

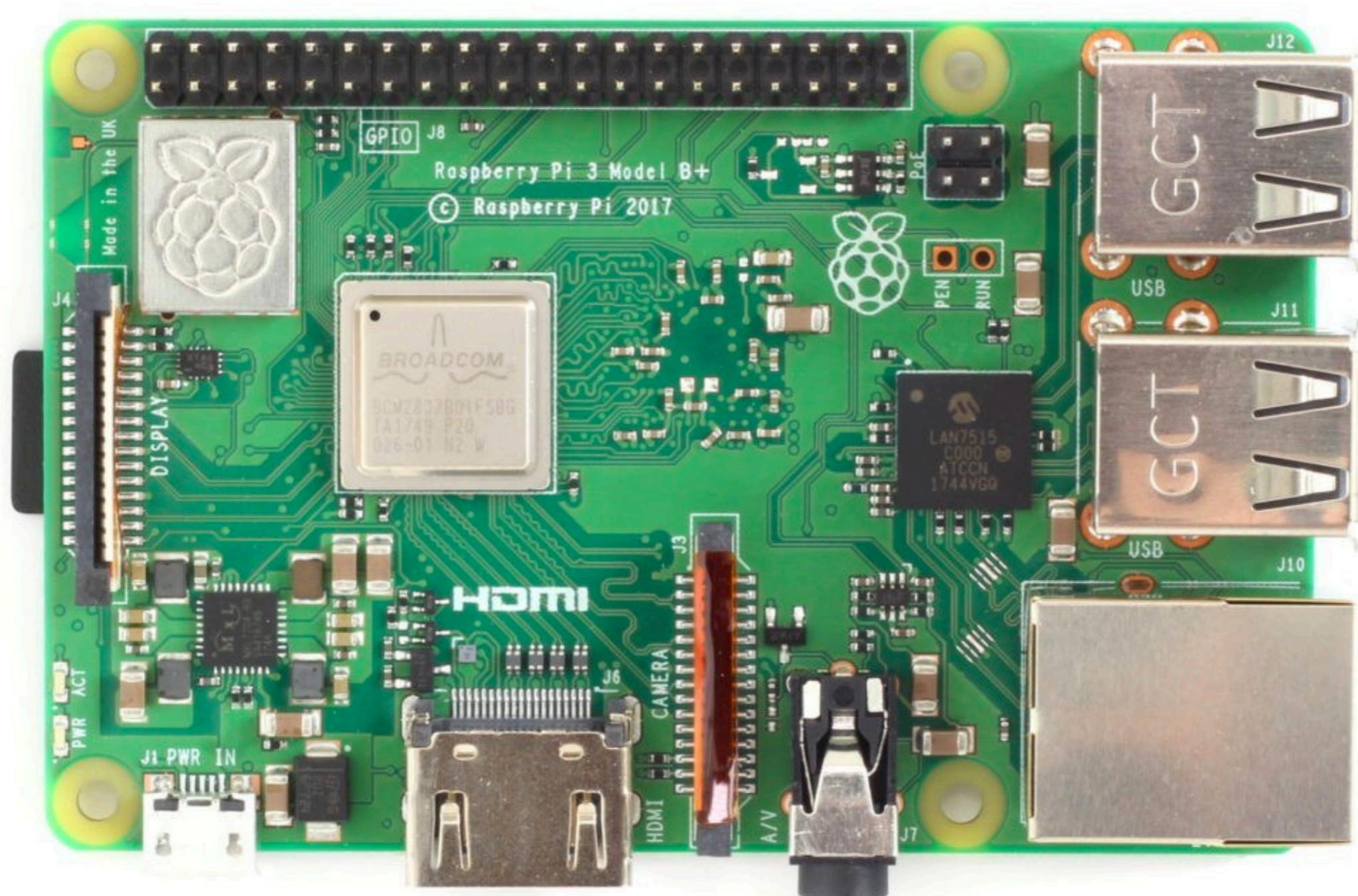
Lecture #5

Raspberry Pi

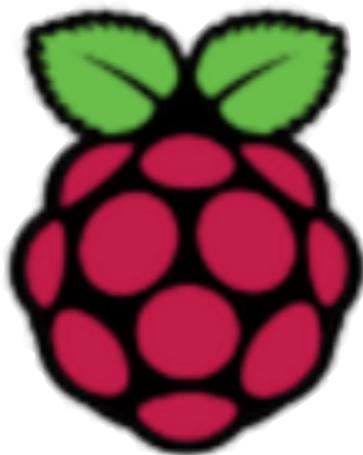
Spring 2024

Options

android
things 



Options



Raspberry Pi

Raspberry Pi Documentation - X +

← → C 🔒 raspberrypi.com/documentation/computers/raspberry-pi.html ⌂ ☆ ⚙ 📸 :

Computers

- Getting Started
- Raspberry Pi OS
- Configuration
- The config.txt file
- The Linux kernel
- Using Linux
- Remote Access
- Raspberry Pi Hardware**

Schematics and Mechanical Drawings

- Raspberry Pi 4 Model B
- Raspberry Pi 3 Model B+
- Raspberry Pi 3 Model B
- Raspberry Pi 2 Model B
- Raspberry Pi Model B+
- Raspberry Pi 3 Model A+
- Raspberry Pi Model A+
- Raspberry Pi Zero
- Raspberry Pi Zero W
- Raspberry Pi Zero 2 W

Product compliance and safety

- Flammability Rating

The Raspberry Pi Documentation is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

Raspberry Pi Hardware

Schematics and Mechanical Drawings

Edit this on GitHub

Schematics for the various Raspberry Pi board versions:

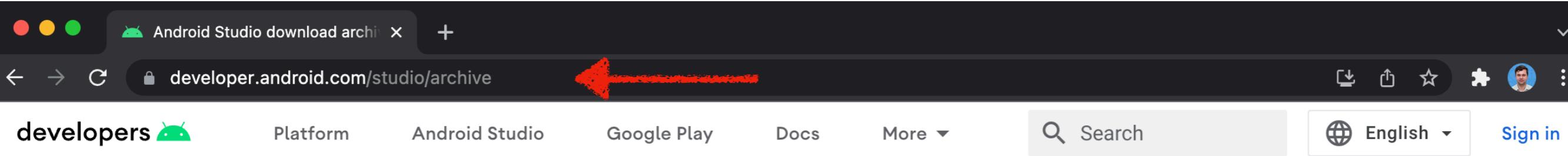
Raspberry Pi 4 Model B

- [Schematics, Revision 4.0](#)
- [Mechanical Drawings, PDF](#)
- [Mechanical Drawings, DXF](#)

Raspberry Pi 3 Model B+

- [Schematics, Revision 1.0](#)
- [Mechanical Drawings, PDF](#)
- [Mechanical Drawings, DXF](#)
- [Case Drawings, PDF](#)

Raspberry Pi 3 Model B



A screenshot of a web browser window. The address bar shows the URL developer.android.com/studio/archive. Above the address bar, there is a navigation bar with icons for back, forward, and search. A red arrow points to the back button icon. The main content area has a header "Android Studio download archives" with a search bar and language selection. Below the header is a list of Android Studio release archives.

Android Studio download archives

This page provides an archive of Android Studio releases.

However, we recommend that you download the [latest stable version](#) or the [latest preview version](#).

▼ [Android Studio Chipmunk \(2021.2.1\) Beta 4](#) March 9, 2022

▼ [Android Studio Dolphin \(2021.3.1\) Canary 5](#) March 3, 2022

▼ [Android Studio Dolphin \(2021.3.1\) Canary 4](#) February 28, 2022

▼ [Android Studio Chipmunk \(2021.2.1\) Beta 3](#) February 28, 2022

▼ [Android Studio Bumblebee \(2021.1.1\) Patch 2](#) February 23, 2022

▼ [Android Studio Dolphin \(2021.3.1\) Canary 3](#) February 14, 2022

▼ [Android Studio Dolphin \(2021.3.1\) Canary 2](#) February 9, 2022

▼ [Android Studio Chipmunk \(2021.2.1\) Beta 2](#) February 9, 2022

▼ [Android Studio Bumblebee \(2021.1.1\) Patch 1](#) February 04, 2022

▼ [Android Studio Dolphin \(2021.3.1\) Canary 1](#) January 31, 2022

Android Studio download archive

developer.android.com/studio/archive

developers  Platform Android Studio Google Play Docs More ▾ Search English Sign in

Android Studio Bumblebee (2021.1.1) Canary 4 July 21, 2021

Android Studio Arctic Fox (2020.3.1) RC 1 July 20, 2021

Android Studio Bumblebee (2021.1.1) Canary 3 July 8, 2021

Android Studio Arctic Fox (2020.3.1) Beta 5 July 2, 2021

Android Studio 4.2.2 June 30, 2021 

Installers

Windows IDE only (64-bit): [android-studio-ide-202.7486908-windows.exe](#) (977126064 bytes)

Chrome OS: [android-studio-ide-202.7486908-cros.deb](#) (848496828 bytes)

Mac: [android-studio-ide-202.7486908-mac.dmg](#) (980425788 bytes)

SHA-256 checksums

517305de6a9558a7075201dfffc2483dc9cc46e31632d781172bd8385c0f2c526 android-studio-ide-202.7486908-windows.exe
ca560d5cb47fdc0850c6d7ce23b068c0b5da047de12dfb21399938c9041ac327 android-studio-ide-202.7486908-cros.deb
63b8e5bed9a772c19398604bd7c897a543d4164a534775ad44a91110f49532c7 android-studio-ide-202.7486908-mac.dmg

Zip files

Windows (64-bit): [android-studio-ide-202.7486908-windows.zip](#) (980051073 bytes)

Mac: [android-studio-ide-202.7486908-mac.zip](#) (981368856 bytes)

