

Atari Punk Console

Assembly, Testing, and Designer's Guide

Board version 1.1

Guide version 1.2

Jesse Hamner

Contents

1	Introduction	2
2	Best Practices	3
3	Useful Equipment For Testing	3
4	Assembly Order	3
5	Hardware Testing	4
6	Board Layout Images	6
7	Device Schematic	11
8	How To Make The Images In This Document	12
9	Change Log	17



Originally developed in collaboration with the UNT Libraries.

1 Introduction

In 1983, or some time before that, [Forrest Mims III](#), a long-time participant in the electronics, hobbyist, and atmospheric science communities, designed the “Stepped Tone Generator” using the wildly popular and hugely versatile [555 timer integrated circuit](#). He published the design in a Radio Shack (Tandy Corp) book of projects*, and then again in the 1992 *Engineer’s Mini-Notebook*†.

I was unable to find an example of this board that presented a simple design, an easy soldering task, but also an easy path to customization. People who are getting in to electronics, soldering, and/or “making” may want a few easy “wins” for their first few projects, and the “[Atari Punk Console](#)”, as it has become known, is a simple circuit with a modest amount of expandability.

I have re-implemented the design with the intent to provide a series of stepping stones from novice to an intermediate “maker.” Maker spaces like UNT’s Factory provide a range of tools to end users, but like many tools, learning to use them can be a challenge. Here again, this design of the APC is up to the task. With easy alignment notches and a downloadable reference design for a 3D-printed case, users can learn to solder, solder the APC, and print a case for it, without facing serious hurdles.

This project and its ancillary pieces allow novices to gain experience in several aspects of hobbyist making, while not facing the full force of electrical or industrial engineering. Put another way: one should not be required to learn typesetting in order to print a Microsoft Word document; neither should one be required to understand G-Code or STL files in order to produce a working and fun electronics project.

This guide provides graphical and instructional information about the assembly of this version of the “Stepped Tone Generator”. Please read this guide all the way through before beginning any work on the boards.

Legal

This document is intended to assist in the assembly of a circuit board project that synthesizes sound waves. This document comes with ABSOLUTELY NO WARRANTY. This document is released under the Creative Commons Attribution-ShareAlike (CC BY-SA) license version 4.0. For more, see creativecommons.org.



*Mims, Forrest M., III. 1984. *Engineers’s Mini-notebook: 555 Timer IC Projects*. Fort Worth, Texas: Radio Shack.

†Mims, Forrest M., III. 1992. *Engineer’s Mini-Notebook*. 3rd edition. Fort Worth, Texas: Radio Shack.

2 Best Practices

- Use a wrist-strap ground during soldering; these components can be easily damaged by static electricity
- Use thin (perhaps 0.032-inch or 0.8mm) solder for through-hole components
- Use a good quality soldering iron
- Clean the tip often on the brass scrubber
- Re-tin the tip often
- Lifting PCB pads and traces is quite possible without much effort—watch the iron temperature, and don't apply heat for more than ten seconds per soldered pin
- Don't buy or use low-quality solder, leaded or not; it gets gummy and is more or less impossible to remove without thermal damage to your components. Use name-brand solder, like [Kester](#)
- Masking tape may be helpful to hold pin headers in place while soldering the underside
- Tweezers or small pliers are useful (but not necessary) for placement of the SMD component

3 Useful Equipment For Testing

- A 9V battery
- A 6V to 13.8V DC power source, capable of providing at least 300 milliamps of current
- A multimeter
- A stereo magnifier, an inspection loupe, and/or a USB microscope
- A third hand or PCB vise
- A set of headphones or a powered speaker

4 Assembly Order

- Discrete resistors
- Diodes (note the orientation!)

- Disc capacitors
- Electrolytic (aluminum, polarized) capacitor(s)
- 2- and 3-pin headers
- LED (note the orientation!)
- 8-pin integrated circuit sockets (note the orientation!)
- DC power jack
- 9V battery cradle
- SMD headphone jack (see text)
- Variable resistors (potentiometers / rheostats)
- Switch

After the sockets are installed, consider the orientation of the 555 timers and press them completely into the sockets. It may be helpful to line up four pins on one side with the socket connector, insert them about 1mm, then gently press the opposite side pins inward until they can also be fit into the socket.

