can this be done simply?
Custom printed circuit boards (PCBs) have come down considerably in price. Even adding electronics can be done either by the builder or by some board houses, providing finished products.

This is the third in a series of PCB art builds, ratcheting up the design by adding capacitive touch and LEDs to come up with a Star Trek LCARS-like panel that lights buttons when touched. This is inspired by smaller LCARS panels used to open doors on The Next Generation and other shows.

Check out previous guides to see the basics of how to make PCB art projects:

- Making PCB Jewelry & Art with Gingerbread and KiCad ()
- Making Wearable Badge Art with Printed Circuit Boards ()
Project Level: Intermediate

This is an intermediate project. It assumes you know the basics of vector based drawing and design in a program like Affinity designer or perhaps Inkscape. It also assumes you know the basics of PCB layout with general familiarity with KiCad 7.0+. Previous guides help with this and it is recommended online documentation and how-to materials be consulted as needed if you intend to design your own smart PCB.

Finally, the components selected are surface mount solder (SMD), to keep all the electronics on the back and have a clean look on the front. This will require some intermediate soldering skills and a pair of tweezers.

Design

PCB costs usually go by the square inch / centimeter and the number of layers, although specific types of finish may cost more. For me, looking at making an LCARS panel, I was inspired by a couple of websites noting the smaller door opening panels in Star Trek: The Next Generation.

Due to copyright I cannot post a picture directly but I can link you to other maker efforts here and here. And one by a Star Trek Production Designer here. The first two use print on plexiglas and the second a dedicated LCD panel.

I knew I could make some very nice panels using techniques in my previous PCB guides mentioned on the Overview page.
Realizing the Design

I opened Affinity Designer 2, a low cost vector graphics program. It allows various artwork to be placed in layers:

Layered Images in a Modern Photo Editor

Stack each of these layers together to produce the final result

The final layers correspond to layers on the circuit board:
To design the front, I would want a background that is white. That maps to the Front Solder Mask. Then the black parts would be Front Silkscreen printed on the white. I wanted a logo that would "pop", so I designed my own, a bit different than some but still iconic, and that is in the Front Copper layer. The logo would be within a rounded square that had no silkscreen or solder mask, allowing it to be illuminated from behind, if desired, through the substrate layer.

Each element is built up. First was the outline which defined the shapes of the rest of the elements within. The "D" shaped keys with identification numbers were drawn next. On top of each key, I chose simple numbers rather than some LCARS panels with random numbers. Drawing ne key, I copy & pasted 4 more in a row down the board. I mirrored the "D" for the left side keys and copied them also.

The rounded square is a standard shape for the logo area and a rounded rectangle for a top Activate button (not functional in this build). The logo was traced from a design I found and scaled appropriately. This can be replaced if another logo is desired.
The final design has appropriate elements grouped into several defined layers, corresponding to the PCB layers above and named for using Winterbloom's free Gingerbread conversion tool, which takes an Affinity Designer file and generates the code to make a KiCad PCB program footprint file.

You can see my design below, load it into Affinity Designer 2 and see how I built up elements into definitions for each layer defined by Gingerbread.

Panel Affinity Designer 2 .afdesign File

For details on how I've been using Affinity Designer 2, see the previous guide Making Wearable Badge Art with Printed Circuit Boards () which goes into more detail on using AD2.

Next, to go from Affinity Designer art to KiCad PCB art.

Exporting for Gingerbread

Save your work at this point in Affinity Designer 2 by using File -> Save. That will result in a .afdesign file with your work.

To export the file from the native .afdesign format to an SVG file for Gingerbread, go to the File menu and click the Export option.
At the top of the dialog box of options, select SVG.

The current Gingerbread site has a picture of the export settings for a previous version of Affinity Designer. The current version is Affinity Designer 2 (or higher) and the menu changed.

The settings below map to the ones Gingerbread wants. If the Raster DPI is blank, do not set it using the dropdown. Manually type in 2540.
Once all the settings are correct, click the Export button in the lower right to save the file as an SVG file. This is the file you will drag and drop in the Gingerbread application (note: not the .afdesign file, Gingerbread only takes especially formatted SVG files).

Using Gingerbread

Open Gingerbread by going to `gingerbread.wntr.dev` in your modern web browser. You'll see a screen like the one below.

