

00A0 2203∃ 2200∀ 2286⊆ 2713x 27FA⇔ 221A√ 221B∛ 2295⊕ 2297⊗ UTF8gbsn

SunFounder pismo

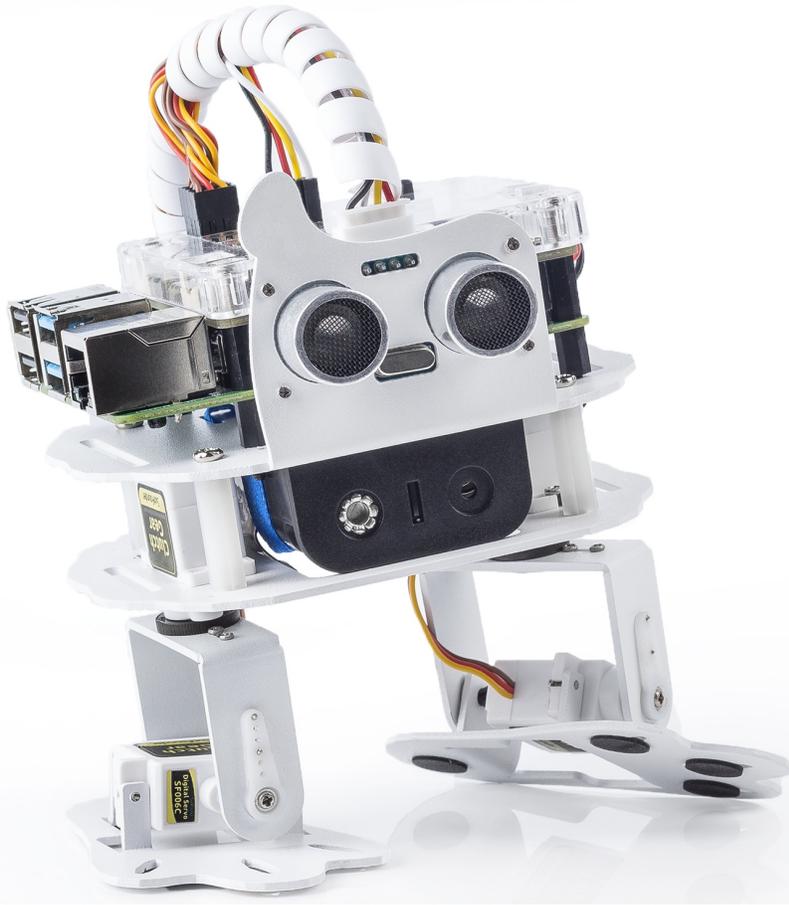
SunFounder

2024 年 03 月 11 日

1	元件清单和组装说明	3
2	关于 Robot HAT	5
3	玩转 EzBlock	9
3.1	EzBlock 快速指南	9
3.2	移动	23
3.3	别碰我	27
3.4	避障	29
3.5	情绪丰富的 PiSloth	31
3.6	跳舞	36
3.7	战斗吧！勇士！	40
3.8	遥控	46
3.9	自定义步态	50
3.10	自定义动作	57
4	玩转 Python	61
4.1	Python 快速指南	61
4.2	移动	81
4.3	跳舞	82
4.4	避障	85
4.5	别碰我	87
4.6	战斗吧！战士！	88
4.7	情绪丰富的 PiSloth	90
4.8	遥控	93
4.9	自定义步态	95
4.10	自定义动作	99
5	附录	101
5.1	网页版 EzBlock	101
5.2	apt 和 pip 更换国内源	105
5.3	Filezilla 软件	107
5.4	I2C 配置	109
5.5	远程桌面	111
5.6	关于电池	120
6	疑难解答	123

6.1	安装 EzBlock 操作系统后，舵机不能转到 0°?	123
6.2	如何在 EzBlock 重新校准机器人	124
6.3	EzBlock 无法连接蓝牙?	125
6.4	APP 搜索到蓝牙，但无法连接	125
6.5	配置 WIFI 后 APP 无法连接	126
6.6	为什么伺服机有时会无缘无故地返回到中间位置?	126
7	版权声明	127

感谢您选择我们的 PiSloth。



PiSloth 是一款铝合金结构的树莓派仿生双足机器人。它可以说话、跳舞，表达快乐和兴奋等情绪。它有 22 种不同的动作，例如跺脚和太空步等，您可以根据需要自定义动作。PiSloth 的眼睛由超声波传感器模块组成，可用于检测距离以实现避障和跟随功能。

本教程包括了清单列表、组装 pdf、Robot HAT 介绍和 PiSloth 编程。

编程部分分为两章：[玩转 EzBlock](#) 和 [玩转 Python](#)，每一章都可以让你掌握如何让 PiSloth 以你想要的方式工作。

Ezblock Studio 是 SunFounder 为初学者开发的开发平台，旨在降低树莓派入门门槛。它有两种编程语言：图形化编程和 Python，几乎可以在所有不同类型的设备上使用。通过蓝牙和 Wi-Fi 支持，您可以在 Ezblock Studio 上下载代码、远程控制树莓派。

更有经验的创客可以使用流行的编程语言——Python。

目录

元件清单和组装说明

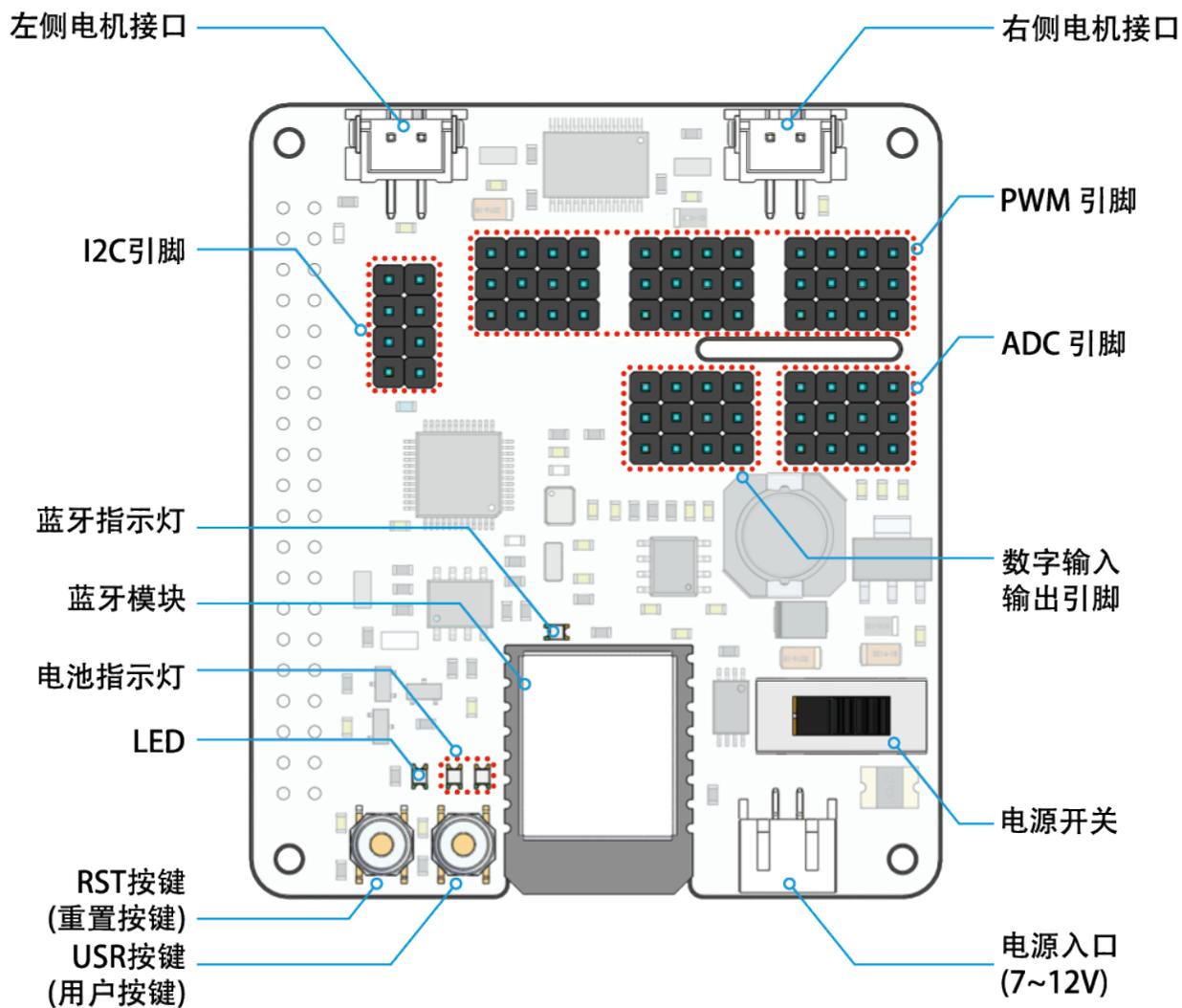
您需要先根据清单检查是否有缺少或损坏的组件。如果有任何问题，请联系我们，我们会尽快解决。
请按照 PDF 上的步骤进行组装。

备注:

1. 组装前需要购买 2 节 18650 电池并充满电，参见[关于电池](#)。
 2. Robot HAT 不能给电池充电，需要额外购买充电器。
-

- [PiSloth 组装教程](#)

关于 Robot HAT



RST 按钮

- 短按 RST 按钮将导致所有正在运行的程序重置。
- 长按 RST 按钮直到 LED 亮起，然后松开将断开 Robot HAT 的蓝牙芯片。

USR 按钮

- USR 按钮的功能可通过编程进行配置。(按下导致输入 0，松开产生 1 输入)

LED

- 通过编程配置 (输出 1 打开 LED，输出 0 关闭 LED。)

电池指示灯

- 电池电压高于 7.8V 将点亮两个 LED 指示灯。电池电压在 6.7V 到 7.8V 之间只会点亮一个 LED，低于 6.7V 的电压将关闭两个 LED。

蓝牙指示灯

- 蓝牙指示灯 LED 将在蓝牙连接稳定时保持亮起，并在信号传输期间快速闪烁。如果蓝牙断开连接，LED 将每隔 1 秒闪烁一次。

EzBlock 是 SunFounder 所使用的一个软件开发平台，用于帮助初学者和新手在树莓派平台上开始学习编程。Ezblock 有两种编程语言环境：一种是图形用户界面环境，一种是命令行 Python 环境。

EzBlock 几乎可用于所有类型的设备，包括 Mac、PC 和 Android。

在使用 EzBlock 之前，请先按照以下教程来完成 EzBlock 下载，安装和使用。

3.1 EzBlock 快速指南

这里是使用 EzBlock 的一些使用指南。在组装的时候，你只需要完成安装 *EzBlock* 镜像和舵机调零。

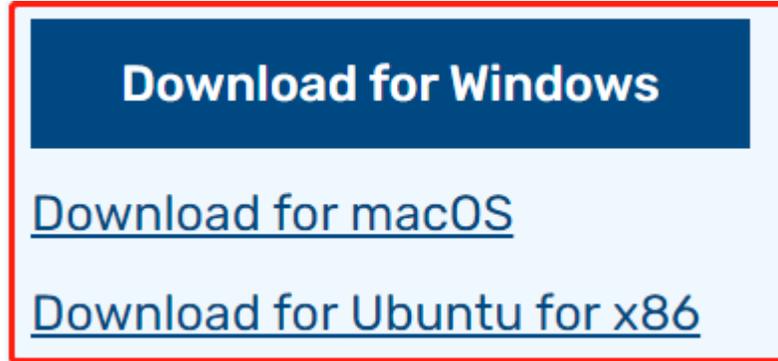
组装完成后再去下载 *EzBlock Studio*，完成快速配置后，就可以根据教程内容对你的机器人进行编程了。

建议按顺序阅读课程内容。

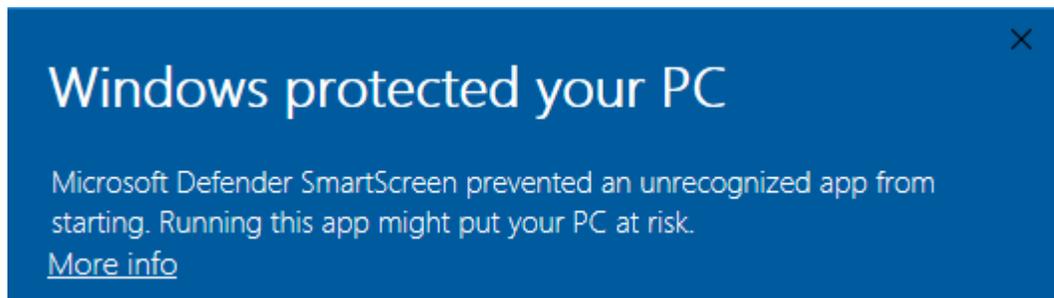
3.1.1 安装 EzBlock 镜像

在组装机器人之前，按照下面的步骤给树莓派的 Micro SD 卡烧录 EzBlock 系统。

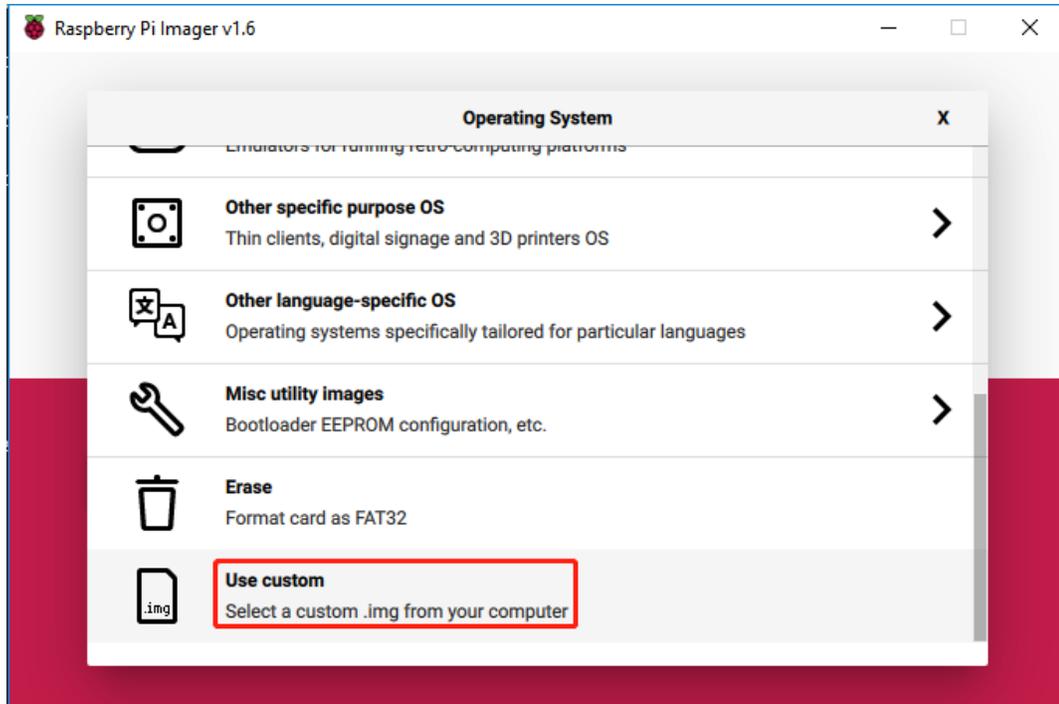
1. 树莓派开发了一个图形 SD 卡写入工具，适用于 Mac OS、Ubuntu 18.04 和 Windows，对于大多数用户来说是最简单的选择，因为它会下载映像并将其自动安装到 SD 卡。访问下载页面：<https://www.raspberrypi.org/software/>。单击与您的操作系统匹配的 Raspberry Pi Imager 链接，下载完成后，单击它以启动安装程序。



2. 当您启动安装程序时，您的操作系统可能会尝试阻止您运行它。例如，在 Windows 上，我收到以下消息：如果出现此消息，请点击 **更多信息**，然后点击 **仍然运行**，然后按照说明安装 Raspberry Pi Imager。

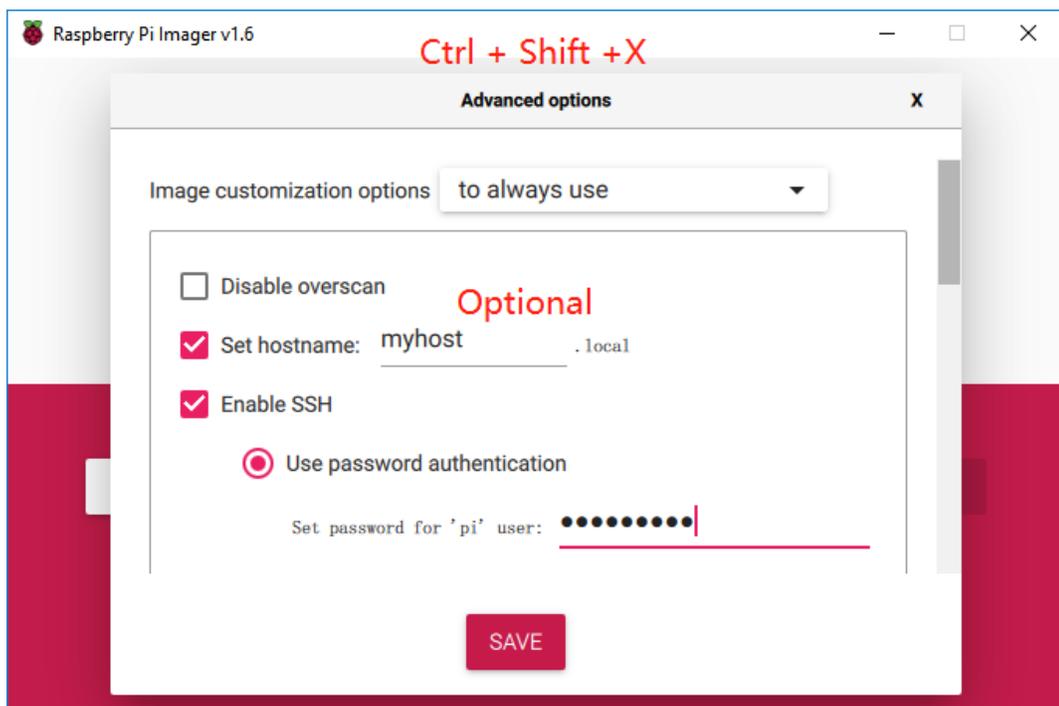


3. 将 SD 卡用读卡器插入计算机或笔记本电脑的 SD 卡插槽。
4. 下载 EzBlock 镜像
 - 天翼网盘：链接：<https://cloud.189.cn/t/QRnEJnEjYNRz>
 - 百度网盘：链接：https://pan.baidu.com/s/1ku1VoukCebChq9-OzkHf_g?pwd=ezbl，提取码：ezbl。
 - 由于文件超过 1G，需要在电脑上下载客户端之后才能下载文件。
5. 在 Raspberry Pi Imager 中选择刚下载的 EzBlock 镜像。



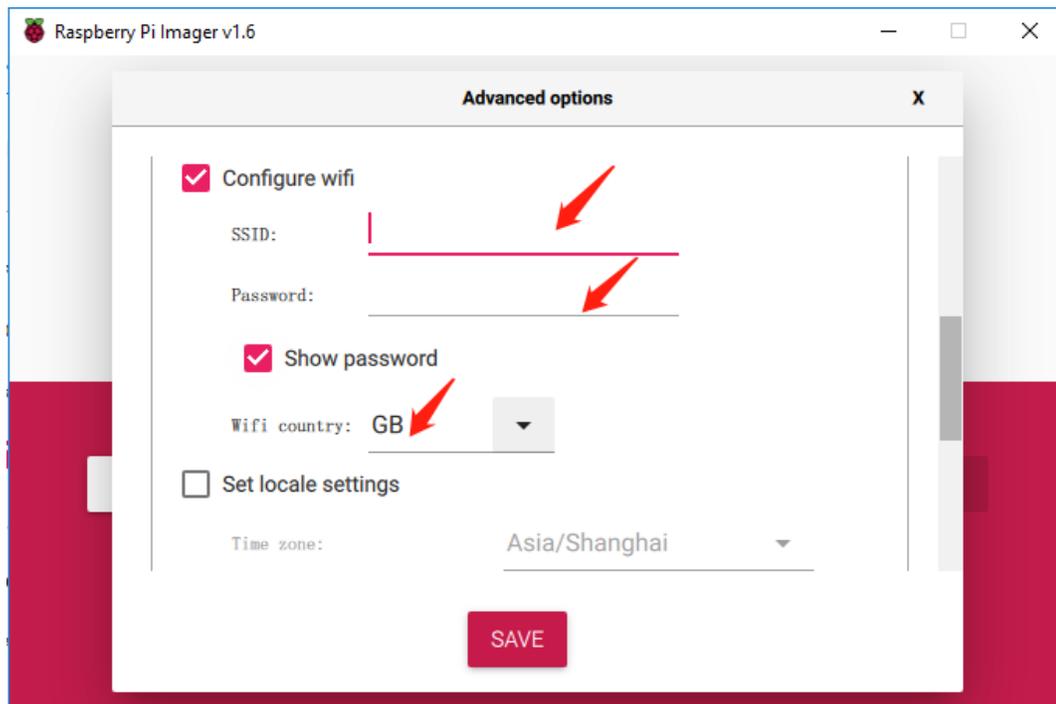
6. 按下 **Ctrl+Shift+X** 或者点击 **设置**按钮来打开 **Advanced options** 页面来设置 hostname 和启动 SSH。你可以选择 “always use this image customization options”（始终使用该定制选项）。

备注： hostname 是让你在使用网页版 *EzBlock*，可以用它来连接到你的产品，你也可以不设置。

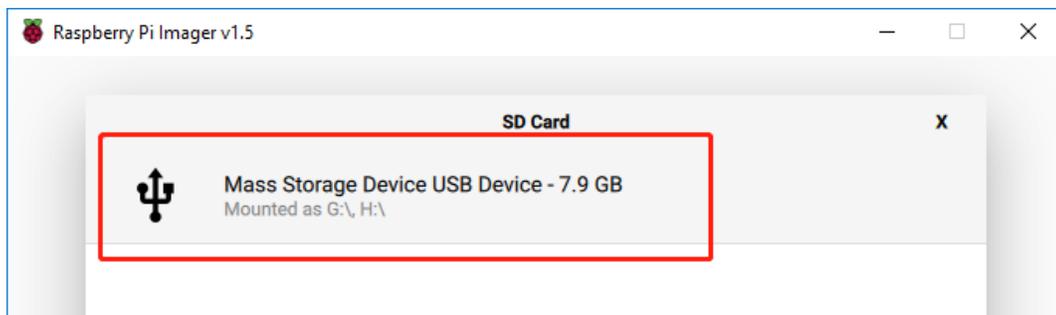


7. 下拉到底完成 WiFi 配置，然后点击 **SAVE**。

备注: wifi country 选择 CN



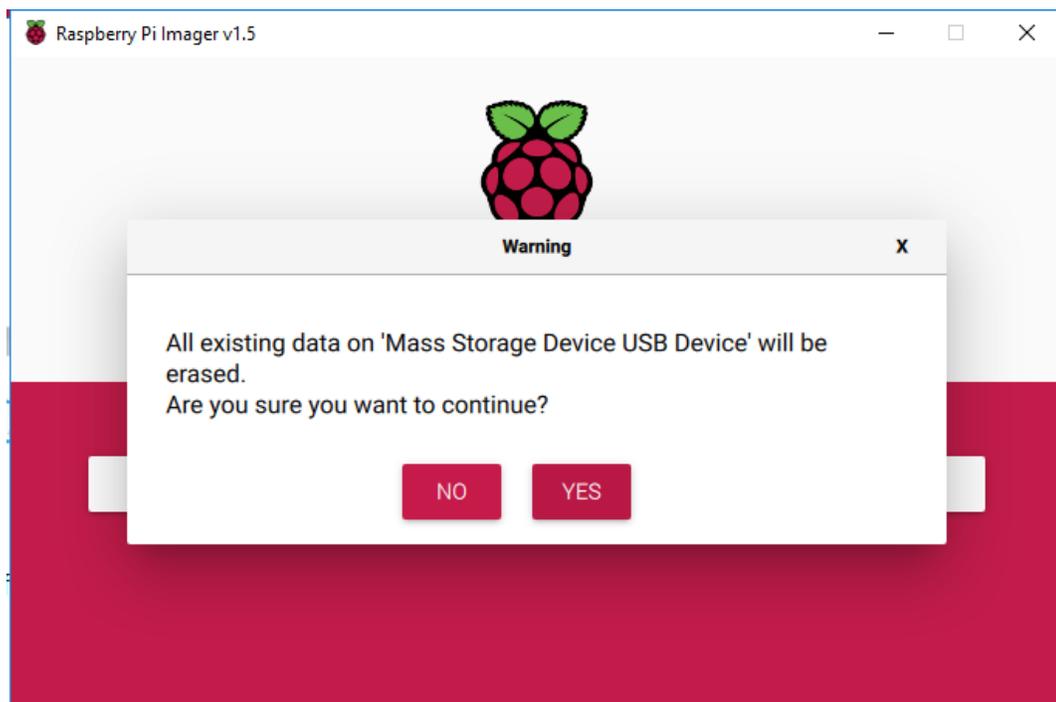
8. 选择您正在使用的 SD 卡。



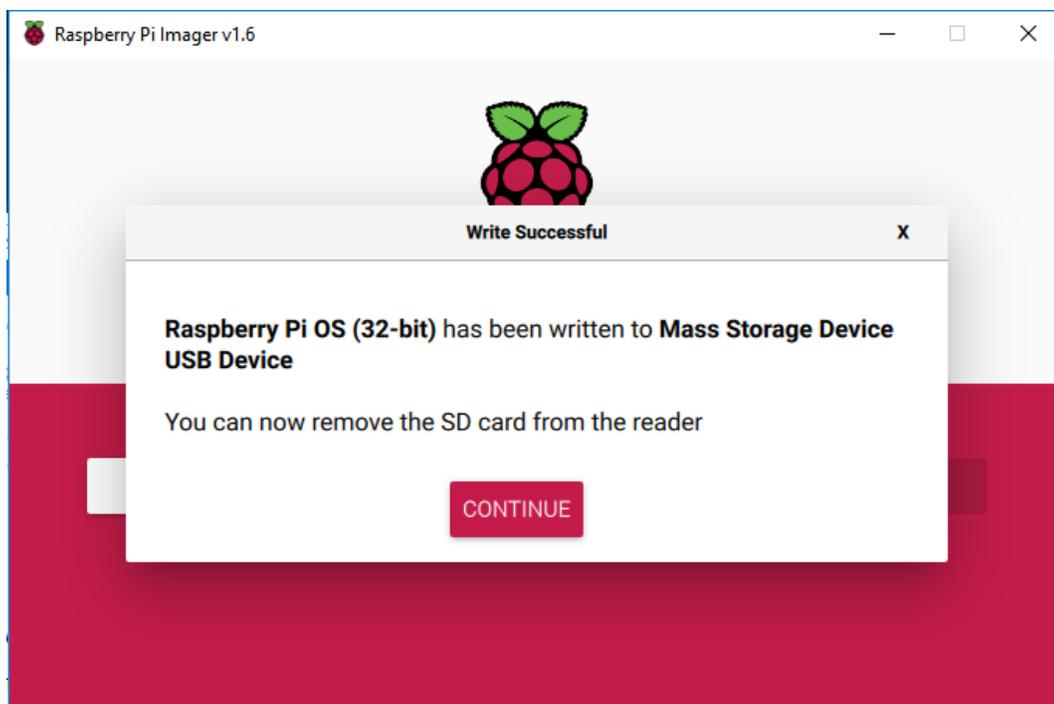
9. 单击 **WRITE** 按钮。



10. 如果您的 SD 卡上当前有任何文件，您可能希望先备份这些文件以防止永久丢失它们。如果没有要备份的文件，请单击 **YES**。

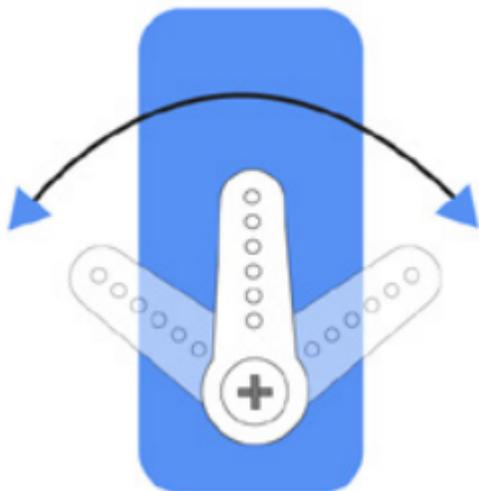


11. 等待一段时间后，会出现如下窗口，代表写入完成。现在你就可以将读卡器从电脑中拔出，并取出 SD 卡插入到树莓派上。

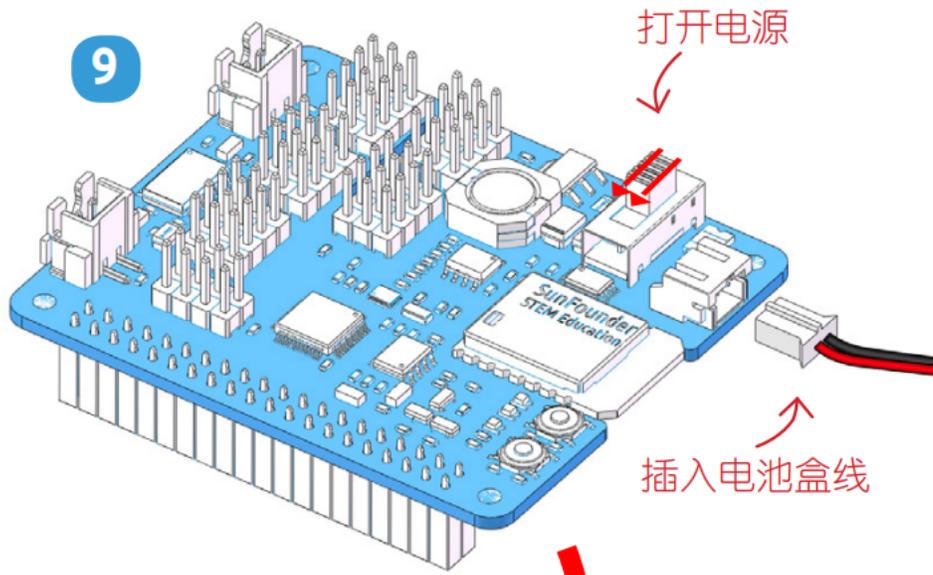


3.1.2 舵机调零

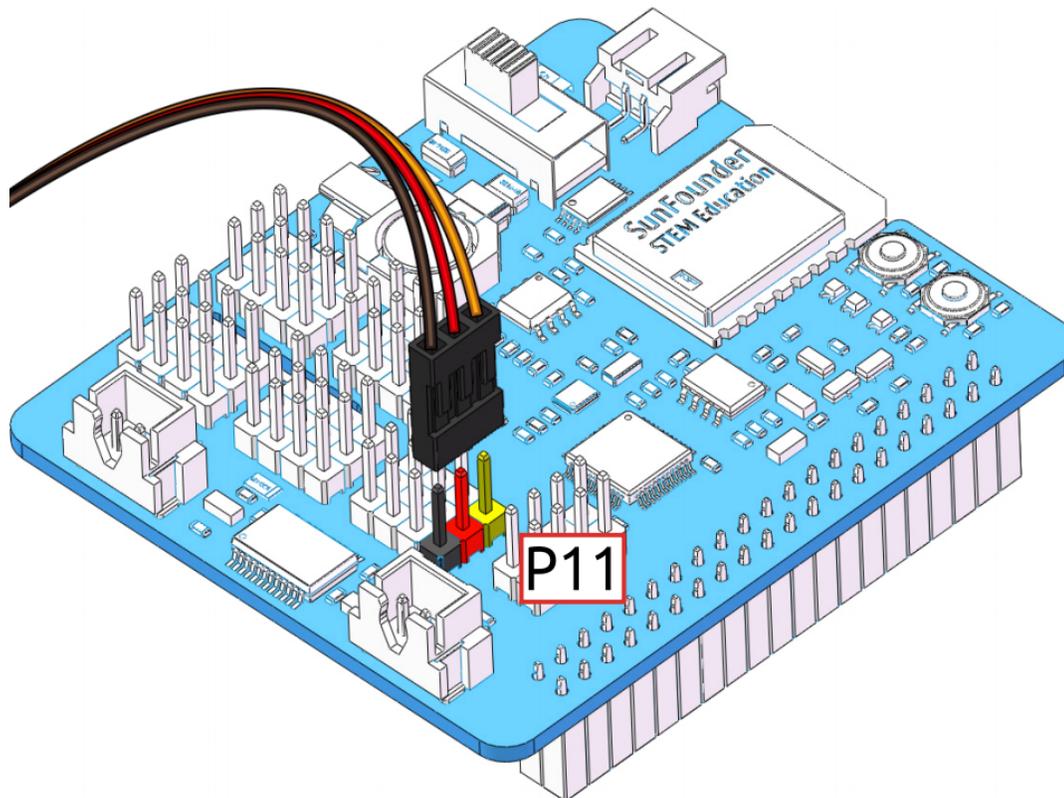
1. 为确保舵机已正确设置为 0° ，首先将摇臂轻轻插入舵机轴，然后将摇臂轻轻旋转到不同的角度。



