

RF Power Measurement Rig



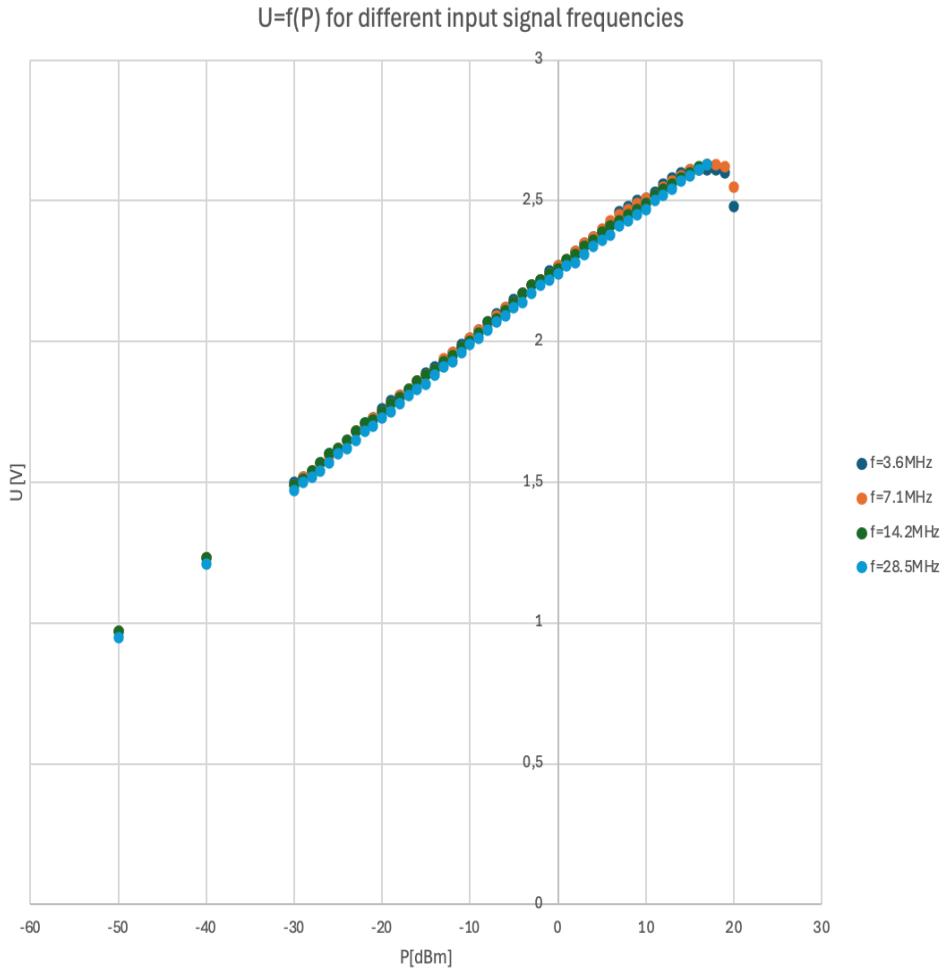
Introduction

The main objective of this project was to utilize capabilities provided by Raspberry Pi Zero platform and its extensive HAT (hardware attached on top) ecosystem to modernize initial RF power meter design published by Wes Hayward W7ZOI and Bob Larkin W7PUA in June 2001 QST Magazine [1]. Similar work has been done in the past by Roger Hayward, KA7EXM [2], Reinhardt Weber, DC5ZM [3] and most recently by Bartosz Krajnik [10] and Mirek Sadowski SP5GNI [4], but all earlier modernizations were based on either PIC microcontrollers or Arduino Nano platform leaving Raspberry Pi gang empty handed.

The core component of all power meters mentioned above is Analog Devices AD83xx logarithmic amplifier, which converts the power level of measured signal to a voltage. AD83xx log amp transfer function describing relation between input power level in dBm and corresponding output voltage in Volts is highly linear and can be expressed as

$$P_{\text{input}}[\text{dBm}] = a * U_{\text{out}}[\text{V}] + b$$

where a and b are respectively linear curve slope and intercept coefficients [5].



