

00A0 2203∃ 2200∀ 2286⊆ 2713x 27FA⇔ 221A√ 221B<sup>3</sup>/ 2295⊕ 2297⊗ UTF8gbsn

---

# SunFounder RasPad 3

[www.sunfounder.com](http://www.sunfounder.com)

2022 年 08 月 03 日



<b>1</b>	<b>组件列表</b>	<b>3</b>
<b>2</b>	<b>组装说明</b>	<b>5</b>
<b>3</b>	<b>快速使用指南</b>	<b>19</b>
3.1	安装树莓派镜像 . . . . .	19
3.2	在 RasPad 上安装虚拟键盘 . . . . .	25
3.3	右键单击 RasPad . . . . .	30
3.4	安装旋转屏幕服务 . . . . .	31
<b>4</b>	<b>RASPAD Launcher</b>	<b>33</b>
4.1	介绍 . . . . .	33
4.2	安装指南 . . . . .	34
<b>5</b>	<b>接口介绍</b>	<b>35</b>
5.1	为 RasPad 充电 . . . . .	36
5.2	音量和亮度设置 . . . . .	39
5.3	连接耳机或其他屏幕 . . . . .	42
<b>6</b>	<b>RASPAD 项目</b>	<b>47</b>
6.1	多媒体选项 . . . . .	47
6.1.1	看视频 . . . . .	47
6.1.2	阅读电子书 . . . . .	48
6.1.3	玩游戏 . . . . .	49
6.2	复古游戏机 . . . . .	50
6.2.1	描述 . . . . .	50
6.2.2	所需组件 . . . . .	51
6.2.3	游戏系统安装 . . . . .	53
6.2.4	RetroPie 配置 . . . . .	56
6.3	3D 打印机 . . . . .	57
6.3.1	Touch UI . . . . .	58
6.3.2	制作 3D 模型 . . . . .	58
6.3.3	打印 3D 模型 . . . . .	63
6.3.4	视频 . . . . .	66
6.4	Home Assistant . . . . .	66
6.4.1	安装 Home Assistant . . . . .	67
6.4.2	安装和配置 ESPHome . . . . .	68



6.4.3	登录	76
6.4.4	添加智能设备	81
6.4.5	添加带有 ESP32 的 DIY 设备	86
6.4.6	在开机时启动 Home Assistant 服务	91
6.4.7	启动时进入 Home Assistant 页面	91
6.4.8	Home Assistant 文件结构	93
6.5	Cloud4RPi	94
6.5.1	Cloud4RPi 的快速使用指南	94
6.5.2	详细了解 control.py	99
6.5.3	项目	100
6.6	GPIO 编程	117
6.6.1	Electronic Kit for Raspberry Pi and Arduino	117
6.6.2	Da Vinci Kit for Raspberry Pi	117
6.7	Scratch 项目	118
6.7.1	会动的名字	119
6.7.2	创作故事	127
6.7.3	制作音乐	137
6.7.4	点击游戏	142
6.7.5	射击游戏	147
6.7.6	击打魔法鱼	155
6.7.7	二进制游戏	164
6.7.8	会飞的鹦鹉	171
6.7.9	油漆盒	183
6.7.10	吃香蕉游戏	197
6.7.11	制作乐器	201
6.7.12	自定义 RasPad 监控设备	207
6.8	在 RasPad 上安装其他系统	212
<b>7</b>	<b>附录</b>	<b>215</b>
7.1	安装摄像头	215
7.2	GPIO 扩展板	219
7.3	获取树莓派的 IP 地址	221
7.3.1	有屏幕	221
7.3.2	无屏幕	222
7.4	用 SSH 远程控制	223
<b>8</b>	<b>常见问题</b>	<b>227</b>
8.1	修复树莓派启动问题	227
8.2	无法检测到 Micro SD 卡?	227
8.3	屏幕不能旋转?	227
8.4	关于睡眠问题	228
8.5	关于关机问题	228
8.6	桌面不是全屏	228
8.7	关于触摸问题	229
8.8	关于音频问题	229
8.9	风扇太吵?	230
8.10	耗电太快?	230
8.11	触摸功能有时会失效?	230
<b>9</b>	<b>版权声明</b>	<b>231</b>



一款完美兼容树莓派 4B 的理想编程树莓派平板电脑。RasPad 内置 10.1 英寸 IPS Pi 触摸屏液晶显示器，分辨率高达 1280×800 像素，10 指触摸，5 小时长电池寿命。只需 2 个步骤就可以轻松组装，随时随地开始使用。

通过利用 HDMI，你可以轻松地将 RasPad 与其他显示器进行扩展，以便更好地进行多任务处理，或者对你的显示器进行镜像。此外，由于三角形的结构设计和六轴加速度传感器，它可以平放或直立在桌子上，也可以旋转 90°/180°/270°，进行惊人的演示！

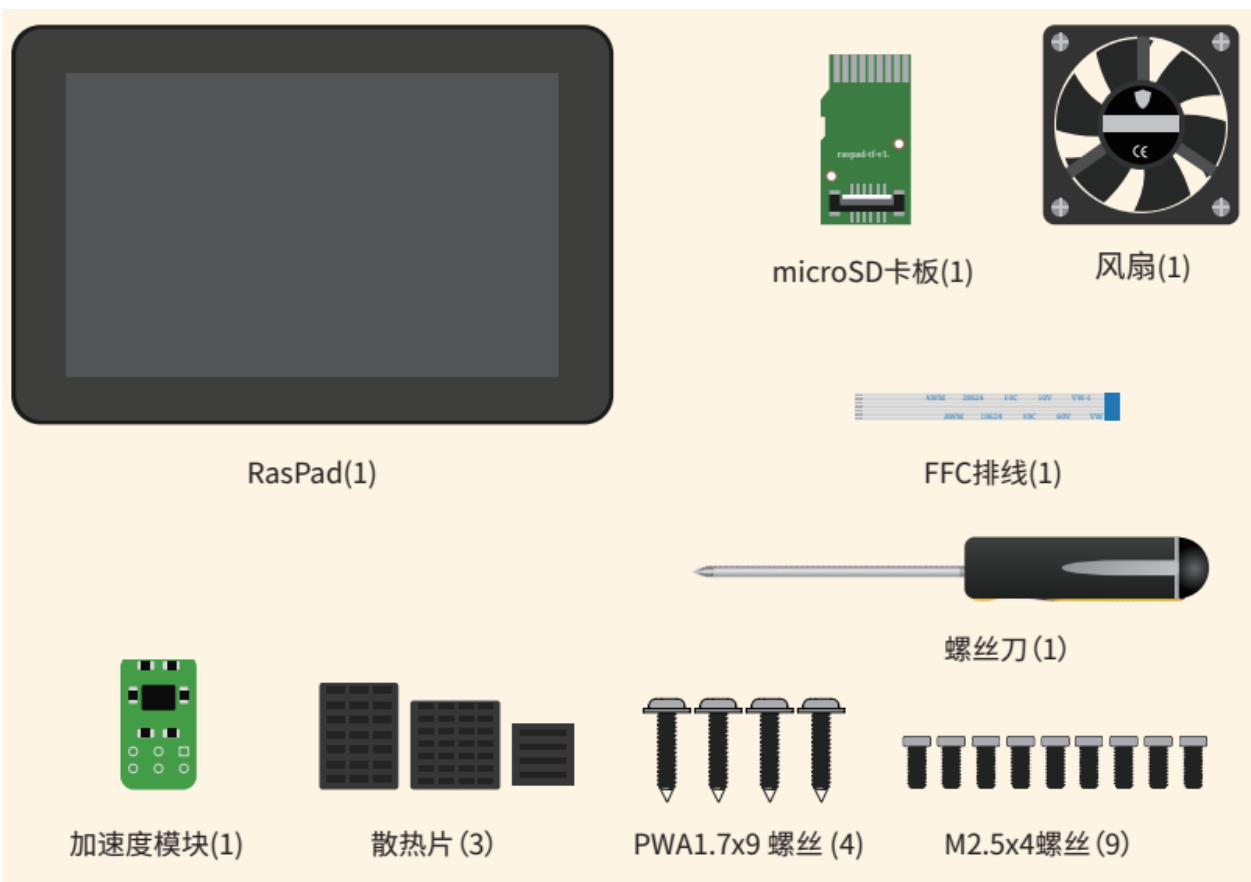
拥有所有的树莓派端口，包括以太网、HDMI、音频、USB 和电源。所有的 RPi I/O 接口，定制的操作系统，立体声扬声器和更多的功能，让开发者有一个方便的硬件黑客体验，特别设计的 GPIO 扩展和摄像头孔使用户为 RasPad 建立项目或安装摄像头。你可以从任何地方开始编程，享受日常工作的乐趣。

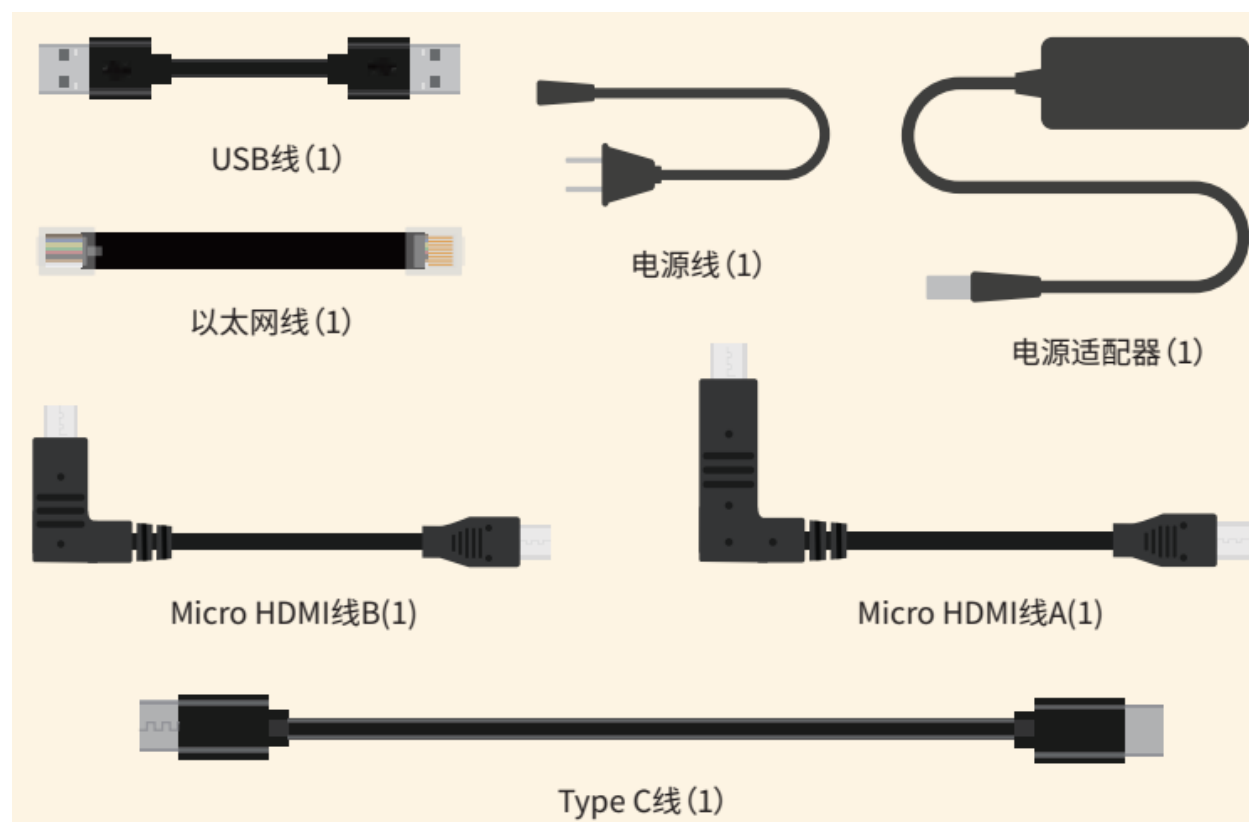
兼容 Arduino、Raspbian、Ubuntu 桌面、Chromium OS、Android 和更多... 实现了平板电脑编程 & 远程控制器 & 备用显示器。除了编程，RasPad 还可以作为一个很好的远程流媒体显示器来制作 3D 打印/玩游戏/建立物联网项目/学习 Scratch 3.0。



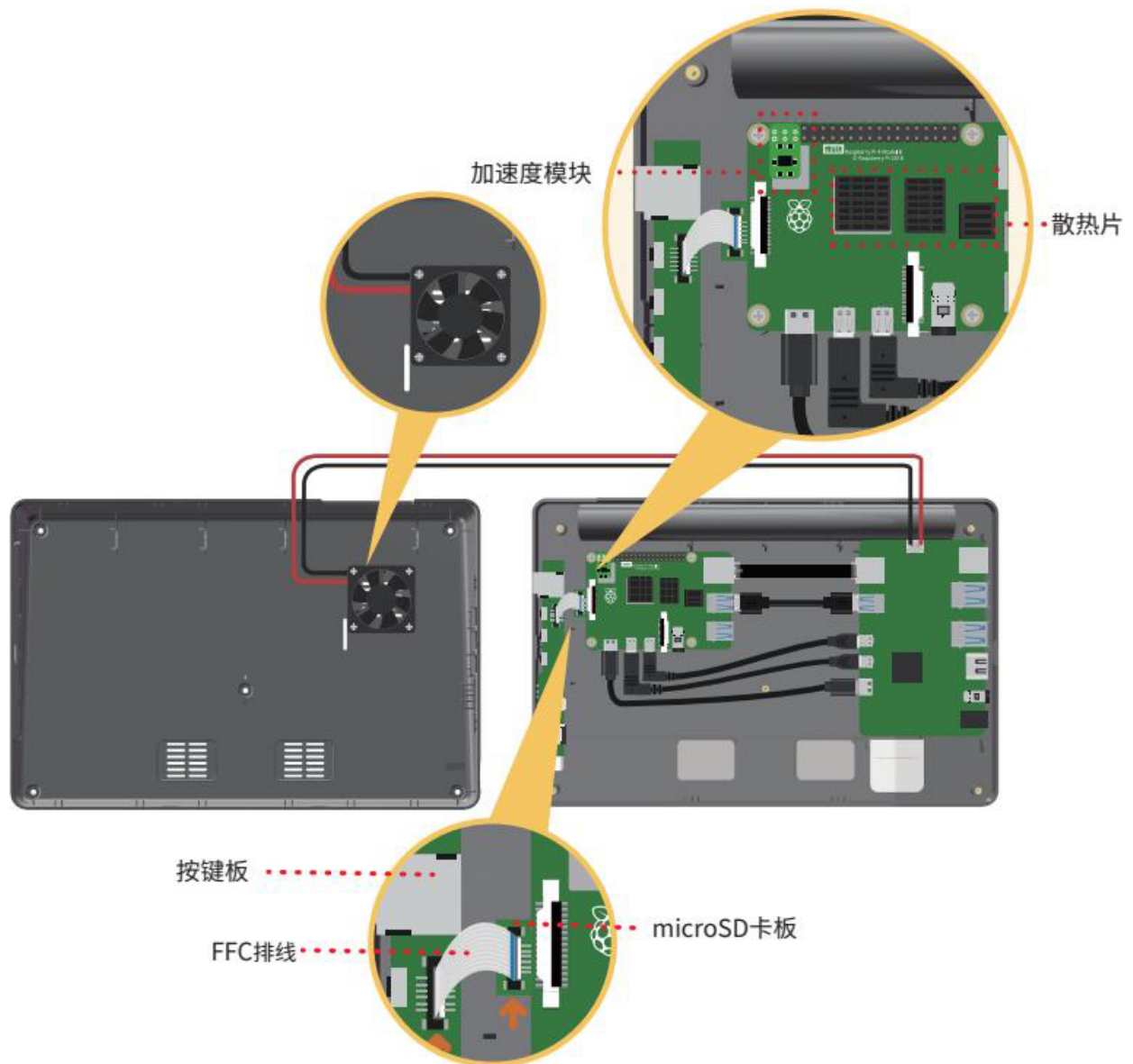
# CHAPTER 1

## 组件列表





1. 在插入 microSD 卡板之前，使用随附的 FFC 排线将按键板连接到树莓派。
2. 连接所有电缆，然后用四颗 M2.5x4 螺丝固定树莓派。
3. 调整风扇的方向，使有贴纸的一侧朝向外部通风口，然后用四颗 PWA1.7x9 螺丝连接风扇。
4. 最后，用五颗 M2.5x4 螺丝固定后盖。



### 详细组装步骤

取出 RasPad，然后将其翻转以取下后盖。



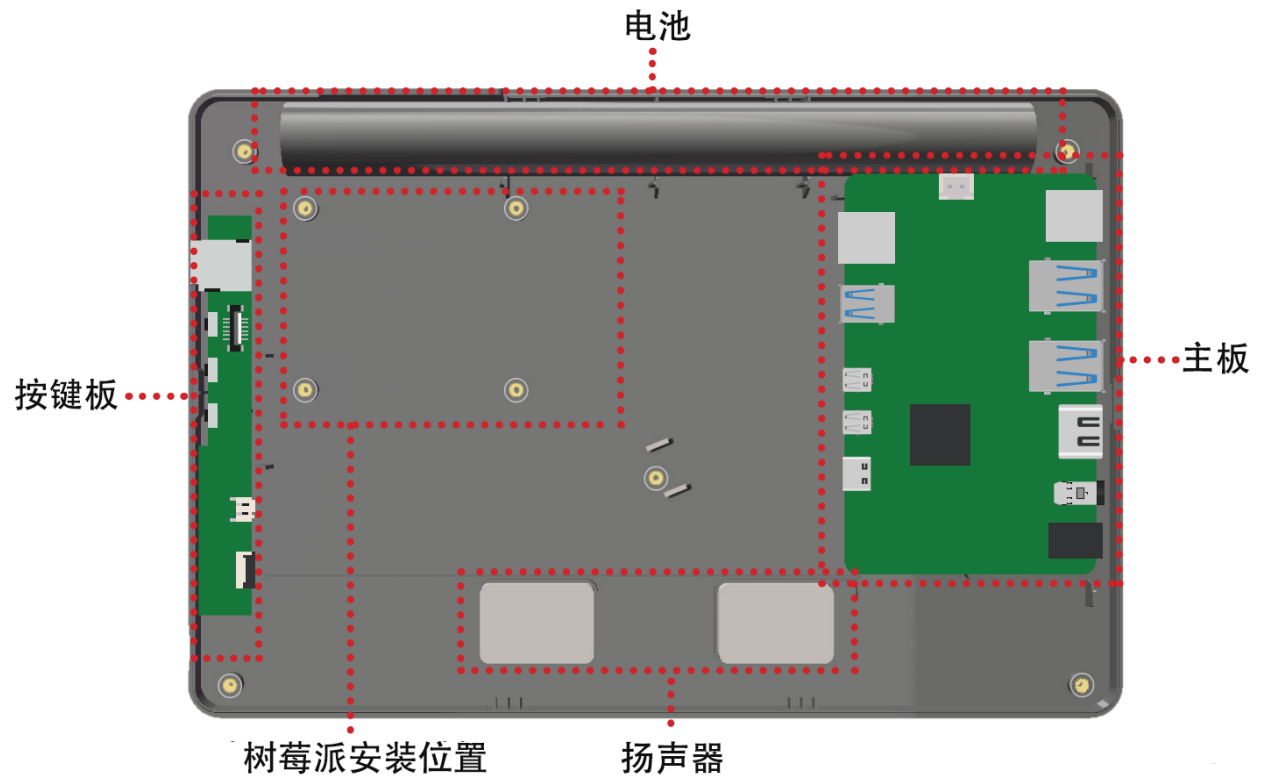
下图是 RasPad 的内部结构图。

---

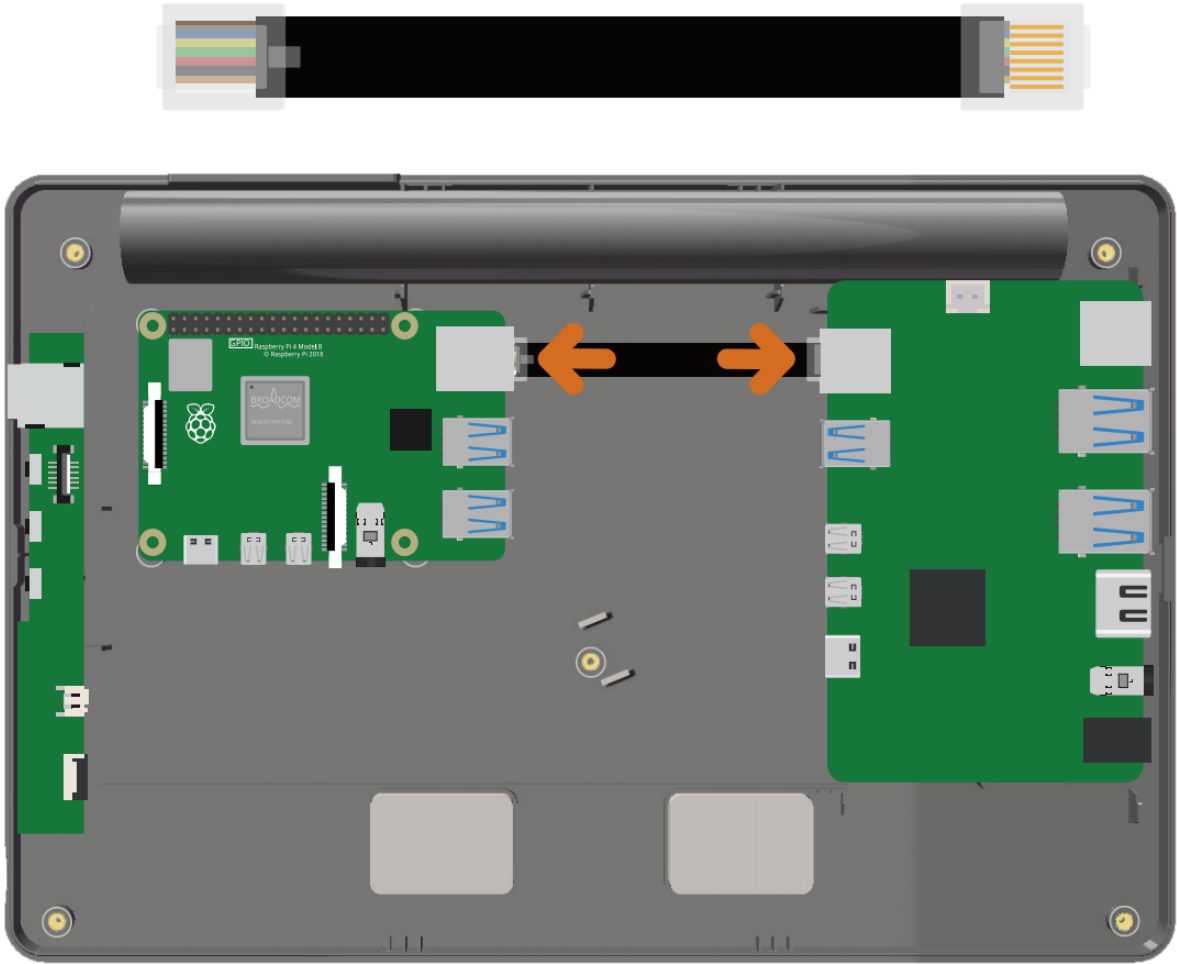
**注解：** 组装时避免按压两个扬声器。

---

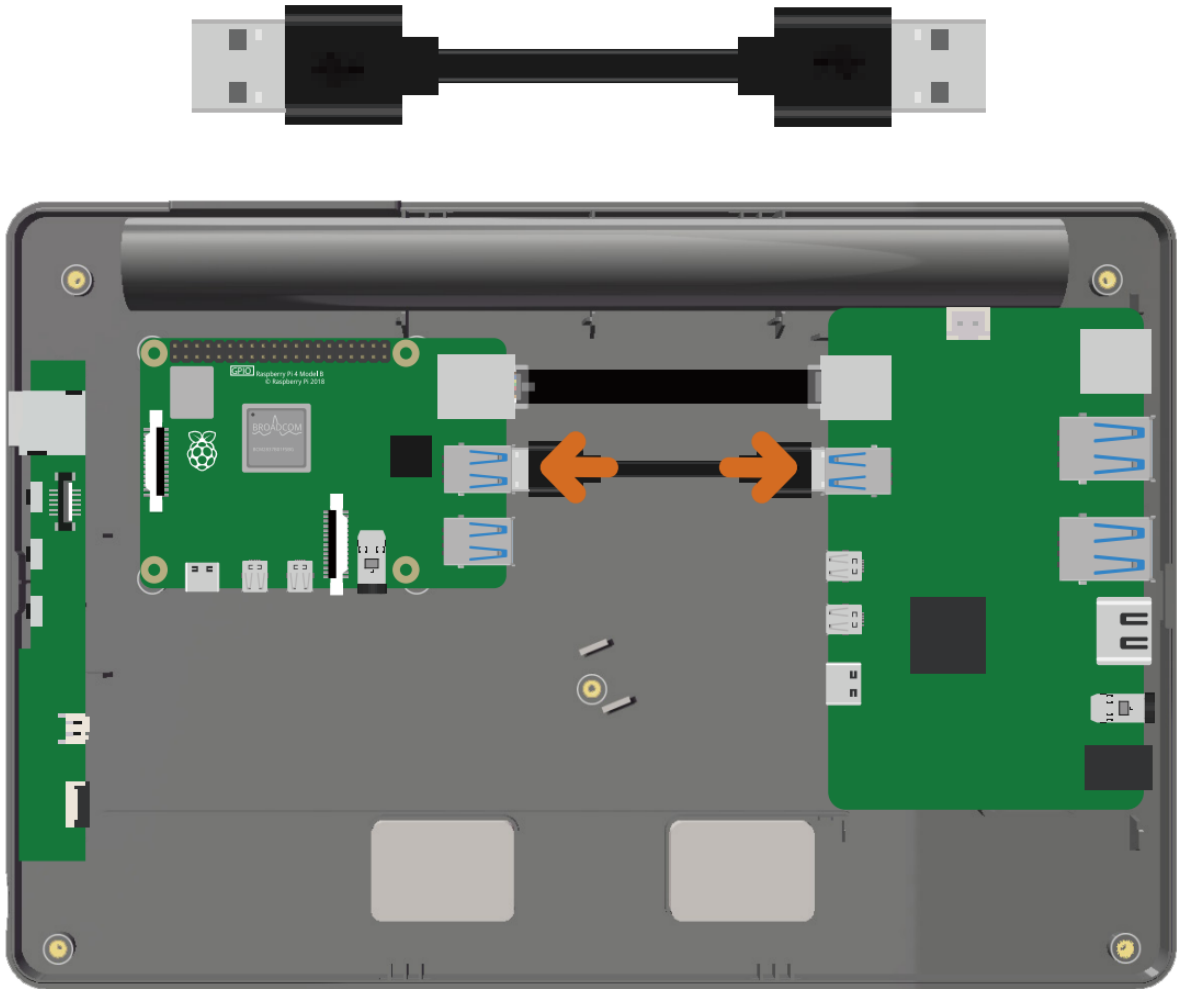




连接树莓派的以太网口和主板。



使用随附的 USB 电缆将树莓派的 USB 3.0 端口连接到 RasPad 主板。

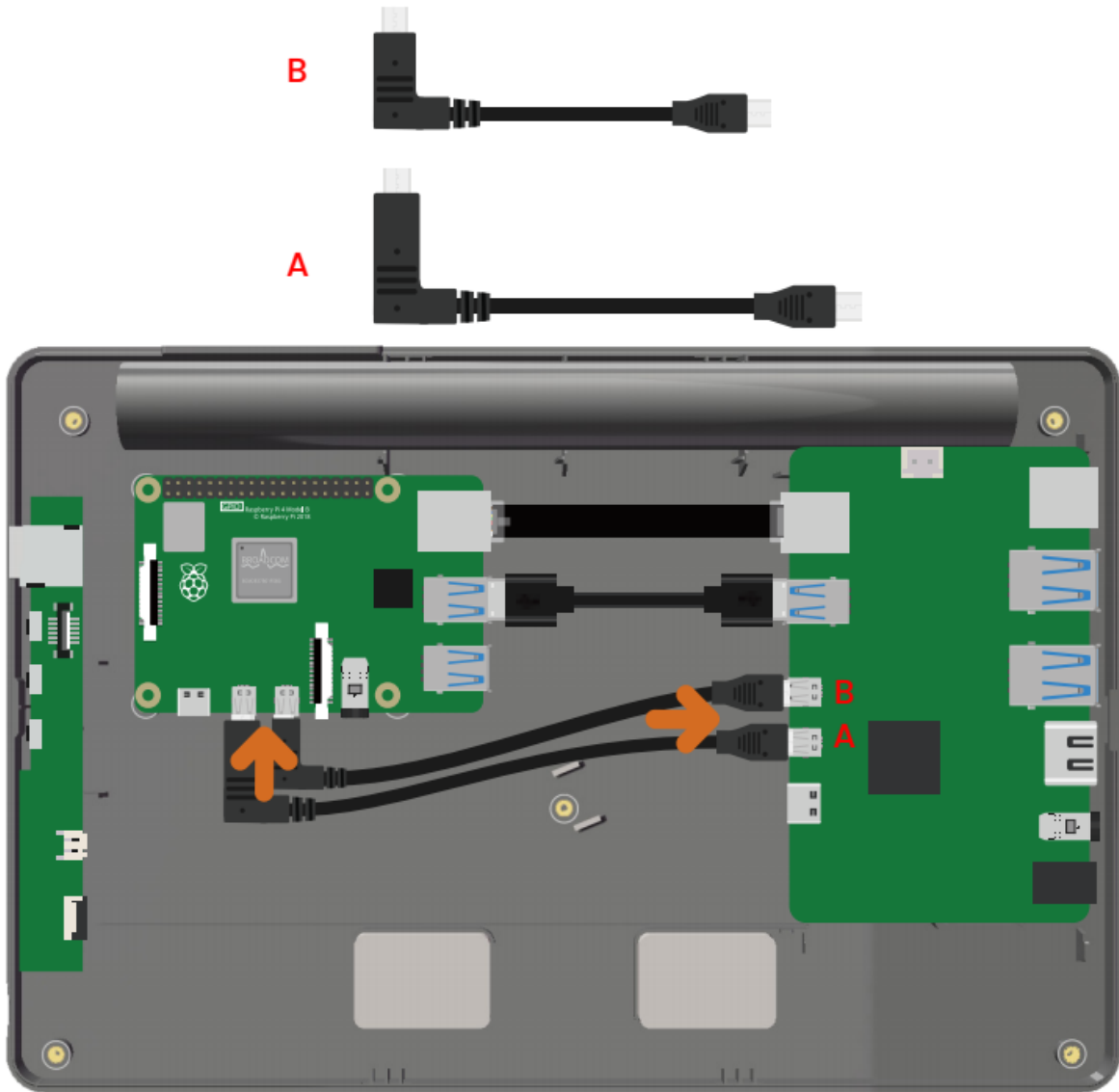


连接两条 Micro-HDMI 线。

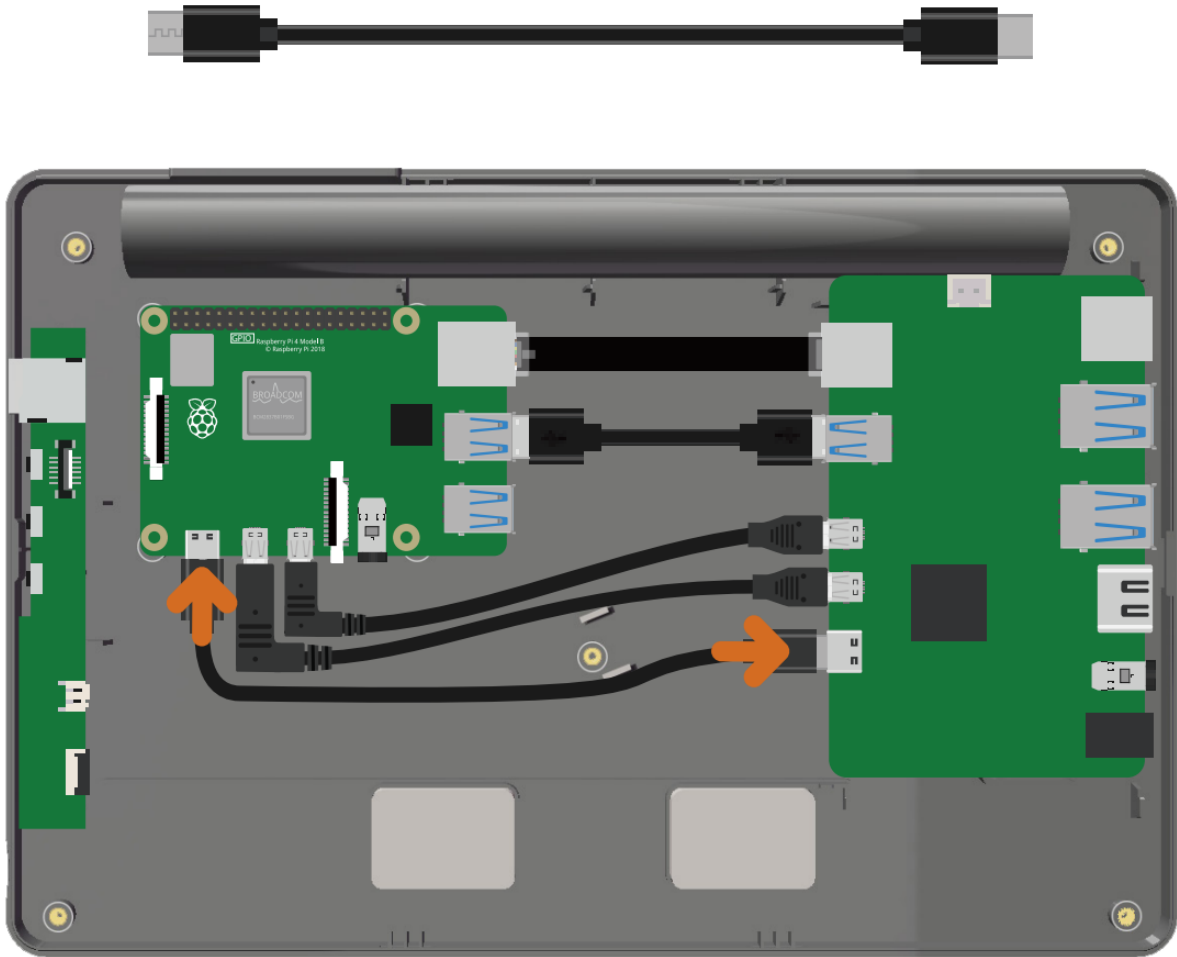
---

**注解：** A 比 B 稍长。

---



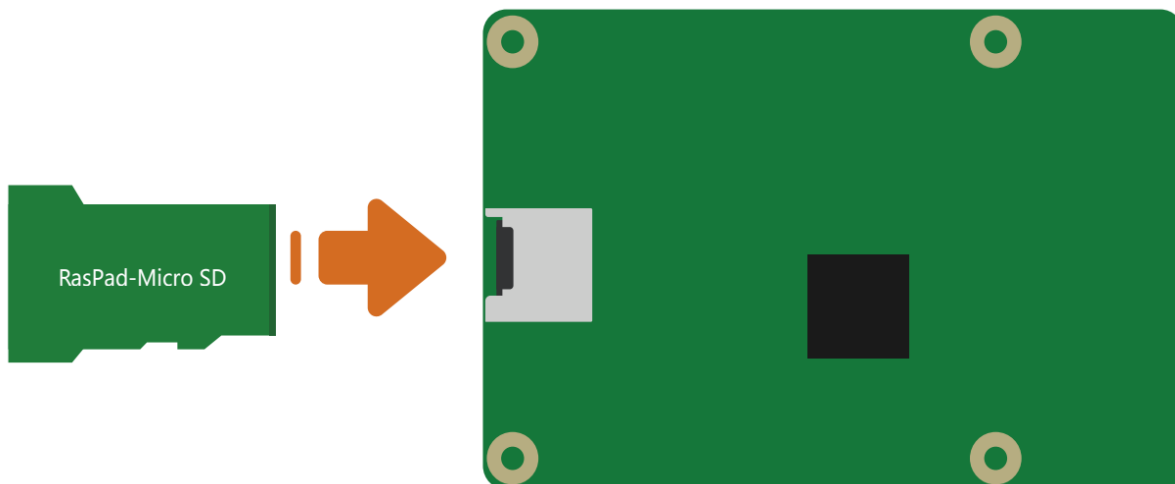
连接 USB-C 线。



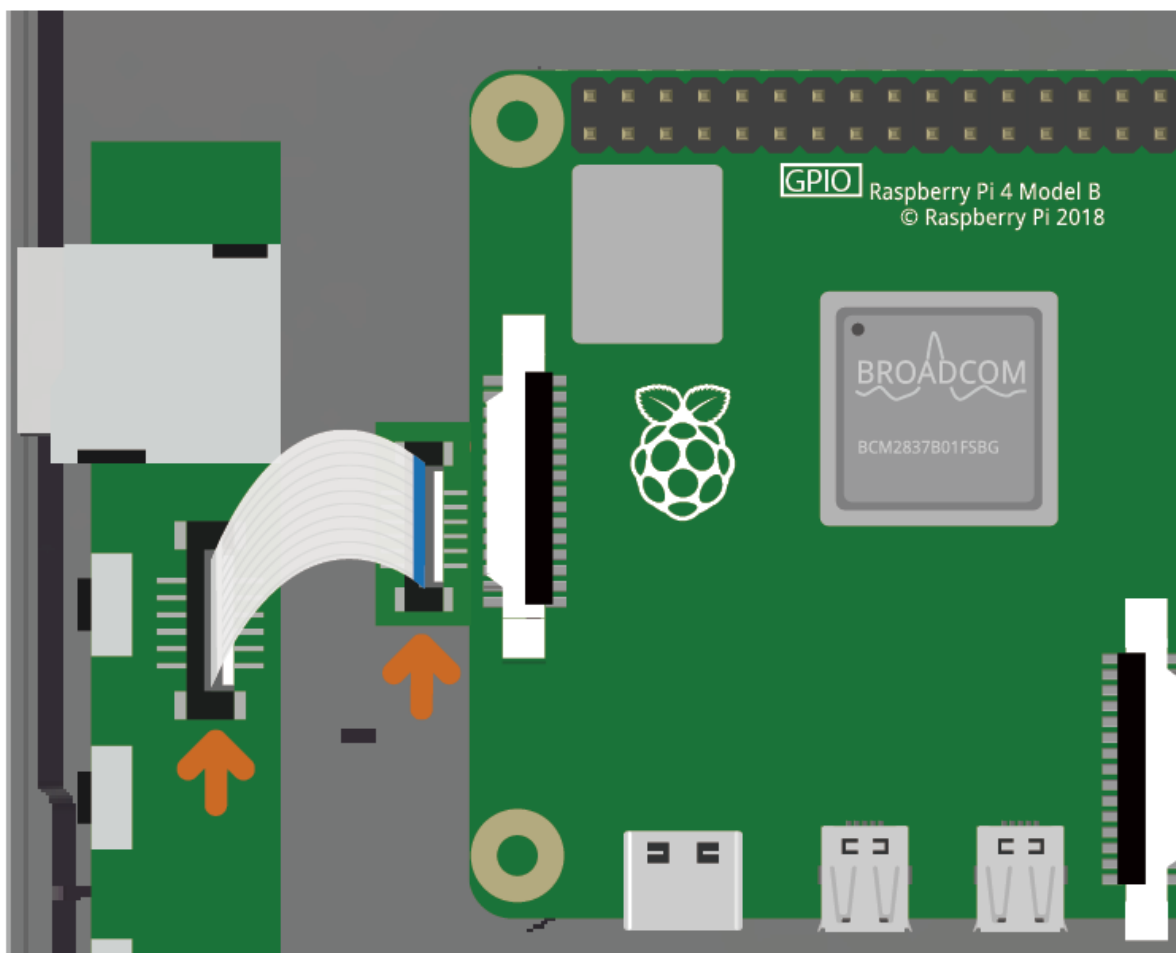
将 FFC 插入 microSD 卡板。



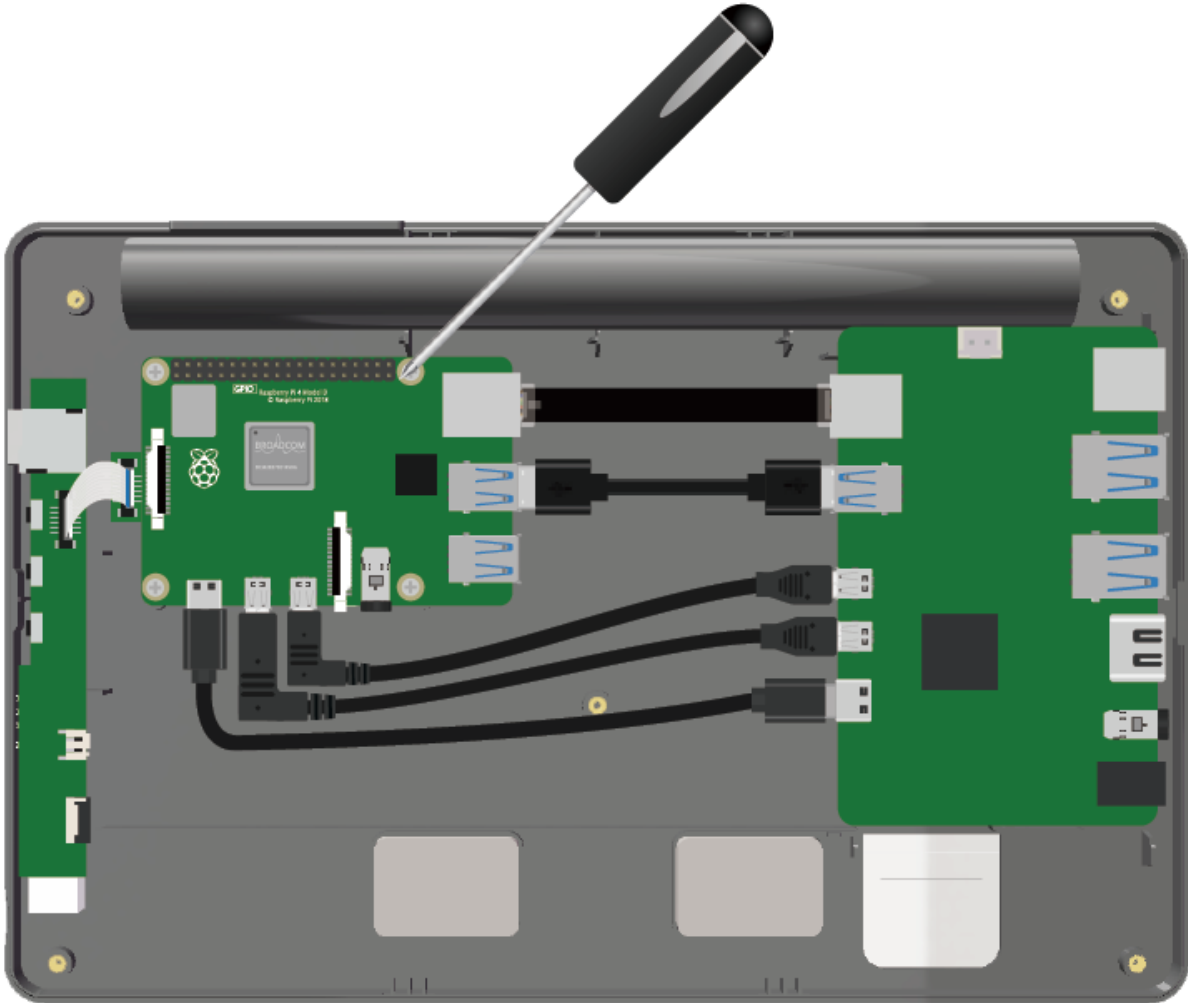
将 microSD 卡板插入树莓派的 Micro SD 卡槽。



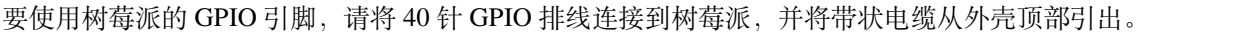
将 FFC 的另一端插入 RasPad 按键板。



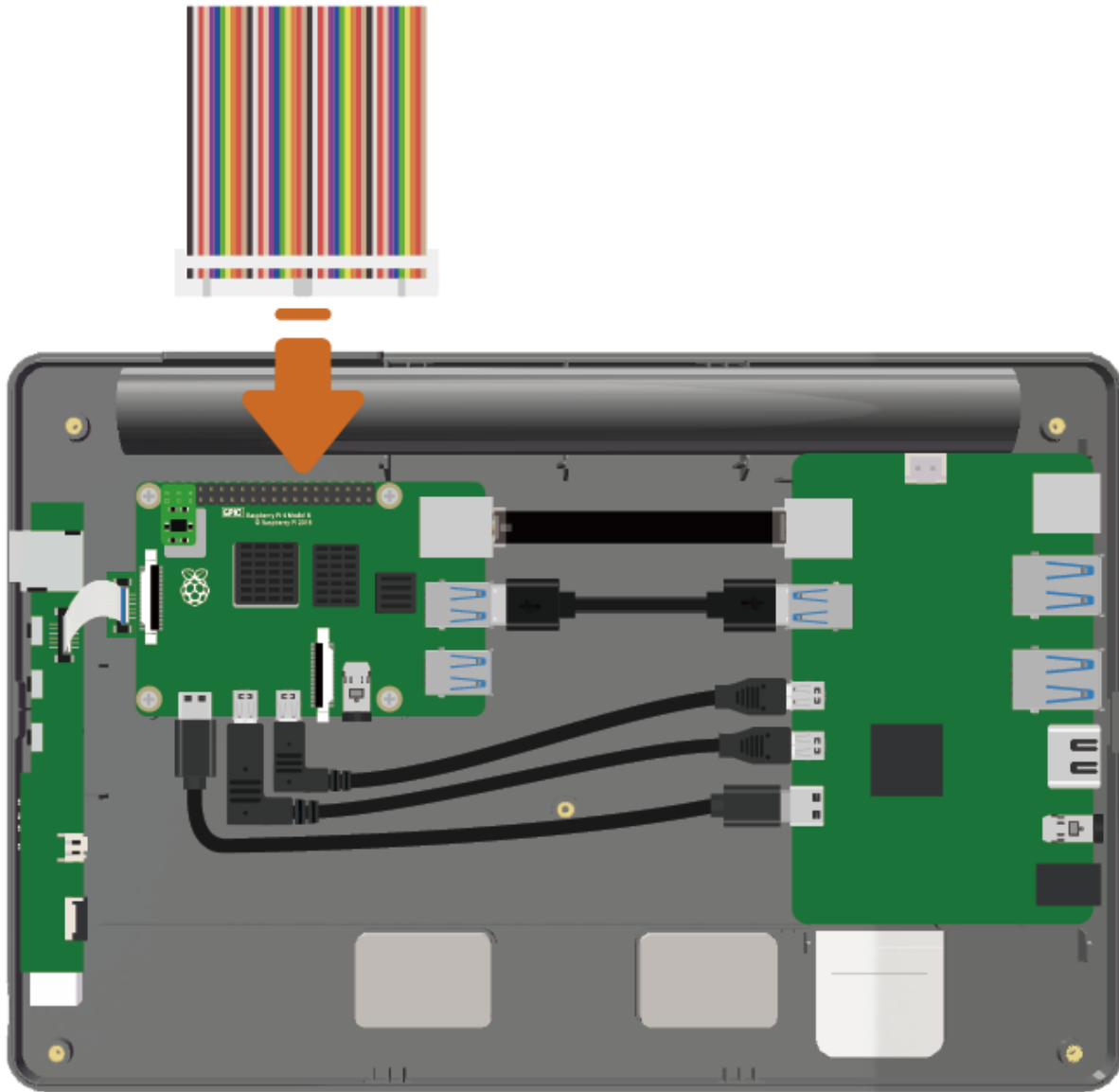
用 4 个 M2.5x4 螺丝固定树莓派。



将三个散热片贴到到树莓派的各个芯片上，并插入加速度模块以实现 RasPad 的自动旋转功能。





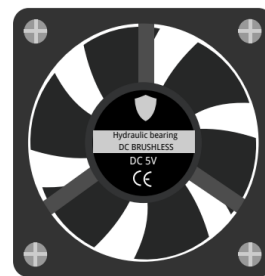
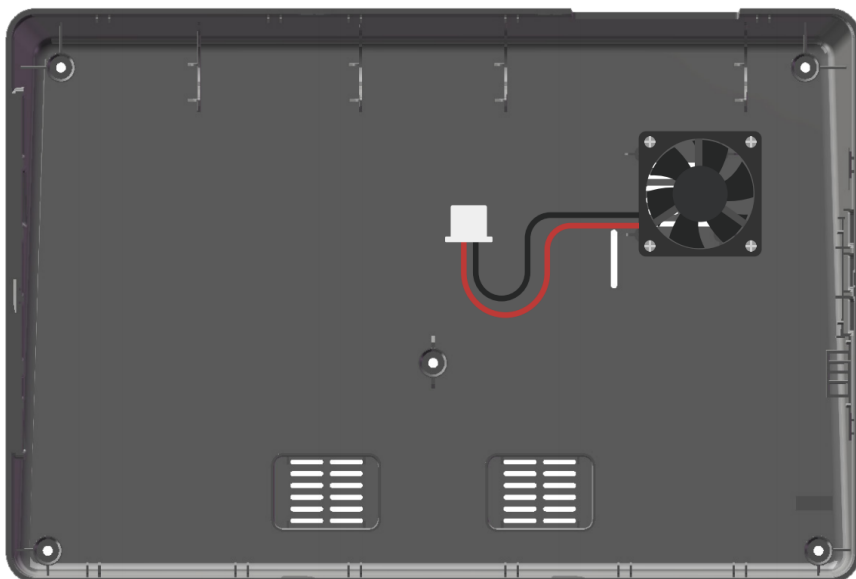


用四颗 M2.5x9 螺丝将风扇固定到后盖上。

---

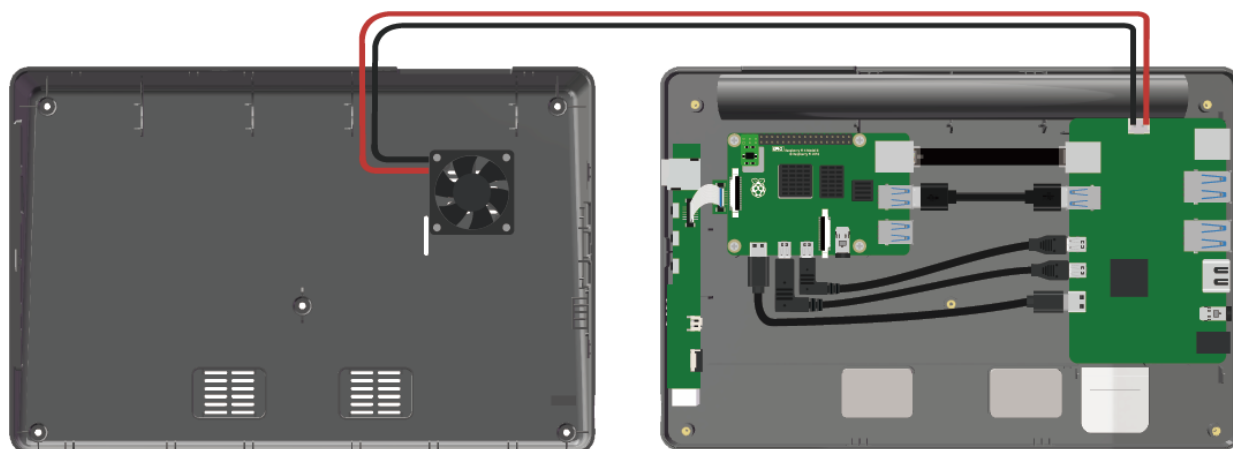
**注解：** 有贴纸的一侧面向外部通风口。

---



图示为有贴纸  
的一侧

将风扇连接到主板。



用五颗 M2.5x4 螺丝固定后盖。

**注解：**请勿强行将后盖固定到位。如果外壳难以合紧，或螺孔未正确对齐，请检查是否线露出或板子没固定好。



本章包括安装树莓派镜像、安装虚拟键盘、设置右键功能和校准旋转屏幕功能。

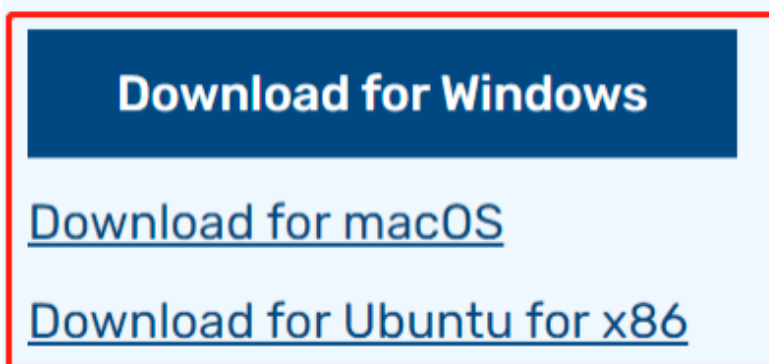
### 3.1 安装树莓派镜像

在将 Micro SD 卡插入 RasPad 按键板之前，需要在 Micro SD 卡上安装操作系统。

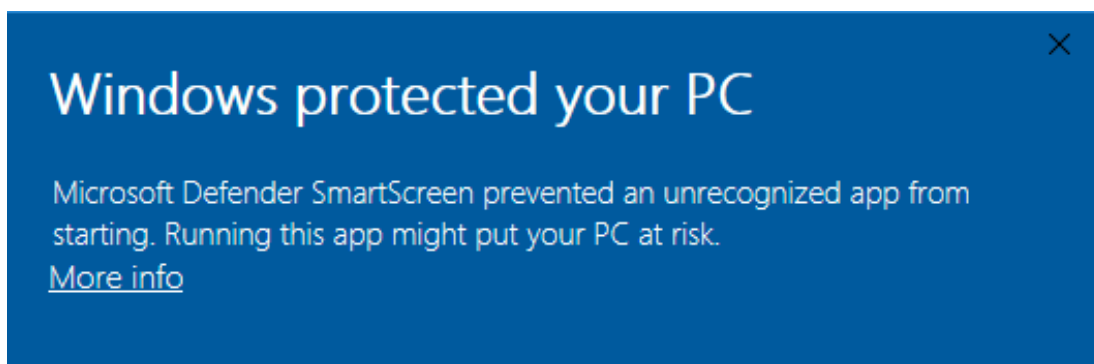
**注解：**如果树莓派已经安装了操作系统，请跳过本章。

Raspberry Pi Imager 是一种将树莓派操作系统从计算机或笔记本电脑安装到 Micro SD 卡的快捷方式。

浏览到 Raspberry Pi Imager 站点并下载与计算机或笔记本电脑的操作系统相匹配的版本 (<https://www.raspberrypi.com/software/>)。下载完成后，启动安装程序并按照提示安装 Raspberry Pi Imager。



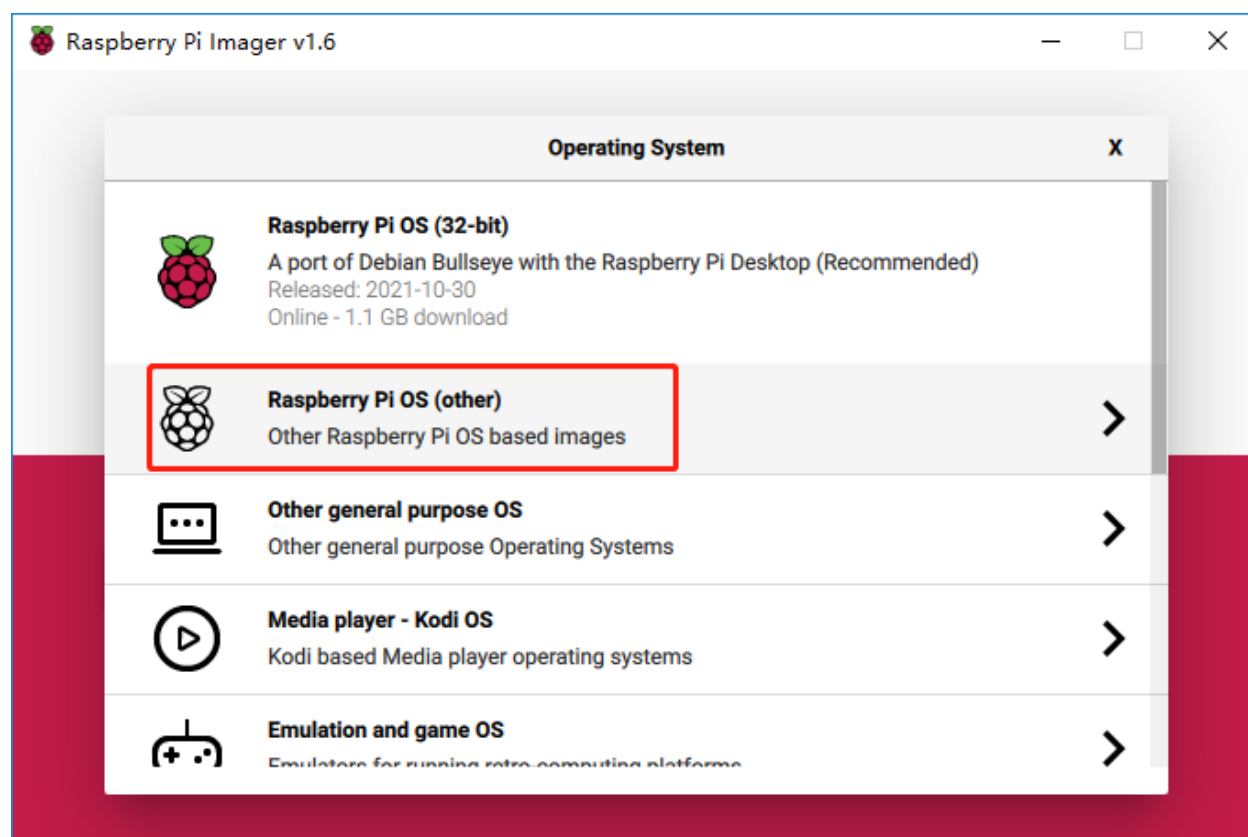
如果 Windows 操作系统阻止应用程序启动，请单击更多信息，然后单击仍然运行以安装 Raspberry Pi Imager。



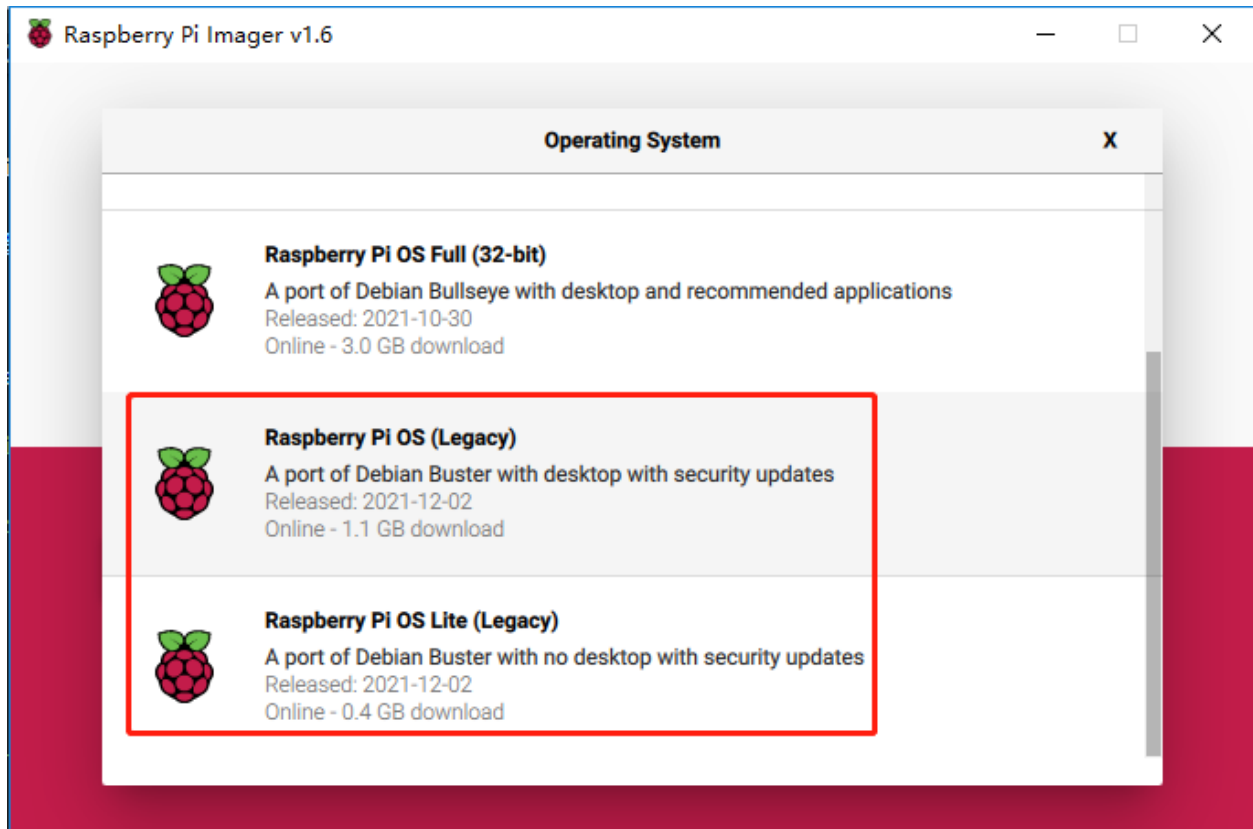
将 Micro SD 卡插入电脑或笔记本电脑的 SD 卡槽中。如果计算机或笔记本电脑只有 SD 卡的全尺寸端口，请使用 Micro SD 卡适配器。

**警告：**树莓派 OS 升级到 **Debian Bullseye** 后，会导致有些功能不能使用，建议还是继续使用 **Debian Buster** 版本。

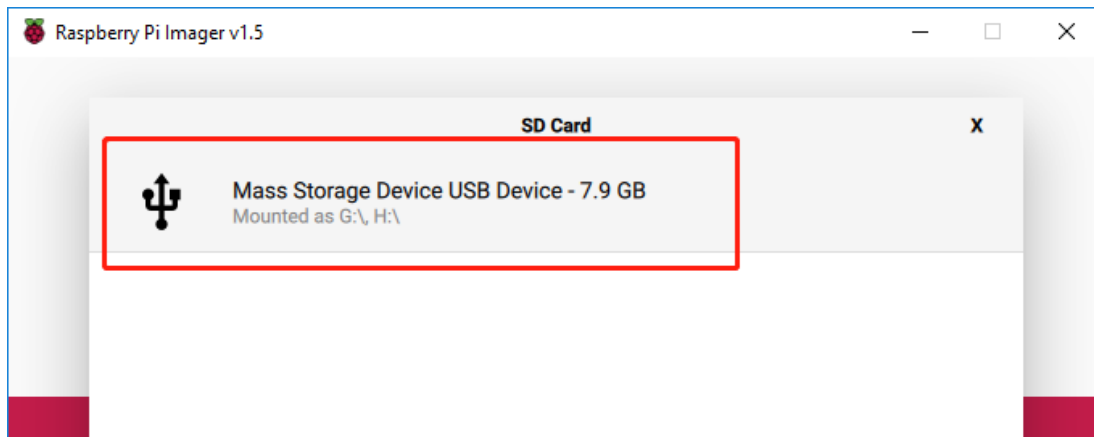
在 Raspberry Pi Imager 中，点击 **CHOOSE OS -> 树莓派 OS(other)**。



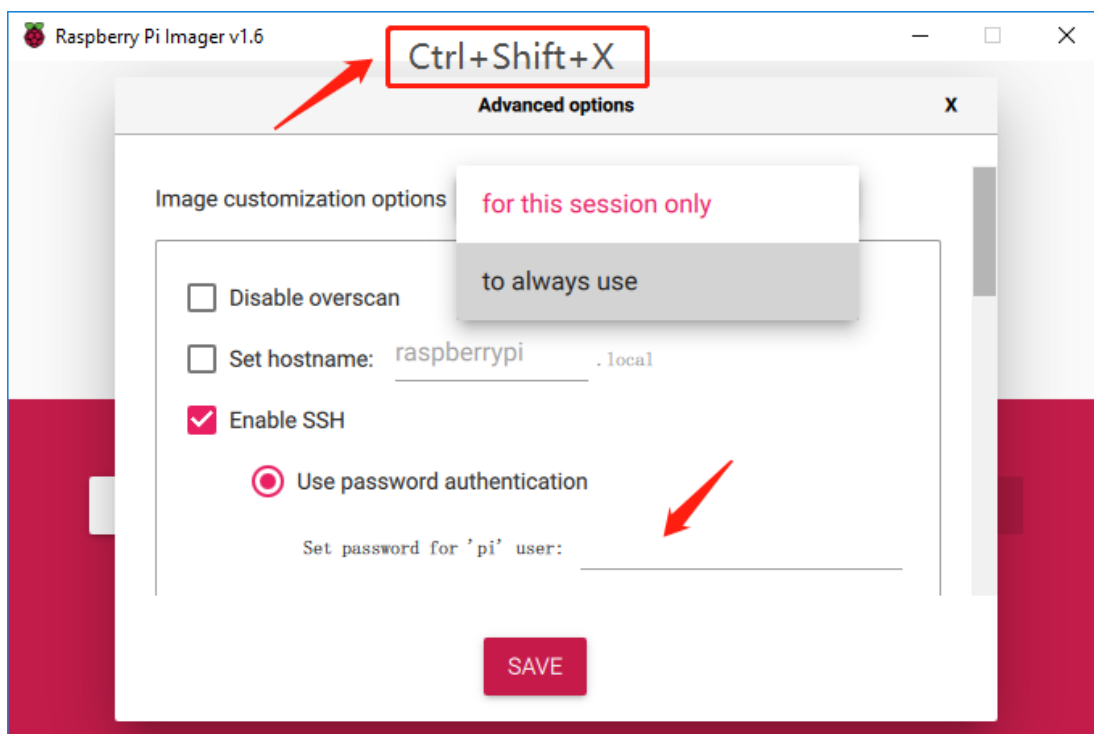
将新打开的页下拉到最后面，你会看到 **树莓派 OS(Legacy)** 和 **树莓派 OS Lite(Legacy)**，这 2 个是对 Debian Buster 安全更新，它们之间的区别是带不带桌面。建议安装 **树莓派 OS(Legacy)**，这个带桌面的系统。



选择您正在使用的 SD 卡。

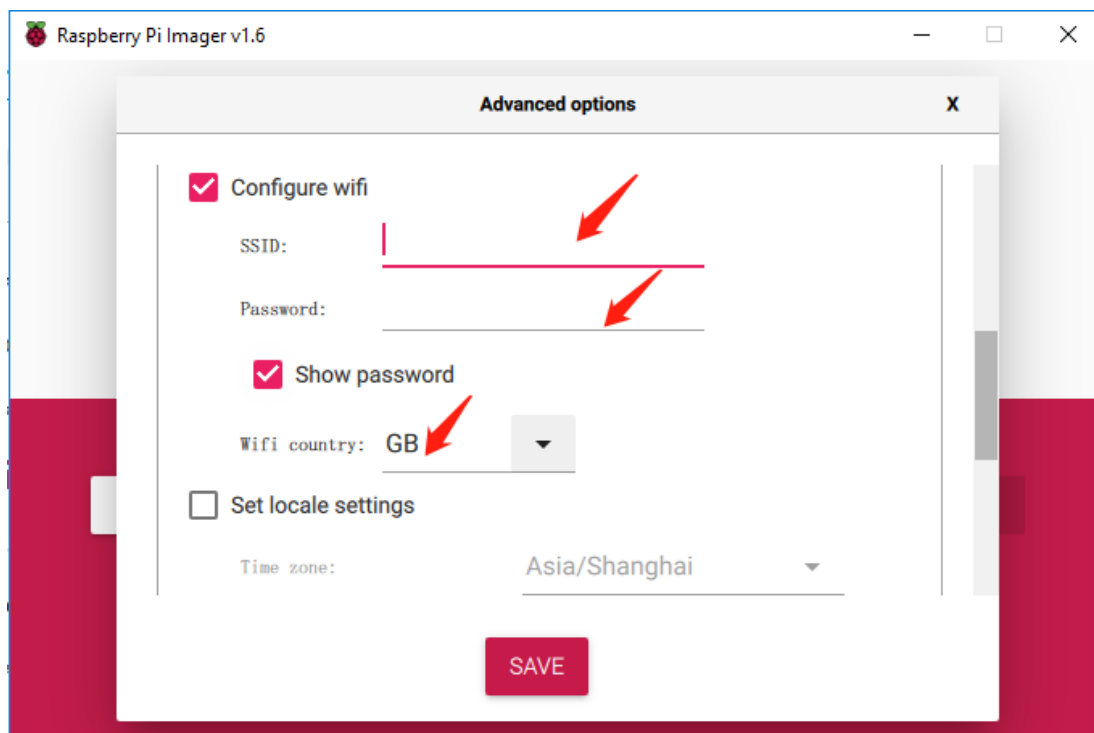


按 **Ctrl+Shift+X** 打开高级选项页面，启用 SSH 和配置 wifi，这 2 项必须设置，其他取决于您的选择。您可以选择始终使用此图像自定义选项。

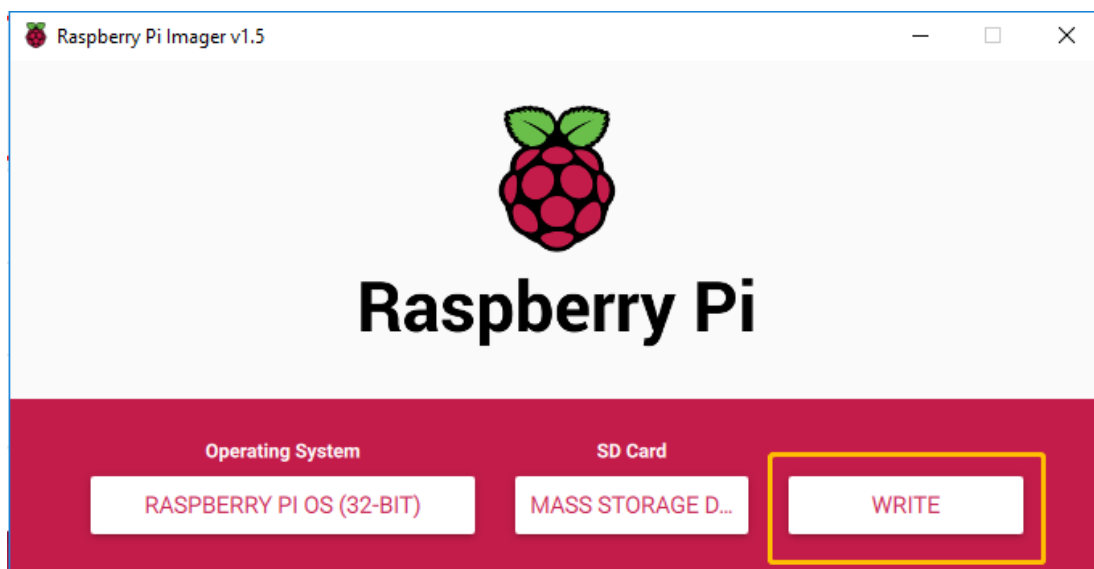


向下滚动以配置 WiFi 设置，然后单击 **SAVE**。

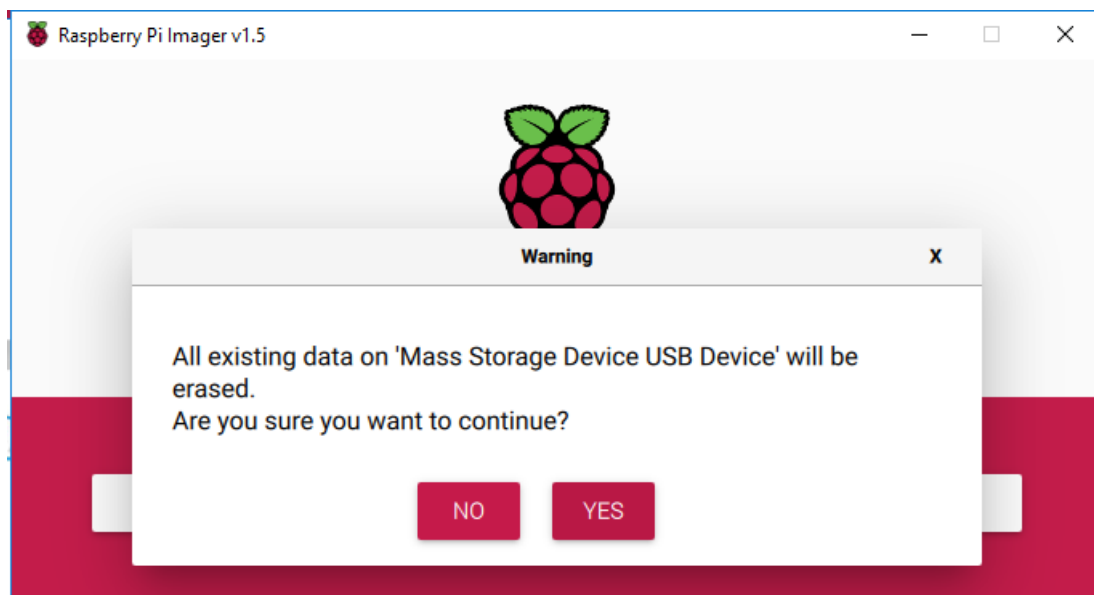
**注解：** **WiFi country** 的条目应设置为指定给该国家/地区的指定的两个字母的 ISO/IEC alpha2 代码。请参考以下链接：[https://en.wikipedia.org/wiki/ISO\\_3166-1\\_alpha-2#Officially\\_assigned\\_code\\_elements](https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2#Officially_assigned_code_elements)。