When soldering the stereo jack, heat a large pad on the PCB and melt a little solder onto it. Align the jack and hold it in place (the tab will be on top of the now-tinned PCB pad) with pliers, tweezers, or whatever you have. Hold the soldering iron tip on top of the tab, or touching both the pad and the tab, and the heat will eventually melt the solder and bond the tab to the board. If you have correctly aligned the jack, the other tabs and pads will be easy to solder. If not, re-melt the solder until you can get the jack to align properly. It may be helpful to press down with the tweezers or pliers (even clamping the board and the jack together with them). You don't want the tab to "float" up on the solder blob.

5 Hardware Testing

- Turn the switch to 'off'
- Install a 9V battery in the battery carrier
- Using a multimeter, check the voltage at the output (left, on this board) terminal of diodes D1 and D2
- Turn the switch to 'on' and check the power LED indicator
- If the LED does not illuminate, use a multimeter to confirm that there is at least 6 volts available at the LED positive terminal

- Set all the knobs to the middle
- plug in the headphones and check for sound

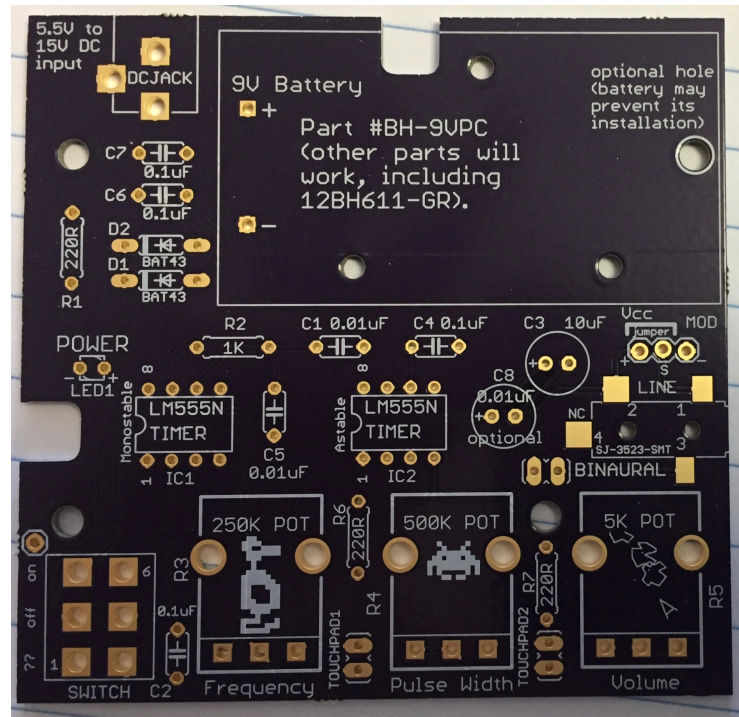


Figure 1: The printed circuit board for this project.

6 Board Layout Images

The next section includes images of the board design and layout for reference during the soldering and testing process.

