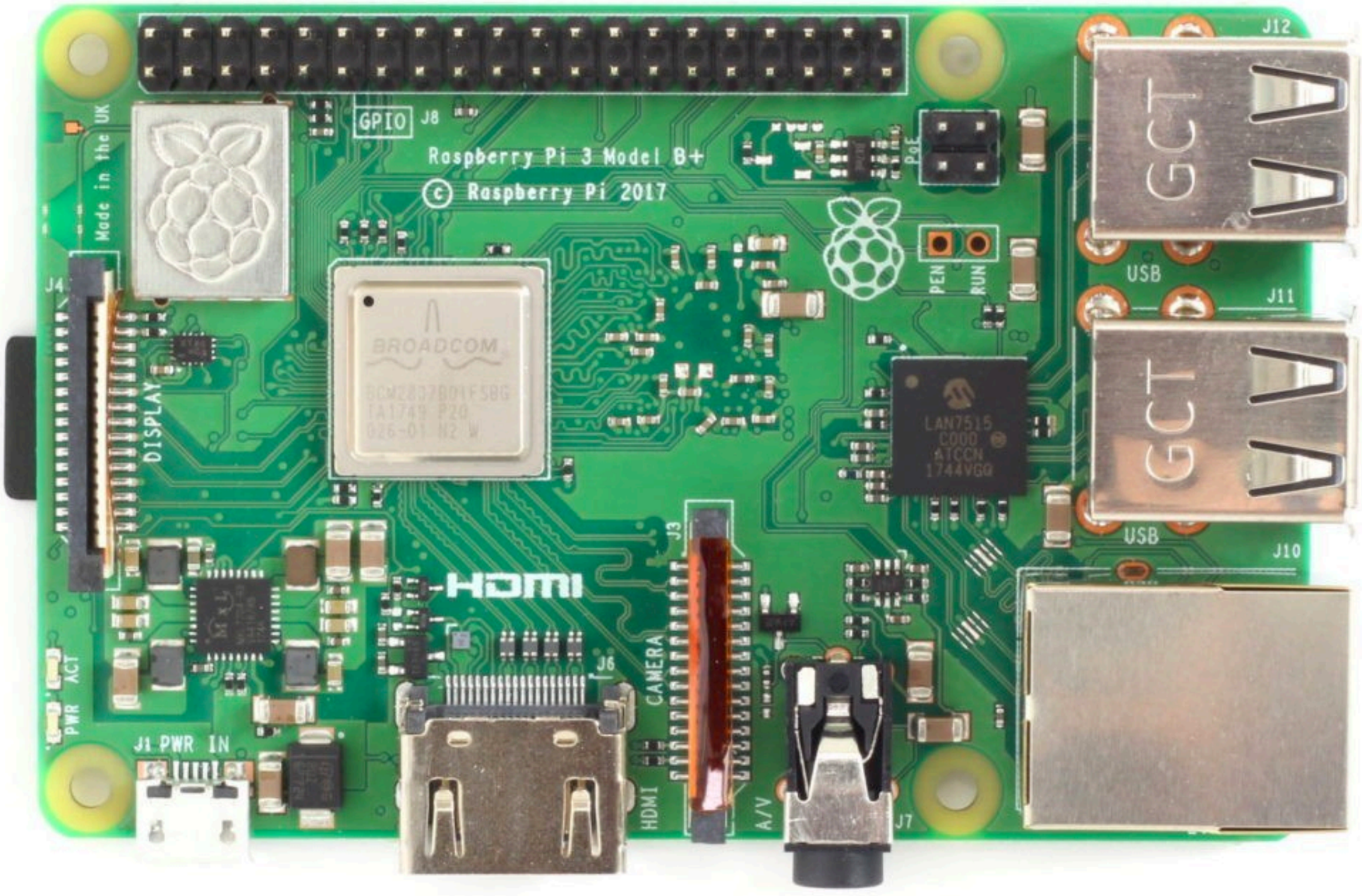
Lecture #5

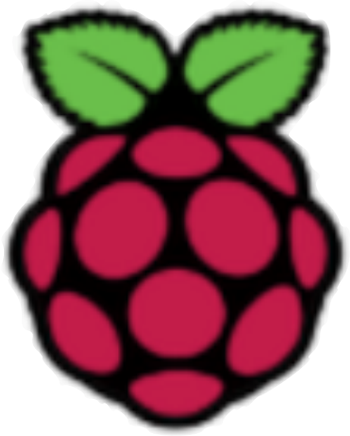
Raspberry Pi

Android Things 2023

Options


GPIO J8
Raspberry Pi 3 Model B+
© Raspberry Pi 2017
Made in the UK
J4
DISPLAY
BROADCOM
LAN7515
C000
ATCCN
1744VGQ
PoE
PEN
RUN
J12
GCT
USB
J11
USB
J10
J3
CAMERA
HDMI
J6
A/V
J7
PWR ACT
J1 PWR IN

Options



Raspberry Pi

Raspberry Pi Hardware

Schematics and Mechanical Drawings

Edit this on GitHub

Schematics for the various Raspberry Pi board versions:

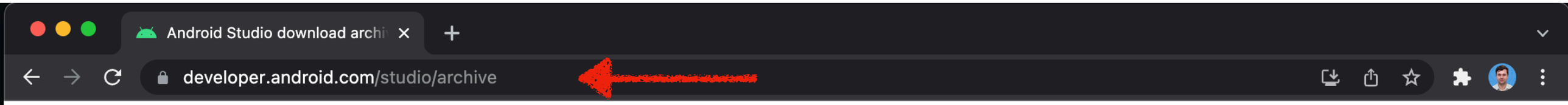
Raspberry Pi 4 Model B

- 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



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 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
517305de6a9558a7075201dffc2483dc9cc46e31632d781172bd8385c0f2c526 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
d816331e1be453d7964c6a6a92e17ba4b22e5dcb505553dd51515e94f86fd95c android-studio-ide-202.7486908-mac.zip
733b04cb66e3ff7f03766aa0222ebd1fef7a63e6340665aa91f0d62e724feca3 android-studio-ide-202.7486908-linux.tar.gz

