

INTRODUZIONE A NERVES



Elixir per sistemi embedded

Christian Sarnataro

Elixir Language Milano - 20 Maggio 2025

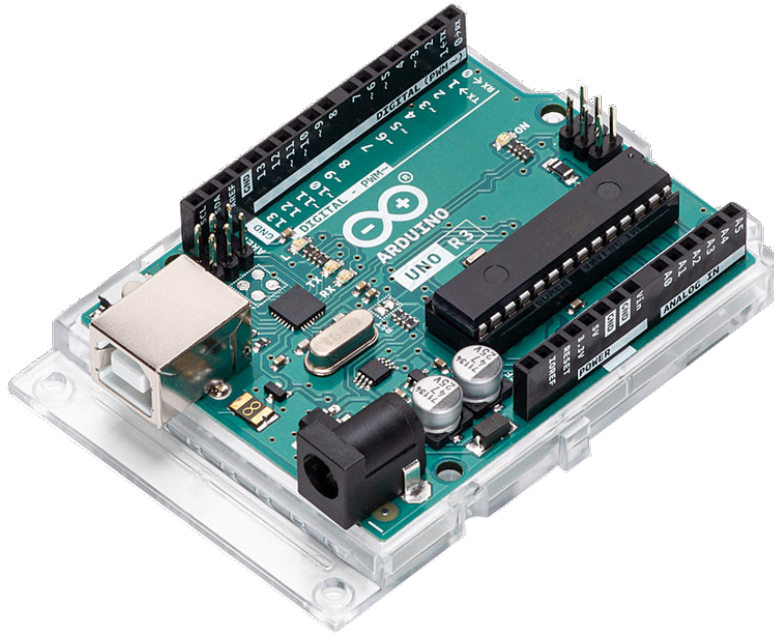
CHI SONO

Christian Sarnataro

- Sviluppatore frontend in Arduino dal 2021
 - Arduino è una piattaforma hardware e software open source progettata per un accesso semplificato all'elettronica
- In precedenza, programmatore full-stack e mobile, web architect
- LinkedIn: <https://www.linkedin.com/in/sarnataro>

ARDUINO UNO

- Più di 10_000_000 schede vendute



ARDUINO IDE

- Gratuita, open source, multi piattaforma



DISCLAIMER

(traduz.: "metto le mani avanti")

- Non sono un esperto di BEAM/OTP/Elixir (ma mi interessa saperne di più)
- Non sono un esperto di Elettronica (ma mi interessa il "physical computing")

AGENDA

1. Cosa sono i sistemi embedded?
2. Introduzione a Nerves
3. Demo:
 - i. Blink, l'*Hello World* dei sistemi embedded (ma ++)
 - ii. App Phoenix/Liveview con Nerves
 - iii. Misurazione di temperatura con Livebook
 - iv. (Opzionale) Pomodoro Timer
4. Q/A

ERLANG, BEAM, OTP, ELIXIR?

- Erlang (1986)
 - Linguaggio funzionale (ispirato a Prolog) ^[1]
 - Applicazioni concorrenti, distribuite, fault-tolerant
- BEAM - Bogdan's Erlang Abstract Machine (1993)
 - Una macchina virtuale per Erlang

[1] [A History of Erlang](#) by Joe Armstrong

ERLANG, BEAM, OTP, ELIXIR?

- OTP - Open Telecom Platform (1996)
 - Erlang, librerie, runtime, documentazione, pattern
 - secondo altre fonti, Outlaw Techno Psychobitch ^[2]
- Elixir (2012)
 - versione moderna di Erlang (ispirato a Ruby)

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

ANALOGIE CON JAVA

(Mia personale interpretazione)

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

NERVES E SISTEMI EMBEDDED

- **Nerves** è una piattaforma open-source che unisce la solidità della virtual machine BEAM con l'ecosistema Elixir per costruire e mettere in produzione sistemi embedded. (*tradotto da nerves-project.org*).
- Normalmente i sistemi embedded sono programmati con linguaggi a basso livello, ad es.: *Bare metal*, Assembly, C/C++ e più recentemente Rust/Zig

COSA SONO I SISTEMI EMBEDDED?

- Molteplici definizioni di sistemi embedded
- Quella che preferisco:

I sistemi embedded sono computer pensati per un singolo scopo specifico (single purpose computer)

- Un calcolatore che risolve un singolo problema **nel mondo reale** (interazioni fisiche e alta affidabilità)

APPLICAZIONI

- Domotica, automazione industriale, automotive, robots, droni, IoT
- Elettronica di consumo "smart"
- Installazioni artistiche interattive
- Vincoli in termini di:
 - costo
 - consumo di elettricità
 - dimensioni ridotte

UN ESEMPIO (ROTTTO)

- Questo dispositivo embedded **NON** usa Nerves, ma sarebbe un ottimo caso d'uso



login: Fri Mar 28 06:56:16 CET 2025 on tty1
raspberrypi 5.4.03-v7+ #1379 SMP Mon Dec 14 13:00:57 GMT 2020 armv7l

Programs included with the Debian GNU/Linux system are free software;
exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

apt-get is currently blocked by rfkill.
Run apt-get config to set the country before use.

server error:
Cannot open log file "/home/pi/.local/share/xorg/Xorg.0.log"

Please consult the The X.Org Foundation support
at <http://wiki.x.org>
for help.