单击 **WRITE** 按钮。

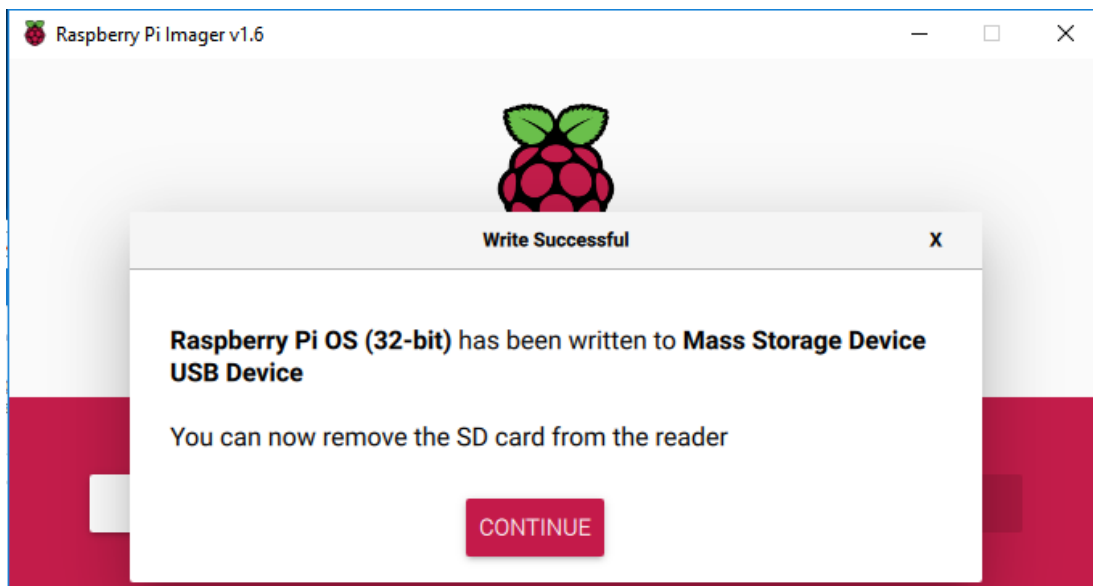


Imager 将覆盖 SD 卡上已有的所有文件。如果 SD 卡上有任何文件，请在单击是之前备份这些文件。



Imager 完成在 SD 卡上安装操作系统后，单击继续以完成安装。





将 Micro SD 卡插入 RasPad 的 Micro SD 端口，然后按住电源按钮三秒以上，RasPad 将开启。  
电源按钮在右侧。按住 3 秒以上可打开或关闭 RasPad。短按将打开或关闭屏幕。

**注解：**使用前建议给为 *RasPad* 充电。



**警告：**SD 卡插槽采用卡扣式连接到后盖。在拆卸或更换后盖之前，请先取出 Micro SD 卡，以免损坏 Micro SD 卡和 RasPad 的内部按键板。

## 3.2 在 RasPad 上安装虚拟键盘

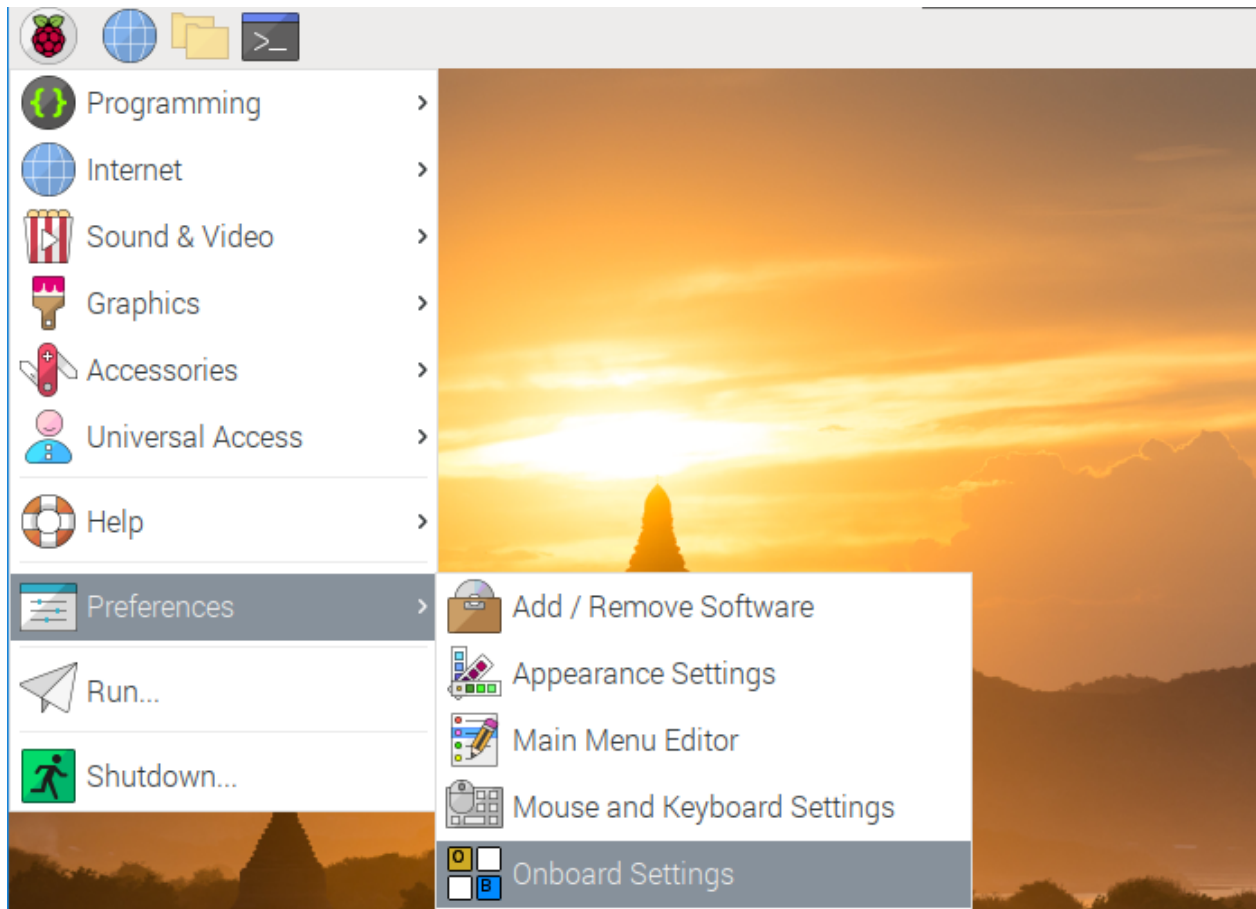
当您使用 RasPad 等触控面板时，可以外接键盘来帮助您进行文字输入操作，但最好安装虚拟键盘。

使用以下命令安装所需的软件。

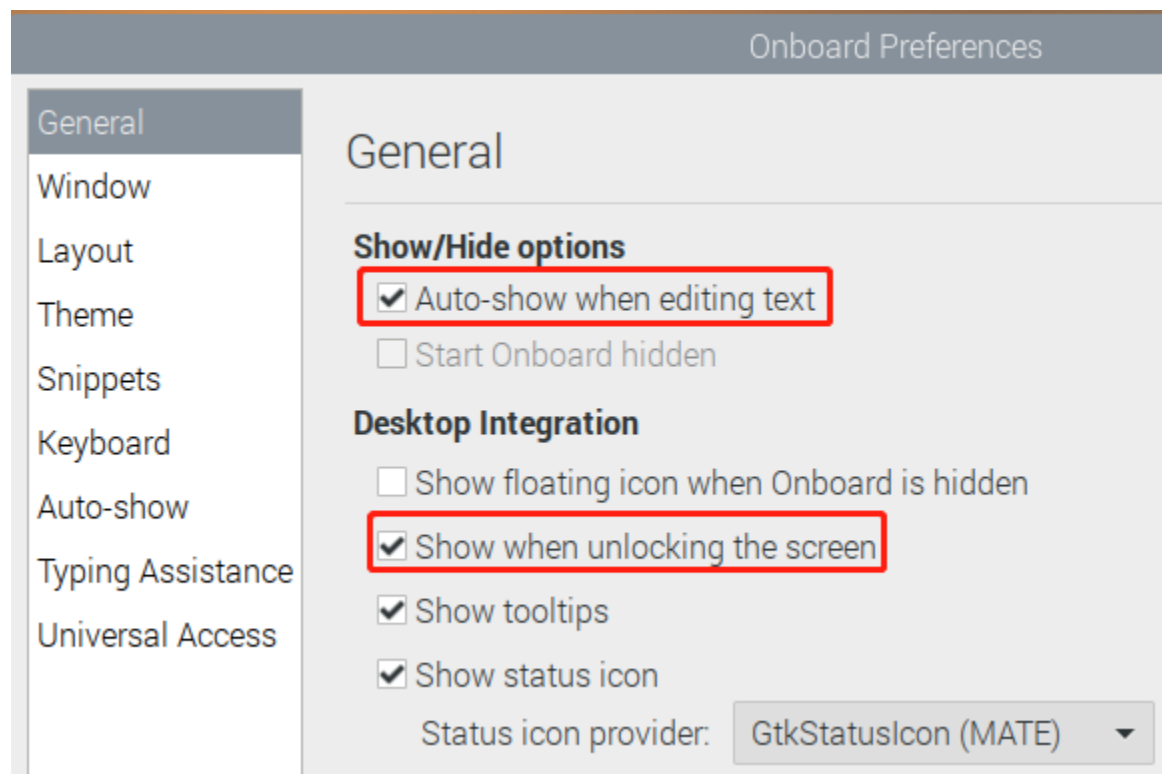
```
sudo apt install onboard -y  
sudo apt install at-spi2-core
```

为了让虚拟键盘有更好的效果，还需要做进一步的设置。

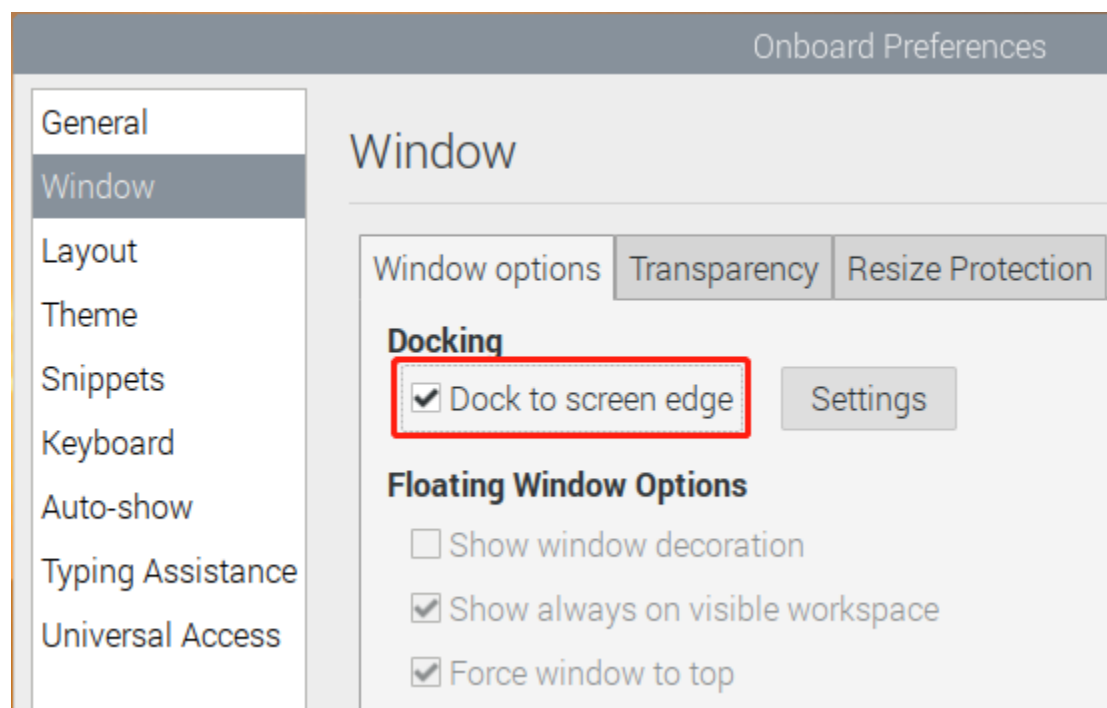
单击左上角的树莓派图标并选择 **Preferences -> Onboard Settings**。



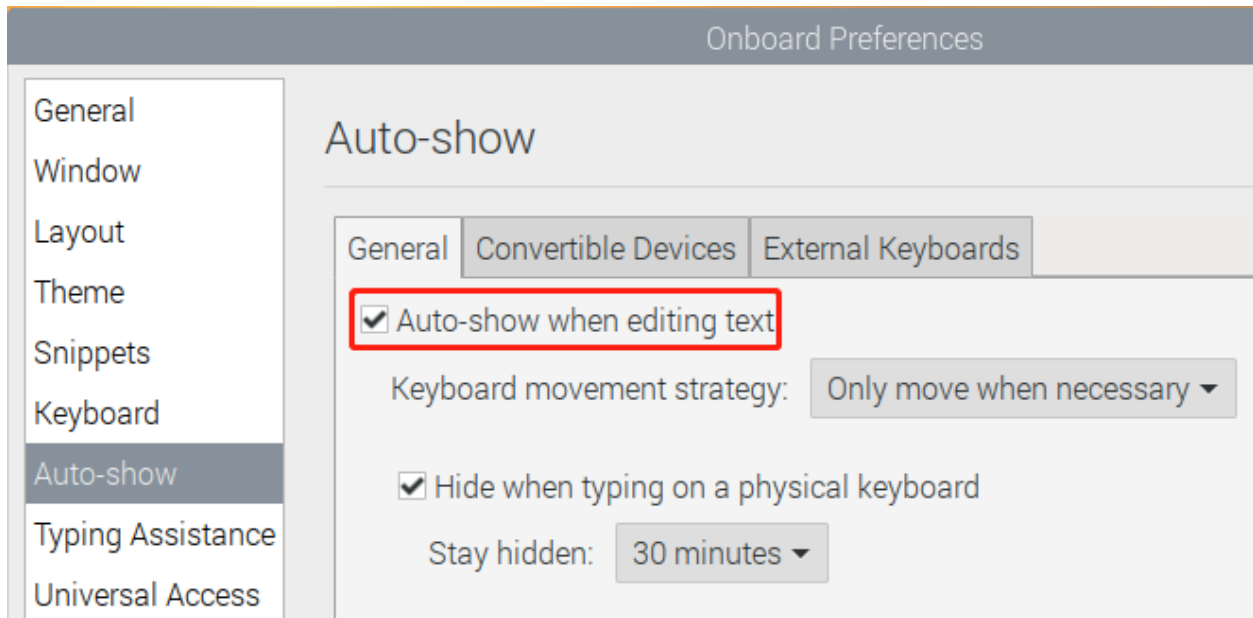
在 **General** 选项中，检查以下两项。当勾选 **Automatically display when editing text** 时，会提示重启，所有设置完成后即可重启。



在 **Window** 选项中, 选中 **Dock to screen edge**。

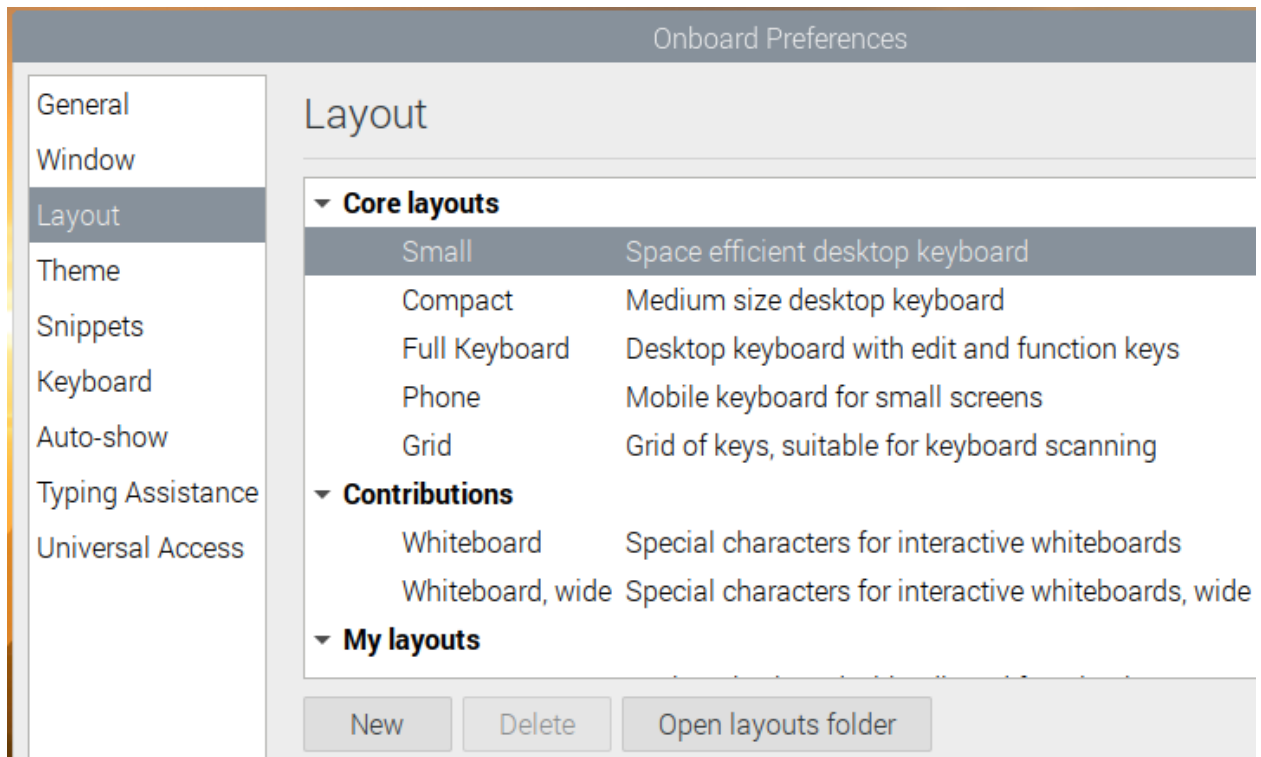


在 **Auto-show** 选项中, 再次选中 **Auto-show when editing text**。

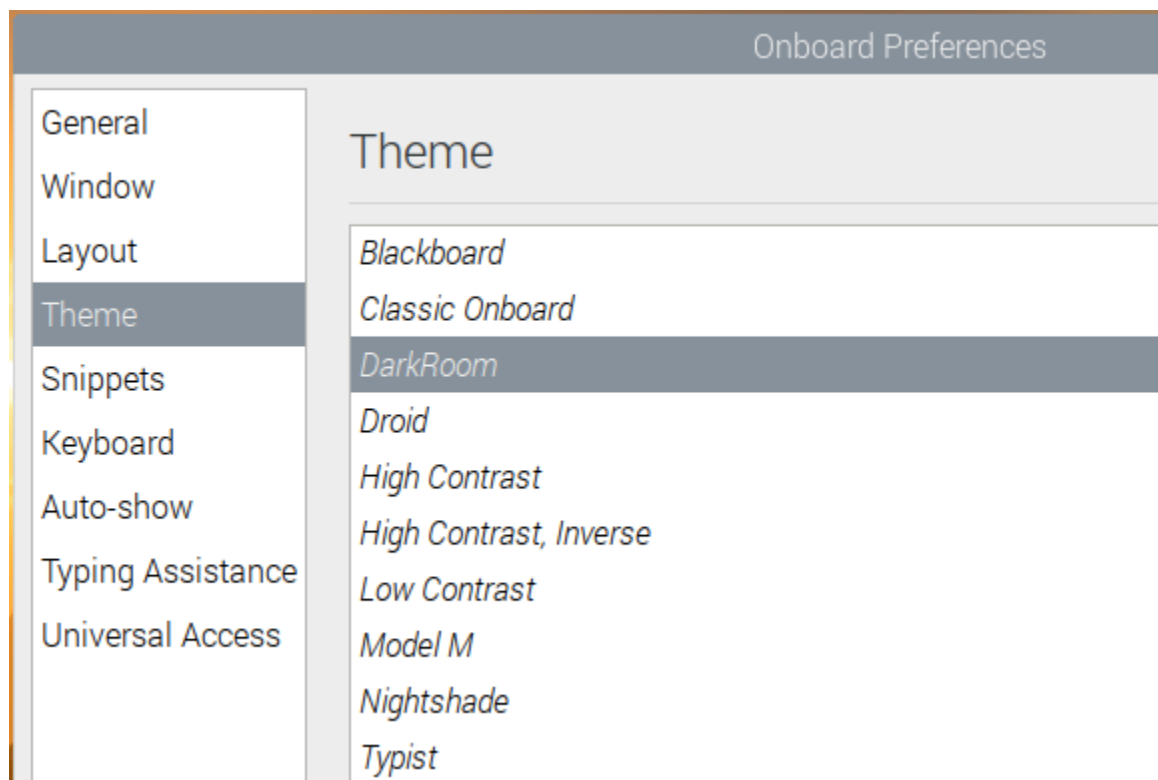


接下来的 2 项是可选的，图中勾选的是我们推荐的，其他的也可以勾选。

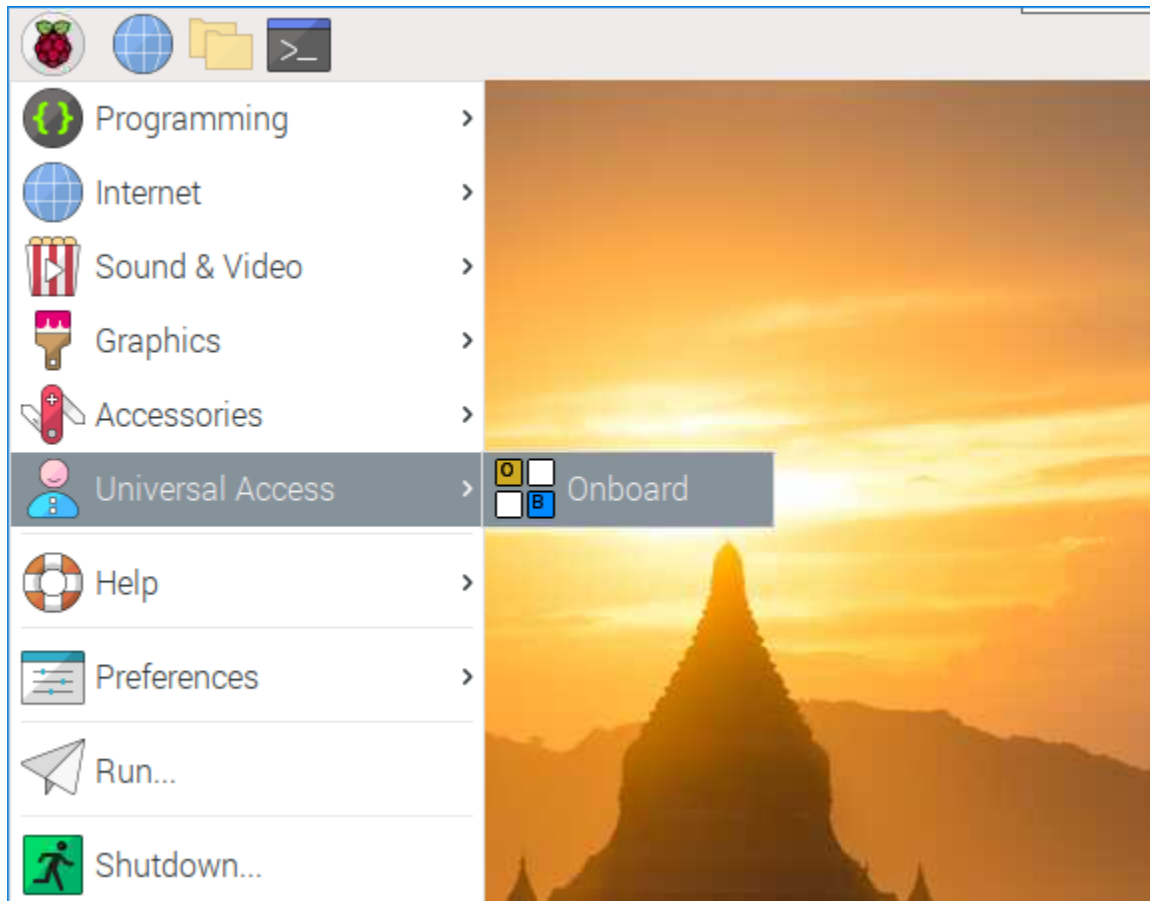
在 **Layout** 选项中, 建议使用 **Small**。



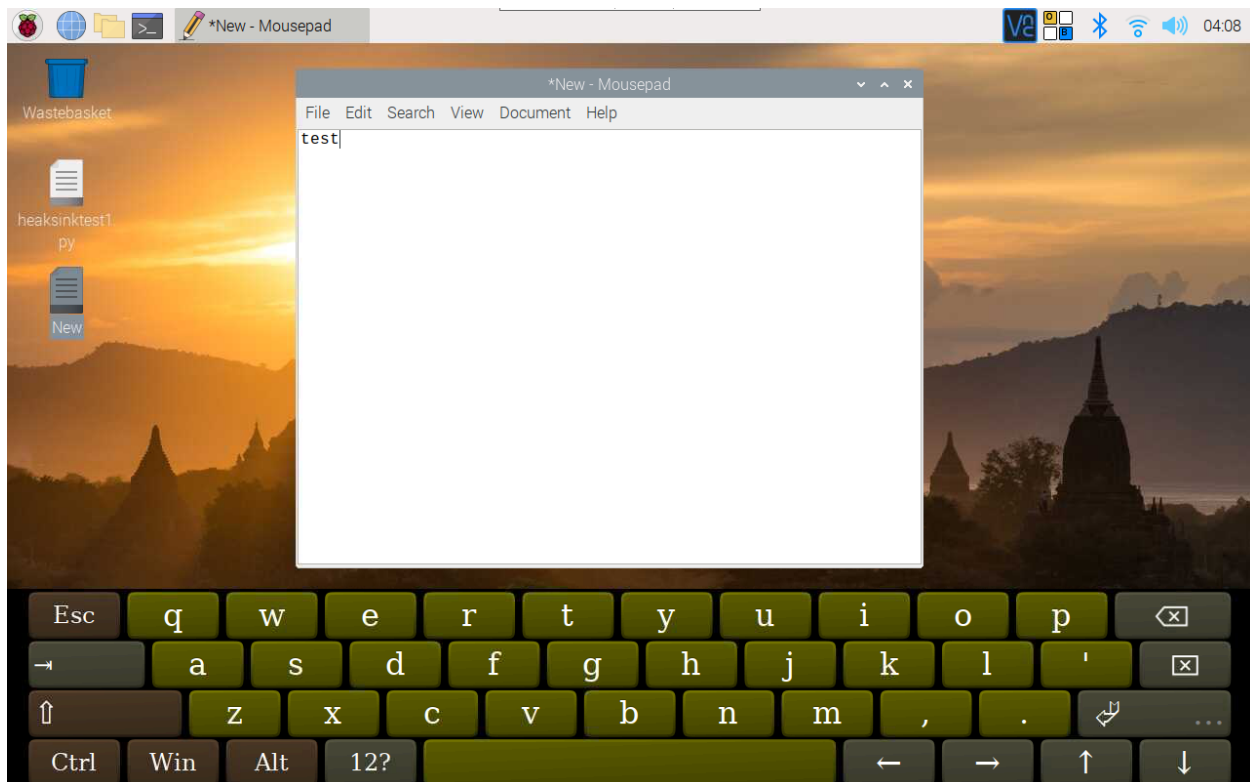
在 **Theme** 选项中, 建议使用 **DarkRoom**。



设置完成后，重启 RasPad。每次重启 RasPad，都需要点击 **General Access -> Onboard** 来启用虚拟键盘。



现在您可以使用此键盘来编辑您的文件或代码。



### 3.3 右键单击 RasPad

触摸屏平板电脑和显示器使您可以轻松地用手指或手写笔执行简单的导航任务，但在某些时候，您可能希望使用右键单击命令来快速访问特定于上下文的快捷方式。

我们用 `evdev-rce` 来使 RasPad 的右键单击命令仍然可用。

输入以下命令安装所需软件。

```
sudo apt install build-essential libevdev2 libevdev-dev -y
git clone 'https://github.com/PeterCxy/evdev-right-click-emulation.git'
cd 'evdev-right-click-emulation'
```

输入以下命令进行构建。

```
make all
```

将文件复制到 `/usr` 目录中。

```
sudo cp 'out/evdev-rce' '/usr/local/bin/'
```

使其可执行。

```
sudo chmod +x '/usr/local/bin/evdev-rce'
```

修改 `/etc/rc.local` 使 `boot-up` 可用。

```
sudo nano /etc/rc.local
```



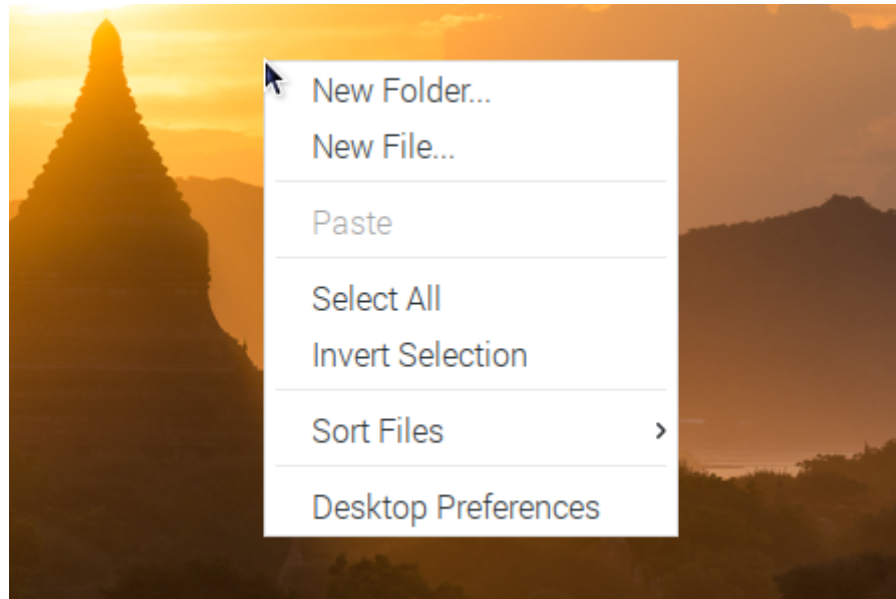
进到 `rc.local` 文件之后, 在 `exit 0` 前面添加如下命令。

```
sudo /usr/local/bin/evdev-rce &
```

按 `Ctrl+C` -> `Y` 退出并保存 `rc.local` 文件, 然后运行命令 `sudo reboot` 以重新启动 RasPad.

```
sudo reboot
```

重启后可以长按 RasPad 桌面, 看看是否出现右键功能。



### 3.4 安装旋转屏幕服务

在组装 RasPad 时将加速度模块插入到树莓派中, 现在我们编写一个脚本使 RasPad 能够自动旋转屏幕。

该脚本目前支持以下系统。

**目前支持**

-树莓派 OS - Twister OS - Ubuntu Desktop 21.04

**运行以下命令进行安装。**

```
git clone https://github.com/raspad-tablet/raspad-auto-rotator
cd raspad-auto-rotator
sudo python3 install.py
```

安装完成后, 可能会提示您重启, 如果是这样, 重启 RasPad 后自动旋转功能会起作用。如果没有提示重启, 自动旋转功能也会起作用。

**校准**

第一次安装后, 有些角度可能不会触发自动旋转功能, 请尝试反方向转动一秒钟, 然后再转动回来, 它会自动校准。或者在三个轴上缓慢旋转 RasPad 720 度, 它也会自动校准。

如果无论您如何旋转和校准它仍然无法在某些角度工作, 在这种情况下, 尝试使用命令重新校准它。

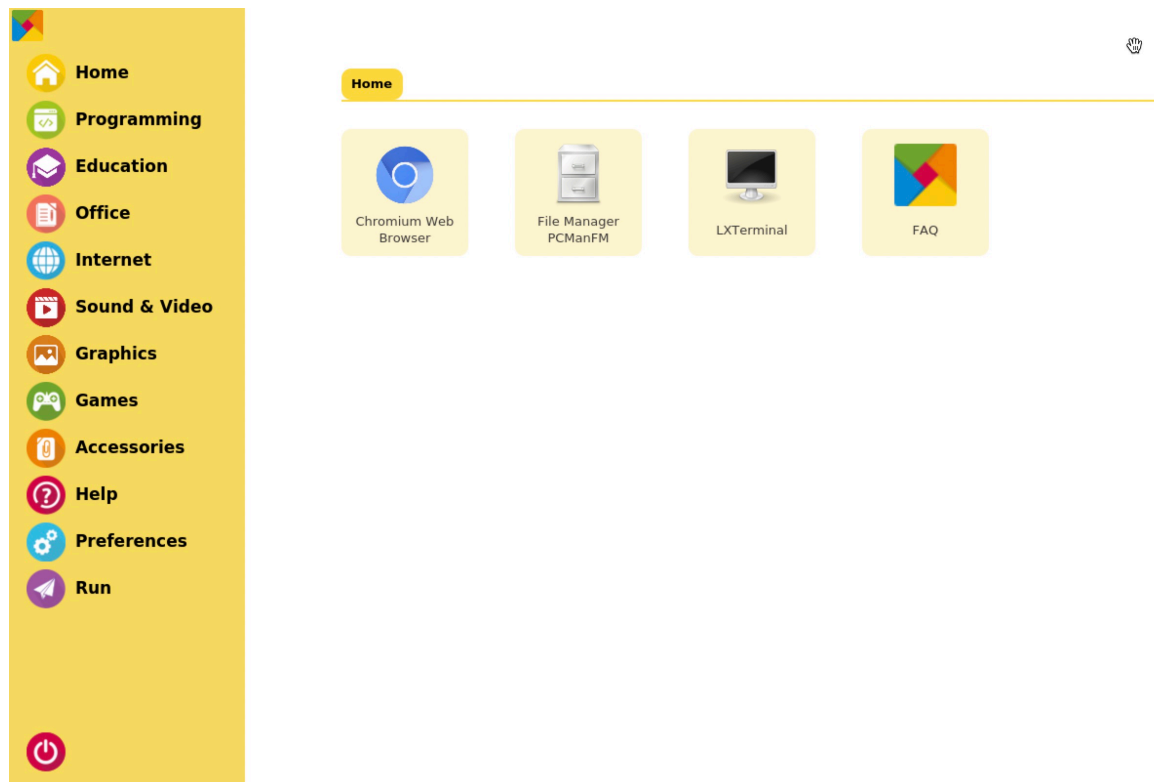


```
raspad-auto-rotator reset
```

运行上述命令后，自动旋转器将重新启动。现在根据前面提到的 2 种方法重新校准。

### 4.1 介绍

RasPad Launcher 是一款模拟启动器菜单的开源软件，专注于改善用户使用 RasPad 或其他触摸屏的触摸屏体验。



---

**注解：**如果你想使用树莓派的原始桌面，你可以跳过这一章。

---

## 4.2 安装指南

下载 RasPad Launcher 安装包并解压

```
wget https://github.com/raspad-tablet/raspad-launcher/releases/latest/download/raspad-  
→launcher.zip  
unzip raspad-launcher.zip  
cd raspad-launcher
```

该脚本将安装以下内容

- 带有桌面配置文件的 RasPad Launcher。
- RasPad 常见问题桌面配置文件（只是浏览器 RasPad 常见问题网页的快速图标）。
- 显示 Accl SHIM 的自动旋转。

运行安装脚本。

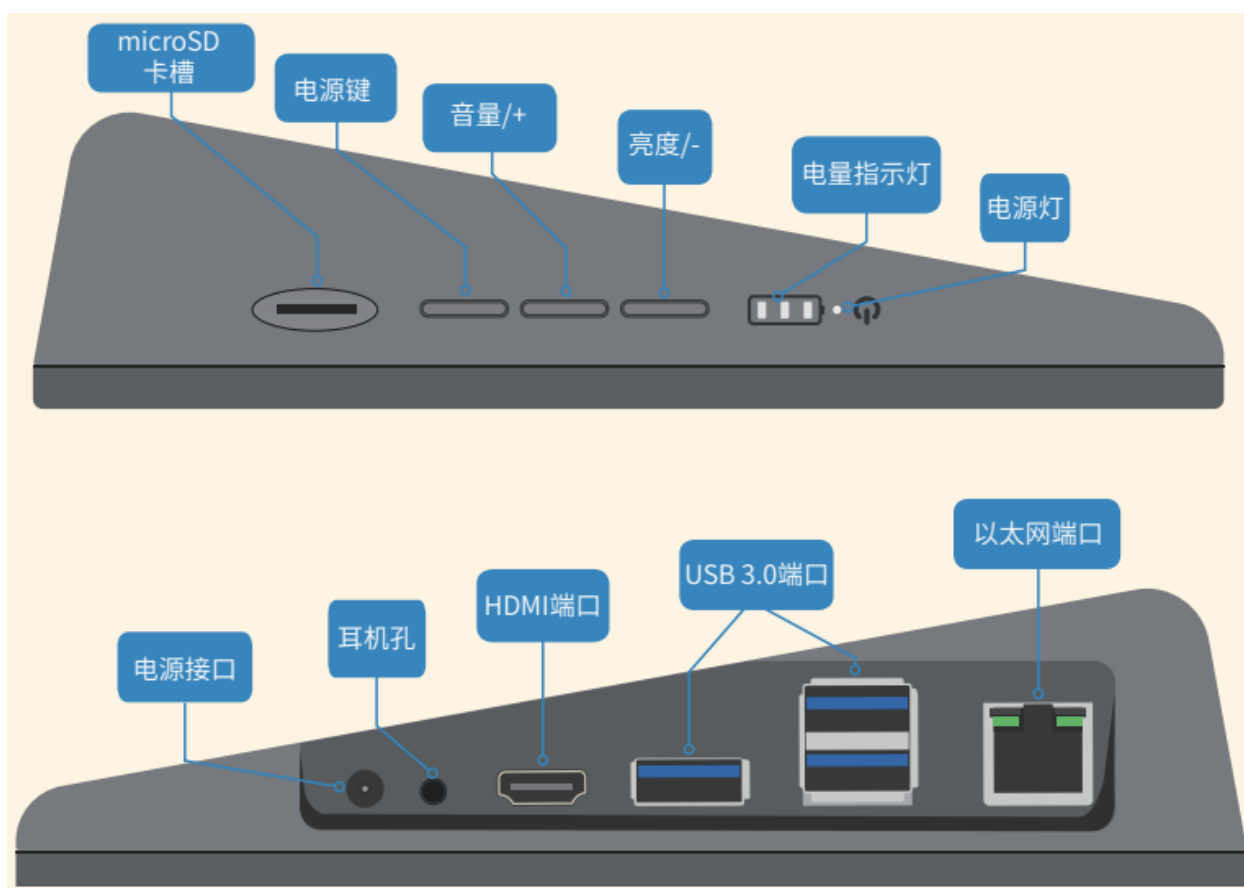
```
chmod +x install  
sudo ./install
```

---

**注解：**要手动安装 RasPad Launcher 和特定组件，请参阅 [RasPad Launcher](#)。

---

## 接口介绍

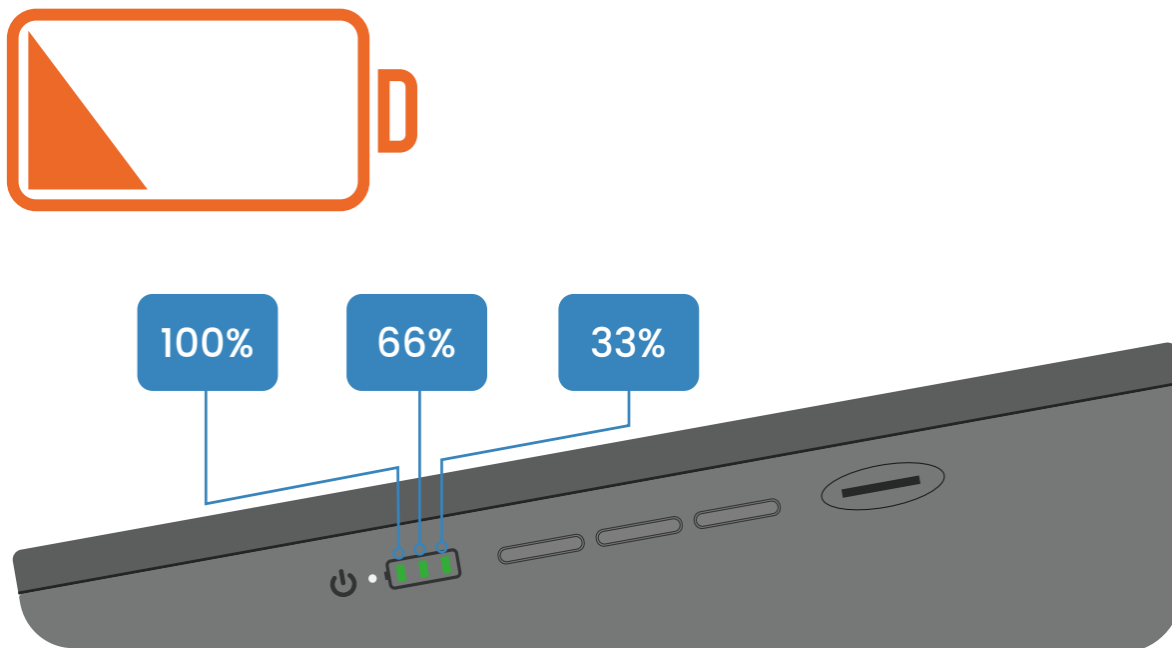


- **microSD 卡槽:** SD 卡插槽是连接到后盖的卡入式。在拆卸或更换后盖之前，请先取出 micro-SD 卡，以免损坏 micro-SD 卡和 RasPad 的内部按键板。
- **电源键:** 按住三秒钟以上可打开或关闭 RasPad。短按将打开或关闭屏幕。

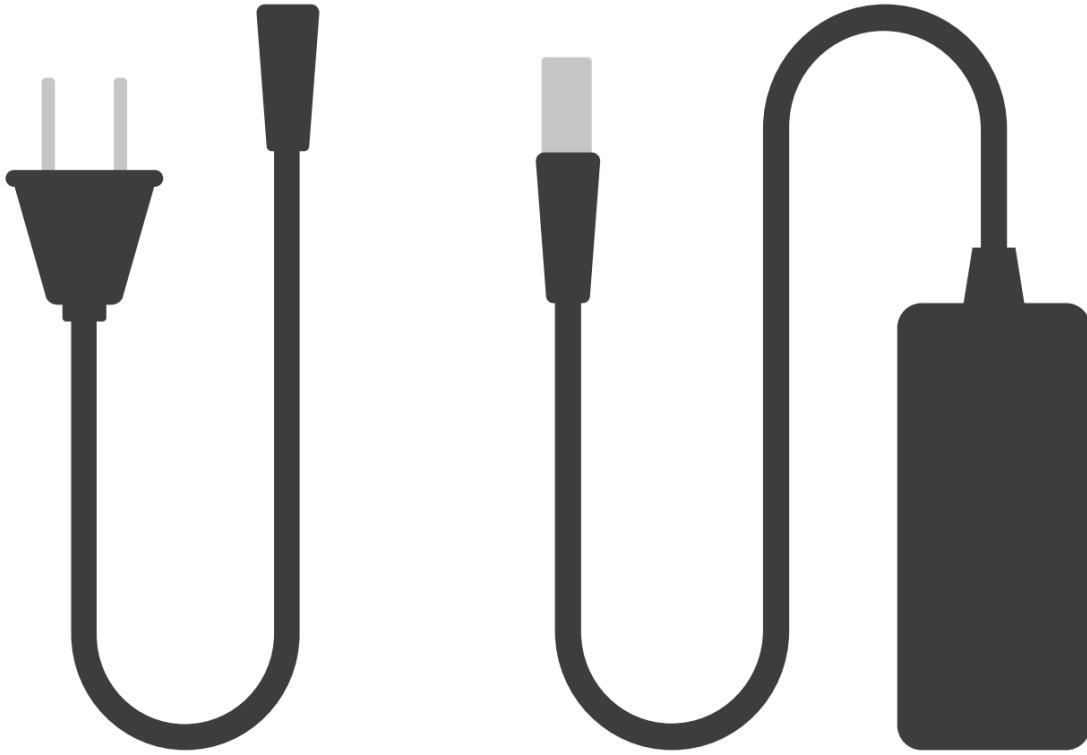
- **音量/+**: 按下进入音量调节状态；如果先按下 **亮度/-**，按下此键可用于提高亮度。
- **亮度/-**: 按下进入亮度调节状态；如果先按下 **音量/+**，按下此按键可用于降低音量。
- **电量指示灯**: 亮绿色。充电时，相应的 LED 闪烁；充满电后，所有 3 个 LED 均亮起。
- **电源灯**: 当 RasPad 开启时，它会亮起红色。
- **以太网端口**: 标准以太网端口。
- **USB 3.0 端口**: 标准 USB 3.0 端口。
- **HDMI 端口**: 通过此端口可用作扩展屏幕。
- **耳机孔**: 标准耳机插孔。
- **电源接口**: 15V/2A DC 端口。

## 5.1 为 RasPad 充电

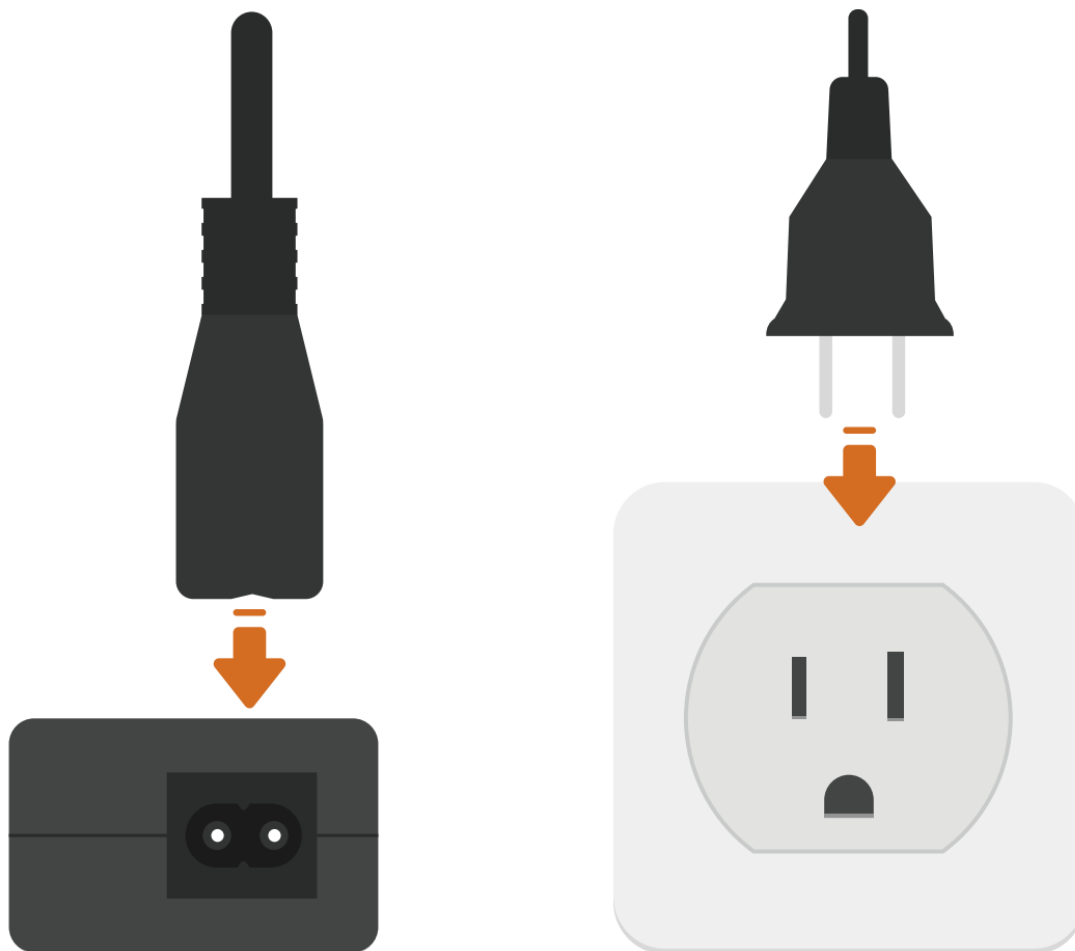
绿色 LED 的数量表示电池百分比。当出现低电量警告图标时，表示需要为 RasPad 充电。



取出电源适配器和电源线。



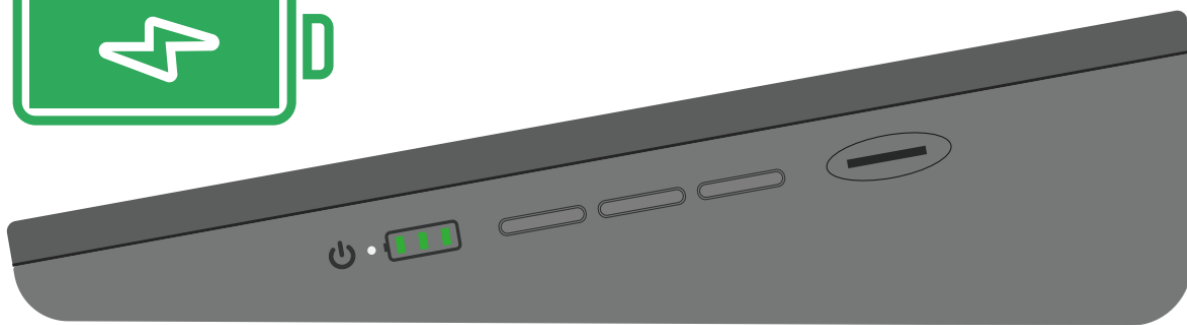
将电源线的一侧插入电源适配器，另一侧插入插座。



将电源适配器插入 RasPad 电源端口。

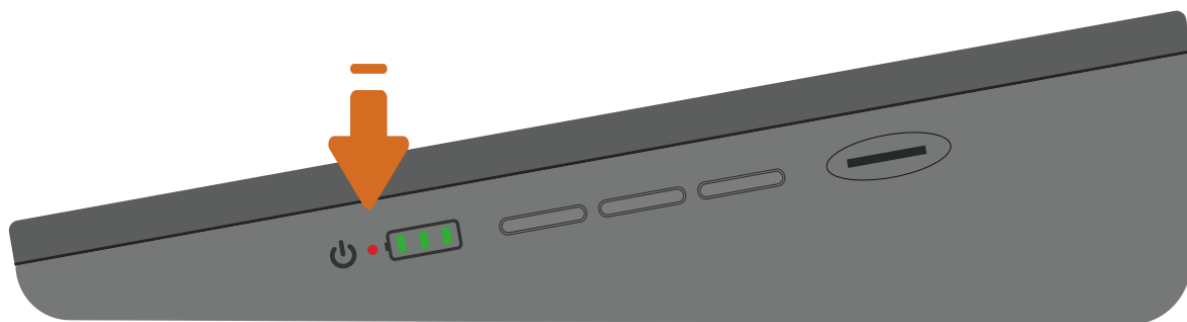


此时，其中一个绿色 LED 可能会闪烁，并且会出现充电图标，表示 RasPad 正在充电。



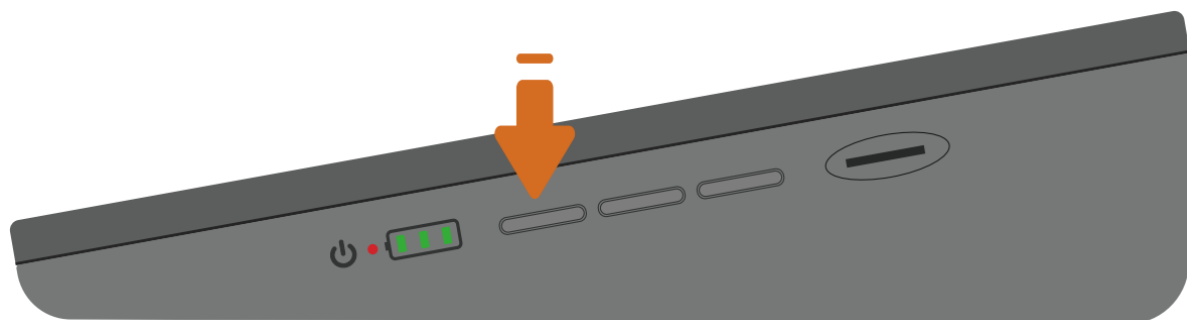
## 5.2 音量和亮度设置

电源按键旁边的红色 LED 现在应该亮起，表示 RasPad 已开机。



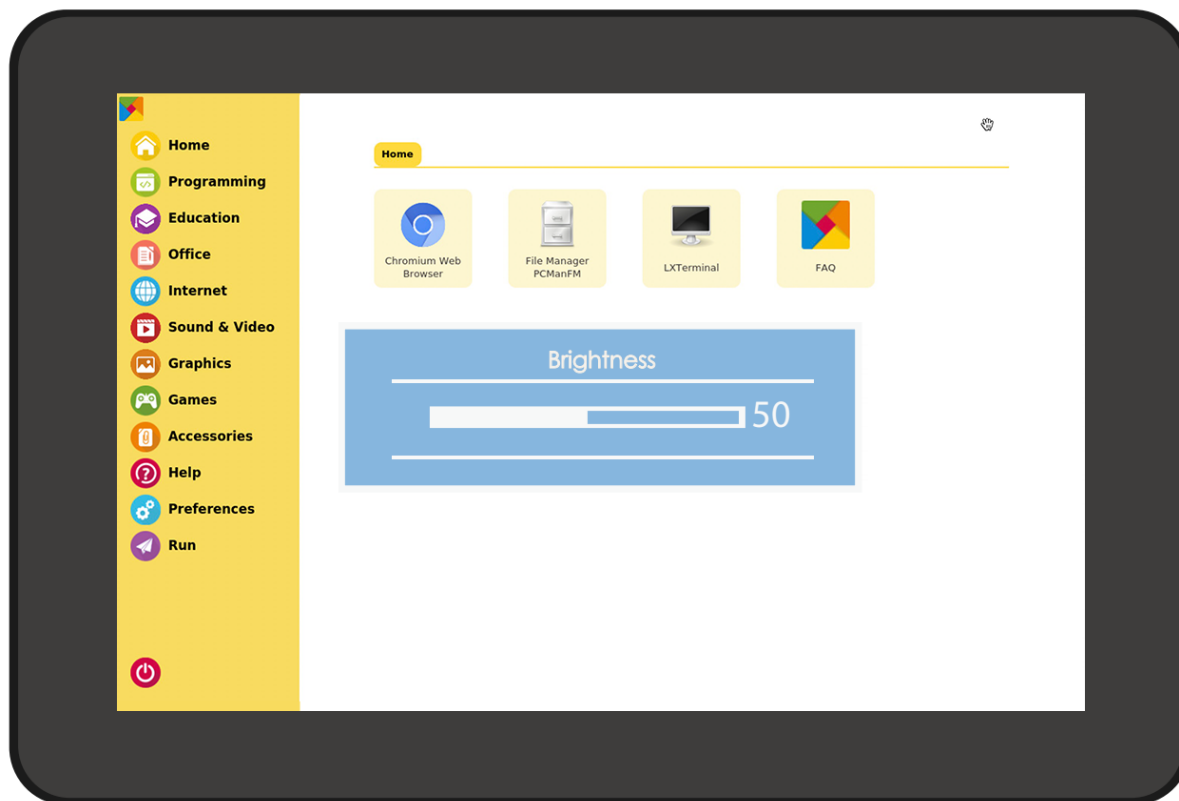
此时，应该启动 RasPad。

如果屏幕太亮，短按亮度键进入屏幕亮度调节页面。

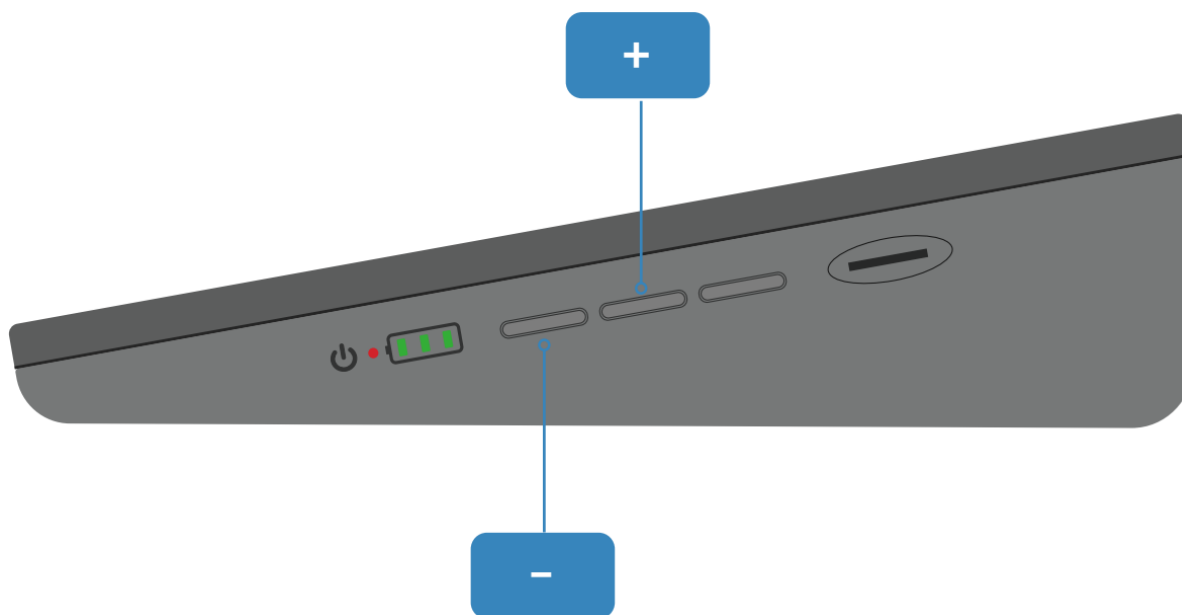


屏幕上会出现亮度调整条。





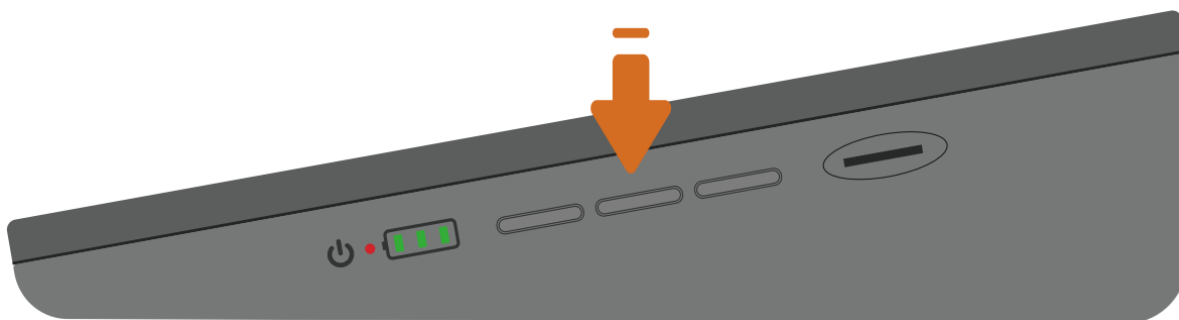
可以通过这两个按钮调节亮度。



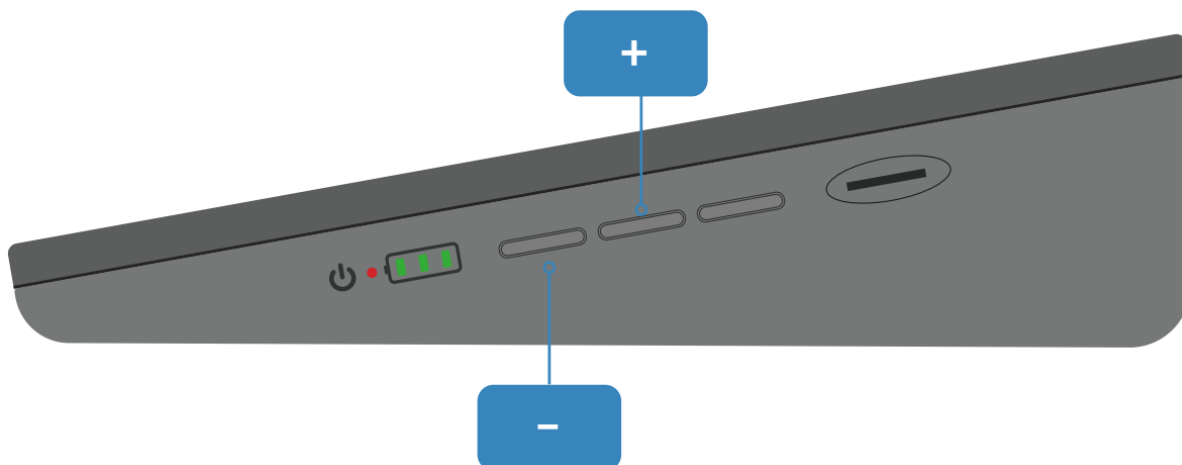
无操作三秒后，屏幕亮度条自动消失。亮度不宜过高或过低，否则对眼睛有害。



要调节音量，按一次音量键进入音量调节页面。

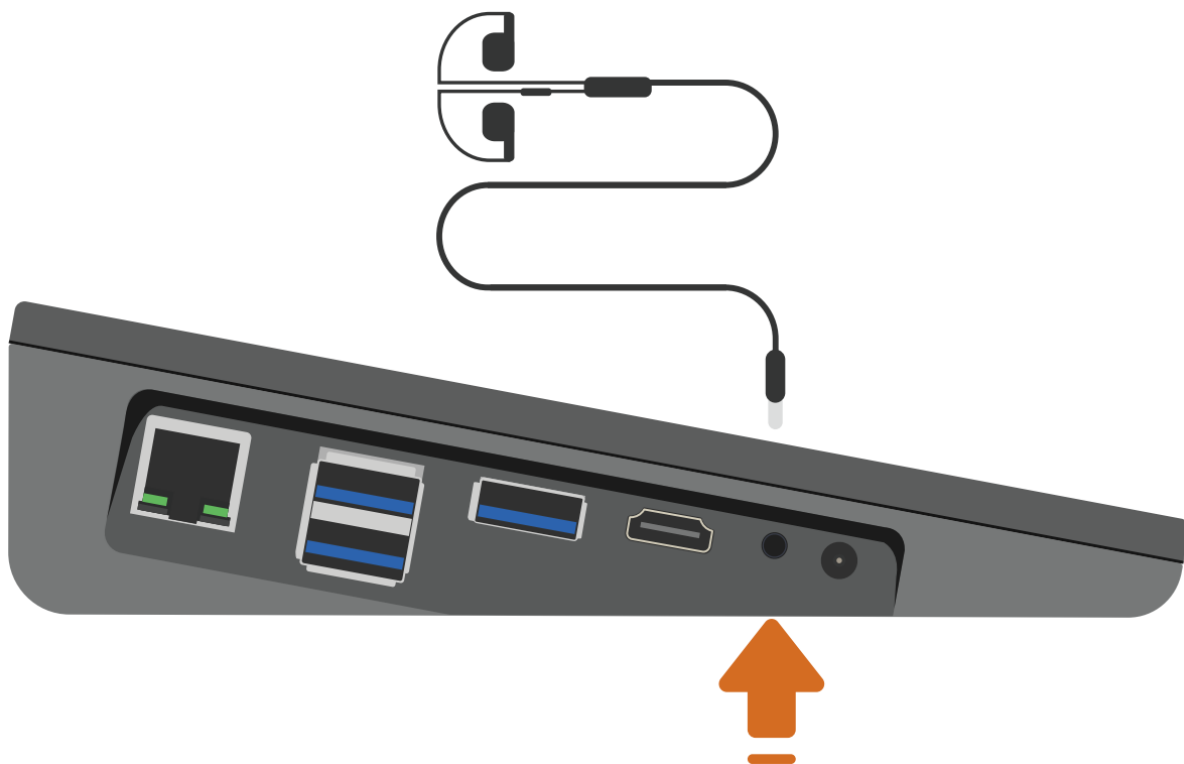


然后按这两个按键调节音量。

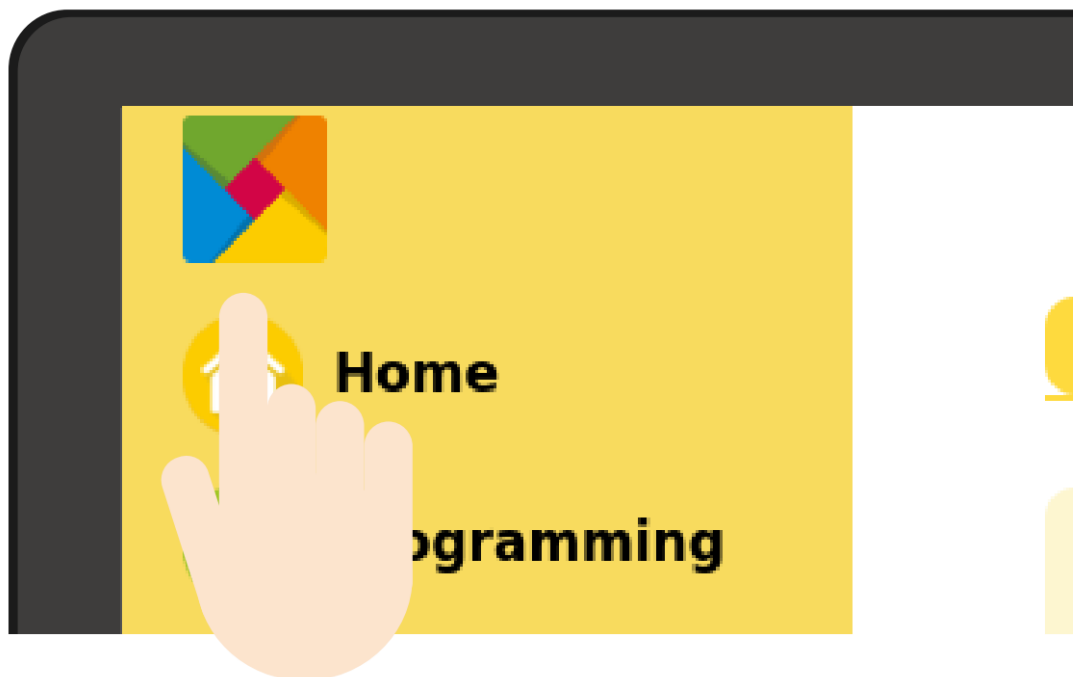


### 5.3 连接耳机或其他屏幕

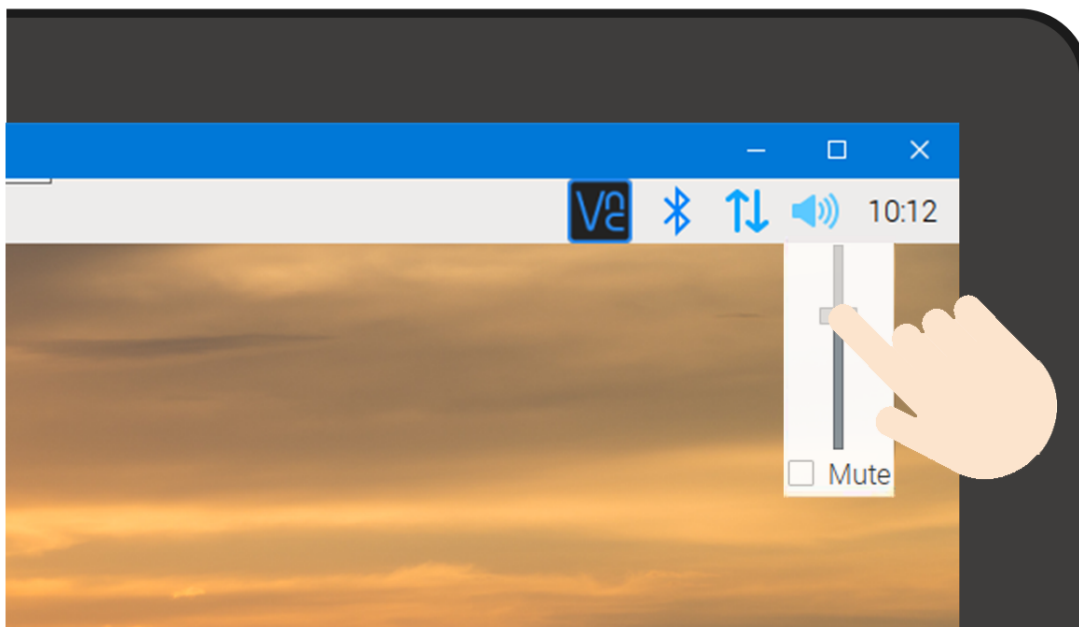
如果需要，请插入耳机。



耳机音量只能在桌面屏幕上调节。通过点击 RasPad Launcher 的左上角菜单图标进入树莓派桌面。



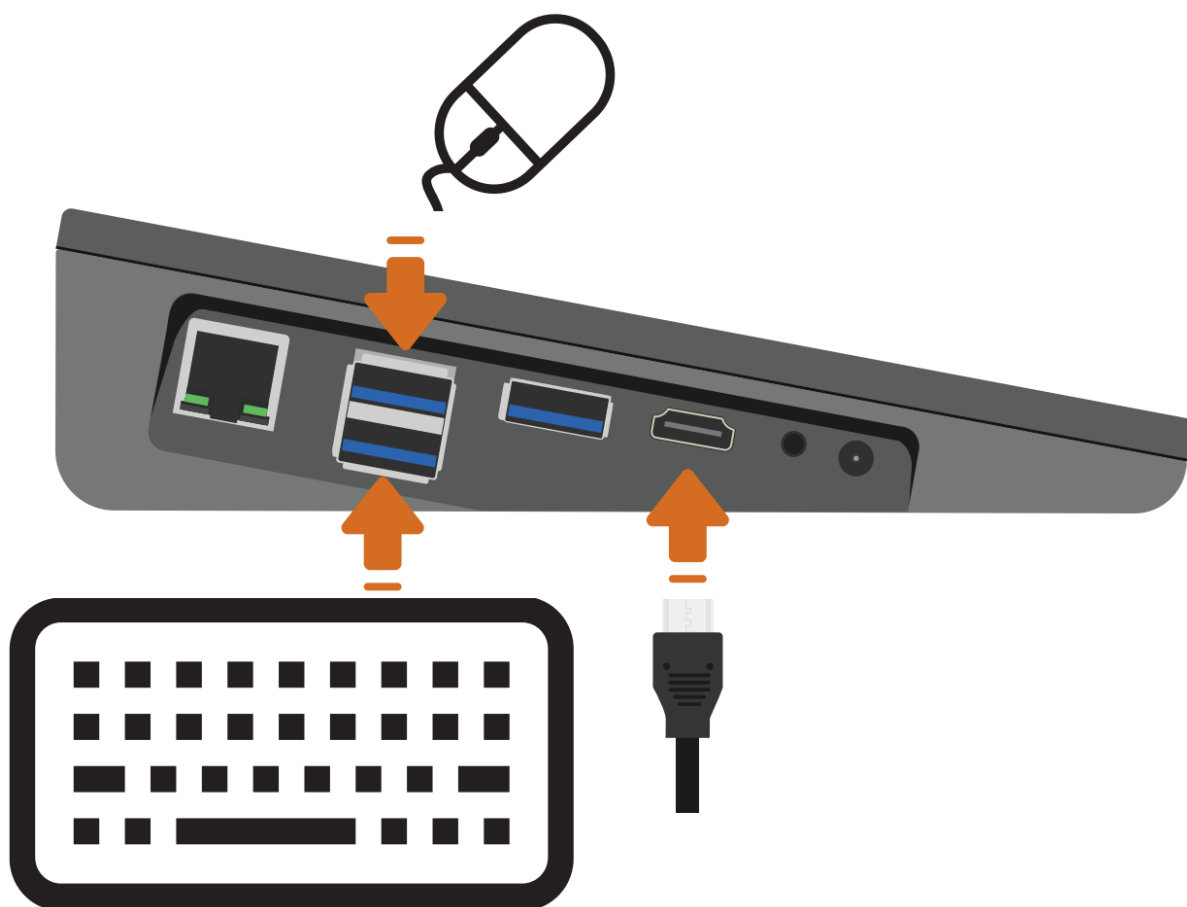
点击扬声器图标，会出现音量控制框。向上或向下滑动音量光标可调节耳机的音量。



**警告：**戴耳机时，不要把音量调得太高。



HDMI 端口可以连接第二台显示器，USB 端口可以连接鼠标和键盘。







要开始使用 RasPad，本章包括一些有趣且具有挑战性的项目可供选择。

## 6.1 多媒体选项

在 RasPad 上观看视频、阅读电子书或玩游戏。

### 6.1.1 看视频

RasPad 有一个内置的视频播放器：VLC 视频播放器。



双击 VLC 视频播放器图标直接观看本地视频，或从谷歌浏览器登录 YouTube 网站观看视频。





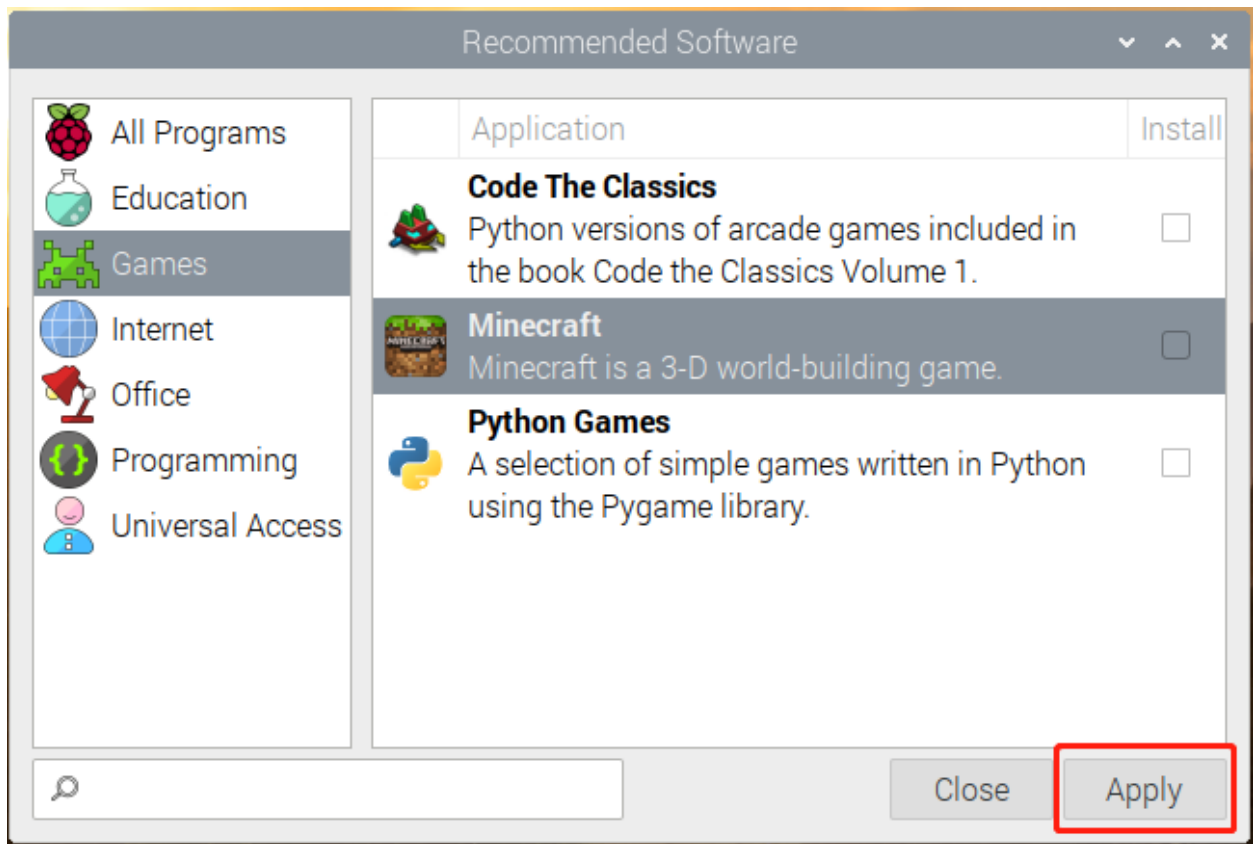
## 6.1.2 阅读电子书

在谷歌浏览器上面在线阅读。

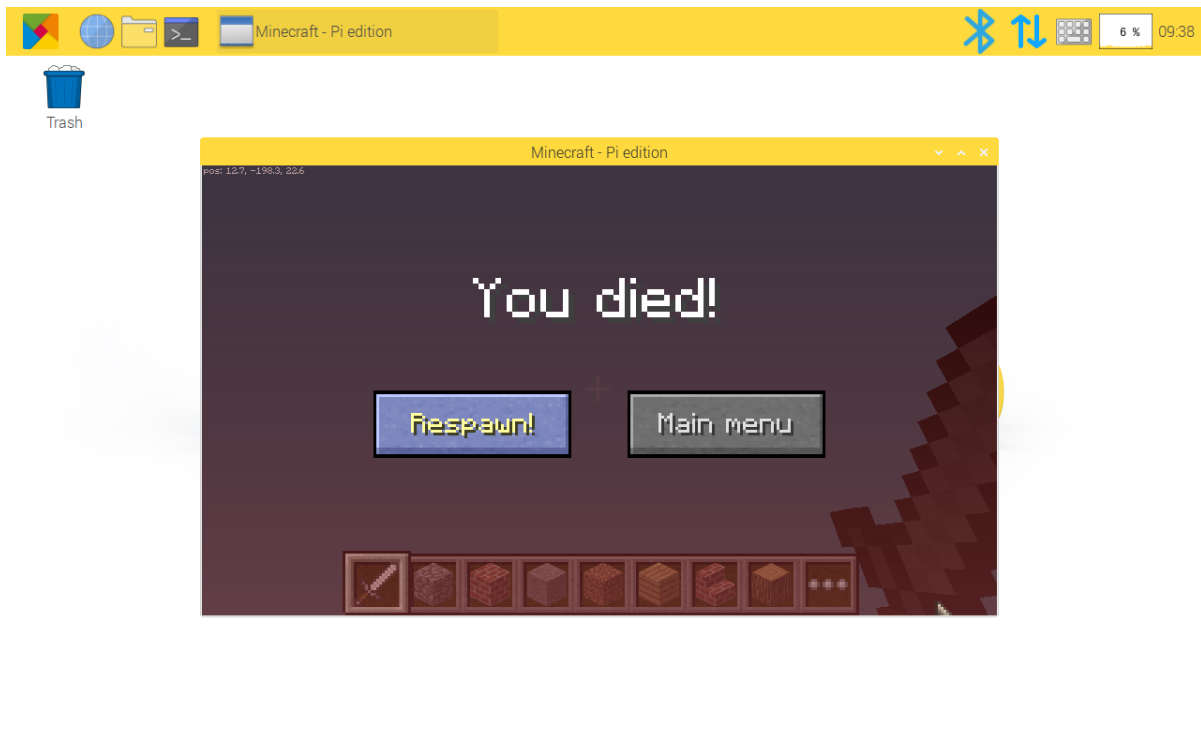


### 6.1.3 玩游戏

点击 **Preferences -> Recommended Software -> Games** 在 RasPad 3 上安装游戏.