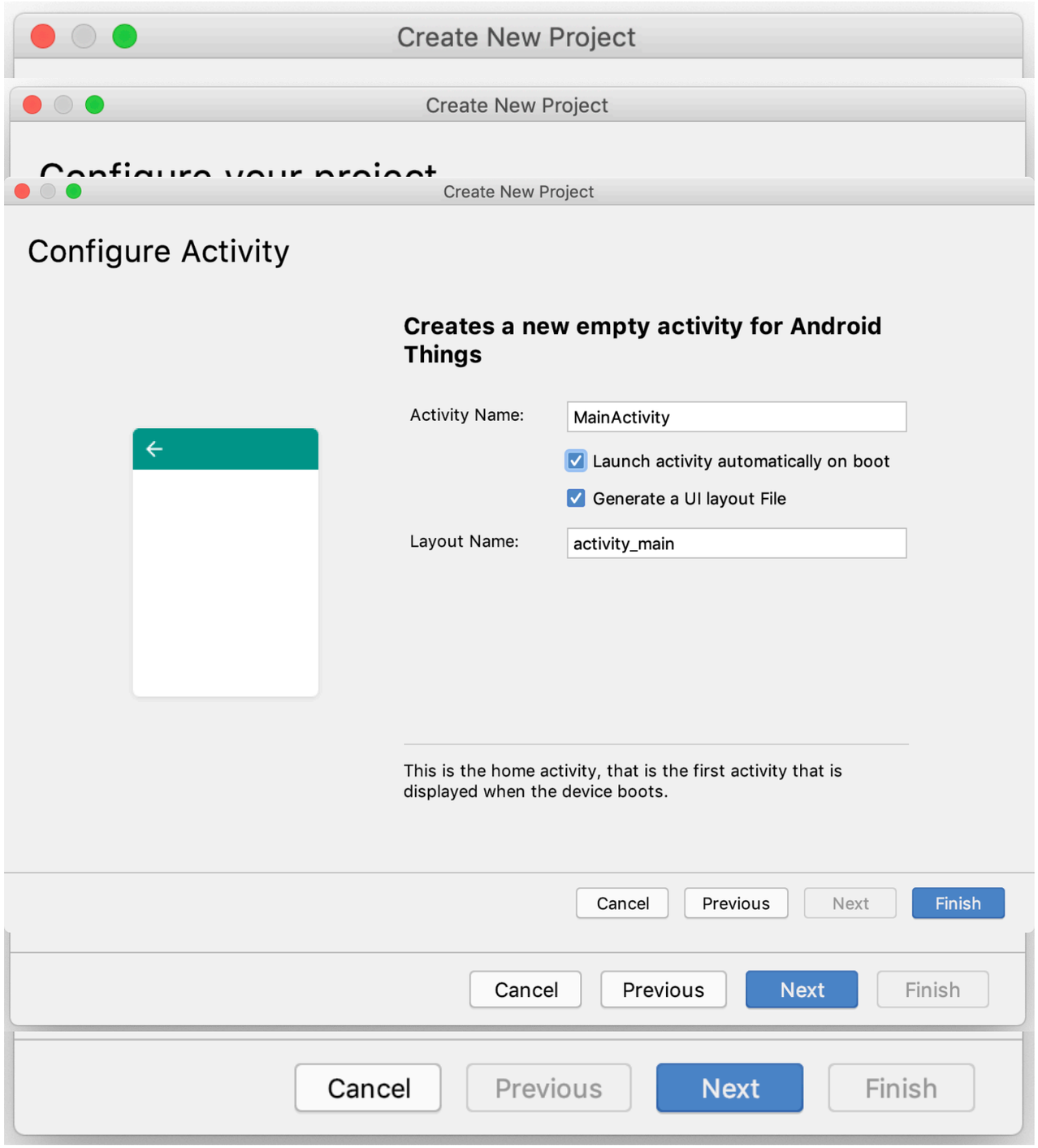
Project Sample

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

ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.0 Ice Cream Sandwich	15	
4.1 Jelly Bean	16	99.6%
4.2 Jelly Bean	17	98.1%
4.3 Jelly Bean	18	95.9%
4.4 KitKat	19	95.3%
5.0 Lollipop	21	85.0%
5.1 Lollipop	22	80.2%
6.0 Marshmallow	23	62.6%
7.0 Nougat	24	37.1% Minimum
7.1 Nougat	25	14.2%
8.0 Oreo	26	6.0%
8.1 Oreo	27	1.1% Recommended
9.0 Pie	28	

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

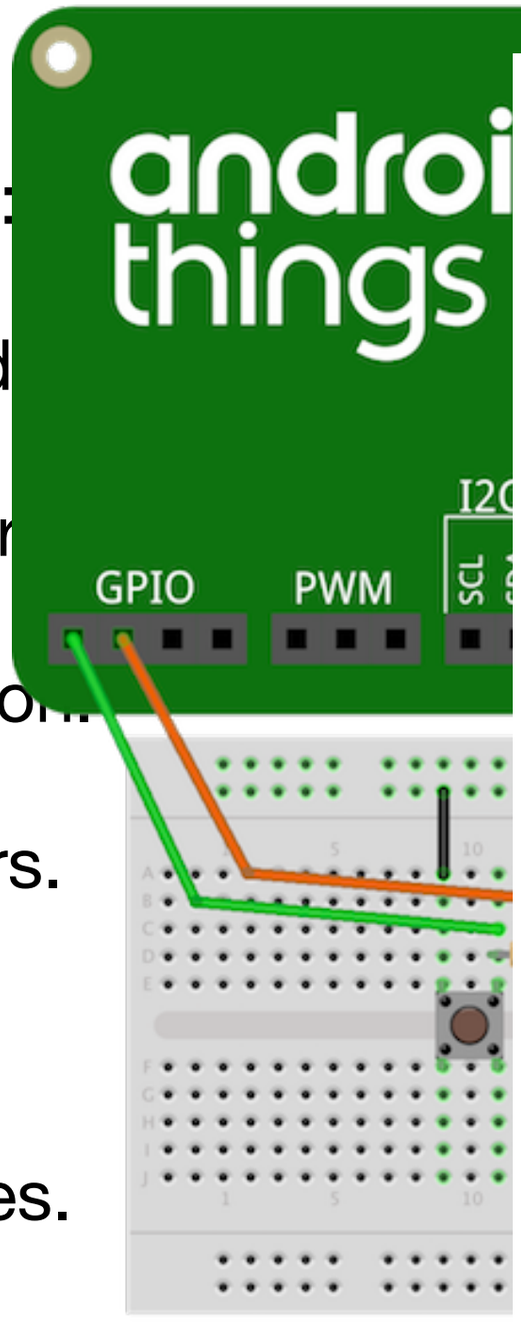
- Manifest file changes.

