Laboratory RF Power Meter

Gary Johnson, NA6O
October, 2023 (V2)

Modern demodulating logarithmic amplifier ICs [1] make it easy to build a versatile wideband RF power meter. They offer wide dynamic range (>90 dB) and reasonable accuracy (±0.4 dB, or about 10%), making them suitable for hobbyist use. I'd previously built a very small battery-powered one based on an AD8307 [2] similar to many other designs published in the ham literature, and it has been very handy especially in the field. For bench use, my goal was to roughly emulate the basic features of commercial power meters including interchangeable external sensors, multiple displays with various scales, and computer control. By changing sensors while sharing a common data acquisition and control unit, it's possible to cover a wider frequency and/or power range, and even consider using other sensing methods sometime in the future. All of this is achieved at a tiny fraction of the cost of commercial equipment.

Controller

My controller is based on a Raspberry Pi 3B+ housed in a 7x 6 x 3 inch enclosure. The display is a 4x20 character LCD with a serial interface (Fig. 1). Analog and digital I/O is supported by a custom interface board (a "Pi hat") built on pre-etched prototyping board (Fig. 2). For data acquisition, an ADS1115 ADC provides 16 bit resolution, a differential input, and a sample rate up to 860 Hz. Since my focus is on CW measurements, a slower 16 Hz rate was selected which automatically includes digital filtering (courtesy of the delta-sigma modulator in the ADC) that is very effective at rejecting noise well beyond the audio range. That makes analog anti-alias filtering easier. A much faster ADC could have been chosen to support peak detection.

Front-panel controls include:

- Switch to select between multiple sensors. Calibration parameters are instantly loaded.
- Rotary encoder to set the value of an arbitrary external attenuator (push to enable).
- Button to switch to Relative mode.
- Button to enable an attenuator built into the sensor.

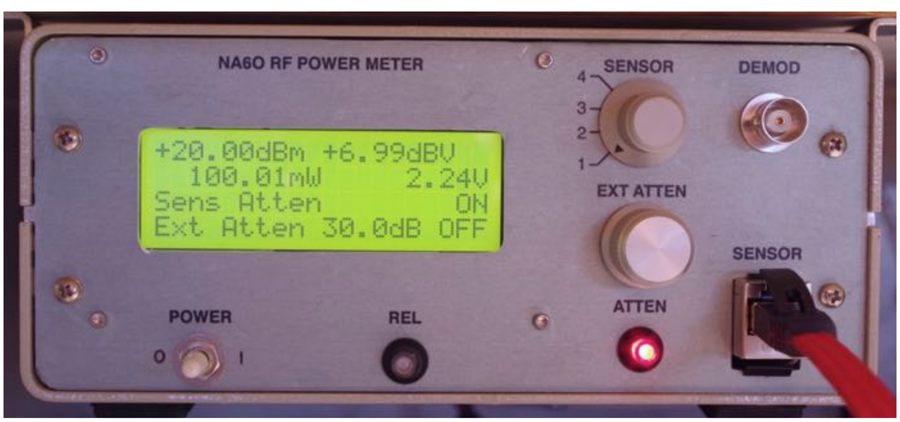


Figure 1. Front panel of the power meter controller.

To make a Raspberry Pi-based system turn on and off gracefully, a couple of tricks are required. At startup, a power relay, arranged in a latching circuit, is enable by the momentary toggle switch on the front panel. At shutdown, the same switch is read by a digital input, which initiates a Shutdown command to the operating system. A special driver was installed (*gpio-poweroff*) [3] that produces a pulse on a digital output after the system has halted. That pulse resets the relay, removing power entirely.

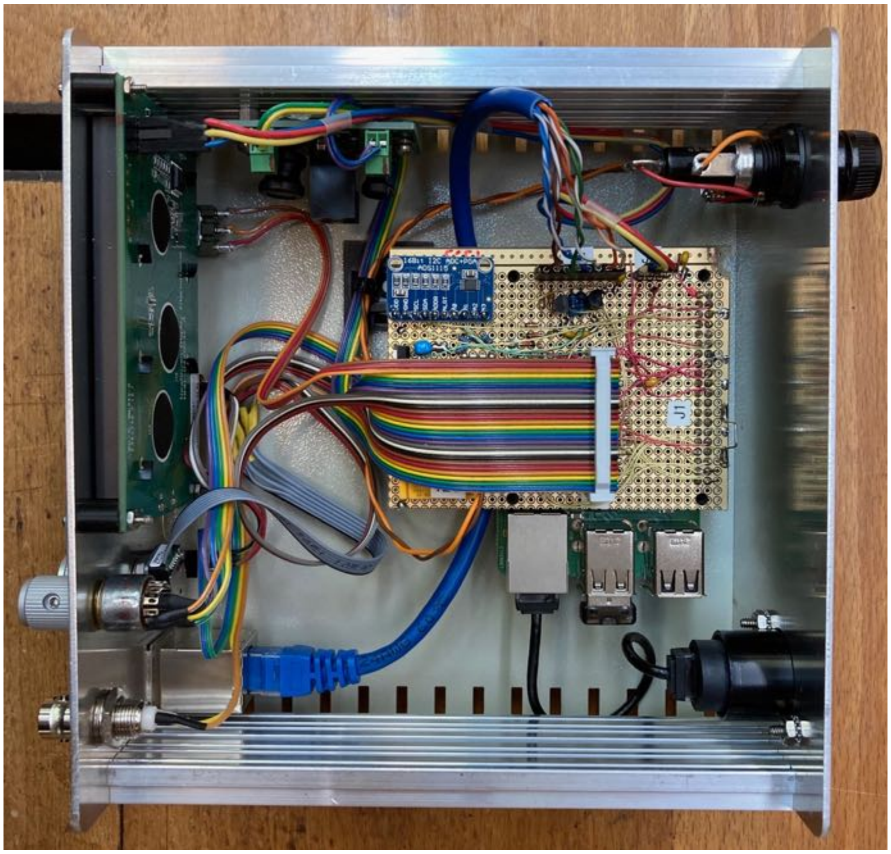


Figure 2. Inside the controller.