比如安装 **Minecraft**, 然后插上键盘鼠标就可以玩了。



## 6.2 复古游戏机

### 6.2.1 描述

您可以将 RasPad 3 变成与您的朋友一起玩的复古游戏机，让我们看看如何做到！



## 6.2.2 所需组件

- A RasPad 3
- 8G+ MicroSD Card
- Micro-SD Card Reader
- Keyboard
- Gamepad

推荐使用树莓派 4 作为主控板，Retro Pie 作为操作系统。

树莓派上传或下载游戏系统和游戏 ROM 需要占用大内存，建议使用大容量 SD 卡，避免配置失败。

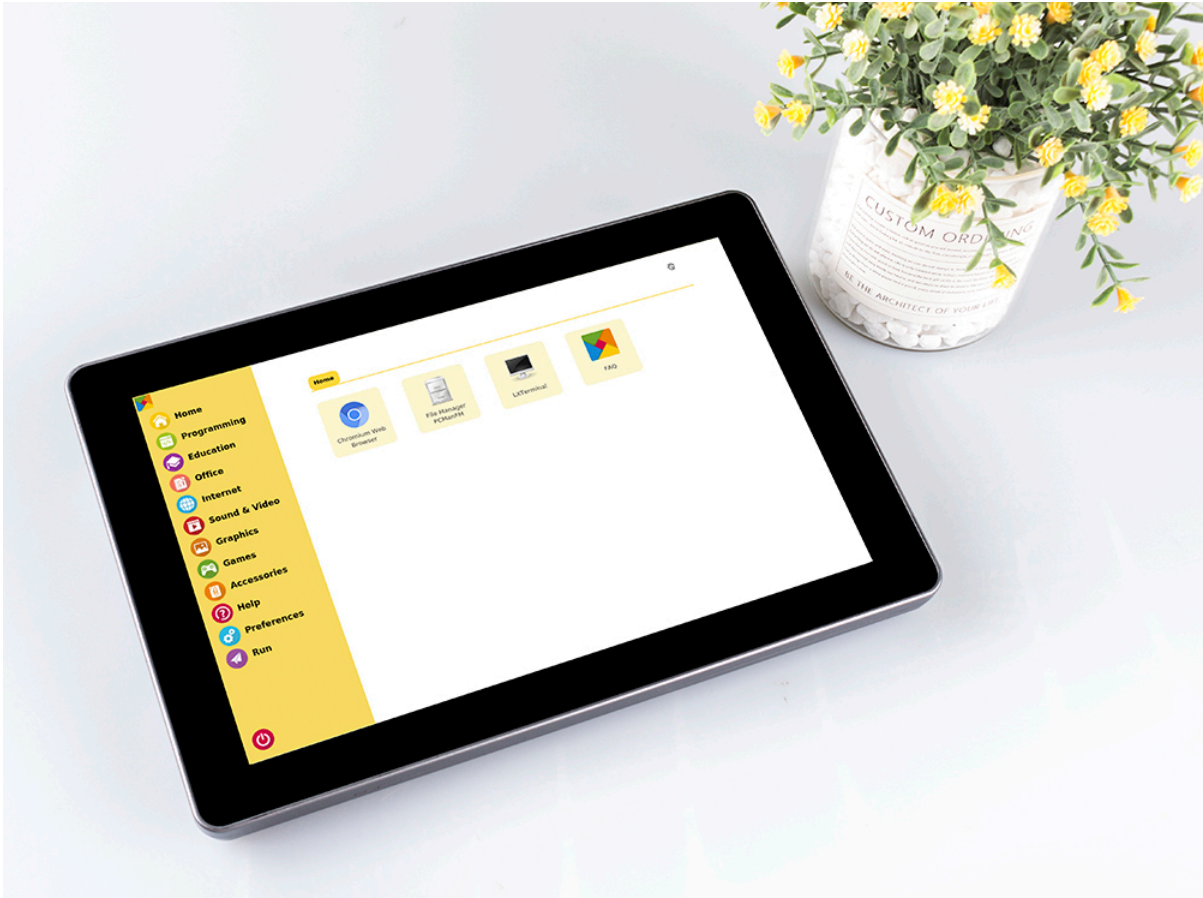


玩游戏时，需要一个游戏手柄和一个键盘。



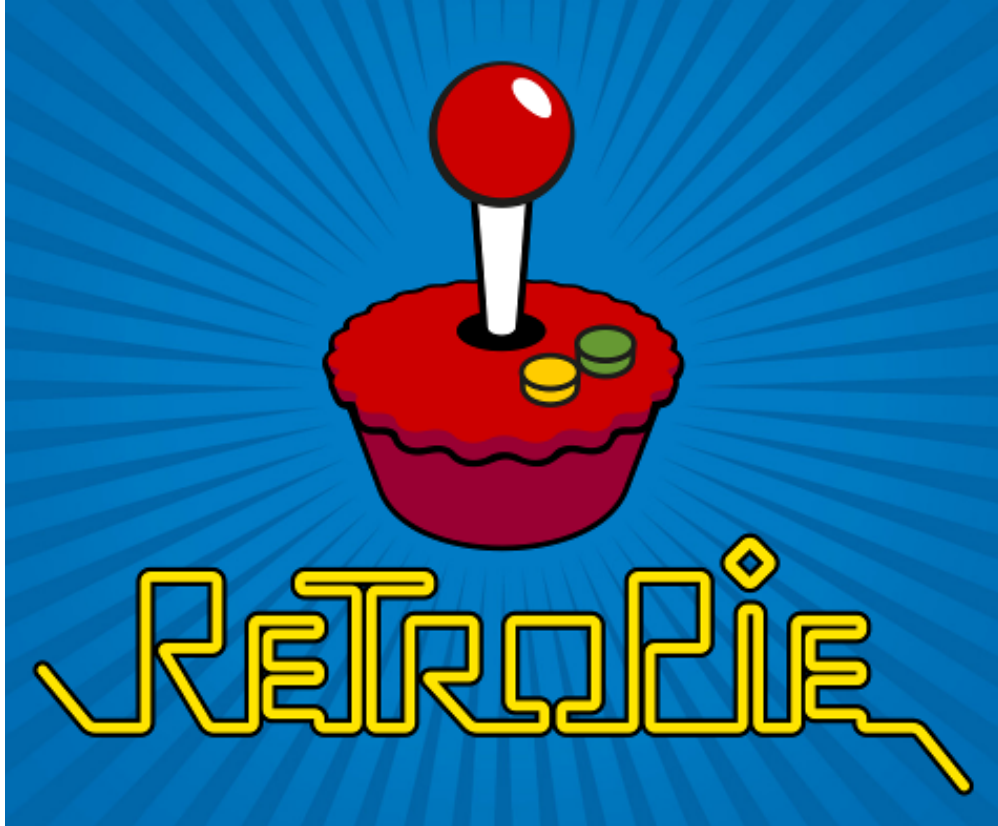
RasPad 配备 1280x800 液晶触摸屏、2 瓦立体声扬声器和三个 USB 3.0 端口，可提供高分辨率和音质，提供出色的游戏体验。





### 6.2.3 游戏系统安装

RetroPie 可让您将树莓派、ODroid C1/C2 或 PC 变成复古游戏机。它建立在 Raspbian OS、Emulation Station、RetroArch 和许多其他项目的基础上，使您能够以最少的设置玩自己喜欢的街机、家用控制台和经典 PC 游戏。



安装 **RetroPie**:

**第 1 步:** 在 [RetroPie 官网](#) 下载兼容树莓派 4 的 SD 镜像。

## Pre-made images for the Raspberry Pi

The latest pre-made image of RetroPie is v4.6 – released April 28, 2020.

Contributions to the project are appreciated, so if you would like to support us with a donation you can do so here.

**Donate**

### Contents [\[hide\]](#)

- 1 Pre-made images for the Raspberry Pi
  - 1.1 BerryBoot
- 2 Installing on top of an existing OS
  - 2.1 Raspbian on a Raspberry Pi
  - 2.2 Debian / Ubuntu on a PC
  - 2.3 Ubuntu on an ODroid-C1/C2
  - 2.4 Ubuntu on an ODroid-XU3/XU4
- 3 PetRockBlock Downloads

If you are installing RetroPie for the first time please follow the **OFFICIAL Installation Guide**

Click button to download

**Raspberry Pi 0/1**

md5: 98b4205ad0248d378c6776e20c54e487

**Raspberry Pi 2/3**

md5: 2e082ef5fc2d7cf7d910494cf0f7185b

**Raspberry Pi 4**

md5: 9154d998cba5219ddf23de46d8845f6c

**第 2 步:** 下载完成后，解压所下载的包。

**第 3 步:** 然后将 RetroPie 镜像安装到 micro-SD 卡。

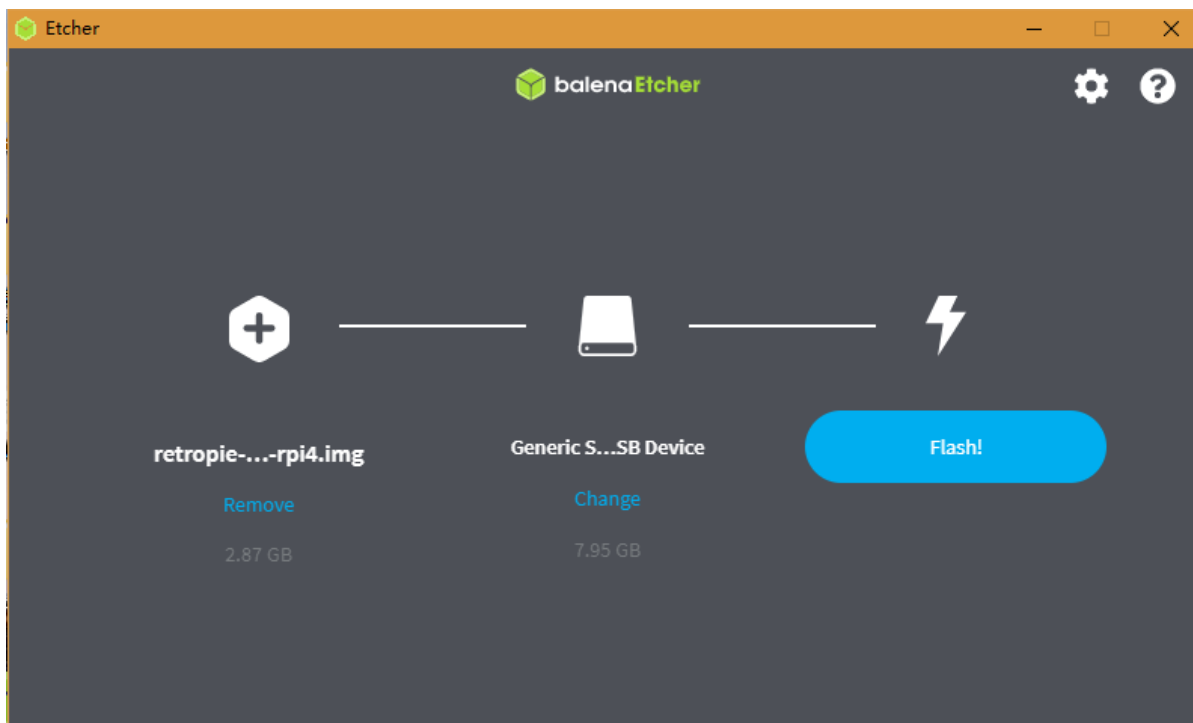
- 对于 Windows 用户，请使用: [Raspberry Pi Imager](#), [Etcher](#), 或者 [Win32DiskImager](#)。

**注解:** Win32DiskImager 需要从第 2 步中下载的.gz 映像中解压的

- 对于 macOS 用户，可用: [Raspberry Pi Imager](#), [Etcher](#), [Apple Pi Baker](#), 或者 `dd` 命令。
- 对于 Linux 用户，可用: [Raspberry Pi Imager](#), [Etcher](#), 或者 `dd` 命令。

**注解:** MacOS/Linux 用户可以用 `gunzip` 从下载的.gz 中提取





**第 4 步:** 将 micro-SD 卡插入 RasPad，按下电源键启动系统。

**警告：** SD 卡插槽采用卡扣式连接到后盖。在拆卸或更换后盖之前，请先取出 micro-SD 卡，以免损坏 micro-SD 卡和 RasPad 的内部按键板。

## 6.2.4 RetroPie 配置

RasPad 启动后，要配置控制器，设置 WiFi，以及传输游戏 ROM。执行这些步骤时需要键盘和游戏手柄。详细步骤如视频所示：

**注解：**

1. 你可以到 RetroPie 官网查看详细教程: [RetroPie Docs](#).
2. RetroPie 可让您将树莓派或 PC 变成复古游戏机。但是由于版权问题，RetroPie 不提供游戏 ROM。如果你想玩更多游戏，可以从论坛或谷歌上下载找到资源，然后将游戏 ROM 放在 RetroPie emulator 目录下。

下面是玩超级马里奥 3 的一个示例。



## 6.3 3D 打印机

使用 3D 打印机时，我们需要使用 OctoPrint。它是一个开源的 3D 打印机控制器应用程序，可为连接的打印机提供 Web 界面。它显示打印机的状态和关键参数，并允许用户安排打印和远程控制打印机。

OctoPrint 的详细安装教程请参考：<https://community.octoprint.org/t/setting-up-octoprint-on-a-raspberry-pi-running-raspbian-or-raspbian-32-bit/2337>。

本教程写了很详细的安装步骤，可能需要很长时间，需要比较耐心。

---

### 注解：

- 在开始本教程之前，需要在 micro-SD 卡上安装树莓派操作系统。请参考[安装树莓派镜像](#)。
- **网络摄像头选项：**需要在 RasPad 上安装摄像头才能使用网络摄像头。请参考教程[安装摄像头](#)。

- **Touch UI 选项:** 新的树莓派图像可能没有 Touch UI 的自动启动功能。请参考教程[Touch UI](#) 来配置 Touch UI 功能的设置。

### 6.3.1 Touch UI

如果 autostart 文件不在 `~/.config/lxsession/LXDE-pi` 路径中, 则需要手动添加该文件。

在 `~/config` 目录中创建 `lxsession` 文件夹和 `LXDE-pi` 文件夹。

```
mkdir ~/.config/lxsession
mkdir ~/.config/lxsession/LXDE-pi
```

将 autostart 文件从路径 `/etc/xdg/lxsession/LXDE-pi` 复制到 `~/.config/lxsession/LXDE-pi` 文件夹。

```
cp /etc/xdg/lxsession/LXDE-pi/autostart ~/.config/lxsession/LXDE-pi/autostart
```

将 autostart 文件的权限设置为可读可写。

```
chmod 644 ~/.config/lxsession/LXDE-pi/autostart
nano .config/lxsession/LXDE-pi/autostart
```

使用 **Nano** 等文本编辑器打开自动启动文件, 并将以下行添加到文件末尾, 以使 RasPad 在启动时执行 **startTouchUI.sh** 脚本文件。

```
@/home/pi/startTouchUI.sh
```

重启 RasPad 后, OctoPrint 的 Touch UI 将以全屏模式打开。按 **F11** 退出全屏模式, 进入桌面。

### 6.3.2 制作 3D 模型

点击这个链接: <https://projects.raspberrypi.org/en/projects?hardware%5B%5D=3d-printer>, 参考树莓派官方教程, 可以得到 .stl 格式的 3D 模型文件。

通常, 3D 打印机无法直接处理 .stl 文件。需要使用 **Ultimaker Cura** 软件进行切片, 然后通过 **OctoPrint** 上传到 3D 打印机再进行打印。

下载 **Ultimaker Cura**。由于 Ultimaker Cura 在树莓派系统上不可用, 您需要在您的计算机上执行切片操作。

## Ultimaker Cura

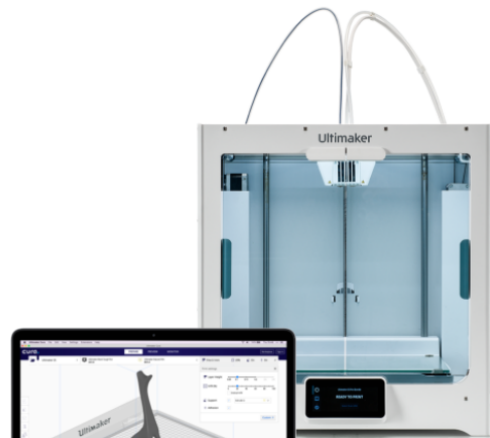
Trusted by millions of users, Ultimaker Cura is the world's most popular 3D printing software. Prepare prints with a few clicks, integrate with CAD software for an easier workflow, or dive into custom settings for in-depth control.



Ultimaker Cura 4.9.1

[Download for free](#)

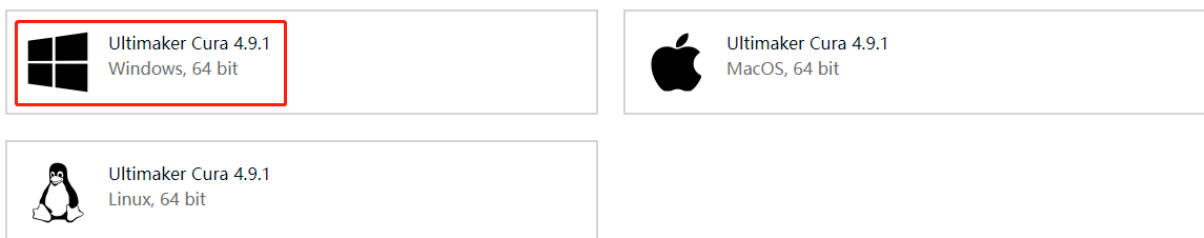
[Get Enterprise edition](#)



选择您需要的版本。

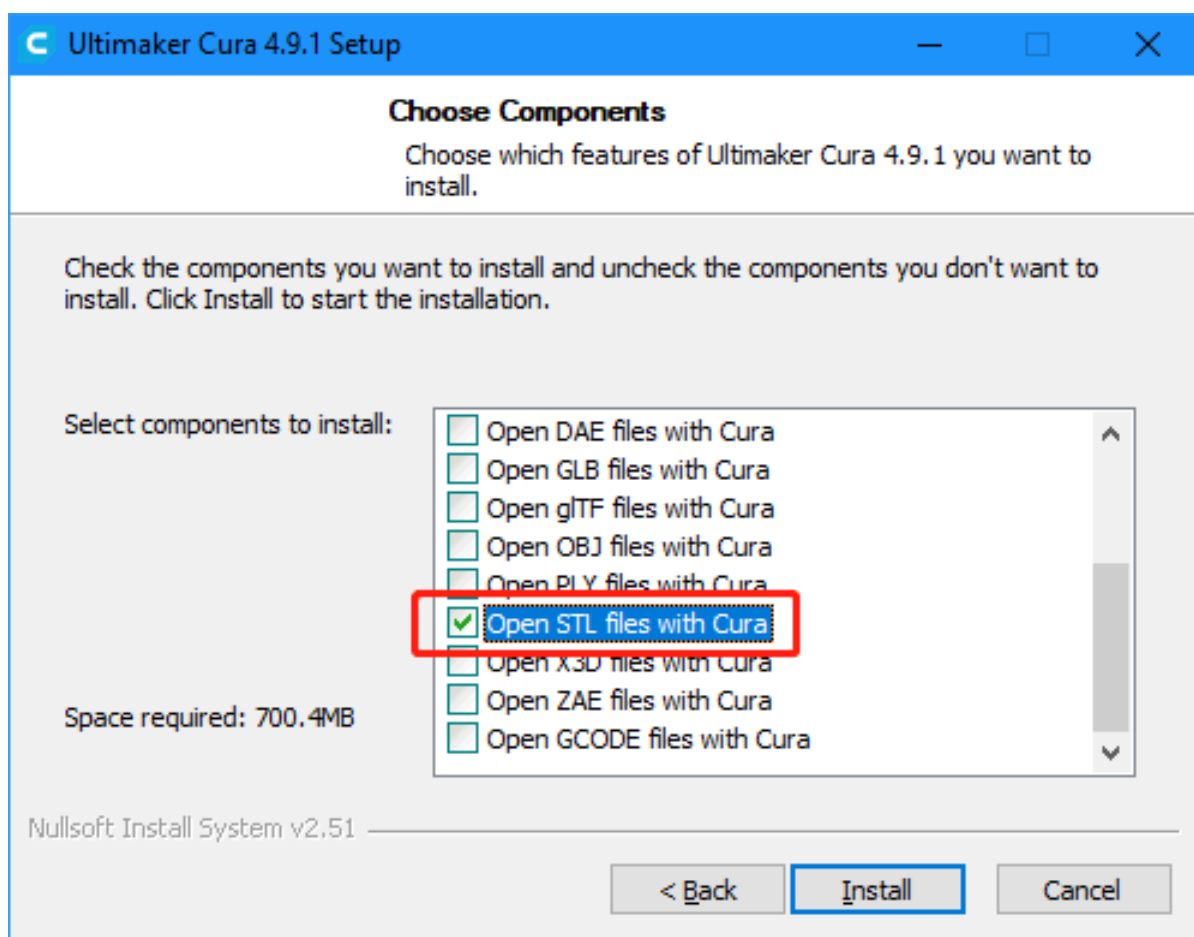
## Choose your operating system

You're almost ready to start 3D printing with Ultimaker Cura. Just let us know which operating system you are using.

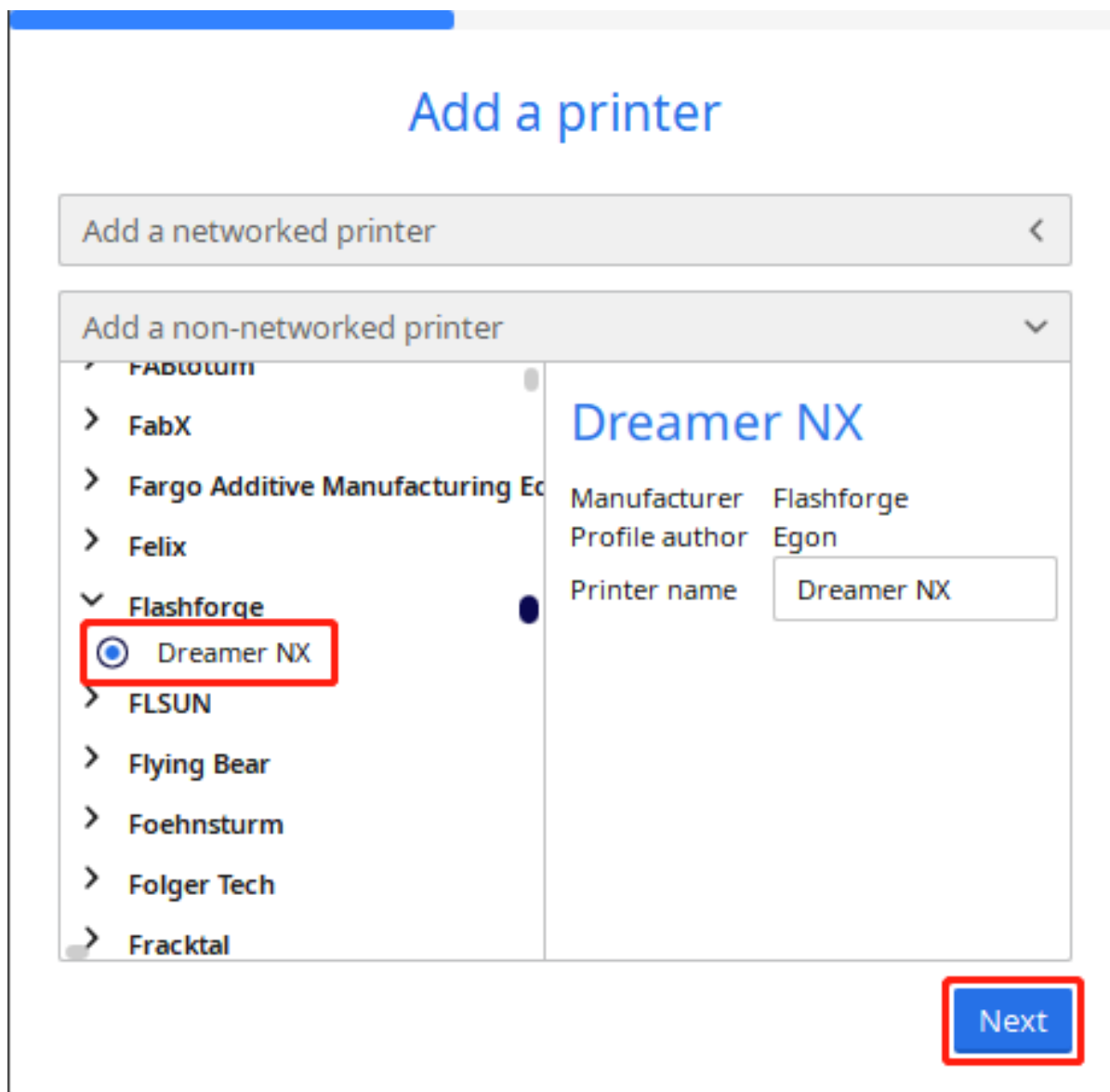


安装 Ultimaker Cura 时，请注意在 **Select components** 步骤中，**Open the STL file with Cura** 已经默认勾选，这样.stl 文件就可以切片了。

如果你想切换其他类型的模型文件，勾选对应选项，否则直接安装即可。



首次打开 Ultimaker Cura 时，会出现一系列配置提示。在 **Add a printer** 步骤中，选择使用的打印机型号，然后单击 **Next step**。



选择正确的打印机型号后，验证 Ultimaker Cura 在机器设置页面中提供的参数是否正确，或者直接更改参数。按照屏幕提示完成 Ultimaker Cura 的配置。

## Machine Settings

### Dreamer NX

Printer

Extruder 1

#### Printer Settings

X (Width)	<input type="text" value="230.0"/> mm
Y (Depth)	<input type="text" value="150.0"/> mm
Z (Height)	<input type="text" value="140.0"/> mm
Build plate shape	<input type="text" value="Recta..."/> ▾
Origin at center	<input checked="" type="checkbox"/>
Heated bed	<input checked="" type="checkbox"/>
Heated build volume	<input type="checkbox"/>
G-code flavor	<input type="text" value="Marlin"/> ▾

#### Start G-code

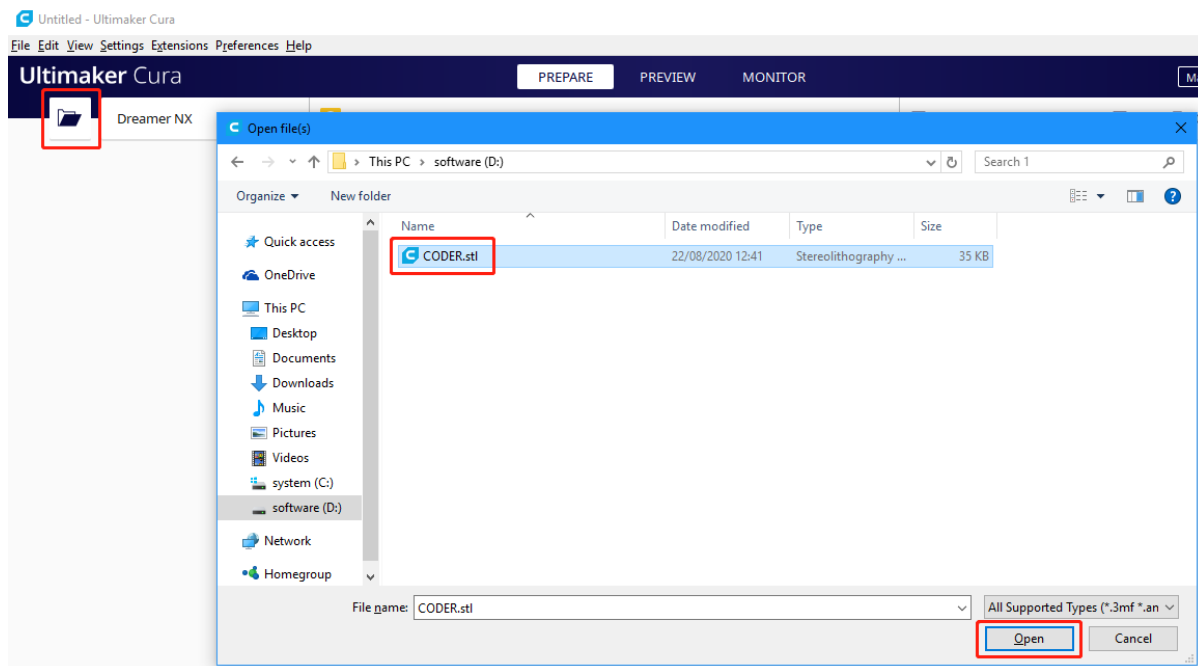
#### Printhead Settings

X min	<input type="text" value="-15"/> mm
Y min	<input type="text" value="-25"/> mm
X max	<input type="text" value="40"/> mm
Y max	<input type="text" value="35"/> mm
Gantry Height	<input type="text" value="30.0"/> mm
Number of Extruders	<input type="text" value="1"/> ▾
Apply Extruder offsets to	<input checked="" type="checkbox"/> Code

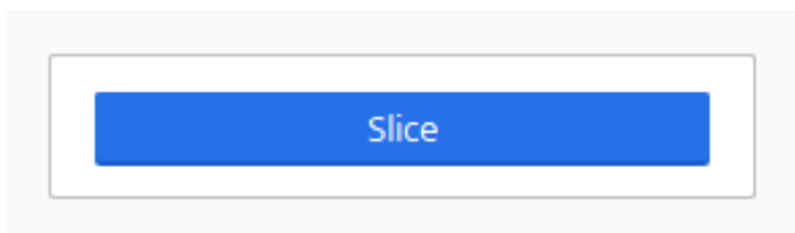
#### End G-code

Next

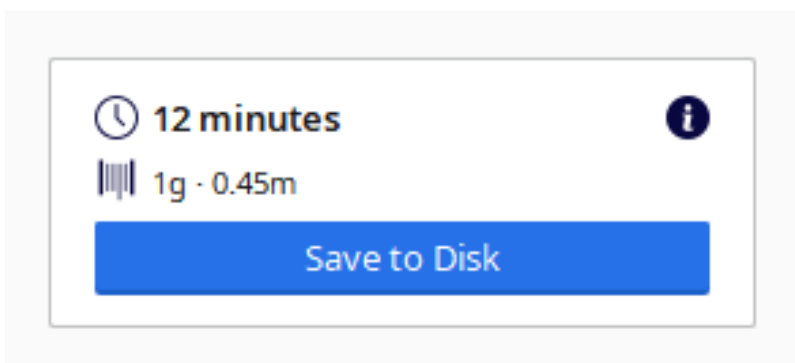
Ultimaker Cura 成功启动后，单击左上角的 **Folder** 图标并浏览到包含需要切片的.stl 3D 模型文件的文件夹，然后单击 **Open** 添加.stl 文件到 Ultimaker Cura 的库。



添加文件后，单击右下角的 **Slice** 选项，Ultimaker Cura 将自动执行切片操作。

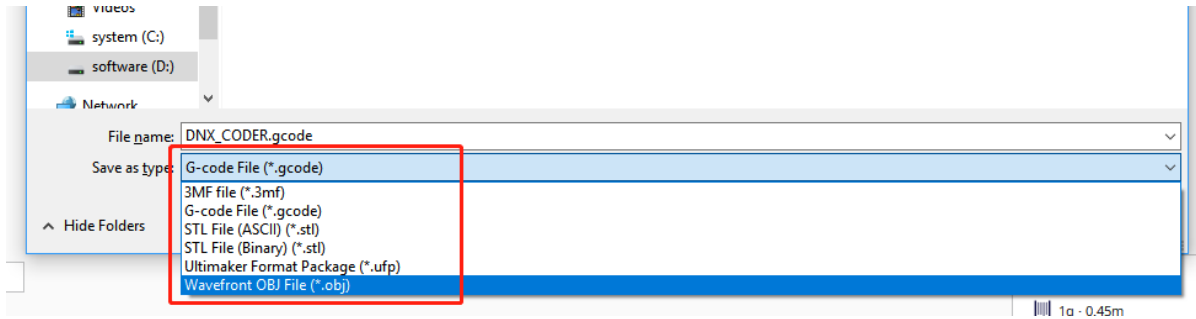


切片完成后，点击右下角的 **Save to Disk** 选项，将切片后的文件保存到本地。



选择 3D 打印机识别的文件扩展名类型，然后单击 **Save**。





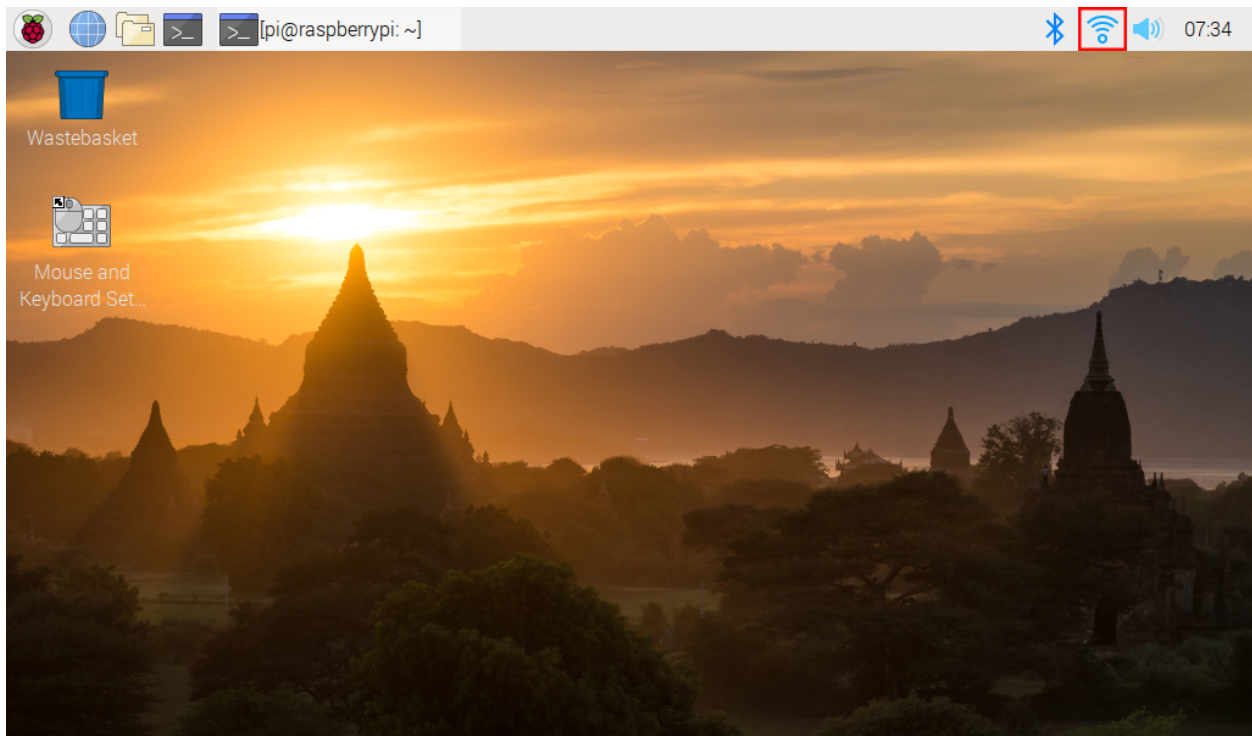
### 6.3.3 打印 3D 模型

.stl 文件切片后，3D 模型文件可以通过 OctoPrint 发送到 3D 打印机进行打印。

打开 RasPad 浏览器，输入 <http://192.168.18.179/?#temp> 登录 OctoPrint。

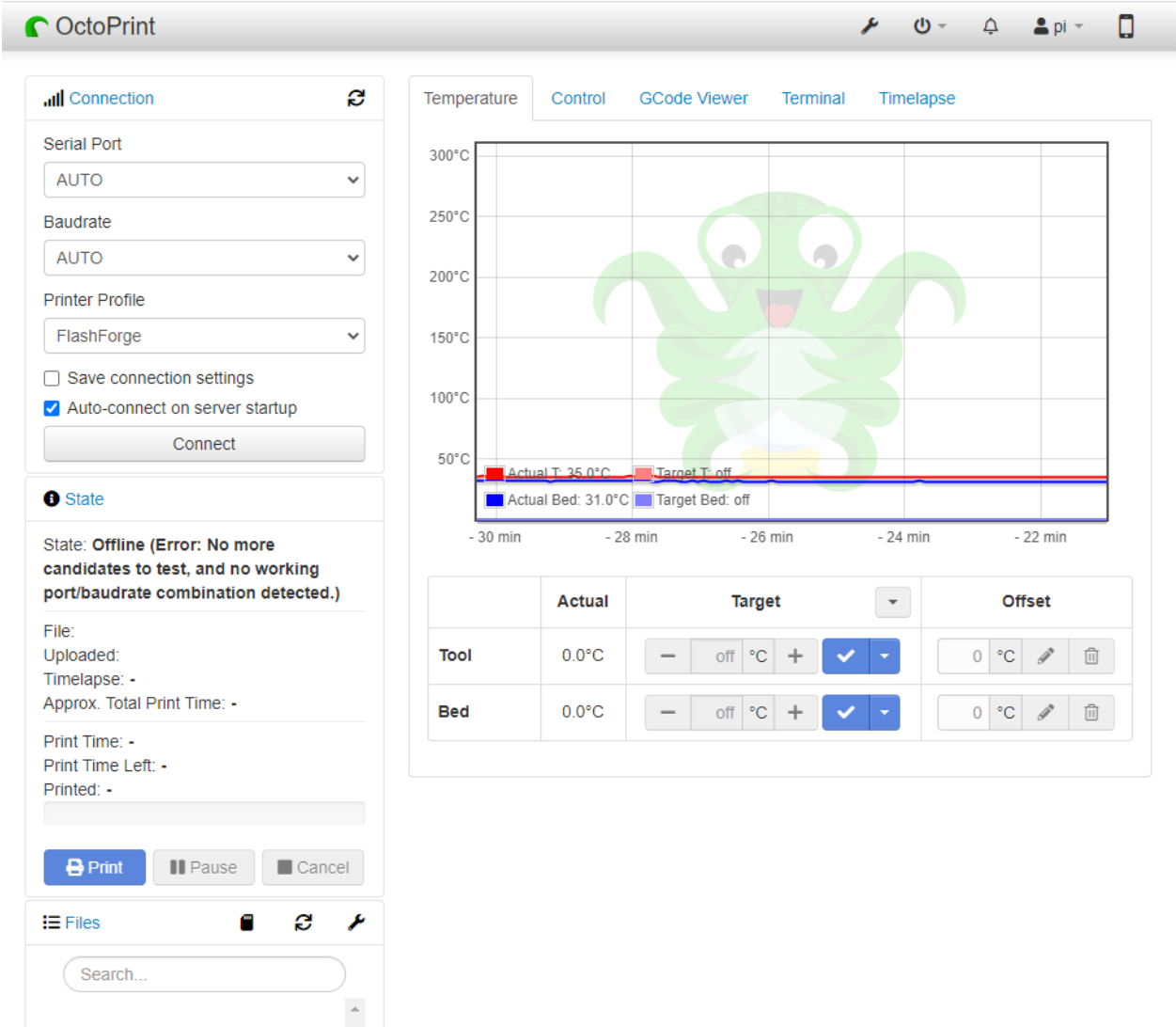
**注解：**在登录 OctoPrint 的 Web UI 之前，OctoPrint 首先需要已成功安装在 RasPad 上。

IP 地址 192.168.18.179 将需要替换为 RasPad 的本地 IP 地址。将光标悬停在 RasPad 桌面上的 WiFi 图标上，将显示本地 IP 地址。

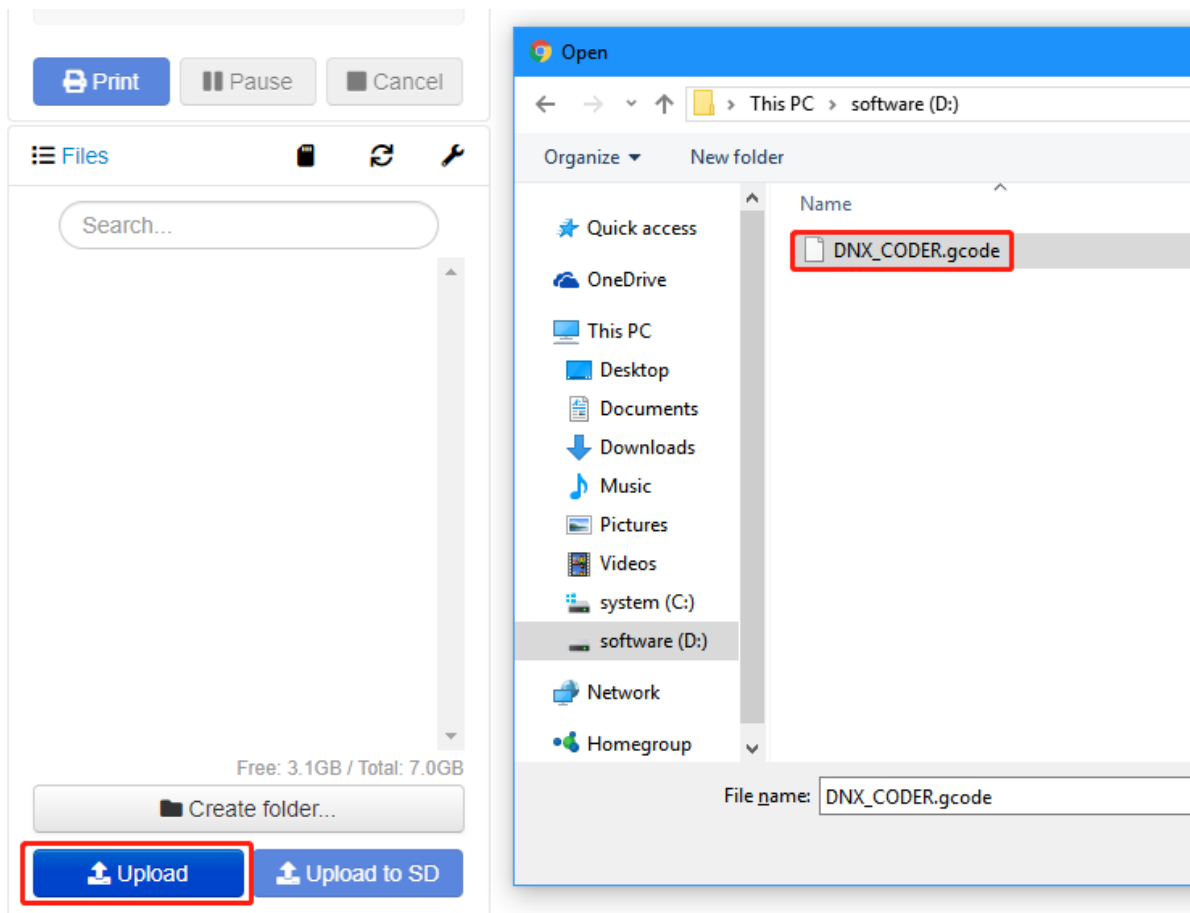


现在您已进入 OctoPrint。

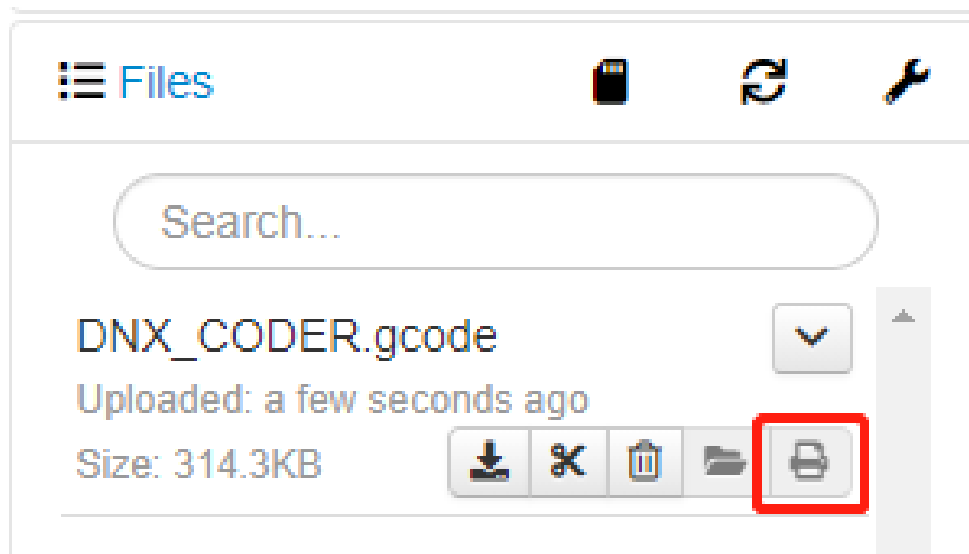




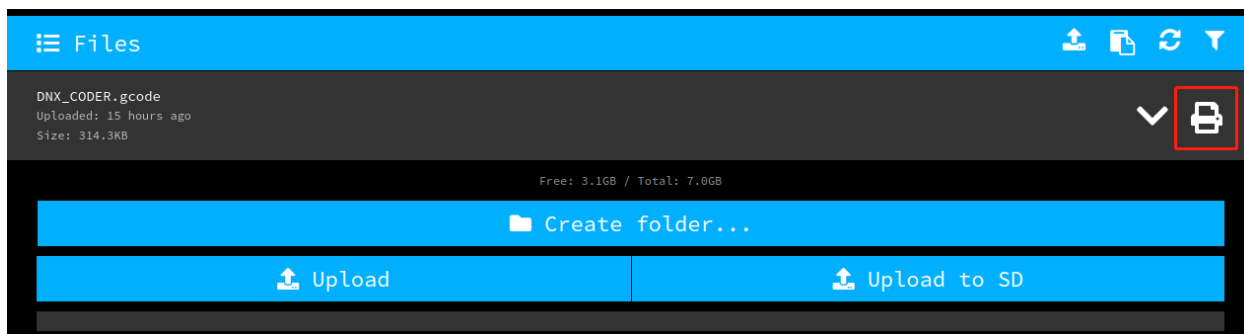
点击 **Upload** 选项，选择切片后的 3D 模型文件。



单击 **print** 图标。切片过程完成后，3D 打印机将开始打印 3D 模型文件。



如果你已经将切片文件传输到树莓派上，你也可以在 RasPad 中打开 OctoPrint UI 进行打印。

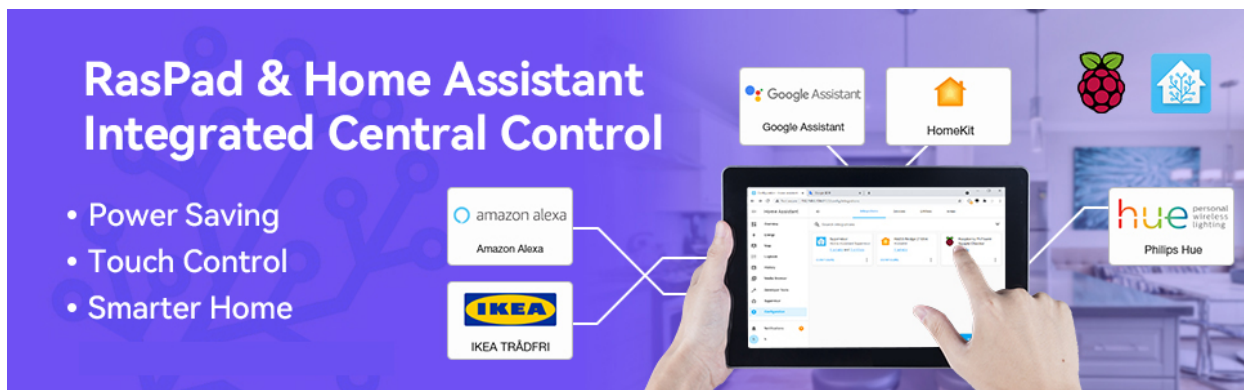


### 6.3.4 视频

以下视频显示安装 OctoPrint 后，通过 USB 线连接 3D 打印机和 RasPad 3，上传设计好的 3D 文件，然后使用摄像头监控打印过程。

还可以监控温度，以防止 3D 打印机过热或过冷，从而影响打印的 3D 模型。

## 6.4 Home Assistant



Home Assistant 是一个基于 Python 的开源智能家居系统，支持多个品牌的智能家居设备，可以轻松实现设备的语音控制和自动化。您可以在树莓派上运行 Home Assistant，而无需将数据发送到互联网上。这意味着您可以完全控制自己的数据，限制来自智能设备的互联网流量，并加强安全性。

RasPad 3 由树莓派 4 供电，配备 10.1 英寸高清触摸屏和 5 小时电池续航时间。我们还为其编写了自启动和全屏启动脚本，因此您可以更轻松、更快速地在 RasPad 上使用 Home Assistant。

### 更多

以下是其他人结合树莓派和 Home Assistant 制作的一些项目。

- 我的廉价智能家居系统
- 开启 Home Assistant - JuanMTech
- DIY 家庭自动化入侵报警系统

### 快速指南

## 6.4.1 安装 Home Assistant

### 安装 Docker

这里将介绍如何使用容器环境来安装和运行 Home Assistant。我们推荐使用 Docker 作为容器环境。安装 Docker 的方法很简单，运行如下命令即可。

```
curl -fsSL https://get.docker.com -o get-docker.sh
sudo sh get-docker.sh
```

**注解：** 你可以在这里查看更多信息 [如何安装 Docker](#)。

### 安装 Home Assistant

安装容器环境后，如果你使用的是 **树莓派 3**，运行以下命令安装 Home Assistant。

```
sudo docker run -d \
--name homeassistant \
--privileged \
--restart=unless-stopped \
-e TZ=MY_TIME_ZONE \
-v /home/pi/homeassistant:/config \
--network=host \
ghcr.io/home-assistant/raspberrypi3-homeassistant:stable
```

如果您使用的是 **树莓派 4**，则请运行以下命令来安装 Home Assistant。

```
sudo docker run -d \
--name homeassistant \
--privileged \
--restart=unless-stopped \
-e TZ=MY_TIME_ZONE \
-v /home/pi/homeassistant:/config \
--network=host \
ghcr.io/home-assistant/raspberrypi4-homeassistant:stable
```

**注解：** 更多详细信息，请查阅 [安装 Home Assistant 容器](#)。

### 启动 Home Assistant 服务

安装完成后，您可以运行以下命令启动 Home Assistant。

```
sudo docker start homeassistant
```

使用以下命令关闭 Home Assistant。

```
sudo docker stop homeassistant
```

使用以下命令重新启动 Home Assistant。

```
sudo docker restart homeassistant
```

### 进入 Home Assistant 页面

现在你可以在浏览器中输入 `http://<localhost>:8123` 来进入 Home Assistant 操作页面。例如我的树莓派 IP 是 192.168.6.136，然后便可以访问 `http://192.168.6.136:8123`。

注解:

1. 如果需要配置开机启动, 请参考: [在开机时启动 Home Assistant 服务](#)
  2. 如果需要配置开机进入 Home Assistant 页面, 请参考: [启动时进入 Home Assistant 页面](#)
- 

## 6.4.2 安装和配置 ESPHome

---

**注解:** 如果你想在 Home Assistant 中添加 ESPHome, 那么你可以通过这个章节来安装和配置它。  
如果你不想添加它, 可以选择跳过这一节, 直接进入下一节[登录](#).

---

ESPHome 是为您的 ESP8266/ESP32 开发板创建自定义固件的完美解决方案。



以下是对 ESPHome 工作原理的快速介绍: ESPHome 是一种工具, 旨在尽可能简化 ESP 板的管理。它读取一个 YAML 配置文件 (就像 Home Assistant) 然后在你的 ESP 设备上安装一个自定义固件。在 ESPHome 的配置中添加的设备或传感器将自动显示在 Home Assistant 的 UI 中。

### 安装

1. 安装依赖项。

```
sudo pip3 install cryptography==2.8
```

2. 安装 ESPHome 前, 需要安装 Python 并通过 pip3 安装控制台脚本。

**注解:** 安装 ESPHome 1.18.0 或更高版本需要 Python 3.7 或更高版本。

---

```
sudo pip3 install esphome
```

或者通过 Docker 镜像安装 (此镜像现在可用于 AND64、ARM 和 ARM64 (AARCH64) 架构); 如果您有其他架构, 请通过 pip 安装 ESPHome 或使用 Home Assistant 插件:

```
docker pull esphome/esphome
```

3. 打开 ESPHome 仪表板。

**注解:** 在 ESPHome 仪表板中操作时, 需要保证此命令正常运行。

---

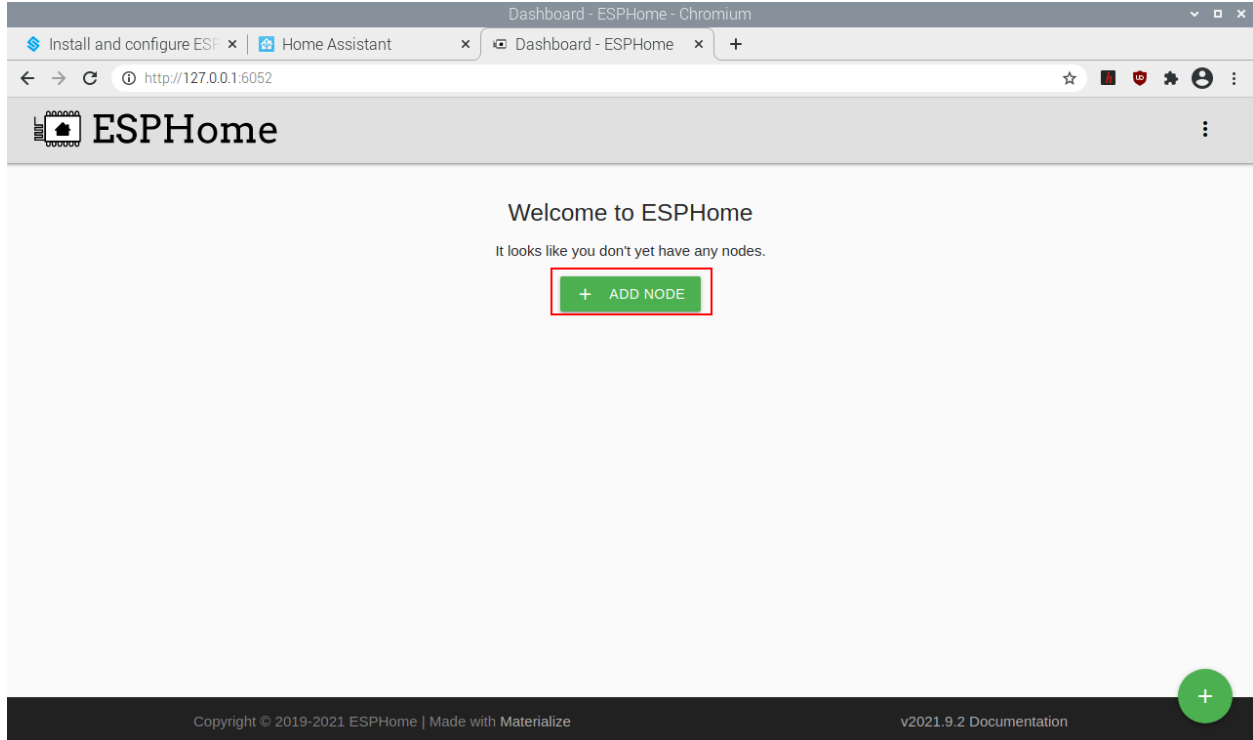
```
esphome dashboard config/
```

### 配置 ESPHome

1. 打开浏览器并通过 <http://localhost:6052> 或 <http://X.X.X.X:6052> (这里的 XXXX 需要替换为您的树莓派 IP 地址) 进入 ESPHome Dashboard。

2. 创建配置

单击“+”开始添加配置。



输入设备连接所需的名称以及 WiFi 和密码，然后单击 **NEXT**。

Dashboard - ESPHome - Chromium

Install and configure ESP x Dashboard - ESPHome x +

← → ↻ ⓘ http://127.0.0.1:6052

# ESPHome

### Create configuration

Limited functionality because you're not browsing the dashboard over a secure connection (HTTPS).

Name\*  
esp-light

Enter your Wi-Fi information so your device can connect to your wireless network.

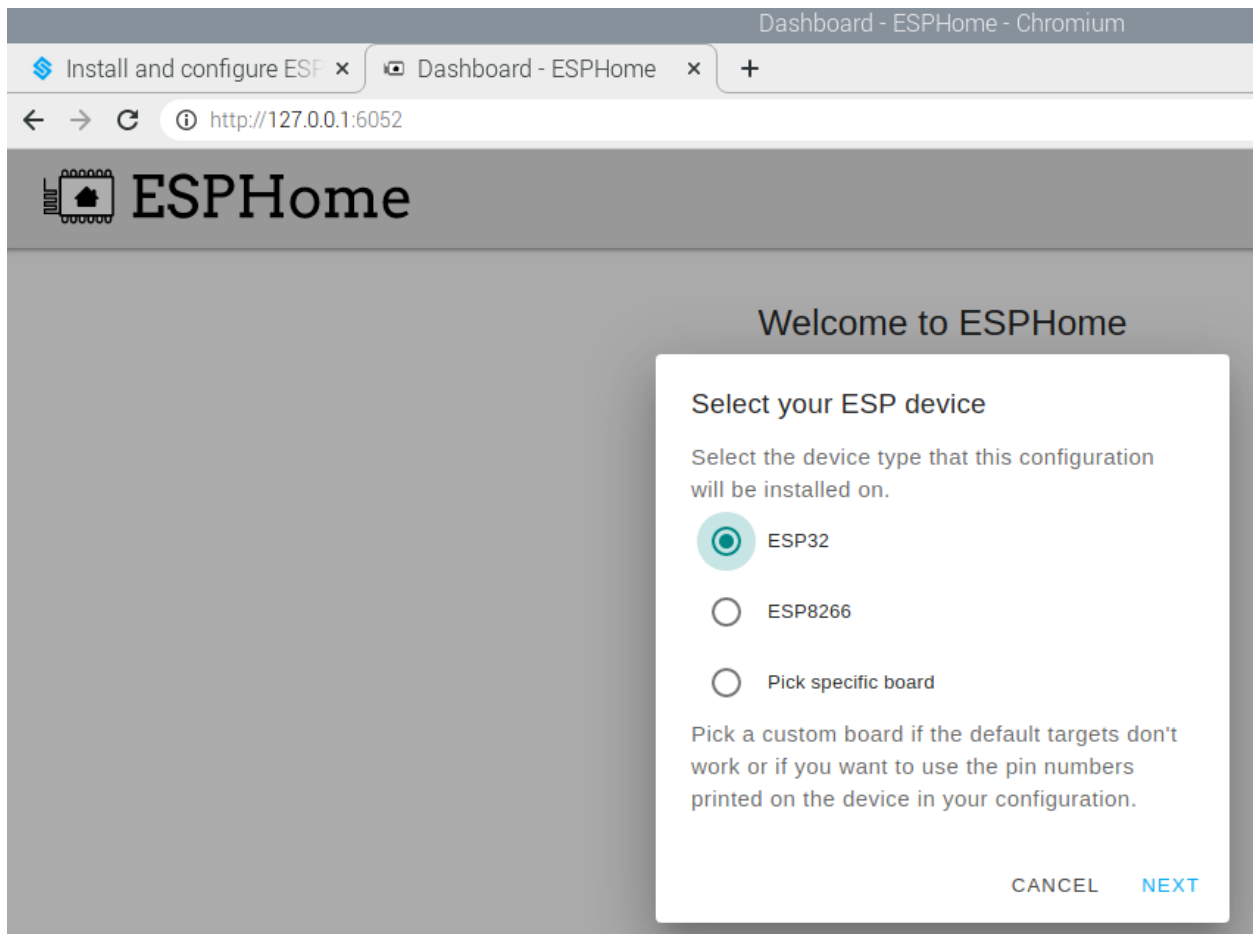
Wi-Fi SSID\*

Wi-Fi password

CANCEL NEXT

Copyright © 2019-2021 ESPHome | Made with Materialize

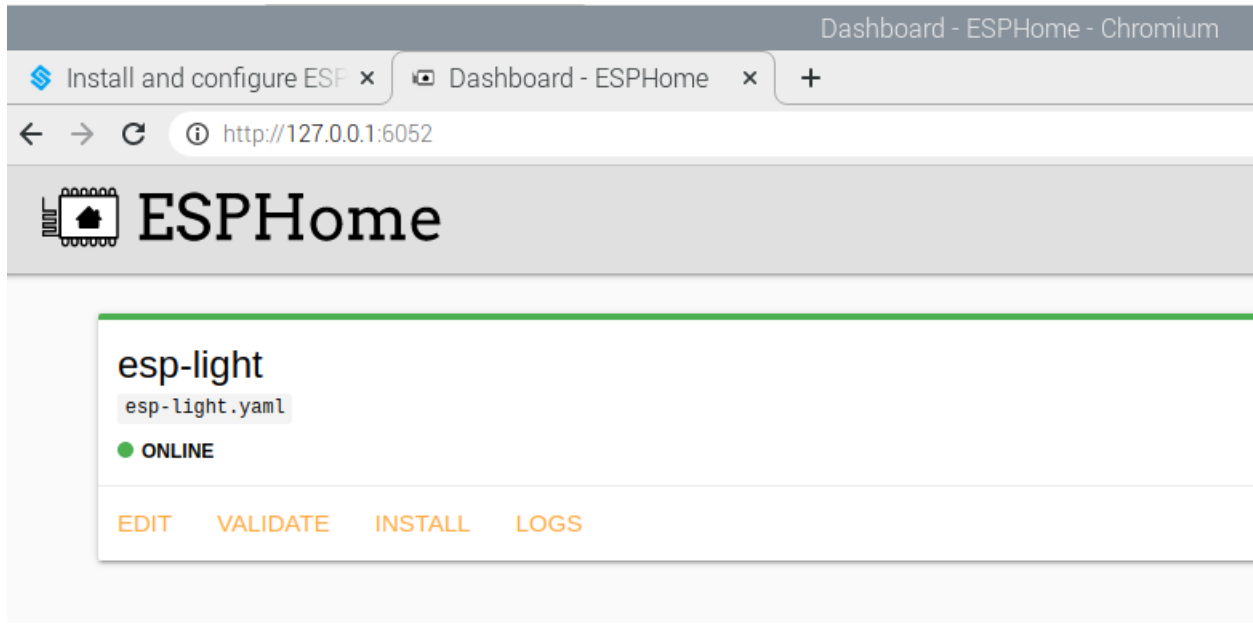
选择设备类型（例如 ESP32）。



点击 NEXT 添加完之后，您将在 ESPHome 的仪表板中看到它们。

**注解：** 当不是第一次添加设备时，在主页面点击“+”，按照提示输入设备名称、WIFI 和密码，选择设备类型。





3. 编辑 .yaml 文件.

