Not-So-Smart Power Plugs, but at Least No Cloud Required

Christian Leitold

christian.leitold@gmail.com

Ostrava Python Pizza, 21 February 2026

I just want to track my energy use, not have Jeff Bezos track it as well

A **smart plug** should

- switch on and off remotely,
- monitor energy consumption,
- and also log everything for later analysis.

Amazon Basics 1 x Smart Wi-Fi Socket for EU Socket, Indoor Use Only Compatible with Alexa, White

Visit the Amazon Basics Store

4.4 ★★★★½ (2,489) | Search this page

2K+ bought in past month

-18% €9⁰⁸

Was: €11.09

✓prime **Tomorrow**

Prices for items sold by Amazon include VAT. Depending on your delivery address, VAT may vary at Checkout. For other items, please see details.

10% off on any 4 qualifying items Shop items ›

Size Name: **1er-Pack**

1er-Pack	2.3*2.3*3.36 in	4er-Pack
€9.08 €11.09	€34.32 FREE Delivery **Tuesday**	See available options

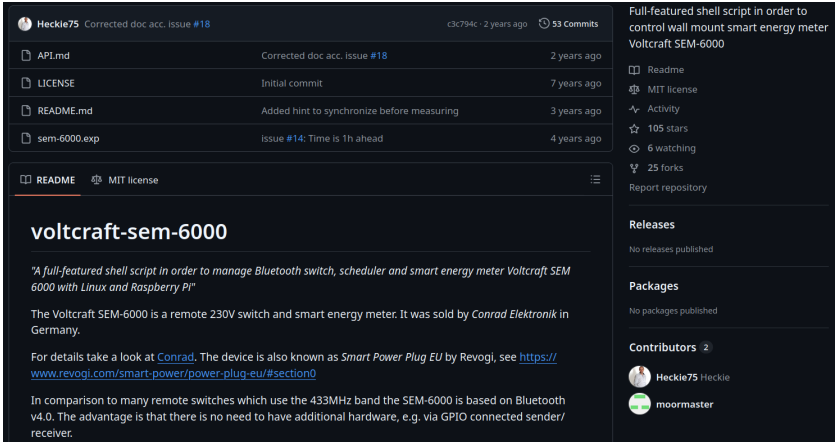
A no-cloud alternative

VOLTCRAFT SEM6000

- A Bluetooth-only *smart plug*.
- With an official Android and iPhone app...
- ...but luckily unofficial open source code is available as well!
- Side note: it is *expensive*!

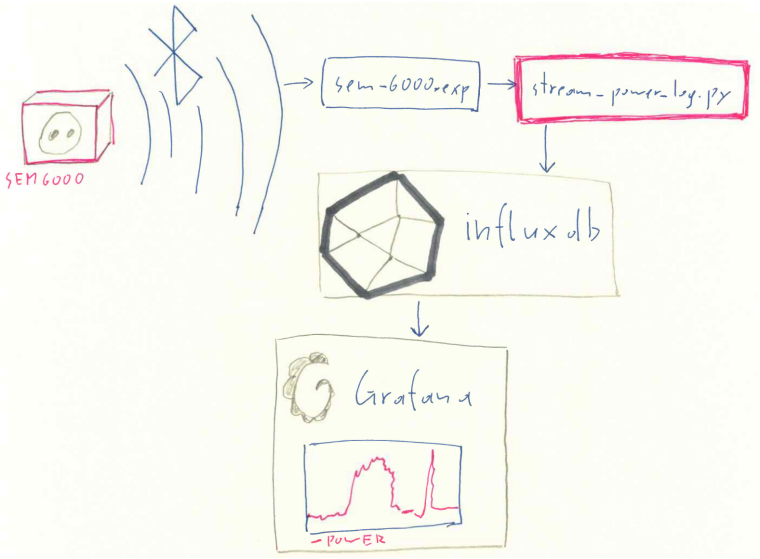
https://www.conrad.at/de/p/voltcraft-sem6000-energiekosten-messgeraet-bluetooth-schnittstelle-datenexport-datenloggerfunktion-trms-stromtarif-e-1558906.html

