Install Jubilinux, then run through setup, but SKIP Offline section. Run setup script with Bluetooth, then return to Offline and finish BT install.

After setting up the loop to run:

sudo apt-get install -y python-smbus

sudo apt-get install -y i2c-tools

Setup configure\_charger to run at bootup (systemd).

First, install the Riled\_up\_tools using git clone <https://github.com/ps2/riled_up_tools.git>

Then put configure\_charger.py in the systemd file:

Create configure\_charger.service in /etc/systemd/system containing:

[Unit]

Description=Configure Power Supply and Charger

[Service]

Type=oneshot

ExecStart=/root/riled\_up\_tools/charger/configure\_charger.py

[Install]

WantedBy=multi-user.target

Now, setup the file to run:

chmod 644 configure\_charger.service

chown root:root configure\_charger.service

systemctl daemon-reload

systemctl –no-pager list-unit-files configure\_charger.service

systemctl enable configure\_charger.service

systemctl start configure\_charger.service

systemctl status configure\_charger.service

Checking the charger:

CD into /root/riled\_up\_tools/charger and run battery-voltage

then ./get\_charger\_status.py

pip install pyserial

git clone https://github.com/ps2/subg\_rfspy

Then CD into the /subg\_rfspy/tools directory and verify it is working with:

./get\_rfspy\_version.py /dev/ttyMFD1

Install I2C Tools:

git clone <https://github.com/groeck/i2c-tools.git>

cd into i2c-tools and run “make install”

If all is good, then program CC1110:

pip install pyserial \*\*\*probably not needed – already installed by this point

Install ccprog:

CD into /riled\_up\_tools and

git clone <https://github.com/ps2/ccprog.git>

NOTE: In ccprog.c, change define RILED\_UP\_RESET 4// GPIO135 to 47//GPIO49 <- in the future, now is 135

CD into ccprog then type “make”, which will build the program

Then you’ll want to program this firmware:

wget <https://github.com/ps2/subg_rfspy/releases/download/v0.8/uart1_alt2_RILEYLINK_US_STDLOC.hex>

./ccprog write uart1\_alt2\_RILEYLINK\_US\_STDLOC.hex

Then, CD .. and git clone <https://github.com/ps2/subg_rfspy.git> cd into the tools directory, and run:

./get\_rfspy\_version.py /dev/ttyMFD1

It should show:

RileyLink OK

Version: subg\_rfspy 0.9

Response: subg\_rfspy 0.9

Already installed with oref0:

Once you have all that working, next step is to install mmeowlink: <https://github.com/oskarpearson/mmeowlink>

Check that out, and run pip install . in the checked out directory.

Once mmeowlink is installed, you should be able to run:

mmtune.py --port /dev/ttyMFD1 --serial 419886, substituting your pump’s serial #, and you’ll get a read-out of RSSI averages at different frequencies.

Here’s an example output from mmtune: <https://gist.github.com/ps2/83be5fa058a4dabc145b34fa3bac2505>

That was from a couple inches away. With a RL that close, we’ll usually be in the -50 or so range.

No match -> -83dBm

18ohm -> 38pf -77dBm

18ohm -> 33pf -74dBm

Changed to measured distance -76dBm

18ohm -> 27pf -78dBm

1.5ohm -> 27pf -75dBm

0 ohm -> 39pf -75dBm

33nH -> 2.4pF -> 33nH -99dBm

0 ohm -> 33pf -75dbm

2.7nH -> 33pf -72dBm

New reference orientation – Up arrow pointed at RiledUp Antenna

2.7nH -> 33pf -71dBm

2.7nH -> 39pf -71dBm

2.7nH -> 8pf -68dBm (narrower filter response centered at 916.760MHz)  
 1.2pf 22nH

Match above plus:

1.2pF shunt at antenna -69dBm

3.3nF shunt at antenna -69dBm

Resoldered filter, no change.

Taoglas FXUB63 antenna, no match -49dBm

Installing Edison Software

Follow these instructions: <https://github.com/oskarpearson/mmeowlink/wiki/Prepare-the-Edison-for-OpenAPS#troubleshooting>

Make sure the Windows Machine has the swap file settings changed to manual min: 1024 max: 2048.

Use both USB cables and external power supply

It will get stuck in endless reboots, so do this to fix it:

\*\*\* Ready to receive application \*\*\*

U-Boot 2014.04 (Feb 09 2015 - 15:40:31)

Watchdog enabled

DRAM: 980.6 MiB

MMC: tangier\_sdhci: 0

In: serial

Out: serial

Err: serial

Hit any key to stop autoboot: 0

1. Hit any key to drop to a prompt and type:  
   printenv bootargs\_target
2. If the answer is  
   bootargs\_target=first-install  
   then type:  
   setenv bootargs\_target multi-user  
   saveenv
3. And to exit that firmware u-boot prompt:  
   run do\_boot

For setting up WiFi access points for autoselect:

**nano /etc/hosts**

- add the host name you chose to the end of the first line

- old: 127.0.0.1 localhost

- new: 127.0.0.1 localhost FIXME-thehostname-you-want

**nano /etc/network/interfaces**

# interfaces(5) file used by ifup(8) and ifdown(8)

auto lo

iface lo inet loopback

auto usb0

iface usb0 inet static

address 192.168.2.15

netmask 255.255.255.0

auto wlan0

iface wlan0 inet dhcp

allow-hotplug wlan0

iface wlan0 inet dhcp

pre-up wpa\_supplicant -Dwext -i wlan0 -c /etc/wpa\_supplicant/wpa\_supplicant.conf -B

iface default inet dhcp

**nano /etc/wpa\_supplicant/wpa\_supplicant.conf**

ctrl\_interface=DIR=/var/run/wpa\_supplicant GROUP=netdev

network={

ssid="Eldorado"

psk="minicorndogs1970"

}

network={

ssid="ISISMindControlStation27"

psk="ILoveTrumpsSpeeches"

}

network={

ssid="CIA\_Drone\_17"

psk="Zandra1967"

}

network={

ssid="EUHSD-BAS"

}

**Install packages per document, then:**

**Install SSH (with existing key) From Git Bash on the Windows Machine:**

cat ~/.ssh/id\_rsa.pub | ssh use@123.45.67.89 "cat >> ~/.ssh/authorized\_keys"

Now install OpenAPS per:

<http://openaps.readthedocs.io/en/latest/docs/walkthrough/phase-2/oref0-setup.html>

Loop should be running now. Then install:

sudo apt-get install -y python-smbus

sudo apt-get install -y i2c-tools

pip install pyserial

git clone <https://github.com/ps2/subg_rfspy>

Then verify it is working with:

./get\_rfspy\_version.py /dev/ttyMFD1

It should show:

RileyLink OK

Version: subg\_rfspy 0.8

Response: subg\_rfspy 0.8

install the Riled\_up\_tools using git clone <https://github.com/ps2/riled_up_tools.git>

Then put configure\_charger.py in the systemd file:

Create configure\_charger.service in /etc/systemd/system containing:

[Unit]

Description=Configure Power Supply and Charger

[Service]

Type=oneshot

ExecStart=/home/root/git/riled\_up\_tools/charger/configure\_charger.py

[Install]

WantedBy=multi-user.target

Now, setup the file to run:

chmod 644 configure\_charger.service

chown root:root configure\_charger.service

systemctl daemon-reload

systemctl –no-pager list-unit-files configure\_charger.service

systemctl enable configure\_charger.service

systemctl start configure\_charger.service

systemctl status configure\_charger.service