单击右下角的 **EDIT** 进入 .yaml 文件内部，您将看到以下默认组件。

# Editing: `esp-light.yaml`

```
1  ▾ esphome:
2      name: esp-light
3      platform: ESP32
4      board: esp32dev
5
6      # Enable logging
7      logger:
8
9      # Enable Home Assistant API
10     api:
11
12  ▾ ota:
13      password: "b741be5b3c53a45b7bca254c5225e7ee"
14
15  ▾ wifi:
16      ssid: "MakerStarsHall"
17      password: "sunfounder"
18
19      # Enable fallback hotspot (captive portal) in
20  ▾ ap:
21      ssid: "Esp-Light Fallback Hotspot"
22      password: "YreDouoMcDRl"
23
24     captive_portal:
```

- `esphome`: 包含您设置的名称、平台和板类型。
- `logger`: `logger` 组件通过串口和 MQTT 主题自动记录所有日志消息。
- `api`: ESPHome 原生 API 用于通过高度优化的网络协议直接与客户端通信。目前，只有 ESPHome 工具和 Home Assistant 使用这个原生 API。
- `ota`: 使用 OTA (Over The Air) 更新组件，您可以将固件二进制文件上传到您的节点，而无需使用 USB 电缆进行上传。
- `wifi`: 这个 ESPHome 组件可以帮您设置 WiFi 连接。必须要配置好，否则 ESPHome 将在配置验证阶段遇到错误。
- `captive_portal`: 这个组件是连接到 WiFi 失败时的回退机制。当尝试连接 WiFi 失败 1 分钟之后，ESP 将启动 WiFi 热点（根据配置启动）。

现在开始添加其他组件。在 ESPHome 官方网站 中还有很多其它的组件, 包括 传感器组件, 输出组件, 光组件等, 下面我们以 光组件 为例开始添加。



# ESPHome

Quick search

Table of Contents

[Binary Light](#)

- [Configuration variables:](#)
- [See Also](#)

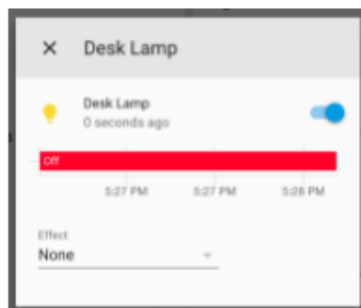
[Join the community](#)

[Follow us on Twitter](#)

[Source Code](#)

## Binary Light

The `binary` light platform creates a simple ON/OFF-only light from a `binary output component`.



```
# Example configuration entry
light:
  - platform: binary
    name: "Desk Lamp"
    output: light_output

output:
  - id: light_output
    platform: gpio
    pin: GPIO16
```

将以下代码添加到 .yaml 文件末尾。

```
# Example configuration entry
light:
  - platform: binary
    name: "Desk Lamp"
    output: light_output

output:
  - id: light_output
    platform: gpio
    pin: GPIO16
```

```

22 | password: YREDOUOMCDRI
23
24 captive_portal:
25
26 # Example configuration entry
27 light:
28   - platform: binary
29     name: "Desk Lamp"
30     output: light_output
31
32 output:
33   - id: light_output
34     platform: gpio
35     pin: GPIO16

```

4. 将 .yaml 文件安装到 ESP32 板上。

编辑完成后点击右下角的 **INSTALL** 进行编译安装。有两种方法供您选择：无线和 USB 端口。但是对于第一次安装，您需要使用 USB 将 ESP32 板连接到树莓派，然后选择第二种安装方法。之后的每一次安装便可以直接通过无线方式安装和编译了。

第一次编译下载依赖大约需要 10 分钟，请耐心等待。

## How do you want to install this on your ESP device?

Wirelessly

Requires the device to be online



Plug into this computer

Dashboard needs to be opened via HTTPS



Plug into the computer running ESPHome Dashboard

For devices connected via USB to the server



**点击这里，如果你的ESP32板子已经插入到树莓派的USB端口。**

Manual download

Install it yourself using ESPHome Flasher or other tools



安装成功后，如果在 ESP32 板的 GPIO16 上连接一个 LED，您将看到 LED 亮起。记好此 IP，将 ESPHome 添加到 Home Assistant 时需要填写此 IP。

```
[18:03:51][C][wifi:359]: SSID: 'MakerStarSHA11'
[18:03:51][C][wifi:360]: IP Address: 192.168.18.254 Record the IP
[18:03:51][C][wifi:362]: BSSID: D0:C7:C0:AC:C5:7F
[18:03:51][C][wifi:363]: Hostname: 'test'
[18:03:51][C][wifi:367]: Signal strength: -53 dB
[18:03:51][C][wifi:371]: Channel: 6
[18:03:51][C][wifi:372]: Subnet: 255.255.255.0
[18:03:51][C][wifi:373]: Gateway: 192.168.18.1
[18:03:51][C][wifi:374]: DNS1: 192.168.18.1
[18:03:51][C][wifi:375]: DNS2: 0.0.0.0
[18:03:51][C][gpio.output:010]: GPIO Binary Output:
[18:03:51][C][gpio.output:011]: Pin: GPIO16 (Mode: OUTPUT)
[18:03:51][C][logger:189]: Logger:
[18:03:51][C][logger:190]: Level: DEBUG
[18:03:51][C][logger:191]: Log Baud Rate: 115200
[18:03:51][C][logger:192]: Hardware UART: UART0
[18:03:51][C][light:097]: Light 'Switch'
[18:03:51][C][captive_portal:148]: Captive Portal:
[18:03:51][C][ota:029]: Over-The-Air Updates:
[18:03:51][C][ota:030]: Address: test.local:3232
[18:03:51][C][ota:032]: Using Password.
[18:03:51][C][api:095]: API Server:
[18:03:51][C][api:096]: Address: test.local:6053
[18:03:54][D][api.connection:730]: Client 'Home Assistant 2021.8.8 (192.168.18.112)' connected successfully!
```


**注解：** 如果点击 **INSTALL** 没有反应，请清除浏览器缓存，重新安装。

```
sudo rm -rf ~/.cache/chromium
```

ESPHome 配置完成后，便可以将其添加到 Home Assistant 了。

### 6.4.3 登录

1. 现在，访问 <http://ip:8123> 以进入您的 Home Assistant，然后为您的 Home Assistant 设置用户名和密码。



# Home Assistant

Are you ready to awaken your home, reclaim your privacy and join a worldwide community of tinkerers?

Let's get started by creating a user account.

Name


Username

Password

Confirm Password

**CREATE ACCOUNT**

2. 为您的家庭创建一个名称并选择所在的地区。



# Home Assistant


Hello Sunfounder, welcome to Home Assistant. How would you like to name your home?

[Name of your Home Assistant installation](#)

Home

We would like to know where you live. This information will help with displaying information and setting up sun-based automations. This data is never shared outside of your network.

We can help you fill in this information by making a one-time request to an external service. [DETECT](#)



Leaflet | © OpenStreetMap, © CARTO

Time Zone

Elevation

0 meters

Unit System

☒ Metric  
Celsius, kilograms

☐ Imperial  
Fahrenheit, pounds


Currency

[Find your value](#)

Currency

[NEXT](#)

3. 选择您想要分享给 Home Assistant 的内容。

 Home Assistant

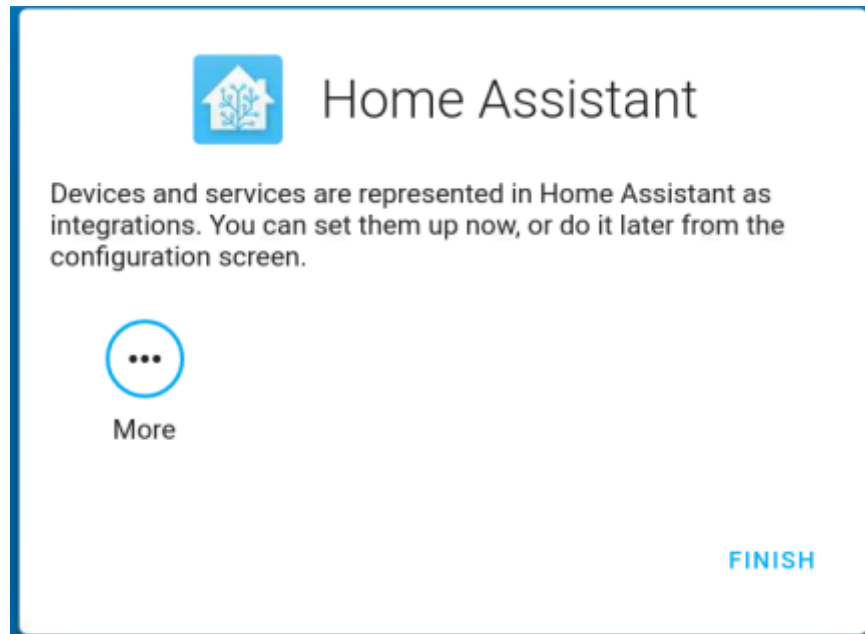
Share anonymized information from your installation to help make Home Assistant better and help us convince manufacturers to add local control and privacy-focused features.

- ☐ Basic analytics  
This includes information about your system.
- ☐ Usage  
Details of what you use with Home Assistant
- ☐ Statistical data  
Counts containing total number of datapoints
- ☐ Diagnostics  
Share crash reports when unexpected errors occur.

[How we process your data](#) **NEXT**

4. 您可以选择现在或稍后配置设备。建议初学者稍后配置设备，配置将在后面的教程中详细介绍。





5. 设置完成后，在登录屏幕中输入您的登录用户名和密码。

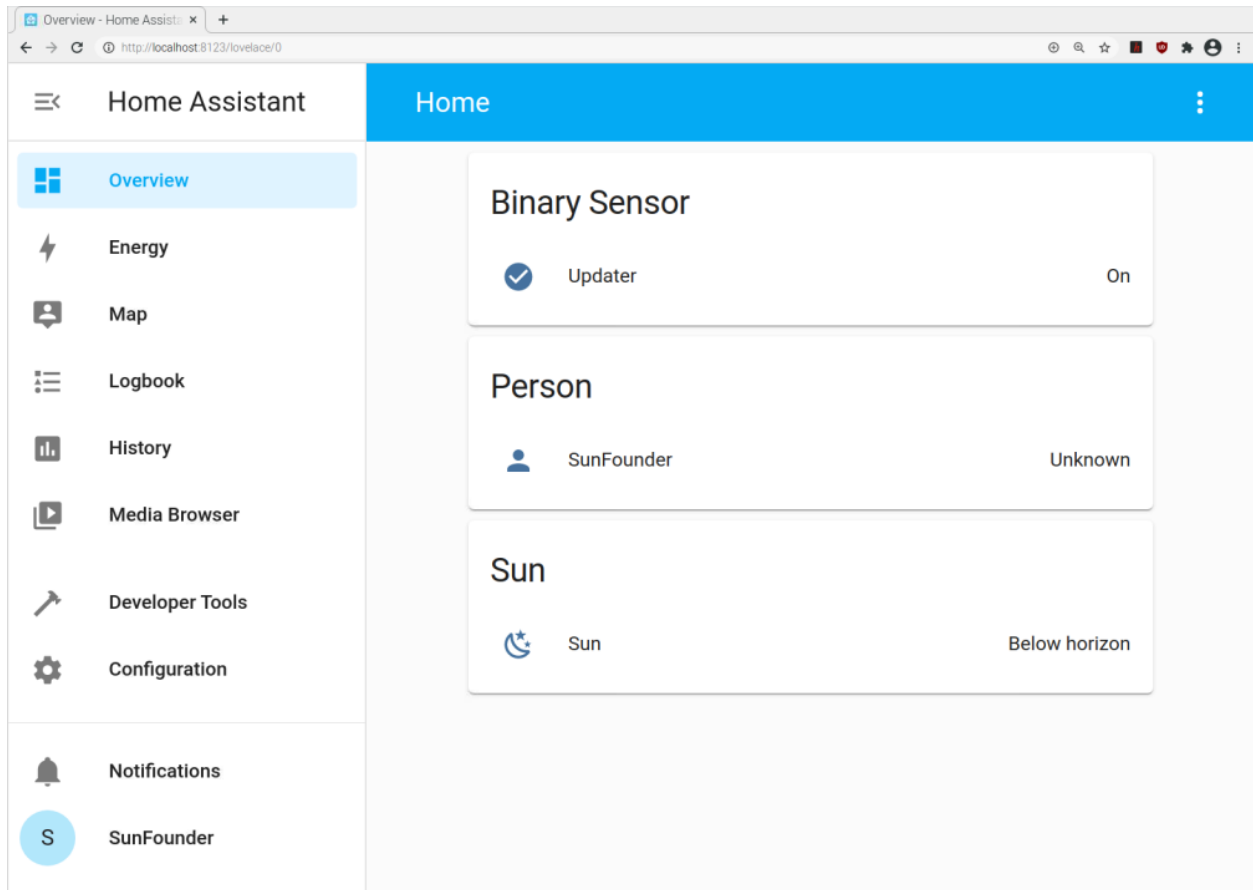
The image shows a Home Assistant login screen. At the top left is the Home Assistant logo (a blue house with a white tree inside). To its right is the text 'Home Assistant'. Below the logo and title, there is a paragraph: 'You're about to give <http://localhost:8123/> access to your Home Assistant instance.' Below this paragraph, there is another paragraph: 'Logging in to **Home** with **Home Assistant Local**.' Below this, there are two input fields. The first is labeled 'Username' and has a blue underline. The second is labeled 'Password' and has a blue underline and a blue eye icon to its right. Below the input fields, there is a blue button labeled 'NEXT'.

---

**注解：** 如果登录界面没有自动跳转，请通过 <http://localhost:8123> 再次打开 Home Assistant。

---

登录成功之后，将会自动进入您的主页。



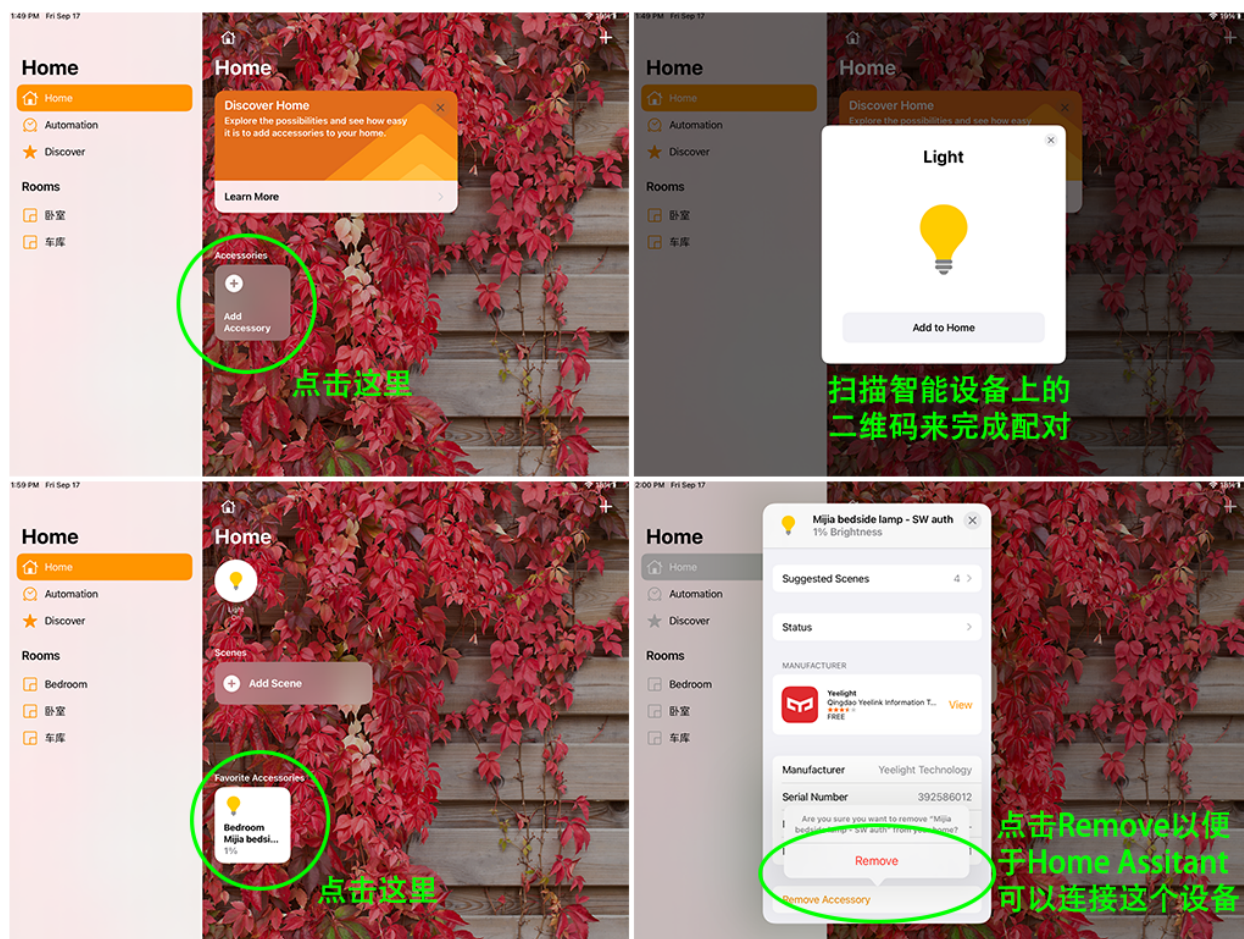
#### 6.4.4 添加智能设备

您可以将您购买的智能设备添加到 Home Assistant 中，例如智能音响、智能台灯等。您也可以使用 ESP32 制作并添加自己的 DIY 设备。

在本节中，您将学习如何添加您的智能设备。

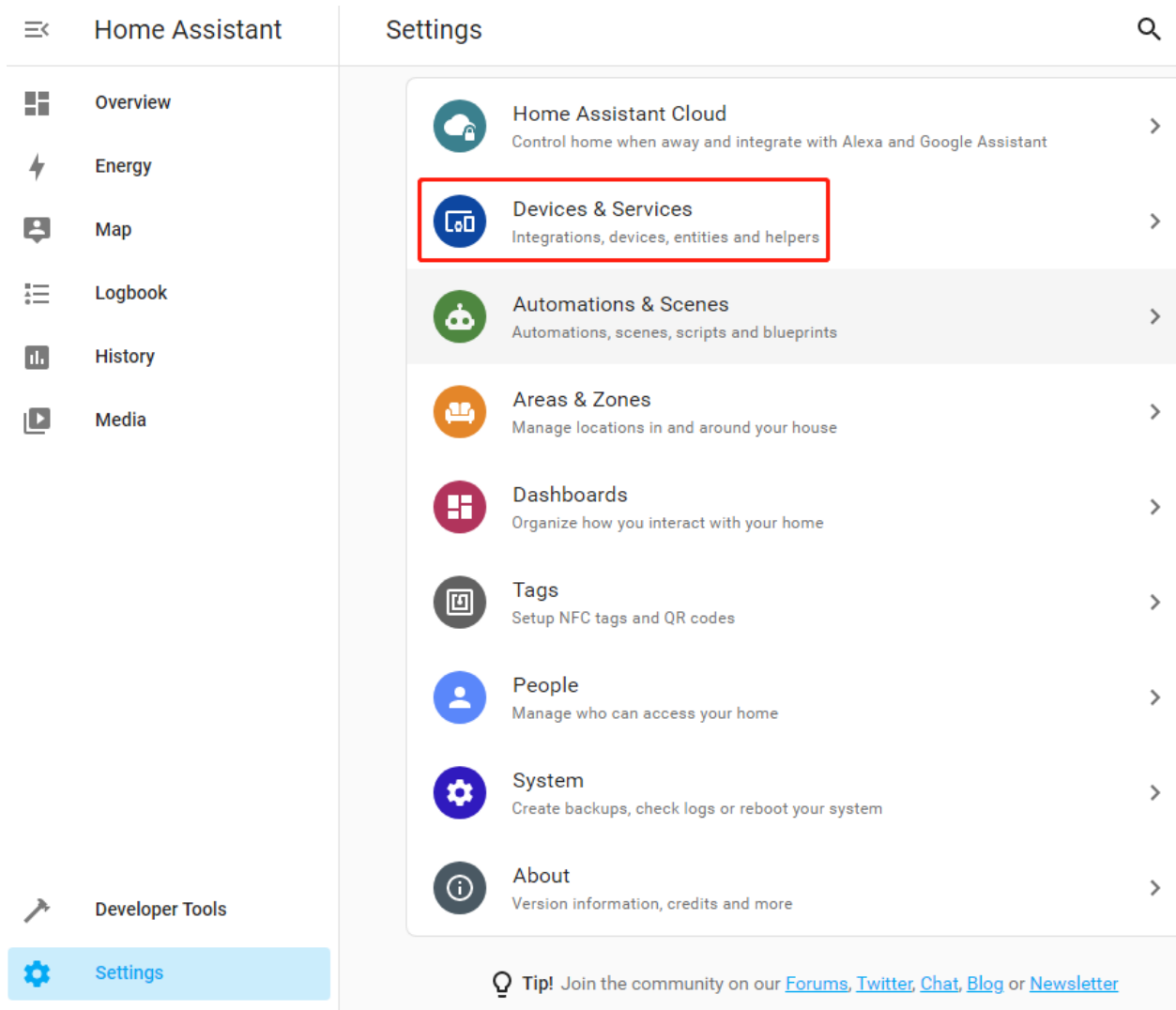
##### 配置智能设备

您需要确保您的智能设备已分配 IP。您可以通过智能设备对应的 APP 完成此步骤。例如，下图显示了一个配置了 **HomeKit** 的床头灯。

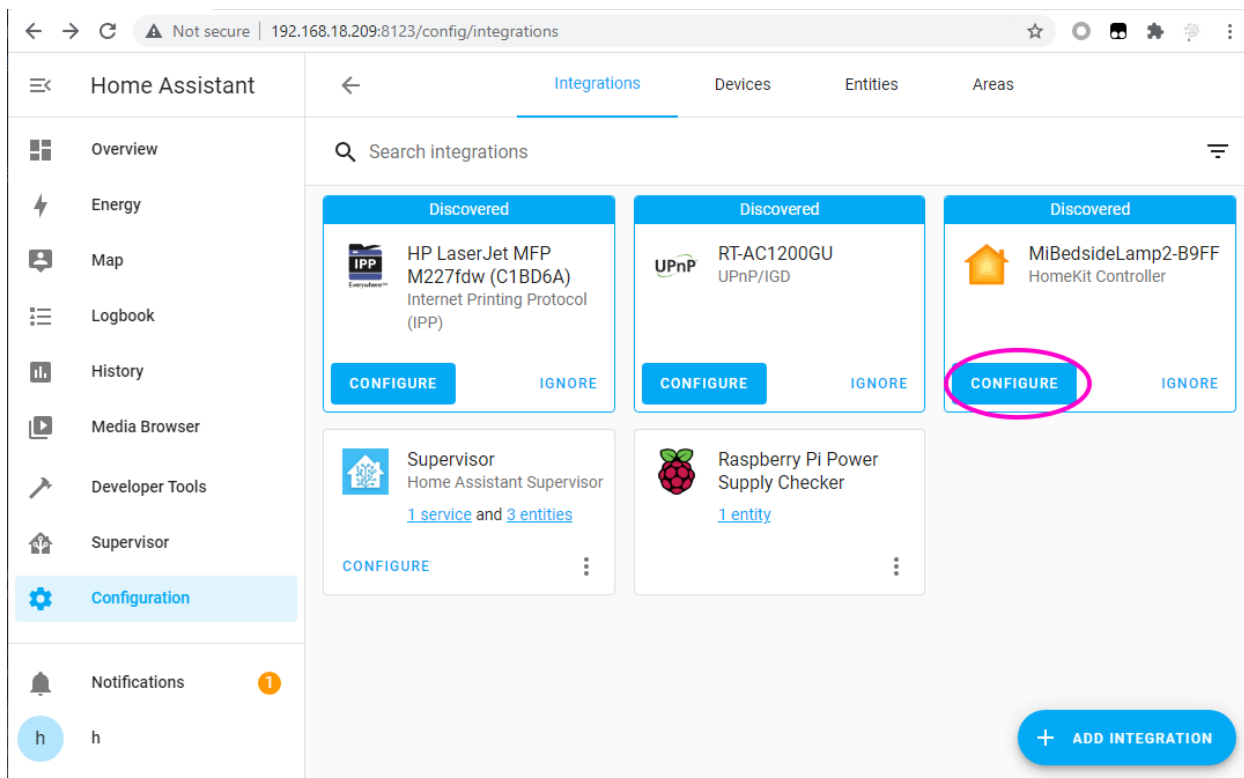


## 添加集成

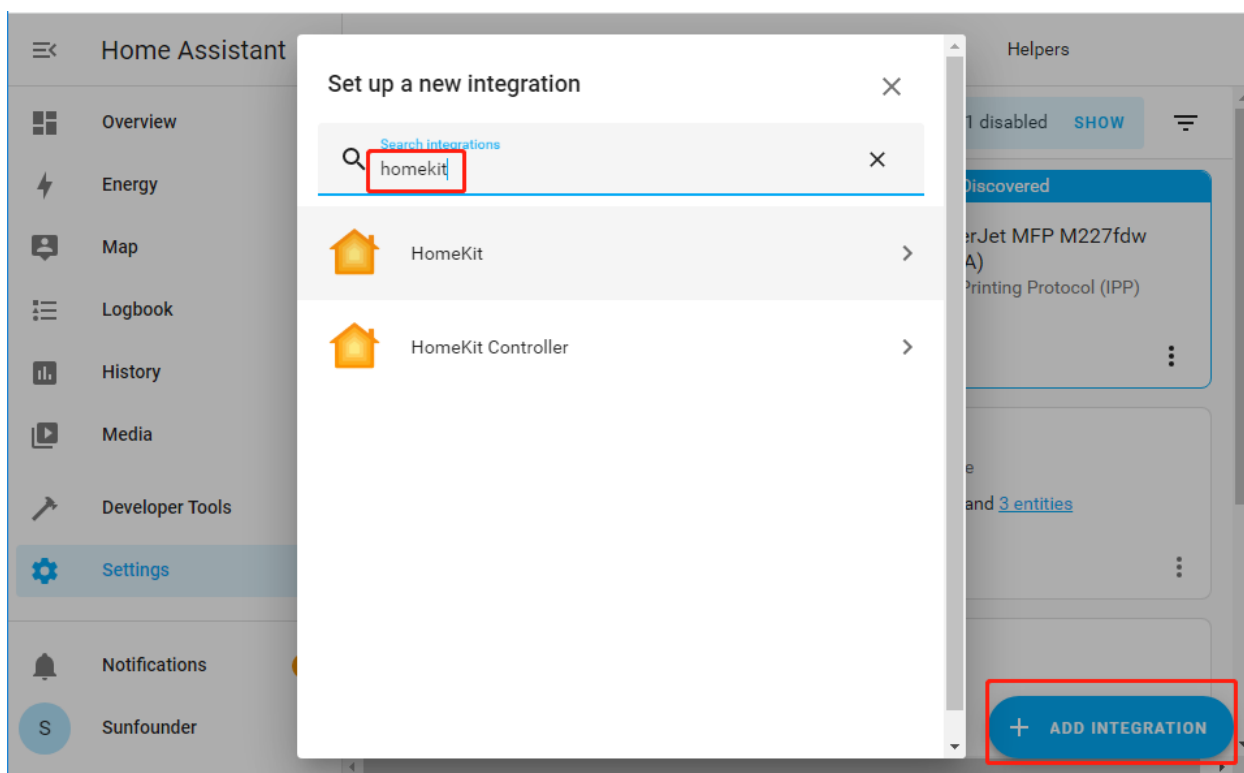
现在，访问 <http://ip:8123> 以进入您的 Home Assistant，然后单击左栏中的 **Settings** 并选择 **Devices/Services**。



如果 Home Assistant 在您的网络上找到一个设备，它会显示这个设备相关的集成，只需点击几下即可轻松添加。



如果您的智能设备没有被找到，请不要担心，单击右下角的 + 添加集成按钮并在列表中搜索您的集成。



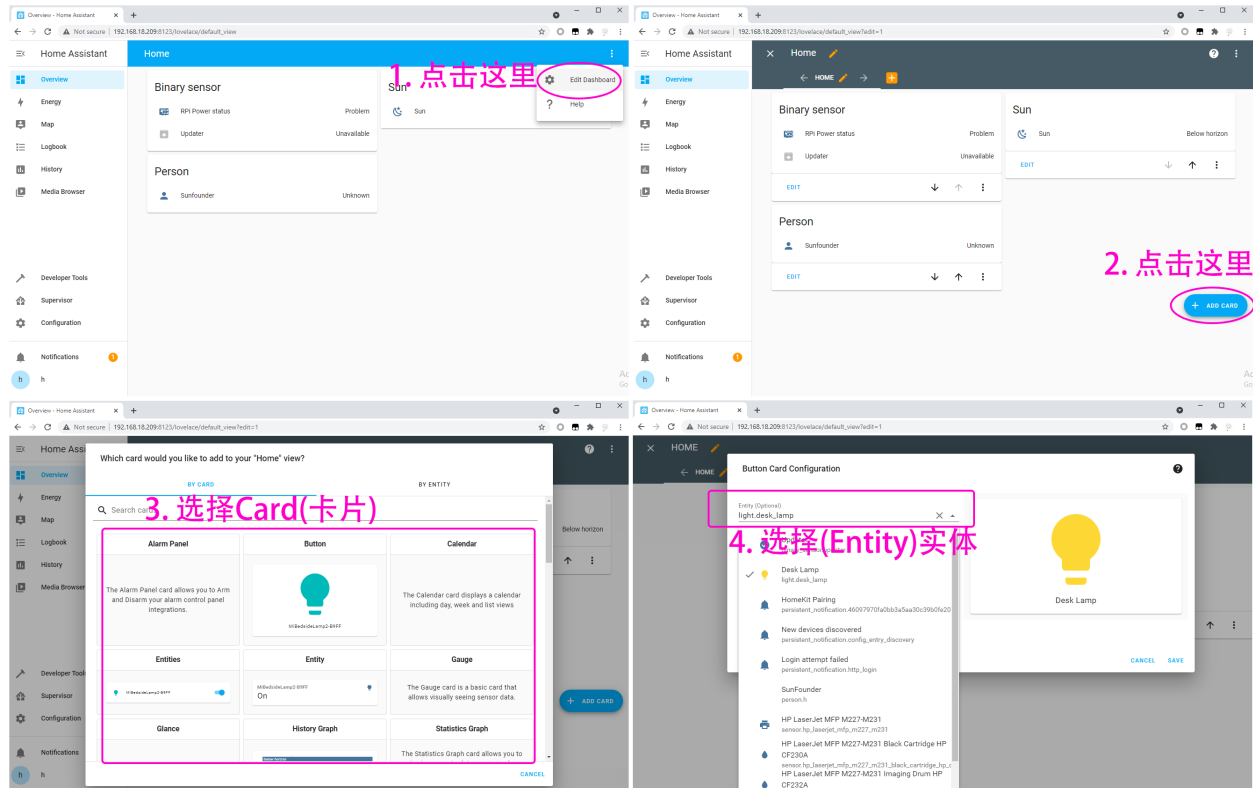
每种集成在使用方法上可能有所不同，您可以访问 [Home Assistant Integration](#) 了解详情。

## 编辑仪表板

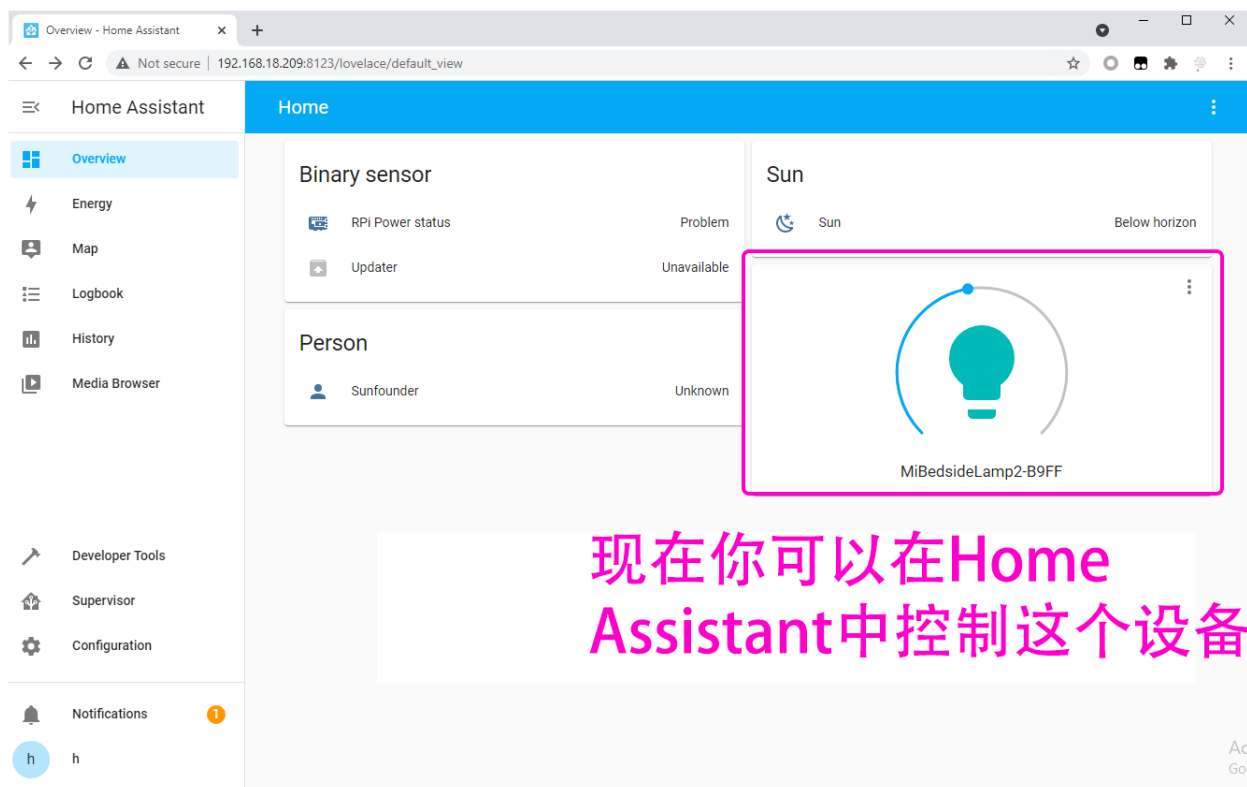
现在需要添加一个 CARD 来控制这个智能设备。

返回概览页面。如果您刚刚添加的设备没有出现，您需要 **编辑仪表板**。

点击 **Overview** --> **Edit Dashboard** --> **ADD CARD**, 可以根据需要选择对应的 CARD, 比如这里选择 **Button**, 然后选择对应的 Entity。



添加完卡片后，点击 Done 来退出编辑，这样你就能开始控制你的智能设备。



### 6.4.5 添加带有 ESP32 的 DIY 设备

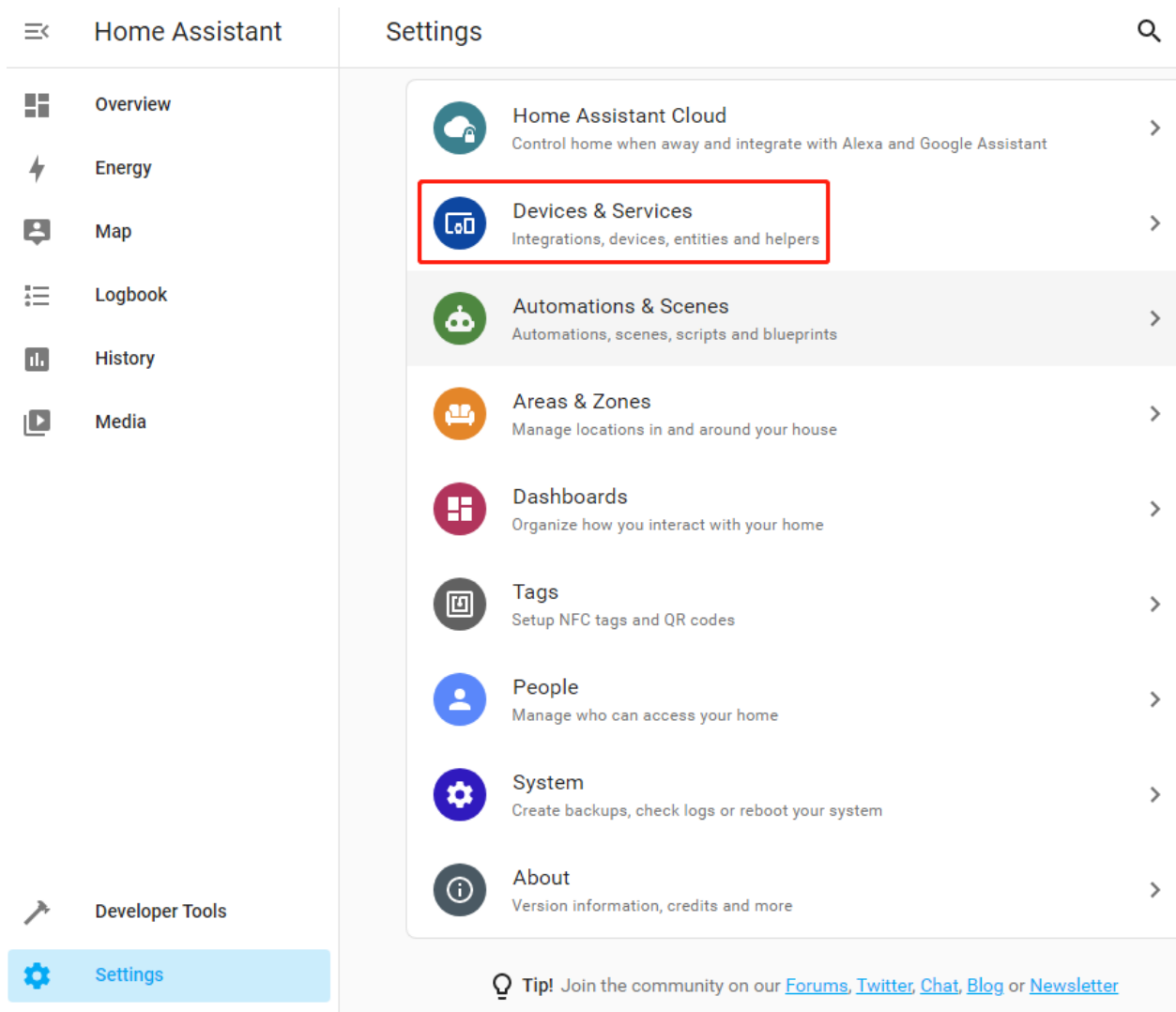
您可以将您购买的智能设备添加到 Home Assistant 中，例如智能音响、智能台灯等。您也可以使用 ESP32 制作您的 DIY 设备并添加。

在本节中，您将学习如何添加 DIY 设备。

在开始以下操作之前，您需要完成安装和配置 *ESPHome*。

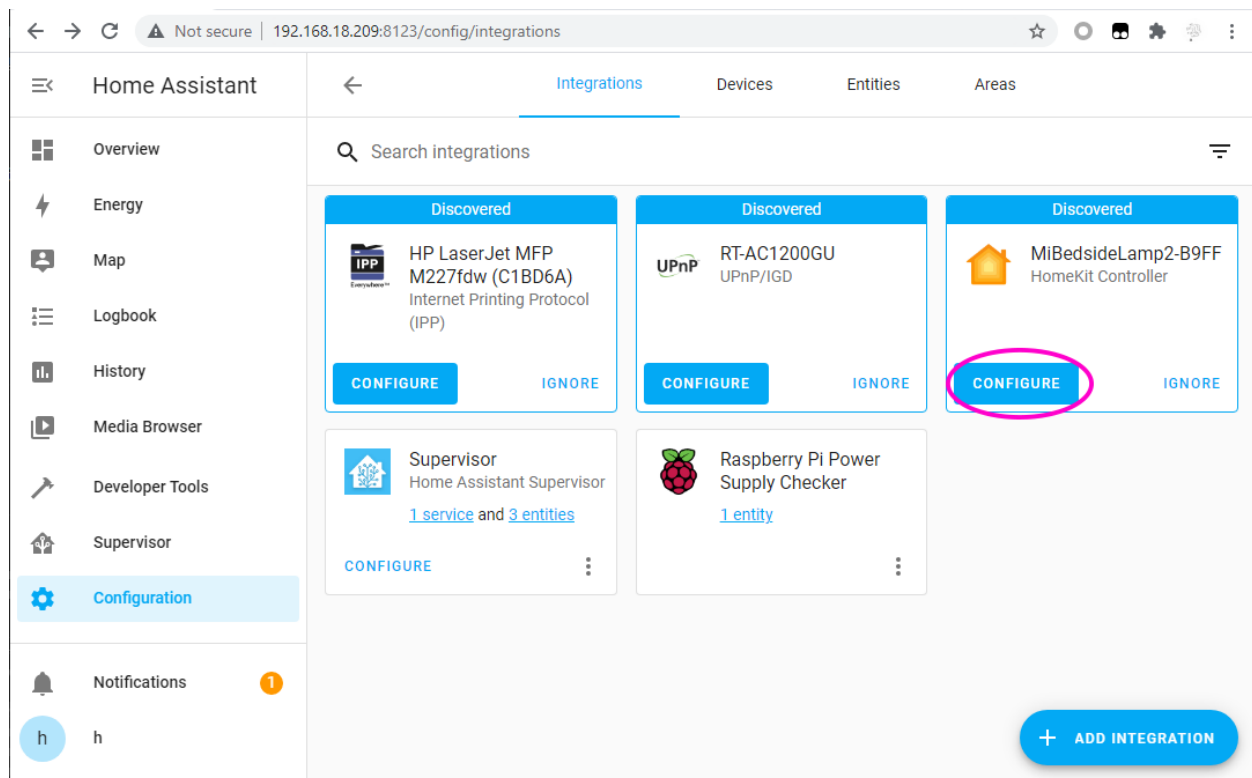
#### 添加集成

1. 现在，访问 <http://ip:8123> 以进入您的 Home Assistant，然后单击左栏中的 **Settings** 并选择 **Devices/Services**。

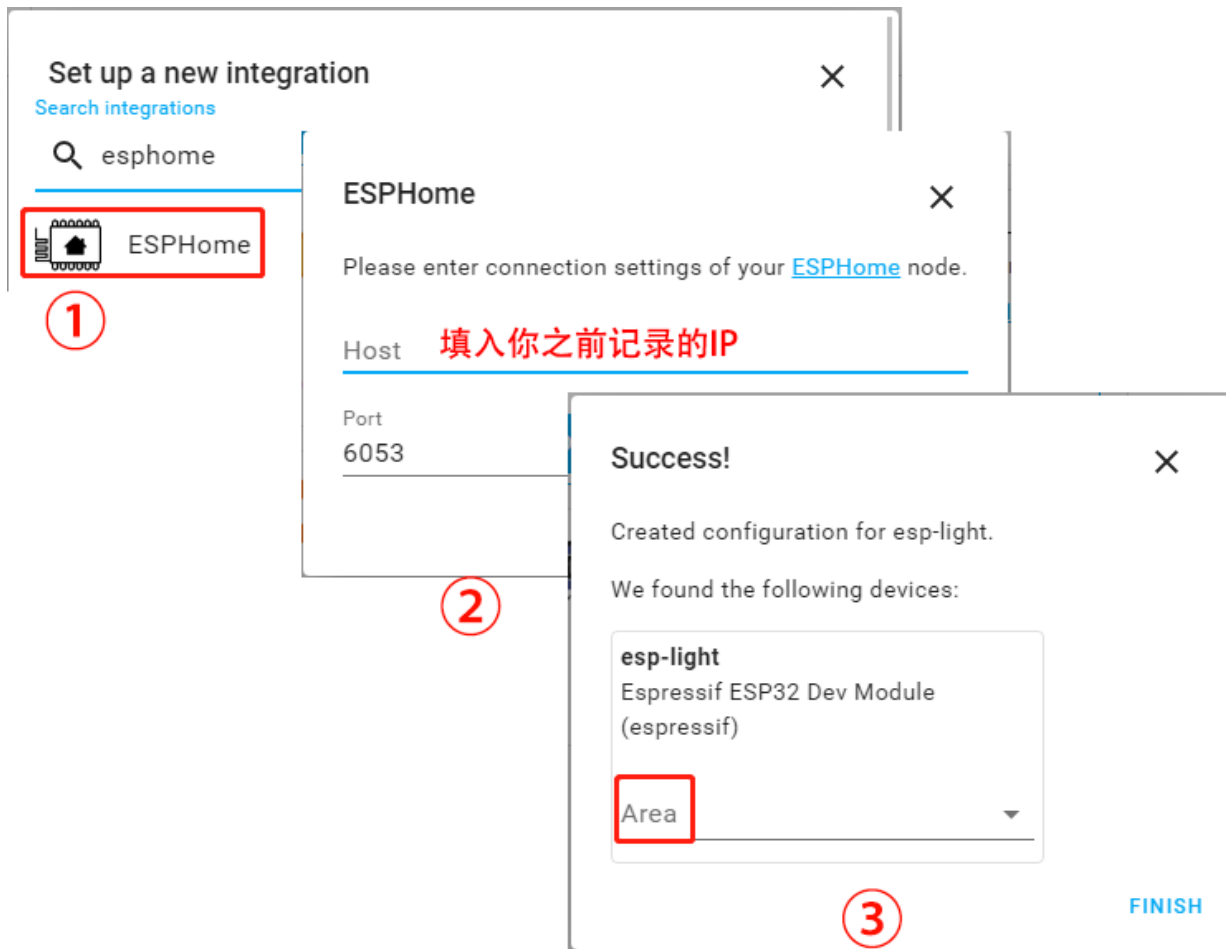


2. 如果 Home Assistant 在您的网络上找到设备，它会显示相应的集成，只需点击即可将其轻松添加。如果您的智能设备尚未被发现，请不要担心，您可以通过右下角的 **+ ADD INTERGATION** 按钮添加它。

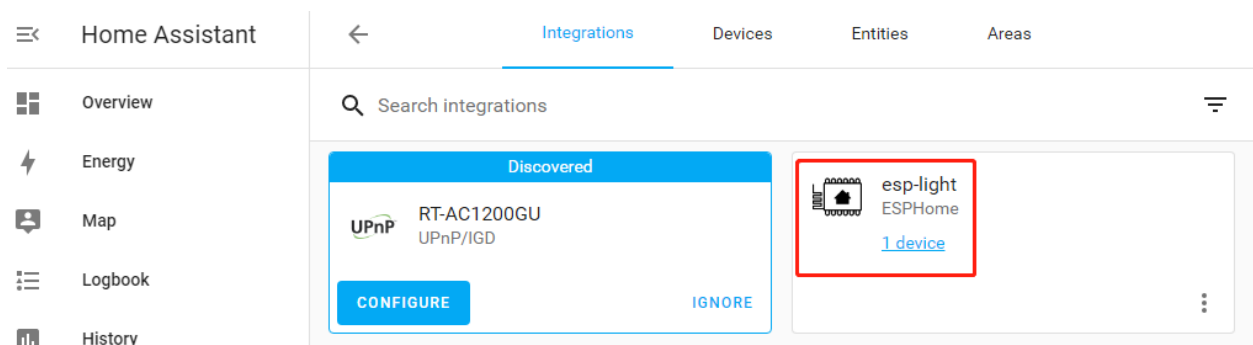




3. 在弹出的窗口中搜索 ESPHome 并填写 Host (IP 在安装和配置 *ESPHome* 章节的 INSTALL 步骤后有记录), 可以将你的 esp-light 放置在卧室、厨房或其他区域。



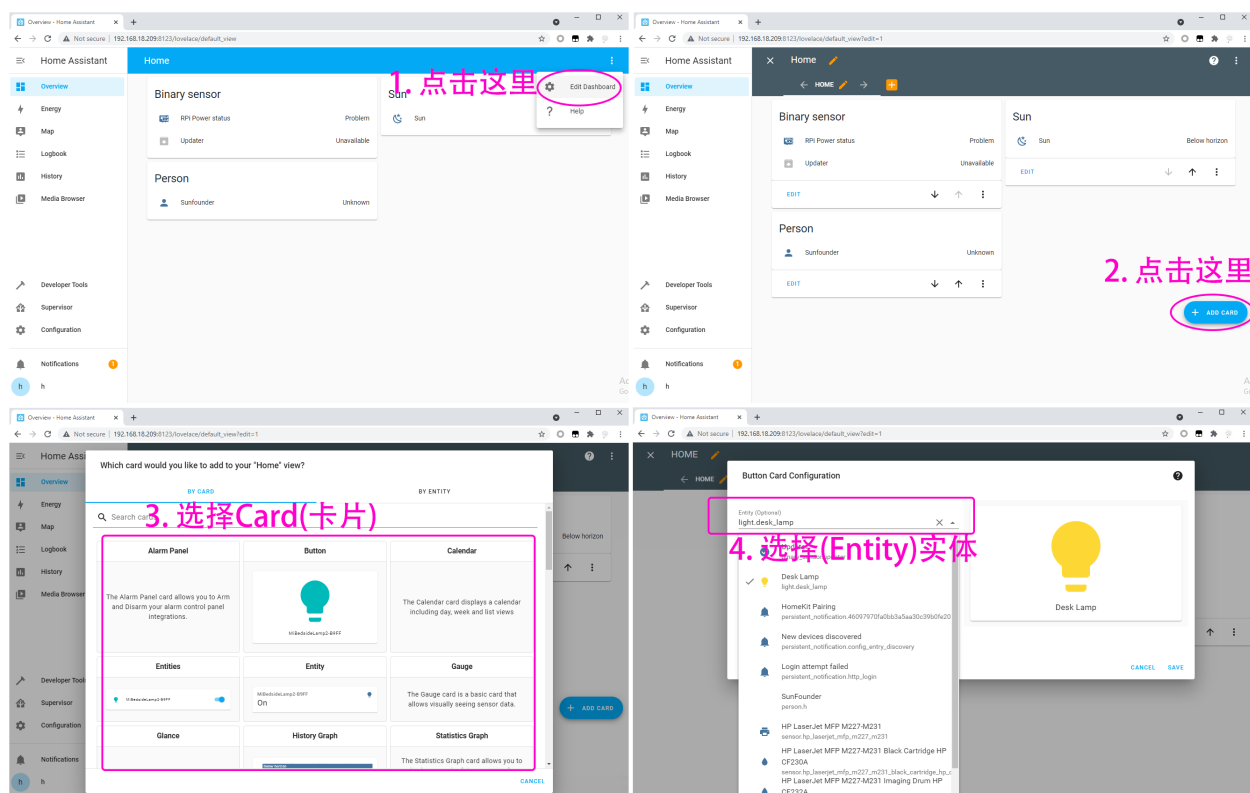
4. 单击完成后，它会在列表中出现。



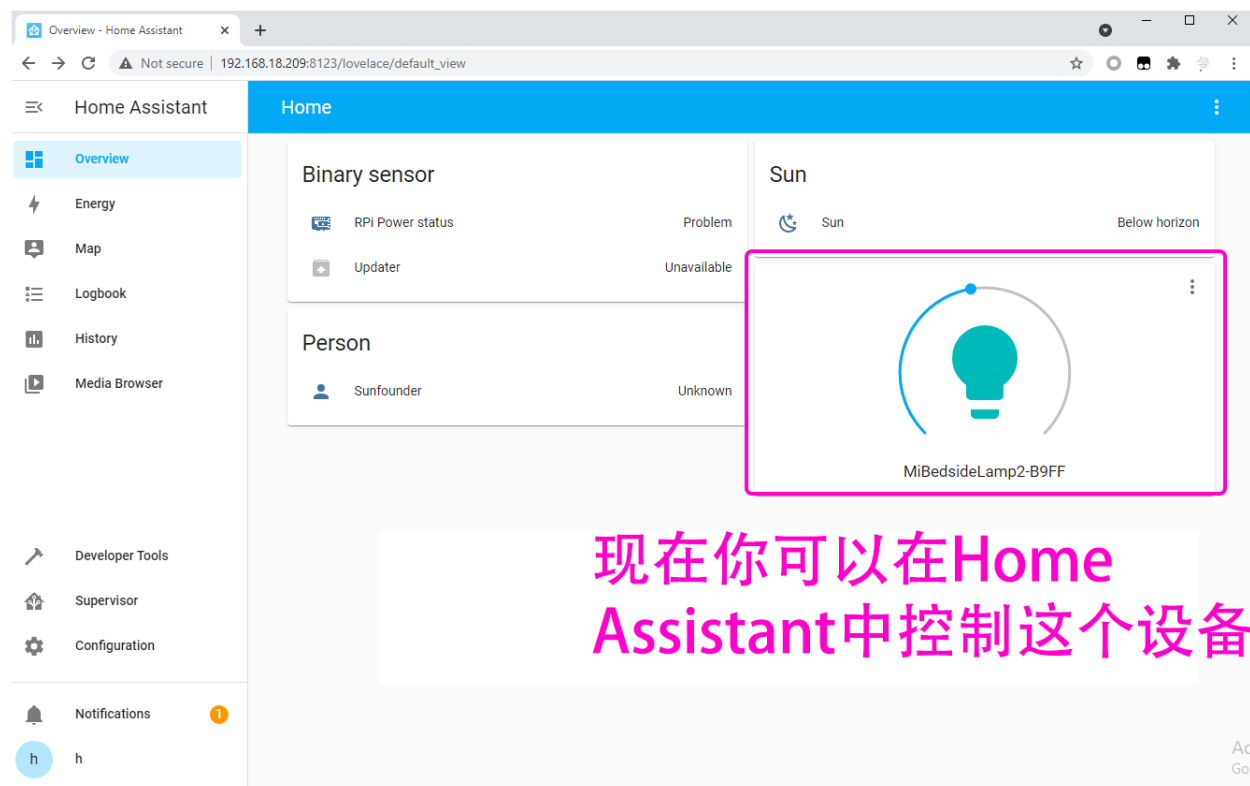
### 编辑仪表板

现在需要添加一个 CARD 来控制这个 esp-light。

点击 **Overview** --> **Edit Dashboard** --> **ADD CARD**, 可以根据需要选择对应的 CARD, 比如这里选择 **Button**, 然后选择对应的 Entity。



点击 SAVE 后，您将可以使用该按钮来控制 ESP-light（在 ESP32 板中将 LED 连接到 GPIO16，然后就可以用该按钮来控制 LED 了）。



参考 [官方 Demo](#) 可以找到更多风格的 DIY 设备。

### 6.4.6 在开机时启动 Home Assistant 服务

正常情况下，Docker 默认开机后是自启动的，这也意味着你安装了 Home Assistant 后，只要启动树莓派就可以使用了。

您可以使用以下命令查看引导列表。确认 `docker.service` 处于 `enabled` 的状态下。

```
sudo systemctl list-unit-files | grep enable
```

想关闭 Docker 的自启动，可以输入以下命令：

```
sudo systemctl disable docker.service
```

开启 Docker 的自启动，则可以输入以下命令：

```
sudo systemctl enable docker.service
```

### 6.4.7 启动时进入 Home Assistant 页面

先体验一下全屏自启动界面。请先关闭已经打开的 Home Assistant 并运行命令：`chromium-browser --start-fullscreen "http://localhost:8123"`。

如果您想在开机后自动全屏显示 Home Assistant 界面，请执行以下操作。

1. 进入 `autostart` 文件夹。

```
cd /home/pi/.config/autostart/
```

**注解：**如果 `autostart` 文件夹不存在，则需要创建一个新文件夹。

```
sudo mkdir -p /home/pi/.config/autostart/
```

2. 创建 `chrome_start_fullscreen.desktop` 文件。

```
sudo nano chrome_start_fullscreen.desktop
```

3. 编辑 `chrome_start_fullscreen.desktop` 文件如下。

```
[Desktop Entry]
Type = Application
Exec = chromium-browser --start-fullscreen "http://localhost:8123"
```

保存并退出依次按：Ctrl + X, Y, Enter.

**注解：**如果要取消全屏启动，将 `.desktop` 文件内容用 “#” 注释掉并重启树莓派。

```
cd /home/pi/.config/autostart/
sudo nano chrome_start_fullscreen.desktop
```

4. 退出全屏。

**计算机:**

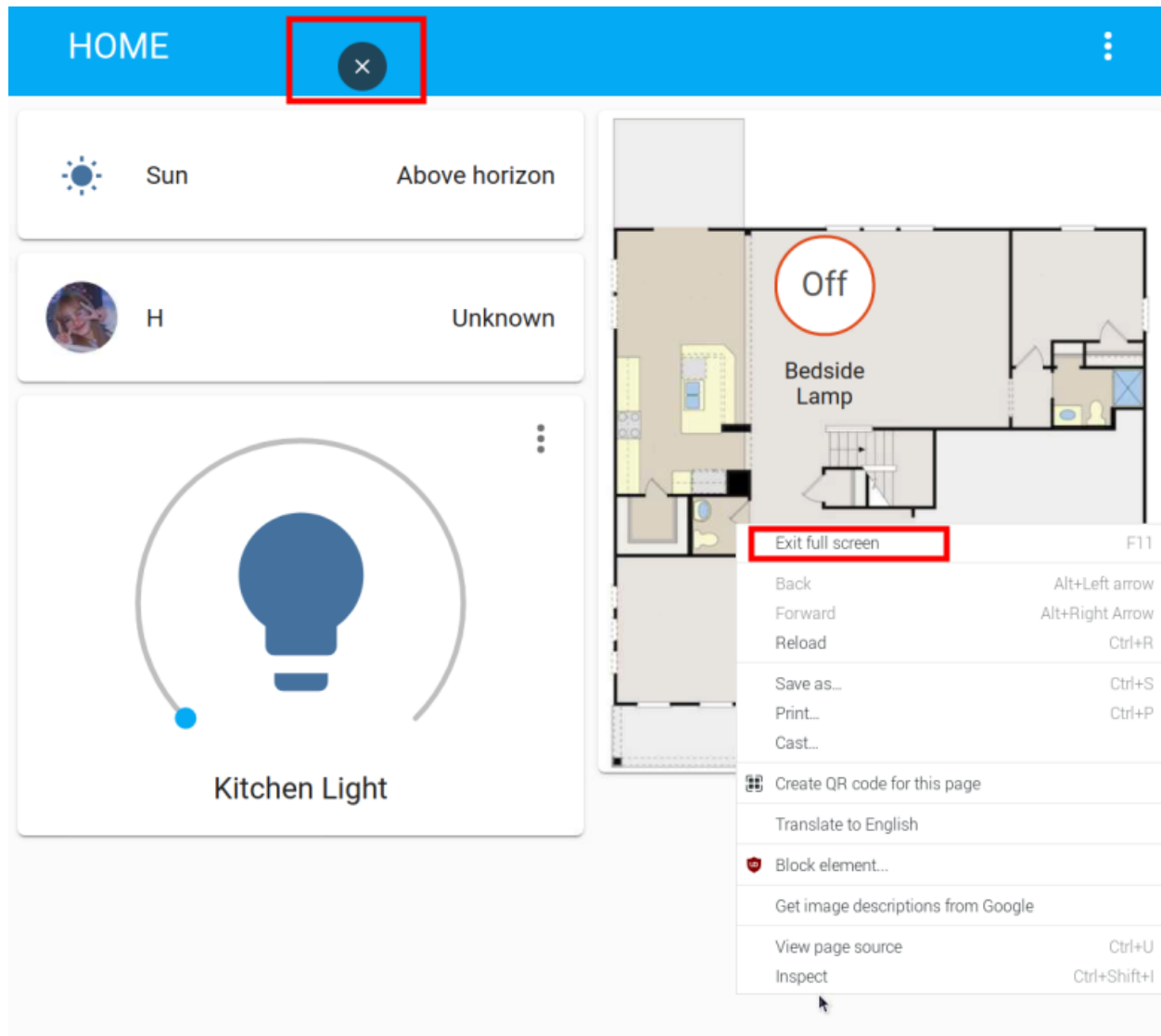
方法一：按 F11。

方法二：将鼠标移动到屏幕上一栏，点击出现的退出按钮。

方法三：右键弹出菜单，选择“退出全屏”。

**触摸屏:**

长按空白处弹出菜单，点击屏幕顶部的退出按钮或选择“退出全屏”。



### 6.4.8 Home Assistant 文件结构

Home Assistant 默认会在 `~/.homeassistant` 路径下创建一个配置文件。

文件目录结构如下：

```
Code
1 → .homeassistant tree -a -I 'deps'
2
3 |— .HA_VERSION
4 |— .cloud
5 |— .storage
6 |   |— auth
7 |   |— auth_provider.homeassistant
8 |   |— cloud
9 |   |— core.area_registry
10 |   |— core.config
11 |   |— core.config_entries
12 |   |— core.device_registry
13 |   |— core.entity_registry
14 |   |— core.restore_state
15 |   |— frontend.user_data_7a04118ad289497dbaf068d2b098b016
16 |   |— onboarding
17 |   |— person
18 |— automations.yaml
19 |— configuration.yaml
20 |— customize.yaml
21 |— groups.yaml
22 |— home-assistant.log
23 |— home-assistant_v2.db
24 |— home-assistant_v2.db-shm
25 |— home-assistant_v2.db-wal
26 |— scripts.yaml
27 |— secrets.yaml
28 |— tts
29
30 3 directories, 23 files
```

- `.storage` 目录包含了很多与用户相关的信息，包括用户登录信息（用户名/密码，在 `auth_provider.homeassistant` 文件中加密）。
- `configuration.yaml`: 用户编辑的配置文件。
- `home-assistant.log`: 运行日志（每次重新启动时会清除）。
- `home-assistant_v2.db`: 数据库
- `.storage`: 前端配置的各种元素。

## 6.5 Cloud4RPI

Cloud4RPI 是用于物联网项目的基于云的控制面板。Cloud4RPI 使用仪表板小部件远程控制物联网设备并显示实时设备数据。

### 6.5.1 Cloud4RPI 的快速使用指南

#### 在您的树莓派中安装 Cloud4RPI

建议在安装前更新您的系统。

```
sudo apt update && sudo apt upgrade -y
```

安装/更新所需的软件包。

```
sudo apt install git python3 python3-pip -y
```

使用以下命令安装 Cloud4RPI:

```
sudo pip3 install cloud4rpi
```

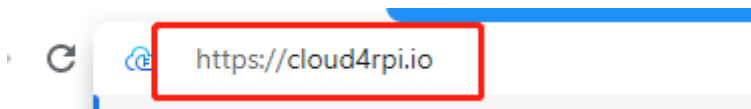
---

**注解:** 本教程提供 Python 3.0 及以上版本的安装方法。对于较低版本的 Python, 可能需要修改命令。

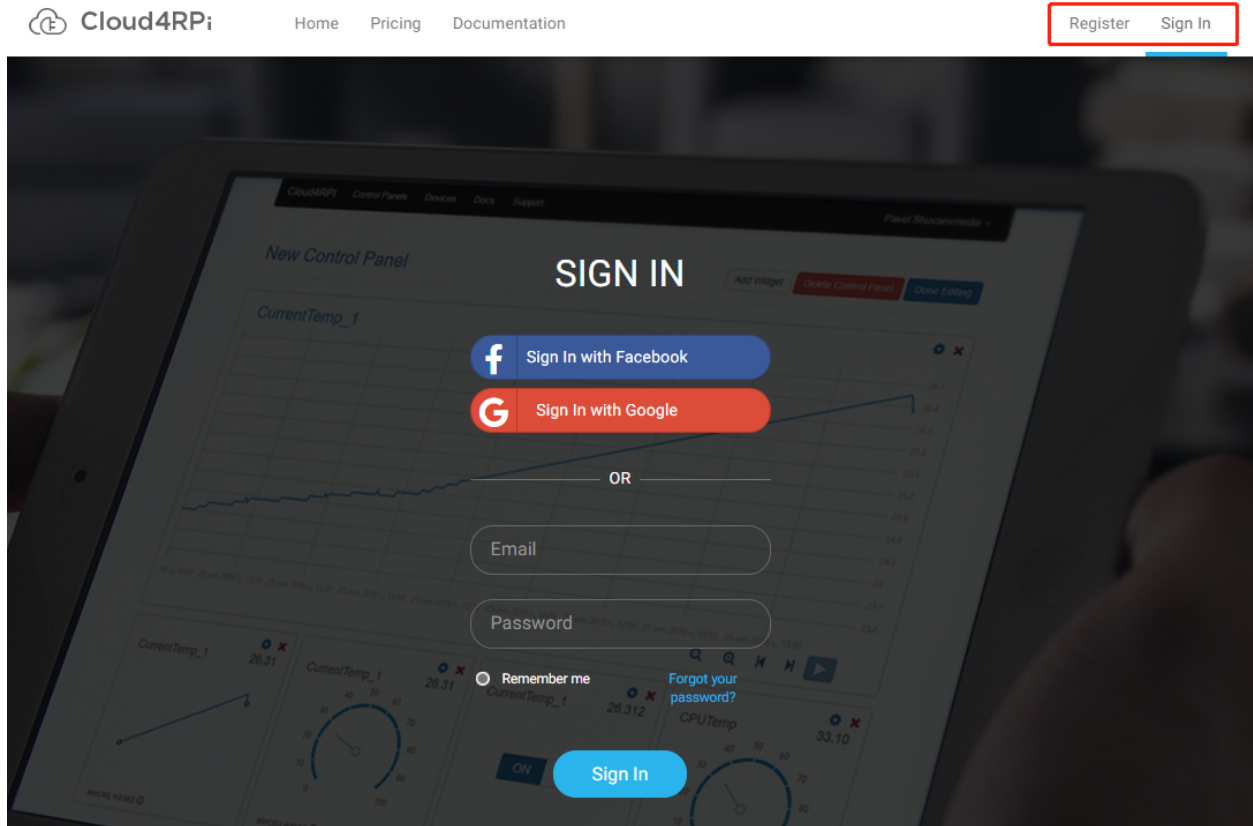
---

#### 在计算机上登录 Cloud4RPI

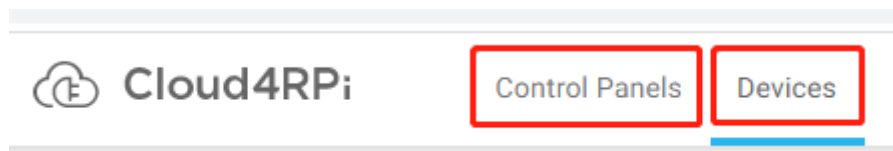
打开 Chromium Web 浏览器并输入链接: <https://cloud4rpi.io>.



在使用之前, 您需要注册并登录您的 Cloud4RPI 帐户。

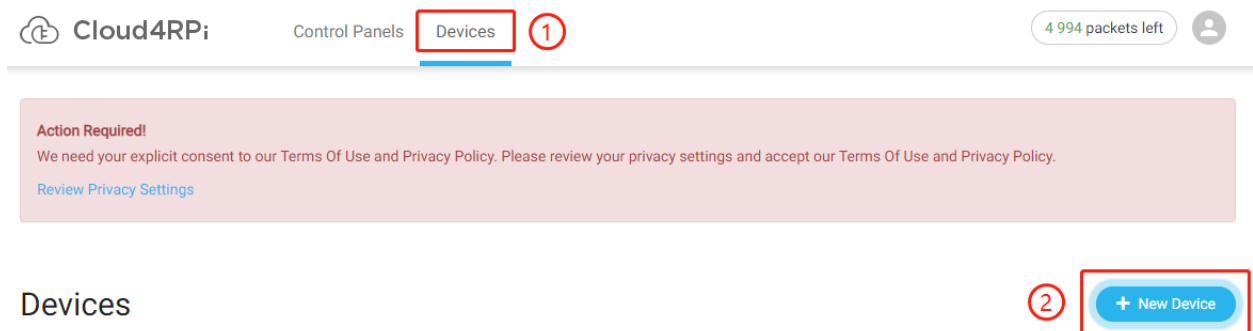


登录完成后可以看见如下图所示的选项，**Devices** 用于显示设备的连接，**Control Panels** 用于显示来自网页或者树莓派的信息。



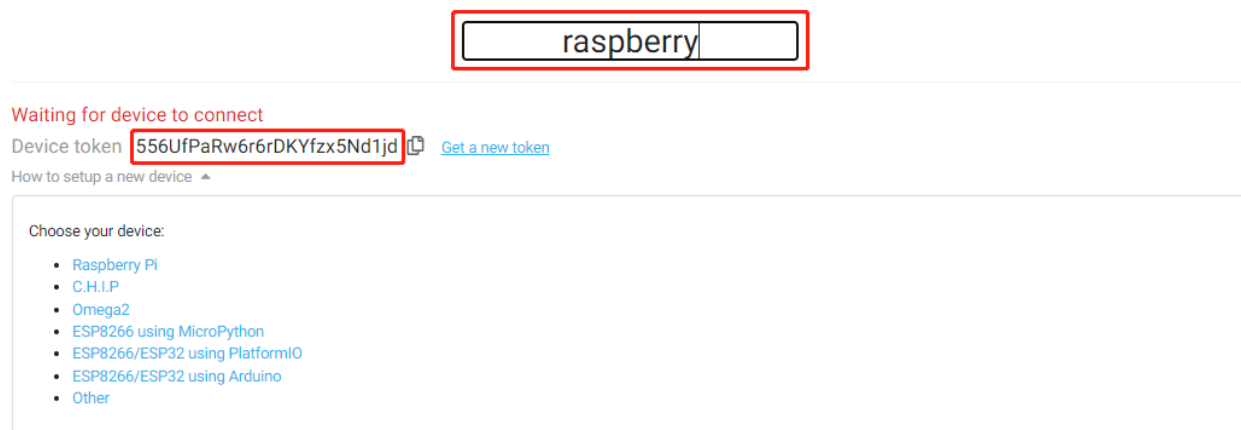
### 将您的树莓派连接到 Cloud4RPI

登录后，单击 **Devices** 选项，然后单击 **New Device**。



为新设备命名，并复制当前的 **Device token**。





打开树莓派终端，下载 Cloud4RPI 提供的测试示例。

```
cd /home/pi
git clone https://github.com/cloud4rpi/cloud4rpi-raspberrypi-python.git
```

使用文本编辑器（如 Nano）修改 control.py 文件内容。

```
cd cloud4rpi-raspberrypi-python
sudo nano control.py
```

找到以下代码行。

```
DEVICE_TOKEN = '__YOUR_DEVICE_TOKEN__'
```

将 \_\_YOUR\_DEVICE\_TOKEN\_\_ 替换成从 Cloud4RPI 上复制的 **Device token** 按 Ctrl + X 保存修改的内容。

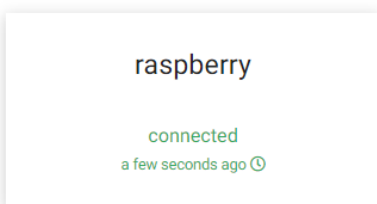
```
DEVICE_TOKEN = '556UfPaRw6r6rDKYfzx5Nd1jd'
```

运行示例文件 control.py。

```
sudo python3 control.py
```

返回 Cloud4RPI 的服务页面。如果您的树莓派连接到 Cloud4RPI，设备信息将变成绿色。

### Devices

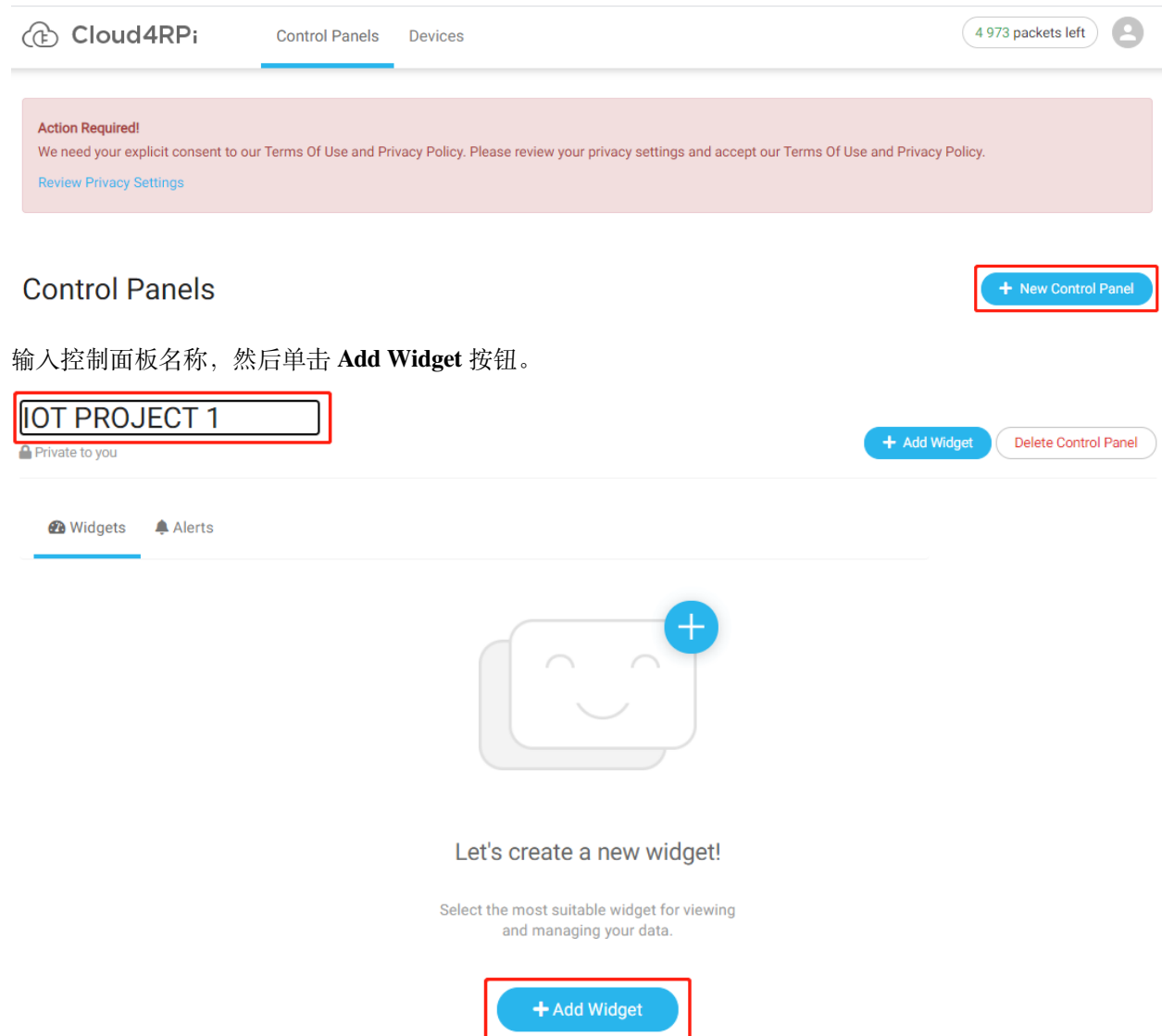
[+ New Device](#)

## 使用 Cloud4RPI 控制面板

Cloud4RPI 服务器上的控制面板为 IoT 应用程序提供了许多有用的小部件。这些小部件可以用来控制组件或显示传感器数据。

进入 **Control Panels** 之后, 单击 **New Control Panel** 按钮。

**注解:** 在使用控制面板之前要保持运行 `control.py` 文件。



Cloud4RPI Control Panels Devices 4 973 packets left

**Action Required!**  
We need your explicit consent to our Terms Of Use and Privacy Policy. Please review your privacy settings and accept our Terms Of Use and Privacy Policy.  
[Review Privacy Settings](#)

### Control Panels

+ New Control Panel

IOT PROJECT 1

Private to you + Add Widget Delete Control Panel

Widgets Alerts

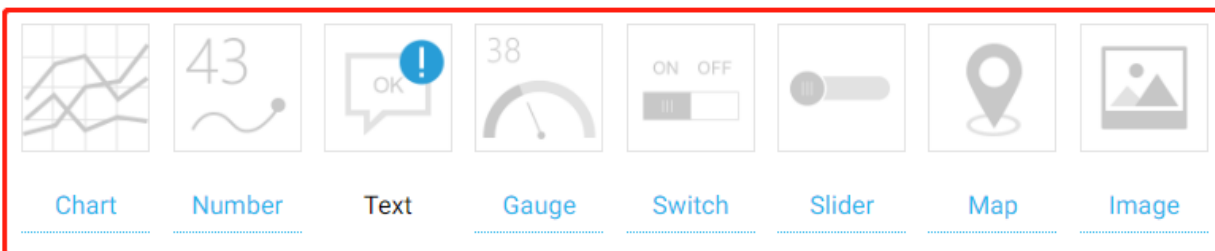
Let's create a new widget!

