Raspberry Pi Crypto-Mining Setup

Manas Jha

June 2023

Overview

I'm working on this personal project to figure out what actually happens behind the scenes when a blockchain gets mined. The idea isn't to make money — it's to see how proof-of-work really functions and how far I can push a Raspberry Pi before it taps out. Running a Monero (XMR) miner on a Pi 4 feels like the perfect way to learn how distributed computing works in the real world.

This project mixes curiosity with a bit of challenge — building, tuning, and experimenting with how a low-power board handles heavy computation while staying cool and efficient.

Project Idea

The main goal is to learn how mining software connects to pools, how proof-of-work validation actually looks from a systems perspective, and what happens when limited hardware gets pushed to its limits. I also want to automate basic monitoring so I can track how performance changes over time and how temperature and thread count interact during long mining sessions.

Setup

Hardware: Raspberry Pi 4
Software: Raspberry Pi OS Lite (64-bit), xmrig 6.x, Python 3.11, Bash.

Implementation

- I compile xmrig from source to make it optimized for the Pi's ARM architecture.
- I configure the mining pool credentials and CPU thread limits inside a custom `config.json`.
- I build a simple Bash script that auto-starts mining and logs system stats in real time.
- I write a Python script that tracks hash rate, CPU temperature, and uptime while the miner runs.
- I experiment with different thread counts and fan speeds to see how they affect temperature and hash rate.

During testing, I spend a lot of time tweaking configurations and watching how even tiny changes affect performance. Every time I hit thermal throttling, I adjust the fan curve or reduce threads. It's basically an ongoing feedback loop — code, test, measure, repeat — and it helps me see how resource management actually works under pressure.

Results and Observations

Mining on a Pi is slow, but it's surprisingly insightful. Once everything stabilizes, I can see how the system balances CPU load and heat. Watching hash rates fluctuate while monitoring temperatures gives a clear idea of how proof-of-work translates into actual energy and computation.

Parameter	**Observation**
Hash Rate	˜44 H/s (4-core load)
CPU Temperature	62–68 °C
Efficiency	˜10.3 H/s per core

What I Learn

- I see how miners communicate with pools using the Stratum protocol.
- I understand how CPU throttling and thermal design affect long-term stability.
- I learn how to automate data collection and visualize performance trends.
- I realize that small devices follow the same optimization logic as large-scale mining rigs — just at a different scale.

Takeaways

This project shows me that mining is less about profit and more about understanding distributed systems. Even on limited hardware, it teaches me how performance scales, how resource bottlenecks appear, and how optimization can stretch the limits of what small systems can do. The process feels like a real-world lesson in system tuning and efficiency.

Final Thoughts

I'm not chasing coins here — I'm chasing understanding. Running a miner on a Raspberry Pi shows me exactly how computational proof-of-work behaves in practice. It's a cool experiment that ties together everything I've learned about performance, automation, and system design. Even though it's not profitable, it's definitely valuable in terms of experience — and it gives me a deeper appreciation for how this tech drives modern cryptography and decentralized computing.