Cyclops is a high-power LED driver that enables precise control of light power for optogenetic stimulation. The circuit was developed by Jon Newman while in Steve Potter's lab at Georgia Tech in order to complete his thesis work, which required the delivery of ultra-precise, continuously time-varying light waveforms for optogenetic stimulation [1]. This was, and still is, not possible with commercial hardware for optogenetic stimulation. Since its first use, the circuit has been improved in terms of speed, precision, programmability, and ease of use. This document provides construction, usage, and performance documentation for the Cyclops LED driver. This document evolves with the repository. To view old revisions, checkout tags or old commits using their SHA.

Contributors

- jonnew
- Sung-Yon Kim

Table of Contents

Features

Circuit Features

- Ultra-precise
- High power
- Up to 1.5A per LED
- Wide bandwidth
  - ~2.5 MHz -3 dB bandwidth
  - Maximum 200 ns 1.0A rise-time
- Current and optical feedback modes
- Built-in waveform generation
- Over-current protection
- Modular
  - Arduino compatible
  - 4 synchronizable optical channels
  - Accepts external analog, gate, or trigger inputs

Multiple stimulus generation options

- External stimulus sequencer
- External digital trigger
  - TTL logic level
- External analog waveform generator
  - 0-5V analog signals
- Internal 12-bit DAC
  - Synchronized across up to 4 drivers
  - Powerful Arduino library
  - Programmable triggering logic
  - Respond to USB input

Performance Specifications

The following oscilloscope traces give indicates of the circuit's precision and speed. Note that time series traces are **not** averaged - these traces display per-pulse temporal characteristics. Optical characteristics and optical feedback signal for the Cyclops driver were provided by a Thorlabs PDA36 amplified photodiode set to 0 dB of transimpedance gain. Measurements were performed a single Osram golden dragon LED.

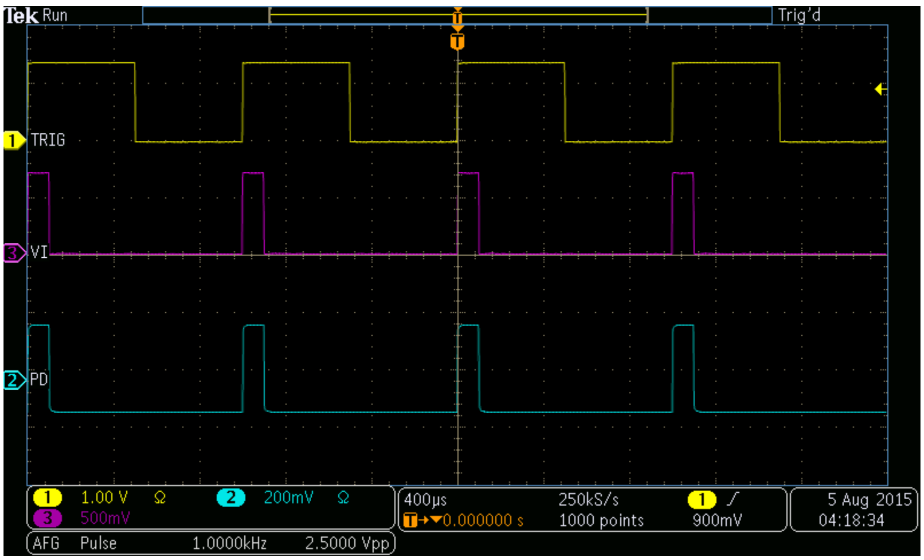


Figure 1: Trigger (yellow), current (pink), and light power (blue) traces during pulsed operation in current feedback mode. Input waveform is a 1 kHz 0 to 750 mV, 10% duty cycle square wave.

The following traces are the same as the previous ones except that the amplified photodiode was used to provide optical feedback. The slowdown compared to current feedback is due to a speed of the photodiode. A faster amplified photodiode would provide crisper rise and fall times

The current-feedback mode -3dB bandwidth was determined by applying a flat noise signal over 50 MHz with mean = 1.0V and Vpp = 500 mV into the `EXT` port with maximal current gain. It occurs at around 2.5 MHz.

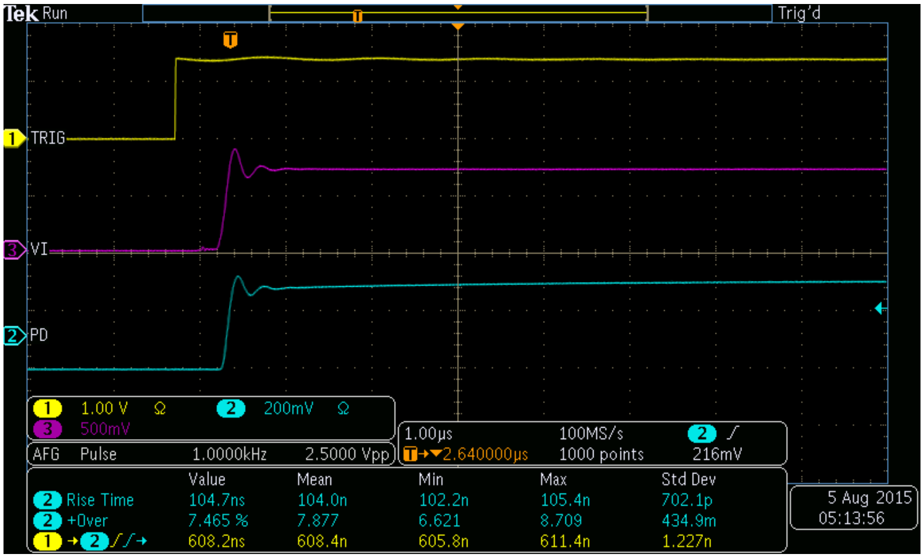


Figure 2: Zoomed traces showing waveform 10-90% rise times. Optical rise time statistics are shown at the bottom of the image.

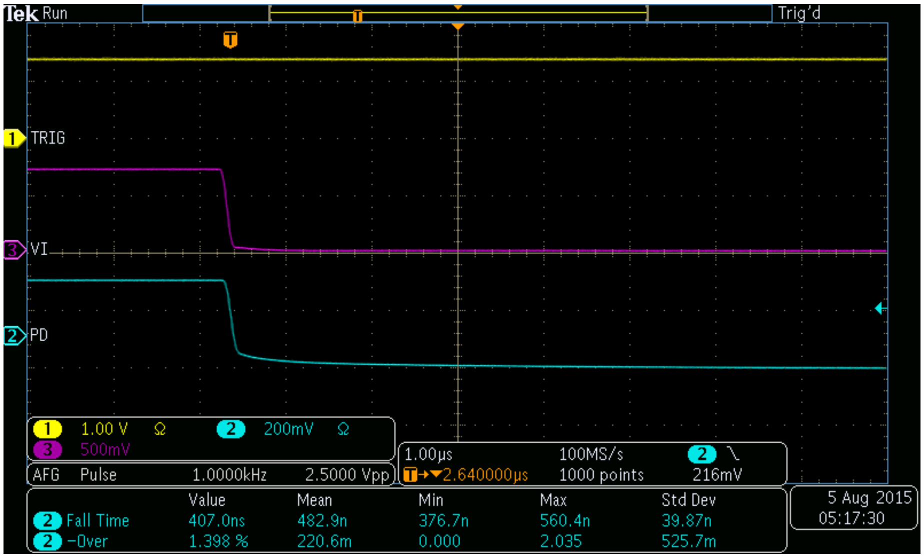


Figure 3: Zoomed traces showing waveform 10-90% fall times. Optical fall time statistics are shown at the bottom of the image.

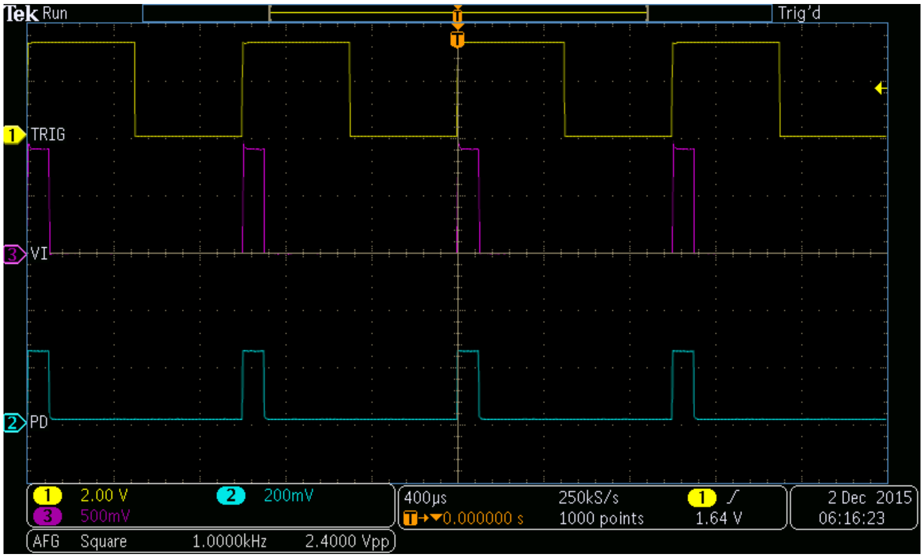


Figure 4: Trigger (yellow), current (pink), and light power (blue) traces during pulsed operation in optical feedback mode. Input waveform is a 1 kHz 0 to 750 mV, 10% duty cycle square wave.