2. 按照组装折页上的提示，插入电池盒线，将电源开关拨向 ON 的位置。等待 1-2 分钟，会有声音提示树莓派启动成功。



3. 接下来，将舵机线插入 P11 端口，如下所示：



4. 此时你会看到舵机臂转动到特定的位置 (0°)。如果伺服臂没有返回到 0° ，请按 **RST** 按钮重新启动 Robot HAT。
5. 现在你就可以按照组装折页上的指示继续安装。

备注：

- 在用舵机螺丝固定该舵机前，不要拔出该舵机线，可在固定完之后拔出。
 - 不要在上电情况下随意转动舵机以免损坏；如果舵机轴插入的角度不对，需把舵机拔出再重新插入。
 - 在组装每个舵机之前，都需要将舵机电缆插入 P11 并打开电源以将它的角度设置为 0°。
 - 若后面用 EzBlock APP 给机器人下载程序后，此调零功能将失效。
-

3.1.3 下载 EzBlock Studio

扫描下方二维码，下载 EzBlock Studio APP。



备注：

- 对于安卓手机，会提示用浏览器打开。下载成功后，根据提示允许浏览器安装该应用
 - 对于 iPhone 手机，会打开 APP Store 来安装该应用。
 - 对于笔记本和电脑可以使用网页版 [EzBlock](#)。
-

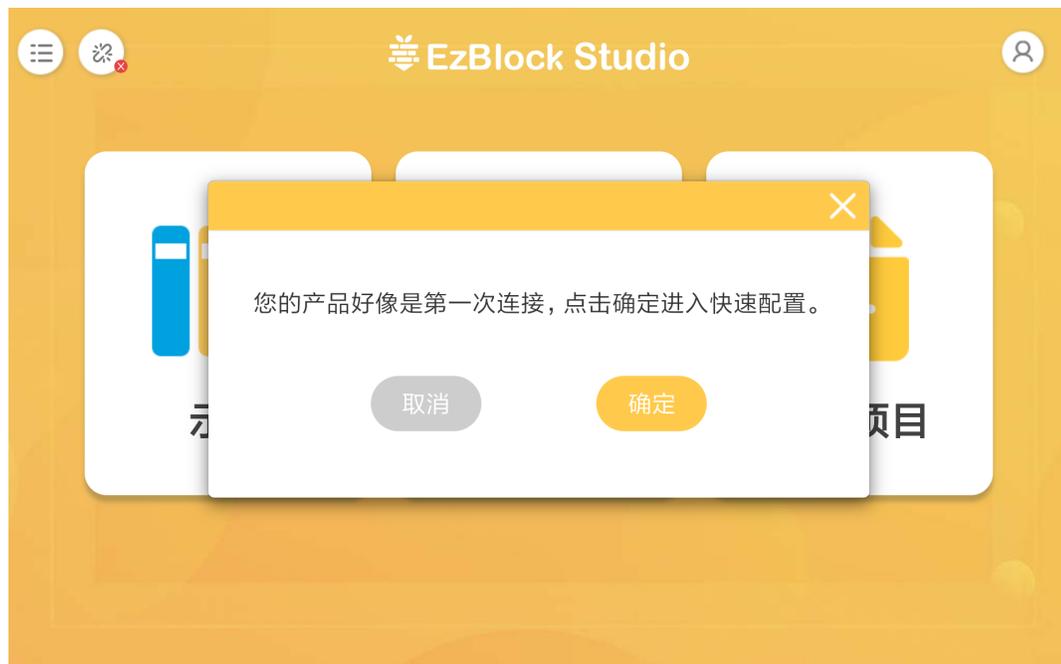
3.1.4 快速配置

在你第一次用 EzBlock 对你的机器人进行编程时，会有一个快速配置的过程。

1. 打开 EzBlock Studio，会弹出一个空设备列表的窗口。您需要在打开产品电源的同时打开移动设备的蓝牙，才会出现产品编号。

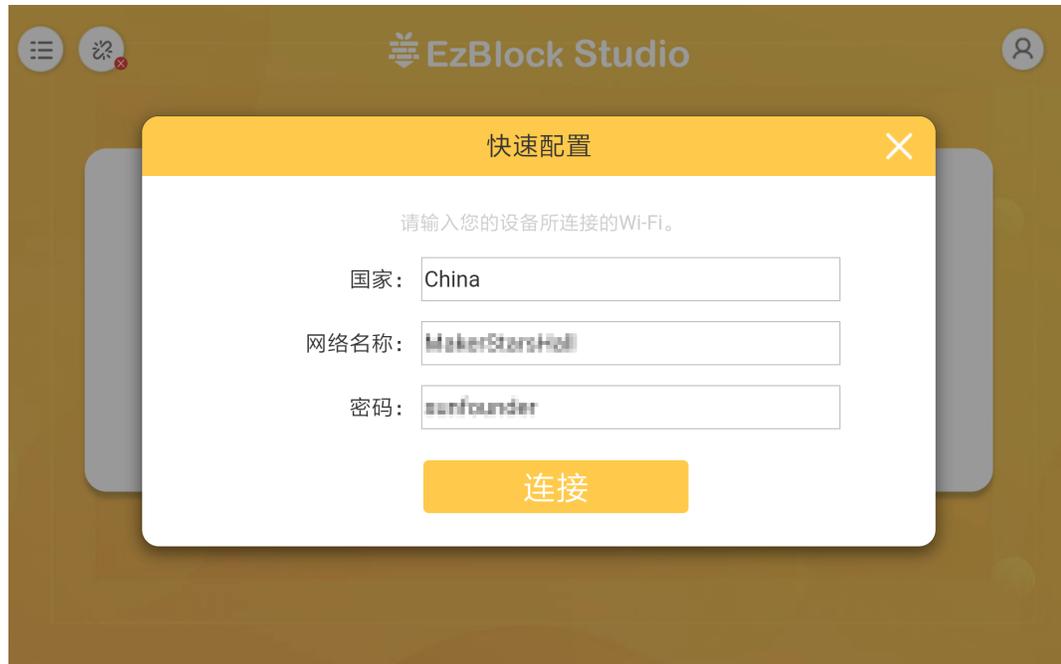


2. 点击右上角的 **完成**，稍等片刻，会提示 **连接成功**。此时您需要点击 **确定** 来快速配置您的产品。



3. 输入您的 Wi-Fi 帐户和密码。

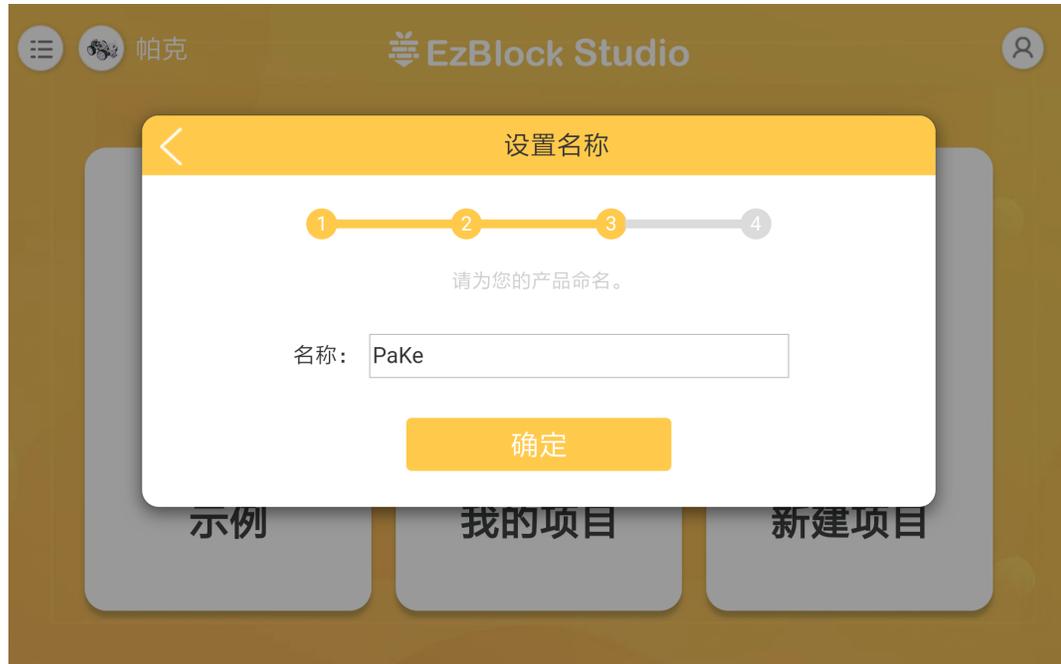
备注：如果您已经提前配置了 Wi-Fi，那么 **快速配置** 这一步就不会出现，直接进入下一步 **设置名称**。



4. 选择与您匹配的产品。



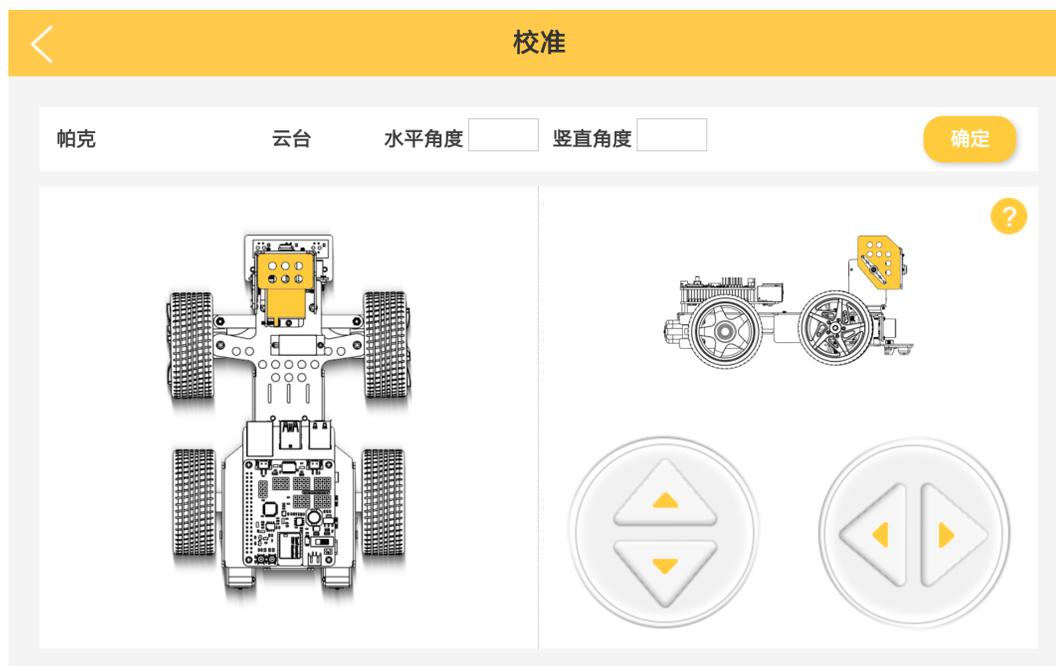
5. 为您的产品输入名称。



6. 如果您的产品需要校准，会有提示告诉您可以通过点击 **立即校准** 进入校准页面。如果不需要，则弹出窗口消失并返回主页。



7. 每个产品的校准页面都不一样，但是都有提示需要校准的部分。您可以点击相应的部分，然后参考 **校准帮助** 进行校准。校准完成后，点击 **确认**。



3.1.5 打开或新建项目

配置完成之后，你就可以开始给你的机器人编写程序，我们已经将所有示例都已上传到 **示例** 页面，你可以直接运行或打开相应的项目。或者根据课程中的提示，新建项目后，通过简单的拖拽来完成相应项目的编程。

打开项目

1. 在首页，点击 **示例**，进入示例页面。如果您只需要简单地测试这些示例，您只需点击 **运行** 即可让您的产品工作。如果要查看和修改里面的代码，那么就需要点击 **编辑** 进入编程页面。



2. 进入编程页面之后，将代码修改之后，点击 **下载/运行** 按钮，就能看到你的机器人做出相应的效果。可点击 **运行/暂停** 按钮来停止运行。



新建项目

1. 点击 新建项目按钮。



2. 填写项目名称。



3. 根据每个项目的提示，编写程序。



4. 点击 **下载/运行** 按钮，就能看到你的机器人做出相应的效果。可点击 **运行/暂停** 按钮来停止运行。



EzBlock 项目

在这里，我们向您展示在 Ezblock Studio 上玩 PiSloth 的项目。如果你是新手，可以参考各个项目里面的代码图片进行编程，可以根据 **提示**来学习积木块的使用。

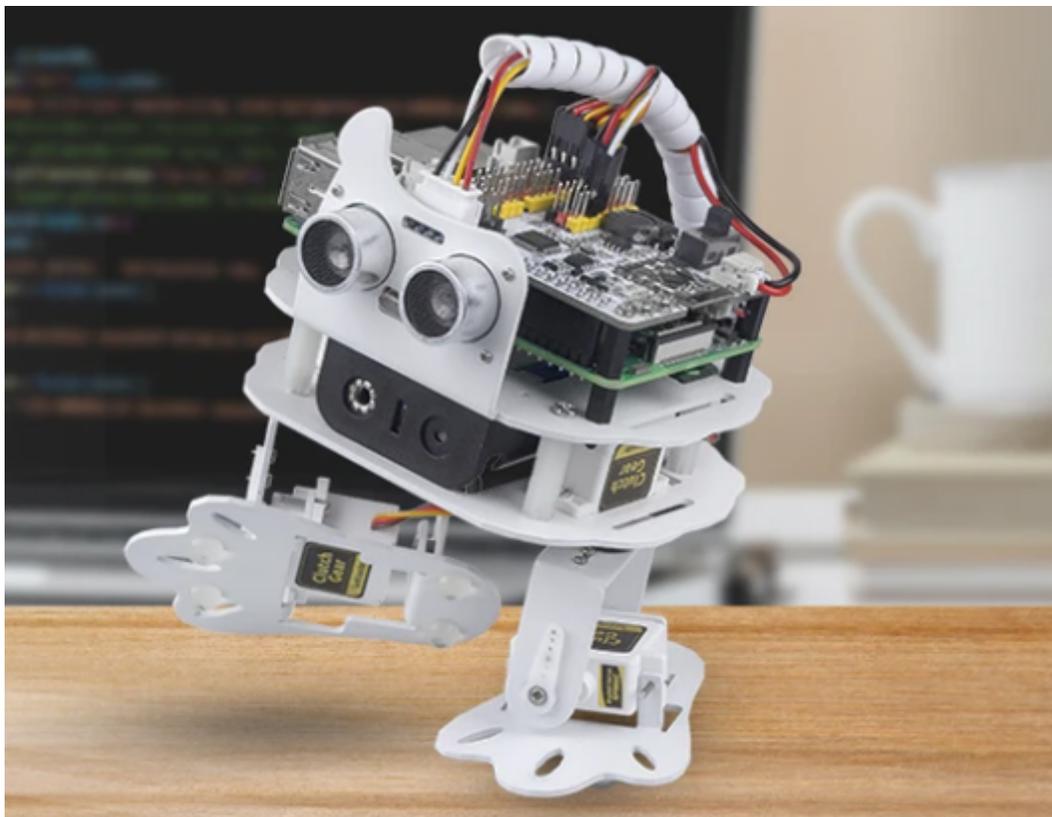
如果你不想一个一个的写这些项目，我们已经把它们上传到 Ezblock Studio 的示例页面，你可以直接运行它们，也可以在运行它们之前稍作编辑。

3.2 移动

备注：在首页，点击 **示例**，进入示例页面，可以直接打开预设好的示例。如果您只需要简单地测试这些示例，您只需点击 **运行**即可让您的产品工作。



这是第一个项目。PiSloth 醒来，并随意移动。



提示

这是程序的基本结构，**开始**块用于做一些初始化（即使没有放置块，也不能删除）而**循环**块，顾名思义，是一个连续的循环，可以让你的程序改变和回应。



该块用于让 PiSloth 以一个速度 (%) 做一个特定的动作几步，例如让 PiSloth 以 50% 的速度前进 1 步。可以从下拉选项中选择不同的动作。



这是一个设置前一个 block 的时长的 block，单位：ms。



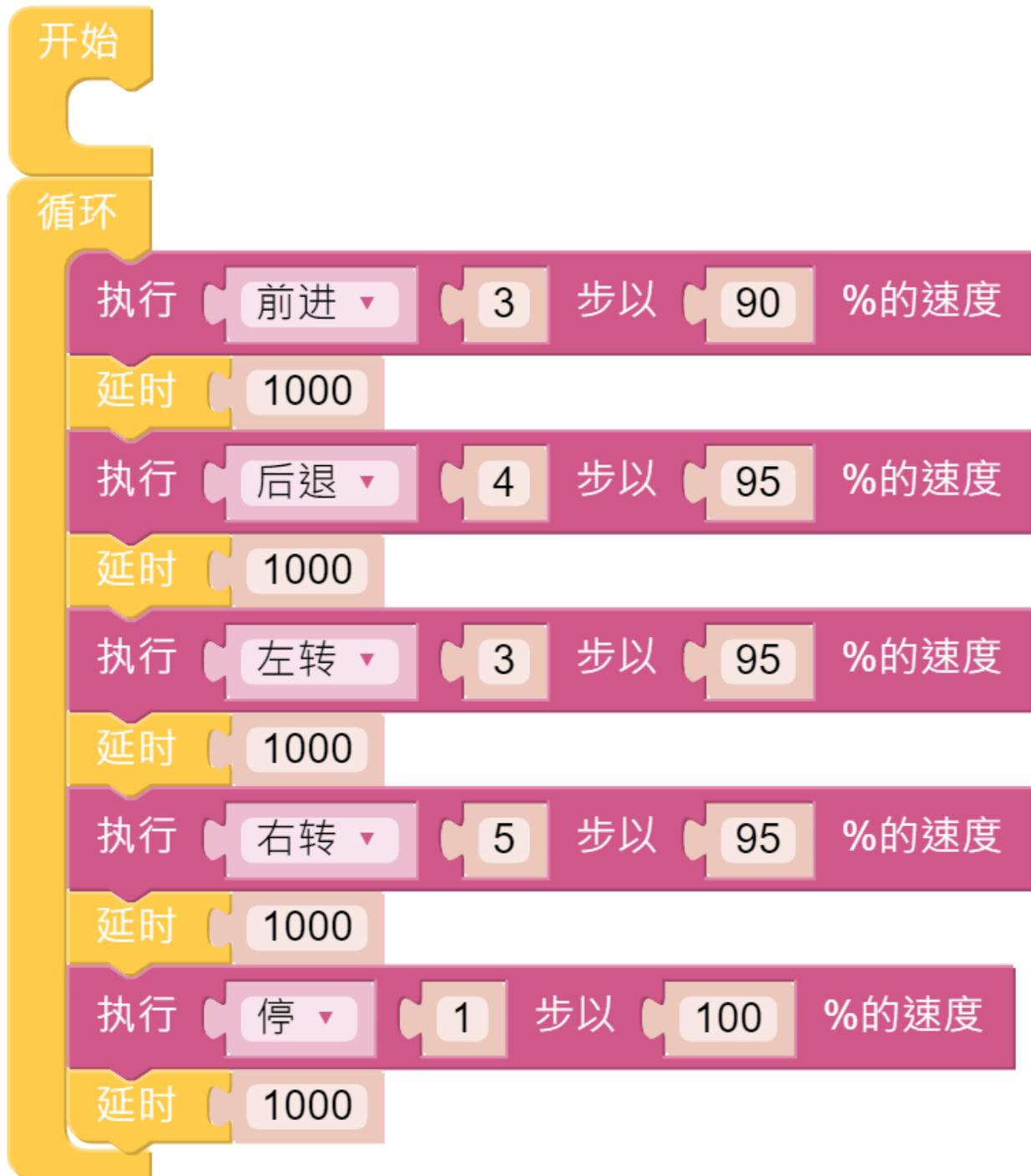
示例

备注：代码完成后，点击右下角的“下载 & 运行”按键，让机器人动起来。



你会看到 PiSloth 前进 3 步，后退 4 步，左 3 步，右 5 步，最后停止。由于整个代码都放在 **循环**块中，PiSloth 会在停止一段时间后重复上述动作。

您可以尝试将 **循环** 块中的代码放入 **开始** 块中，看看有什么不同。



3.3 别碰我

如果它看到你伸手想去摸它，它会生气并离你远远的。

提示

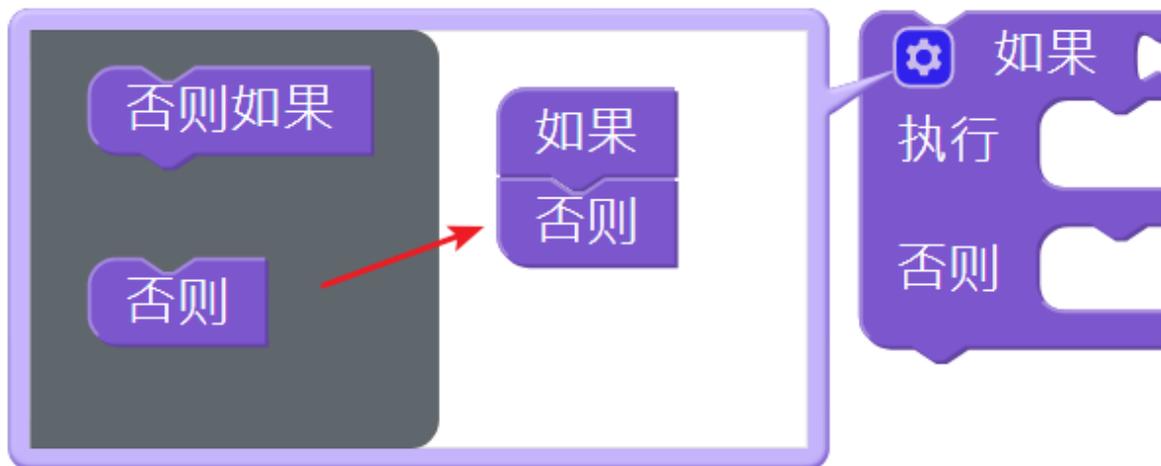
您可以直接使用此块读取与前方障碍物的距离。

备注：如在组装时，Trig 和 Echo 分别接在 D2 和 D3 上，编程时也需要同时更改。



要实现条件判断，需要使用 **如果-执行** 块。

当需要实现多个条件判断时，就得把 if do 改成 if else do。这可以通过单击设置图标来实现。



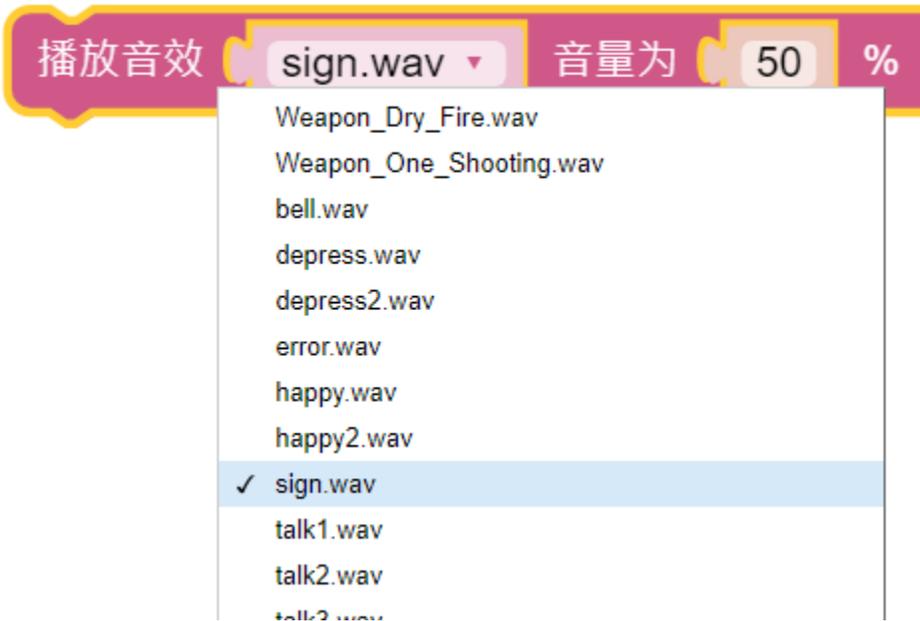
您需要将条件语句块与 **如果-执行** 结合使用。判断条件可以是 “=”、“>”、“<”、“≥”、“≤”、“≠”。



一个数字块。

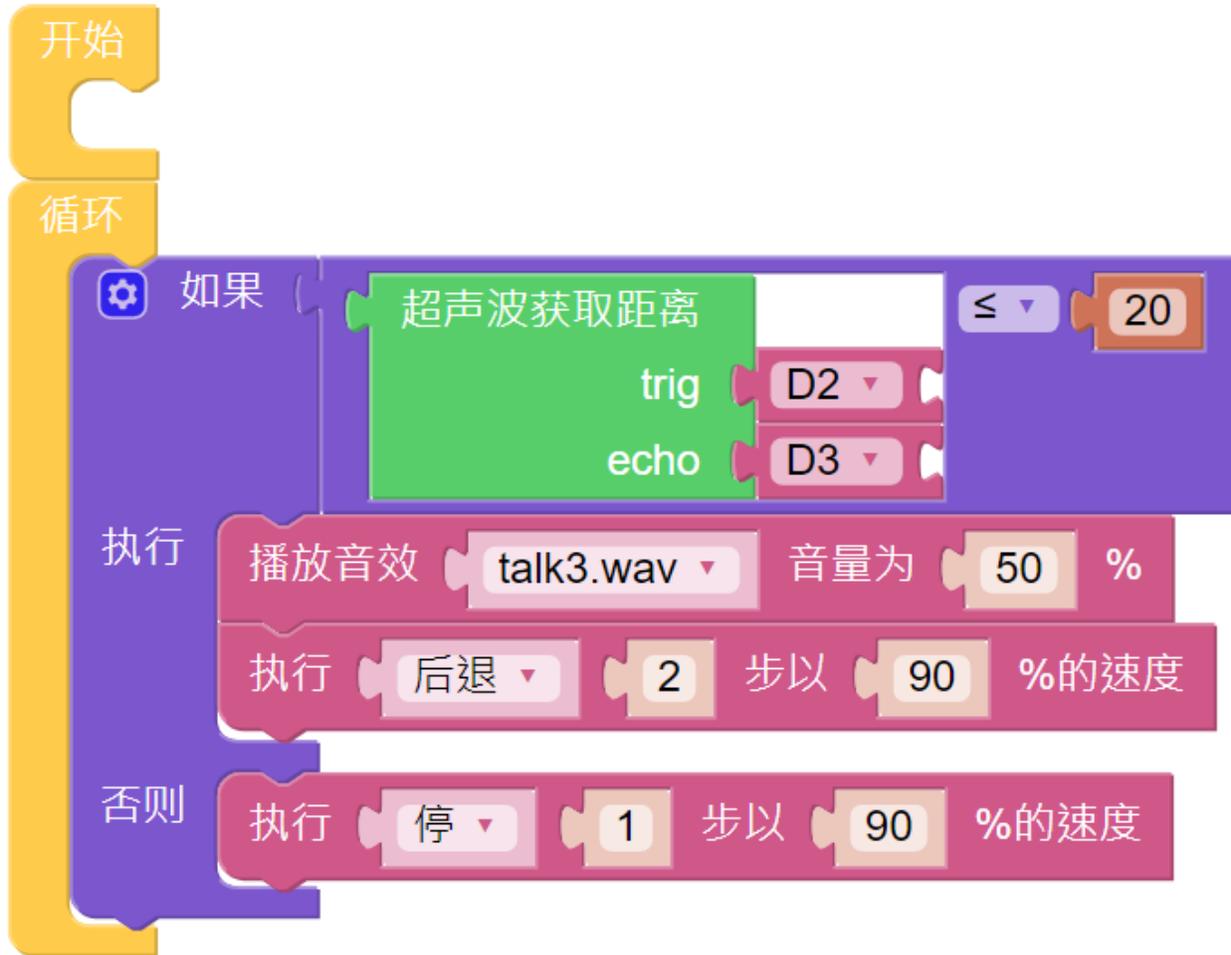


该块可以发出一些预设的音效，如警笛声、枪声等。音量范围为 1~100。



示例

备注： 你可以直接打开我们提供的示例或者是按照下图来编写程序，详细教程请参考[打开](#)或[新建项目](#)。



代码运行后，当你的手靠近 PiSloth，它会发出警告声并后退。

3.4 避障

在这个项目中，当 PiSloth 检测到障碍物时，它会发送信号并寻找另一个方向前进。

提示

这是在之前的项目的基础上，增加了自主判断，让 PiSloth 可以在移动中避开前面的障碍物。

示例

备注： 你可以直接打开我们提供的示例或者是按照下图来编写程序，详细教程请参考[打开或新建项目](#)。



代码运行后，PiSloth 会向前走。如果检测到前方障碍物的距离小于 10cm，它会停下来并发出警告，然后左转。如果左转后方向没有障碍物或障碍物距离大于 10，则继续向前移动。