Since the sensor is a low-bandwidth device, it connects via a standard CAT 7 shielded RJ45 cable. This is remarkably good cable for instrumentation. It consists of four individual twisted shielded pairs with an overall shield that is connected to the RJ45 connector body shield. By properly terminating the shield on the *outside* of each enclosure, all EMI energy is kept out of the equipment. This avoids the well-known "Pin 1 Problem" [4] where the shield is incorrectly terminated inside the enclosure.

Demodulating amplifiers provide an output that not only represents the average input power, but simultaneously gives you a higher-bandwidth signal that can be viewed on an oscilloscope. In this way, the data acquisition displays the average while the 'scope shows the dynamics. A BNC connector provides that raw demodulated output. I've found it very useful when looking at modulated or unstable signal sources. Because the signal is logarithmic, the full dynamic range of the sensor is visible, allowing you to see low-level residual noise on a keyed or modulated signal.

Displayed values are in dBm, Watts, dBV, and Volts. When an attenuator is activated, its value is accounted for in all readings. For years I had to do the math for attenuation and units conversion... At last, my power meter does it for me.

Choice of programming language is usually driven by your personal experience and expertise. In my case, I have decades of experience with LabVIEW, a graphical programming language that was in fact conceived for instrumentation applications. I used the free LabVIEW Community Edition [5] which supports the Raspberry Pi as a real-time target. I have an existing code base that has been used for many applications and includes analog and digital I/O, LCD display, and TCP/IP based communications for external user interfaces (Fig. 3). The user interface was also written in LabVIEW and can run on any platform. It includes a strip chart and access to all calibration factors. Additional features can be added easily when needed (Fig. 4).

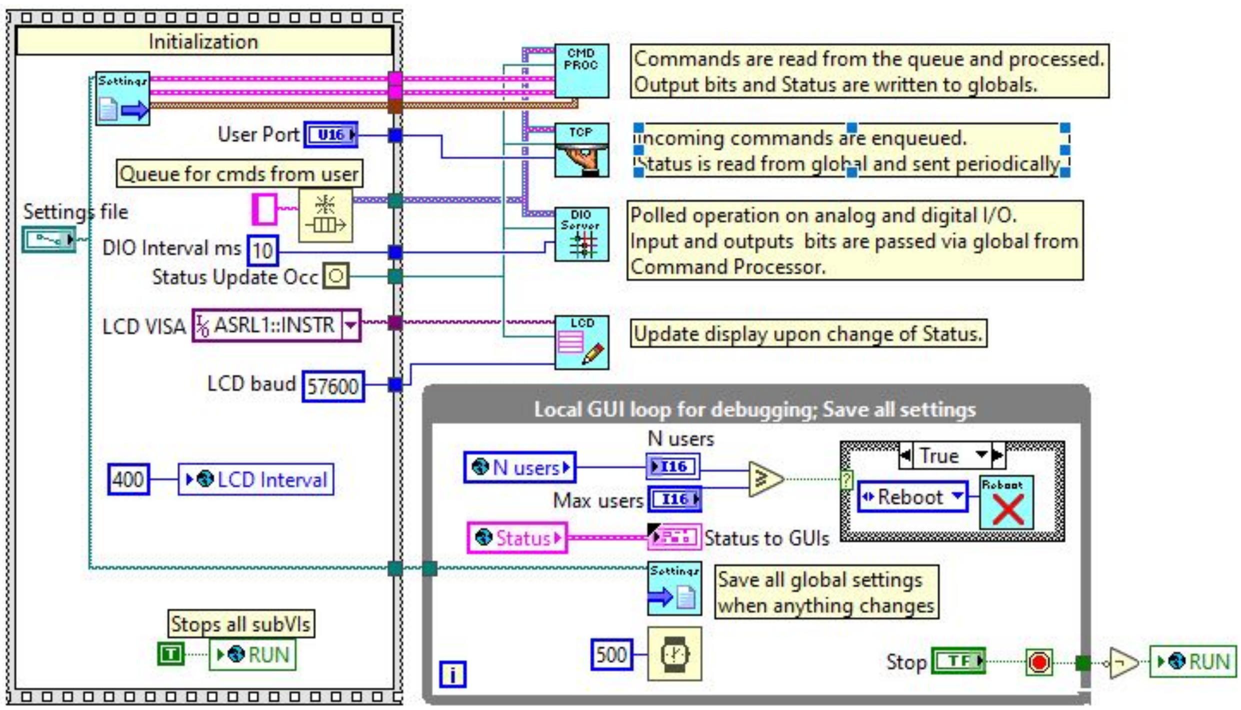


Figure 3. Main LabVIEW diagram of the embedded controller. There are five tasks executing in parallel, simply by being on the diagram. Try that with any other language...

