AtomVM

About me (Davide Bettio)

https://github.com/bettio/ | davide@uninstall.it | https://uninstall.it/

- Tinker with hardware and embedded systems since 2004.
- **Long-time open-source dev** (since ~2005 contributed to KDE Plasma and others).
- **Fell in love with Elixir in 2017** & started the AtomVM the same year
- I love hiking!

AtomVM

The C/C++ Experience on MCUs (microcontrollers)

- Concurrency? *Manual, tricky.*
- Binary parsing? *Boring & dangerous.*
- Async? *Callback hell, anyone?*
- Memory?

AtomVM

AtomVM: **Elixir, Erlang, and Gleam** on Microcontrollers

AtomVM

Elixir, Erlang and Gleam

- **Functional languages** running on the **BEAM virtual machine** (the reference Erlang VM)
- Designed for building **highly testable and reliable software**
- **Erlang**: the original BEAM language for highly reliable, distributed systems (OTP, supervision)
- **Elixir:** a modern, highly productive BEAM language
- **Gleam**: a statically typed BEAM language

Similar foundations, different strengths

AtomVM

Actor Model: Processes & Messages

- Spawning a process is extremely cheap: you can run millions of them (they are not OS threads)
- Shared-nothing design: no shared memory or global state; processes interact only by exchanging messages
- Asynchronous programming becomes simple and natural

[Maybe not million of processes, but definitely a killer feature for an IoT project]

AtomVM

Fault Tolerant by Design

- Erlang was created for Ericsson's telephony switches, which had to be continuously available. Goal: "nine nines" (~31 milliseconds/year of downtime)
- It embraces the idea that failures are inevitable and should be handled, not avoided
- Fault-tolerance best practices and features (such as supervisors) are built into Erlang/OTP

[Definitely a killer feature for an embedded project]

AtomVM

Pattern Matching

```
def fact(0), do: 1                    # match the base case
def fact(n), do: n * fact(n - 1)      # match any other n
```

```
case reply do
  %{status: 200, body: "{}"} -> :ignore
  %{status: 200, body: body} -> parse(body)
  %{location: location} -> handle_redirect(location)
  %{} = payload -> handle_error(payload)
end
```

AtomVM

Pattern Matching on Binaries

Pattern matching on binaries is one of the Erlang VM strengths.

You can match any single bit out of a payload you received somehow. e.g. from a LoRa antenna, from a sensor, etc…:

```
def parse(
    <<dest::little-unsigned-32, src::little-unsigned-32, pkt_id::little-unsigned-32,
      hop_start::size(3), via_mqtt::size(1), want_ack::size(1),
      hop_limit::size(3), channel_hash::8, 0::16, encrypted_data::binary>>) when src != 0 do

  Logger.debug “Parsing packet #{pkt_id} from #{src}”

end
```

[Definitely a killer feature for an IoT project]

AtomVM

AtomVM: **Elixir, Erlang, and Gleam** on **Microcontrollers**

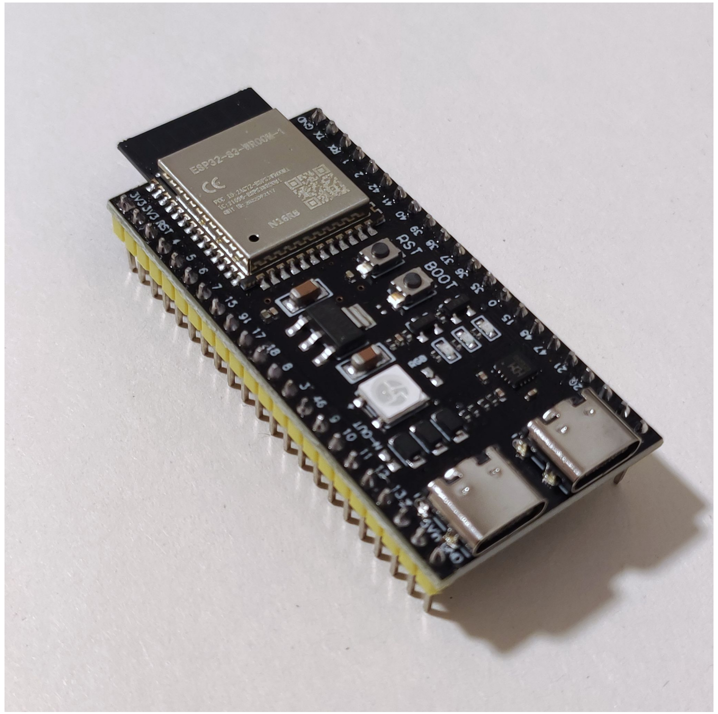
AtomVM

Modern MCU: ESP32 Example

ESP32:

- Cost < 5 €
- Dual Core @ 240MHz
- RAM: ~500KB - 8MB
- Flash: 4MB - 16MB
- Connectivity: WiFi, Bluetooth, etc.
- Lot of GPIOs & integrated peripherals
- Low Power / Battery-friendly

Powerful, but **not enough for running the BEAM**, the reference Erlang VM.



AtomVM

What if we could bring *somehow* the safety, **concurrency, and productivity** of the BEAM ecosystem to these **tiny devices?**

AtomVM

To the Rescue

AtomVM, A lightweight virtual machine designed to run compiled Erlang, Elixir and Gleam code on microcontrollers with limited resources.

- Key Trade-offs:
  - **Memory First**: RAM & Flash are precious
  - **Portability**: New targets in hours, not days
  - **Flexible Requirements**: Adaptable core

AtomVM

AtomVM

- **Already available today:** https://github.com/atomvm/AtomVM
- Runs on *nix, ESP32, RaspberryPi Pico, STM32 (more are coming)
- **Production-ready** (and ready for your next project)
- Capable of running complex applications
- Advanced features such as JIT and Erlang Distribution
  - Yes, you can cluster MCUs together: and even mix MCUs with traditional BEAM nodes

AtomVM

What Can I Use AtomVM For?

- I used AtomVM for some projects like home automation and LoRa (radio) nodes
- Mostly any kind of embedded, IoT, and automation projects
- A solid alternative to Arduino, MicroPython, and similar stacks
- In general when you need embedding an Erlang VM in new or unconventional environments
    - They did it with emscripten and the browser, see also: https://popcorn.swmansion.com/

AtomVM

Join Us

https://atomvm.org/

Discord: https://discord.gg/QA7fNjm9Nw

Telegram: https://t.me/atomvm

Sponsor: https://github.com/sponsors/atomvm

AtomVM