备注：由于超声波传感器模块的有效检测距离为 2-400cm，当检测距离太近或者太远时，它将无法准确读取距离。

3.5 情绪丰富的 PiSloth

PiSloth 拥有非常多的情绪。时而快乐，时而害羞，时而迷茫。

提示

您可能希望使用 **函数** 来简化程序，尤其是当您多次执行相同的操作时。把这些操作放到一个新声明的函数中，可以大大方便你的使用。

单击 **函数** 类别并选择相应的功能块，您创建的功能也会出现在这里。



此处使用无输出的 **函数** 块。



示例

备注：你可以直接打开我们提供的示例或者是按照下图来编写程序，详细教程请参考[打开或新建项目](#)。

Shy(害羞)



Confuse(迷惑)



Happy(快乐)



Fear(害怕)



Sad(伤心)



Fall(摔倒)



在 **循环**块中的调用所有自定义函数。



3.6 跳舞

当当当当，有请 PiSloth 向大家展示他新学的舞蹈。



备注：您可以为您的 PiSloth 下载并打印卡通面具。

- [卡通面具 \(.pdf\)](#)

提示

PiSloth 除了可以播放音效和说话，还可以播放设定好的背景音乐，背景音乐的音量可以调节 (0%-100%)。



重复块可以帮助您多次执行相同的代码以减少代码大小。

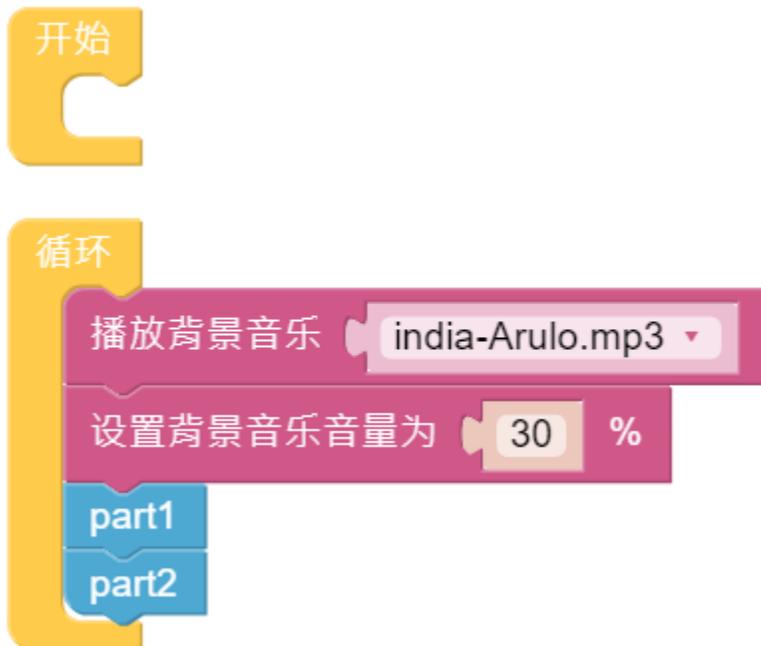


示例

备注： 你可以直接打开我们提供的示例或者是按照下图来编写程序，详细教程请参考[打开或新建项目](#)。



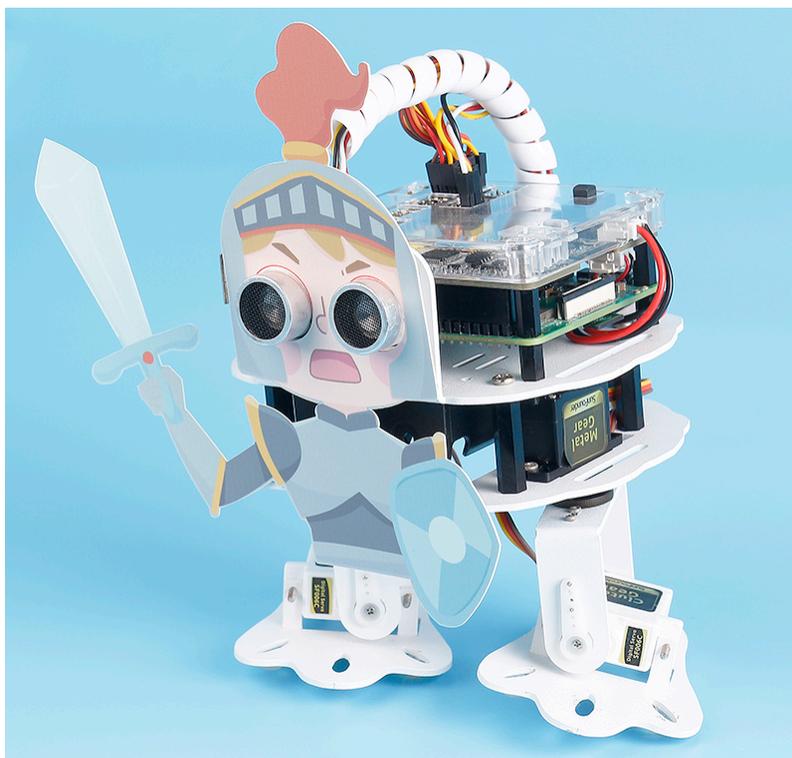




整个舞蹈分为 2 个部分，PiSloth 会随着音乐完成这 2 个部分。如果你不暂停代码，它会重复跳舞一直跳下去。

3.7 战斗吧！勇士！

在这里，PiSloth 是一个勇敢的战士，当它出现在敌人面前时，它会发出咆哮冲向敌人。

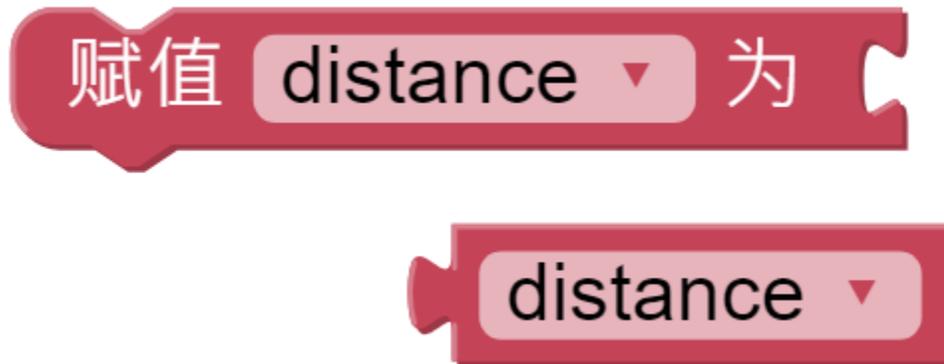


备注：您可以为您的 PiSloth 下载并打印卡通面具。

- 卡通面具 (.pdf)
-

提示

您可能希望使用 **变量**来简化您的程序。例如，当你有多个函数需要读取障碍物距离时，你不需要读取每个函数的值，只需将值加载到一个 **变量**中并多次使用即可。



单击 **变量**类别上的 **创建变量**按钮以创建名为距离的变量。



您可以使用此块来设置无限循环。



这是一个跳出循环的块，它有两个选项，只能在循环内使用。

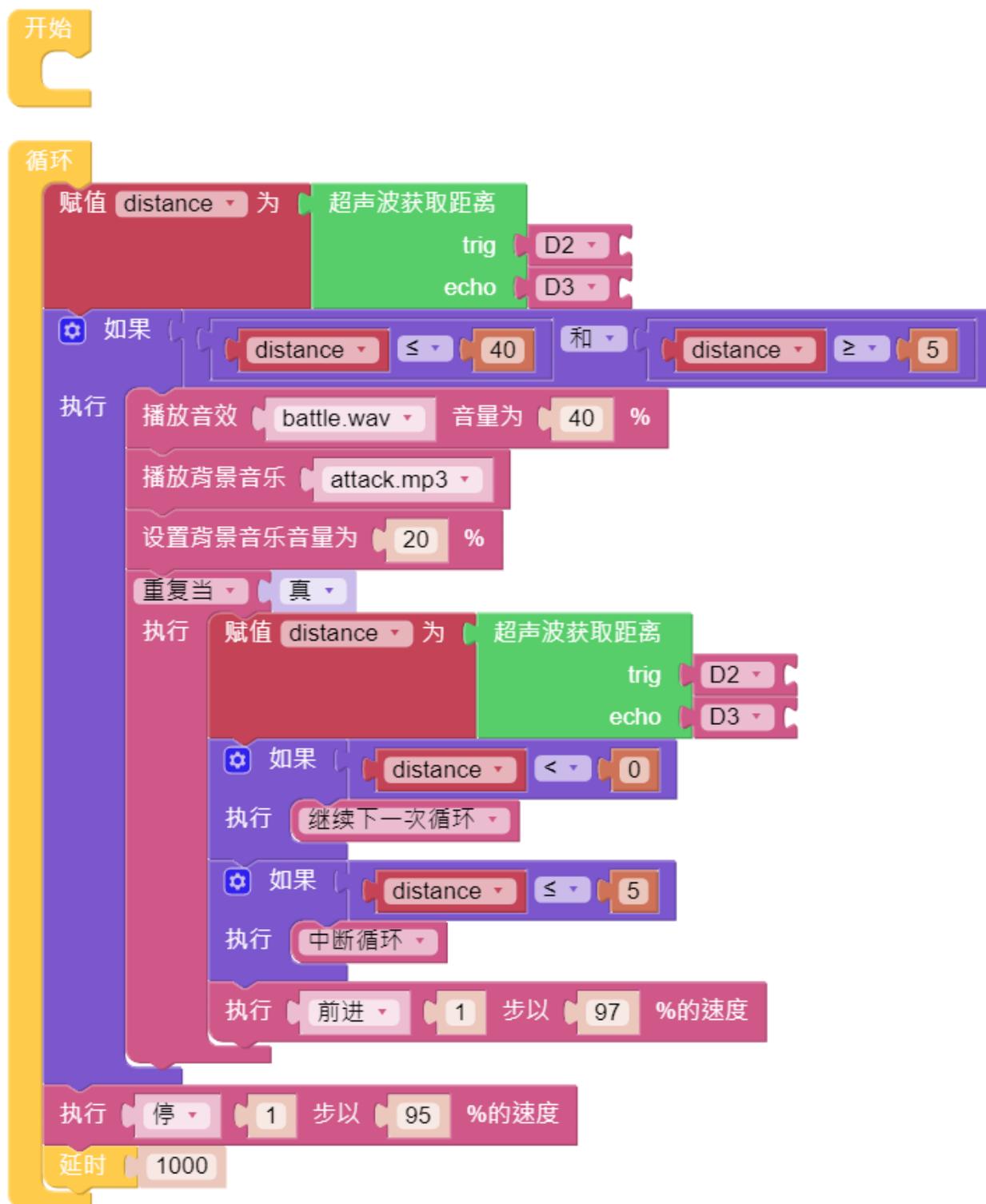
- 中断循环：跳出整个循环。

- 继续下一次循环：跳出当次循环，进入下一次。



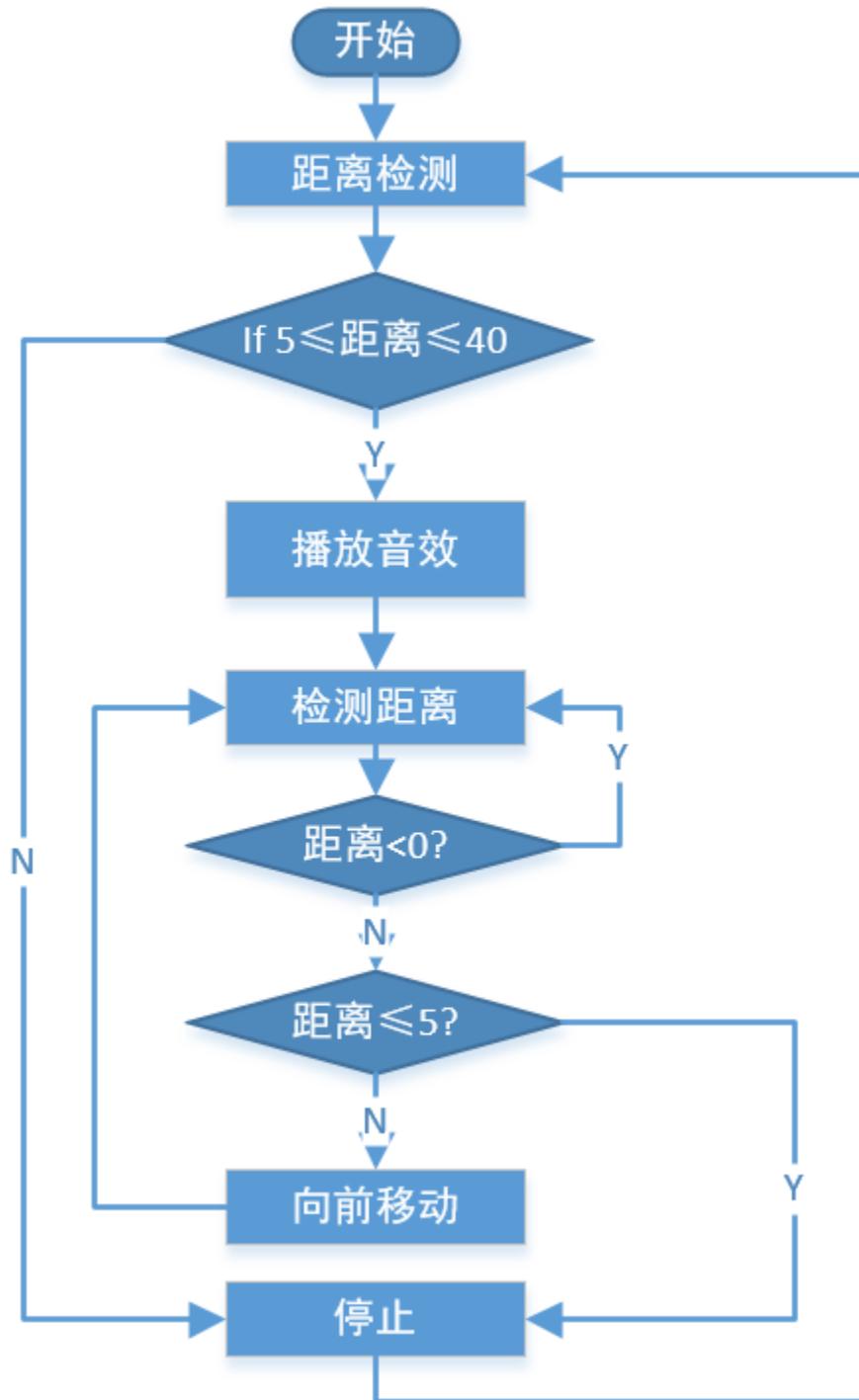
示例

备注：你可以直接打开我们提供的示例或者是按照下图来编写程序，详细教程请参考[打开或新建项目](#)。



代码运行后，PiSloth 会不断检测障碍物的距离，当距离在 5 到 40 之间时，PiSloth 会发出轰鸣声并向前冲去；当障碍物的距离小于 5 时，PiSloth 将停止。

流程图



3.8 遥控

您还可以使用 Ezblock Studio 上的小部件使 PiSloth 移动。



提示

要使用远程控制功能，请从主页面左侧打开远程控制页面。



回到编程页面，您将看到一个附加的远程类别，其中出现方向键和按钮块。

- **按钮 () 获取值:** 该块用于读取按钮的值，如果按钮被按下，则值为 1，否则为 0。
- **按钮 () (按下/松开):** 这个块与上一个块具有相同的效果，可以直接用来判断一个按钮是否被按下。
- **方向盘 () 获取 () 值:** 此块用于读取上/下/左/右（通过下拉菜单选择）键值，按下为 1，松开为 0。



示例

备注： 你可以直接打开我们提供的示例或者是按照下图来编写程序，详细教程请参考[打开或新建项目](#)。

开始

循环



代码运行后，再次进入到 **远程控制** 页面。按下 **方向盘 A** 小部件来控制 PiSloth 的移动，按下按键 A,B,C,D 来让 PiSloth 发出不同的音效。

3.9 自定义步态

在之前的项目中，我们使用了很多我们自己写的动作，那么这些动作是如何组合和完成的呢？一般来说，一个动作由一个或多个步态组成。

在这个项目中，我们将学习如何自定义 PiSloth 的步态。

我们要做的就是使用遥控器页面中的按钮，让 PiSloth 完成下图所示的步态，然后得到 4 个舵机此时的角度。



备注：您可以为您的 PiSloth 下载并打印卡通面具。

- 卡通面具 (.pdf)

第 1 步：在遥控器中拖出 9 个按钮来控制 PiSloth 上 4 个舵机的旋转角度。



第 2 步： 创建 4 个变量来存储 4 个舵机的角度。

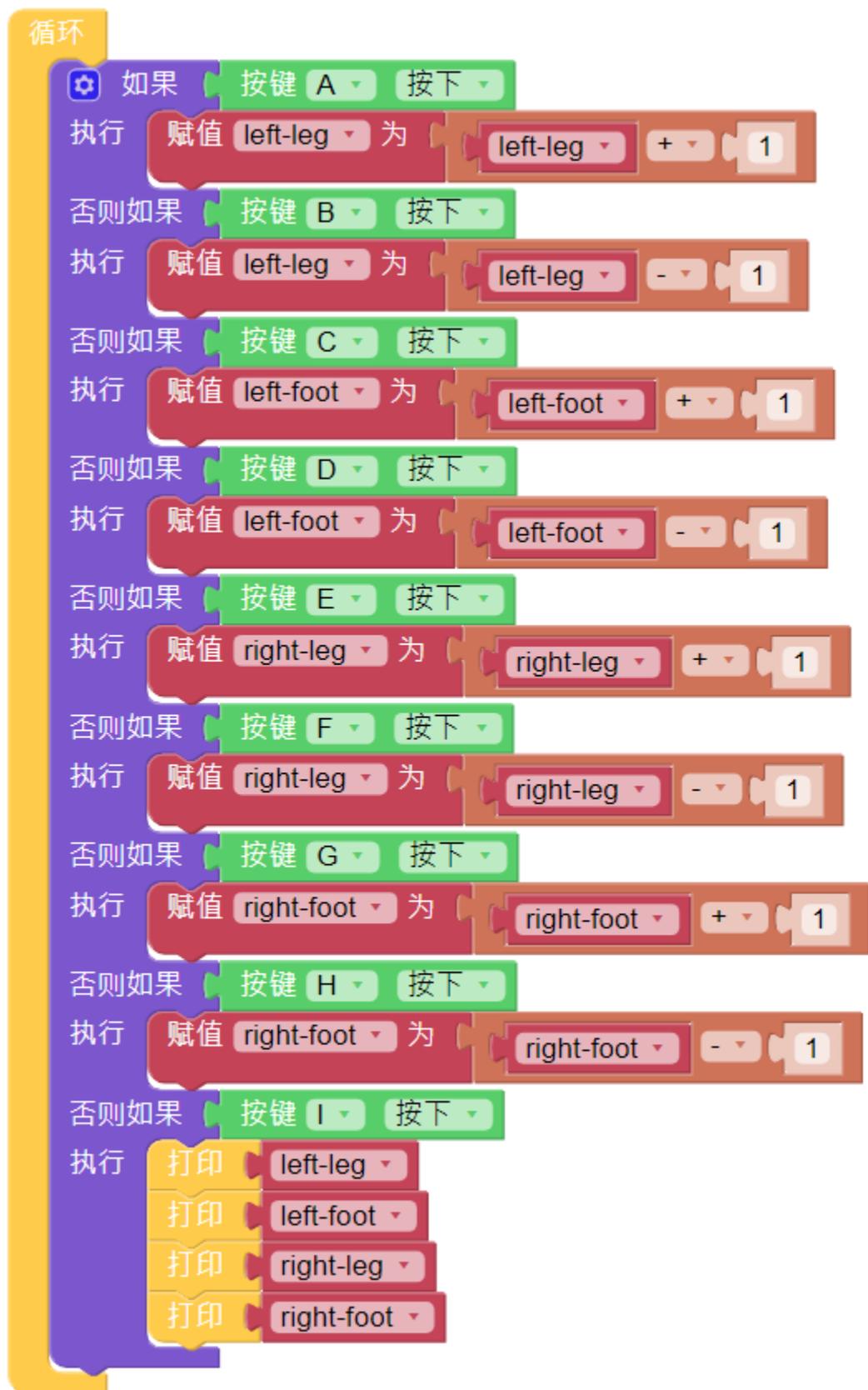


然后将角度初始化为 0。



第3步：读取用于控制舵机角度的不同按钮的值。

- **按钮 AB** 控制 **左腿**。
- **按钮 CD** 控制 **左脚**。
- **按钮 EF** 控制 **右腿**。
- **按钮 GH** 控制 **右脚**。
- 按 **按钮 I**，在调试监视器中打印 4 个舵机的角度。



第4步：在循环块的末尾，填写读取到4个舵机中的角度值，并使用**执行动作块**让PiSloth执行此步骤。

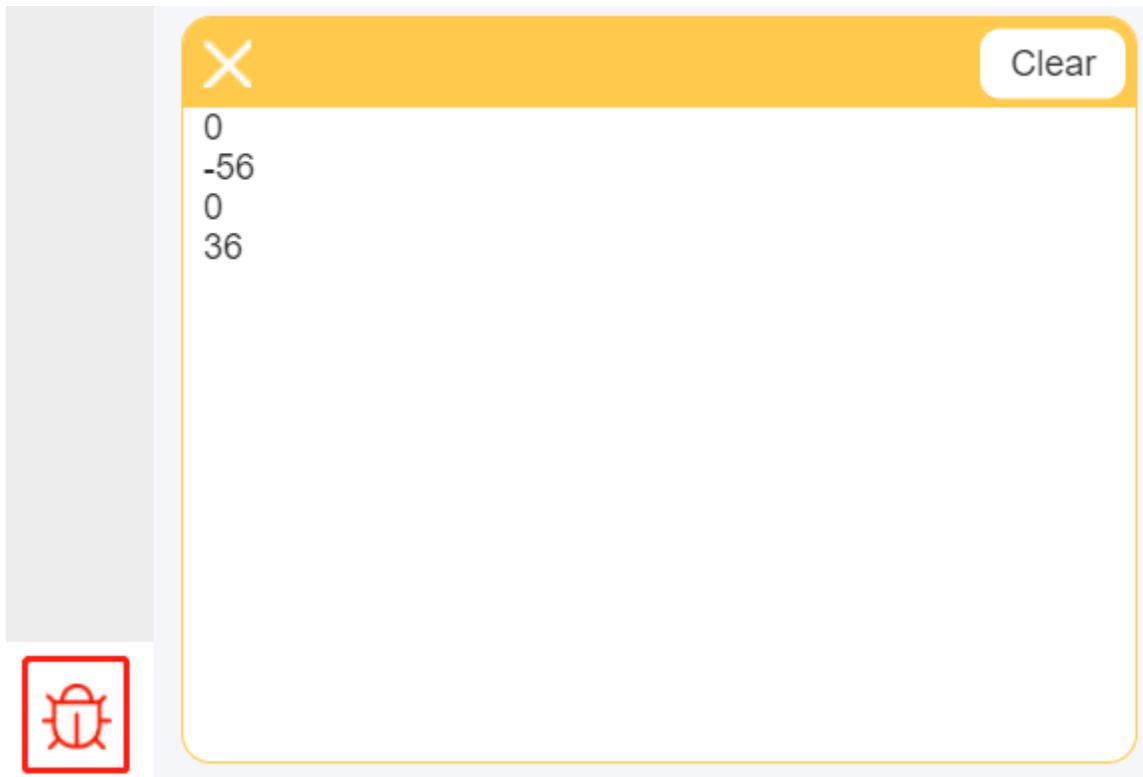


第5步：代码完成后，点击右下角的**下载**图标下载并运行代码。现在我们可以点击**按钮 CD**和**按钮 GH**（根据实际代码）使PiSloth姿势像这样，你也可以让它做其他步骤。



第6步：点击左下角的小虫子图标打开调试监视器，当你按下**按钮 I**时，你会在调试监视器中看到4个舵机的角度。

备注：如果出现多组数据，是因为点击按钮的时间长了一点，Ezblock 会认为按钮被点击了数次。如果数据看着很杂乱，可以点调试监视器的右上角的清除按钮。



完整代码如下：



3.10 自定义动作

在之前的项目中，我们能够给 PiSloth 自定义步态，那么我们如何将这此步态组合成动作呢？

例如，让 PiSloth 摆出上一个项目中步态，然后回到站立姿势，如此循环实现蹦蹦跳跳的效果。



备注：您可以为您的 PiSloth 下载并打印卡通面具。

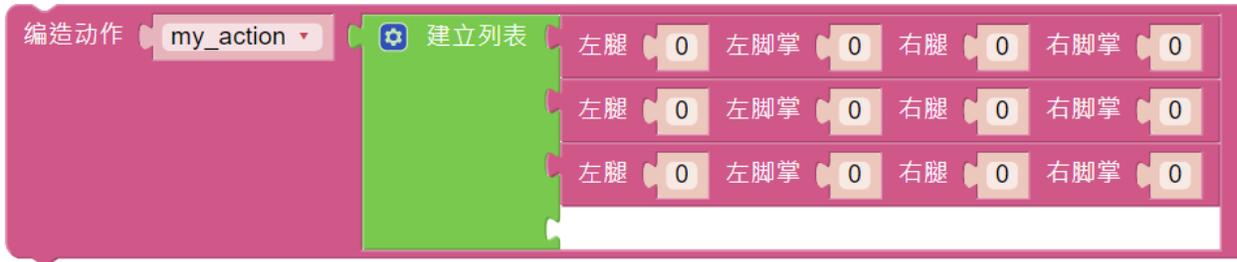
- 卡通面具 (.pdf)

提示

创建一个变量 `up_down` 来存储这个动作。



您可以使用 **编造动作** 块让 PiSloth 执行这些动作。这些值代表了 PiSloth 上 4 个舵机的角度。范围是 (-90~90)。



在这里，您可以通过拖动来增加或减少项目的数量。



填写上一个项目中获得的角度，并将此操作命名为 **up_down**（从 **变量** 中拖出它）。



使用 **执行块** 让 PiSloth 以 50% 的速度执行一次此操作。



示例

备注： 你可以直接打开我们提供的示例或者是按照下图来编写程序，详细教程请参考 [打开或新建项目](#)。

开始

编造动作 建立列表

左腿	0	左脚掌	-56	右腿	0	右脚掌	36
左腿	0	左脚掌	0	右腿	0	右脚掌	0

循环

执行 1 步以 50 %的速度

如果你想用 python 编程，那么你需要学习一些基本的 Python 编程技巧和树莓派的基础知识，请先根据 Python 快速指南配置树莓派。

4.1 Python 快速指南

本节教大家如何安装树莓派操作系统，配置树莓派 wifi，远程访问树莓派运行相应代码。

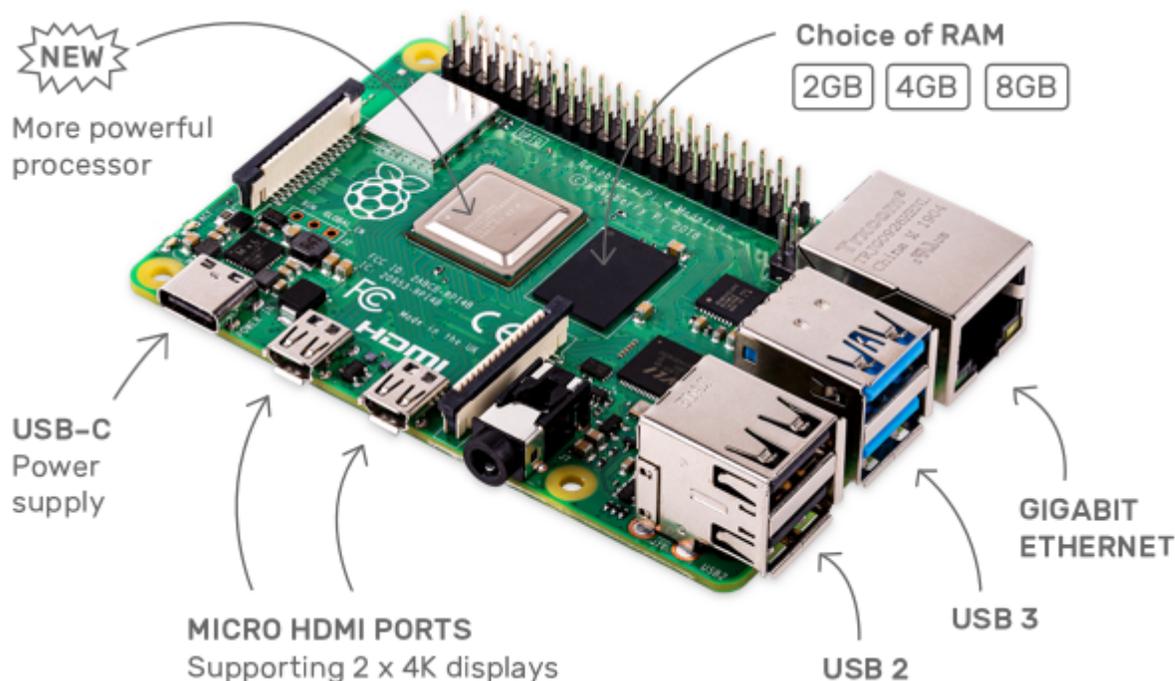
如果您熟悉树莓派并且可以成功打开命令行，那么您可以跳过前 3 部分，然后完成最后一部分。

