# μRAT

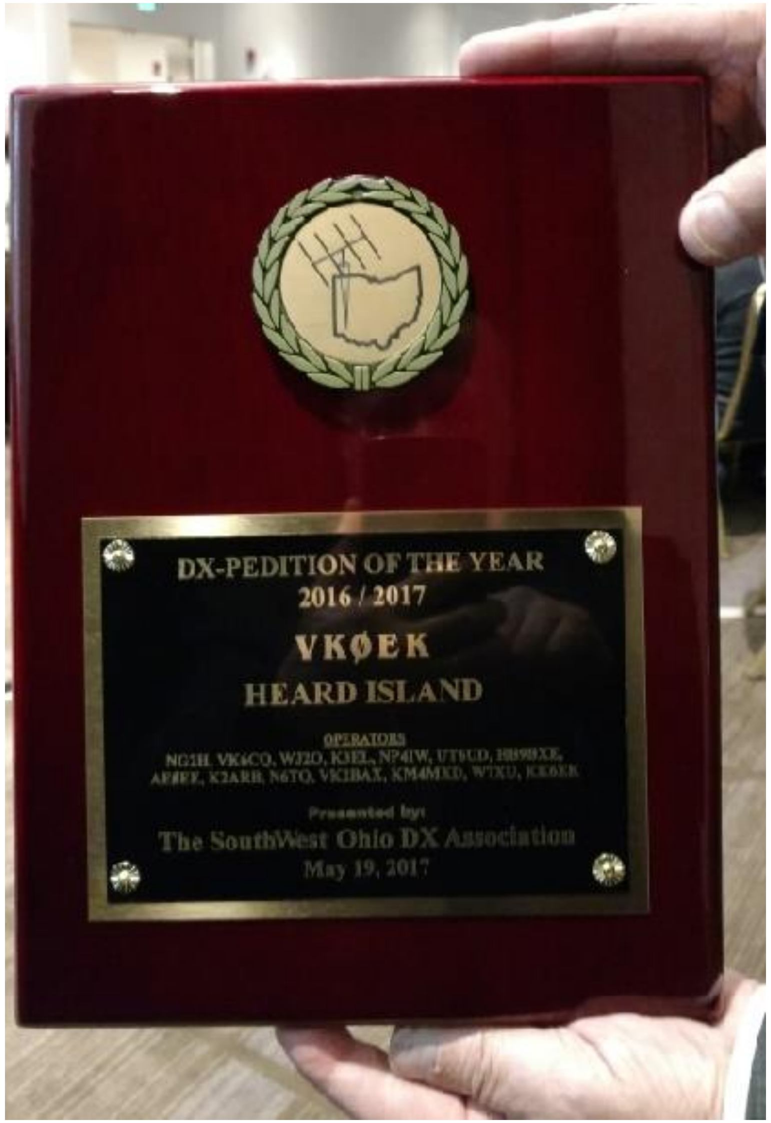
Universal Remote Antenna Tuner

Rich Holoch, KY6R

# KY6R - Background

- First licensed as WN2QHN in Newton, NJ – 1973
- Off air from 1977 – 2001
- Earned DXCC Honor Roll in 11 years – 2013
- 2 away from Top of Honor Roll after 16 years of DX-ing
- 36 years in IT – Staff Data Architect and Data Engineer at Credit Karma in San Francisco

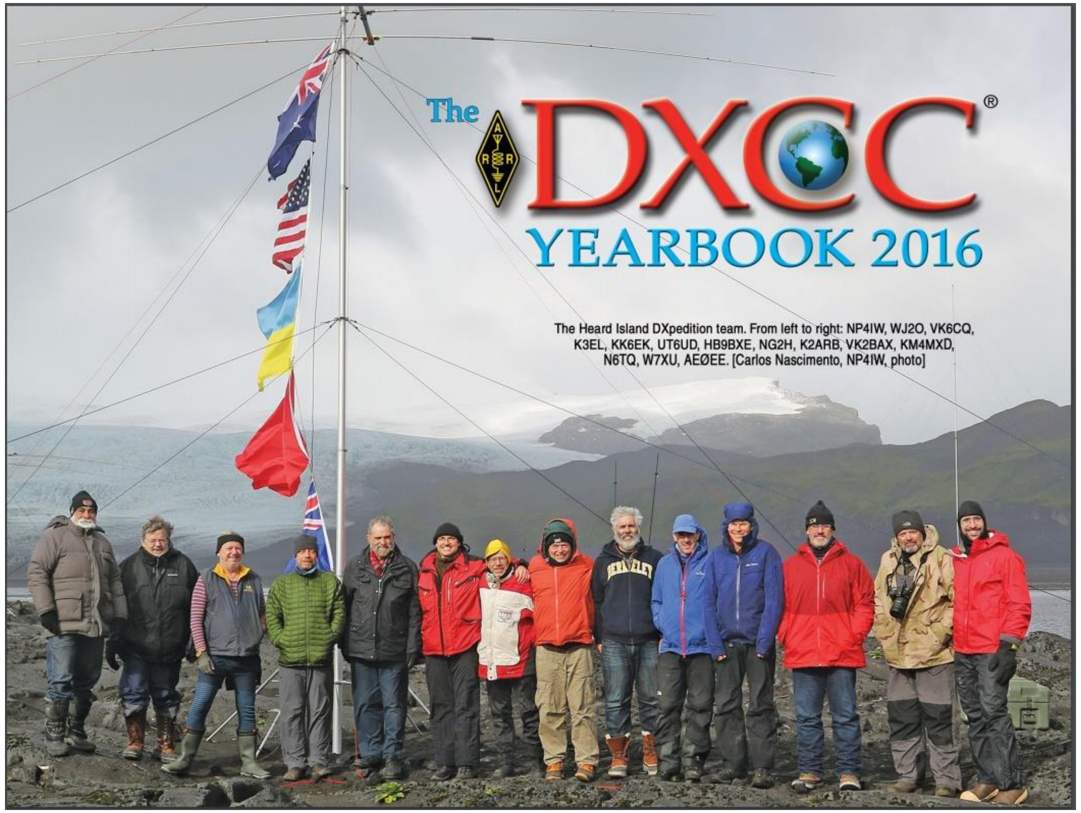
# VK0EK – Heard Island Co-Organizer



DX-PEDITION OF THE YEAR
2016 / 2017
VKØEK
HEARD ISLAND

OPERATORS
NG2H, VK6CQ, WJ2O, K3EL, NP4IW, UT6UD, HB9BXE,
AEØEE, K2ARB, N6TQ, VK2BAX, KM4MXD, W7XU, KK6EK

Presented by:
The SouthWest Ohio DX Association
May 19, 2017



The DXCC®
YEARBOOK 2016

The Heard Island DXpedition team. From left to right: NP4IW, WJ2O, VK6CQ, K3EL, KK6EK, UT6UD, HB9BXE, NG2H, K2ARB, VK2BAX, KM4MXD, N6TQ, W7XU, AEØEE. [Carlos Nascimento, NP4IW, photo]

The January 2016 K5P DXpedition to Palmyra Atoll PAGE 69

QST

DEVOTED ENTIRELY TO AMATEUR RADIO

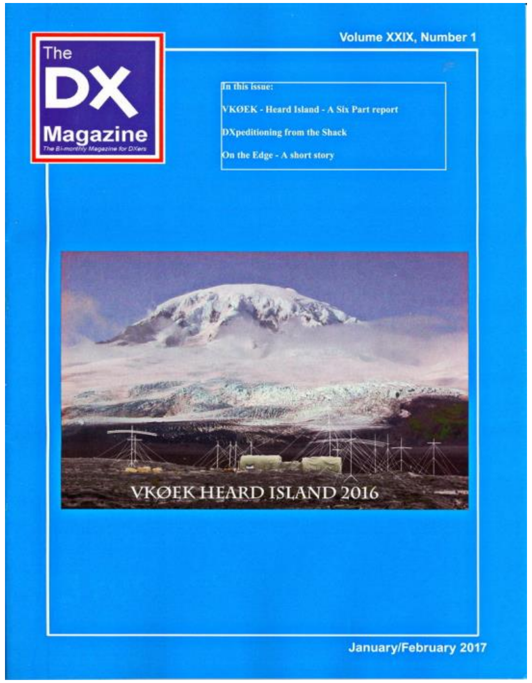
October 2016 WWW.ARRL.ORG

Extreme DX at the Bottom of the World!

*QST* reviews:

- Expert Electronics SunSDR2 PRO HF/VHF Software Defined Transceiver
- SSB Electronic SP-200 and SP-70 Mast-Mounted Preamplifiers
- Alinco DR-735T VHF/UHF FM Transceiver

VKØEK HEARD ISLAND 2016



The DX Magazine
The Bi-monthly Magazine for DXers

Volume XXIX, Number 1

In this issue:

VKØEK - Heard Island - A Six Part report

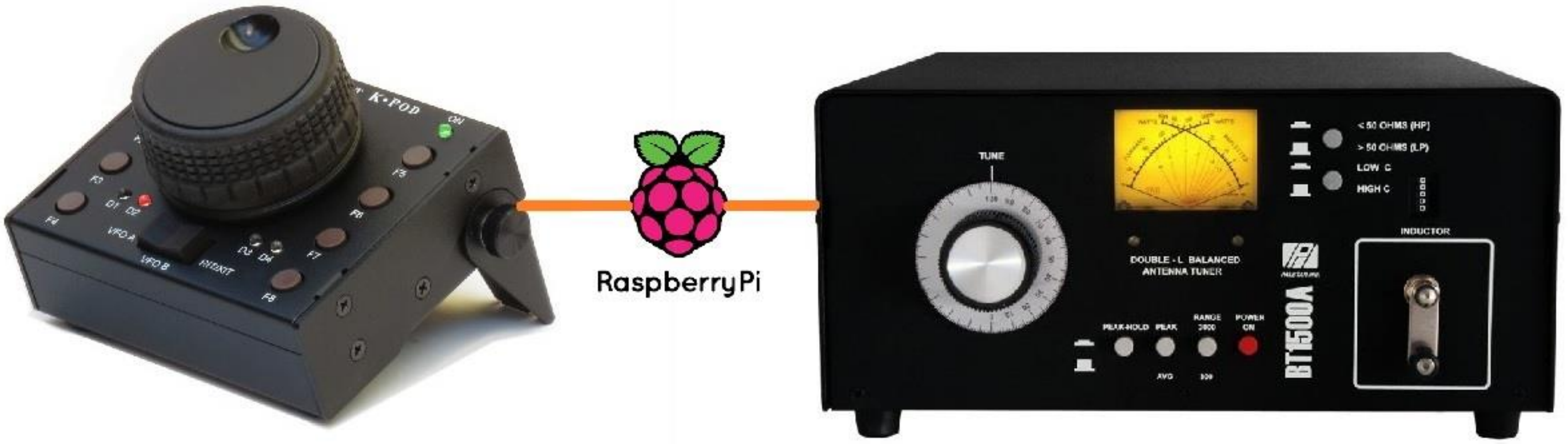
DXpeditioning from the Shack

On the Edge - A short story

VKØEK HEARD ISLAND 2016

January/February 2017

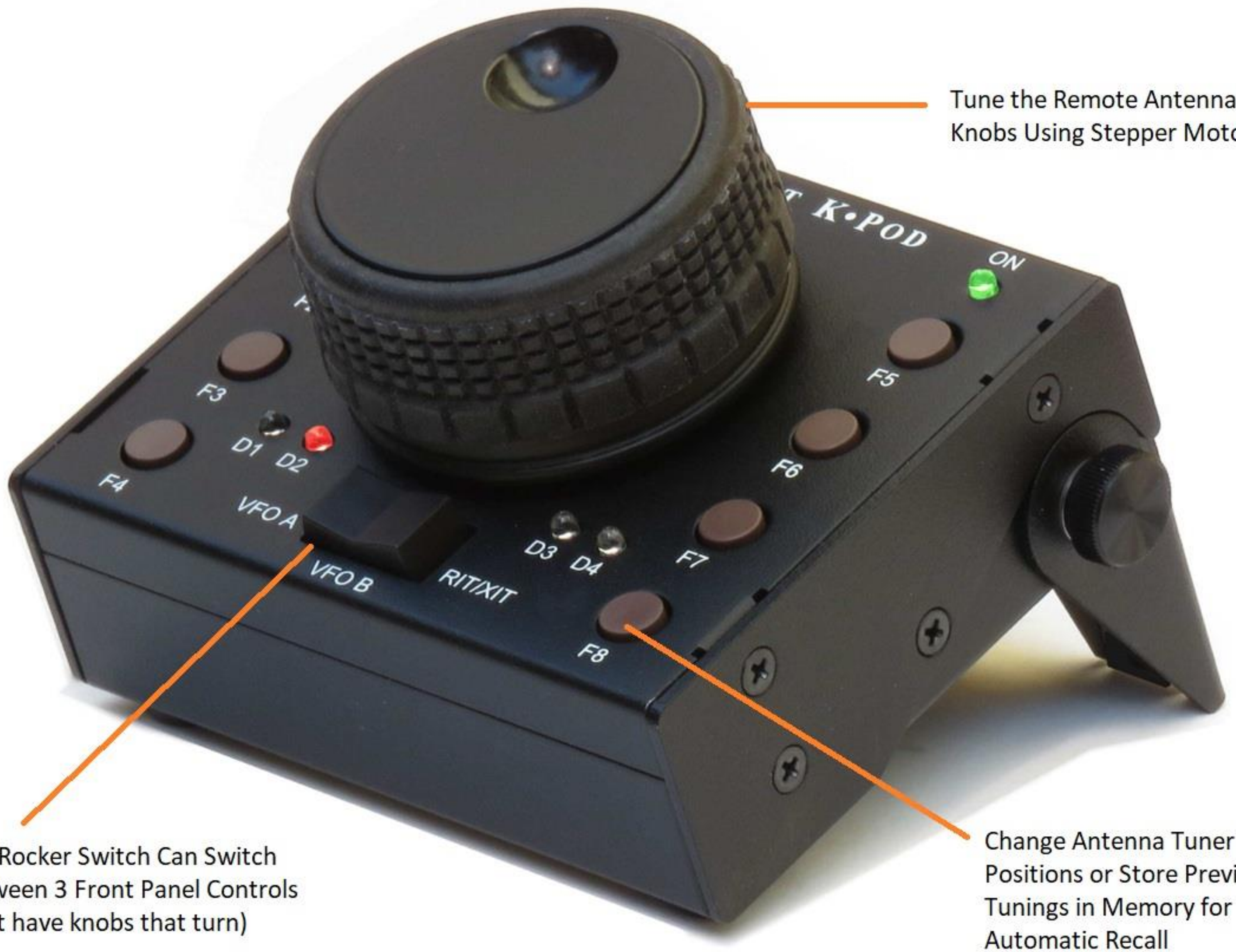
# u.RAT – Ham Meets Maker



Elecraft KPOD

Palstar BT1500A

# The Elecraft KPOD

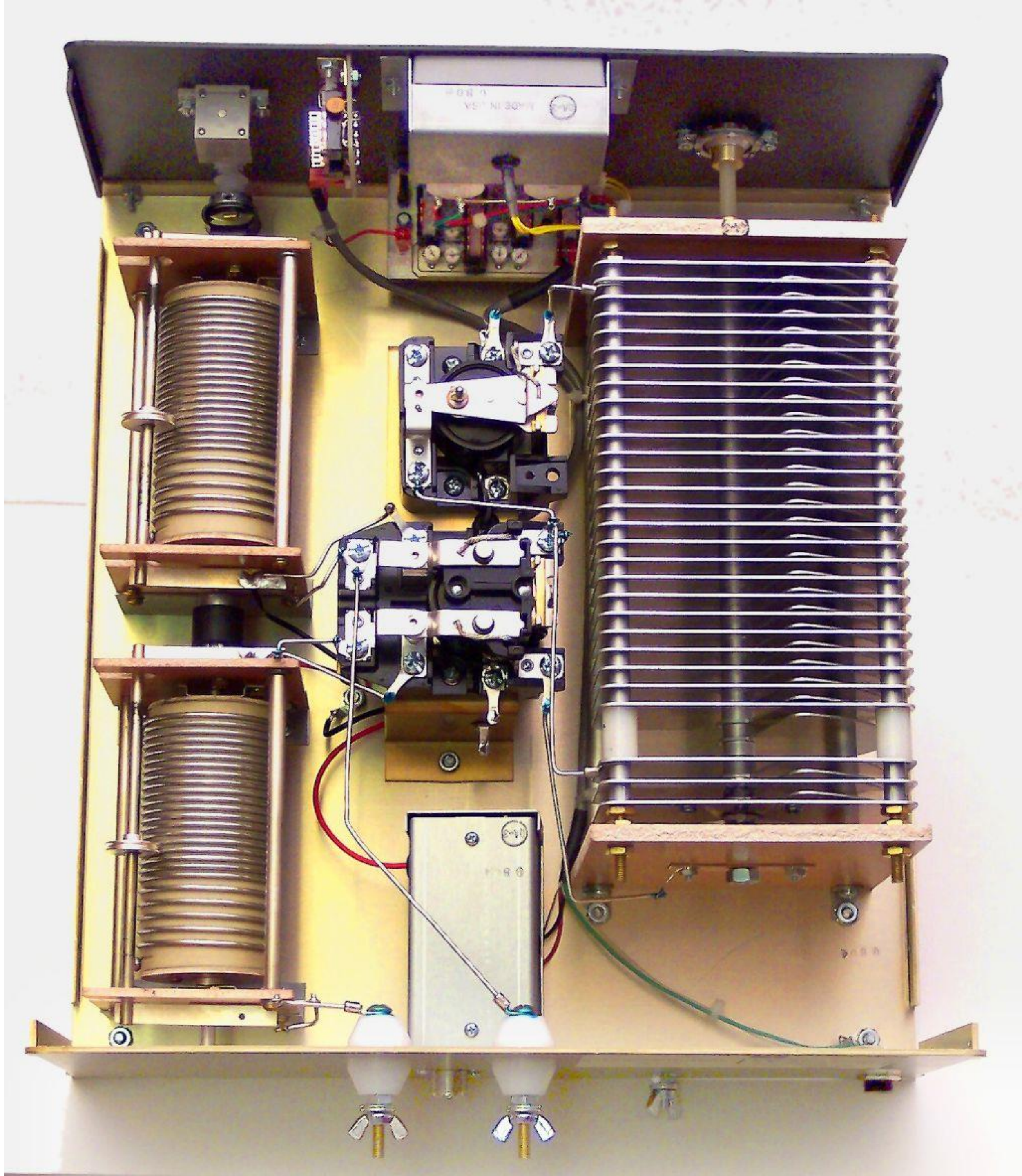


Tune the Remote Antenna Tuner Knobs Using Stepper Motors

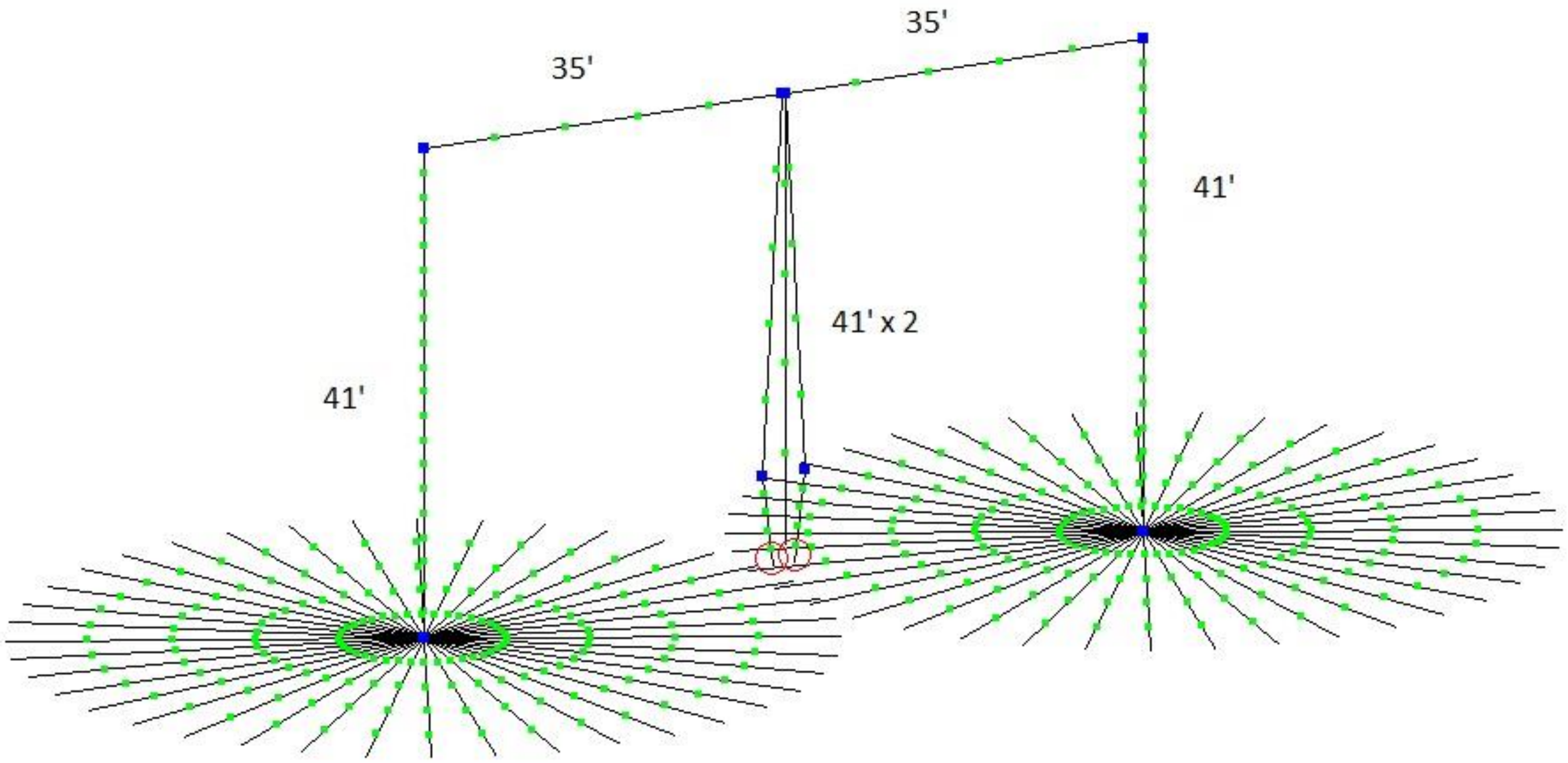
The Rocker Switch Can Switch Between 3 Front Panel Controls (that have knobs that turn)

Change Antenna Tuner Switch Positions or Store Previous Tunings in Memory for Automatic Recall

