Arm Energy Probe Basics

Andy Green
andy.green@linaro.org
V10 2012-08-14

Linaro

Arm Energy Probe Basics

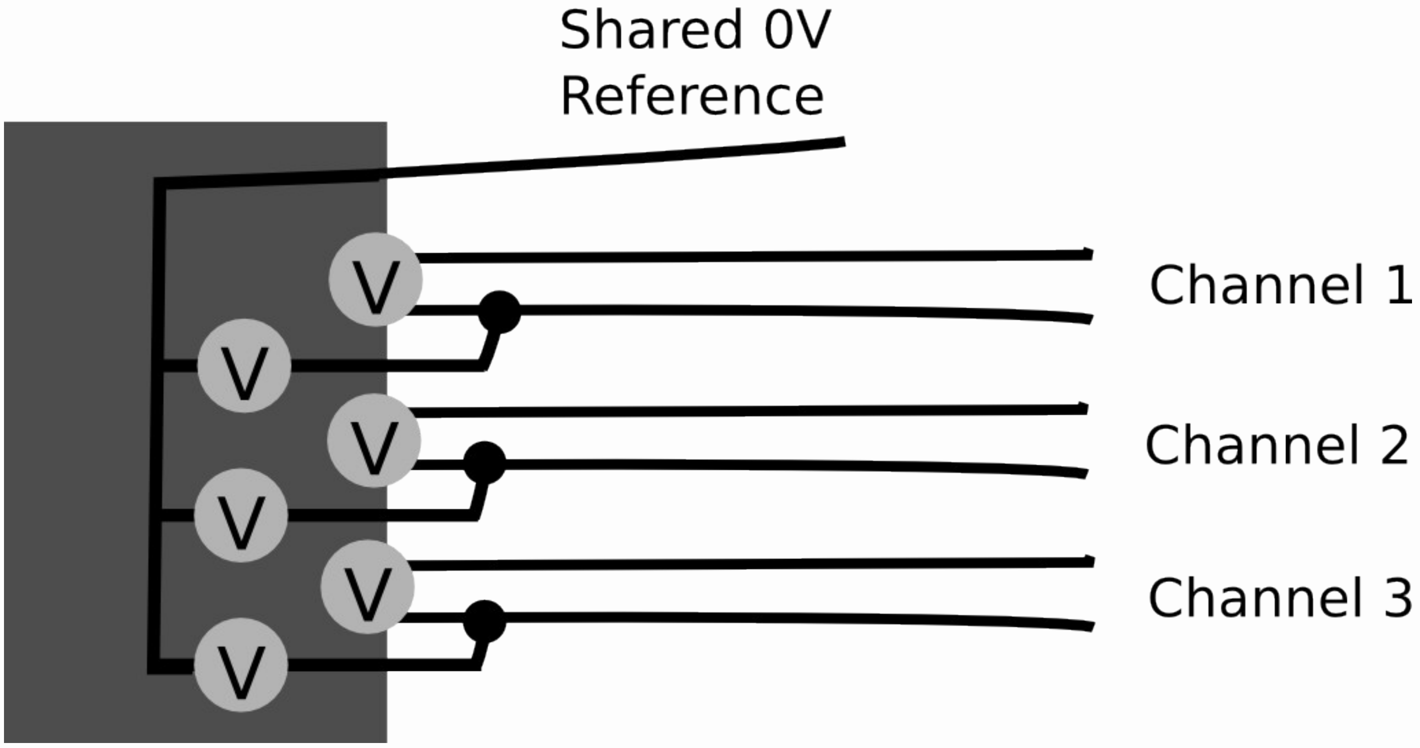
- What it is
- Quick electronics 101 recap
- Measuring around regulators
- Wiring your board for AEP usage
- Some actual measurements
- Major sources of measurement error
- Linux Commandline tool

Arm Energy Probe Basics

- What it is
- Quick electronics 101 recap
- Measuring around regulators
- Wiring your board for AEP usage
- Some actual measurements
- Major sources of measurement error
- Linux Commandline tool

Understanding what it is

- It's a 3-channel USB voltmeter, two voltages reported for each channel



Understanding what it is

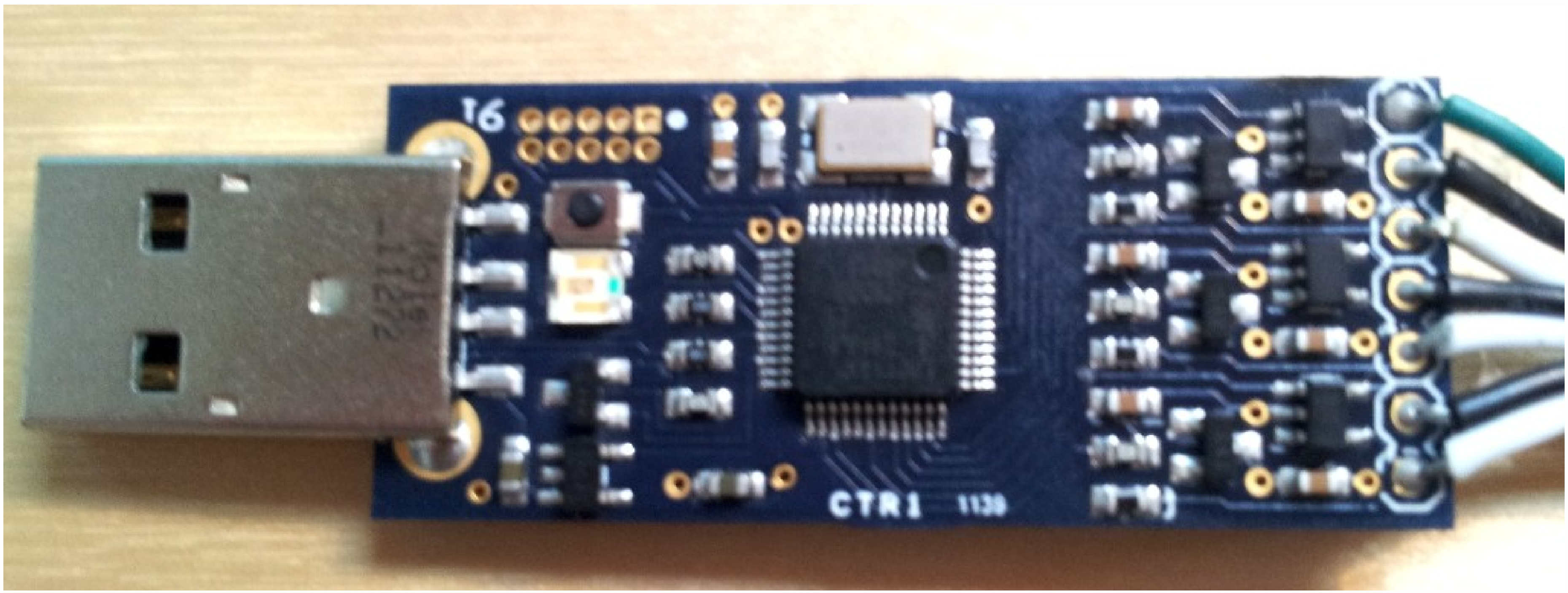
- The first voltage (up to 30V) is measured between one of the sense leads and 0V
- The other is an amplified (differential) measurement of the voltage between the two sense leads, limited to 165mV
  - These amplified channels are used to measure the voltage across shunts to calculate current as we will explain
  - 165mV limit affects shunt resistance selection

Linaro

USB side

- It has a Cortex M3 LPC1343 that appears to the host PC as a ttyACM CDC serial port class device
- Linux knows what to do with it
- Linaro has a commandline tool “arm-probe” which can drive it
- Check for other things touching ttyACM0!
  - Modem-manager from NetworkManager wants to fiddle with any ttyACM device it sees

Hardware



Practical problems...

- **Can only measure one channel at a time!**
  - **Simultaneous measurements on multiple probes possible but hard to synchronize**
- Probes do not have unique USB serial #
  - Cannot reliably be identified in multi-probe setup
  - Can probably be fixed by poking firmware and reflashing by hand
  - Reflashing only possible on Windows ^^

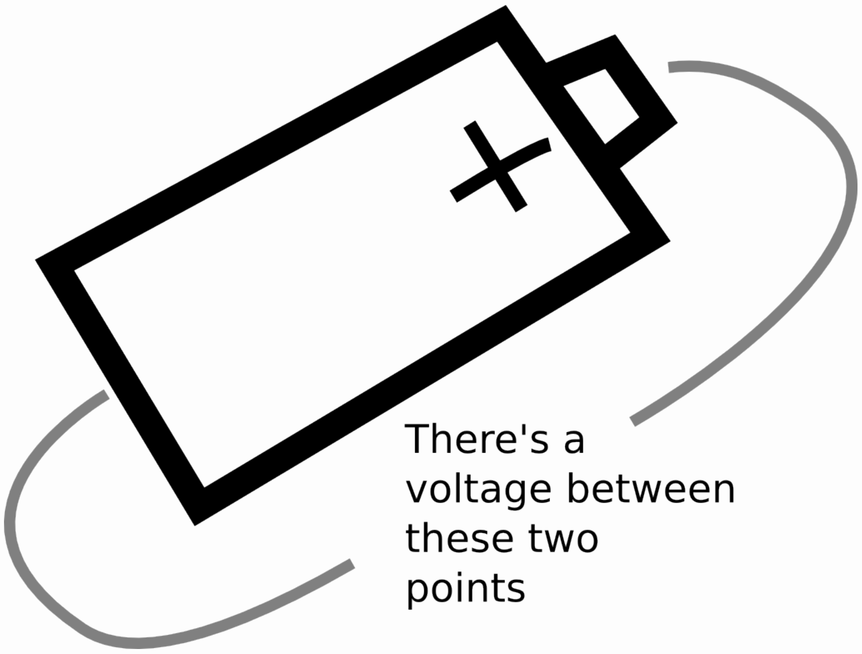
Linaro

Arm Energy Probe Basics

- What it is
- Quick electronics 101 recap
- Measuring around regulators
- Wiring your board for AEP usage
- Some actual measurements
- Major sources of measurement error
- Linux Commandline tool

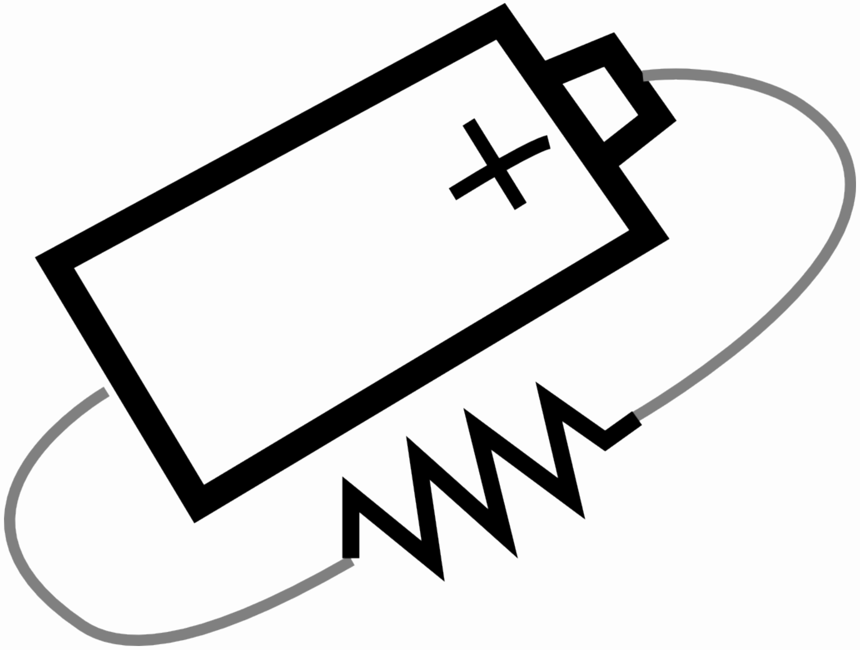
Voltage

- Voltage is the “potential difference” between two points, usually the power rail and a common “0V” or “ground” point, measured in Volts (V)



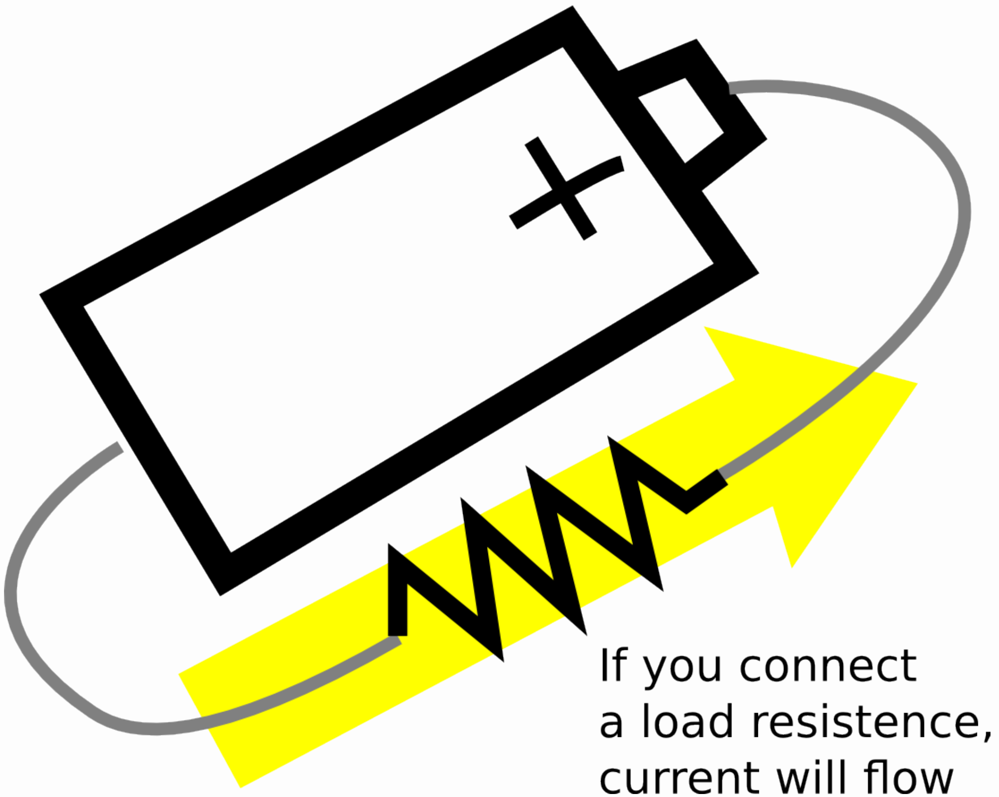
Load

- Load is a resistance (measured in Ohms or "R") you put between a voltage to make current flow



Current

- Current is a measure (in Ampere or A) of how much charge is moving through the circuit.



Power

- Power is the **voltage multiplied by the current** and is measured in Watts (W)
  - Even at low voltages a lot of power can be used if a lot of current is flowing
  - At high voltages, very little current needs to flow to use a lot of power
- Power is unique because it's the only way you can compare currents at different voltages

Some identities

- $I = V / R_{(load)}$
  - Into the same load, higher V makes more current flow
  - At the same voltage, higher load resistance makes less current flow
- $P = I \times V$
  - Lower V or I, less power
- $P = V/R \times V = V^2/R$
  - **Half voltage --> quarter power!**

Power is boss

- To talk about power, talking about voltage or current alone is useless
- P = IV so we need to talk about current AND voltage if we talk about either
  - Eg, “it takes 50mA”... 50mA at 1.2V == 60mW, 50mA at 5V == 250mW... which is it?
- Converting voltage and current measurements to power lets you compare measurements made at different voltages

Linaro

However...

