Remote Wattmeter with Band Decoder

By Mike Lewis, K7MDL
March 13, 2024

I wanted a cost-effective means to measure the SWR and power output of my high-power VHF, UHF and SHF amplifiers. Later I need that capability to be placed outside the ham shack near the antennas.

I built a variety of small milli-watt meters, initially on a M5Stack, then PSoc5LP, then Arduino. Each of them leverages some version of a log RF power detector such as the single channel AD8307 or AD8318. Those were usable to 6GHz. Later I discovered the dual channel ADL5519 available from SV1AFN. It is usable to 10GHz. ADL5519 RF Power Detector - sv1afn.com. With 2 channels on the chip, they are built to be accurate relative to each channel and are affected by temperature changes equally. It has an onboard temperature sensor that can be read to apply your own corrections if desired. I use it to display the rough ambient temperature of the unit, thus the outdoor amplifier enclosure that it is placed inside of. The input power range of an ADL5519 is around +5 to -60dBm with linear results closer to 0 to -55dBm. The outputs of the ADL5519 are inverted and with zero signal output is around 2.5VDC. The output decreases linearly to 0.5VDC at max input.

Below is the first version built on a M5Stack CPU module and 2 single channel AD8313 RF log detector modules. Multiple screens are presented for power, SWR and per band calibration.

I use reasonably priced surplus bi-directional couplers chosen for the frequency and power level I needed for each band. Some couplers cover fairly wide frequency ranges, all that is needed is a means to conveniently apply calibration factors. For a KW amplifier a total of 60-70dBm attenuation is required on the forward power channel and about 10dB less on the reflected channel. A typical coupler provides between 10-30dB attenuation so you need to add 10-40dB additional attenuation on the coupler ports. I use standard coaxial SMA or N attenuators.

Below are examples of external bi-directional couplers.

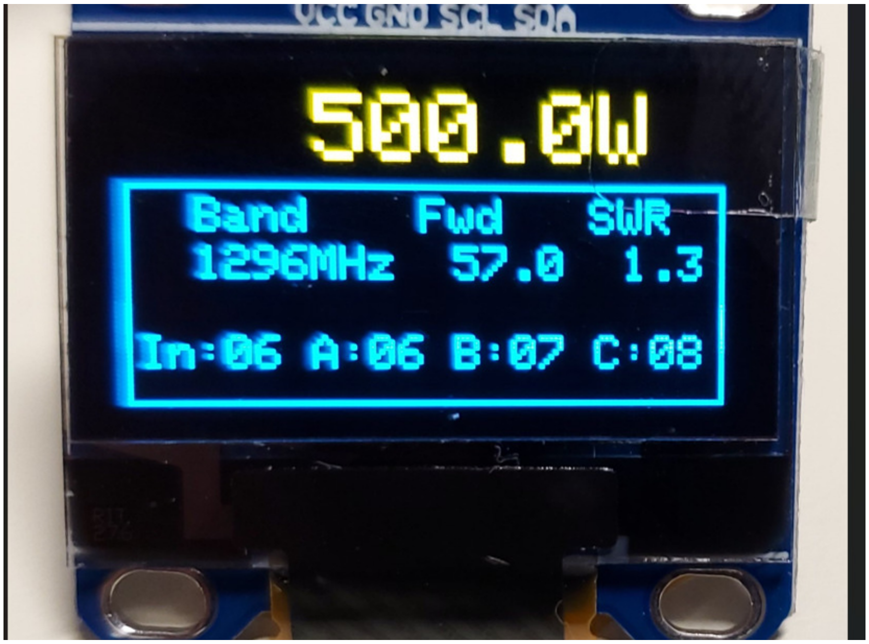
I included band decoding capability so that the correct calibration would be applied on each band change. It was natural to extend this to include programmable outputs to control transverter, LO, coupler selection, and amplifier selection outputs.