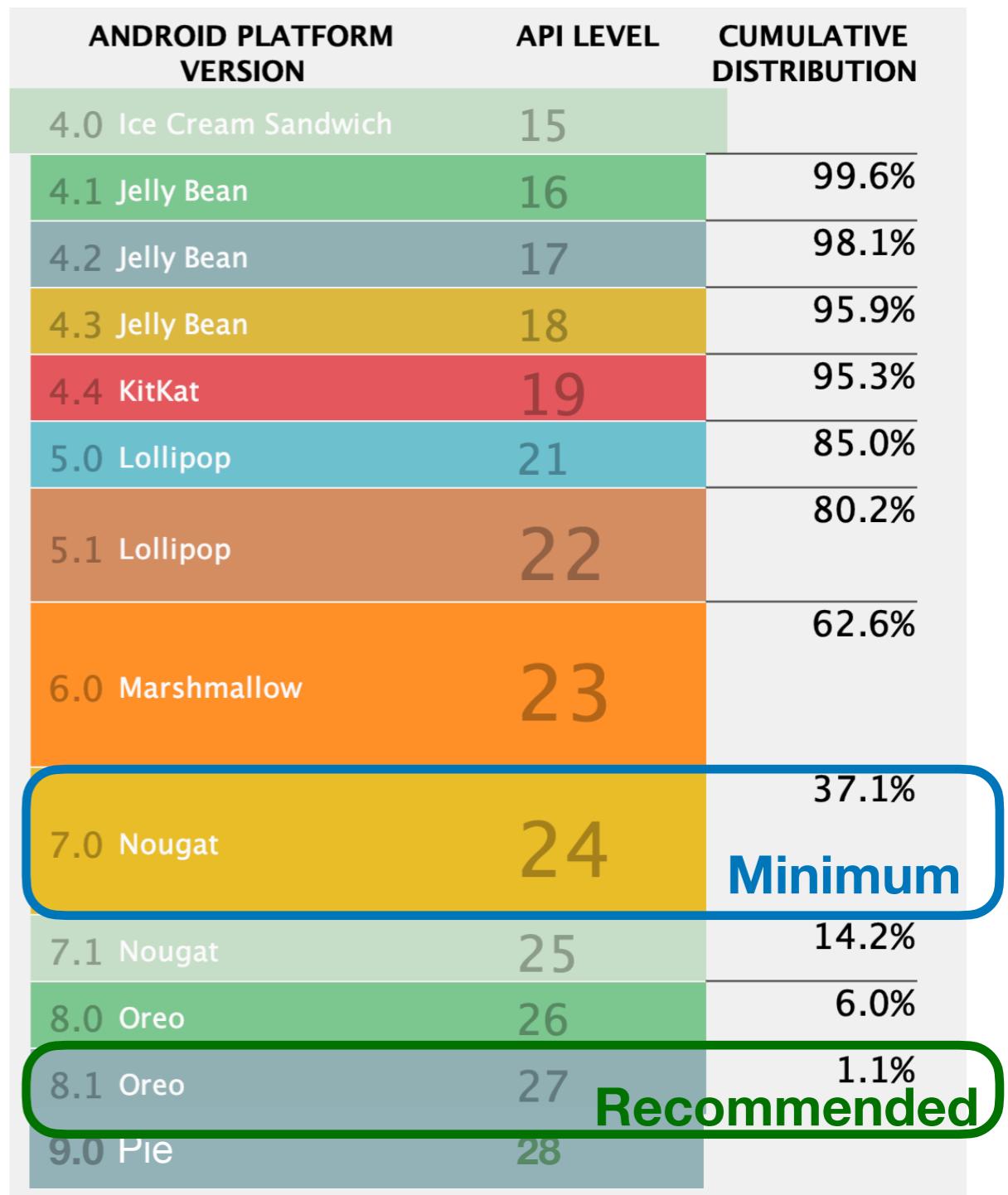
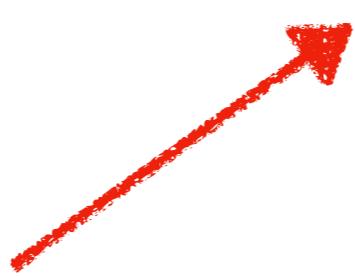
Linux: [android-studio-ide-202.7486908-linux.tar.gz](#) (996627232 bytes)

SHA-256 Checksums

61054257662fc81aaf95334b74081959fe44e50c824f68d8083b440dfb5dc88d android-studio-ide-202.7486908-windows.zip
d816331e1be453d7964c6a6a92e17ba4b22e5dc505553dd51515e94f86fd95c android-studio-ide-202.7486908-mac.zip
733b04cb66e3ff7f03766aa0222ebd1fef7a63e6340665aa91f0d62e724feca3 android-studio-ide-202.7486908-linux.tar.gz

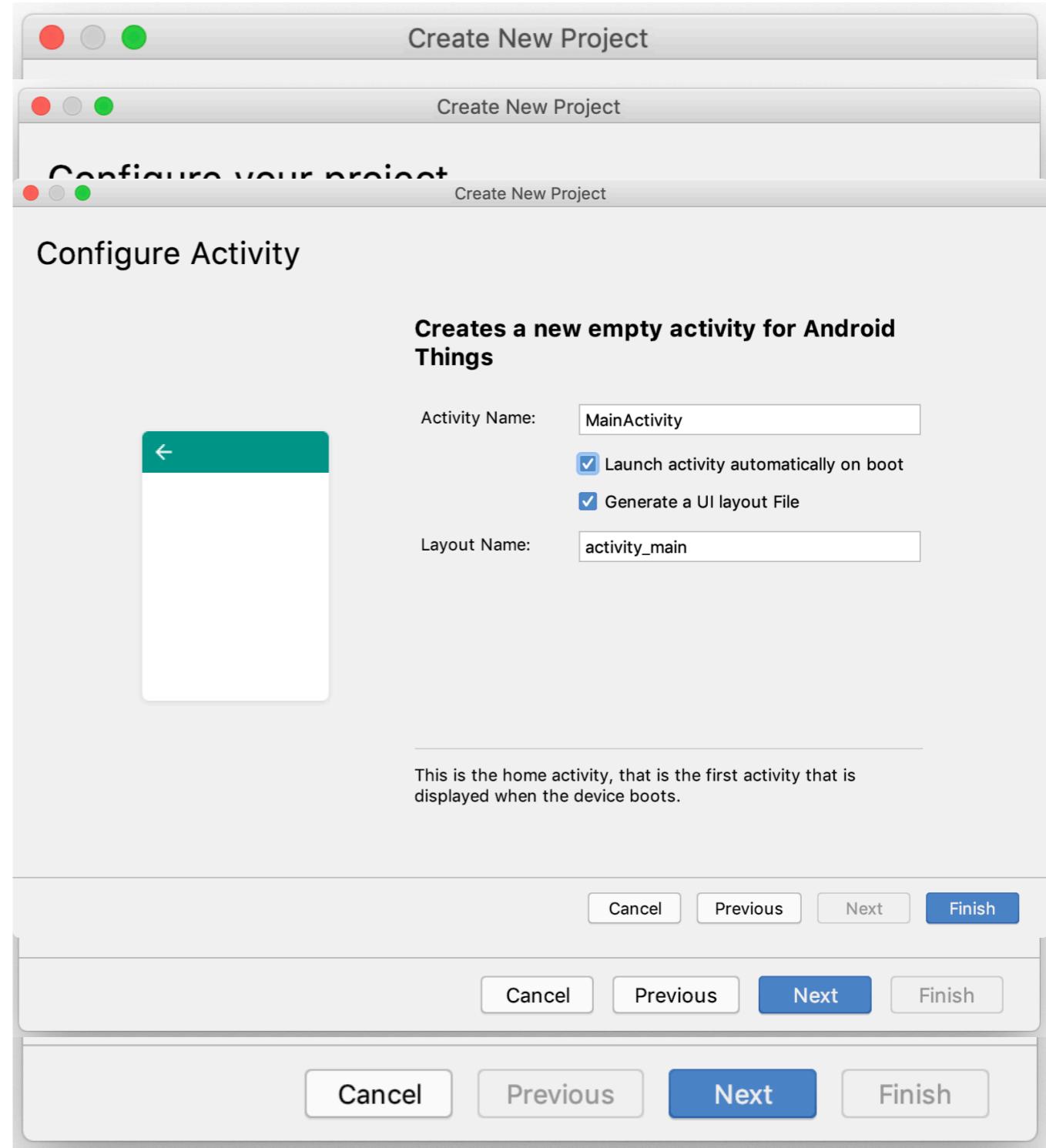
Project Sample

- Prerequisites
 - SDK Tools at least 25.0.3.
 - SDK with API 27 or higher.



Create the Project

- Select **Android Things** as the only form factor.
- Ensure that API 27 is selected.
- Check AndroidX artifact.
- Ensure that the activity will start automatically on boot



Key Generated Changes

- Build.gradle changes.

```
dependencies {  
    ...  
    compileOnly 'com.google.android.things:androidthings:+'  
}
```