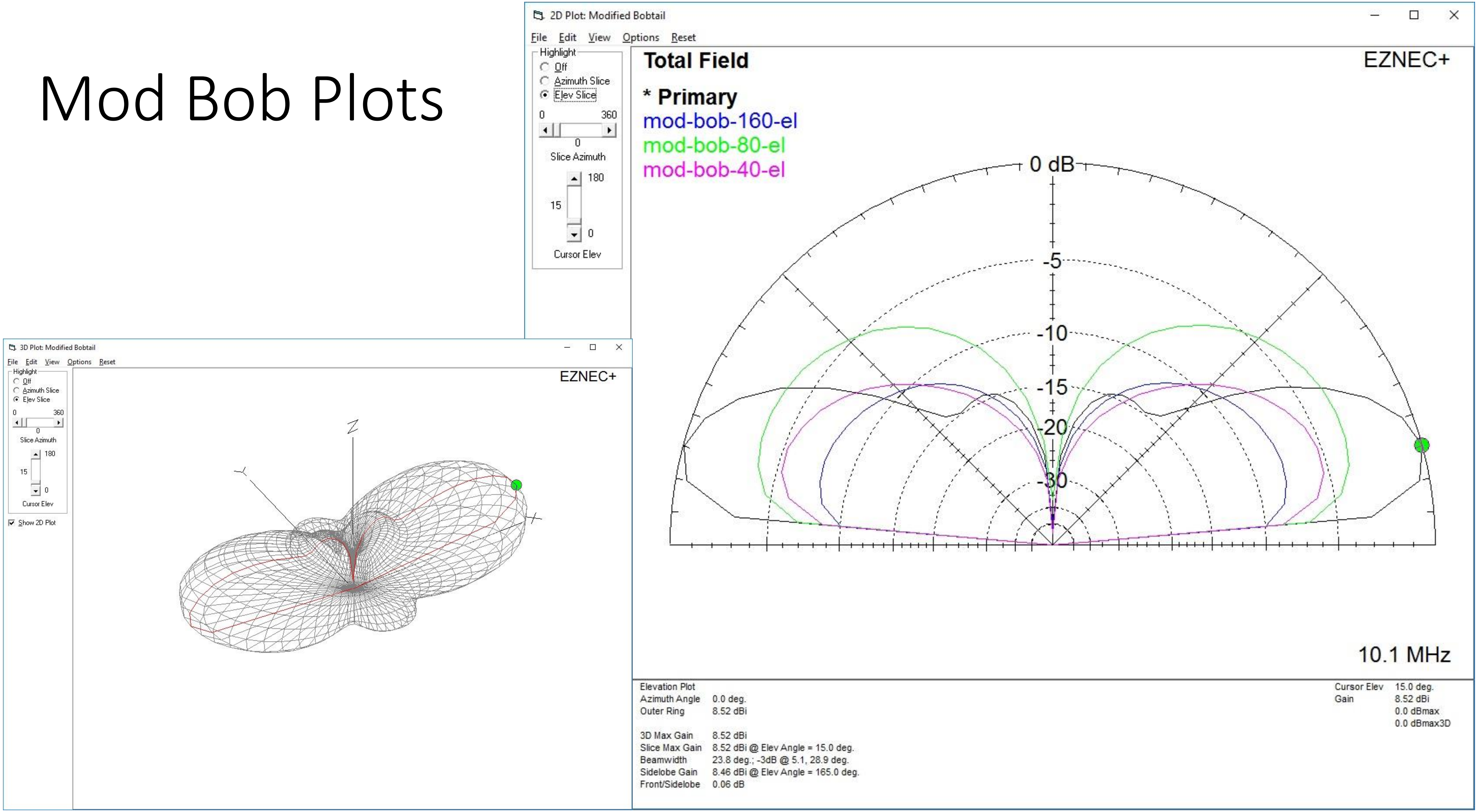
# The Palstar BT1500A



# Mod-Bob Low Band Antenna



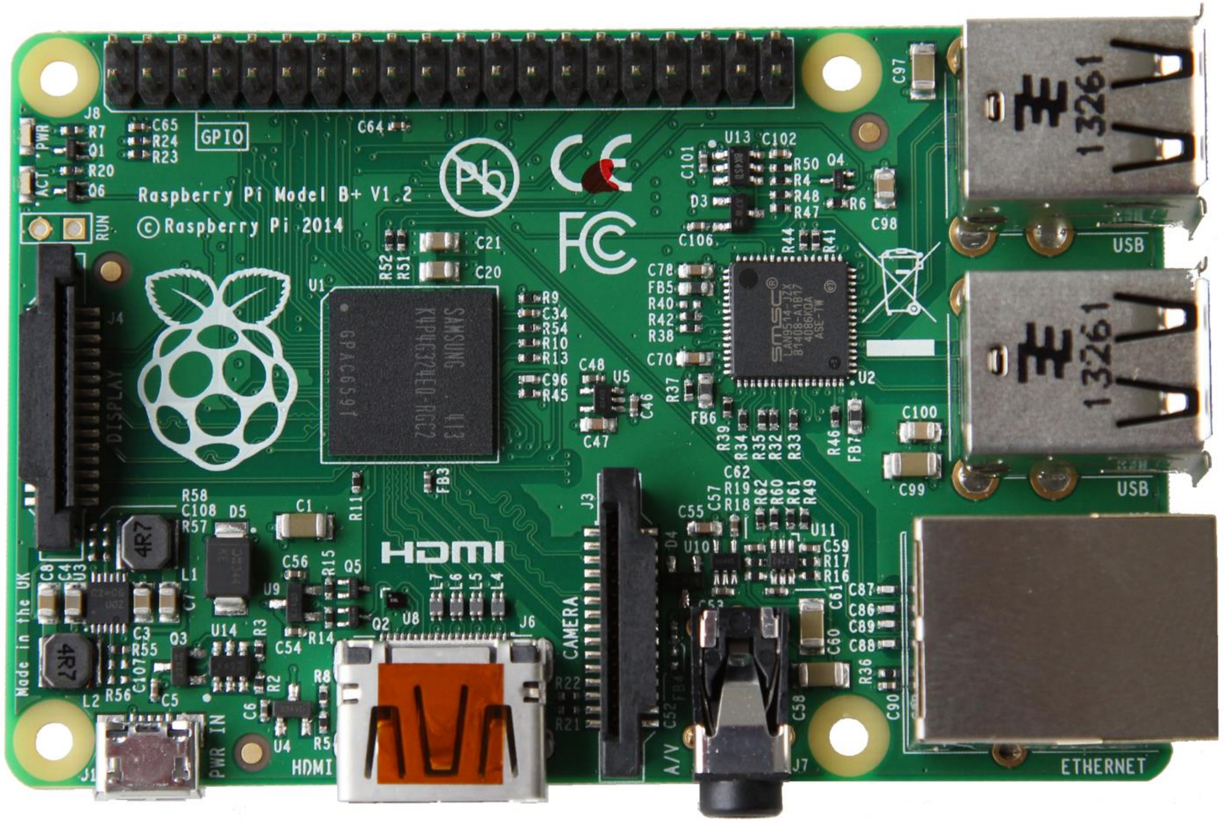
# Mod Bob Plots


3D Plot: Modified Bobtail
File Edit View Options Reset
EZNEC+
Z
Y
X
2D Plot: Modified Bobtail
File Edit View Options Reset
Total Field
EZNEC+
* Primary
mod-bob-160-el
mod-bob-80-el
mod-bob-40-el
0 dB
-5
-10
-15
-20
-30
10.1 MHz

```
Highlight
○ Off
○ Azimuth Slice
● Elev Slice
0          360
     0
Slice Azimuth
      180
15
      0
Cursor Elev
☑ Show 2D Plot
```

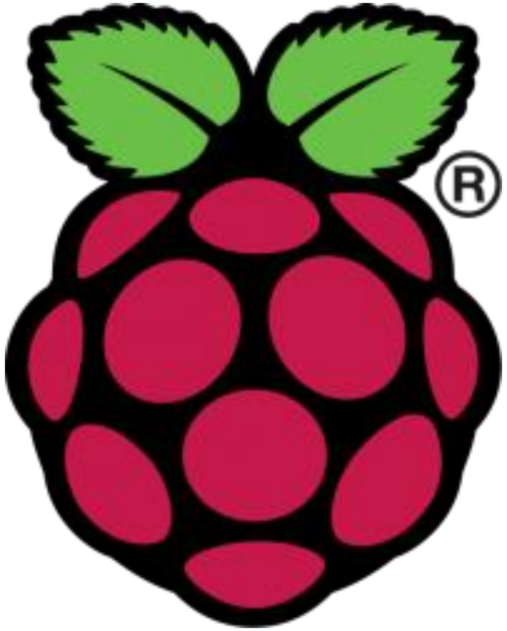
```
Elevation Plot                                              Cursor Elev   15.0 deg.
Azimuth Angle    0.0 deg.                                   Gain          8.52 dBi
Outer Ring       8.52 dBi                                                 0.0 dBmax
                                                                          0.0 dBmax3D
3D Max Gain      8.52 dBi
Slice Max Gain   8.52 dBi @ Elev Angle = 15.0 deg.
Beamwidth        23.8 deg.; -3dB @ 5.1, 28.9 deg.
Sidelobe Gain    8.46 dBi @ Elev Angle = 165.0 deg.
Front/Sidelobe   0.06 dB
```

# Arduino or Raspberry Pi?

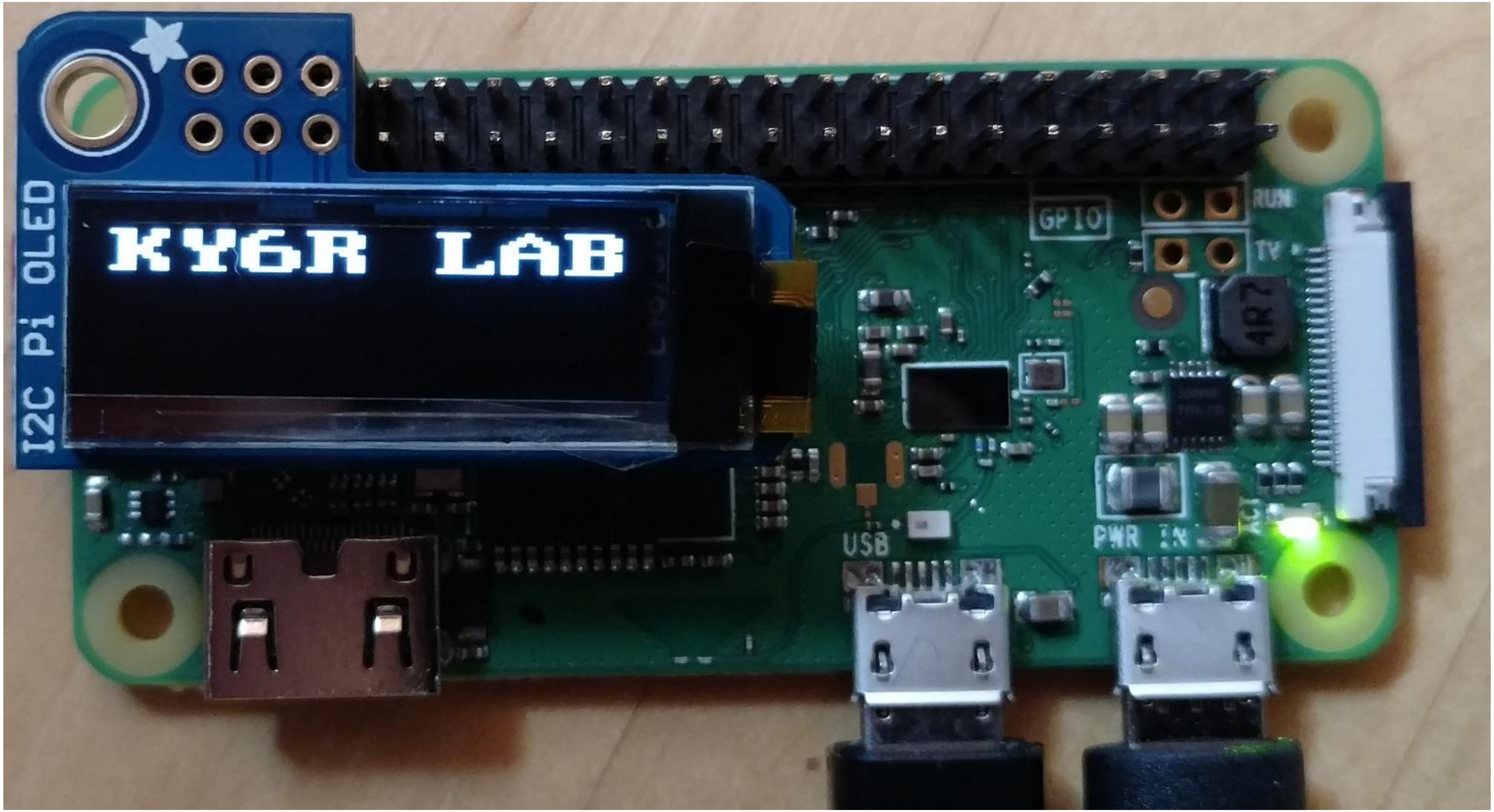


# Elecraft KPOD Driver Decided For Me!

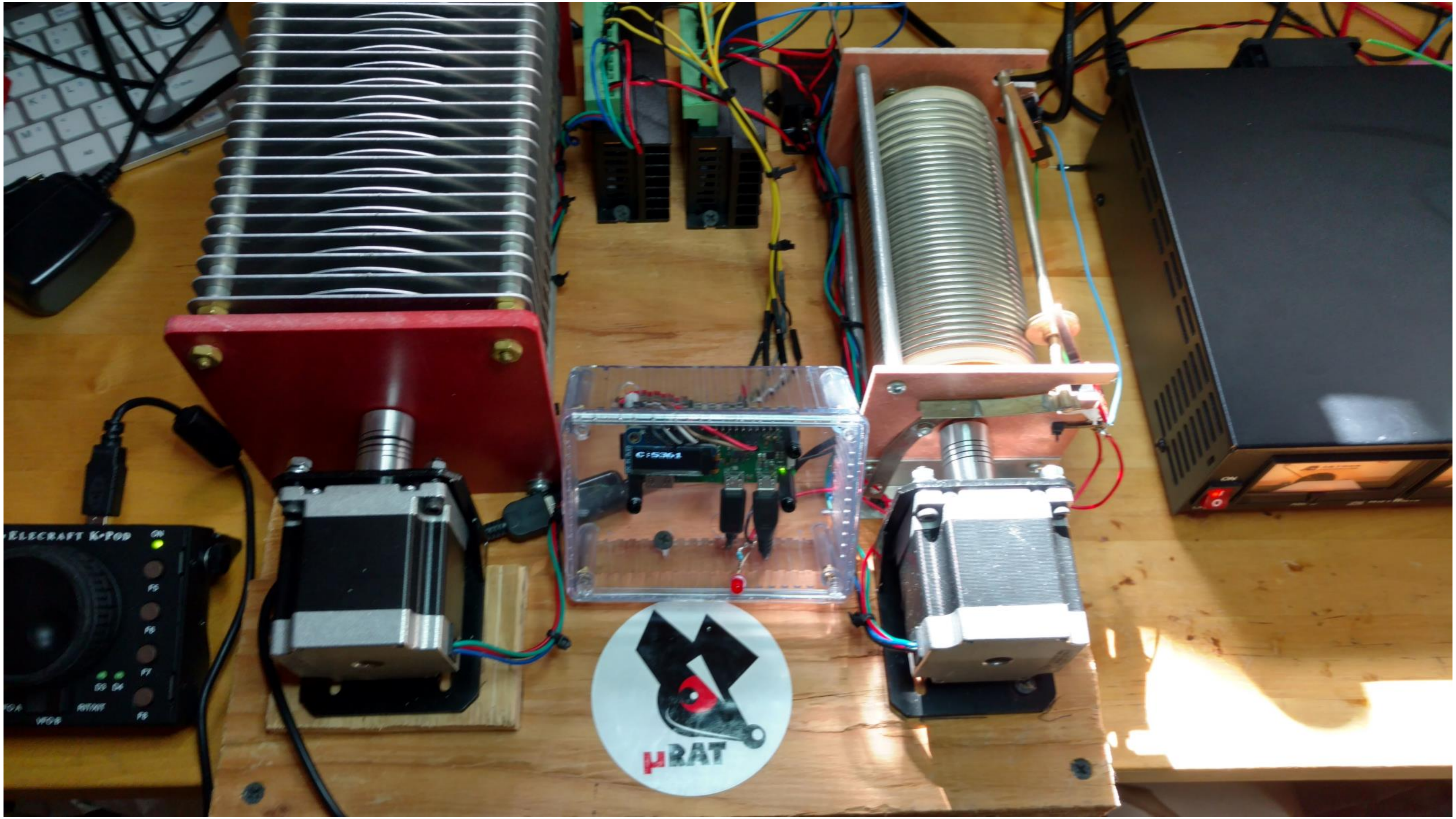
- KPOD driver written (by Paul, N6HZ) in C required Linux to compile
- Raspberry Pi is a Debian Linux based computer (“Raspbian”)