The device was initially built for local usage and has the option of a Nextion color touchscreen in 2.8" or 3.5" sizes. I added support for a small OLED display when I embedded a version of this into a 1296 150W RF amplifier. When I needed it located with my power amps outside in outdoor boxes, close to the antennas with high RF levels, I added a headless configuration and ethernet to overcome the USB distance limitation and be more RFI resistant.

The picture below is a local box with a 3.5" Nextion touchscreen. It has multiple pages for calibration, setting alarm thresholds, and to graph power and SWR vs. time. Serial OTRSP is enabled from N1MM+ for band decode input. Also, CW mode is detected. Could be useful later for helping to determine corrections for modulated signals. Currently I calibrate it against a Bird 43 wattmeter.

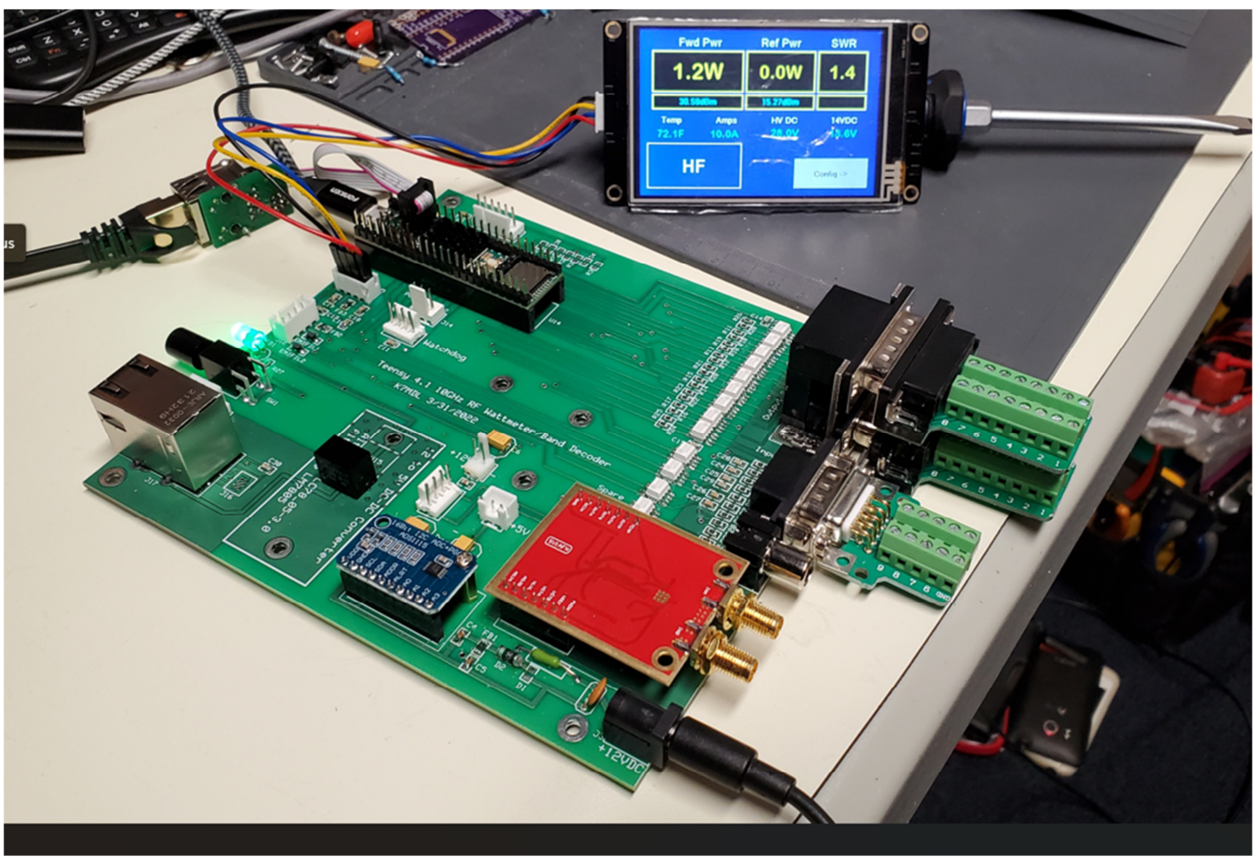
At the time I first built these the Nextion display was a great option for low powered CPUs I started with the like the Arduino Nano or Cypress PSoc5LP. I now use a Teensy 4 that can handle complex SPI connected graphics displays. I use them in my Teensy SDR project. If I were to revisit the code today, I would use a simple graphics controller display. My favorite displays use RA8875 and RA8876 controllers for high performance spectrum drawing but for this project any display will work and would be easier to code up.