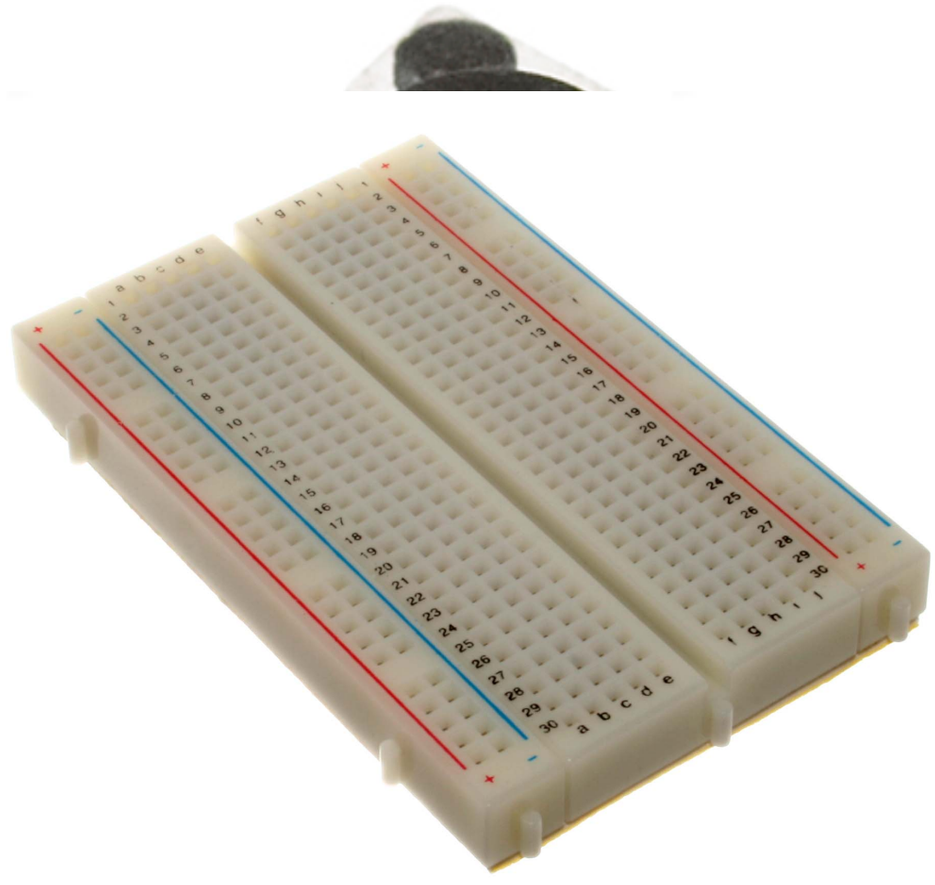
```
<application>
  <uses-library android:name="com.google.android.things"/>
  <activity android:name=".HomeActivity">
    <!-- Launch activity as default from Android Studio -->
    <intent-filter>
      <action android:name="android.intent.action.MAIN"/>
      <category android:name="android.intent.category.LAUNCHER"/>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

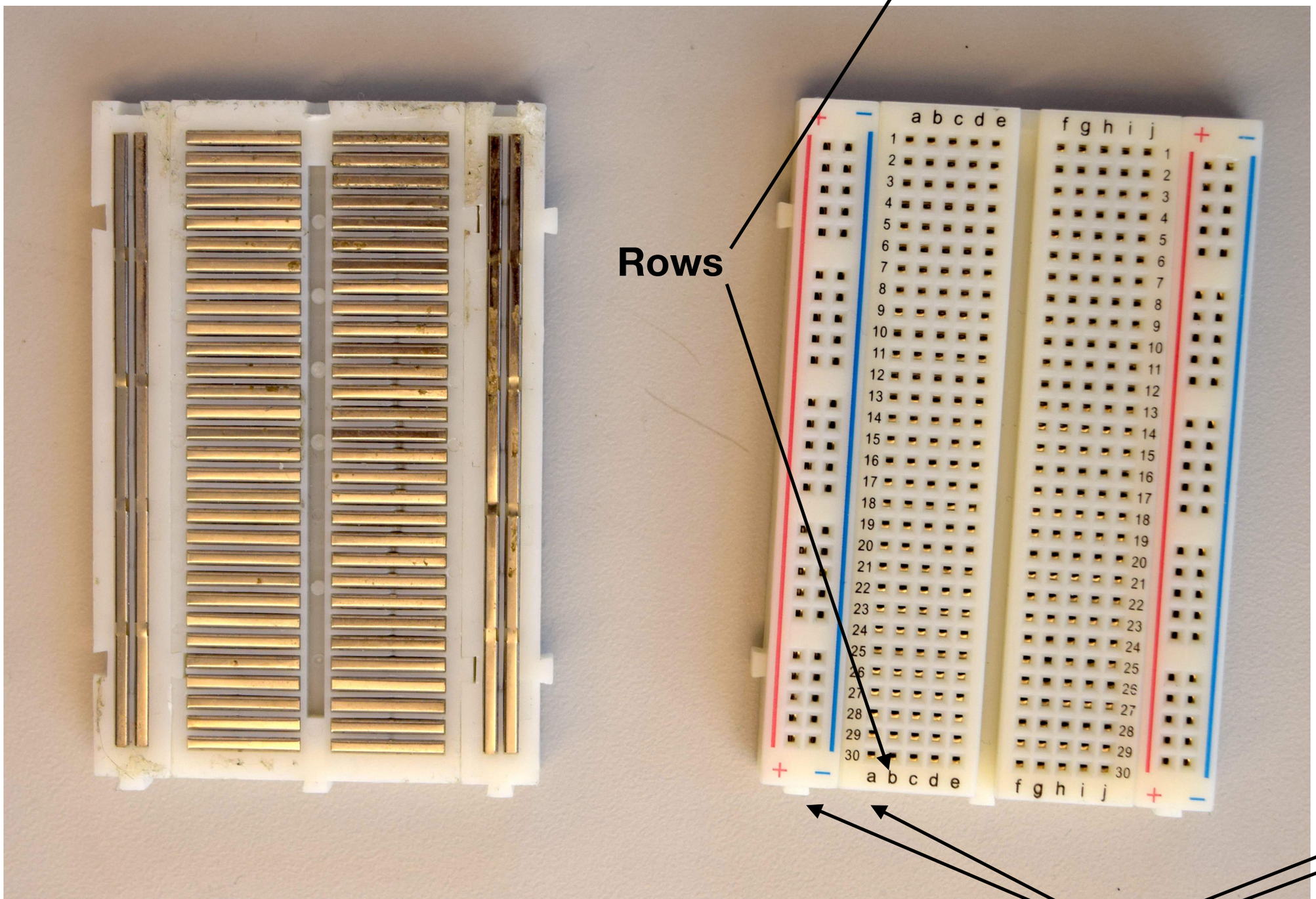
- Requirements:
  - The Android
  - A breadboar
  - A push button.
  - Two resistors.
  - A LED.