Select the most suitable widget for viewing and managing your data.

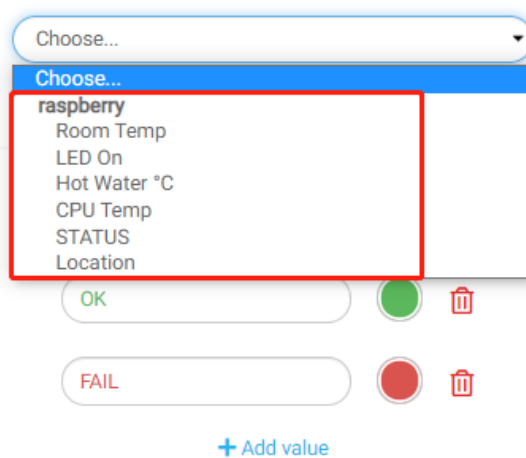
+ Add Widget

再次点击 **Add Widget** 按钮之前, 需要选择合适的小部件和要显示的变量。

## New Widget



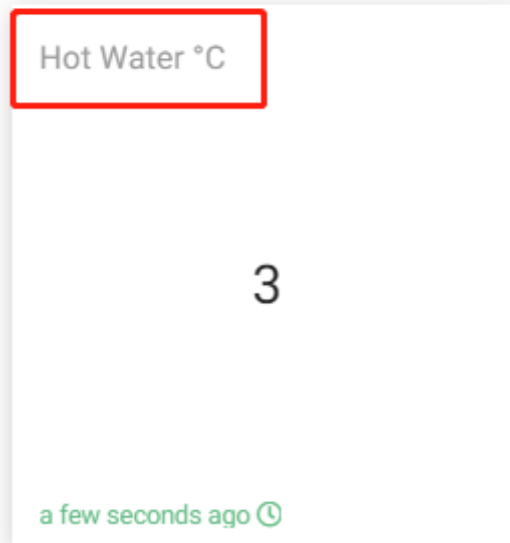
## Variable to display



例如，选择 **Text** 之后再选择 **Hot Water °C**，这样我们就能再新创建的控制面板上看到它了。

# IOT PROJECT 1

Private to you



**注解：**我们会在[详细了解 control.py](#) 章节详细介绍这些显示的数据。

## 6.5.2 详细了解 control.py

为了更好的理解后面的项目，我们先来看看 Cloud4RPI 官方提供的示例代码。

**注解：**在查看本章之前，请先完成上一章[Cloud4RPI 的快速使用指南](#)。

打开 control.py 文件。

```
cd cloud4rpi-raspberrypi-python
sudo nano control.py
```

Cloud4RPI 会为每个设备设置一个设备令牌，您需要用正确的设备令牌填写变量 `DEVICE_TOKEN` 才能连接到相应的设备。

```
DEVICE_TOKEN = '556UfPaRw6r6rDKYfzx5Nd1jd'
```

`variables` 是一个二维字典，它的每个键对应的值也是一个字典。

`variables` 字典的键包括 'Room Temp', 'LED On', 'CPU Temp', 'STATUS', 'Location', 它们都是显示在控制面板小部件上的数据。

`variables` 字典中每个键的值也是一个字典，这些字典的键是一样的，其中 'bind' 键的值是一个可以返回传递内容的函数，是传递内容 'type' 的数据类型 ('numeric' 是数字类型, 'string' 是字符串类型, 'bool' 是布尔类型, 'location' 是包含两个字典的列表)。

通过对 `variables` 字典的分析，我们可以知道 `Cloud4RPi` 会读取 `variables` 字典的键值，并在控制面板中显示键值对应的值。显然我们不能改变 `variables` 字典的键，但是我们可以改变键中的值（'bind' 字典的值），让 `widgets` 显示我们想要的内容。

比如我们要向 `Cloud4RPi` 发送湿度值，虽然不能在 `variables` 字典中添加一个新的键，但是可以借用其中的键，比如 'Room Temp'，然后在其 'bind' 的值里面编写返回湿度值的函数。

```
variables = {
    'Room Temp': {
        'type': 'numeric' if ds_sensors else 'string',
        'bind': ds_sensors[0] if ds_sensors else sensor_not_connected
    },
    'LED On': {
        'type': 'bool',
        'value': False,
        'bind': led_control
    },
    'CPU Temp': {
        'type': 'numeric',
        'bind': rpi.cpu_temp
    },
    'STATUS': {
        'type': 'string',
        'bind': listen_for_events
    },
    'Location': {
        'type': 'location',
        'bind': get_location
    }
}
```

`vdiagnostics` 中存储有关树莓派的信息，并用于验证，防止错接。

```
vdiagnostics = {
    'CPU Temp': rpi.cpu_temp,
    'IP Address': rpi.ip_address,
    'Host': rpi.host_name,
    'Operating System': rpi.os_name,
    'Client Version': cloud4rpi.__version__,
}
```

### 6.5.3 项目

在本节中，您将开始学习创建 IoT 项目。

在开始项目之前，应该确保已经下载了我们提供的代码。

打开终端并输入以下命令从 `github` 上下载。

```
git clone https://github.com/sunfounder/raspad3.git
```

然后将代码从 `iot` 文件夹复制到 `cloud4rpi-raspberrypi-python` 文件夹。

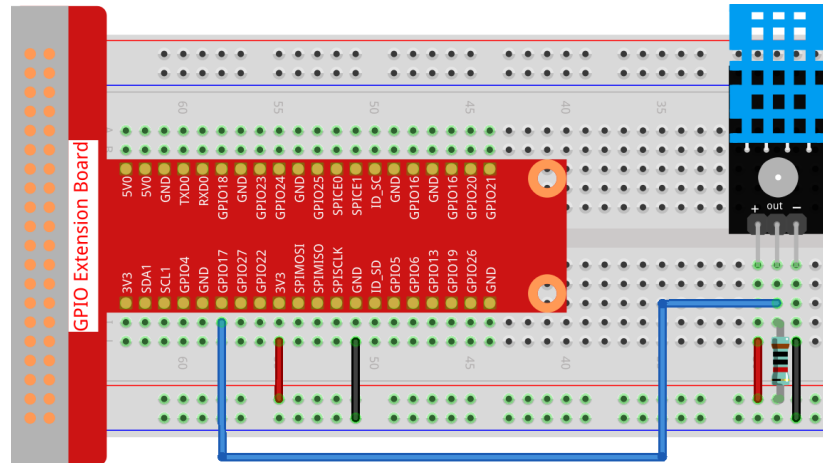
```
cp -r /home/pi/raspad3/iot/* /home/pi/cloud4rpi-raspberrypi-python
```

## 温湿度采集

在这个项目中，我们使用 DHT11 读取周围的温湿度，然后在 Cloud4RPI 上显示出来。

## 实验步骤

搭建电路。



打开代码。

```
cd /home/pi/cloud4rpi-raspberrypi-python
sudo nano temperature_and_humidity_acquisition.py
```

找到以下部分并填写正确的令牌。

```
DEVICE_TOKEN = '__YOUR_DEVICE_TOKEN__'
```

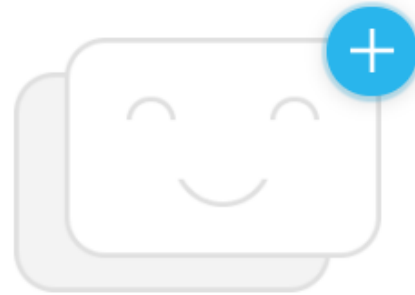
运行代码。

```
sudo python3 temperature_and_humidity_acquisition.py
```

转到 Cloud4RPI 页面, 添加一个名为 project1 的新控制面板, 然后单击 **Add Widget**.

project1

Private to you

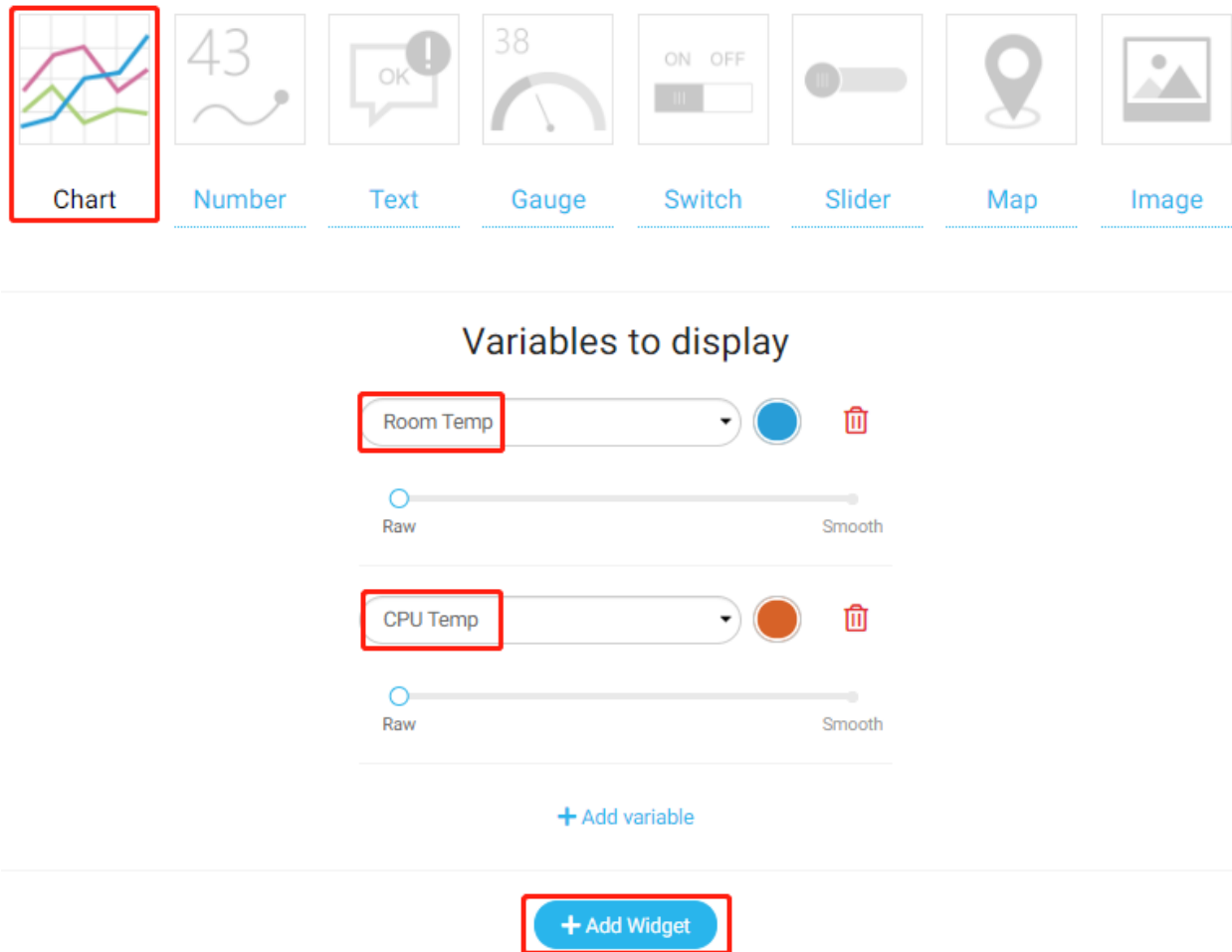


Let's create a new widget!

Select the most suitable widget for viewing  
and managing your data.

+ Add Widget

选择一个同时包含 **Room Temp** 和 **CPU Temp** 的 **Chart** 小部件作为要显示的变量。



添加后，您将看到温度（Room Temp）和湿度（CPU Temp）随时间变化的曲线，可以通过单击右下角的选项来调整显示窗口。

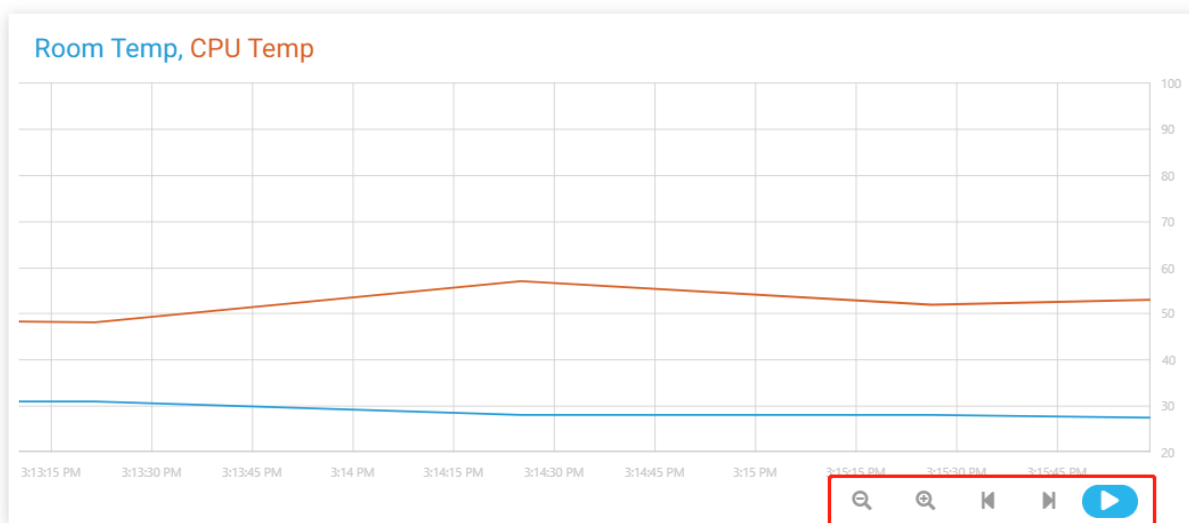


project1

Private to you

Share

Edit



## 代码说明

```
dht11 = DHT11,DHT11(17)
```

实例化一个 dht11 对象。

```
def getTem():
    result = dht11.get_result()
    while True:
        if result:
            break
        else:
            result = dht11.get_result()
    return result[1]
```

温度 (result[1]) 由该函数获取，直到读取成功才会中断该函数。

```
def getHum():
    result = dht11.get_result()
    while True:
        if result:
            break
        else:
            result = dht11.get_result()
    return result[0]
```

湿度 (result[0]) 是通过该函数获取的，同样是读取成功后才会中断。

```
variables = {
    'Room Temp': {
        'type': 'numeric',
        'bind': getTem
```

(下页继续)

(续上页)

```

    },
    'CPU Temp': {
        'type': 'numeric',
        'bind': getHum
    },
}

```

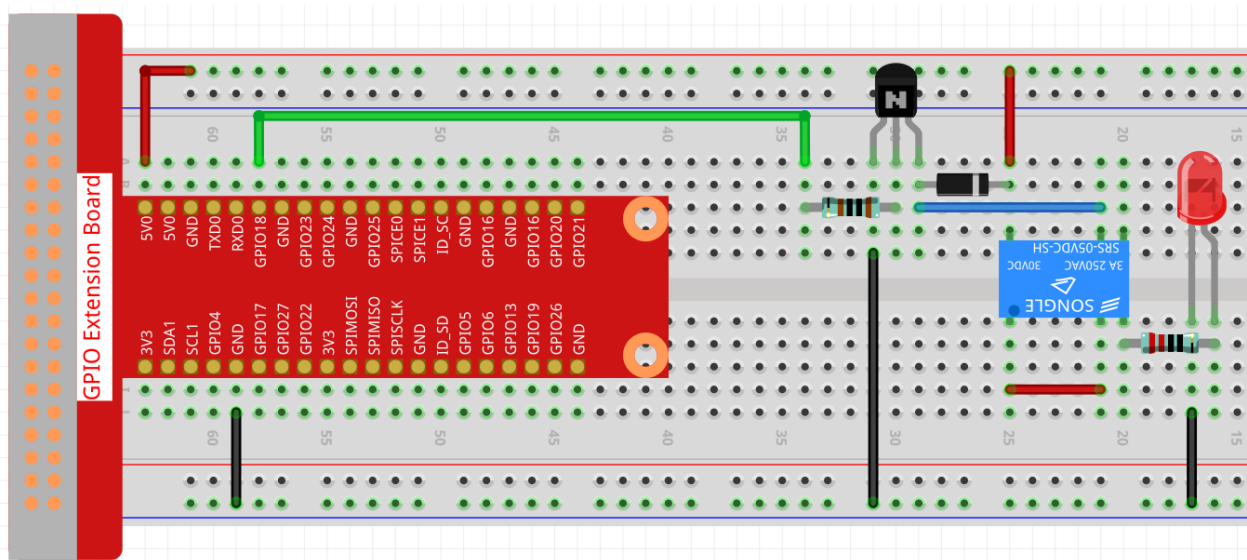
温度信息存储在 'Room Temp' 键中的 'bind' 里, 湿度信息存储在 'CPU Temp' 键中的 'bind' 里. 当 Cloud4RPI 收到 variables 时, 它会将其转换为网页可以识别的 JavaScript 对象。

## 遥控开关

在这个项目中, 我们将学习使用 Cloud4RPI 的开关来控制继电器, 从而控制继电器的负载 LED。您也可以将继电器的负载更改为家用电器, 但要注意安全。

## 实验步骤

搭建电路。



打开代码。

```

cd /home/pi/cloud4rpi-raspberrypi-python
sudo nano remote_switch.py

```

找到下面的部分并填写正确的设备令牌。

```

DEVICE_TOKEN = '__YOUR_DEVICE_TOKEN__'

```

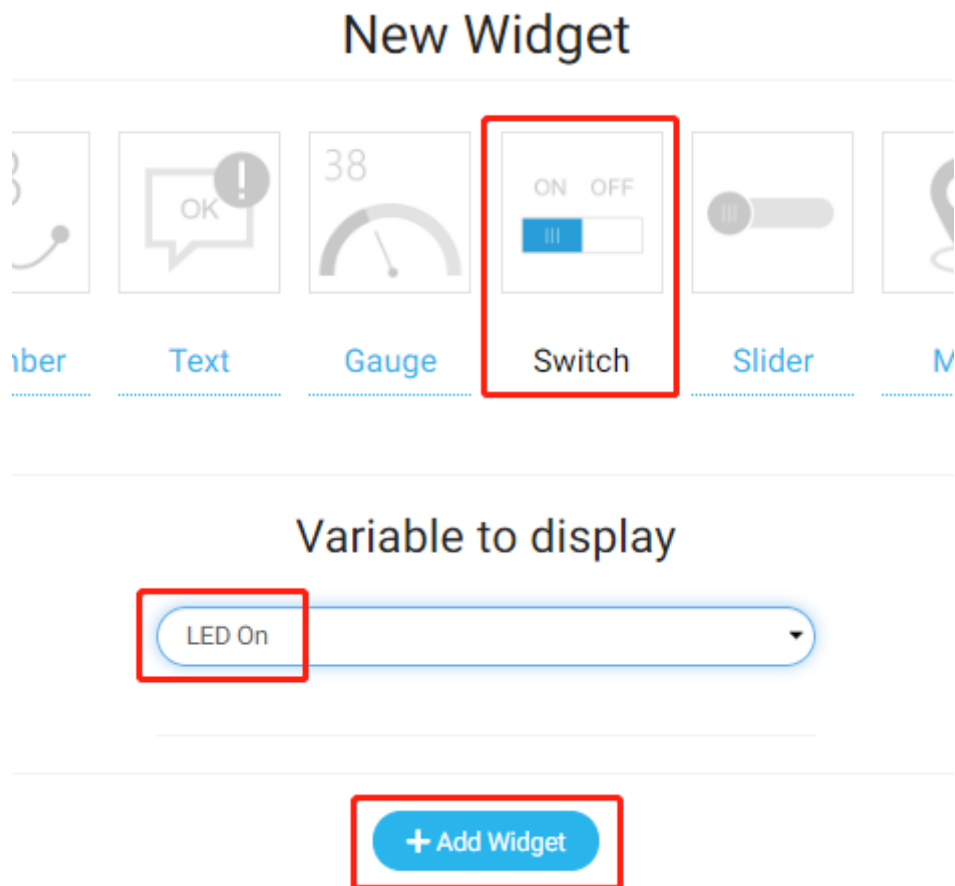
运行代码。

```

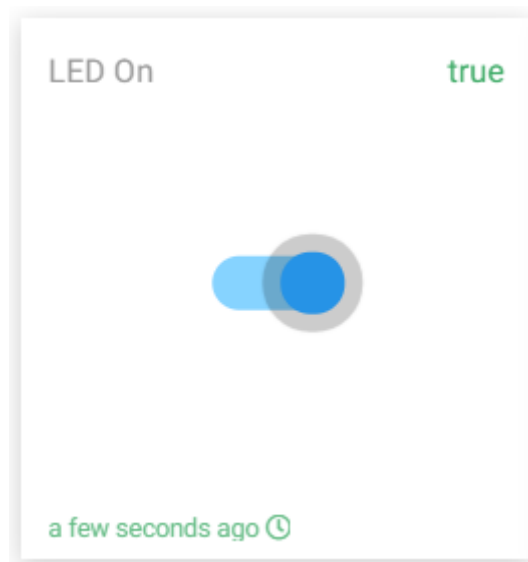
sudo python3 remote_switch.py

```

转到 Cloud4RPI 页面, 添加一个名为 project2 的新控制面板, 然后单击 **Add Widget** 添加一个 **Switch** 小部件。



添加后，您可以使用“Switch”小部件来控制继电器。



## 代码说明

```
RELAY_PIN = 18
```

继电器连接到 T 型扩展板的 GPIO18 引脚。

```
GPIO.setmode(GPIO.BCM)
GPIO.setup(RELAY_PIN, GPIO.OUT)
```

选择 BCM 命名模式并设置 RELAY\_PIN 为输出模式。

```
def relay_control(value=None):
    GPIO.output(RELAY_PIN, value)
    return GPIO.input(RELAY_PIN)
```

该函数通过 value 控制继电器，并返回继电器当前的电平。

**注解：** 由于继电器工作在高电平，当开关状态为 Ture 时继电器闭合，当开关状态为 False 时继电器打开

```
variables = {
    'LED On': {
        'type': 'bool',
        'value': False,
        'bind': relay_control
    },
}
```

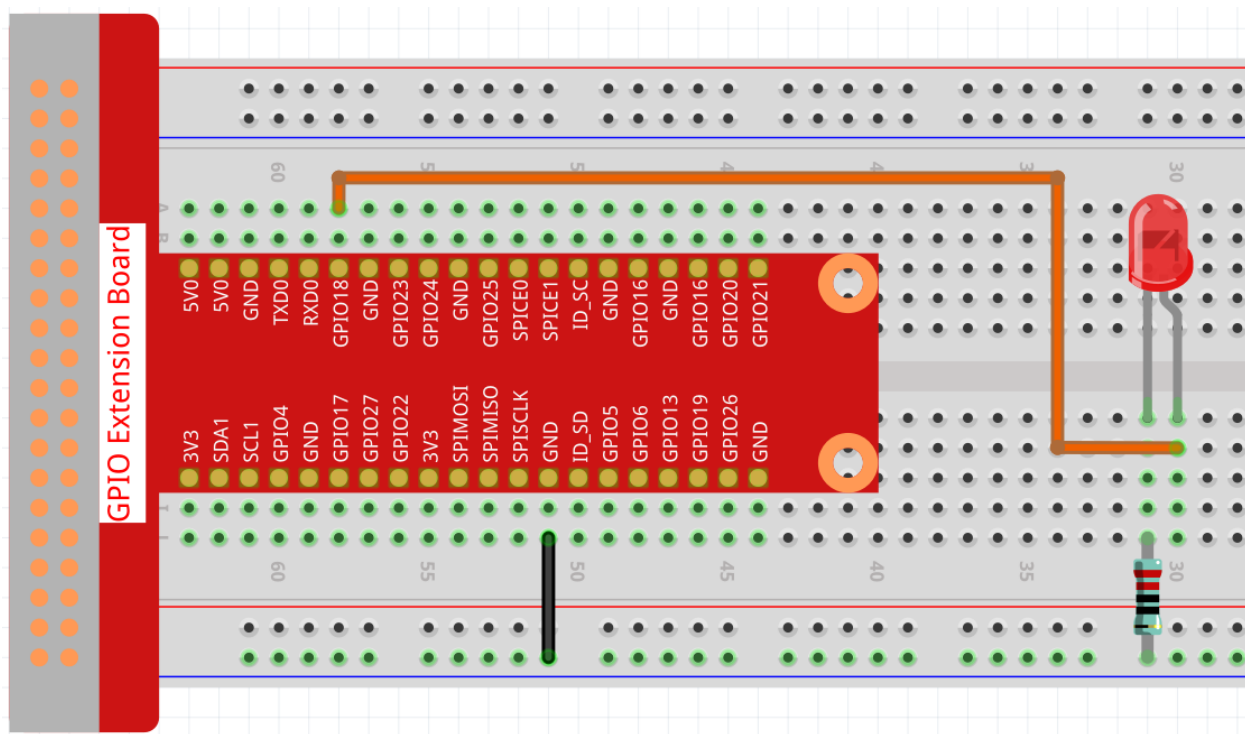
通过设置 'value' 键的值, 我们可以设置 Switch 部件的初始值, 然后循环读取 'value' 键的值, 并将继电器当前电平 RELAY\_PIN (函数 relay\_control() 的返回值) 发送到 Cloud4RPI。

## 智能灯

在这个项目中, 我们将学习使用 Cloud4RPI 的 Silder 小部件控制 LED 的亮度。

## 实验步骤

搭建电路。



打开代码。

```
cd /home/pi/cloud4rpi-raspberrypi-python
sudo nano smart_light.py
```

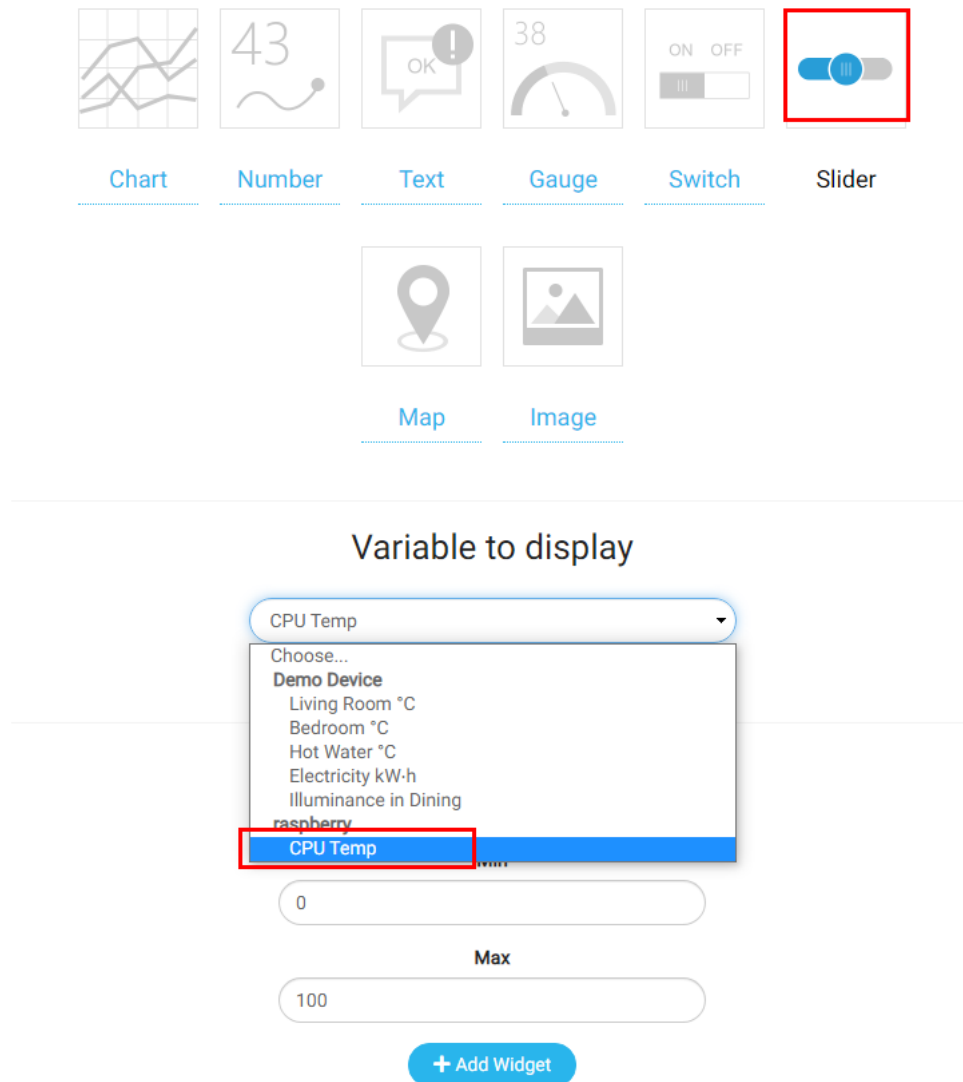
找到下面的部分并填写正确的设备令牌。

```
DEVICE_TOKEN = '__YOUR_DEVICE_TOKEN__'
```

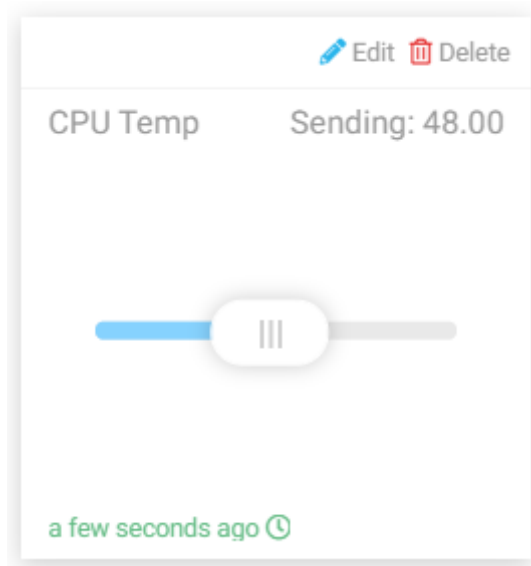
运行代码。

```
sudo python3 smart_light.py
```

转到 Cloud4RPI 页面，添加一个名为 project3 的新控制面板，然后单击 **Add Widget** 添加一个 **Slider** 小部件。



添加完后，就可以使用 **Slider** 小部件来控制 LED 的亮度了。



### 代码说明

```
led_val = GPIO.PWM(LED_PIN, 2000)
led_val.start(0)
```

调用 `GPIO.PWM()` 函数定义 `LED_PIN` 为 PWM 引脚并设置 PWM 引脚的频率为 2000Hz, 然后使用 `start()` 函数将初始占空比设置为 0。

```
led_val.ChangeDutyCycle(value)
```

Slider 小部件发送到树莓派的值用作 PWM 占空比来改变 LED 的亮度。

```
variables = {
  'CPU Temp': {
    'type': 'numeric',
    'value': 0,
    'bind': led_control
  },
}
```

通过设置 'value' 键的值, 我们可以设置 Slider 部件的初始值, 然后循环读取 'value' 键的值, 并将当前电平 `LED_PIN` (函数 `led_control()` 的返回值) 发送到 Cloud4RPi。

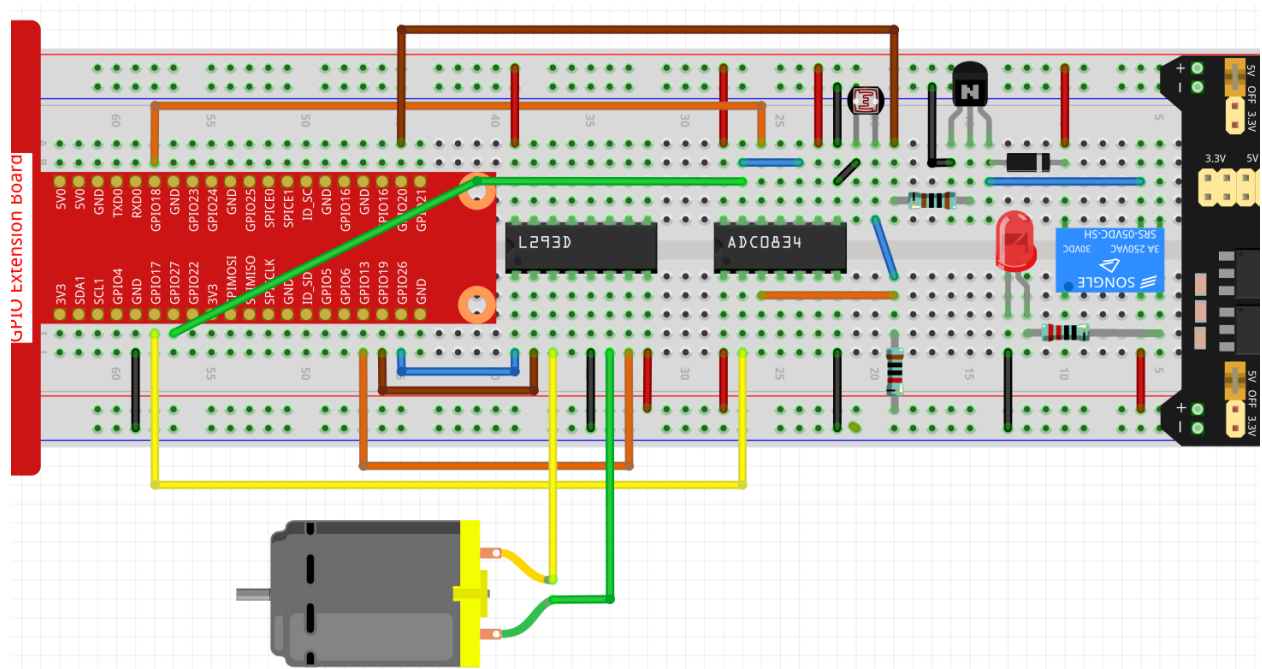
### 智能窗帘

在这个项目中, 我们将模拟一个根据房间亮度自动打开或关闭窗帘的场景。当光敏电阻检测到房间亮度过亮时, 树莓派会驱动电机关闭窗帘; 当房间亮度太暗时, 窗帘会被拉开。

您也可以通过 Cloud4RPi 上的文本小部件查看室内亮度变化, 或使用开关小部件控制灯 (为了用电安全, 本实验使用 LED 代替)。

## 实验步骤

搭建电路。



打开代码。

```
cd /home/pi/cloud4rpi-raspberrypi-python
sudo nano smart_curtain.py
```

找到下面部分并填写正确的设备令牌。

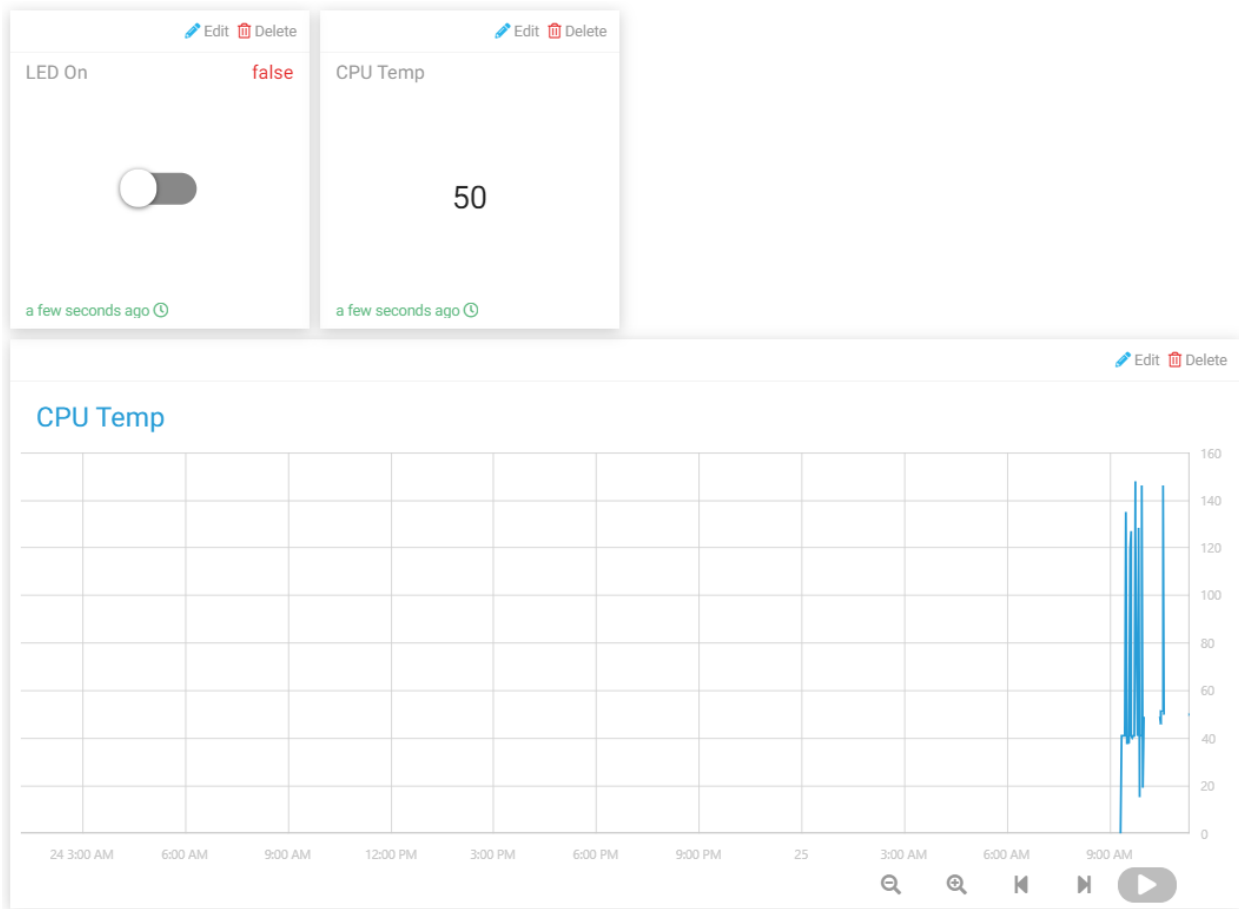
```
DEVICE_TOKEN = '__YOUR_DEVICE_TOKEN__'
```

运行代码。

```
sudo python3 smart_curtain.py
```

转到 Cloud4RPI 页面，添加一个新的控制面板命名为 **project4**，并通过点击 **Add Widget** 添加 3 只小部件 (**Switch**, **Text** 和 **Chart**)。





添加后，您可以使用开关小部件来控制 LED；Text 小部件上的光敏电阻值每 1 分钟更新一次，当值 > 100 时，电机顺时针旋转 5 秒；如果该值小于 20，则电机逆时针旋转 5 秒。您可以在 Chart 部件中看到随着时间推移发生的变化。

### 代码说明

```
def getValue():
    analogVal = ADC0834.getResult()
    if analogVal > 100:
        move.clockwise()
        sleep(5)
        move.stop()
    elif analogVal < 20:
        move.contrarotate()
        sleep(5)
        move.stop()
    return analogVal
```

读取 ADC0834 的 CH0 通道（连接光敏电阻）的值。如果该值 > 100，则电机顺时针旋转 5 秒。如果该值小于 20，则电机逆时针旋转 5 秒。

```
'CPU Temp': {
    'type': 'numeric',
```

(下页继续)

(续上页)

```
'bind': getValue
},
```

将 `getValue()` 函数放在 'bind' 键的值中，使 led 的亮度 (`getValue()` 函数的返回值) 可以被发送到 **Text** 部件。

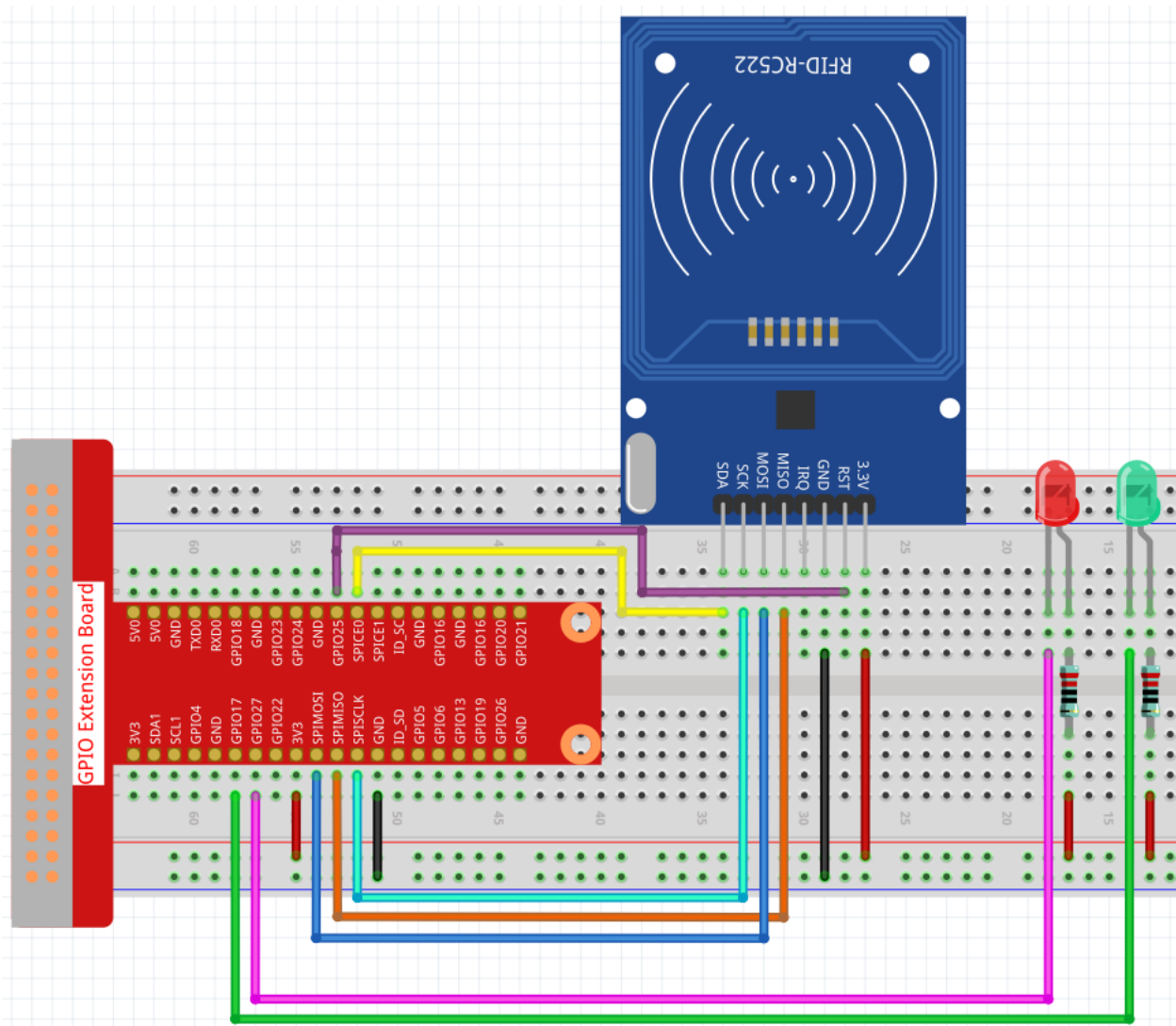
## 考勤系统

在这个项目中，我们使用 MFRC522 RFID Module 和 Cloud4RPI 做一个系统，可以在 Cloud4RPI 上查看考勤信息，或者打开本地.csv 文件查看考勤记录。

由于网络数据传输延迟，本项目只允许绿灯亮时打卡，红灯亮时上传数据。

## 实验步骤

搭建电路。



打开代码。

```
cd /home/pi/cloud4rpi-raspberrypi-python  
sudo nano attendance_system.py
```

找到下面的部分并填写正确的设备令牌。

```
DEVICE_TOKEN = '__YOUR_DEVICE_TOKEN__'
```

运行代码。

```
sudo python3 attendance_system.py
```

转到 Cloud4RPI 页面, 添加一个名为 **project5** 的新控制面板, 并通过点击 **Add widget** 按钮添加 2 个小部件 (**Text** 和 **Chart** 部件)。



添加完后, 您可以在 **Chart** 部件上查看不同时间段的打卡记录 (当 LED ON 的值为 True 时, 表示有人打卡), 然后在 **Text** 部件上查看打卡人数。

你也可以在 /home/pi/cloud4rpi-raspberrypi-python 路径下找到带有日期的 .csv 文件。

```

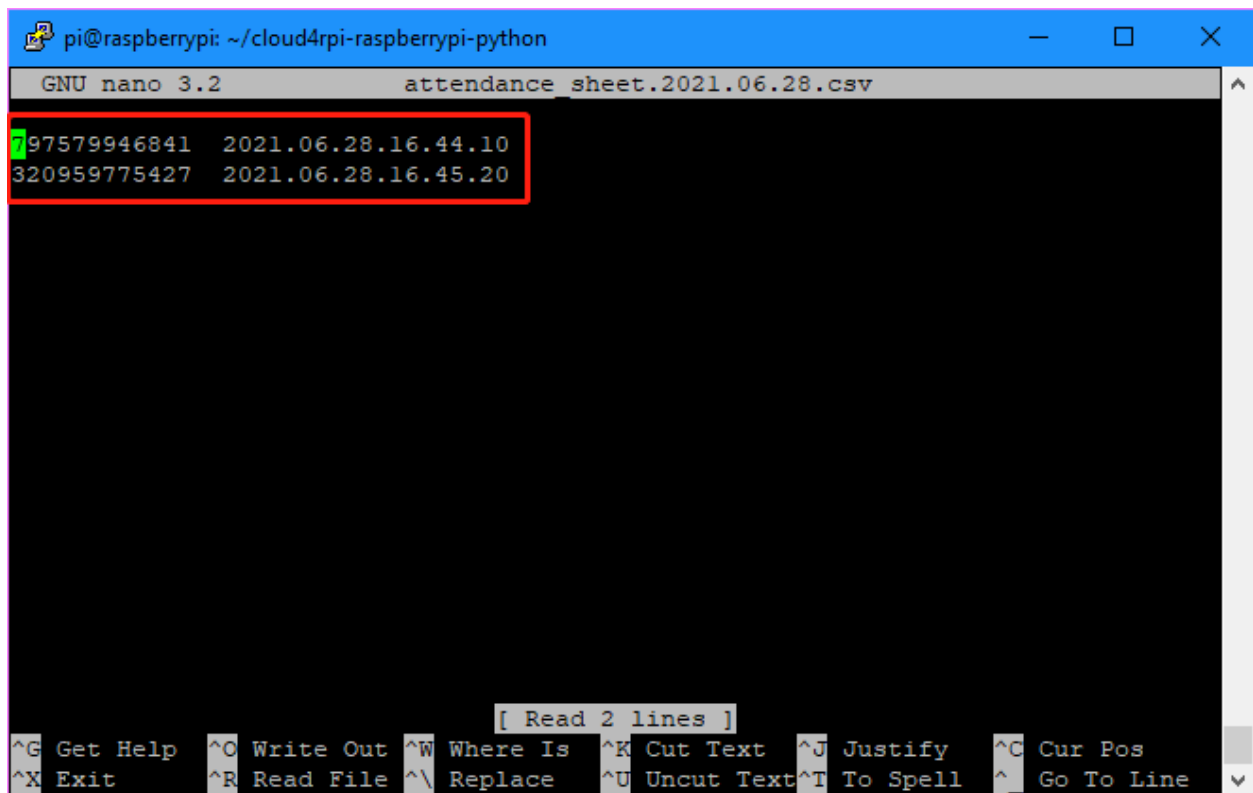
pi@raspberrypi:~/cloud4rpi-raspberrypi-python $ ls
ADC0834.py          remote_switch.py
attendance_sheet.2021.06.28.csv requirements.txt
attendance_system.py rpi.py
control.py          service
DHT11.py           service_install.sh
dsl8b20.py          setup.cfg
LICENSE             smart_light.py
Makefile            smart_window.py
MOTOR.py            temperature_and_humidity_acquisition.py
README.md           test

```

使用以下命令打开文件。

```
sudo nano attendance_sheet.2021.06.28.csv
```

这样就可以读取不同时间段的打卡记录。左边是不同 MFRC522 RFID 模块的 ID，右边是打卡时间。



```

GNU nano 3.2 attendance_sheet.2021.06.28.csv
797579946841 2021.06.28.16.44.10
320959775427 2021.06.28.16.45.20

```

[ Read 2 lines ]

^G Get Help    ^O Write Out    ^W Where Is    ^K Cut Text    ^J Justify    ^C Cur Pos  
 ^X Exit        ^R Read File    ^\ Replace    ^U Uncut Text ^T To Spell    ^\_ Go To Line

**注解：**如果事先将人名写入 MFRC522 RFID Module，然后将 id 替换为 code 中的文字（后面在 code 解释中会提到），这样考勤表就记录了每个人的名字，而不是卡的 ID。

## 代码说明

```
RedPin = 13
GreenPin = 11

GPIO.setmode(GPIO.BOARD)
GPIO.setup(RedPin, GPIO.OUT, initial=GPIO.HIGH)
GPIO.setup(GreenPin, GPIO.OUT, initial=GPIO.HIGH)
```

设置 2 个 LED 作为输出并将初始电平设置为高电平。

```
def get_time():
    time.time()
    year = str(time.strftime('%Y',time.localtime(time.time())))
    month = str(time.strftime('%m',time.localtime(time.time())))
    day = str(time.strftime('%d',time.localtime(time.time())))
    hour = str(time.strftime('%H',time.localtime(time.time())))
    minute = str(time.strftime('%M',time.localtime(time.time())))
    second = str(time.strftime('%S',time.localtime(time.time())))
    present_time = year + '.' + month + '.' + day + '.' + hour + '.' + minute + \
    + '.' + second
    present_date = year + '.' + month + '.' + day
    return present_date, present_time
```

使用 get\_time() 函数获取当前时间戳并返回两个值，一个是 present\_date 精确到天，一个是 present\_time 精确到秒。

```
attendance = False
GPIO.output(GreenPin, GPIO.LOW)
GPIO.output(RedPin, GPIO.HIGH)
print("Reading...Please place the card...")
id, text = reader.read()
print("ID: %s\nText: %s" % (id,text))
GPIO.output(RedPin, GPIO.LOW)
GPIO.output(GreenPin, GPIO.HIGH)
```

首先，我们设置 attendance 为 False，表示没有人打卡。

然后将 GreenPin 置低电平点亮绿灯，RedPin 置高电平保持红灯熄灭，表示当前考勤系统工作正常。

当有人打卡时，会打印卡片的 id 和文本信息。如果红灯亮绿灯灭，表示签到成功，结果会发送到 Cloud4Rpi。

在此期间，考勤系统处于休眠状态，直到绿灯亮进入下一个周期。

```
if not id in attendance_list:
    attendance = True
    attendance_list.append(id)
    present_date, present_time = get_time()
    attendance_statistics[id] = present_time
    with open('attendance_sheet.' + present_date + '.csv', 'w') as f:
        [f.write('{0} {1}\n'.format(key, value)) for key, value in \
    attendance_statistics.items()]
```

首先通过 if 语句判断 attendance\_list 中的 id 是否重复，如果没有，则表示打卡有效，将信息传入 attendance\_list。然后我们得到当前时间戳，并使用 id 作为 attendance\_statistics 字典的键和其对应的值 present\_time。这样字典 attendance\_statistics 就存储了当前 id 的打卡时间。

最后我们将字典 attendance\_statistics 写入一个.csv 文件，并将文件命名为 'attendance\_sheet.' + present\_date + '.csv'，以便我们按时间顺序存储考勤表。

**注解：** 如果之前在 MFRC522 RFID 中写入了一个人的名字，则替换 id 和 text 内容，那么考勤表将会记录下这个人的名字。

```
def get_num():
    return len(attendance_list)
```

返回 attendance\_list 的长度, 即打卡的人数

## 6.6 GPIO 编程

RasPad 3 可以将树莓派的 40 个 I/O 引脚连接到面包板，因此您可以使用这些引脚来做一些有趣的项目。

**注解：** 请参阅 [GPIO 扩展板](#) 说明，将 RasPad 内部树莓派的 40 个引脚扩展到外部面包板。

SunFounder 有非常完整的关于树莓派 GPIO 学习包的课程体系，比如下面两个学习包。

- *Electronic Kit for Raspberry Pi and Arduino*

此基础学习套件适用于树莓派和 Arduino 新手。该套件教授基本电路设计并为初学者介绍 Python 编程。

- *Da Vinci Kit for Raspberry Pi*

这是一个先进的树莓派教学套件，包含一系列强大的组件。该套件为初学者和高级用户教授物理计算和 Python 编程原理。

### 6.6.1 Electronic Kit for Raspberry Pi and Arduino

#### 教程链接

<https://docs.sunfounder.com/projects/electronic-kit/en/latest/index.html>

产品链接全球: <https://www.sunfounder.com/products/electronics-fun-kit>

美国: <https://www.amazon.com/dp/product/B07QT78FXF>

德国: <https://www.amazon.de/dp/B07QT78FXF>

加拿大: <https://www.amazon.ca/dp/product/B07RQ7Z6G4>

### 6.6.2 Da Vinci Kit for Raspberry Pi

#### 教程链接

<https://docs.sunfounder.com/projects/davinci-kit-for-raspberry-pi/en/latest/>

#### 产品链接

全球: <https://www.sunfounder.com/products/sunfounder-davinci-kit>

美国: <https://www.amazon.com/dp/product/B07WV2HYC6>

日本: <https://www.amazon.co.jp/dp/product/B083SJQXCL>

加拿大: <https://www.amazon.de/dp/B07WYX8M76>

德国: <https://www.amazon.ca/dp/product/B07KH2C215>

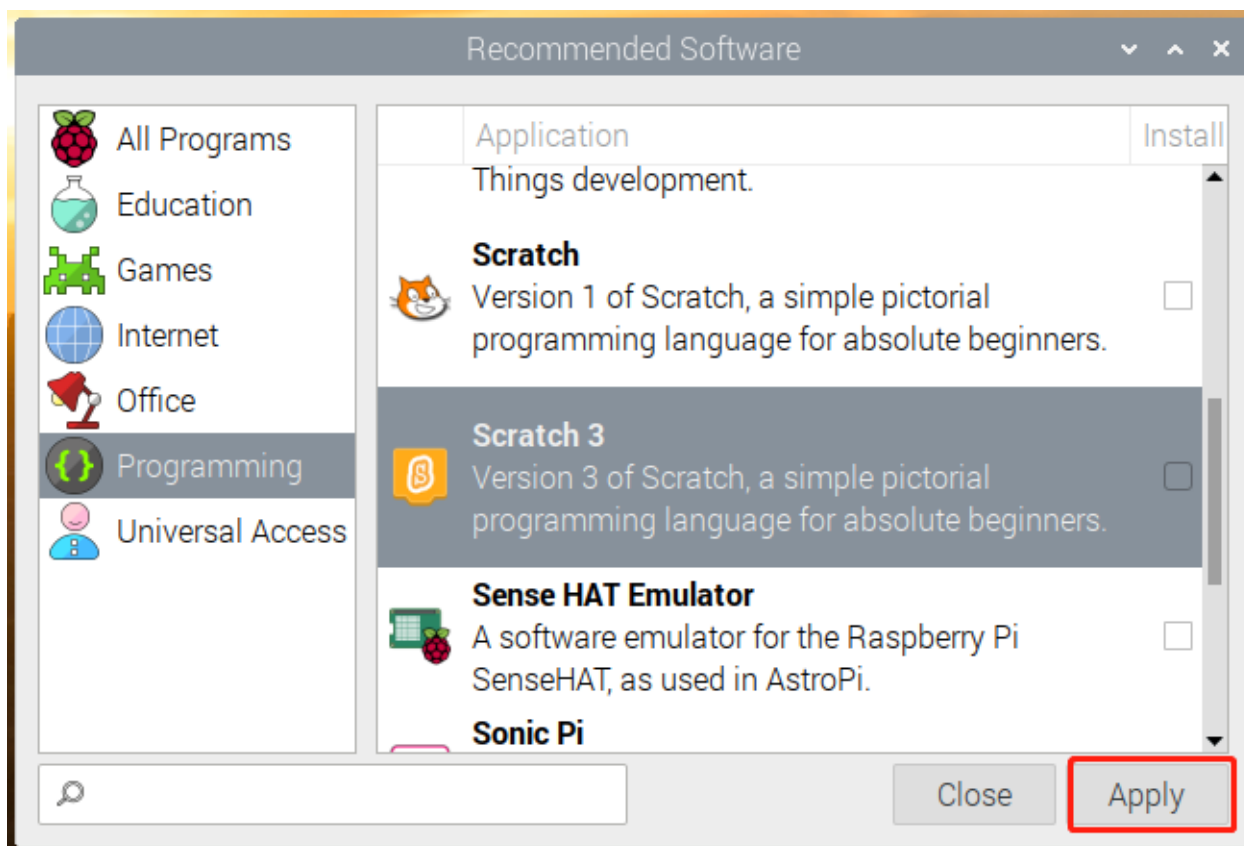
## 6.7 Scratch 项目

在本章中，您将学习在 RasPad 3 上使用 Scratch，其中包含 10 个示例。

如果您是刚使用 Scratch 的用户，我们建议您按顺序尝试教学示例，以便您可以快速上手 Scratch。

如果你下载的树莓派系统自带推荐软件，那么可以在 **Programming** 中找到 Scratch 3。

如果您下载的系统只有桌面，那么可以点击 **Preferences -> Recommended Software -> Programming** 来在 RasPad3 上安装 Scratch。



**注解：**在尝试教学示例之前，确保已经下载了相关资料和代码文件。

打开终端并输入以下命令从 github 进行下载。

```
git clone https://github.com/sunfounder/raspad3.git
```

### Projects

### 6.7.1 会动的名字

#### 描述

今天我们将学习在 Scratch 上写我们的名字。当我们点击名称中的字母时，它们将会动起来。

点击绿旗开始。

或者点击 [Animate A Name](#), 然后在 Scratch 官网学习在线教程。

#### 所需组件

- A RasPad 3
- Scratch 3 (either online or offline)

#### 你将学会

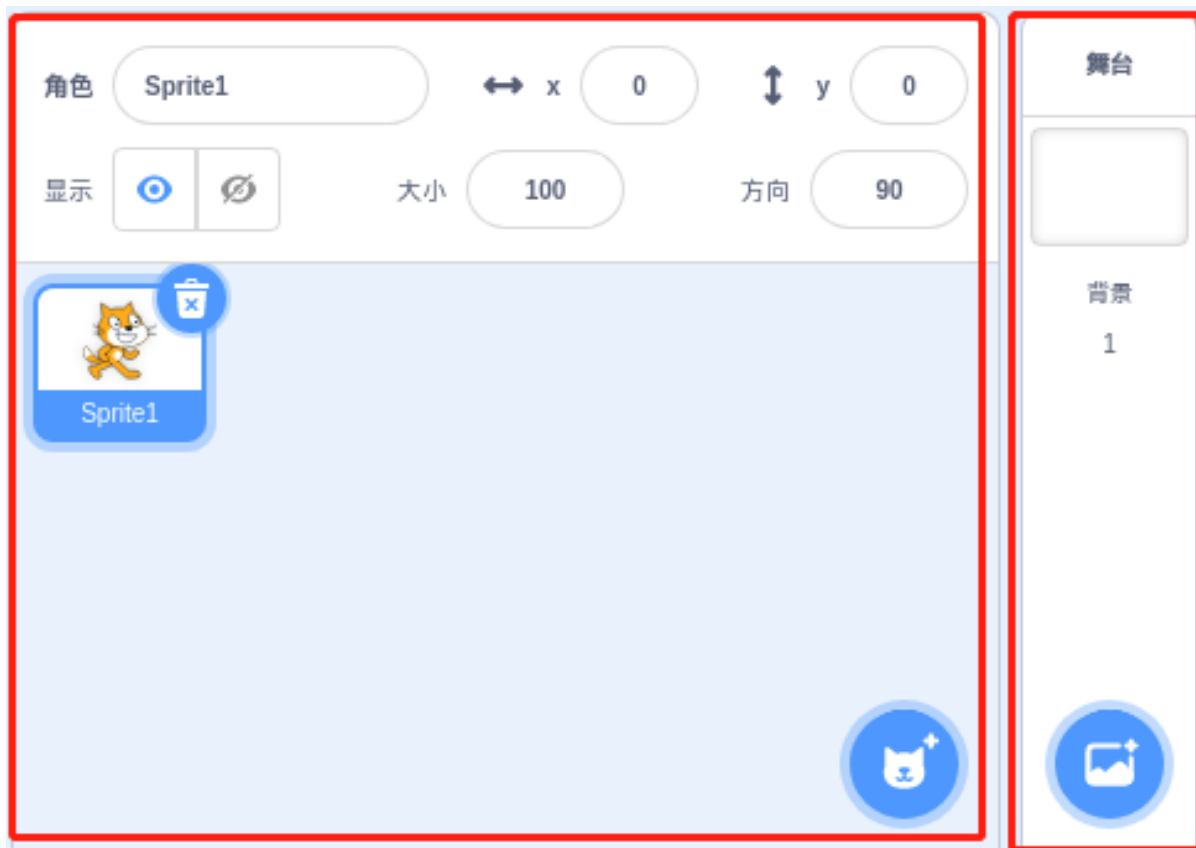
- 选择你想要的角色。
- 分别编程不同的角色。
- 使角色变色。
- 让角色旋转。
- 让角色缩放。
- 让角色闪烁。



## 课程指南

## 我的名字是 Abby

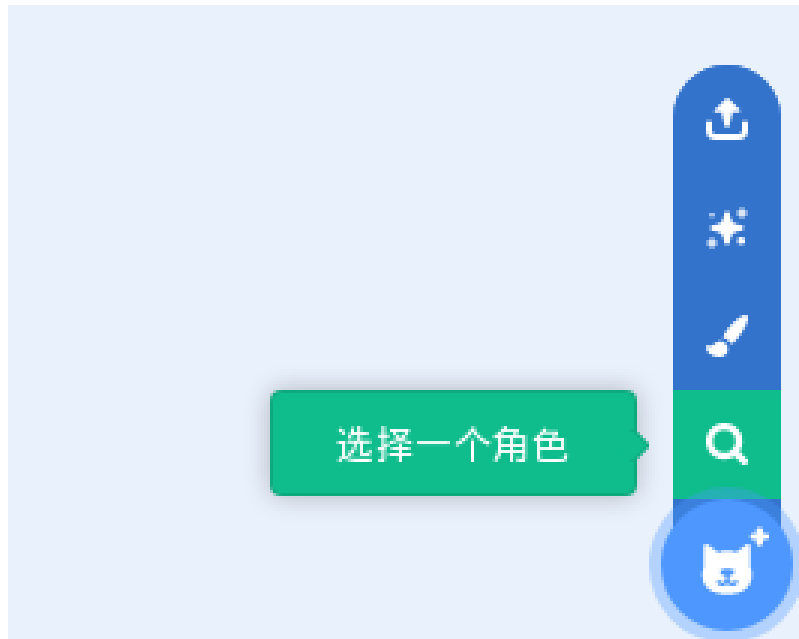
在 Scratch 的右下角，有两个区域。我们可以在这里添加我们想要的角色和背景。



首先删除原来的角色，我们不需要它。



然后选择新的角色。



找到字母角色。



重复几次拼出你的名字，就像我的名字“Abby”一样。

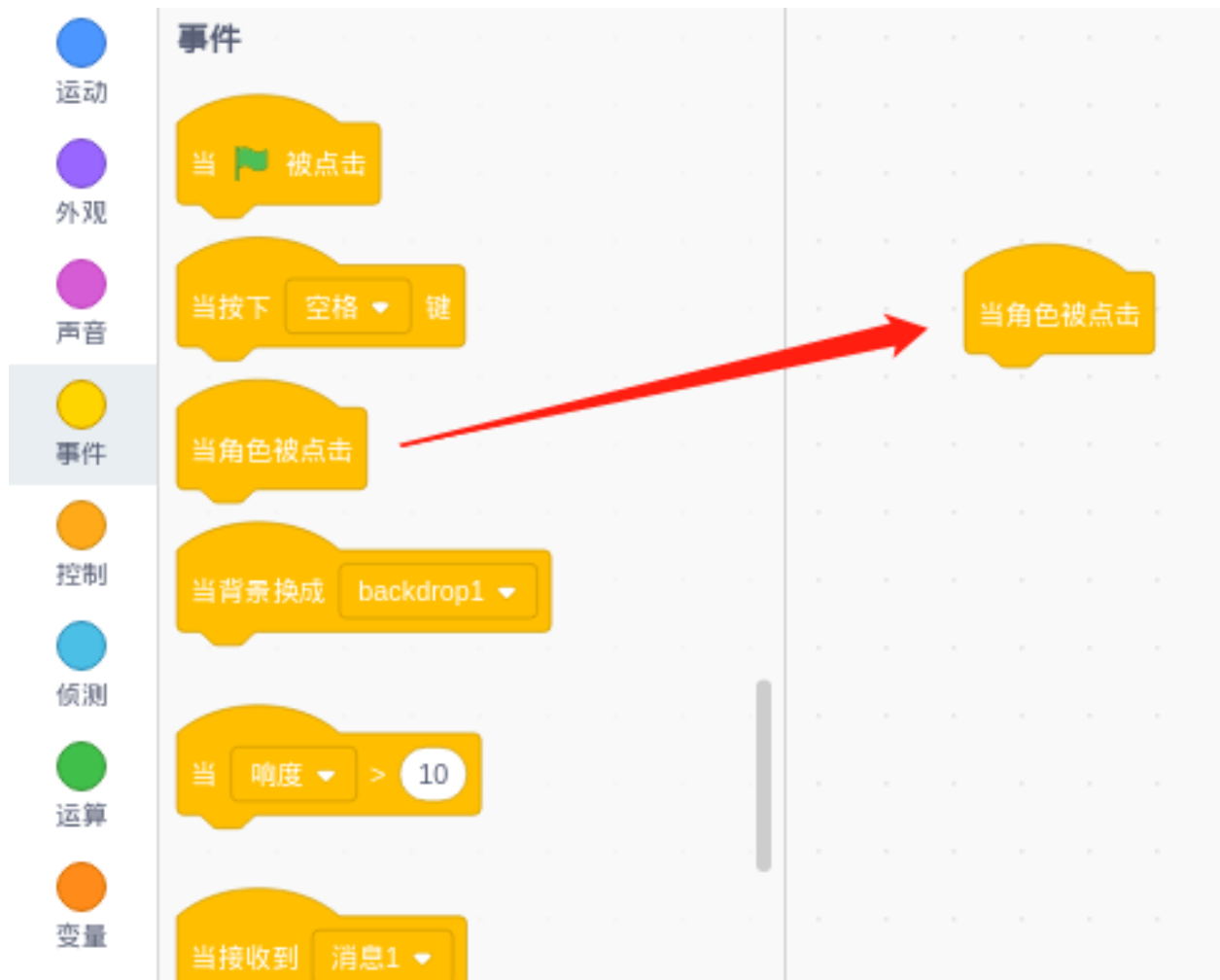


字母 A 可以改变自己的颜色

单击 Block-A 角色，让我们为它编写一些代码。

当点击 A 时，它会改变它的颜色。

点击角色，在事件中找到「当角色被点击」，拖到右边的编码区。



在 外观里找到「将颜色特效增加…」把它拖到「当角色被点击」的底部.