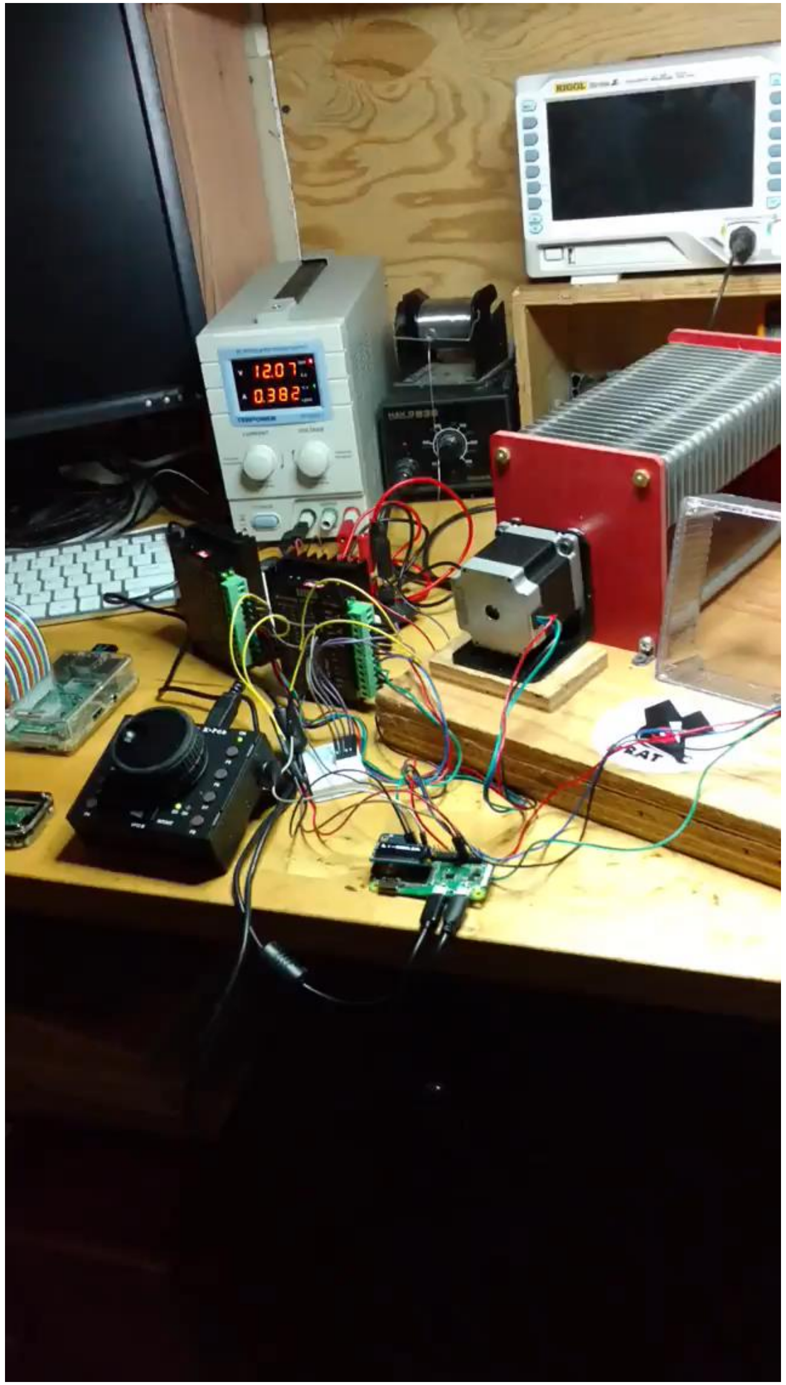


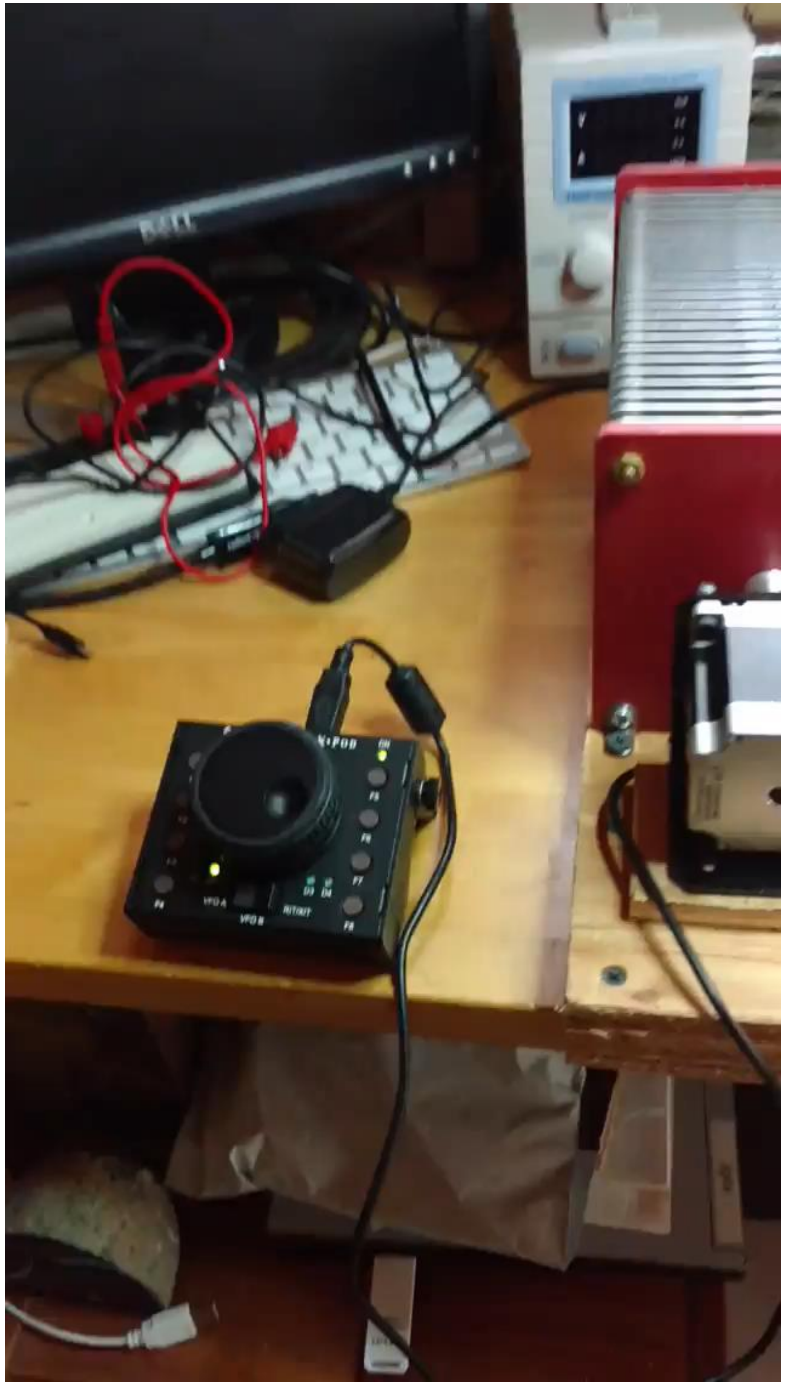
# Raspberry Pi Zero W Plus Adafruit OLED

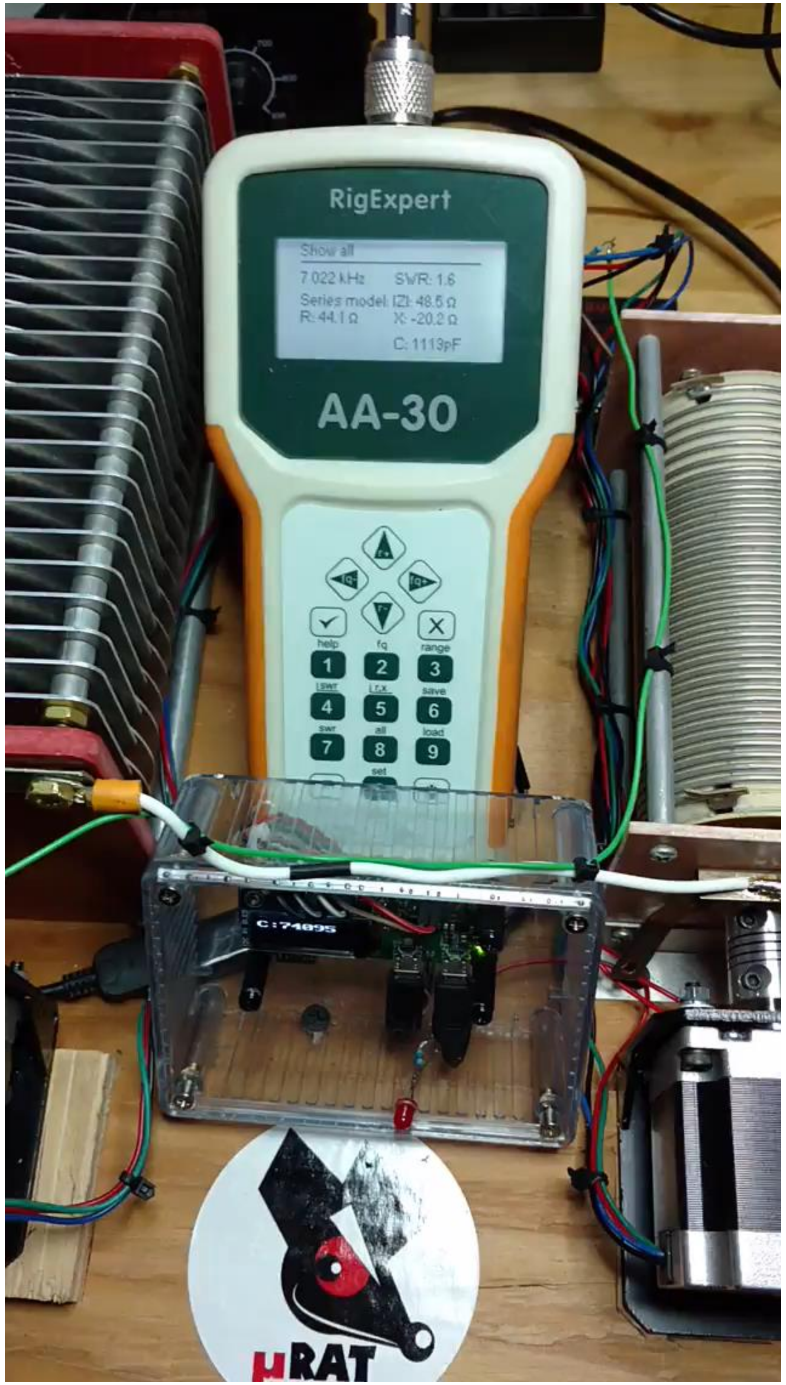


# The Completed u.RAT

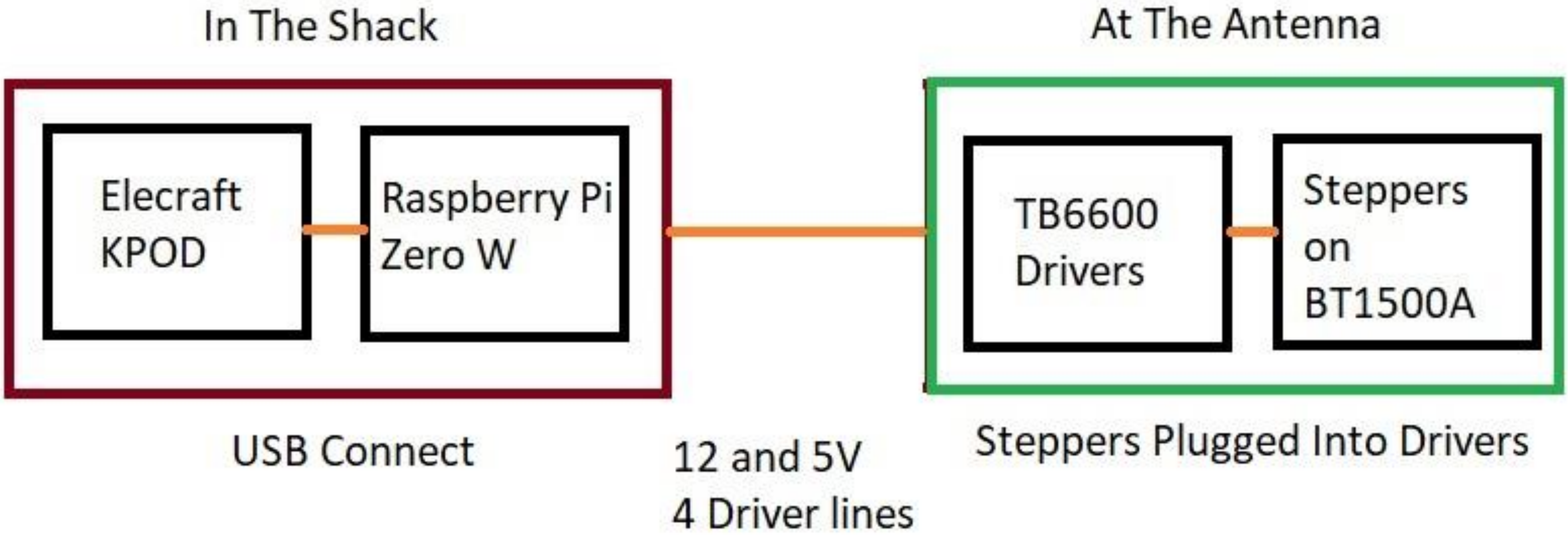



12.07
0.382
µRAT


DELL
VFO A
VFO B


RigExpert
Show all
7.022 kHz SWR: 1.6
Series model: |Z|: 48.5 Ω
R: 44.1 Ω X: -20.2 Ω
C: 1113pF
AA-30
help
fq
range
SWR
save
all
load
set
C:74095
µRAT

# Where Would I Install The u.RAT?



# Maker Projects – Its All About Creativity

- The best Maker – Ham projects seem to design themselves
- Ask yourself “I wonder if” or “What if I put A together with B”
- In my case, the problem I had to solve evolved after I purchased an Expert SPE 1.3K FA solid state amplifier
- My goal was to use the Mod Bob antenna on all low bands – and the Mod Bob is resonant on 160M