  - Jumper wires.





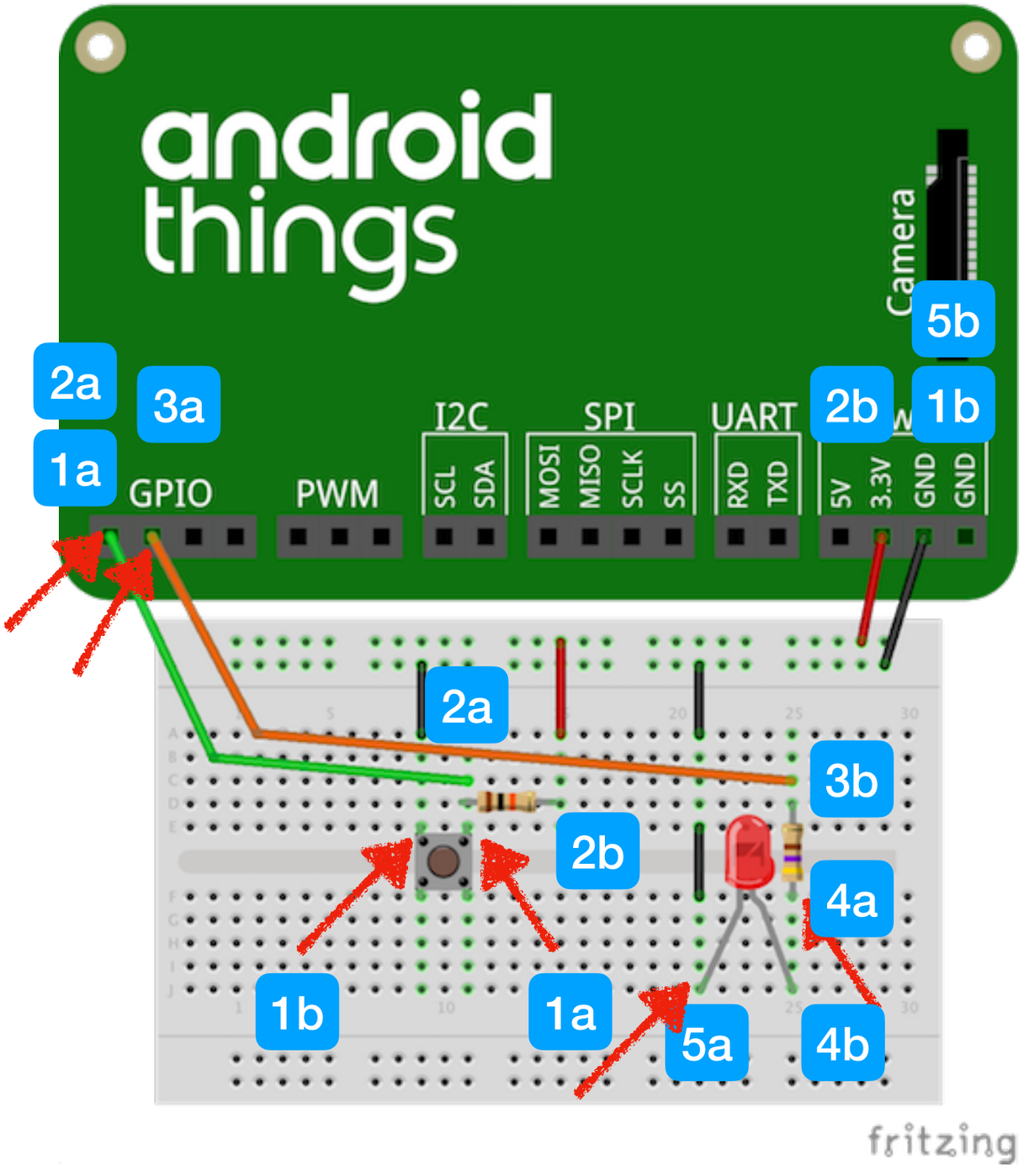
fritzing

The breadboard

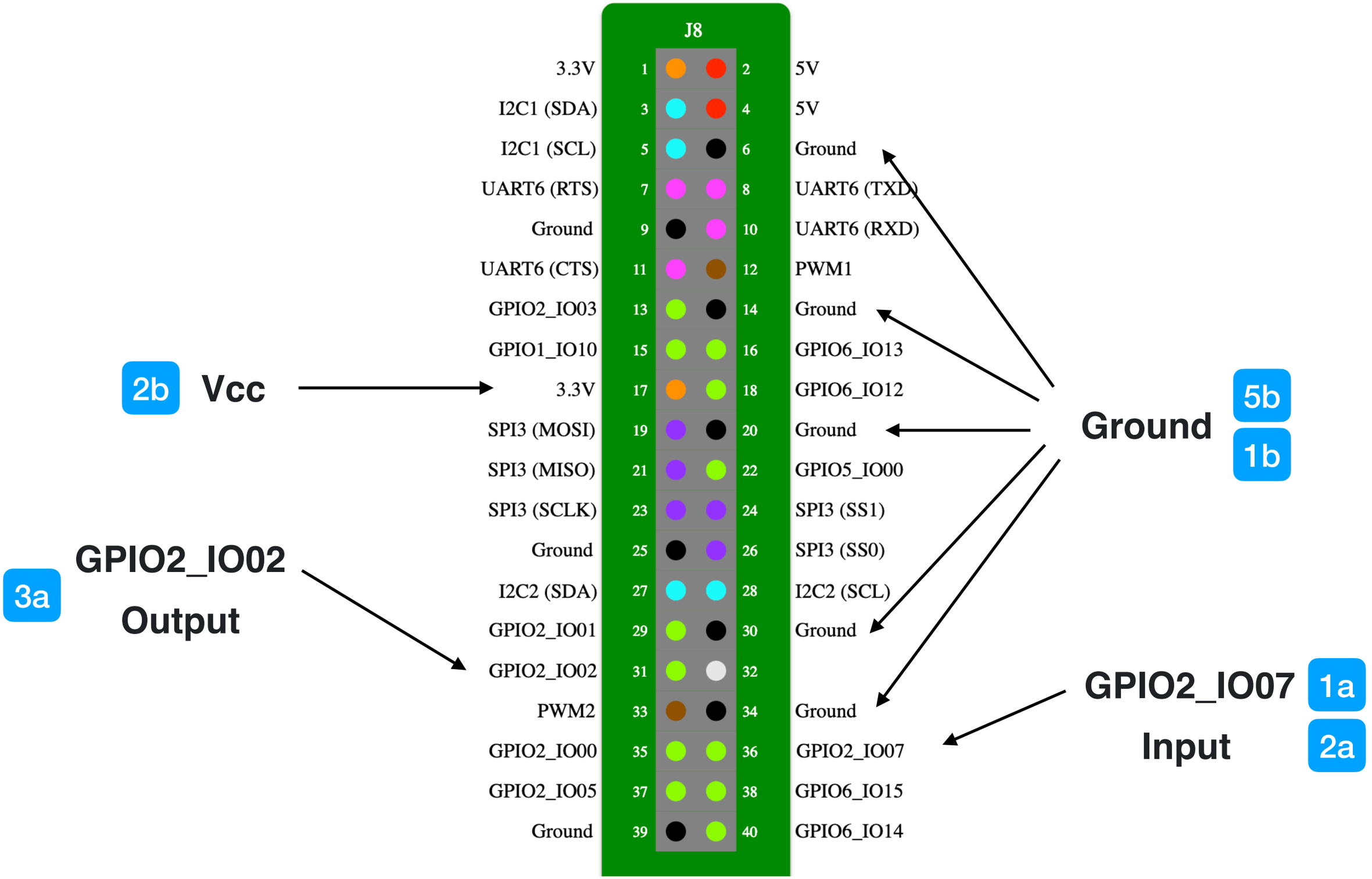


Create the Connections

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.

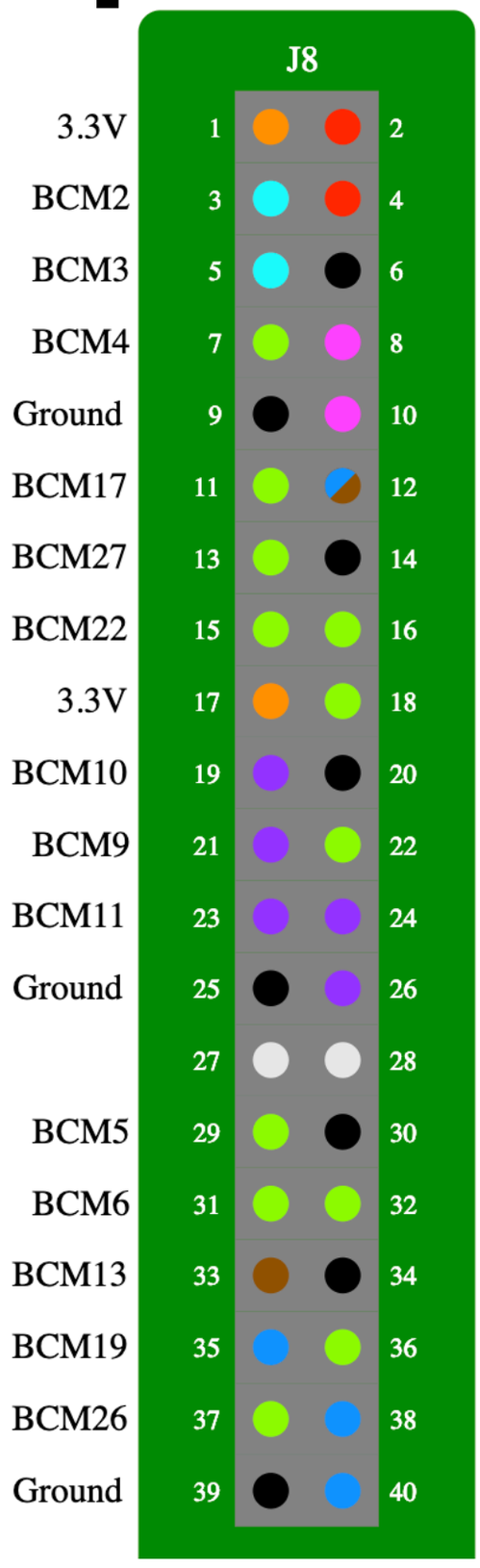


NXP i.MX7D



https://developer.android.com/things/hardware/imx7d#io-pinout

Raspberry Pi



2b **Vcc**

Ground 5b 1b

3a **BCM6** **Output**

BCM21 **Input** 1a 2a

https://developer.android.com/things/hardware/raspberrypi#io-pinout

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}")
  }
}
```

