

The chip that draws more power uses less of it

October 19, 2026 · by Mark Bartlett

[Download this post as PDF](#)

InsiderLLM Weekly issue 26 – October 19, 2026

The ESP32-P4 board draws more power than the ESP32-S3, and it still spends about 46% less energy on every token. More on that below. Also this week: used GPU prices on 39 guides caught up with the market, llama.cpp removed flags our older commands still use, and DeepSeek, Meta, NVIDIA and Mistral all shipped weights.

New This Week

- **ESP32-P4 vs ESP32-S3: Energy per Token, Measured at the USB Port.** The main piece, below.

Updated:

- **Used GPU prices, 39 guides:** a used 3090 now runs about \$1,500, a 4090 about \$3,000 and a 5090 about \$6,000 (eBay US sold, October 15). If you priced a build from our pages before October, redo the math. ([RTX 5090 page](#); [used 3090 buying guide](#))
- **llama.cpp flags, 15 guides:** `--no-mmap` and `--mlock` are now `--load-mode`, and llama-server's default port is 9931 from v0.7.0. Reader commands are updated; benchmark commands stay as run. ([CPU-only guide](#))

Quick Hits

- **Meta and DeepSeek ship weights you can't run.** Muse Spark Open (816B) and DeepSeek V4.1 Flash 1017 (552B) are both open, but the smallest quants are 176 GB and 233 GB. That's 192 GB-of-RAM territory, not a 3090. ([Meta](#); [DeepSeek](#); [unsloth GGUF](#); [llama.cpp support PR](#); ours: [open weights you can't run](#))
- **Nemotron Bolt is built for 24 GB.** NVIDIA's 2-bit, 74B mixture-of-experts model ships as one 22 GB file, sized to fit a used 3090. NVIDIA hasn't shown it on one, so that's my next test. ([NVIDIA](#))

- **llama.cpp v0.7.0 caches hot experts in VRAM.** The new `--moe-cache-mib` flag took a 4090 from 25 to 41 tok/s in its pull request. If you offload experts to RAM, try it, after the flag changes above. ([release notes](#); [the pull request](#))
 - **Mistral's Timestral forecasts on a CPU.** Its time-series models run from 5M to 209M parameters and answer in 10–140 ms with no GPU. That's good news for sensor and machine monitoring, unless your company makes over \$20M a month. ([Mistral](#); [licence](#))
 - **Next-Concept Modeling writes the whole answer at once.** A 25M-parameter Stanford model solved 86% of hard Sudokus, where a token-by-token model solved 1%. It's research only, but the code and weights are MIT-licensed. ([paper](#))
 - **The RTX 5090 halt is still unconfirmed.** Micro Center lists it as "Not Carried", and used cards sell for about \$6,000. If 24 GB will do, a \$1,500 used 3090 has the same VRAM as a \$3,000 4090. ([Tom's Hardware](#); ours: [RTX 5090 for local AI](#))
-

The Faster Chip Is the Cheaper One

Last week's issue ended on a promise: the same 28.9M-parameter story model on the ESP32-P4, and what it costs in energy. Here it is. If you want a small model running on a battery, tokens per second is the wrong number to stare at. The number that matters is energy per token, and on a microcontroller the board around the chip can burn as much as the work itself.

So I built a logger for it: an INA226 and a Raspberry Pi Pico on the USB line, sampling about 826 times a second, calibrated against a multimeter. I measured every joule over five minutes of back-to-back generation, three runs per board. A cheap inline USB meter read 0.00 A on all three boards, because they draw too little for it to register.

The ESP32-P4 board used 7.909 mJ per token. The in-spec S3 build used 14.709 mJ, and the S3 with experimental 120 MHz memory clocks used 12.357 mJ. The P4 draws the most power of the three while generating, about 0.54 W, but it finishes each token in 13.6 ms against the S3's 33.4 ms, and that time saving outweighs the extra draw. The two S3 builds tell the same story on one chip: the faster memory clock draws more power on every run and still costs less per token.

The other finding is the idle tax. Sitting at its prompt doing nothing, each board draws about half of what it draws while generating. On the P4 board that includes an Ethernet port nothing here uses. Pick the chip for speed, then look hard at the board.

I locked five predictions with a hash before the runs. Four held. The fifth, that the P4 board would idle at least 0.2 W above the S3, failed at about 0.09 W, and I already had that result when I locked it. The article explains why I kept it as written.

 **Full breakdown [here](#).**  **The S3 speed work it builds on: <https://insiderllm.com/guides/llm-on-microcontroller-esp32-s3/>**

 **Every number we've measured, with configs and raw logs: insiderllm.com/benchmarks/.**

That's the week. If you're putting a model on a battery, measure energy per token, and pick for speed.

New here, reading this on the web? [Subscribe](#) and the next one lands in your inbox.

— Mark, InsiderLLM

Have you measured power on a board running a model? What did it idle at? Reply, or hit me at hello@insiderllm.com. I read everything.

A weekly email with every new guide and measured benchmark.

[Subscribe — free, no spam](#)

Source: <https://insiderllm.com/blog/newsletter-2026-10-19/>

Free guides for running AI locally