Bringing More People Closer To The Metal

Who am I?

👋 Hi! I'm Rahul

Who am I?

👋 Hi! I'm Rahul

- 🇲🇺 From Mauritius

Who am I?

👋 Hi! I'm Rahul

- 🇲🇺 From Mauritius
- Increasing Rust's Reach 2018

Who am I?

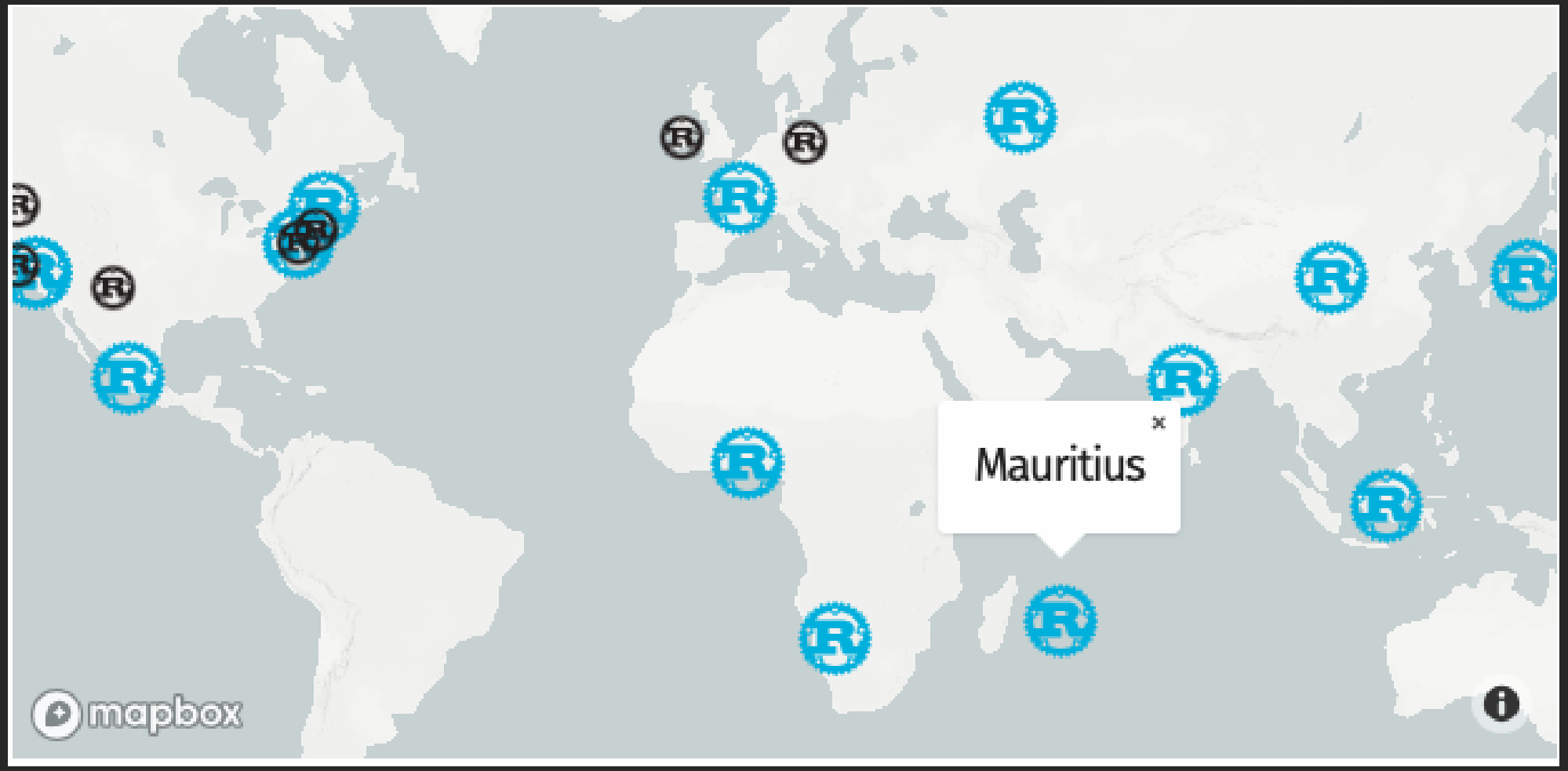
👋 Hi! I'm Rahul

- 🇲🇺 From Mauritius
- Increasing Rust's Reach 2018
- Creating **rust_gpiozero**

Who am I?