https://developer.android.com/reference/com/google/android/things/pio/PeripheralManager.html

Handle button events

```
class ButtonActivity : Activity() {
    companion object {
        const val BUTTON_PIN_NAME = "GPIO2_IO07" // GPIO port wired to the button
    }

    private lateinit var buttonGpio: Gpio
    // Step 4. Register an event callback.
    private val mCallback = GpioCallback {
        logi("GPIO changed, button pressed")
        // Step 5. Return true to keep callback active.
        true
    }

    override fun onDestroy() {
        super.onDestroy()
        // Step 6. Close the resource.
        buttonGpio.unregisterGpioCallback(mCallback)
        try {
            buttonGpio.close()
        } catch (e: IOException) {
            loge("Error on PeripheralIO API", e)
        }
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        val manager = PeripheralManager.getInstance()
        try {
            // 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.
            buttonGpio.registerGpioCallback(mCallback)
        } catch (e: IOException) {
            loge("Error on PeripheralIO API", e)
        }
    }
}
```

Blink an LED

```
class BlinkActivity : Activity() {
  private companion object {
    const val LED_PIN_NAME = "GPIO2_IO02" // 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)
    }
  }
```

Raspberry Pi OS – Raspberry P

raspberrypi.com/software/

Install Raspberry Pi OS using Raspberry Pi Imager

Raspberry Pi Imager is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card, ready to use with your Raspberry Pi. Watch our 45-second video to learn how to install an operating system using Raspberry Pi Imager.

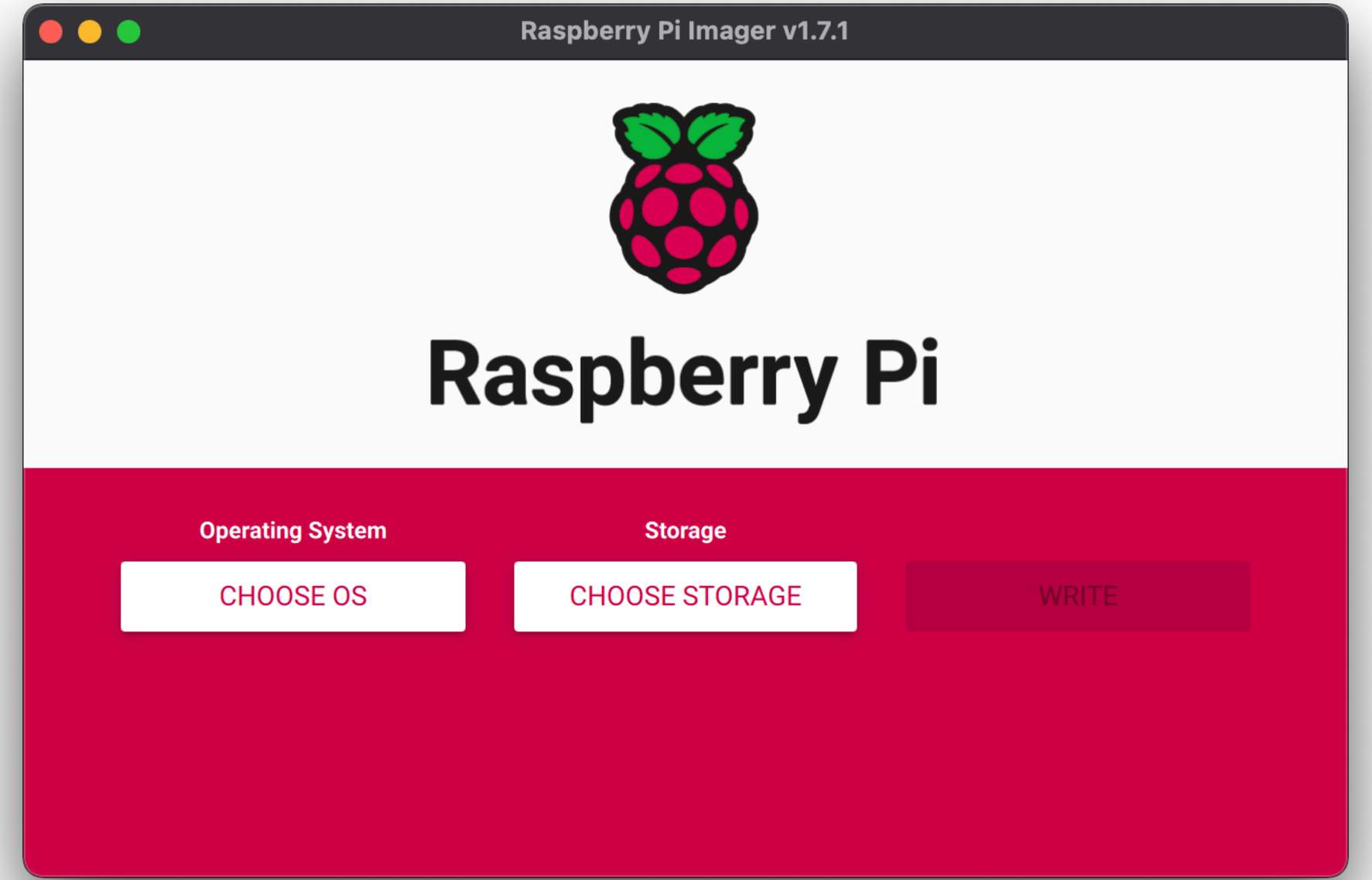
Download and install Raspberry Pi Imager to a computer with an SD card reader. Put the SD card you'll use with your Raspberry Pi into the reader and run Raspberry Pi Imager.