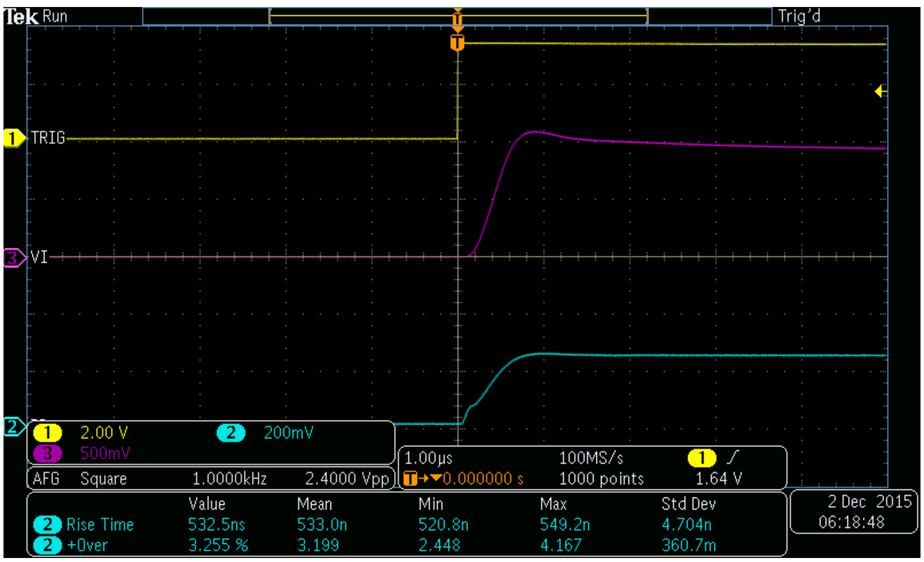


Figure 5: Zoomed traces showing waveform 10-90% rise times in optical feedback mode. Optical rise time statistics are shown at the bottom of the image.

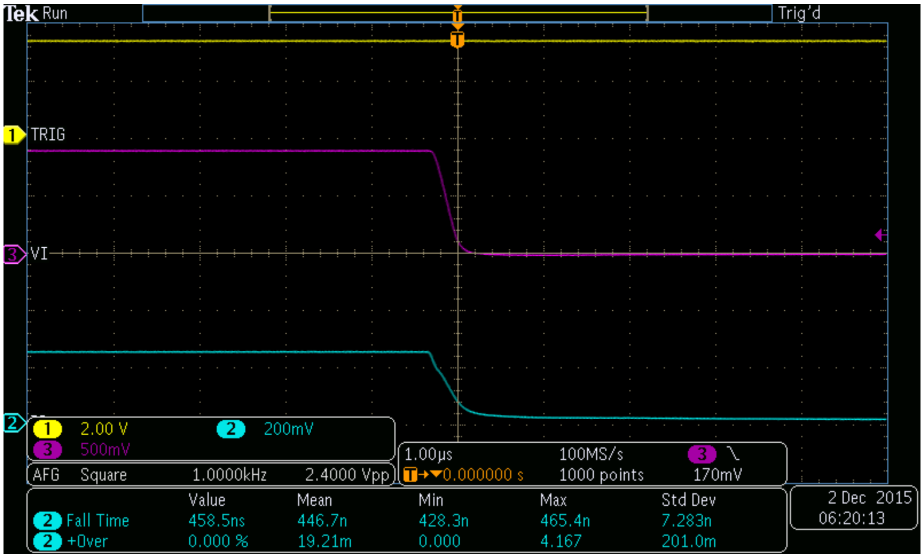


Figure 6: Zoomed traces showing waveform 10-90% fall times in optical feedback mode. Optical fall time statistics are shown at the bottom of the image.

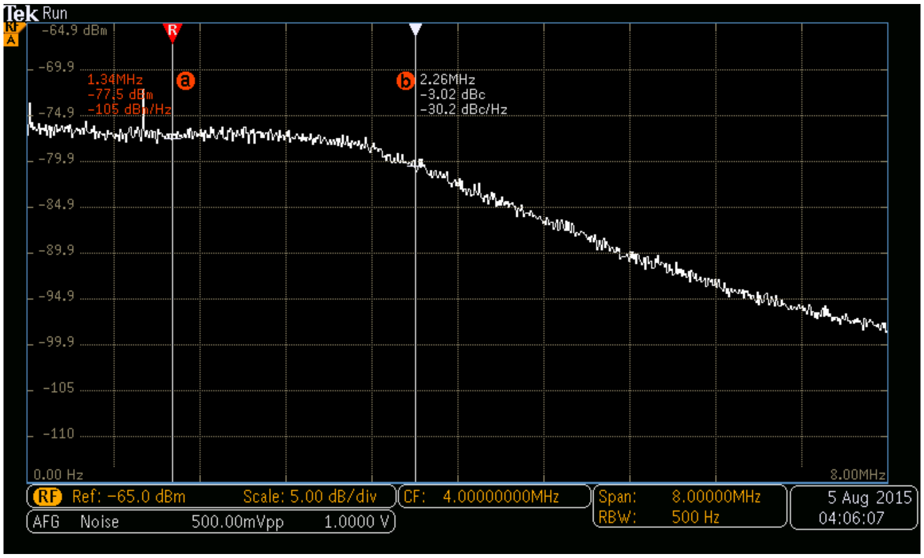


Figure 7: Optical bandwidth in current feedback mode. -3dBm occurs at ~2.4 MHz. Input signal was was noise, flat over 50 MHz, mean = 1.0V, Vpp = 500 mV.

Usage

The cyclops is a device that is capable of transforming voltage signals (e.g. sine waves, square pulses, etc.) into optical signals from high-power LEDs. Voltage signals to drive the device can be generated internally using an on-board digital to analog converter or can be delivered from an external source, such as a function generator or stimulus sequencer. The cyclops provides numerous measurements of circuit operation that can be measured and recorded during an experiment such as LED current and stimulus reference voltages. The device can be controlled over a USB interface using its Arduino library. The device also can be configured to drive commercially available LED modules from Thorlabs and Doric.

Below we provide an explanation of the operational modes of the device and the different ways it can be used to generate optical stimuli. Refer to the above diagram to locate the physical switches, dials, and connectors corresponding to verbal or iconic descriptions device settings.

Feedback modes

Current Feedback Mode

To use current feedback mode, push the `F.B. MODE` slide switch to the `CURR` position (
). Using the circuit in current feedback mode ensures that the forward current across the LED is precisely regulated according the voltage at the `VREF` pin. This configuration is a standard method for driving LEDs because the relationship between current and LED irradiance is smooth and monotonic. This means that more current across the LED will generate more light power (while staying within the LED's maximum ratings, of course). However, the relationship between current and irradiance is not linear. For most LEDs, it looks like a logarithmic function. Additionally, the efficiency of the LED is inversely related to its temperature. So, as the LED operates and heats up, the amount of light it produces drops even when the current is held constant. The severity of an LED's temperature dependence and current/irradiance nonlinearity depend on the type of LED (roughly, the color and who made it). These properties should be clearly documented in the LED's data sheet. With a quality LED and proper thermal management, the effects of temperature and static current/irradiance nonlinearity are fairly minimal and can be ignored in most situations.

Auxiliary Feedback Mode

To use auxilary feedback mode, push the `F.B. MODE` slide switch to the `AUX` position (
). When extremely stable, linear control of light power is required, the auxiliary feedback input can be used to used to compensate for the temperature dependence and static nonlinearity of the current/irradiance

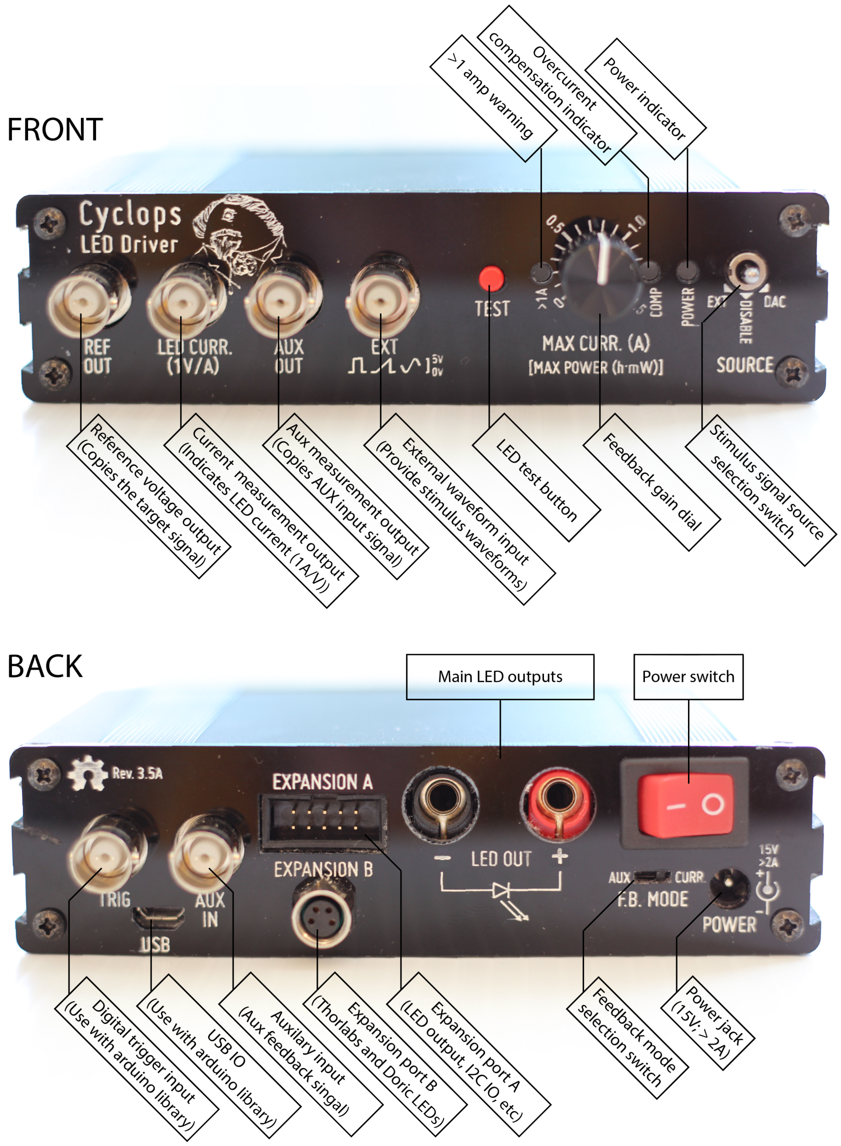


Figure 8: Cyclops physical interface.

