

OpenAPS Cheat Sheet

If you're coming to try OpenAPS from a Loop system, there's going to be some pretty obvious differences. And it starts with "building" the system. **Loop** is built using XCode app on an Apple computer. The brains of the system sit on your iPhone. The communications reside in the RileyLink. **OpenAPS** is built using "script" commands (can be wide variety of computers that are used). The brains and communications of the system reside on a "rig" which acts as a mini-computer.

HUGE DISCLOSURE: For the purposes of this cheat sheet, I'm going to assume a few things and therefore I am not covering all the possible options for setting up your OpenAPS rig. If you want to find out options that exist beyond my assumptions, please refer back to [OpenAPS's documentation](#). In fact, please refer back to there anyways. It really has a lot of information. This document is more of a quick cheat sheet rather than a thorough setup guide.

Assumptions

1. Using an explorer board and Edison
2. Using an Apple computer
3. Using an iPhone
4. Using a Loop-compatible Medtronic pump

Rig parts list

Explorer Board

<https://enhanced-radio-devices.myshopify.com/products/900mhz-explorer-block-pre-order>

Edison

<https://www.sparkfun.com/products/13024>

Nuts and Bolts

<https://www.sparkfun.com/products/13187>

Lithium Battery

(I'd recommend using the larger battery for a little longer battery life.)

<https://www.adafruit.com/products/328> (2500mAh battery)

<https://www.sparkfun.com/products/8483> (2000mAh battery)

Cables

<https://www.adafruit.com/products/592> (3 ft long cable, USB-microB)

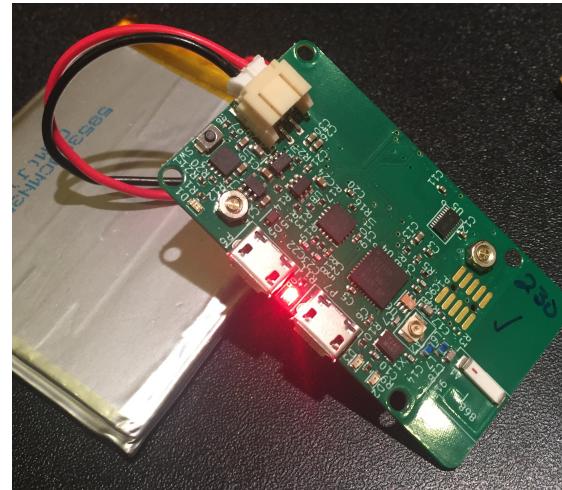
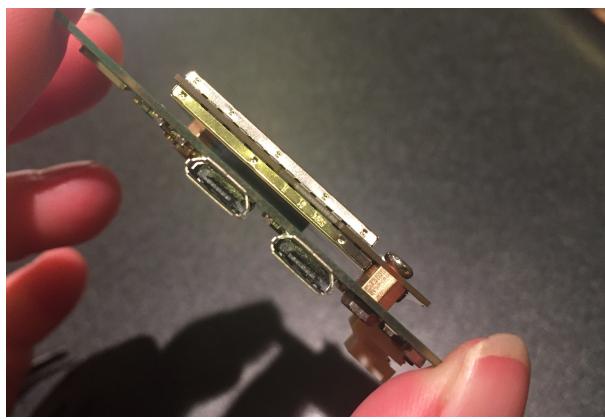
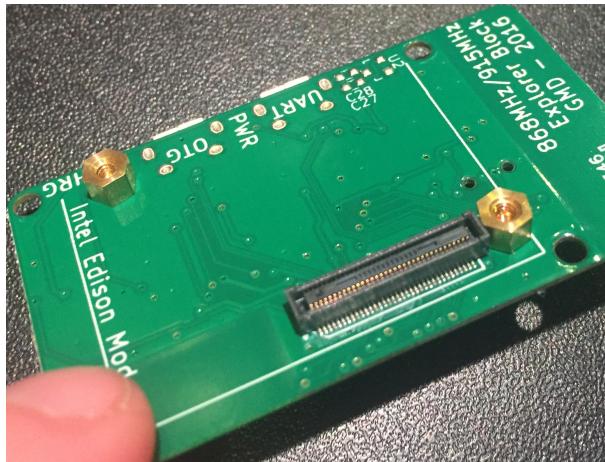
<https://www.adafruit.com/products/898> (6 inch long cable, USB-microB)

Physical-build your rig

The Explorer board is where all the communications are housed for the rig, as well as the battery charger. The Edison is the mini-computer where all the OpenAPS code will be sent and used. In order for this to work, first you have to screw and connect the Edison and Explorer Board together with the nuts and bolts you order.

The nuts and bolts are tiny, and the spaces are a little tight. I find it really helps to use a set of tweezers and a small Phillips head screwdriver.

