



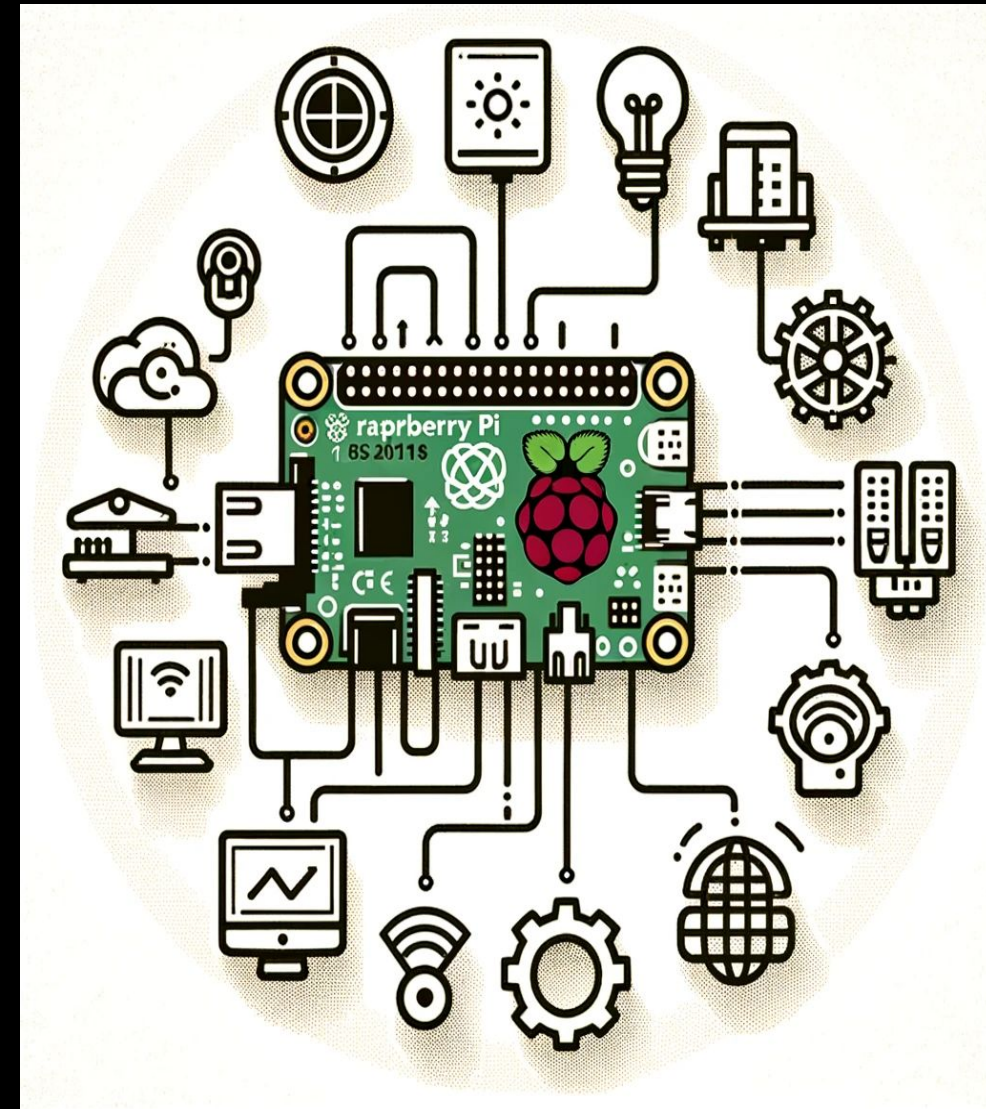
Pi4Miconaut

- Ruthvik Mannem



IOT & Raspberry Pi

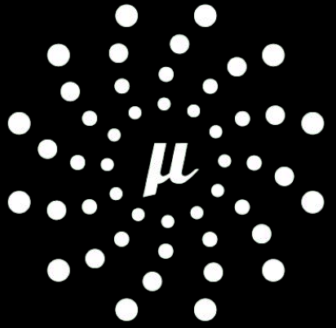
- **Internet of Things:**
 - Network of devices communicating and exchanging data.
 - Automation, remote control, and data analysis.
 - Advancements in smart homes, cities, and industries.
- **Raspberry Pi:**
 - Compact, affordable, powerful computer.
 - Ideal for educational, hobbyist, and industrial projects.
 - Features GPIO pins, Wi-Fi/Bluetooth connectivity, and low power usage.
- **Raspberry Pi in IoT:**
 - Acts as a central controller for IoT devices.
 - Supports sensors and actuators for data collection



What is the problem?