4.1.1 我们需要什么？

所需组件

树莓派

树莓派是一款低成本、信用卡大小的计算机，可插入进入电脑显示器或电视，并使用标准键盘和鼠标。这是一款功能强大的小型设备，可让所有年龄段的人探索计算，并学习如何使用 Scratch 和 Python 等语言进行编程。



电源适配器

为了连接到电源插座，树莓派有一个微型 USB 端口（在许多手机上都可以找到）。您将需要一个提供至少 2.5 安培电流的电源。

微型 SD 卡

您的树莓派需要一张 Micro SD 卡来存储其所有文件和树莓派操作系统。您需要一张容量至少为 8 GB 的微型 SD 卡。

可选组件

屏幕

要查看树莓派的桌面环境，需要使用的屏幕可以是电视屏幕，也可以是电脑显示器。如果屏幕有内置扬声器，Pi 通过它们播放声音。

鼠标和键盘

使用屏幕时，还需要一个 USB 键盘和一个 USB 鼠标。

HDMI

树莓派有一个 HDMI 输出端口，与大多数现代电视和计算机显示器的 HDMI 端口兼容。如果您的屏幕只有 DVI 或 VGA 端口，则需要使用合适的转换线。

外壳

你可以把树莓派放在一个盒子里；通过这种方式，您可以保护您的设备。

声音或耳机

树莓派配备了一个 3.5mm 左右的音频接口，可以在你的屏幕没有内置扬声器或者没有屏幕操作的情况下使用。

4.1.2 安装操作系统

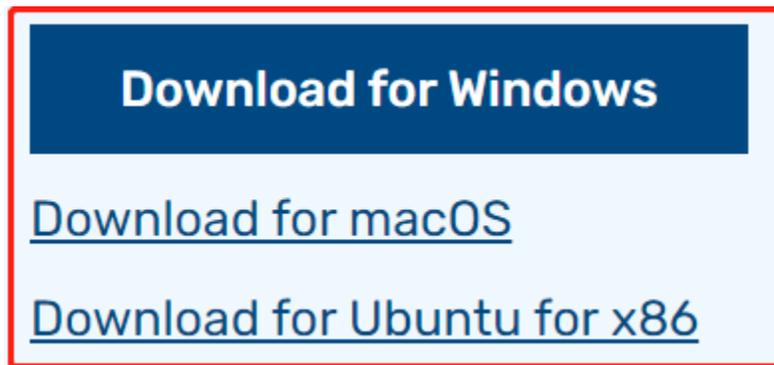
必需组件

任意树莓派	1 * 个人计算机
1 * 微型 SD 卡	

第 1 步

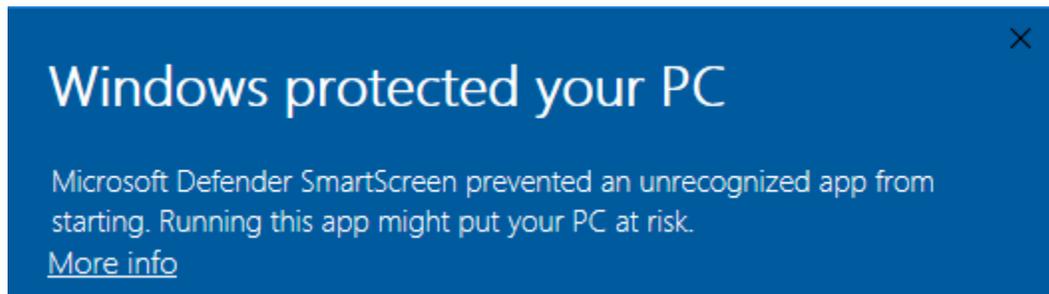
树莓派开发了一个图形 SD 卡写入工具，适用于 Mac OS、Ubuntu 18.04 和 Windows，对于大多数用户来说是最简单的选择，因为它会下载映像并将其自动安装到 SD 卡。

访问下载页面：<https://www.raspberrypi.org/software/>。单击与您的操作系统匹配的 Raspberry Pi Imager 链接，下载完成后，单击它以启动安装程序。



第 2 步

当您启动安装程序时，您的操作系统可能会尝试阻止您运行它。例如，在 Windows 上，我收到以下消息：如果出现此消息，请点击 **更多信息**，然后点击 **仍然运行**，然后按照说明安装 Raspberry Pi Imager。



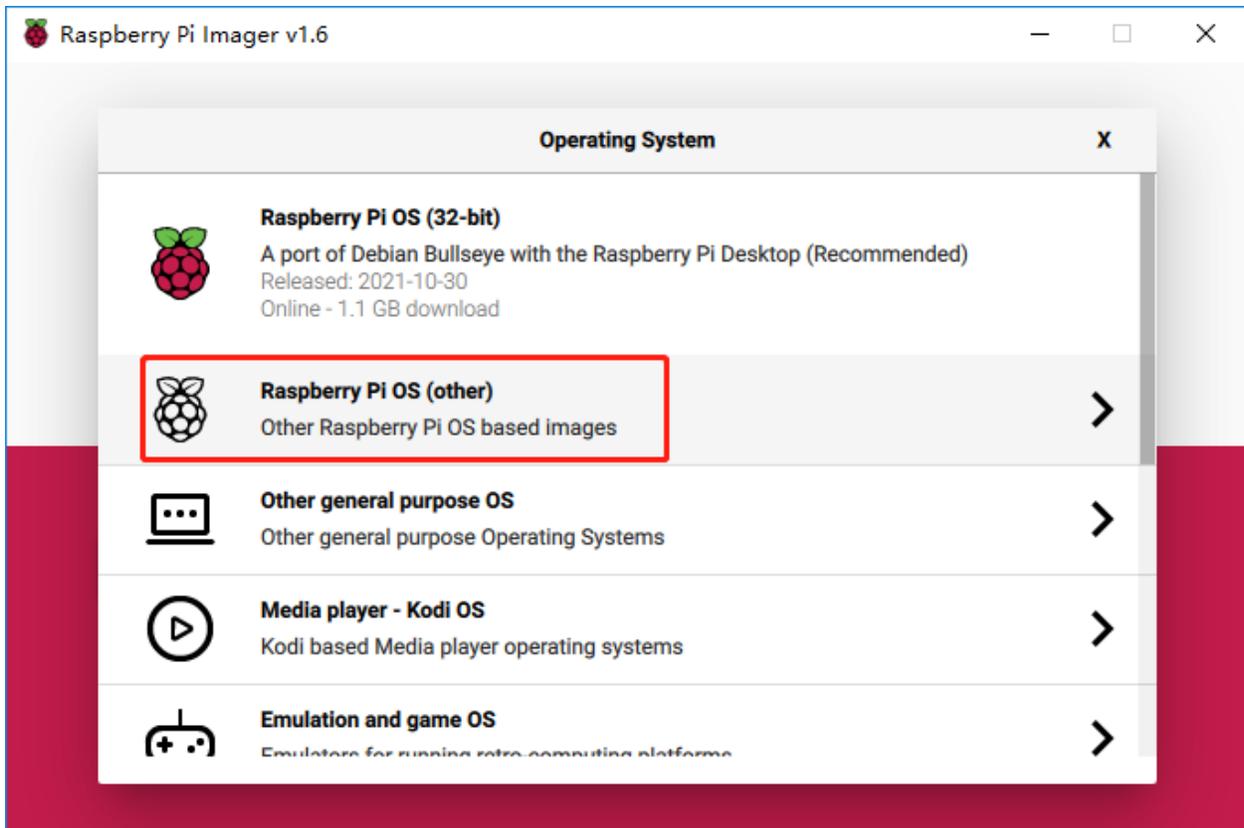
第 3 步

将 SD 卡插入计算机或笔记本电脑的 SD 卡插槽。

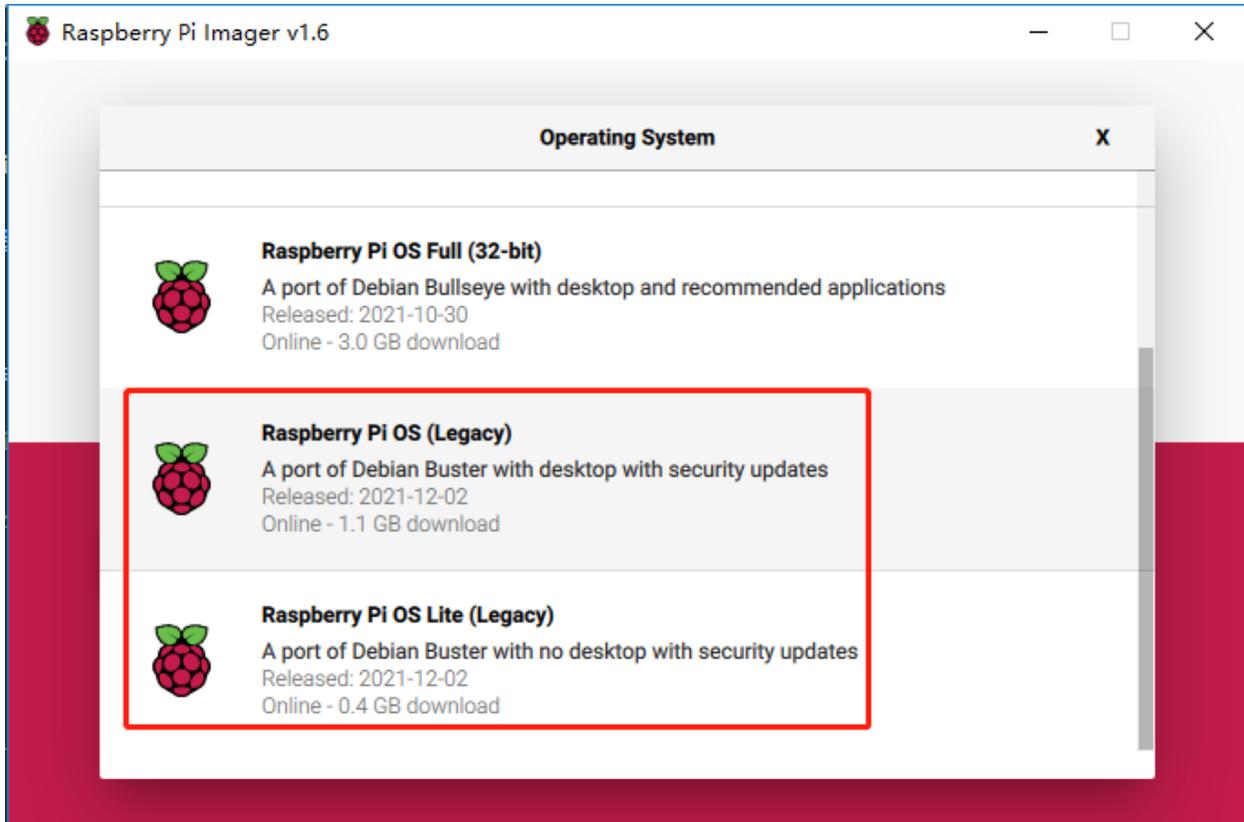
第 4 步

警告： Raspberry Pi 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 -> Raspberry Pi OS(other)**。

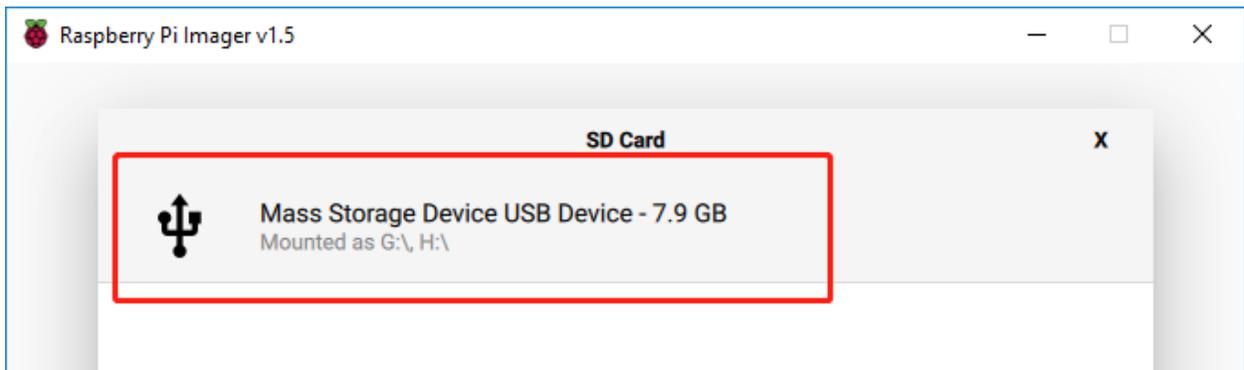


将新打开的页下拉到最后面，你会看到 **Raspberry Pi OS(Legacy)** 和 **Raspberry Pi OS Lite(Legacy)**，这 2 个是对 Debian Buster 安全更新，它们之间的区别是带不带桌面。建议安装 **Raspberry Pi OS(Legacy)**，这个带桌面的系统。



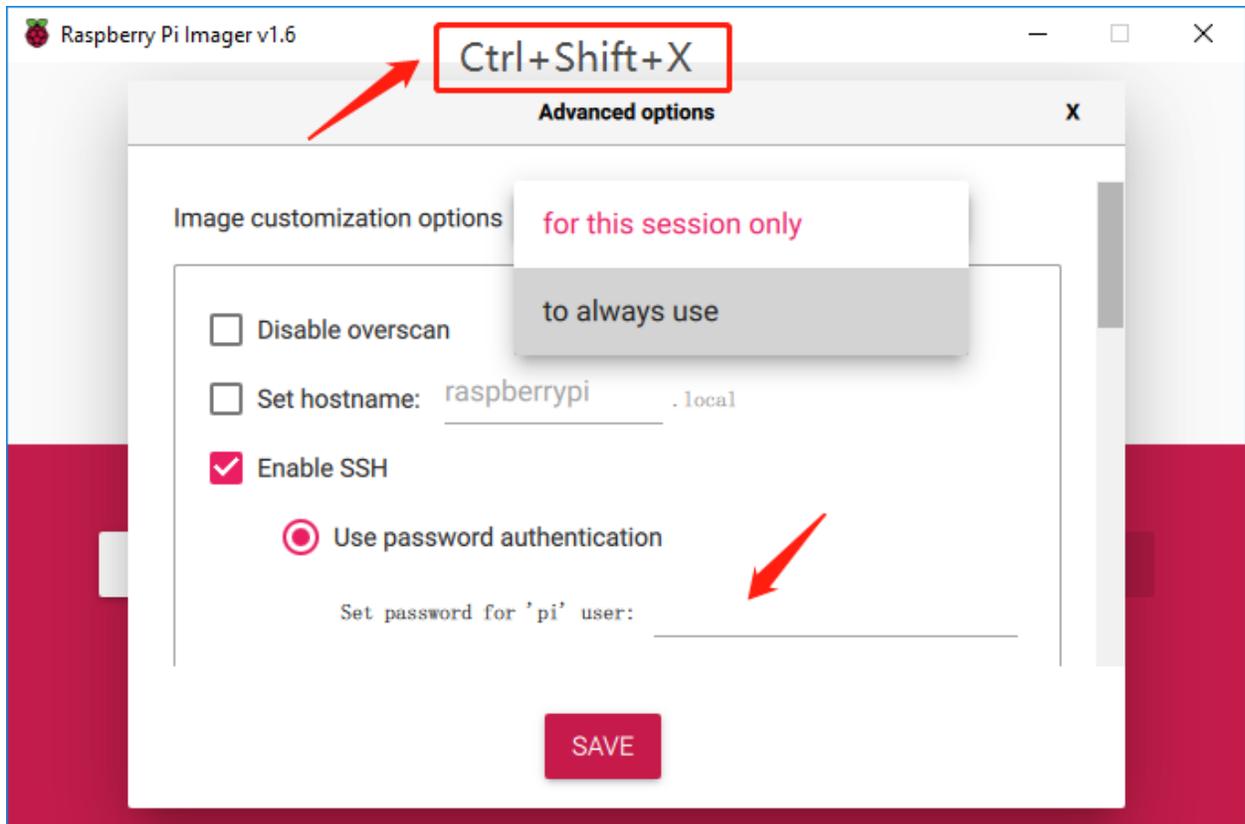
第 5 步

选择您正在使用的 SD 卡。



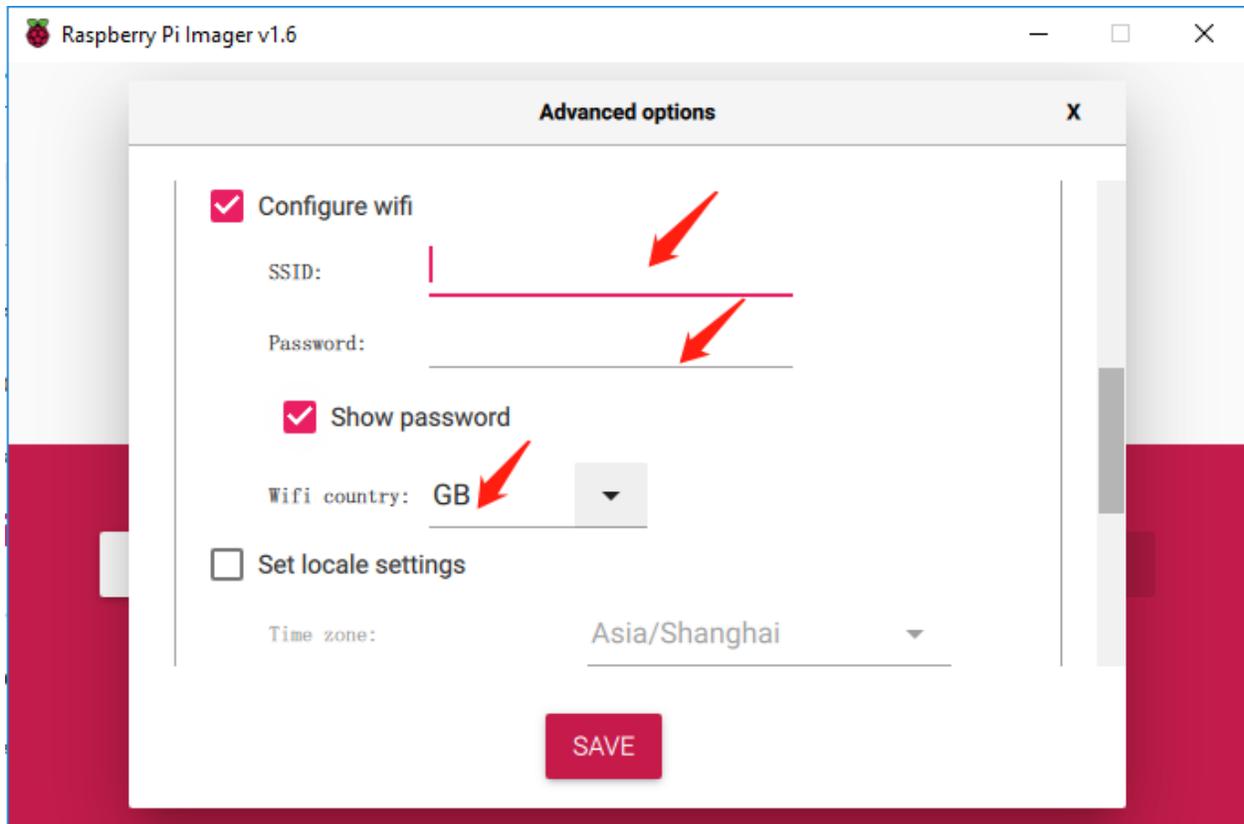
第 6 步

按 **Ctrl+Shift+X** 或者点击 **设置**按钮来打开 **高级选项**页面启用 SSH 和配置 wifi，这 2 项必须设置，其他取决于你的选择。您可以选择始终使用此图像自定义选项。



然后向下滚动以完成 wifi 配置并单击 **SAVE** 。

备注: **wifi country** 选择 CN。



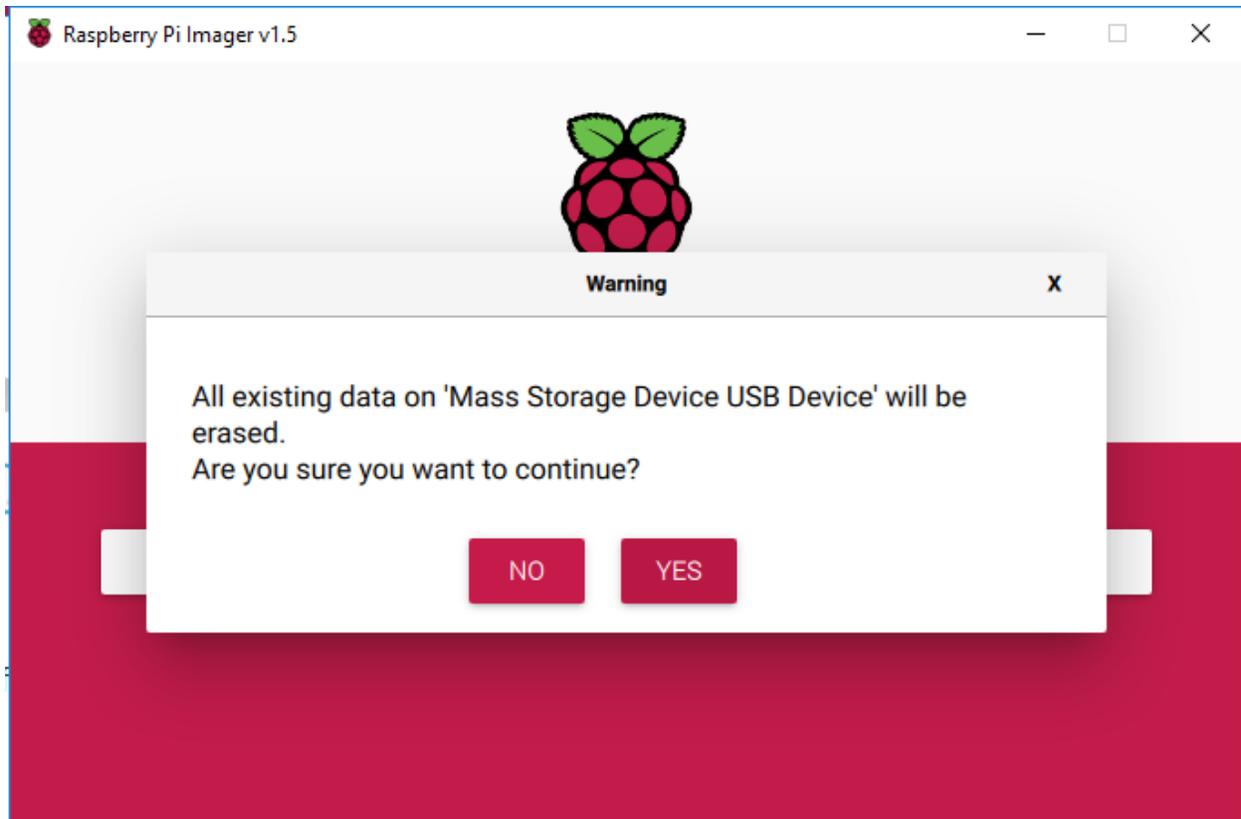
第7步

单击 **WRITE** 按钮。



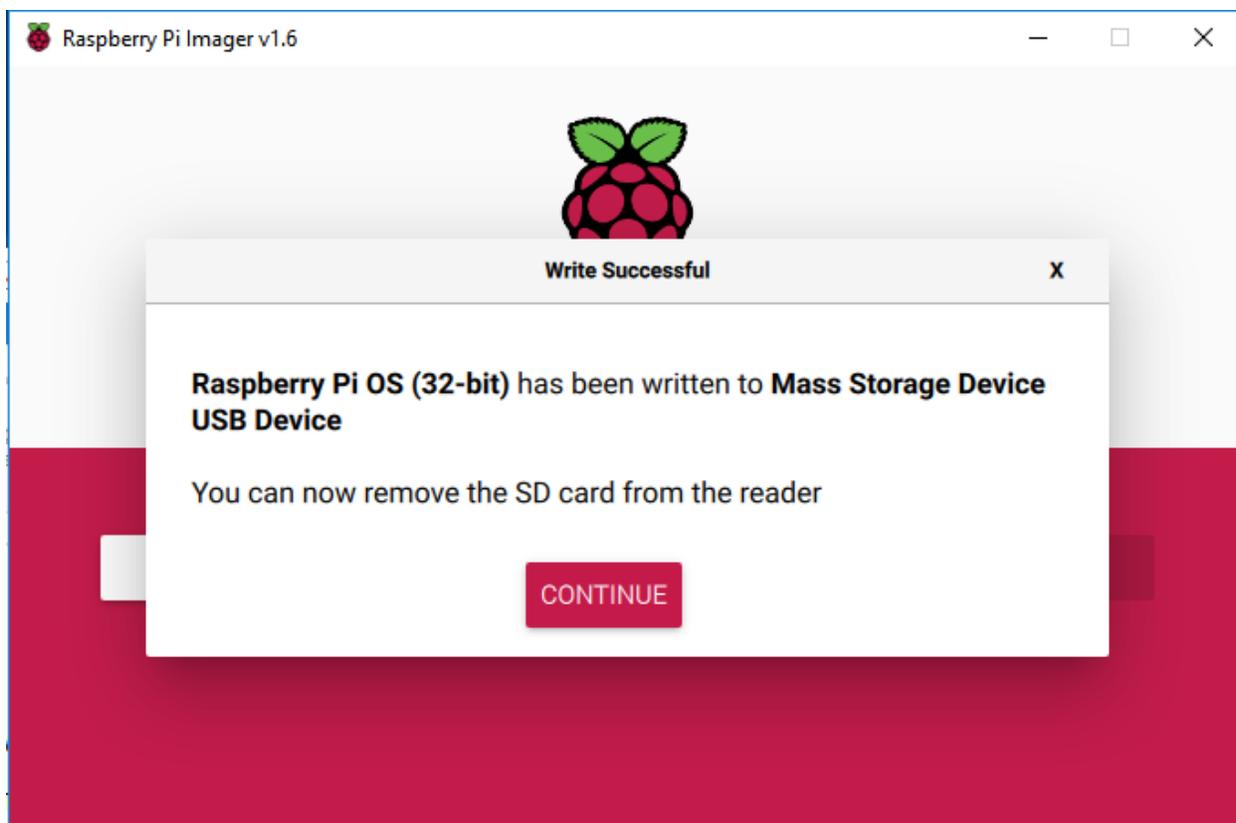
第 8 步

如果您的 SD 卡上当前有任何文件，您可能希望先备份这些文件以防止永久丢失它们。如果没有要备份的文件，请单击 **YES**。



第9步

等待一段时间后，会出现如下窗口，代表写入完成。



4.1.3 设置你的树莓派

如果你有屏幕

如果你有一个屏幕，你会很容易在屏幕上操作树莓派。

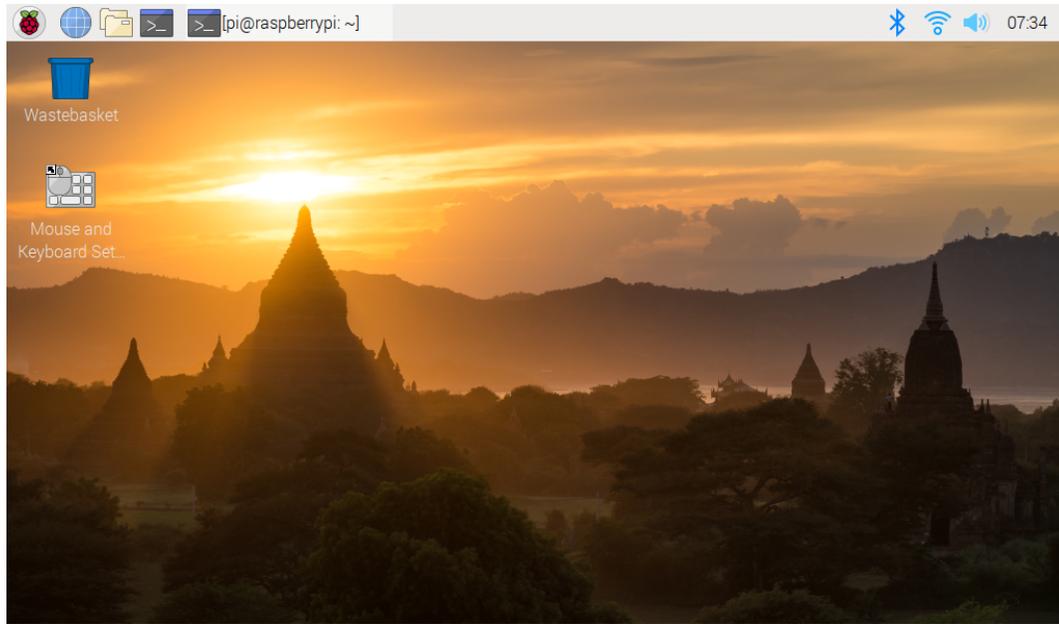
必需组件

任意树莓派	1 * 电源适配器
1 * Micro SD 卡	1 * 屏幕电源适配器
1 * HDMI 线	1 * 屏幕
1 * 鼠标	1 * 键盘

1. 将您在 Raspberry Pi OS 上设置的 SD 卡插入到树莓派底部的 micro SD 卡插槽中。
2. 插入鼠标和键盘。
3. 将屏幕连接到树莓派的 HDMI 端口，并确保您的屏幕已插入壁式插座并已打开。