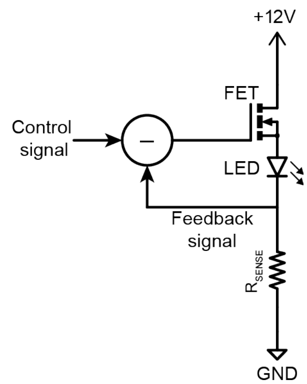


Figure 9: Current feedback configuration.

relationship of the LED. For example, when the auxiliary voltage is supplied by an amplified photodiode that is somewhere indecent to radiation from the LED, or is sampled from the fiber transporting LED light, the gate voltage is adjusted such that the measured light power matches a DAC-supplied reference voltage. This is the case in the circuit diagram. This configuration is referred to as optical feedback mode. The PDA36A adjustable amplified photodiode from Thorlabs is a good option for supplying optical feedback. However, you can make your own amplified photodiode for a fraction of the price, and a design is included within the cyclops repository. Optical feedback completely linearizes the relationship between a supplied reference voltage and the light power produced by the LED by compensating for the current/irradiance nonlinearities and temperature dependence.

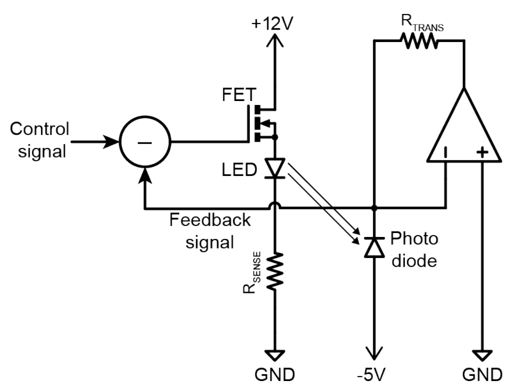
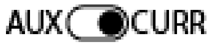
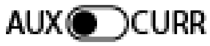


Figure 10: Optical feedback configuration.

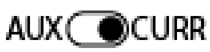
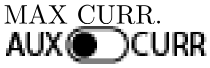
Stimulus Generation Options

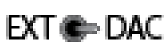
There are three ways to generate light signals using the driver. The behavior of each of these options is dependent on the feedback mode being used. The behavior of each input option is described in relation to the feedback mode of the driver.

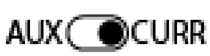
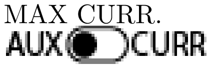
1.  The test button is always available and will override all other input modes. Using the `TEST` button the behavior of the circuit is:

-  Source the current specified by the `MAX CURR.` dial.
-  Generate the optical power specified by the h * mW level that is specified by the `MAX POWER` dial. The intensity of the LED will be dependent on the auxiliary feedback signal used which defines the ‘h’ parameter.

2.  External input mode is engaged when the `SOURCE` switch is moved to the `EXT` position and user supplied voltage waveforms are present at the `EXT` BNC input. If the user attempts to supply more than 5V to the `EXT` input, the circuit will clamp the input signal to 5V. Using `EXT` mode, the behavior of the circuit is:

-  Source the current specified by (EXT Voltage / 5V) * MAX CURR.
-  Generate the optical power specified by (EXT Voltage/5V) * h * mW. The intensity of the LED will be dependent on the auxiliary feedback signal used which defines the ‘h’ parameter.

3.  The internal digital to analog converter (DAC) is engaged when the `SOURCE` switch is moved to the `DAC` position and can be used to generate pre-programmed waveforms and waveform sequences triggered by a digital pulse to the `TRIG` input. This feature relies on optional Arduino installation and programming the device using its API. Using the `DAC` mode, the behavior of the circuit is:

-  Source the current specified by (DAC Voltage / 5V) * MAX CURR.
-  Generate the optical power specified by (DAC Voltage/5V) * h * mW. The intensity of the LED will be dependent on the auxiliary feedback signal used which defines the ‘h’ parameter.

Programming the onboard microcontroller

TODO

Construction

If you have questions during device assembly, please direct them to the open-ephys forum so that others may benefit. Pull requests improving this documentation are welcome.

Components

The bill of materials (BOM) is available on this google doc. Most of the parts can be purchased from Digikey, but there are a few components that need to be bought from other sources such as Newark, Adafruit, and Samtec. All vendor, part number, and quantity information is listed on the BOM. If you are having trouble getting a part, check the BOM since there are alternative suppliers listed for some parts. As a convenience, the spreadsheet contains links to a pre-populated Digikey cart (which are optimized for quantity discounts).

The cyclops PCB can be constructed by purchasing from one of the pre-uploaded options:

- OSH Park - made in America, excellent quality. Minimum of 3 boards per order
- Seeed Studio - made in China, very good quality. Minimum of 5 boards per order.

Alternatively, the gerber files can be uploaded to the PCB fabrication service of your choice. The layer of each gerber file is identified by its file extension:

```
*.GKO = board outline
*.GTS = top solder mask
*.GBS = bottom solder mask
*.GTO = top silk screen
*.GBO = bottom silk screen
*.GTL = top copper
*.G2L = inner layer 2 copper
*.G3L = inner layer 3 copper
*.GBL = bottom copper
*.XLN = drill hits and sizes
```

PCB stencils, which are useful for applying solder paste to the boards, can be purchased from a service like OSH stencils using the gerber files located in

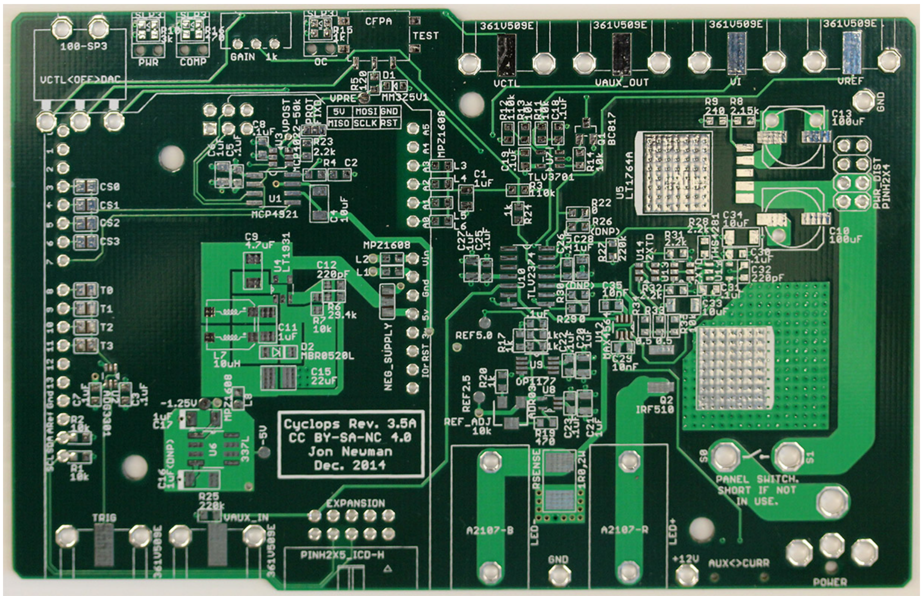


Figure 11: A bare Cyclops PCB, top side, fabricated by Seeed Studio.

./cyclops/stencil/. If you plan to hand solder the board, or don't mind dispensing solder paste yourself, then you do not need to purchase these stencils.

The BOM includes several optional components, which are not in the pre-populated Digikey cart. These include:

- An **extruded aluminum enclosure**, which houses the completed board. The enclosure is recommended because the large voltages and current transients used to drive high power LEDs can cause capacitive and inductive interference with nearby recording equipment. Acrylic front and rear panels can be purchased from Ponoko using the links supplied in the BOM. The instructions below show how these plastic pieces are modified to provide proper electrical shielding.
- An **M8-4 connector**. This is a rather expensive connector that allows cyclops to drive Thorlabs LED modules or Doric LED modules.

Board Assembly

To assemble a Cyclops board, you will need the following materials

- A soldering iron and, if possible, a hot-air reflow device.
  - At minimum, a soldering iron regulated to ~370 deg. c) will do the job.