这时，我们点击舞台上的字母 **A**，它就会改变颜色。

### 字母 **B** 能够旋转 360 度

单击 Block-B 角色，让我们为 Block-B 编写代码。当舞台上的字母 **B** 被点击时，它会旋转一圈。



你知道为什么要设置让 B 每次旋转 15 度，共旋转 24 次吗？因为  $24 \times 15 = 360$ ，所以转了 360 度！

### 第二个字母 B 可以放大和缩小

单击 Block-B2 角色，让我们为 Block-B2 编写代码。

当舞台上的第二个 B 被点击时，它会先放大再缩小。



### 点击字母 Y 让它闪烁

单击 Block-Y 角色，让我们为 Block-Y 编写一段代码。当舞台上的 Y 被点击时，它会反复消失和出现。



## 挑战

我相信你很快就能掌握和实现这个游戏。接下来，我们将添加一些内容使你的游戏更丰富。

- 添加两个新角色，其中一个会在点击时闪烁并改变颜色；点击另一个将缩放和旋转。

## 6.7.2 创作故事

### 描述

你喜欢听故事吗？比如，睡前听父母讲的小故事，晚上可以做一个甜蜜的梦。今天我们要“讲述”一只小猴子在 Scratch 上寻找香蕉的故事。

点击绿旗开始。

或者点击 [Create a Story](#), 然后在 Scratch 官网学习在线教程。

### 所需组件

- A RasPad 3
- Scratch 3 (either online or offline)

### 你会学到

- 选择你想要的角色。
- 让角色四处走动。
- 让角色互相交谈。
- 切换场景。

### 课程指南

**这是一只快乐的猴子，名叫 Harley。**

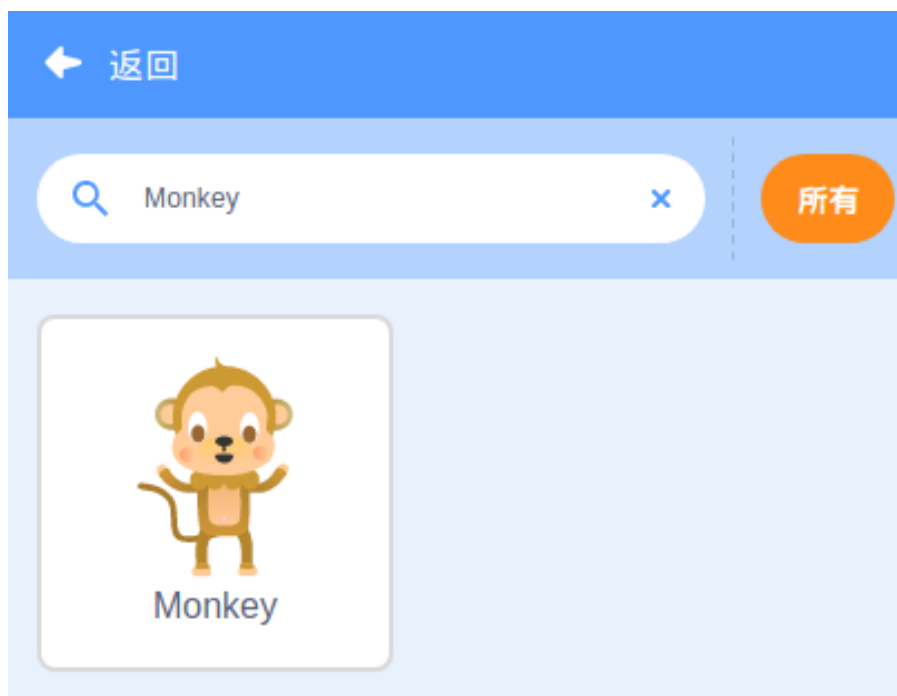
在 Scratch 的右下角，有两个区域——角色区和舞台区，在这里我们可以添加我们想要的角色和舞台背景。





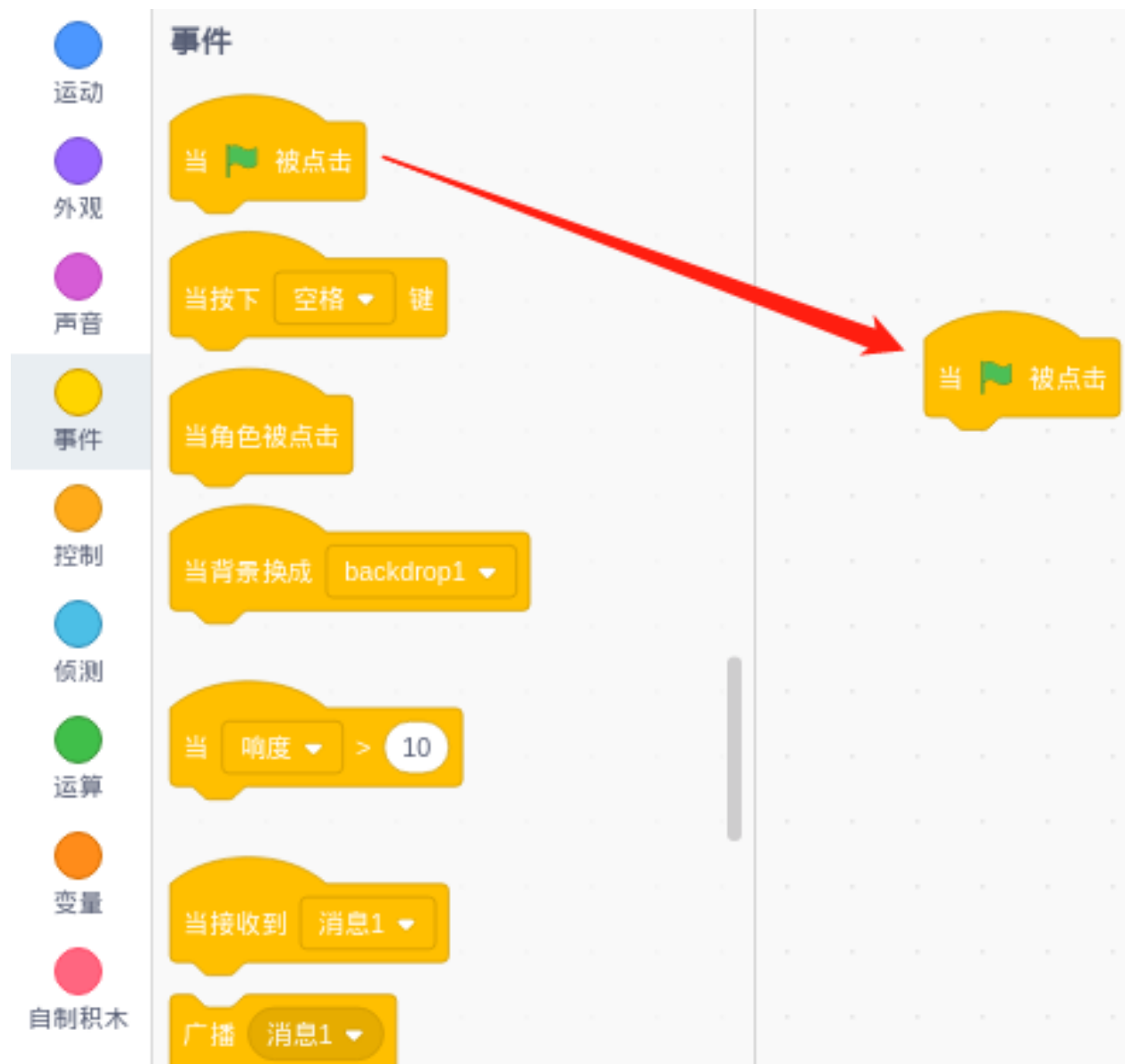
首先是当前角色，然后点击 **选择一个角色** 按钮来添加角色。

我们需要找到 Monkey，点击它！你可以用上面的搜索框进行查找。然后给它起一个名字 **Harley**。



Harley 饿了。他问他的朋友 Perry 香蕉在哪里？

单击您的 Harley 角色，然后单击事件类别。将 当 被点击 拖到右边的编码区。



点击外观类别并拖动 说”你好!”2 秒到 当 被点击的底部。

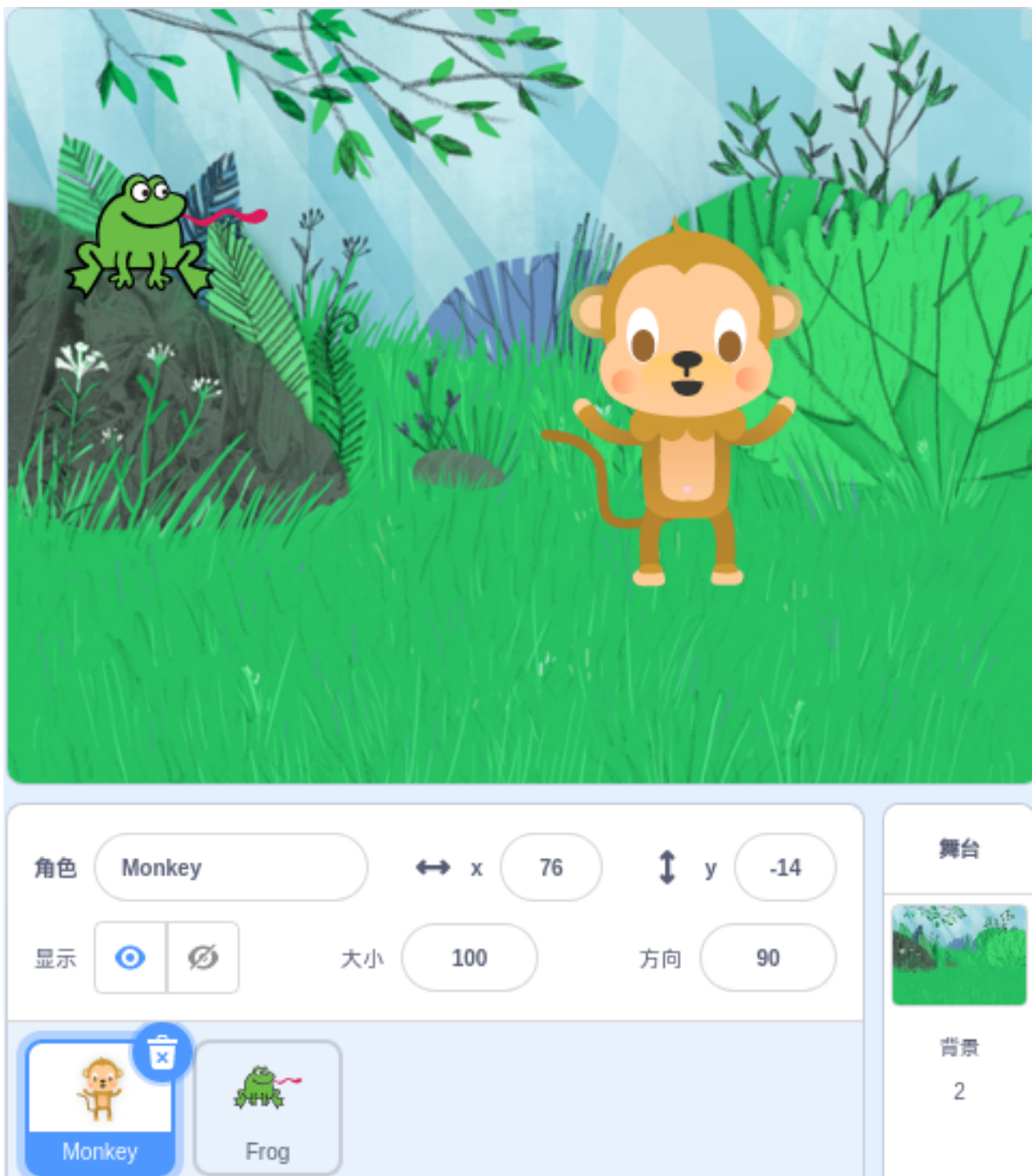


在输入框中更改它说话的内容以及它说话的时长。

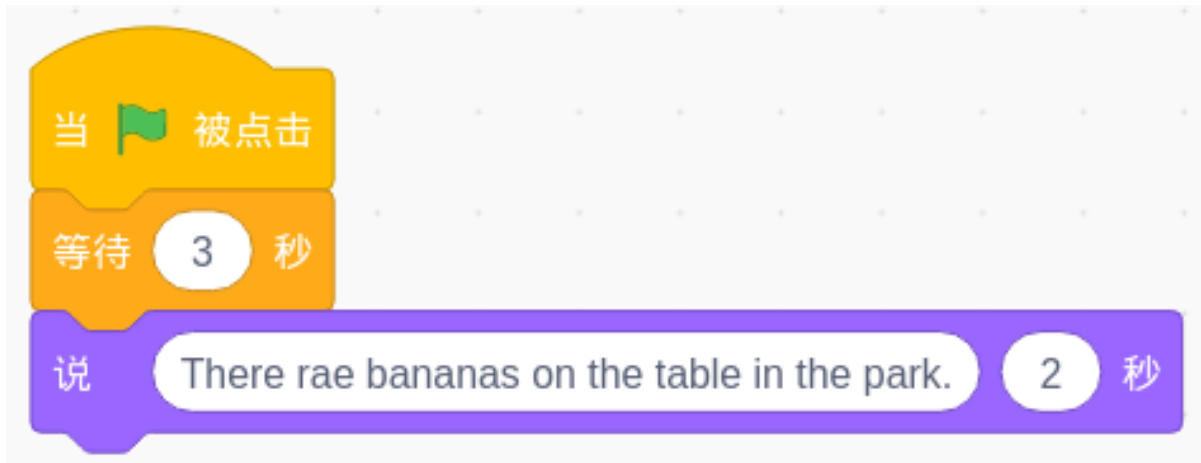


Perry 在公园里看到了香蕉，所以他告诉了 Harley.

Perry 是一只青蛙，添加青蛙角色和森林背景。



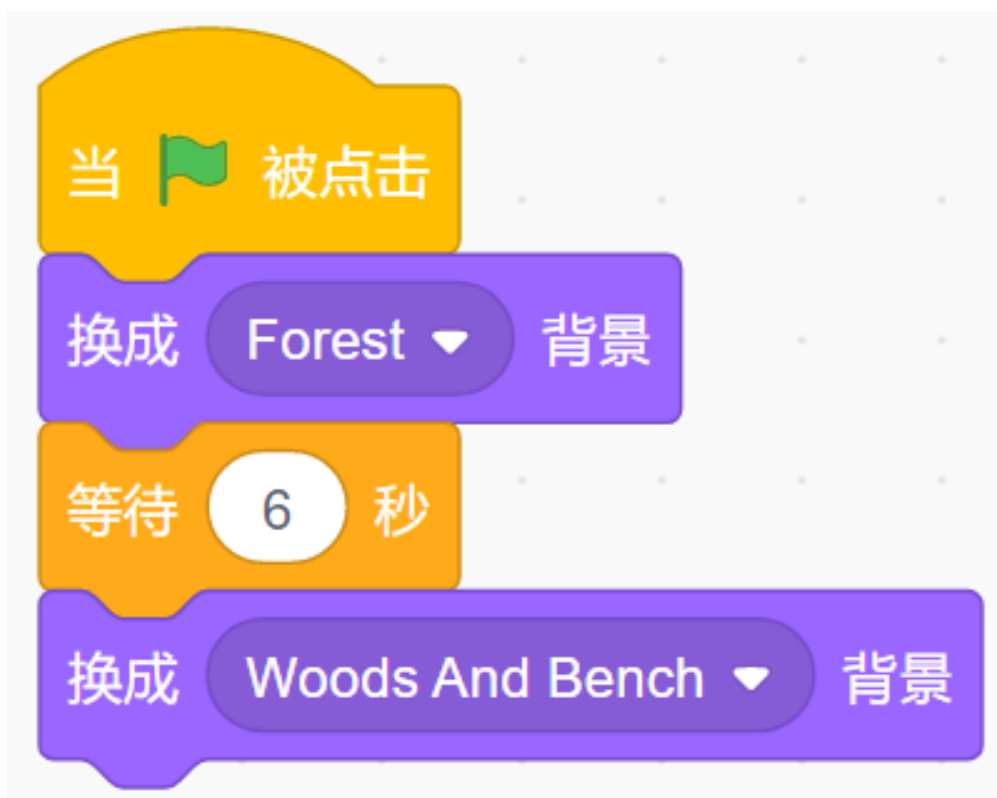
点击青蛙精灵，添加一个[等待 () 秒] 块（来等 Harley 把话说完），然后青蛙开始说话。



现在，点击绿旗，我们就可以看到 Harley 和 Perry 的聊天。

**Harley 去了公园，在桌子上发现了香蕉。**

现在，单击背景并让它在 6 秒后（对话结束后）切换。

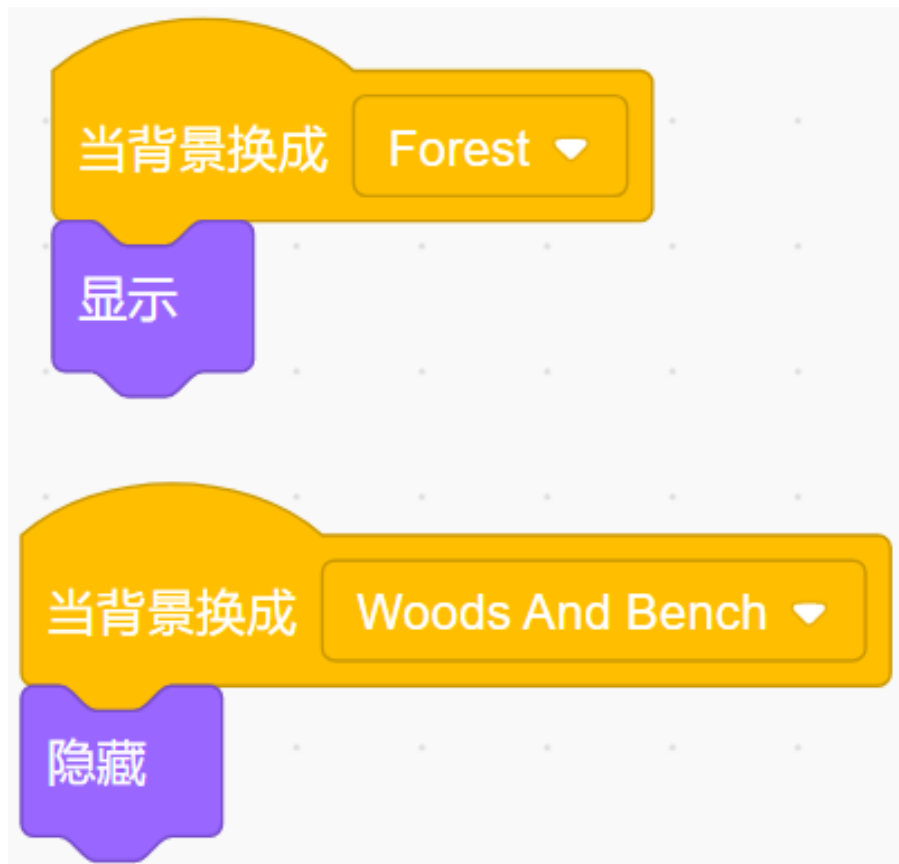


现在，通过点击绿旗，我们可以看到 Perry 和 Harley 在对话后一起来到公园。

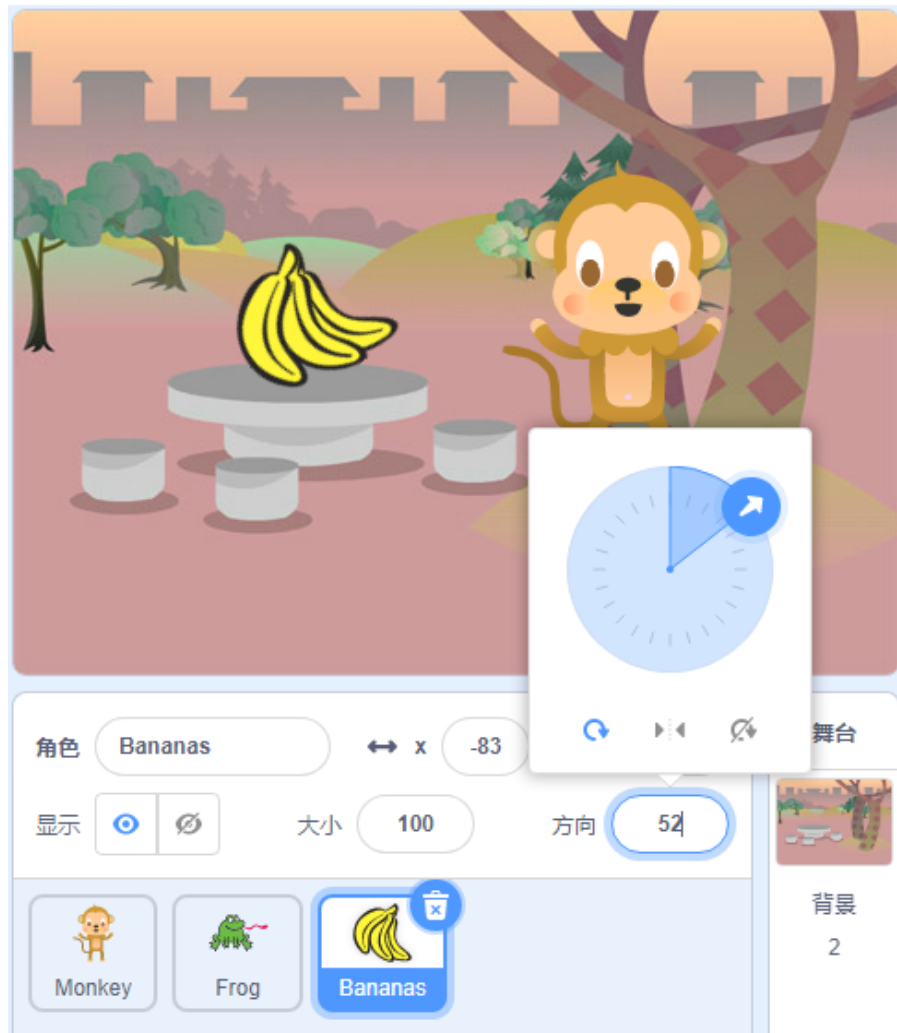


但是 Harley 应该一个人来公园，桌子上应该有香蕉。所以，我们必须让 Perry 消失。

点击青蛙角色，添加一个事件，当背景换成... 时，在外观中找到“显示”和“隐藏”。让 Perry 适时出现！

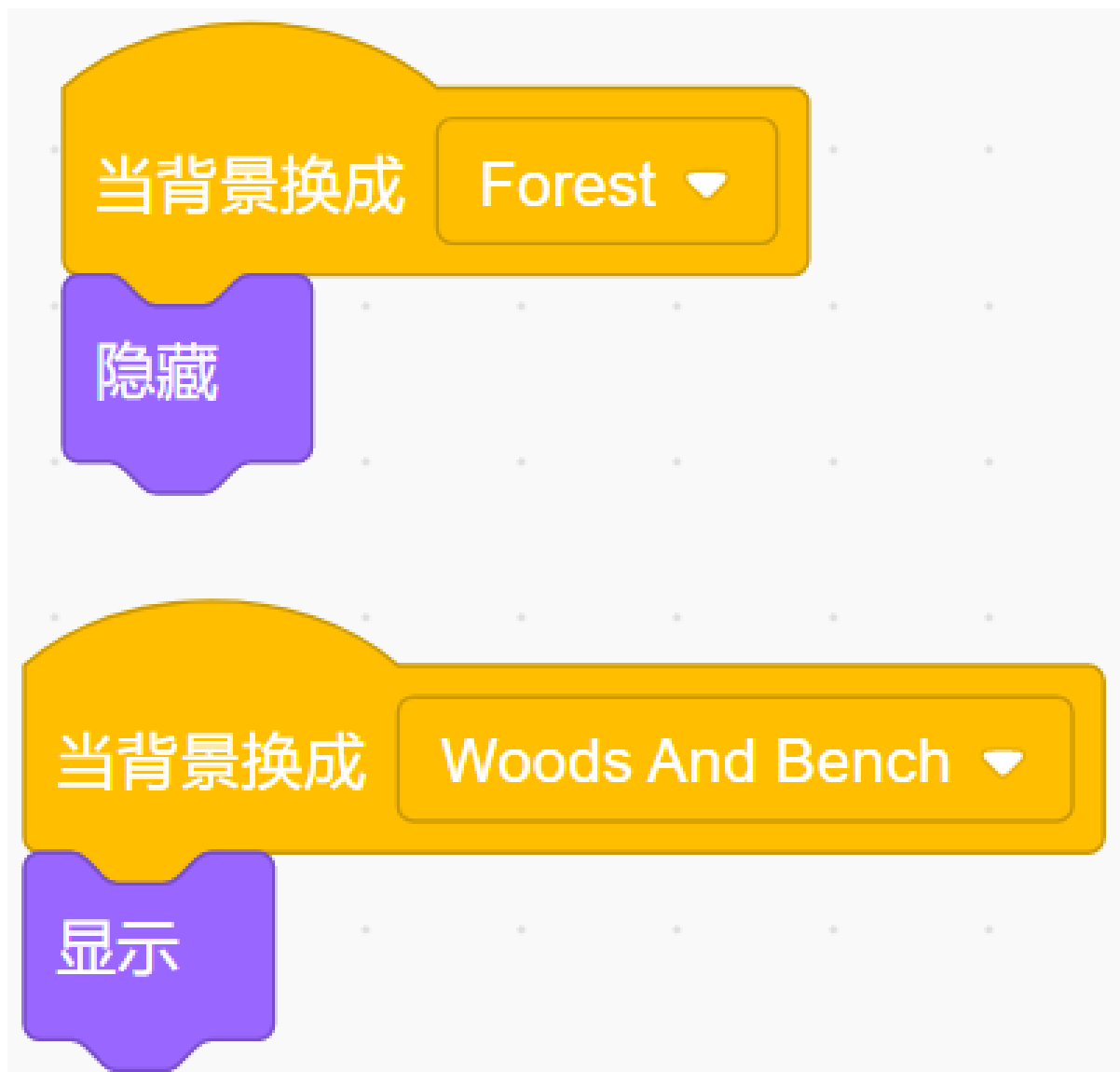


添加香蕉角色并调整角度。



也让它在正确的时间出现。





现在，点击绿旗，你可以看到一个完整的故事。

### 挑战

相信聪明的你很快就能编程实现这个游戏，接下来我们会增加一些挑战来丰富我们的故事内容。

- 让 Harley 将香蕉带回森林，与 Perry 分享。

### 6.7.3 制作音乐

#### 描述

你去过音乐厅吗？它充满了琳琅满目的乐器。这些乐器在音乐家的演奏下发出美妙的声音。

今天我们也将成为音乐家，在 Scratch 的舞台上添加一些乐器。当您点击不同的乐器时，它们会发出相应的乐器声音。

点击绿旗开始。

或者点击 [Make Music](#)，然后在 Scratch 官网学习在线教程。

#### 所需组件

- A RasPad 3
- Scratch 3 (在线或离线)

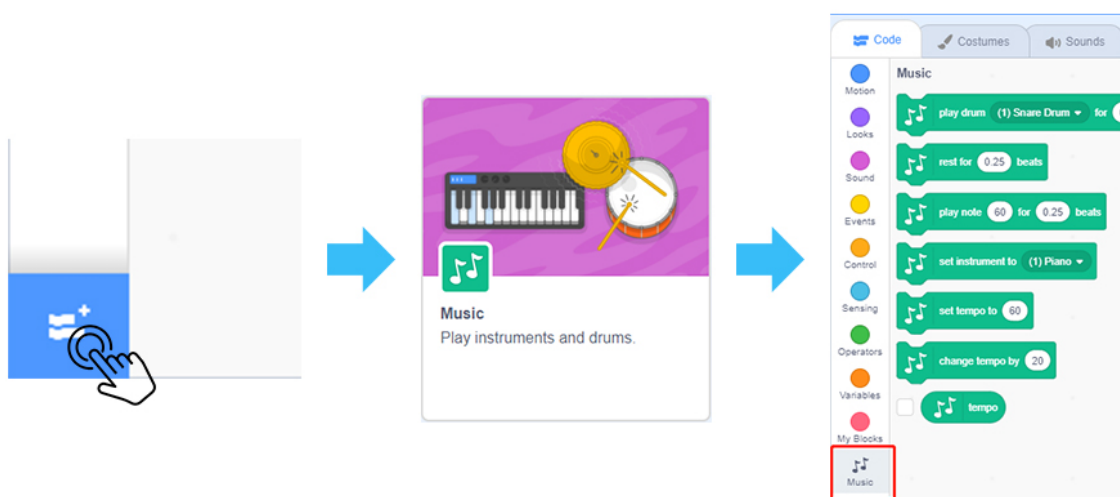
#### 你会学到

- 用 Scratch **添加扩展**功能添加音乐扩展。
- 修改角色的初始大小。
- 让角色做一些动作。

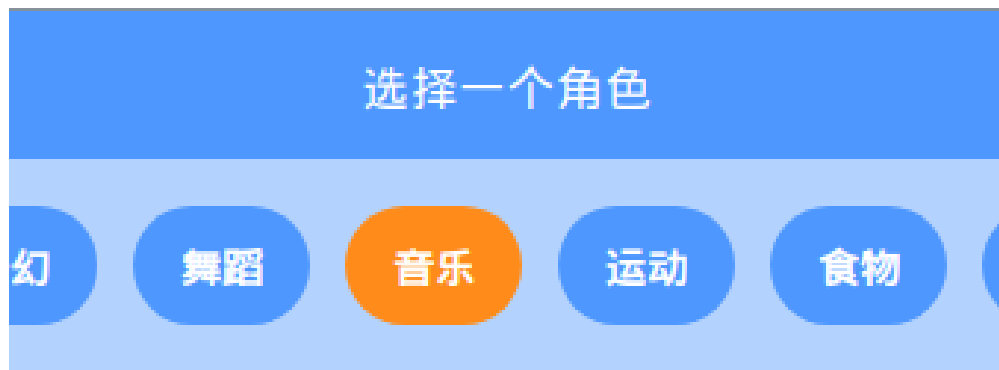
#### 课程指南

#### 我们需要鼓

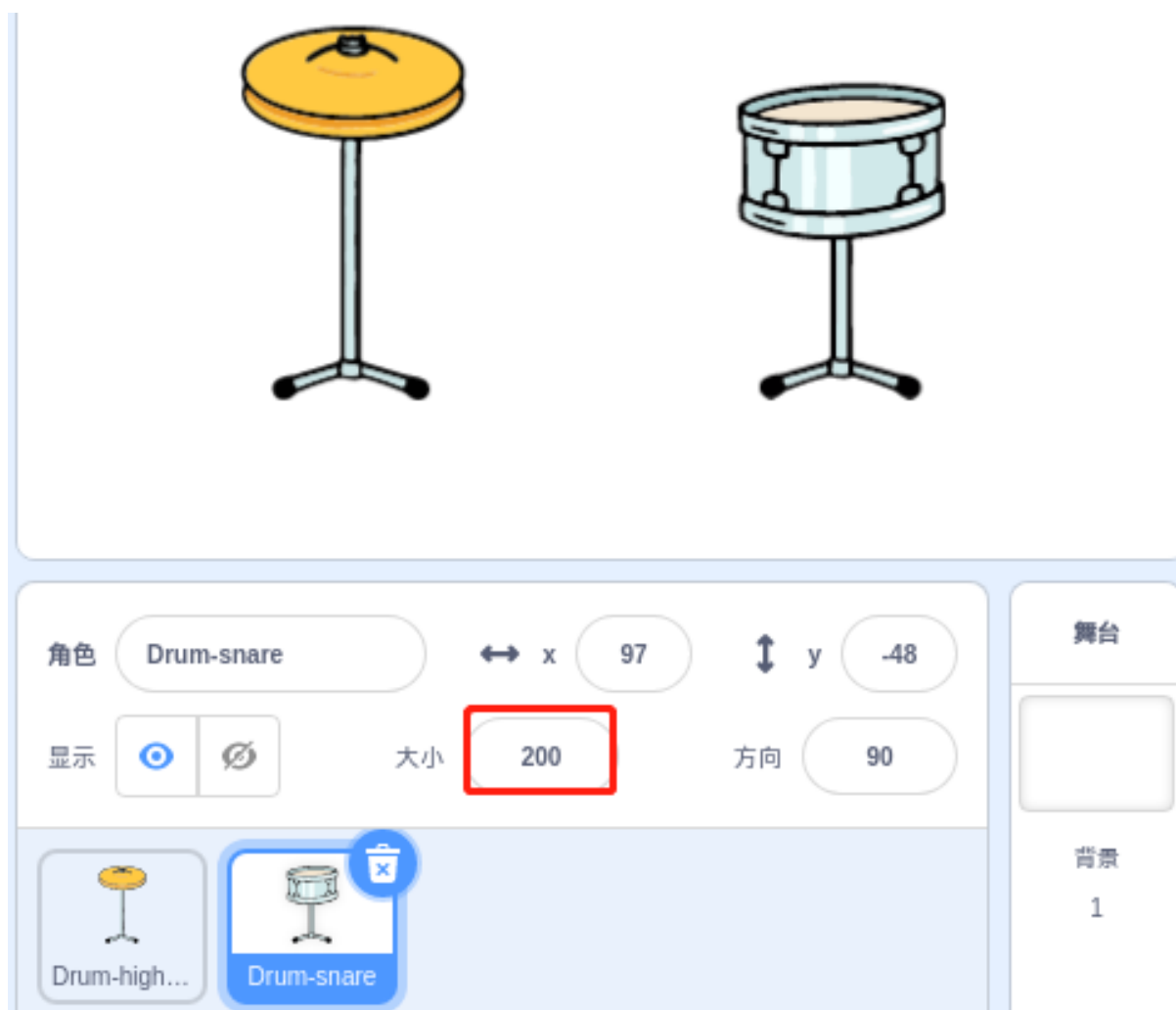
点击 Scratch 左下角的“添加扩展”图标，选择 **音乐**，然后你将会在 Scratch 的左侧会发现一个新的扩展——音乐。



删除原来的角色, 添加 Drum-highhat 角色和 Drum-snare 角色。

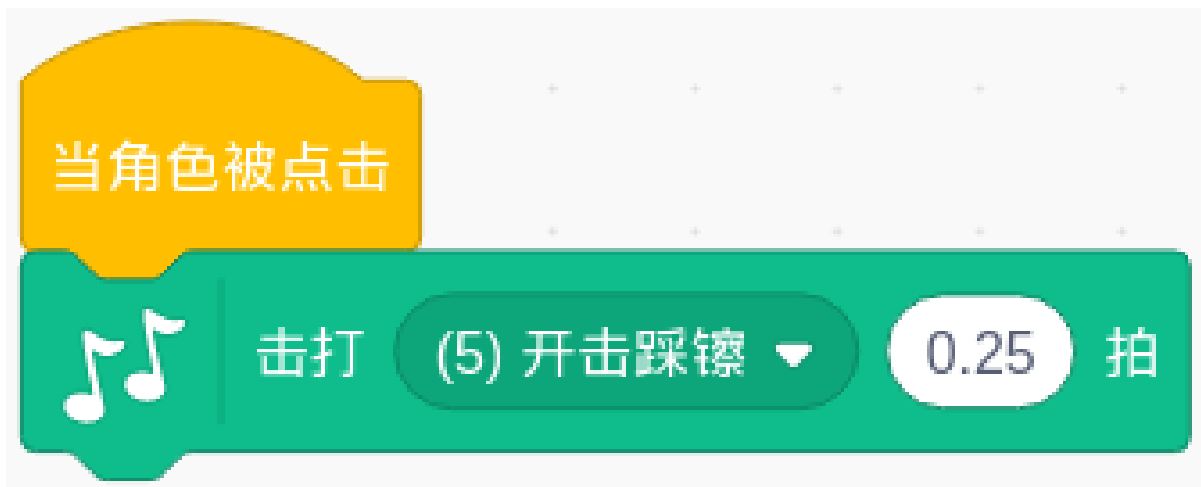


将 Drum-highhat 角色和 Drum-snare 角色调整到合适的大小。



播放 Drum-highhat。

点击 Drum-highhat 角色，在音乐扩展模块中拖出「击打...」代码块。



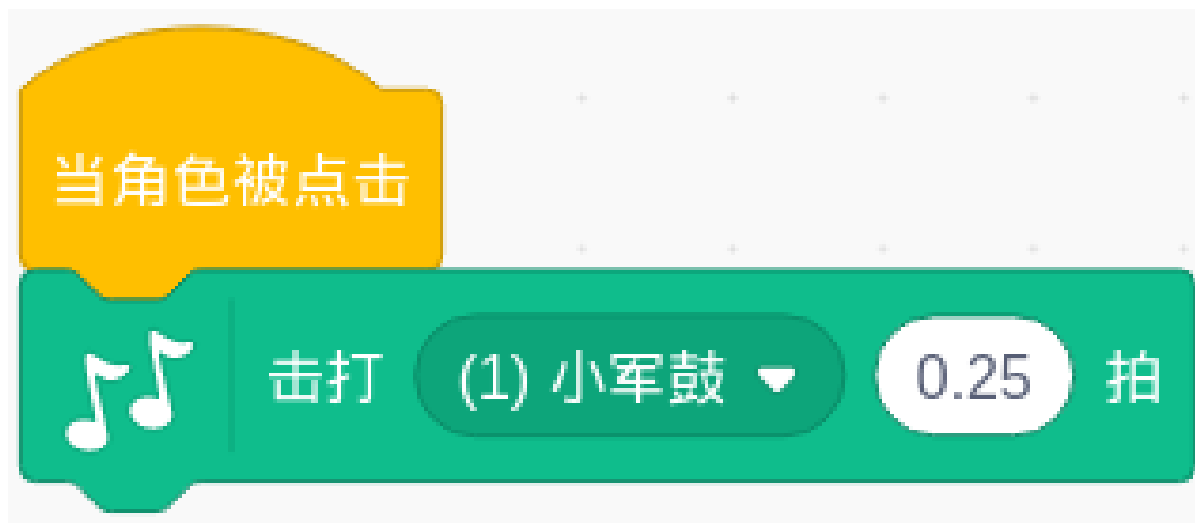
将选项更改为 (5) 开击踩镲，然后使用“下一个造型”块来切换鼓的外观。



然后你就可以演奏 Drum-highhat 了。

**演奏 Drum-snare.**

点击 Drum-snare 角色并在音乐模块中拖出「击打...」代码块。



将选项更改为 (1) 小军鼓, 然后使用 下一个造型块 来切换小军鼓的外观。



然后你就可以演奏小军鼓了。

### 挑战

相信你能很轻松的掌握并实现这个游戏。接下来我们将添加一些更难的内容来丰富我们的游戏。

- 添加 Drum 角色, Drum Kit 角色和 Drum-cymbal 角色, 修改它们的大小、并选择合适的音效。这样我们就制作了一套架子鼓。

角色	大小	乐器选项
Drum	200	(3) 敲鼓边
Drum Kit	150	(2) 低音鼓
Drum-cymbal	200	(4) 碎音钹

## 6.7.4 点击游戏

### 描述

今天我们来玩一个戳气球的游戏。

点击绿旗开始。

或者点击 [Clicker Game](#), 然后在 Scratch 官网学习在线教程。

### 所需组件

- A RasPad 3
- Scratch 3 (either online or offline)

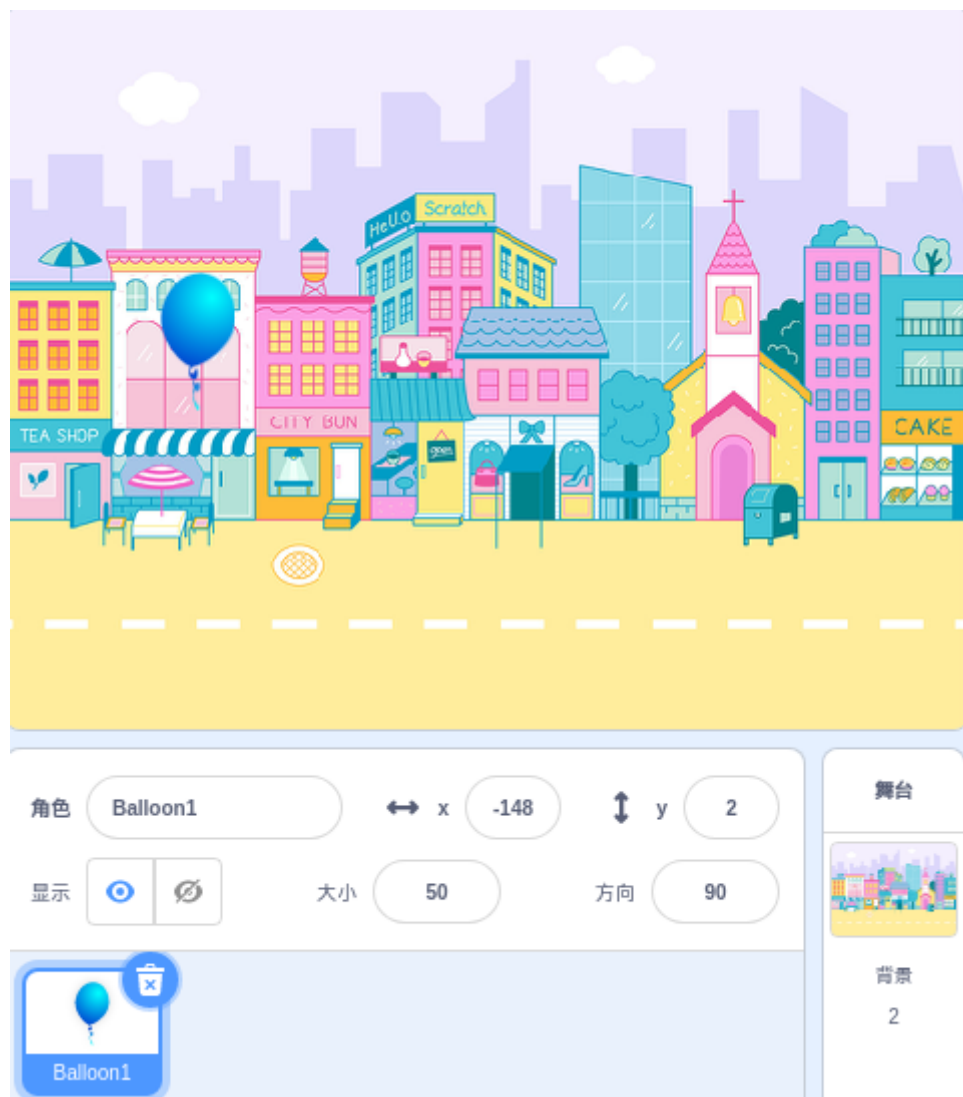
### 你会学到

- 选择随机数。
- 改变角色的位置、大小和颜色。
- 设置变量。

### 课程指南

让气球随机出现。

首先，添加 **Balloon1** 角色和多彩的城市背景。

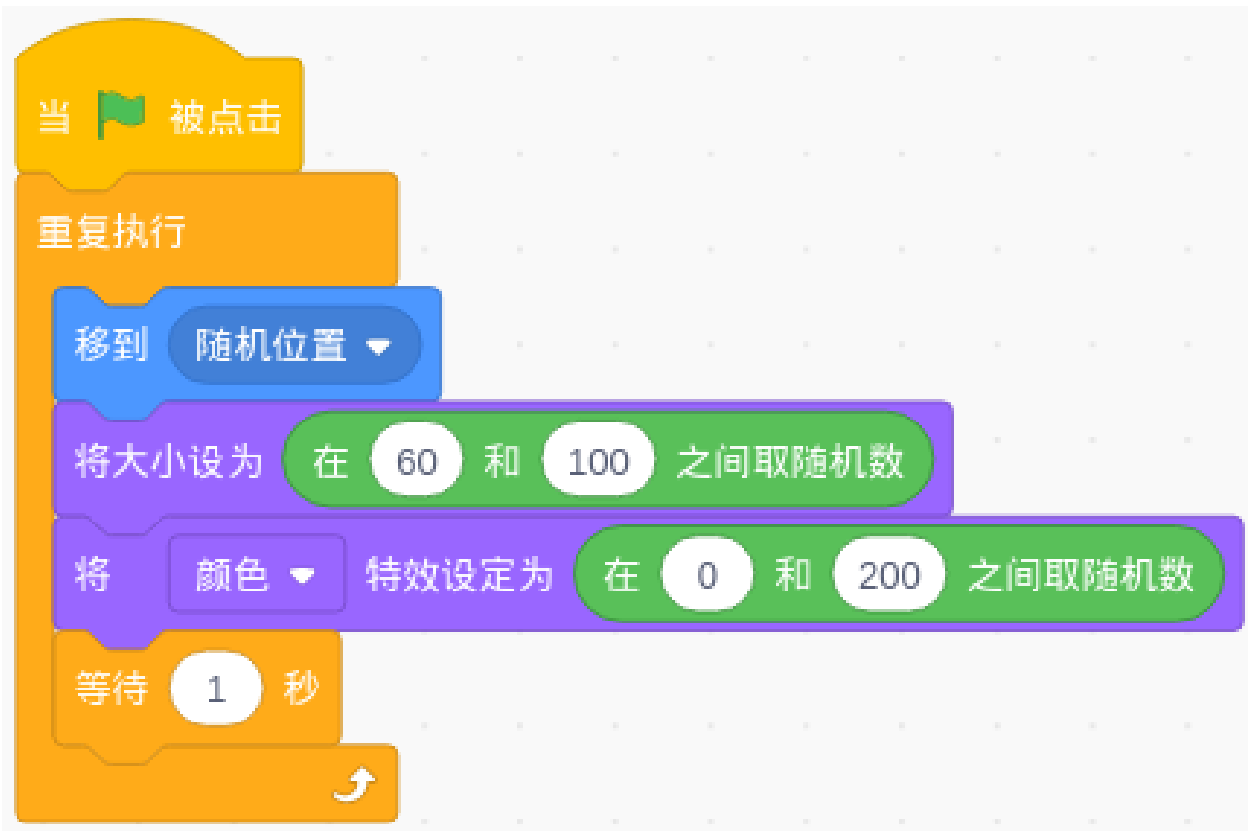


将气球随机放在一个位置。





随机改变它的大小和颜色。



现在，点击绿旗，您可以看到舞台区域中的气球开始随机出现变化。

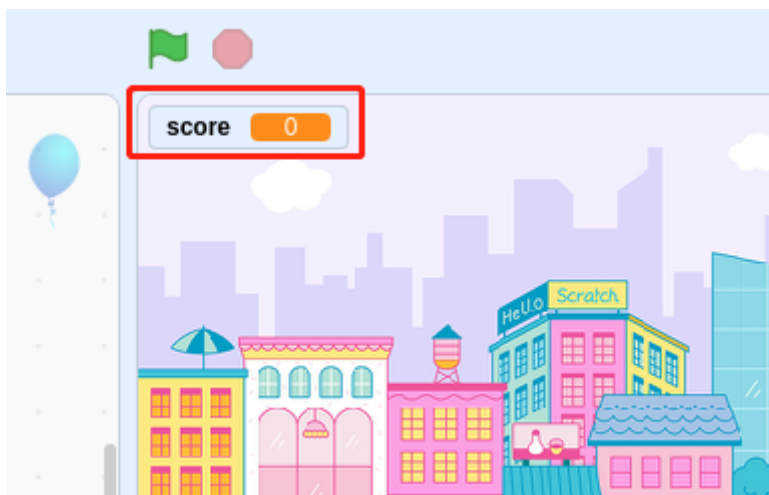
#### 单击气球以提高分数。

现在，我们需要建立一个评分机制。每当点击气球时，分数就会增加一。

创建一个变量（它可以帮助我们跟踪和存储分数）。在变量中，选择 **建立一个变量** 并命名为 **score**。



我们将在舞台的左上角看到它。



当气球被点击时，分数增加一。当您点击绿旗（游戏开始）时，分数将重置为零。



现在，点击舞台区域上方的绿色旗帜开始游戏，点击气球来增加你的分数！

### 挑战

我相信你很快就会足够聪明来编程和实现这个游戏。接下来，可以添加一些额外的内容来丰富你的游戏。

- 点击气球时，会发出“POP”的音效。
- 让河豚角色也随机出现在舞台上，如果不小心点到它会扣分！

## 6.7.5 射击游戏

### 描述

你在电视上看过那些射击游戏吗？参赛者向靶心射出的子弹越接近靶心，他的得分就越高。

今天我们也在 Scratch 做一个射击游戏。游戏中，用十字准星尽量射中靶心以获得更高的分数。

点击绿旗开始。使用空格键发射子弹。

或者点击 [Shooting Game](#), 然后在 Scratch 官网学习在线教程。

### 所需组件

- A RasPad 3
- Scratch 3 (either online or offline)
- Crosshair(十字线) 精灵
- Target（标靶）背景

### 你会学到

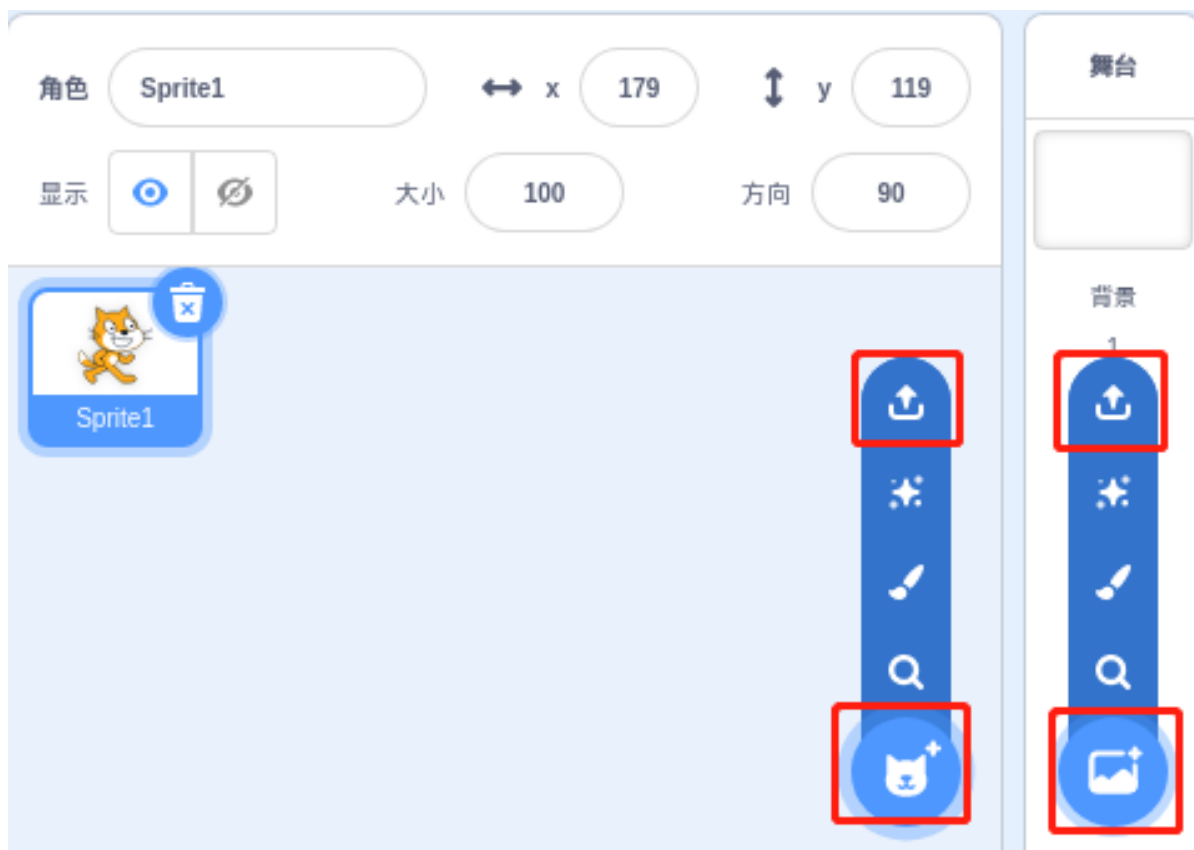
- 将新的角色和背景图片上传到 Scratch。
- Scratch 阶段的坐标范围。
- 停止其他角色运行的脚本并获取游戏结果。
- 使用 Scratch 颜色选择器。

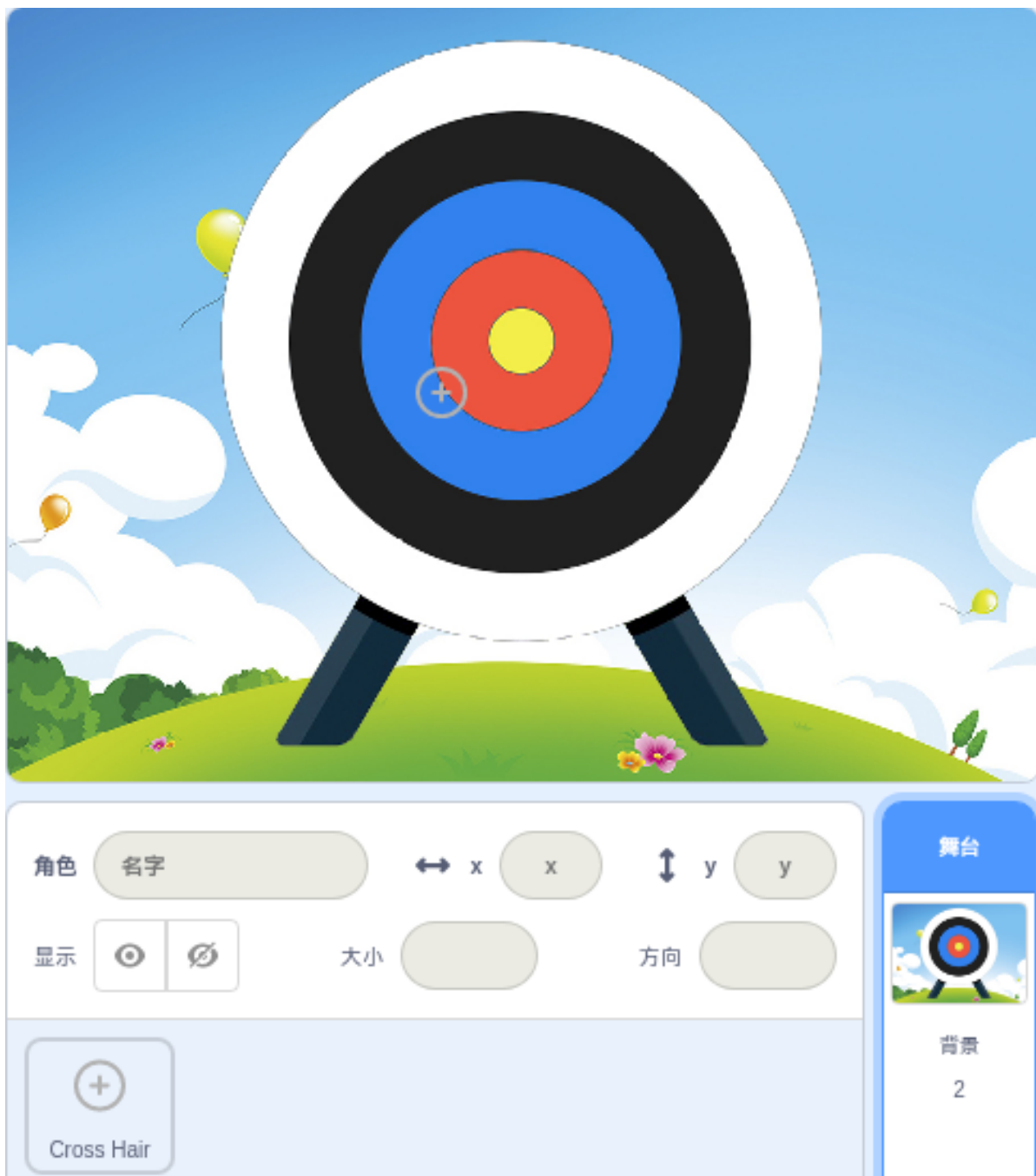
## 课程指南

## 排列十字准星和靶子

首先删除之前的角色，上传角色十字准星和背景靶子，在路径 `./raspad3/scratch/materials`（之前从 `raspad3.git` 上下载的）下可以找到这些要上传的图片。

点击角色列表下方的图标，选择“上传角色选项”，找到 `Cross Hair.svg` 文件，上传。同样，上传背景 `Target.jpg`。





### 瞄准

瞄准的时候，十字准线会轻微晃动，我们可以用「在... 秒内滑行到...」的代码块来达到这个效果。



设置十字准星角色的随机位置和大小。



现在，点击绿旗，看看这个瞄准的效果……非常好。让我们猜猜，在「在... 秒内滑行到...」方块中改变时间，准星会变得更稳定吗？还是更摇摆不定？

### 发射！

我们现在要实现子弹射击的效果。

如何做到呢？先停止移动十字准线，然后让它快速收缩以产生射击效果。

这意味着当按下空格键时，我们需要停止摇动十字准线的代码，并让十字准星的尺寸缩小。





现在点击绿旗试试效果吧。

### 计算分数!

然后我们需要建立一个评分机制。射击不同的色环会得到不同的分数。离靶心越近，得分越高。

例如，在红色区域射门可以得分 200 分。让我们用「碰到颜色...?」块来判断。这个块有一个颜色选择器，我们用它来识别目标上的红色。



用同样的方法为不同的色环设置相应的分数。



## 挑战

相信你很快就能掌握这个游戏，下面可以尝试添加更多的内容来丰富你的游戏。

- 在一场比赛中，你可以打三轮，然后计算总分。

## 6.7.6 击打魔法鱼

### 描述

在项目开始之前，我会告诉你一条神奇鱼的故事。在浩瀚的海底世界，有一条神奇的魔鱼。当你用海星击中它时，它会变成另一条鱼。

今天我们用 scratch 来模拟一个海底世界，然后加入海星精灵和魔鱼精灵来制作一个“打魔鱼”的小游戏。

点击绿旗开始。按空格键发射海星。

或者点击 [Hit the magic fish](#), 然后在 Scratch 官网学习在线教程。

### 所需组件

- A RasPad 3
- Scratch 3 (either online or offline)

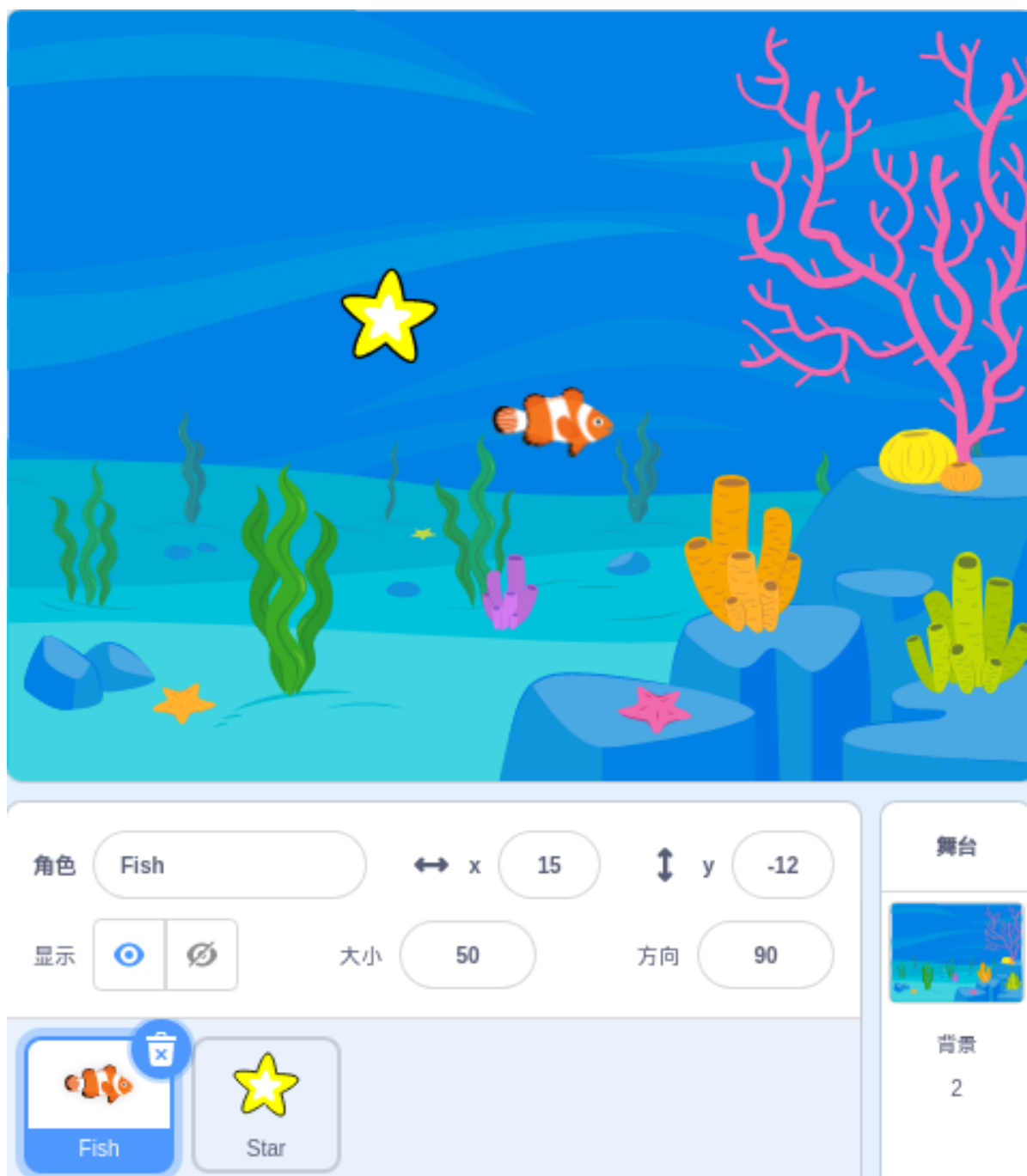
### 你会学到

- 让角色在碰到舞台边缘时反弹。
- 更改坐标以使角色移动。
- 做一个计时器。
- 使用广播告诉角色发生了什么。

### 课程指南

**这条鱼在游泳。**

添加角色 **Fish** , **Star** 以及背景 **Underwater1** 。



让鱼出现在坐标 (-240,120) 处, 然后让它左右游动。

使用「**碰到边缘就反弹**」代码块使鱼从边缘开始游动, 当碰到墙壁时, 它会朝着与原来相反的方向折回。



### 星星升起

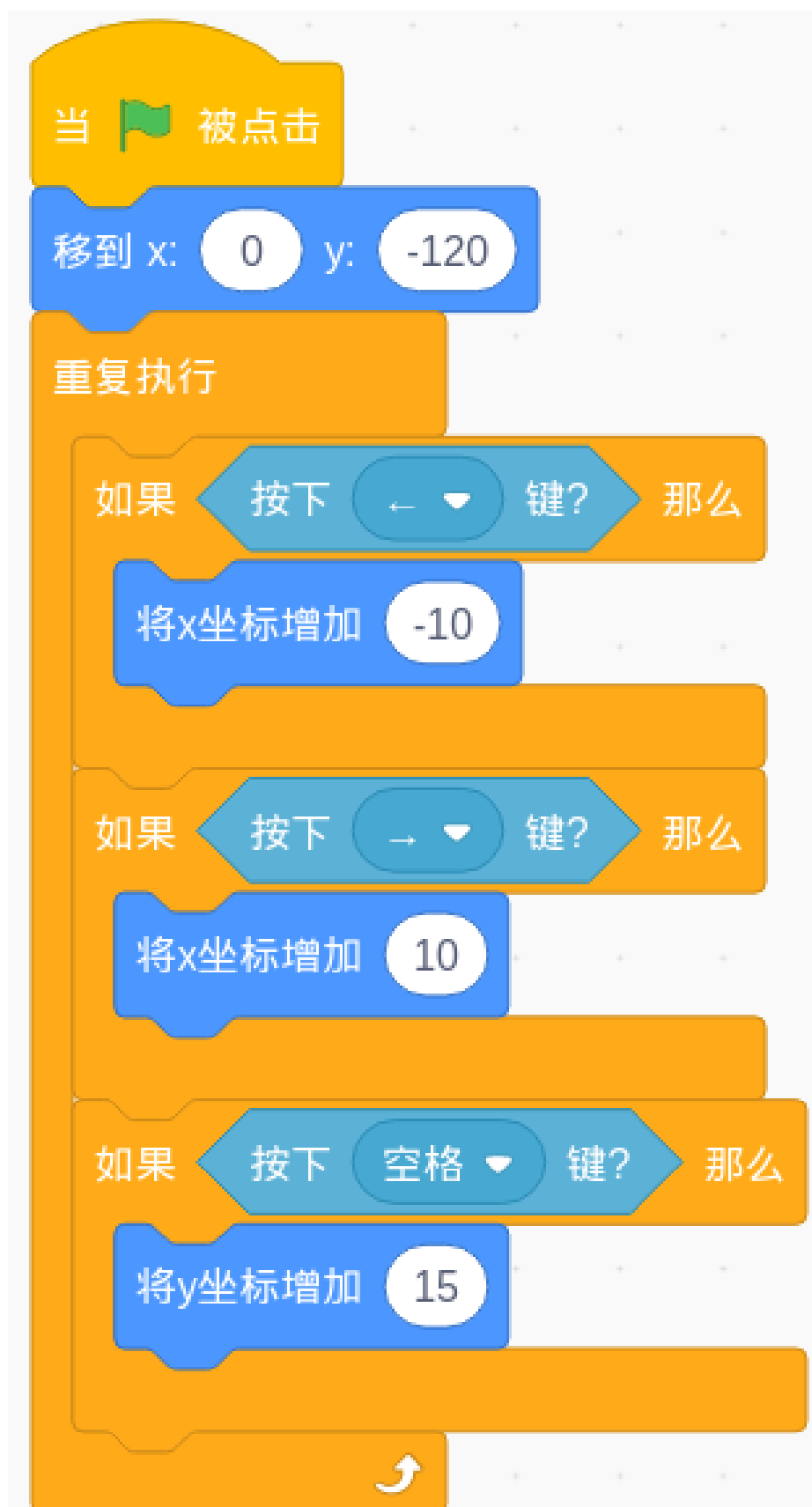
让星星出现在坐标 (0,-120) 上。如果按下 ← 或者 → 键，星星会向着左右移动。

用「按下... 键？」代码块来检测按键是否按下，并用「将 X 坐标增加...」来移动角色。



然后让星星动起来，如果按下空格键，星星的  $y$  坐标值会增加，星星上升。





有没有感觉到哪里不对劲？星星应该继续上升，直到达到和鱼一样的高度（还记得吗？我们为鱼设置的 y 坐标是 120）然后返回。



这样看起来就好多了。

### 用星星砸鱼

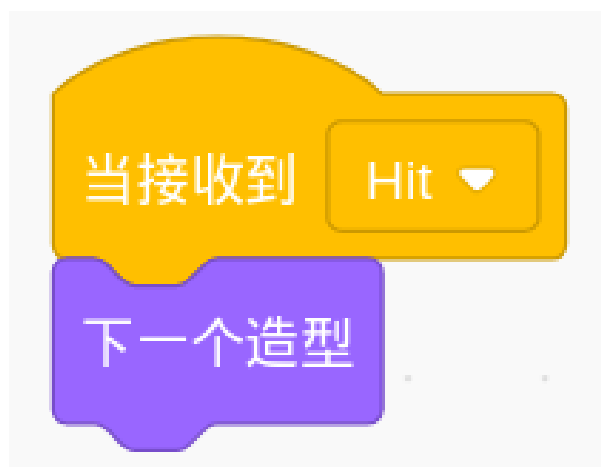
星星升起之后可以用「碰到...?」代码块来判断是否砸到了鱼。

当星星砸到鱼之后，我们需要发出一个消息告诉鱼，并让鱼做出相应的反应--改变鱼的形状。

我们设置了“Hit”广播。这个广播可以发出一个“Hit”的消息，让所有角色或者背景知道并作出相应的反应。



回到鱼的编程界面编写鱼接收到“Hit”之后的反应，用「当接收...」代码块判断是否接收到信息，然后用「下一个服装」来改变鱼的形态。



### 计算得分

为了让游戏更有趣，可以添加评分机制。当单击绿旗时，分数将重置为零。当接收到“Hit”消息时，分数增加 1。

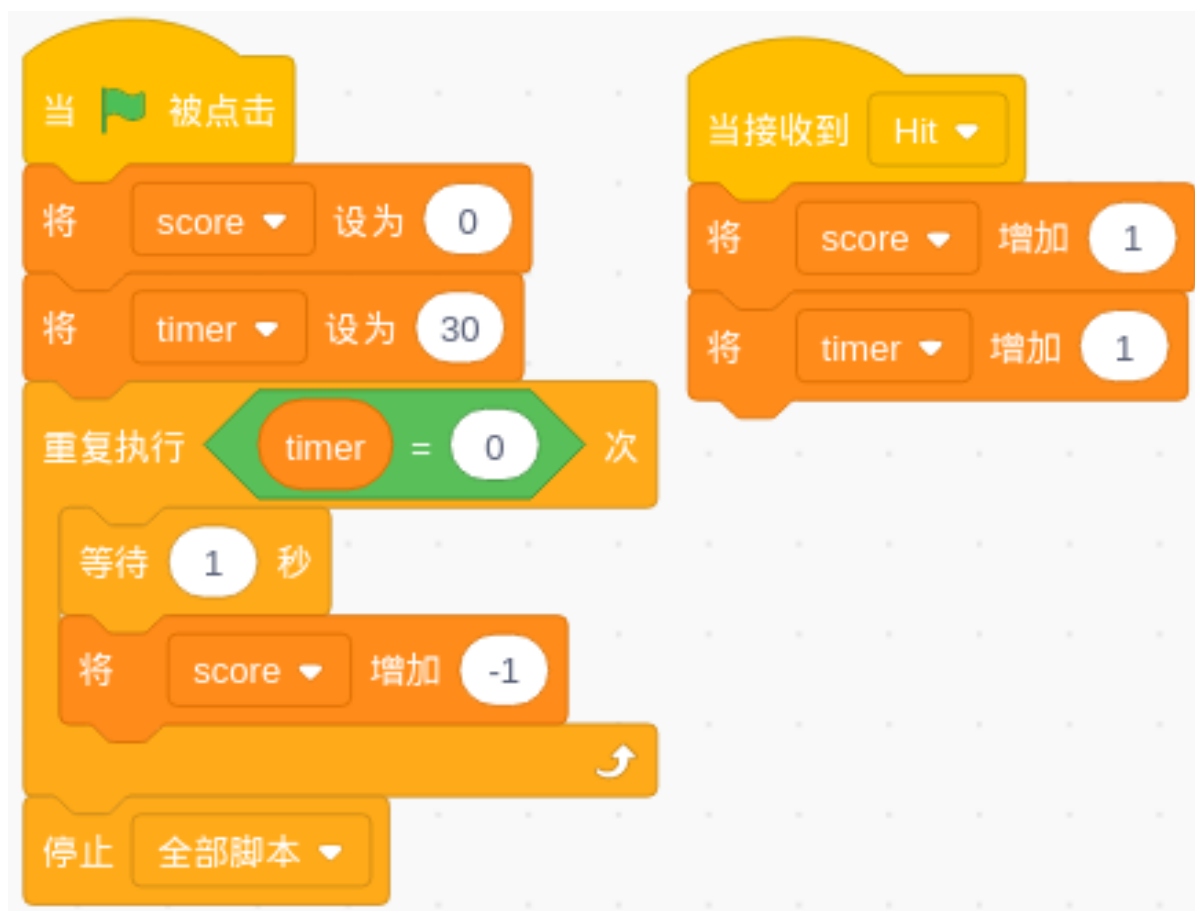


**注解：** 你可以将计算得分的代码块放在背景中。

### 倒计时

使游戏更丰富还可以添加时间机制。当点击绿旗时，计时器设置为 30s，然后每秒递减 1，直到时间减为 0 的时候结束游戏。

但如果接收到“Hit”的信息时，可以让计时器加 1 以增加游戏时间。



### 挑战

相信你很快就能掌握并实现这个游戏。接下来可以添加下面的内容来使你的游戏更丰富。

- 添加三条不同的魔法鱼
- 击中其中一条会获得双倍积分
- 几种另外一条会减少游戏时间
- 击中第三条会更换背景

## 6.7.7 二进制游戏

### 描述

你知道二进制吗？二进制是计算技术中广泛使用的数字系统，因此了解二进制数是了解计算机世界的重要一步。

点击绿旗开始。屏幕中间会出现一个随机数，你需要知道对应的二进制数。比如 5，对应的二进制数应该是 0101，按右边的熊掌确认。如果正确，分数将增加 1。

或者点击 [Binary Game](#)，然后在 Scratch 官网学习在线教程。

## 所需组件

- A RasPad 3
- Scratch 3 (either online or offline)

## 你将学会

- 二进制数基本概念。
- 如何将二进制数转换为十进制数。

## 课程指南

在屏幕底部布置一个小的二元算盘。

首先，我们删除之前的角色。上传角色列表中之前从 [raspad3.git](https://github.com/raspad3) 下载的 **Zero.png** 文件和 **Enter.png** 文件。然后我们在 **Zero** 角色的 **Costumes** 选项中上传 **One.png** 文件，让 **Zero** 角色拥有 **Zero** 服装和 **One** 服装。



然后我们用鼠标右键复制 **Zero** 角色，舞台区域就会出现 3 个新的 **Zero** 角色。



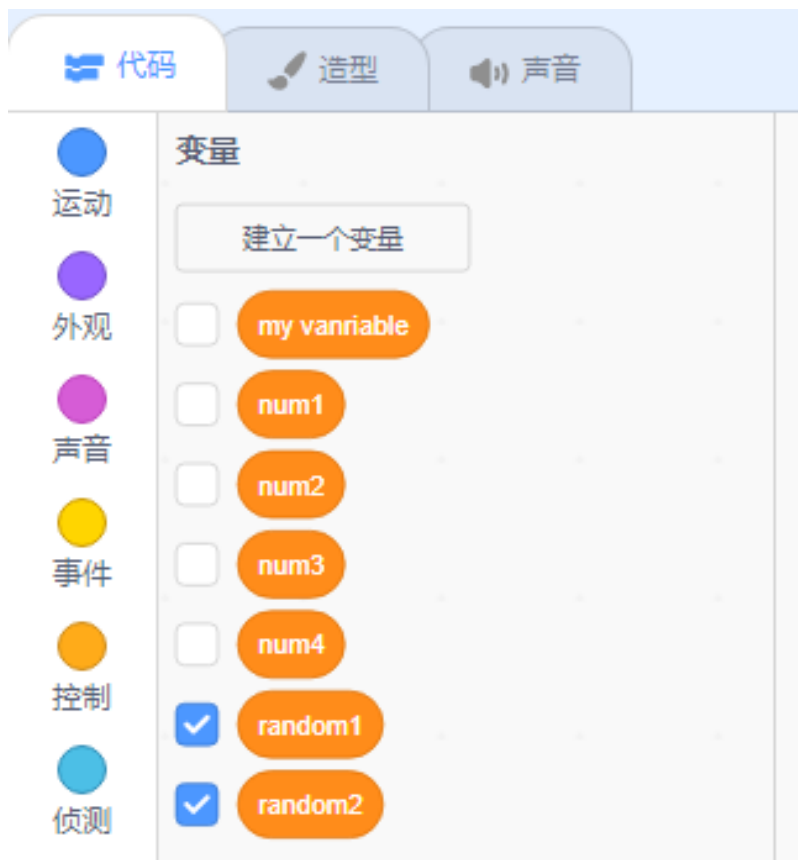
将四个精灵的默认服装设置为 **Zero**，分别用变量“num1”、“num2”、“num3”和“num4”进行记录，并初始化四个角色的位置。

下图中的程序是 **Zero** 角色的程序。其他三个 **Zero** 角色则修改变量名称，并将 X 坐标增加 80，使效果从左到右依次排列。



当我们点击变量前面的 ✓ 符号时，变量将从舞台上消失。





通过单击并记录二进制值来改变形状。

如果点击 **Zero** 角色，它会切换它的服装到下一个服装，并切换变量 num1 的值，如果 num1 = 0，则将 num1 更改为 1。

这样每次 **Zero** 角色切换服装时，num1 变量就可以实时记录黑板上的数值。

复制其余三个 **Zero** 角色的程序并修改变量名称。



### 二进制数和十进制数的转换。

在开始编程之前，我们先来了解一下 [二进制数](#)。

二进制数是以 2 为基数的数字系统或二进制数字系统表示的数字，它只使用两个符号：通常是“0”（零）和“1”（一）。

如何将二进制转换为十进制？

对于  $n$  位二进制数：

```
dn-1 ... d3 d2 d1 d0
```

十进制数等于二进制数字的总和 ( $dn$ ) 乘以它们的 2 次方 ( $2^n$ ):

