

Swift on Raspberry Pi

宋旭陶 / 靛青 K

SwiftGG Swift 爱好者



收获国内外一线大厂实践 与技术大咖同行成长

- ✓ 演讲视频
- ✓ 干货整理
- ✓ 大咖采访
- ✓ 行业趋势



自我介绍

宋旭陶（靛青K），热衷于 Swift 语言及各场景下的应用。

参与 Swift 文档翻译、SwiftGG 文章翻译，
开源 TransitionTreasury、Flix Swift 框架。

对 Swift 生成代码应用上也有些许的实践，比如 RouterBuilder、Sketch2Code。

目录

1. Swift 在 Linux 上的发展现状
2. 树莓派和 Swift
3. 为什么尝试 Swift
4. Swift 实际应用 - 智能灯泡

Swift 在 Linux 上发展得怎么样了



Swift for TensorFlow



Kitura



Vapor

深度学习

Web 框架

Swift 在 Linux 上发展得怎么样了

Projects

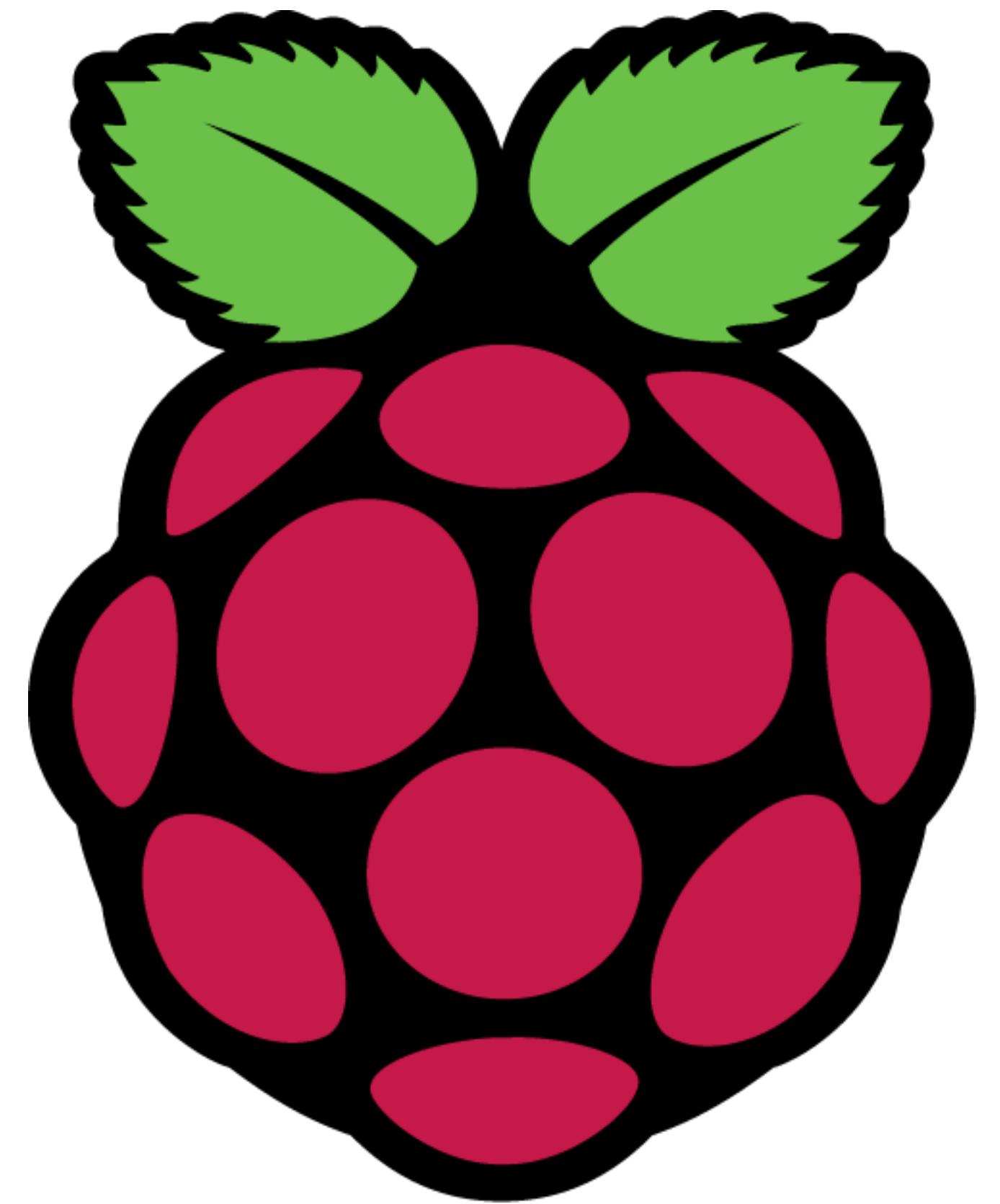
<https://swift.org/server/>

Project	Maturity Level	Pitched	Accepted
SwiftNIO	Graduated	N/A	9/7/2018
Logging API	Sandbox	9/10/2018	2/7/2019
Metrics API	Sandbox	1/8/2019	4/4/2019
Postgres Client	Sandbox	11/18/2018	5/16/2019
Redis Client	Sandbox	1/7/2019	6/27/2019
HTTP Client	Sandbox	4/18/2019	6/27/2019
APNS Client	Sandbox	2/5/2019	6/27/2019
Statsd Client	Sandbox	6/2/2019	8/11/2019
Prometheus Client	Sandbox	11/18/2018	8/11/2019

目录

1. Swift 在 Linux 上的发展现状
2. 树莓派和 Swift
3. 为什么尝试 Swift
4. Swift 实际应用 - 智能灯泡

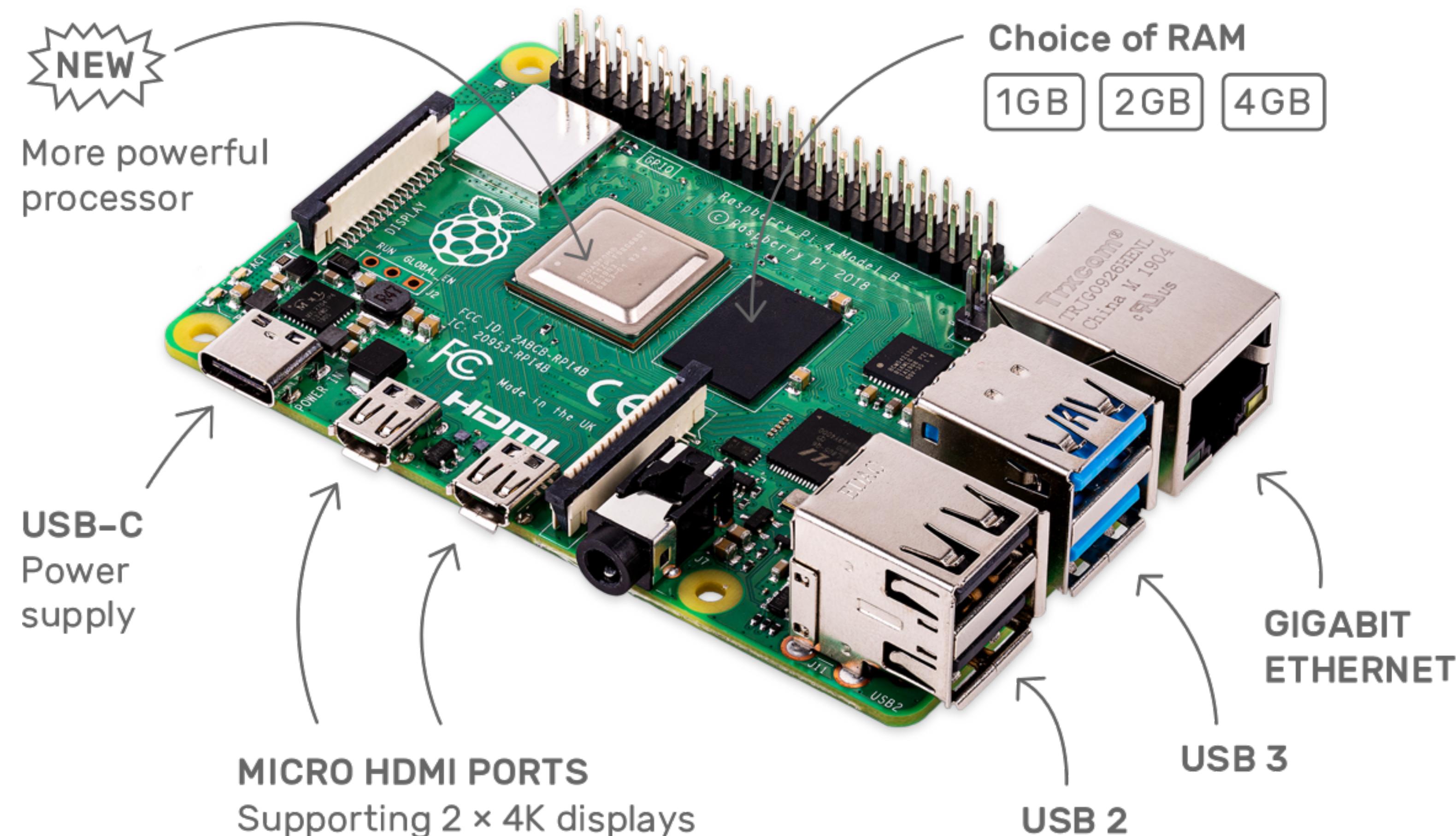
树莓派和 Swift



+



树莓派 4B



树莓派 Zero W

- 802.11 b/g/n wireless LAN
- Bluetooth 4.1
- Bluetooth Low Energy (BLE)
- 1GHz, single-core CPU
- 512MB RAM
- Mini HDMI and USB On-The-Go port
- Micro USB power
- HAT-compatible 40-pin header
- Composite video and reset headers
- CSI camera connector



和 iPhone X 对比



在树莓派上运行 Swift

安装

```
$ curl -s https://packagecloud.io/install/repositories/swift-arm/  
release/script.deb.sh | sudo bash
```

```
$ sudo apt install swift5
```

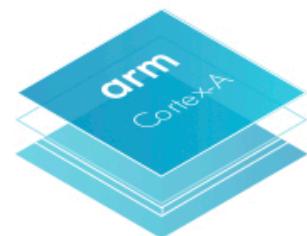
运行

```
$ echo "print(\"hello world\")" > helloworld.swift
```

```
$ swift helloworld.swift
```

在 ARM 下编译 Swift

<https://github.com/uraimo/buildSwiftOnARM>



buildSwiftOn ARM

Scripts to clone, configure, patch and build Swift 5.1.2 on Linux ARM devices.

For precompiled Swift 5.1.2 binaries see the [Prebuilt binaries](#) section, if you want to build Swift on your own instead, check out the [Building on ARM](#) section and the step-by-step instructions.