– In addition to the iron, a hot-air rework tool or reflow oven are recommended and the assembly instructions below assume you are using one of these two options. A low cost, high-quality hot-air rework station can be purchased from SparkFun here.

Figure 12: A soldering iron can be used to assemble the PCB, but a hot air rework station makes things much easier. These can be purchased from Sparkfun.

- Copper braid ('solder wick') for solder removal (e.g this)
- Liquid flux (`no-clean` variants are easiest since they don't have to be thoroughly removed after use)
- Solder paste (e.g. this)
- Stereoscope or loupe (optional but nice for tracking down shorts.)
- Isopropyl alcohol for cleaning flux off the board (e.g. this; optional)
- An anti-static mat (e.g. this; optional but recommended to protect your work. . .)

PCB component population and soldering is fairly straightforward and requires standard surface mount construction techniques.

- A tutorial on hot-air soldering can be found here.

Figure 13: Wire solder and an soldering iron can be used to construct the PCB, but solder paste combined with a hot air rework station or a reflow oven makes things much easier. We use Chipquik 291ax10.

- A great tutorial filled with general tips and tricks for surface mount soldering can be found here.

The following steps provide a visual guide to construct your own board. The goal is to create a fully populated PCB like this one:

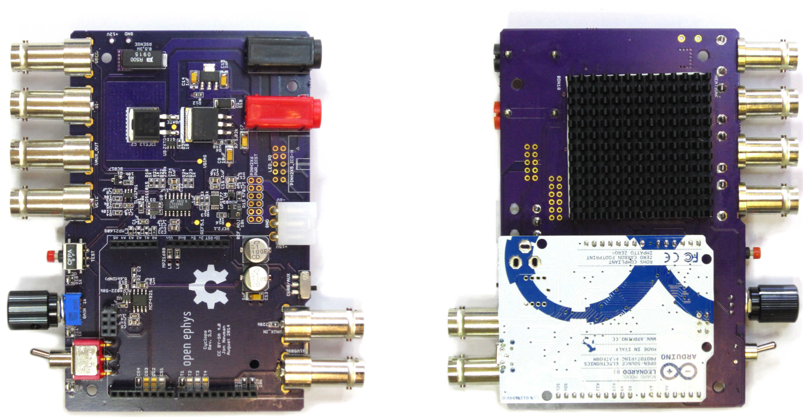


Figure 14: Finished device (revision 3.3).

Following board construction, you should run through the electrical tests outlined in the next section before applying power.

1. Place the bare PCB on a flat surface, preferably one that is static dissipative or anti-static. Alternatively, the board can be mounted in a PCB vice.
2. The silkscreen layer on the PCB (white text) has almost all the information you will need to properly populated the PCB. However, its a good ideal to

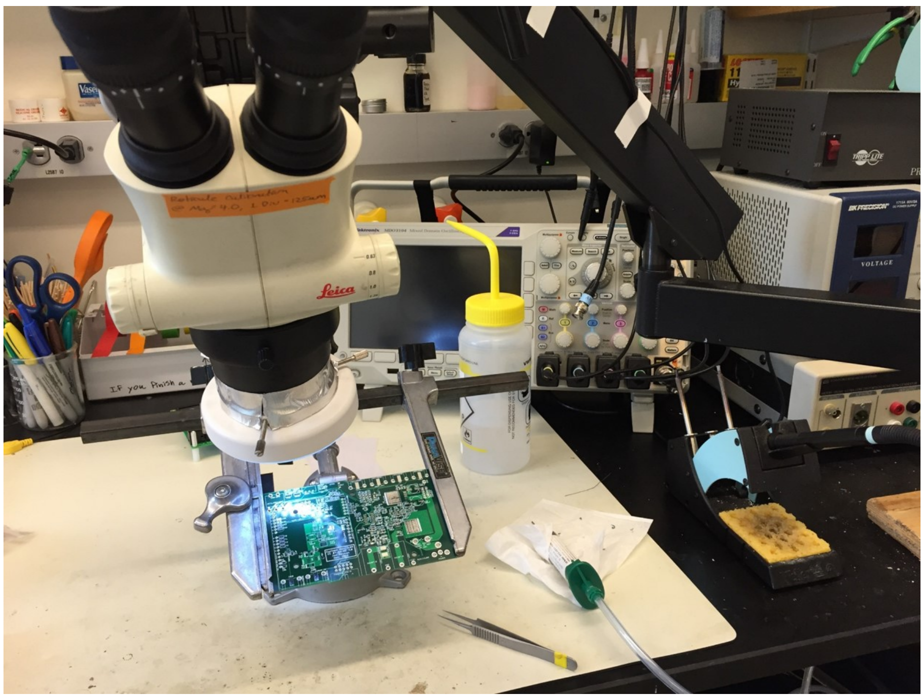


Figure 15: Instead of populating components on a table, holding the PCB using a PanaVise can be helpful.

to open the cyclops design in EAGLE. This will allow you to get detailed information on components before placing them on the board.

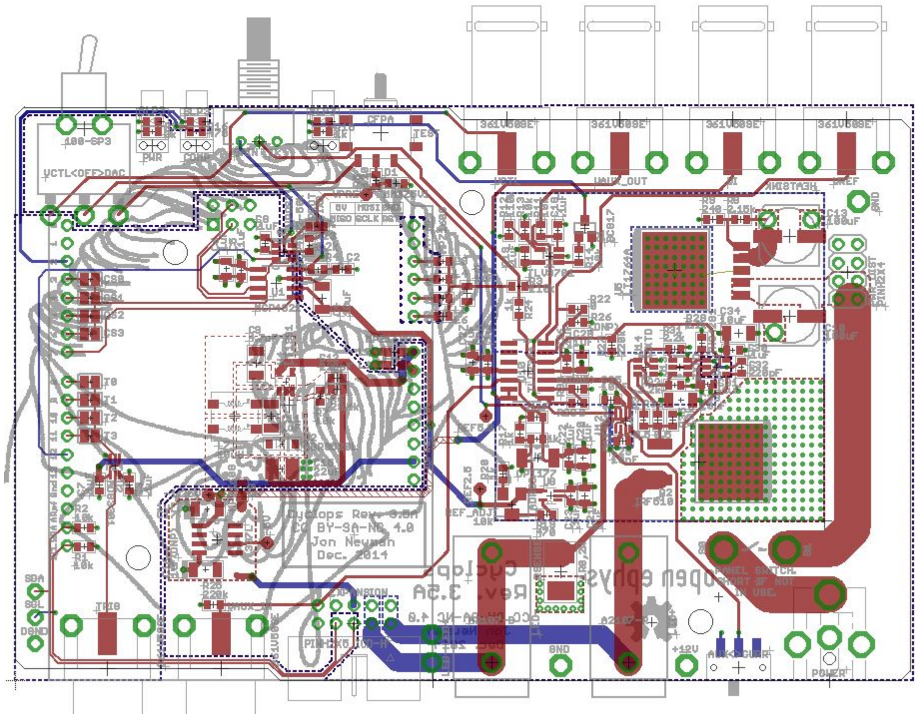


Figure 16: The cyclops PCB design in CadSoft EAGLE

You can then the `information` tool to get detailed information on each component, e.g. to ensure you are placing the correct value resistor or capacitor.

3. After cleaning the surface of the board with isopropyl acholhol or similar, apply solder paste to each of the pads. For an excellent series of tips on effective methods for dispensing solder paste, see Mike's video on the subject. Do not apply solder paste to through-holes or the pads shown outlined in red in the following image. These will be hand soldered later in the assembly process.

   TODO: Image

   The correct amount of solder paste to apply is 'enough'. Each component contact should sit in a small amount of paste, but blobs of paste that envelop the component pad or pin may later result in a short. The following images show examples of good and bad solder placement.

   If you need to pause at any point, you should store place the PCB in the fridge to prevent the flux in the solder paste from breaking down.

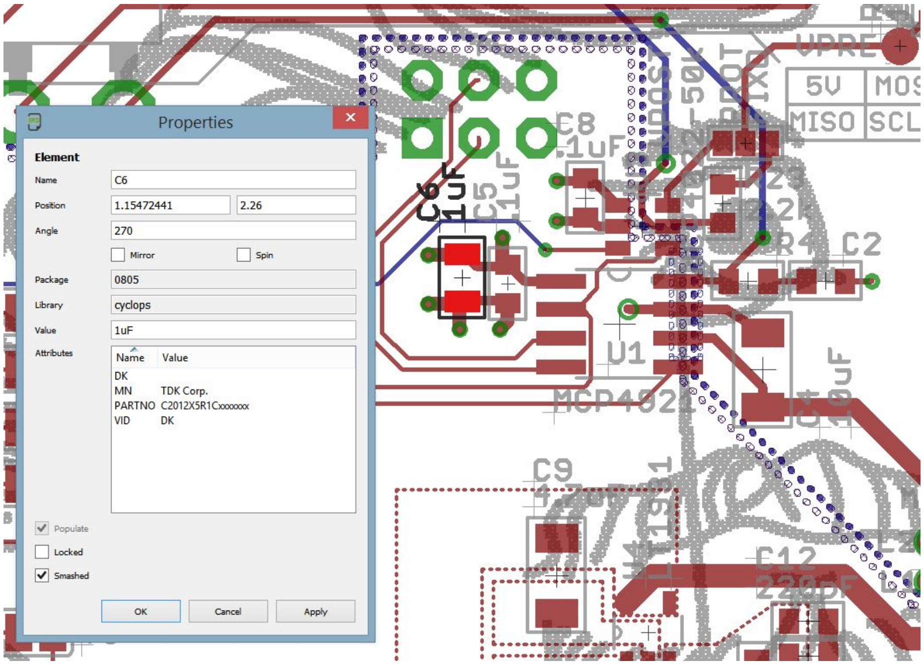


Figure 17: By selecting the information tool and clicking the cross at the center of a component, you can pull up detailed info (e.g. part number)