Key Generated Changes

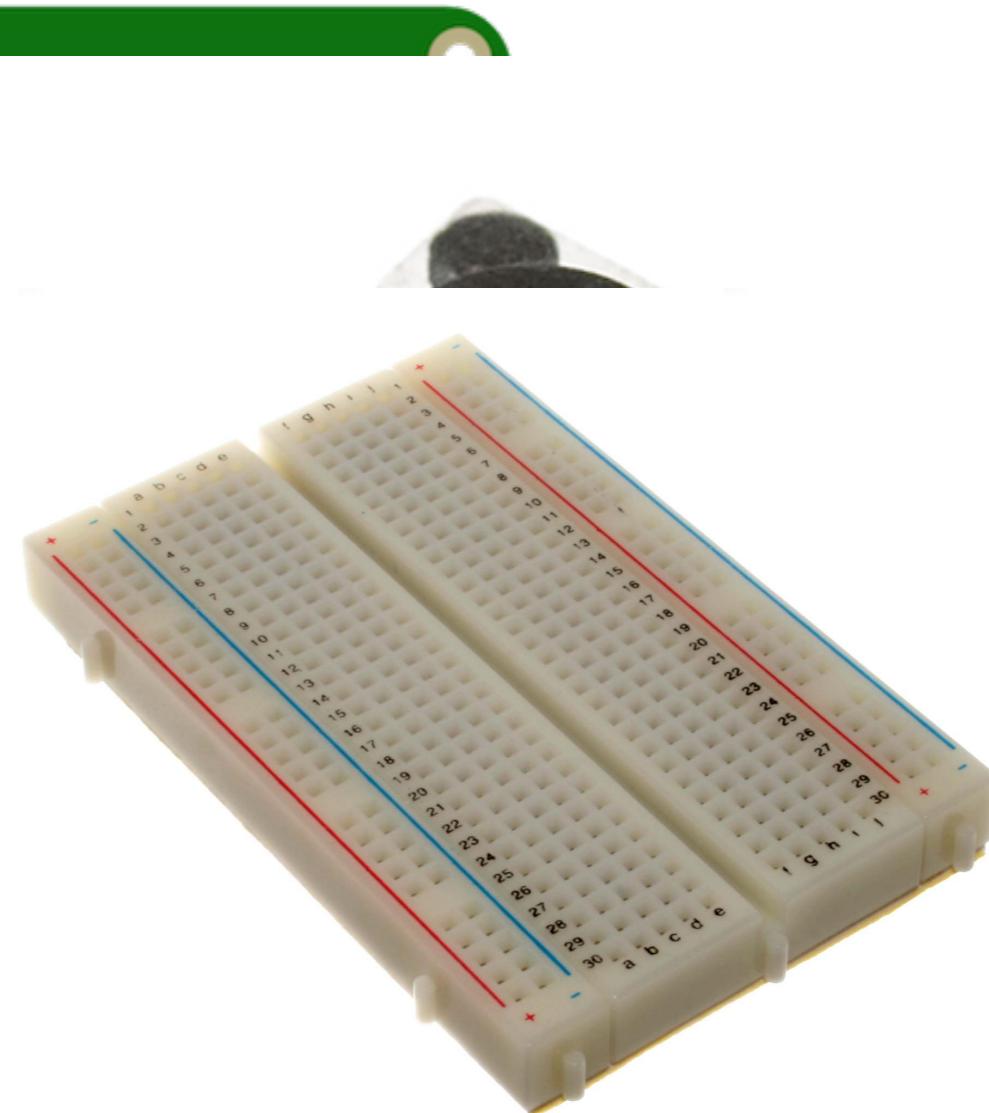
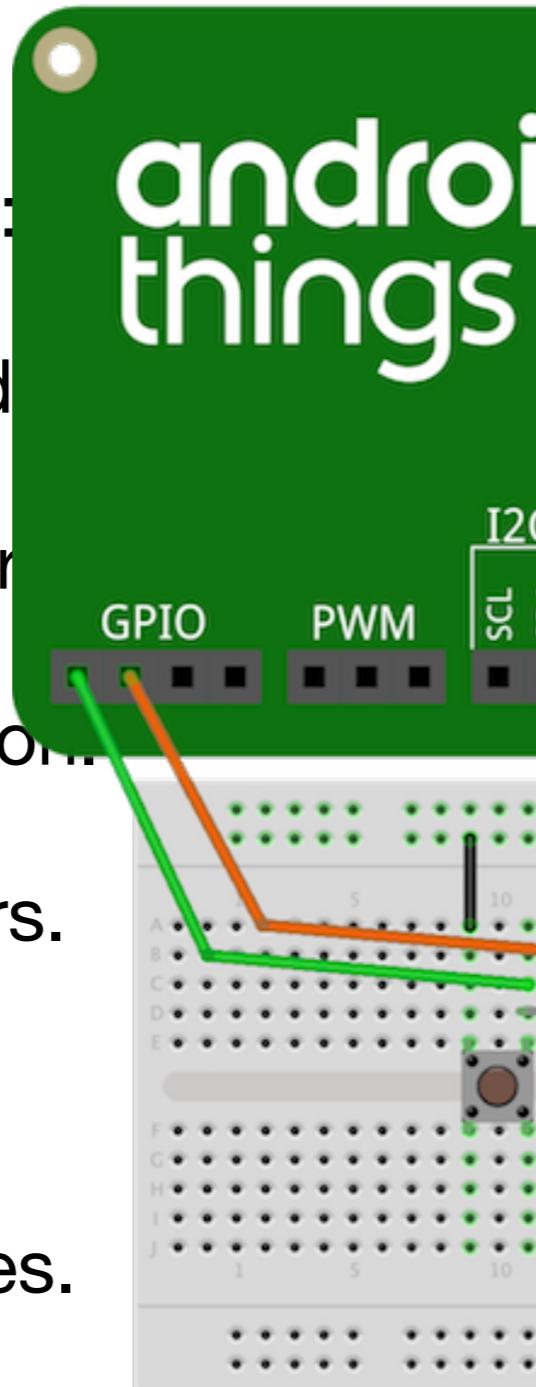
- Manifest file changes.

```
<application>
    <uses-library android:name="com.google.android.things"/>
    <activity android:name=".HomeActivity">
        <!-- Launch activity as default from Android Studio -->
        <intent-filter>
            <action android:name="android.intent.action.MAIN"/>
            <category android:name="android.intent.category.LAUNCHER"/>
        </intent-filter>

        <!-- Launch activity automatically on boot,
            and re-launch if the app terminates. -->
        <intent-filter>
            <action android:name="android.intent.action.MAIN"/>
            <category android:name="android.intent.category.HOME"/>
            <category android:name="android.intent.category.DEFAULT"/>
        </intent-filter>
    </activity>
</application>
```

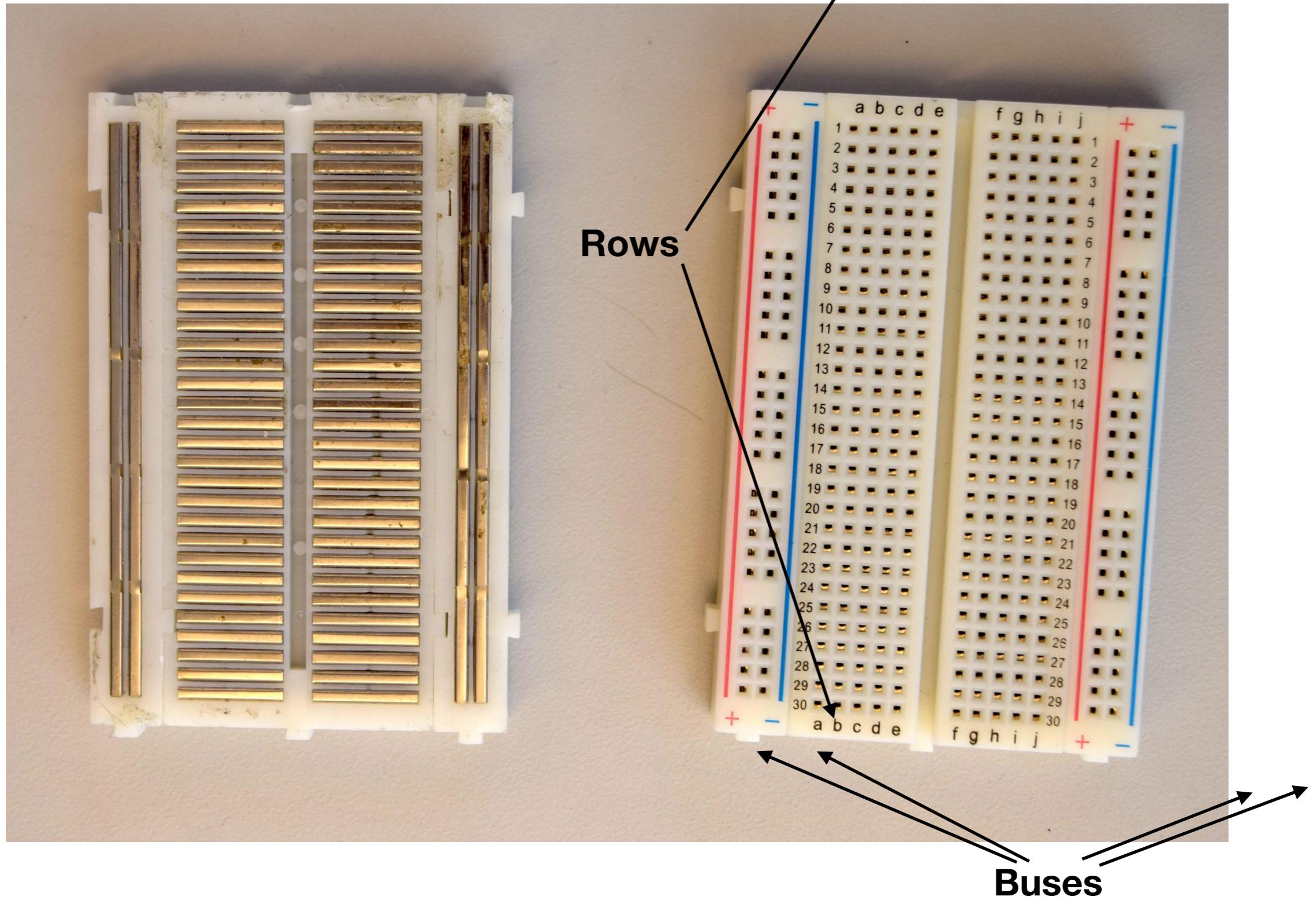
Connect the Hardware

- Requirements:
 - The Android Things Dev Board.
 - A breadboard.
 - A push button.
 - Two resistors.
 - A LED.
 - Jumper wires.