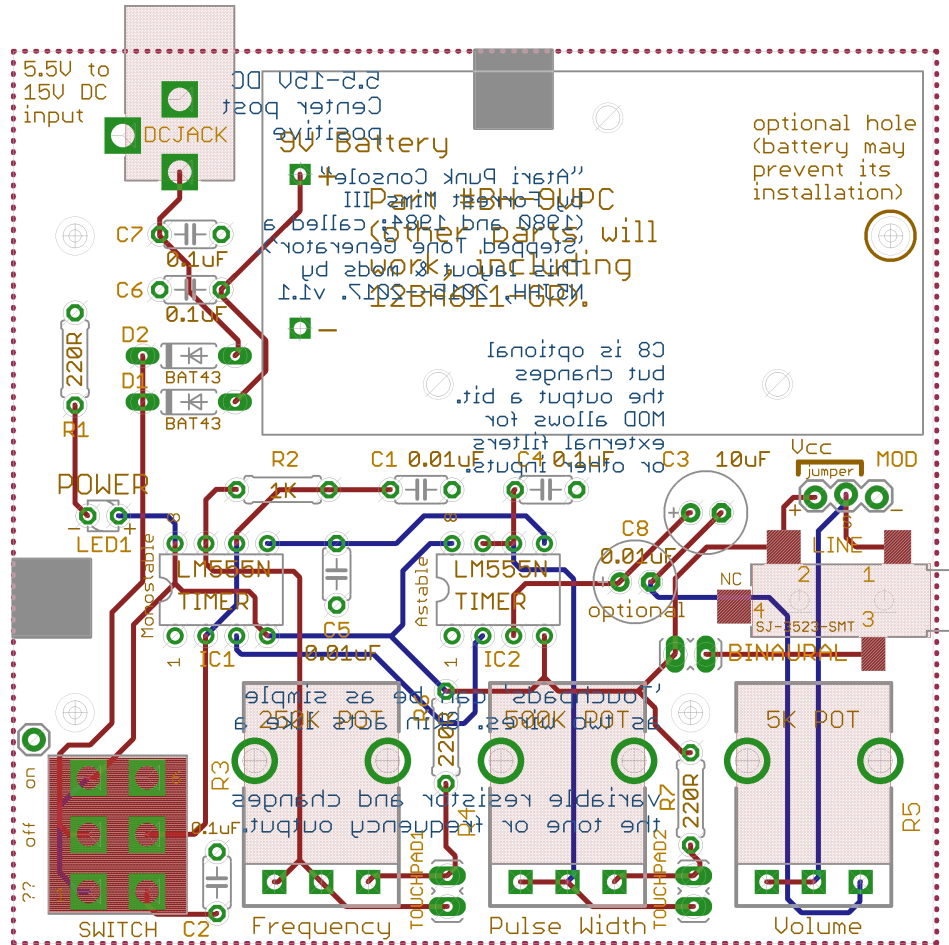


Figure 2: Color image unsuitable for monochrome printing. Both layers visible. Red traces are on the top layer, and blue traces are on the bottom layer. As would be expected if the board was transparent, all components on the bottom (although there are none for this board), and text, are reversed.

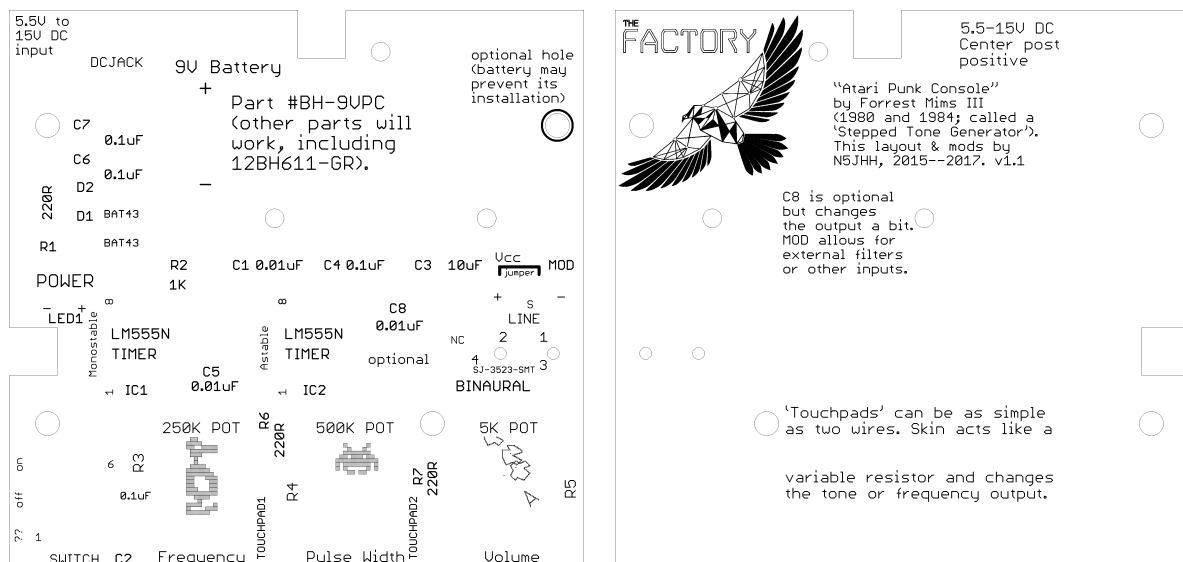


Figure 6: Silkscreens for top and bottom of the board. On the top silkscreen, note that the 8-bit Atari and Taito characters will be hidden under the rotary potentiometers once installed. From left to right: a dragon from the Atari 2600 *Adventure* game; a Taito Space Invader; and a screen from *Asteroids*. On the bottom silkscreen: note optional modification information and the Factory logo, copyright UNT Libraries.