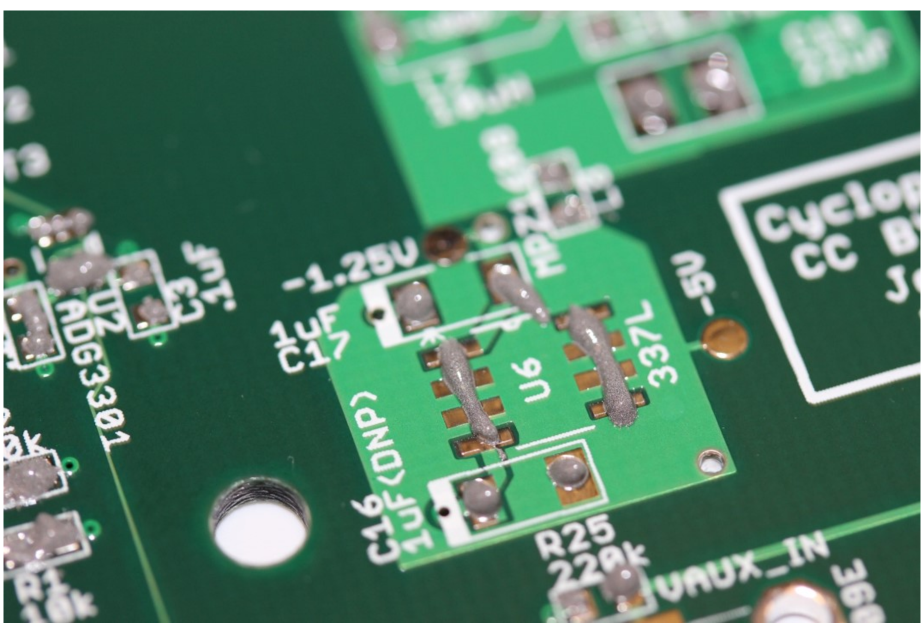


Figure 18: Good solder placement.

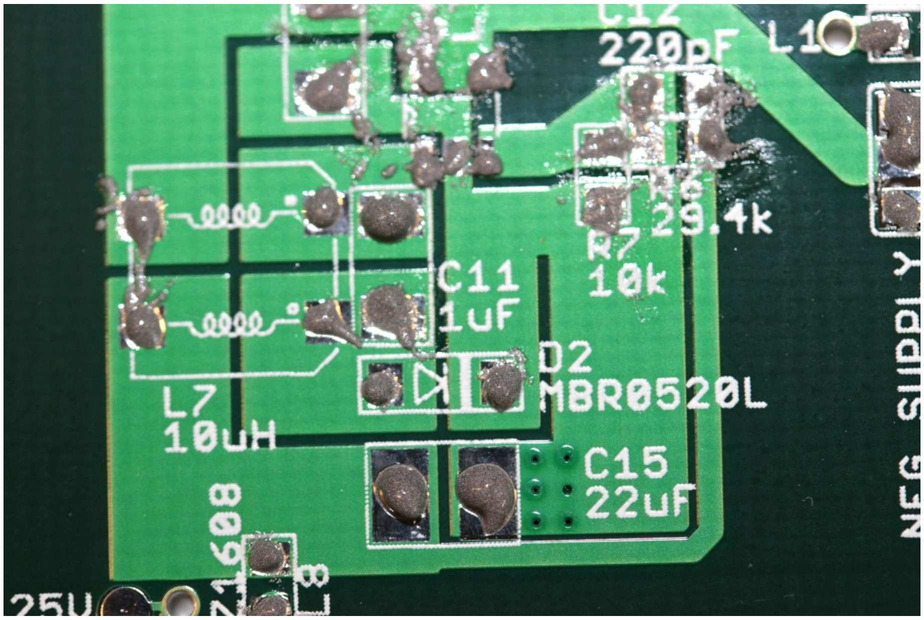


Figure 19: Bad solder placement. Too much paste!

4. Populate all **top-side surface mount** components on the board. There is a single surface mount switch on the back of the board that will be hand soldered later. Additionally, all through hole components (e.g. power jack, BNC connectors, etc) will be populated later. Start by placing the integrated circuits (ICs). Use the stereoscope or loupe to ensure that pads are making contact with the pins of the placed components. Precise component alignment is not necessary. Components will self-align during the reflow process.

5. After placing the ICs, place the passive components (resistors, capacitors, inductors, diodes, and ferrite chips).

6. Next, reflow solder the board. We use a homemade reflow oven constructed from a toaster oven, Arduino board, reflow oven control shield, and mains relay. You can make a similar one, use a commercial reflow oven, or use the hot air station. Reflow the solder paste on the board using your oven or hot air gun as described in the links above.

7. After the solder has cooled, examine solder pads using the stereoscope of loupe for solder bridges between pins, solder that has not melted, or pads lacking a decent solder joint. Fix any issues using a standard soldering iron If there are solder bridges present, get rid of them using some solder wick before moving on. Solder through-hole components in place using a standard soldering iron. A low cost reflow oven can be made form a

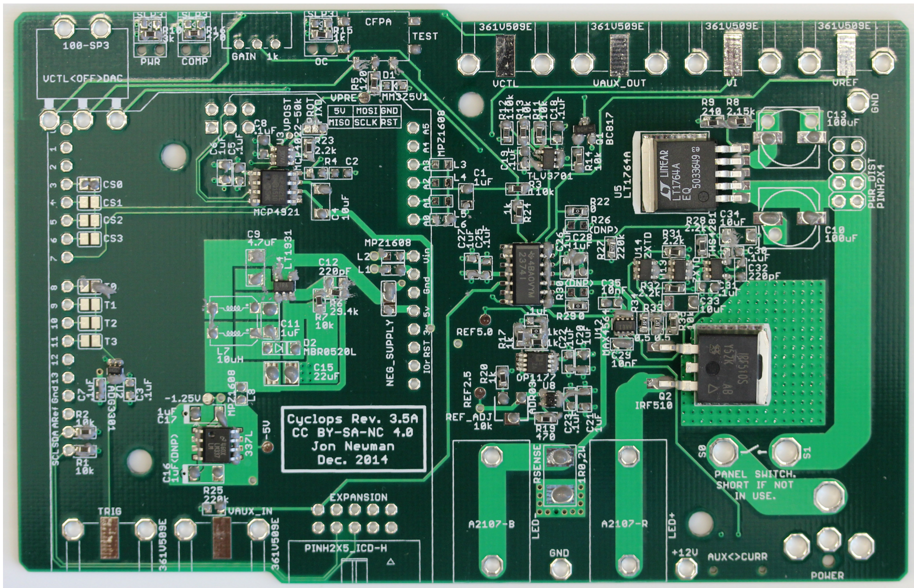


Figure 20: Ingegrated circuit population.

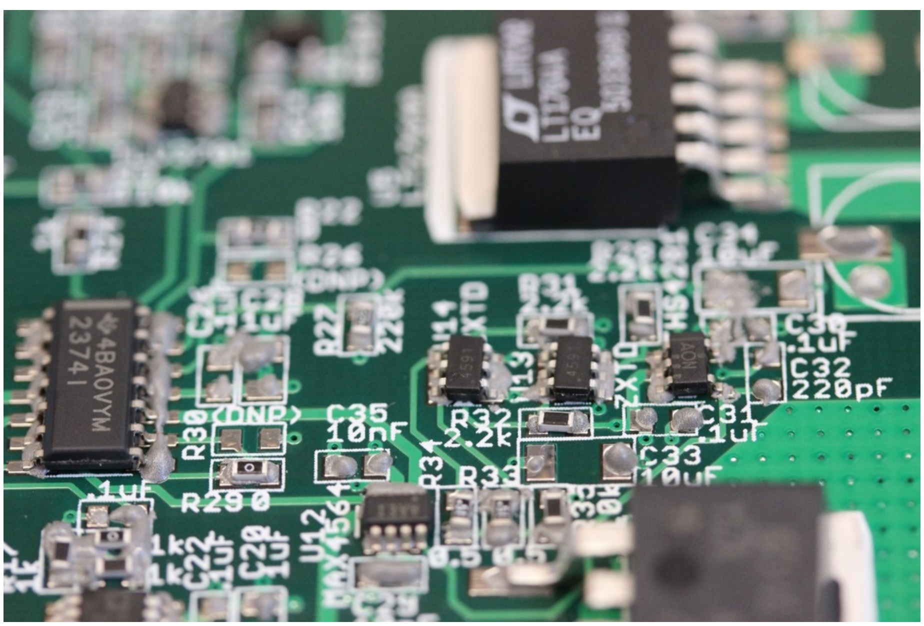


Figure 21: Zoomed view of ingegrated circuit placement.