备注：如果您使用的是树莓派 4，则需要将屏幕连接到 HDMI0（最靠近电源输入端口）。

4. 使用电源适配器为树莓派供电。几秒钟后，将显示 Raspberry Pi OS 桌面。



如果你没有屏幕

如果没有显示器，可以远程登录树莓派，但在此之前，您需要获取树莓派的 IP。

获取 IP 地址

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

如果您有权限登录路由器（如家庭网络），您可以在路由器的管理界面查看分配给树莓派的地址。

树莓派操作系统的默认主机名是 **raspberrypi**，你需要找到它。（如果你使用的是 ArchLinuxARM 系统，请找 **alarmpi**。）

2. 网段扫描

您还可以使用网络扫描来查找树莓派的 IP 地址。您可以应用软件，**Advanced IP scanner** 等。

扫描设置的 IP 范围，将显示所有已连接设备的名称。同样，树莓派操作系统的默认主机名是 **raspberrypi**，如果你没有修改过的话。

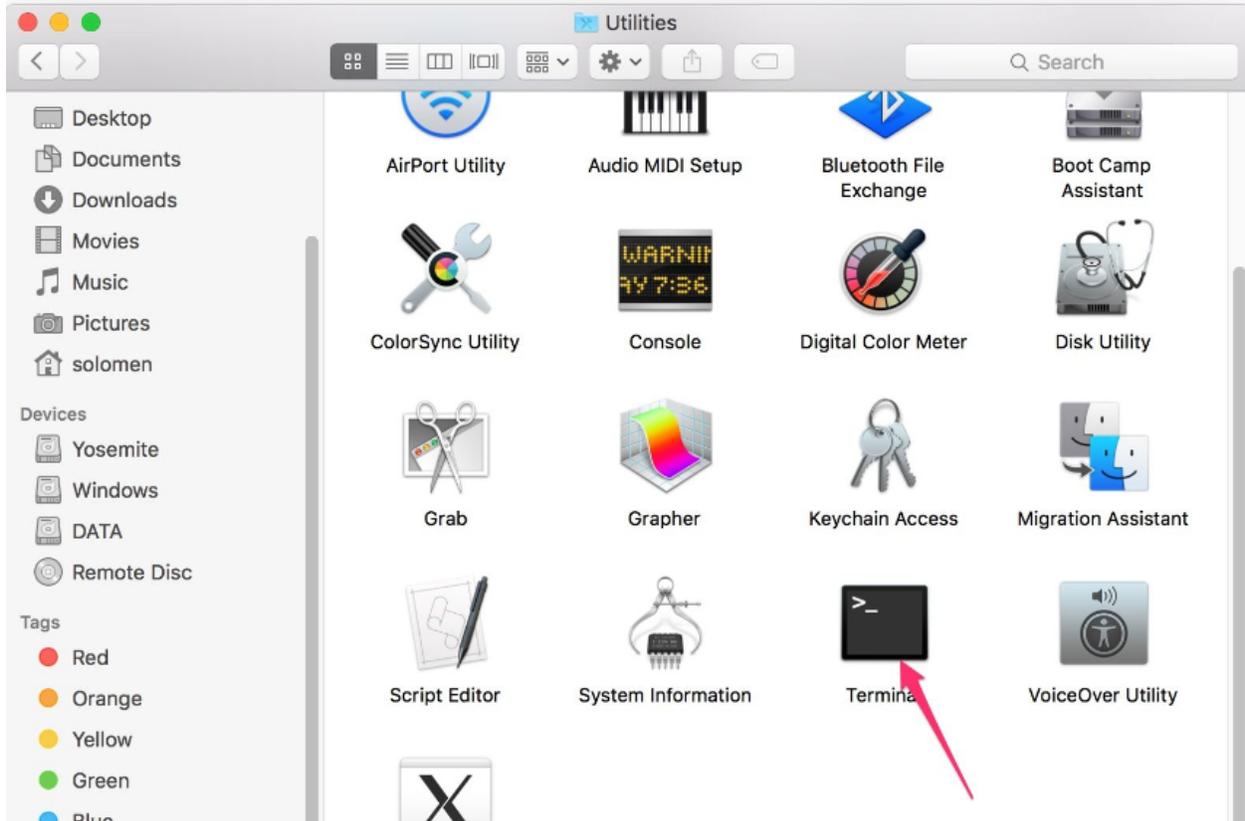
使用 SSH 远程控制

我们可以通过应用 SSH 打开树莓派的 Bash Shell。Bash 是 Linux 的标准默认 shell。Shell 本身是一个用 C 编写的程序，它是连接客户和 Unix/Linux 的桥梁。此外，它可以帮助完成大部分所需的工作。

适用于 Linux 或 Mac OS X 用户

第 1 步

进入 **Applications -> Utilities**，找到 **Terminal**，然后打开它。



第 2 步

输入 `ssh pi@ip_address`。"pi" 是您的用户名，"ip_address" 是您的 IP 地址。例如：

```
ssh pi@192.168.18.197
```

第 3 步

输入 "yes"。

```
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)?
```

第 4 步

输入密码，默认密码为 **raspberrypi**。

```
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: ?
```

第5步

我们现在已连接树莓派，并准备进行下一步。

```
1. pi@raspberrypi: ~ (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:
Linux raspberrypi 4.9.80-v7+ #1098 SMP Fri Mar 9 19:11:42 GMT 2018 armv7l

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 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:~ $
```

备注：当您输入密码时，字符不会相应显示在窗口中，这是正常的。您只需要输入正确的密码即可。

对于 Windows 用户

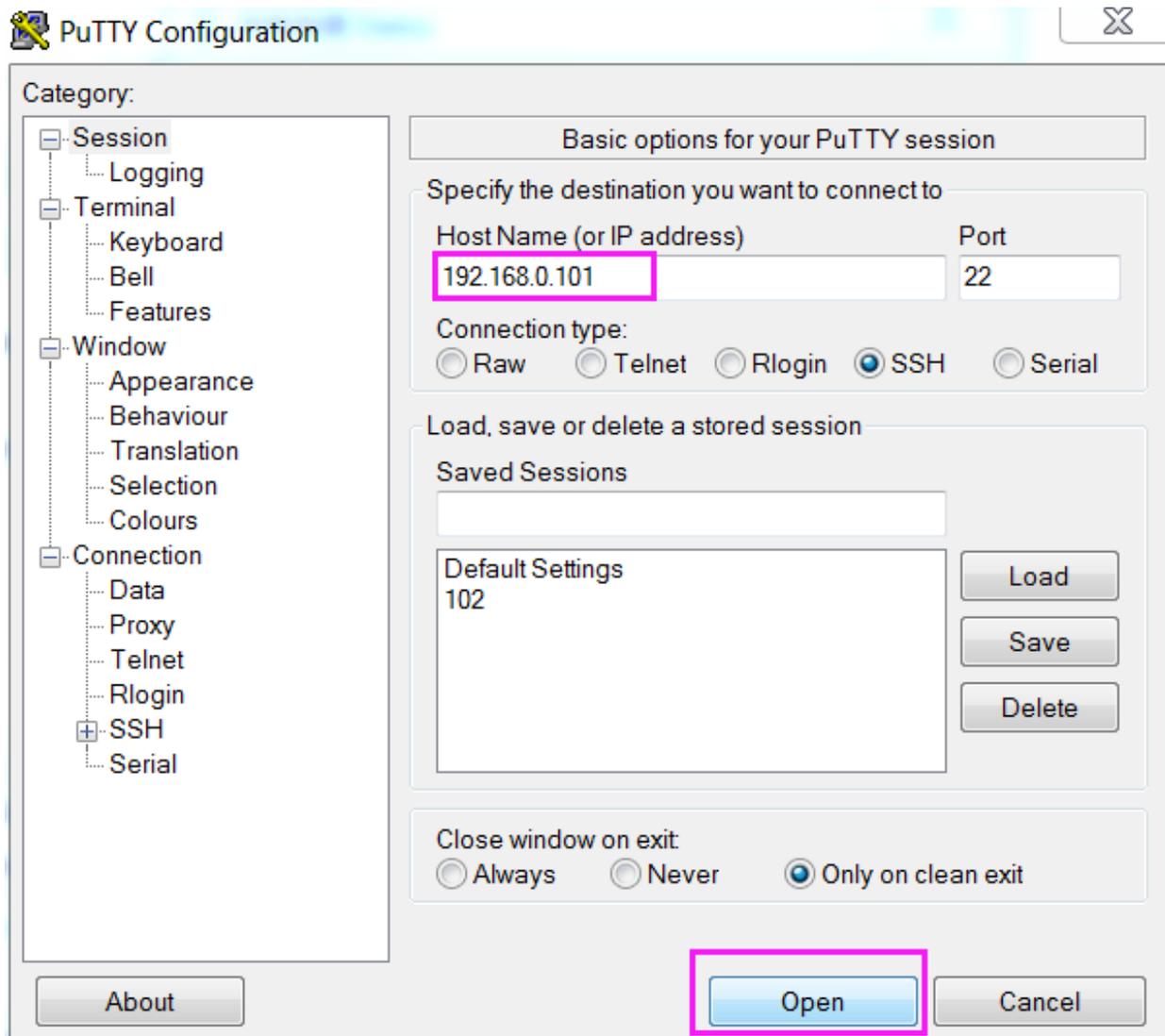
如果您是 Windows 用户，则可以通过某些软件的应用程序使用 SSH。在这里，我们推荐 **PuTTY**。

第 1 步

下载 PuTTY。

第 2 步

打开 PuTTY 并单击左侧树状结构上的 **Session**。在 **Host Name (or IP address)** 下的文本框中输入 RPi 的 IP 地址，**端口**下输入 **22**（默认为 22）。



第 3 步

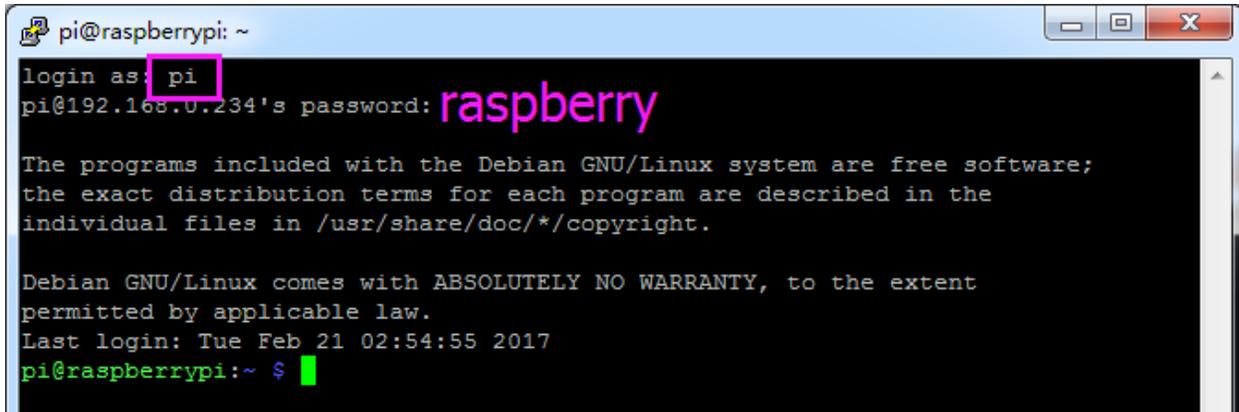
点击 **Open**。注意第一次用 IP 地址登录树莓派时，会提示安全提示。只需单击 **Yes**。

第四步

当 PuTTY 窗口提示 **login as:** 时，输入“pi”（树莓派的用户名）。提示 **password:** 时候，输入“raspberrypi”（默认密码）。

备注： 当您输入密码时，字符不会相应显示在窗口中，这是正常的。您只需要输入正确的密码即可。

如果 PuTTY 旁边出现 **inactive**，则表示连接已断开，需要重新连接。



```
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:~$
```

第 5 步

在这里，我们连接了树莓派，是时候进行下一步了。

备注：如果您对使用命令窗口控制树莓派不满意，还可以使用远程桌面功能，它可以帮助我们轻松管理树莓派中的文件。

有关如何执行此操作的详细信息，请参阅[远程桌面](#)。

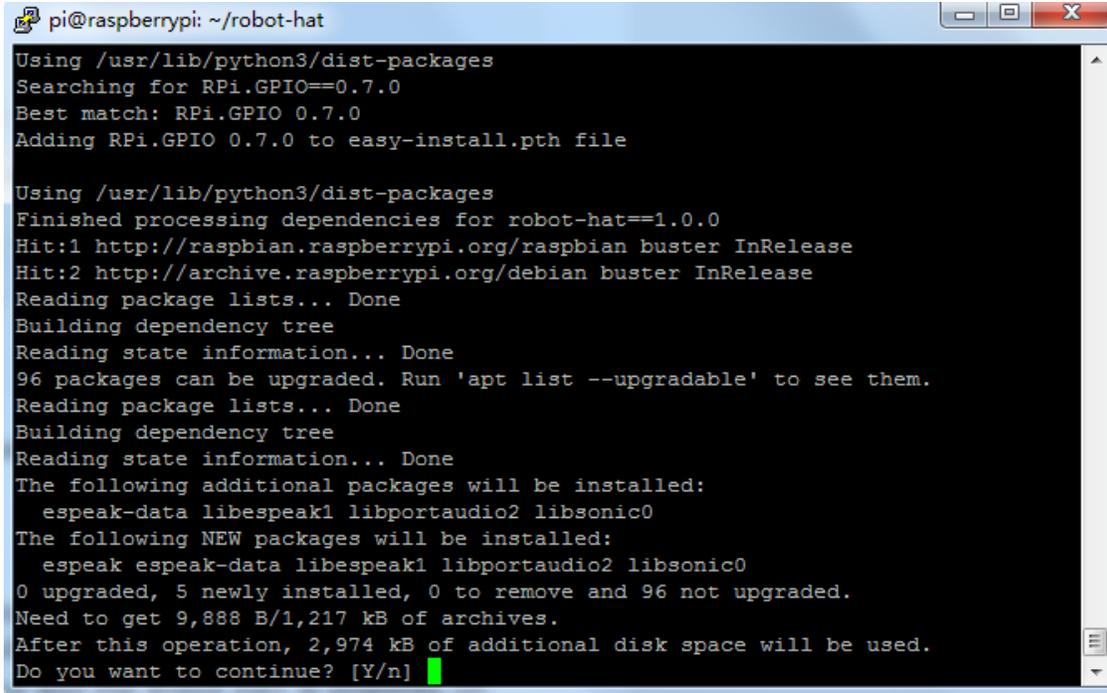
4.1.4 下载代码和库

首先通过在命令行中使用 `git clone` 下载库文件 `robot-hat`。

备注：在下面的安装过程中，可能会由于网络问题导致失败，你需要参考[apt](#) 和 [pip](#) 更换国内源 来修改配置。

```
cd /home/pi/  
git clone https://gitee.com/sunfounder/robot-hat.git  
cd robot-hat  
sudo python3 setup.py install
```

备注：运行 `setup.py` 将下载一些必要的组件。由于网络问题，您可能无法下载成功。您可能需要重新下载。在这种情况下，输入 `Y` 并按 `Enter`。



```
pi@raspberrypi: ~/robot-hat
Using /usr/lib/python3/dist-packages
Searching for RPi.GPIO==0.7.0
Best match: RPi.GPIO 0.7.0
Adding RPi.GPIO 0.7.0 to easy-install.pth file

Using /usr/lib/python3/dist-packages
Finished processing dependencies for robot-hat==1.0.0
Hit:1 http://raspbian.raspberrypi.org/raspbian buster InRelease
Hit:2 http://archive.raspberrypi.org/debian buster InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
96 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  espeak-data libespeak1 libportaudio2 libsonic0
The following NEW packages will be installed:
  espeak espeak-data libespeak1 libportaudio2 libsonic0
0 upgraded, 5 newly installed, 0 to remove and 96 not upgraded.
Need to get 9,888 B/1,217 kB of archives.
After this operation, 2,974 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

然后下载代码并安装 pislloth 库。

```
cd /home/pi/
git clone -b v2.0 https://gitee.com/sunfounder/pislloth.git
cd pislloth
sudo python3 setup.py install
```

这一步需要一点时间，所以请耐心等待。

最后需要运行脚本 `i2samp.sh` 安装 i2s 功放所需的组件，否则它可能会没有声音。

```
cd /home/pi/pislloth
sudo bash i2samp.sh
```

```
pi@raspberrypi: ~/pisloth
File "/usr/local/lib/python3.7/dist-packages/robot_hat-1.0.0-py3.7.egg/robot_hat/robot.py", line 91, in servo_move
    time.sleep(step_delay)
KeyboardInterrupt
pi@raspberrypi:~/pisloth/examples $ cd ..
pi@raspberrypi:~/pisloth $ sudo bash i2samp.sh
Support for your operating system is experimental. Please visit
forums.adafruit.com if you experience issues with this product.

This script will install everything needed to use
i2s amplifier

--- Warning ---

Always be careful when running scripts and commands
copied from the internet. Ensure they are from a
trusted source.

If you want to see what this script does before
running it, you should run:
    \curl -sS github.com/adafruit/Raspberry-Pi-Installer-Scripts/i2samp
Do you wish to continue? [y/N] █
```

输入 y 并按 Enter 继续运行脚本。

```
pi@raspberrypi: ~/pisloth
running it, you should run:
    \curl -sS github.com/adafruit/Raspberry-Pi-Installer-Scripts/i2samp
Do you wish to continue? [y/N] y
Checking hardware requirements...

Adding Device Tree Entry to /boot/config.txt
dtoverlay=hifiberry-dac
dtoverlay=i2s-mmap

Commenting out Blacklist entry in
/etc/modprobe.d/raspi-blacklist.conf

Disabling default sound driver
Configuring sound output

Installing aplay systemd unit

You can optionally activate '/dev/zero' playback in
the background at boot. This will remove all
popping/clicking but does use some processor time.

Activate '/dev/zero' playback in background? [RECOMMENDED] [y/N] █
```

输入 y 并按 Enter 让 /dev/zero 在后台运行。

```
pi@raspberrypi: ~/pisloth
/etc/modprobe.d/raspi-blacklist.conf

Disabling default sound driver
Configuring sound output

Installing aplay systemd unit

You can optionally activate '/dev/zero' playback in
the background at boot. This will remove all
popping/clicking but does use some processor time.

Activate '/dev/zero' playback in background? [RECOMMENDED] [y/N] y

Created symlink /etc/systemd/system/multi-user.target.wants/aplay.service → /etc
/systemd/system/aplay.service.

All done!

Enjoy your new i2s amplifier!

Some changes made to your system require
your computer to reboot to take effect.

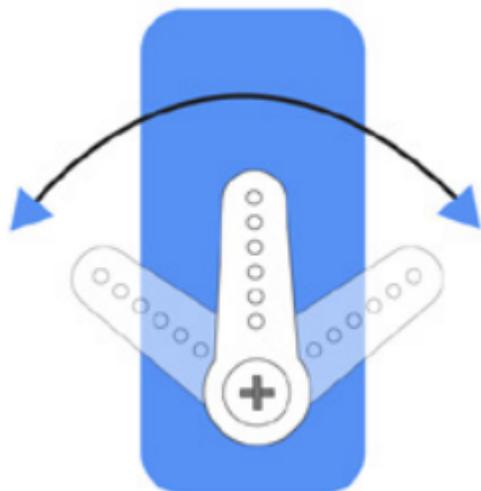
Would you like to reboot now? [y/N] █
```

输入 y 并按 Enter 重新启动机器。

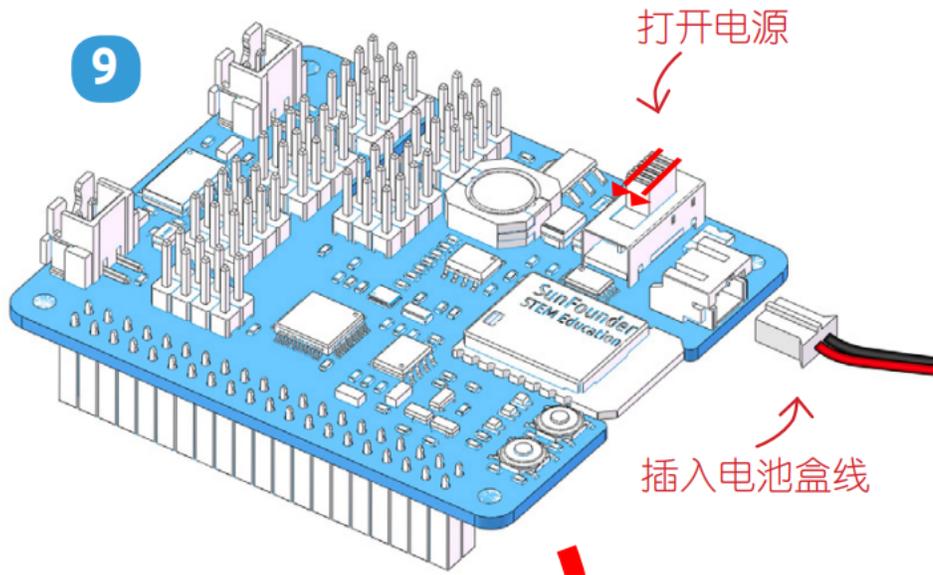
备注：如果重启后没有声音，你需要多次运行 `i2samp.sh`。

4.1.5 舵机调零

1. 为确保舵机已正确设置为 0° ，首先将摇臂轻轻插入舵机轴，然后将摇臂轻轻旋转到不同的角度。



2. 按照组装折页上的提示，插入电池盒线，将电源开关拨向 ON 的位置。等待 1-2 分钟，会有声音提示树莓派启动成功。

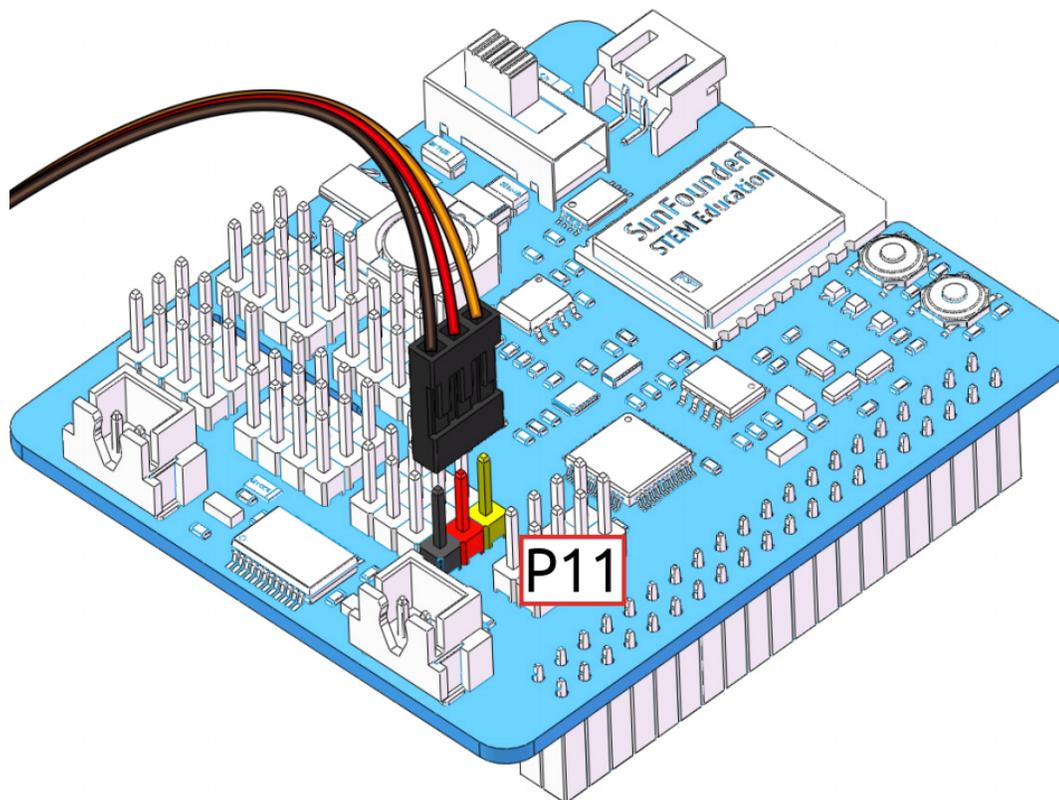


3. 现在，运行 examples/ 文件夹中的 servo_zeroing.py .

```
cd /home/pi/picrawler/examples
sudo python3 servo_zeroing.py
```

备注：如果报错，请尝试重新启用树莓派的 I2C 端口，请参阅：[I2C 配置](#)。

4. 接下来，将舵机线插入 P11 端口，如下所示：



5. 此时你会看到舵机臂转动到特定的位置 (0°)。如果伺服臂没有返回到 0° ，请按 **RST** 按钮重新启动 Robot HAT。
6. 现在你就可以按照组装折页上的指示继续安装。

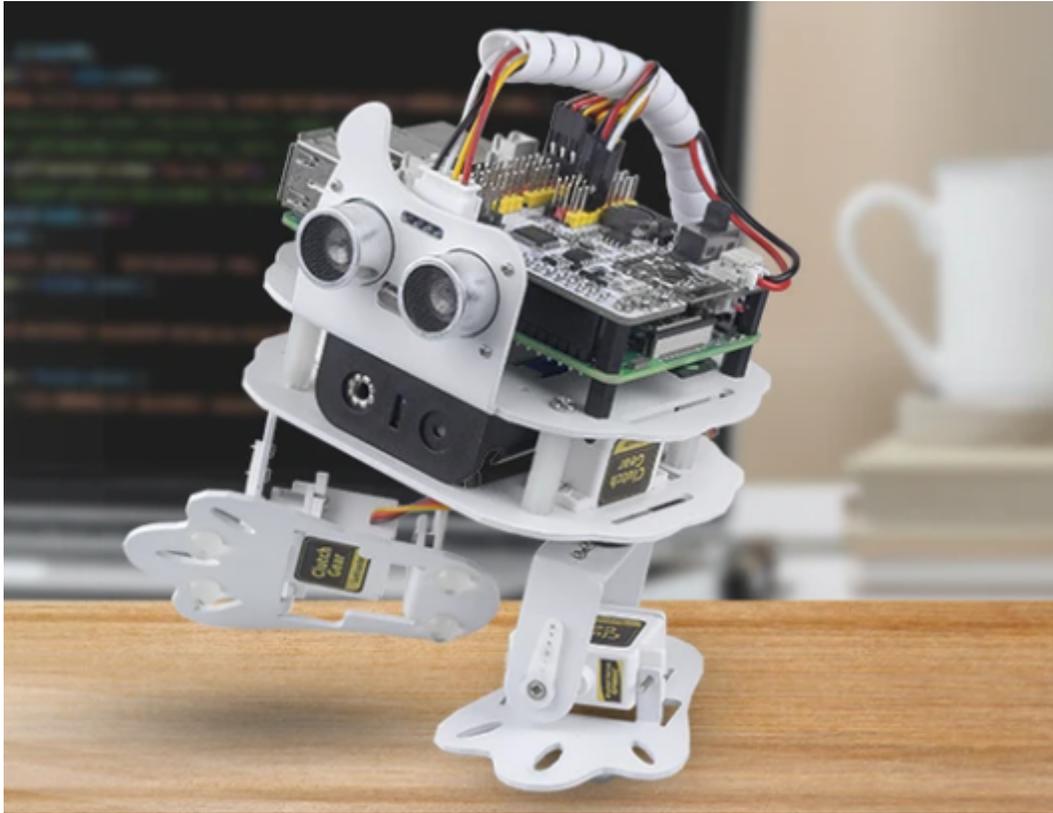
备注:

- 在用舵机螺丝固定该舵机前，不要拔出该舵机线，可在固定完之后拔出。
- 不要在上电情况下随意转动舵机以免损坏；如果舵机轴插入的角度不对，需把舵机拔出再重新插入。
- 在组装每个舵机之前，都需要将舵机电缆插入 P11 并打开电源以将它的角度设置为 0° 。

组装完成后，您可以尝试运行下面的项目。

4.2 移动

这是第一个项目。PiSloth 已经醒来，它可以自由移动。



运行代码

```
cd /home/pi/pislloth/examples
sudo python3 move.py
```

运行代码后，你会看到 PiSloth 向左移动 7 步，向前移动 5 步，向右移动 7 步，向前移动 5 步。

代码

```
from pislloth import Sloth

sloth = Sloth([1,2,3,4])
sloth.set_offset([0,0,0,0])

def main():
    sloth.do_action('turn left', 7, 90)
    sloth.do_action('forward', 5, 90)
    sloth.do_action('turn right', 7, 90)
    sloth.do_action('forward', 5, 90)

if __name__ == "__main__":
    while True:
        main()
```

这个怎么运作？

首先, 在 `pisloth` 库中导入 `Sloth` 类, 它包含 `PiSloth` 的所有操作和实现它们的函数。

```
from pisloth import Sloth
```

然后实例化 `Sloth` 类。

```
sloth = Sloth([1,2,3,4])
sloth.set_offset([0,0,0,0])
```

最后使用 `sloth.do_action()` 函数使 `PiSloth` 移动。

```
sloth.do_action('turn left', 7, 90)
sloth.do_action('forward', 5, 90)
sloth.do_action('turn right', 7, 90)
sloth.do_action('forward', 5, 90)
```

一般来说, `PiSloth` 的所有动作都可以通过 `sloth.do_action()` 函数来实现. 它有四个参数:

- `motion_name` 是具体动作的名称, 包括: `forward`, `turn right`, `turn left`, `backward`, `stand`, `moon walk left`, `moon walk right`, `hook`, `big swing`, `swing`, `walk boldly`, `walk backward boldly`, `walk shyly`, `walk backward shyly`, `stomp right`, `stomp left`, `close`, `open`, `tiptoe left`, `tiptoe right`, `fall left`, `fall right`.
- `step` 表示每个动作执行的次数, 默认为 1。
- `speed` 表示动作的速度, 默认为 50, 范围为 0~100。
- `bpm` 表示节奏, 在后面跳舞示例中会用到。

备注: 您可以通过 [Filezilla](#) 软件 向 `musics` 或 `sounds` 文件夹添加不同的音效或音乐。

4.3 跳舞

当当当, 有请 `PiSloth` 向大家展示他新学的舞蹈。



备注：您可以为您的 PiSloth 下载并打印卡通面具。

- 卡通面具 (.pdf)

运行代码

```
cd /home/pi/pislloth/examples
sudo python3 dancing.py
```

整个舞蹈分为 2 个部分，PiSloth 会随着音乐完成这 2 个部分。如果你不暂停代码，它会重复跳舞一直跳下去。

代码

```
from pislloth import Sloth
from robot_hat import Music
from robot_hat import Ultrasonic
from robot_hat import Pin
import time
import os

music = Music()

sloth = Sloth([1,2,3,4])
sloth.set_offset([0,0,0,0])

def main():
```

(续下页)