Solution Architecture - Key Hardware Modules

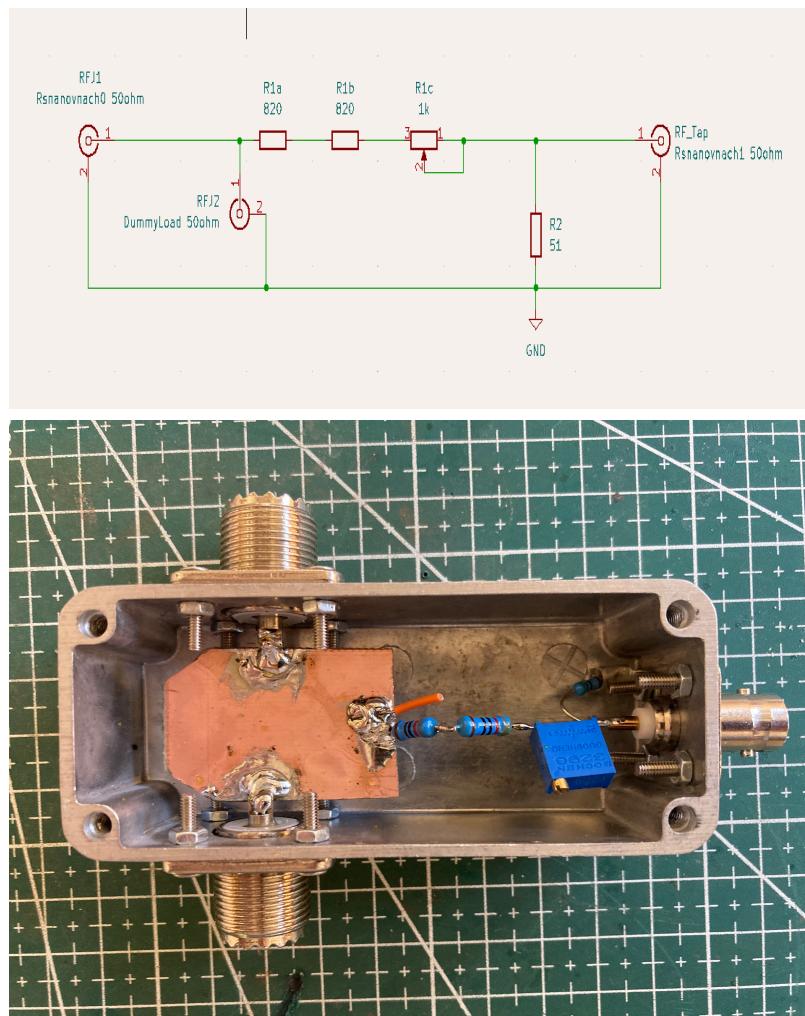
Power Sensor Module Based On AD8307 Amplifier

AD8307 [6] used in this implementation, accepts input signal frequencies from DC to 500 MHz and input signal levels from -95dBm to +17dBm. Readily available AD8307 detector board purchased at Amazon is used for this project (see below)



RF Tap / Attenuator

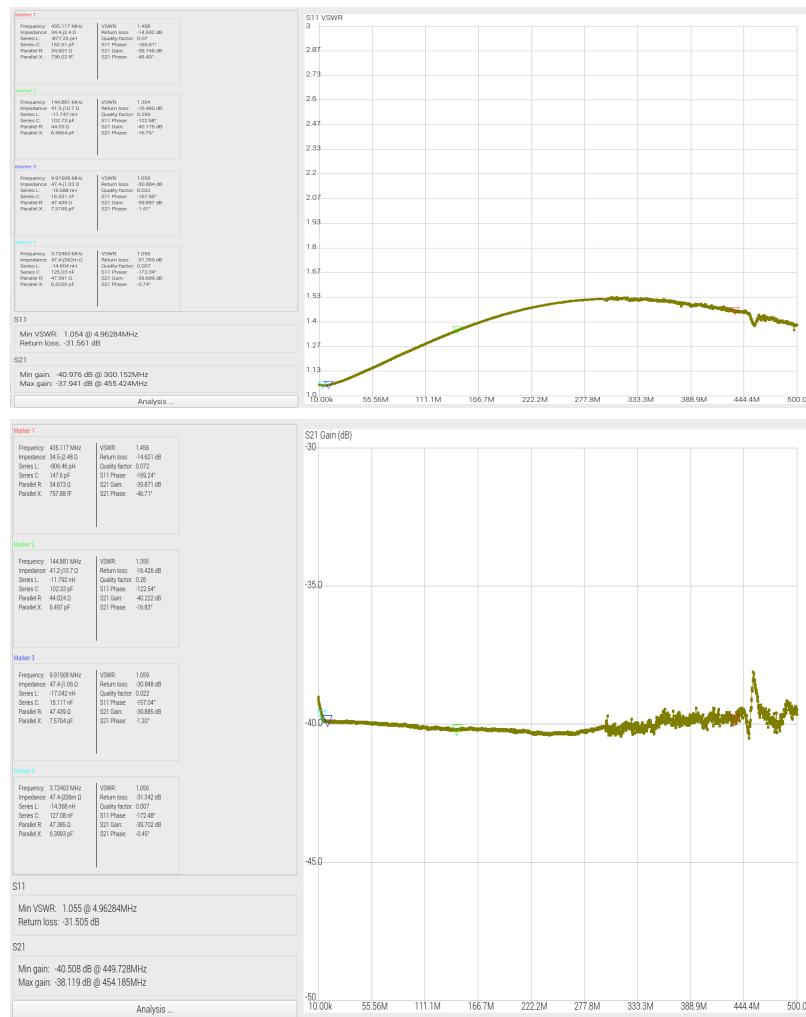
Since presented device is intended to be used to measure output power of the typical ham radio transceivers, the input signal level accepted by the power meter must be extended to 100 Watts or 50 dBms. This is done by the use of -40dB RF Tap/Attenuator (see below).