⌚ Summary

- [Supported Architectures](#)
- [Prebuilt binaries](#)
- [Usage](#)
 - [Dependencies](#)
- [Building on ARM](#)
 - [Step by step instructions](#)
 - [Setup your own build infrastructure](#)
- [GitHub CI on ARM](#)
- [REPL Issues](#)
- [Acknowledgments](#)
- [Previous Releases](#)

目录

1. Swift 在 Linux 上的发展现状
2. 树莓派和 Swift
3. 为什么尝试 Swift
4. Swift 实际应用 - 智能灯泡

为什么尝试 Swift



我学 Swift 是因为我爱 Swift

为什么尝试 Swift

高性能、低功耗

TensorFlow

高级语言

多线程

面向协议/对象编程



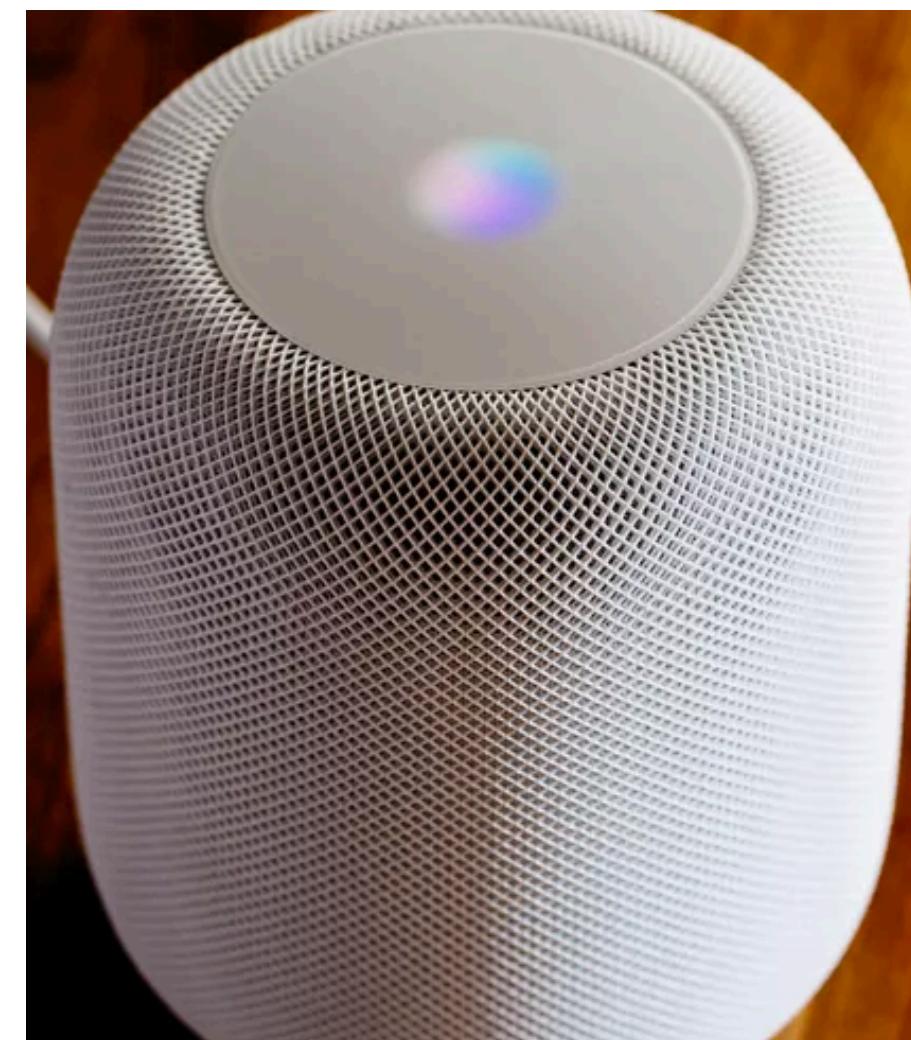
目录

1. Swift 在 Linux 上的发展现状
2. 树莓派和 Swift
3. 为什么尝试 Swift
4. **Swift 实际应用 - 智能灯泡**

嘿 Siri, 开电视

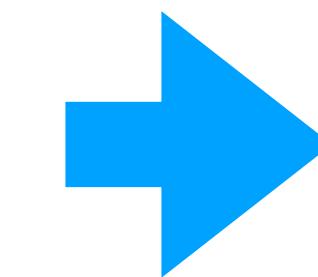


嘿 Siri, 开电视

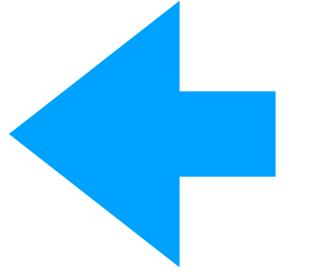




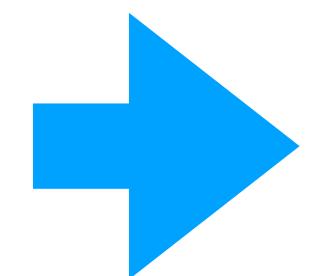
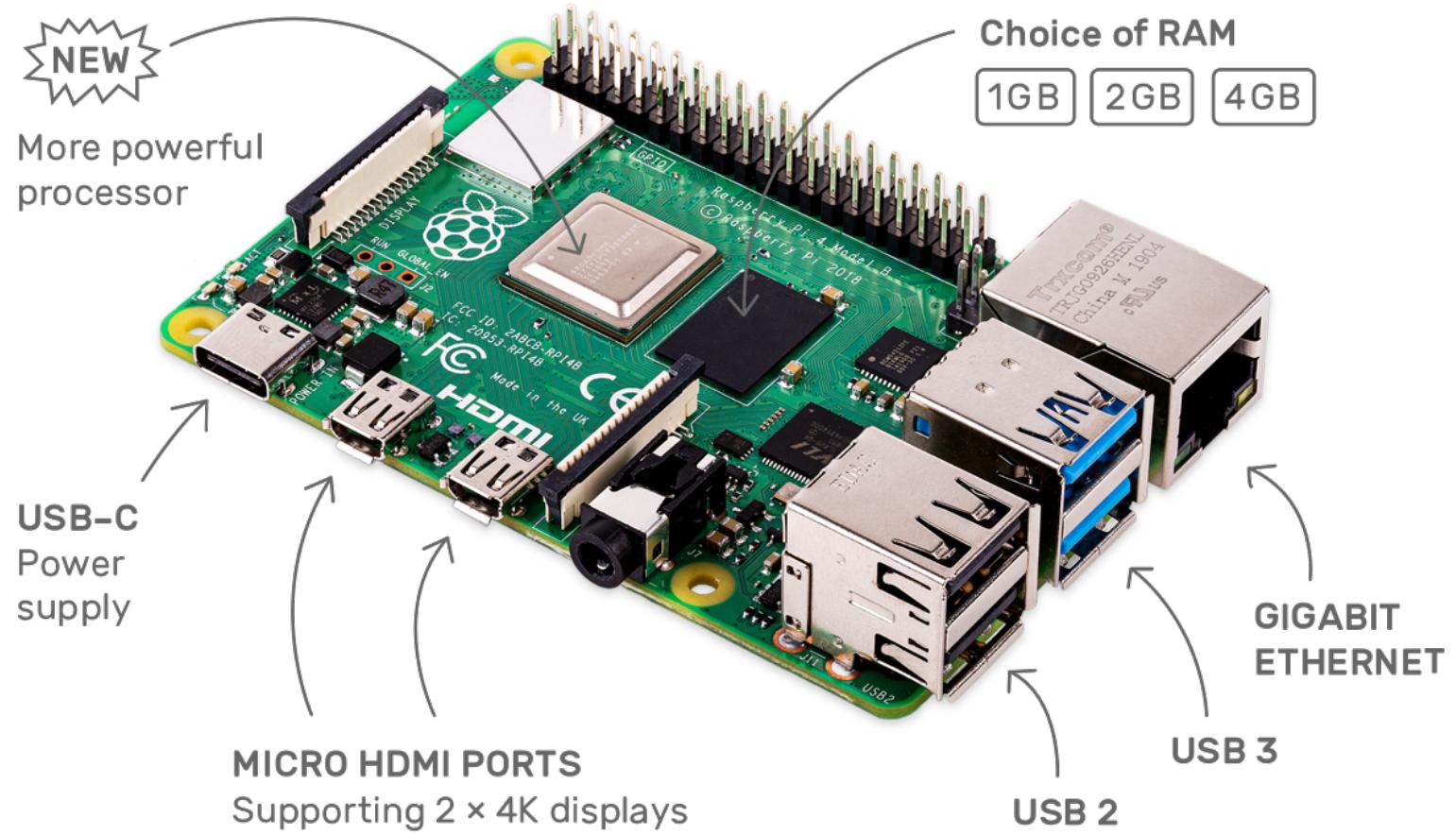
实际应用 - 智能灯泡