There are three example designs you can look at to see how Gingerbread does its thing. You can also download the associated .afdesign files and look at how they were created in Affinity Designer 2.
Open your SVG

In your file application (Windows, for example, would be File Explorer), open up the directory containing your SVG file made with Affinity Designer 2. Drag the SVG file to the blank area below the existing examples.

If Gingerbread parses the SVG file as valid, you will have a screen similar to the one below. If it doesn't work, check the settings used to export an SVG graphics file from Affinity Designer 2 from the previous page.
You can toggle the layers on and off in the lower right. You can change the color of the solder mask and silk layers for better viewing.

Use the view to make sure all the layers are the way you want. You can separate the layers and view the back with the black buttons at the bottom of the frame.

The colors you select for solder mask and silkscreen are only for visualization in Gingerbread. They will not be exported to KiCad. Generally these are specified during submission to make the boards and not in the data file.

As an example of the warning above, you can use any colors you wish in Gingerbread, but different board house fabricators have different processes, not the full range of colors.

In the OSH Park normal service, the Solder Mask color is purple and the silkscreen is white.

Other PCB fabricators, such as JCLPCB and PCBWay, may have more flexible options.

Search different manufacturers for the colors and processing you want.
Ready to Export to KiCad

If the design is looking like you want it, you then click the "Export to Clipboard" button in the lower right part of the window. This will load the clipboard with the data the KiCad Footprint Editor expects for a new footprint (part).

Pasting the Gingerbread Data into the KiCad Footprint Creator

At this point, you should have used Gingerbread to load your clipboard with footprint data.

Open KiCad. As of this guide, the version is 7.x. You should get the following screen if you’ve opened the main KiCad application and not one of the submodules:

Open the KiCad main window and create a new project (I use New LCARS below), then Open the Footprint Editor by going to Tools -> Footprint Editor.
Here is the KiCad Footprint Editor:
Click the New Footprint Icon (circled in red on the left) or use File -> New Footprint.

Select a name for your footprint -- since it isn't a standard component, for the Footprint Type select Other.

The text that appears is used in a standard circuit. As we do not want our design labeled by KiCad, click each line of text and press the Delete key to get rid of it, giving a blank slate in the central footprint area.

In the central footprint window, right-click and select Paste. Or use File -> Paste.

If the Gingerbread data is still in the clipboard, it will paste your design into the Footprint Creator. If it doesn't, go back to the Gingerbread web window and press the Export to Clipboard button again and try to paste once more in KiCad.
You should then see your design. Move it so it is centered in the footprint area and click your mouse to "drop" it.

Don't worry about the colors at this point.

You'll rarely need to do anything else here. If there is anything you want on the PCB that is not available in the Affinity Designer 2 -> Gingerbread -> KiCad workflow, you can do that now. Most times, you can just select File -> Save and you'll get a dialog to save the footprint.
You should look to make a new library for your design - I selected "Anne". Do this using the New Library button located on the lower left. I named this footprint "LCARS Front". Click Save.

Now you have your design in a footprint library, congratulations.

KiCad

If you don't already have a project, click File -> New Project. Select the name representing your design.

Go back to the main KiCad menu and click on PCB Editor (the green one).

The PCB Editor looks like the screen below:
Near the upper right of the PCB Editor screen is an icon of a chip. That is the Add a Footprint tool. Click it and you get a window to select a footprint. Select your footprint library and your design.

Place it in the center of the PCB editor window.

Zoom out if necessary to see your entire sheet.

In the 3D Viewer (View -> 3D Viewer), you can see your front design.

You should save your design into a project file and the PCB file will have a .kicad_pcb file extension.
This project features capacitive touch for each of the ten keys. The keys 1-10 use the PCB as a capacitive touch pad. The inputs, from large copper pads under the buttons, are connected by a PCB via to a 10K ohm resistor and then into an AT42QT1070-S capacitive touch chip. Each AT42 can handle 5 inputs and 5 outputs. So to handle 10, two identical circuits are needed.

The design is directly from the Adafruit AT42QT1070 breakout board and its product guide. It is used in momentary mode, when you touch the pad, an LED comes on.