👋 Hi! I'm Rahul

- 🇲🇺 From Mauritius
- Increasing Rust's Reach 2018
- Creating **rust_gpiozero**
- Physical computing with Rust


Mauritius
mapbox

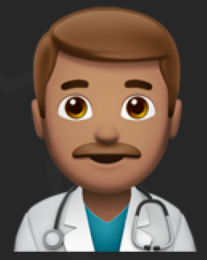
> In more than a decade of teaching physical computing at New York University's Tisch School of the Arts, we have found people from very **diverse backgrounds** looking to bridge this gap between the physical and the virtual.
> ~ *Physical Computing, Dan O'Sullivan and Tom Igoe*

The maker movement

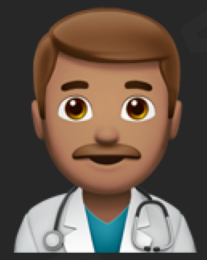
> **Maker culture** encourages novel applications of technologies, and the exploration of **intersections** between traditionally separate domains and ways of working including metal-working, calligraphy, film making, and computer programming.
> ~ *Maker Culture*, *Wikipedia*

My own journey

My own journey



My own journey



Why Should You Care?

> **Empowering everyone*** to build reliable and efficient software.
> ~ *The Rust Programming Language*

> The Rust programming language is fundamentally about **empowerment**: no matter what kind of code you are writing now, Rust empowers you to **reach farther**, to program with confidence in **a wider variety of domains** than you did before.
> ~ *The Rust Programming Language Book*

Barriers to entry

⚡ Basic Electronics & Circuits

⚡ Basic Electronics & Circuits

 Computers and Microcontrollers

⚡ Basic Electronics & Circuits

💻 Computers and Microcontrollers

👨🏻‍💻 Programming

⚡ Basic Electronics & Circuits

💻 Computers and Microcontrollers

👨🏻‍💻 Programming

📡 Communication and protocols

Why physical computing?

Computers as learning instruments

> The computer is the Proteus of machines. Its essence is its universality, its **power to simulate**. Because it can take on a **thousand forms** and can serve a **thousand functions**, it can appeal to a **thousand tastes**.
> ~ *Mindstorms*, *Seymour Papert*

> The computer is the Proteus of machines. Its essence is its universality, its **power to simulate**. Because it can take on a **thousand forms** and can serve a **thousand functions**, it can appeal to a **thousand tastes**.
> ~ *Mindstorms*, *Seymour Papert*

source: Rodrigo Mesquita, Wikimedia.org

How?

Abstraction

High Level Libraries

`rust_gpiozero` 📦

```rust
use rust_gpiozero::*;

fn main() {
    // Create a new LED attached to Pin 17
    let mut led = LED::new(17);

    // on_time = 2 secs, off_time=3 secs
    led.blink(2.0,3.0);

    // prevent program from exiting immediately
    led.wait();
}
```

Buzzer	Represents a digital buzzer component.
DigitalOutputDevice	Represents a generic output device with typical on/off behaviour. Extends behaviour with a blink() method which uses a background thread to handle toggling the device state without further interaction.
LED	Represents a light emitting diode (LED)
Motor	Represents a generic motor connected to a bi-directional motor driver circuit (i.e. an H-bridge). Attach an H-bridge motor controller to your Pi; connect a power source (e.g. a battery pack or the 5V pin) to the controller; connect the outputs of the controller board to the two terminals of the motor; connect the inputs of the controller board to two GPIO pins.
OutputDevice	Represents a generic GPIO output device.
PWMLED	Represents a light emitting diode (LED) with variable brightness. A typical configuration of such a device is to connect a GPIO pin to the anode (long leg) of the LED, and the cathode (short leg) to ground, with an optional resistor to prevent the LED from burning out.
PWMOutputDevice	Generic output device configured for software pulse-width modulation (PWM). The pulse width of the signal will be 100µs with a value range of [0,100] (where 0 is a constant low and 100 is a constant high) resulting in a frequenzy of 100 Hz.
Servo	Represents a PWM-controlled servo motor connected to a GPIO pin.