Resistors R1a, R1b and R1c shall be 500mW rated. R2 can be 125mW rated.
 Please note that if you do not terminate RF path with 50ohm dummy load or antenna, the average power dissipated on one of R1 resistors goes up to 1.3W, which may cause its permanent damage!

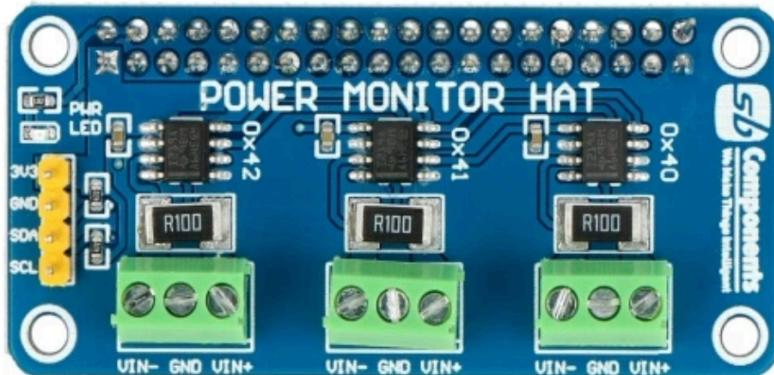
To minimize attenuation increase for frequencies above 144MHz the capacitor made of wire connected parallel to R1a shall be added (orange wire on the picture above - not shown on the circuit diagram).

RF Tap VSWR and Attenuation curves are shown below:



ADC Module Based On Ina219

Power Monitor HAT based on Texas Instruments Ina219 chips [7] and manufactured by SB Components [8] is used to perform analog to digital conversion of the AD8307 output signal (see below).



This particular HAT offers four 12-bit ADC channels, which can be handy if SWR measurements are considered in the future. Ina219 chips are controlled by Pi Zero using I2C data bus.

AD83xx output voltage level, decreases with the frequency of measured signal even though input signal power level is kept constant. This effect can be compensated with simple two point calibration procedure [5] performed separately for each ham band. Calculated coefficient pairs are then incorporated into Python measurement script executing on Pi Zero. Band selection is made by use of up/down buttons on meters front pannel.

Display Module

To present measurement results Mini PiTFT 1,3" 240x240px ST7780 based display HAT [9] from AdaFruit is used (see below).



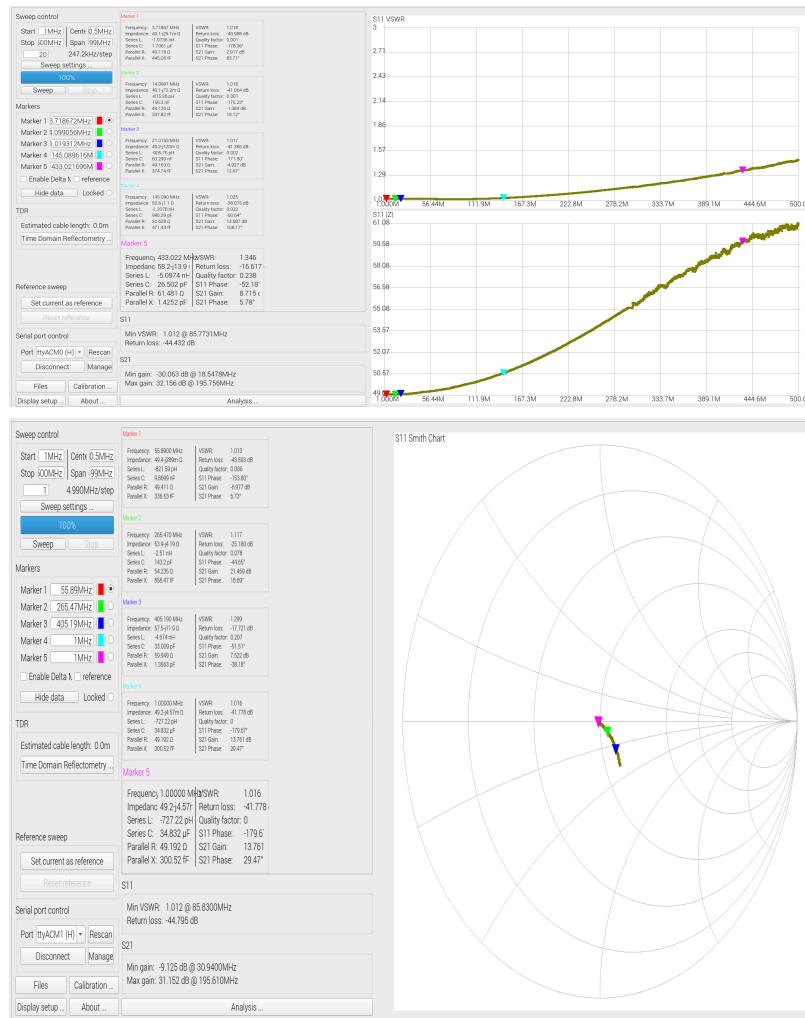
Display module is controlled by Pi Zero using SPI bus. PiTFT HAT is also equipped with two tact switches, which are used in this project as Up/Down band selectors.