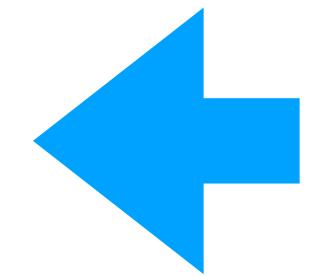
开关



亮灭



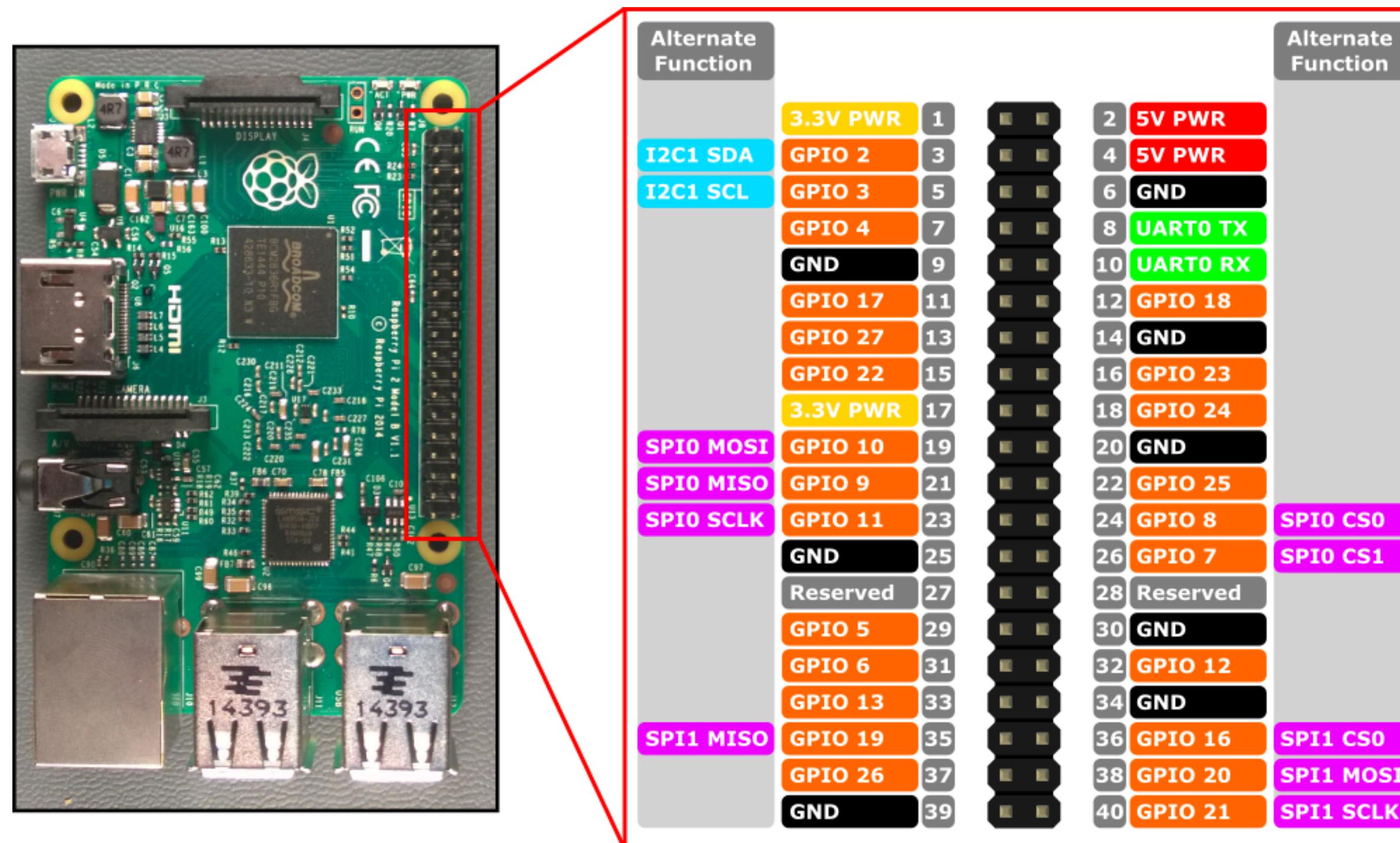
亮灭状态



开关

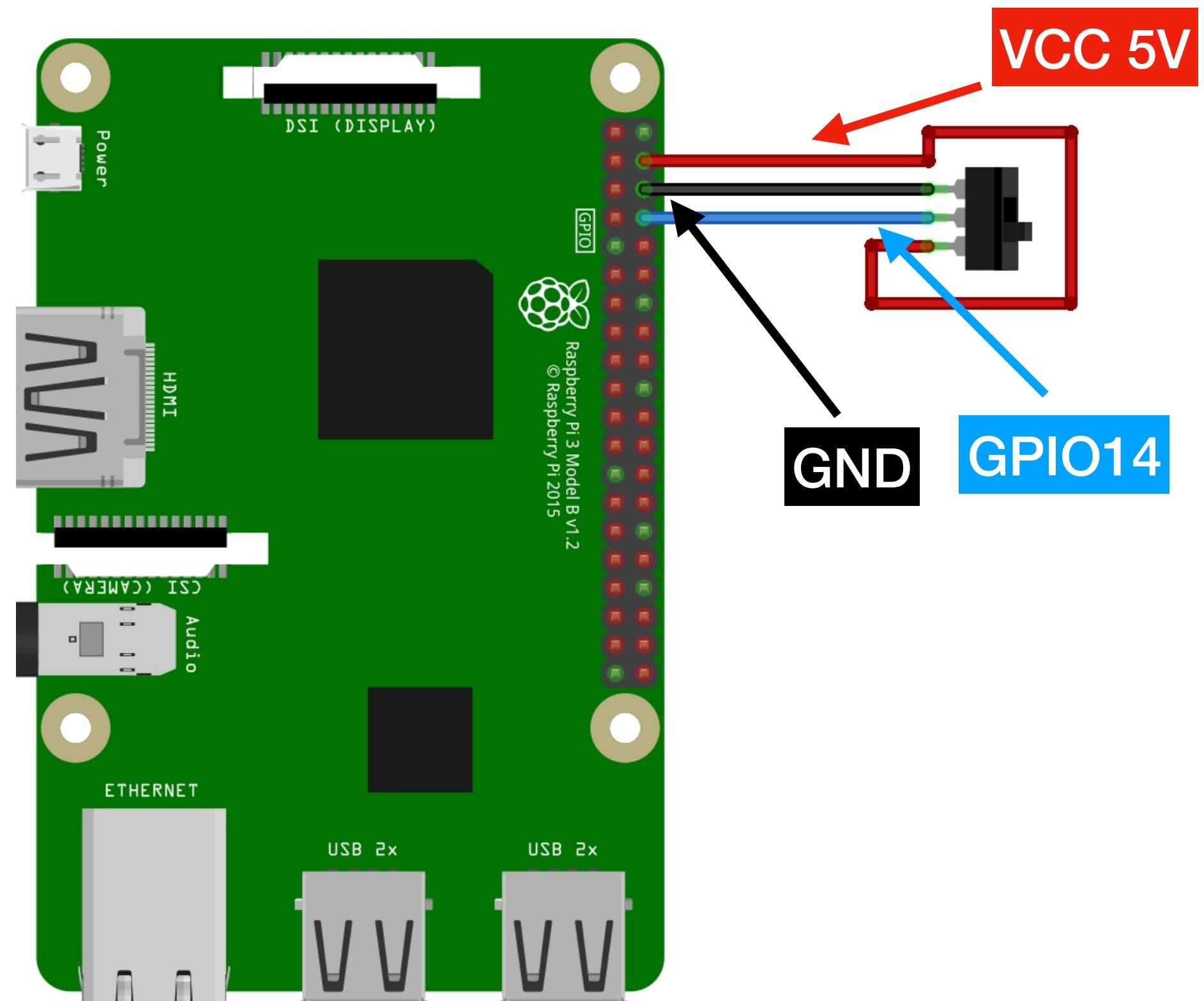


General-purpose input/output (GPIO)



作为输出，输出高电压/低电压
作为输入，接收高电压/低电压

接收按钮输入信号



fritzing

拨到下方 VCC 和 GPIO14 连接
呈高电平

拨到上方 GND 和 GPIO14 连接
呈低电平

引入 SwiftyGPIO 控制引脚



SwiftyGPIO

License MIT Swift 5.x Slack swift/arm build passing

A Swift library for hardware projects on Linux/ARM boards with support for GPIOs/SPI/I2C/PWM/UART/1Wire.

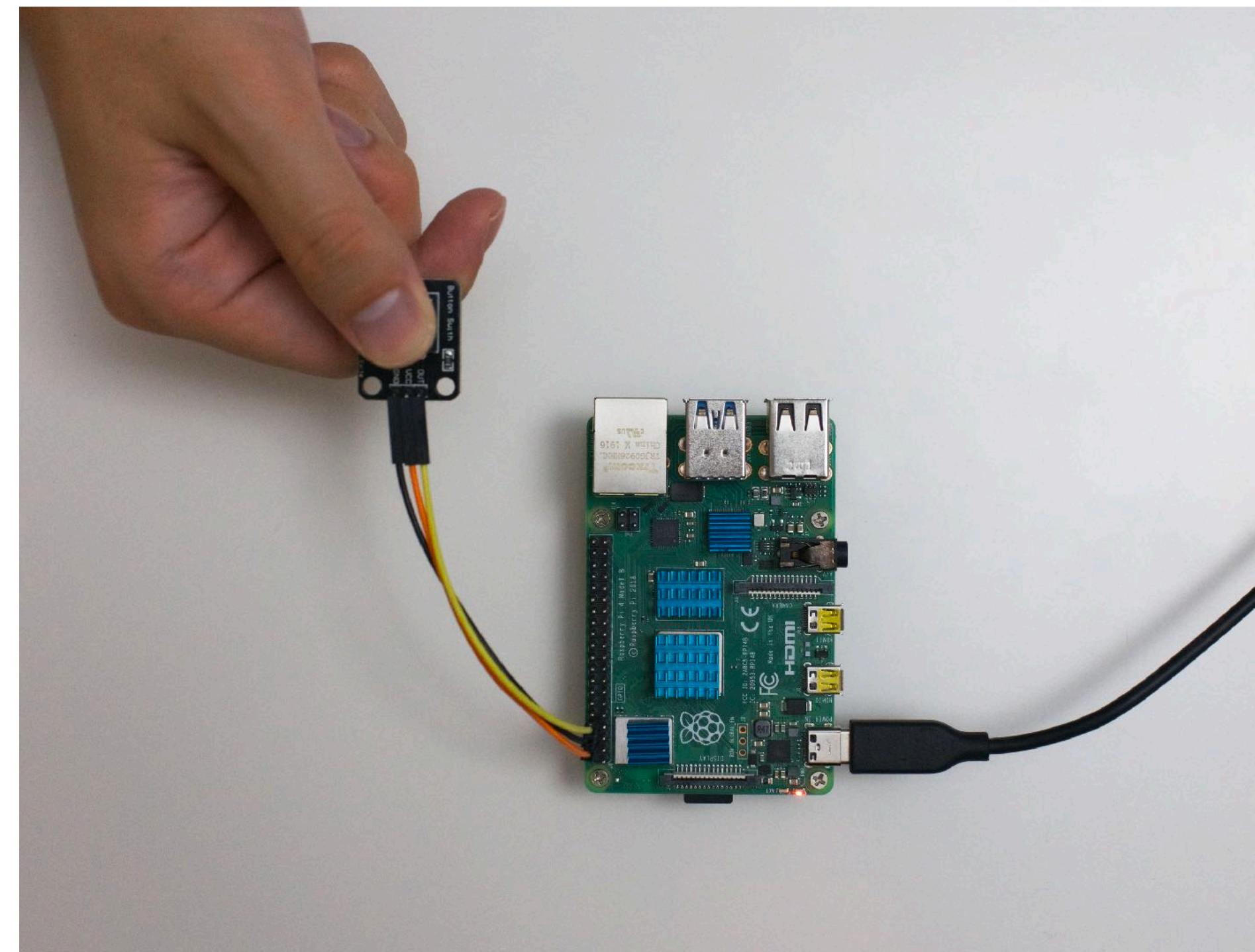


接收按钮输入信号

```
import SwiftyGPIO

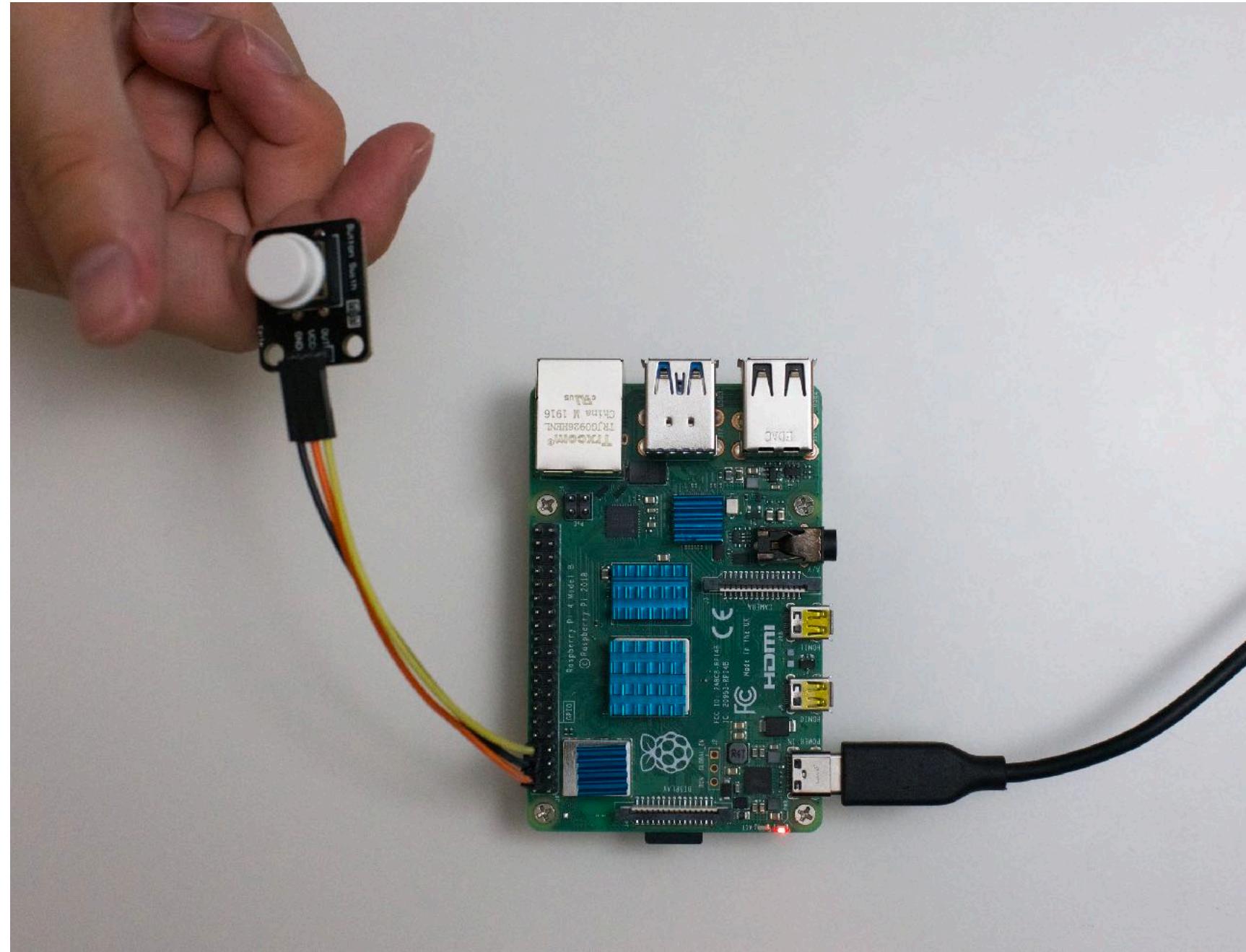
let gpios = SwiftyGPIO.GPIOs(for:.RaspberryPi3)
let button = gpios[.P14]!
button.direction = .IN ← 作为输入接收电压变化
button.onChange { (gpio) in
    print("GPIO Value: \(gpio.value).")
}
```