Below is a small OLED display. The bottom line is the current band decoder input and output patterns. You can run with no display, either display, or even both displays.

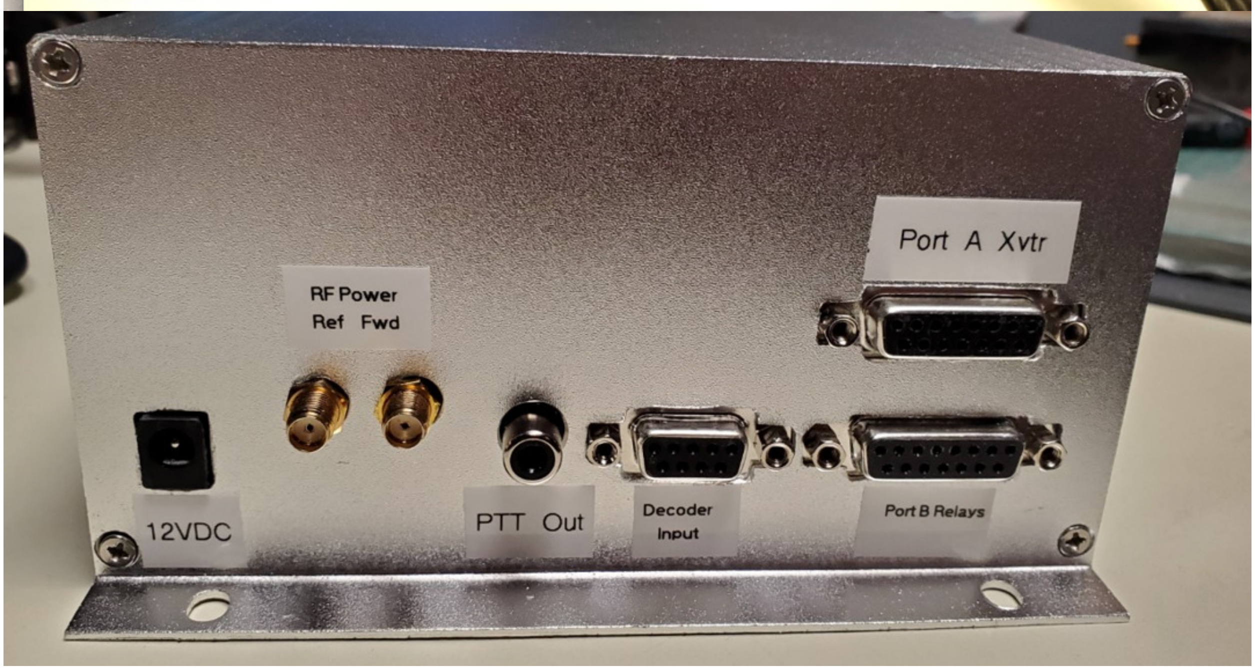


I built the first remote unit using proto boards for the Teensy 4.1 CPU and several ULN2803 buffers in a plastic case with a OLED display. That meant lots of small wire jumpers. When I rotated the antennas toward the unit, high RFI would lock up the CPU and the display was corrupted. I added an independent watchdog reset board to help recover, buying me time to design a more RFI tolerant solution. The Teensy 4.1 offers high speed CPU, lots of IO pins, IC2, and a built-in ethernet controller, just add an optional mag-jack. Also useful for some of my projects is the USB 2 host port. For my Teensy SDR project I use the USB host port to control an Arduino Nano based 5W QRP SDR from Hobby PC, the RS-HFIQ.