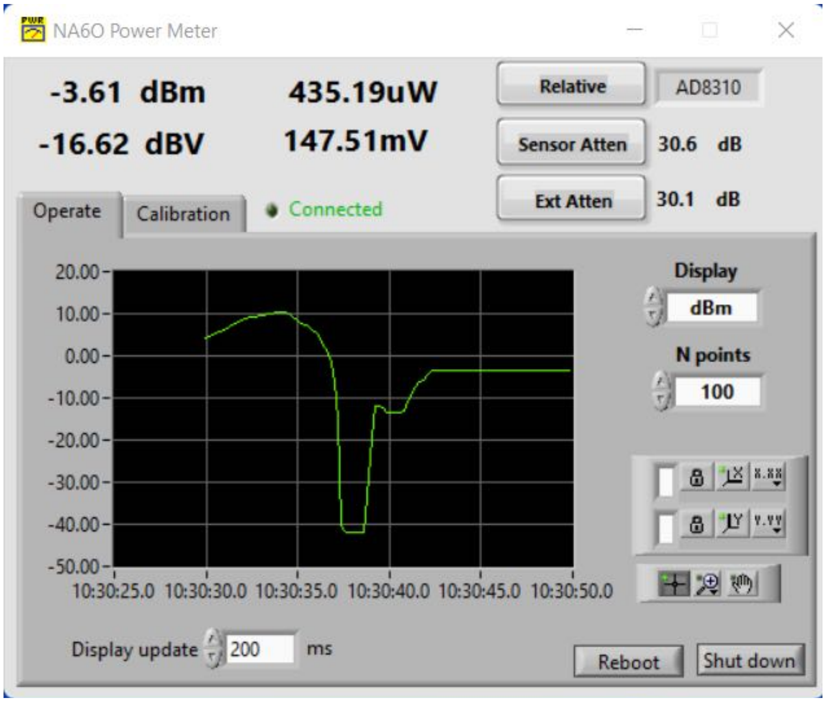


Figure 4. User interface, written in LabVIEW.

RF Sensor #1

The first sensor I built is based on an AD8310 (Fig. 5). Usable from audio to at least 440 MHz, it has a 95 dB dynamic range and typical error over all frequencies and amplitudes of ±0.4 dB. All that in a tiny 8-lead package that costs $13. Clearly this is an enabling technology for low-cost instrumentation! At the moment, this meets my needs, though I may also build one using an AD8363 that operates into the microwave region. Also, it's important to note that these log amps respond to *average envelope voltage*, not true RMS power like thermal sensors. For non-sinusoidal waveforms and modulated signals, it's an important difference. That could be motivation for an alternative sensor as well.

Figure 5. The completed RF sensor #1.

To expand the maximum power handling capability, an onboard wideband 30 dB, 10 W attenuator can be inserted via a small RF relay. A temperature switch IC indicates overheating. It all runs on 5 VDC at 10 mA. The input is a BNC connector and a shielded RJ45 connects to the controller. The custom PC board was laid out in CircuitMaker (a community version of Altium Designer). I used four layers (three are ground planes) to achieve 50 Ohm microstrips with narrow lines, and also to improve heat spreading for the high-power attenuator.

Frequency response of the AD8310, like all of its relatives, rolls off with a first-order lowpass response. This can be accurately compensated with an equalizer consisting first-order highpass filter as proposed by W7ZOI [7]. A small inductor is also added to cancel the input capacitance of the sensor, extending the frequency response to nearly 500 MHz. I used a trimmer capacitor to allow fine-tuning of the response up to about 350 MHz. Above that, the inductor must be changed.

A convenient way to assess frequency response is to use an RF sweep generator while observing the demodulated output on an oscilloscope. I used the signal generator output of my TinySA Ultra that unfortunately generates transients at each step. These were filtered out by adding a 20 Hz RC lowpass filter at the oscilloscope input. Sweeping from zero to 1 GHz or a bit less gives a complete picture of the system response while allowing you to adjust the trimmer capacitor for maximum flatness. Figure 6 shows the frequency response. The raw data is fairly noisy due to limitations of the oscilloscope's resolution plus residual noise from the sweep generator steps. Digital smoothing improves it but still leave a lot of aritificial ripples. A 3 dB attenuator improves the input match at high frequencies (Fig. 7).

The enclosure was machined from a block of aluminum (Fig. 8) and the board is held in place by its BNC connector and small screws. Phase-change thermal interface material is placed under the board to reduce thermal resistance to the enclosure, which serves as a heatsink. The final product is a very rugged little brick. Specifications appear in Table 1.

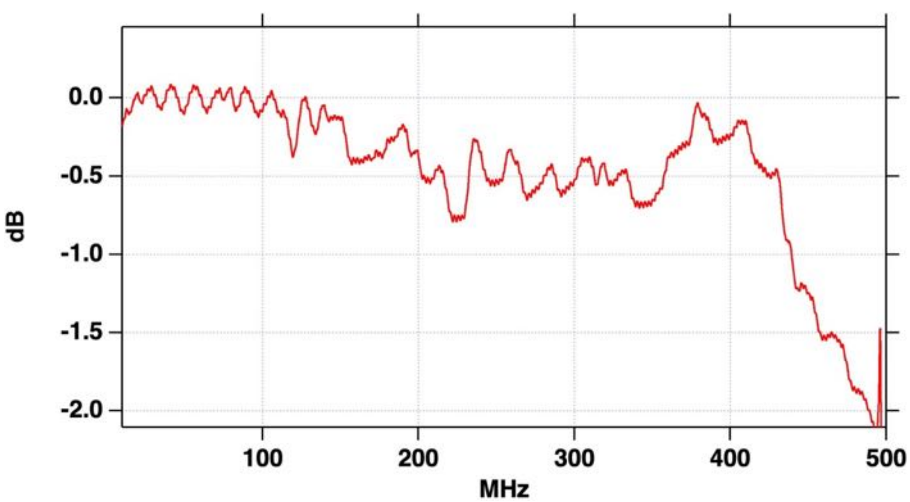


Figure 6. Frequency response after equalization. Ripples are artifacts of sweep generator transients and limitation of oscilloscope resolution.

