

# ACME performance and precision with multiple Powercapes

using HWMON/Sigrok

# Measurement principle

- Measure the time from update to update for the first probe @0x40 and verify there is always enough time to read out all registers until next sample time is due
- Write trace “a” when entering *update\_device*
- Write trace “b” after completion of i2c operations
- Measure time  $b_k - a_k$  ***cost of i2c transfer***
- Measure time  $b_k - b_{k-1}$  ***achievable sampling period***
- Do this for both versions of the hwmon driver

# Test commands

- **Data acquisition**

*trace-cmd start -p nop*

*sigrok-cli -d baylibre-acme --samples 200 --config samplerate=\${freq}*

*trace-cmd stop*

*trace-cmd extract*

- **Possible frequency values with sigrok-cli**

Libsigrok/src/hardware/baylibre-acme will set the update interval through sysfs, possible values are discrete however, resulting of the possible averaging parameter times the constant Integration Time setting of 2.2ms

Averaging values	1	4	16	64	128	512	1024
Periods (s)	0.0022	0.0088	0.0352	0.1408	0.2816	1.1264	2.2528
Freq (Hz)	454	113.63	28.41	7.1	3.55	0.887	0.443
<b>Tested (Hz)</b>	<b>454</b>	<b>113</b>	<b>28</b>	<b>7</b>			

- **Visualization**

*Done with kernelshark*

# Preliminary discussion

- **Expected achievable sample rates**

With the 4.3 hwmon driver, upon accessing sysfs with the show function, all 8 registers are read once per programmed update interval. The 4.4 driver will read only the required value register.

Reading all registers takes at best  $500\mu\text{s} * 8 = 4\text{ms}$  per probe, but this can be much longer if the threaded IQR completion is delayed (i2c-omap).

The table bellow provides the **best sampling frequency (Hz)** based on the number of connected probes, and the driver version, where no old value is repeated.

#probes #channels	1 Probe	2 Probes	3 Probes	4 Probes	5 Probes	6 Probes	7 Probes	8 Probes
Driver 4.3 Any ch #	113	113	28	28	28	28	28	28
Driver 4.4 1 channel	454	454	454	454	113	113	28	28
Driver 4.4 2 channels	454	454	454	113	113	113	28	28
Driver 4.4 3 channels	454	454	113	113	113	28	28	28
Driver 4.4 4 channels	454	454	113	113	28	28	28	28

# 4.3 Driver, 8 probes, 7 Hz

- **Command and driver setup trace verification**

*sigrok-cli -d baylibre-acme --samples 200 --config samplerate=7*

*[ 4232.068759] Setup averaging = 64*

$64 * 2.2\text{ms} = 141\text{ms} \sim 7.1\text{ Hz}$  **OK**

*[ 4232.072163] Actual interval = 140800 us*

- **Results**

Elapsed time between sigrok-cli readout:

~200 ms **KO, see [1]**

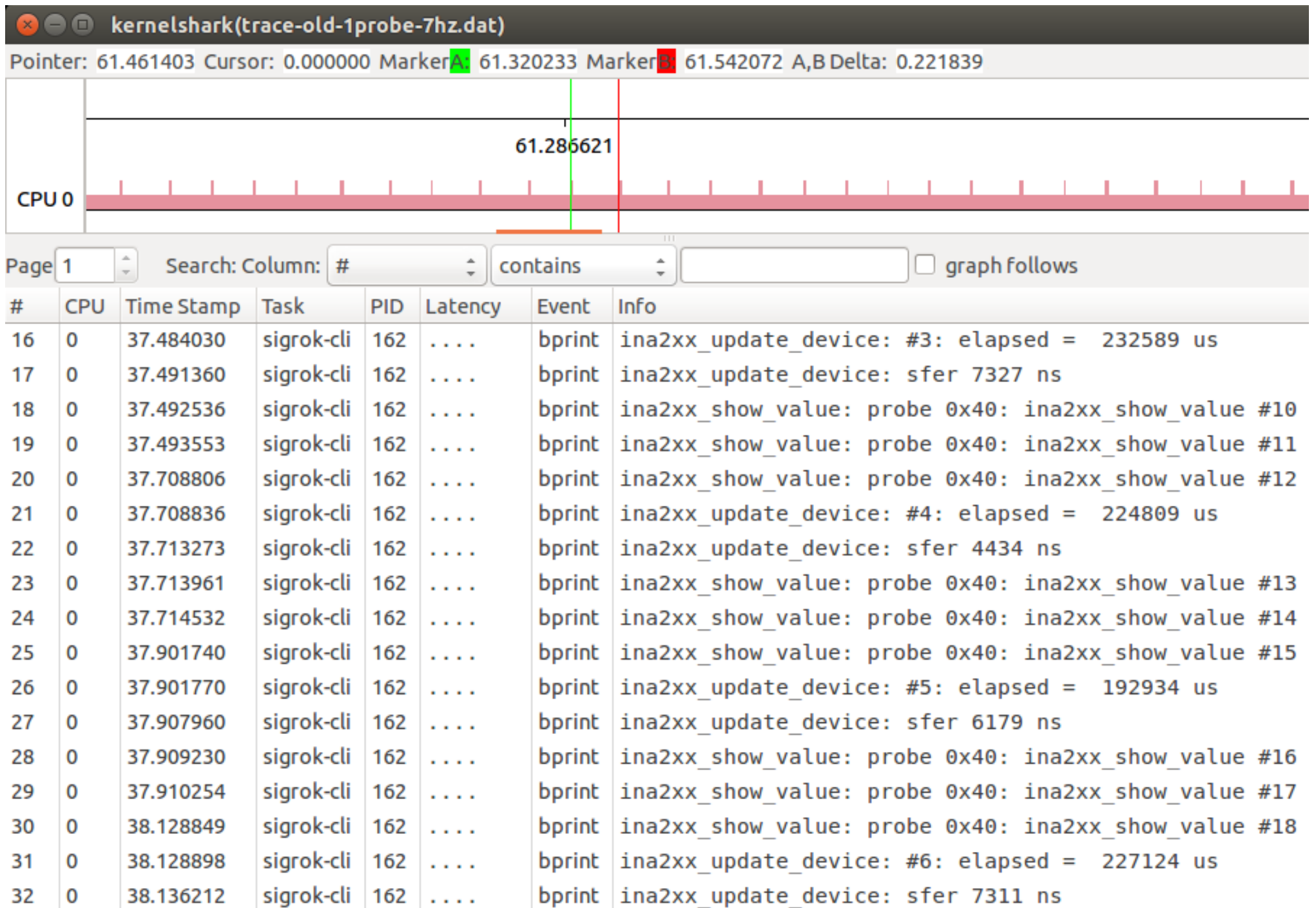
Elapsed time between fresh values:

~200 ms **KO, see [2]**

- **Comment**

**[1] with driver 4.3 bl\_acme\_receive\_data requires a small patch to prevent introducing an additional delay.**

**[2] driver 4.3 will prevent the read out every second time, this might be related to how timer events for sleeping are managed.**



The remaining tests and measurements are done with hwmon driver v4.4 (the one using regmap) because v4.3 is even worse, and linux-next will provide v4.4 anyhow.

## 4.4 Driver, 8 probes, 28 Hz

- **Command and driver setup trace verification**

```
sigrok-cli -d baylibre-acme --samples 200 --config samplerate=28
```

```
[ 39.494589] Setup interval = 35 us  
[ 39.498304] Setup averaging = 16  
[ 39.501685] Actual interval = 35200 us          OK
```