Dummy Load Module

During power measurements antenna is replaced by dummy load based on inductance-less RFP-250 resistor manufactured by Anaren (or similar such as RFR 50-250). To allow for sufficient heat disipation resistor has been enclosed in aluminium Hammond box (model: 1590LLB) attached to heat sink (see below).

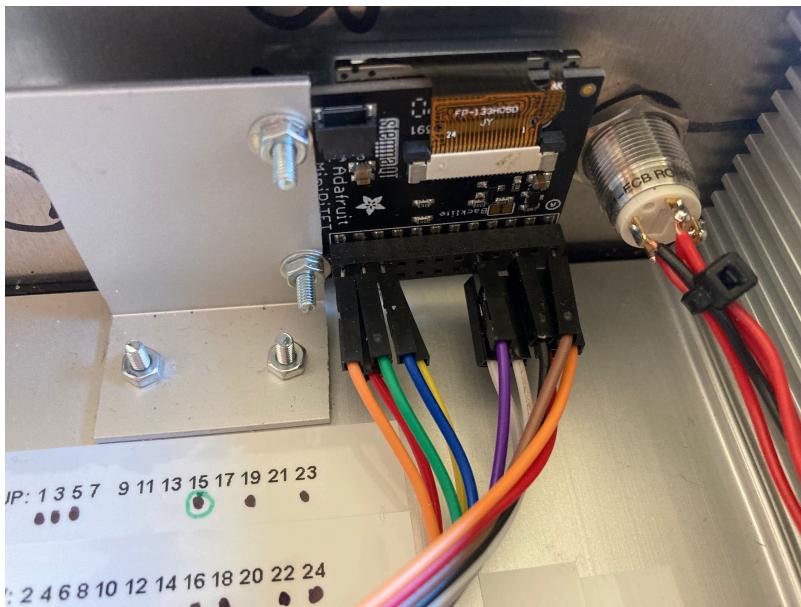
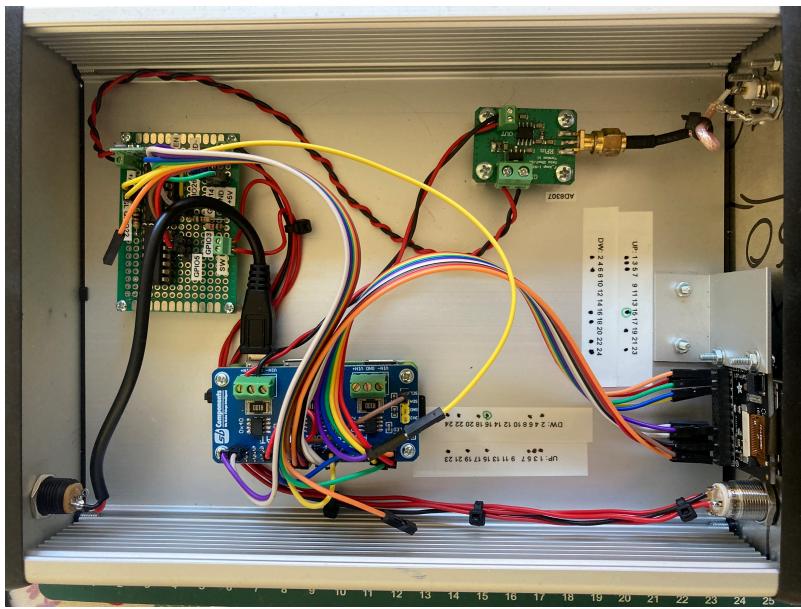