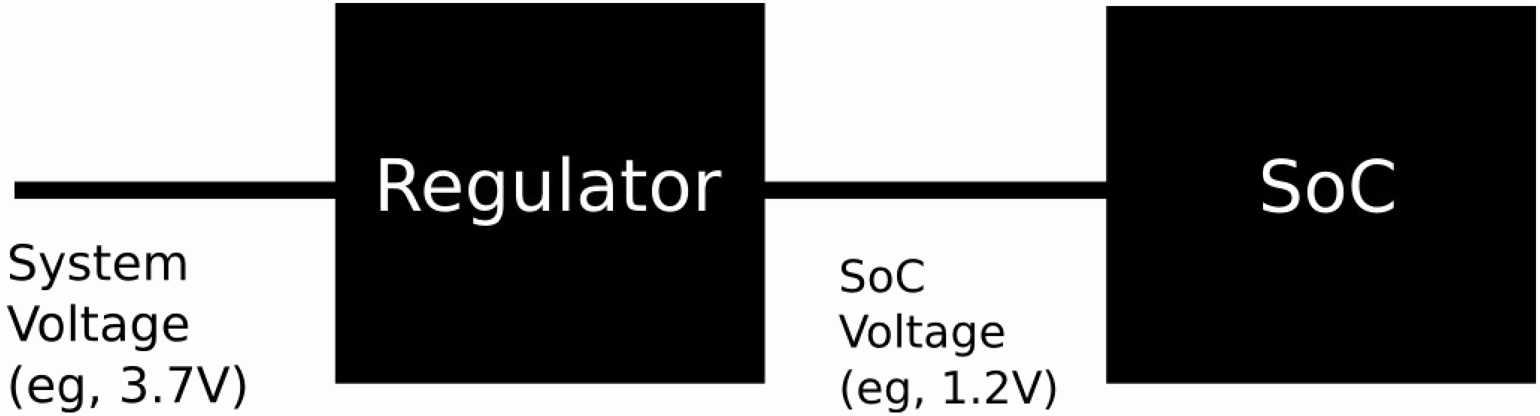
- If the voltage part of your measurement is constant, you can treat current part as a stand-in for being a scaled version of power
  - Shortcut is true if you are interested in relative changes in power local to same measured rail

Arm Energy Probe Basics

- What it is
- Quick electronics 101 recap
- Measuring around regulators
- Wiring your board for AEP usage
- Some actual measurements
- Major sources of measurement error
- Linux Commandline tool

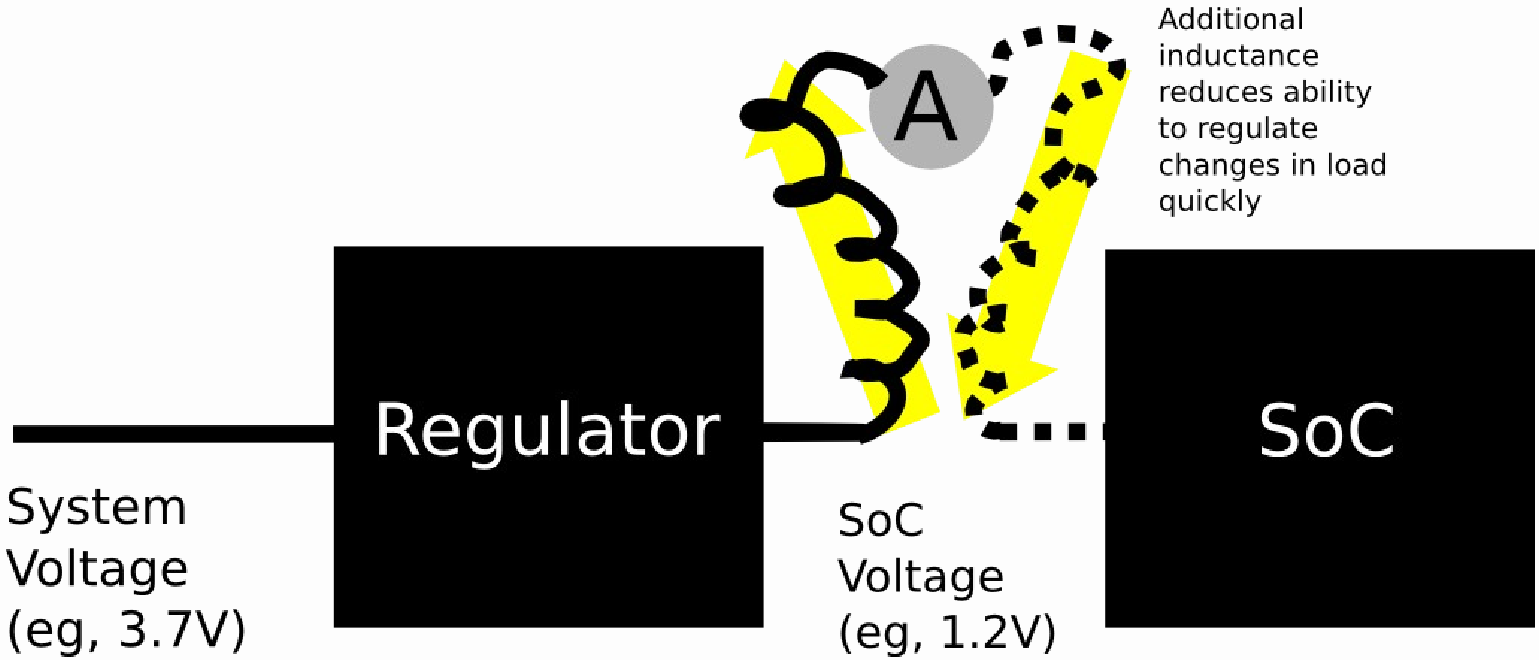
Regulator structure

- Regulators adapt a voltage source to provide a different voltage, even if the load is changing dynamically



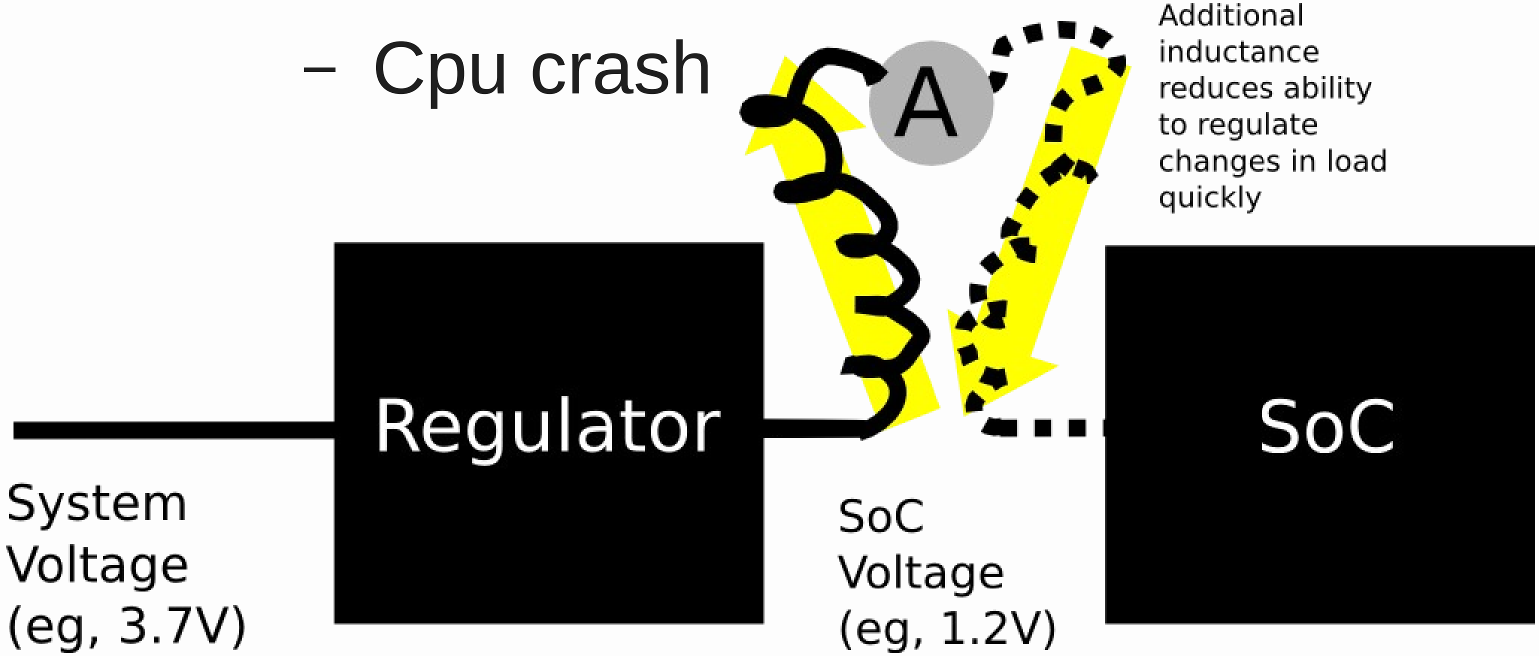
Traditional current measurement

- Normally we would stick an ammeter in series with the load



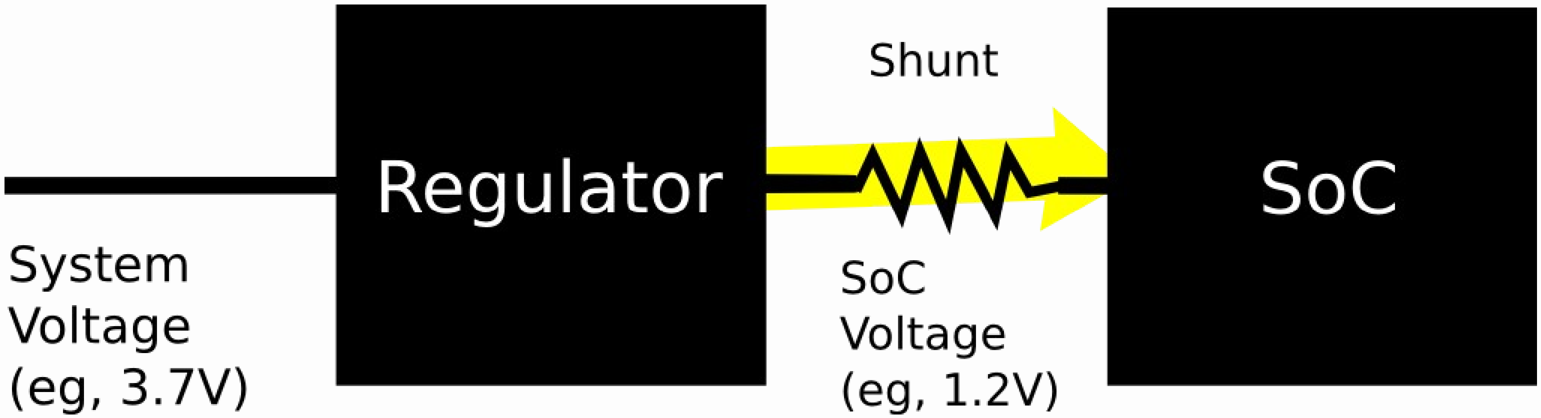
Traditional current measurement

- Regulator cannot “see” a fast-changing load clearly and fails to regulate
  - Cpu crash



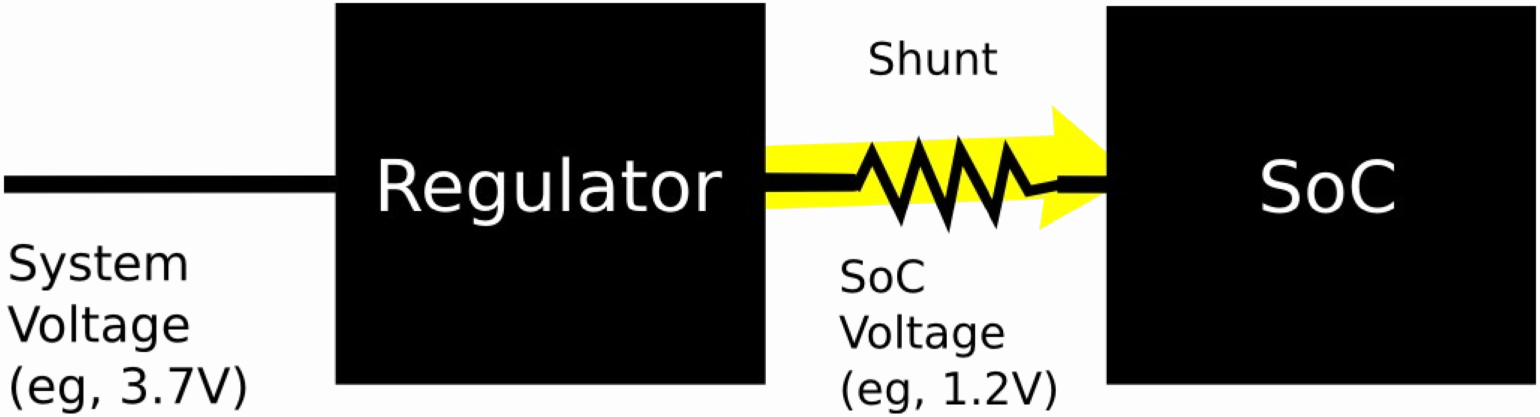
Shunt current measurement

- A shunt is a very small resistance (typ 33mR) placed in series with the load right on the PCB



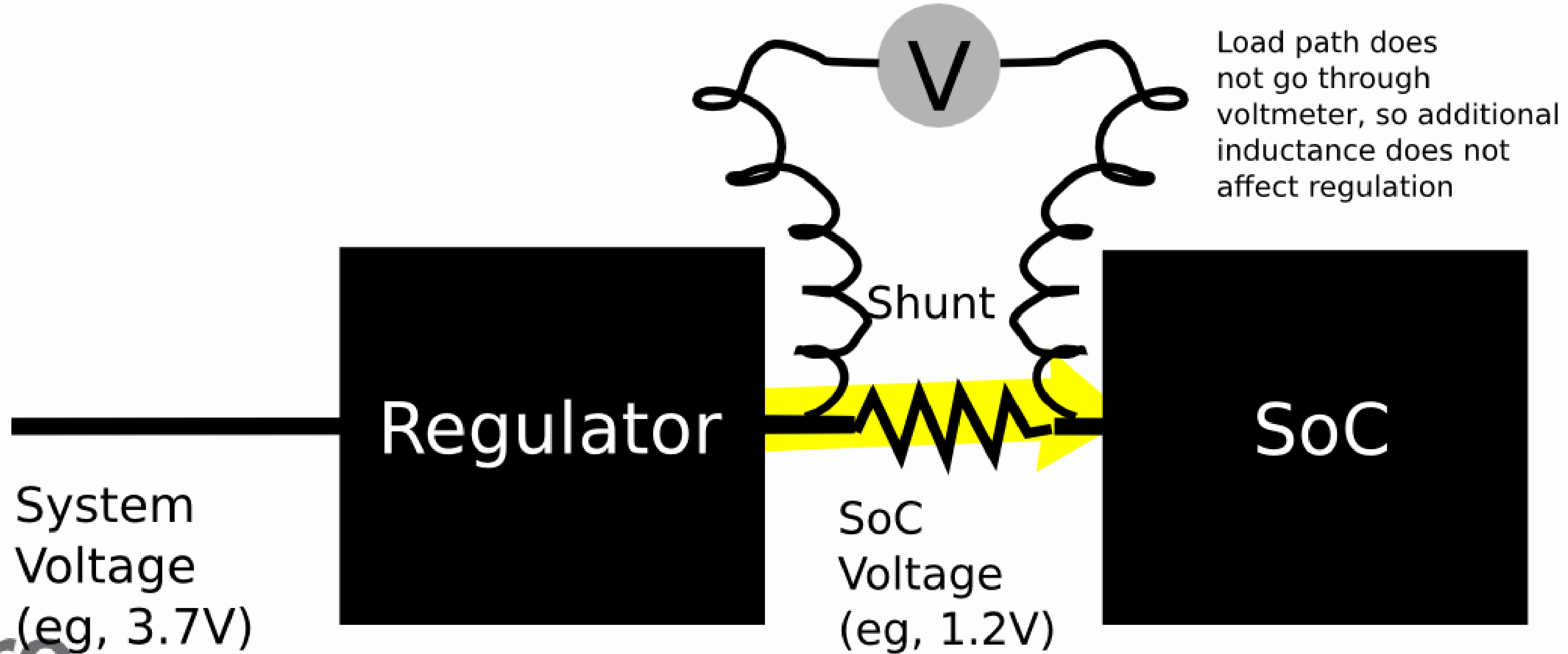
Shunt current measurement

- It introduces a very small reduction in voltage in proportion to current flowing