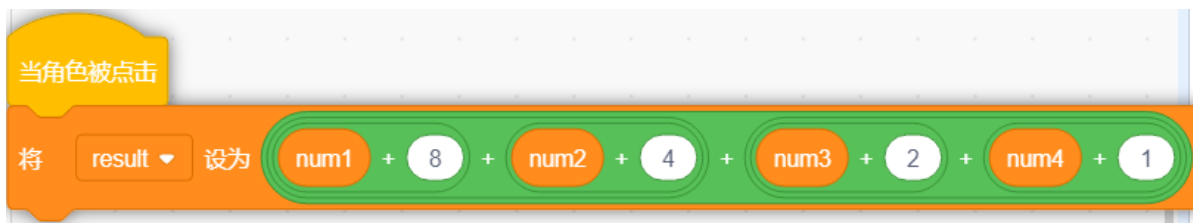
```
decimal = d0×2^0 + d1×2^1 + d2×2^2 + ...
```

如果二进制是 1000，对应的十进制应该是  $1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 8$

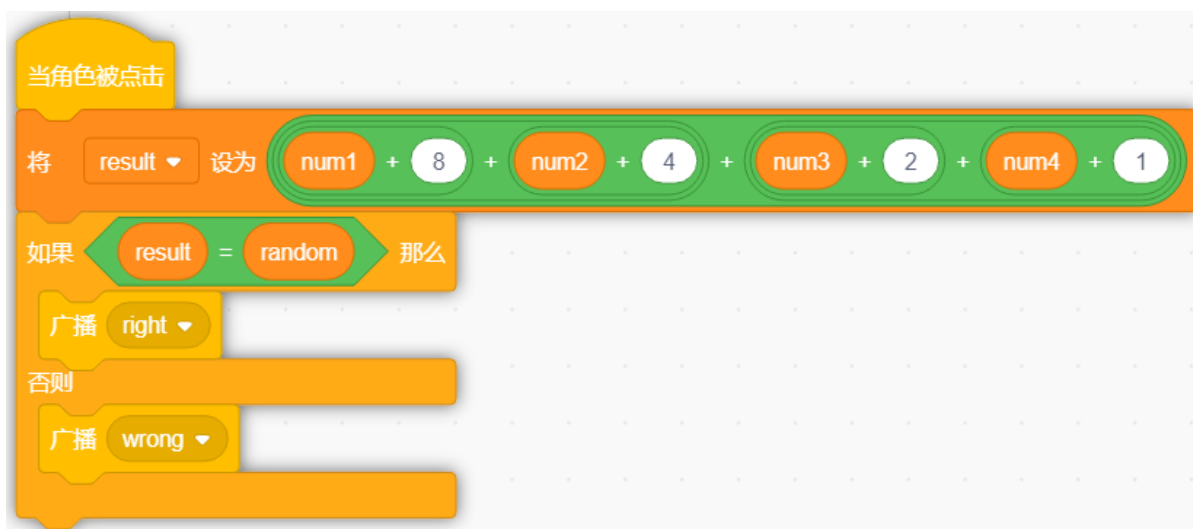
那么，1010 的十进制数是多少？ $1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 10$ 。

## 为背景编程

游戏开始时，我们的屏幕中间会出现一个 1-16 之间的随机整数，作为一个十进制数，需要进行二进制转换，存储在变量 `random` 中。



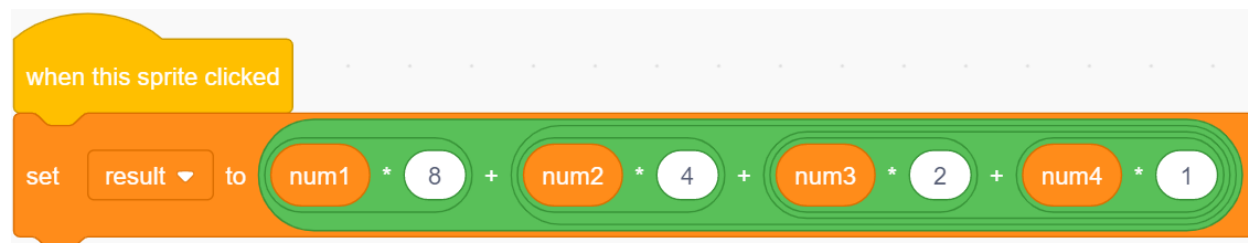
无论我们收到“正确”的结果还是“错误”的结果，我们都必须重新生成随机数并开始下一轮游戏。



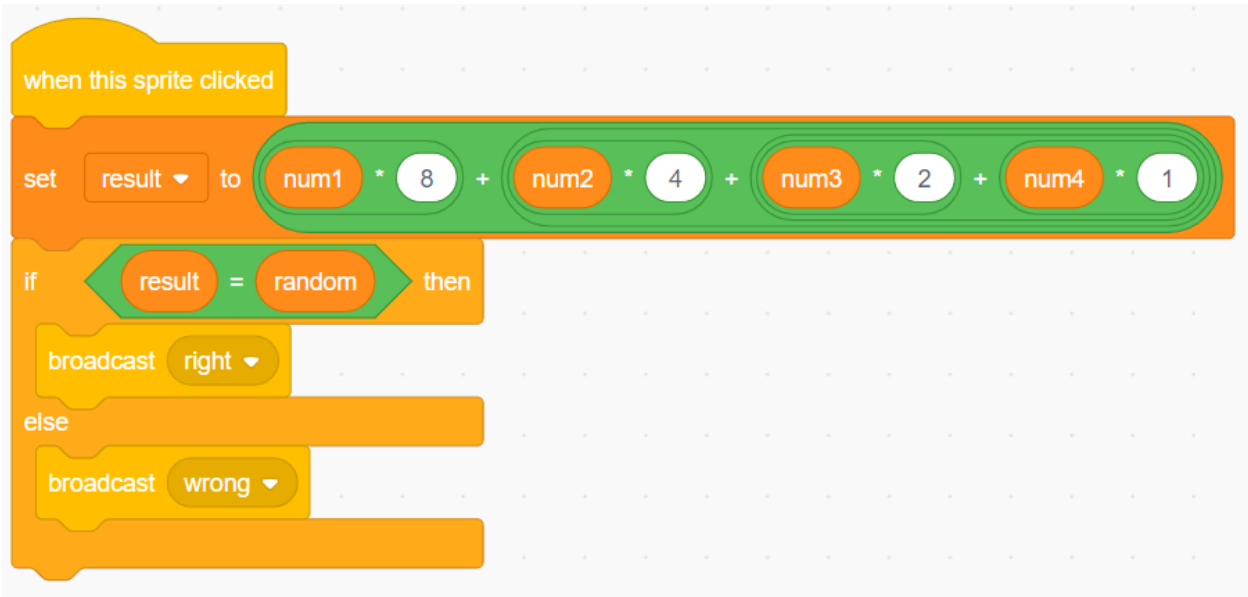
## 为 Enter 角色编程

了解了二进制数和十进制数的转换后，我们来编写 **Enter** 角色。

首先，我们创建一个结果变量，当点击 **Enter** 角色时，我们将 `num1` 变量、`num2` 变量、`num3` 变量和 `num4` 变量组成的二进制数转换为十进制数，并将结果保存在 `result` 变量中。



然后比较这个 `result` 变量和随机变量的值，如果相等就返回一个“正确”的消息，否则返回一个“错误”的消息。



现在，点击舞台区域上方的绿旗开始游戏并探索神奇的二进制！

### 挑战

我相信你很快就能掌握并实现这个游戏。接下来，可以添加一些额外的内容来使你的游戏更丰富。

- 添加一个 **Zero** 角色来转换 1-32 之间的随机整数。
- 当您单击 **Zero** 角色时，将播放流行音乐。
- 答错扣 1 分！

## 6.7.8 会飞的鹦鹉

### 描述

今天我们让鹦鹉 **德文** 避开树木觅食。

点击绿旗开始。

或者点击 [Flappy Parrot](#), 然后在 [Scratch 官网](#) 学习在线教程。

### 所需组件

- A RasPad 3
- Scratch 3 (either online or offline)

## 你会学到

- 为角色添加更多服装，让它们改变外观
- 逻辑运算
- 确定角色是否相互接触

## 课程指南

### 安排鹦鹉和背景图片

首先，我们删除默认角色并添加鹦鹉角色和蓝天背景。将鹦鹉的大小更改为 20。

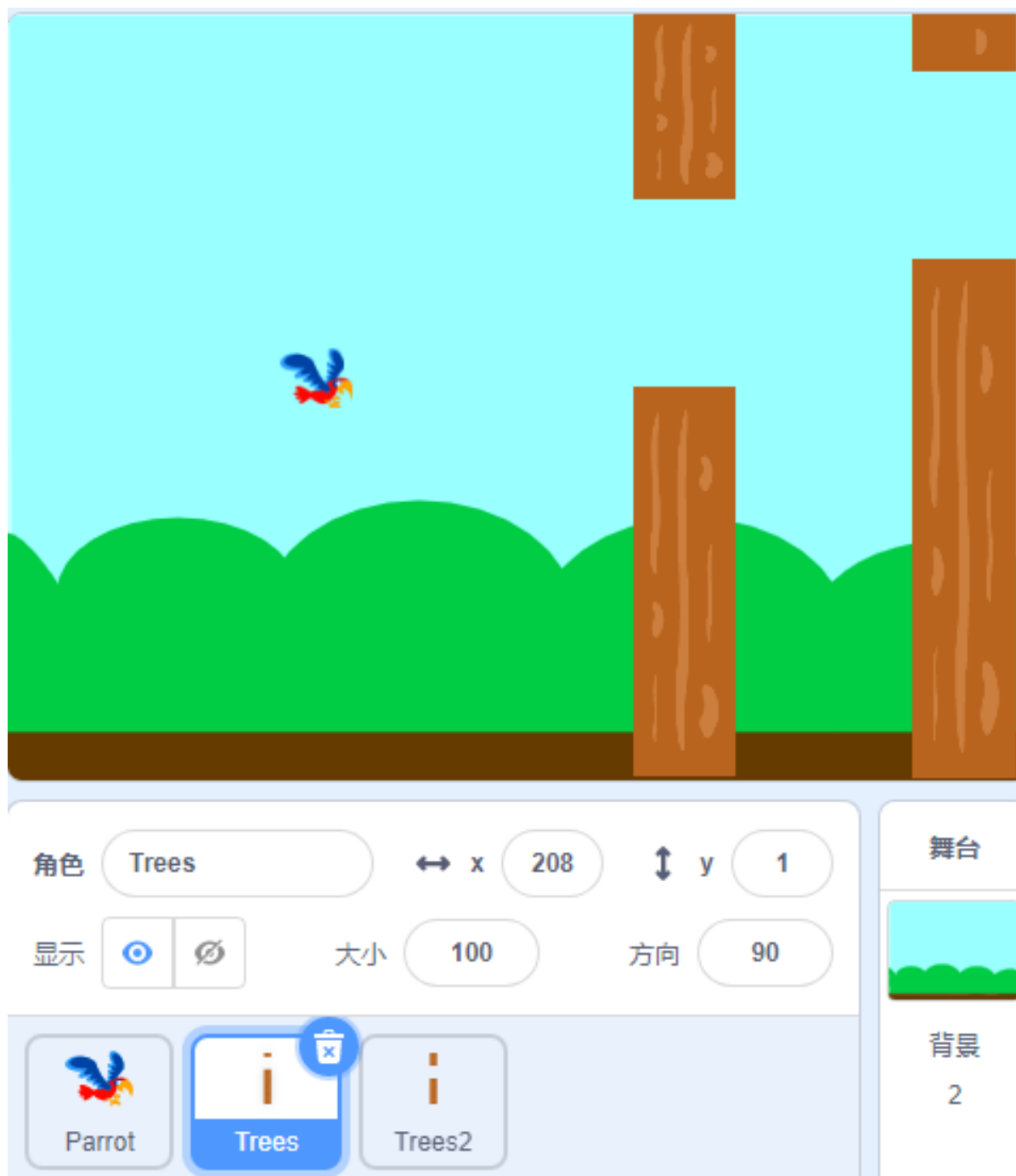


上传 Trees 角色，选择 Upload Sprite 选项，找到 “Trees.png” 文件，上传。

单击 “Costumes”，然后单击 “Upload Costume” 选项以上传 Trees2.png、Trees3.png、Trees4.png 和 Trees5.png。



复制第二棵树。



### 让鸚鵡在舞台上飞翔

鸚鵡一动不动，Trees 和 Trees2 角色一直向左移动，所以看起来鸚鵡在舞台上飞翔！

设置 Trees 的初始位置和移动方向，使其可以从舞台的最右侧到达最左侧。然后，随机变化的服装重新出现在舞台的最右侧。



角色 Trees2 的代码与 Trees 几乎相同，唯一不同的是它在开始时隐藏了 1.5 秒。





现在，点击绿旗，看看这次飞行的效果……很好，鹦鹉好像飞到了舞台的右侧。

### 按空格键控制鹦鹉飞

游戏开始时，鹦鹉会继续往下掉，也就是 y 轴坐标会继续往下递减。当我们按下空格键时，鹦鹉的 y 轴坐标会增加。



### 通过变量不断提高鹦鹉下落的速度

鹦鹉掉落有点慢，我们添加飞行变量让鹦鹉掉落越来越快！



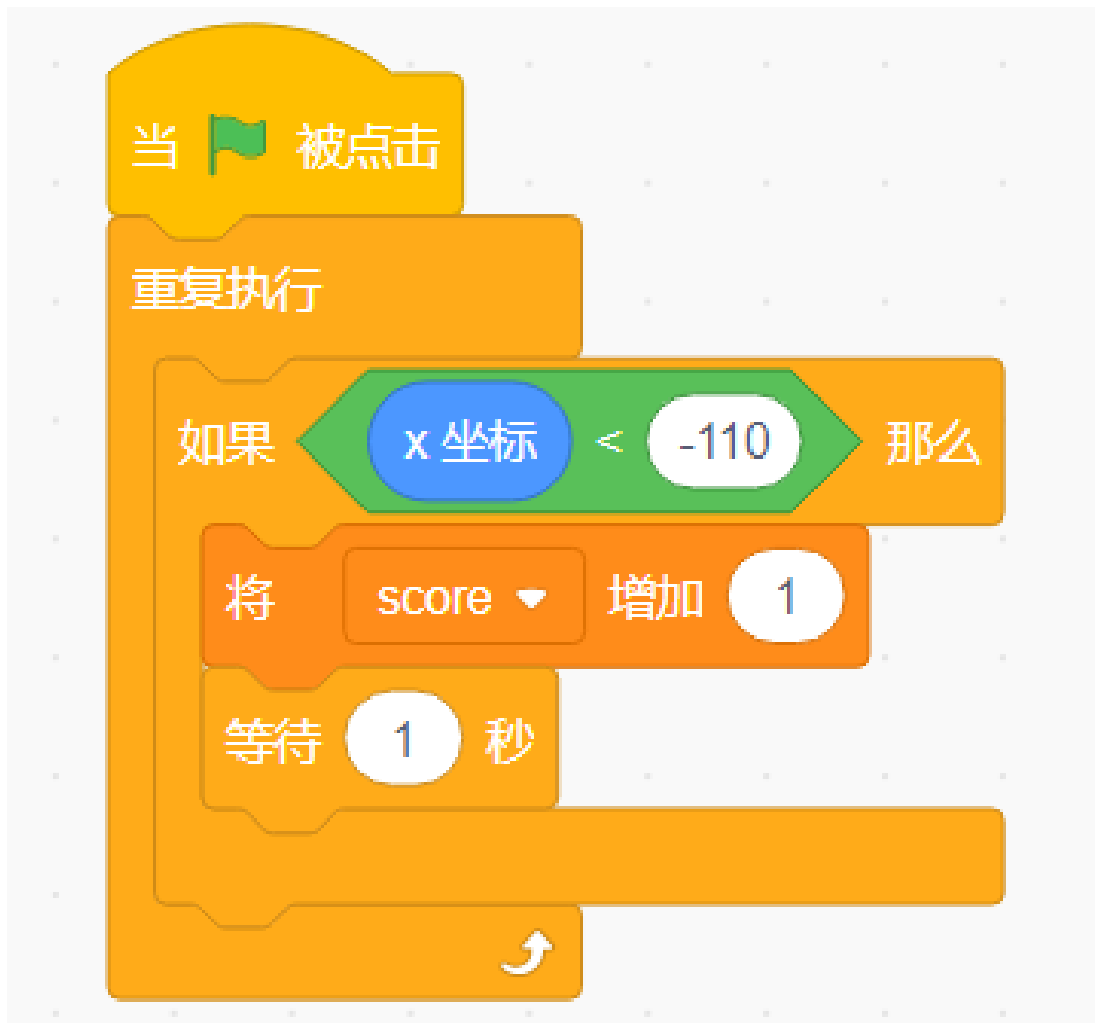
### 改变服装让鹦鹉有飞行动画

当我们按下空格时，鹦鹉的衣服会切换。



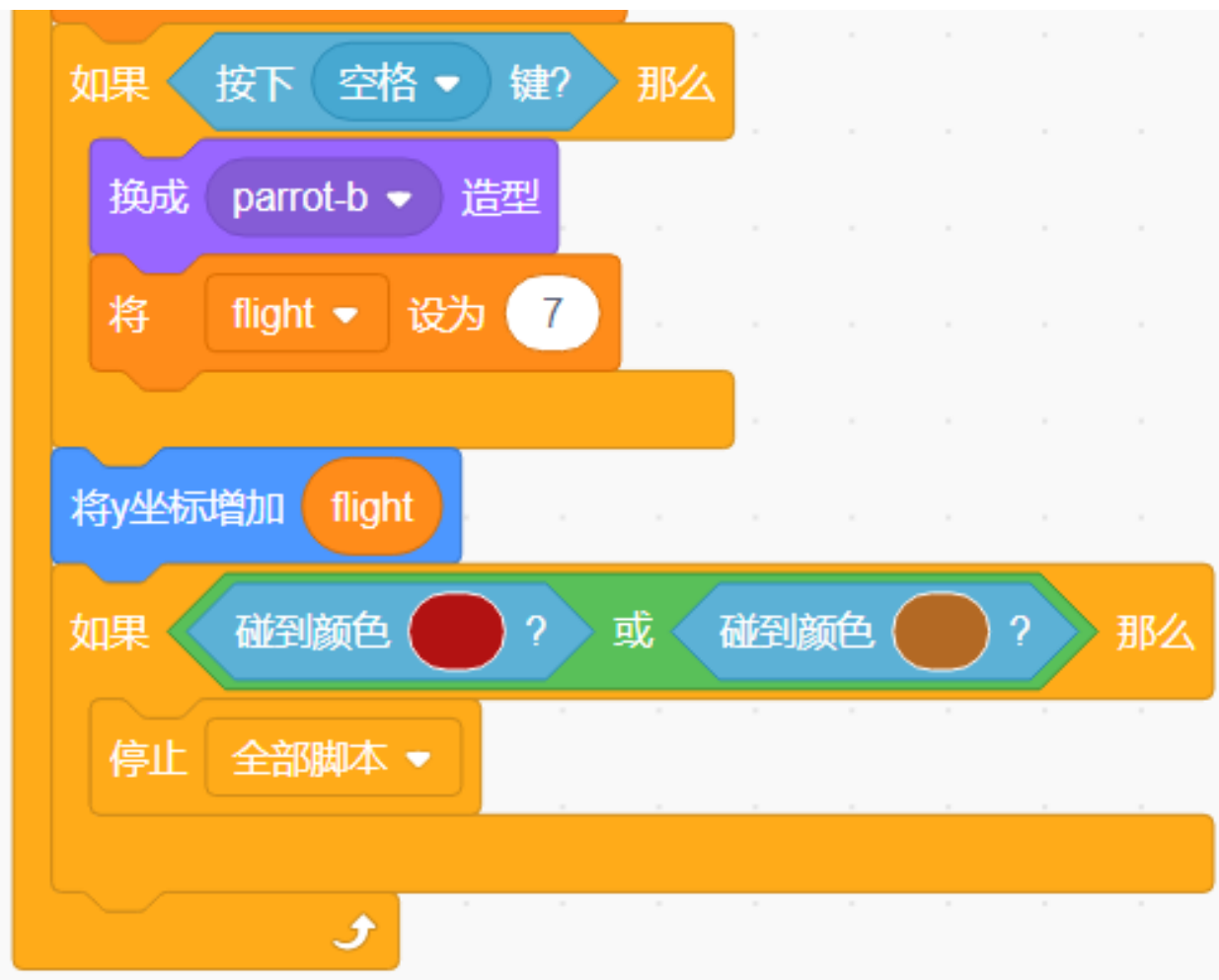
### 计算分数!

创建一个新的分数变量，当 Trees 和 Trees2 移动到鹦鹉的右侧时，表示鹦鹉已经穿过 Trees 和 Trees2。这时候给得分加一。为了避免总是加分，我们需要在加完分数后面加上一秒的延时。（此代码在 Trees 和 Trees2 角色中都需要添加!）



### 游戏失败机制

当鹦鹉在飞行过程中遇到一棵树或掉到地上时，我们判断游戏结束，然后我们停止所有脚本。



### 挑战

我相信你很快就能掌握并实现这个游戏。下面可以添加额外的内容来使你的游戏更丰富。

- 当游戏失败时，它会切换到诅咒背景屏幕。
- 游戏开始画面新增倒计时画面，让我们为比赛做好准备。

### 6.7.9 油漆盒

#### 描述

你用过油漆盒吗？油漆盒中的彩色画笔可以用来绘制多彩的世界。今天，我们将制作出一个独特的油漆盒。下面让我们来一起看看吧。

点击绿旗开始。

或者点击 [Paint Box](#), 然后在 [Scratch 官网](#)学习在线课程。



### 所需组件

首先，需要准备以下组件：

- A RasPad 3
- Scratch 3 (在线或者离线)

### 你会学到

- 使用画笔扩展功能进行绘制。
- 让角色随着鼠标移动。
- 修改服装的大小和光标的位置。
- 使用滑块修改变量的值。

### 课程指南

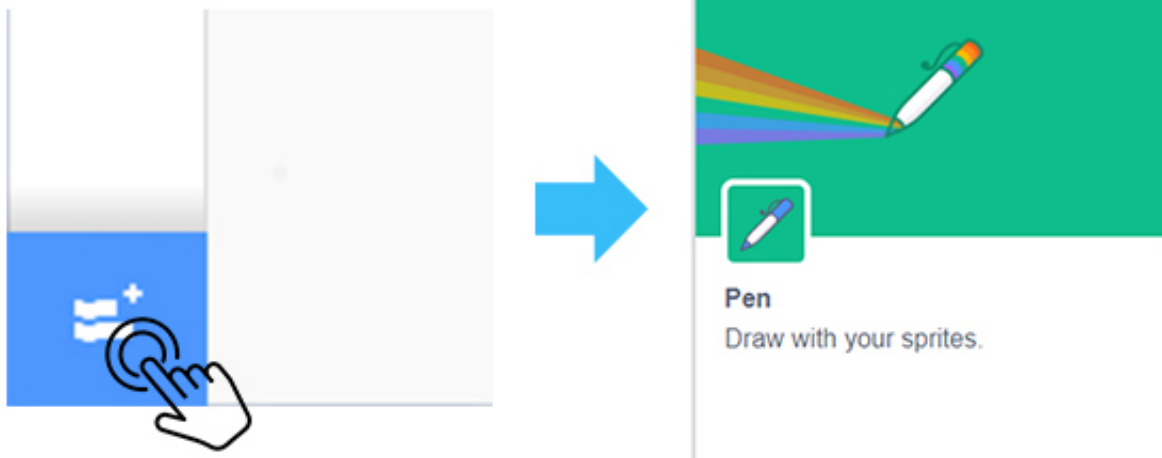
#### 布置画板，增加画笔扩展功能

首先，我们删除之前的角色，然后将 Pencil 角色和 Block-X 角色添加到 Scratch。

上传角色列表中的 Eraser.svg、Red.sprite、Blue.sprite、Green.sprite、Yellow.sprite 等文件（之前从 [raspad3.git](#) 上下载的），最后将 Pencil 角色和 Block-X 角色的大小分别更改为 50 和 25。

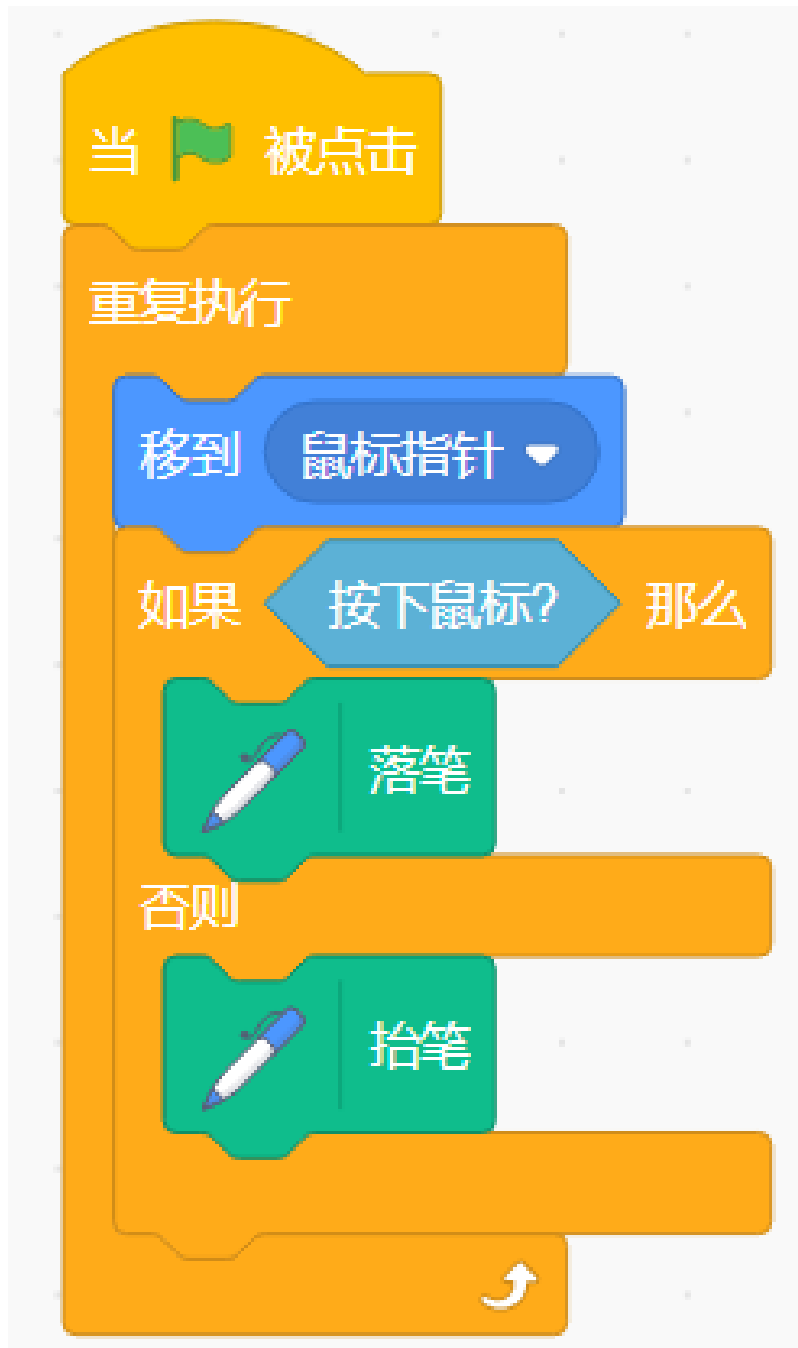


然后添加扩展功能 **Pen**。



### 为 Pencil 添加绘图功能

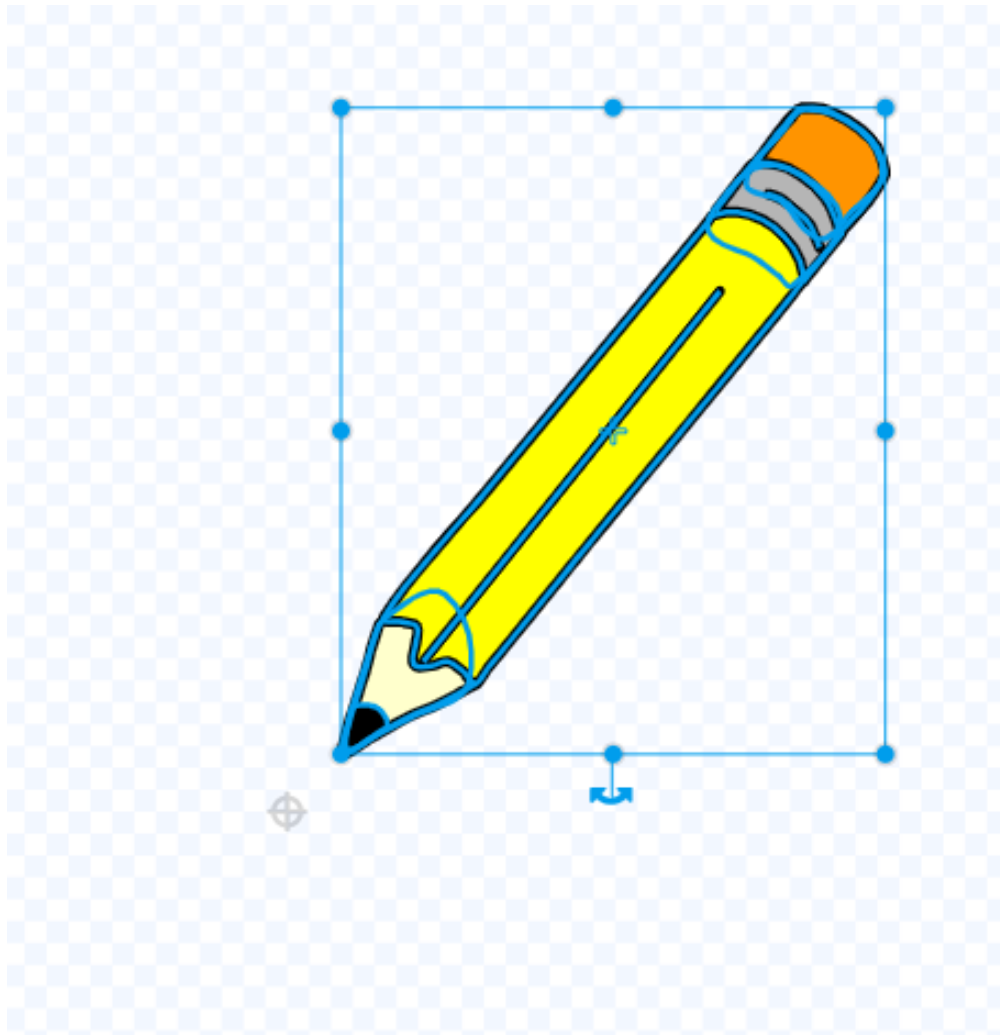
单击 Pencil 角色。首先，我们需要让 Pencil 随着鼠标移动。如果按下鼠标按钮，我们会调用 Pen 扩展函数中的「落笔」块，它可以沿着光标的轨迹进行绘制。否则就会调用「抬笔」块停止绘图。



点击绿旗，尝试用鼠标控制铅笔画一个笑脸。您会发现图像不是从 Pencil 的尖端绘制的。

**注解：** 由于光标不在铅笔尖上，所以绘制的线是从铅笔的中间绘制的。

点击 **造型**, 选择整个 Pencil 角色并移动它直到画笔尖端与十字光标对齐。



改变铅笔的绘画颜色。

如何实现当我们点击舞台下方的颜色选项时，Pencil 会切换相应的颜色？

首先我们为红色部分编写代码，当它被点击时，广播发出一个“红色”消息。



然后返回到铅笔部分的代码，当它收到“红色”消息时，我们将铅笔的颜色更改为红色。



同样的方法，为蓝色部分编写代码，当它被点击时，广播发出一个“蓝色”消息。



再次返回铅笔部分的代码。当它收到“blue”消息时，我们将 Pencil 的颜色更改为蓝色。在绘画之前，我们清除舞台上的场景并将铅笔颜色设置为红色。



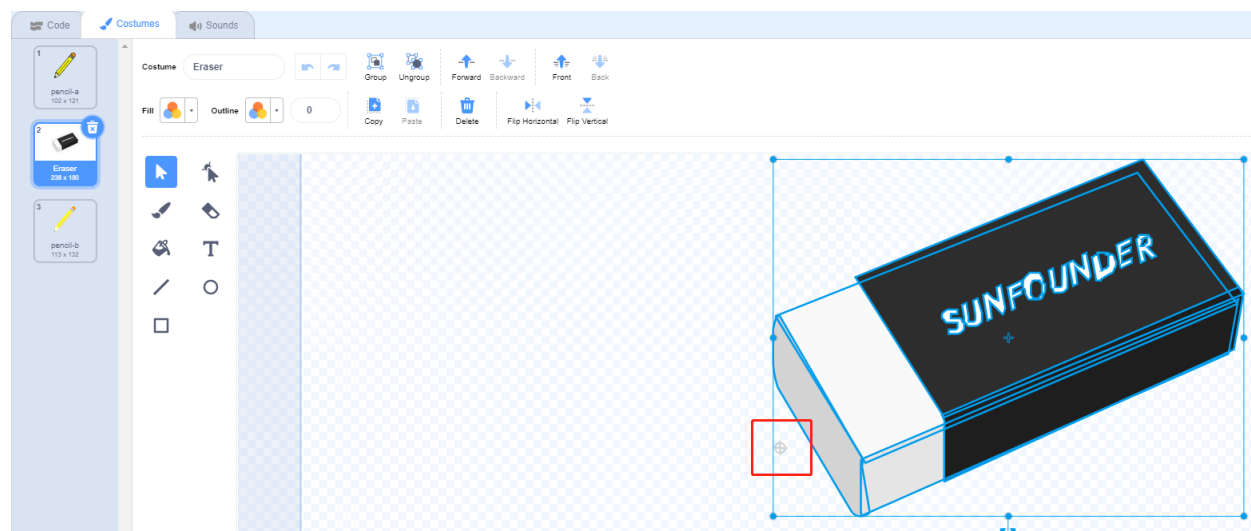
### 增加清晰绘图功能。

有时我们在绘制时会发生意外，所以我们需要添加一个擦除的功能。首先选择 **Block-X** 角色，当它被点击时，清除所有屏幕。



然后为铅笔设置一个新的造型, 单击”造型”选项, 从左下角的图标中选择 **上传造型** 选项来上传 **Eraser.svg** 文件。

适当修改橡皮擦的大小, 然后将其放置在光标的右上角。



开始为橡皮擦编写代码, 当它被点击时, 广播发出一条“橡皮擦”消息。



然后继续为铅笔添加代码，当收到橡皮擦发来的信息时，铅笔就会切换成橡皮擦。这样当我们点击橡皮擦时，就可以让铅笔切换变成橡皮擦了。然后将颜色改为与舞台背景色一致的白色，从而实现局部擦除的功能。





### 切换画笔以及它的宽度

当铅笔切换成橡皮擦时，我们可以通过点击绘图板上的红色和蓝色将橡皮擦重新切换回铅笔。



我们还可以随时改变铅笔的宽度。

新建一个名为“width”的变量，然后右击舞台上的“width”变量，选择滑杆，你就可以在舞台上看到一个滑块，通过它可以改变“width”变量的值。



用变量“width”来调整铅笔的大小。

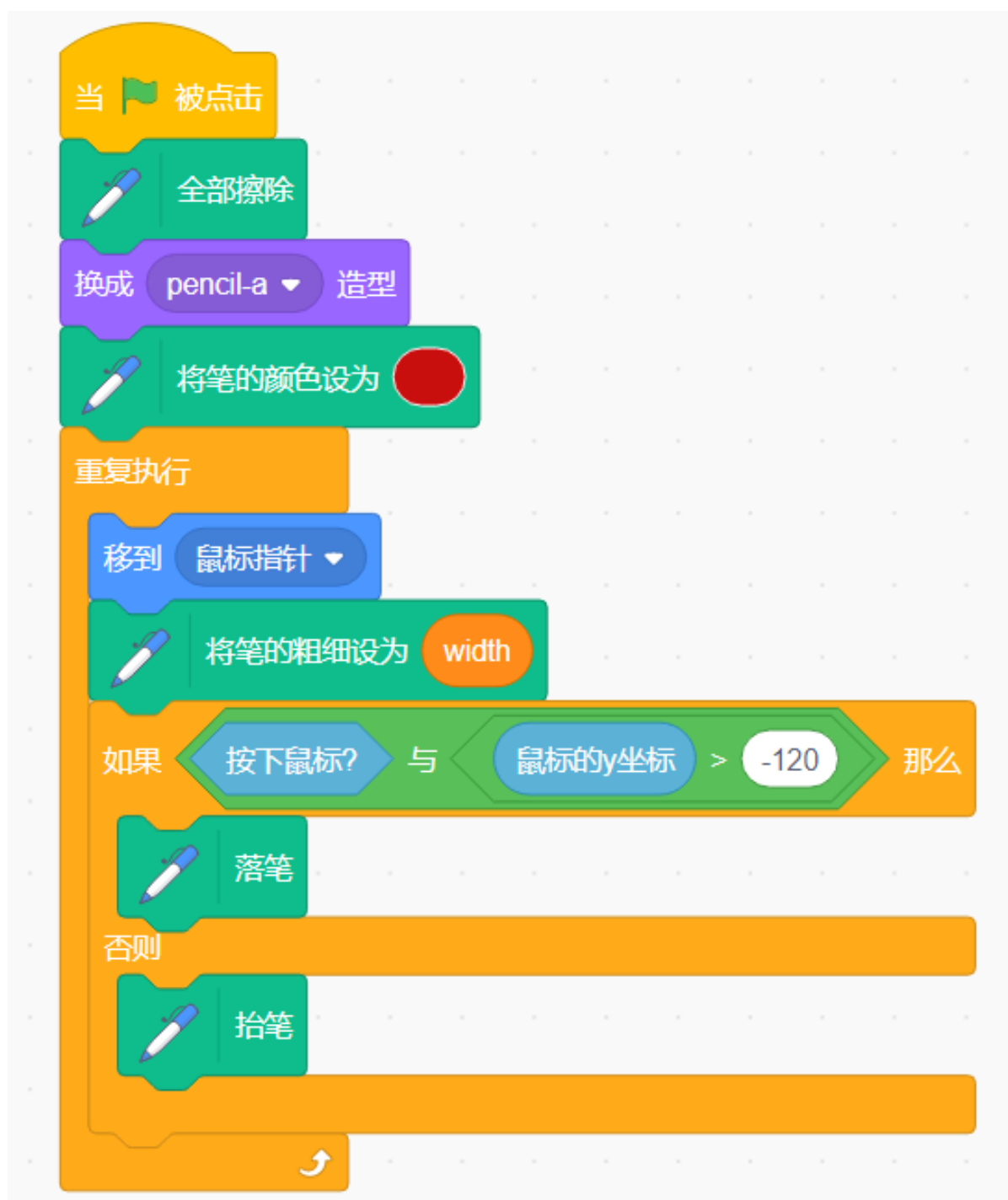


### 限制铅笔的绘画范围

只有当铅笔进入绘图区，才可以进行绘画。



当钢笔进入工具栏时，也就是当  $y$  坐标小于等于 -120 时，我们需要调用「pen up」函数停止绘画。



## 挑战

相信你很快就能掌握并实现这个游戏，下面可以添加额外的内容来使你的游戏更丰富。

- 添加更多颜色画笔来制作属于您的油漆盒。

## 6.7.10 吃香蕉游戏

### 描述

Scratch 有一个 **Video Sensing** 扩展模块，可以在 Scratch 中开启摄像头，检测摄像头屏幕上物体的移动。

今天，我们将使用相机制作一个吃香蕉的游戏。在规定的时间内，帮助猴子多吃香蕉。

要在白色背景下玩游戏，请单击绿旗开始。移动相机前的彩色物体来控制猴子精灵。

或者点击 [Eating Banana Game](#), 在 Scratch 官网学习在线教程。

### 所需组件

- A Raspad
- Scratch 3 (离线或者在线)

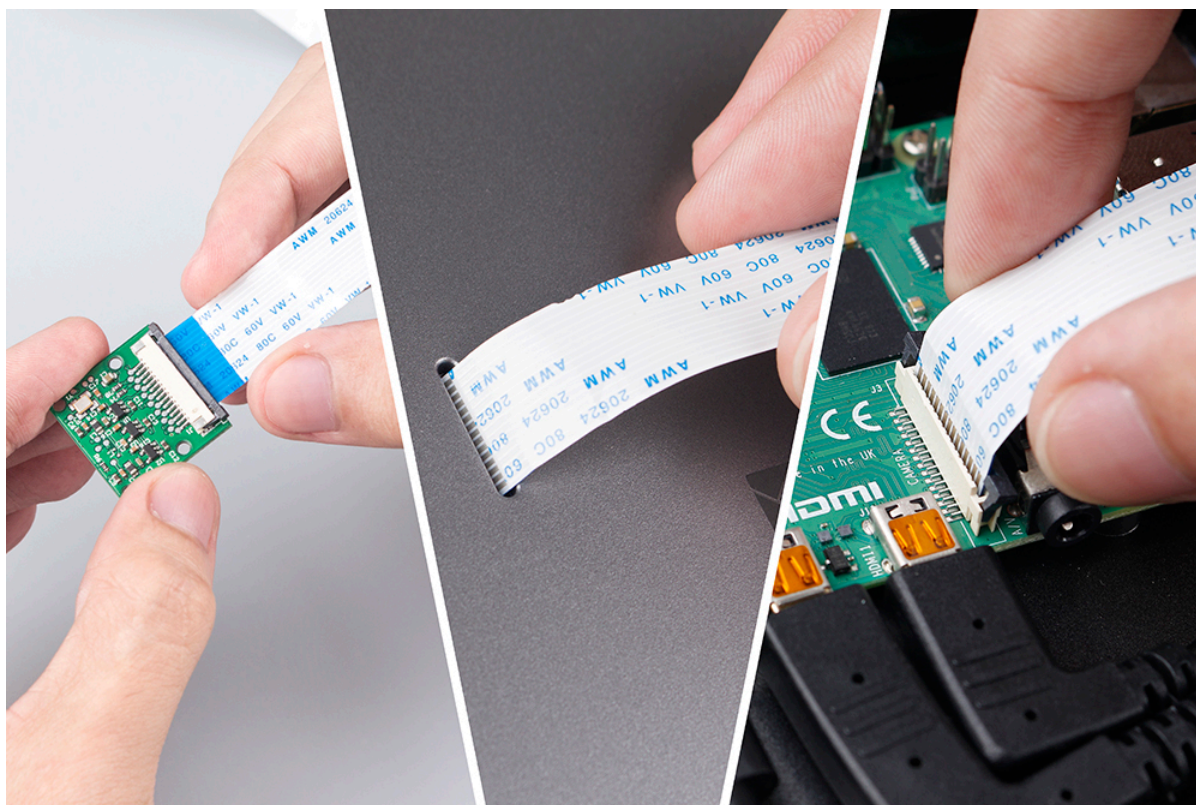
### 你会学到

- 在 RasPad 上安装摄像头。
- 增加 Video Sensing 扩展模块，使用摄像头功能
- 让相机检测我们的动作

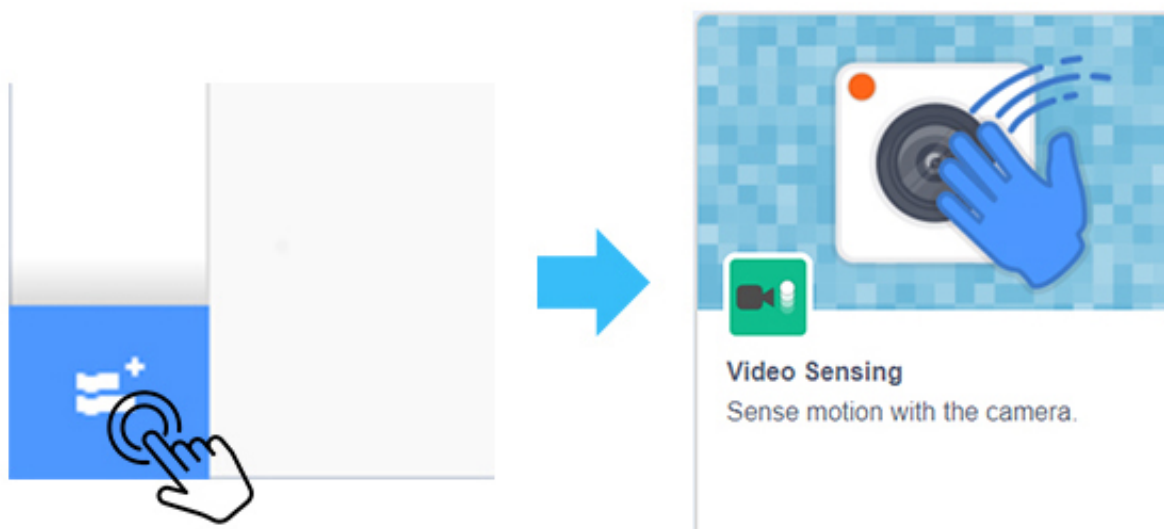
### 课程指南

#### 安装和使用摄像头

摄像头安装教程请参考[安装摄像头](#)。



单击 scratch 左下角的 **Add Extension** 图标，然后选择 **Video Sensing** 扩展。



此时，您可以看到在 Scratch 舞台上拍摄的相机。

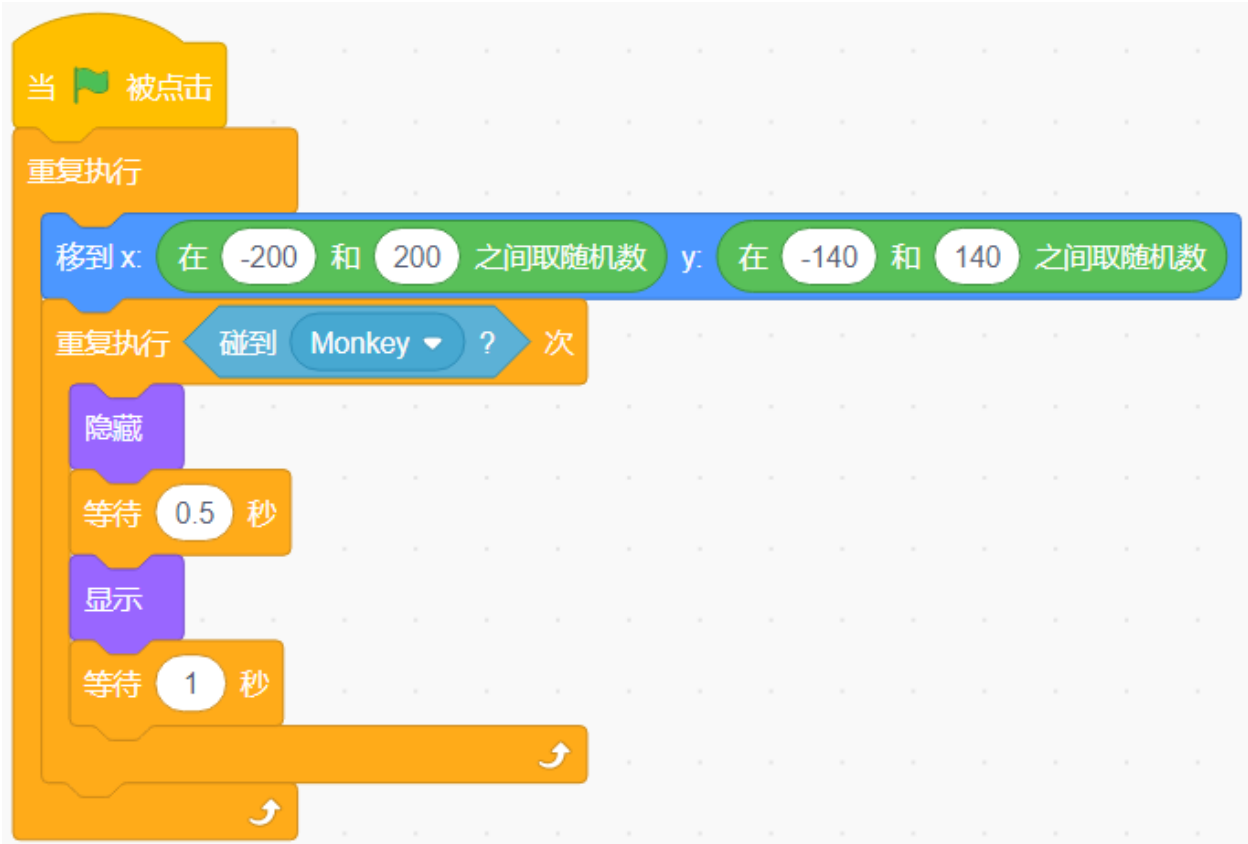
### 安排猴子和香蕉

首先，我们删除原始角色，然后添加猴子和香蕉角色，并将它们的大小更改为 50。

让香蕉随机出现。



香蕉在遇到猴子后消失，这意味着它被猴子吃掉并随机重新出现。

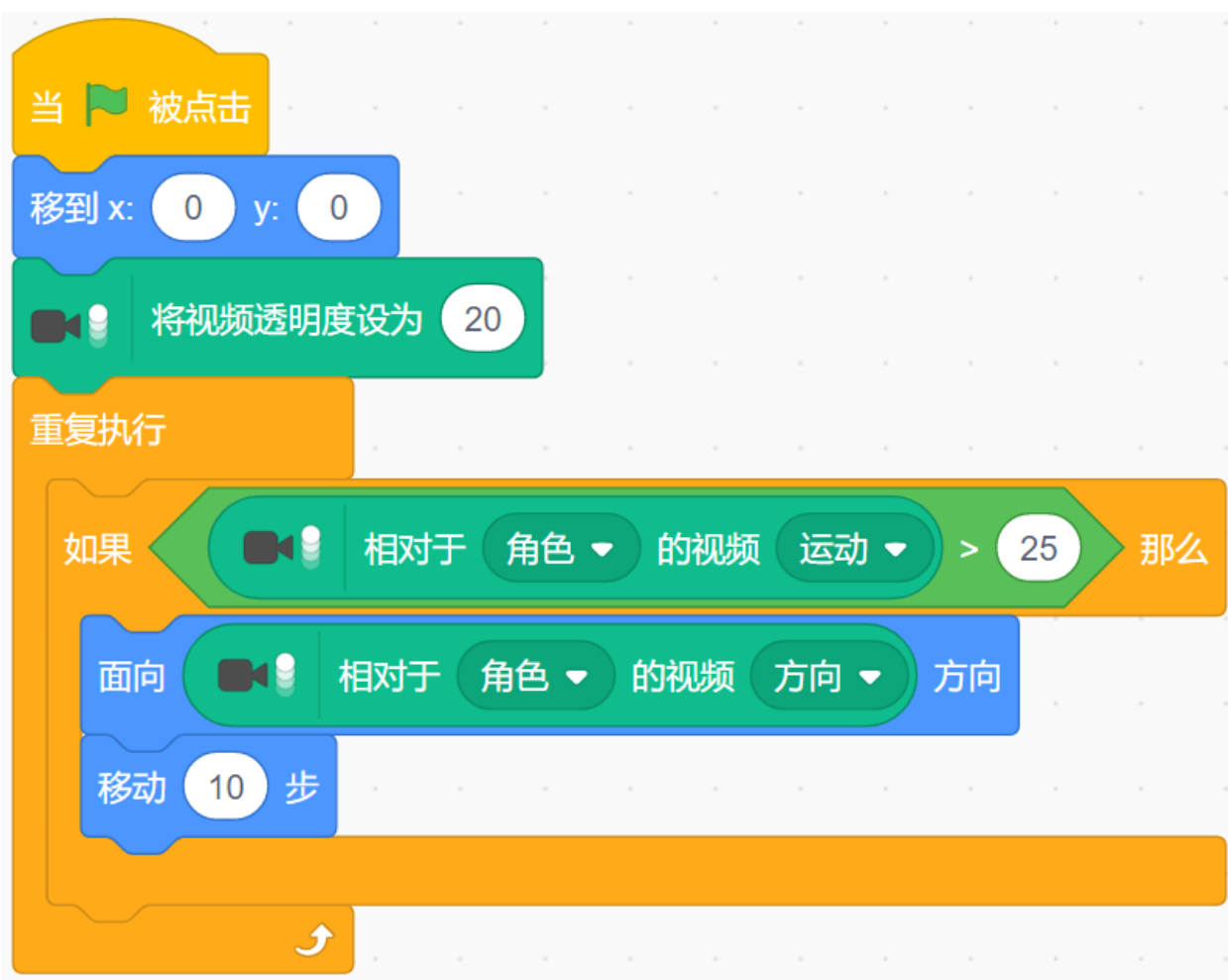


让猴子出现在舞台中央并初始化摄像头数据（透明度设置为 20）。





如果摄像头检测到物体在移动，让猴子朝着物体移动。



现在，点击舞台区域顶部的绿旗开始游戏。

让猴子吃香蕉，它很饿！尽量在白色背景上玩这个游戏，以防止其他物体的干扰。

## 挑战

我相信你很快就能掌握并实现这个游戏。接下来可以添加一些额外的内容来使你的游戏更丰富。

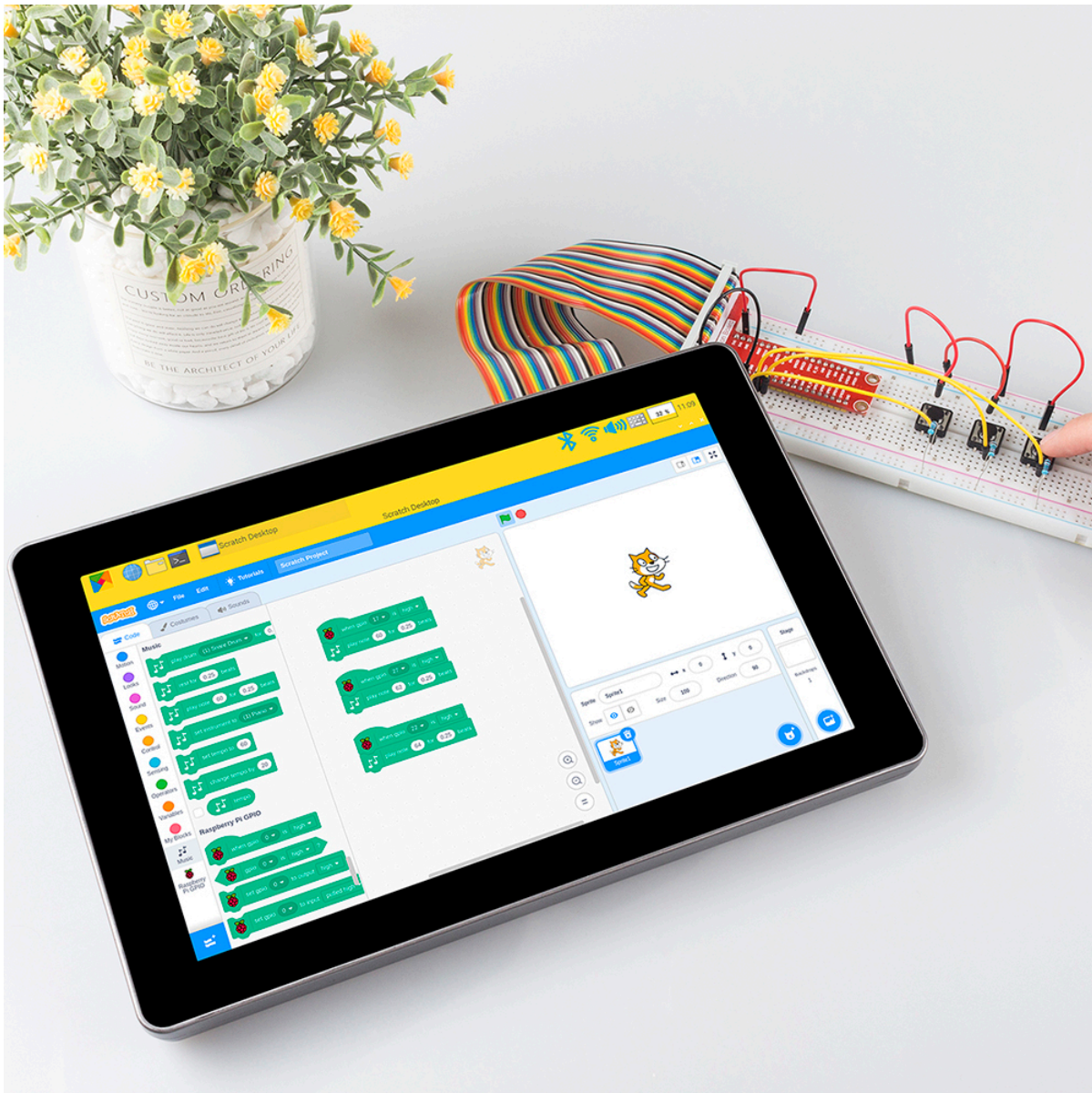
- 当猴子吃香蕉时，我们给分数加 1。30 秒内，看谁得分最高！
- 当猴子吃香蕉时，它会发出合适的声音效果。

## 6.7.11 制作乐器

### 描述

您可以将 RasPad 3 变成与朋友一起玩的游戏屏幕、显示天气和时间的智能闹钟、监控机器人动作的显示器以及许多其他功能。

本文将向您展示如何使用您的 RasPad 3 DIY 乐器。让我们一起来看看吧！



### 所需组件

- A RasPad 3
- 8G+ SD Card
- Scratch 3 (either online or offline)
- Micro SD 卡读卡器
- 40pin 排线
- T 型拓展板
- 面包板
- 按键
- 10k 电阻
- 一些跳线

### 你会学到

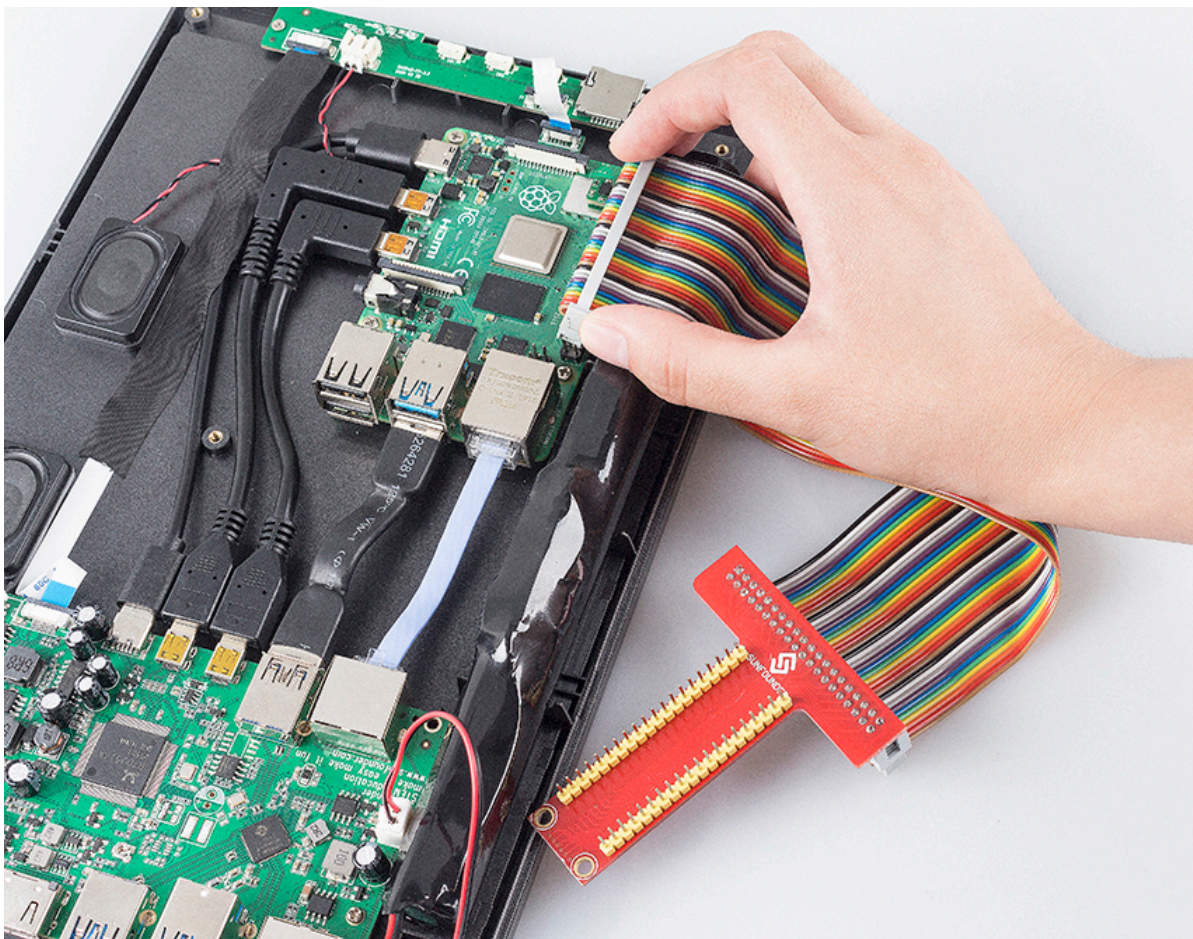
- 在 Scratch 上使用笔记功能。
- 从 GPIO 引脚输入按钮值。

### 课程指南

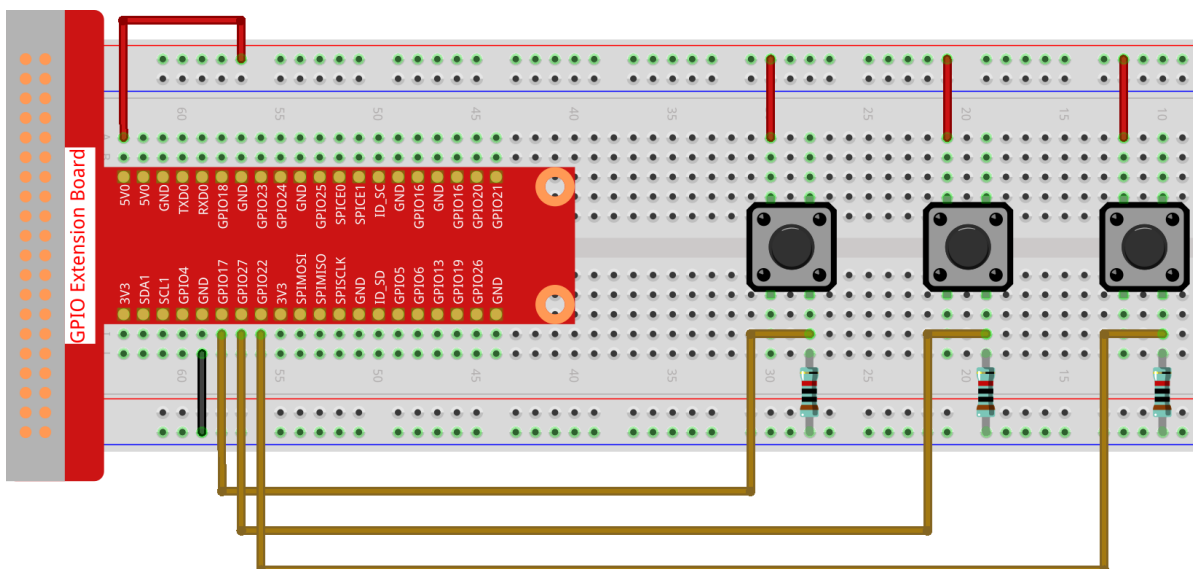
#### 搭建电路

首先连接 GPIO 扩展板，具体步骤请阅读[GPIO 扩展板](#)。





将 T 型 GPIO 扩展板插入面包板，搭建电路如下图。



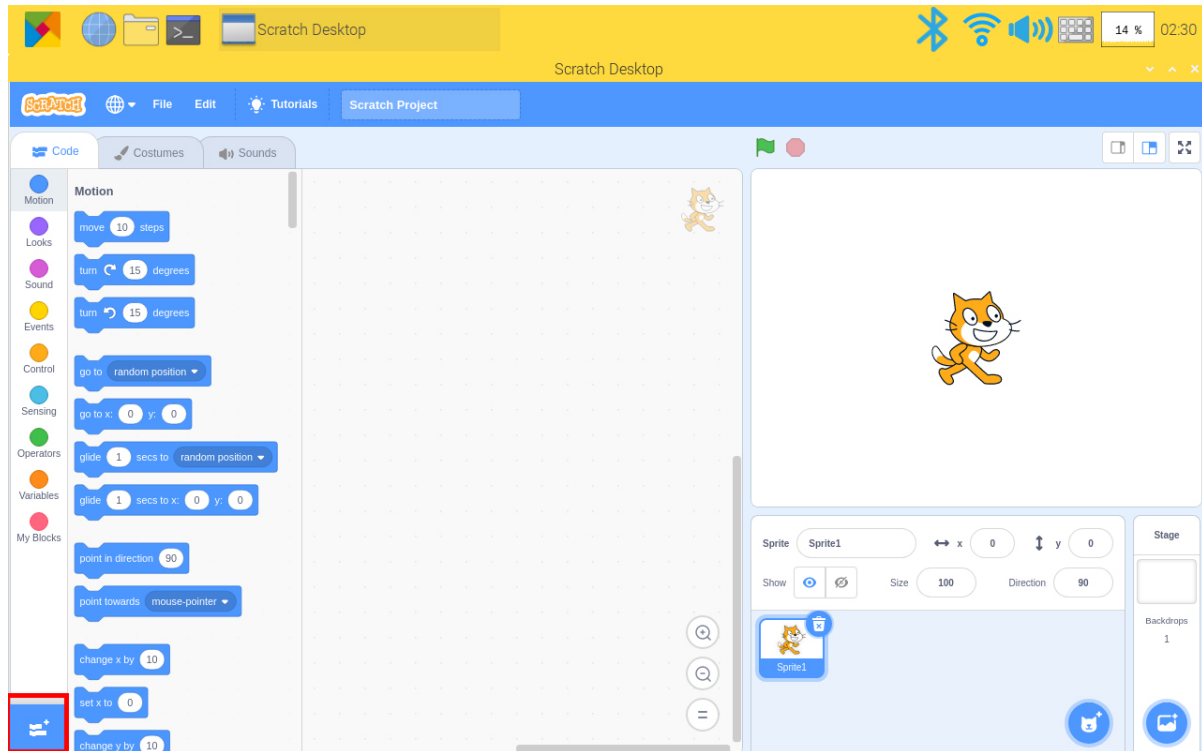
### 用 Scratch 3 编程

将 SD 卡插入 RasPad 3 的插槽，然后长按电源键启动 RasPad 3。

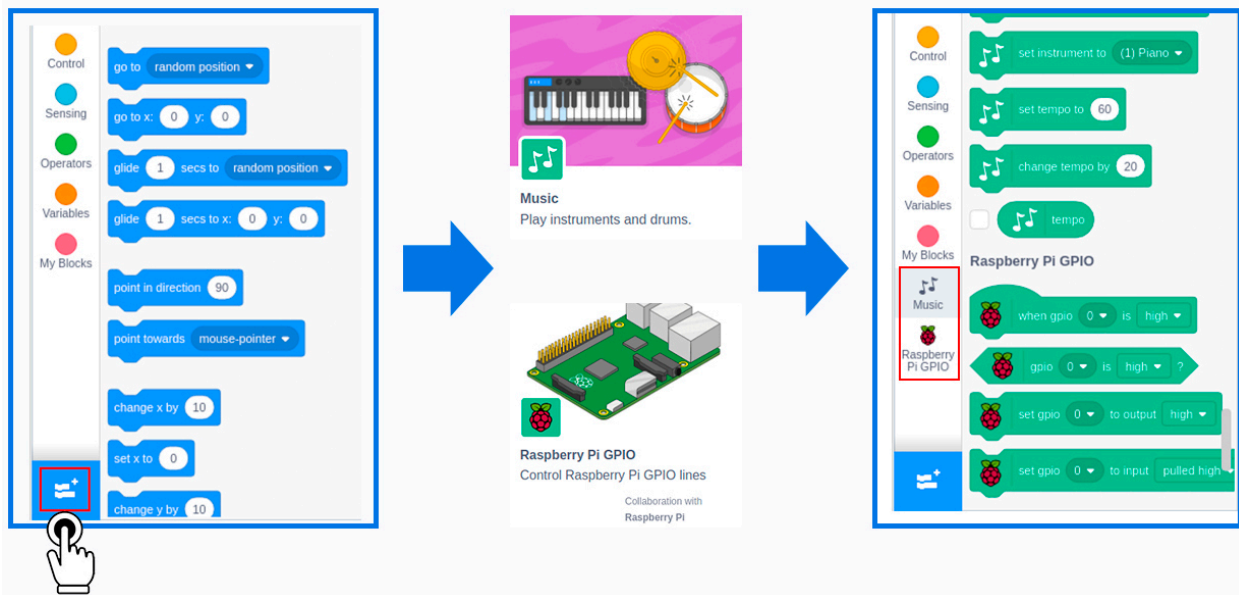


**警告：**卡槽位于后盖，采用卡扣式。因此在打开或关闭后盖之前，您需要取出 microSD 卡，以免损坏您的 microSD 卡和内部按键板。

Scratch 3 主页的左侧是一些排列整齐的代码块，您可以拖动它们来编程。在这个项目中，我们需要添加另外两个功能：音乐和树莓派 GPIO。演奏乐器和鼓的音乐功能，以及树莓派 GPIO 功能可用于控制树莓派的整个引脚。