Bunnisher.com @bunnisher · Mar 10
Replying to @rahulthakoor @rustlang
Rahul, I really like your rust_gpizero. Thank you for your efforts on it. I'm new to Rust, so I can learn Rust and apply it to my @Raspberry_Pi = Very cool. You have a Vegas fan here.

Top-Down Approach

Standing on the shoulders of giants

Standing on the shoulders of giants

`rppal`

embedded-hal

PWM

SPI

I2C

Guides, Tutorials and Documentation

How to draw an owl

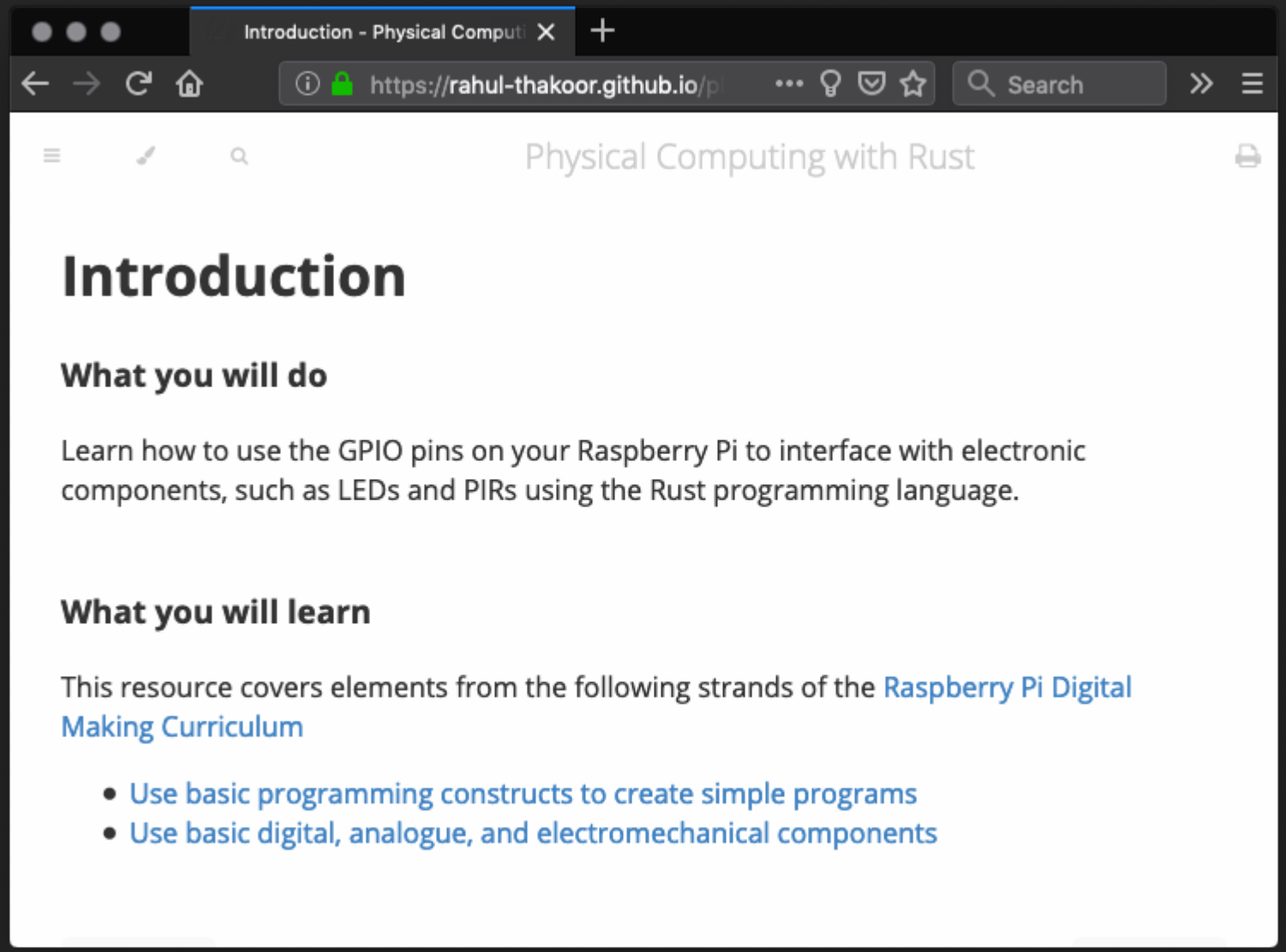
1.