Key characteristics of presented dummy load are shown below:

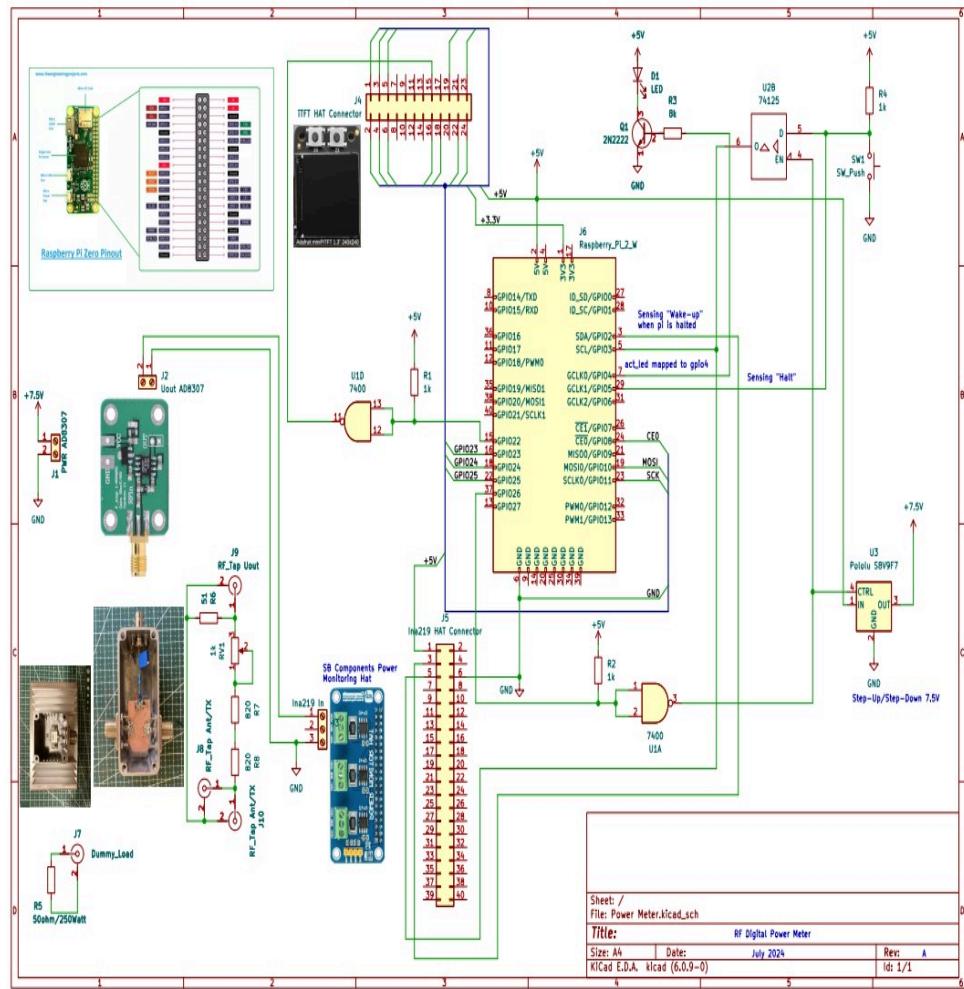


RF Power Meter

AD8307 module together with Pi Zero Platform integrated with Power Monitor HAT (Ina219 ADC) and PiTFT HAT (Display and Band Selector Buttons) are enclosed into separate Hammond Box chassis and form a digital power meter, which is the central part of presented power measurement rig (see below).