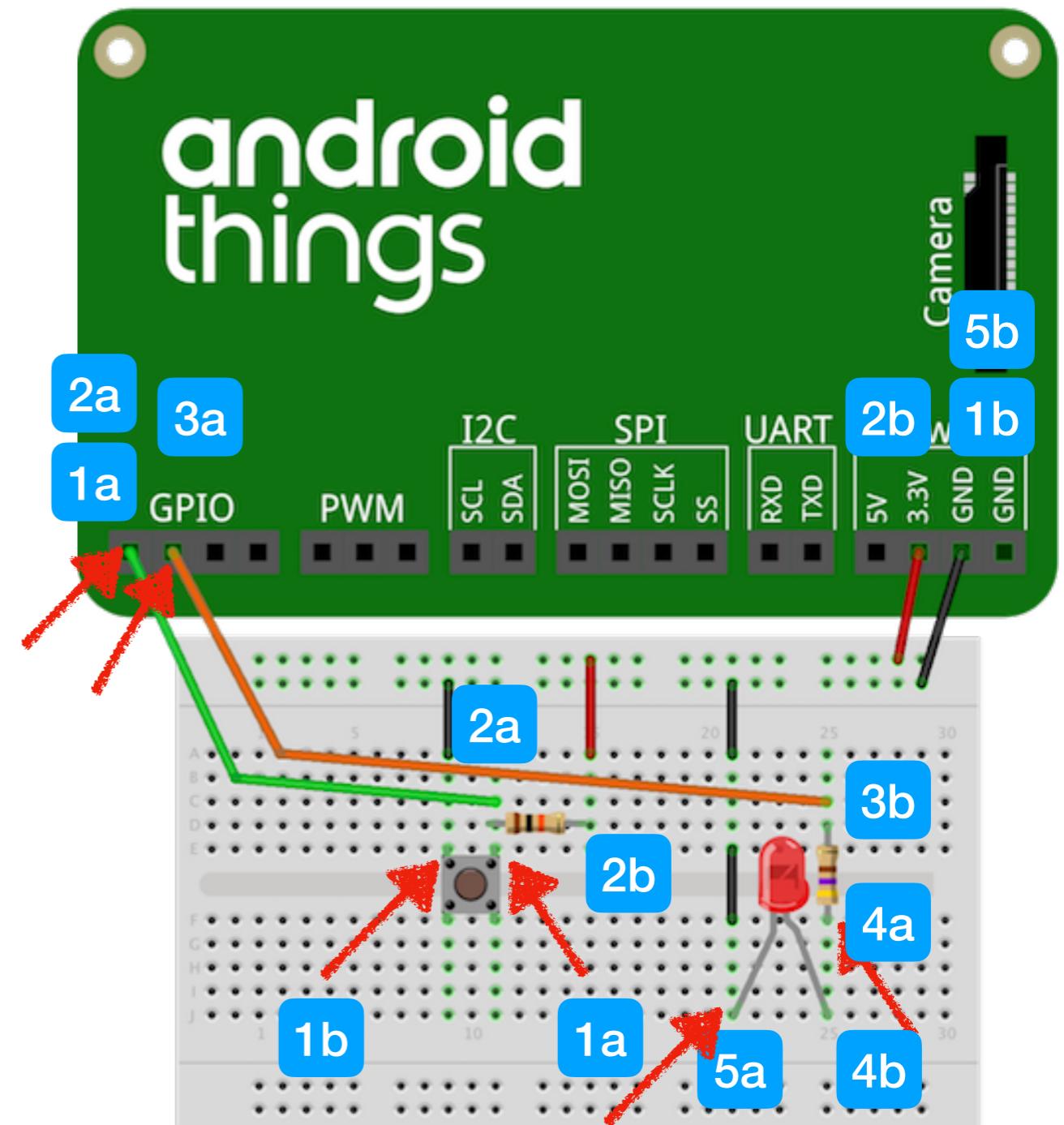
fritzing

The breadboard

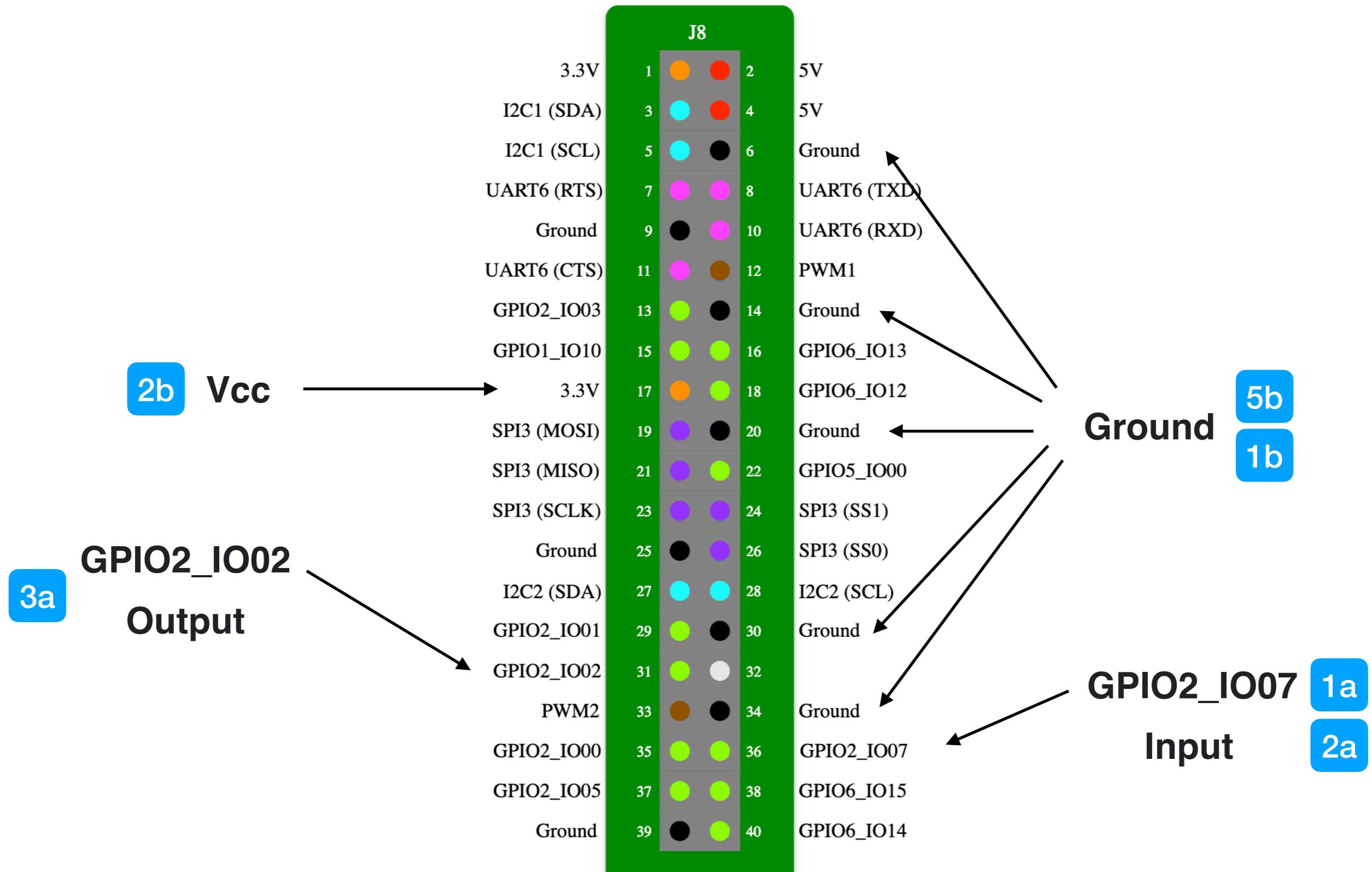


Create the Connections

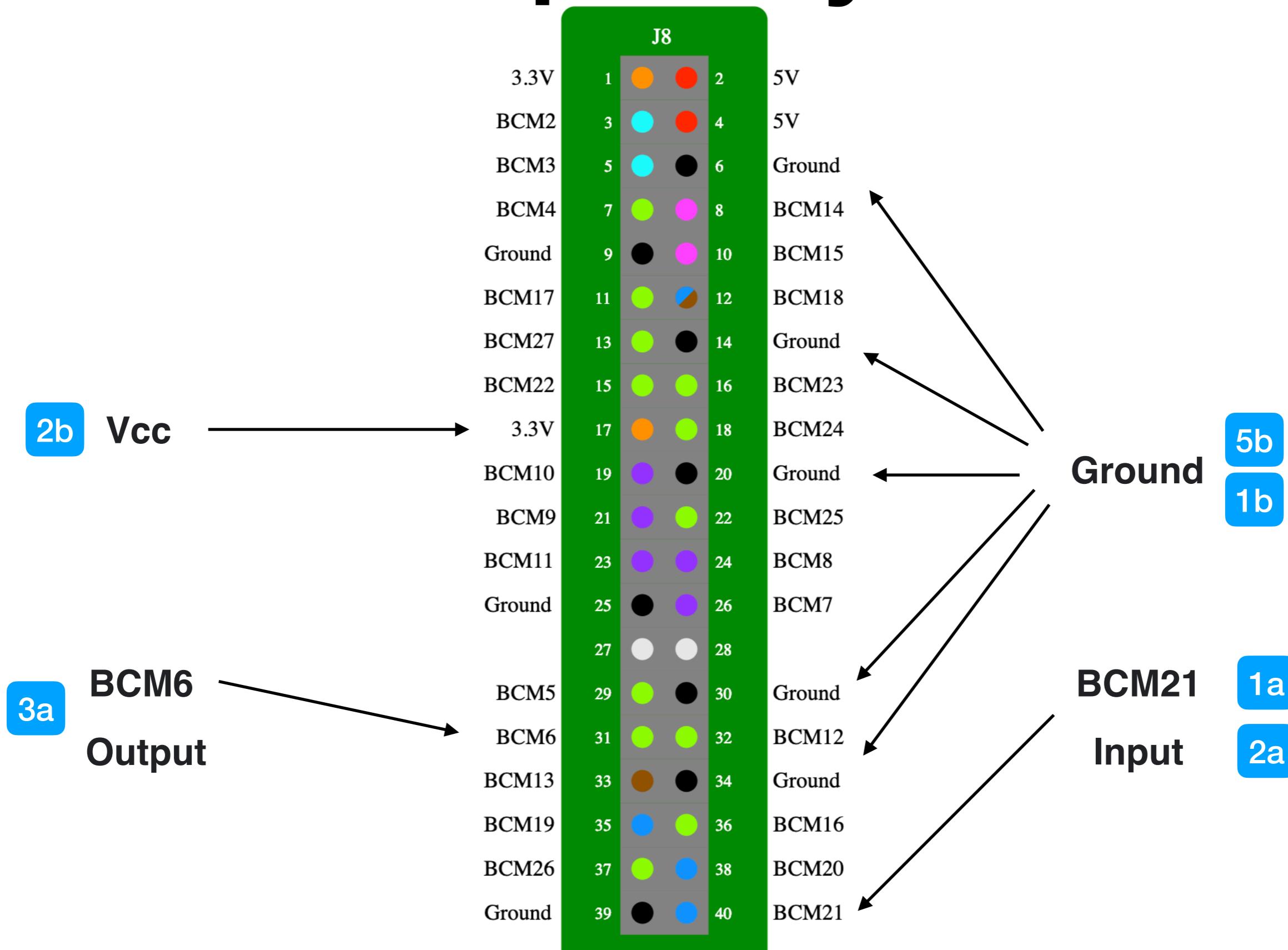
- 1 Connect one side of the button to the chosen GPIO input pin, and the other side to ground.
- 2 Connect the same GPIO input pin to +3.3V through a pull-up resistor.
- 3 Connect the chosen GPIO output pin to one side of a series resistor.
- 4 Connect the other side of the resistor to the anode side (longer lead) of the LED.
- 5 Connect the cathode side (shorter lead) of the LED to ground.



NXP i.MX7D



Raspberry Pi



List available peripherals

```
import com.google.android.things.pio.PeripheralManager  
...  
  
class MainActivity : Activity() {  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
  
        val manager = PeripheralManager.getInstance()  
        logd("Available GPIO: ${manager.gpioList}")  
    }  
}
```

```
// Step 1. Create GPIO connection.  
buttonGpio = manager.openGpio(BUTTON_PIN_NAME)  
// Step 2. Configure as an input.  
buttonGpio.setDirection(Gpio.DIRECTION_IN)  
// Step 3. Enable edge trigger events.  
buttonGpio.setEdgeTriggerType(Gpio.EDGE_FALLING)  
// Step 4. Register an event callback.
```

```
class ButtonActivity : Activity() {  
    private Companion object Companion {  
        const val BUTTON_PIN_NAME = "GPIO2_I007" // GPIO port wired to the button  
    }  
}
```

```
private lateinit var buttonGpio: Gpio  
// Step 4. Register an event callback.  
private val mCallback = GpioCallback {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        Log.i("GPIO changed, button pressed")  
        super.onCreate(savedInstanceState)  
        // Step 5. Return true to keep callback active.  
        true  
    }  
    val manager = PeripheralManager.getInstance()  
} try {  
    // Step 1. Create GPIO connection.  
    override fun onDestroy() {  
        buttonGpio = manager.openGpio(BUTTON_PIN_NAME)  
        super.onDestroy()  
        // Step 2. Configure as an input.  
        buttonGpio.setDirection(Gpio.DIRECTION_IN)  
        // Step 3. Enable edge trigger events.  
        buttonGpio.setEdgeTriggerType(Gpio.EDGE_FALLING)  
    } try {  
        // Step 4. Register an event callback.  
        buttonGpio.registerGpioCallback(mCallback)  
    } catch (e: IOException) {  
        Log.e("EFF8F 8R PeripheralIO API", e)  
    }  
}
```