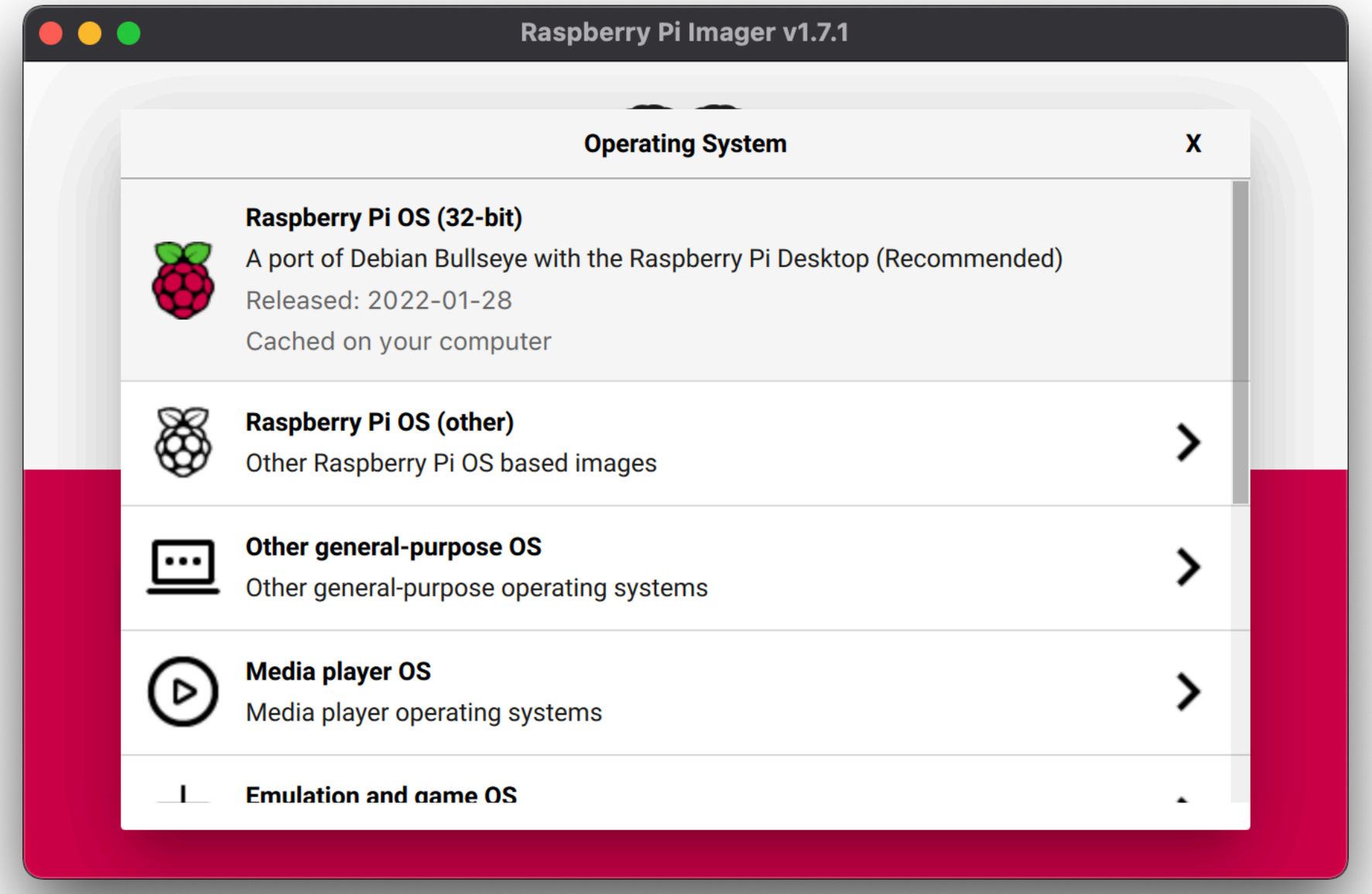
Download for macOS

Download for Windows

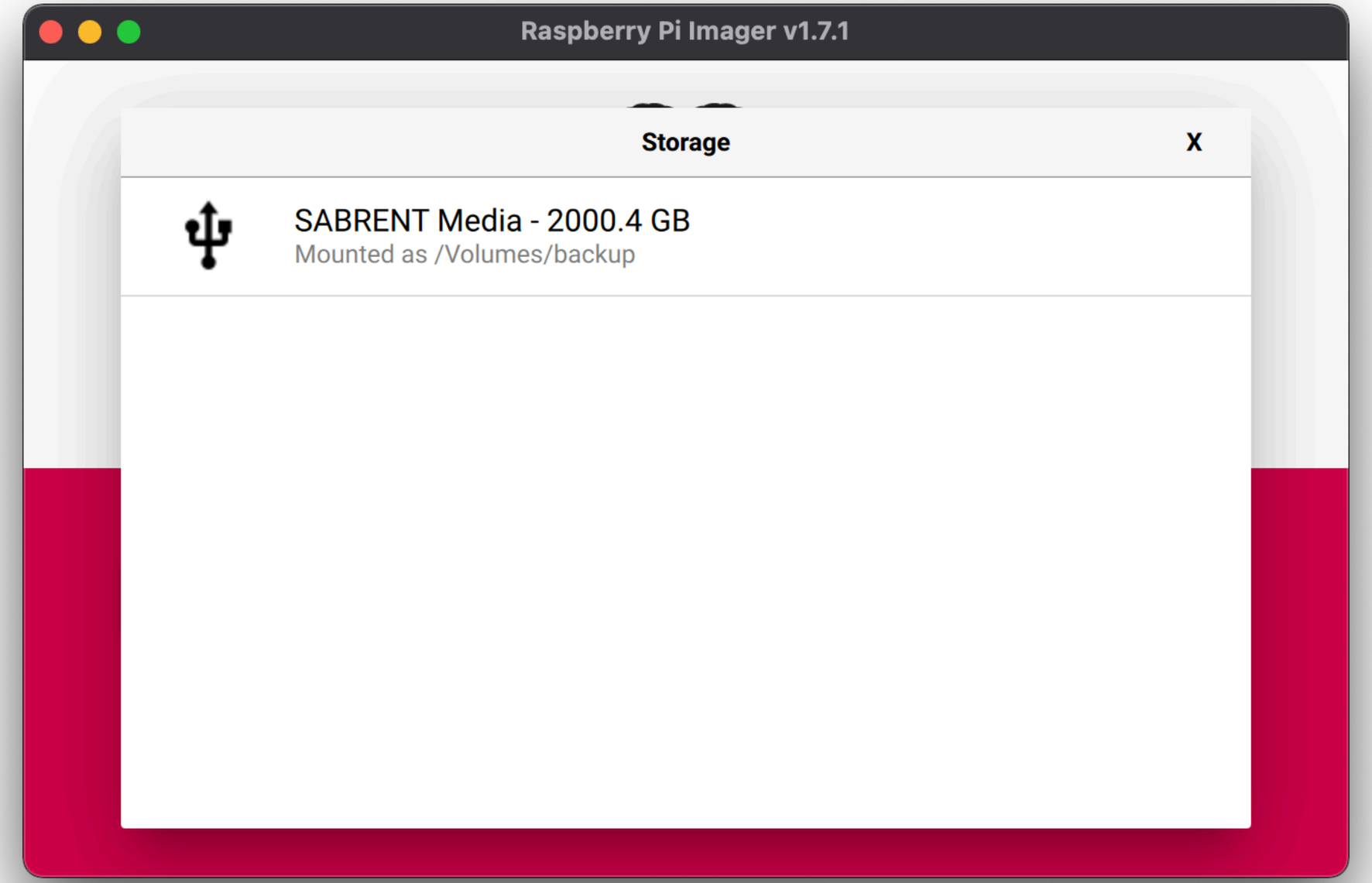
Download for Ubuntu for x86

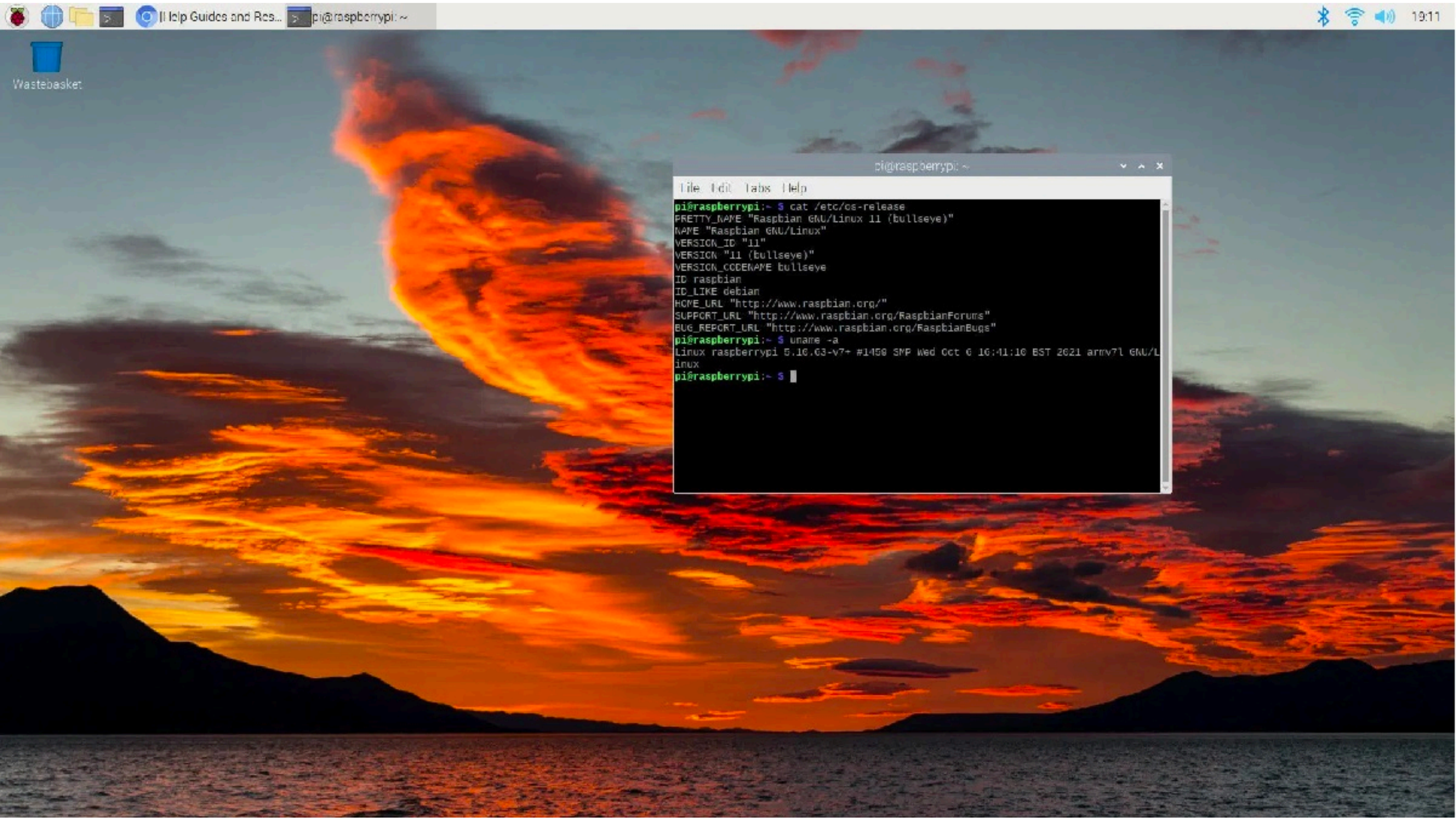
To install on **Raspberry Pi OS**, type `sudo apt install rpi-imager` in a Terminal window.


Raspberry Pi Imager v1.7.1
Raspberry Pi
Operating System
CHOOSE OS
Storage
CHOOSE STORAGE
WRITE

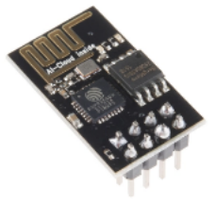

Raspberry Pi Imager v1.7.1
Operating System
X
Raspberry Pi OS (32-bit)
A port of Debian Bullseye with the Raspberry Pi Desktop (Recommended)
Released: 2022-01-28
Cached on your computer
Raspberry Pi OS (other)
Other Raspberry Pi OS based images
Other general-purpose OS
Other general-purpose operating systems
Media player OS
Media player operating systems
Emulation and game OS

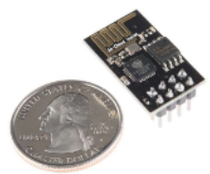

Raspberry Pi Imager v1.7.1
Raspberry Pi
Operating System
RASPBERRY PI OS (32-BIT)
Storage
CHOOSE STORAGE
WRITE