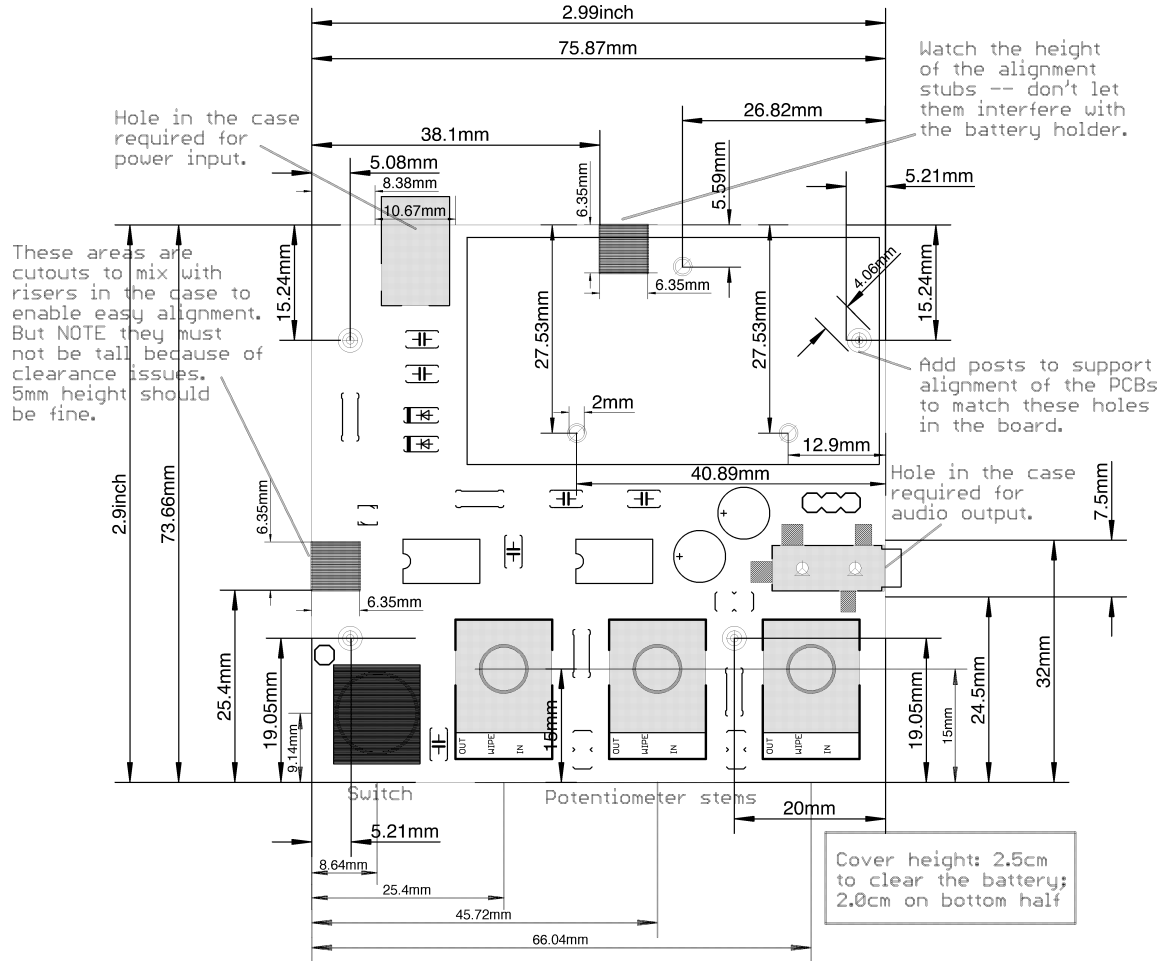


Figure 7: Board dimensions, should you want to design an enclosure for it. The three 2mm holes in the battery cradle could be attached to the underside of the enclosure with machine screws and nuts, but all fasteners must be flush with top surface of the cradle, meaning the underside of the enclosure would need recessed pockets for the nuts, or else rubberized feet that were taller than the portion of the fasteners that protrude through the bottom of the enclosure. It is also possible to create standoffs *within* the case that provide underside clearance for fasteners such that the fasteners do not have exposed surfaces outside the case, at the slight expense of a taller overall case.