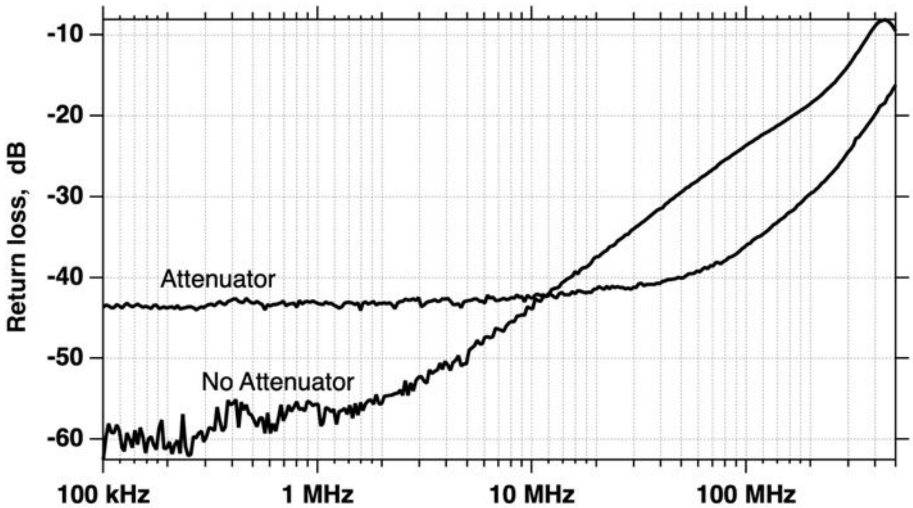


Figure 7. Input return loss. Activating the 30 dB attenuator improves the match at the highest frequencies.

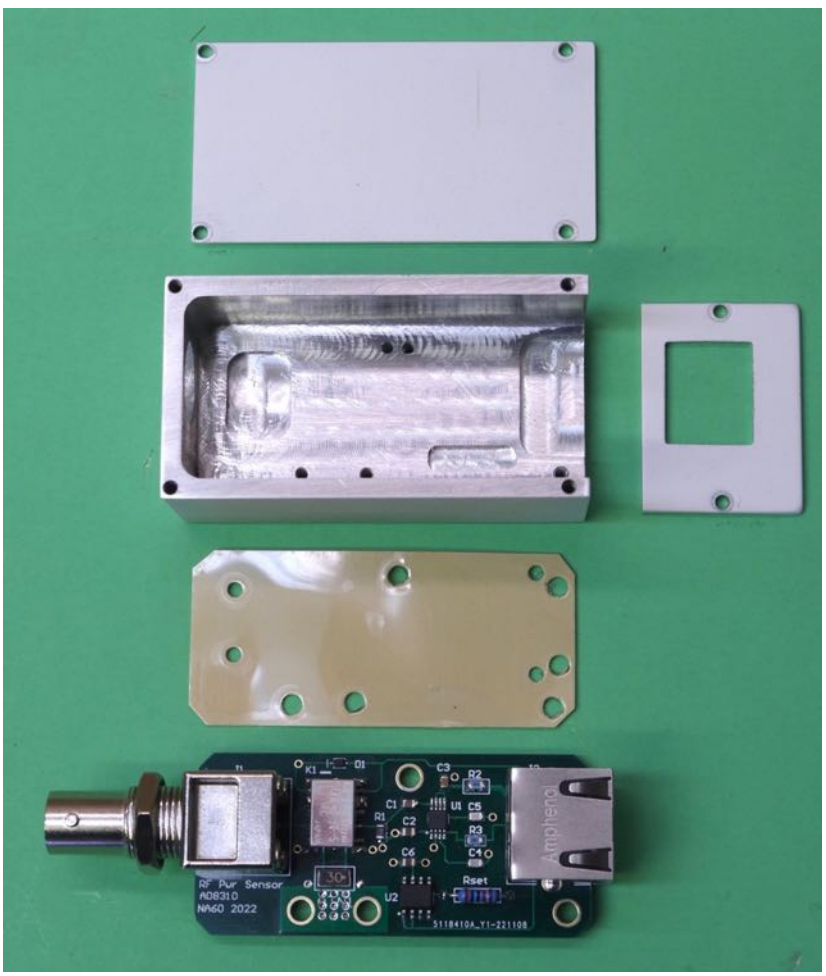


Figure 8. Parts of RF sensor #1. The tan material near the bottom is the phase-change thermal interface material.