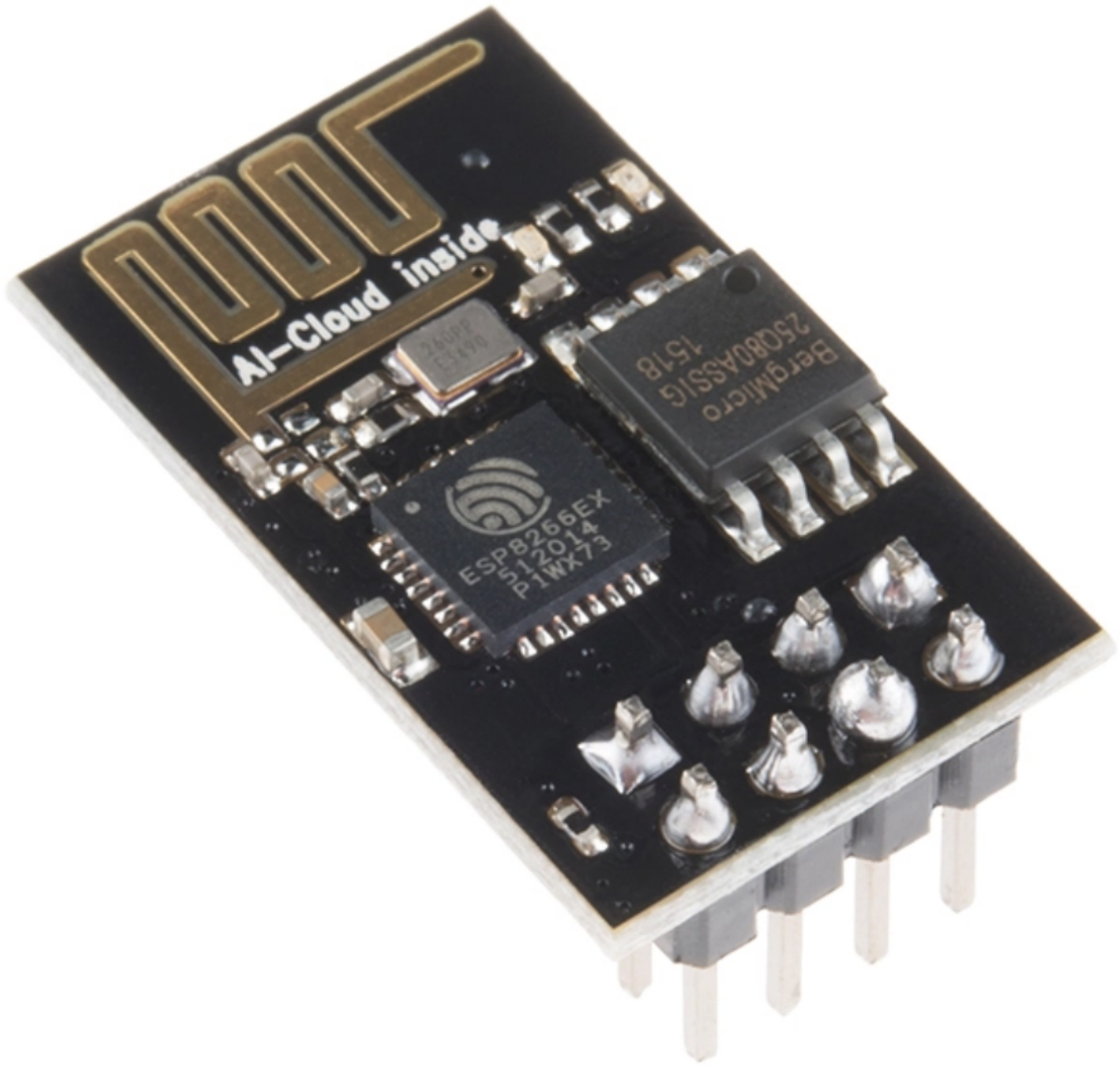

Raspberry Pi Imager v1.7.1
Storage
X
SABRENT Media - 2000.4 GB
Mounted as /Volumes/backup


ll Help Guides and Res...
pi@raspberrypi: ~
19:11
Wastebasket
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ cat /etc/os-release
PRETTY_NAME "Raspbian GNU/Linux 11 (bullseye)"
NAME "Raspbian GNU/Linux"
VERSION_ID "11"
VERSION "11 (bullseye)"
VERSION_CODENAME bullseye
ID raspbian
ID_LIKE debian
HOME_URL "http://www.raspbian.org/"
SUPPORT_URL "http://www.raspbian.org/RaspbianForums"
BUG_REPORT_URL "http://www.raspbian.org/RaspbianBugs"
pi@raspberrypi:~ $ uname -a
Linux raspberrypi 5.10.63-v7+ #1459 SMP Wed Oct 6 16:41:10 BST 2021 armv7l GNU/L
inux
pi@raspberrypi:~ $

Connect Arduino2Internet







ESP8266 WiFi

12 Review-uri

19,00 Lei

Stoc sku: WIR-46

STOC EPUIZAT

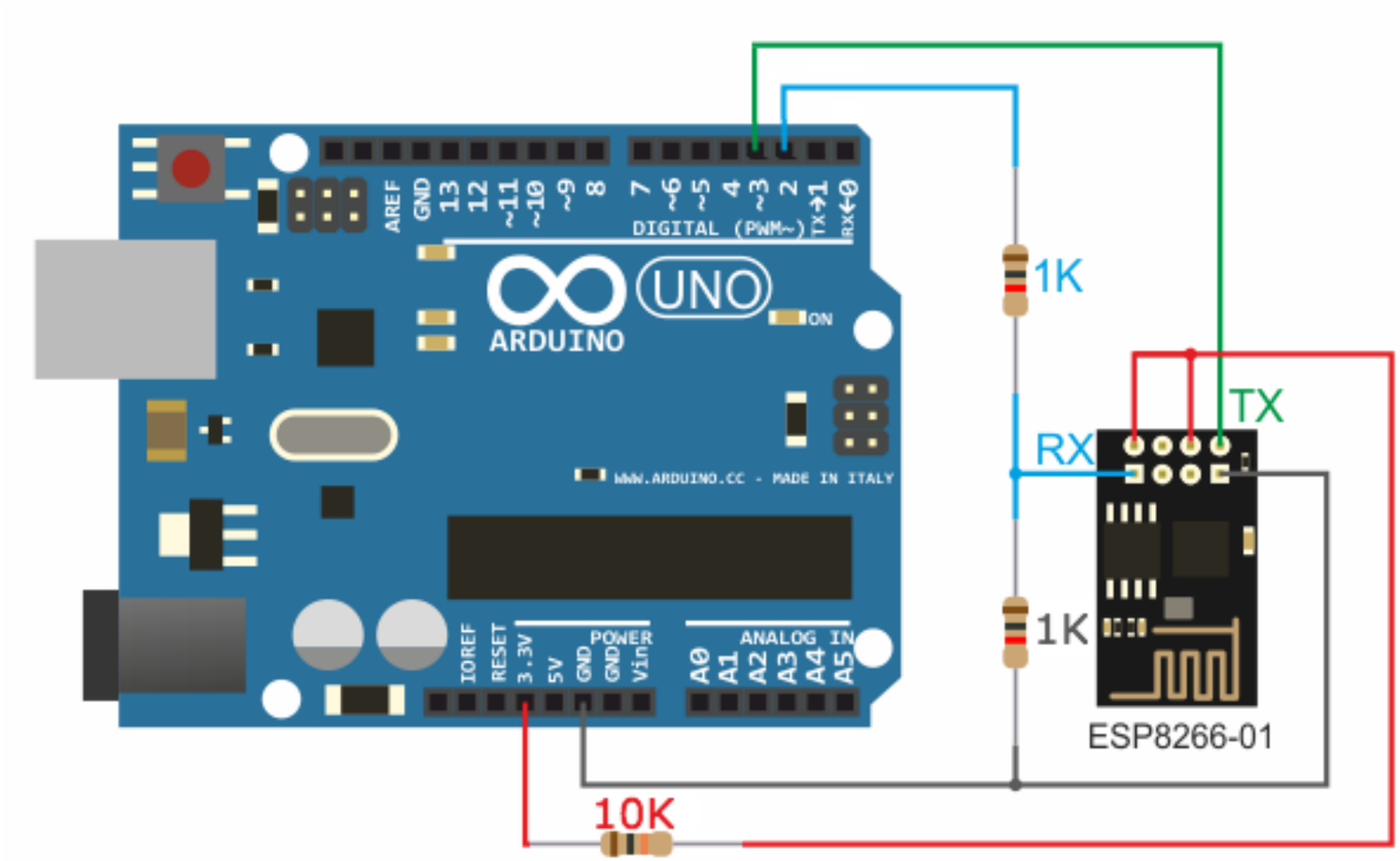
Durata de livrare: 1 - 3 zile lucratoare

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
     }
}
```

https://github.com/imjeffparedes/iot-esp8266-arduino-interface/

Lecture outcomes

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