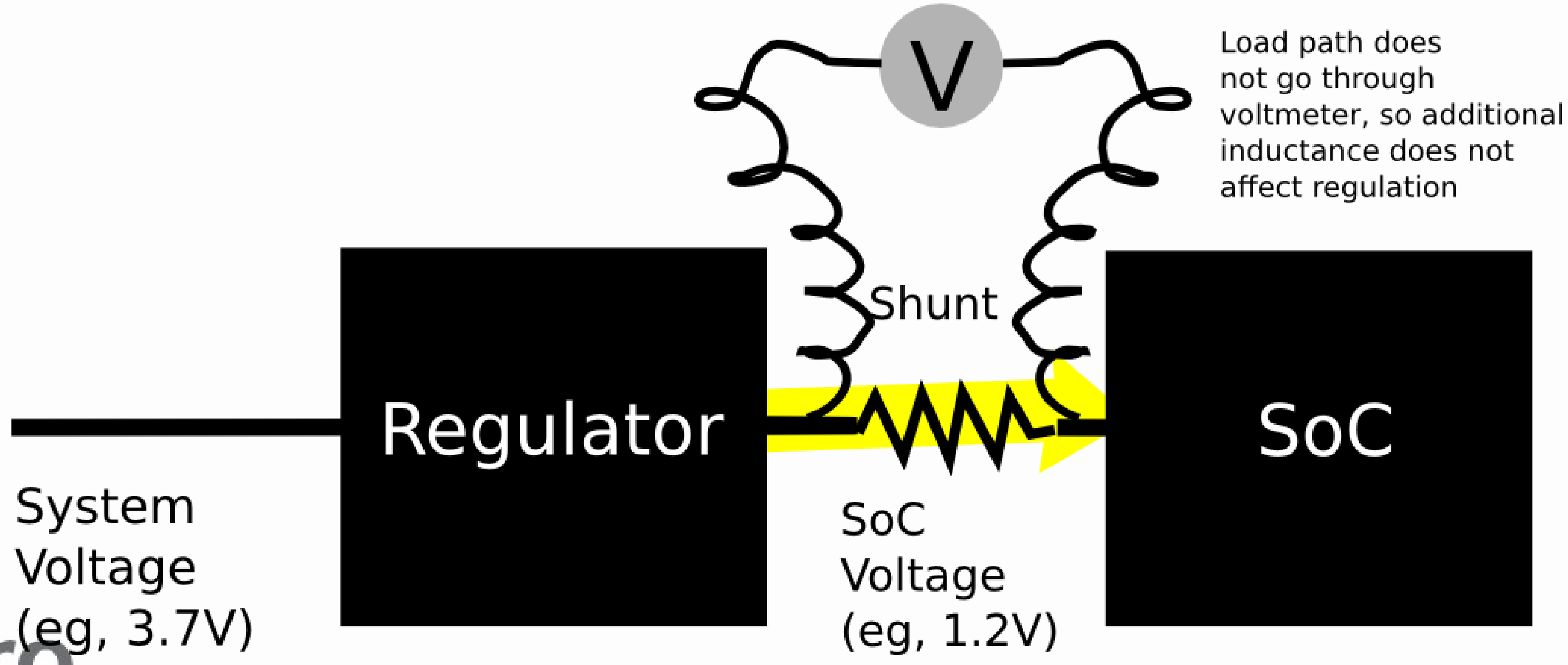
Shunt current measurement

- The energy probe adds an amplified voltmeter to measure the tiny voltage drop across the shunt



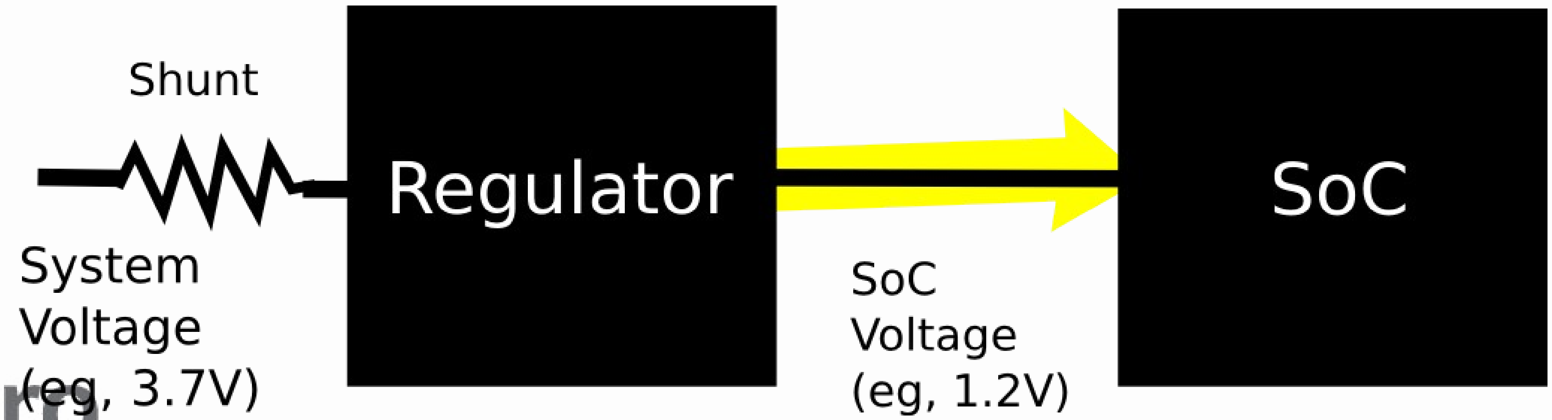
Shunt current measurement

- Because the shunt is a small resistor or metal staple, the regulator can usually “see through it”



Shunt current measurement

- Some regulators are only barely stable. You can measure the input side in those cases
  - Your measurement includes regulator efficiency losses



ARM Energy Probe

- So the probe measures and reports **two voltages** on each channel
  - The voltage at one side of the shunt compared to 0V
  - An amplified version of the tiny voltage **across** the shunt

Linaro

ARM Energy Probe

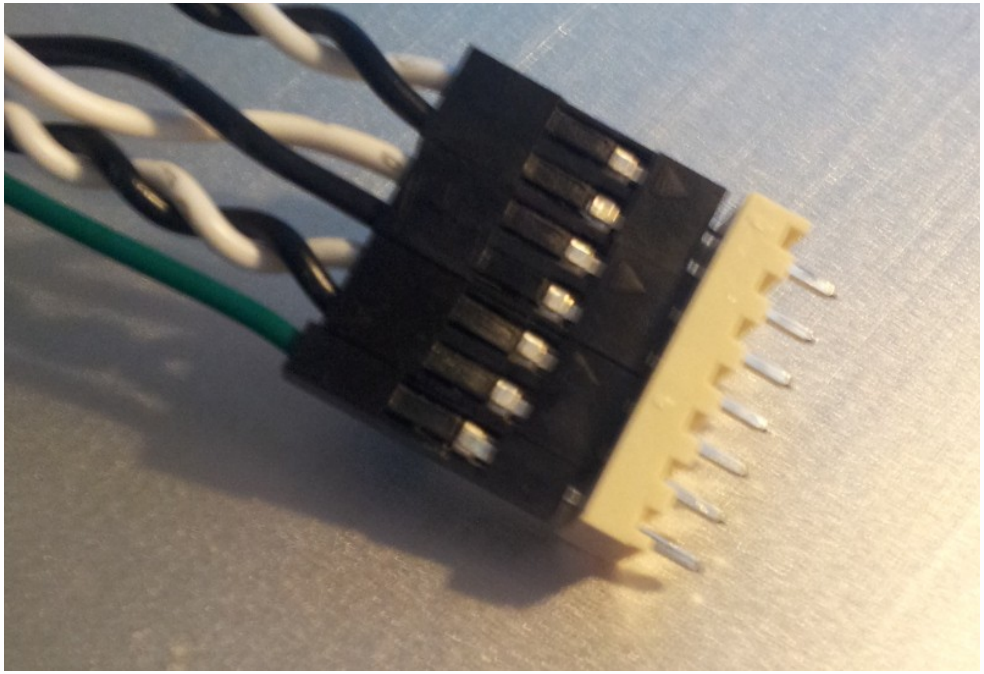
- If we know the shunt resistance, $I = V_{(shunt)} / R_{(shunt)}$ tells us current flowing
  - Probe cannot deduce current without knowing $R_{(shunt)}$
- Since the probe also measures the shunt voltage compared to 0V, we can calculate **power** from P=IV

Arm Energy Probe Basics

- What it is
- Quick electronics 101 recap
- Measuring around regulators
- Wiring your board for AEP usage
- Some actual measurements
- Major sources of measurement error
- Linux Commandline tool

Wiring the probe

- You'll need a 7-pin 0.1” header to plug the probe on so you can detach it



Wiring the probe

- Wire the green 0V lead to 0V on the PCB ideally near the regulators of interest
- Your board will work fine without the probe connected since current will flow through the shunt as usual

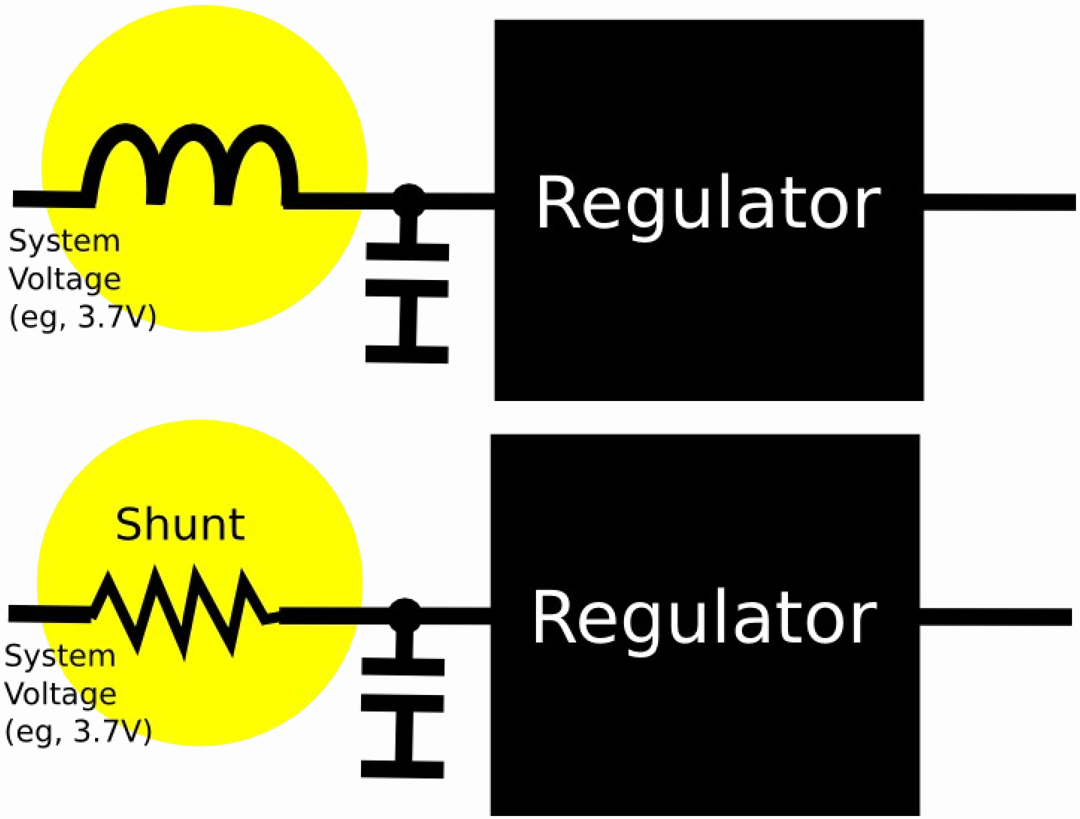
Choosing where to place shunt

- Most regulators have no convenient way to place shunt in series with output
- However there is almost always a convenient series inductor at the **input** designed to limit EMI going back up the power supply cable...

Linaro

Choosing where to place shunt

- We can replace this inductor with the shunt

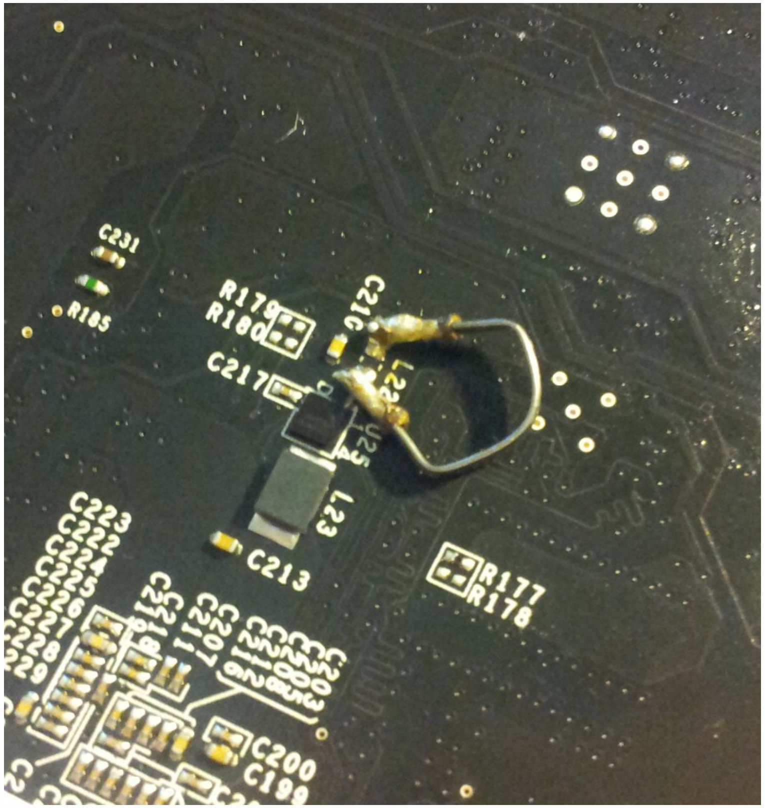


Choosing where to place shunt

- Measuring from regulator input side is fine but
  - You measure the input voltage, not the output
    - You cannot see regulator output DVFS voltage directly
  - Regulator efficiency overhead also measured
    - All real designs must include real regulators, so it's sane

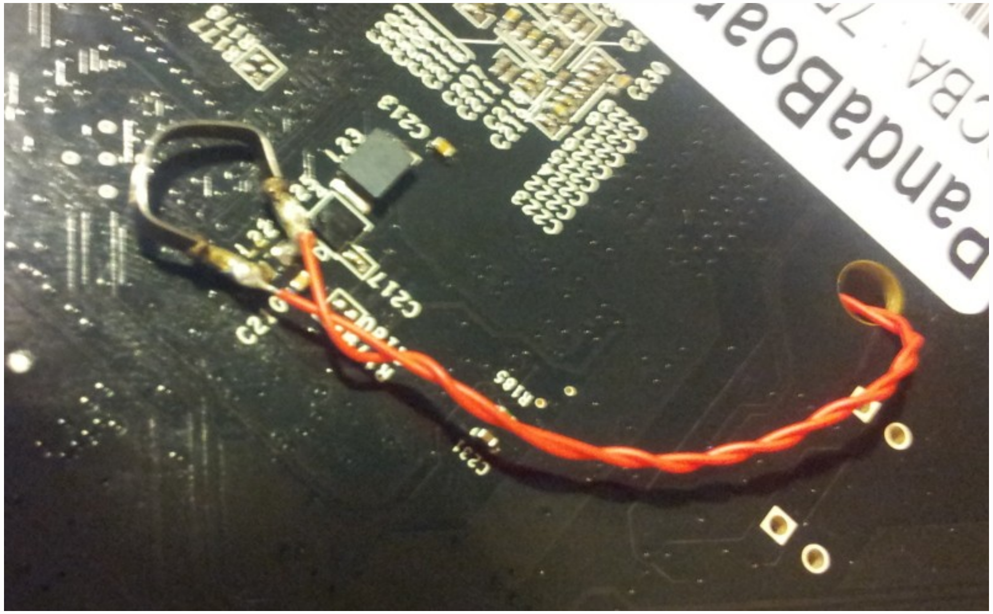
Placing the shunt

- This is a metal precision shunt replacing L22 on 4460 Panda (VDDCORE)