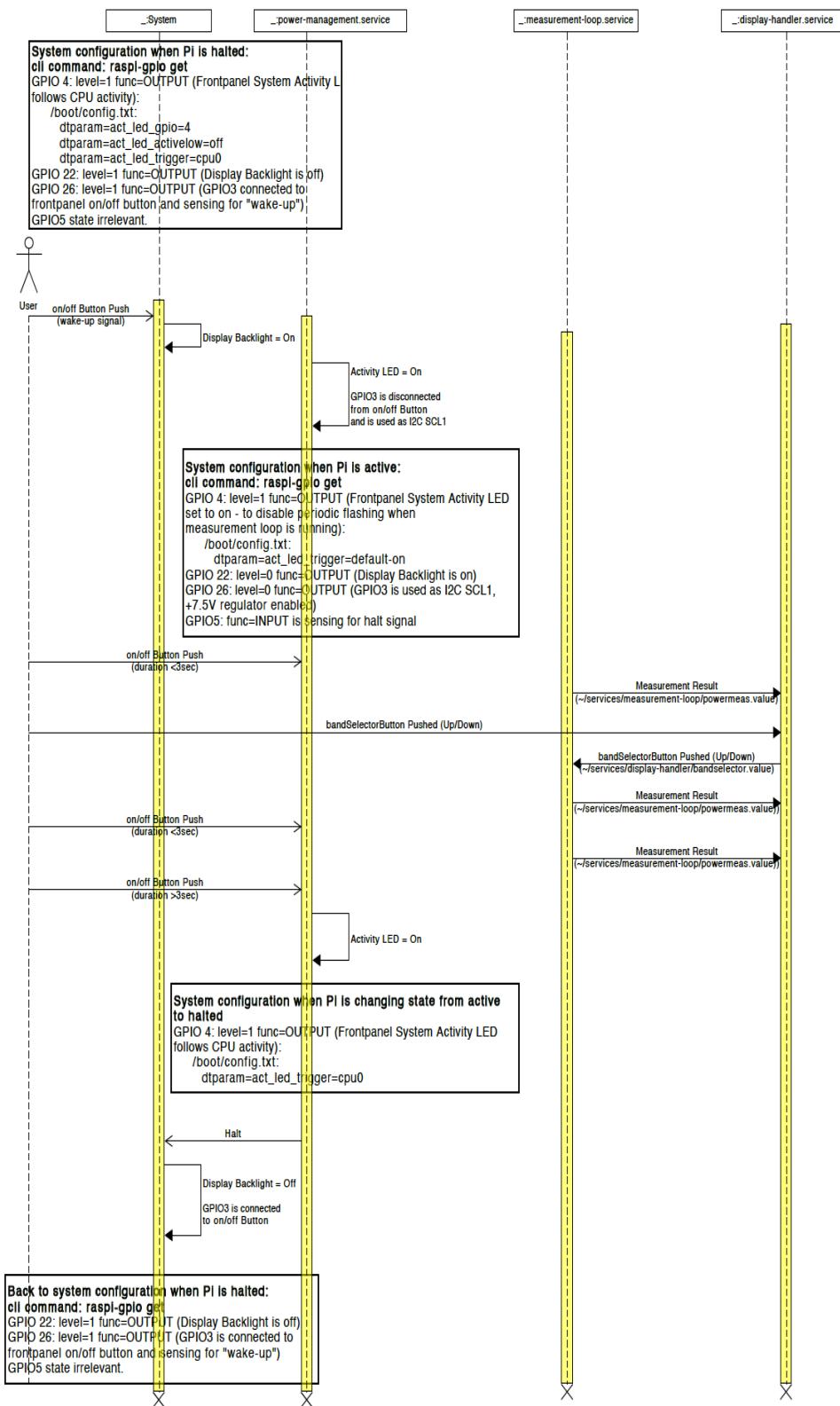


RF Power Measurement Rig - Electrical Diagram



Power Meter - Software Architecture

Key building blocks of power meter software include three systemd services written in python. Interaction between those services, operating system and the user are depicted on the below flow diagram:



Brief description of each power meter service can be found below:

Display Handler

Display Handler service (`display-handler.service/display-handler.py`) provides power meter graphical interface and implements band selector function.

Band selector function is based on two tact switches available on PiTFT display HAT. Display handler continuously reads power data from the `powermeas.value` text file stored in measurement loop service directory and displays it on TFT screen. It also continuously checks if one of the up/down tact switches have been pressed and based on this information calculates currently selected measurement band. Latest selected band information is stored in `bandselector.value` text file located in display handler service directory.

Display Handler service utilizes the following Circuit Python modules:

- `board` - SPI bus support
- `digitalio` - tact switches status handling (including switch debouncing function)
- `adafruit_rgb_display` - support for ST7780 TFT display controller

Display Handler also uses Python Imaging Library (PIL) to implement graphical operations such as font handling and drawing.

Measurement Loop

Measurement loop service (`measurement-loop.service/measurement-loop.py`) continuously measure the input power level in dBm and Watts.

Measurement loop service (`measurement-loop.service/measurement-loop.py`) continuously reads AD8307 output voltage over I2C interface of Ina219 12-bit ADC. It also continuously reads selected measurement band stored in file `bandselector.value` located in display handler service directory. Based on selected band the appropriate voltage to power conversion function is used to calculate power value in dBm, which corresponds to measured output voltage of AD8307. Additionally power level in dBm is converted to its equivalent in Watts. Both power values are stored in `powermeas.value` text file located in measurement loop service directory.

