INTRODUCTION TO NERVES

Elixir for embedded systems

Christian Sarnataro

Elixir Language Milano - May 20th, 2025

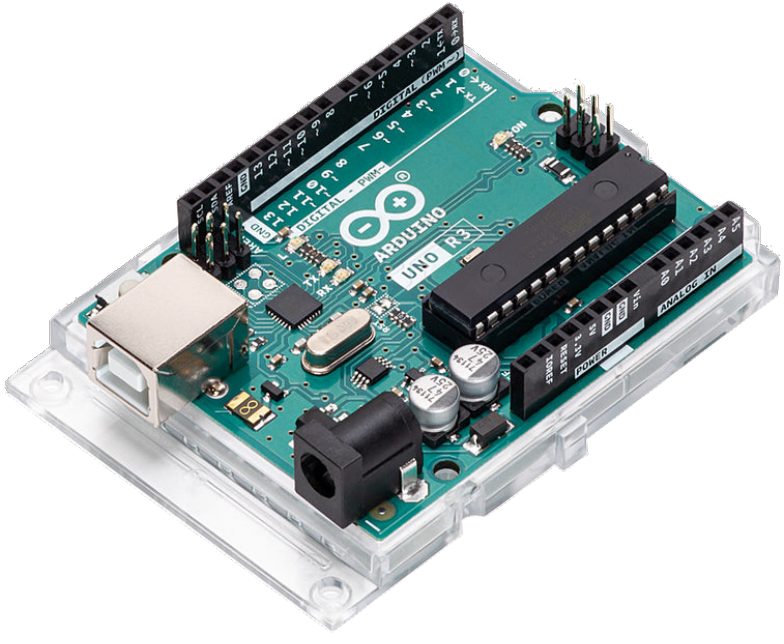
WHO AM I

Christian Sarnataro

- Frontend software engineer at Arduino since 2021
  - Arduino is an open-source electronics platform based on easy to use hardware and software
- Previously, full stack and mobile software engineer, web architect
- LinkedIn: https://www.linkedin.com/in/sarnataro

ARDUINO UNO

- More than 10_000_000 units sold



ARDUINO IDE

- Free, open source, cross platform

DISCLAIMER

(italian translation: "metto le mani avanti")

- Not a BEAM/OTP/Elixir expert (but interested in learning more)
- Not an electronics expert (but interested in "physical computing")

AGENDA

1. What are embedded systems?
2. Introduction to Nerves
3. Demos:
    i. Blink, the *Hello World* of embedded system (here, ++)
    ii. Phoenix/Liveview app with Nerves
    iii. Temperature measurement and Livebook
    iv. (Optional) Pomodoro Timer
4. Q/A

WHAT ARE ERLANG, BEAM, OTP, ELIXIR, ANYWAY?

- Erlang (1986)
  - Functional programming language (ispirato by Prolog) [1]
  - Used to build concurrent, distributed, fault-tolerant applications
- BEAM - Bogdan's Erlang Abstract Machine (1993)
  - A virtual machinge for Erlang

> *[1] A History of Erlang by Joe Armstrong*

WHAT ARE ERLANG, BEAM, OTP, ELIXIR, ANYWAY?

- OTP - Open Telecom Platform (1996)
  - Erlang, libraries, runtime, docs, patterns
  - according to other sources, Outlaw Techno Psychobitch [2]
- Elixir (2012)
  - modern version of Erlang (inspired by Ruby)

> *[2] Erlang The Movie II: The Sequel minuto 3:03*

ANALOGIES WITH JAVA

(My own interpretation)

Erlang	=>	Java
BEAM	=>	JVM
OTP	=>	JDK/JEE
Elixir	=>	Kotlin

NERVES AND EMBEDDED SYSTEMS

- **Nerves** is an open-source platform that combines the rock-solid BEAM virtual machine and Elixir ecosystem to easily build and deploy production embedded systems *(from nerves-project.org)*.
- Traditionally embeddes systems are programmed in low-level languages like: *Bare metal*, Assembly, C/C++ and more recently Rust/Zig

WHAT ARE EMBEDDED SYSTEMS

- Different and diverse definitions of embedded devices
- The one I like the most:

> *Embedded systems are single purpose computers*

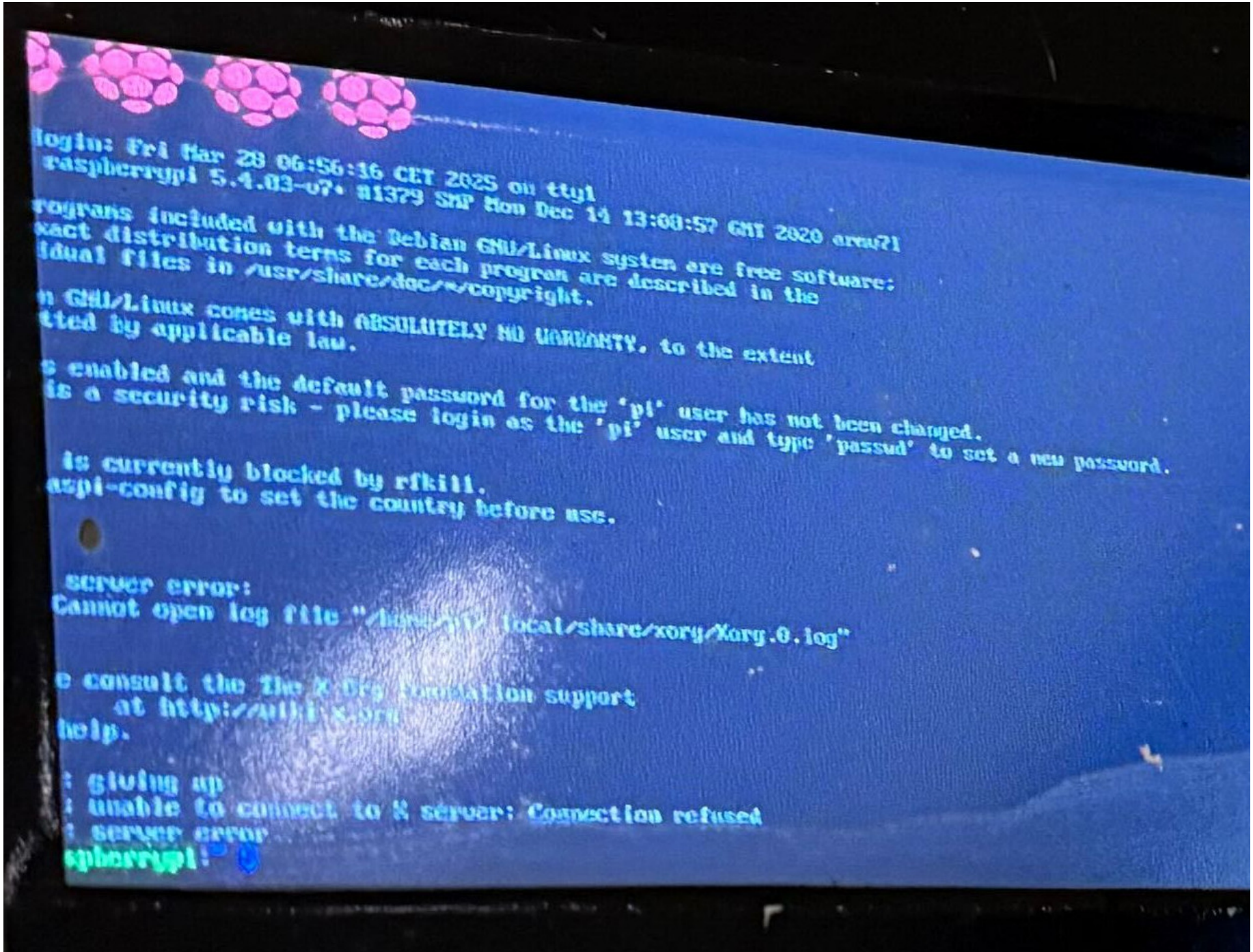
- A computer which solves a problem **in the real world** (physical interactions, high reliability)

APPLICATIONS

- Home and industrial automation, automotive, robots, drones, IoT
- Smart consumer electronics
- Interactive art, exhibitions
- Usually resource constrained:
  - cost
  - power consumption
  - small size

A BROKEN EXAMPLE

- This project **IS NOT** using Nerves, but it would a perfect use case


login: Fri Mar 28 06:56:16 CET 2025 on tty1
raspberrypi 5.4.83-v7+ #1379 SMP Mon Dec 14 13:08:57 GMT 2020 armv7l
rograms included with the Debian GNU/Linux system are free software:
xact distribution terms for each program are described in the
idual files in /usr/share/doc/*/copyright.
n GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
tted by applicable law.
s enabled and the default password for the 'pi' user has not been changed.
is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.
is currently blocked by rfkill.
aspi-config to set the country before use.
server error:
Cannot open log file "/home/pi/.local/share/xorg/Xorg.0.log"
e consult the The X.Org Foundation support
at http://wiki.x.org
help.
: giving up
: unable to connect to X server: Connection refused
: server error
spberrypi:~ $