按下低电平



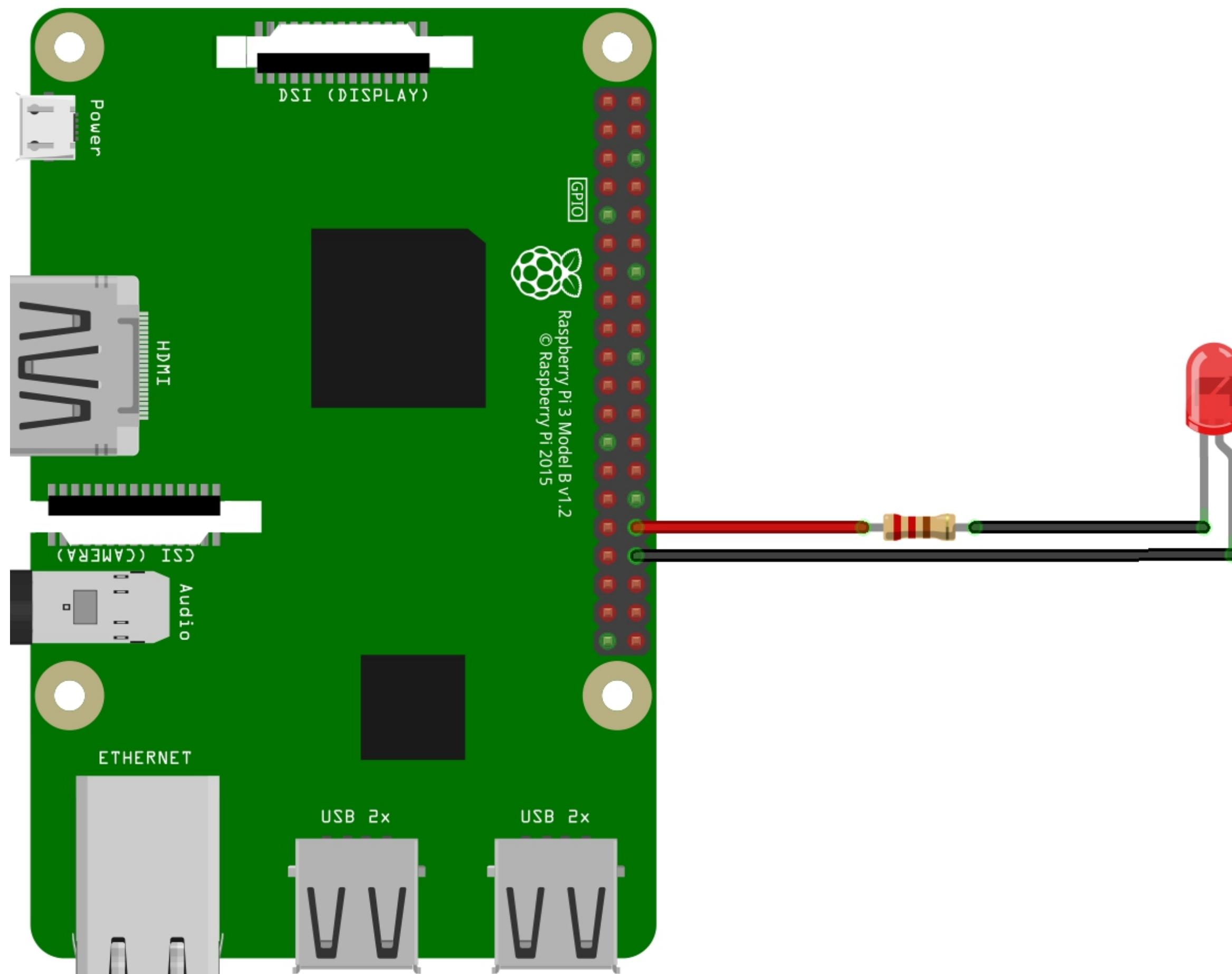
```
pi@raspberrypi:~/GMTCT-Swift-on-Pi/Server $ swift run Button  
GPIO Value: 0.
```

松开高电平



```
pi@raspberrypi:~/GMTC-Swift-on-Pi/Server $ swift run Button
GPIO Value: 0.
GPIO Value: 1.
```

点亮 LED



fritzing

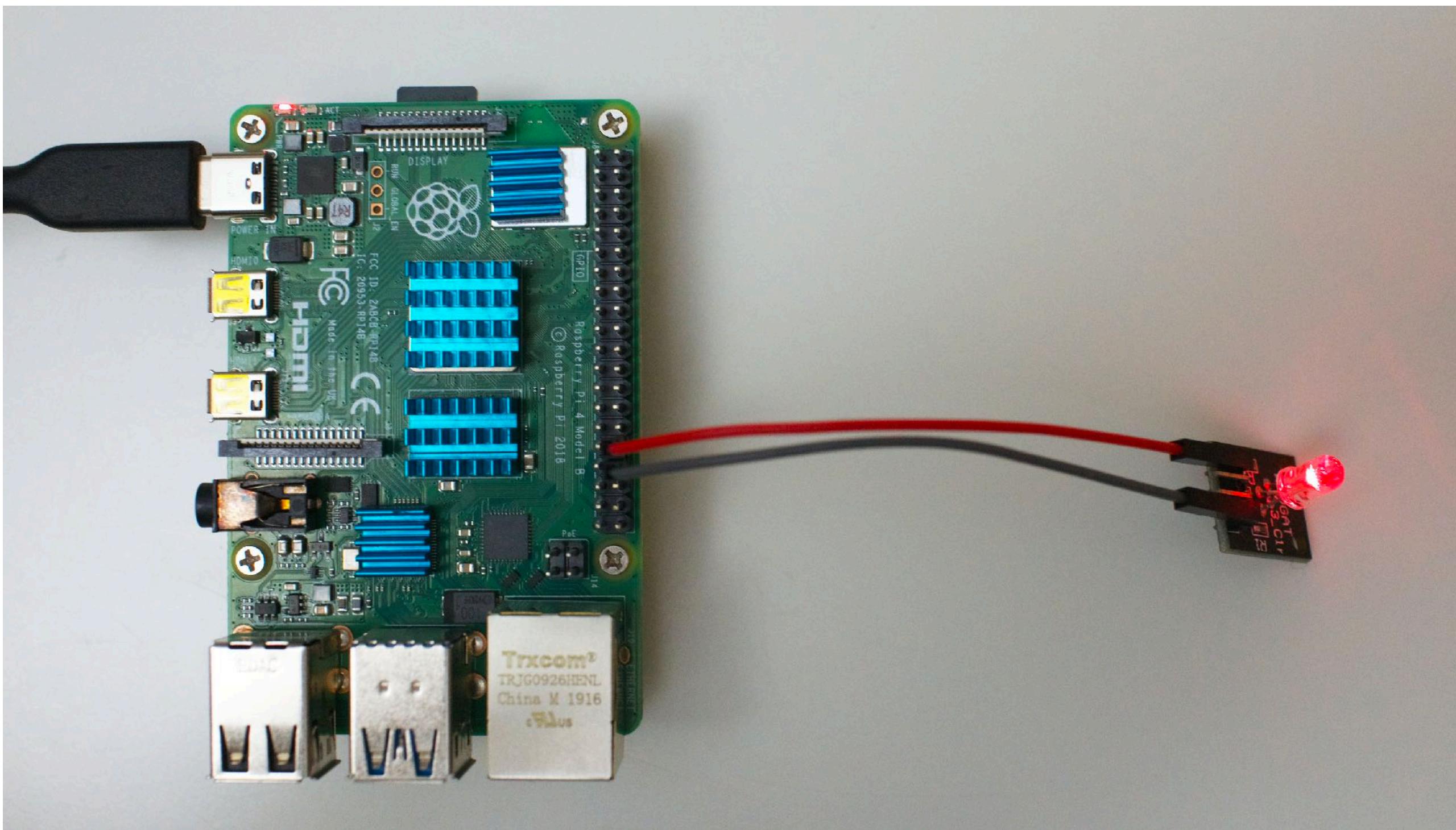
点亮 LED

```
import SwiftyGPIO

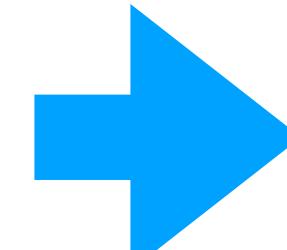
let gpios = SwiftyGPIO.GPIOs(for:.RaspberryPi3)
let led = gpios[P12]! ← 作为输出改变输出电压
led.direction = .OUT
led.value = 1
```

点亮 LED

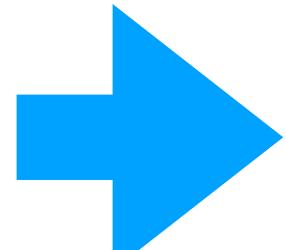
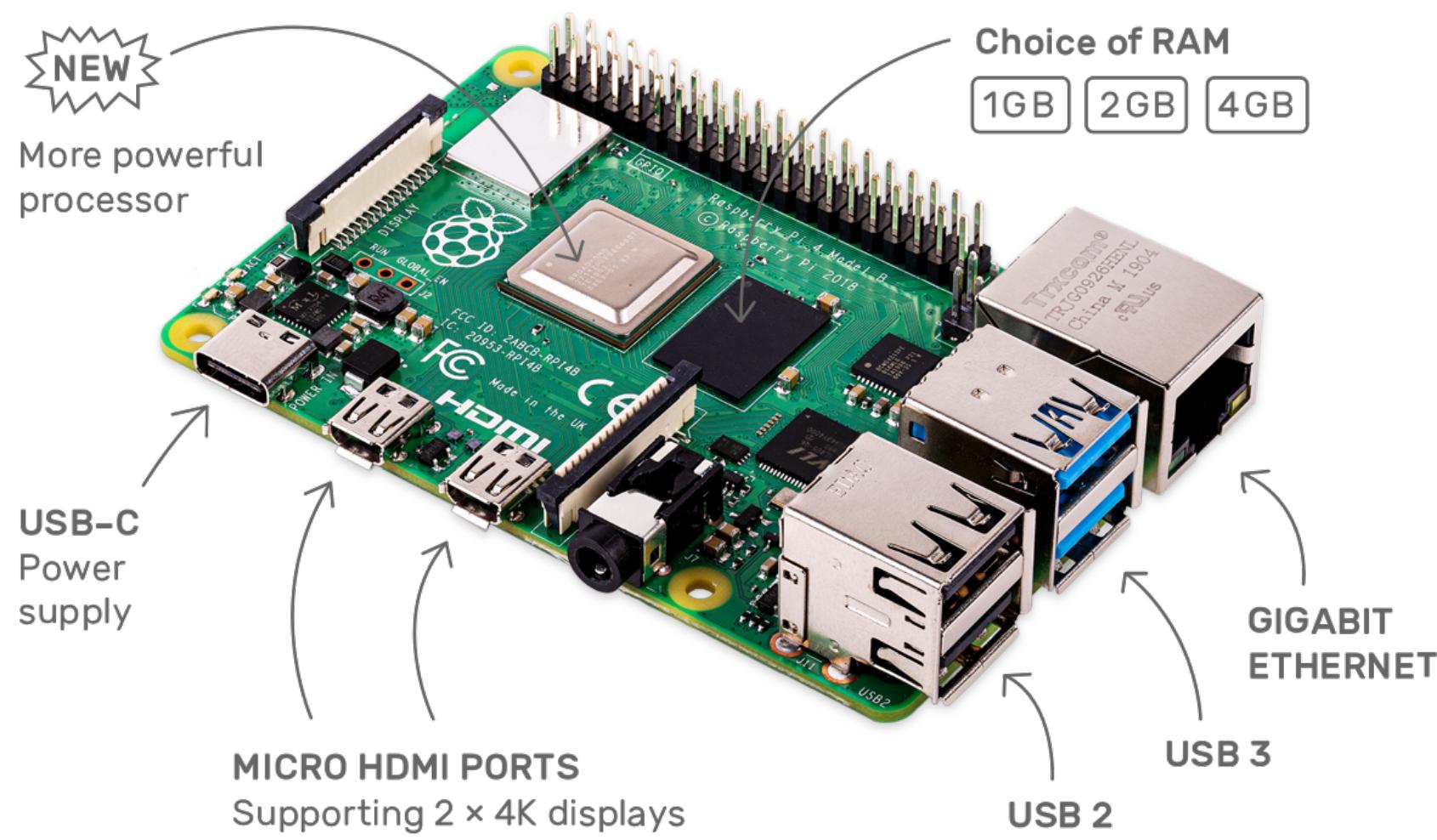
```
pi@raspberrypi:~/GMTC-Swift-on-Pi/Server $ swift run LED  
[14/14] Linking LED  
pi@raspberrypi:~/GMTC-Swift-on-Pi/Server $ █
```