```

music.background_music('./musics/india-Arulo.mp3')
music.music_set_volume(20)
sloth.do_action('stomp left',3,bpm=129)
sloth.do_action('stomp right',3,bpm=129)
sloth.do_action('moon walk left',3,bpm=129)
sloth.do_action('moon walk right',3,bpm=129)
for i in range(3):
    sloth.do_action('swing',1,bpm=129)
    sloth.do_action('stand',1,bpm=129)
for i in range(3):
    sloth.do_action('close',1,bpm=129)
    sloth.do_action('stand',1,bpm=129)
    sloth.do_action('open',1,bpm=129)
    sloth.do_action('stand',1,bpm=129)
sloth.do_action('tiptoe left',2,bpm=129)
sloth.do_action('tiptoe right',2,bpm=129)

sloth.do_action('stomp left',3,bpm=129)
sloth.do_action('stomp rihgt',3,bpm=129)
sloth.do_action('moon walk left',3,bpm=129)
sloth.do_action('moon walk right',3,bpm=129)
for i in range(3):
    sloth.do_action('hook',1,bpm=129)
    sloth.do_action('stand',1,bpm=129)
for i in range(4):
    sloth.do_action('swing',1,bpm=129)
    sloth.do_action('big swing',1,bpm=129)
    sloth.do_action('swing',1,bpm=129)
    sloth.do_action('stand',1,bpm=129)

sloth.do_action('tiptoe right',2,bpm=129)
sloth.do_action('stand',2,bpm=129)

music.music_stop()
time.sleep(10)

if __name__ == "__main__":
    while True:
        main()

```

这个怎么运作？

您可以通过导入以下库让 PiSloth 播放音乐。

```
from robot_hat import TTS, Music
```

播放 pisloth/examples/musics 目录中的背景音乐，音量设置为 20。您也可以通过 *Filezilla* 软件将音乐添加到 musics 文件夹中。

```
music.background_music('./musics/india-Arulo.mp3')
music.music_set_volume(20)
```

一般情况下，PiSloth 的所有动作都可以通过该 `sloth.do_action()` 函数实现。它有四个参数：

- motion_name 是具体动作的名称，包括: forward, turn right, turn left, backward, stand, moon walk left, moon walk right, hook, big swing, swing, walk boldly, walk backward boldly, walk shyly, walk backward shyly, stomp rihgt, stomp left, close, open, tiptoe left, tiptoe right, fall left, fall right.
- step 表示每个动作执行的次数，默认为 1。
- speed 表示动作的速度，默认为 50，范围为 0~100。
- bpm 表示节奏，这里的 bpm 参数影响 PiSloth 运动的间隔时间。值越高，间隔时间越短。当我们通过 bpm 计算器知道一首歌的节拍时，我们可以让 PiSloth 随着音乐跳舞。

备注：您可以通过 *Filezilla* 软件向 musics 或 sounds 文件夹添加不同的音效或音乐。

4.4 避障

在这个项目中，PiSloth 将使用超声波模块来检测前方的障碍物。当 PiSloth 检测到障碍物时，它会发出信号并寻找另一个方向前进。

运行代码

```
cd /home/pi/pislloth/examples
sudo python3 avoid.py
```

代码运行后，PiSloth 会向前走。如果检测到前方障碍物的距离小于 10cm，就会停车并发出警告，然后左转。如果左转后方向没有障碍物或障碍物距离大于 10，则继续向前移动。

代码

```
from pislloth import Sloth
from robot_hat import TTS, Music
from robot_hat import Ultrasonic
from robot_hat import Pin
import time
import os

tts = TTS()
music = Music()

sloth = Sloth([1,2,3,4])
sloth.set_offset([0,0,0,0])
sonar = Ultrasonic(Pin("D2"), Pin("D3"))

alert_distance = 10

def main():
    distance = sonar.read()
    if distance < 0:
        pass
    elif distance <= alert_distance:
        try:
            music.sound_effect_threading('./sounds/sign.wav')
        except Exception as e:
            print(e)
            sloth.do_action('hook', 1,95)
```

(续下页)

```

    time.sleep(0.5)
    sloth.do_action('stand', 1,95)
    time.sleep(0.5)
    sloth.do_action('turn left',7,90)
    sloth.do_action('stand', 1,95)
    time.sleep(0.2)
    else :
        sloth.do_action('forward', 1,90)

if __name__ == "__main__":
    while True:
        main()

```

这个怎么运作？

您可以通过导入 Ultrasonic 类来实现距离检测。

```
from robot_hat import Ultrasonic
```

然后初始化超声波引脚。

```
sonar = Ultrasonic(Pin("D2") ,Pin("D3"))
```

这里是主程序。

- 读取超声波模块检测到的 distance (距离)，过滤掉小于 0 的值（当超声波模块距离障碍物太远或无法正确读取数据时，会出现“距离 <0”）。
- 当 distance 小于等于 alert_distance (之前设置的阈值，即 10) 时，播放音效 sign.wav。PiSloth 依次做 hook (勾脚), stand (站立), left turn (左转) 和 stand (右转)。
- 当 distance 大于 alert_distance 时，PiSloth 会向前移动。

```

distance = sonar.read()
if distance < 0:
    pass
elif distance <= alert_distance:
    try:
        music.sound_effect_threading('./sounds/sign.wav')
    except Exception as e:
        print(e)
    sloth.do_action('hook', 1,95)
    time.sleep(0.5)
    sloth.do_action('stand', 1,95)
    time.sleep(0.5)
    sloth.do_action('turn left',7,90)
    sloth.do_action('stand', 1,95)
    time.sleep(0.2)
else :
    sloth.do_action('forward', 1,90)

```

备注：您可以通过 *Filezilla* 软件向 musics 或 sounds 文件夹添加不同的音效或音乐。

4.5 别碰我

如果它看到你伸手想去摸它，它会生气并离你远远的。

运行代码

```
cd /home/pi/pislloth/examples
sudo python3 dont_touch_me.py
```

代码

```
from pislloth import Sloth
from robot_hat import Music
from robot_hat import Ultrasonic
from robot_hat import Pin
import time
import os

music = Music()

sloth = Sloth([1,2,3,4])
sloth.set_offset([0,0,0,0])
sonar = Ultrasonic(Pin("D2"), Pin("D3"))

alert_distance = 20

def main():
    distance = sonar.read()
    print(distance)
    if distance <= alert_distance :
        try:
            music.sound_effect_threading('./sounds/talk3.wav')
        except Exception as e:
            print(e)
        sloth.do_action('backward', 2, 90)
    else:
        sloth.do_action('stand', 1, 90)
        time.sleep(1)

if __name__ == "__main__":
    while True:
        main()
```

这个怎么运作？

实例化 Music, Sloth 和 Ultrasonic 以供使用。

```
music = Music()

sloth = Sloth([1,2,3,4])
sloth.set_offset([0,0,0,0])
sonar = Ultrasonic(Pin("D2"), Pin("D3"))
```

这里是主程序。

- 读取超声波模块检测到的数据 distance 并打印出来。

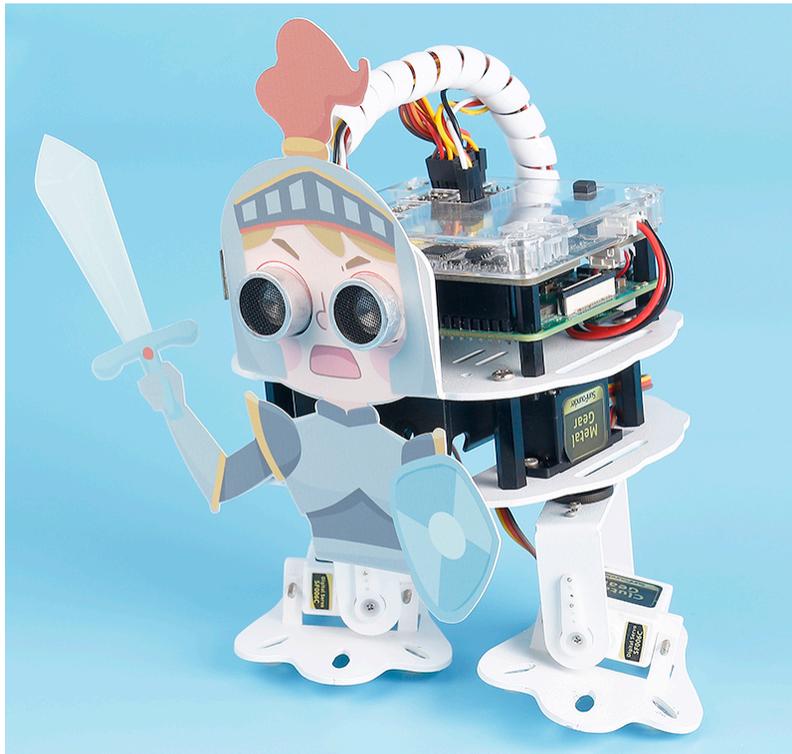
- 当 distance 小于或等于 alert_distance（之前设置的阈值，即 20）时，播放音效 talk3.wav 并向后移动。
- 当 distance 大于 alert_distance 时，PiSloth 将站立。

```
distance = sonar.read()
print(distance)
if distance <= alert_distance :
    try:
        music.sound_effect_threading('./sounds/talk3.wav')
    except Exception as e:
        print(e)
    sloth.do_action('backward', 2, 90)
else:
    sloth.do_action('stand', 1, 90)
    time.sleep(1)
```

备注：您可以通过 *Filezilla* 软件 向 musics 或 sounds 文件夹添加不同的音效或音乐。

4.6 战斗吧！战士！

在这里，PiSloth 是一个勇敢的战士，当它出现在敌人面前时，它会发出咆哮冲向敌人。



备注：您可以为您的 PiSloth 下载并打印卡通面具。

- 卡通面具 (.pdf)
-

运行代码

```
cd /home/pi/pislloth/examples
sudo python3 lets_fight.py
```

代码运行后，PiSloth 会不断检测障碍物的距离，当距离在 5 到 40 之间时，PiSloth 会发出轰鸣声并向前冲去；当障碍物的距离小于 5 时，PiSloth 将停止。

代码

```
from pislloth import Sloth
from robot_hat import Music
from robot_hat import Ultrasonic
from robot_hat import Pin
import time
import os

music = Music()

sloth = Sloth([1,2,3,4])
sloth.set_offset([0,0,0,0])
sonar = Ultrasonic(Pin("D2"), Pin("D3"))

alert_distance = 40
contact_distance = 5

def main():
    distance = sonar.read()
    if distance <= alert_distance and distance >= contact_distance :
        try:
            music.sound_effect_play('./sounds/battle.wav')
            music.background_music('./musics/attack.mp3')
            music.music_set_volume(20)
        except Exception as e:
            print(e)
        while True:
            distance = sonar.read()
            print(distance)
            if distance < 0:
                continue
            if distance <= contact_distance:
                break
            sloth.do_action('forward', 1,90)
        sloth.do_action('stand', 1, 90)
        time.sleep(1)

if __name__ == "__main__":
    while True:
        main()
```

这个怎么运作？

这里是主程序。

- 读取 distance 超声波模块检测到的值，过滤掉小于 0 的值（当超声波模块距离障碍物太远或无法正确读取数据时，distance 将会是一个小于 0 的无效值）。
- 当 distance 在 5 和 40 之间，PiSloth 将发出 warning.wav 和 attack.mp3 音效并向前移动。

- 当 distance 小于 5 时, PiSloth 将保持 stand 位置。

```
distance = sonar.read()
if distance <= alert_distance and distance >= contact_distance :
    try:
        music.sound_effect_play('./sounds/battle.wav')
        music.background_music('./musics/attack.mp3')
        music.music_set_volume(20)
    except Exception as e:
        print(e)
    while True:
        distance = sonar.read()
        print(distance)
        if distance < 0:
            continue
        if distance <= contact_distance:
            break
        sloth.do_action('forward', 1, 95)
sloth.do_action('stand', 1, 90)
time.sleep(1)
```

备注: 您可以通过 *Filezilla* 软件 向 musics 或 sounds 文件夹添加不同的音效或音乐。

4.7 情绪丰富的 PiSloth

PiSloth 拥有非常多的情绪。时而快乐, 时而害羞, 时而迷茫。

运行代码

```
cd /home/pi/pisloth/examples
sudo python3 emotional_pisloth.py
```

代码

```
from pisloth import Sloth
from robot_hat import TTS, Music
import time

tts = TTS()
music = Music()

sloth = Sloth([1,2,3,4])
sloth.set_offset([0,0,0,0])

def confuse():
    try:
        music.sound_effect_threading('./sounds/sign.wav')
    except Exception as e:
        print(e)
    sloth.do_action('hook', 1, 90)

def happy():
    try:
        music.sound_effect_threading('./sounds/happy2.wav')
```

(续下页)

(接上页)

```
except Exception as e:
    print(e)
for i in range(3):
    sloth.do_action('hook', 1, 90)
    sloth.do_action('stand', 1, 90)

def fear():
    try:
        music.sound_effect_threading('./sounds/warning.wav')
    except Exception as e:
        print(e)
    sloth.do_action('hook', 1, 90)
    sloth.do_action('stand', 1, 90)
    try:
        music.sound_effect_threading('./sounds/warning.wav')
    except Exception as e:
        print(e)
    sloth.do_action('walk backward boldly', 1, 90)
    sloth.do_action('stand', 1, 90)

def sad():
    try:
        music.sound_effect_threading('./sounds/depress.wav')
    except Exception as e:
        print(e)
    sloth.do_action('big swing', 1, 90)

def angry():
    try:
        music.sound_effect_threading('./sounds/error.wav')
    except Exception as e:
        print(e)
    sloth.do_action('walk backward boldly', 1, 90)
    sloth.do_action('stand', 1, 90)

def fail():
    try:
        music.sound_effect_threading('./sounds/depress2.wav')
    except Exception as e:
        print(e)
    sloth.do_action('fall left', 1, 90)

def shy():
    try:
        music.sound_effect_threading('./sounds/talk3.wav')
    except Exception as e:
        print(e)
    sloth.do_action('close', 1, 90)
    time.sleep(1)
    try:
        music.sound_effect_threading('./sounds/talk2.wav')
    except Exception as e:
        print(e)
    sloth.do_action('stand', 1, 90)

def main():
```

(续下页)

```
print("shy")
shy()
time.sleep(1)
sloth.do_action('stand', 1, 90)
time.sleep(2)

print("confuse")
confuse()
time.sleep(1)
sloth.do_action('stand', 1, 90)
time.sleep(2)

print("happy")
happy()
time.sleep(1)
sloth.do_action('stand', 1, 90)
time.sleep(2)

print("fear")
fear()
time.sleep(1)
sloth.do_action('stand', 1, 90)
time.sleep(2)

print("sad")
sad()
time.sleep(1)
sloth.do_action('stand', 1, 90)
time.sleep(2)

print("angry")
angry()
time.sleep(1)
sloth.do_action('stand', 1, 90)
time.sleep(2)

print("fail")
fail()
time.sleep(1)
sloth.do_action('stand', 1, 90)
time.sleep(2)

if __name__ == "__main__":
    while True:
        main()
```

这个怎么运作？

在这个项目中，动作 + 音效组合成不同的情感动作，你也可以自己修改。

备注：这个 fail 动作会使 PiSloth 摔倒，注意不要让它从桌子上掉下来弄坏它。

您可以通过 *Filezilla* 软件 向 musics 或 sounds 文件夹添加不同的音效或音乐。

4.8 遥控

在这个项目中，我们将学习如何使用键盘远程控制 PiSloth。您可以控制 PiSloth 上下左右移动并通过特定键说话。

运行代码

```
cd /home/pi/pislloth/examples
sudo python3 keyboard_control.py
```

代码运行后，按下 wasd 按键来控制 PiSloth，按下 1234 来播放不同的音效，按下 qe 来让 PiSloth 说话。

按下 esc 按键来退出。

- w: 前进
- a: 左转
- s: 后腿
- d: 右转
- 1: 音效: talk1
- 2: 音效: talk2
- 3: 音效: talk3
- 4: 音效: depress2
- q: Say: "Oh hello there"
- e: Say: "bye"
- esc: Quit

代码

```
from pislloth import Sloth
from robot_hat import Music
from robot_hat import TTS
import sys
import tty
import termios
import time

sloth = Sloth([1,2,3,4])
tts = TTS()
music = Music()
sloth.set_offset([0,0,0,0])

def readchar():
    fd = sys.stdin.fileno()
    old_settings = termios.tcgetattr(fd)
    try:
        tty.setraw(sys.stdin.fileno())
        ch = sys.stdin.read(1)
    finally:
        termios.tcsetattr(fd, termios.TCSADRAIN, old_settings)
    return ch

manual = '''
```

(续下页)

```

Press keys on keyboard to control PiSloth!

w: Forward
a: Turn left
s: Backward
d: Turn right
1: Sound effect: talk1
2: Sound effect: talk2
3: Sound effect: talk3
4: Sound effect: depress2
q: Say: "Oh hello there"
e: Say: "bye"
esc: Quit
...

def main():
    print(manual)
    while True:
        key = readchar().lower()
        # print(key)
        if key == "w":
            sloth.do_action('forward', 1, 90)
        elif key == "a":
            sloth.do_action('turn left', 1, 90)
        elif key == "s":
            sloth.do_action('backward', 1, 90)
        elif key == "d":
            sloth.do_action('turn right', 1, 90)
        elif key == "1":
            music.sound_effect_play('./sounds/talk1.wav')
        elif key == "2":
            music.sound_effect_play('./sounds/talk2.wav')
        elif key == "3":
            music.sound_effect_play('./sounds/talk3.wav')
        elif key == "4":
            music.sound_effect_play('./sounds/depress.wav')
        elif key == "q":
            tts.say("Oh hello there")
        elif key == "e":
            tts.say("bye")
        elif key == chr(27): # 27 for ESC
            break
        time.sleep(0.05)
    print("\nQuit")

if __name__ == "__main__":
    main()

```

这个怎么运作？

该功能引用标准输入流并返回读取的数据流的第一个字符。

- `tty.setraw(sys.stdin.fileno)` 就是将标准输入流改为 raw 模式，即传输过程中所有字符都不会被转义，包括特殊字符。
- `old_settings = termios.tcgetattr(fd)` 和 `termios.tcsetattr(fd, termios.TCSADRAIN, old_settings)` 并起到备份和恢复的作用。

```
def readchar():
    fd = sys.stdin.fileno()
    old_settings = termios.tcgetattr(fd)
    try:
        tty.setraw(sys.stdin.fileno())
        ch = sys.stdin.read(1)
    finally:
        termios.tcsetattr(fd, termios.TCSADRAIN, old_settings)
    return ch
```

最后，根据读取的键盘字符，让 PiSloth 做我们设置的动作，调用 `tts.say()` 函数说话或播放事先准备好的音效。

```
key = readchar().lower()
# print(key)
if key == "w":
    sloth.do_action('forward', 1, 90)
elif key == "a":
    sloth.do_action('turn left', 1, 90)
elif key == "s":
    sloth.do_action('backward', 1, 90)
elif key == "d":
    sloth.do_action('turn right', 1, 90)
elif key == "1":
    music.sound_effect_play('./sounds/talk1.wav')
elif key == "2":
    music.sound_effect_play('./sounds/talk2.wav')
elif key == "3":
    music.sound_effect_play('./sounds/talk3.wav')
elif key == "4":
    music.sound_effect_play('./sounds/depress.wav')
elif key == "q":
    tts.say("Oh hello there")
elif key == "e":
    tts.say("bye")
elif key == chr(27): # 27 for ESC
    break
```

备注：您可以通过 *Filezilla* 软件向 `musics` 或 `sounds` 文件夹添加不同的音效或音乐。

4.9 自定义步态

在之前的项目中，我们使用了很多我们自己写的动作，那么这些动作是如何组合和完成的呢？一般来说，一个动作由一个或多个步态组成。

在这个项目中，我们将学习如何自定义 PiSloth 的步态。

备注：您可以为您的 PiSloth 下载并打印卡通面具。

- 卡通面具 (.pdf)

运行代码

```
cd /home/pi/pisloth/examples
sudo python3 custom_step.py
```

代码运行后，按下以下按键来调整 PiSloth 的各个舵机的角度。

- q: 增加左腿的角度
- w: 减小左腿的角度
- z: 增加左脚的角度
- x: 减小左脚的角度
- i: 增加右腿的角度
- o: 减小右腿的角度
- n: 增加右脚的角度
- m: 减小右脚的角度
- 空格键: 打印所有角度
- ESC: 退出

比如按 `zx` 键和 `nm` 键，我们让它做出做如图所示的步态。



此时按 **空格键** 打印 4 个舵机的角度。您需要记录这些角度值，这些值将在下一个项目自定义动作中使用。

```

pi@raspberrypi:~/pislloth/examples $ sudo python3 custom_step.py
Press keys on keyboard to control PiSloth!
Q: Increase left-up servo angle
W: Decrease left-up servo angle
Z: Increase left-down servo angle
X: Decrease left-down servo angle
I: Increase right-up servo angle
O: Decrease right-up servo angle
N: Increase right-down servo angle
M: Decrease right-down servo angle
SPACE: Print all servo angle
ESC: Quit
0 -45 0 40

```

代码

```

from pislloth import Sloth
# from robot_hat import Music
# from robot_hat import TTS
from robot_hat import PWM
from robot_hat import Servo

import sys
import tty
import termios
import time

sloth = Sloth([1,2,3,4])
# tts = TTS()
# music = Music()
sloth.set_offset([0,0,0,0])

right_leg_servo = Servo(PWM('P0'))
right_foot_servo = Servo(PWM('P1'))
left_leg_servo = Servo(PWM('P2'))
left_foot_servo = Servo(PWM('P3'))

def readchar():
    fd = sys.stdin.fileno()
    old_settings = termios.tcgetattr(fd)
    try:
        tty.setraw(sys.stdin.fileno())
        ch = sys.stdin.read(1)
    finally:
        termios.tcsetattr(fd, termios.TCSADRAIN, old_settings)
    return ch

manual = '''
Press keys on keyboard to control PiSloth!
q: Increase the servo angle of the left leg
w: Decrease the servo angle of the left leg
z: Increase the servo angle of the left foot
x: Decrease the servo angle of the left foot
i: Increase the servo angle of the right leg
o: Decrease the servo angle of the right leg

```

(续下页)

```
n: Increase the servo angle of the right foot
m: Decrease the servo angle of the right foot
SPACE: Print all angle
ESC: Quit
'''

def main():
    print(manual)

    left_leg=0
    left_foot=0
    right_leg=0
    right_foot=0
    while True:
        key = readchar().lower()
        # print(key)
        if key == "q":
            left_leg = left_leg+5
        elif key == "w":
            left_leg = left_leg-5
        elif key == "z":
            left_foot = left_foot+5
        elif key == "x":
            left_foot = left_foot-5
        elif key == "i":
            right_leg = right_leg+5
        elif key == "o":
            right_leg = right_leg-5
        elif key == "n":
            right_foot = right_foot+5
        elif key == "m":
            right_foot = right_foot-5
        elif key == chr(32): # 32 for space
            print(right_leg,right_foot,left_leg,left_foot)
        elif key == chr(27): # 27 for ESC
            break

        right_leg_servo.angle(right_leg)
        right_foot_servo.angle(right_foot)
        left_leg_servo.angle(left_leg)
        left_foot_servo.angle(left_foot)
        # time.sleep(0.05)

    print("\nQuit")

if __name__ == "__main__":
    main()
```

4.10 自定义动作

在之前的项目中，我们能够给 PiSloth 自定义步态，那么我们如何将这此步态组合成动作呢？例如，让 PiSloth 摆出上一个项目中步态，然后回到站立姿势，如此循环实现蹦蹦跳跳的效果。



备注：您可以为您的 PiSloth 下载并打印卡通面具。

- 卡通面具 (.pdf)

第1步：进入 `/home/pi/pislloth/examples` 路径。

```
cd /home/pi/pislloth/examples
```

第2步：打开 `custom_action.py` 文件。

```
nano custom_action.py
```

第3步：修改 `sloth.add_action()` 中的角度，每组代表 1 个步态，这里只设置了 2 个步态，您可以无限添加下去。

```
sloth.add_action("my_action", [  
    [ 0, -45, 0, 40],  
    [0, 0, 0, 0]  
])
```

第 4 步: 运行代码.

```
sudo python3 custom_action.py
```

代码

```
from pisloth import Sloth
import time

sloth = Sloth([1,2,3,4])
sloth.add_action("my_action", [
    [ 0,-45 ,0, 40],
    [0,  0, 0,  0]
])

def main():
    sloth.do_action("my_action", 1, 80)
    time.sleep(1)

if __name__ == "__main__":
    while True:
        main()
```

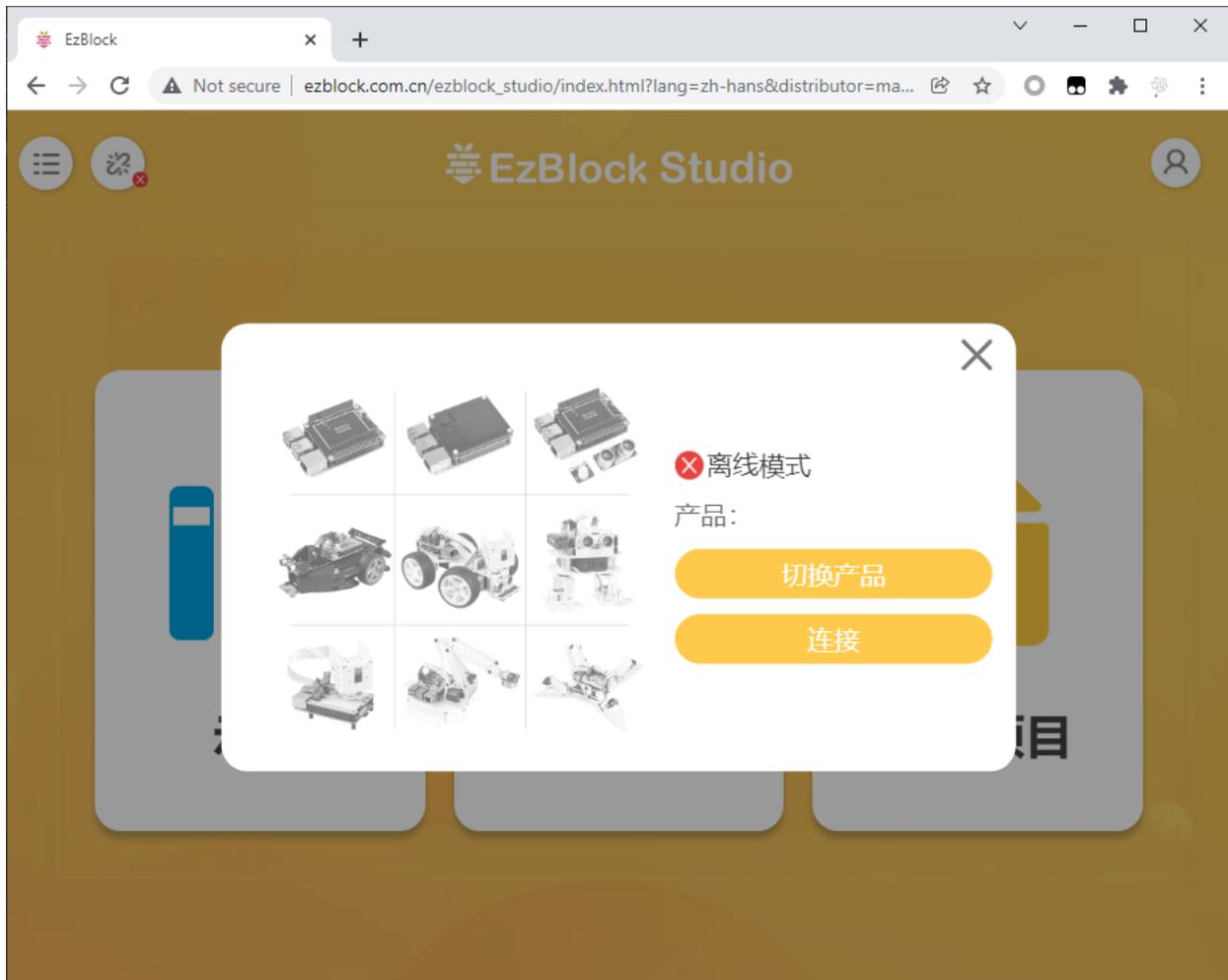
5.1 网页版 EzBlock

进入 [网页版 EzBlock](#)。

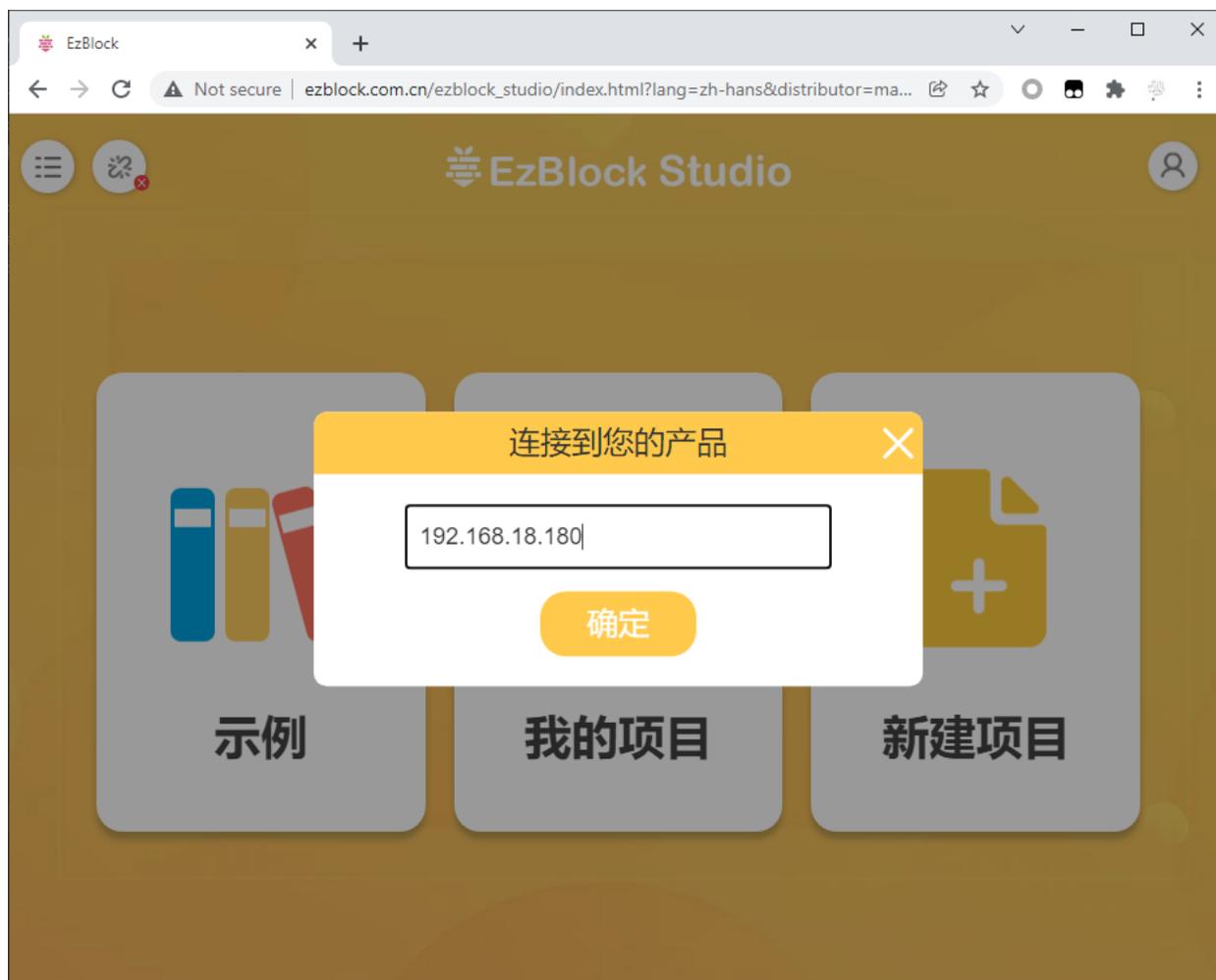
在首页点击左上角的产品按钮。



点击连接。



输入 IP 地址。



随后将弹出“连接成功”的提示，此时你可以回到玩转 *EzBlock*。



5.2 apt 和 pip 更换国内源

树莓派系统默认的 apt 源和 pip 源都是国外的服务器，使用国内网络访问可能会发生超时（ReadTimeoutError），或被拒绝访问的情况，如果是这样我们可以将 apt 和 pip 更改为国内的源，步骤如下所示：

1. apt 更换国内源

- 1) 访问链接：<https://mirrors.tuna.tsinghua.edu.cn/help/raspbian/> 以了解配置文件修改详情。
- 2) 不同的树莓派系统版本修改不同，先选择对应的版本，比如我的是 Debian 10（buster），如果你的树莓派系统是 bullseye 则选择对应版本，若没有符合的版本则请重新安装 buster 及以下版本的系统。

使用说明

选择你的 Raspbian 对应的 Debian 版本:



```
# 编辑 `/etc/apt/sources.list` 文件，用以下内容取代：
deb [arch=armhf] http://mirrors.tuna.tsinghua.edu.cn/raspbian/raspbian/ buster main non-free contrib rpi
deb-src http://mirrors.tuna.tsinghua.edu.cn/raspbian/raspbian/ buster main non-free contrib rpi

# 如果需要 arm64 软件源，在 `/etc/apt/sources.list` 中加上
deb [arch=arm64] http://mirrors.tuna.tsinghua.edu.cn/raspbian/multiarch/ buster main

# 编辑 `/etc/apt/sources.list.d/raspi.list` 文件，删除原文件所有内容，用以下内容取代：
deb http://mirrors.tuna.tsinghua.edu.cn/raspberrypi/ buster main
```

备注：一般不删除原内容，可以将其用 # 注释掉。

3) 我们这里以 raspbian buster 版本为例，用 nano 命令打开 /etc/apt/sources.list 文件。

```
sudo nano /etc/apt/sources.list
```

4) 然后用 # 将原本的内容注释掉，在最后面附上下面的代码。

```
# deb http://raspbian.raspberrypi.org/raspbian/ buster main contrib non-free rpi
# Uncomment line below then 'apt-get update' to enable 'apt-get source'
#deb-src http://raspbian.raspberrypi.org/raspbian/ buster main contrib non-free rpi

deb [arch=armhf] http://mirrors.tuna.tsinghua.edu.cn/raspbian/raspbian/ buster main
↪non-free contrib rpi
deb-src http://mirrors.tuna.tsinghua.edu.cn/raspbian/raspbian/ buster main non-free
↪contrib rpi
```

5) 按下 Ctrl+O 保存，按下 Ctrl+X 和 Y 退出。

6) 用 nano 命令打开 etc/apt/sources.list.d/raspi.list 文件。

```
sudo nano /etc/apt/sources.list.d/raspi.list
```

7) 然后用 # 将原本的内容注释掉，在最后面附上 deb... 代码。

```
# deb http://archive.raspberrypi.org/debian/ buster main
# Uncomment line below then 'apt-get update' to enable 'apt-get source'
#deb-src http://archive.raspberrypi.org/debian/ buster main

deb http://mirrors.tuna.tsinghua.edu.cn/raspberrypi/ buster main ui
```

8) 按下 Ctrl+O 保存，按下 Ctrl+X 和 Y 退出。

9) 用以下命令更新软件列表：

```
sudo apt update
```

2.(Pypi) pip 更换国内源

可以参考链接 <https://mirrors.tuna.tsinghua.edu.cn/help/pypi/> 了解详情。

有两种方法修改配置文件。

方法一：使用 pip 指令设置

```
pip3 config set global.index-url https://pypi.tuna.tsinghua.edu.cn/simple
```

方法二：手动编辑文件

Linux/Mac os 环境中，配置文件位置在 `~/.pip/pip.conf`（如果不存在则手动创建该目录和文件）。

```
sudo mkdir -p ~/.pip  
sudo nano ~/.pip/pip.conf
```

然后按如下编辑文件内容

```
[global]  
index-url = https://pypi.tuna.tsinghua.edu.cn/simple  
[install]  
trusted-host = https://pypi.tuna.tsinghua.edu.cn
```

最后 apt 和 pip 都已经更换了国内下载源，这样下载速度就会提高很快不会导致下载失败了。

5.3 Filezilla 软件



文件传输协议 (FTP) 是一种标准通信协议，用于在计算机网络上将计算机文件从服务器传输到客户端。

Filezilla 是一款开源软件，不仅支持 FTP，还支持 FTP over TLS (FTPS) 和 SFTP。我们可以使用 Filezilla 将本地文件（如图片和音频等）上传到树莓派，或者从树莓派下载文件到本地。

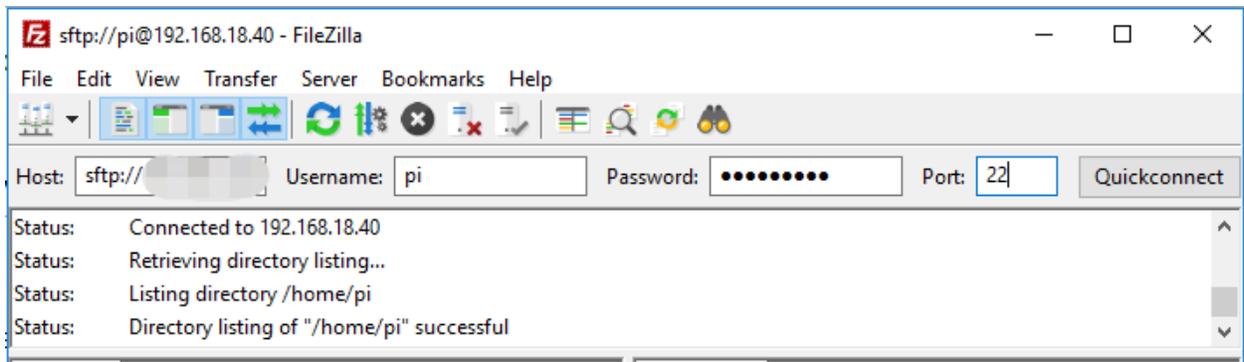
第 1 步：下载 Filezilla。

在 [Filezilla 官方网站](#) 下载客户端，Filezilla 有一个很好的教程，请参考：[Filezilla 文档](#)。

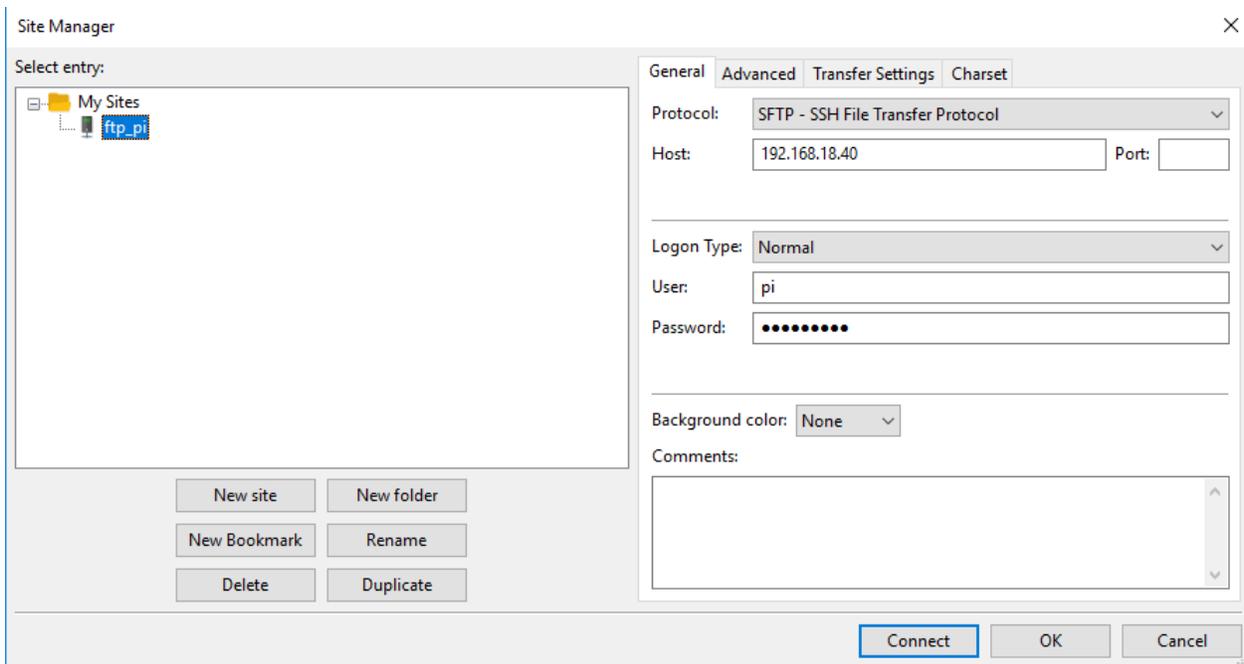
第 2 步：连接树莓派

快速安装后打开它，现在 [连接到 FTP 服务器](#)。

它有 3 种连接方式，这里我们使用 [快速连接](#) 栏。输入 [主机名/IP](#)、[用户名](#)、[密码](#) 和 [端口 \(22\)](#)，然后点击 [快速连接](#) 或按 [回车](#) 连接到服务器。

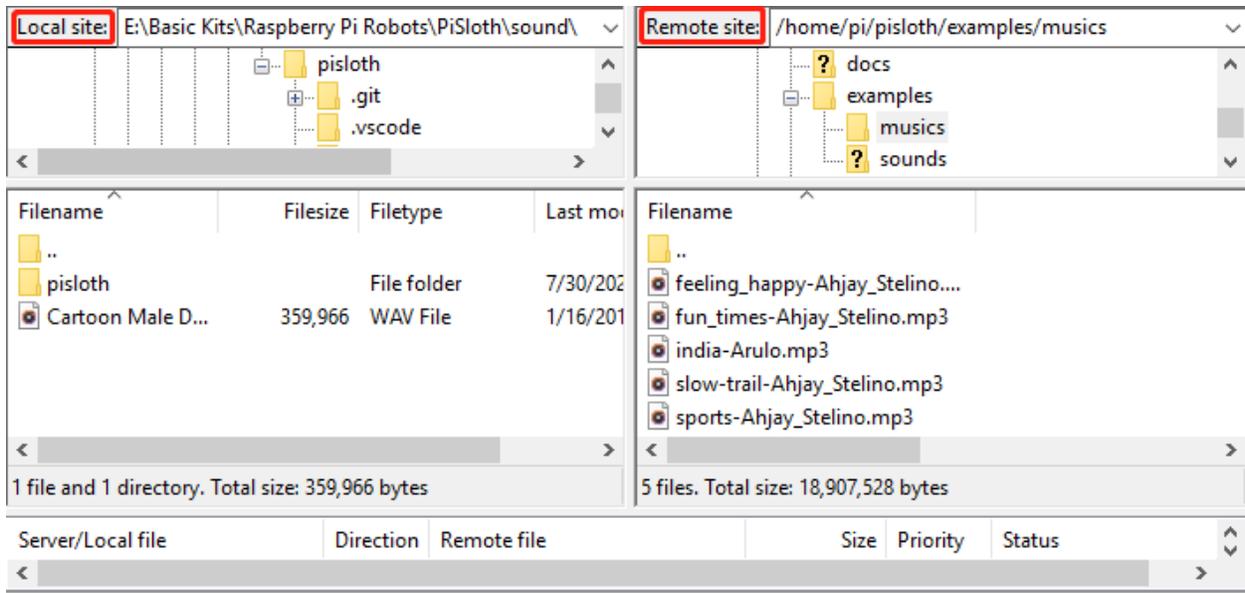


备注：快速连接是测试您的登录信息的好方法。如果要创建永久条目，可以在快速连接成功后选择 **File-> Copy Current Connection to Site Manager**（文件-> 将当前连接复制到站点管理器），输入名称并单击 **确定**。下次您将能够通过 **File -> Site Manager**（文件-> 站点管理器。）中选择先前保存的站点进行连接。



第3步：上传/下载文件。

您可以通过拖放将本地文件上传到树莓派，也可以将树莓派文件中的文件下载到本地。

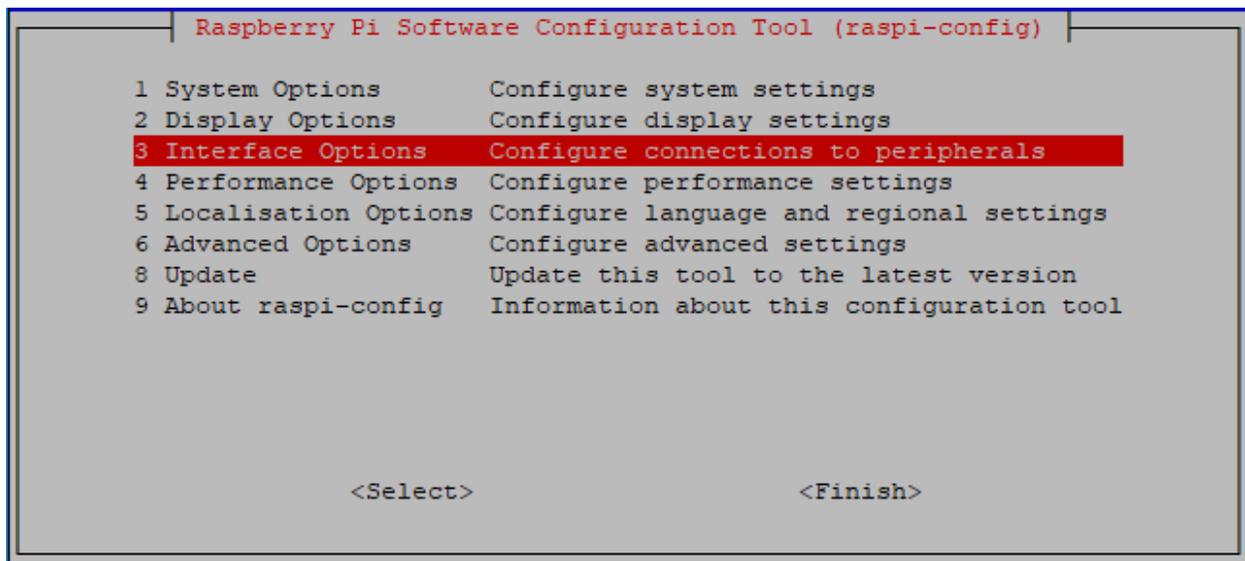


5.4 I2C 配置

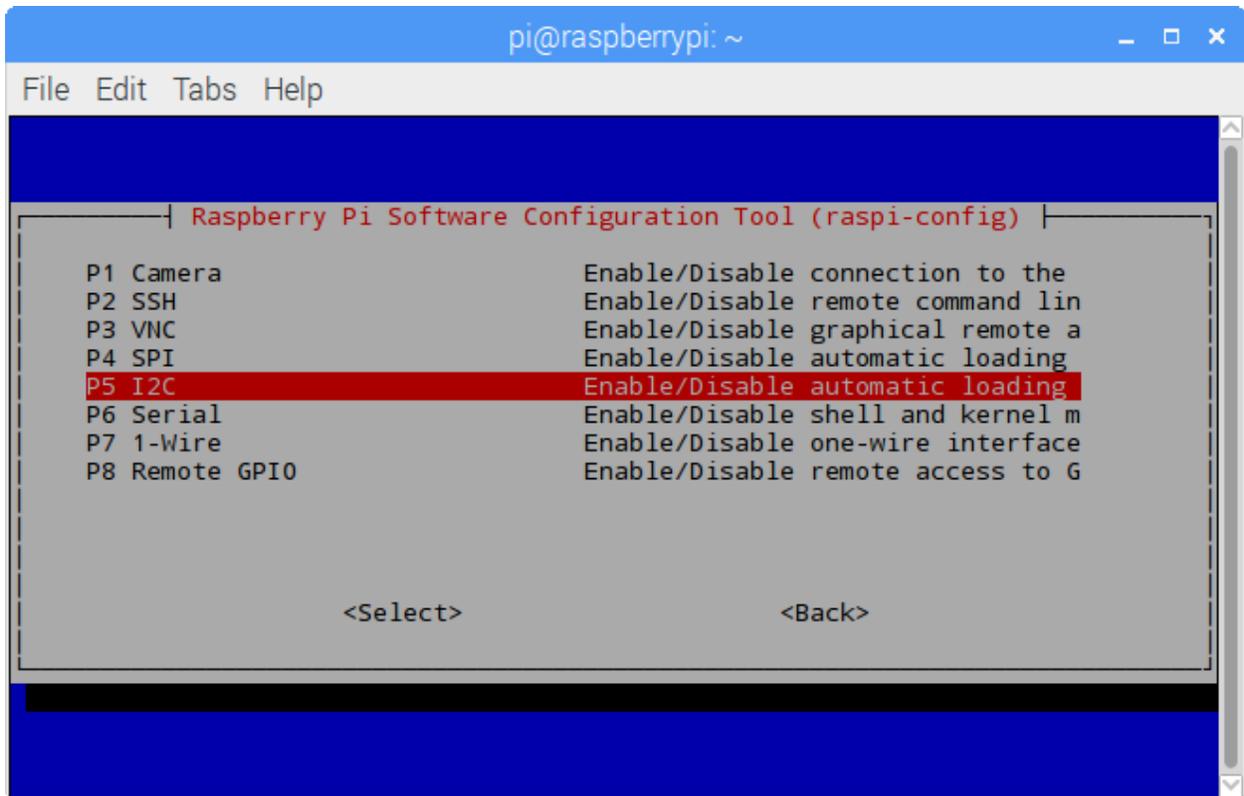
启用你的树莓派的 I2C 端口。如果你已经启用了它，请跳过这一步；如果你不知道你是否已经做了，请继续。

```
sudo raspi-config
```

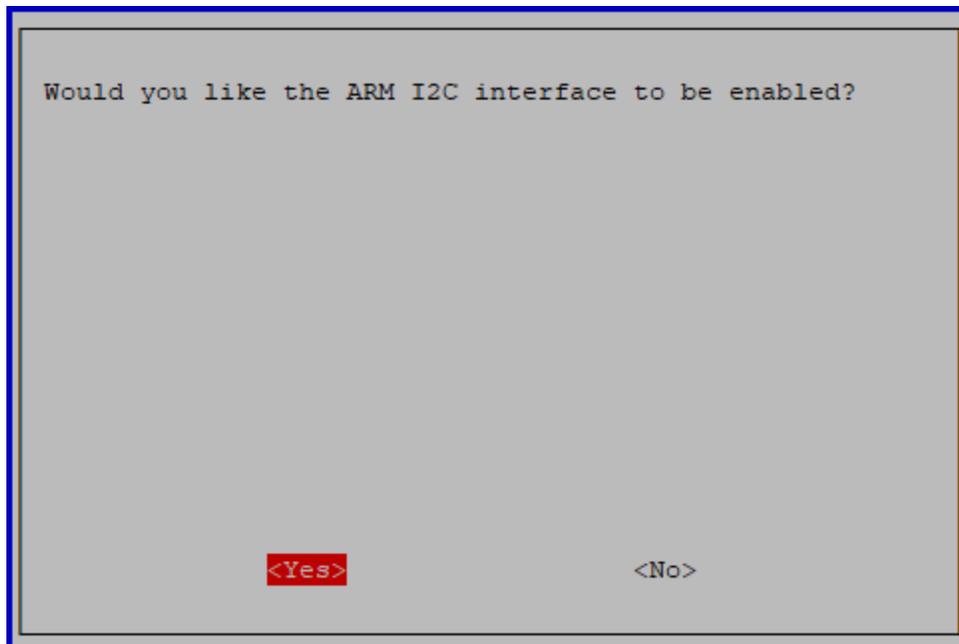
3 Interfacing options



P5 I2C



<Yes>, 然后 <Ok> -> <Finish>.



5.5 远程桌面

有两种方法可以远程控制树莓派的桌面。

VNC 和 XRDP，你可以使用它们中的任何一种。

5.5.1 VNC

你可以通过 VNC 使用远程桌面的功能。

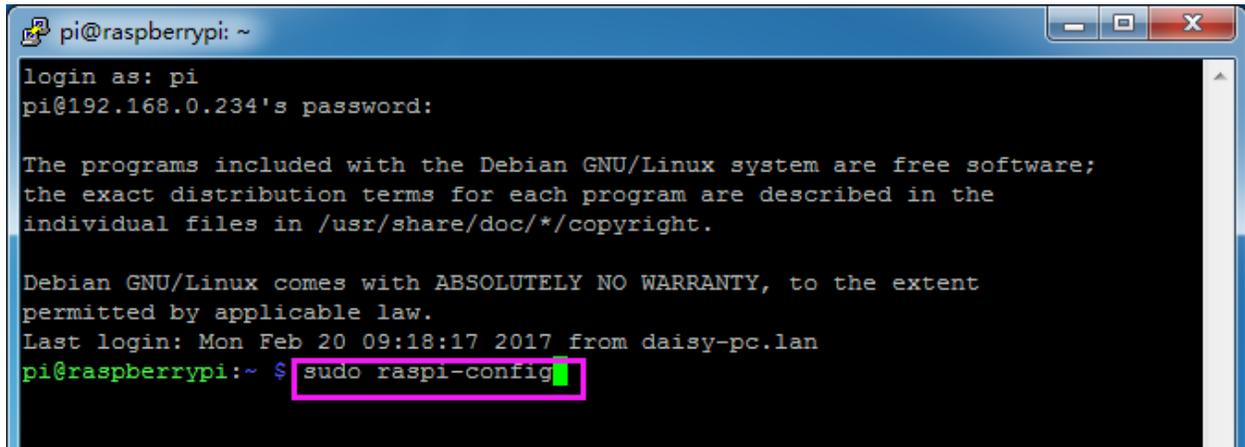
启用 VNC 服务

系统中已经安装了 VNC 服务。默认情况下，VNC 被禁用。默认情况下，VNC 是禁用的。你需要在配置中启用它。

第 1 步

输入以下命令。

```
sudo raspi-config
```



```
pi@raspberrypi: ~  
login as: pi  
pi@192.168.0.234's password:  
  
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: Mon Feb 20 09:18:17 2017 from daisy-pc.lan  
pi@raspberrypi:~ $ sudo raspi-config
```

第 2 步

选择 **3 Interfacing Option**，按你键盘上的向下箭头键。键盘上的向下箭头键，然后按 **Enter** 键。

```
Raspberry Pi Software Configuration Tool (raspi-config)

1 System Options      Configure system settings
2 Display Options    Configure display settings
3 Interface Options   Configure connections to peripherals
4 Performance Options Configure performance settings
5 Localisation Options Configure language and regional settings
6 Advanced Options   Configure advanced settings
8 Update              Update this tool to the latest version
9 About raspi-config Information about this configuration tool

<Select>                <Finish>
```

第3步

P3 VNC

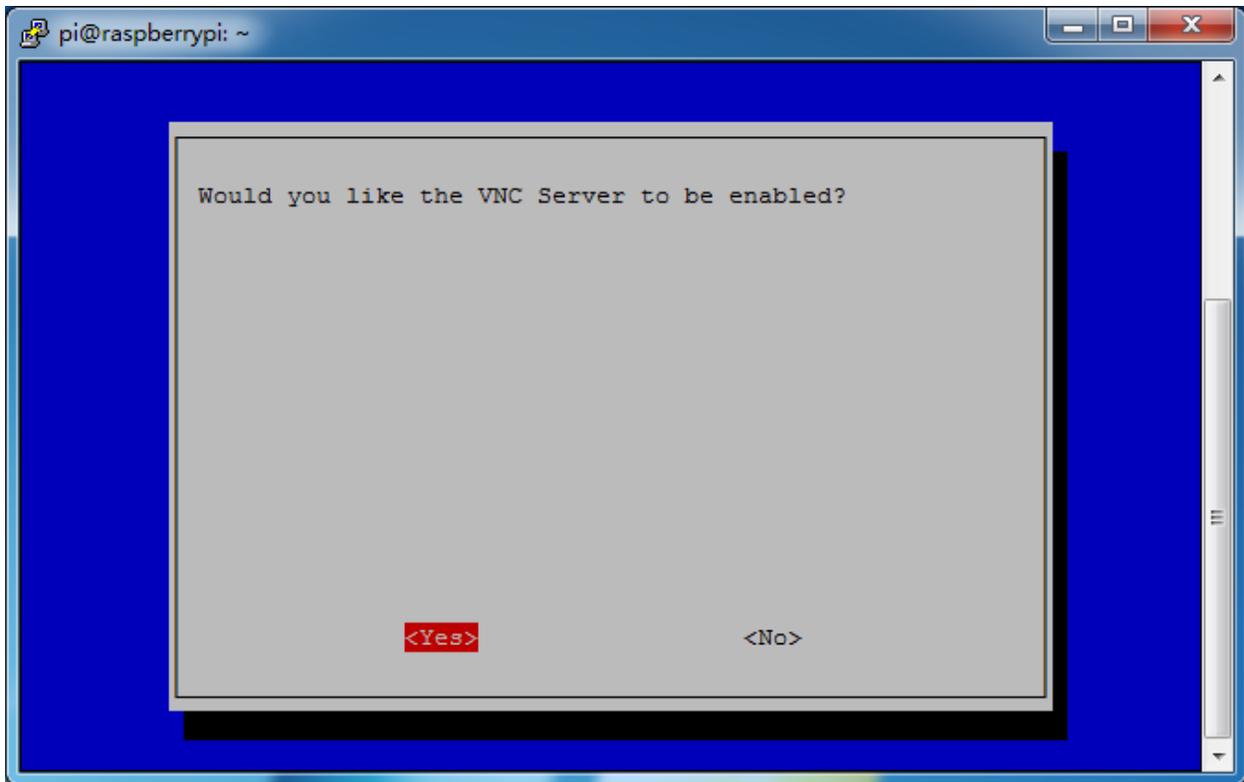
```
Raspberry Pi Software Configuration Tool (raspi-config)

P1 Camera      Enable/disable connection to the Raspberry Pi Camera
P2 SSH         Enable/disable remote command line access using SSH
P3 VNC         Enable/disable graphical remote access using RealVNC
P4 SPI         Enable/disable automatic loading of SPI kernel module
P5 I2C         Enable/disable automatic loading of I2C kernel module
P6 Serial Port Enable/disable shell messages on the serial connection
P7 1-Wire      Enable/disable one-wire interface
P8 Remote GPIO Enable/disable remote access to GPIO pins