Table 1. Sensor #1 Specifications

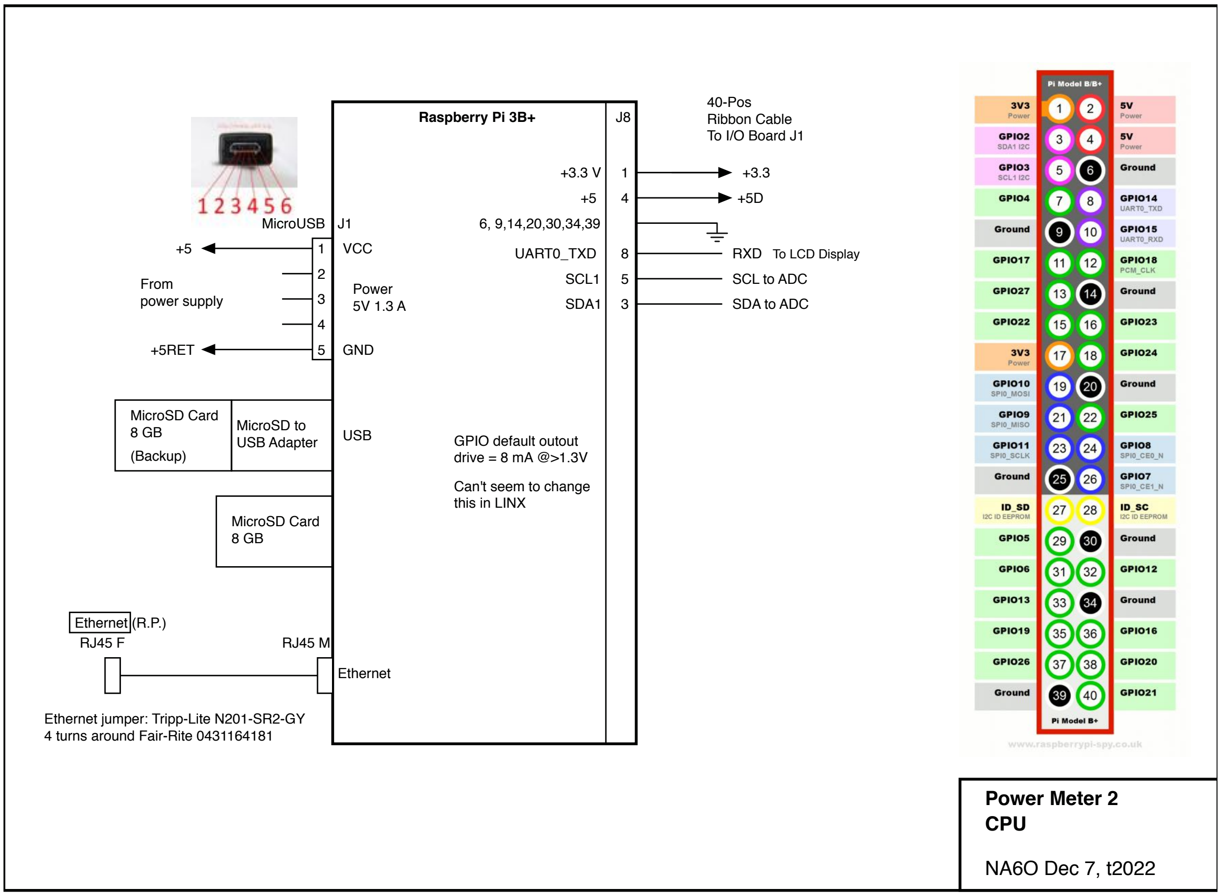
Input impedance	50 Ω, see return loss graphs
Bandwidth (±0.4 dB)	50 kHz - 440 MHz
Frequency Response	50 kHz - 450 MHz ±0.4 dB
Input power, direct	-70 to +24 dBm
Input power, with attenuator	-40 to +40 dBm
Attenuator	30 dB, 10 W
Linearity	±0.4 dB
Scale temperature coefficient	-0.04 dB/°C
Demodulated output scale	23.7 mV/dB
Demodulated output bandwidth	50 kHz, Tr = 7.5 us
Temperature alarm	70°C
Power supply	5V, 10 mA