: giving up
: unable to connect to X server: Connection refused
: server error

raspberrypi:~\$

TIPOLOGIE

Microprocessori

Più veloci (4 GHz)

GB di RAM

Con OS (Linux)

Per es: Raspberry PI

Arduino X8?

Nerves 

Microcontrollori

Più lenti (200 MHz) *(ma in real time)*

KB di RAM

Bare metal o RTOS

Per es: Arduino, ESP32

AtomVM

Nerves 

Adattato da: Frank Hunleth (core maintainer di Nerves e autore di "Build a Weather Station with Elixir and Nerves")
youtu.be/P_xryjmG35I minuto 11:19

PERCHÉ NERVES

- Nerves è testato e utilizzato in produzione in svariati progetti (per es. automazione industriale o agricoltura)
 - Orientato a **networking, concorrenza e affidabilità**
-

Per fortuna, non devo spiegare i punti di forza della BEAM in un meetup Elixir ()*

() Rubata a Giacomo Cavalieri*

INTRODUZIONE A NERVES

- Alcuni pre-requisiti
 - Raspberry PI (o simili)
 - Scheda Micro SD
 - Hardware (LED, motori, sensori, cavetti, display) e, eventualmente, saldatore a stagno
 - Iniziare a familiarizzare con i datasheet e il gergo utilizzato nell'ambito dei sistemi embedded
- Iniziare a sperimentare con hardware e **Livebook**

"HELLO WORLD" CON NERVES

- Con Elixir e mix già installati:

```
$ 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
```

- Inserire la SD card nel device (nel nostro caso Raspberry Pi Zero)

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

PAIN POINTS

- Mettere/togliere la scheda SD, `mix firmware` e `mix firmware.burn` richiedono un sacco di tempo
 - esiste `script upload.sh` per aggiornare il firmware senza rimuovere la scheda SD dalle volte successive
 - esiste **NervesHub**, un servizio per aggiornamenti over-the-air (OTA).
Nota: non testato, ma molto interessante
- Quando si crea un nuovo firmware con `firmware.burn` la chiave ssh cambia e va rigenerata con `ssh-keygen -R nerves.local`

DEMO 1: ACCENDERE UN LED

(a.k.a. "Blink", a.k.a. l'*Hello World* dei sistemi embedded)

- "Blink++": usa un LED per codificare un messaggio in codice Morse (con effetti sonori)
- `Circuits.GPIO` per gestire GPIO (***G***eneral ***P***urpose ***I***nput ***O***utput)

https://hexdocs.pm/circuits_gpio

- `Pigpiox` per gestire PWM (***P***ulse ***W***idth ***M***odulation)

<https://hexdocs.pm/pigpiox>

- **Show me the code!** [Linux](#) / [Mac](#) / [GitHub](#)

DEMO 2: APP PHOENIX/LIVEVIEW

- Se vedete queste slide, *questa* è la demo 2
 - Phoenix è un framework web per Elixir
- App Phoenix in esecuzione su Raspberry Pi 4
- Aggiorniamo le slide

DEMO 3: LIVEBOOK CON NERVES

COS'È LIVEBOOK?

- Applicazione web per eseguire *notebook* interattivi con Elixir direttamente nel browser
- Simile a Python Jupyter Notebooks
- Supporta Markdown, "celle" Elixir, grafici interattivi, integrazione con Hugging Face

DEMO 3: LIVEBOOK CON NERVES

- Distribuzione di Livebook specifica per Nerves
- Ottimo per docenti/formatori: Elixir in azione nel browser
- Ottimo per sperimentare con l'hardware in maniera iterativa
- [Livebook](#)

DEMO 4. POMODORO TIMER

- `Oled` + `Chisel` per gestione schermo e font
 - <https://hexdocs.pm/oled>
 - <https://hexdocs.pm/chisel>
- `Circuits.GPIO` per gestione pulsante
 - https://hexdocs.pm/circuits_gpio
- Implementa una macchina a stati finiti con `:gen_statem`
- Codice: [Linux](#) / [Mac](#)

CONCLUSIONI

- Nerves permette lo sviluppo di applicazioni embedded complesse, connesse e affidabili con Elixir
- Livebook semplifica la prototipazione con l'hardware
- L'elettronica è oggi molto accessibile, anche per chi ha un background prevalentemente software
- Le possibilità, combinando "a piacere" sensori, motori, luci, pulsanti, telecamere - e magari una spruzzatina di AI - sono praticamente infinite

RIFERIMENTI

- Demo e slide https://github.com/csarnataro/nerves_talk
- Nerves project <https://nerves-project.org/>
- Elixir <https://elixir-lang.org/>
- Erlang: The Movie <https://www.youtube.com/watch?v=xrljfljssLE>

DOMANDE?

Fine della presentazione