Easiest to start with the explorer board and put on 2 nuts and gold screws (nuts on the side with most of the wiring). Inside the little outline where the Edison will eventually sit. Gold screws, as shown, with nuts on the backside. Then lay the Edison board on top, aligning the screw holes. Use small Phillips head screw driver to tighten the screws into the gold screws beneath them. Edison board should not wobble, and should feel secure when you are done. Attach your battery into the explorer board plug. A single red light should appear and stay lit.



Software-build your rig

Building the software into your rig is comprised of three steps:

1. preparing the Edison (aka flashing the Edison)
2. installing the “looping” code (aka setup script for oref0)
3. customizing your loop

1. Preparing the Edison

The Edison comes with an older operating system that doesn’t work the best with OpenAPS. The first step is to replace the operating system with a new one. This is called “flashing” the Edison.

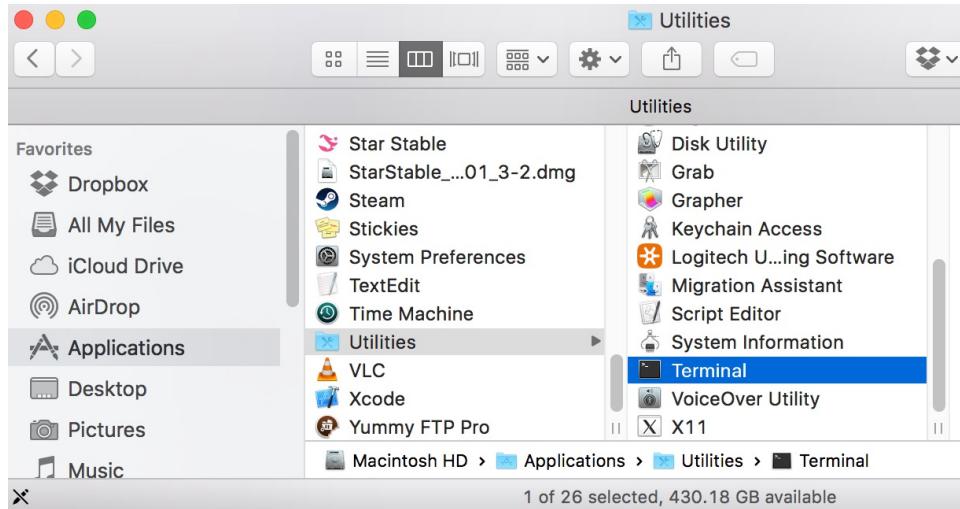
Let’s start by downloading the updated operating system (it’s called Jubilinux) to your computer so that we can install it later onto the Edison. Go to Safari and download [jubilinux.zip](http://www.robinckirkman.com/jubilinux/jubilinux.zip) (<http://www.robinckirkman.com/jubilinux/jubilinux.zip>)

Now we move to the Edison. You’ll see two microB USB ports on your explorer board. One is labeled OTG (that’s for flashing) and one is labeled UART (that’s for logging into the Edison from a computer). We will need both to flash. We’re going to plug both of those into our computer’s USB ports using the cables listed in the parts list (dexcom charging cable will work. too).

Once you plug in the cables, you should see your Edison board in your Finder folder as a connected “device”. If you don’t...try different cables. If your USB port is bad and not recognizing the device, you may need to [reset your SMC first](#) (it’s not hard to do. Takes 2 minutes <http://osxdaily.com/2010/03/24/when-and-how-to-reset-your-mac-system-management-controller-smc/>).

The OpenAPS uses Terminal app, kind of like Loop uses Xcode. It’s our interaction with the code that forms the basis of the loop. You may have never even used the Terminal app. Go to your Applications folder and find the Terminal App in the Utilities folder. Double click to open it.

Terminal app is an ugly interface...but it does what we need to do, communicate with the Edison. Basically, the Edison is a computer that lacks a keyboard and display. By using a cable connected to the rig, we can login to the Edison and use the Terminal app as a way of interacting with the Edison.



When you first launch Terminal, you will probably see something rather plain like below. The important thing to know is that the terminal app helps show you WHERE you are in your computer or Edison. So, in the screen shot below, it's telling me I am in my "iMac4K" user account. If you are ever a little confused where you are...you can look to the left of the \$ prompt and get an idea.

iMac4K — -bash — 80x24

```
Last login: Thu Jan 26 09:56:14 on ttys000
Kathryns-iMac:~ iMac4K$
```

"you are here"

A screenshot of a Mac OS X terminal window titled "iMac4K — -bash — 80x24". The window shows the command line history: "Last login: Thu Jan 26 09:56:14 on ttys000" followed by the prompt "Kathryns-iMac:~ iMac4K\$". A red arrow points from the text "you are here" to the user identifier "Kathryns-iMac:~".

If you're like me, you don't "speak linux" (or python or java or...) nor do you really know what linux is. So, you'll be comforted to know that most of this setup is cut and paste commands into Terminal. You won't need to suddenly learn linux...just will need to follow directions and be willing to learn some basics.