**Design Notes for the NA6O Power Meter Project**

NA6O Jan 2023

These are general design notes in no special order.

**NOTE ON ALL GPIO: Some have weak pulldowns. Use 10K or lower for all pullups.**

**LCD Notes**

New Haven Display NHD-0420D3Z-FL-GBW-V3 20x4 $27 at Mouser

Chars 4.15mm = 0.16 in, pretty small.

TTL serial interface, connects directly to Pi GPIO. Write-only.

Benchmark:

(Default) 9600 baud, write 4x20 chars: 96 ms

57,600: 24 ms <-- using this

115,200: 8 ms

To change baud rate, run the Change LCD Baud Rate Utility.vi, located in the Pwr Meter SubVIs folder. The LCD needs to be connected to the Pi. The resulting baud rate must be the same as the LCD Baud constant on the Power Meter Controller.vi diagram.

**ADC Notes**

ADS1115 (16 bit, 2 diff) and ADS1256 (24 bit, 2 diff) via i2C now supported by LINX.

<https://www.vipm.io/package/mediamongrels_lib_linx_raspberry_pi_addons/>

Install from VIPM (LINX Raspberry Pi Addons)

Little Amazon ADS1115 adapter board I2C adr 0x48 on bus 1.

Run on 5V to get max input range.

With 5V supply, 5V I/O but running straight off pi I2C ok without level shifter.

Internal ref.

Has PGA, up to Vdd range.

Single-shot or various sample rates up to 860 sps. Driver only does single reads but ADC is actually sampling and taking advantage of digital filter.

**AD8310 Info**

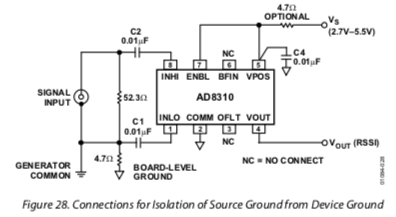
Note: Not RMS responding.

(AD makes some RMS chips that go up to uWave. Less dynamic range, demod output not fast enough for nice risetime testing. Difficult package to solder.)

2. to 5.5V, 7.5 mA. Internal Vref.

-74 to +16 dBm => 2.5V out with 5V supply

8310 has buffered output. Is MSOP (.025 pin spacing)



AD8310 module – Get on Ebay for testing and evaluation. Here is preliminary data from that:

Had to change to 51 ohm Rin, came with 200 ohms for some reason.

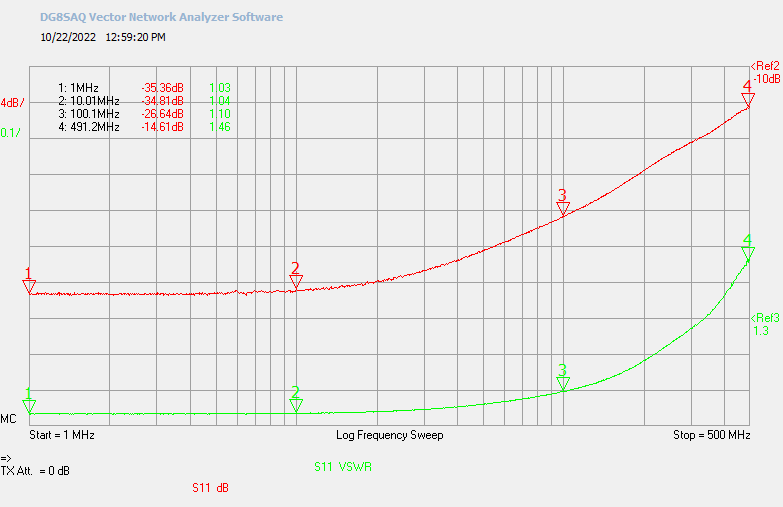
|  |  |
| --- | --- |
| dBm | Vout |
| 15 | 2.628 |
| 10 | 2.514 |
| 0 | 2.265 |
| -10 | 2.014 |
| -20 | 1.771 |
| -30 | 1.524 |
| -40 | 1.271 |
| -50 | 1.033 |
| -60 | 0.792 |
| -70 | 0.545 |
| -75 | 0.410 |
| -80 | 0.320 |
| -85 | 0.276 |
| nosig | 0.254 |

dBm= -91.923 + 40.624 \* Volts

No output filter. Ripple at -15 dBm = 420 mV p-p at carrier freq

1.0 nf output filter. Step to 0 dBm, Tr = 6.6 us. Mbe 5 mV p-p noise 0-20 MHz BW.