Handle button events

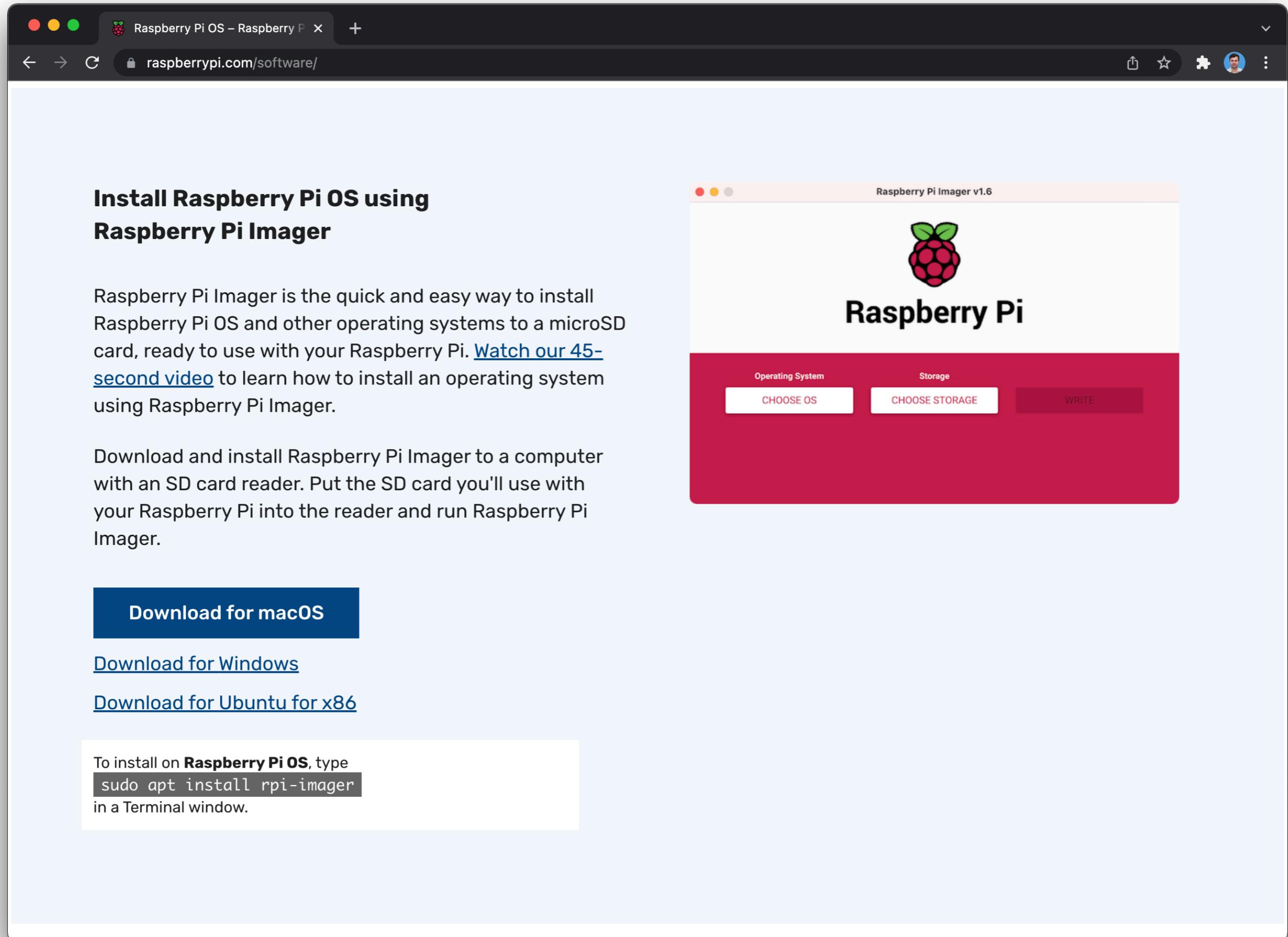
Blink an LED

```
class BlinkActivity : Activity() {
    private companion object {
        const val LED_PIN_NAME = "GPIO2_I002" // GPIO port wired to the LED
        const val INTERVAL_BETWEEN_BLINKS_MS = 1000L
    }
    private val mHandler = Handler()
    private lateinit var ledGpio: Gpio

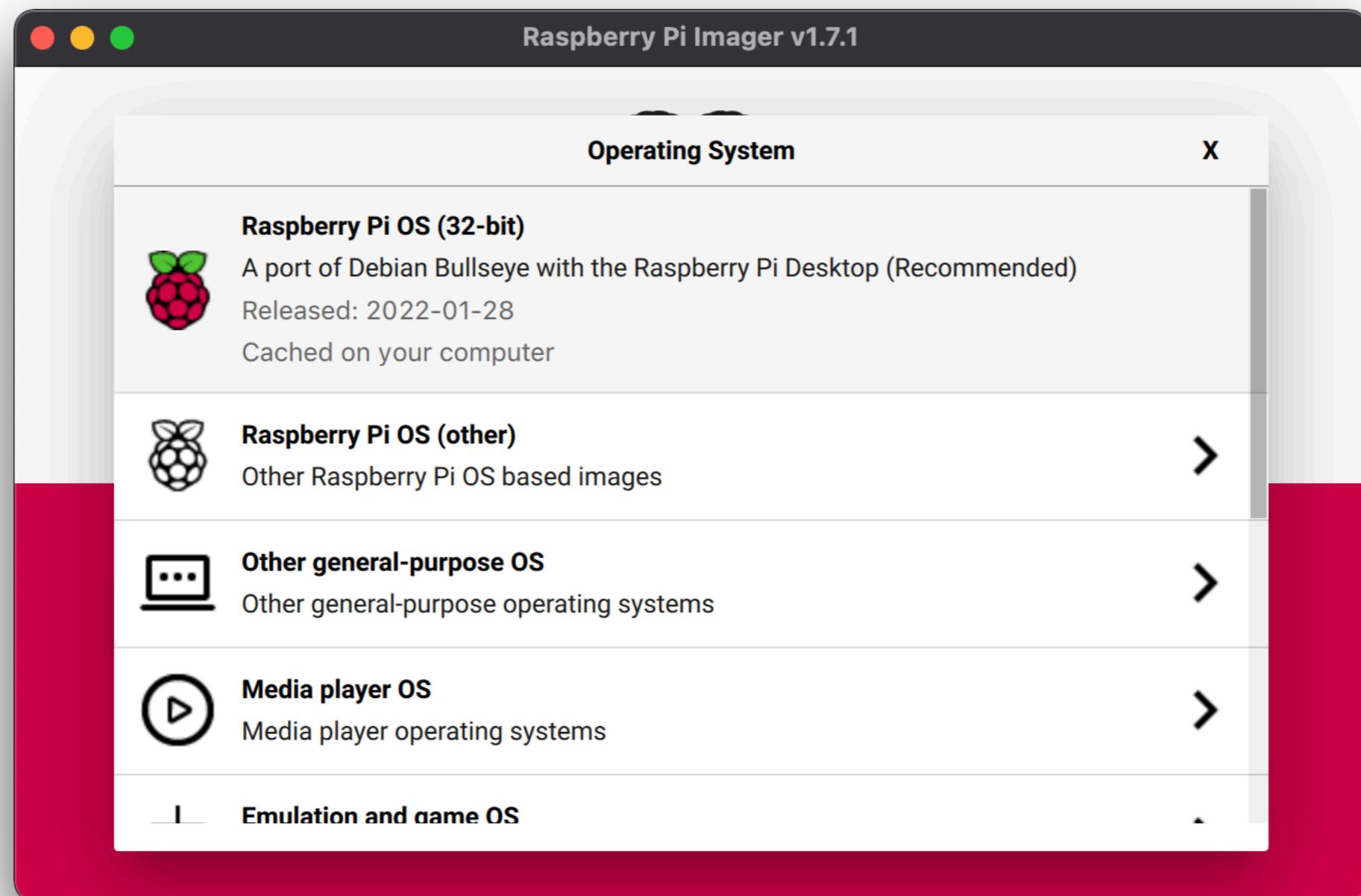
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        // Step 1. Create GPIO connection.
        val manager = PeripheralManager.getInstance()
        try {
            ledGpio = manager.openGpio(LED_PIN_NAME)
            // Step 2. Configure as an output.
            ledGpio.setDirection(Gpio.DIRECTION_OUT_INITIALLY_LOW)

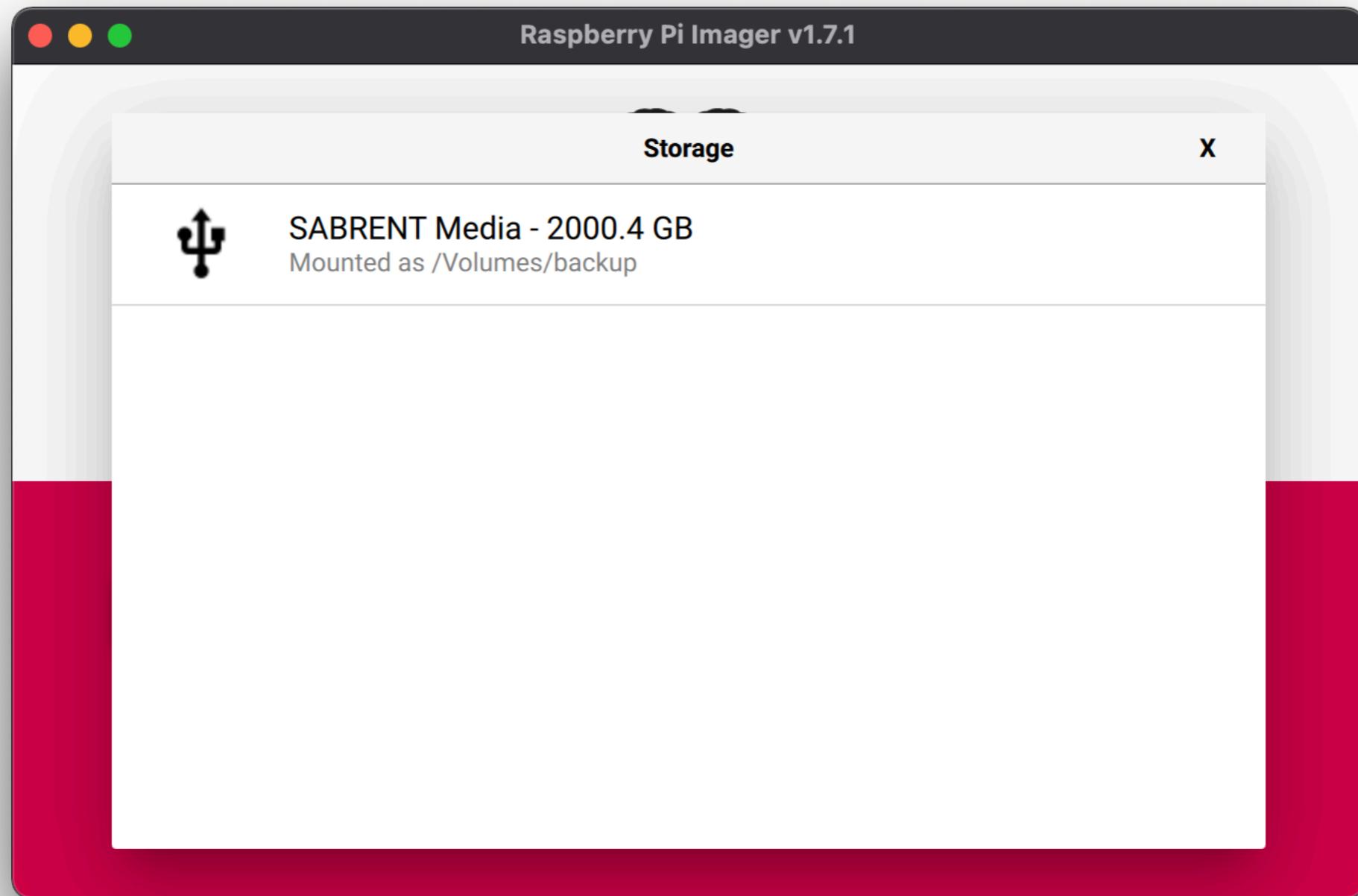
            // Step 4. Repeat using a handler.
            mHandler.post(blinkRunnable)
        } catch (e: IOException) {
            loge("Error on PeripheralIO API", e)
        }
    }
}
```

