## Getting access to PowerMon designs and source code

The main repo for PowerMon is located at https://github.com/beppodb/PowerMon. There is a 1.0 and 1.1 version of the hardware and software.

Directories are as follows:

Application – A simple application that initializes PowerMon and reads samples based on command-line parameters.

Assembly – Instructions for adding SMT and through-hole components to the circuit board and specifications of cables used to connect devices to PowerMon.

Firmware – Firmware that programs the onboard microcontroller; program with AVR-Dragon or AVR-Dude

Gerber – Gerber files that can be given to a PCB fab to create more PowerMon circuit boards

PCB – schematic and layout in EAGLE format

## What does the PowerMon hardware consist of? What limitations are there?

The best explanation can be found in the PowerMon tech report (http://www.renci.org/wp-content/pub/techreports/TR-09-04.pdf) but briefly, PowerMon provides the following:

1) Inline monitoring of up to 8 DC channels for current and voltage, which means that 5 rails for a standard ATX MB can be measured along with peripherals like HDDs and PCIE cards. 3 "devices" can be measured at once, assuming there are enough pins available.
2) Higher frequency measurements (1 KHz for up to three components) that are streamed to a host PC via USB.
3) Compact design that fits in a 3.5" hard drive slot.
4) At scale, the device is relatively cheap to fab and have assembled - $250 for batches of 50 and < $500 for small batches.

**Restrictions:**

PowerMon2 is built around a custom PCB that includes Atmel ATmega168 8-bit microcontroller connected to 8 Analog Devices ADM1191 digital power monitor chips. It does not provide total system power, and due to the specific chips it can safely measure up to 300 W on all three channels without any additional cooling. The valid range for the current and voltage sensors is 3.15V - 26V and up to 11.3 Amps.

(From Dan B.) There are three factors that determine the current limit:
- Voltage sag. This is most relevant on the 3.3V rail. The voltage cannot go below 3.15V, or the sensor will not work. For the other rails, the limit is 10%, according to the ATX spec.
- Dynamic range of the sensor. For a .005 ohm resistor, the limit is .10584/.005 = 21.168A.
- Heat dissipation of the circuit board. I believe the values I quoted earlier (11.3A probably comes from

this) were for a 20°C temperature rise. I think you can probably exceed this safely or increase the airflow over the board to keep the temperature low.

**\*\*For measuring a high-end GPU like a Tesla, the safest bet is to just measure that separately from other components since these devices can use up to 250 W.**

## How does PowerMon differ from Watts Up? How about PowerInsight?

**Watts Up?** is generally used to measure system power and is designed to measure standard 120 V electrical connections. PowerMon has a much higher measurement resolution than Watts Up?, 1 KHz vs. 60 Hz, and it allows for the measurement of circuits at a smaller granularity (for instance ATX rails vs input at the power supply). PowerMon also streams measurement values via a USB cable to a host PC while Watts Up stores results in an onboard memory of limited size.

https://www.wattsupmeters.com/secure/index.php

**PowerInsight** is a more recent hardware tool developed at Sandia. PowerInsight uses Hall effect transducers to measure current without needing to be inline with the circuit, providing more accurate results with less interference. The main disadvantage of this infrastructure is that it is not an open-source design and it can only be purchased in certain configurations of Penguin Computing servers.

Link:http://www.sandia.gov/~jhlaros/Laros_23_PowerInsight.pdf

Another good paper that talks about Watts Up? vs. Hall Effect Transducers can be found in "Goel, et al, Chapter two – Techniques to Measure, Model, and Manage Power, Advances in Computers, November 2012".

http://www.sciencedirect.com/science/article/pii/B978012396528800002X

## How to build PowerMon hardware

PowerMon is an open-sourced project released via RENCI open source license, which means that the PCB layout, schematic, and Gerber files are available online along with assembly instructions. You can review the schematic using EAGLE, found at http://www.cadsoftusa.com/download-eagle/

To build a new PowerMon device you will need to do the following:

1) Get a quote from a PCB manufacturer for fab of the PCB by providing them with the Gerber files (gerber directory in Github repo). If you don't want to attach the components yourself, then you should request a turnkey assembly quote and provide a list of the components in assembly/v1.0/powermon-bom.ods. **<TODO> This document was updated.**
2) Either get a quote for the cables specified in assembly/v1.1/pm-cablea.ods and assembly/v1.1/pm-cableb.ods or buy the components and a molex micro-connector crimper. **<TODO> Link Engineering Tool**.