使用按钮控制 LED 灯



开关



亮灭



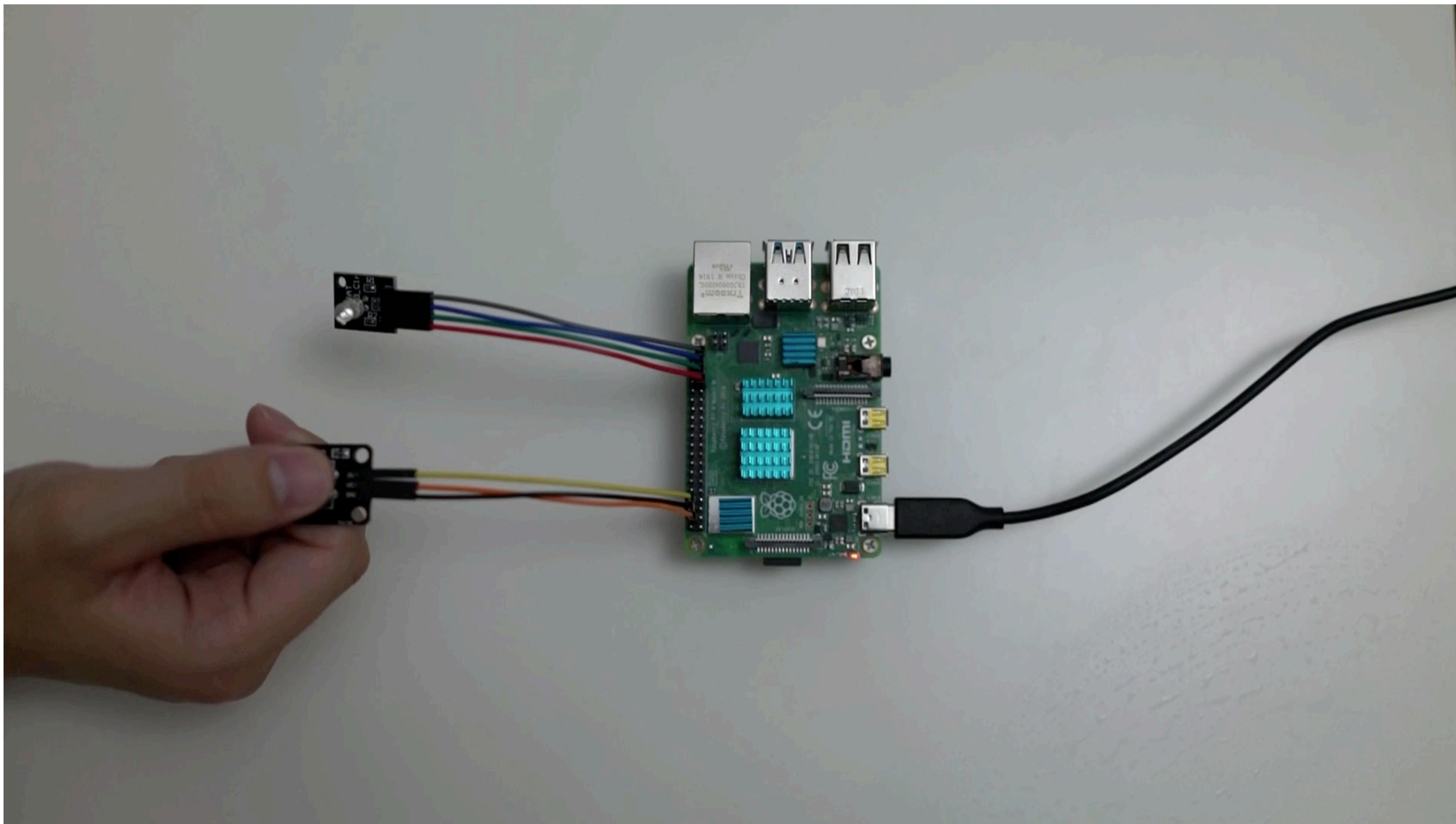
使用按钮控制 LED 灯

```
import SwiftyGPIO

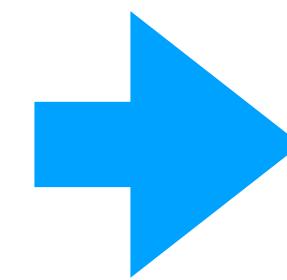
let gpios = SwiftyGPIO.GPIOs(for: .RaspberryPi3)
let rgbLED = RGBLED(red: gpios[.P13]!, green: gpios[.P19]!, blue: gpios[.P26]!)

let button = gpios[.P14]!
button.direction = .IN
button.onRaising { (gpio) in
    rgbLED.switchToNextState()
}
```

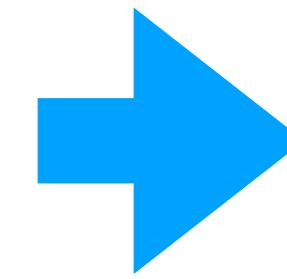
使用按钮控制 LED 灯



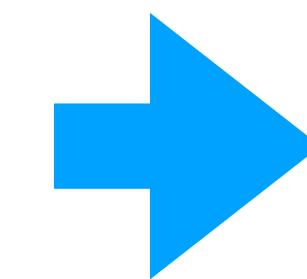
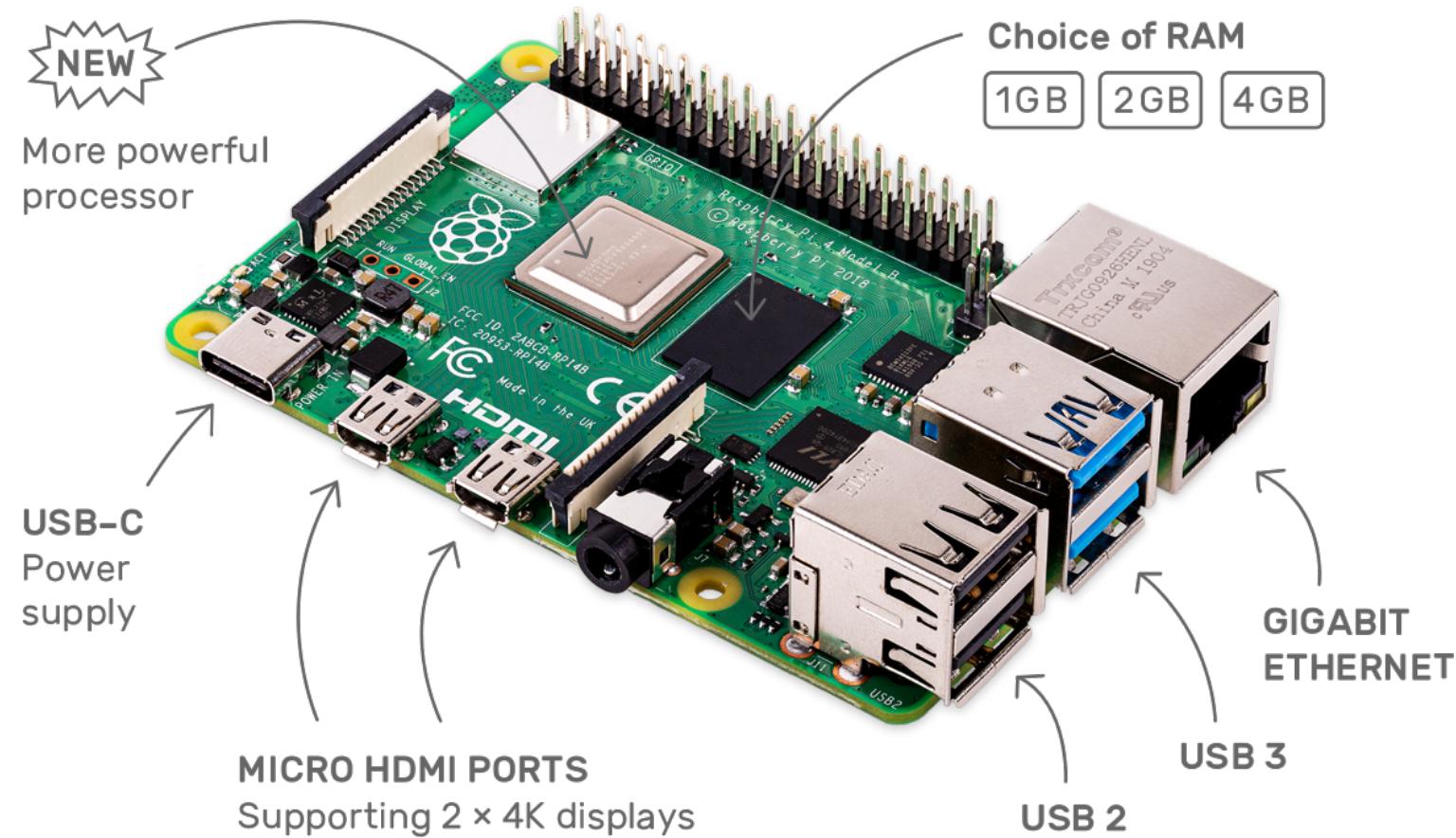
与 iPhone 通信



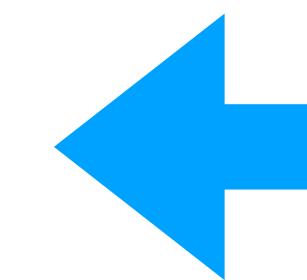
开关



亮灭



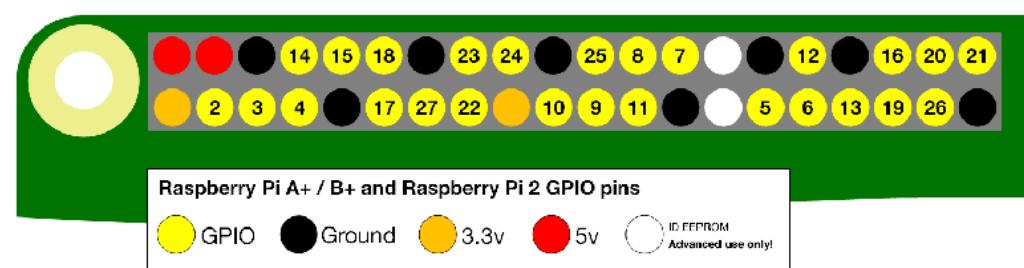
?



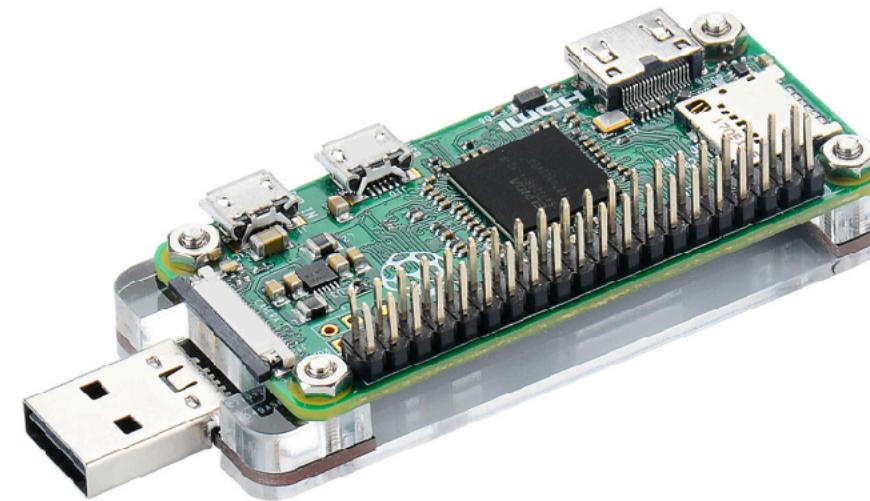
?