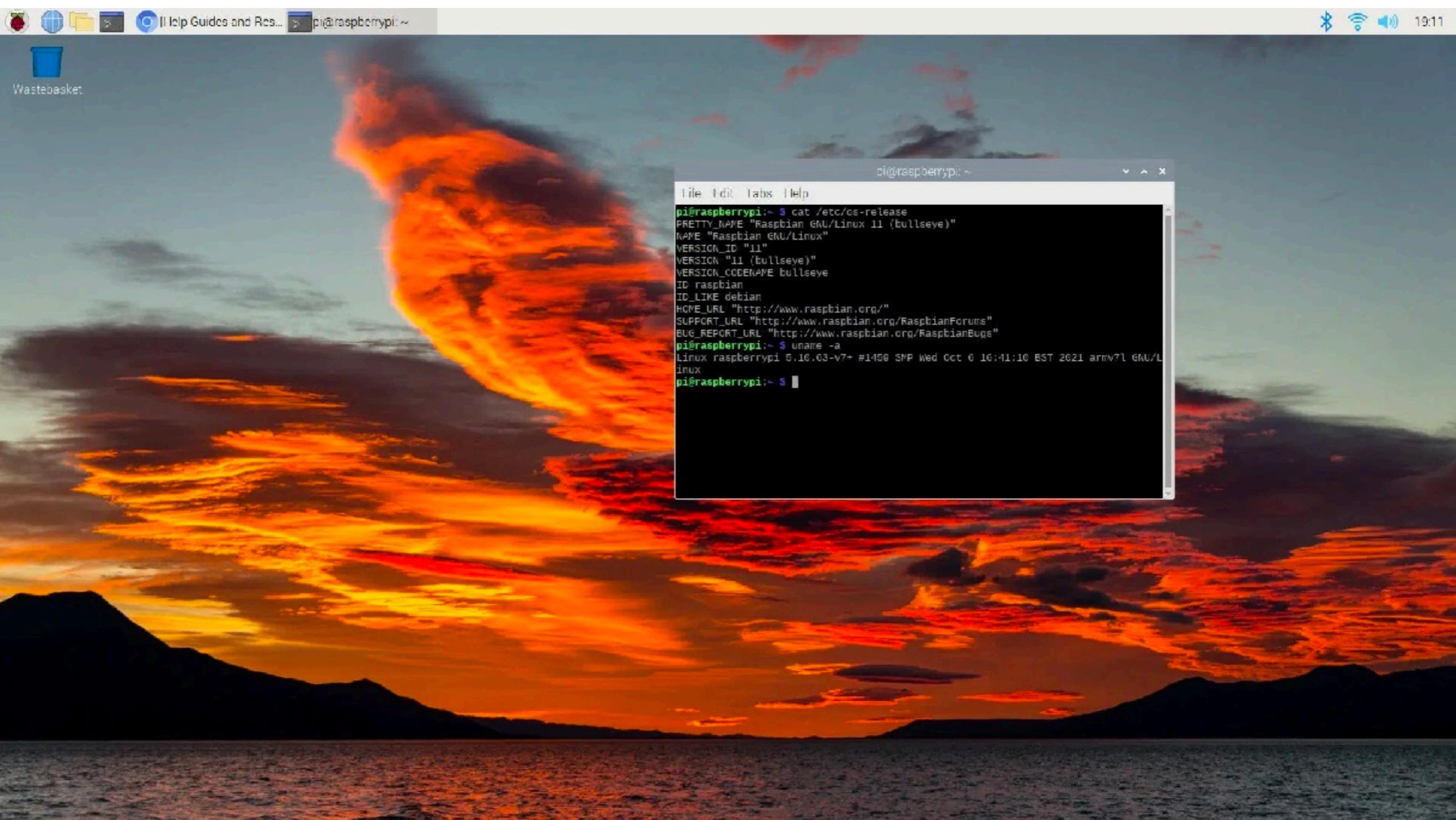






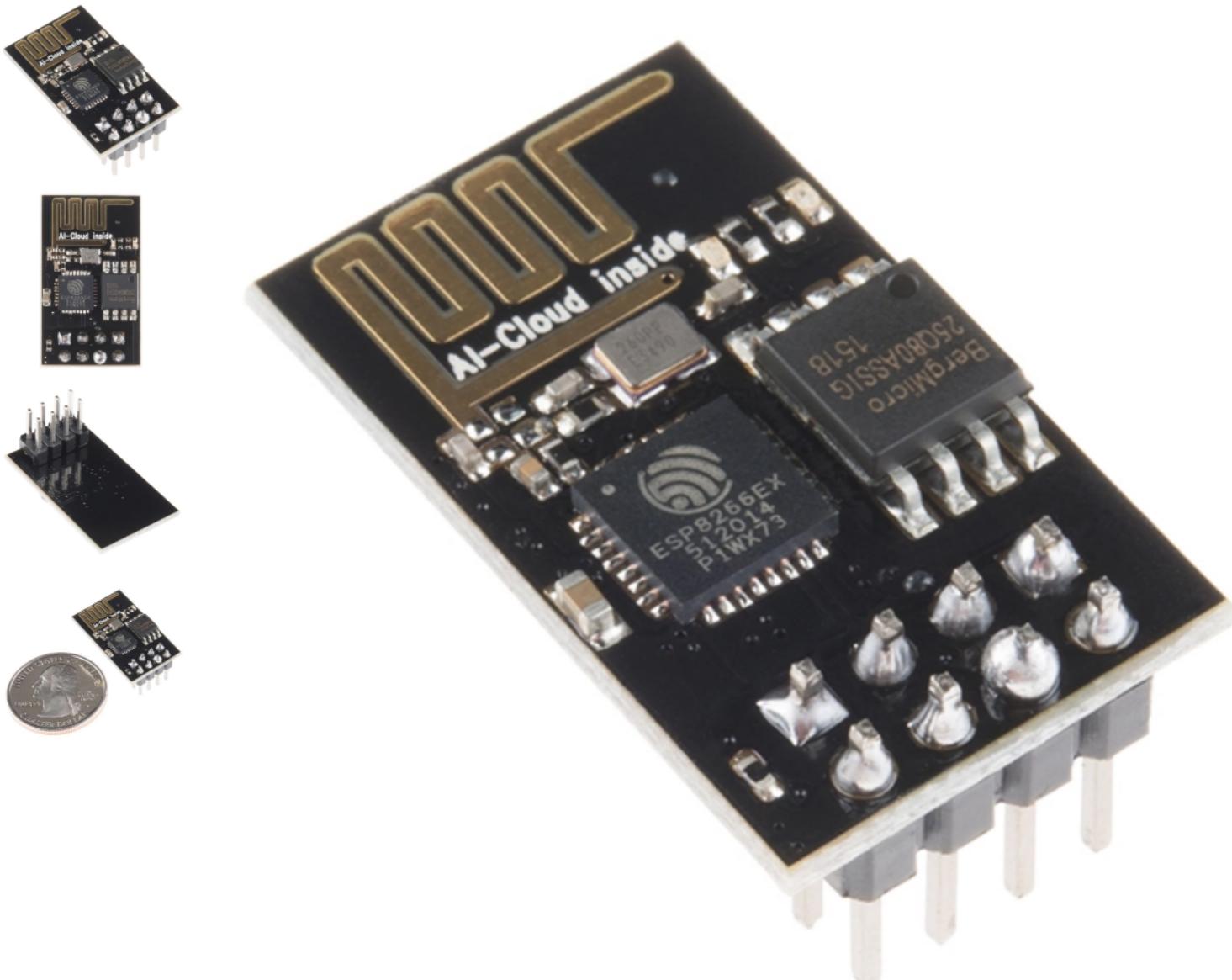






Connect Arduino2Internet

Home / Wireless / [ESP8266 WiFi](#)