# How to build PowerMon cables

## How to check the version of and program PowerMon firmware

Connect to the PowerMon port with Minicom at 1MB/s and send "v<lf>" (where <lf> is a line feed. ctrl-j in minicom, I think). That will show you the hardware and firmware version number according to https://github.com/beppodb/PowerMon/blob/master/firmware/system/powermon.c.
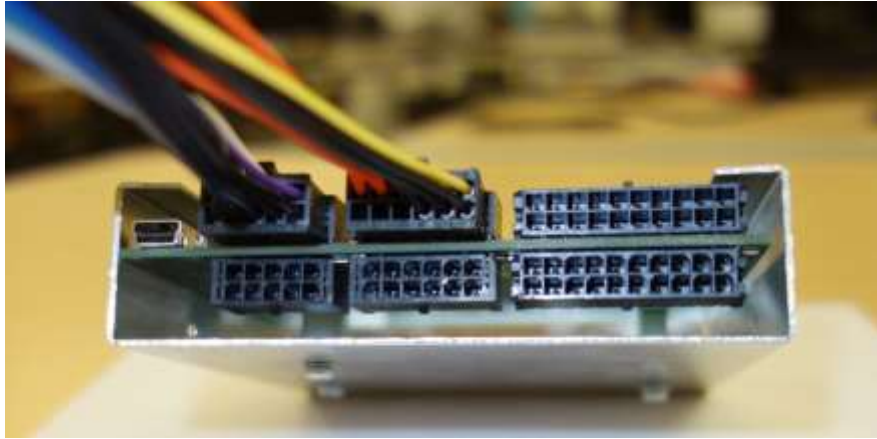
## How to hook up devices to PowerMon



### ATX motherboard power:
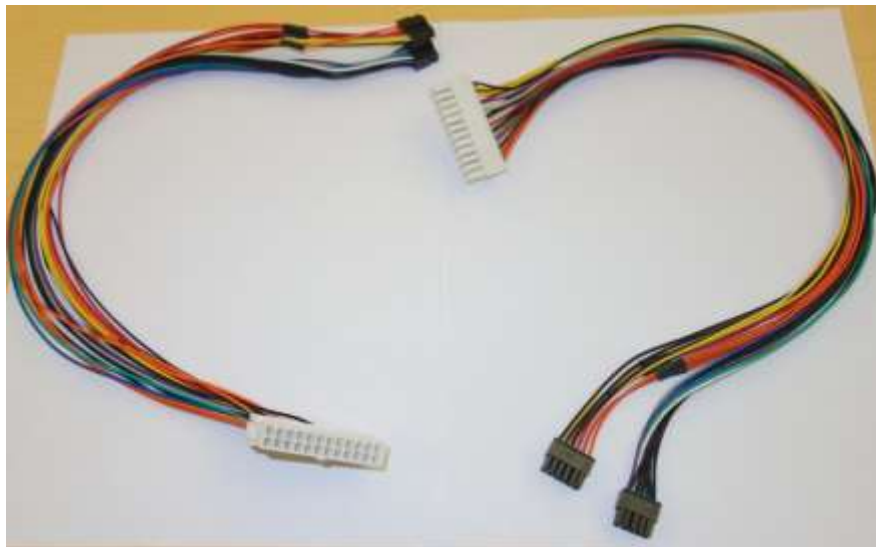


ATX motherboard power is measured using "Cable A" specified in Github under the assembly folder, and there is one white female and male connector.