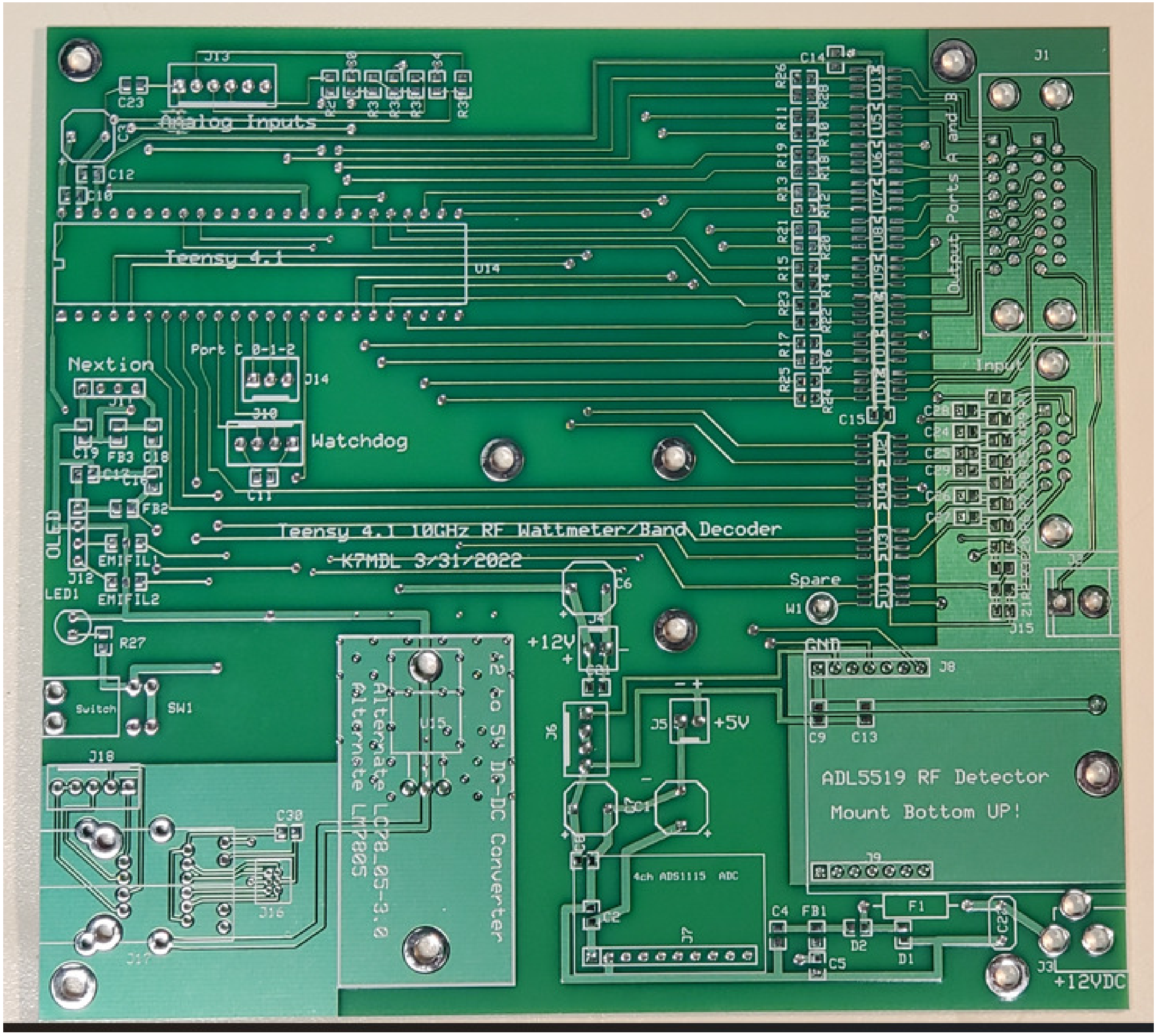
To address the RFI challenge I designed a PCB sized to fit an aluminum case I selected resulting in a board about 4"x5". Big enough to fit several PCB mounted connectors. I used opto-couplers on the band decoder inputs and outputs with the option to float them – i.e. isolate them from ground or power in the box. They were placed close to the connectors to keep the traces very short inside the box. Wired connections to displays, if used, are filtered. 5V is from a DC-DC converter. The ADL5519 plugs in to the PCB as does a very common ADC1115 4-channel 16-bit ADC module. Used in single ended configuration it is good for 15bit resolution and has a decent internal voltage reference. It communicates via the I2C bus. A USB+Ethernet combo jack is used on the front panel along with a power switch and power LED. I replaced the external watchdog board with an internal Teensy watchdog feature though a connector is still provided for an external watchdog board.