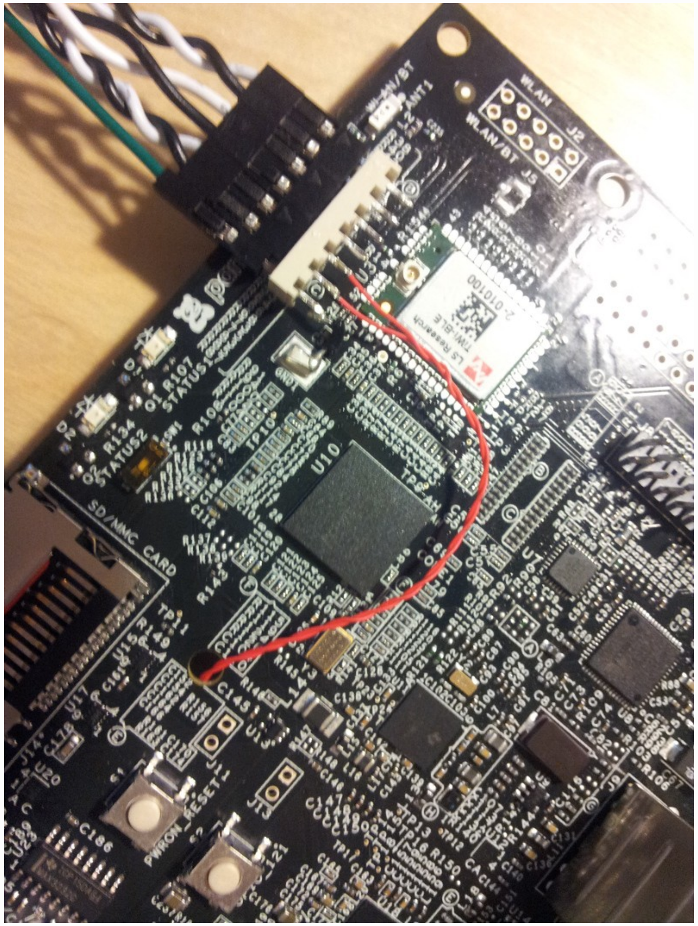
Wiring the shunt

- Twist two thin insulated wires together to wire the two sides of the shunt to the probe connector



Choosing where to place shunt

- Glue the header to one edge of the PCB



Probe cares about sense leads

- The white side of the sense leads needs to go on the pre-shunt side of the shunt, black to post-shunt
- It won't damage the probe to get it wrong but current readings will always be near zero

Linaro

Shunt resistance selection

- Low resistance shunt is preferable to minimize regulation disruption
- Probe ADC resolution can be a problem then
  - With 33mR shunt, one ADC count == 4mA resolution
  - Considerable “noise” or aliasing

Arm Energy Probe Basics

- What it is
- Quick electronics 101 recap
- Measuring around regulators
- Wiring your board for AEP usage
- Some actual measurements
- Major sources of measurement error
- Linux Commandline tool

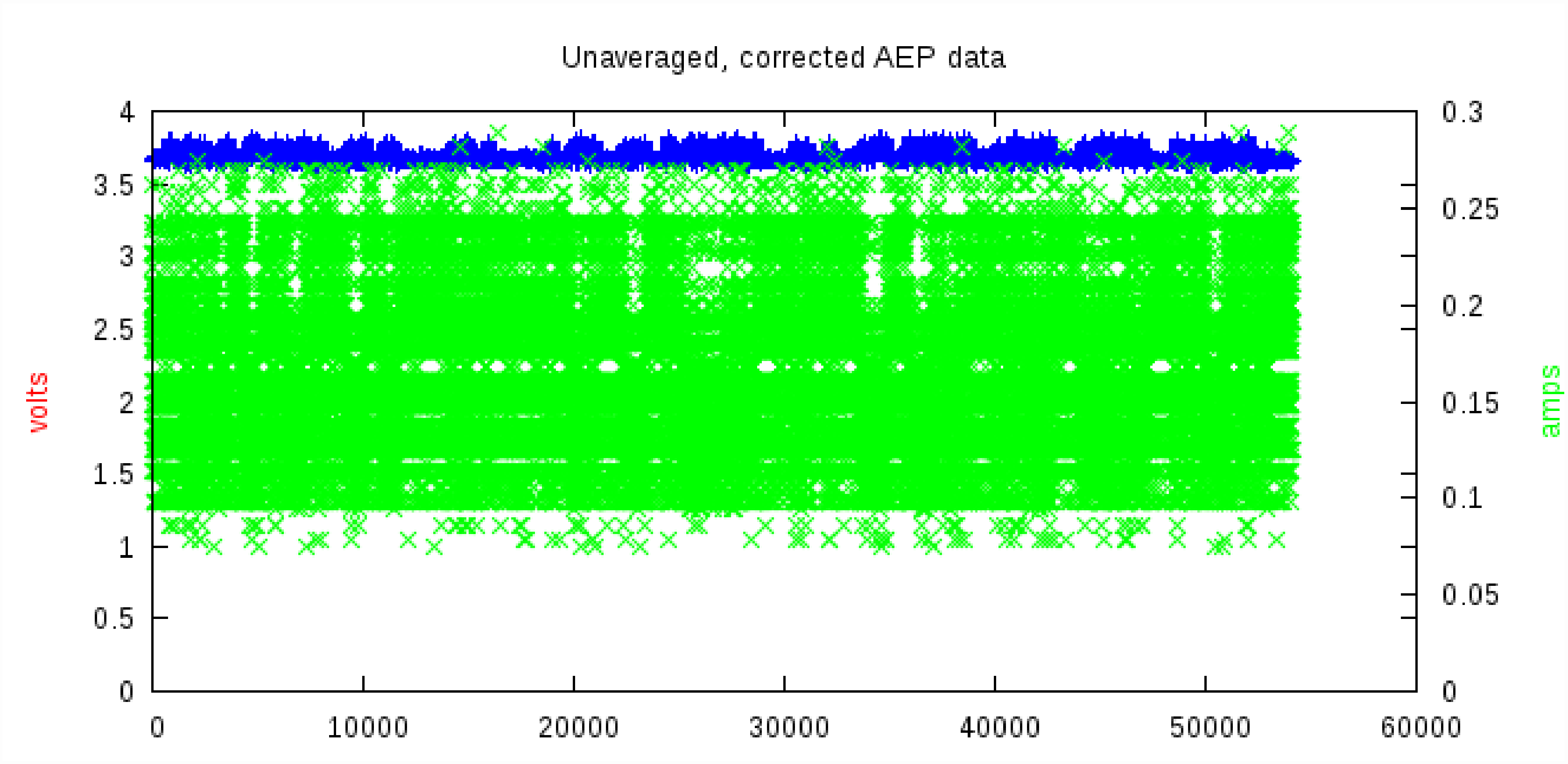
Linaro

Real measurement

- 4460 Panda VDDCORE
- Input side of regulator
- 33mR and 470mR shunt
- idle at U-Boot prompt
- 600MHz dynamic load
- Corrected for slope error
- Using linux commandline

Linaro

Unaveraged, 33mR shunt

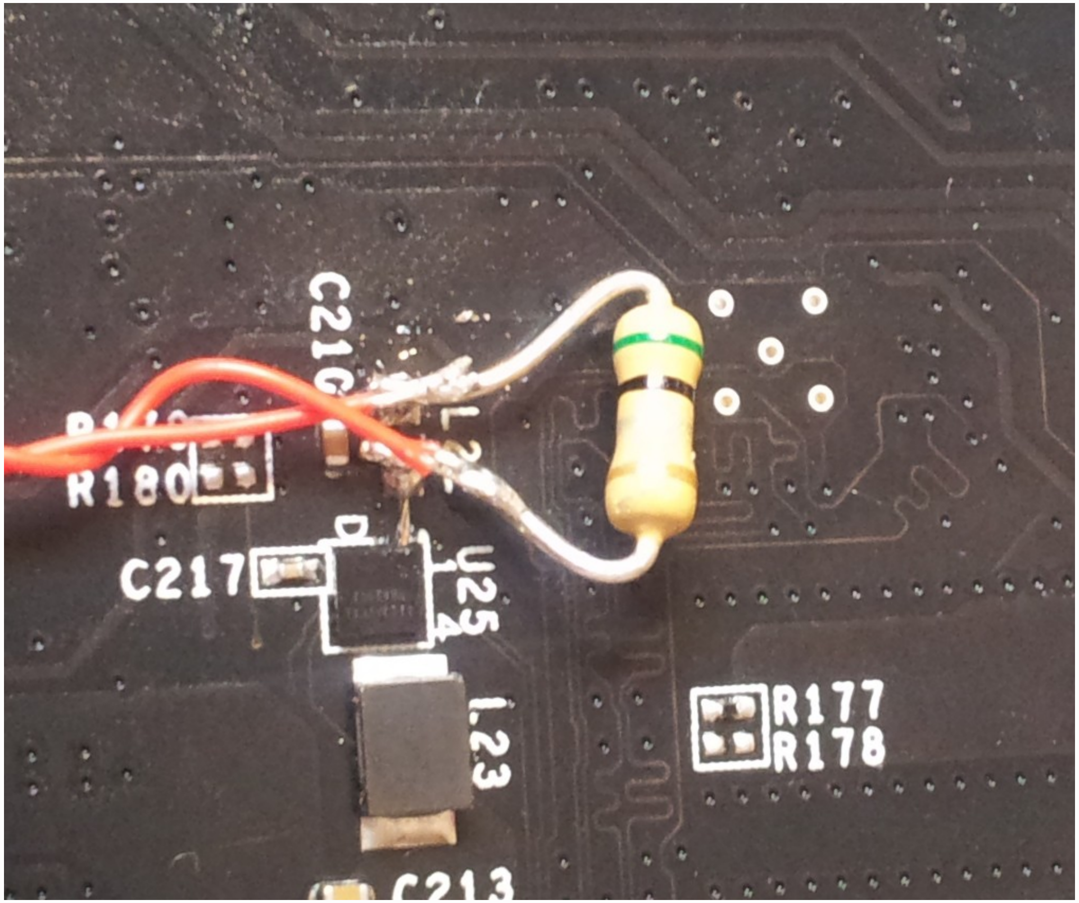


Interpretation

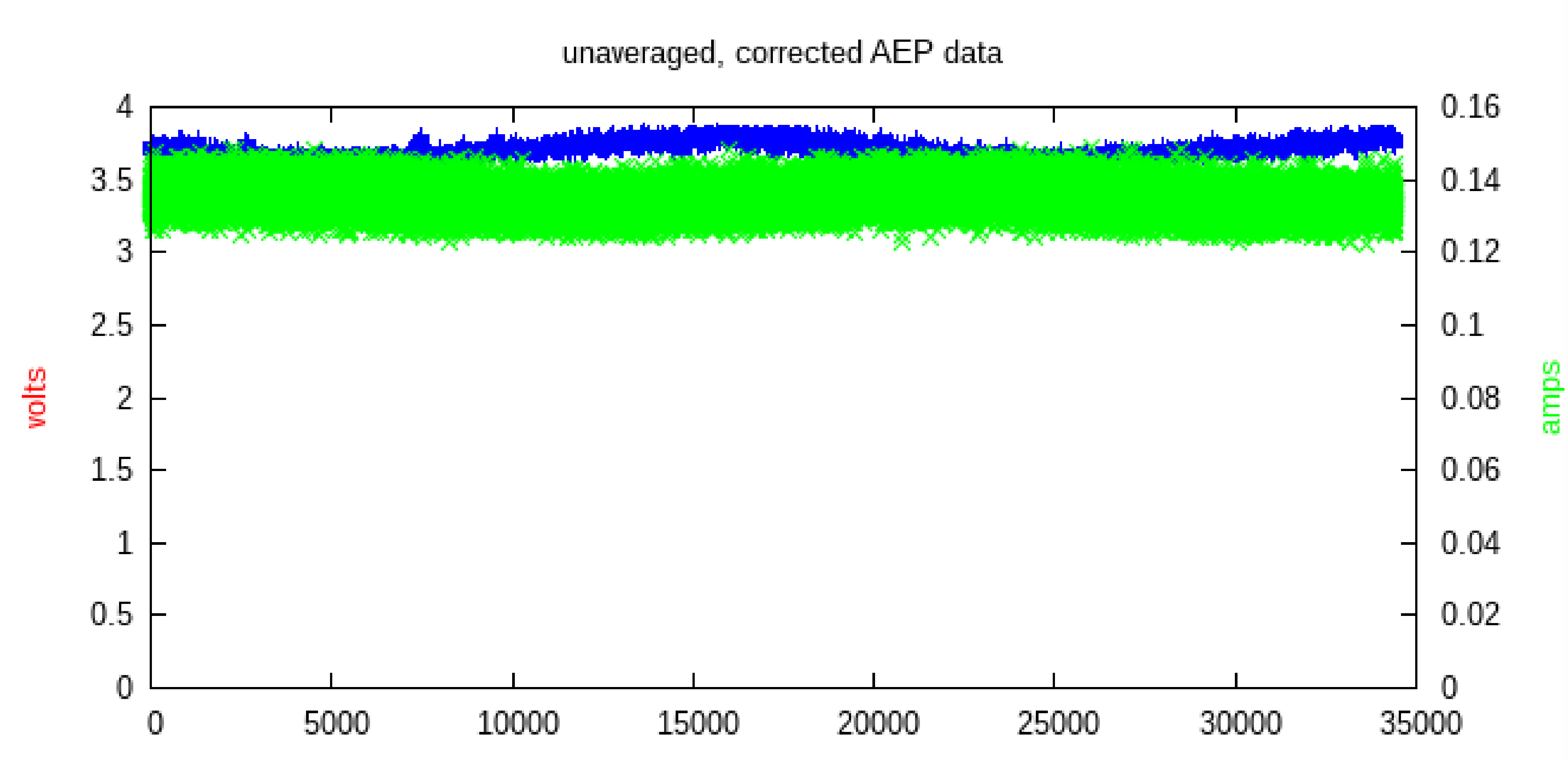
- 33mR is too low resistance shunt for the ARM probe
- limits the ADC count for the measurement to around 5% of ADC range
  - SNR reduced drastically
- Change to a higher resistance shunt so we have a bigger voltage to measure