SimSmith says Cin is about 2.4 pF. Add 5 nH in series to improve HF match. Chip is rated 1.4 pF.

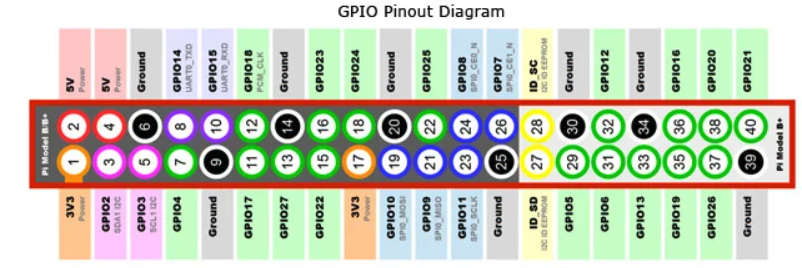
**Attenuator Info**

Extend power range upward from +16 dBm to a few watts. Relay switched. Watch power dissipation, could cause drift in detector.

Anaren 20 dB/10W, D10AA20Z4, 200x100 mil chip. (Or 30 dB D10AA30Z4) Needs heatsink on back of board. Cheap. Then max power is +36 dBm = 4W. A bit more with loss of accuracy. $5.29 DigiKey or Mouser.

Relay for attenuator:

Omron [G6K-2F-RF-T-TR03 DC5](https://www.mouser.com/ProductDetail/Omron-Electronics/G6K-2F-RF-T-TR03-DC5?qs=bR9%2Fh8%2FHa7Q3LGohyKHVjA%3D%3D) DPDT 5V/21 mA $18 DigiKey or Mouser



**Signals to Sensor - 4 pair**

1 +5 power

2 Power return

4 Sig+

5 Sig-

3 Control bit 0 (out)

6 Control bit 1 (out)

7 Status bit 0 (in)

8 Status bit 1 (in)

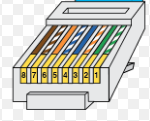
**Sensor Cable and Connector**

8 pins plus overall connector shield,

CAT7 cable: ITSP plus overall shield and nice metal RJ45.

Shielded jack, metal housing. Make sure it makes good connection to panel.

Cable Matters shielded RJ45 from Amazon

Inside, use a stub of CAT5 or whatever.

**Power and Boot/Shutdown Manager for Pi**

Button that triggers Reboot, Halt,vi These are the commands that it implements:

/usr/bin/sudo /sbin/shutdown -h now

or -r for reboot

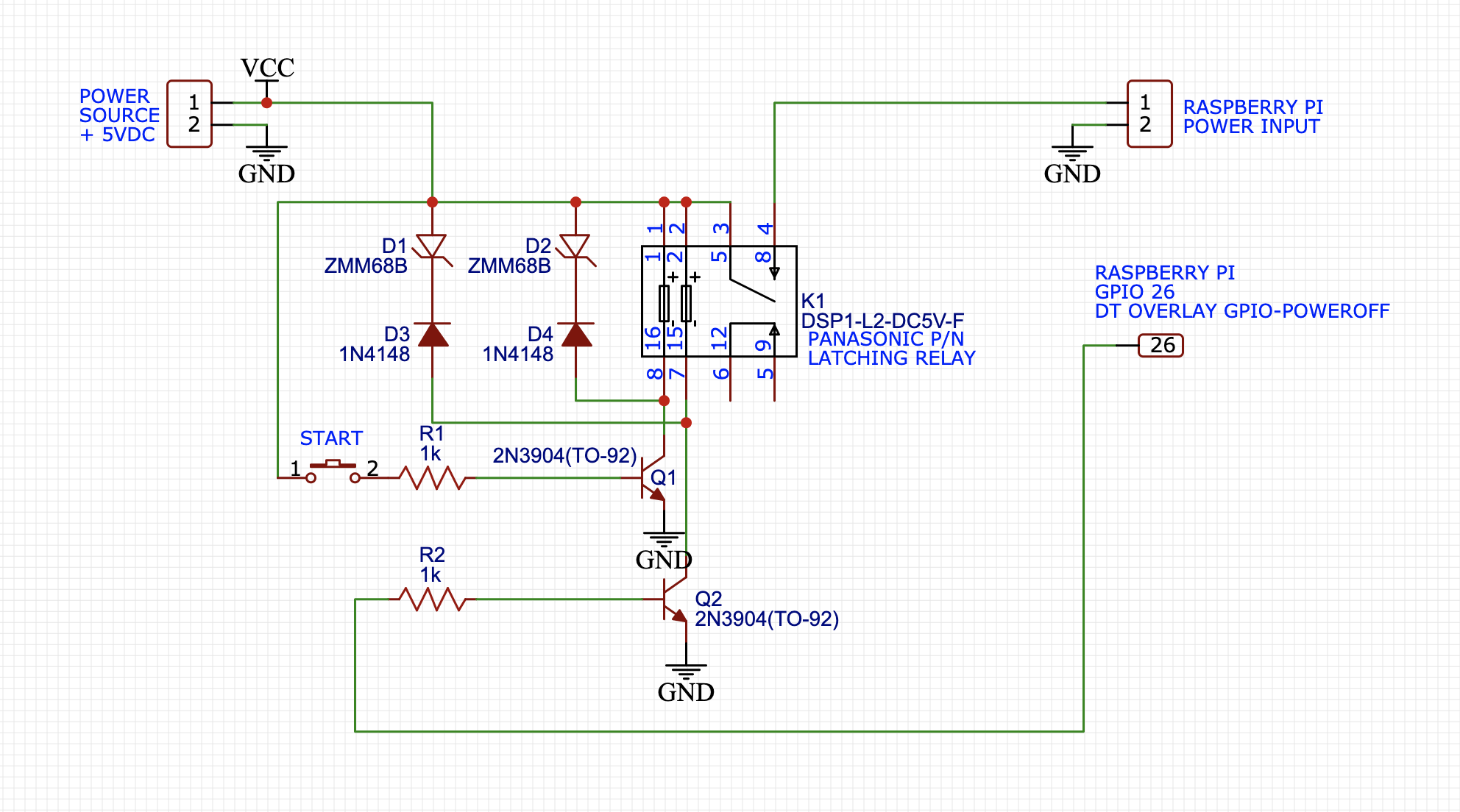
Solution to hardware powe-off:

<https://raspberrypi.stackexchange.com/questions/133966/cut-power-supply-to-pi-after-shutdown-and-power-on-after-button-press>

More details:

<https://raspberrypi.stackexchange.com/questions/89297/how-long-to-wait-after-shutdown-to-cut-power/130443#130443>

Uses 5V dual-coil latching relay. But we use a regular DPDT relay with a self-sealing contact arrangement.



Special use of GPIO26 (pin 37), generates low-hi step about 15 sec after HALT command. Must edit config.txt. Display “shutting down” message on LCD before halt.

**Software Architecture Notes**

Start with my standard Pi app.

Include TCP GUI.

Queue and send buffers with 1 Hz heartbeat. Limit queue size.

Calibration entry via GUI

Has to reflect current values, then allow updates without race cond

Settings stored on Pi in .ini file. Path: /home/pi/filename.ini

Simple way to change sensors: FP switch

Position indicated on GUIs

How to change sensors: Depends on how different.

Calibration coeffs and/or linearization algo

Behavior of FP buttons could change

Different control sigs to sensor

At the moment, only similar sensors.

Select SubVIs for calibration

Nothing else... So FP switch would be easy.

Display GUI items:

Sensor selected

Numeric values (large) 4 with units

Stripchart(s)

Reboot button

Shutdown button

External atten setting

Enter value to add to readings.

Since it's an encoder, value can also be entered numerically into FP/GUI.

Encoder has push switch to enable/disable ext atten.

Encoder: Grayhill 62SG22-M5-060C, 24 detents with momentary PB, 6" cable with conn.

**LCD Contents**

4-lline version

±xx.xxdBm ±xx.xxdBV

xxx.xxmW xxx.xxmV

Sens Atten OFF \*REL\*

Ext Atten 31.2dB OFF

12345678901234567890

At shutdown, write "Shutting down..." It's >10 sec til actual power off.

On Temp\_Hi alarm, overwrite line 3

Benchmark:

(Default) 9600 baud, write 4x20 chars: 96 ms

57,600: 24 ms <-- using this

115,200: 8 ms

**TCP Commands to Controller**

Make these text for compatibility with other apps. Send one of these commands per packet.