Measurement loop service utilizes the following Circuit Python modules:

- `board` - I2C bus support
- `adafruit_ina219` - ina219 ADC support

Power Management

Power Management service (`power-management.service/power-management.py`) controls status LED and ensures clean halt/power-off of Pi Zero platform.

When Pi platform is woken-up from halt or powered on, power management service starts and switches status LED from tracking CPU activity (default set in `/boot/config.txt`) to the continuous on state. This prevents LED from periodic blinking while measurement loop is running. It also changes GPIO26 output level from default High to Low, which disconnects GPIO3 from on/off button preventing possible interruption of data flow on I2C bus when on/off button is pressed and at the same time enabling Vcc powering AD8307 sensor.

Power measurement service then continuously monitors the state of halt button connected to GPIO3 line. As soon as halt button is pressed for at least 3 seconds the power management service switches status LED back to tracking CPU activity and sends halt signal to the operating system, which puts Pi Zero into sleep.

When Pi Zero is in sleep mode GPIO26 state is high, which connects GPIO3 to on/off button allowing Pi to respond to wake-up trigger. GPIO26 high state also cuts Vcc from AD8307 sensor.

This is critical functionality which ensures clean halt of Pi Zero platform providing protection against SSD data card corruption, which is likely to occur when Pi is shut down in uncontrolled manner. It also prevents measurement results from being corrupted by on/off button presses, which could cause data flow interruption on I2C bus.

Power Management service utilises the following Python modules:

- RPi - GPIO support

Power Meter - Software Configuration

A detailed description of the power meter software configuration can be found in [11] – [powermeter_pi_config.txt](#). A brief description of this procedure is presented below.

A 32-bit version of Debian Bookworm was used as the base system image. Basic configuration changes such as user account definition (powermeter) and wifi configuration were made from the Raspberry Pi Imager tool.

Before the first boot of the system, the ssh and ssh via USB were enabled and the default behavior of the LED indicating the device status was changed. This

part of the configuration was done by directly modifying the /boot/firmware/config.txt and /boot/firmware/cmdline.txt files and creating an empty /boot/firmware/ssh file.

The next step was to install the power management service. To do this, copy the source code of the service (file: [power-management.py](#)) available in this repository [12] within ~/sources/services/power-management directory to the /home/powermeter/services/power-management directory on the Raspberry Pi platform, and the service configuration file (file: power-management.service) to the /lib/systemd/system/ directory. Then, using the systemctl commands, activate the power management service and verify its correct operation (systemctl options: start, enable and status).

In the same manner measurement loop and display services shall be installed, but in these cases we need to install the necessary hardware drivers and activate i2c and spi buses before services are started.

To communicate with the Ina219 ADC chip, the Adafruit driver (adafruit-circuitpython-ina219) is used.

To communicate with the TFT ST7789 controller, the Adafruit driver (adafruit-circuitpython-rgb-display) is used. To operate correctly it requires the installation of raspi-blinka libraries. Dejavu fonts and python libraries: PIL and Numpy have also been added to the system.

Basic information about the power meter has been placed in the /etc/motd file, the content of which is displayed automatically after logging in to the power meter platform with the ssh command.

The power meter software also has the ability to log basic information about its status using the journal system service.

Calibration Procedure

In order to get accurate results, the measurement setup has to be calibrated. Two and four point calibration methods have been experimented with and two point calibration has been ultimately selected as more accurate within 0-100W nominal working range [5]. Calibration procedure has been run for complete measurement setup including power meter, attenuator, dummy load and cabling. Since power reading depends on frequency range calibration procedure had to be repeated for each band.

Calibrated system provides +/- 2 Watt accuracy within 0 - 100 Watt range of measured power and frequencies covering all HF ham bands.

Before the start of calibration process measurement-loop service shall be disabled from the power meter cli:

```
systemctl stop measurement-loop.service
```

and instead, [measurement-loop.py](#) script shall be run:

```
python ~/services/measuremenet-loop/measurement-loop.py
```

[measurement-loop.py](#) script prints voltage level, which corresponds to measured power.