Same tests with 470mR shunt

- “0.5R” resistor



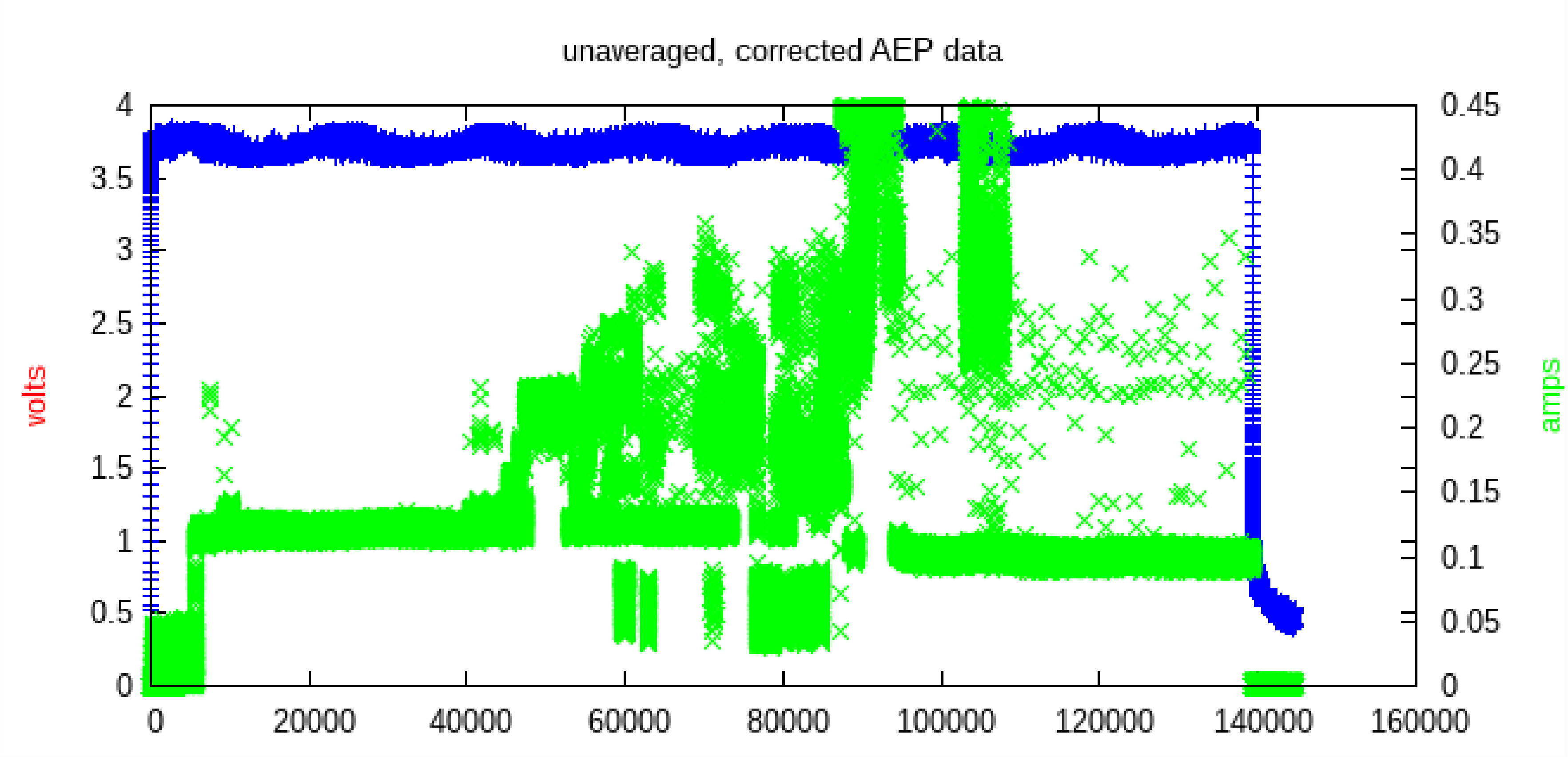
Unaveraged, 470mR shunt



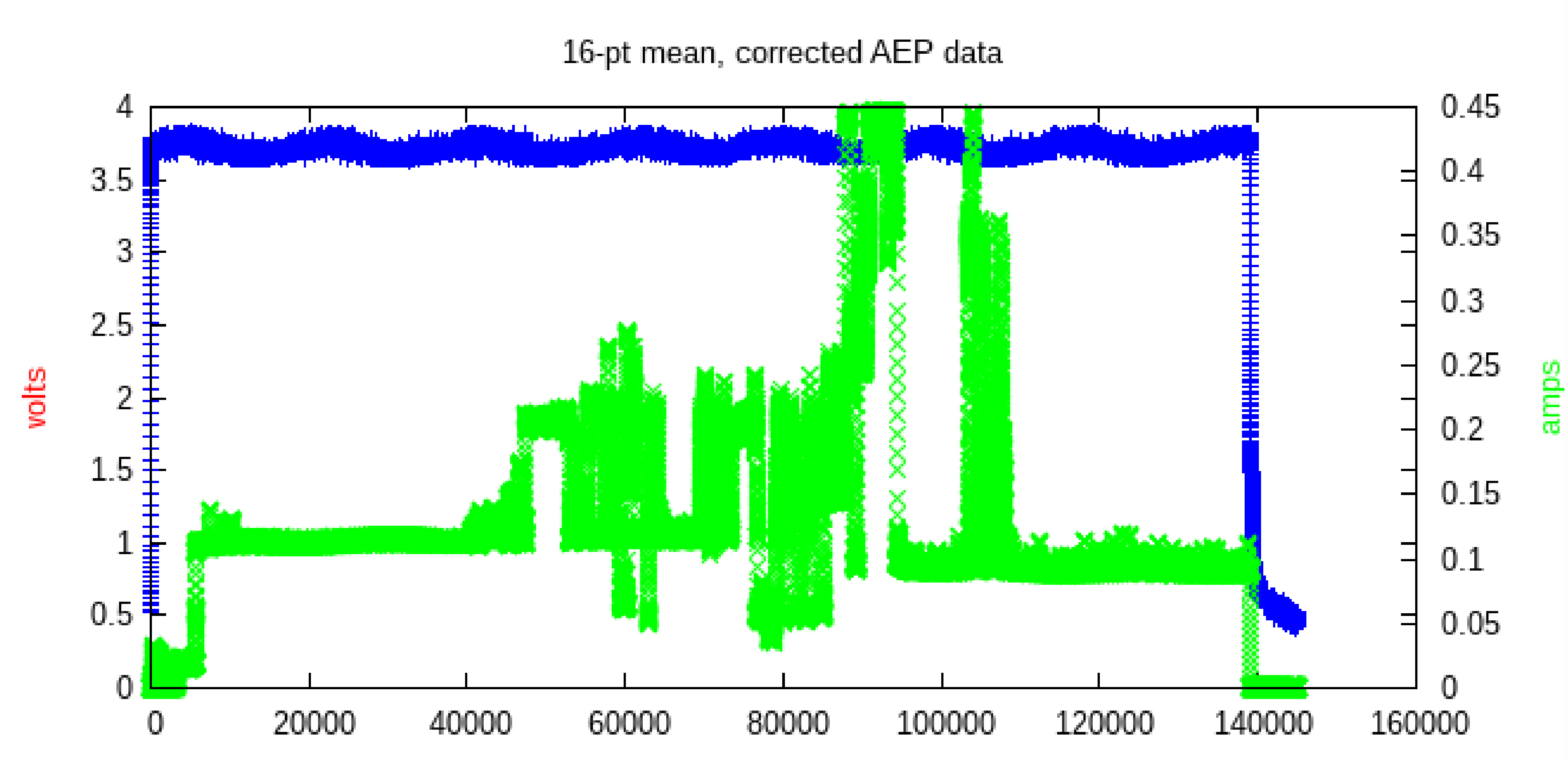
Interpretation

- Much better SNR
- Resolution improved to 280uA / ADC step
  - For static currents, high averaging will give even more precise results

Unaveraged, 470mR Linux



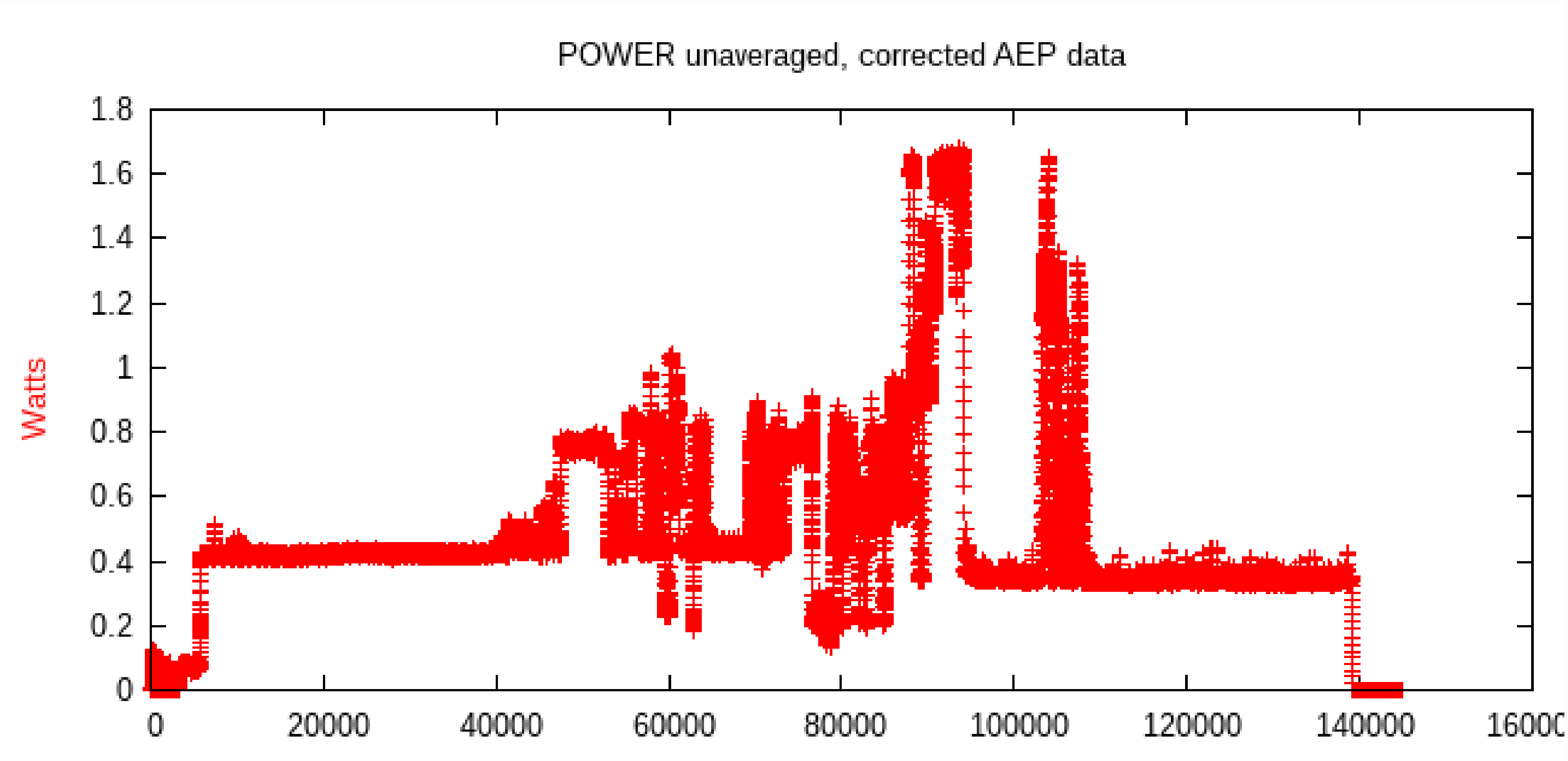
16-pt mean, 470mR Linux



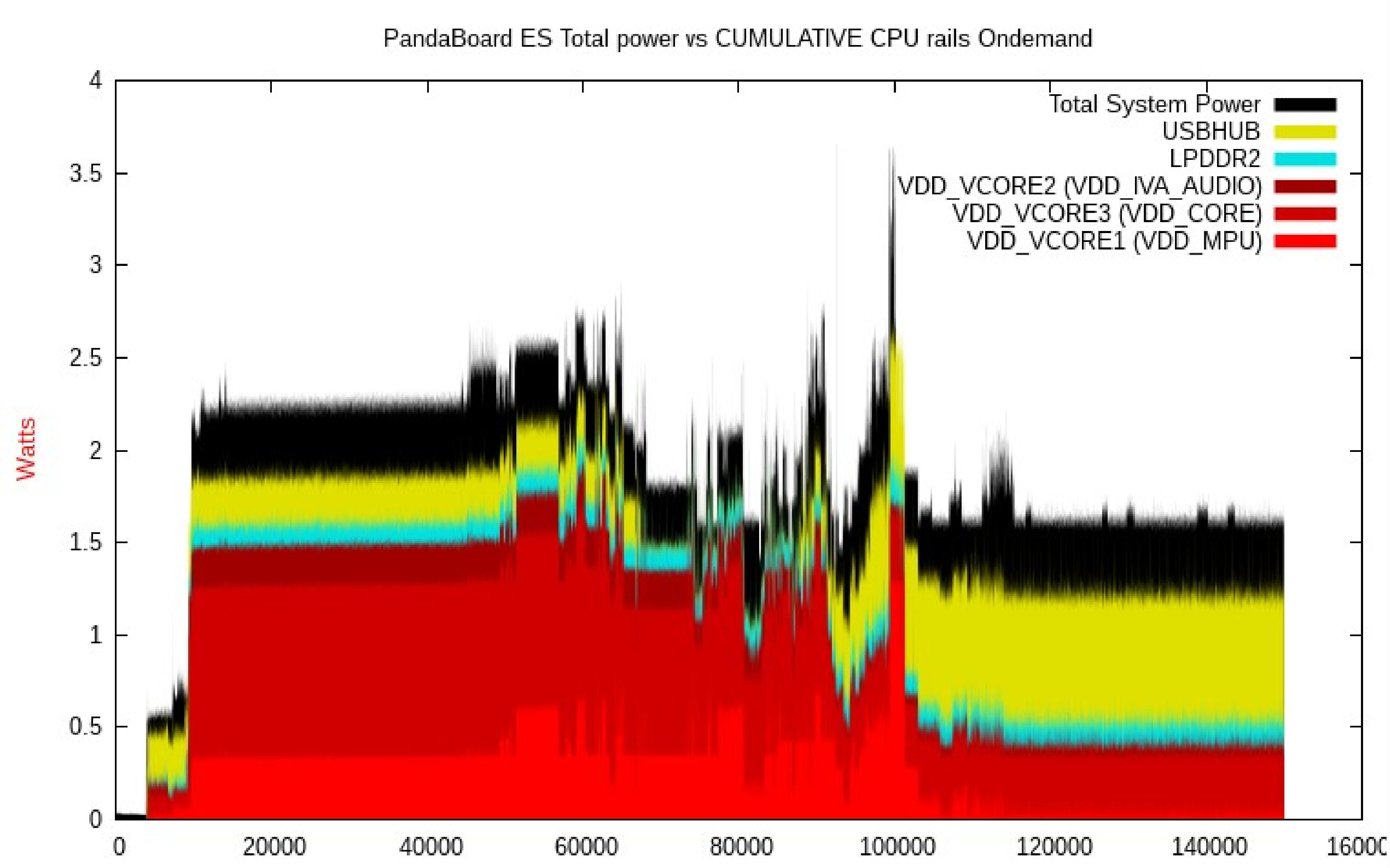
Interpretation

- 100us sample hides spikes
  - Short increases in current may be missed completely
  - Some rails dynamic load changes at 1GHz
  - 1 AEP sample covers 100,000 CPU clocks...
  - Any averaging makes it worse
- Rest of the graphs show power, not separate voltage and current