<Select>                <Back>
```

第4步

选择 Yes -> OK -> Finish 来退出配置。



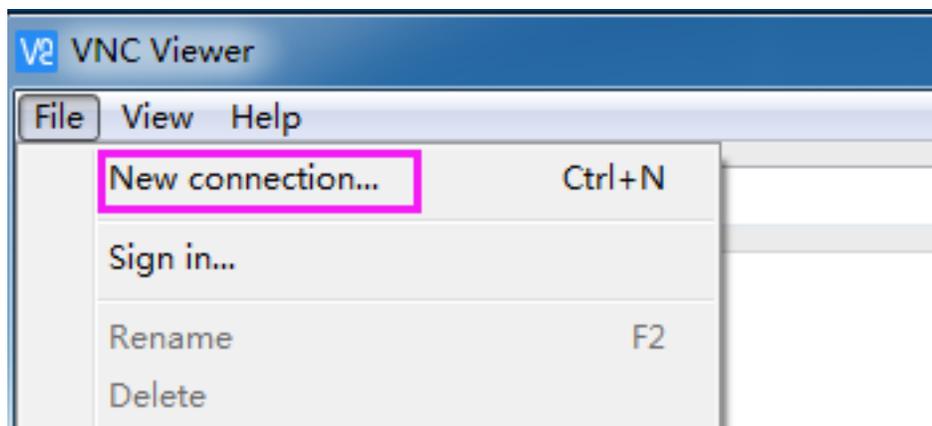
登录 VNC

第 1 步

你需要在个人电脑上下载并安装 VNC Viewer。安装完成后，打开它。

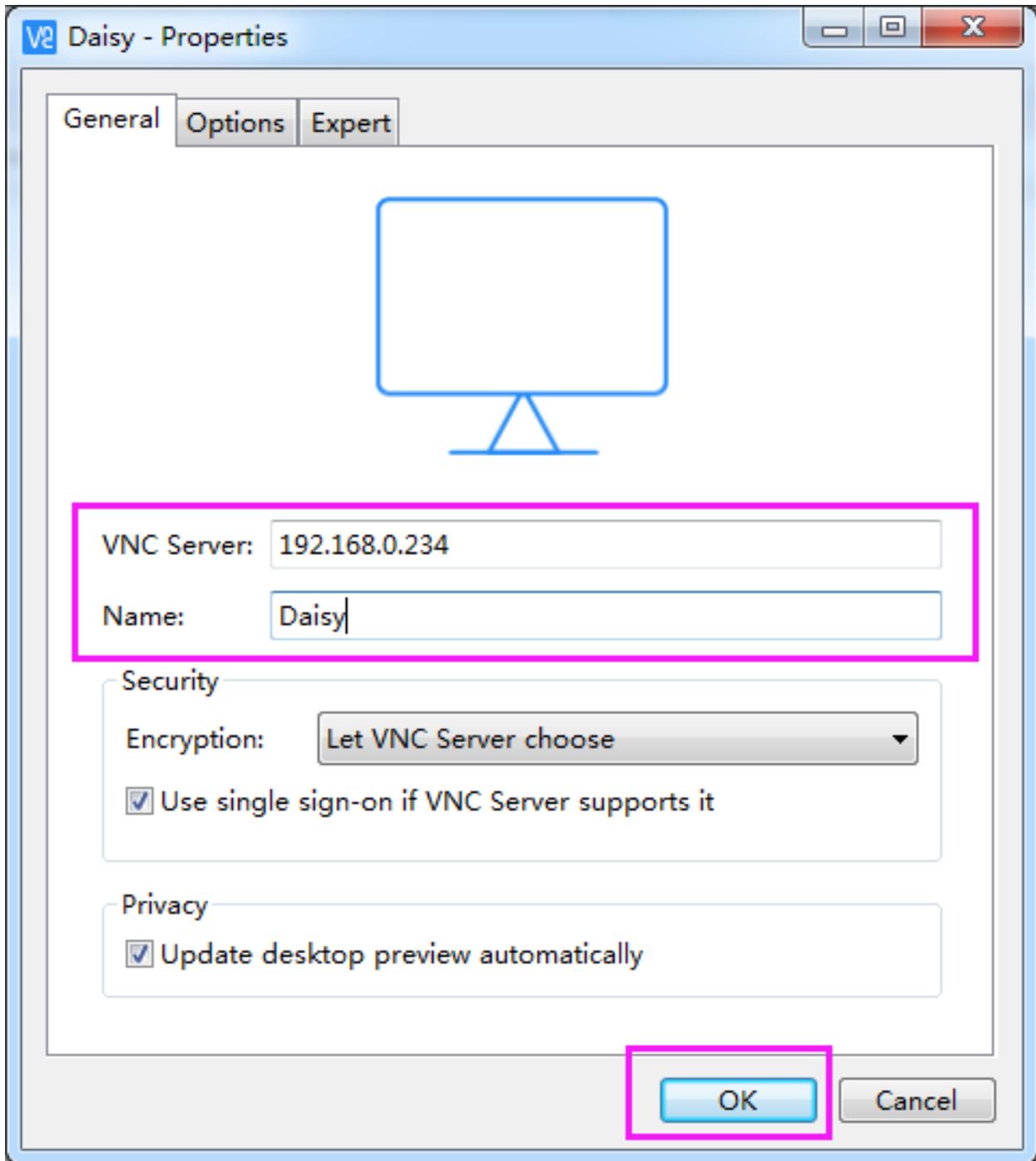
第 2 步

然后选择“New connection”。



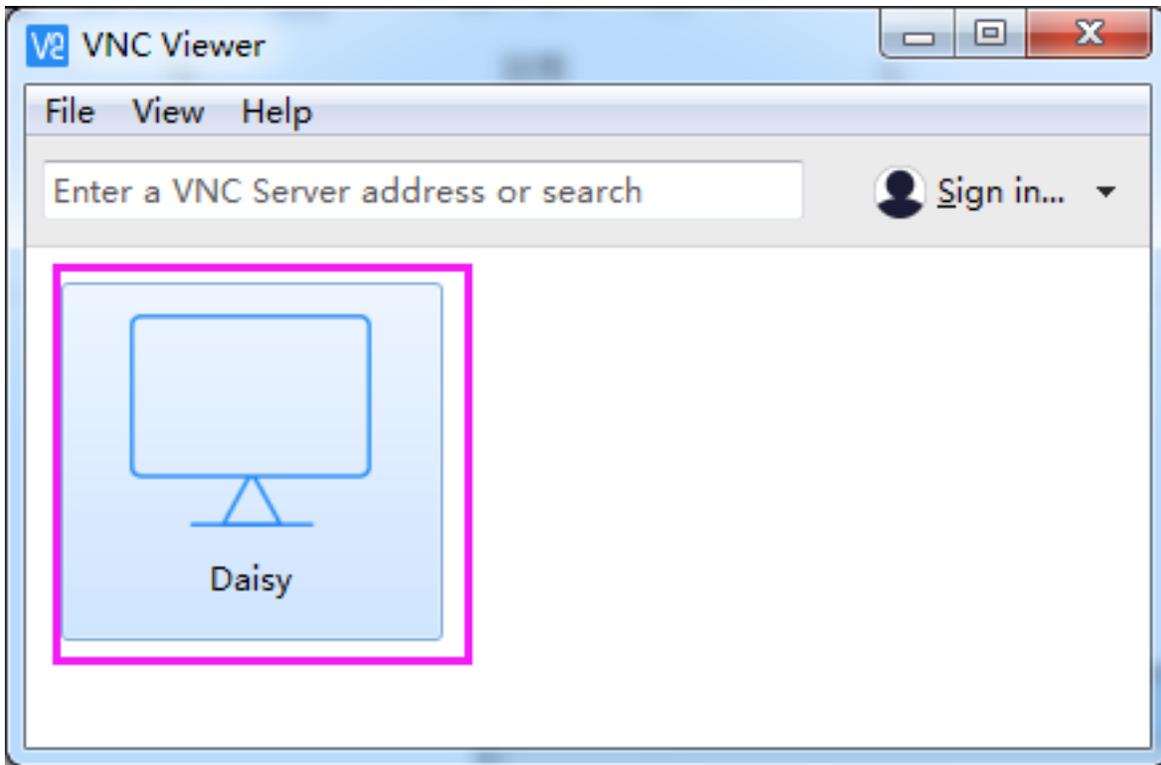
第 3 步

输入树莓派的 IP 地址和任意 名称。



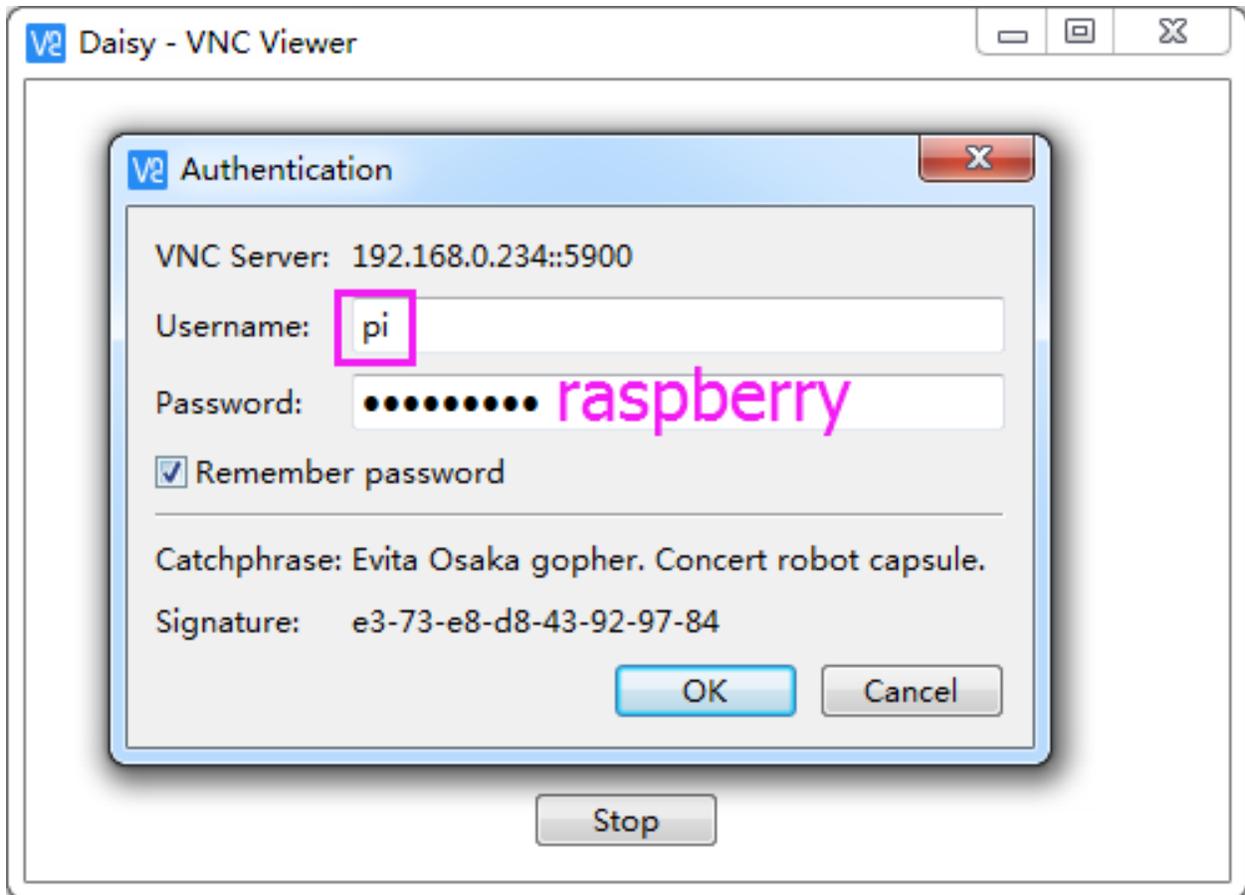
第 4 步

双击刚刚创建的 连接。



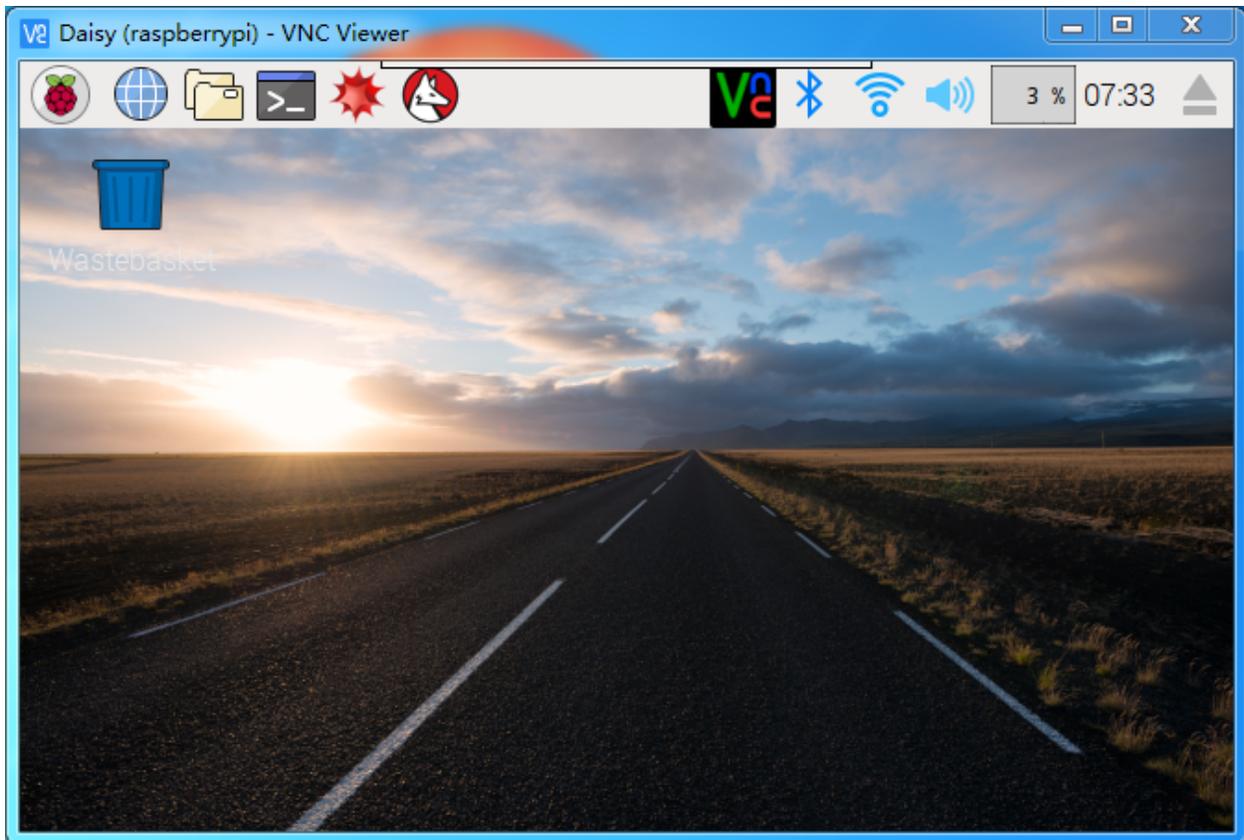
第 5 步

输入用户名 (**pi**) 和密码 (默认为 **raspberr**)。



第 6 步

现在你可以看到树莓派的桌面了。



VNC 部分就到此为止了。

5.5.2 XRDP

另一种远程桌面的方法是 XRDP，它使用 RDP（微软远程桌面协议）提供图形化登录到远程机器。远程桌面协议）。

安装 XRDP

第 1 步

通过使用 SSH 登录到树莓派。

第 2 步

输入以下说明来安装 XRDP。

```
sudo apt-get update
sudo apt-get install xrdp
```

第 3 步

后来，安装开始了。

按（“Y”），然后按（“Enter”）来确认。

```
pi@raspberrypi: ~  
pi@raspberrypi:~ $ sudo apt-get install xrdp  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following extra packages will be installed:  
  vnc4server x11-apps x11-session-utils xbase-clients xbitmaps xfonts-base  
Suggested packages:  
  vnc-java mesa-utils x11-xfs-utils  
The following NEW packages will be installed:  
  vnc4server x11-apps x11-session-utils xbase-clients xbitmaps xfonts-base  
  xrdp  
0 upgraded, 7 newly installed, 0 to remove and 0 not upgraded.  
Need to get 8,468 kB of archives.  
After this operation, 17.1 MB of additional disk space will be used.  
Do you want to continue? [Y/n] y
```

第 4 步

安装完成后，你应该使用 Windows 远程桌面应用程序登录到你的树莓派。使用 Windows 远程桌面应用程序。

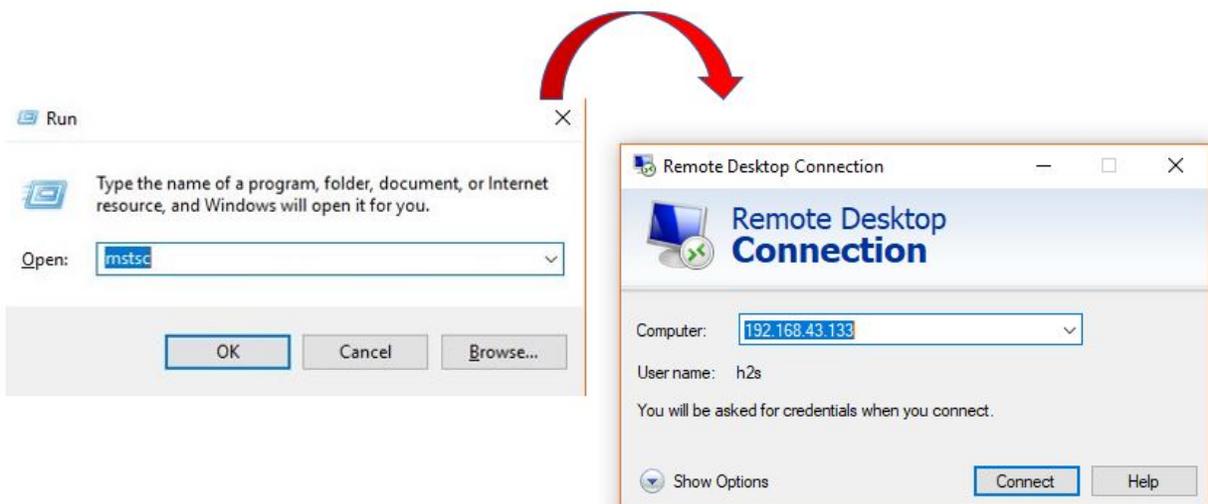
登录到 XRDP

第 1 步

如果你是一个 Windows 用户，你可以使用 Windows 自带的远程桌面功能。自带的远程桌面功能。如果你是 Mac 用户，你可以从 APP Store 下载并使用微软远程桌面，两者之间没有太大区别。两者之间没有什么区别。接下来的例子是 Windows 远程桌面。

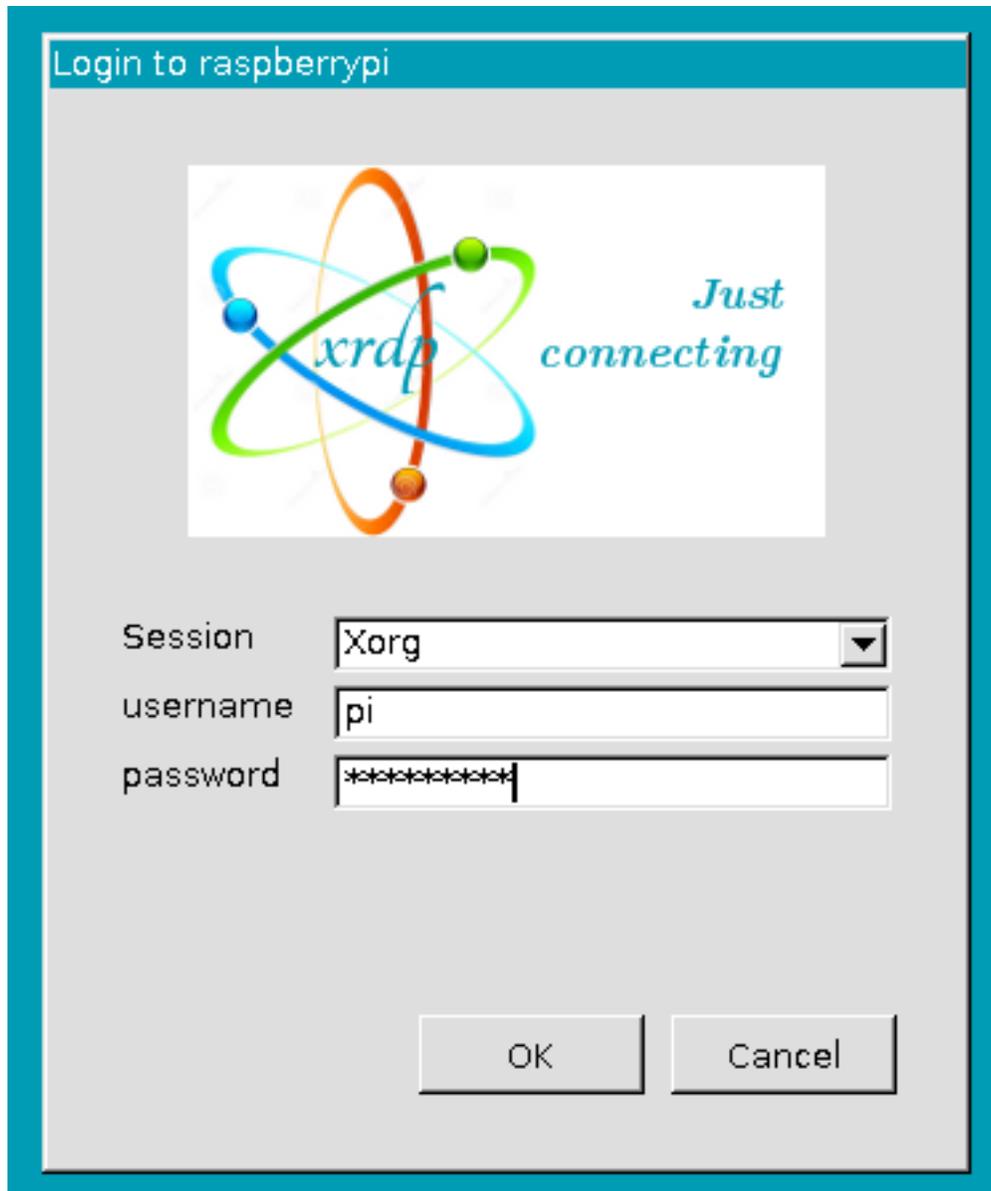
第 2 步

在运行（WIN+R）中键入“mstsc”，打开远程桌面连接，并输入树莓派的 IP 地址，然后点击“Connect”。



第 3 步

然后弹出 xrdp 登录页面。请键入您的用户名和密码。之后，请点击“OK”。在你第一次登录的时候。你的用户名是“pi”，密码是“raspberrypi”。



第 4 步

在这里，你通过使用远程桌面成功登录到 RPi。



5.6 关于电池

适用参数

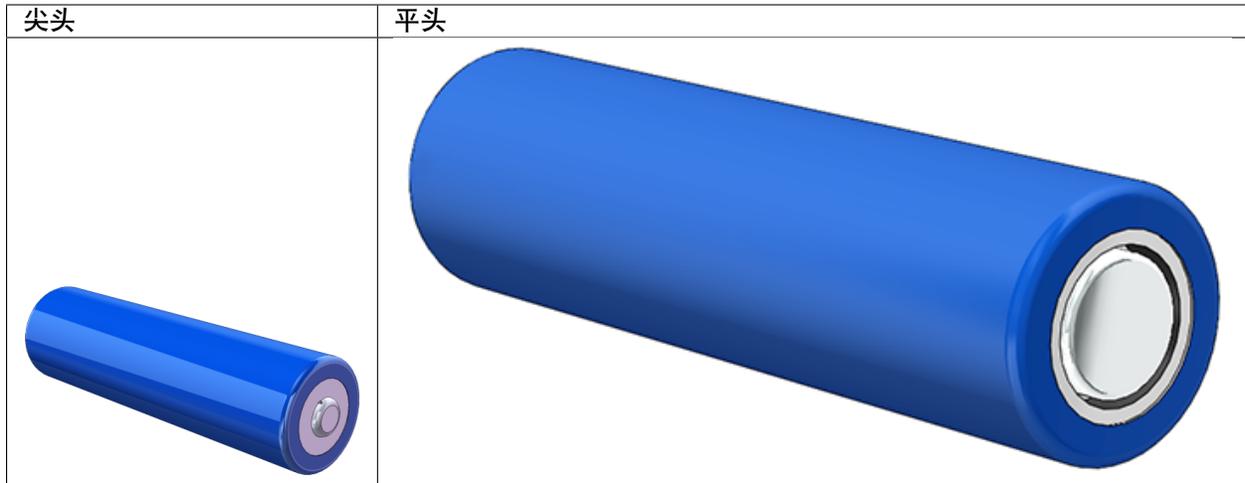
- 3.7V
- 18650
- 可充电的
- 锂离子电池
- 尖头电池
- 无保护板

备注:

- Robot HAT 无法给电池充电，需要购买电池充电器。
 - 当 Robot HAT 上的两个电量指示灯熄灭时，表示电量过低，需要给电池充电。
-

尖头 vs 平头？

请选择尖头电池，以确保电池与电池座之间的连接良好。

**没有保护板？**

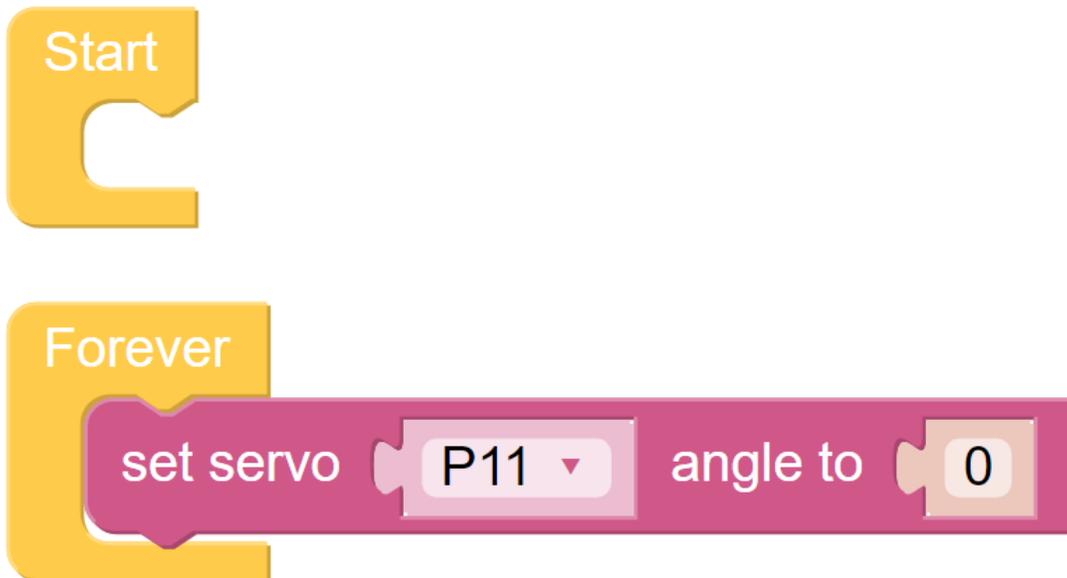
建议使用没有保护板的 18650 电池。否则可能会因为保护板的过流保护而导致机器人断电停止运行。

电池容量？

为了让机器人长时间工作，尽量使用大容量电池。建议购买容量为 3000mAh 及以上的电池。

6.1 安装 EzBlock 操作系统后，舵机不能转到 0° ？

- 检查伺服电缆是否正确连接，Robot HAT 的电源是否打开。
- 按下复位按钮。
- 如果您已经在 EzBlock Studio 中运行了程序，那么 P11 的自定义程序就不再可用。您可以参考下图，在 EzBlock Studio 中手动编写程序，将伺服角度设置为 0。

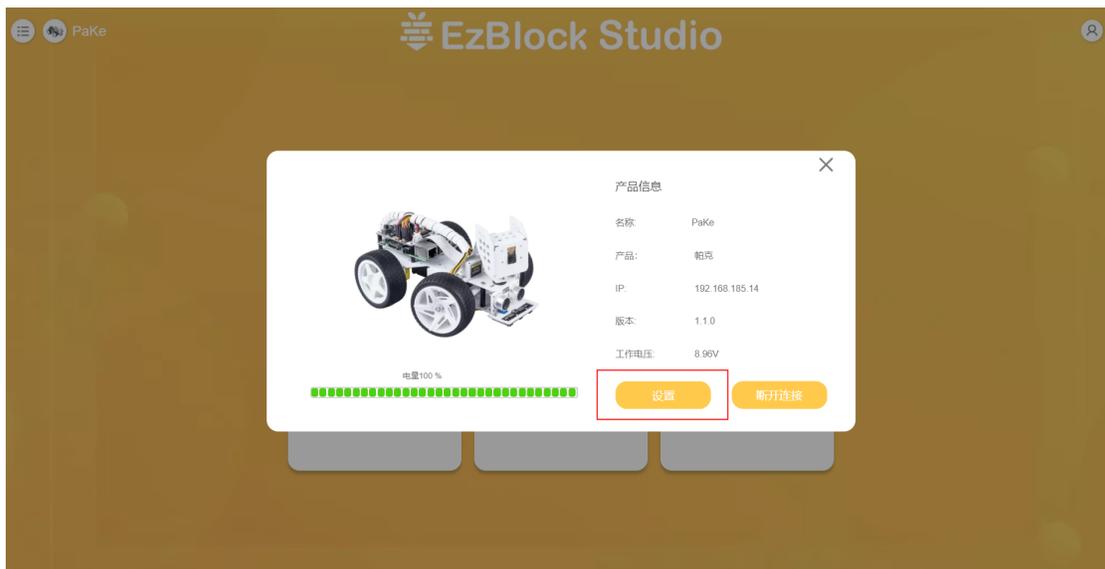


6.2 如何在 EzBlock 重新校准机器人

在首页，点击左上角的产品连接图标。



进入到产品页面之后，点击 **设置**按钮。



进入设置页面后，点击 **校准**按钮就能进入到校准页面，然后按照提示来校准。



6.3 EzBlock 无法连接蓝牙？

- 先检查是否有给你的 SD 卡烧录安装 *EzBlock* 镜像。
- 打开产品的电源开关后，等蓝牙指示灯变得更亮并且出现“zi~”声后，代表树莓派成功启动，此时再去连接。
- 检查您的移动设备的蓝牙是否打开。
- 检查 EzBlock 是否被允许访问设备的位置。
- 有些移动设备还需要打开位置服务。
- 检查电池电量。如果两个电源指示灯都关闭，或者只有一个指示灯在闪烁；电源电量低，请给电池充电。
- 如果以上方法都试过了，请尝试按下 RST 按钮，或重新启动产品和 APP。

6.4 APP 搜索到蓝牙，但无法连接

- 先检查是否有给你的 SD 卡烧录安装 *EzBlock* 镜像。
- 打开产品的电源开关后，等蓝牙指示灯变得更亮并且出现“zi~”声后，代表树莓派成功启动，此时再去连接。
- 检查 ROBOT HAT 上的 BLE 或 USR 灯是否一直亮着（这意味着产品被其他设备连接），如果是，请断开其他设备的连接或重新启动产品。
- 如果以上方法都试过了，请尝试按下 RST 按钮，或者重新启动产品和 APP。

6.5 配置 WIFI 后 APP 无法连接

- 检查国家、账号和密码是否正确。
- 检查该 WIFI 的网络状态。
- 检查电源电量。如果两个电源指示灯都熄灭或只有一个电源指示灯闪烁，则说明电源电量不足，请给电池充电。
- 检查配置的 WiFi 和移动设备连接的 WiFi 是否相同。

6.6 为什么伺服机有时会无缘无故地返回到中间位置？

当舵机被结构物或其他物体挡住，无法到达预定位置时，舵机会进入断电保护模式，以防止舵机被过大的电流烧坏。

断电一段时间后，如果没有给伺服机提供 PWM 信号，伺服机将自动恢复到原来的位置。

CHAPTER 7

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。根据相关规定和版权法，您只能将其用于个人学习、调查、欣赏或其他非商业或非营利目的，不得侵犯作者和相关权利人的合法权利。对于任何个人或组织未经许可将其用于商业利益，本公司保留采取法律行动的权利。