2.

1. Draw some circles

2. Draw the rest of the fucking owl

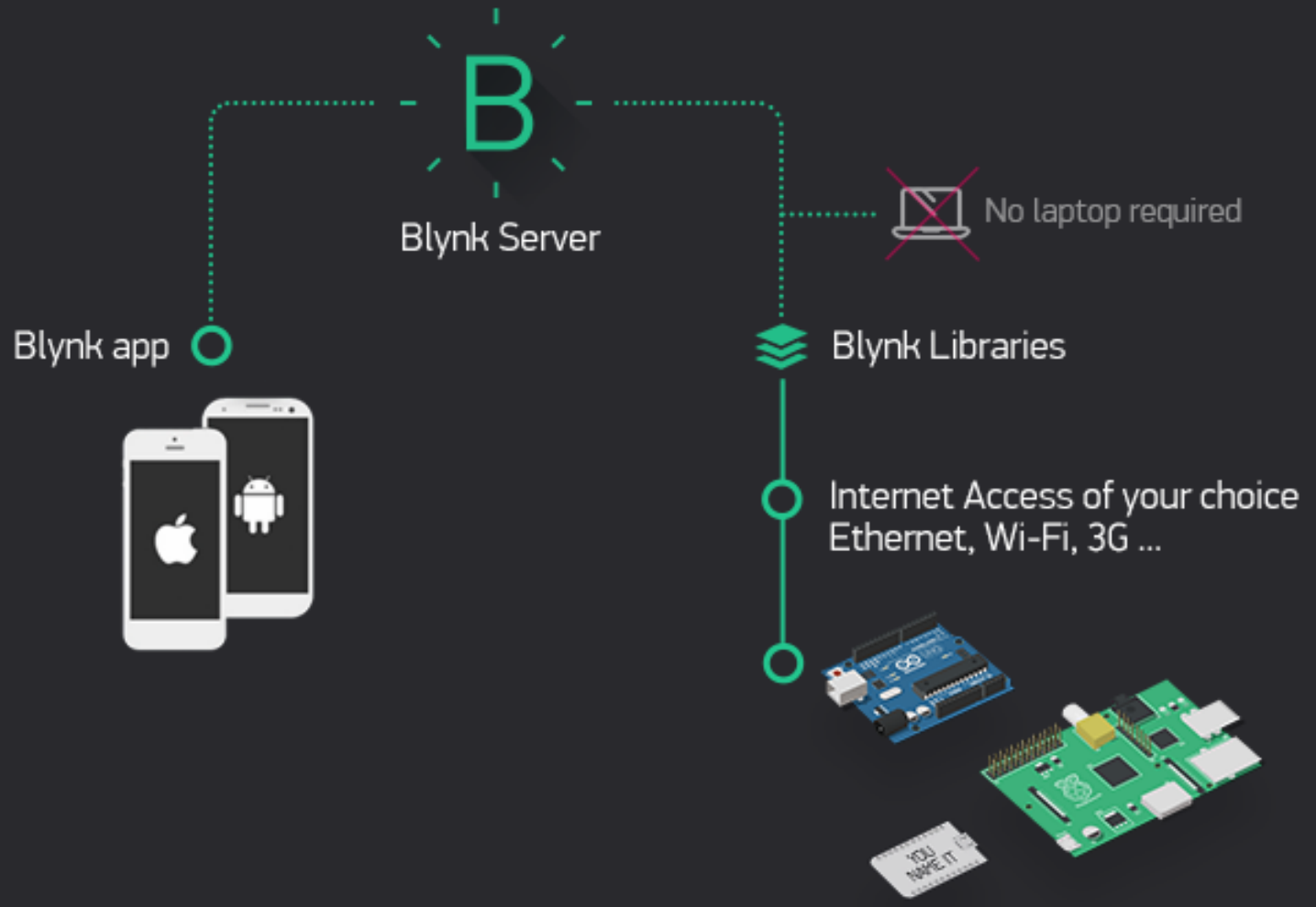
https://www.flickr.com/photos/centralasian/5229725173