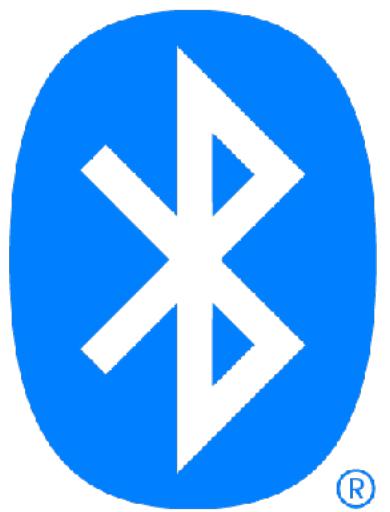
使用什么和树莓派通信？



GPIO



USB

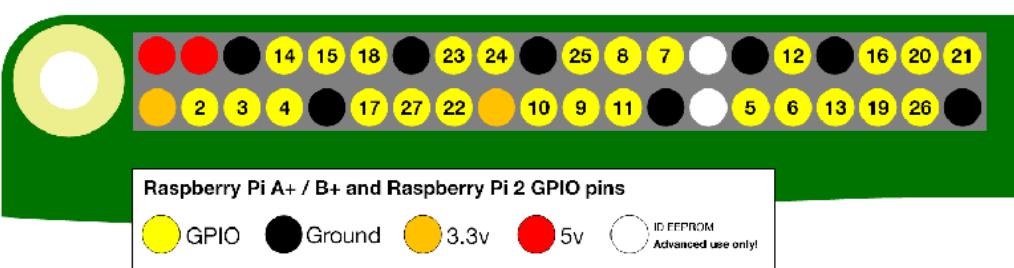


Bluetooth

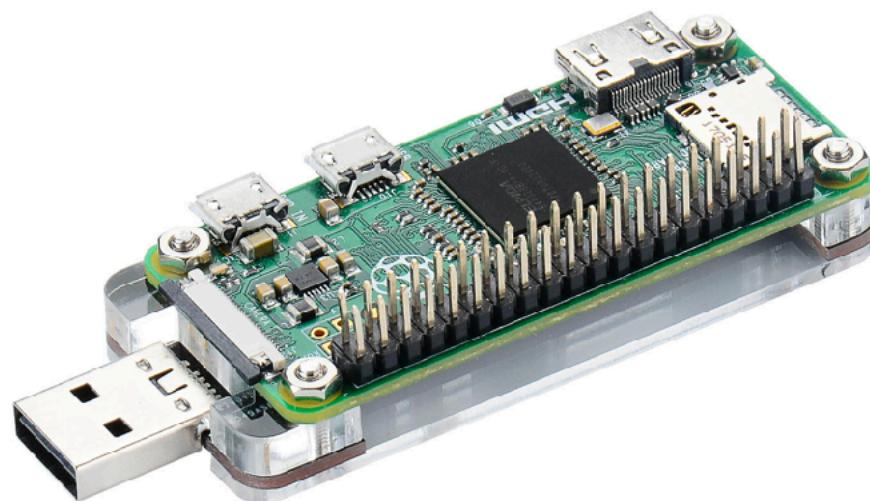


Network

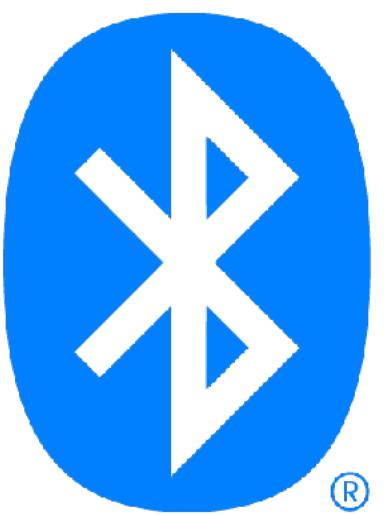
使用什么和树莓派通信？



GPIO



USB



Bluetooth



Network



远程
无距离限制

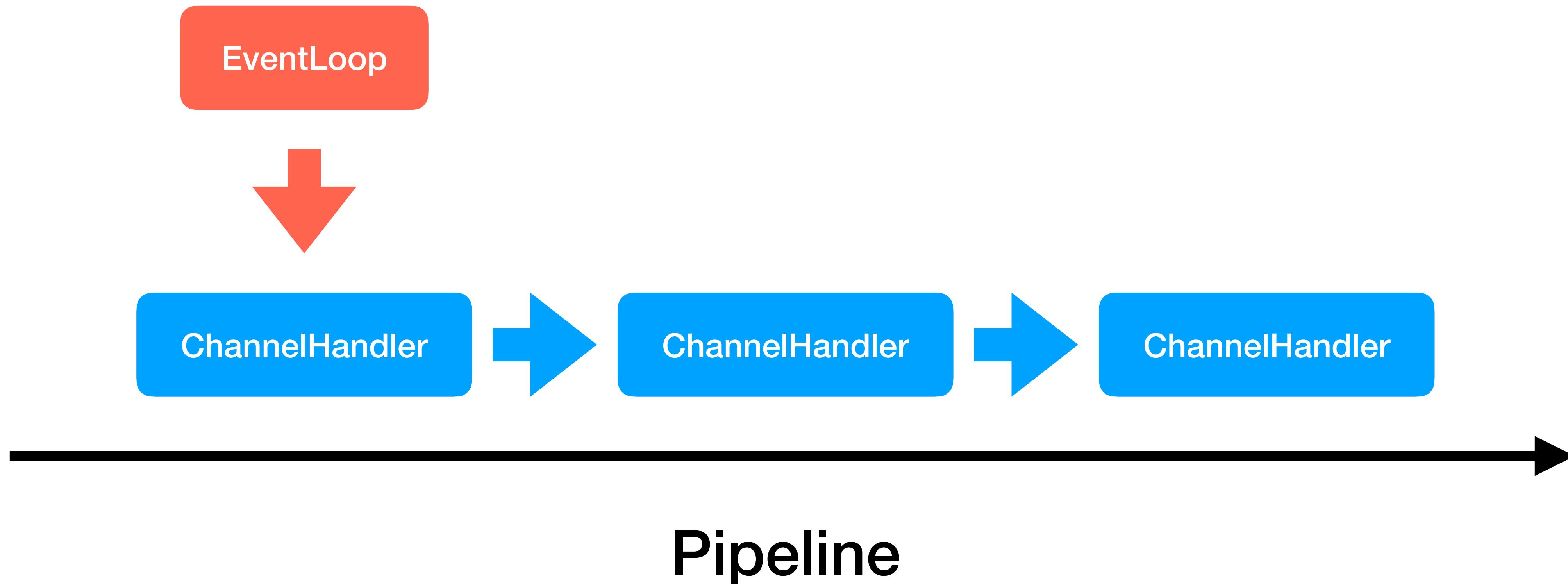
SwiftNIO

- <https://github.com/apple/swift-nio>
- 跨平台，可用于 Linux、macOS、iOS 和 tvOS
- 基于事件驱动的非阻塞式异步网络框架
- 支持丰富的协议，TCP、UDP、HTTP/1、HTTP/2、WebSocket、TLS

EventLoop



Channel



Bootstrap (高阶 API)

Bootstrap

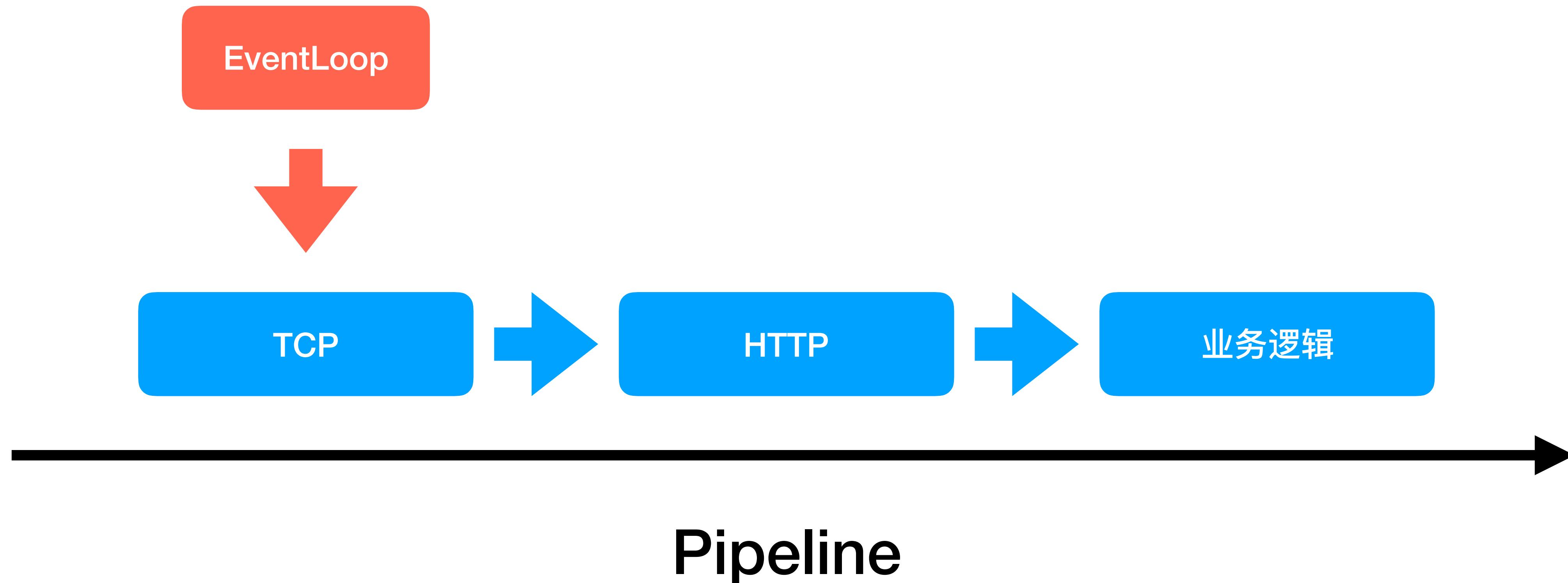
创建

Channel

注册到

EventLoop

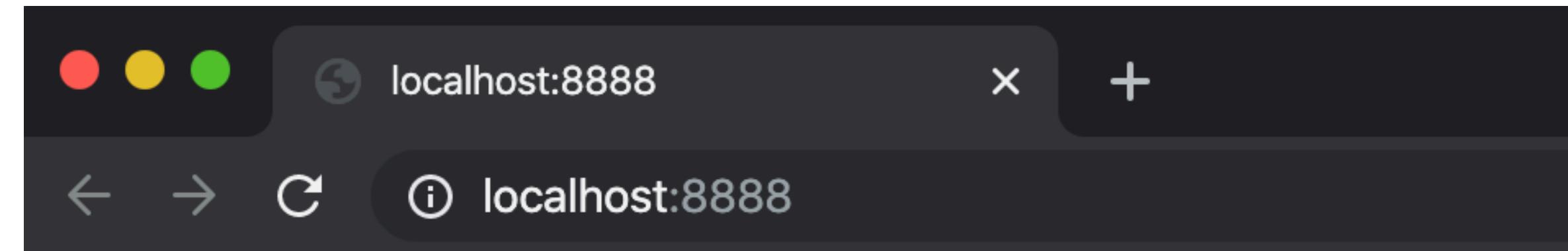
HTTP 服务的 Pipeline



Hello World - 简单 HTTP 服务

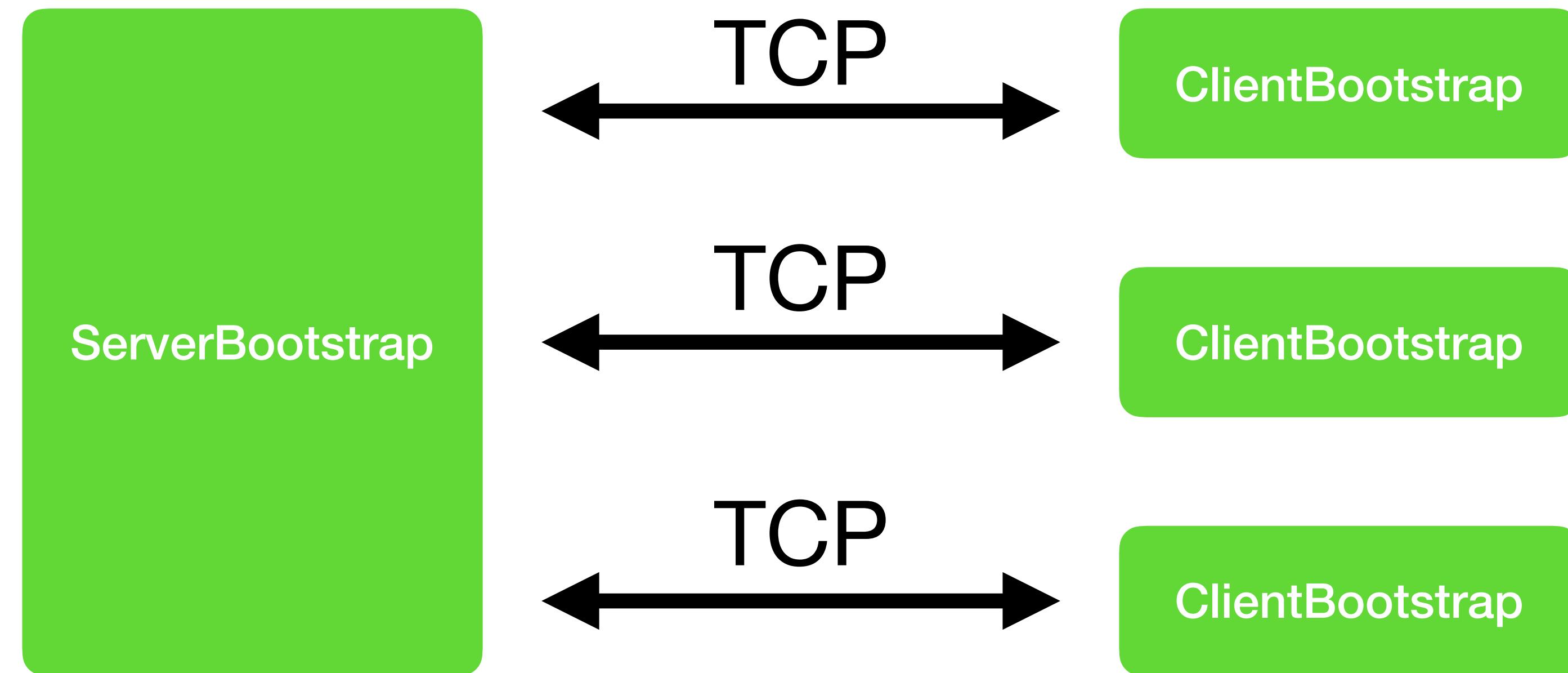
```
let eventLoopGroup = MultiThreadedEventLoopGroup(numberOfThreads: System.coreCount)
let bootstrap = ServerBootstrap(group: eventLoopGroup)
    .serverChannelOption(ChannelOptions.socket(SocketOptionLevel(SOL_SOCKET), SO_REUSEADDR), value: 1)
    .childChannelInitializer { channel in
        channel.pipeline.configureHTTPServerPipeline().flatMap {
            channel.pipeline.addHandler(HelloWorldHTTPHandler())
        }
    }
}

let serverChannel = try bootstrap.bind(host: "localhost", port: 8888).wait()
print("Server running on:", serverChannel.localAddress!)
```



