目錄

软件简介 1.1
- 下载与安装 1.1.1
- 软件界面 1.1.2
- 设备连接 1.1.3
- 创作模式 1.1.4

Scratch教程 2.1
- 程序块介绍 2.1.1
- 舞台&角色 2.1.2
- Scratch编写 2.1.3

Python教程 3.1
- Python舞台 3.1.1
- 内置模块 3.1.2
- 画笔 3.1.3
- Arduino 3.1.4
- 人工智能(AI) 3.1.5
- 百变小强(IBB) 3.1.6
- mCookie 3.1.7
- ideaBox 3.1.8

扩展 4.1
- 姿态检测 4.1.1

mCookie套件 5.1
- 连接mCookie 5.1.1
- mCookie使用教程 5.1.2

百变小强 6.1
- 连接小强 6.1.1
- 小强编程 6.1.2

百变小钢 7.1
- 连接小钢 7.1.1
- 小钢编程 7.1.2

附录 8.1
- 更新日志 8.1.1
- 开源软件鸣谢 8.1.2

mDesigner 3

mDesigner 3 是基于Scratch 3 开发的图形化编程软件，延续了Scratch 3 操作简单、所编即所见的设计理念，并增加Arduino、Python代码编程、AI（人工智能）和IoT（物联网）等功能，用户通过拖拽代码块的操作方式，可以轻松控制自己的智能硬件，创作生动有趣创意作品。

图形化编程

延续了Scratch操作简单、所编即所见的设计理念，易学易用，功能更多。

文本编程

支持一键查看积木块对应的Python 代码，学生只需掌握图形化编程知识，即可轻松理解Python语法并学习。还可以直接用Python编辑器编写代码，无需更换其它软件，就能从图形化编程无缝衔接到代码编程。

软硬件结合

支持硬件编程，编程结果通过硬件直接呈现，给学生带来更强的成就感，让课堂教学的内容和形式更丰富。

将AI引入教学

可编程实现如人脸识别，检测年龄，姿态检测等好玩的应用，让孩子轻松理解 AI 技术背后的逻辑和原理，实现更多创造。

更多的IoT应用

接入专为IoT教学设计的云服务，结合电子模块，可实现天气预报、自动浇花、智能开关灯等多种实用、有趣的应用，让学生在生活场景中了解IoT。

全平台支持

Web端、PC客户端、移动Pad端及ChromeBook

功能对比

	Web端	ChromeBook	pc端	Pad端
产品形态	网页 + mConnector PC客户端	网页 + Chrome应用	mDesigner PC客户端	mDesigner 移动客户端
运行环境	Win7、MacOS10.9 以上，Chrome浏览器	ChromeBook本	Win7、MacOS10.9 以上	iOS10、安卓6.0以上
是否需要网络	是	是	无网可用	是
强制登录	是	是	否，不登录可用	是
国内/国际服务切换	1、依赖业务：例：从国内平台选择创作，直接跳转国内服务 2、登录时选择：登录页面用户自己选择		登录时选择	
存储方式	云端	云端	本地 / 云端	云端
发布作品	支持	支持	暂不支持	暂不支持
连接方式	串口 / Mac 蓝牙	串口	串口 / Mac 蓝牙	蓝牙
支持套件	mCookie、百变小强、百变小钢、ideaBox、指间Lite			不支持指间Lite（无蓝牙模块）
Arduino代码编译	云端编译	云端编译	本地编译	云端编译
Python代码	online模式下的角色和硬件			暂不支持（移动版输入代码不便）
UI交互	统一UI、交互、工具栏			独立UI、交互