7 Device Schematic

Although not required to successfully assemble and use the Stepped Tone Generator, some may find the schematic useful or interesting. There are also many schematics of similar implementations of Forrest Mims's work online. This implementation stays very close to the original, with a few small additions like the touchpad headers and the optional capacitors, that can be ignored if desired. I varied a few of the other components based on experience. To create this image, from the schematic pane, type (no quotes!)

```
EXPORT IMAGE $HOME/some/useful/path/APCschematic.png 800
```

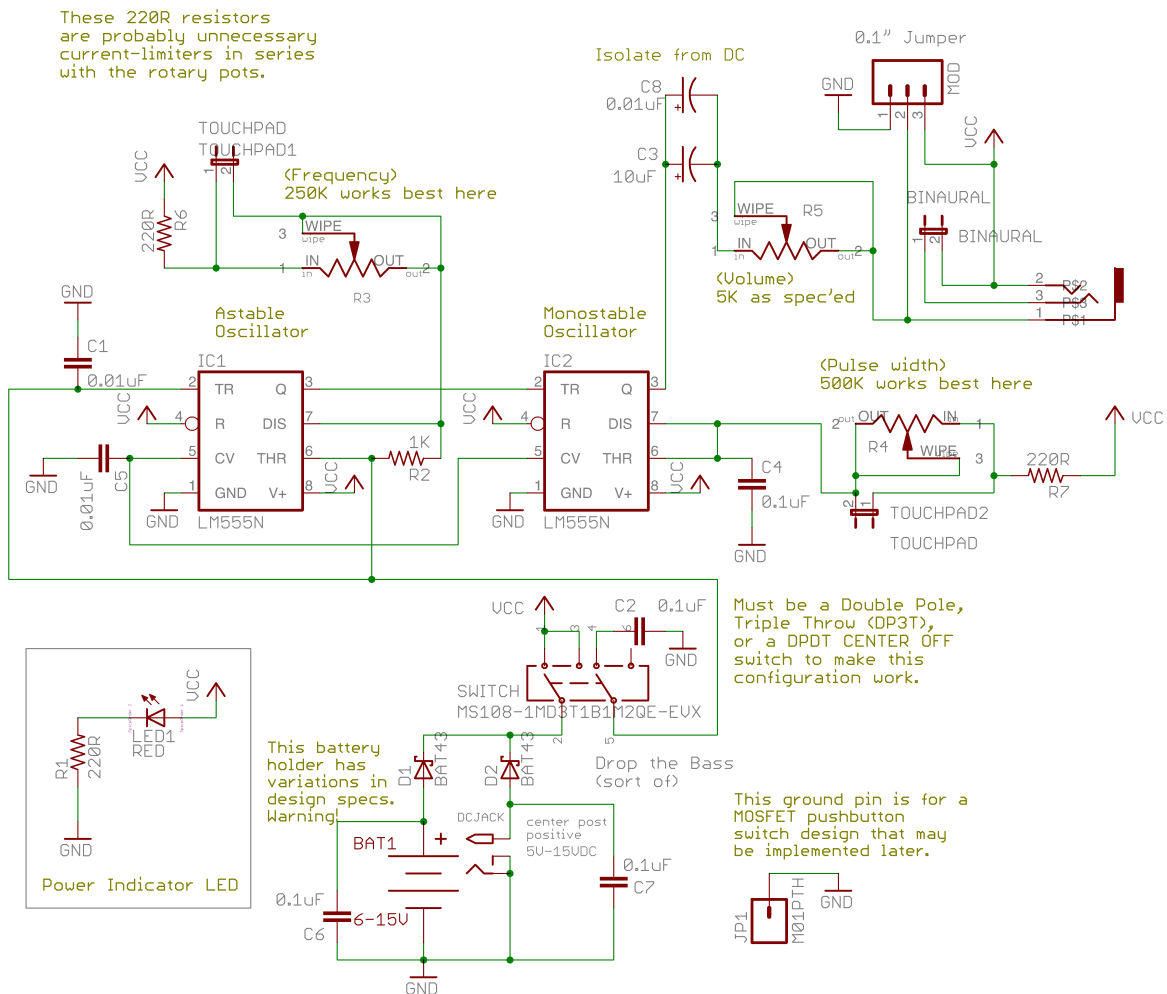


Figure 8: Schematic, with notes, as implemented in the board for this version.

8 How To Make The Images In This Document

This document includes images created from within Eagle CAD and using external programs. To show how it is done, and to provide any hints to those who would like to see how the documentation was made, I include the following mini-tutorial.