K7MDL
10GHz RF Wattmeter
& Band Decoder
RX 0.0W
Ethernet
Power
CPU
USB


Port A Xvtr
RF Power
Ref Fwd
12VDC
PTT Out
Decoder
Input
Port B Relays

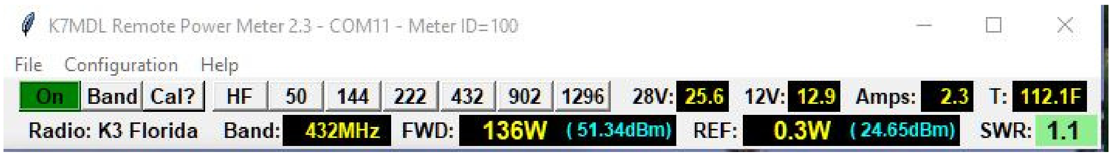
Rev 0.01 PCB



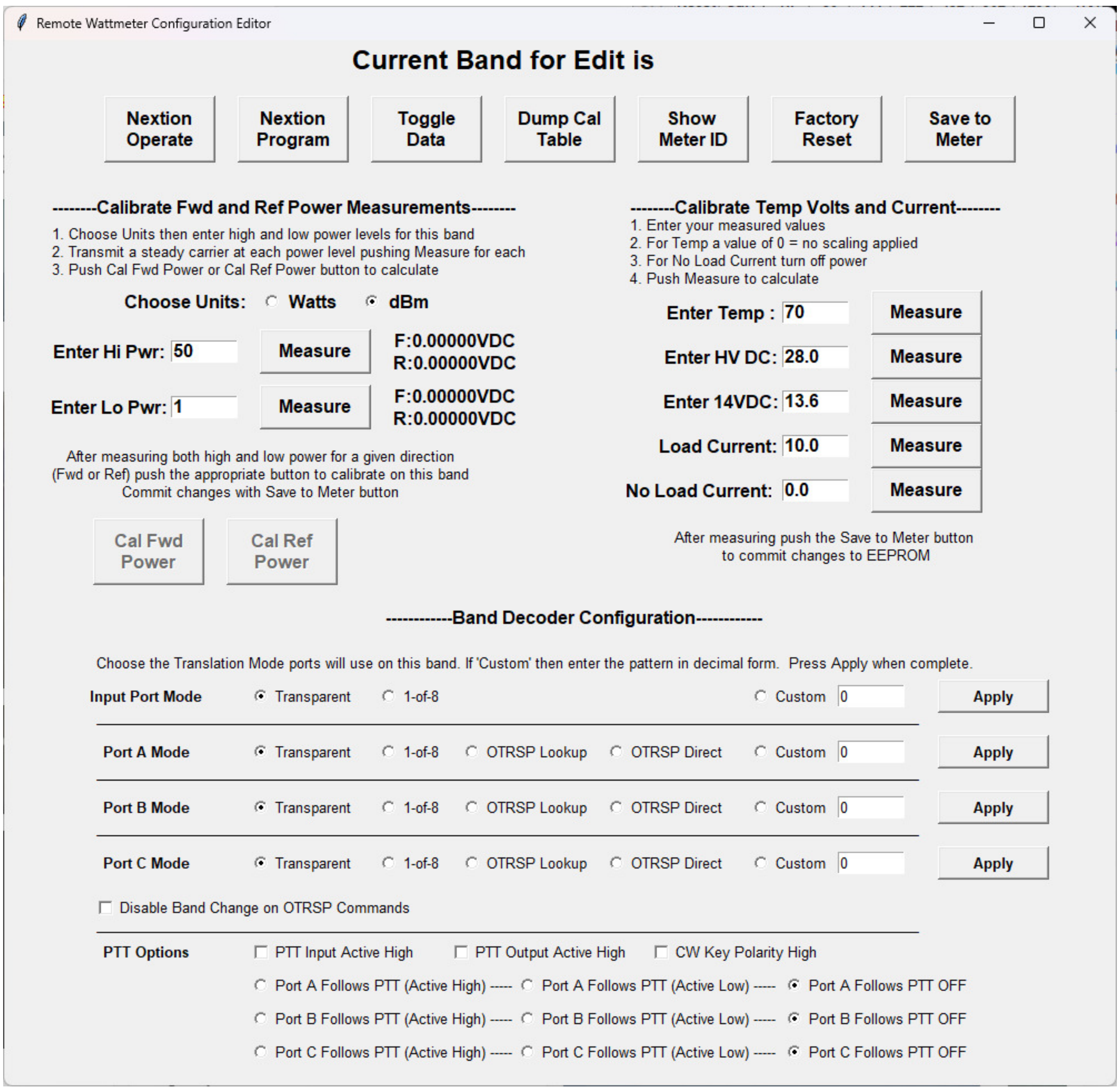
I provided a PTT input and 2 PTT outputs that pass through the CPU and buffer outputs. The current firmware simply relays the PTT signal straight through, but time delay and/or inversion can be applied to act as a sequencer. There are 2 8-bit output ports and 6 input lines. The inputs can be configured to recognize a custom pattern, BCD, or a 1-of-8 pattern. One of the outputs ports can also mask each line with PTT. This permits "routing" PTT signals to a chosen amp or transverter. In my own usage I had a bank of 9 relays. 1 for PTT where all my amps PTT lines were connected in common through diodes. The other relays to controlled antenna select coax switches and a remote power enable signal for each amplifier. Since only 1 amplifier was powered on at any time, the PTT was common in my case.

There is a companion Python desktop app. It can be used to remotely monitor the SWR and forward and reflected power in Watts and dBm, manually select a band, and has a configuration and calibration UI. I also built a Teensy based headless rotator controller. The control panel optionally displays in the desktop UI. Other features were added such as support for OTRSP serial connection to N1MM+. In the beginning I did not have a physical connection to a radio for band selection. Since I am a big WSJT-X user I added a UDP listener to monitor the current band a configured WSJT-X instance is running on.

WSJT-X is connected to the radios so always knows what band it is on even when I am not using any digital modes. Now that I have a physical band decoder input this feature is configured to off. The app can use multicast permitting connection to a logger, WSJT-X and JT_Alert.



Above is the desktop companion App showing the power meter panel only and transmitting on 432MHz. Pressing any band button would change the outputs and related calibration for that band. Useful if you do not have a way to connect a radio to the band decoder direct. Below is the Configuration UI page.