软件下载

Web版

访问地址：https://mdesigner.microduino.cn/

系统要求：Win7 +，OSX 10.10 +，Chrome浏览器

PC版

下载地址：https://www.microduino.cn/download/mdesigner/

系统要求：Win7 +，OSX 10.10 +

Pad版

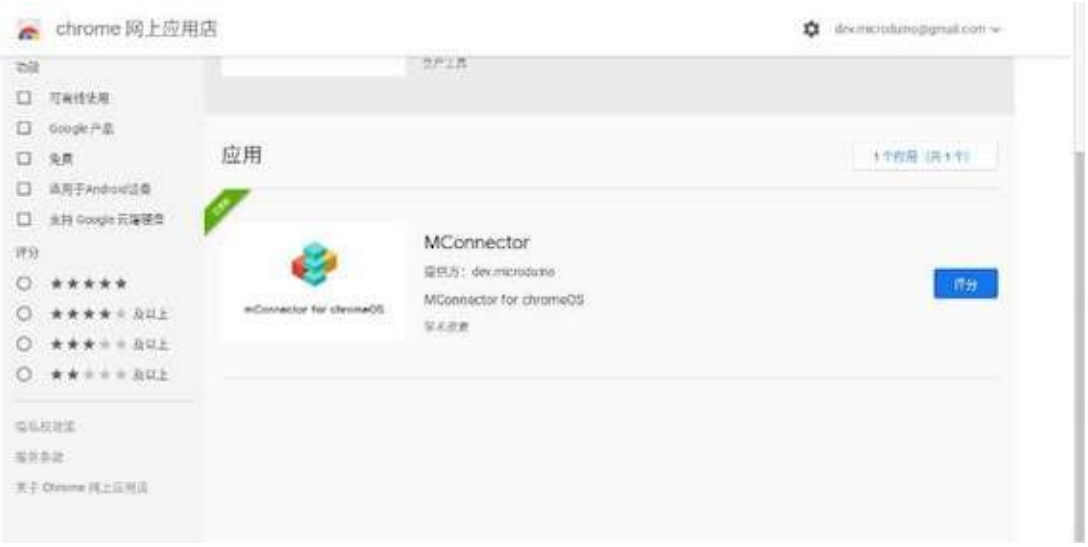
AppStore 搜索 “mDesigner” 下载

扫码下载Pad客户端

系统要求：iOS 10 + ， Android 6.0 +

ChromeBook版

Chrome 网上应用商店 搜索 “MConnector” 下载

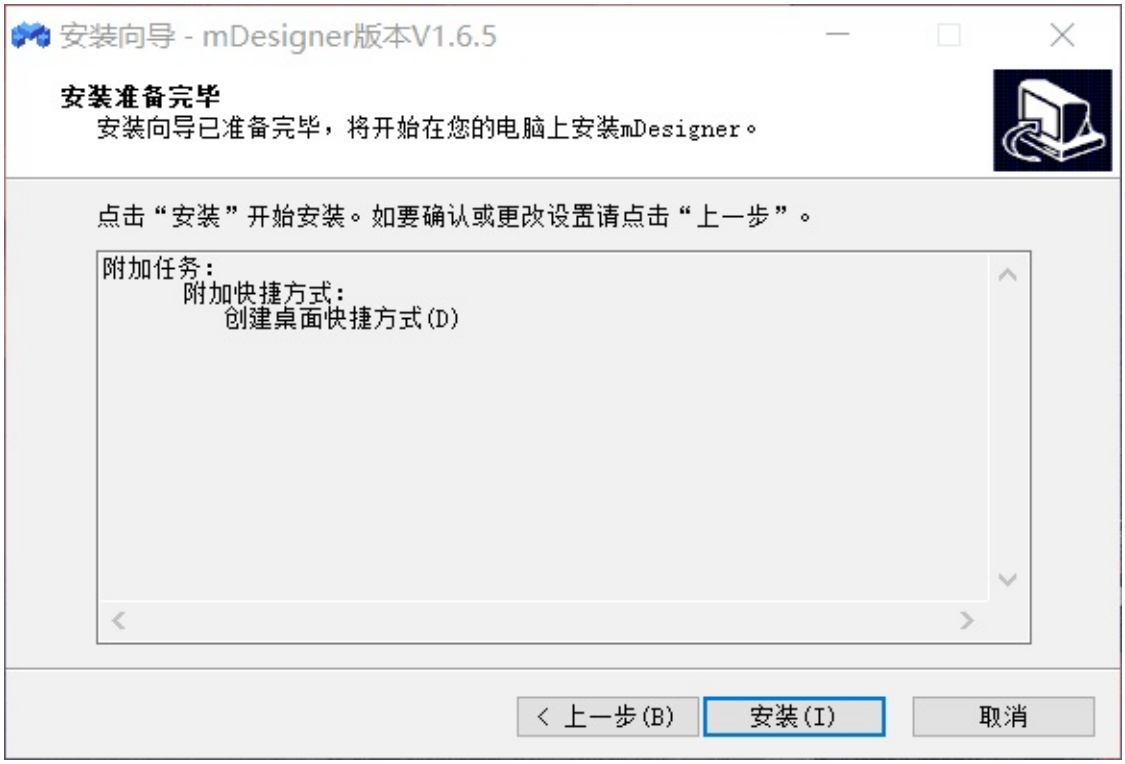


系统要求：ChromeBook电脑