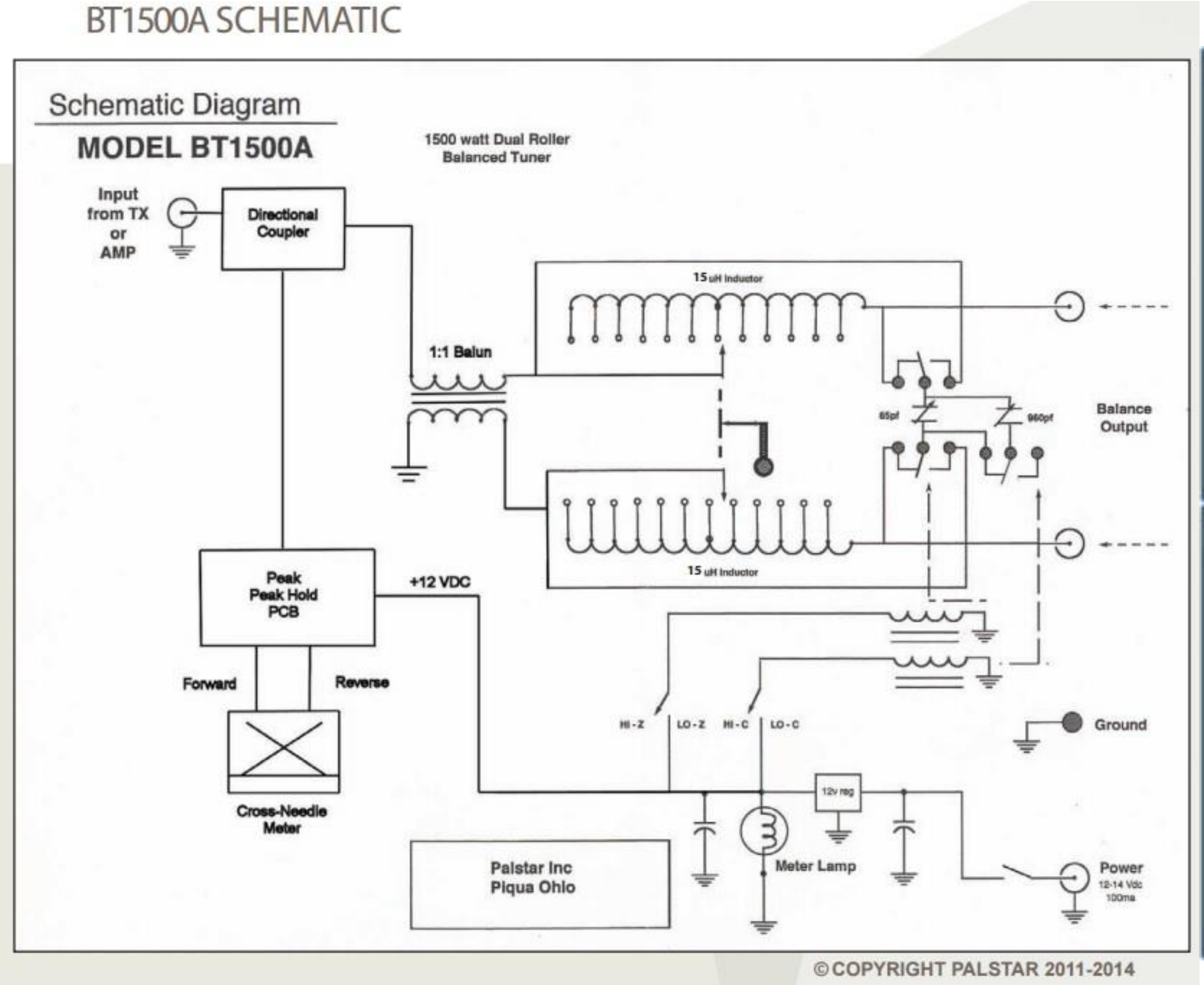
# Solid State Amplifiers Need Low SWR

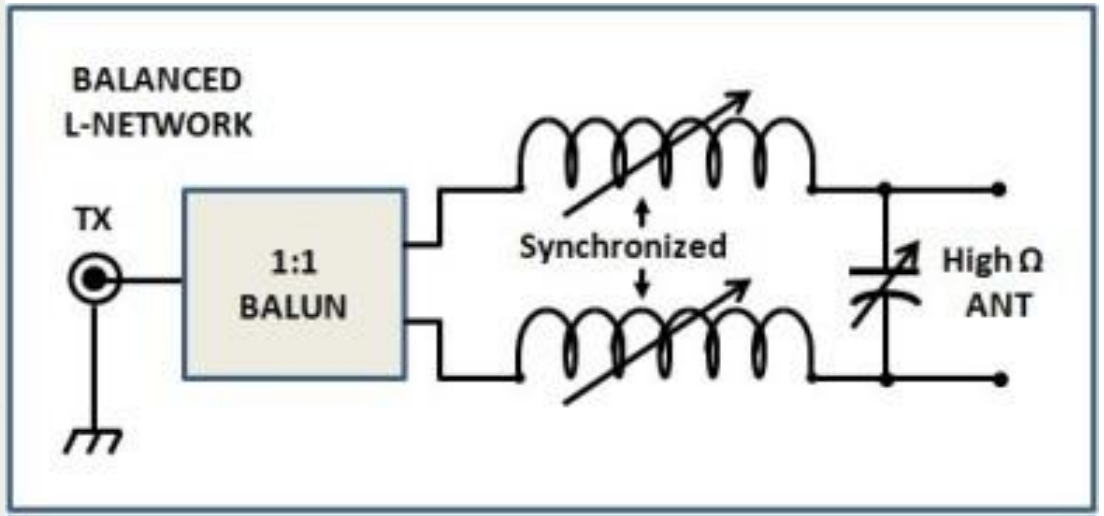
SPE Expert 1.3 – FK Solid State Linear Amplifier – has ATU but range is small

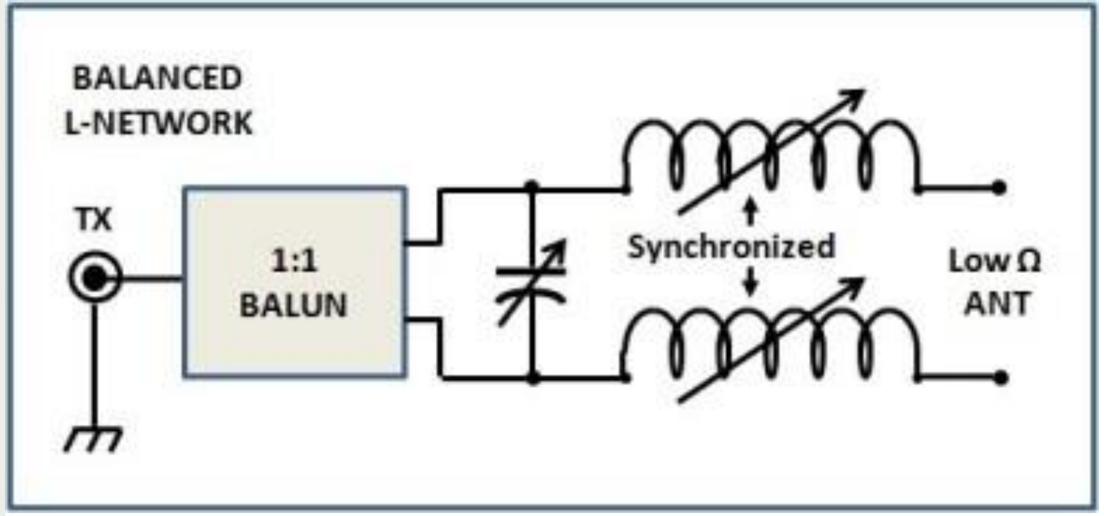
# What Would Be The Best Antenna Coupler?

The answer was easy, the Palstar BT-1500A fit like a glove

# BT1500A is a Balanced Antenna Tuner







# BT1500A Can Switch to Match Hi and Low Z

- The BT1500 can switch the variable capacitor as input to the inductors or on their output
- The inductors are synchronized on the same shaft – control
- The Mod Bob is an open wire balanced “Double Inverted L” antenna – a weird variant on the Bobtail Curtain with far less loss, but which requires exactly the flexibility that the BT1500A offers impedance wise

# The BT1500A Only Requires Two Steppers

- While playing with my KPOD tuning my K3, I saw that the controls were perfect for tuning an antenna
    - The knob has a great feel – even better than the mechanical knobs on the tuner itself
    - The rocker switch could switch up to three stepper motors that could turn any kind of manual antenna tuner meaning any old antenna tuner could be retrofitted and now become a new “digitally controlled” remote antenna tuner
    - If you did not want to spend the money on a KPOD – a simple rotary encoder and set of switches could be used (requiring different and greatly simplified code though)

# The “Universal” Aspect of the u.RAT

- Turning any old manual antenna tuner into an automatic remotely controlled “digital” tuner is what turned this project into an “Open Source” idea (than just solving my own backyard antenna problem)
- A quick email to Elecraft resulted in a collaboration with Paul, N6HZ
- Since the code was C and it required Linux to compile, Raspberry Pi was the only way to go
- Once I started – all of the decisions that I worried about at first just answered themselves – it was like magic!

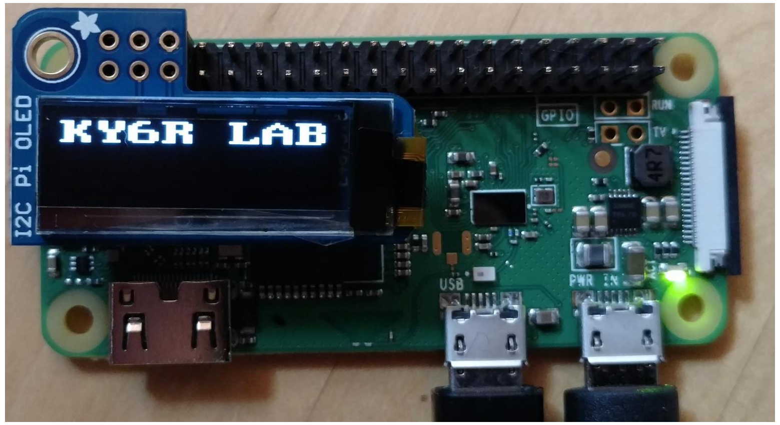
# Decision Steps (For KPOD + BT1500A)

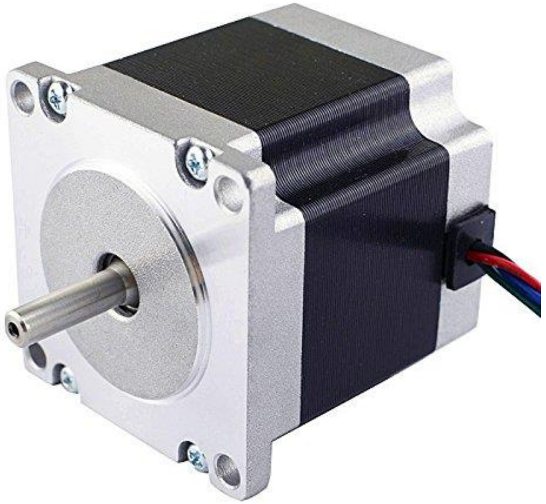
1. Which processor - Arduino or Raspberry Pi => Raspberry Pi
2. Which Stepper Library => WiringPi
3. Which Stepper Motors and Drivers => Nema 23 and TB6600
4. Which Raspberry Pi => Raspberry Pi Zero W
5. Which display => Adafruit Pi-OLED
6. Which Pi-OLED Library => OLED96

# Things I Tried But Didn’t Decide to Do

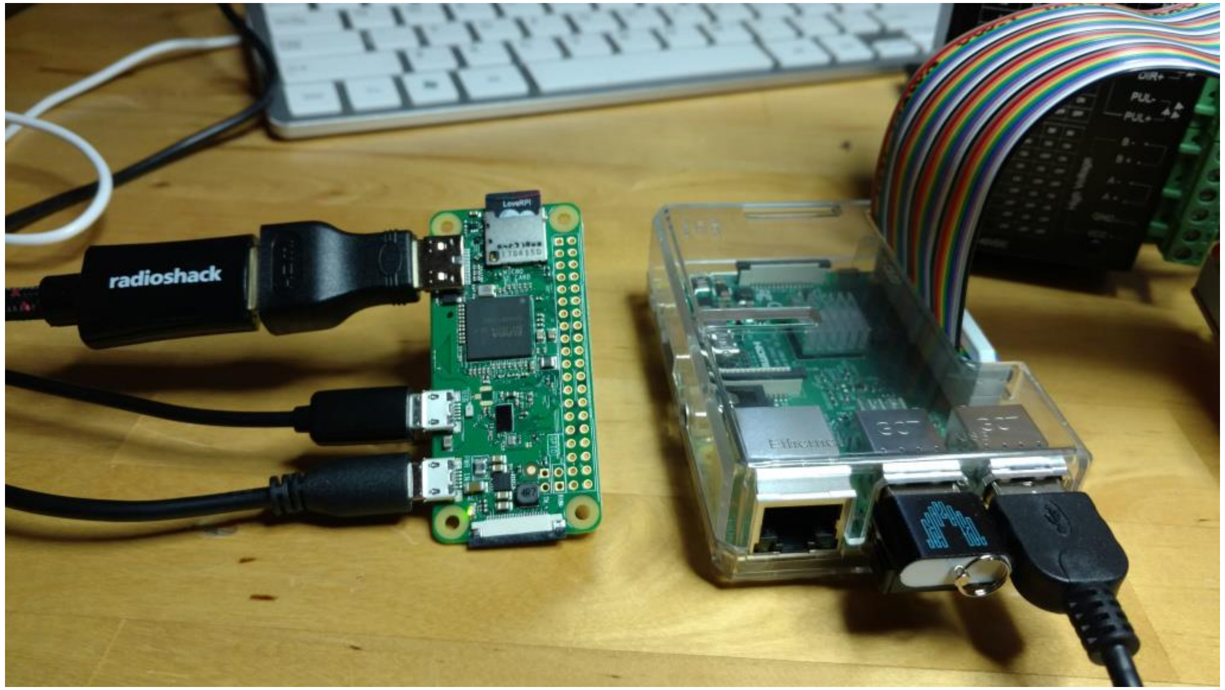
- Just build a very simple rotary encoder and rocker switch controller of my own – instead of using the $250 KPOD
- Just use a simple L tuner based on components I already had in the junk box
- Use an Arduino that I had on hand collecting dust