REBOOT Reboot the OS

SHUTDOWN Shut down CPU

HEARTBEAT Keep-alive msg. No parameters.

INTERVAL <integer> Status xmit interval, ms. Range: 10-1000 ms.

EXTATTENSW (toggle) Use external attenuator value

EXTATTENVAL <integer> External attenuator value. Divide by 10 for dB.

SENSATTEN (toggle) Use sensor attenuator (= Sensor control bit 0)

REL 1 | 0 Relative readings, 1 = on.

SENSCTLB 1 | 0 Sensor control bit 1, 1 = high

CAL <float> \*3 Calibration, applies to currently-selected sensor.

Offset, slope, atten

**TCP Status Report from Controller**

Make these text for compatibility with other apps. Sent as one packet. Lines terminated with \r. Send periodically. Interval is user-adjustable.

COUNT <integer> Msg counter

DBM <float> Power in dBm

DBV <float> Power in dBV

WATTS <float> Power in Watts

DVOLTS <float> "Power" in Volts

ADCVOLTS <float> Raw ADC Volts

SENSORSEL <integer> Sensor selected (0-3)

EXTATTENSW 1 | 0 Using external attenuator value

EXTATTENVAL <integer> External attenuator value. Divide by 10 for dB.

SENSATTEN 1 | 0 Sensor attenuator, 1 = on. (= Sensor control bit 0)

REL 1 | 0 Relative readings, 1 = on.

SENSTATA 1 | 0 Sensor status bit 0

SENSTATB 1 | 0 Sensor status bit 1

SENSCTLB 1 | 0 Sensor control bit 1, 1 = high

CAL <float> \*3 Calibration, applies to currently-selected sensor.

Offset, slope, atten

**RELATIVE Mode Notes**

For dBm, save dBm value at transition then subtract from real-time readings.

For relative Watts, save Watts value at transition then subtract from real-time readings.

Display units as dB and Watts-Relative

Don't bother with Volts, blank that.

**Sensor Board Notes**

Board dims: 2.4 x 1.1. .063 thick. 4 layers.

HASL $7, ENIG $21.30. Shipping $19.05. Total $40.85. 4-day fab, 4-5 days DHL.

JLCPCB, 4-layer impedance calc

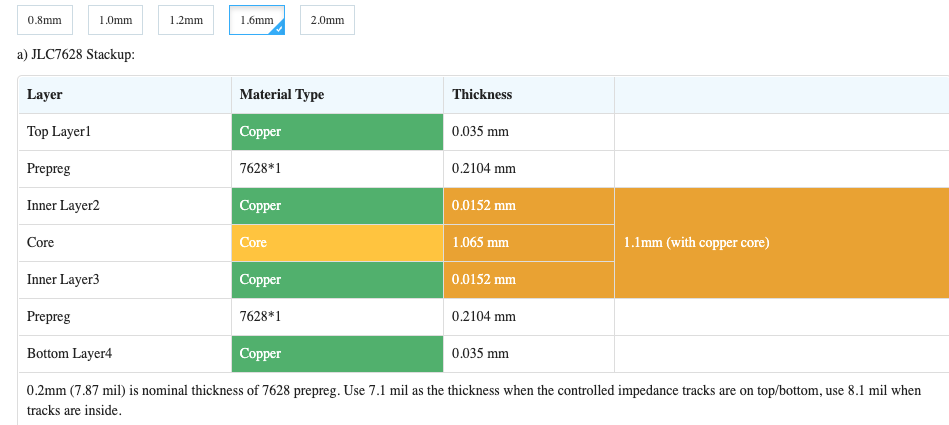
Type JLC7628

For 50 ohms, use 11.55 mil. I used 12.

Min via 0.2mm = 7.8 mil dia

Prepregs are 0.2104 mm = 8.28 mil

Core is 1.065 mm = 41.9 mil



**Enclosure**

Aluminum billet with big pocket.

Board hangs by BNC, plus 2 heatsink screws plus 1 RJ45 screw.

Some side clearance.

Angle board corners to clear inside radius.

Cover screwed in 4 corners.

Back cover (for RJ45) screwed on. Otherwise, can't install board!

Mount board on Wurth 0.2mm thermal pad material to avoid shorts with vias.

Cut relief pockets for thru-hole parts.