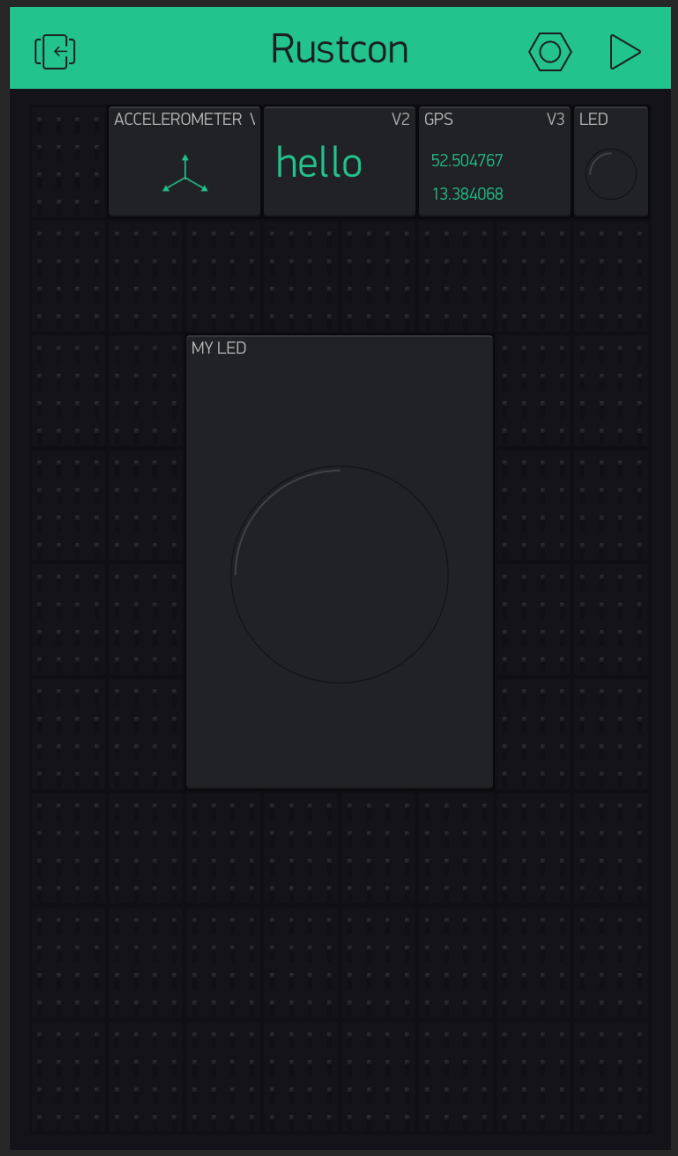

Introduction - Physical Computi
https://rahul-thakoor.github.io/p
Search
Physical Computing with Rust
Introduction
What you will do
Learn how to use the GPIO pins on your Raspberry Pi to interface with electronic components, such as LEDs and PIRs using the Rust programming language.
What you will learn
This resource covers elements from the following strands of the Raspberry Pi Digital Making Curriculum
Use basic programming constructs to create simple programs
Use basic digital, analogue, and electromechanical components

Teaching & Learning

Workshops and Bootcamps


RUSTCON ASIA
2019 BEIJING


Blynk Server
No laptop required
Blynk app
Blynk Libraries
Internet Access of your choice
Ethernet, Wi-Fi, 3G ...


Rustcon
ACCELEROMETER \
V2
hello
GPS
V3
52.504767
13.384068
LED
MY LED

Your Turn

Thanks 🙏

Questions?

Acknowledgement

- Theme inspired by rocket.rs
- Floating Cogs by Hero Patterns