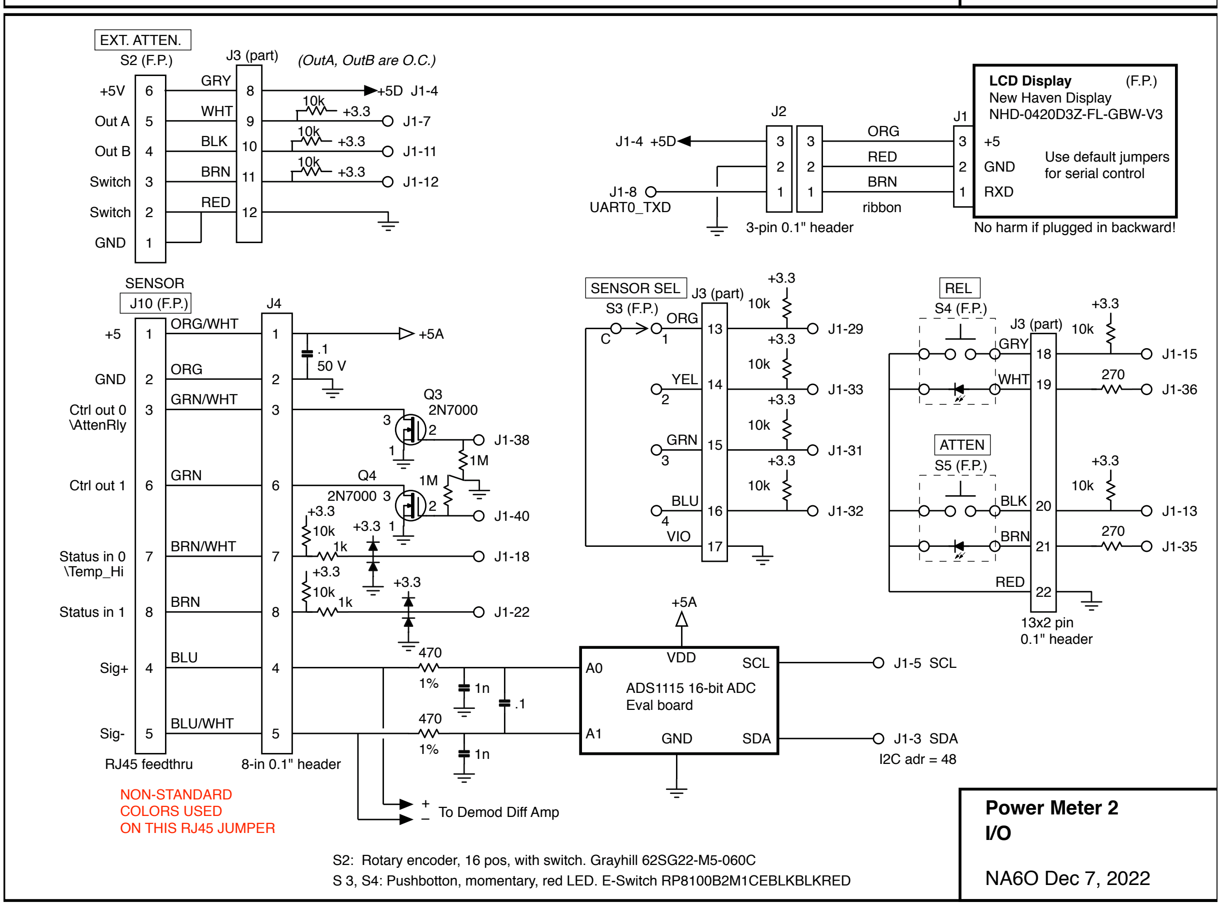
Conclusion

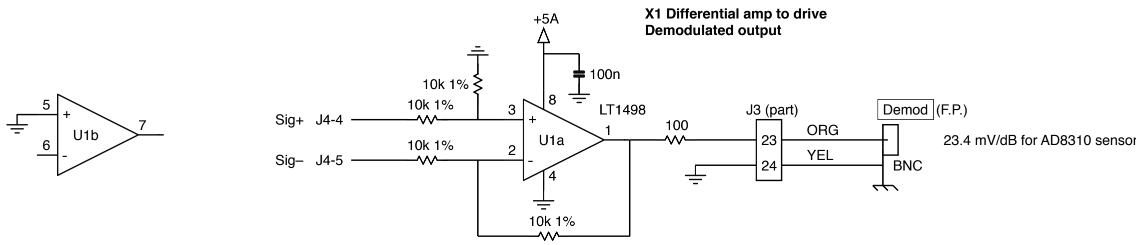
I'm very pleased with the final product. This one of many pieces of test equipment that I've built in my lifetime. For some reason, I was drawn to instrumentation design and application from a very early age, and it become the core of my long career as an engineer and craftsman. Building custom test equipment is personally satisfying, we end up with something that does exactly what we want, and at a price we can afford.

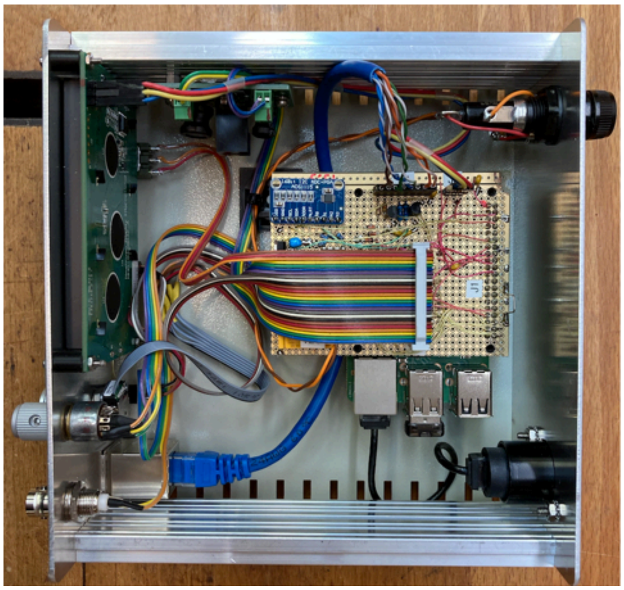
References

1. Logarithmic Amplifiers Explained (Analog Devices). https://www.analog.com/en/analog-dialogue/articles/logarithmic-amplifiers-explained.html
2. Miniature RF power meter. https://na6o.com/main/Electronics_files/WB9JPS+RF+Power+Meter.pdf
3. Discussion on how to properly halt and power-down a Raspberry Pi. https://raspberrypi.stackexchange.com/questions/89297/how-long-to-wait-after-shutdown-to-cut-power/130443#130443
4. The Pin 1 Problem. http://pin1problem.com/
5. LabVIEW Community Edition. https://www.ni.com/en-us/shop/labview/select-edition/labview-community-edition.html
6. CircuitMaker PCB design software. https://www.altium.com/circuitmaker/download/b
7. Hayward, Wes, W7ZOI, "Simple RF Power Measurement." *QST*, June, 2001.


Raspberry Pi 3B+
J8
40-Pos
Ribbon Cable
To I/O Board J1
+3.3 V 1 +3.3
+5 4 +5D
6, 9,14,20,30,34,39
UART0_TXD 8 RXD To LCD Display
SCL1 5 SCL to ADC
SDA1 3 SDA to ADC
1 2 3 4 5 6
MicroUSB J1
+5
1 VCC
2
From
power supply
3
4
Power
5V 1.3 A
+5RET
5 GND
MicroSD Card
8 GB
(Backup)
MicroSD to
USB Adapter
USB
GPIO default outout
drive = 8 mA @>1.3V
Can't seem to change
this in LINX
MicroSD Card
8 GB
Ethernet (R.P.)
RJ45 F
RJ45 M
Ethernet
Ethernet jumper: Tripp-Lite N201-SR2-GY
4 turns around Fair-Rite 0431164181
Pi Model B/B+
3V3 Power
5V Power
GPIO2 SDA1 I2C
5V Power
GPIO3 SCL1 I2C
Ground
GPIO4
GPIO14 UART0_TXD
Ground
GPIO15 UART0_RXD
GPIO17
GPIO18 PCM_CLK
GPIO27
Ground
GPIO22
GPIO23
3V3 Power
GPIO24
GPIO10 SPI0_MOSI
Ground
GPIO9 SPI0_MISO
GPIO25
GPIO11 SPI0_SCLK
GPIO8 SPI0_CE0_N
Ground
GPIO7 SPI0_CE1_N
ID_SD I2C ID EEPROM
ID_SC I2C ID EEPROM
GPIO5
Ground
GPIO6
GPIO12
GPIO13
Ground
GPIO19
GPIO16
GPIO26
GPIO20
Ground
GPIO21
Pi Model B+
Power Meter 2
CPU
NA6O Dec 7, t2022