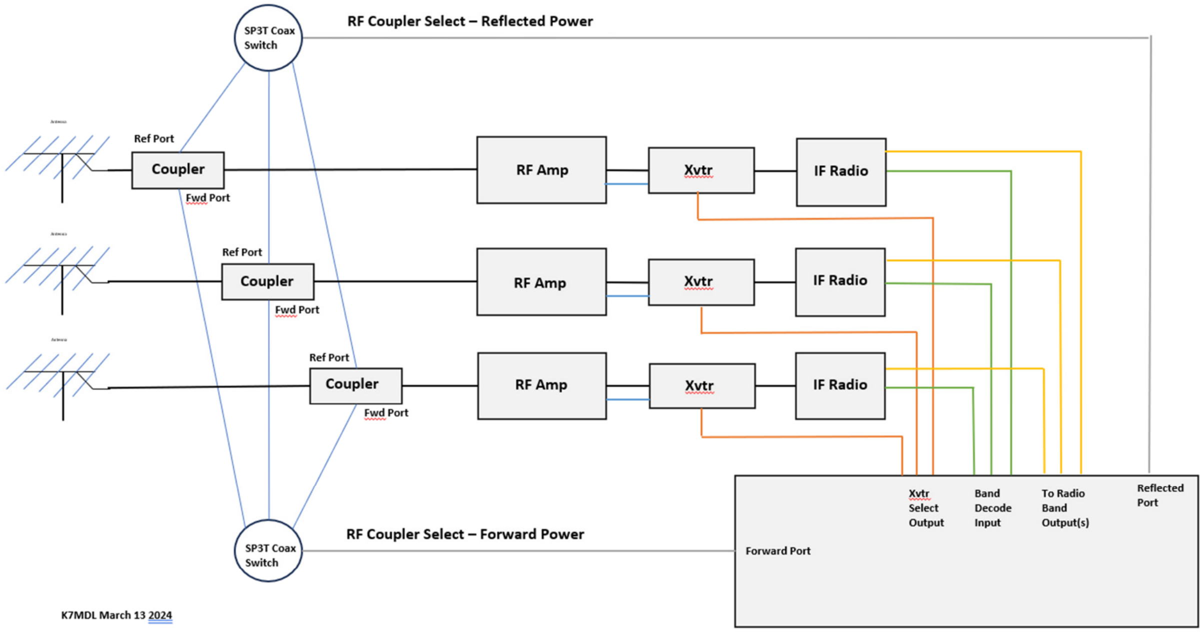
Note in the graphic above the band decoder is very flexible and a port can follow PTT state.   Power is calibrated by simply measuring 2 points with an external wattmeter, one at high power and one low power. At each point you transmit, record the external wattmeter value in the appropriate entry fields. When measuring is done hit Cal Fwd Power or Cal Ref Power button as appropriate to send the values to the remote CPU where it computes the slope and intercept for that band and saves values in the Teensy's (emulated) EEPROM. The Band decode patterns and options are also saved in EEPROM. You can run multiple instances of the desktop app for multiple remote units, each unit configured to a unique meter ID number via command line arguments you can place in a desktop shortcut. IP addresses are set in the Python script variables. If you start the app with no command line arguments, then a small GUI pops up to let you select from a list of USB serial ports found, or UDP, and specify an alternate meter ID number. The default meterID of 100 is good when running only a single meter unit.

The project is open-source. The PCB files, BOM, and Arduino firmware are found at [K7MDL2/RF-Power-Meter-V1: RF wattmeter and Band Decoder on Arduino or PSoC5 modules with Python desktop monitoring and control app. Remote operation via USB Serial or Ethernet (new). (github.com)]