EagleCAD Schematic and Layout Images

An EagleCAD script is provided in Figure 9 and in the GitHub repository. Run the script from within EagleCAD and the first set of images will appear. Basically, the script turns on the layers for each particular image, sets color or monochrome, sets the resolution to 600 DPI, and exports each resulting layer set to a PNG file. For instance, before making the *top* layer, the script turns off the set of layers in EagleCAD that would make it hard to parse the top components – i.e. all of the “bottom” layers, like 16, 22, 24, 26, 28, 30, 32, 34, 36, 40, 42, 52, 122, 126, 128, and, as noted above, 200 (where the logo image silkscreen resides). To make an image of the underside, a similar operation is performed, blanking the “top” layers. The script writes image files called:

- fullcolorboard.png
- boardlayout.png
- boardtoplayout.png
- boardbottomlayout.png
- boardbottomsilk.png
- boardtopsilk.png
- boarddimensions.png

The script to appropriately convert the images is seen in Figure 10. The resulting files are called:

- colorlayout.png
- whiteboardlayout.png
- toplayout.png
- bottomlayout.png
- topsilk.png
- bottomsilk.png
- dimensions.png

This script was written for Mac OS X but should run on any system with [bash](#) and [ImageMagick](#) installed.

```

1 # An EagleCAD script to export four versions of a circuit board:
2 # 2 layers in color; and B/W, two layers, top layer, bottom layer,
3 # and silkscreen layers for each top and bottom.
4
5 # Jesse Hamner, July 2016
6
7 # Turn on the layers we want to see (note the RIPUP; won't render polygons)
8 # 200bmp is off, because it obscures the bottom layer
9
10 SET COLOR_LAYER fp3 7;
11 RIPUP @ GND;
12 DISPLAY NONE TOP BOTTOM PADS VIAS DIMENSION UNROUTED TPLACE BPLACE TNames BNames
   TCREAM BCREAM TKEEPOUT BKEEPOUT TRESTRIC BRESTRIC VRESTRIC DRILLS HOLES _tsilk
   _bsilk _tNames _bNames _tValues _bValues Drill_legend _bDocu fp3
13
14 # Export the color image:
15 EXPORT IMAGE $HOME/AtariPunkConsole/docs/images/fullcolorboard.png 800;
16
17 # Monochrome image, both layers:
18 EXPORT IMAGE $HOME/AtariPunkConsole/docs/images/boardlayout.png MONOCHROME 800;
19
20 # Turn on the top layers only:
21 SET COLOR_LAYER fp3 8;
22 DISPLAY NONE TOP PADS VIAS DIMENSION TNames TCREAM DRILLS HOLES _tsilk _tNames
   _tValues Drill_legend TKEEPOUT TRESTRIC fp3
23
24 # Monochrome image, top layer:
25 EXPORT IMAGE $HOME/AtariPunkConsole/docs/images/boardtoplayout.png 800;
26
27 # Turn on the bottom layers only:
28 DISPLAY NONE BOTTOM PADS VIAS DIMENSION BNames BCREAM DRILLS HOLES _bsilk _bNames
   _bValues Drill_legend _bDocu BKEEPOUT BRESTRIC fp3
29
30 # Monochrome image, bottom layer:
31 EXPORT IMAGE $HOME/AtariPunkConsole/docs/images/boardbottomlayout.png 800;
32
33 # Top silk:
34 DISPLAY NONE DIMENSION _tsilk TNames 201bmp 202bmp 203bmp
35 EXPORT IMAGE $HOME/AtariPunkConsole/docs/images/boardtopsilk.png MONOCHROME 800;
36
37 # Bottom silk:
38 DISPLAY NONE 200bmp DIMENSION _bsilk _bNames BNames
39 EXPORT IMAGE $HOME/AtariPunkConsole/docs/images/boardbottomsilk.png MONOCHROME 800;
40
41 # Dimensions for the case:
42 DISPLAY NONE DIMENSION tPlace tCream tKeepout tRestrict Holes Milling Document
   Reference Notes
43 EXPORT IMAGE $HOME/AtariPunkConsole/docs/images/boarddimensions.png MONOCHROME 800;
44
45 # Turn all the layers back on, including 200bmp this time
46 DISPLAY NONE TOP BOTTOM PADS VIAS UNROUTED TPLACE tDocu bDocu BPLACE TORIGINS BORIGINS
   TNames BNames TVALUES BVALUES TSTOP BSTOP TCREAM BCREAM TKEEPOUT BKEEPOUT
   TRESTRIC BRESTRIC VRESTRIC DRILLS HOLES _tsilk _bsilk _tNames _bNames _tValues
   _bValues _bDocu 200bmp DIMENSION
47
48 # <EOF>