- Python dependency when it comes to Raspberry Pi.
- It's hard to Interact with hardware components using traditional java approach.
- Understanding the anatomy of components.
- Knowledge required on the communication protocols like Digital Output, PWM, I2C etc.

Micronaut & Pi4J



M I C R O N A U T[®]

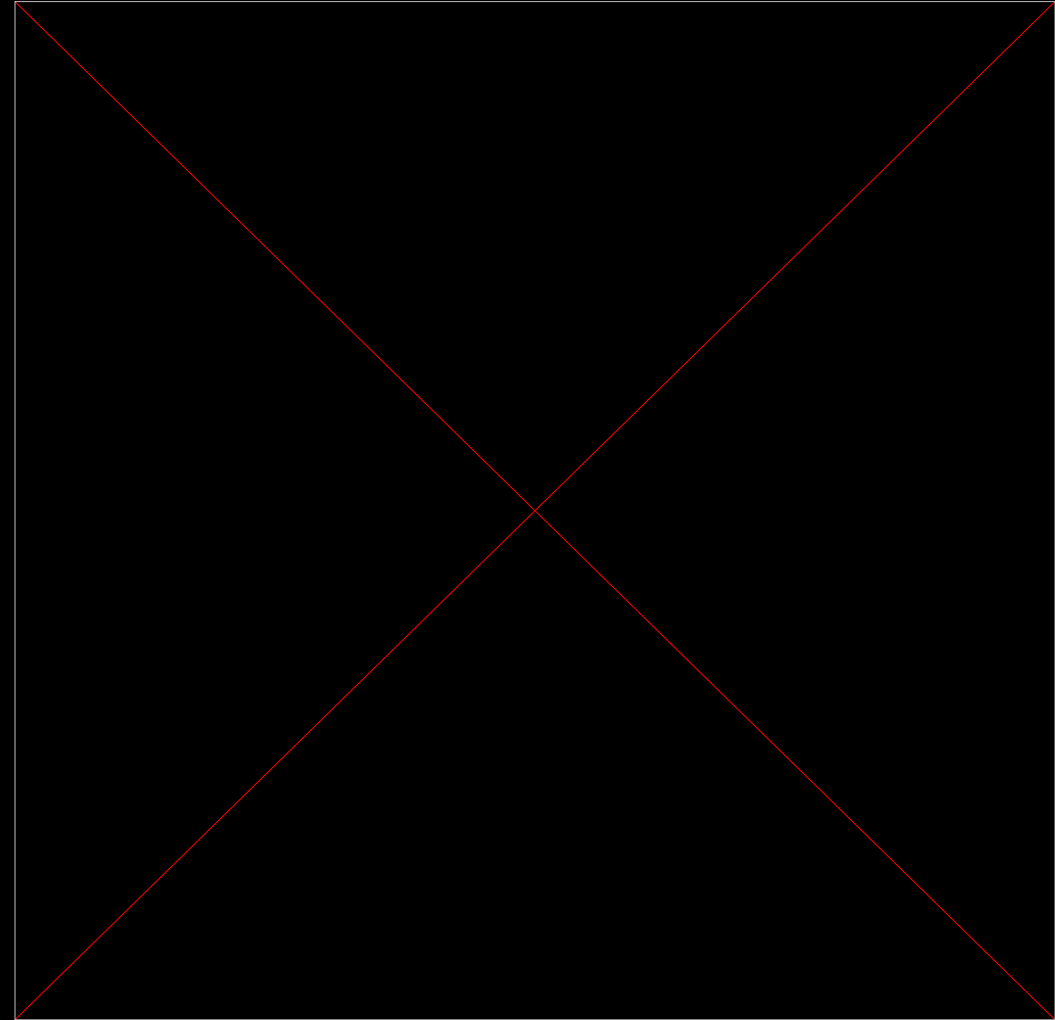
- Java based framework for building microservices.
- Lightweight
- Fast startup time
- Low memory footprint.
- Simplifies building scalable applications.



- Open-source library for Java programmers.
- Easy access to the GPIO and other hardware interfaces on the Raspberry Pi.
- Control and communicate with external sensors, motors, and other peripherals.