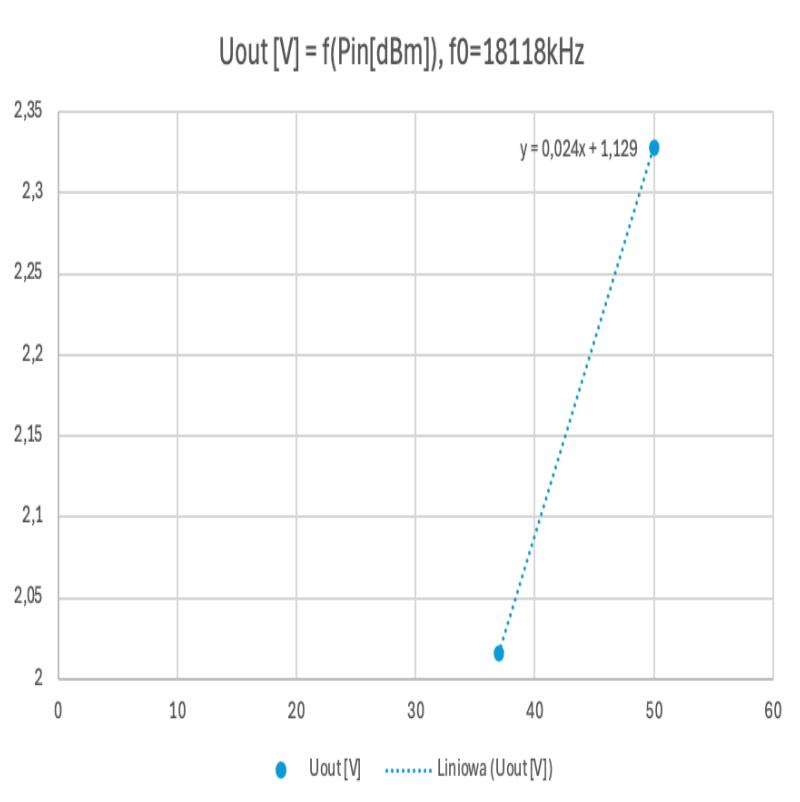
As a source of reference signal Yaesu FT-710 transceiver in CW mode has been used.

Calibration procedure consisting the following steps:

1. Set transceiver output power to minimum value of 5W and record corresponding voltage reading
2. Set transceiver output power to maximum value of 100W and record corresponding voltage reading. Avoid taking full power measurement for more than 10-15 seconds.
3. Convert reference power levels to dBm's
4. In Excel or similar tool run regression analysis to find a and b parameters of linear curve:

$$U(P[dBm])[V] = a * P[dBm] + b$$

Example of linear curve fitting for 17m band using Excel ($a = 0,024$, $b = 1,129$):



5. Convert the curve to the format, which can be applied to [measurement-loop.py](#) service and update the script accordingly:

$$P[dBm] = (U[V] - b)/a$$

6. Repeat for each band

Measurement Setup

The operation of the setup is very easy assuming the rig has been already calibrated. In such scenario it is enough to connect transceiver output to the input of the RF tap marked with red arrow on the picture below. For obtaining most accurate results, the relevant frequency band shall be selected using UP/DOWN keys next to the TFT display.



For the novice user it might be surprising that when transceiver is in LSB/USB mode and PTT button is pressed, there is practically no signal present at the output of the transceiver unless one starts speaking or better whistling. To obtain results, which can be interpreted the transceiver shall be set into CW mode. As soon as the straight key is pressed the power of clean carrier sine wave dispating on dummy laod can be measured.

Acknowledgments

Special thanks to Zygmunt (Zygi) Szumski, SP5ELA for lots of relevant technical comments and editorial support!

Otwock, August 2024

References

[1] Simple RF-Power Measurement, Wes Hayward, W7ZOI, Bob Larkin, W7PUA - June 2001 QST

[2] A PIC-based HF/VHF Power Meter, Roger Hayward, KA7EXM - May/June 2005 QEX and the June 2005 QST

[3] Digital Power Meter, Reinhardt Weber, DC5ZM - FUNKAMATEUR (1/2018 page. 38) (in German) or at English transcript at <https://www.dl2mdu.de/rf-power-level-meter/>

[4] Miernik Poziomu Sygnału RF z AD8318, Mirek Sadowski, SP5GNI (in Polish) - <https://hf5l.pl/miernik-poziomu-sygnalu-z-ad8318/>

[5] Obscurities & Applications of RF Power Detectors, Carlos Calvo, Analog Devices 2007

[6] AD8307 Data Sheet, Analog Devices

[7] Ina219 Zerø-Drift, Bi-Directional CURRENT/POWER MONITOR with I²C™ Interface, Texas Instruments

[8] Power Monitor Hat Product Page, SB Components - <https://shop.sb-components.co.uk/blogs/posts/how-to-use-power-monitor-hat>

[9] Mini PiTFT 1,3" Display Product Page, Adafruit - <https://www.adafruit.com/product/4484>

[10] Warsztatowy miernik mocy w.cz, Bartosz Krajnik SP2Z, Zjazd Techniczny Krótkofalowców, Burzenin 2018 (in Polish)

[11] RF Power Meter, Andrzej Mazur SP5GW,
https://github.com/SP5GW/RF_Power_Measurement_Rig