ESP8266 WiFi

★★★★★ 12 Review-uri

19,00 Lei

Stoc sku: WIR-46

STOC EPUZAT

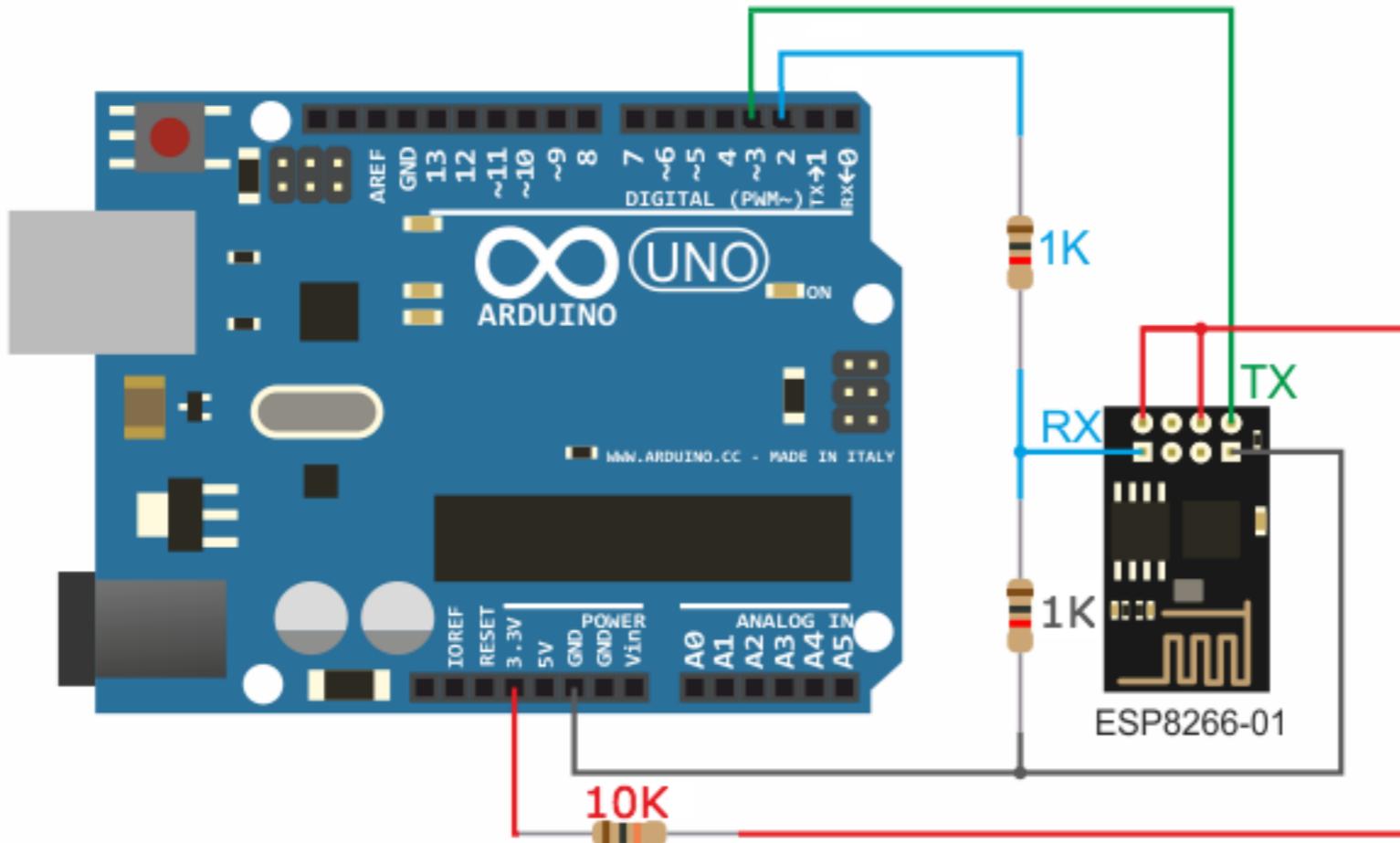
Durata de livrare: 1 - 3 zile lucratoare

ALERTA STOC

Cod Produs: **00003605**

Adauga la Favorite

Build the Circuit



Build the Circuit

ESP8266	Arduino
GND	GND
VIN	3.3v
ENABLE	3.3v
TX	RX
RX	TX

IMPORTANT do not use voltages more than 3.3V!

Talk to Module

Commands	Description	Type
AT+RST	restart module	basic
AT+CWMODE	wifi mode	wifi
AT+CWJAP	join AP	wifi
AT+CWLAP	list AP	wif
AT+CWQAP	quit AP	wifi
AT+CIPSTATUS	get status	TCP/IP
AT+CIPSTART	set up TCP or UDP	TCP/IP
AT+CIPSEND	send data	TCP/IP
AT+CIPCLOSE	close TCP or UDP	TCP/IP
AT+CIFSR	get IP	TCP/IP
AT+CIPMUX	set multiple connections	TCP/IP
AT+CIPSERVER	set as server	TCP/IP

Process Module Data

```
if(wifiSerial.available()>0){

    String message = readWifiSerialMessage();

    if(find(message,"esp8266:")){
        String result = sendToWifi(message.substring(8,message.length()),responseTime,DEBUG);
        if(find(result,"OK"))
            sendData("\n"+result);
        else
            sendData("\nErrRead"); //At command ERROR CODE for Failed Executing statement
    }else
        if(find(message,"HELLO")){
            //receives HELLO from wifi
            sendData("\nHI!"); //arduino says HI
        }else if(find(message,"LEDON")){
            digitalWrite(13,HIGH);
        }else if(find(message,"LEDOFF")){
            digitalWrite(13,LOW);
        }
        else{
            sendData("\nErrRead");//Command ERROR CODE for UNABLE TO READ
        }
    }
}
```

RPPAL - Raspberry Pi Peripheral Access Library

 CI passing  crates.io v0.17.1  license MIT  rustc v1.60.0

RPPAL provides access to the Raspberry Pi's GPIO, I2C, PWM, SPI and UART peripherals through a user-friendly interface. In addition to peripheral access, RPPAL also offers support for USB to serial adapters.

The library can be used in conjunction with a variety of platform-agnostic drivers through its `embedded-hal` trait implementations. Both `embedded-hal` v0.2.7 and v1 are supported.

RPPAL requires Raspberry Pi OS or any similar, recent, Linux distribution. Both GNU and musl `libc` targets are supported. RPPAL is compatible with the Raspberry Pi A, A+, B, B+, 2B, 3A+, 3B, 3B+, 4B, 5, CM, CM 3, CM 3+, CM 4, 400, Zero, Zero W and Zero 2 W. Backwards compatibility for minor revisions isn't guaranteed until v1.

This library is under development on the [master branch](#) of the repository on GitHub. If you're looking for the `README.md` or the `examples` directory for the latest release or any of the earlier releases, visit [crates.io](#), download an archived release from the GitHub [releases](#) page, or clone and checkout the relevant release tag.

Operating System development tutorials in



```
#[no_mangle]
unsafe fn kernel_init() -> ! {
    use interface::mm::MMU;

    arch::enable_exception_handling();

    if let Err(string) = arch::mmu().init() {
        panic!("MMU: {}", string);
    }

    for i in bsp::device_drivers().iter() {
        if let Err(() ) = i.init() {
            panic!("Error loading driver: {}", i.compatible());
        }
    }
}
```

Minipush 1.0

[MP] 🚧 Waiting for /dev/ttyUSB0
[MP] ✓ Connected

Raspberry Pi 3

[ML] Requesting binary
[MP] ► Pushing 64 KiB ===== 100% 32 KiB/s Time: 00:00:02
[ML] Loaded! Executing the payload now

[2.909178] Booting on: Raspberry Pi 3
[2.910263] MMU online. Special regions:
[2.912174] 0x00080000 - 0x0008ffff | 64 KiB | C RO PX | Kernel code and RO data
[2.916257] 0x3f000000 - 0x3fffffff | 16 MiB | Dev RW PNX | Device MMIO
[2.919820] Current privilege level: EL1
[2.921731] Exception handling state:
[2.923512] Debug: Masked
[2.925075] SError: Masked
[2.926639] IRQ: Masked
[2.928203] FIQ: Masked
[2.929766] Architectural timer resolution: 52 ns
[2.932069] Drivers loaded:
[2.933415] 1. GPIO
[2.934675] 2. PL011Uart
[2.936152] Echoing input now

Hello OS dev!

<https://github.com/rust-embedded/rust-raspberrypi-OS-tutorials>

Pi-hole®

The screenshot shows the official Pi-hole website at <https://pi-hole.net>. The header features the Pi-hole logo and navigation links for GITHUB, COMMUNITY, ABOUT, BLOG, and SHOP. Below the header, the main title "Pi-hole® Network-wide Ad Blocking" is displayed, along with "INSTALL", "SPONSOR US", and "DONATE" buttons. The main content area is divided into two sections: "1. Install a supported operating system" and "2. Install Pi-hole". The first section includes a description of supported systems (Docker, Fedora, OpenShift, Docker Compose, Ubuntu, Debian, and Raspberry Pi) and buttons for "DOCKER INSTALL" and "SUPPORTED OPERATING SYSTEMS". The second section features a large green leaf icon.

1. Install a supported operating system

You can run Pi-hole in a container, or deploy it directly to a supported operating system via our automated installer.

[DOCKER INSTALL](#)

[SUPPORTED OPERATING SYSTEMS](#)

2. Install Pi-hole

<https://pi-hole.net/>

Pi-hole®

Pi-hole

hostname: Test-PiZeroW Pi-hole

Status
Active Temp: 42.2 °C
Load: 0.22 0.18 0.19
Memory usage: 18.8%

MAIN NAVIGATION

- Dashboard
- Query Log
- Long term data
- Whitelist
- Blacklist
- Group Management
- Disable
- Tools
- Network
- Settings
- Local DNS Records
- Donate
- Help

Total queries (5 clients) **3,416**

Queries Blocked **491**

Percent Blocked **14.4%**

Domains on Blocklist **82,309**

Total queries over last 24 hours

Client activity over last 24 hours

https://pi-hole.net/

Built-in DHCP Server

The screenshot shows the Pi-hole web interface on a Raspberry Pi device. The top navigation bar includes the Pi-hole logo, the host name "raspberrypi", and the Pi-hole logo again.

The main menu on the left is titled "Pi-hole" and includes the following items:

- Status (Active, Temp: 40.8 °C, Load: 0 0 0, Memory usage: 59.4 %)
- MAIN NAVIGATION
 - Dashboard
 - Query Log
 - Long term data
 - Whitelist (selected)
 - Blacklist
 - Disable
- Tools
- Settings (selected)
- Logout
- Donate
- Help

The "DHCP" tab is selected in the top navigation bar. The main content area is divided into two main sections:

- DHCP Settings**:
 - A checkbox labeled "DHCP server enabled" is unchecked.
 - A "Range of IP addresses to hand out" section with "From" set to 192.168.1.201 and "To" set to 192.168.1.251.
 - A "Router (gateway) IP address" section with "Router" set to 192.168.1.1.
- Advanced DHCP settings**:
 - "Pi-hole domain name": Domain lan
 - "DHCP lease time": Lease time in hours 24
 - Hint: 0 = infinite, 24 = one day, 168 = one week, 744 = one month, 8760 = one year
 - A checkbox for "Enable IPv6 support (SLAAC + RA)" is unchecked.

Below these sections is the "DHCP leases" section, which displays the following information:

- Currently active DHCP leases**: No data available in table.
- Showing 0 to 0 of 0 entries
- Search: [empty input field]

Finally, the "Static DHCP leases configuration" section is shown:

- A table with columns: MAC address, IP address, Hostname.
- No data available in table.
- Showing 0 to 0 of 0 entries
- Specifying the MAC address is mandatory and only one entry per MAC address is allowed. If the IP address is omitted and a host name is given, the IP address will still be generated dynamically and the specified host name will be used. If the host name is omitted, only a static lease will be added.
- A green "+" button for adding new entries.