# Stepper Motors, Drivers, etc

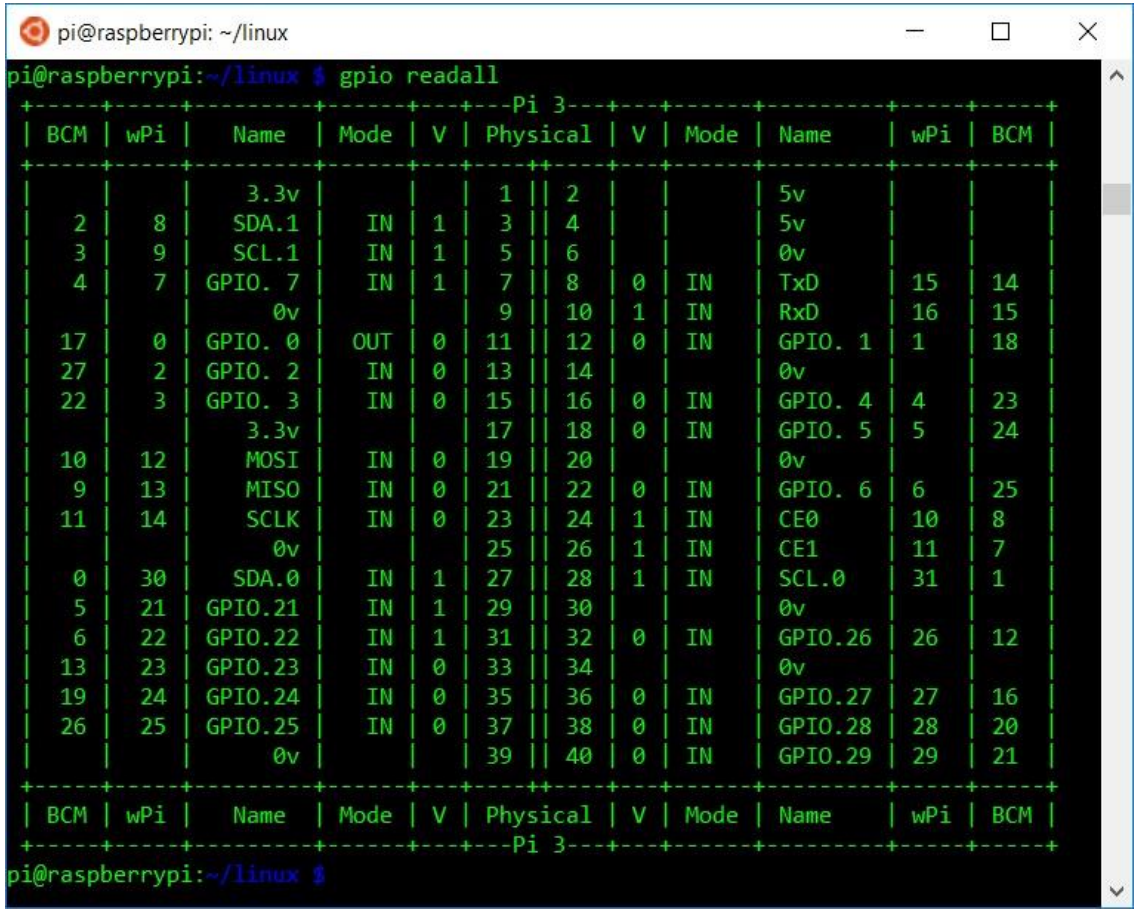




# Raspberry Pi Development Environment

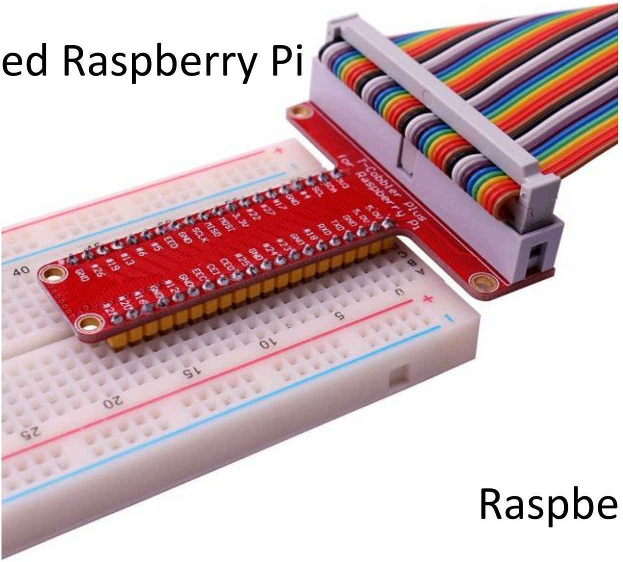


Raspberry Pi Zero W and full sized Raspberry Pi



```
pi@raspberrypi:~/linux $ gpio readall
 +-----+-----+---------+------+---+---Pi 3---+---+------+---------+-----+-----+
 | BCM | wPi |   Name  | Mode | V | Physical | V | Mode | Name    | wPi | BCM |
 +-----+-----+---------+------+---+----++----+---+------+---------+-----+-----+
 |     |     |    3.3v |      |   |  1 || 2  |   |      | 5v      |     |     |
 |   2 |   8 |   SDA.1 |   IN | 1 |  3 || 4  |   |      | 5v      |     |     |
 |   3 |   9 |   SCL.1 |   IN | 1 |  5 || 6  |   |      | 0v      |     |     |
 |   4 |   7 | GPIO. 7 |   IN | 1 |  7 || 8  | 0 | IN   | TxD     | 15  | 14  |
 |     |     |      0v |      |   |  9 || 10 | 1 | IN   | RxD     | 16  | 15  |
 |  17 |   0 | GPIO. 0 |  OUT | 0 | 11 || 12 | 0 | IN   | GPIO. 1 | 1   | 18  |
 |  27 |   2 | GPIO. 2 |   IN | 0 | 13 || 14 |   |      | 0v      |     |     |
 |  22 |   3 | GPIO. 3 |   IN | 0 | 15 || 16 | 0 | IN   | GPIO. 4 | 4   | 23  |
 |     |     |    3.3v |      |   | 17 || 18 | 0 | IN   | GPIO. 5 | 5   | 24  |
 |  10 |  12 |    MOSI |   IN | 0 | 19 || 20 |   |      | 0v      |     |     |
 |   9 |  13 |    MISO |   IN | 0 | 21 || 22 | 0 | IN   | GPIO. 6 | 6   | 25  |
 |  11 |  14 |    SCLK |   IN | 0 | 23 || 24 | 1 | IN   | CE0     | 10  | 8   |
 |     |     |      0v |      |   | 25 || 26 | 1 | IN   | CE1     | 11  | 7   |
 |   0 |  30 |   SDA.0 |   IN | 1 | 27 || 28 | 1 | IN   | SCL.0   | 31  | 1   |
 |   5 |  21 | GPIO.21 |   IN | 1 | 29 || 30 |   |      | 0v      |     |     |
 |   6 |  22 | GPIO.22 |   IN | 1 | 31 || 32 | 0 | IN   | GPIO.26 | 26  | 12  |
 |  13 |  23 | GPIO.23 |   IN | 0 | 33 || 34 |   |      | 0v      |     |     |
 |  19 |  24 | GPIO.24 |   IN | 0 | 35 || 36 | 0 | IN   | GPIO.27 | 27  | 16  |
 |  26 |  25 | GPIO.25 |   IN | 0 | 37 || 38 | 0 | IN   | GPIO.28 | 28  | 20  |
 |     |     |      0v |      |   | 39 || 40 | 0 | IN   | GPIO.29 | 29  | 21  |
 +-----+-----+---------+------+---+----++----+---+------+---------+-----+-----+
 | BCM | wPi |   Name  | Mode | V | Physical | V | Mode | Name    | wPi | BCM |
 +-----+-----+---------+------+---+---Pi 3---+---+------+---------+-----+-----+
pi@raspberrypi:~/linux $
```

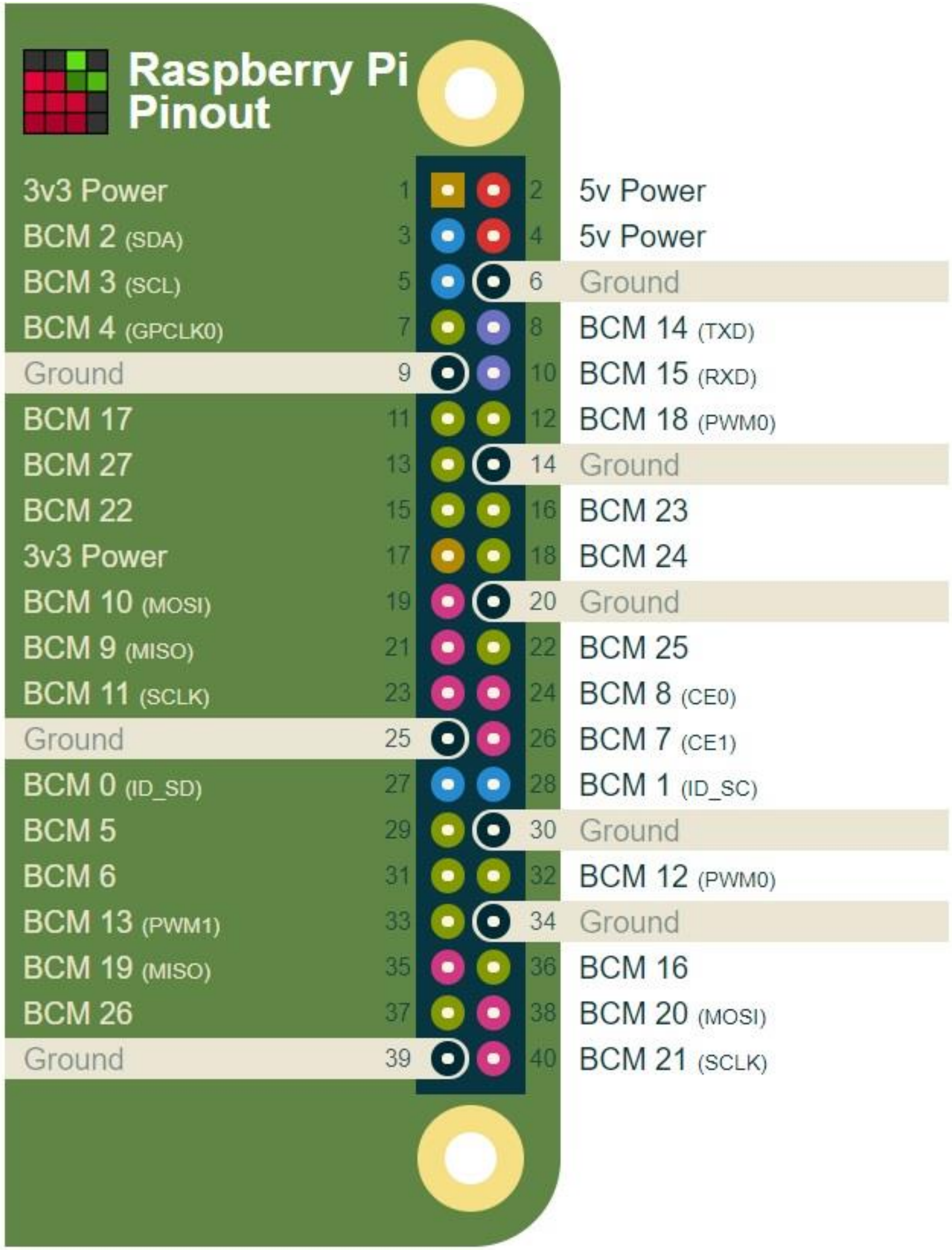
WiringPi Utility “GPIO”



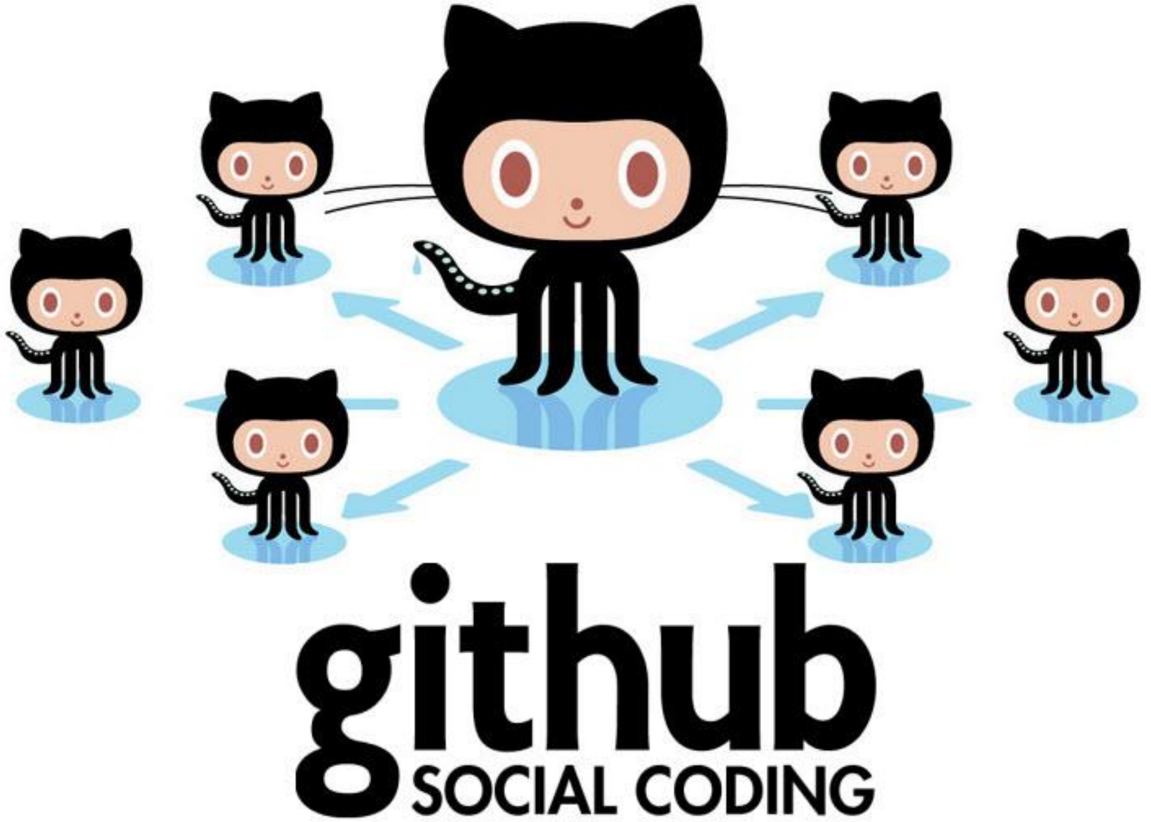
Raspberry Pi Breakout Board (“Cobbler”)

# Raspberry Pi GPIO Pinout

| HY-DIV268N-5A | Connects to: |
|---|---|
| DIR- | RPi GPIO GND |
| DIR+ | RPi GPIO #18 |
| PUL- | RPi GPIO GND |
| PUL+ | RPi GPIO #23 |
| EN- | RPi GPIO GND |
| EN+ | RPi GPIO #4 or GND |
| A+ | stepper green wire |
| A- | stepper gray wire |
| B+ | stepper red wire |
| B- | stepper yellow wire |
| DC- | GND (black on PC power supply) |
| DC+ | +12v (yellow on PC power supply) |