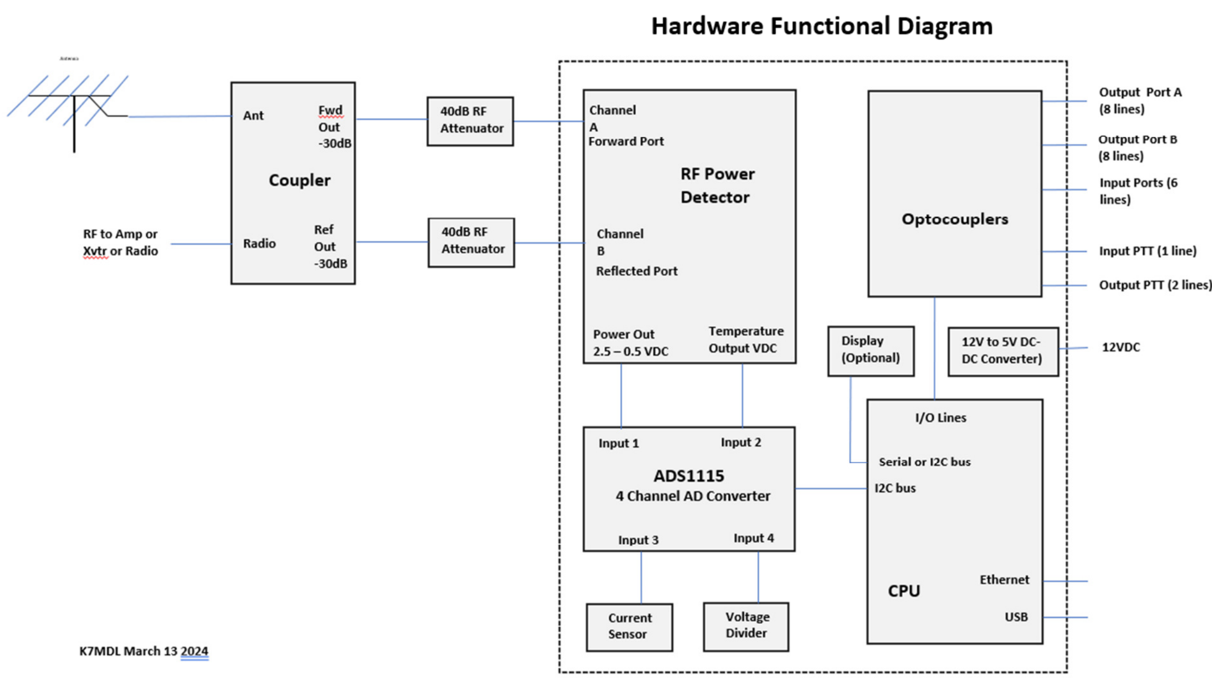
The PCB revision is at 0.04 at the time of this article (March 2024). I built the 0.01 version PCB and had 10 PCBs produced. It has been in use for 2 years now in outdoor cabinets. I made several small improvements to the PCB layout while I built the first board. I have not ordered and built the new version PCB yet. Some of the corrections fix a shorted trace, a missing trace, and I increased the spacing between the back panel connectors so the connector shells can fit properly.

Below is a picture of my home built 1296 amp. I used a small 1" square PSoC5 CPU module driving an OLED display to show Power in Watts and dBm, voltage, and ambient temperature. An ADL5519 is connected to a Narda coupler with attenuators. This version produces a 0-4VDC output corresponding to an SWR of 1:1 to 4:1 and higher. That voltage feeds the High SWR shutdown input on the internal sequencer board.

Possible control connection example



Block Diagram



The voltage divider is connected to the 12V supply but can be connected to outside Voltage source by adjusting the voltage divider appropriately and installing a connector. Temperature is sensed from the RF Power Detector chip. If desired it can be used to apply corrections.

As long as the insertional loss is low, they do not need to be within the exact frequency specs. Calibration will take care of the likely higher than specified coupler attenuation. Size the RF Coupler to the needed power handling. For 1KW 40dB 10W attenuators are a good choice for 20dB couplers, 5W sized for 500W. 30dB couplers can use 30dB 1W attenuators.   The goal is to ensure the maximum RF power at the detector input is 0dBm or less. Software will correct for total attenuation.