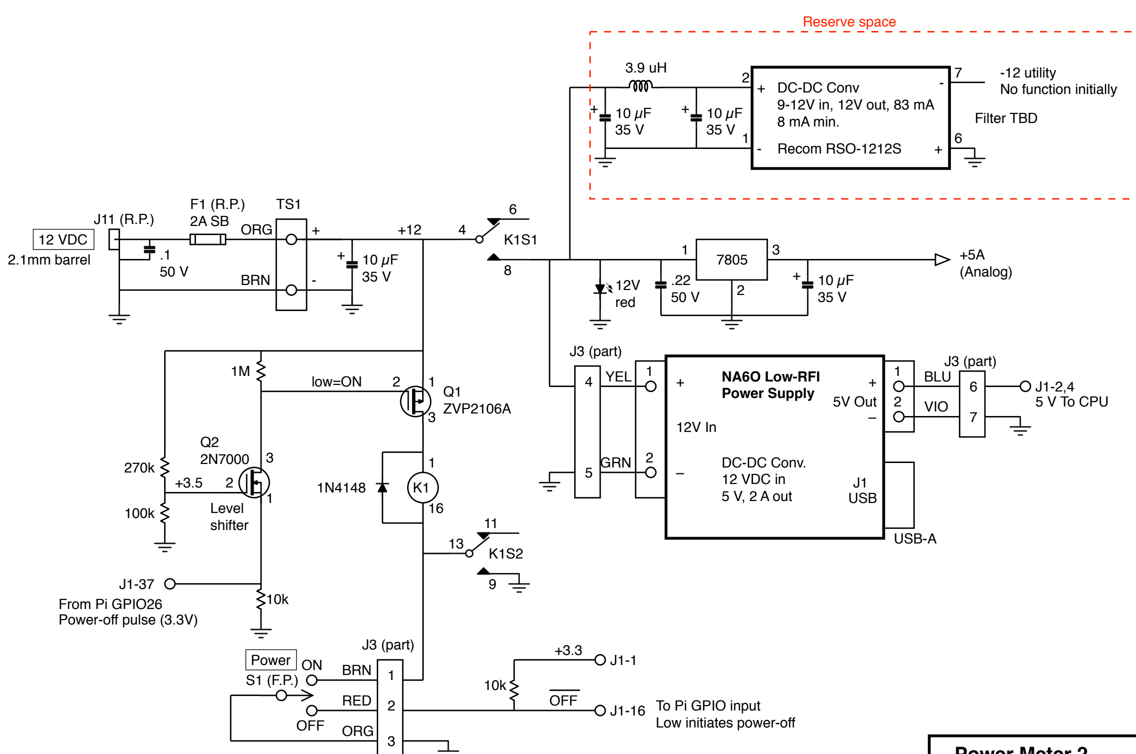

EXT. ATTEN.
S2 (F.P.)
J3 (part)
(OutA, OutB are O.C.)
+5V 6 GRY 8 +5D J1-4
Out A 5 WHT 9 10k +3.3 J1-7
Out B 4 BLK 10 10k +3.3 J1-11
Switch 3 BRN 11 10k +3.3 J1-12
Switch 2 RED 12
GND 1
LCD Display (F.P.)
New Haven Display
NHD-0420D3Z-FL-GBW-V3
J2
J1
J1-4 +5D
ORG
RED
BRN
J1-8
UART0_TXD
ribbon
3-pin 0.1" header
+5
GND
RXD
Use default jumpers
for serial control
No harm if plugged in backward!
SENSOR
J10 (F.P.)
J4
+5 1 ORG/WHT 1 +5A
.1
50 V
GND 2 ORG 2
Ctrl out 0
\AttenRly
3 GRN/WHT 3
Q3
2N7000
J1-38
1M
Ctrl out 1 6 GRN 6
Q4
2N7000
1M
J1-40
+3.3
10k
1k
Status in 0
\Temp_Hi
7 BRN/WHT 7 J1-18
Status in 1 8 BRN 8 J1-22
Sig+ 4 BLU 4
470
1%
1n
.1
Sig- 5 BLU/WHT 5
470
1%
1n
RJ45 feedthru
8-in 0.1" header
NON-STANDARD
COLORS USED
ON THIS RJ45 JUMPER
+ To Demod Diff Amp
−
SENSOR SEL
S3 (F.P.)
J3 (part)
+3.3
10k
ORG 13 J1-29
YEL 14 J1-33
GRN 15 J1-31
BLU 16 J1-32
VIO 17
REL
S4 (F.P.)
J3 (part)
GRY 18 J1-15
WHT 19 270 J1-36
ATTEN
S5 (F.P.)
BLK 20 J1-13
BRN 21 270 J1-35
RED 22
13x2 pin
0.1" header
+5A
A0 VDD SCL
ADS1115 16-bit ADC
Eval board
A1 GND SDA
J1-5 SCL
J1-3 SDA
I2C adr = 48
S2: Rotary encoder, 16 pos, with switch. Grayhill 62SG22-M5-060C
S 3, S4: Pushbotton, momentary, red LED. E-Switch RP8100B2M1CEBLKBLKRED
Power Meter 2
I/O
NA6O Dec 7, 2022