While the sensor cables can apparently plug into either side, the input traditionally goes into the top set of female plugs while the output goes into the bottom set of female plugs.
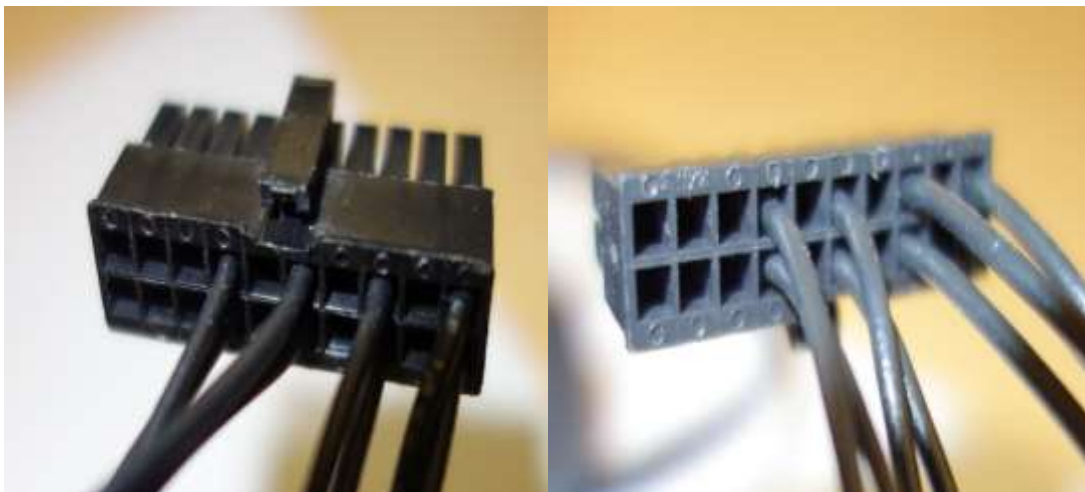


### CPU power:

### AC Power adapters:

Power Brick inputs into PowerMon



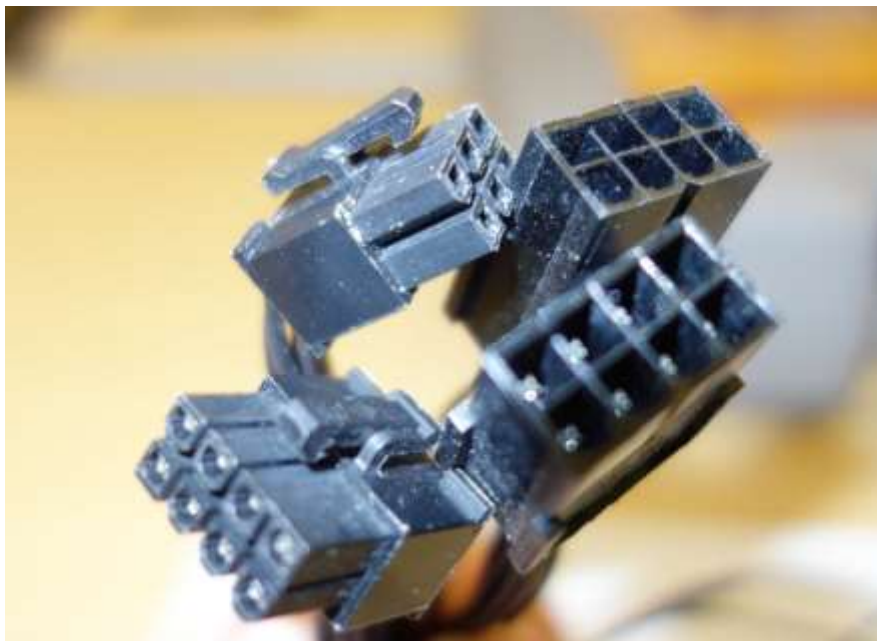Power Brick outputs from Powermon – three different power bricks and a USB power output.



Power Brick sensor connectors Powermon – note that each sensor uses two pins.

To measure PCIe power requires 3 sensors and a PCIe interposer card. The card for our infrastructure has been designed by GTRI and has measurement points for the two 3.3V and 12V rails.



PCIe 6 and 8 pin power sensors are used to

## How to check the firmware version and update it

You can connect to the PowerMon port with Minicom at 1MB/s and send "v<lf>" (where <lf> is a line feed. ctrl-j in minicom, I think). That will show you the hardware and firmware version number according to https://github.com/beppodb/PowerMon/blob/master/firmware/system/powermon.c .

## How to compile and run the PowerMon application

Ideally use a different system to host the USB connection to reduce interference

<program_exe> <port_dev> <mask> <sample_pd> <num_samples>

The following example samples sensor 6 at 1 KHz for approximately 4 seconds (4000 samples).

$./run_powermon /dev/ttyUSB0 6 1024 4000

## How to reduce interference from included system components

Hard drive, memory, GPU/network devices