软件安装

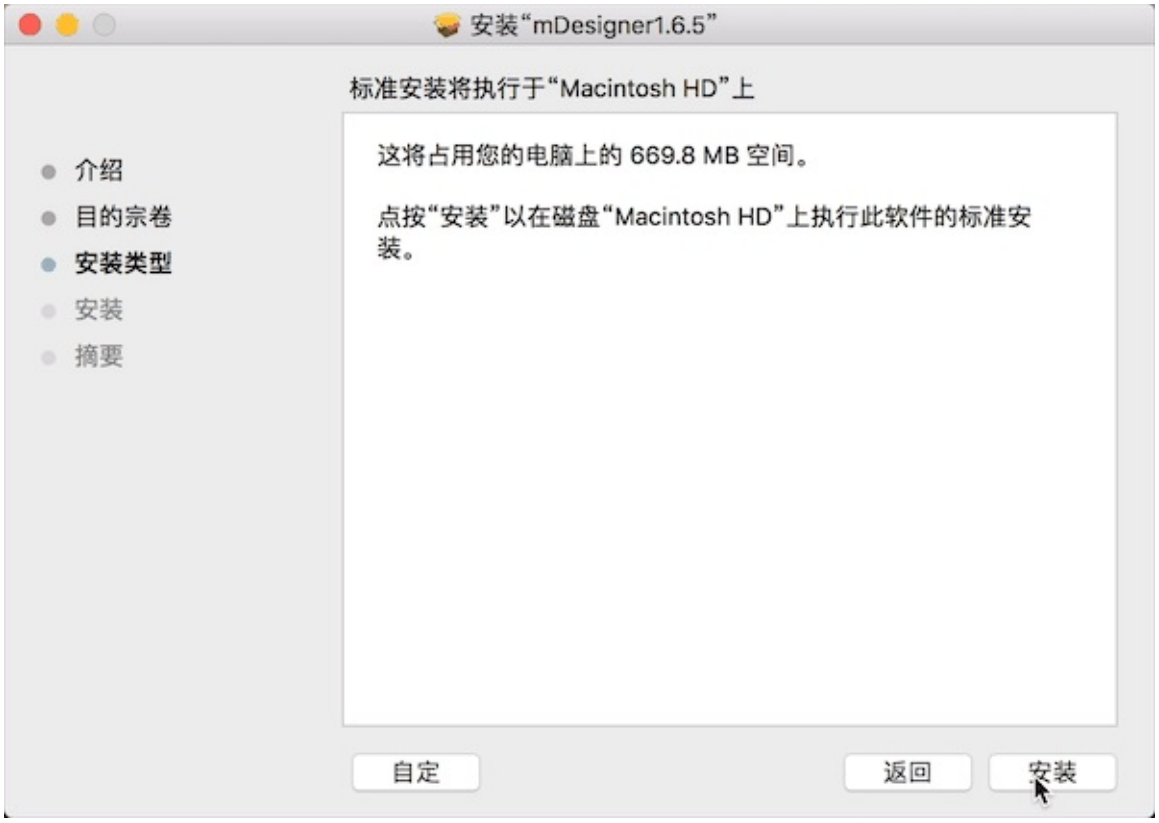
windows安装

按步骤开始安装软件即可。

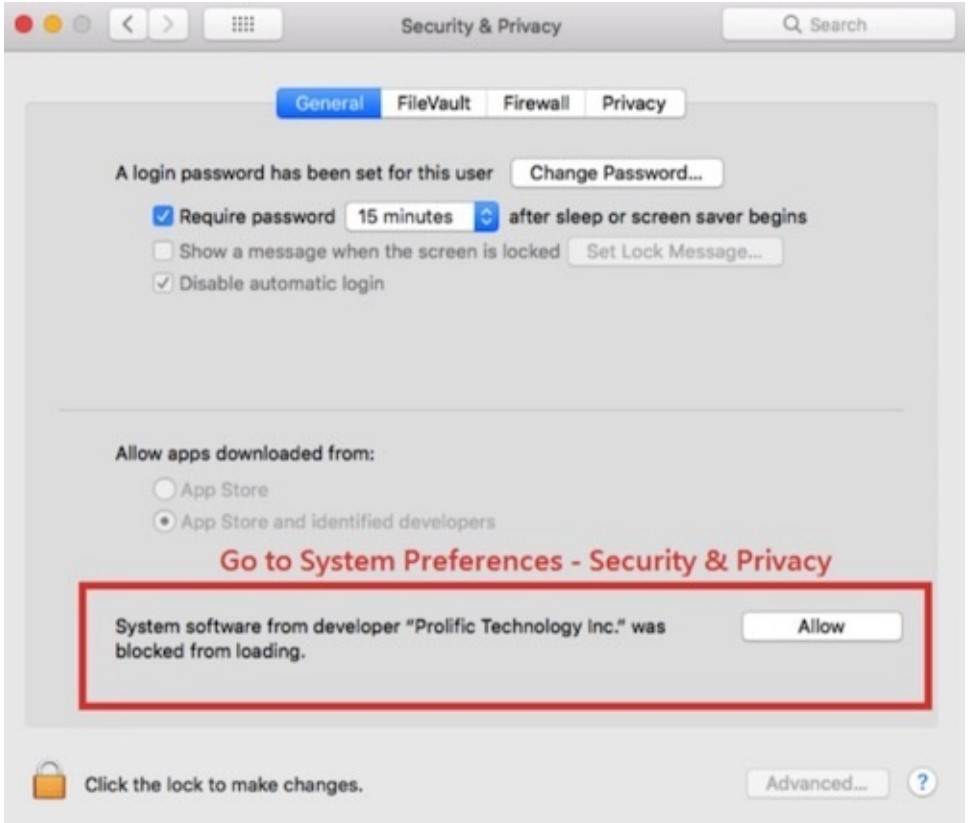


Mac 安装

按步骤开始安装, 安装完成后，需要**重启电脑**。

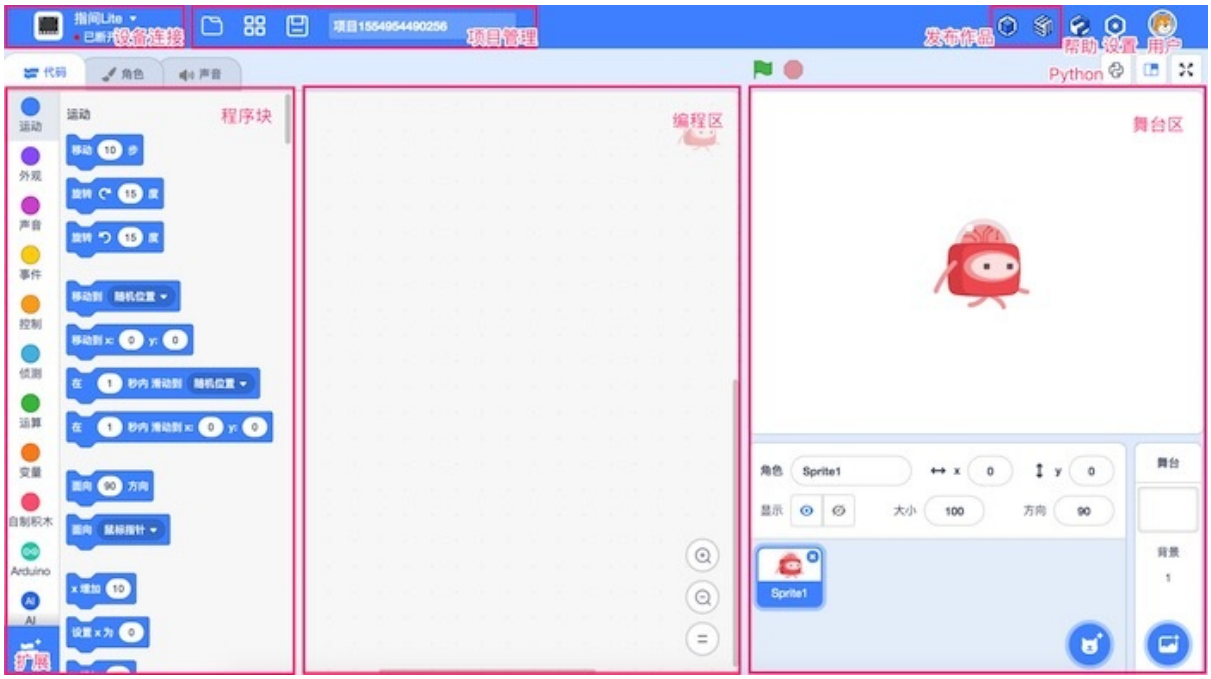


注意：打开「系统偏好设置」-「安全性与隐私」，有如下字样，点击「允许」


Security & Privacy
Search
General
FileVault
Firewall
Privacy
A login password has been set for this user
Change Password...
Require password
15 minutes