Power Meter 2
Diff Amp

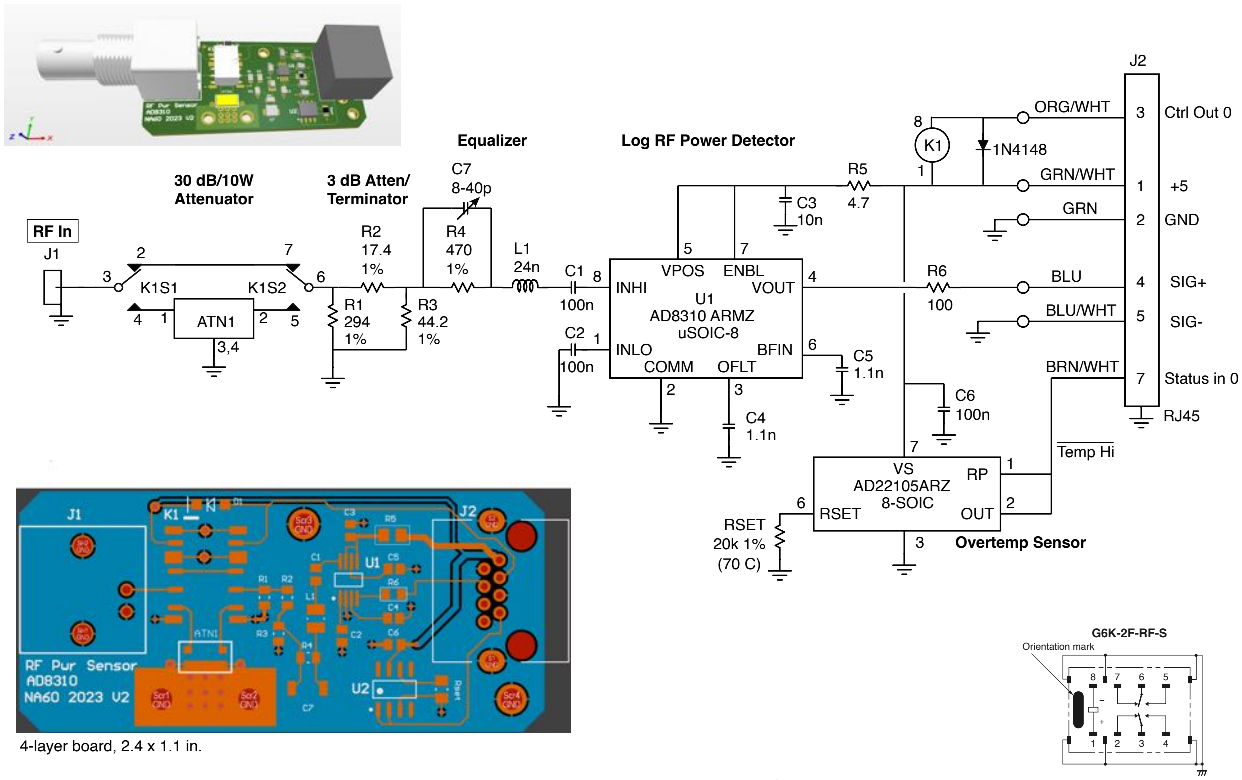
NA6O Dec 7, 2022



Power Meter 2
Power Supply

NA6O Dec 7, 2022

K1: Pansonic DS2E-S-DC12V, DPDT 12V
S1: NKK toggle, SPDT (on)-off-(on) M2018SD1W01



4-layer board, 2.4 x 1.1 in.

Demod BW = 1/2pi*3k*C3
Set to ~50 kHz. C = 1.06 nF

Offset comp loop freq = 1/2pi*2.6k*C4
Set to ~60 kHz

$$R_{SET} = \frac{39M\Omega\ °C}{T_{SET}\ (°C) + 281.6°C} - 90.3k\Omega$$

All passives 0805.
J1 BNC right angle. TE 227677-1
J2 RJ45, shielded. Amphenol RJHSE-5380
ATN1 Attenuator, 30 dB, 10W, Anaren D10AA30Z4
K1 Relay, DPDT, RF, 5V Omron G6K-2F-RF-S-TR03 DC5

RF Power Sensor #1, AD8313, V2

NA6O 10-16-23

AD8310 RF Power Sensor Parts List   NA6O Oct 2023

All parts available at Mouser

Designator	Description	Manufacturer P/N
ATN1	RF attenuator, 30 dB, 10 W, SMT	Anaren D10AA30Z4
C1, C2, C6	CAP CER 0.1UF 25V 10% X7R 0805	
C3	CAP CER 10000PF 50V X7R 10% 0805	
C4, C5	CAP CER 1000PF 50V 5% X7R 0805	
C7	CAP, trimmer, 8-40PF, SMT	Knowles Voltronics JR400
D1	1N4148, 2-pin SOD-123	Onsemi 1N4148WS
J1	Conn BNC 50 Ohm Solder Thru-Hole	Molex 73138-5033
J2	Modular Jack - Right Angle, Shielded, Without LEDs, Thru-Hole	Amphenol RJHSE-5380
K1	Relay, DPDT, RF	Omron G6K-2F-RF-S-TR03 DC5
L1	Inductor, RF, 24NH, unshielded wirewound 0805	Coilcraft 0805CS-240XJLC
R1	Thick Film Resistors - SMD 1/8W 294 ohm 1%	
R2	Thick Film Resistors - SMD 1/8W 17.4 ohm 1%	
R3	Thick Film Resistors - SMD 1/8W 44.2 ohm 1%	
R4	Thick Film Resistors - SMD 1/8W 470 ohm 1%	
R5	Thick Film Resistors - SMD 1/8W 4.7 ohm 1%	
R6	Thick Film Resistors - SMD 1/8W 100 ohm 1%	
Rset	Thick Film Resistors - SMD 1/8W 20K ohm 1%	
U1	AD8310ARM DC to 440 MHz, 95 dB Logarithmic Amplifier, 8-pin MSOP	AD8310ARM
U2	AD22105ARZ Resistor Programmable Thermostatic Switch, 8-pin SOIC	AD22105ARZ