```

Figure 9: The EagleCAD script to export images from EagleCAD.

```

1  #!/bin/bash
2  # Program to convert image output from EagleCAD to a print-friendly format.
3  # Note this script requires ImageMagick to be installed.
4  # Jesse Hamner, July 2016
5
6  function trimnegate {
7      echo "Converting_$1.png_to_$2.png:"
8      convert "$1.png" -trim -negate $3 "$2.png"
9      rm "$1.png"
10 }
11
12 # To create the full color image but keep the color information:
13 IMG="fullcolorboard"
14 DEST="colorlayout"
15 echo "Converting_${IMG}.png_to_${DEST}.png:"
16 convert ${IMG}.png -transparent black ${IMG}NB.png
17 convert ${IMG}NB.png -alpha off -trim -fuzz 10% -fill white -opaque black \
18 ${DEST}.png
19
20 # To invert the colors of the multi-layer monochrome image:
21 IMG="boardlayout"
22 trimnegate "${IMG}" "white${IMG}"
23
24 # To invert the colors of the top/bottom layer and convert to grayscale:
25 DEST="toplayout"
26 IMG="board${DEST}"
27 trimnegate "${IMG}" "${DEST}" "-colorspace_gray-gamma_0.3" # improve contrast
28
29 DEST="bottomlayout"
30 IMG="board${DEST}"
31 trimnegate "${IMG}" "${DEST}" "-colorspace_gray-flop-gamma_0.3" # and reverse image
32
33 # Do the silkscreen layers :
34 DEST="topsilk"
35 IMG="board${DEST}"
36 trimnegate "${IMG}" "${DEST}"
37
38 DEST="bottomsilk"
39 IMG="board${DEST}"
40 trimnegate "${IMG}" "${DEST}" "-flop" # reverse the bottom image
41
42 DEST="dimensions"
43 IMG="board${DEST}"
44 trimnegate "${IMG}" "${DEST}" "-colorspace_gray-gamma_0.9" # improve contrast
45
46 convert APCschematic.png -gamma 0.95 -brightness-contrast -20x60 APCschembetter.png
47
48 # Clean up:
49 for F in fullcolorboardNB fullcolorboard
50 do
51     if [ -e "${F}.png" ]; then
52         rm ${F}.png
53     fi
54 done
55 # <EOF>

```

Figure 10: The bash script to correctly color the exported images from EagleCAD.

8-bit Art in Eagle CAD

The PCB also includes three pieces of art from the ancient era of video games. The Asteroids ship was converted using the `import-bmp.ulp` user language script that is included in Eagle CAD. The other two presented a fun challenge, which was to draw the 8-bit sprites using the Eagle CAD scripting language. It's pretty easy to draw squares and rectangles using the language, so I wrote a Python script that reads in a very simple csv file, composed of rows of comma-separated "1" or "0" cells. The Python script then writes out a nice (and easily scalable, if you check out the top of the script) Eagle CAD script that draws the figure in the layer you specify. It's much faster than drawing it yourself, and the CAM script runs much faster too, since the sprites are so simple.

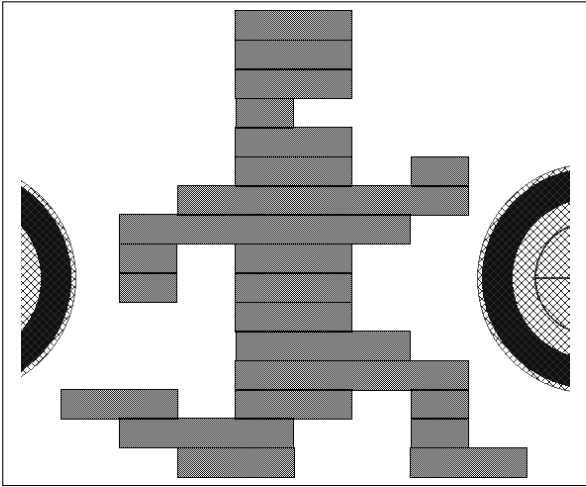
CSV code	Image
<pre> 1 0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0 2 0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0 3 0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0 4 0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0 5 0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0 6 0,0,0,0,0,0,1,1,1,1,0,0,1,1,0,0 7 0,0,0,0,1,1,1,1,1,1,1,1,1,1,0,0 8 0,0,1,1,1,1,1,1,1,1,1,1,0,0,0,0 9 0,0,1,1,0,0,1,1,1,1,0,0,0,0,0,0 10 0,0,1,1,0,0,1,1,1,1,0,0,0,0,0,0 11 0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0 12 0,0,0,0,0,0,1,1,1,1,1,1,0,0,0,0 13 0,0,0,0,0,0,1,1,1,1,1,1,1,1,0,0 14 1,1,1,1,0,0,1,1,1,1,0,0,1,1,0,0 15 0,0,1,1,1,1,1,1,0,0,0,0,1,1,0,0 16 0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1 </pre>	