In the schematic editor, draw two copies of the AT42 circuit with values changed between them appropriately. The battery is specified separately.

Bill of Materials (BOM)

KiCad has built-in PCB footprints for some of these items:

- A Keystone CR2032 battery holder
- 1206 size 470 and 10K ohm surface mount resistors

The LEDs are the same Inolux IN-S124AR LEDs used for the NASA badge. The footprint was tweaked to make the LED hole larger to accommodate reverse
mounting. This was not quite sufficient (the board house doesn't like a hole so close to the LED pads) so a bead file is used on each hole to make it even a bit bigger.

Looking at DigiKey, the AT42QT1070-S in a SOIC-14 package (3.9x8.7mm, 1.77mm pitch) was chosen. These were placed above the buttons on the PCB back. The battery holder was placed at the top opposite the Activate button. Care was made to avoid placing components in the bare board area to facilitate light shining through, if desired.

The final bill of materials is below. Prices may change of course and I ordered more of the small parts in case of lose (they are so tiny) and price breaks on the LEDs at 100. Parts were bought to populate 5 boards, the typical minimum number from board houses. But also a good number to have parts for board revisions.

Note: You can buy parts from any supplier, this is not a DigiKey sponsored project.

Routing and Component Placement

Each button has an LED and 470 ohm resistor for showing it's been touched. It also has a 10K ohm resistor connected to a via, a connection between the top copper and bottom copper. The copper under each button made in Affinity Designer acts as a touchpad and this is connected by the via to a trace leading to the 10K resistor to the AT42 chip. It appears the touchpad copper is left unconnected as a circuit but, trust me, it works.
ith the art design on front, it wouldn’t be as nice to route copper traces between components on that side. So all traces between components was routed on the bottom copper layer, effectively making the routing only on one side (barring the capacitive touch keys on the front). With two 14 pin chips, that is a mess of connections and very tricky to lay out neatly and effectively.

Tips used to do the layout:

- The LEDs were placed on the outside of the buttons due to ergonomics, so it made sense to route the 10K sense resistors on the interior portion of the board.
- First trick was checking the pinout of the chips. It made the most sense to put one with pin 1 pointing up, and the other rotated 180 degrees with pin 1 facing down (the bottom of the board). This had the most pins routing to the sense resistors and the opposite pins to the LEDs.
- The choice above is NOT 100% ideal as not all pins for a function are limited to one side of the chip.
- The second trick was to route some traces under the some resistors.
- An ideal layout is a clean, orderly one but this is much harder in this complex circuit compared to the NASA badge project which had a very simple circuit.

There are many tutorials for making good cad designs and in using KiCad, so I will not repeat them here.
Fabricating Your Design

Creating the Gerber and Drill Files

While some PCB manufacturers can take a KiCad data file directly for a design, the introduction of KiCad 7 in 2023 has some manufacturers behind on supporting the new file format. No worries, one can export the files needed. Select File -> Fabrication Outputs -> Gerbers shown below.

Use the Plot button to generate the Gerber files. I recommend making a subdirectory under your project to put them in. After that press the Generate Drill Files button. Save the files in the same directory as the Gerber Files.
These files describe exactly what each layer should have with the drill files showing the exact place to have holes drilled.

Go to the directory with all the generated files. Zip all the files into one Zip file package. If a board house cannot use KiCad 7 files, they will definitely use the raw files, most often sent in a single Zip file.

Upload the Gerber and Drill Files as a zip file (latest)

Ordering

As mentioned earlier, you can select from a number of manufacturers in several countries. One of the main considerations is if they make boards in the colors you want.

The colors for this project are white solder mask, black silkscreen and silver (bare metal). In the end I also wanted it thinner for the LEDs, 0.8mm thickness. I could get this combination at PCBWay.com (Note: this is not an endorsement and there no consideration given / sponsorship / discount for using them).

Note that colors and thickness might slow production from generic green boards as the manufacturer must gather enough boards to panelize for manufacturing. If you want a board to just look at and do electrical tests, you might specify default colors and thickness then use special processing on boards that will be final use.

Parameters

There are a wide variety of manufacturing choices available.

FR-4 is the most common board type and the one described earlier. Two layer boards are more common and this guide's design uses both sides.

Choose the color for the board and silkscreen.