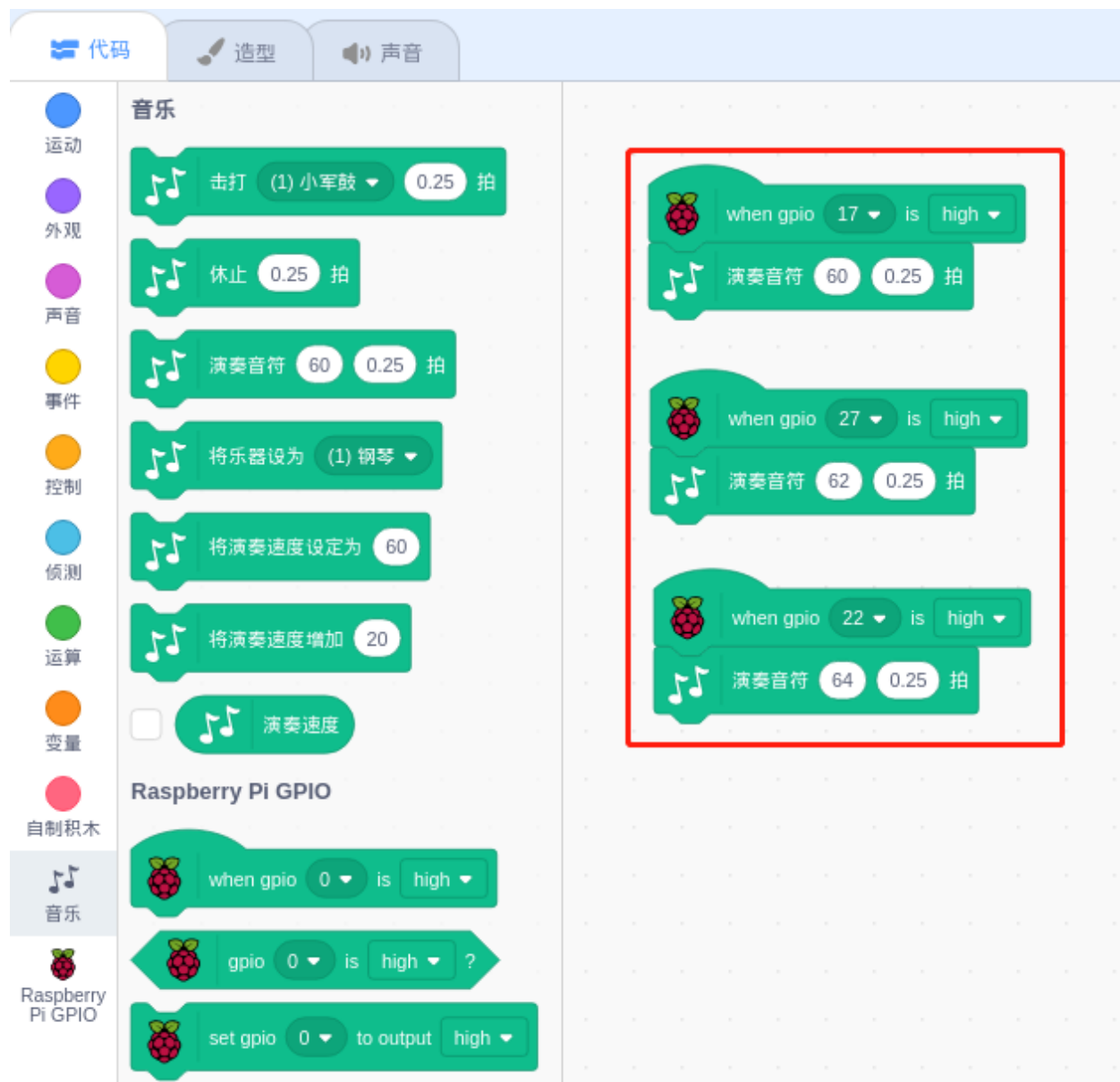


点击左下角的添加图标，选择音乐和树莓派 GPIO，在 Scratch 3 主页面左侧添加两个功能。

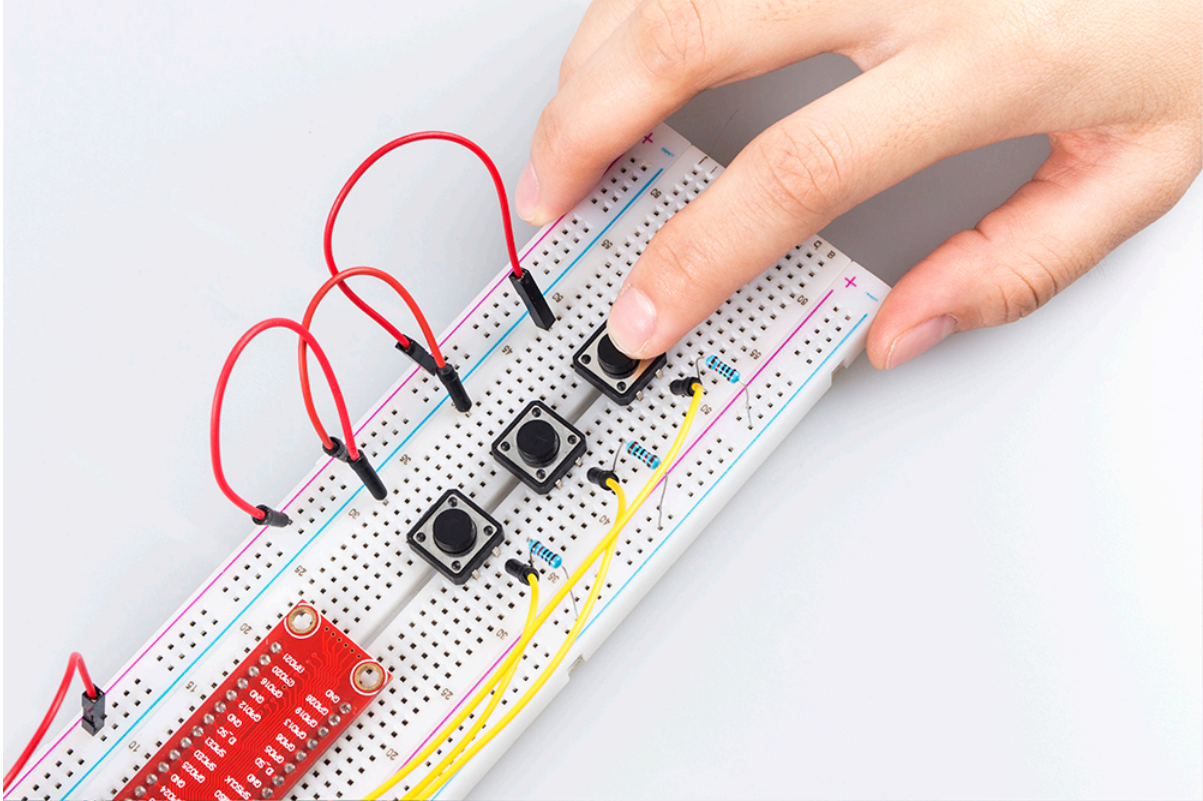


结束程序





按下面板上的这三个按钮，就会发出“哆、来、咪”的声音。



如果你不喜欢看文字，下面可以观看视频

在本文中，我们将介绍如何使用 RasPad 3 创建乐器。当然，你可以发挥你所有的想象力和灵感来升级你的乐器，例如你可以添加更多的按钮、音符和 LED 来制作一个很酷的八音盒。

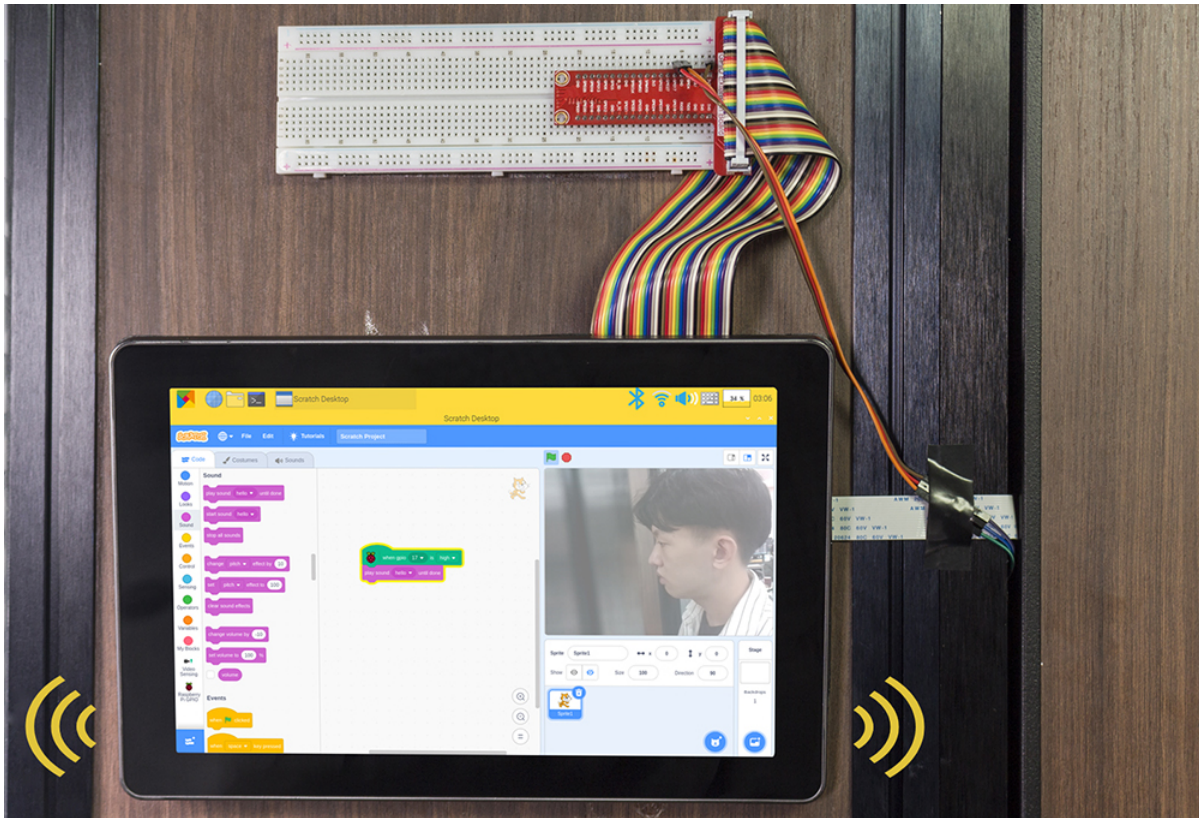
### 6.7.12 自定义 RasPad 监控设备

#### 描述

您可以将 RasPad3 变成与朋友一起玩的游戏屏幕、显示天气和时间的智能警报、监控机器人动作的显示器以及许多其他有趣的东西。

本文将向您展示如何在您的 RasPad 3 上自定义制作一个 RasPad 监控设备。让我们来看看如何制作吧！





## 所需组件

- A RasPad 3
- 8G+ SD 卡
- Scratch 3 (either online or offline)
- Micro SD 卡读卡器
- 40pin 排线
- T 型拓展板
- 面包板
- PIR 运动传感器模块
- 摄像头模块
- FFC 线
- 公-母跳线

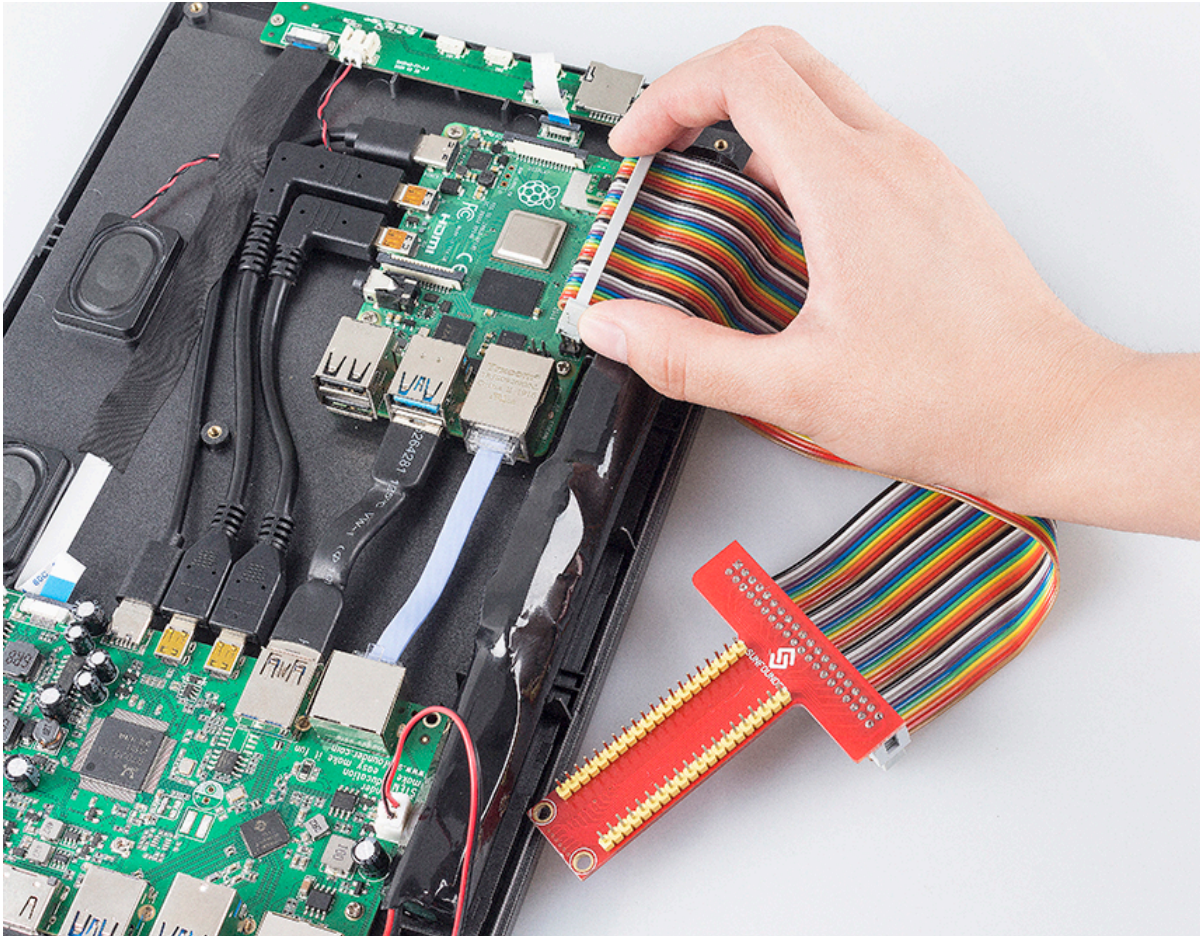
## 你会学到

- 在 Scratch 上使用树莓派扩展。
- 在 Scratch 上使用音频功能。
- 使用 PIR 模块。

## 课程指南

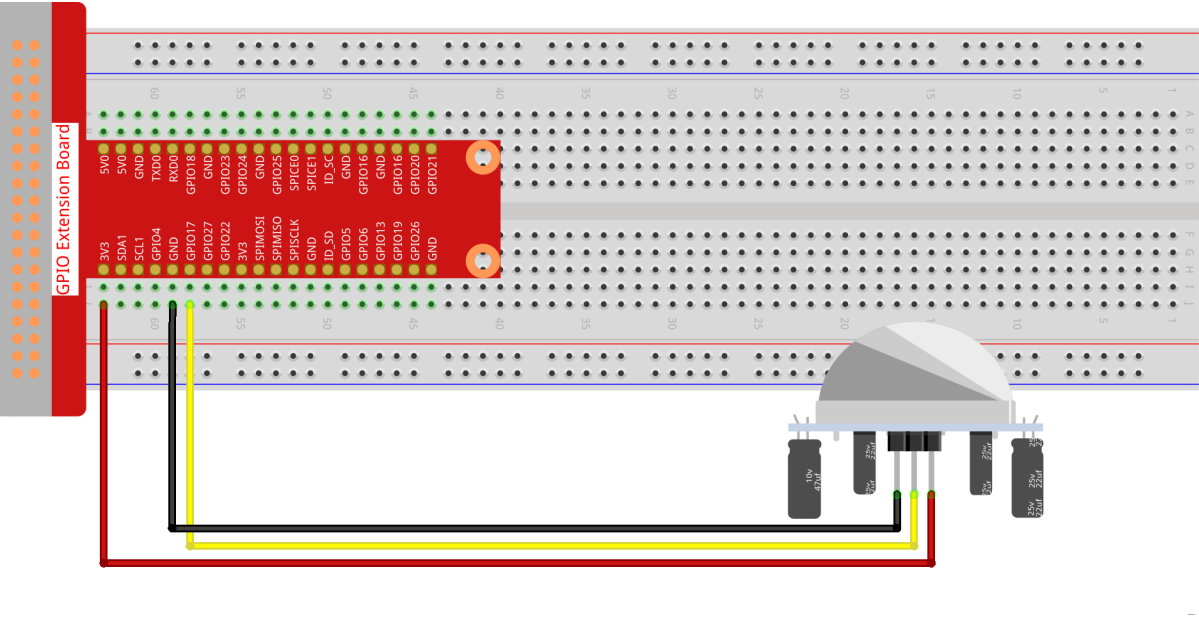
### 搭建电路

首先链接 GPIO 扩展板, 具体步骤请阅读[GPIO 扩展板](#)。

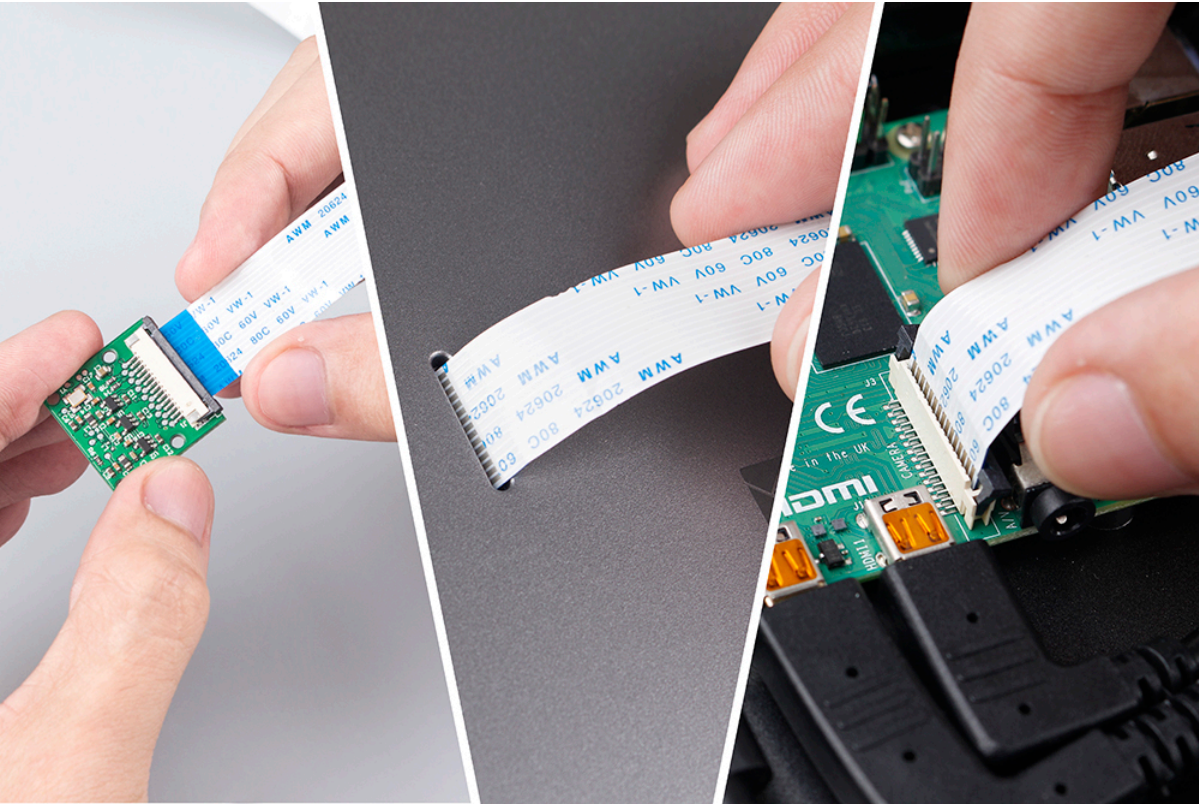


将 T 型 GPIO 扩展板插入面包板，搭建电路。



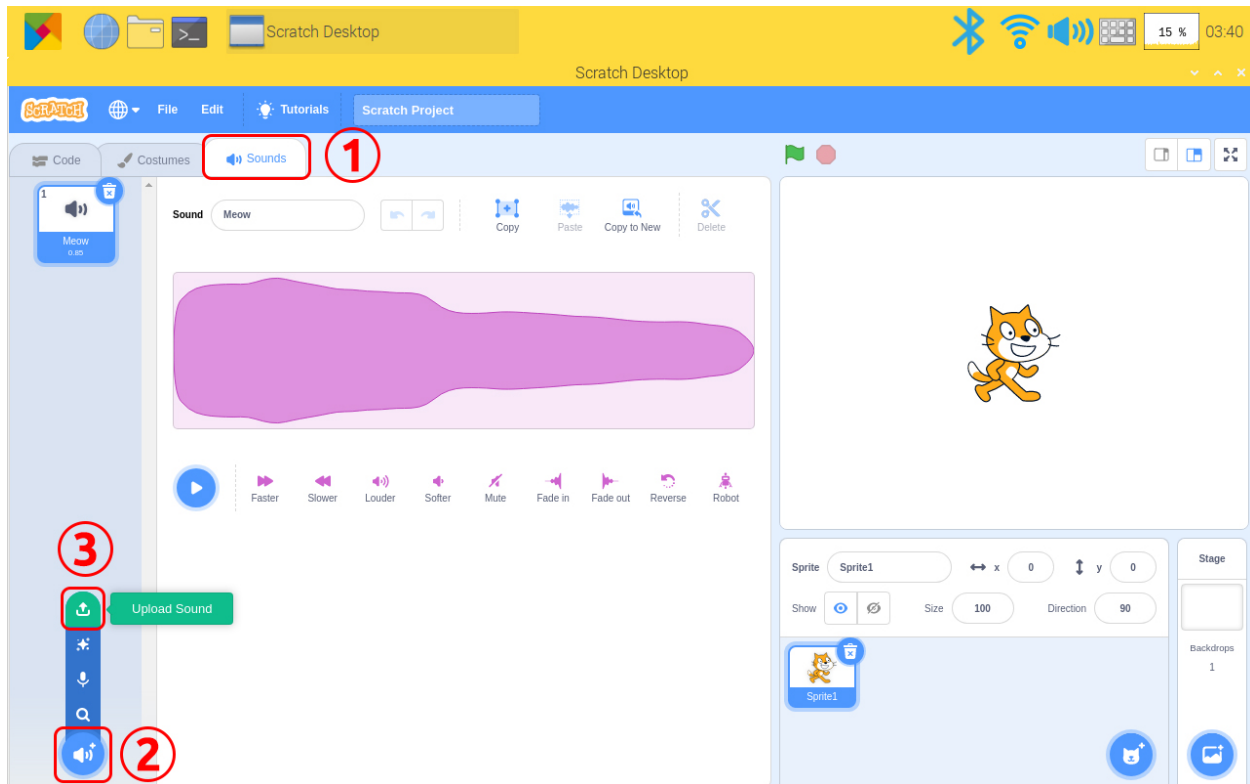


摄像头安装教程请参考安装摄像头.

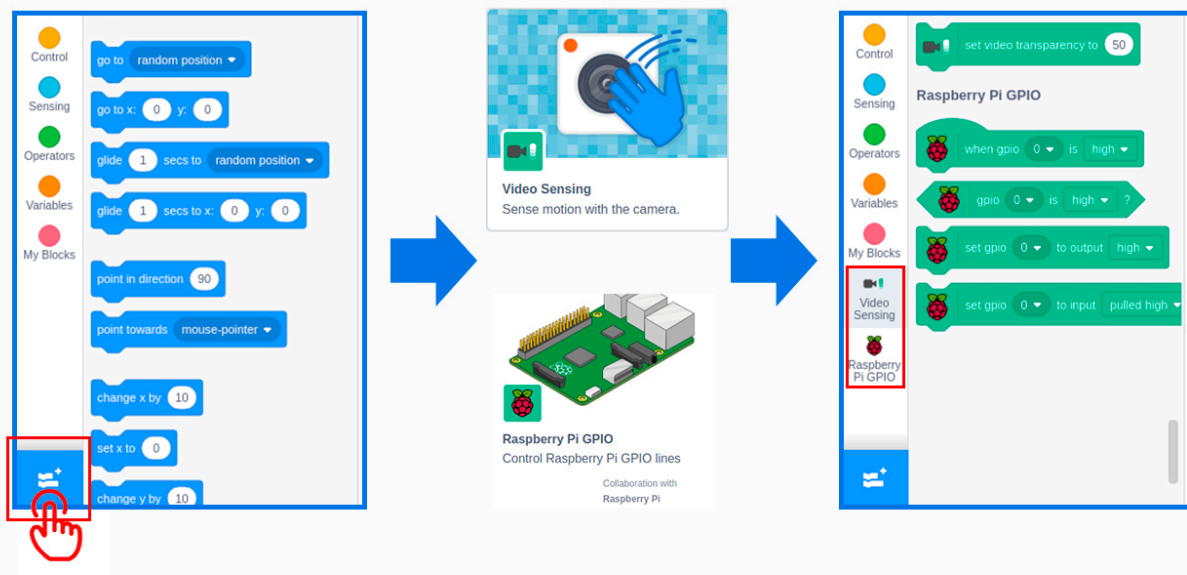


用 Scratch 3 编程

在这一步中，您将学习如何将准备好的音乐上传到 Scratch。点击左上角的“声音”选项，然后点击“扬声器”图标并选择“上传声音”图标上传准备好的音乐文件 - 你好，最后点击“打开”确认。



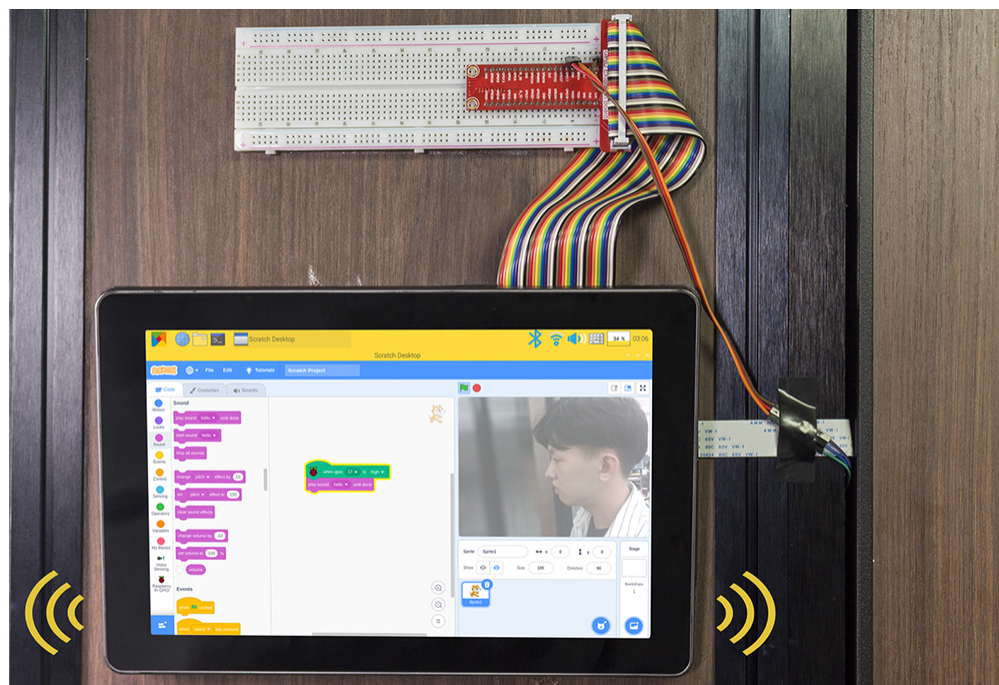
点击左下角的添加图标，选择“Video Sensing”和“树莓派 GPIO”添加两个功能。



回到主页面，将树莓派 GPIO 功能中的“when gpio 0 is high”和“play sound (hello) until done”拖到编程区。



将 pir 模块和摄像头贴在门外的墙上，并将 RasPad 贴在门内或任何地方的墙上。当门打开时，您会听到音乐，然后看到谁在那里。



如果你不喜欢阅读文字，可以观看视频。

在本文中，我们将介绍如何使用 RasPad 3 自定义制作一个 RasPad 监控设备。当然，您还可以添加一个继电器和一个步进电机，以便在门口检测到入时可以自动打开门。

## 6.8 在 RasPad 上安装其他系统

除了我们之前使用的 树莓派 OS 和 RetroPie 操作系统外，RasPad 3 还支持其他操作系统。

这里只提供学习参考的链接。您可以按照链接上面的操作步骤尝试安装。

**注解：**我们只在 RasPad 上测试了树莓派 OS、Ubuntu MATE、RetroPie。如果您安装了其他系统，但其中一些可能存在一些错误，导致某些功能无法使用，我们只能建议您咨询该系统的开发人员。

23 种可在您的树莓派上运行的操作系统。.



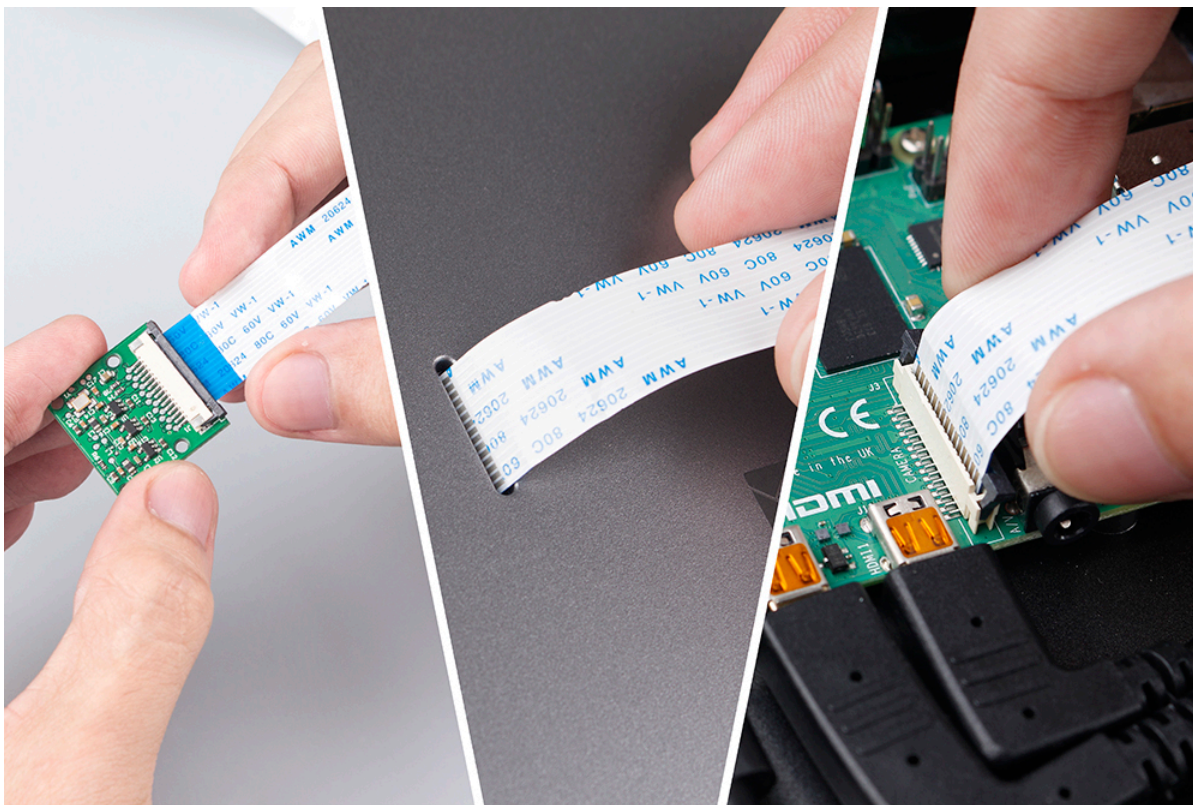
对于新的 RasPad 用户，以下部分是执行某些项目需要使用的 basic 操作。请仔细阅读这些部分。

### 7.1 安装摄像头

打开 RasPad 的后盖。将 FFC 电缆连接到摄像头模块的 CSI 接口。小心地将 CSI 接口电缆穿过 RasPad 后盖上的插槽。连接树莓派的 CSI 接口。然后小心地关闭 RasPad 的后盖。

**警告：** SD 卡插槽采用卡扣式连接到后盖。在拆卸或更换后盖之前，请先取出 micro-SD 卡，以免损坏 micro-SD 卡和 RasPad 的内部按键板。

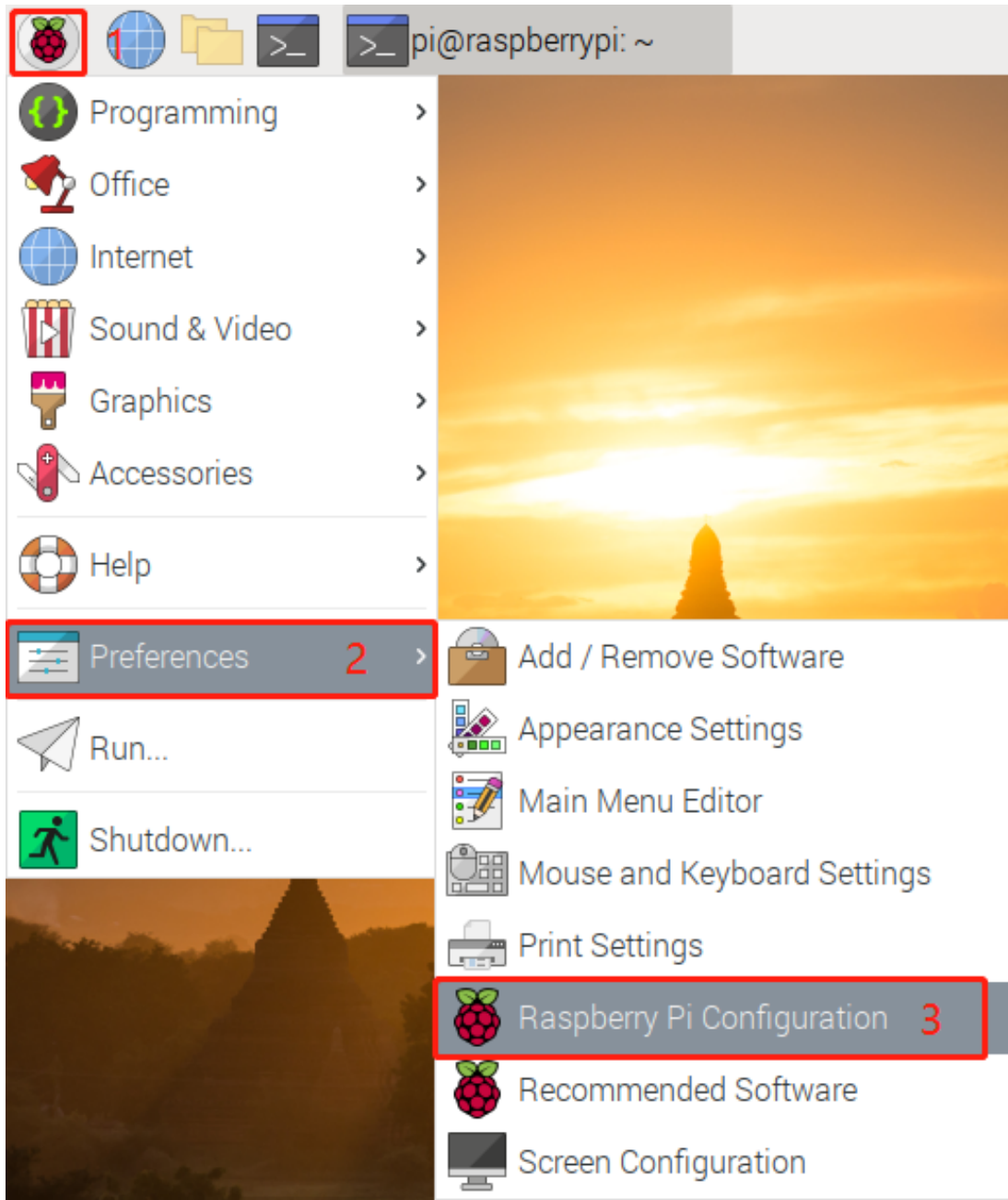




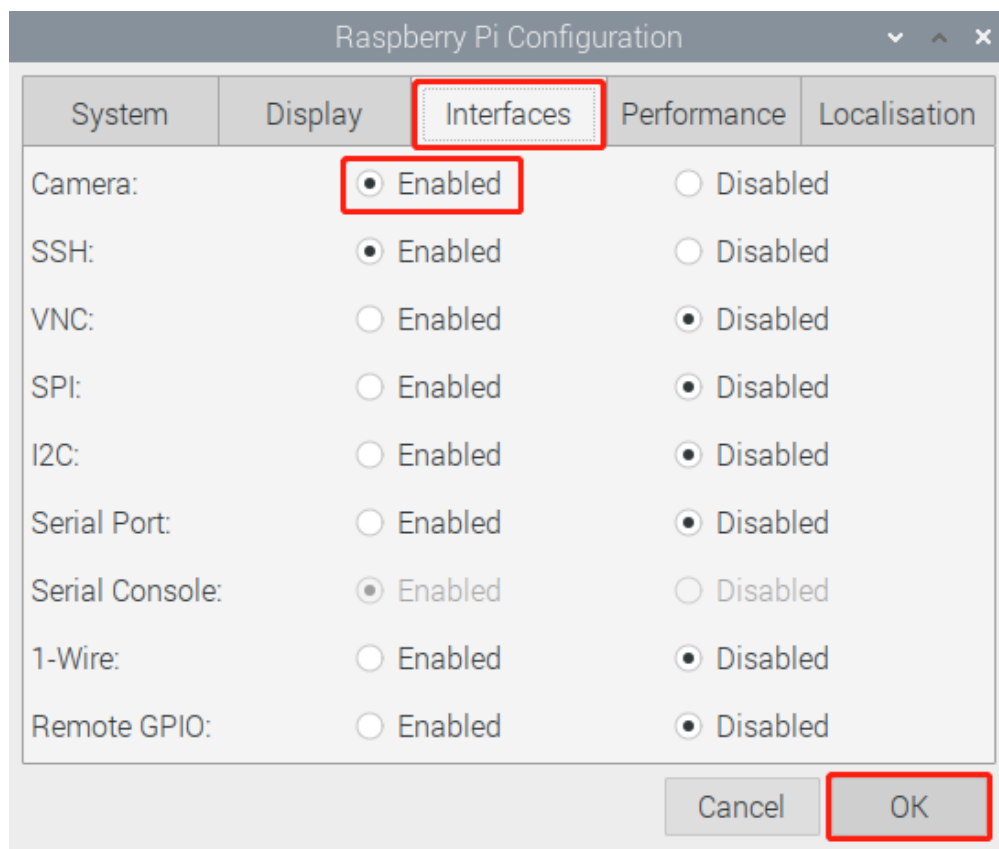
将带有树莓派操作系统镜像的 micro-SD 卡插入插槽，长按电源键启动 RasPad。



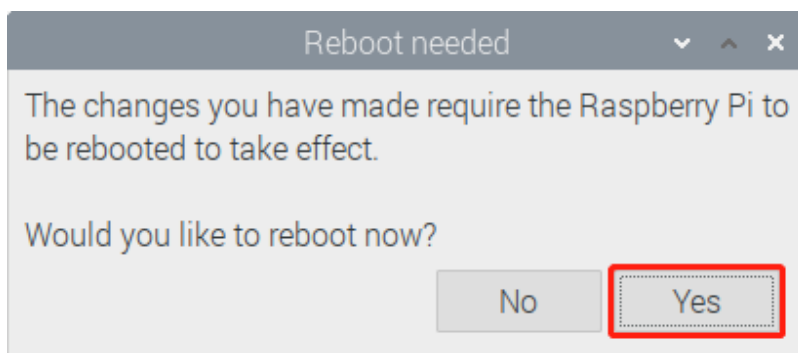
打开 树莓派配置。



在 **Interfaces** 选项中, **Enable** 使用摄像头, 然后单击 **OK**。



在弹出的提示框中，选择立即重启。



重启完成后，使用如下命令行查看摄像头是否可用。

如果出现摄像头画面，则表示摄像头安装成功。否则需要拔下 FFC 电缆线并重新插入。

```
raspivid -o vid.h264
```

## 7.2 GPIO 扩展板

在开始项目之前，您首先需要更多地了解树莓派的引脚，这是搭建电路的关键。

树莓派的引脚有三种命名方式。它们是 `wiringPi`、`BCM` 和 `Board`。40 针 GPIO 扩展板使用 `BCM` 命名方式。

下表显示了三种命名方式以及 GPIO 扩展板上每个引脚的固有名称。

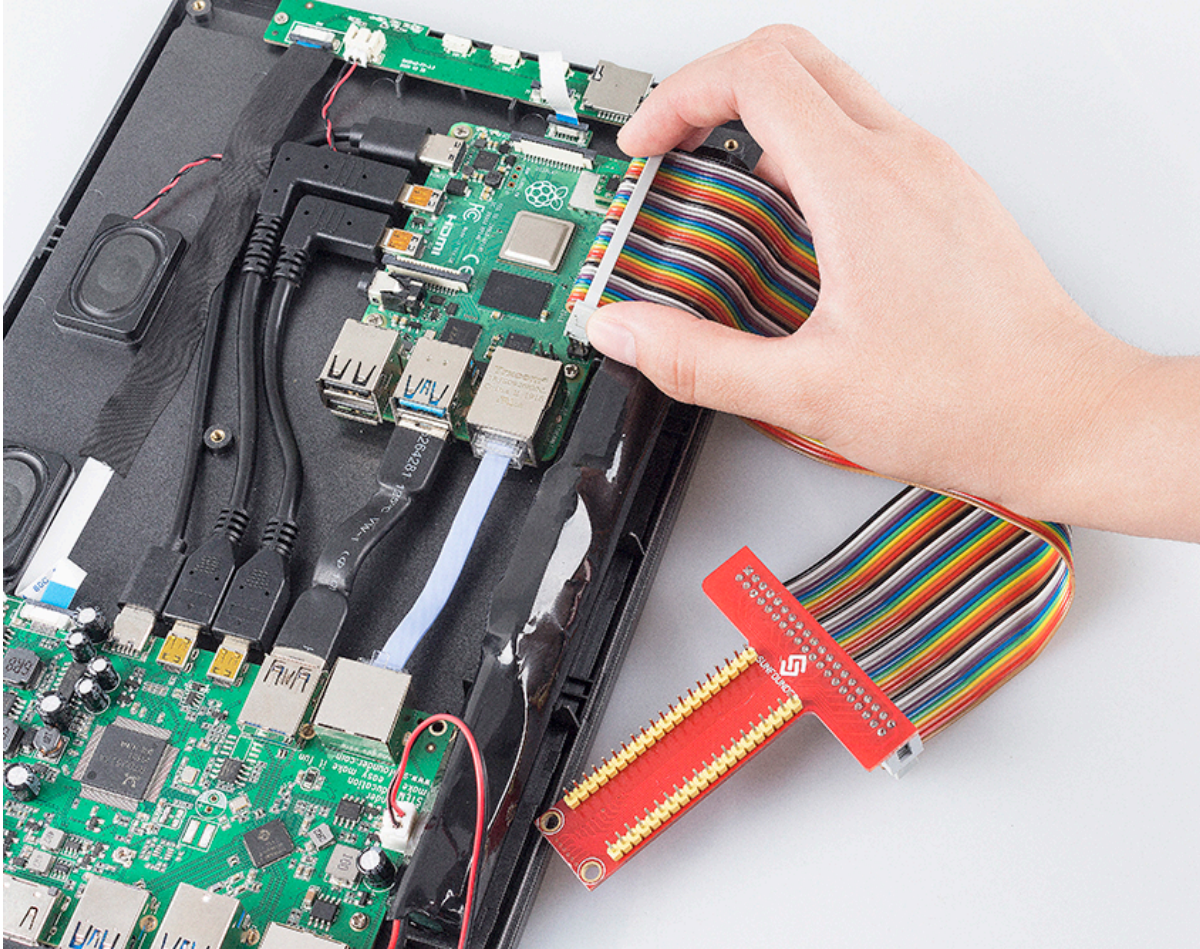
例如 `GPIO17`，其 `Board` 命名方式为 11，`wiringPi` 命名方式为 0，其固有命名方式为 `GPIO0`。

Name	WiringPi	Board	BCM		Board	WiringPi	Name
		GPIO Extention Board					
3.3V	3V3	1	3V3	5.0V	2	5.0V	5V
SDA	8	3	SDA	5.0V	4	5.0V	5V
SCL	9	5	SCL	GND	6	GND	0V
GPIO7	7	7	GPIO4	TXD	8	15	TXD
0V	GND	9	GND	RXD	10	16	RXD
GPIO0	0	11	GPIO17	GPIO18	12	1	GPIO1
GPIO2	2	13	GPIO27	GND	14	GND	0V
GPIO3	3	15	GPIO22	GPIO23	16	4	GPIO4
3.3V	3.3V	17	3.3V	GPIO24	18	5	GPIO5
MOSI	12	19	MOSI	GND	20	GND	0V
MISO	13	21	MISO	GPIO25	22	6	GPIO6
SCLK	14	23	SCLK	CE0	24	10	CE0
0V	GND	25	GND	CE1	26	11	CE1
IN_SDA	30	27	EED	EEC	28	31	ID_SCL
GPIO21	21	29	GPIO5	GND	30	GND	0V
GPIO22	22	31	GPIO6	GPIO12	32	26	GPIO26
GPIO23	23	33	GPIO13	GND	34	GND	0V
GPIO24	24	35	GPIO19	GPIO16	36	27	GPIO27
GPIO25	25	37	GPIO26	GPIO20	38	28	GPIO28
0V	GND	39	GND	GPIO21	40	29	GPIO29

打开 RasPad 的后盖，将 40 针带状电缆线插入树莓派。将带状电缆线穿过 RasPad 底座上的可用插槽，并将 40 针带状电缆线的另一端连接到 GPIO 扩展板。然后更换 RasPad 的后盖。

**警告：** SD 卡插槽采用卡扣式连接到后盖。在拆卸或更换后盖之前，请先取出 micro-SD 卡，以免损坏 micro-SD 卡和 RasPad 的内部按键板。





## 7.3 获取树莓派的 IP 地址

### 7.3.1 有屏幕

如果你有一个屏幕，就可以很容易得到树莓派的 IP 地址。

#### 所需组件

- 任何树莓派
- 1 \* 电源适配器
- 1 \* 微型 SD 卡
- 1 \* 屏幕电源适配器
- 1 \* HDMI 线
- 1 \* 屏幕
- 1 \* 鼠标
- 1 \* 键盘

1. 将带有树莓派 OS 镜像的微型 SD 卡插入树莓派底部的微型 SD 卡插槽。

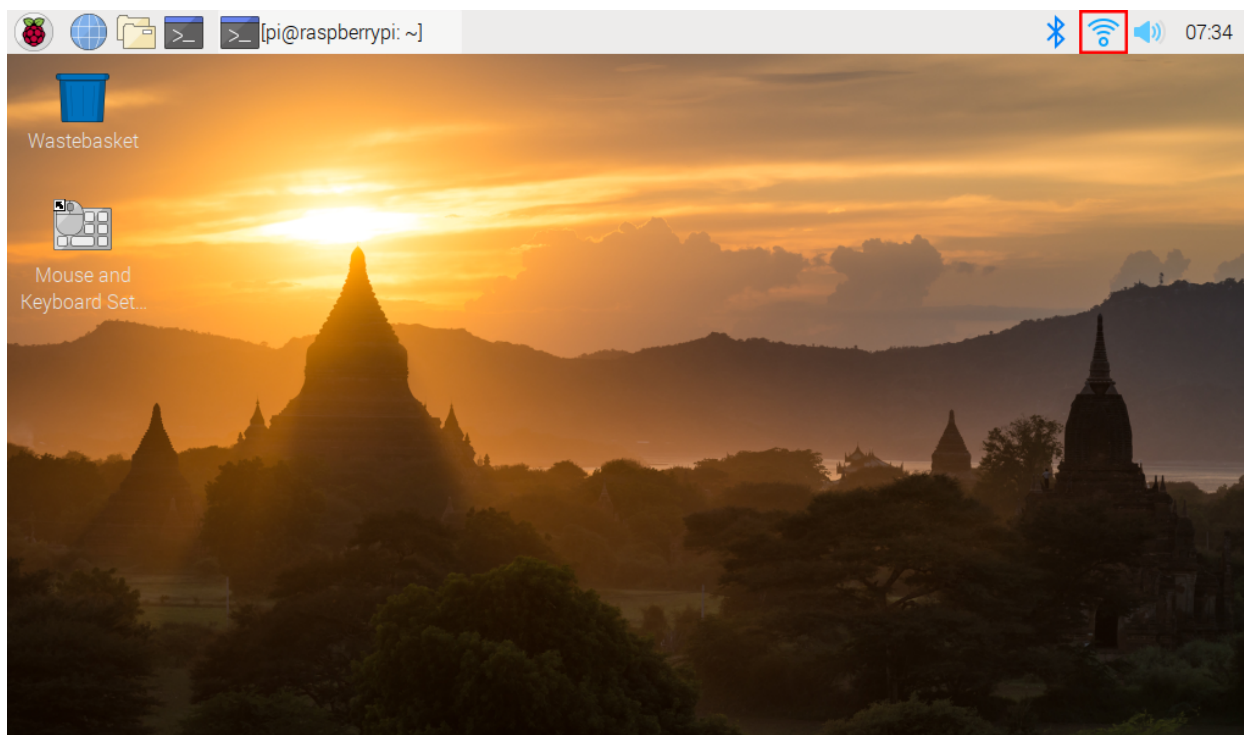
2. 插入鼠标和键盘。
3. 将屏幕连接到树莓派的 HDMI 端口。确保屏幕已插入壁式插座并已打开。

---

**注解：**对于树莓派 4，将屏幕连接到最靠近电源插座的 HDMI0 端口。

---

4. 使用电源适配器为树莓派供电。几秒钟后，将显示树莓派 OS 桌面。
5. 将光标悬停在 WiFi 图标上，将显示树莓派的 IP 地址。



### 7.3.2 无屏幕

树莓派连接 WIFI 后，我们需要获取它的 IP 地址。知道 IP 地址的方法有很多种，下面列出了其中的两种。

#### 1. 通过路由器检查

在路由器的管理界面上检查分配给树莓派的地址。

树莓派操作系统的默认主机名是 raspberrypi。如果您使用的是 ArchLinuxARM 系统，请查找 alarmpi。

#### 2. 网段扫描

有一些网络扫描的应用程序可用于查找树莓派的 IP 地址，例如 Advanced IP Scanner。

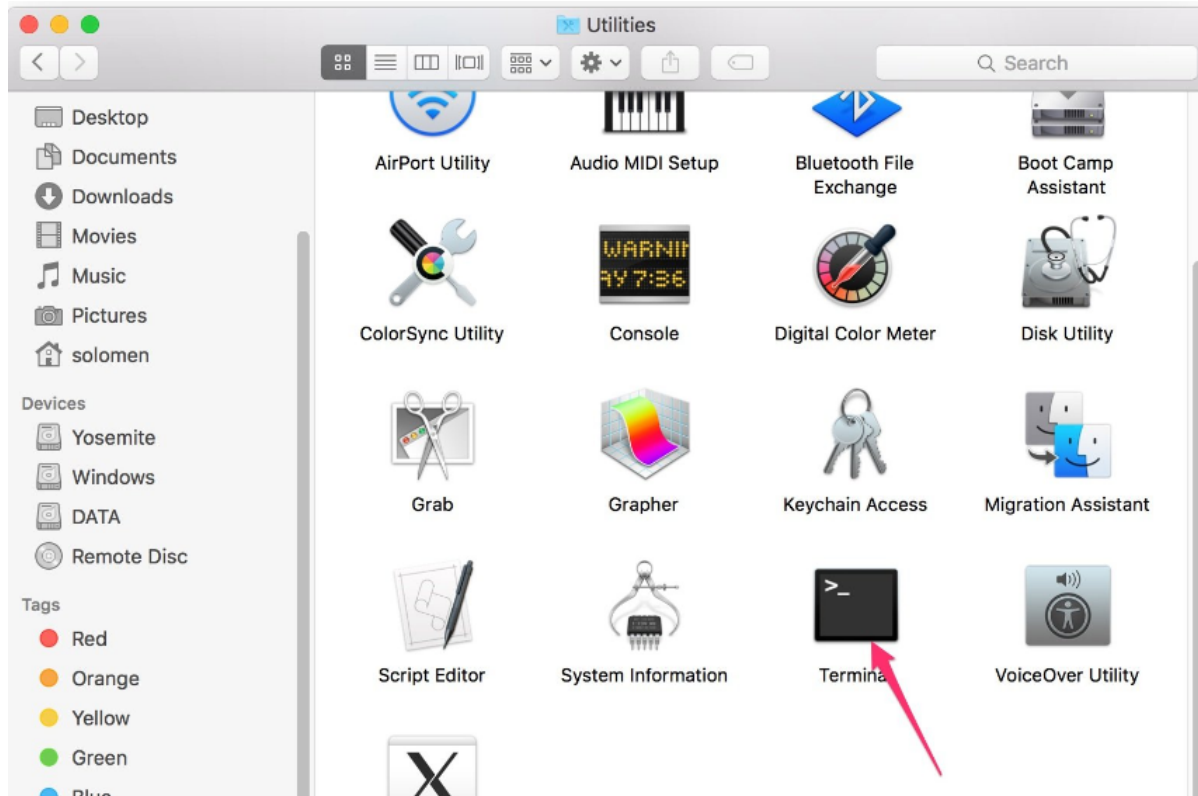
扫描设置的 IP 范围，将显示所有已连接设备的名称。树莓派操作系统的默认主机名是 raspberrypi。

## 7.4 用 SSH 远程控制

Bash 是 Linux 的标准默认 shell，可以在树莓派上使用 SSH 打开。Shell 本身是一个用 C 编程语言编写的程序，充当连接客户和 Unix/Linux 系统的桥梁。

对于 Linux 或/Mac OS X 用户

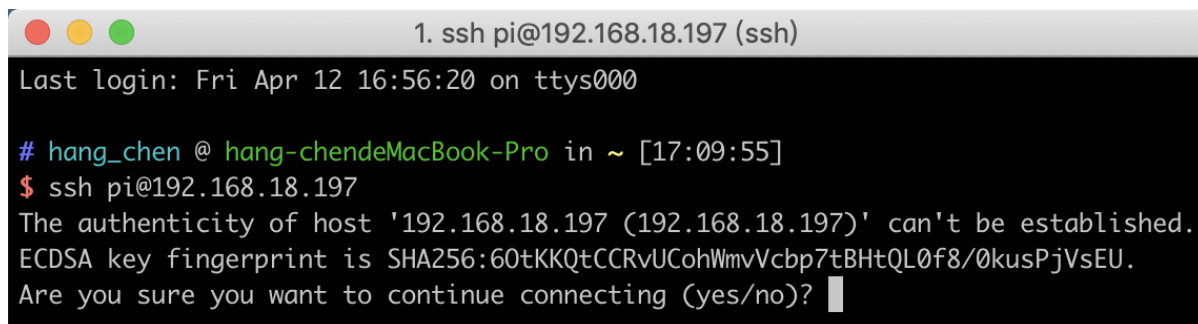
依次点击 **Applications -> Utilities**, 找到 **Terminal**, 然后打开它。



在 `ssh pi@ip_address` 这里可以输入命令，pi 表示用户名 and ip\_address 表示树莓派的 ip 地址。例如：

```
ssh pi@192.168.18.197
```

根据提示，输入小写的 yes。

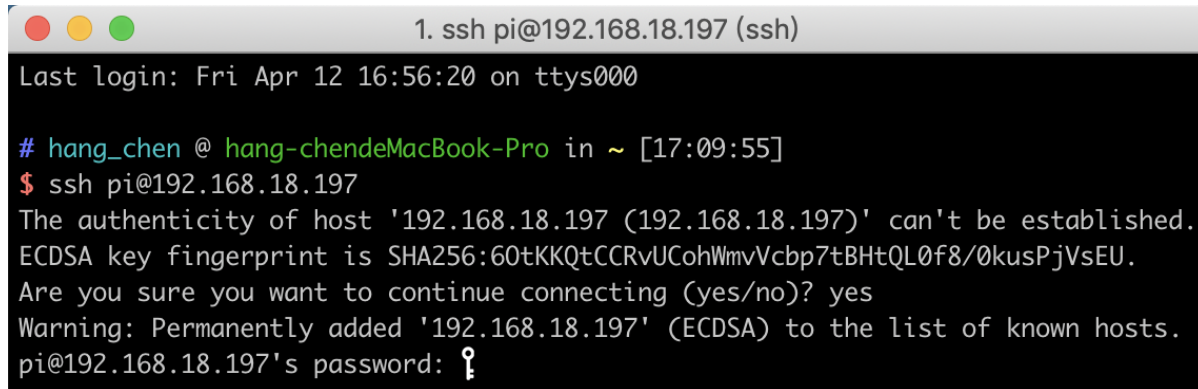


默认密码时 raspberry。



**注解：** 输入密码时不会显示字符。确保你输入的是正确的密码。

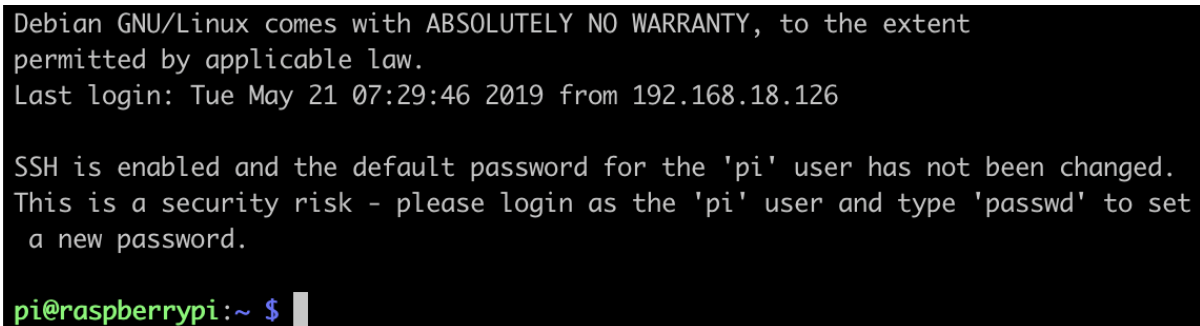
---

A terminal window titled "1. ssh pi@192.168.18.197 (ssh)" with standard macOS window controls. The terminal shows the output of an SSH command from a Mac. It displays the last login for the pi user, the user's prompt, the command to connect to 192.168.18.197, a warning about the host's authenticity, a confirmation to continue, a warning about adding the host to the known hosts list, and the password prompt.

```
1. ssh pi@192.168.18.197 (ssh)
Last login: Fri Apr 12 16:56:20 on ttys000

# hang_chen @ hang-chendeMacBook-Pro in ~ [17:09:55]
$ ssh pi@192.168.18.197
The authenticity of host '192.168.18.197 (192.168.18.197)' can't be established.
ECDSA key fingerprint is SHA256:60tKKQtCCRvUCohWmvVcbp7tBHtQL0f8/0kusPjVsEU.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.18.197' (ECDSA) to the list of known hosts.
pi@192.168.18.197's password: 
```

到这里，我们就已经连接上了树莓派。

A terminal window showing the login message on a Raspberry Pi. It displays the Debian GNU/Linux warranty disclaimer, the last login time, a message about SSH being enabled and the default password for the pi user, and the user's prompt.

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue May 21 07:29:46 2019 from 192.168.18.126

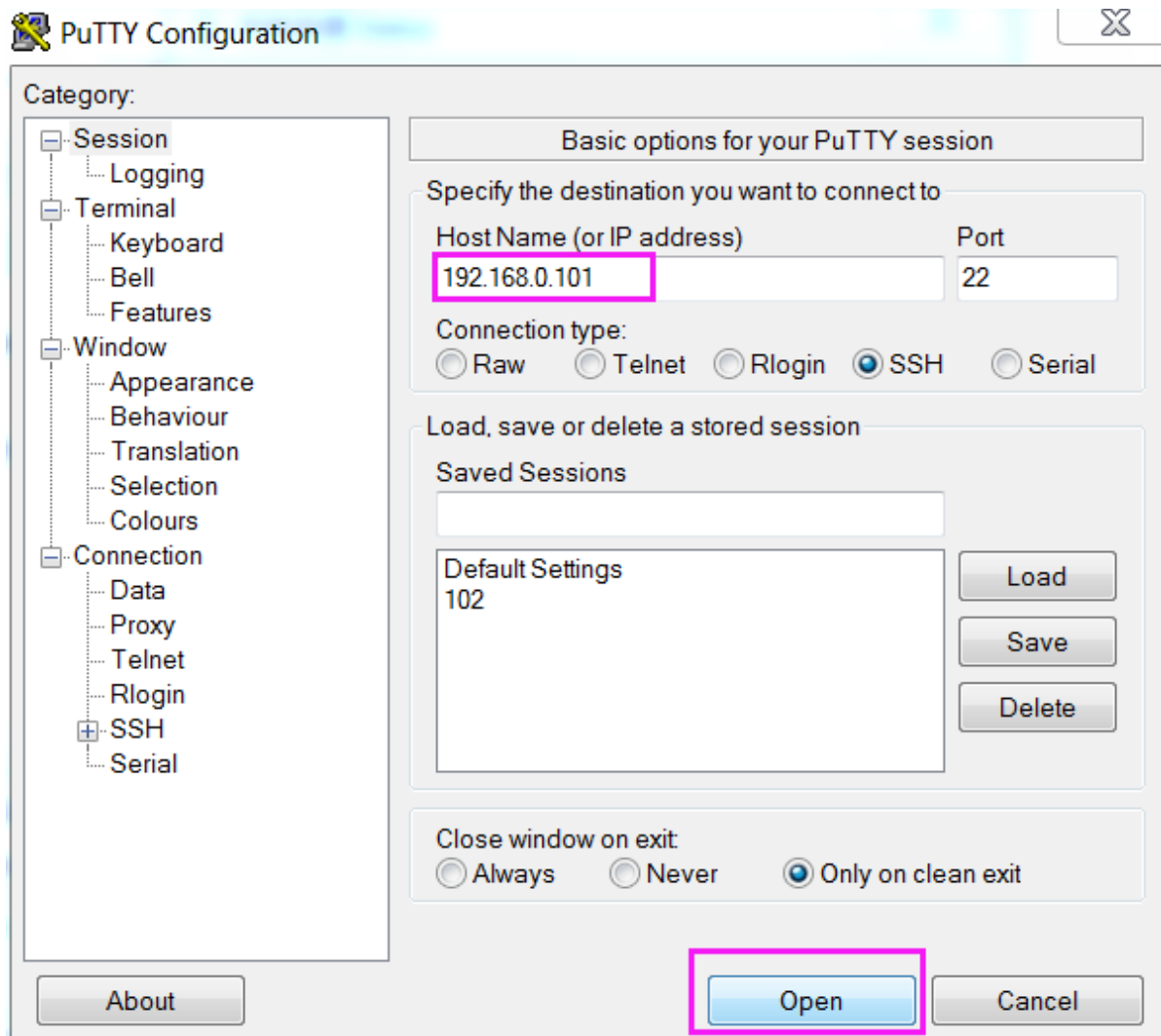
SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

pi@raspberrypi:~ $ 
```

### 对于 Windows 用户

Windows 用户可以通过 PuTTY 应用程序使用 SSH。

下载 PuTTY，打开应用，点击树状结构左侧的 Session。在 Host Name 下的文本框中输入树莓派的 IP 地址，在 Port 下输入 22。




单击打开。

**注解：**第一次用 IP 地址登录树莓派时，会有安全提示。单击继续。

当 PuTTY 窗口提示 **login as:** 的时候，输入用户名 `pi` 以及默认的初始密码 `raspberrypi`。

**注解：**输入密码时不会显示字符。请确保输入的是正确的密码。



```
pi@raspberrypi: ~  
login as: pi  
pi@192.168.0.234's password: raspberrypi  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Tue Feb 21 02:54:55 2017  
pi@raspberrypi:~ $
```

到这里，我们就已经连接上了树莓派。

#### 8.1 修复树莓派启动问题

1. 检查电源指示灯是否亮，如果不亮，则需要给 RasPad 充电或尝试不同的电源。最好使用 SunFounder 提供的电源。
2. 您可能正在启动但没有获得视频输出。首先，检查您的电缆是否正确连接。将它们全部取出并重新连接。如果您有 HDMI 电缆，请尝试使用不同的 HDMI 电缆。
3. 取出 microSD 卡并使用 Raspberry Pi Imager 将新安装的树莓派 OS 刷入其中（详见[安装树莓派镜像](#)）。
4. 尝试不同的 microSD 卡。

#### 8.2 无法检测到 Micro SD 卡？

1、Micro SD 卡板上的 FFC 是否连接正确，参考[组装说明](#)。2、检查 SD 卡或 Micro SD 卡板上的金手指（传输信号）是否变黑。如果是，说明金手指已经氧化，需要用橡皮擦擦掉氧化层。3、检查 SD 卡是否损坏。如果在读取 SD 卡时拔掉 SD 卡的插头，导致卡的供电中断，可能会损坏卡。

#### 8.3 屏幕不能旋转？

原因是内置的加速度模块没有经过校准。

请参考[安装旋转屏幕服务](#)。

## 8.4 关于睡眠问题

- 如果您的 RasPad 屏幕亮起但黑屏，这是树莓派进入睡眠模式。您可以通过点击触摸屏或按外部键盘或鼠标来唤醒。
- 如果 RasPad 屏幕不亮或您短按电源按钮，这是 RasPad 进入睡眠模式。您可以通过短按电源按钮唤醒。

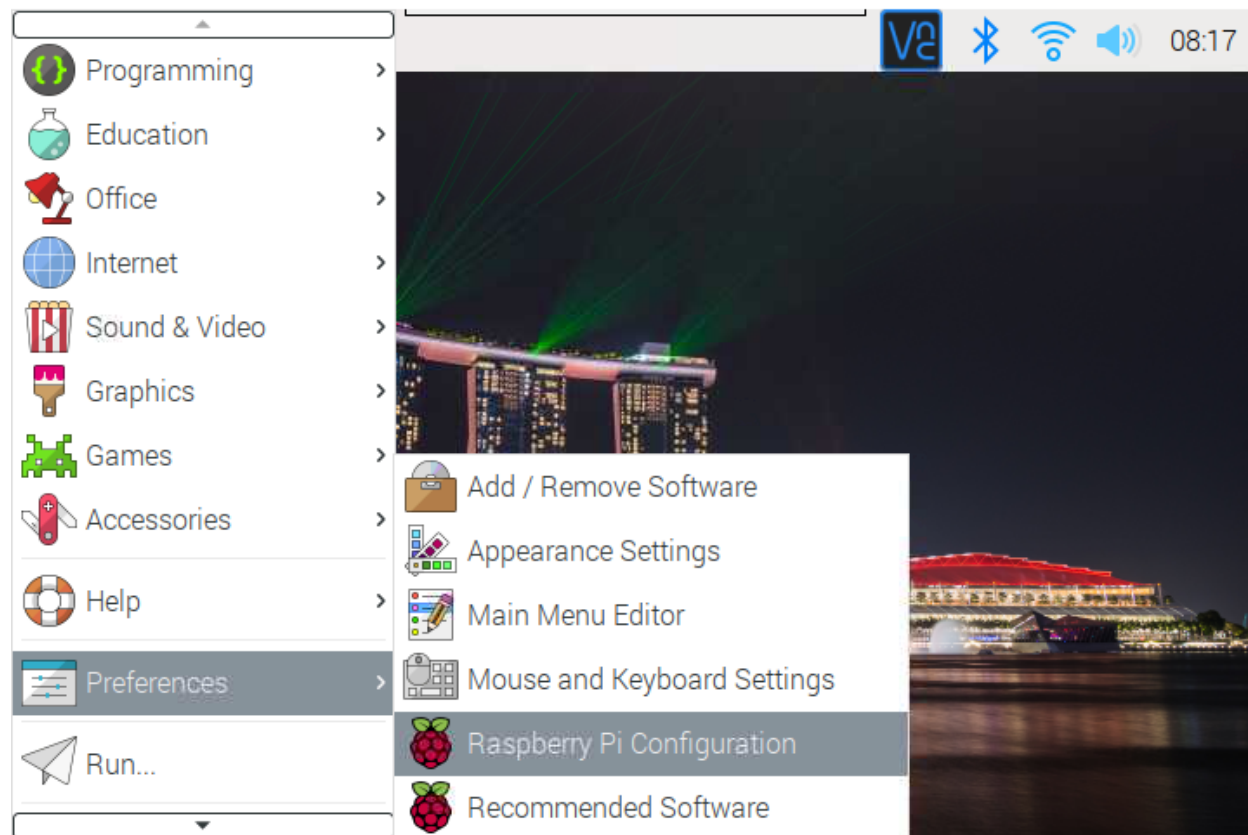
## 8.5 关于关机问题

- 当您在树莓派桌面点击关机按钮或在命令行关闭树莓派时，树莓派会关机，但 RasPad 没有关机，过一段时间会自动进入睡眠状态，因为它不会检测不到 HDMI 信号。您需要长按 RasPad 的电源按钮才能关机。
- 如果直接长按 RasPad 的电源键，树莓派和 RasPad 会同时关机。但需要注意的是，这是直接关闭树莓派的电源，可能会损坏 Micro SD 卡。建议先关闭树莓派，再关闭 RasPad。

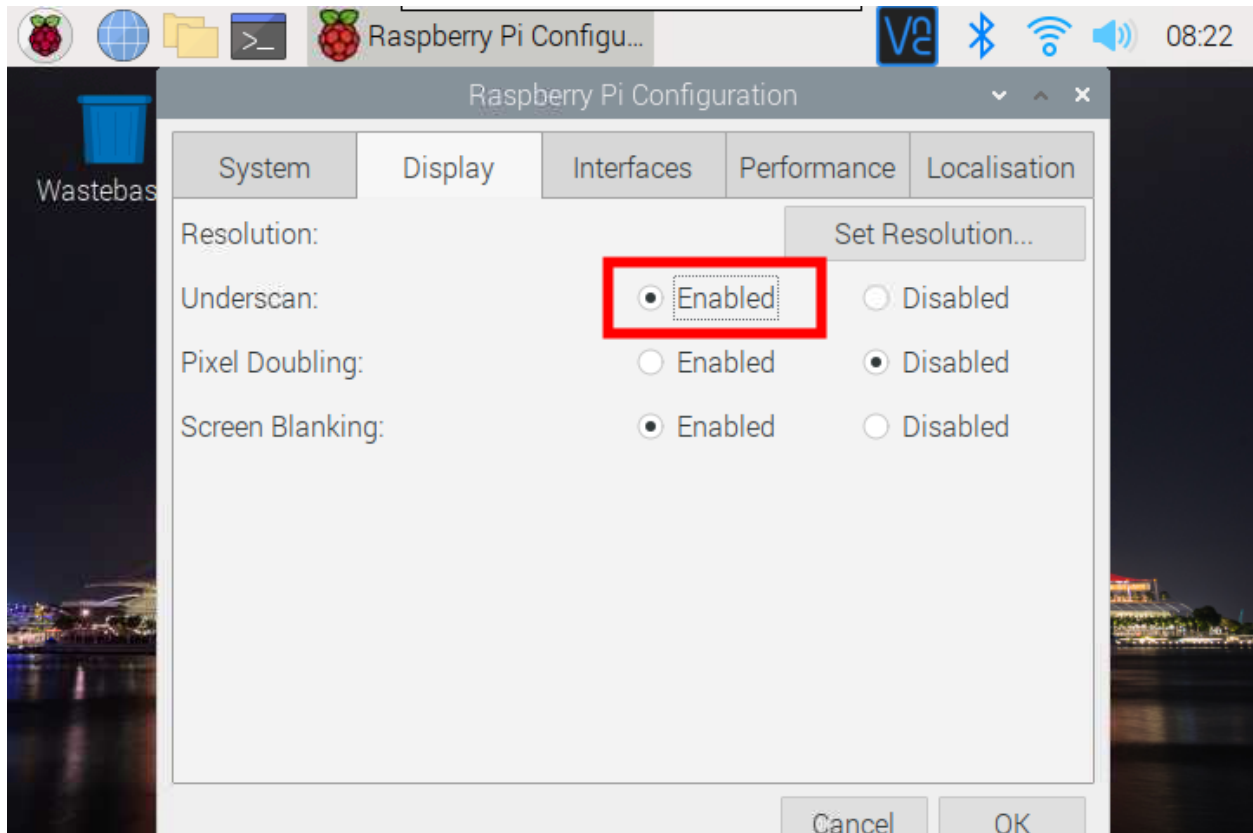
## 8.6 桌面不是全屏

如果树莓派启动后桌面没有全屏显示，则需要开启 **Underscan**。

单击左上角的菜单图标，然后选择 **Preferences -> 树莓派 Configuration**。



在 **Display** 页面，开启 **Underscan**，然后重启树莓派。



## 8.7 关于触摸问题

### A. 触摸不起作用

首先，请检查内部电缆连接是否稳定。

对于老版本的 RasPad 3，因为外接显示器后触摸会变得很奇怪，所以关闭触摸功能。

### B. 触摸不准确

由于树莓派系统是基于 PC 鼠标控制的，人手不能像鼠标那样精准，如果触控精度问题造成不便，建议您增大图标大小，或者使用 *RASPAD Launcher* 来安装 RasPad 主题。

## 8.8 关于音频问题

### A. 没有声音。

如果 RasPad 和树莓派的音量设置为 0，将导致没有声音。

### B. 耳机插孔不能正常工作

由于 3.5mm 耳机插孔位于 RasPad 3 外壳内部，请检查耳机是否完全插入。

## 8.9 风扇太吵？

这是因为风扇转速太高，本来是为了增强 RasPad 3 的散热能力。可以通过将主板左上角的风扇开关设置为's'来降低噪音，但是这个将降低风扇速度以及冷却能力。

## 8.10 耗电太快？

RasPad 的电池容量为  $3 \times 3.7V \times 3Ah = 33Wh$ ，比 iPad pro 2020 的 28.65Wh 大一点。但是因为树莓派本身功耗比较大，待机 3W 左右，而且树莓派系统和平板甚至 iOS 的省电机制都不一样。相比上一代，我们将原来的 7w 屏驱动功耗降低到了 3W 以下，使用时间也得到了提升。

我们将继续优化这种时间的使用。

## 8.11 触摸功能有时会失效？

打开树莓派的一些软件，会发现有些地方不能触摸，这是树莓派操作系统没有对触摸屏做适配造成的。我们解决不了，需要等待树莓派 OS 优化解决方案。

---

### 版权声明

---

本手册中包括但不限于文字、图片、代码等所有内容均归 SunFounder 公司所有。您只应将其用于个人学习、娱乐或其他非商业或非营利目的，根据相关法规和版权法，在不侵犯作者及相关权利人合法权益的前提下。对于任何个人或未经许可将其用于商业利益的组织，本公司保留采取法律行动的权利。