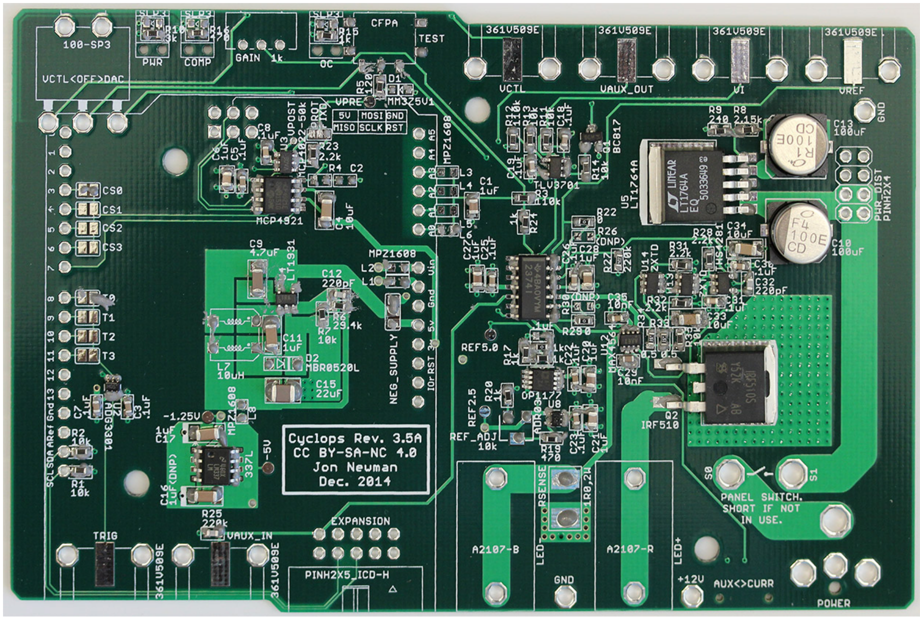


Figure 22: Board following resistor and capacitor population.

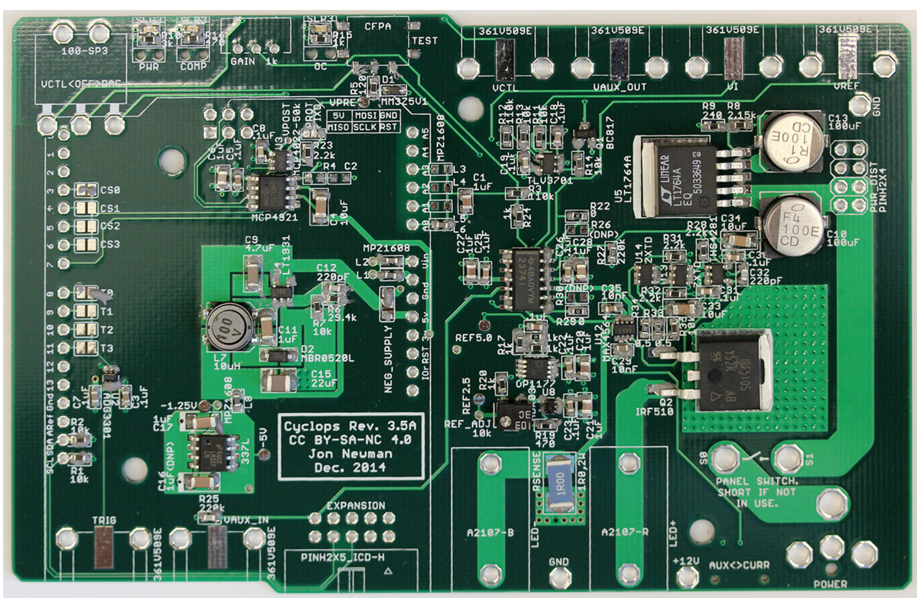


Figure 23: Board with all top-side surface mount components installed.

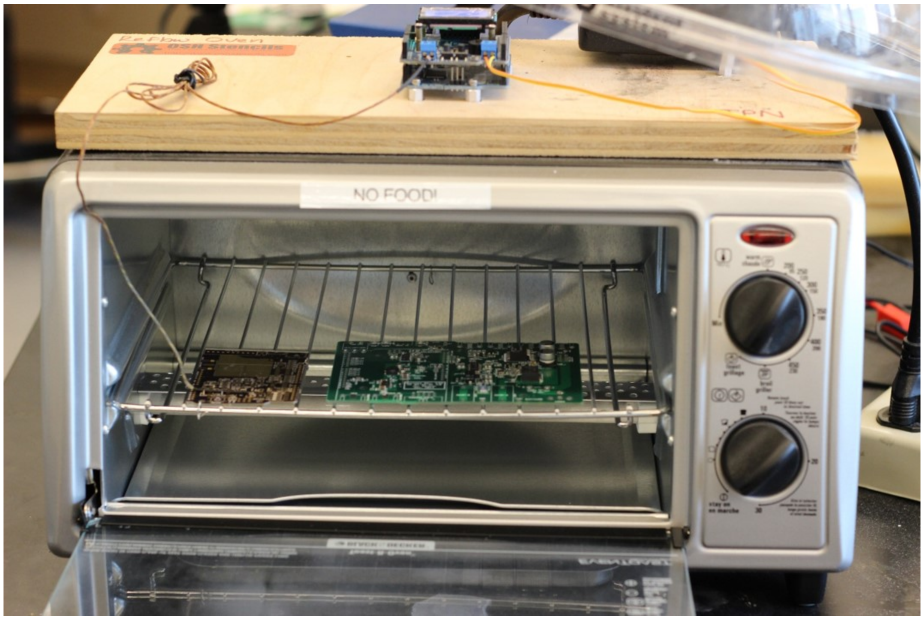


Figure 24: Homemade reflow oven with the populated board inside.

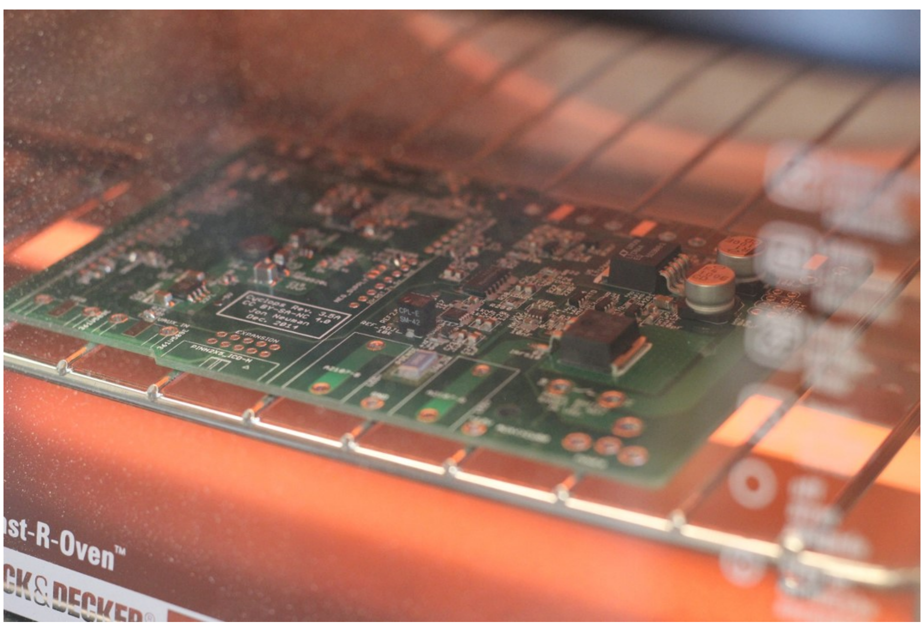


Figure 25: The board is shown after the reflow temperature has been reached. Reflow will occur at different temperatures depending on the specification of the solder paste you are using.

toaster oven as shown here. This link also contains useful information on the basics of the reflow soldering process,

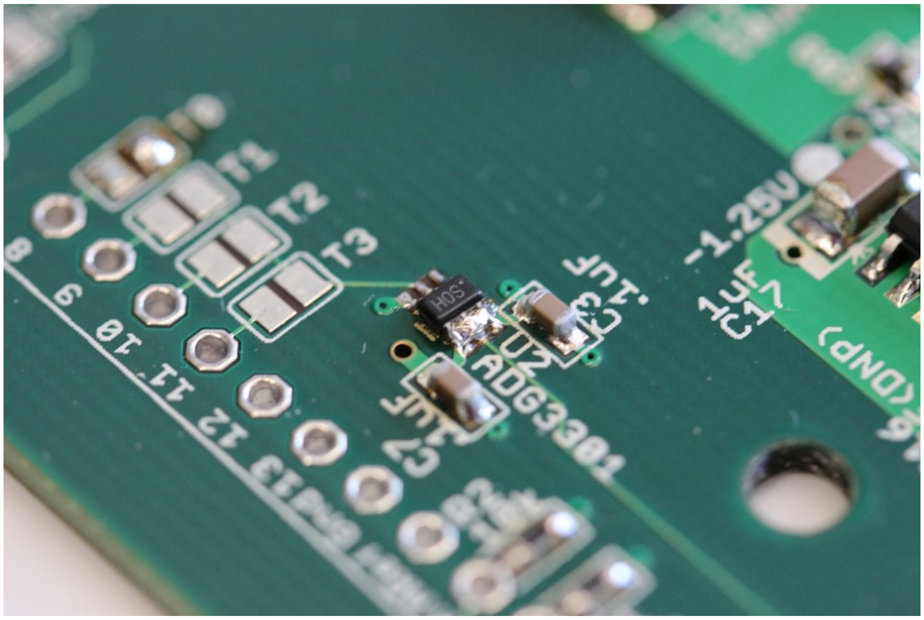


Figure 26: Example of a short between IC pins. This must be resolved before moving forward.