FLAVOURS

Microprocessors	**Microcontrollers**
Faster (4 GHz)	Slower (200 MHz) *(but react in real time)*
GB of RAM	KB of RAM
With OS (Linux)	Bare metal or RTOS
E.g. Raspberry PI	E.g. Arduino, ESP32
Arduino Portenta X8?	*AtomVM*
Nerves ✅	**Nerves** ❌

> *Inspired by: Frank Hunleth (core maintainer of Nerves and author of "Build a Weather Station with Elixir and Nerves")*
> *youtu.be/P_xryjmG35I, minute 11:19*

WHY NERVES

- Nerves is used in production in many project (in industrial or agricultural automation)
- Focus on **networking, concurrency and reliability**

Luckily, I don't need to explain the strengths of the BEAM in an Elixir meetup (*)

(*) Shamelessly stealed from Giacomo Cavalieri

GETTING STARTED WITH NERVES

- Some prerequisites
  - Raspberry PI (or similar)
  - Micro SD Card
  - Some hardware (LEDs, sensors, wires, displays) and, possibly, some iron soldering
  - Get familiar with embedded systems jargon and datasheets
- Experiment with your hardware and **Livebook**

"HELLO WORLD" WITH NERVES

- Assuming Elixir and `mix` are already installed:

```
$ mix archive.install hex nerves_bootstrap #nerves generators
$ mix nerves.new hello_nerves #creates new project
$ cd hello_nerves
$ export MIX_TARGET=rpi0 # VERY IMPORTANT
$ mix deps.get
$ mix firmware # builds the firmware for MIX_TARGET
$ mix burn # burns your firmware on an SD card
```

- Insert SD card it into your device (e.g. our Raspberry Pi Zero) and boot it up.

```
1 $ ssh nerves.local
2 $ HelloNerves.hello
3 :world
```

PAIN POINTS

- Manually plugging the SD card, `mix firmware` and `mix firmware.burn` take some time
  - you can use an `upload.sh` script to update the firmware without removing the SD card
  - **NervesHub** is a service for over-the-air (OTA) updates
    Warning: never tried, but looks promising
- When you create a new firmware with `firmware.burn`, the ssh key changes and you have to refresh it with `ssh-keygen -R nerves.local`

DEMO 1: BLINK A LED

(a.k.a. *Hello World* of embedded device)

- "Blink++", uses an LED to encode a message in Morse code (with sound)
- `Circuits.GPIO` for controlling GPIO (***G****eneral* ***P****urpose* ***I****nput* ***O****utput*)

  https://hexdocs.pm/circuits_gpio
- `Pigpiox` for controlling PWM (***P****ulse* ***W****idth* ***M****odulation*)

  https://hexdocs.pm/pigpiox
- **Show me the code!** Linux / Mac / GitHub

DEMO 2: PHOENIX/LIVEVIEW APP

- If you're seeing this slides, *this* is demo 2
  - Phoenix is a web framework for Elixir
- Liveview app running on Raspberry Pi 4
- Let's update our slides

DEMO 3: LIVEBOOK WITH NERVES

WHAT IS LIVEBOOK?

- Web application to write and execute interactive notebooks in the browser
- Similar to Python Jupyter Notebooks
- Supports Markdown, Elixir cells, interactive charts, integrated with Hugging Face

DEMO 3: LIVEBOOK WITH NERVES

- Specific Livebook distro for Nerves
- Ideal for educators: it can run Elixir code in the browser
- Ideal for quick iterative prototyping with hardware
- Livebook

DEMO 4. POMODORO TIMER

- `Oled` + `Chisel` for font rendering
  - https://hexdocs.pm/oled
  - https://hexdocs.pm/chisel
- `Circuits.GPIO` for the button
  - https://hexdocs.pm/circuits_gpio
- Implements a finite state machine with `:gen_statem`
- Code: Linux / Mac

RECAP

- Nerves allows development of rather complex, connected and reliable embedded applications with Elixir
- Livebook can be used to easily prototype with hardware
- Electronics is nowadays more accessible than ever, even for engineers with a mostly software background
- Combining sensors, motors, lights, buttons, cameras - and maybe a sprinkle of AI - there are no limits in what you can build

QUESTIONS?

End of presentation