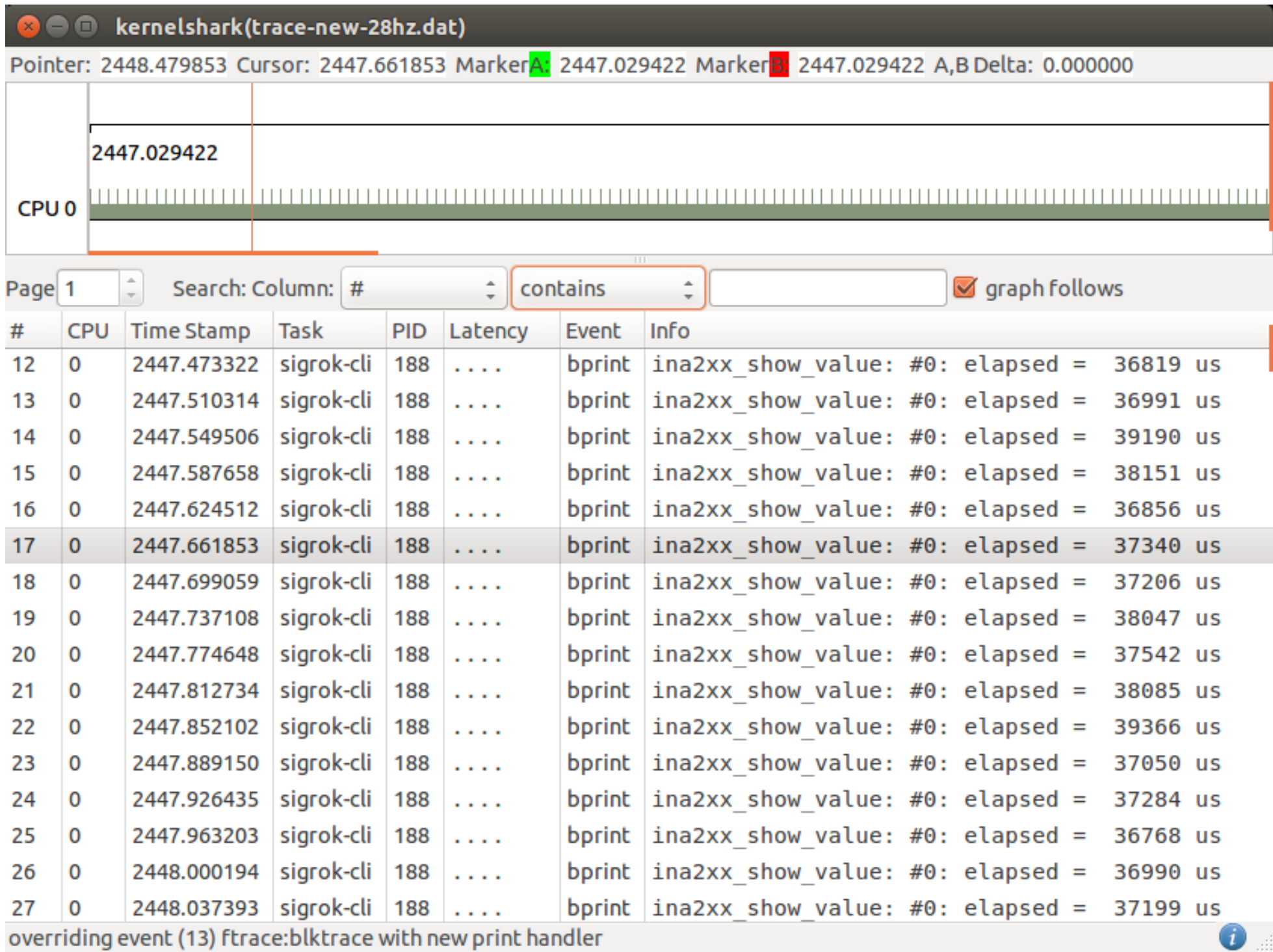
- **Results**

Elapsed time between fresh values:                      ~37 ms    (sigrok timer)      OK

- **Comment**

Kernelshark shows very regular readout times for probe0/vshunt, meaning that this configuration can be sustained in terms of performances.





## 4.4 Driver, 8 probes, 113 Hz

- **Command and driver setup trace verification**

```
sigrok-cli -d baylibre-acme --samples 200 --config samplerate=113
```

```
[ 4418.708234] Setup interval = 8 us  
[ 4418.711755] Setup averaging = 4  
[ 4418.715047] Actual interval = 8800 us      OK
```

- **Results**

Elapsed time between fresh values:                      32 ms      (sigrok timer)      **KO**

- **Comment**

As expected, setting a readout period under the required time spent in i2c-omap will result in an erratic sample rate (bottleneck effect).

**Sample rate above 28 Hz result in time stretching (repeated samples).**

# Conclusions

*Using 8 probes is functional, but with sample freq restrictions (28 Hz max) for accurate signal properties.*

## Recommended changes in Current End-user package

It is important to provide users with:

- the version of sigrok that uses the timer device.
- the hwmon driver 4.4, as it will allow for better sampling frequencies for simple use-cases (see table in slide 4)

## Changes in Documentation

I recommend providing table 4 (remove the driver-4.3 line) to the user, with explanation Using higher sampling rates is always possible to avoid a slow GUI refresh rate, but any sampling freq above the compatible one will result in time stretching (repeated samples). Signal process software can probably do a better job with proper interpolation algos.

## Corrective action and evolutions

- **Support smaller integration times with hwmon than the fixed 2.2ms to increase sample freq, and switch to IIO driver when sigrok plugging is available (no sysfs)**
- **Investigate using a bitclock higher than 400kHz, might not be possible with the current gpio routers ? Might be difficult to upstream**
- **Investigate using a non-threaded irq instead of a threaded\_irq + completion routine for i2c-omap. Might be difficult to upstream.**