8. Each board has an address (0 through 3) that is defined by two solder jumpers and the location of a ferrite chip. This allows cyclops boards to be stacked to share a power supply while being driven by a common microcontroller. For each board that will share a microcontroller, a unique address must be specified and the solder jumpers and ferrite chip must be soldered in appropriate positions to reflect this address. See the picture below to better understand this addressing scheme.

9. Flip the board over and install the final surface mount component, the `AUX<>CURR` slide switch, by hand soldering.

10. Next, populate all electromechanical components. This can be soldered in place with a standard soldering iron and a large chisel tip.

    Note: The the barrel power jack (name: `POWER`, value: PJ-063BH on the schematic) should be mounted on the **bottom** of the board. It fits on both the top and the bottom, and will properly supply the board with power if mounted on the top. However, if the barrel jack is mounted on the top side of the board, it will not fit inside the enclosure.

11. Install the power switch. You need to use hookup wire capable of handling the currents that the driver requires. AWG 20 (~1.8 mm diameter) braided

Figure 27: Dipping the copper braid in flux will make the solder wick much more redily.

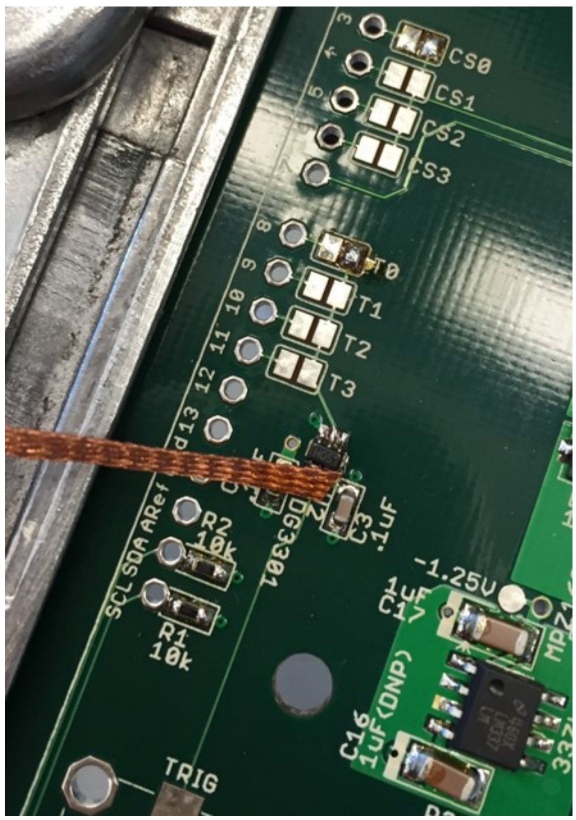


Figure 28: Place the copper braid over the solder blob and then press with the soldering iron. You should see the excess solder wick up the copper braid.

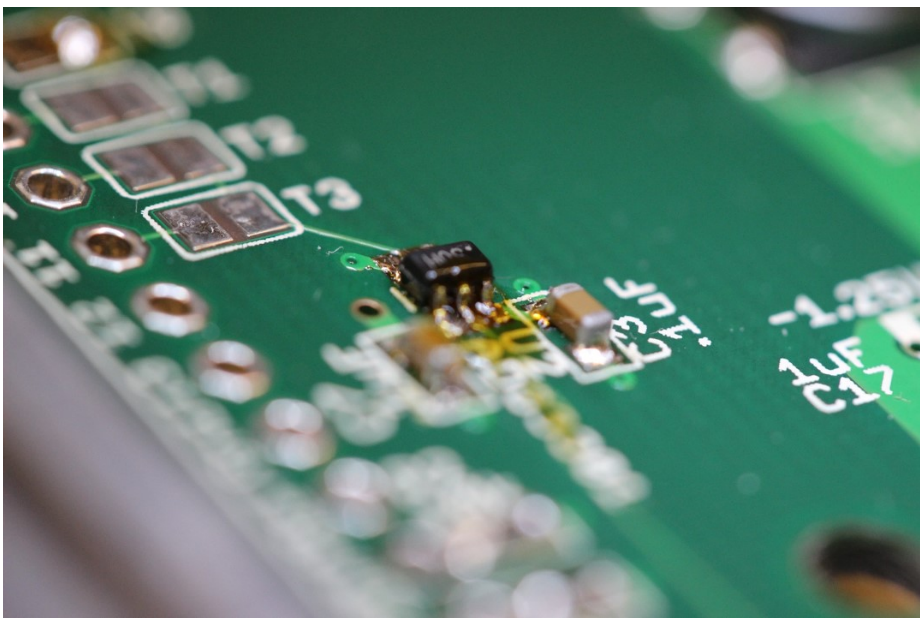


Figure 29: Often you will not have to re-apply solder after this process because there will be enough left over to maintain a good electrical contact. Once you are satisfied the flux residue can be cleaned using isopropyl alcohol

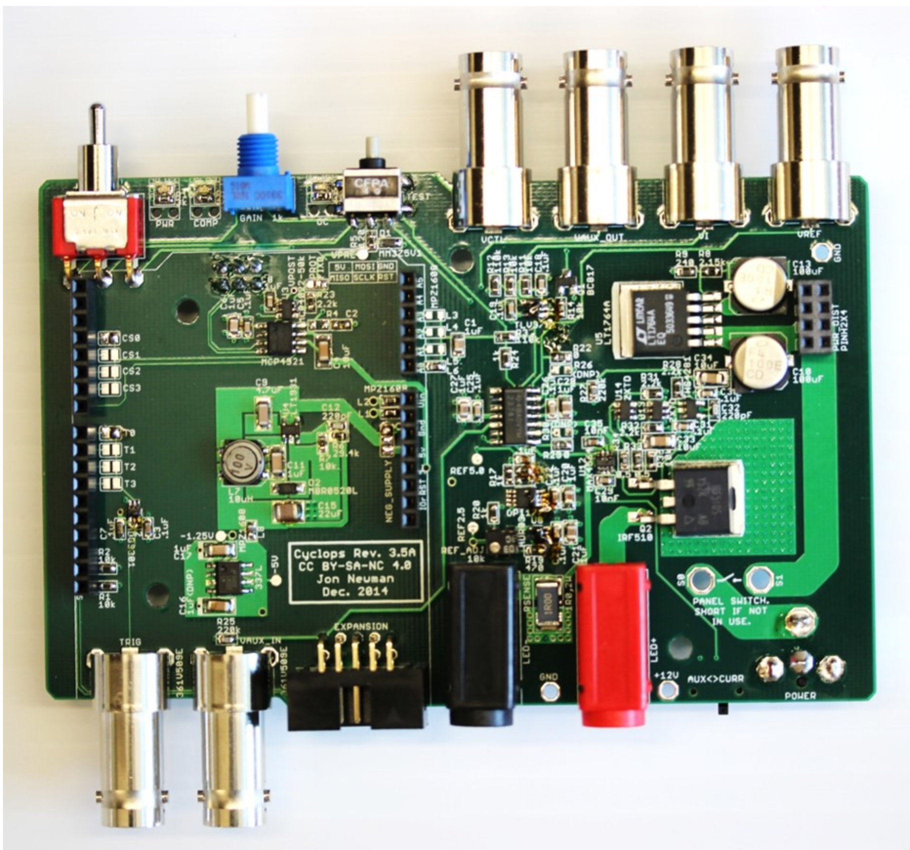


Figure 30: Top side of board following electromechanical component installation.

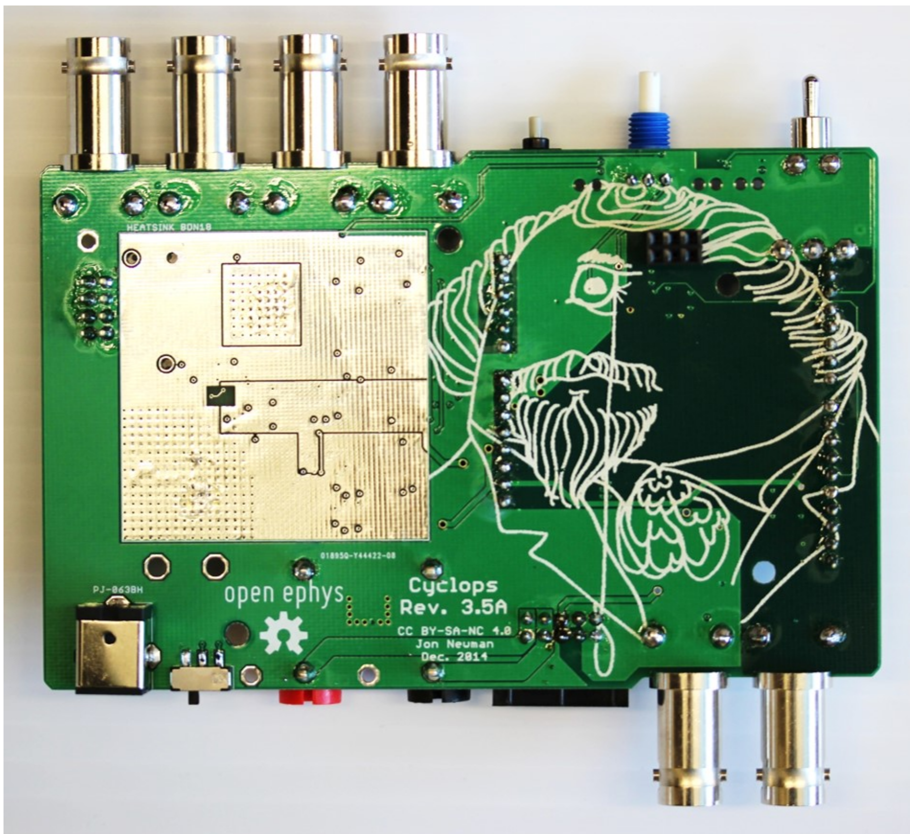


Figure 31: Bottom side of board following electromechanical component installation.

copper wire or thicker is recommended. Use heat-shrink tubing to cover electrical contacts. If you don't want to use the power switch, jumper the switch solder points using AWG 20 wire or thicker.