Two different finishes are available for exposed metal: HASL and ENIG.

HASL () is when the board is dipped in liquid solder, making a silver finish. Lead & Lead-free (RoHS compliance) is usually available.
**ENIG** consists of an electroless nickel plating, covered with a thin layer of gold, which protects the nickel from oxidation. While a bit more expensive, it provides a pretty finish that will not oxidize readily.

For the LCARS, the logo is the only exposed copper layer and I just ordered HASL but if you want it in gold, use ENIG for an additional charge.

The choices for this project are circled in green in the screen above. Note that this particular manufacturer has a minimum order of 5 boards.

<table>
<thead>
<tr>
<th>Parameter Information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Board type</strong></td>
</tr>
<tr>
<td>Different design in panel</td>
</tr>
<tr>
<td>Size</td>
</tr>
<tr>
<td>Layer</td>
</tr>
<tr>
<td>Thickness</td>
</tr>
<tr>
<td>Min hole size</td>
</tr>
<tr>
<td>Silkscreen</td>
</tr>
<tr>
<td>Surface finish</td>
</tr>
<tr>
<td>Via process</td>
</tr>
</tbody>
</table>

**Checks**

The board house will look at your design to see if it is manufacturable according to their processes. Likely they will note any issues and bet back to you with issues and if you should go into KiCad to make adjustments.

They did ask if the LED holes should be plated, I said no.

Now sit back and wait until your boards come.
The parts arrive and I have five beautiful panels, art in their own right. But there is more to add to make them shine.

Ladyada said 1206 resistors are fine but it takes time, tweezers and a steady hand to attach them. Be sure you have a soldering iron with a fine tip. I did not use solder paste myself. If you are familiar with paste, you can order a stencil with the boards at an extra cost, place the parts and heat the board.

There are two types of tweezers in the Featured Products page for holding SMD parts while soldering.

If you have not soldered surface mount (SMD) parts with a soldering iron, see this tutorial.

For the ICs, I referred back to the schematic for orientation. As seen above, the left handed chip U2 has pin 1 facing the battery connector, U1 on the right pin 1 is pointing down to the components below.

I lightly tinned all the IC pads - not much solder at all. Then I carefully placed the chip in the correct orientation. Heating up a corner leg, it anchors the chip and you can review to see if the chip is still square on the pads. If not, heat and reposition. Once done, go to the opposite leg and tack that down. Then go one by one heating a leg to solder in place. If you are proficient at solder flux, you have an easier time of it.
Solder the battery holder with the + pointing down towards the bottom of the PCB.

Next solder the resistors. Be sure to get the 470 ohm on the right pads and then do the 10K resistors. Tin one pad, place the resistor on with tweezers, heat to affix then use light solder to get the other end affixed. Repeat for all resistors.

The holes for the LEDs to face into the board may be a bit small. I used a bead file, also called a needle file, to slightly enlarge the holes so the chunky center of the LED fit inside the hole. Drilling would be possible if the bit is only slightly bigger than the existing hole, be careful.

![Image of LED](image)

For the diodes, I found the small green vertical green line from the larger horizontal green line faced the top of the board (as shown, the small square faces to the battery holder). You can test yourself: put a battery in and then put an LED on the pads and touched the button to ensure it would light, then solder it (with the power off) correctly oriented.

Once all soldering is done, install a good battery and flip to the top side. Touch each number pad and the corresponding LED should light up.

Debugging

If one doesn't light:

1. Be sure the LED is positioned correctly. I found the small green vertical green line from the larger horizontal green line faced the top of the board.
2. Ensure the IC, resistors and LED for that button are making contact with the circuit board. Do not heat too much or a pad may lift off the board.
3. If a whole row of LEDs doesn't light, double check the battery is good, the column IC is oriented correctly has power to the correct legs and all solder connections are good. A magnifying glass or illuminated microscope help here.

Use

The video above demonstrates the panel. As buttons are pressed, the corresponding LED is lit.

Going Further

Instead of driving LEDs, the signals could be fed to a microcontroller to do all sorts of actions. The panel may be independently lit from behind to make the logo area stand out:

Having the art in the form of a printed circuit board opens up these types of interactive props to a wide range of creative opportunities at a low cost.