Figure 11: The comma-separated source text that `eagledraw.py` uses to write the script to draw Pitfall Harry, and the resulting image in Eagle CAD. Pitfall Harry is not displayed on the the board, but is used here as an example.

Bill of Materials

Part	Value	Package	Description	Digi-Key Part
<input type="checkbox"/> BAT1	6-15V	BH9VPC	Battery holder is BH-9VPC, but part 12BH611-GR is substitutable.	BH9V-PC-ND
<input type="checkbox"/> BINAURAL	0.1" header	JP1	jumper for two-channel audio output.	
<input type="checkbox"/> C1,C5,C8	0.01 μ F	C050-024X044	ceramic disc capacitor	BC1078CT-ND
<input type="checkbox"/> C2,C4,C6,C7	0.1 μ F	C050-024X044	ceramic disc capacitor	399-9870-1-ND
<input type="checkbox"/> C3	10 μ F	E2,5-7	polarized capacitor	P15799CT-ND
<input type="checkbox"/> D1,D2	BAT43	DO35-7	Schottky diode, 0.45V fwd, 30V, 200mA	497-2492-1-ND
<input type="checkbox"/> DCJACK	PJ102A	PJ-102A	DC jack, 2.0mm center pin, 5.5mm plug	CP-102A-ND
<input type="checkbox"/> IC1, IC2	LM555N	DIL08	555 Timer IC	296-9684-5-ND
<input type="checkbox"/> JP1	M01PTH	1X01	1-pin header to ground	S1011EC-40-ND
<input type="checkbox"/> LED1	859-LTL2R3KRD-EM	5mm, 2.54mm spacing	red LED, 2mA, 1.8V fwd	160-1853-ND
<input type="checkbox"/> LINE	SJ-3523-SMT	SJ352 Audio Jack	CUI inc Stereo Audio SMT jack, 3.5mm	CP-3523SJCT-ND
<input type="checkbox"/> MOD	0.1" header	1X03	3 pin header for mods	(see above)
<input type="checkbox"/> R1,R6,R7	220R	0207/7	through-hole resistor	CF18JT220RCT-ND
<input type="checkbox"/> R2	1K	0207/7	through-hole resistor	S1KQCT-ND
<input type="checkbox"/> R3	250K	CTS UM296RE pot	Frequency potentiometer	CT2269-ND
<input type="checkbox"/> R4	500K	CTS UM296RE pot	Pulse-width potentiometer	CT2270-ND
<input type="checkbox"/> R5	5K	CTS UM296RE pot	Volume potentiometer	CT2264-ND
<input type="checkbox"/> SWITCH	108-1MD3T1B1M2QE-EVX	—	Mountain Switch DPDT center-off switch	(only at Mouser)
<input type="checkbox"/> TOUCHPAD1,2	0.1" header	JP1, JP2	optional touch pad connection	(see above)

9 Change Log

Revision 1.1: March 2017

- Fixed wiring on two potentiometers (I outsmarted myself and wired them in reverse, so the volume control, e.g., worked backwards)
- Re-routed traces to make the design friendly for desktop CNC mills
- Added two current-limiting resistors just to use best practices for the rotary potentiometers; the extra 220 Ohms doesn't affect the sound, but prevents runaway current should the wiper fail.

1.0: February 2017

- Initial release
- Changed switch to one in current production
- Re-routed and simplified some traces
- Moved and added labels
- Moved some discrete components

0.4: July 2016

- Shrunk the footprint of discrete resistors
- Renumbered a few components
- Altered the 9V cradle part's footprint to improve pin placement
- Changed a few labels

0.3: June 2016

- Changed switch
- Moved a few components for clearance

0.2: January 2016

- Revealed a problem with the switch layout

0.1: December 2015

- Corrected problems with component layout, clearance, and part names
- Added board cutouts to ensure it only fits in the enclosure in one specific orientation
- Shrunk discrete component pin spacings from 10mm to 7mm

This is the last page.