<https://pi-hole.net/>

Query Log

Pi-hole

raspberrypi Pi-hole

Status
● Active Temp: 41.3 °C
● Load: 0 0 0
● Memory usage: 59.4 %

MAIN NAVIGATION

- Dashboard
- Query Log
- Long term data
- Whitelist
- Blacklist
- Disable
- Tools
- Settings
- Logout
- Donate
- Help

Recent Queries (showing up to 100 queries), [show all](#)

Show 10 entries

Search:

Previous 1 2 3 4 5 ... 10 Next

Time	Type	Domain	Client	Status	Reply	Action
2018-12-19 17:49:46	A	api-global.netflix.com	192.168.1.131	OK (forwarded)	CNAME (25.0ms)	Blacklist
2018-12-19 17:49:10	A	api-global.netflix.com	192.168.1.131	OK (forwarded)	CNAME (26.1ms)	Blacklist
2018-12-19 17:48:48	A	api-global.netflix.com	192.168.1.131	OK (forwarded)	CNAME (31.9ms)	Blacklist
2018-12-19 17:48:25	A	device-metrics-us.amazon.com	echoplus	Pi-holed	- (0.9ms)	Whitelist
2018-12-19 17:48:19	A	checkip.synology.com	holobuffer	OK (forwarded)	CNAME (31.1ms)	Blacklist
2018-12-19 17:47:58	A	canireachthe.net	192.168.1.132	OK (forwarded)	IP	Blacklist
2018-12-19 17:47:53	A	logsink.devices.nest.com	nest	OK (forwarded)	IP (21.3ms)	Blacklist
2018-12-19 17:47:51	A	api-global.netflix.com	192.168.1.131	OK (forwarded)	CNAME (23.3ms)	Blacklist
2018-12-19 17:47:24	A	api-global.netflix.com	192.168.1.131	OK (forwarded)	CNAME (24.6ms)	Blacklist
2018-12-19 17:47:22	A	d3p8zr0ffa9t17.cloudfront.net	echoplus	OK (forwarded)	IP	Blacklist

Showing 1 to 10 of 100 entries

Apply filtering on click on Type, Domain, and Clients

Previous 1 2 3 4 5 ... 10 Next

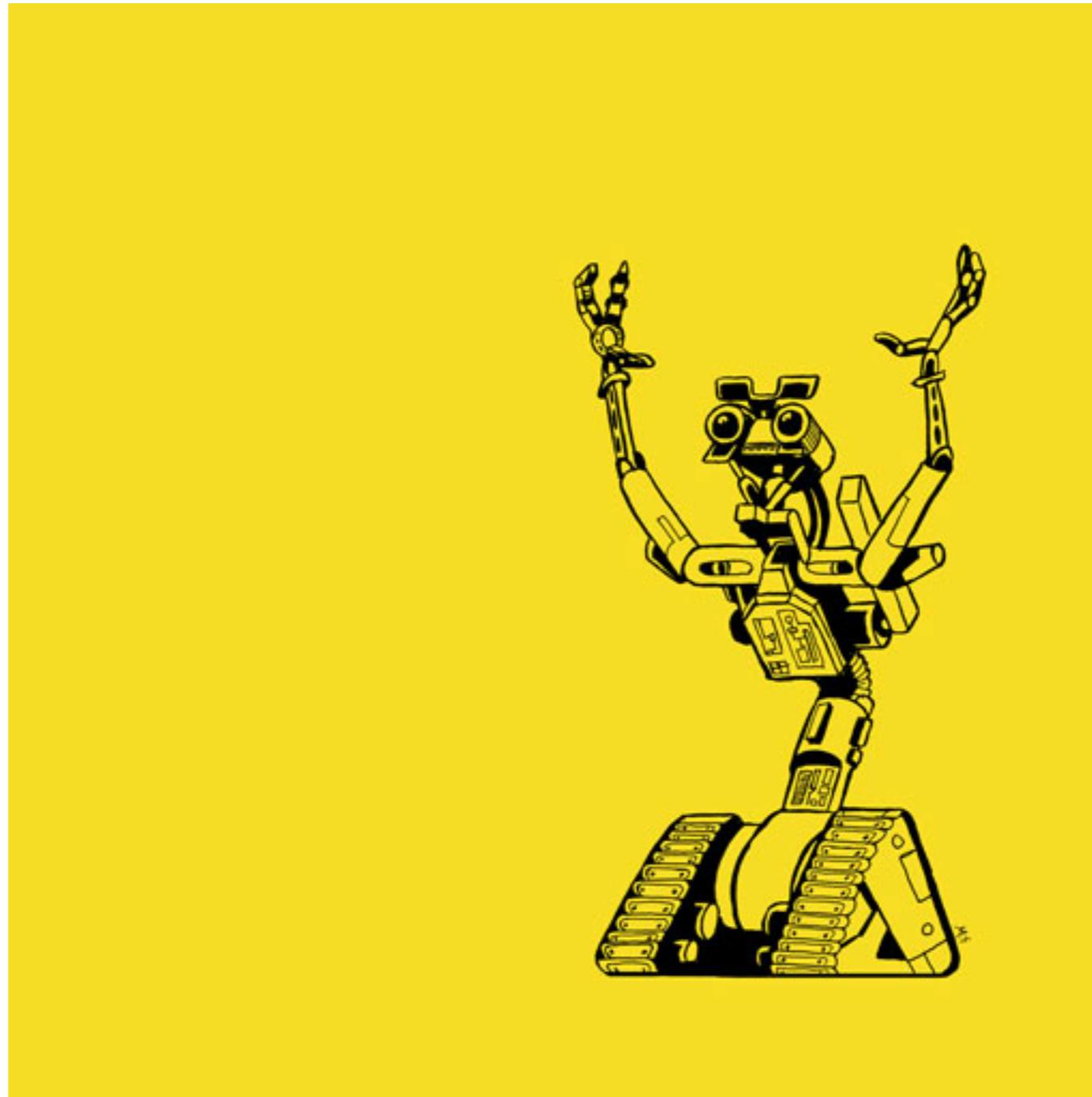
<https://pi-hole.net/>

Build your own weather station



<https://projects.raspberrypi.org/en/projects/build-your-own-weather-station>

The JavaScript Robotics Programming



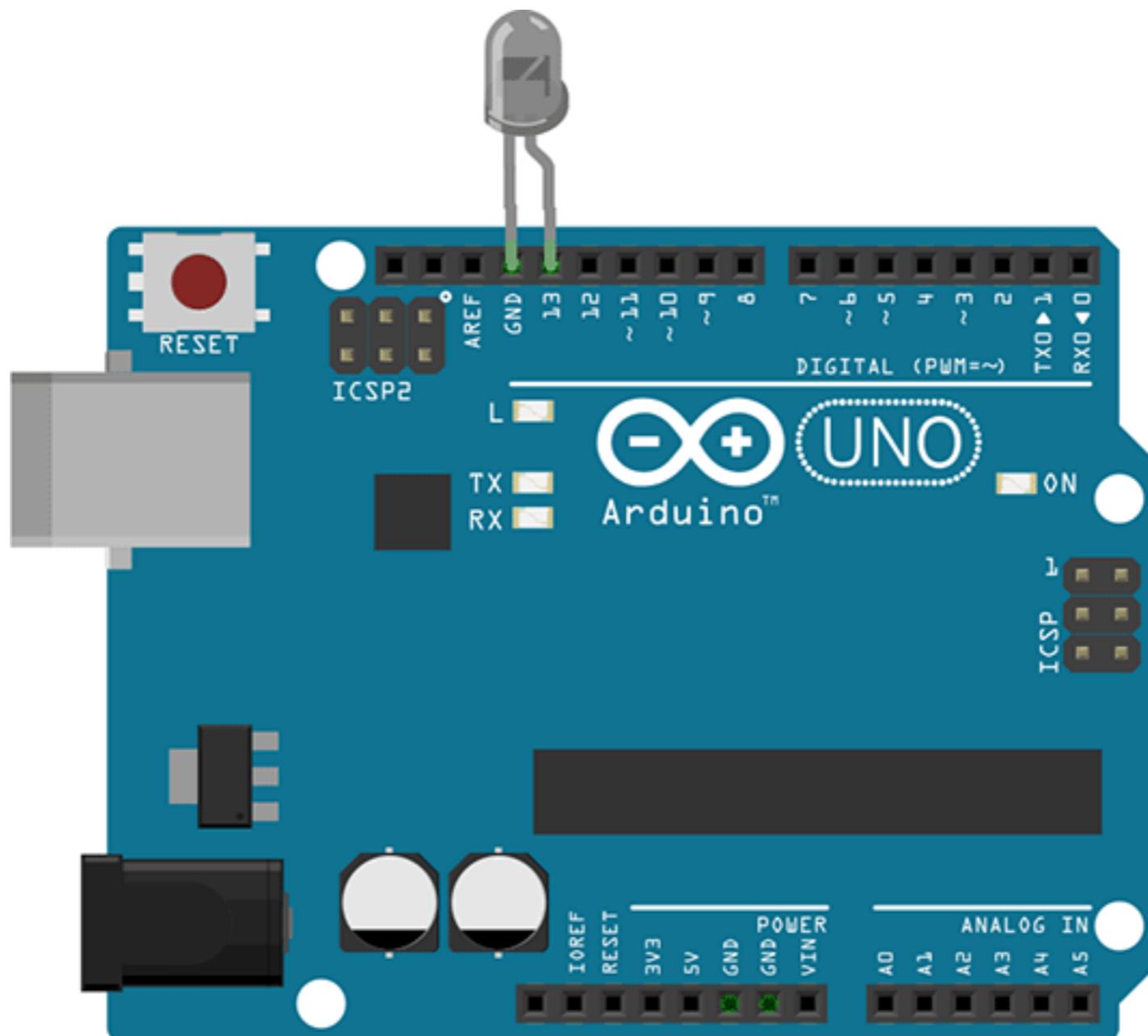
<https://github.com/rwaldron/johnny-five>

The JavaScript Robotics Programming

```
const { Board, Led } = require("johnny-five");
const board = new Board();

board.on("ready", () => {
  // Create an Led on pin 13
  const led = new Led(13);
  // Blink every half second
  led.blink(500);
});
```

The JavaScript Robotics Programming



<https://johnny-five.io/examples/>

The JavaScript Robotics Programming

```
const Raspi = require('raspi-io').RaspiIO;
const five = require('johnny-five');
const board = new five.Board({
  io: new Raspi()
});

board.on('ready', () => {

  // Create an Led on pin 7 (GPIO4) on P1 and strobe it on/off
  // Optionally set the speed; defaults to 100ms
  (new five.Led('P1-7')).strobe();

});
```

Lecture outcomes

- Connect the hardware components.
- Send and receive data.