# Github is Your Best Friend



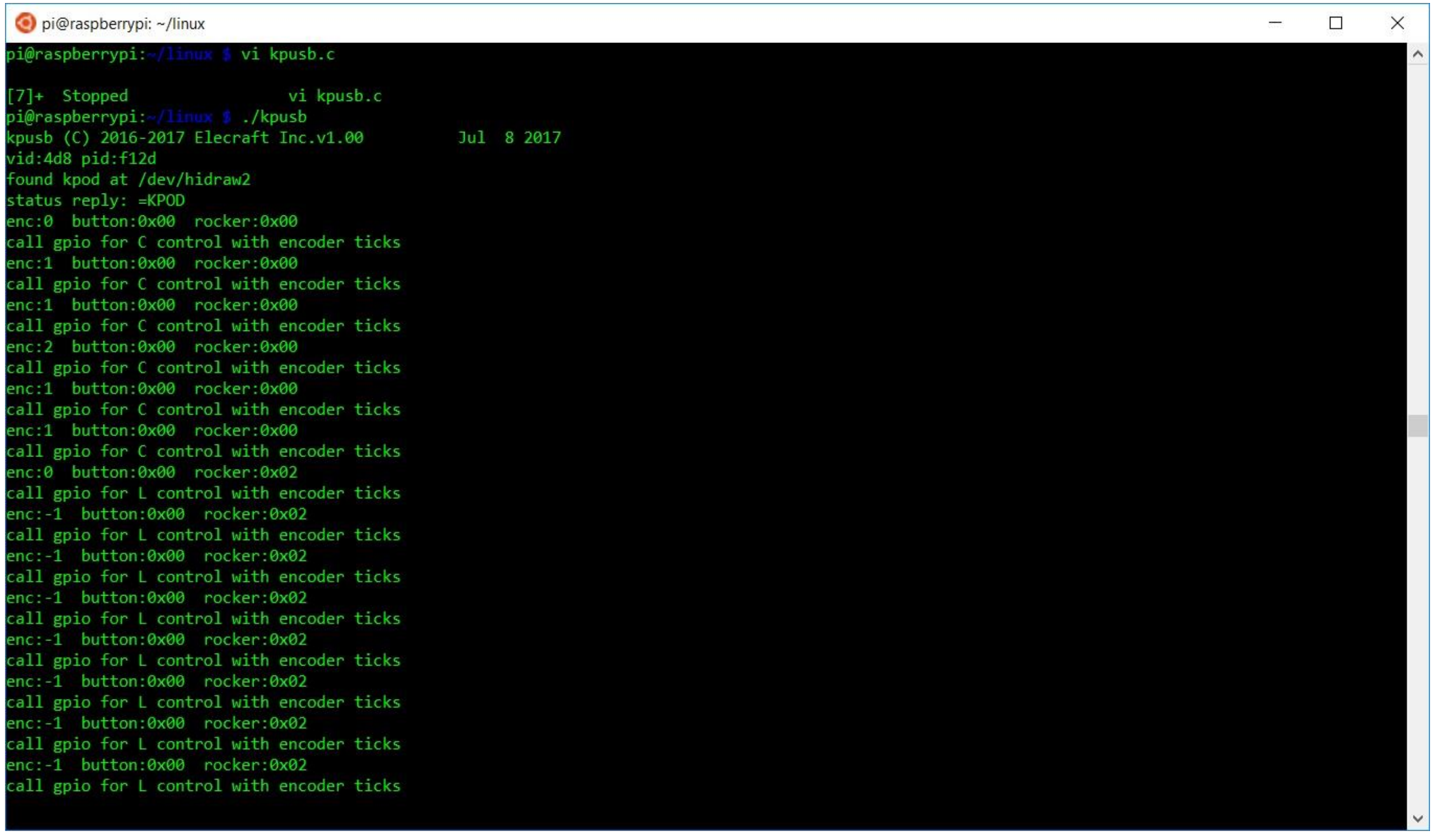
# The Code

- ***Elecraft KPOD device driver*** – written in C (Kernighan and Ritchie)
    - HID RAW USB device
- ***Wiring Pi*** – emulates an Arduino wiring scheme but written in C and runs on Raspberry Pi
- ***Oled96*** – library for the Adafruit OLED

```
#include <stdint.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <time.h>
#include <errno.h>
#include <string.h>
#include <sys/ioctl.h>
#include <ctype.h>
#include <linux/types.h>
#include <linux/hidraw.h>
#include <wiringPi.h>
#include "oled96.h"
```

# Code

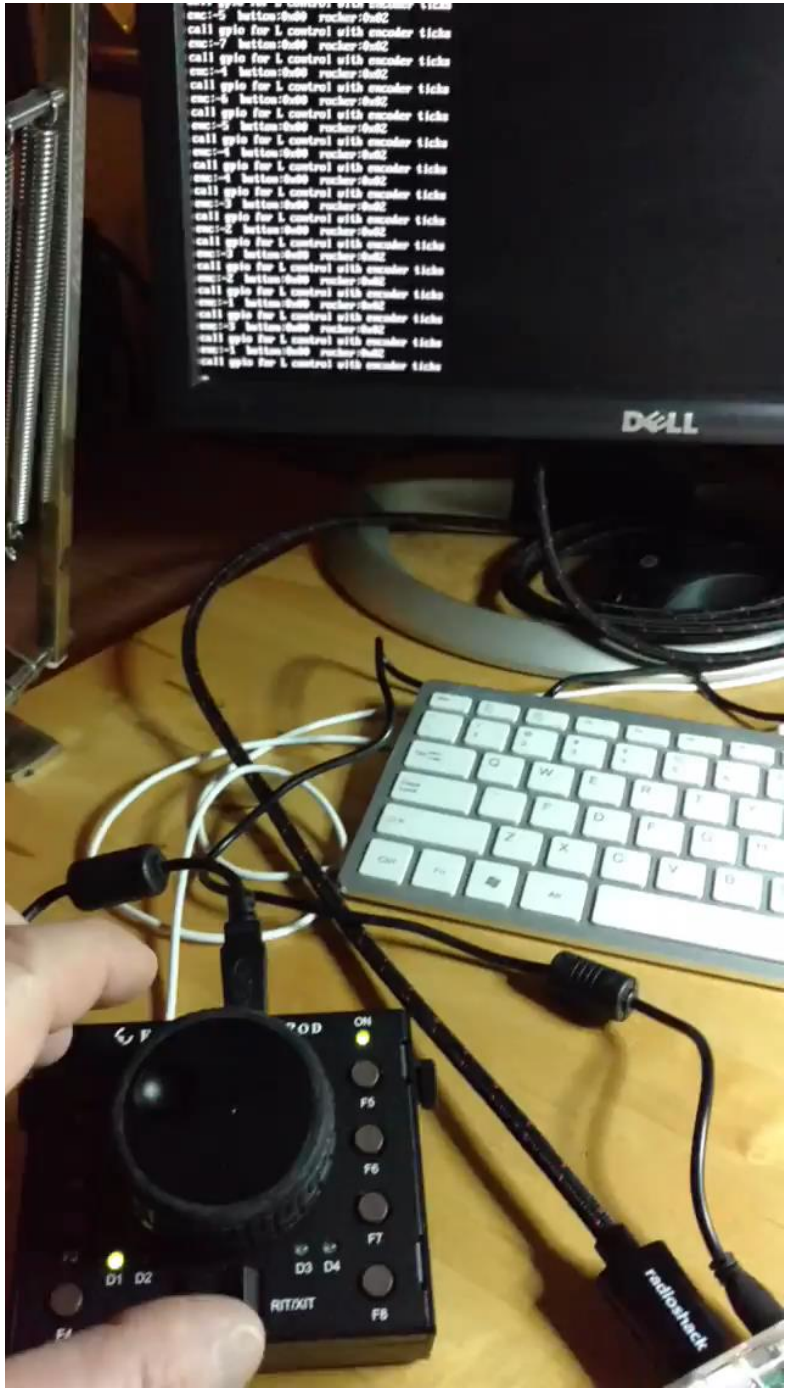
- Took it one step at a time
  - Got the KPOD to be seen by the Raspberry Pi
  - Got the steppers to move when I turned the KPOD knob and used its rocker switch
  - Switched from the regular Raspberry Pi 3 to the Zero W and then got the OLED to work



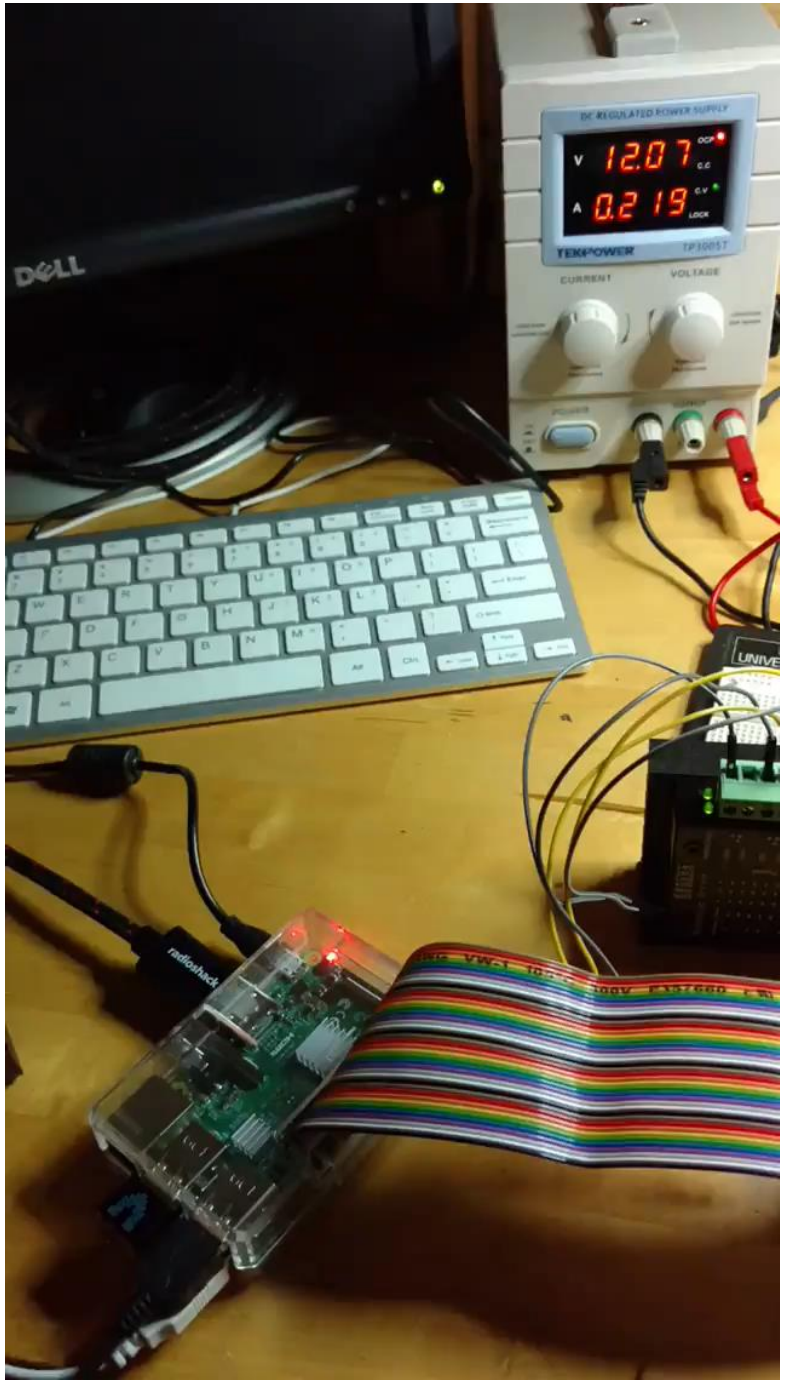
```
pi@raspberrypi:~/linux $ vi kpusb.c

[7]+  Stopped                 vi kpusb.c
pi@raspberrypi:~/linux $ ./kpusb
kpusb (C) 2016-2017 Elecraft Inc.v1.00          Jul  8 2017
vid:4d8 pid:f12d
found kpod at /dev/hidraw2
status reply: =KPOD
enc:0  button:0x00  rocker:0x00
call gpio for C control with encoder ticks
enc:1  button:0x00  rocker:0x00
call gpio for C control with encoder ticks
enc:1  button:0x00  rocker:0x00
call gpio for C control with encoder ticks
enc:2  button:0x00  rocker:0x00
call gpio for C control with encoder ticks
enc:1  button:0x00  rocker:0x00
call gpio for C control with encoder ticks
enc:1  button:0x00  rocker:0x00
call gpio for C control with encoder ticks
enc:0  button:0x00  rocker:0x02
call gpio for L control with encoder ticks
enc:-1  button:0x00  rocker:0x02
call gpio for L control with encoder ticks
enc:-1  button:0x00  rocker:0x02
call gpio for L control with encoder ticks
enc:-1  button:0x00  rocker:0x02
call gpio for L control with encoder ticks
enc:-1  button:0x00  rocker:0x02
call gpio for L control with encoder ticks
enc:-1  button:0x00  rocker:0x02
call gpio for L control with encoder ticks
enc:-1  button:0x00  rocker:0x02
call gpio for L control with encoder ticks
enc:-1  button:0x00  rocker:0x02
call gpio for L control with encoder ticks
```