Pi4Micronaut

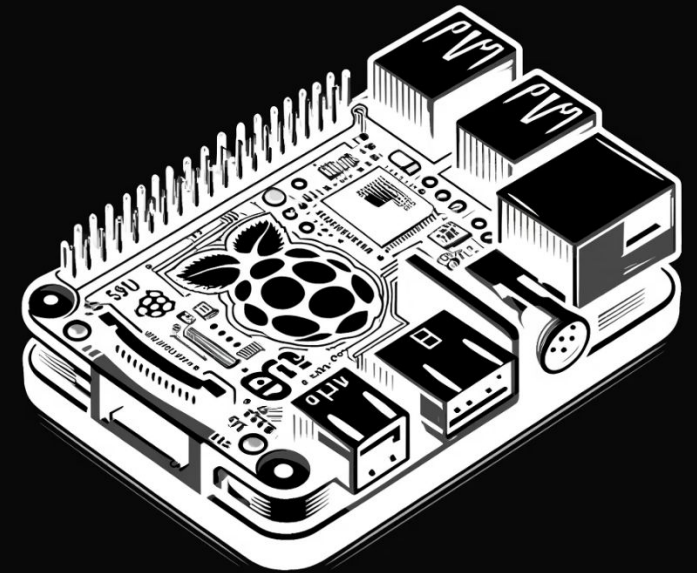
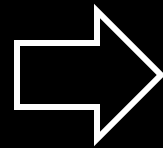
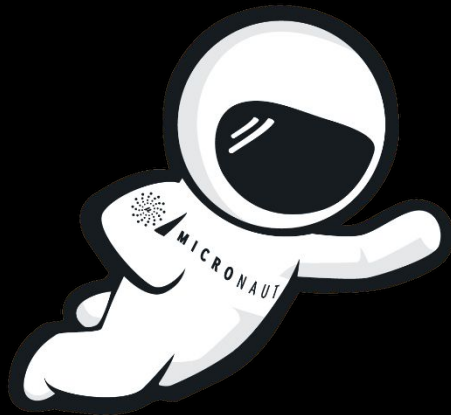
- Open Source Java library.
- Enables interaction with the hardware components connected to Raspberry pi via GPIO.
- A fusion of **Micronaut framework** and the **Pi4J** library.
- Develop and deploy the hardware interfaces as microservices on Raspberry Pi.
- Pi4Micronaut's availability in Maven Central.
- One stop shop for documentation, example implementations for hardware interactions and circuit setup.



How we made it?



 **Pi4Micronaut**



Why to use Pi4Micronaut?

- Provides a powerful platform for IoT developers to leverage Java's ecosystem.
- Enhances the productivity of developers by providing a cohesive development environment.
- Makes it easier to manage and deploy IoT services on Raspberry Pi devices.

Where it can be used?

- Home automation systems.
- Industrial monitoring and control
- Smart city infrastructure
- Educational projects and research

Example Projects

- Home Security
- GitHub Merger

Supported Components

- Currently Pi4Micronaut v1.0 support 13 components.
 - Push Button
 - Slide Switch
 - Rotary Encoder
 - RFID Scanner
 - LED
 - RGB LED
 - LCD1602
 - Photosensor
 - Touch Switch Sensor
 - Active Buzzer
 - Passive Buzzer
 - PIR Motion Sensor
 - Ultrasonic Sensor

Developing Pi4Micronaut @OpenSourcewithSLU

- Collaboration of **Open Source with SLU** and **Unity Foundation**.
- When it started? **August 2022**
- Two teams with a group of 4 undergraduate developers and a tech-lead.
- Contributed 20 hours/week for 18 months.
- We have released **Pi4Micronaut v1.0** on Pi Day - March 14th 2024.

Developers

- Ruthvik Mannem (Tech Lead)
- Greih Murray (Tech Lead)
- Adrian Swindle
- Austin Howard
- Alex Delgado
- Traison Diedrich
- Joe Folen
- Sinuo Liu
- John Yanev

How to get started?

- Choose the components which you want to use.
- Refer to Pi4Micronaut Documentation.
- Setup circuit.
- Configure and download Micronaut Application.
- Add Pi4Micronaut v1.0 in your dependencies.
- Add the configuration of your setup in **Application.yml** file.
- Create a controller class to make use of the Pi4Micronaut methods for your components.
- Build your application.
- Copy the Jar file on to Raspberry Pi.
- SSH on to Pi and Run the Jar file.
- Use Curl commands using the terminal of Pi to interact with the component.

Demo with Micronaut Application

- Link to Documentation - <https://oss-slu.github.io/Pi4Micronaut/>



Useful Links

- [Link to Repository](#)
- [Documentation](#)
- [Maven Central Repository](#) – Pi4Micronaut v1.0 library
- [Library Artifacts](#)

Thank You!!!