voltcraft-sem-6000 by Heckie75



https://github.com/Heckie75/voltcraft-sem-6000

Putting the pieces together

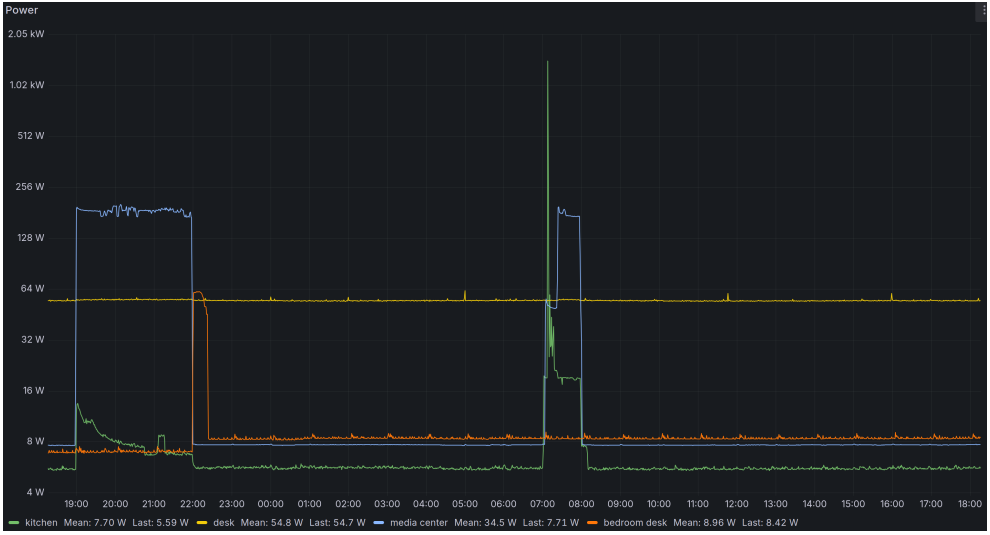


Some lessons learned

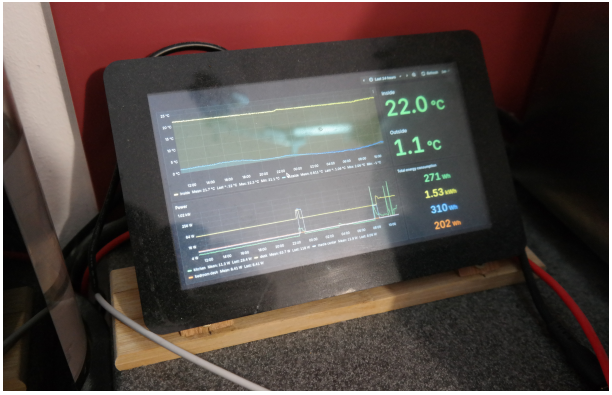
- Set the Bluetooth device inside the Expect script!
- It will work just fine for some time...
- ...until it does not any more.
- Fix: queues, threads, sanity checks for invalid or stuck values.
- Rinse and repeat!
- Caveat: security is not great.

```
def measure_stream_function(device: str, the_queue: queue.SimpleQueue):
    exe = '/usr/bin/expect'
    measure_script = '/home/christian/projects/voltcraft-sem-6000/sem-6000.exp'
    while True:
        process = subprocess.Popen(
            [exe, measure_script, ALL_PLUG_IDS[device], '0000',  '--measure'],
            stdout=subprocess.PIPE
        )
        power_ringbuffer = queue.deque(maxlen=RING_BUFFER_SIZE)
        sent_power_ringbuffer = queue.deque(maxlen=RING_BUFFER_SIZE)
        invalid_ringbuffer = queue.deque(maxlen=RING_BUFFER_SIZE)
        n_repetitions = 0

        while True:
            line = process.stdout.readline()

            line = line.decode().strip()
            columns = line.split('\t')
            try:
                power = columns[4]
            except IndexError:
                print(f'# No power reading found for {device}, terminating.')
                process.kill()
                break

            real_power = float(power)
```

Some final results, visualized using a Grafana panel



Is it all worth it in the end?



Python script on Github:
`https://github.com/leitold/powerlogger`