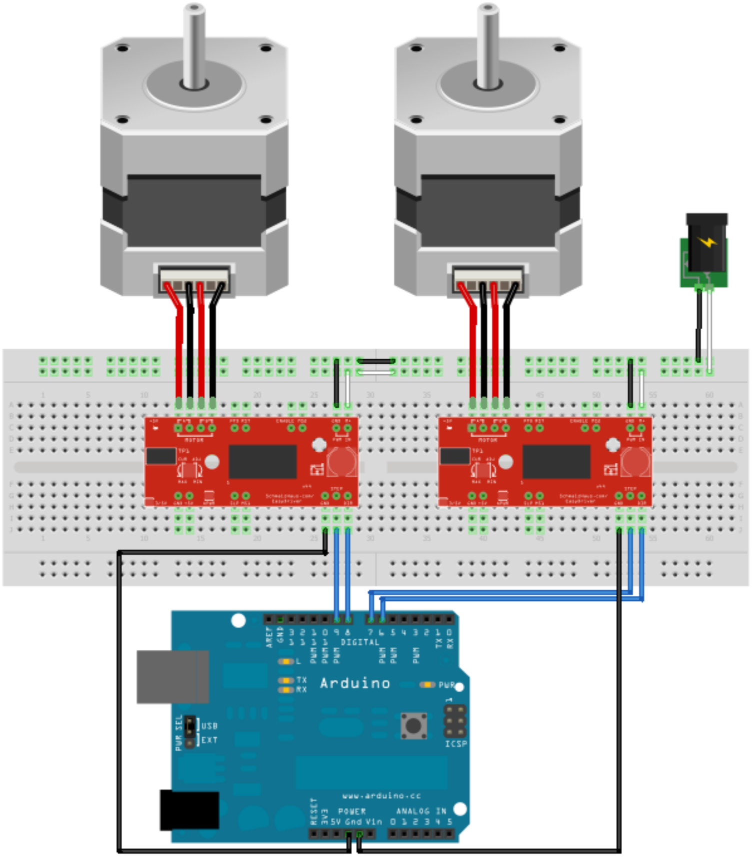
# Early Tests



```
call gpio for L control with encoder ticks
enc:-5  button:0x00  rocker:0x02
call gpio for L control with encoder ticks
enc:-7  button:0x00  rocker:0x02
call gpio for L control with encoder ticks
enc:-4  button:0x00  rocker:0x02
call gpio for L control with encoder ticks
enc:-6  button:0x00  rocker:0x02
call gpio for L control with encoder ticks
enc:-5  button:0x00  rocker:0x02
call gpio for L control with encoder ticks
enc:-4  button:0x00  rocker:0x02
call gpio for L control with encoder ticks
enc:-4  button:0x00  rocker:0x02
call gpio for L control with encoder ticks
enc:-3  button:0x00  rocker:0x02
call gpio for L control with encoder ticks
enc:-2  button:0x00  rocker:0x02
call gpio for L control with encoder ticks
enc:-3  button:0x00  rocker:0x02
call gpio for L control with encoder ticks
enc:-2  button:0x00  rocker:0x02
call gpio for L control with encoder ticks
enc:-1  button:0x00  rocker:0x02
call gpio for L control with encoder ticks
enc:-3  button:0x00  rocker:0x02
call gpio for L control with encoder ticks
enc:-1  button:0x00  rocker:0x02
call gpio for L control with encoder ticks
```

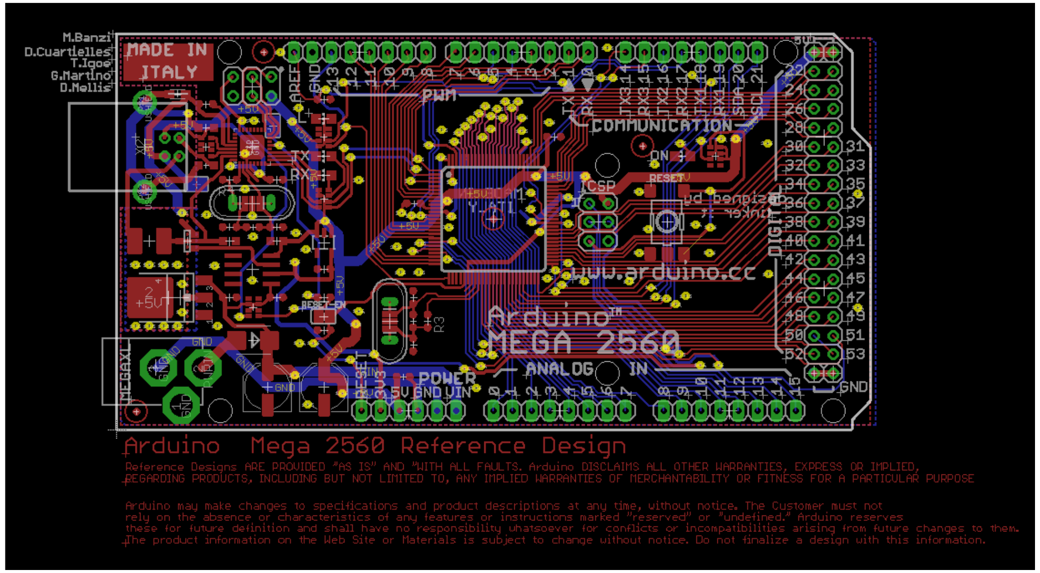


# Documenting Your Project



Fritzing – generate schematics and PC boards from drawings

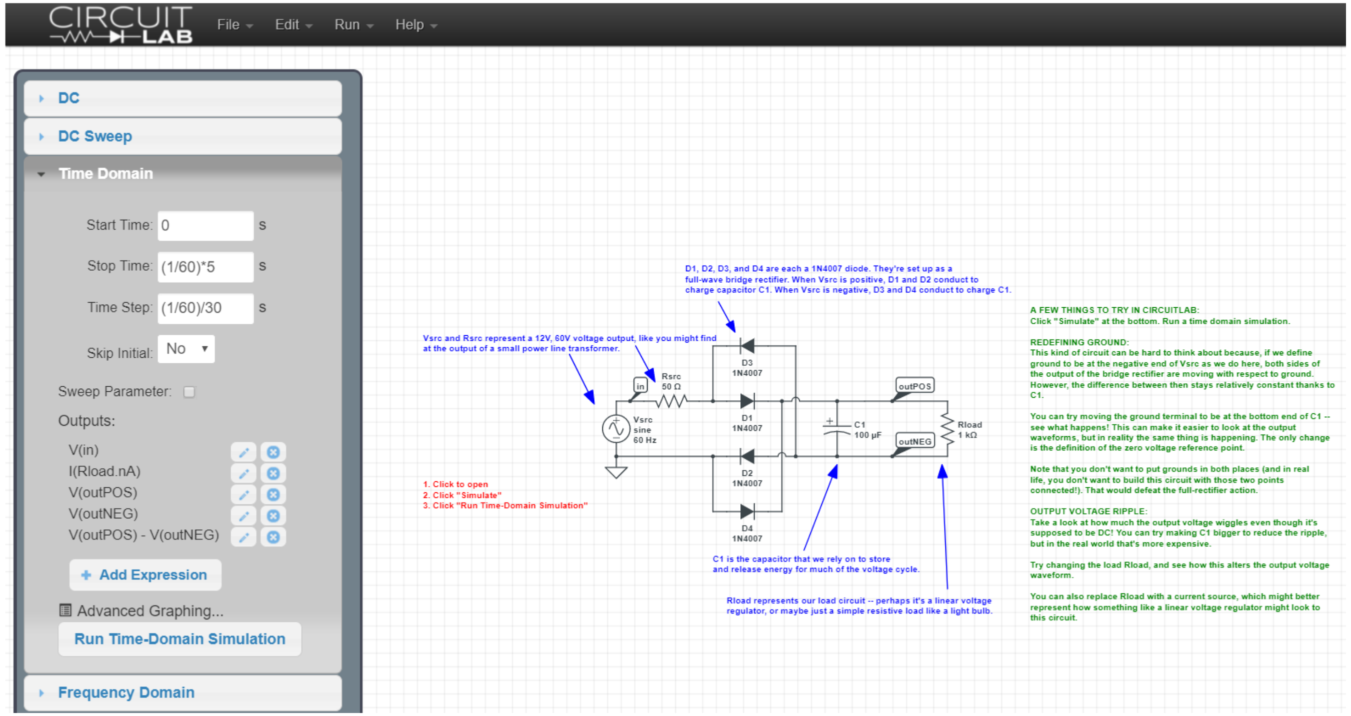
Eagle by AutoCad



Arduino Mega 2560 Reference Design

Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS. Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Arduino reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.



Start Time: 0 s

Stop Time: (1/60)*5 s

Time Step: (1/60)/30 s

Skip Initial: No

Sweep Parameter: □

Outputs:

V(in)

I(Rload.nA)

V(outPOS)

V(outNEG)

V(outPOS) - V(outNEG)

Advanced Graphing...

D1, D2, D3, and D4 are each a 1N4007 diode. They're set up as a full-wave bridge rectifier. When Vsrc is positive, D1 and D2 conduct to charge capacitor C1. When Vsrc is negative, D3 and D4 conduct to charge C1.

Vsrc and Rsrc represent a 12V, 60V voltage output, like you might find at the output of a small power line transformer.

1. Click to open
2. Click "Simulate"
3. Click "Run Time-Domain Simulation"

C1 is the capacitor that we rely on to store and release energy for much of the voltage cycle.

Rload represents our load circuit -- perhaps it's a linear voltage regulator, or maybe just a simple resistive load like a light bulb.

A FEW THINGS TO TRY IN CIRCUITLAB:
Click "Simulate" at the bottom. Run a time domain simulation.

REDEFINING GROUND:
This kind of circuit can be hard to think about because, if we define ground to be at the negative end of Vsrc as we do here, both sides of the output of the bridge rectifier are moving with respect to ground. However, the difference between then stays relatively constant thanks to C1.

You can try moving the ground terminal to be at the bottom end of C1 -- see what happens! This can make it easier to look at the output waveforms, but in reality the same thing is happening. The only change is the definition of the zero voltage reference point.

Note that you don't want to put grounds in both places (and in real life, you don't want to build this circuit with those two points connected!). That would defeat the full-rectifier action.

OUTPUT VOLTAGE RIPPLE:
Take a look at how much the output voltage wiggles even though it's supposed to be DC! You can try making C1 bigger to reduce the ripple, but in the real world that's more expensive.

Try changing the load Rload, and see how this alters the output voltage waveform.

You can also replace Rload with a current source, which might better represent how something like a linear voltage regulator might look to this circuit.

CircuitLab

# Key Learnings

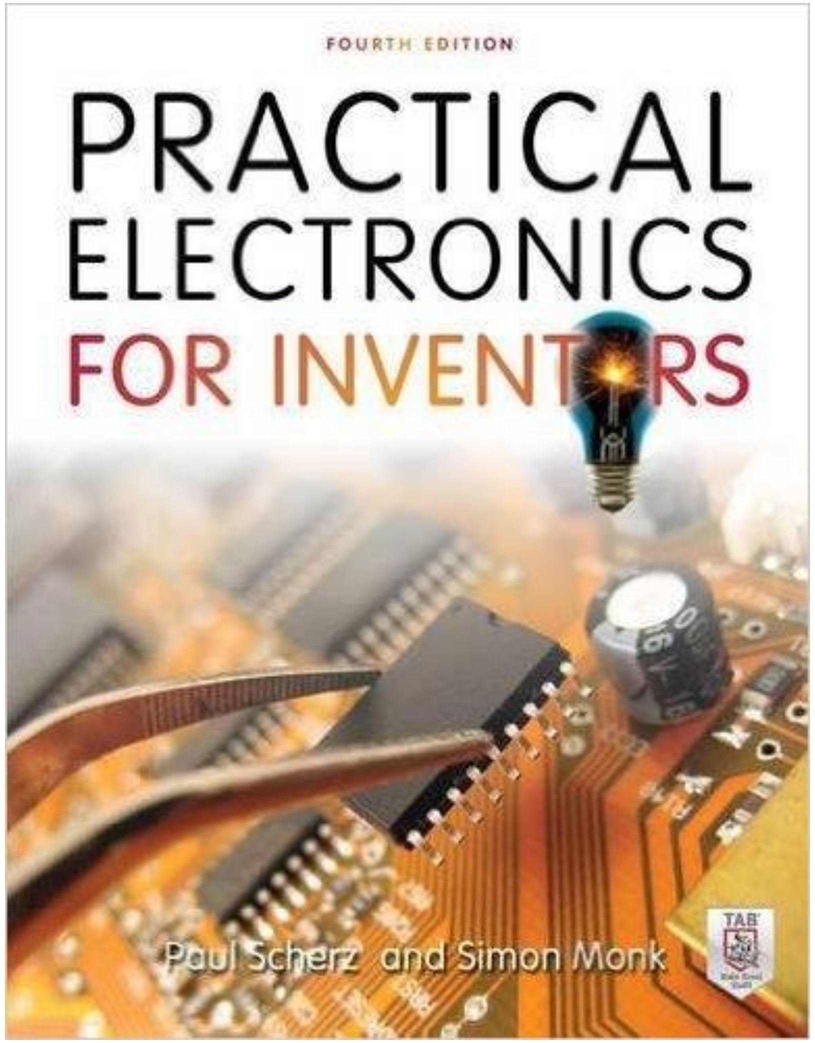
- The Raspberry Pi Zero W and its uses, plus “headless” operation
- Controlling Stepper motors with the Raspberry Pi, the GPIO and WiringPi – that emulates the Arduino
- Embedded devices and the Internet of Things (IOT)
- Ham Shack Remote control – wired or wireless
- Antenna tuners and when you need a balanced antenna tuner
- Arduino vs. Raspberry Pi
- C and Python as Maker languages
- USB device drivers and HID vs. HIDRAW
- OLED devices on the Raspberry Pi Zero W
- Github and Open Sourcing a project
- How antenna tuners work – and the different circuits employed (esp. Balanced Tuners)
- Antenna impedance matching and solid state amplifiers and what they “need” . ..

# How Can This Project Be Extended?

- Someone could easily add a third stepper for a 3 control tuner
- Someone could use a tuner that has a fixed switchable inductor
- Someone could remote the entire project and use WiFi
- Someone could make tuning automatic for any band by tapping into the SWR circuit on their tuner
- Someone could add the ability to save band settings as KPOD button memory loctions
- Someone could also add a remote wireless antenna switch so that it follows the u.RAT switching

# Resources

- My blog – Top Band Chordal Hopper (Google it)
- Adafruit
- Hackaday
- Github
- Amazon
- Youtube - https://youtu.be/Dc16mKFA7Fo
  - Google stepper, etc


FOURTH EDITION
PRACTICAL ELECTRONICS FOR INVENTORS
Paul Scherz and Simon Monk
TAB