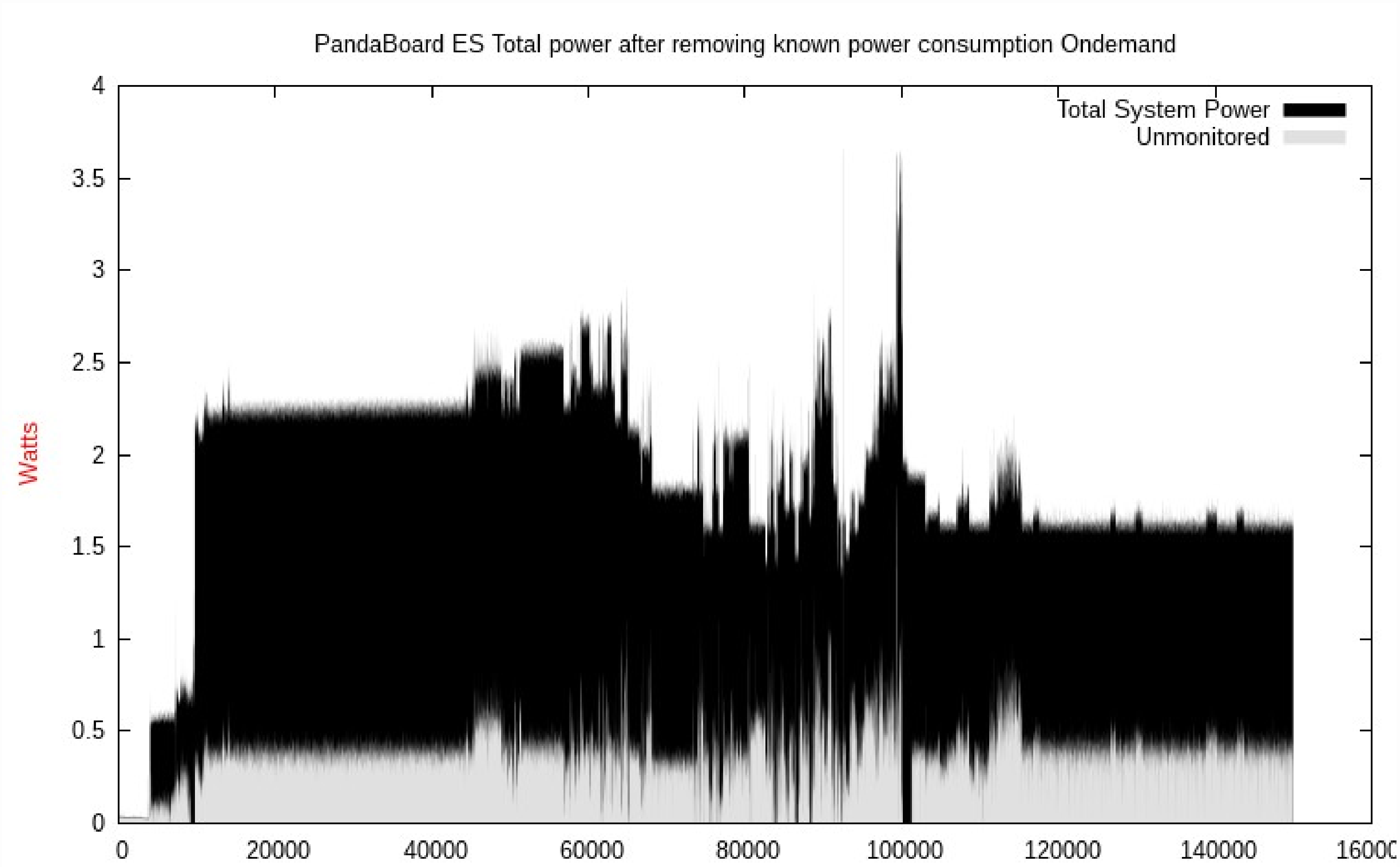
POWER, unaveraged, 470mR Linux



6 channel “cumulative” view



Unmonitored power vs DC jack view



Summary

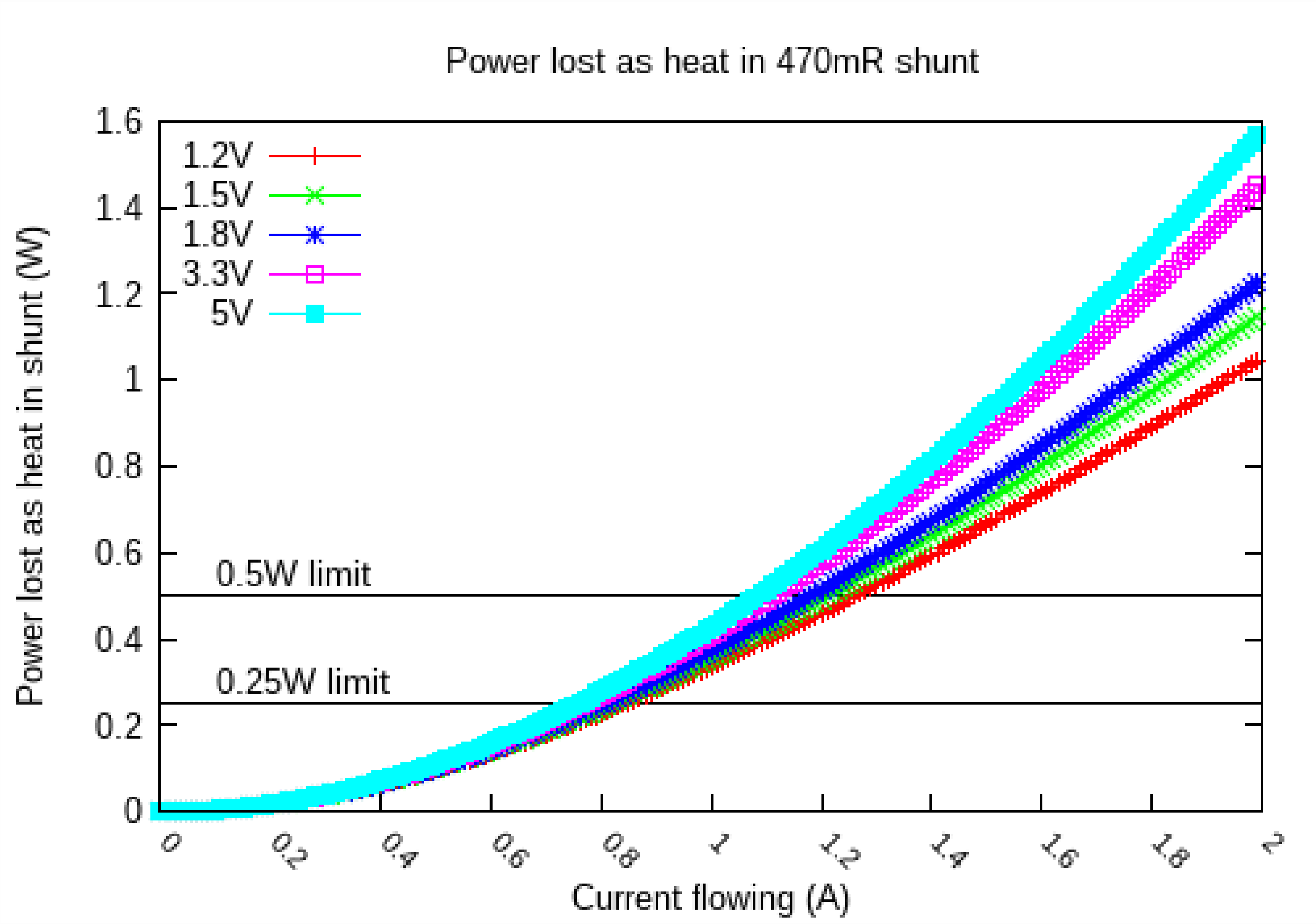
- ARM probe optimized to measure many amps
- Unable to use small value shunts well with normal currents
- Try 470mR shunt first
- Use 0.1% or 1% tolerance resistor if available

Caution

- Higher resistance shunt == more voltage drop... measured current flows through the voltage drop and P=IV for the shunt
- It has to dissipate the power as heat
- 470mR resistor used here rated 0.25W
- Only good for 0.7A at common voltages
- Higher power resistors available

Linaro

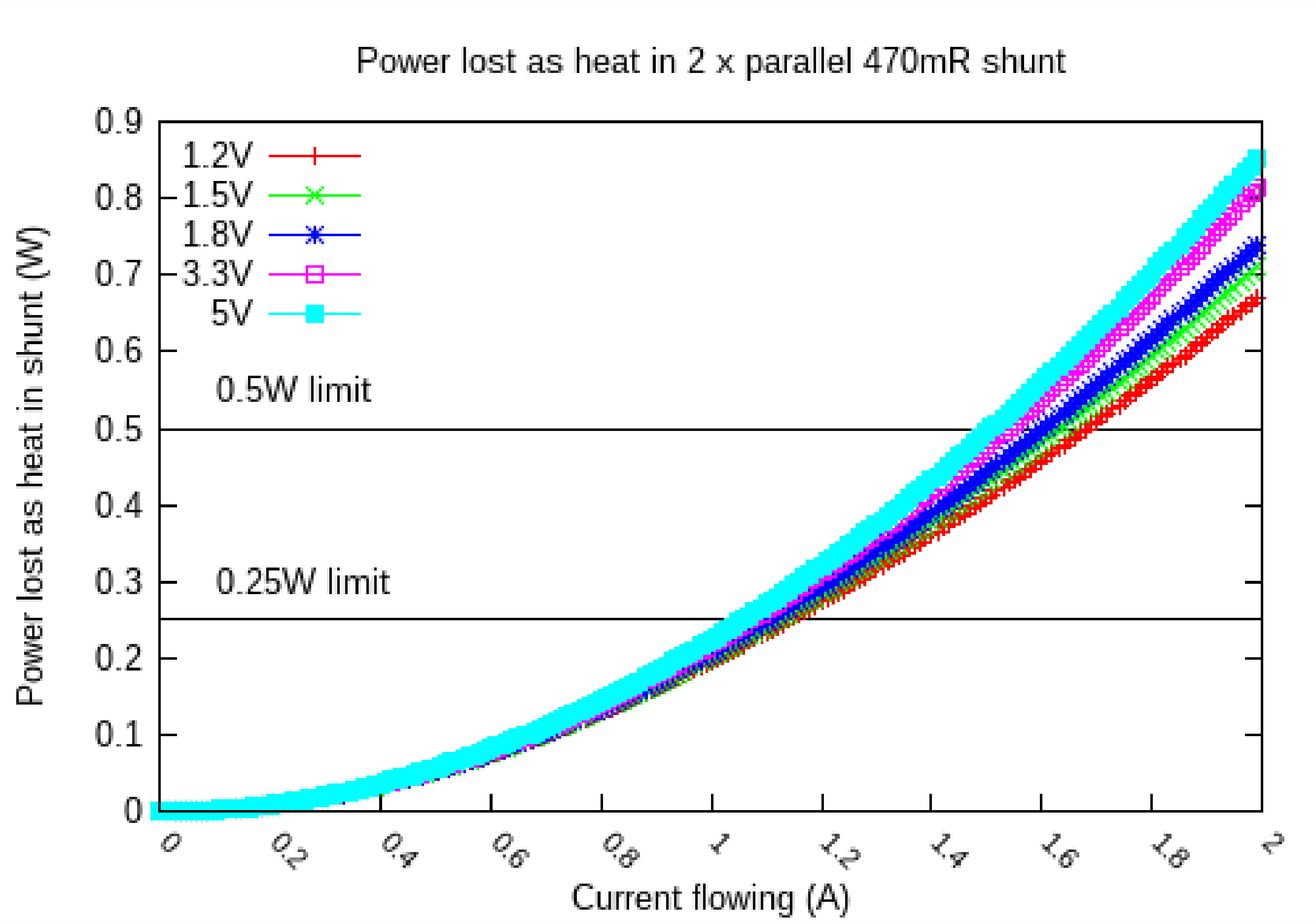
Power dropped in shunt



Parallel up shunts to cope

- 2 x 470mR, 0.25W shunts in parallel becomes 235mR, 0.5W capable
- The voltage drop is halved and the dissipation limit doubled (resolution /2...)
- Can cope with ~1.5A at the common voltages

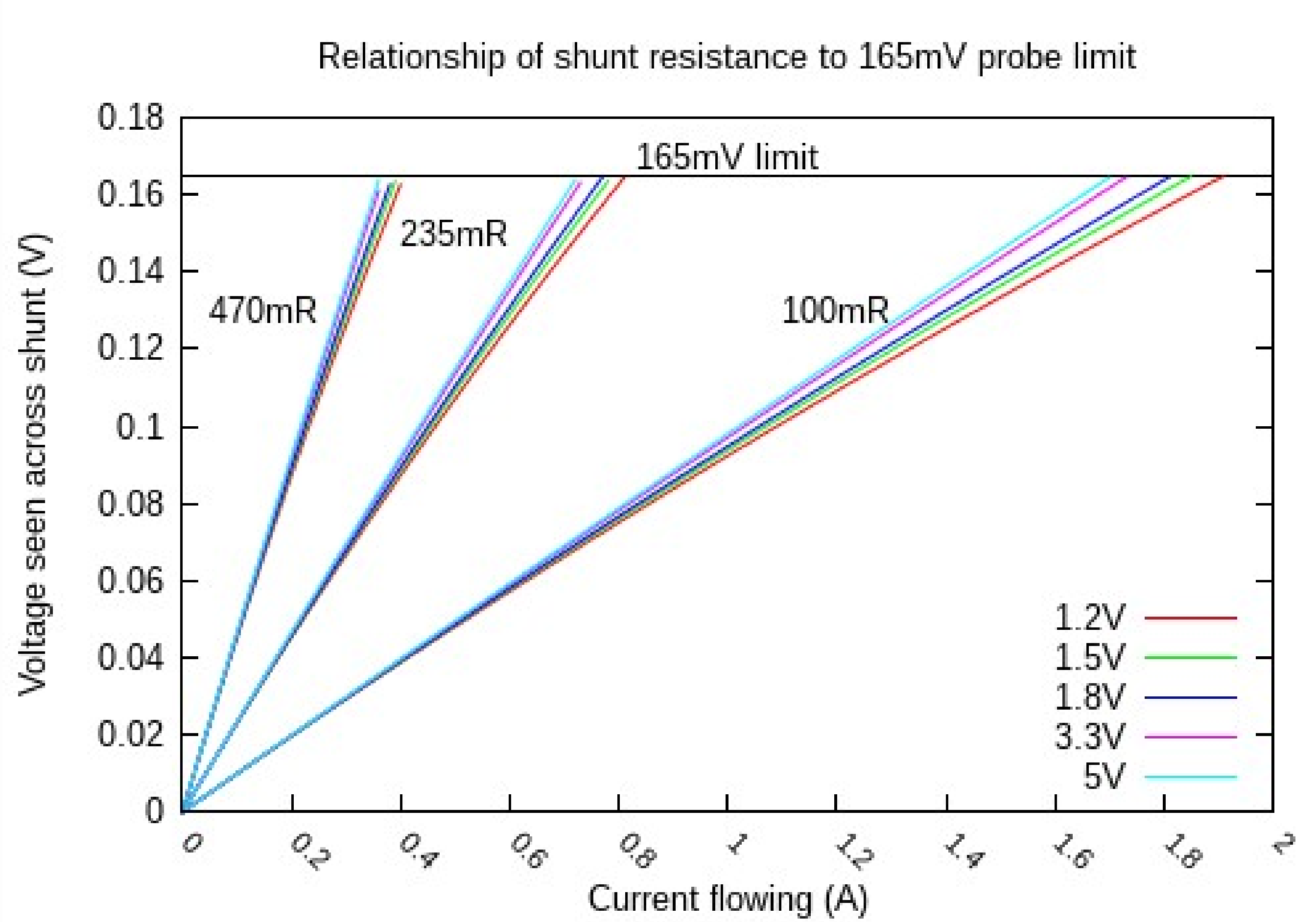
Power dropped in shunt



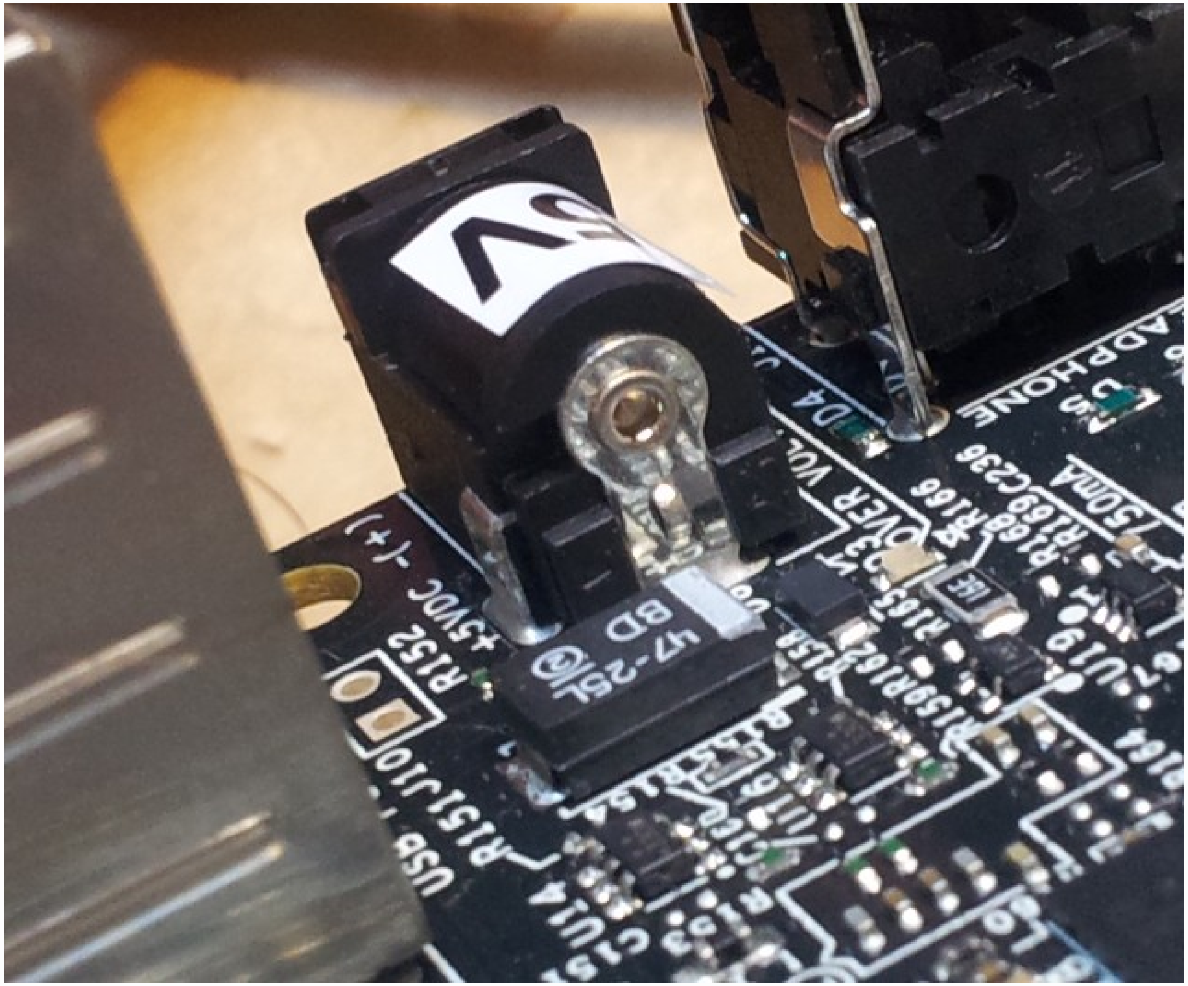
Shunt selection vs current range

- Probe can only measure up to 165mV across the shunt
- It means you have to select the shunt resistance according to the maximum current expected
- Following chart shows effect of common shunt values
  - Lower Rshunt --> high noise, low resolution