Hello World

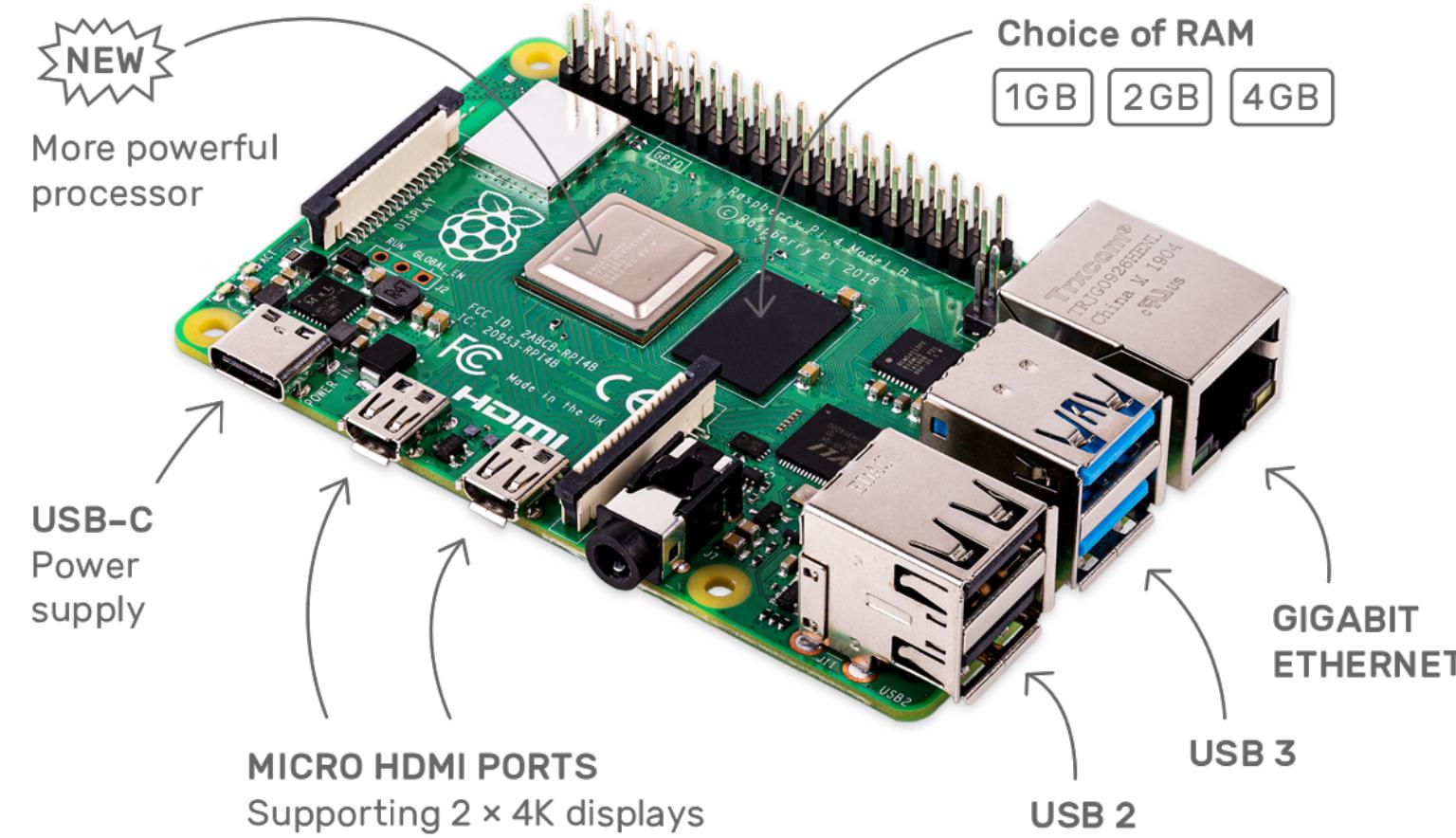
通过 TCP 长连接进行通信



使用 SwiftNIO 连接树莓派和 iPhone



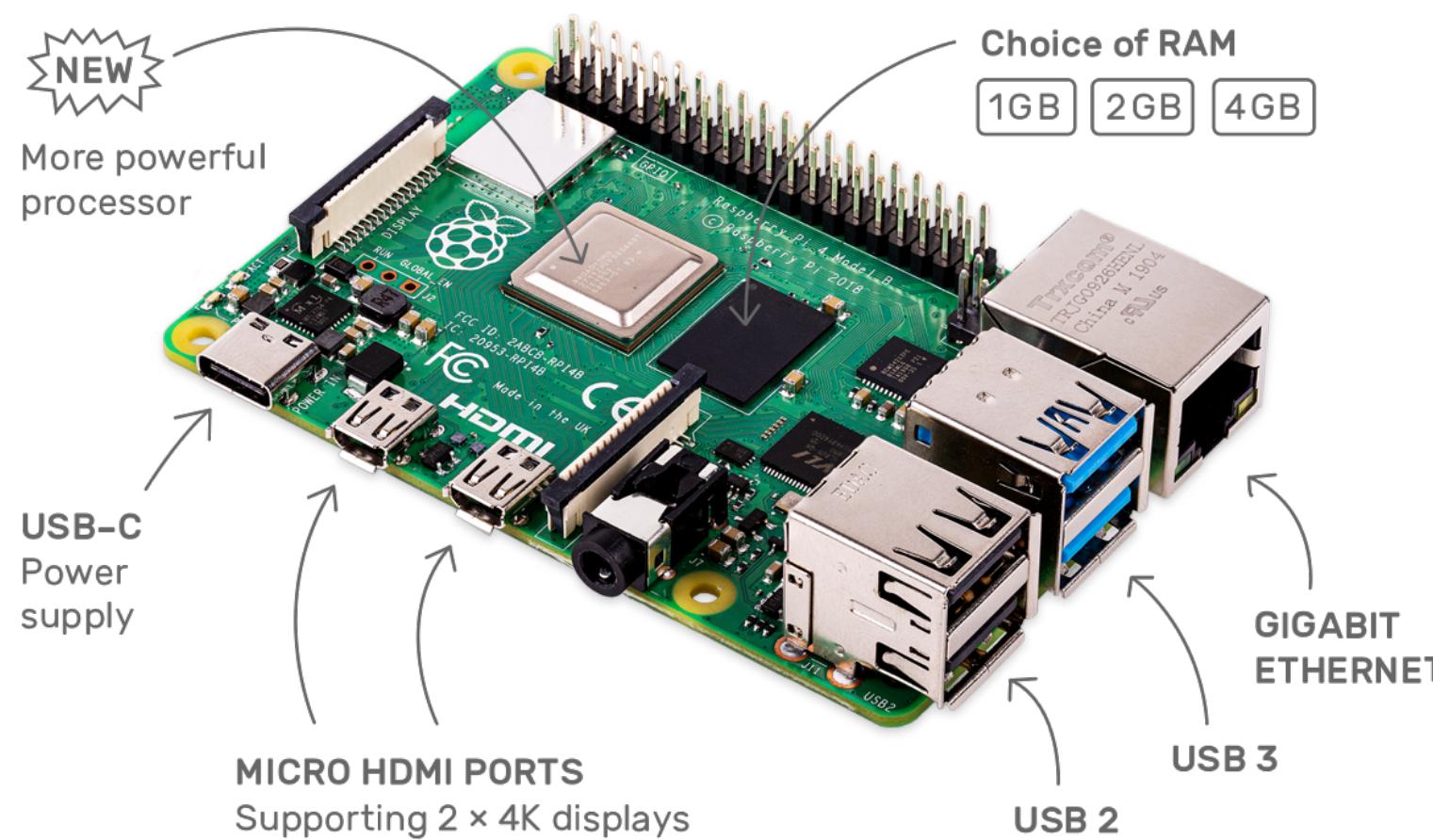
↔
GPIO



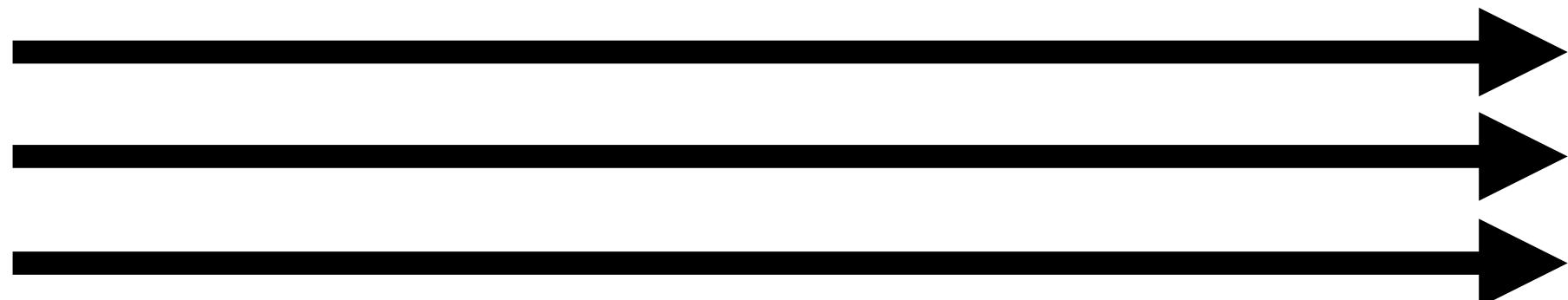
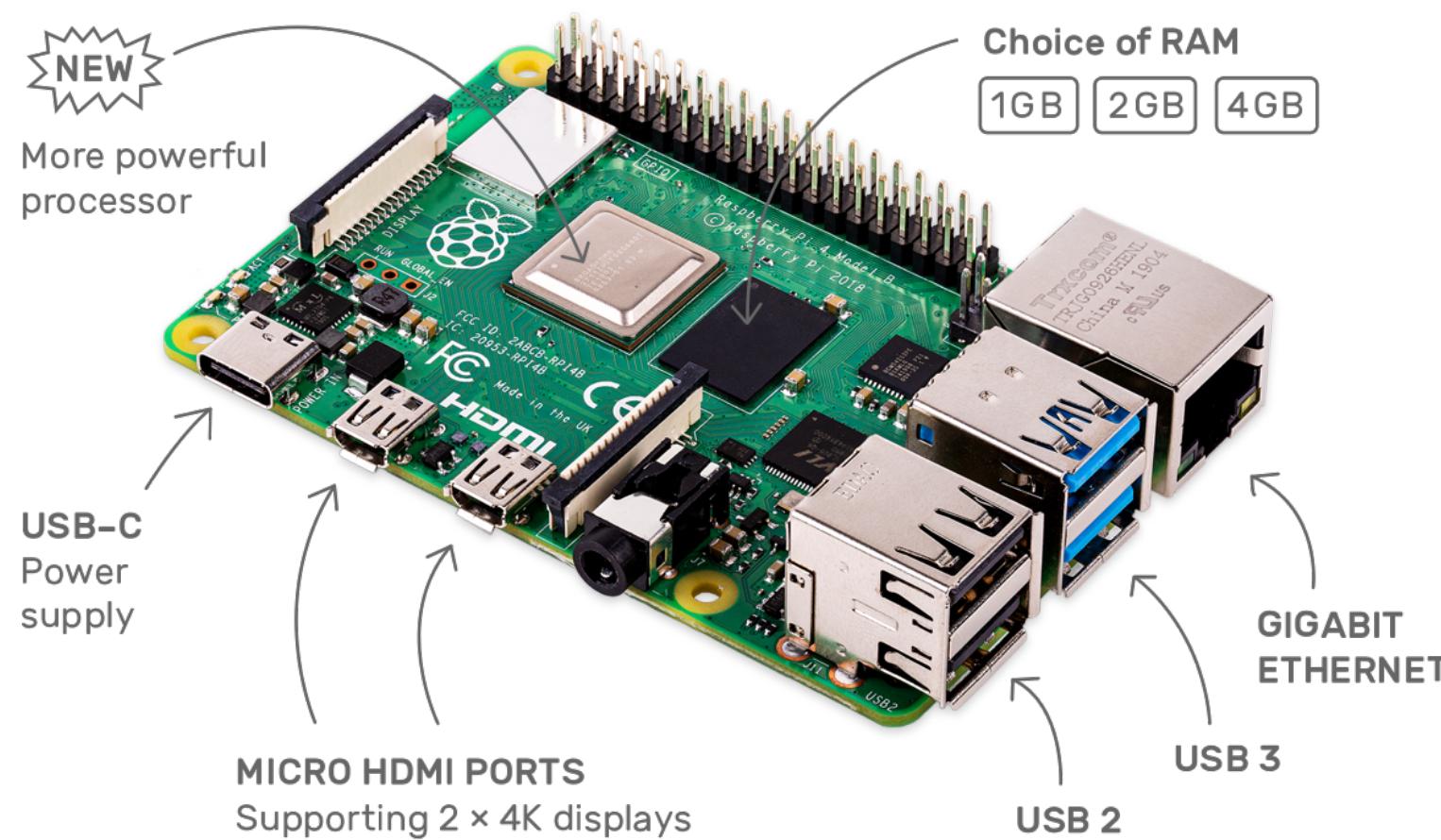
↔
TCP



使用 SwiftNIO 连接树莓派和 iPhone



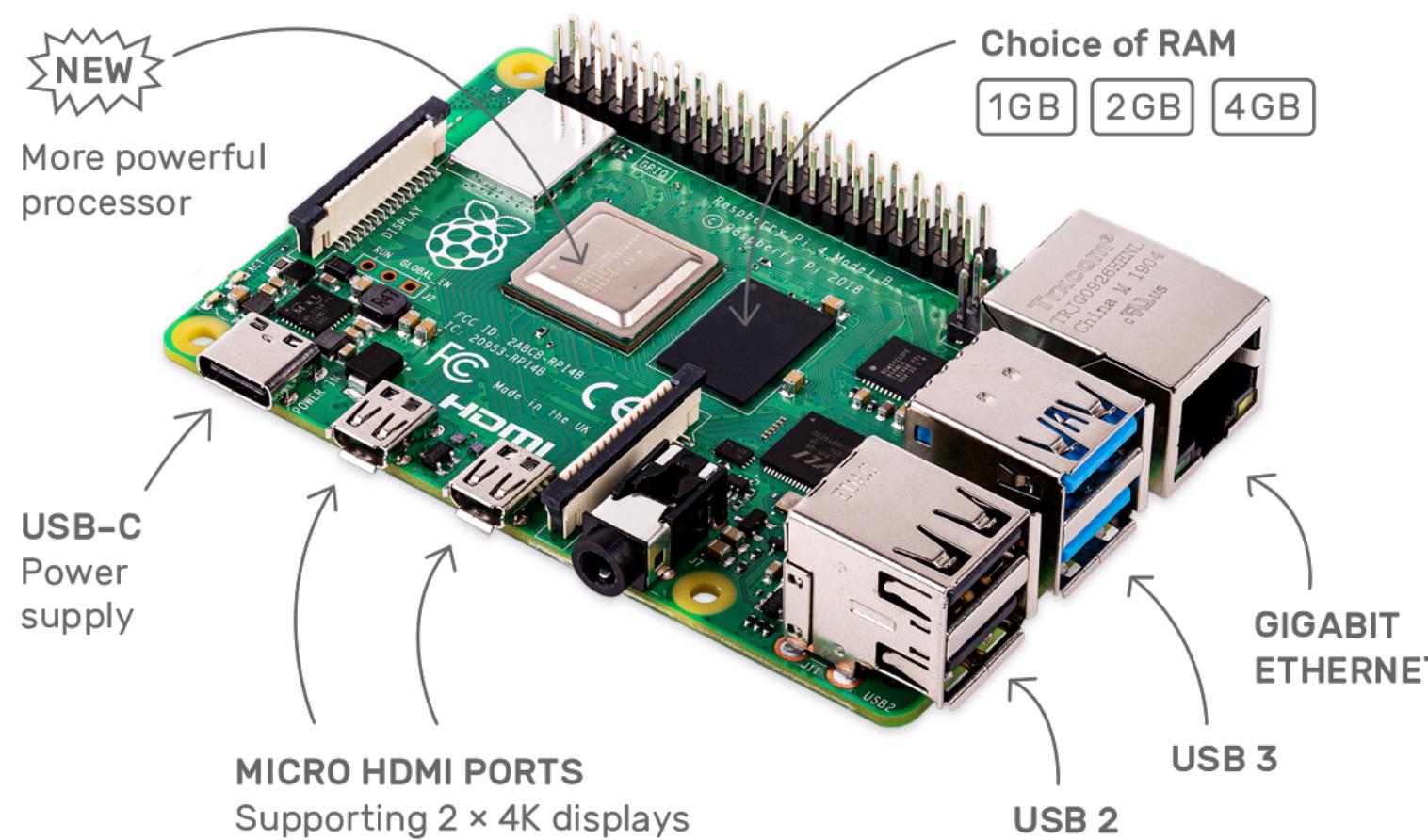
使用 SwiftNIO 连接树莓派和 iPhone



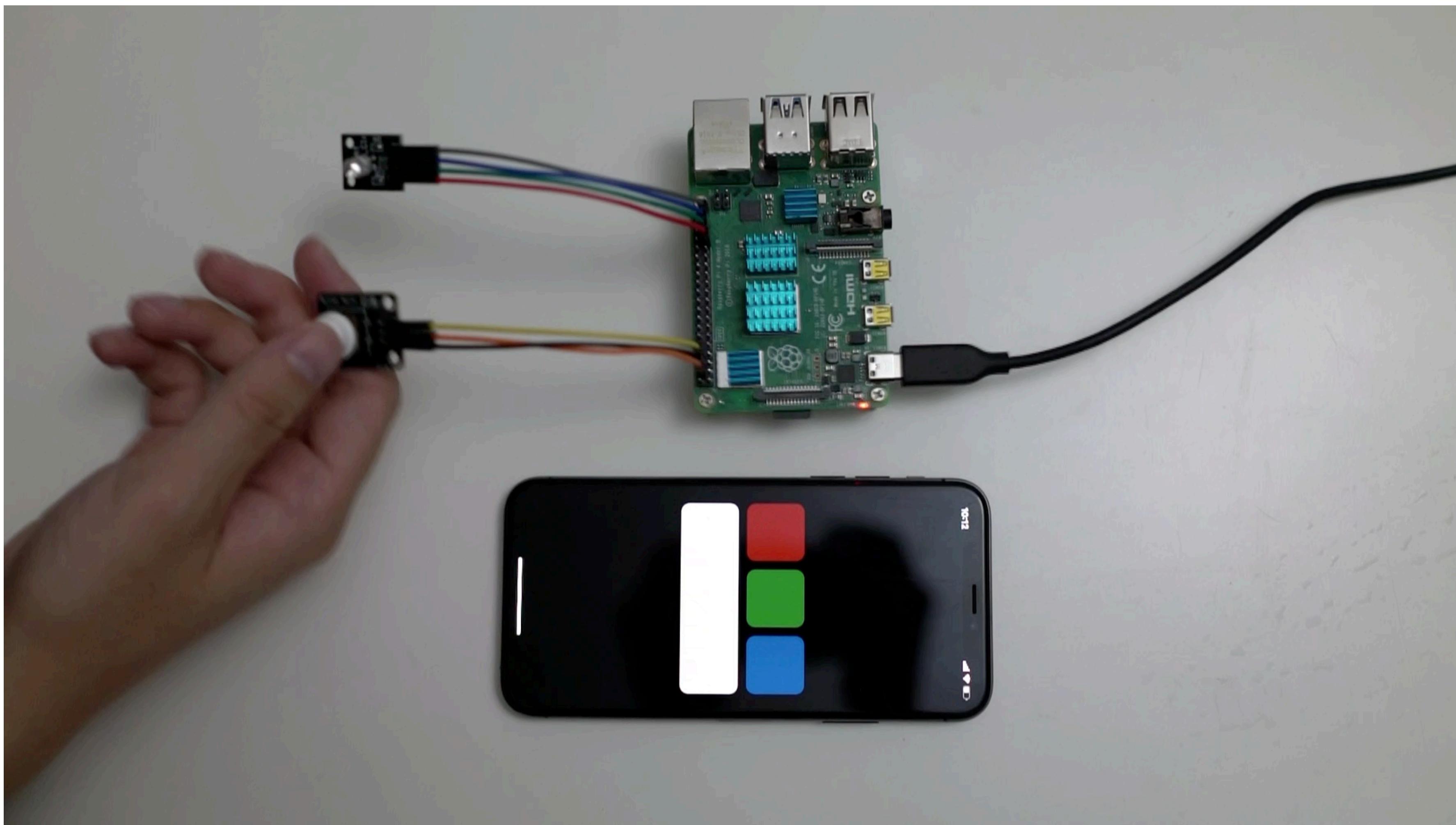
推送物理开关修改状态



使用 SwiftNIO 连接树莓派和 iPhone



使用 SwiftNIO 连接树莓派和 iPhone



提个醒

- 一个 Server (智能灯泡) 对多 Client (多个手机) 问题
- 实际应用会采用标准协议, 比如 MQTT
- 实现 UDP 组播 / Bonjour 自动发现设备

总结

- Swift 已经可以在 ARM 设备上进行嵌入式开发，我们有了从 0 到 1 的过程
- 现在如何上手 Swift 嵌入式开发？从树莓派开始
- Demo 工程：<https://github.com/DianQK/GMTC-Swift-on-Pi>
- 除了灯泡做什么？所有电子设备改造，比如空气净化器、饮水机、电饭煲



提问送树莓派



批量购课
扫码联系顾问

5G 时代大前端进阶之路

重学前端 | 每天 10 分钟，重构你的前端知识体系

iOS 开发高手课 | 从原理到实战，带你解决 80% 的开发难题

Vue 开发实战 | 从 0 开始搭建大型 Vue 项目

Flutter 核心技术与实战 | 来自 Google 的高性能跨平台开发框架

Android 开发高手课 | 由表及里，给你的 App“把脉”

React 实战进阶 45 讲 | 掌握当下热门的前端利器

Node.js 开发实战 | 带你开发一个完整的 Node.js 项目

DevOps 实战笔记 | 精要 30 计，让 DevOps 快速落地

批量购课特惠

购买本系列课程总价满 ¥1000，享**8** 折优惠。

获取优惠，请联系客服「豆包」  |  13167596032



60天学透算法与数据结构

600+名企内推通道向你打开！

大厂内推 实战演练 闭环体系 社群连接

THANKS

