Power & Temperature Monitoring System
Setup & Configuration

http://www.lungStruck.com/projects/power-temp-monitor

This is not meant to be a complete step-by-step guide to setting up the system – it's only intended as a reference based on how I configured my own system. I'm providing it for the benefit of others but you may need to change some steps to adapt to your own needs or changes in the OS, etc.

OS & Software Setup

(These instructions were written and based on Raspian Jessie Lite. They may still apply to future versions of Raspian, but some changes may be required.)

Before you begin, copy the necessary files from my website to a USB thumb drive and modify the email, server name, etc. to match the settings you need. Then connect it to the Raspberry Pi.

- Run `raspi-config`
    - Enable One-Wire Interface support (required for DS16B20 temperature sensor)
    - Exit and reboot
- Install updates and packages required by PowMon and PowWeb:
    - `sudo apt-get update`
    - `sudo apt-get dist-upgrade`
    - `sudo apt-get install python3 python3-pip`
    - `sudo pip3 install rpi.gpio retrying psutil flask waitress`
- Copy PowMon/PowWeb files to the pi. The following assumes all files are on the root of a USB drive seen by the system as /dev/sda1 (You can verify if it's /dev/sda1 by running sudo `fdisk -l`.)
    -
    - `mkdir ~/PowMon`
    - `sudo mkdir /media/usb`
    - `sudo mount /dev/sda1 /media/usb`
    - `cp /media/usb/powmon.py ~/PowMon/`
    - `cp /media/usb/powweb.py ~/PowMon/`
    - `cp /media/usb/ds18b20.py ~/PowMon/`
    - `cp /media/usb/email-list-power.txt ~/PowMon/`
    - `cp /media/usb/email-list-temperature.txt ~/PowMon/`
    - `sudo cp /media/usb/powmon.service /lib/systemd/system/`
    - `sudo cp /media/usb/powweb.service /lib/systemd/system/`
    - `chmod +x ~/PowMon/powmon.py`
    - `chmod +x ~/PowMon/powweb.py`

    - `chmod +x ~/PowMon/ds18b20.py`
    - `sudo chmod 644 /lib/systemd/system/powmon.service`
    - `sudo chmod 644 /lib/systemd/system/powweb.service`
    - `sudo umount /media/usb`
- Test the PowMon program to make sure it runs and all required packages are installed and to allow support files to be created. Run it from the PowMon folder so the log and INI files are created there.
    - `cd ~/PowMon`
    - `./powmon.py`
    - The program should run, rebuild the log and INI files and may send an email if it runs long enough.
    - Press CTRL+C to close the program once it's stable.
- Test the PowWeb program to make sure it runs and all required packages are installed.
    - ./powweb.py
    - It should start on port 8080. Browse to `http://[IP Address]:8080` from another PC to test it.
- Configure systemd to run the PowMon/PowWeb services automatically on boot
    -
    - `sudo systemctl daemon-reload`
    - `sudo systemctl enable powmon.service`
    - `sudo systemctl enable powweb.service`
    - Reboot: `sudo shutdown -r now`
    - The service should start about the time the login prompt appears. Login and verify the service is working:
        -
        - `sudo systemctl status powmon.service`
- Verify the system is working properly
    -
    - One of the status LEDs should be lit.
    - `tail -f ~/PowMon/powmon.log` to view the log file and automatically display new entries
    - Disconnect power long enough for an email to be sent. The log file should update. Verify the email is received.
    - Reconnect power. The log file should update and another email should be sent. Verify the email is received.
    - Browse to http://[IP Address]:8080 from another PC. You should see temperature, service status and the log file entries.
    - If the emails were sent and the website works, everything is working properly and the system is ready. Press CTRL+C to finish viewing the log.

Hardware Setup

DO NOT PLUG THE SYSTEM INTO A UPS! It won't be able to detect that the power is out if it's on a UPS. It should be connected to a surge protector w/o battery backup instead.