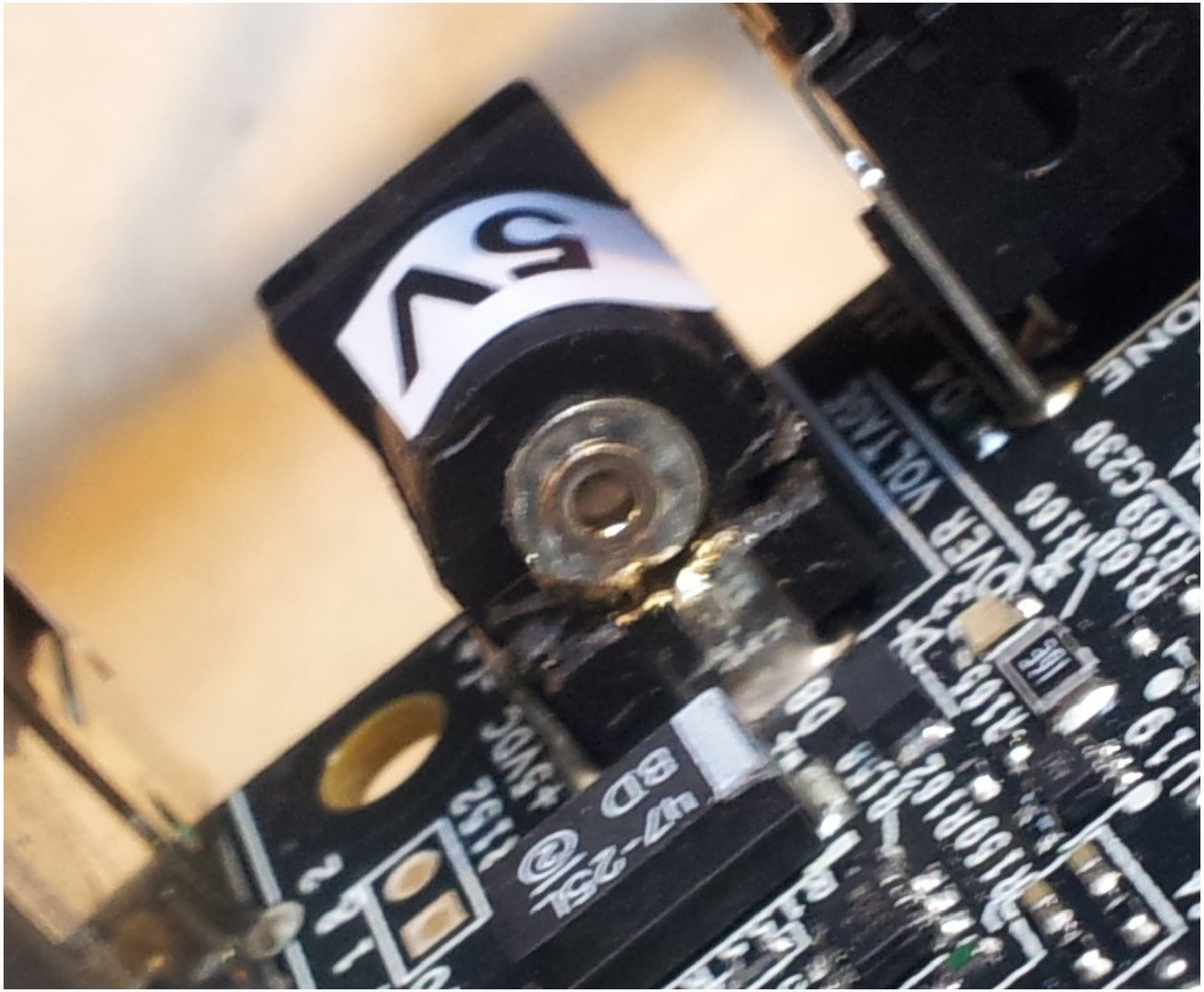
Effect of shunt resistance on range



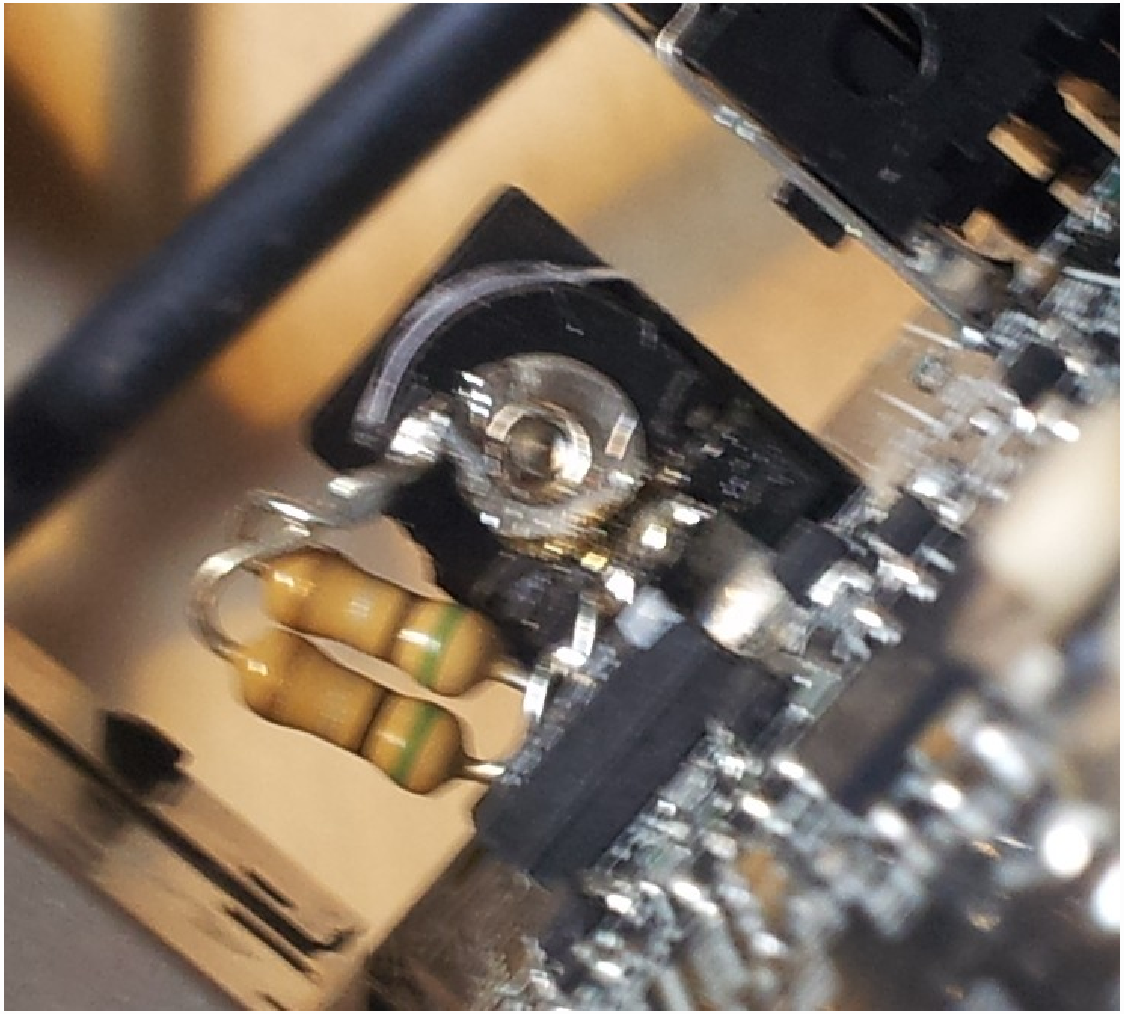
Attaching shunt to DC jack



Cut the + conductor



Apply shunts in parallel cf power



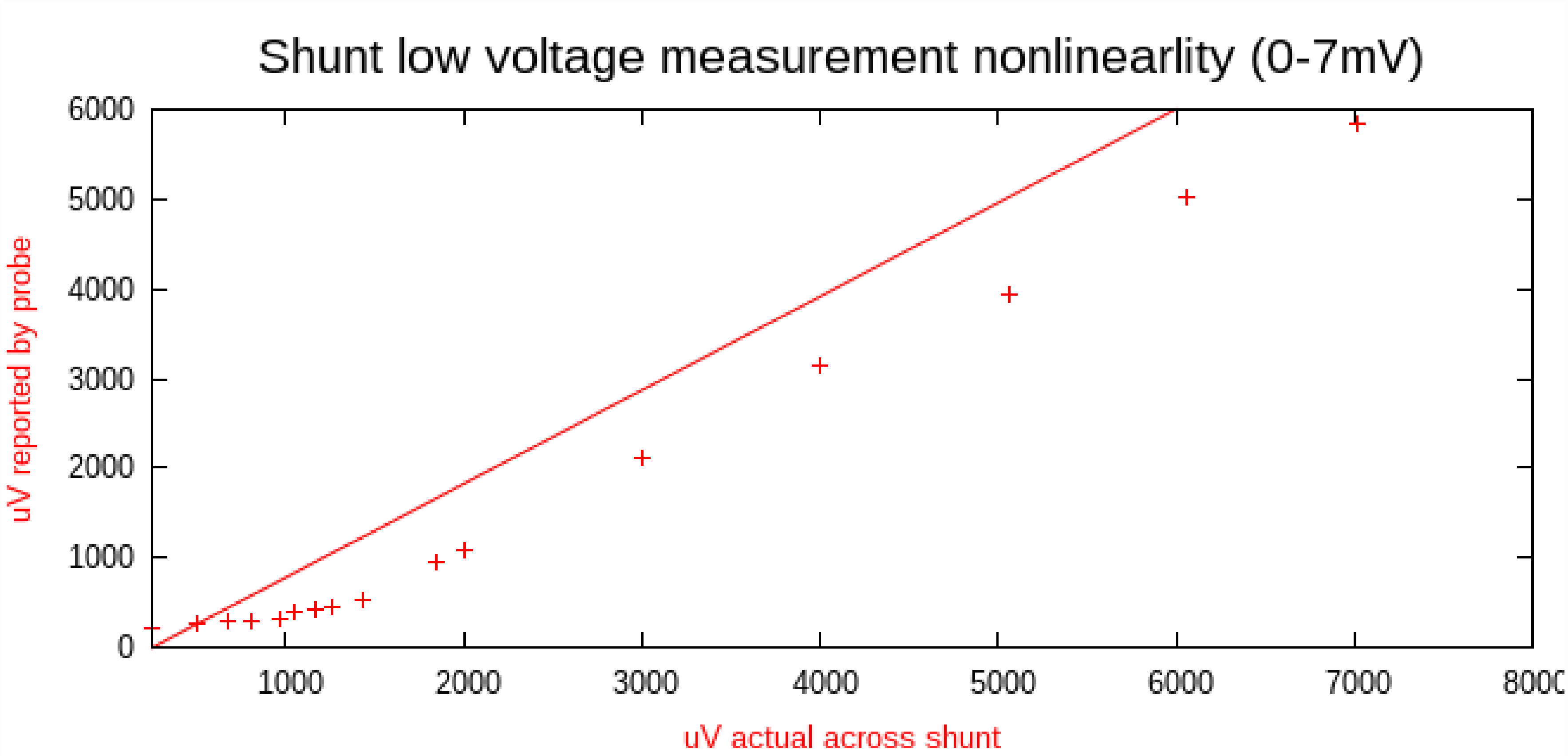
Arm Energy Probe Basics

- What it is
- Quick electronics 101 recap
- Measuring around regulators
- Wiring your board for AEP usage
- Some actual measurements
- Major sources of measurement error
- Linux Commandline tool

Linaro

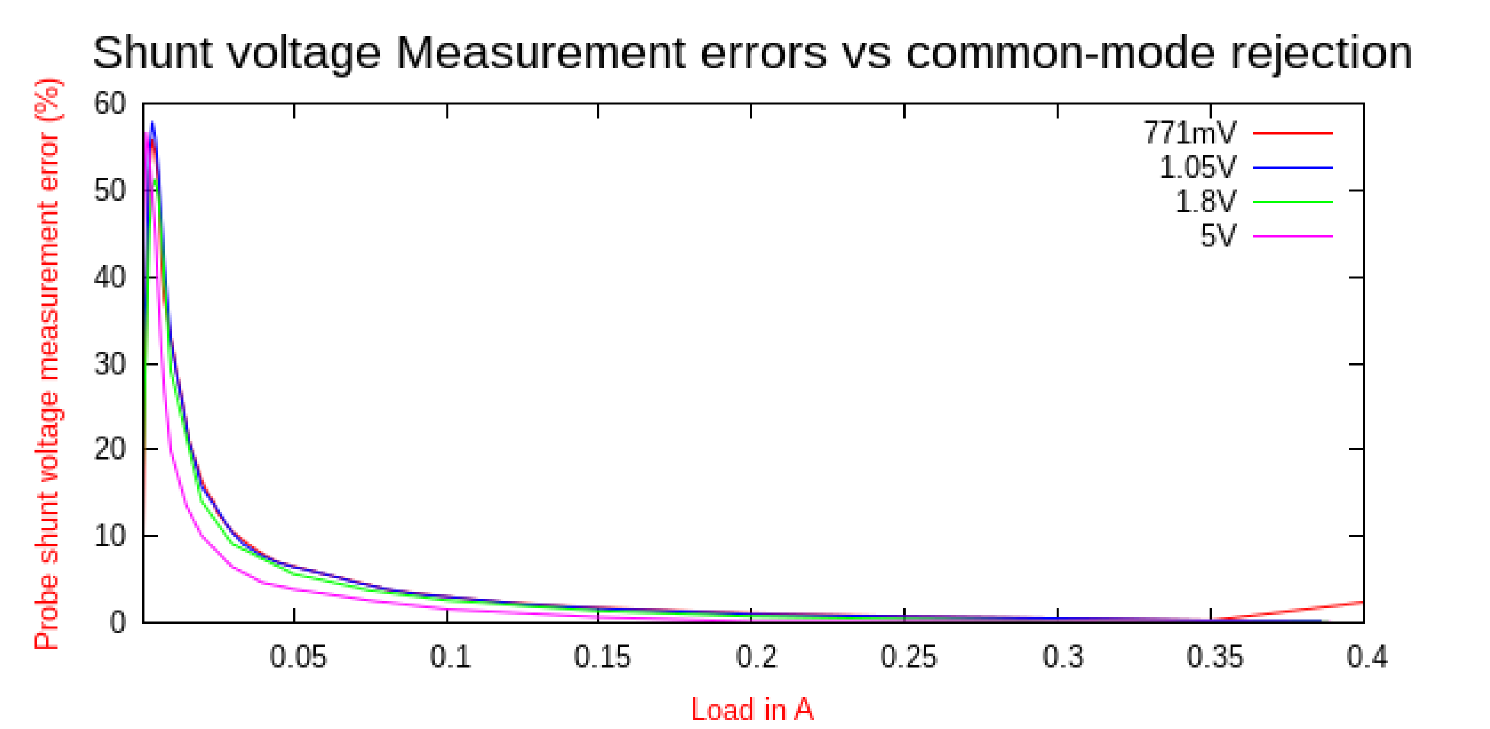
Performance characteristics

- Nonlinear at low cross-shunt voltages



Performance characteristics

- Raw results < 50mA (470mR shunt) **poor**
  - Very nonlinear, also temp, common-mode V?