Figure 32: Power switch components.

12. Install the heatsink.

13. Install the light pipes over the front LEDs. These need to be seated firmly for the board to fit inside the enclosure.

    TODO: Image

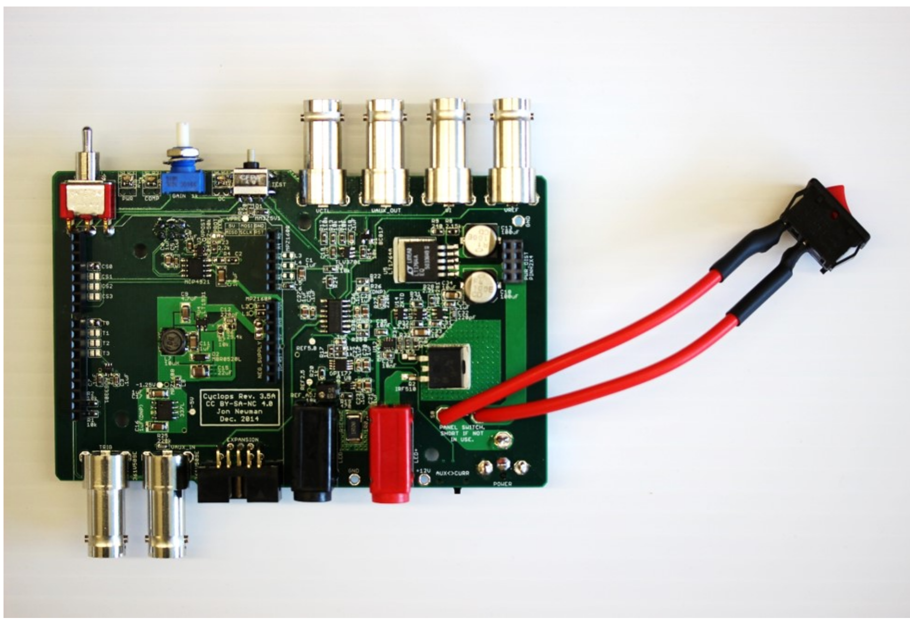


Figure 33: Power switch installation.

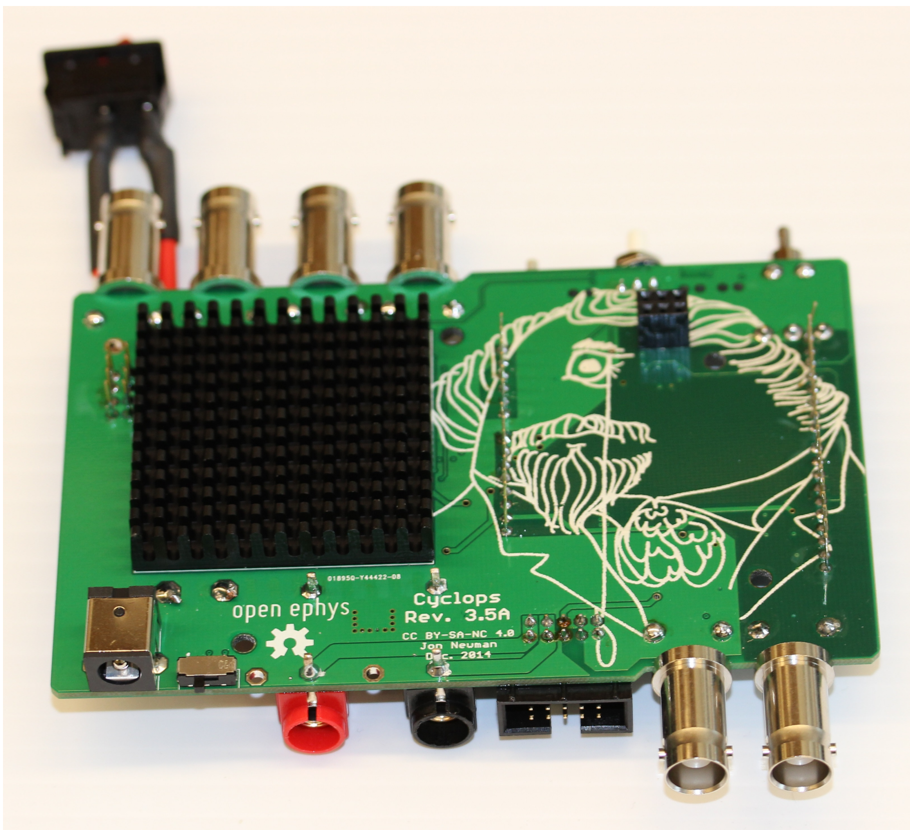


Figure 34: Botton of board with ahesive-backed heatsink in place.

Enclosure

To construct the enclosure, you we will use the following materials

- Phillips head screwdriver (if you are using the enclosure)
- A white paint pen (e.g. these)
- Conductive coating for EMI suppression (e.g. this).

Circuit testing

To perform basic electrical testing, you we will use the following materials

- Digital multimeter (DMM). A low cost mulitmeter is available from sparkfun.
- Jeweler's flat head screwdriver.
- Oscilloscope (optional, but recommended for performance verification)

1. Before powering on the device, check for shorts between power traces on the board. Put your DMM in continuity mode and check for shorts between GND and the various DC voltage supplies on the board. These include

    - `+12V` Positive analog rail
    - `-5V` Negative analog analog rail 1
    - `-1.25V` Negative analog rail 2
    - `REF5.0` 5V voltage reference
    - `REF2.5` 2.5V voltage reference
    - `5v` Digital rail

    TODO: image

    If there is a short, you must track it down and get rid of it before applying power. If you find a short, test the same contact points on an unpopulated PCB to ensure that it is not due to a PCB fabrication defect. If so, contact your PCB fabricator for a return.

2. Obtain a power supply which can source at least 2 amps at 15 volts. You can use a switching supply, since current sourced to the LED is regulated. The BOM includes a reasonably priced option that is capable of powering a single device. Plug this power supply into the barrel jack and turn the power switch on. You should see the power LED illuminate.

3. Put your DMM in DC voltage measurement mode. Touch the negative probe to the GND test point and measure the voltage on each DC voltage supply. They should have the following approximate values:

    - `+12V` 12 volts
    - `-5V` -5 volts
    - `-1.25V` -1.25 volts
    - `REF5.0` 5.0 volts
    - `REF2.5` 2.5 volts
    - `5v` 5 volts

4. While measuring the `REF2.5` testpoint, use the jeweler's screwdriver to turn the `REF_ADJ` trimpot until it reads exactly 2.50 volts. `REF2.5` provides an internal reference voltage for the TEST switch. It serves the purpose of the `VCTL` signal, but does not require an external source. It also provides the reference voltage for the onboard DAC if that is used.

5. Next, we need to ensure that upon the first test of our LED driver, we will not accidentally source too much current to the LED and destroy it. Ensure the device is set to current feedback mode using the rear panel slide switch. Using your DMM in voltage measurement mode, probe the VREF pin pad on the front BNC connector. Depress the `TEST` button and turn the `GAIN` potentiometer until the voltage measurement reads ~100mV. This indicates that the circuit will attempt to drive 100 mA through an LED attached to the LED port. Obtain a high power LED. Ensure that it can handle the 100 mA current that we are about to supply to it. Connect its anode to LED+ and cathode to LED-, respectively.

6. Use the DMM in voltage measurement mode to probe the voltage at the `VI` BNC connector. Depress the `TEST` button. The LED should light up. **Don't look directly at the LED - your eyes' lenses are very good at focusing light to dangerously high levels at your retina**. In current feedback mode, the voltage at the `VI` port reflects the current through the LED with the scale factor of 1V = 1A. Examine the voltage at `VI` port, which should read 100mV, corresponding to 100 mA through the LED. If the LED does not illuminate, ensure that you switched the device to current feedback mode. If the device is left in AUX mode, and there is a high impedance at the AUX BNC connector (e.g. nothing is plugged in), the circuit will appear not to function.

7. Now you are ready to supply time-varying input, from 0-5 volts, to the V_CTL pin to drive the LED or to program the onboard DAC to control the LED output.

License

Hardware Licensing

Software Licensing

References

[1] T. Tchumatchenko*, J.P. Newman*, M.-f. Fong, S.M. Potter. Delivery of time-varying stimuli using ChR2. (* - equal contributions, co-first authors) Front. Neural Circuits (7:184) 2013. doi: 10.3389/fncir.2013.00184