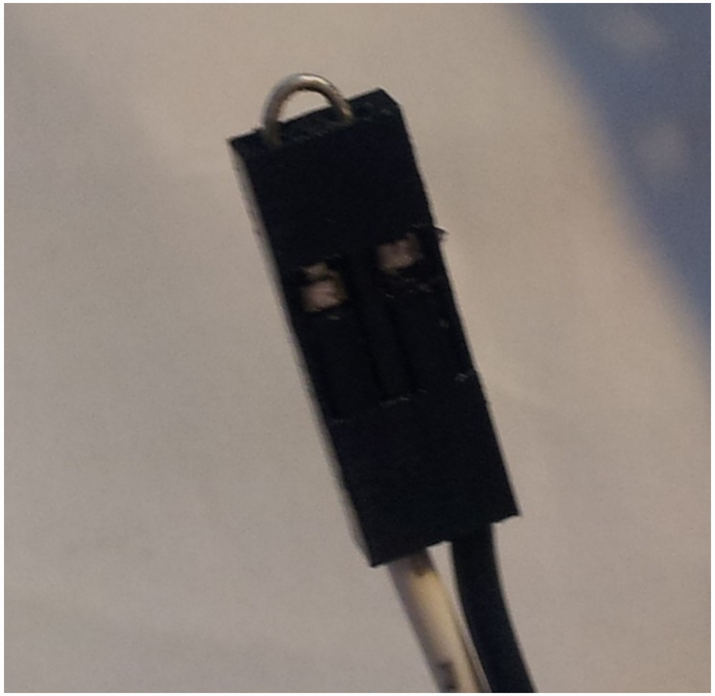


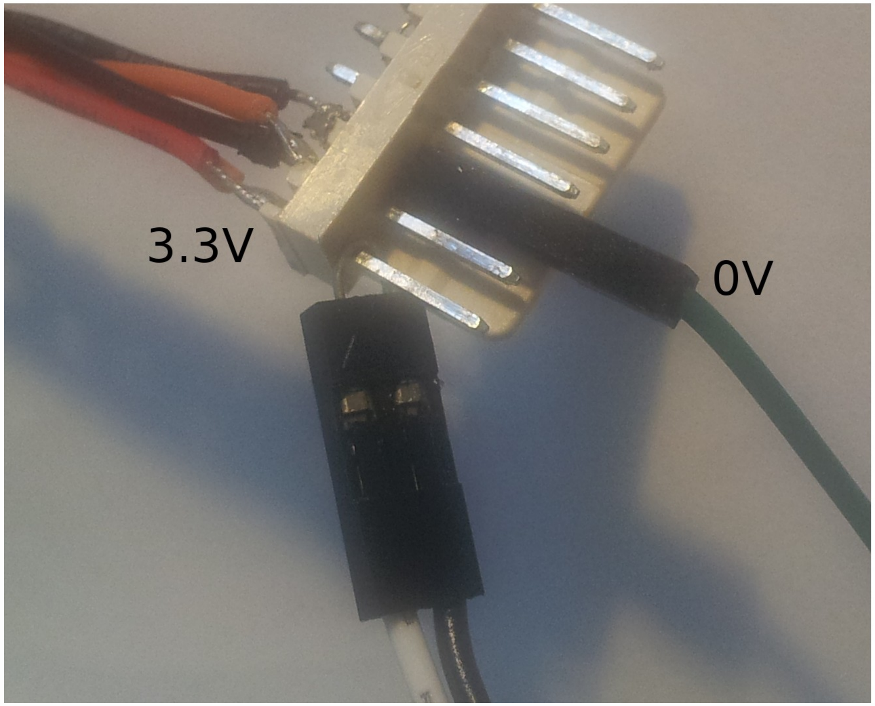
Analysis

- Raw probe current numbers below ~100mA (470R shunt) have significant error, that's **all common measurements**!
- Software correction needed
- Below 2mA measurements unrepeatable
  - Caused by unipolar ADC not reporting noise below 0, leading to increasing +ve offset
  - Temperature drift

Measure 0A offset error vs °C

- Short shunt inputs with wire, ~0R
- Connect shorting link to 3.3V, ~0A

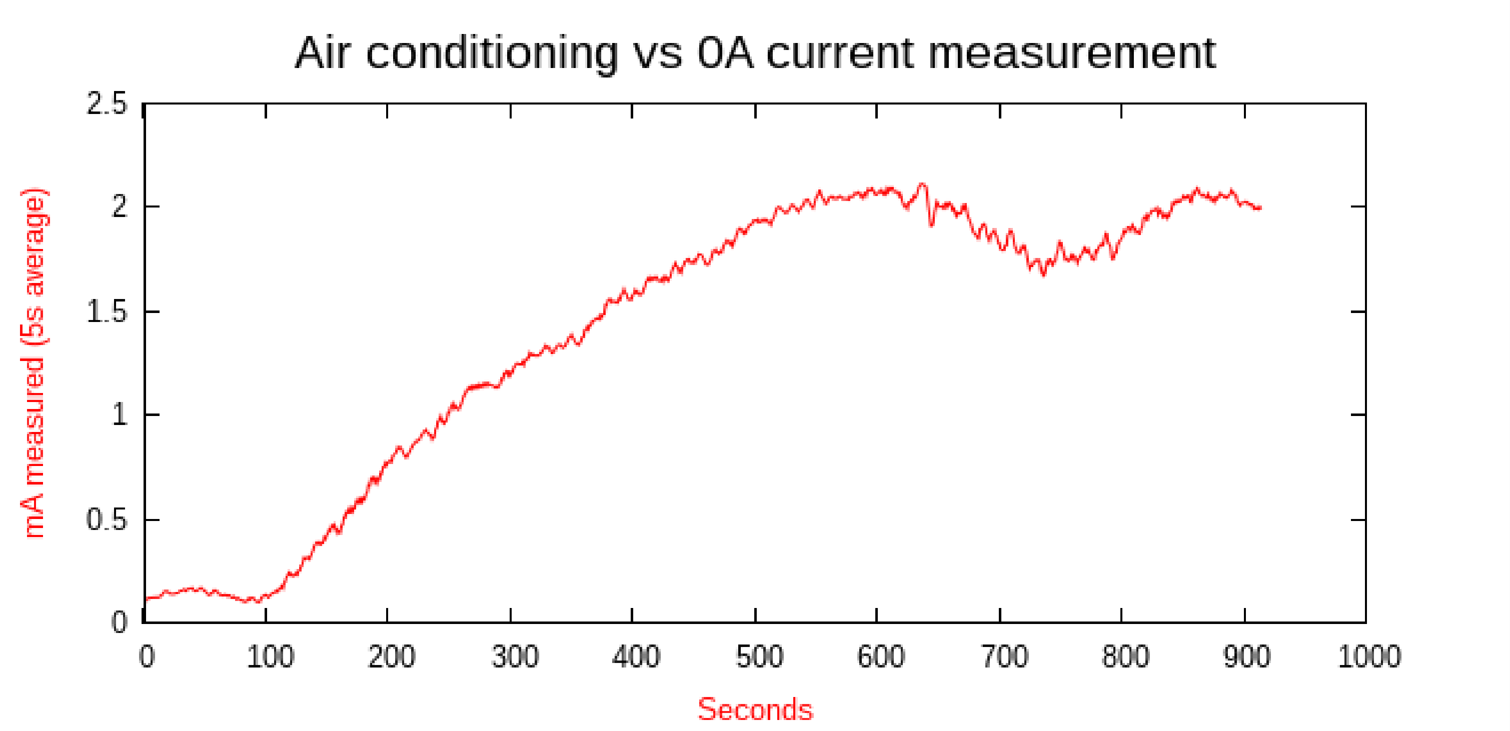




Linaro

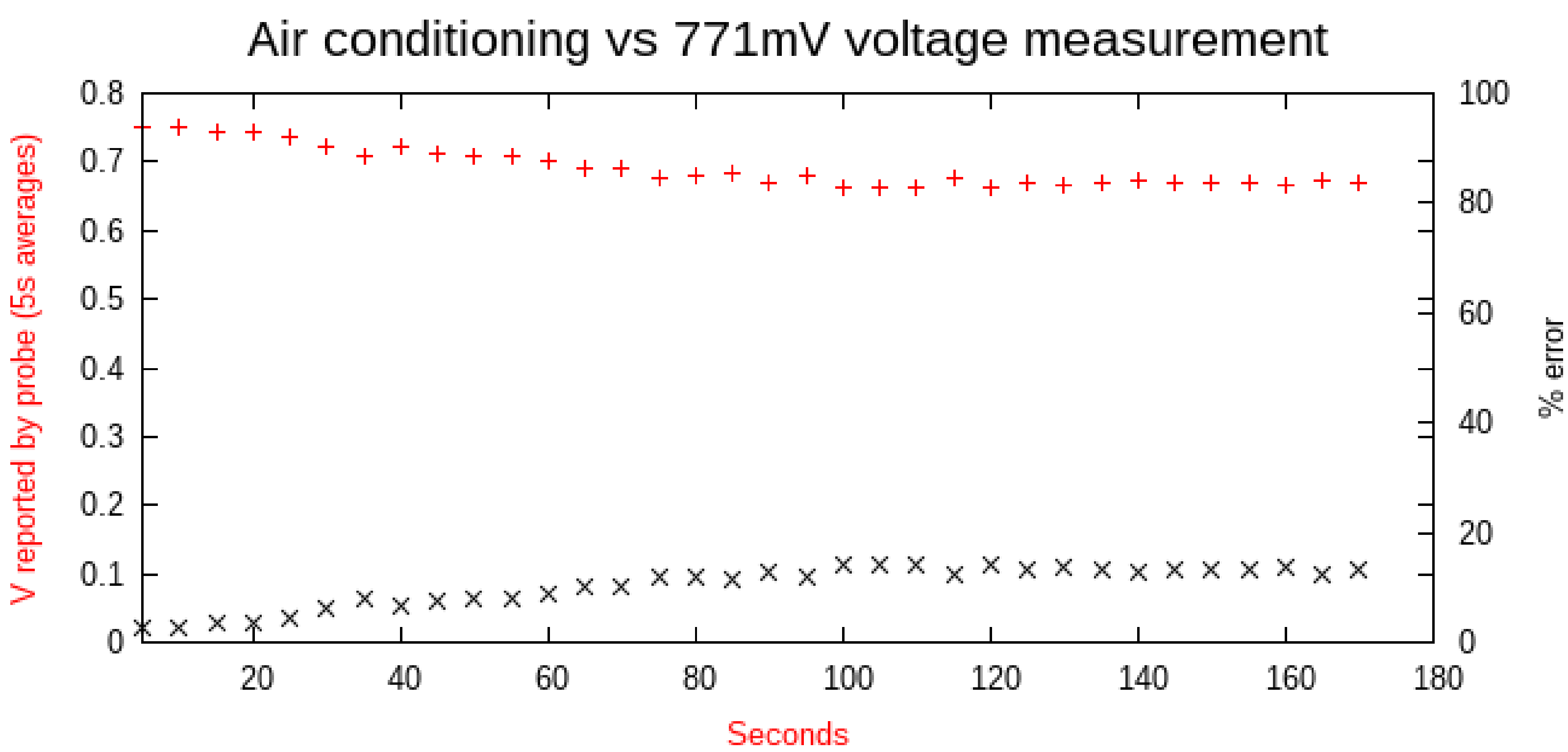
Temperature drift vs 0A current

- +1mV offset from 10°C drop in room temp
  - == “0A” error of +2.1mA with 470mR shunt !



Temperature vs voltage

- Same problem on voltage channel, air conditioning offsets by -15%



Analysis

- Current measurements have > 2mA offset uncertainty due to temperature
- Can combat it a bit by keeping temperature as stable as possible and measuring offset before real measurements (**--autozero**)
- Very low current measurements are going to be unrepeatable and have high error, due to air currents, sunlight on probe etc

Linaro

Correction model

- Software has error slope tables for 771mV and 6.02V
- Interpolates on both tables and then interpolate between the results based on measured common-mode voltage
- Software has offsets for voltage and current measurements per channel stored in config
  - Needs constant autozero to follow temperature

Shunt tolerance

- Actual shunt resistance is not known precisely
  - Typically +/- 1% or 5% at room temp
  - Also varies with temperature
  - At high currents, may self-heat
  - Error here directly affects current and power calculations

Arm Energy Probe Basics

- What it is
- Quick electronics 101 recap
- Measuring around regulators
- Wiring your board for AEP usage
- Some actual measurements
- Major sources of measurement error
- Linux Commandline tool

Linux arm-probe commandline tool

- Scripting and gnuplot-friendly
- Output on stdout (info on stderr)
  - Gnuplot style ascii floating point
    - `Sample# volts amps watts` (default) or
    - `Sample# watts` (**--justpower / -j**)
- If previous output piped into stdin, adds column
  - Friendly for combining multiple runs
    - `Sample# watts1 watts2`

Config file

- By default, ./config, # for comments
- First non-comment line is name for setup the config represents, eg, PandaES-B1-ANDY
- Then device path like /dev/ttyACM0
  - Followed by 3 x channel setups indented by space
  - <space> <channel name> <Rshunt> <Vslope> <Voffset for current> <fixed pretrigger> <"pretty name">

Linaro

Config file

- Example config file
  - PandaBoardES-B1-ANDY
  - /dev/ttyACM0
    - VDD_VCORE1 0.470 1 -0.000253 0 VCORE1/MPU
    - VDD_VCORE2 0.470 1 -0.000262 0 VCORE2/IVA_AUDIO
    - VDD_VCORE3 0.470 1 -0.000304 0 VCORE3/CORE
- Space indent at start of channel lines is critical!
- Can define multiple probes in one config

Channel Autozero

- **--autozero / -z** takes 5s average on selected channel and writes the voltage and current seen into config for that channel
  - `arm-probe -c VDD_VCORE1 -z`
- During this, short both channel sense lines to 0V line so you are measuring 0V, 0A offset
  - Unfortunately the offset varies by channel and temperature...

Channel names

- Select the channel you will capture using the channel name from the first config column
  - **--channel / -c** <channel name>
  - If you have multiple probes, will find the correct probe according to /dev/ttyACM section in config
  - Take care to connect probes to USB in same order
  - “Pretty name” is used for channel in output
    - Gnuplot needs '_' to be '_' in pretty name

Trigger threshold and filtering

- Specify “trigger” threshold in mV or mW
  - In volts (**--mvtrigger / -q** <mV>), and / or
  - In power (**--mwtrigger / -w** <mW>)
  - Default, 400mV
- Specify how long trigger must remain true to be accepted (**--ustrigfilter / -f** <us>)
  - Default, 400us
  - 200mV / mW hysteresis applied

Pretrigger

- Until trigger conditions seen, probe in “pretrigger” state
  - Nothing on stdout
  - Displays live volts / amps / watts on stderr
- Can capture to pretrigger buffer so you can see what led up to trigger event
  - (**--mspretrigger / -p** <ms to capture>)
  - Pretrigger buffer flushed on stdout first

Linaro

Trigger holdoff

- Amount of time to wait after trigger event seen before actually triggering
  - **--mstrighold / -t** <ms>
  - Default 0ms
  - Allows you to target events that occur a fixed period after, eg, power-on, without capturing everything before

Capture length and autoexit

- Can define how long to capture data
  - **--mslength / -l** <ms> (default: no limit)
    - Needed when combining captures on stdin
  - **--exitoncap / -x** exits the program when this capture limit is reached
  - **--offexit / -o** waits for trigger conditions to be false before exiting the program
    - Perfect to sync multiple scripted captures so next capture can trigger at power-on

Averaging

- Can apply mean averaging
  - **--mean / -m** <samples> (default: none)
    - Use with a large mean buffer to get single figure results instead of graph data
    - Ten samples per ms, so **-m 50000 -l 5000** gives perfectly mean-averaged 5s capture
    - Set your device to loop performing use-case
    - Choose a capture interval several times one loop period for best accuracy

Decimation

- Reduces output to once per n samples
  - **--decimate / -d** <samples> (default: 1)
    - Allows long period monitoring of rails without being overwhelmed by data
    - Can be combined with mean averaging, eg **-m 50000 -d 50000** issues one new fully averaged sample every five seconds
    - Sample# in output still counts real input samples, so you can follow real time

Multicapture helper scripts

- Two common multichannel capture cases supported by scripts
  - **aep-capture.sh** - capturing on all channels during or after boot for fixed period
    - Manages observing power cycling of target for each channel and synchronzing captures
  - **aep-average.sh** – for each channel captures unsynchronized sample interval and fully mean-averages the power
    - Provides single power number for use-case

Gnuplot scripts

- Gnuplot scripts will need customizing for your setup but have the basics
  - aeplot-average -
  - (continued next time...)