after sleep or screen saver begins
Show a message when the screen is locked
Set Lock Message...
Disable automatic login
Allow apps downloaded from:
App Store
App Store and identified developers
Go to System Preferences - Security & Privacy
System software from developer "Prolific Technology Inc." was blocked from loading.
Allow
Click the lock to make changes.
Advanced...

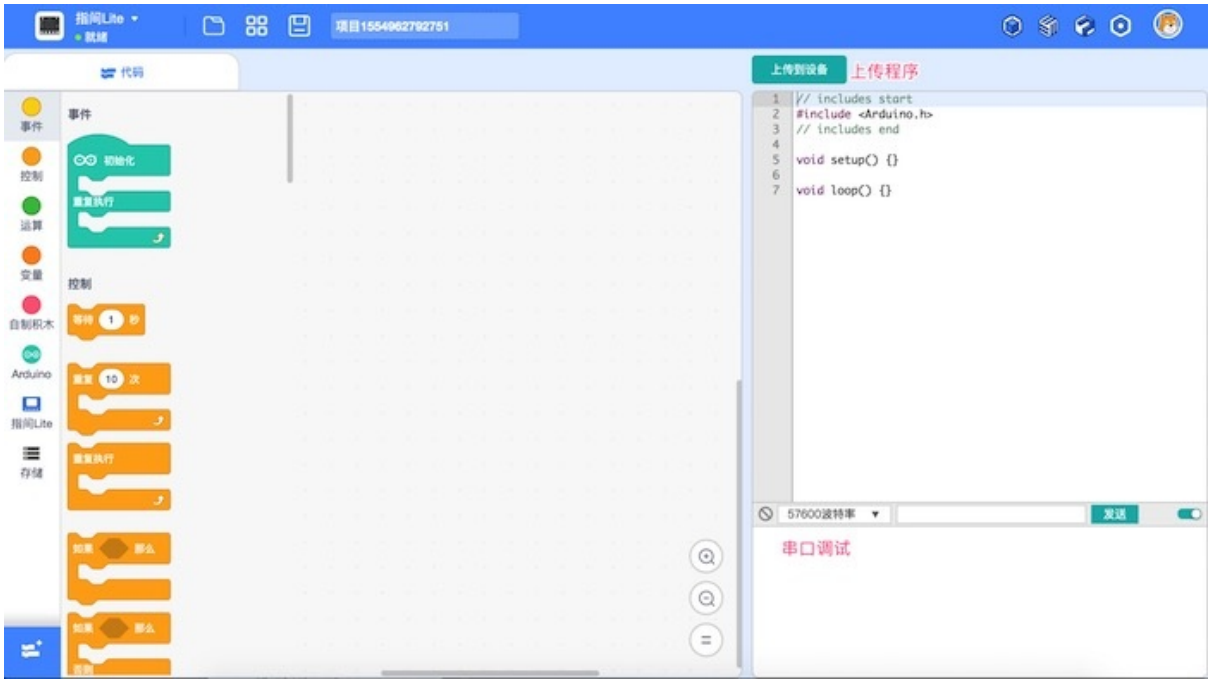
软件界面

「实时模式」online



- **设备连接**：套件选择、连接、烧录固件、重置设备等功能；
- **项目管理**：项目的新建、打开、复制、保存、重命名、删除等功能；
- **发布作品**：项目发布到ideaxlab平台，查看作品页；
- **帮 助**：查看用户使用手册；
- **设 置**：切换语言、意见反馈、版本号、在线更新、更新日志等功能；
- **用 户**：注册、登录、修改密码、退出；
- **Python**：打开或关闭Python编辑窗口；
- **程 序 块**：提供编程所需的程序块，可以按照分类及颜色查找；
- **编 程 区**：对程序块进行操作的区域，可以将程序块拖放到这个区域来编写程序；
- **舞 台 区**：展示程序效果，添加角色与背景等功能；
- **扩 展**：添加或取消功能扩展；

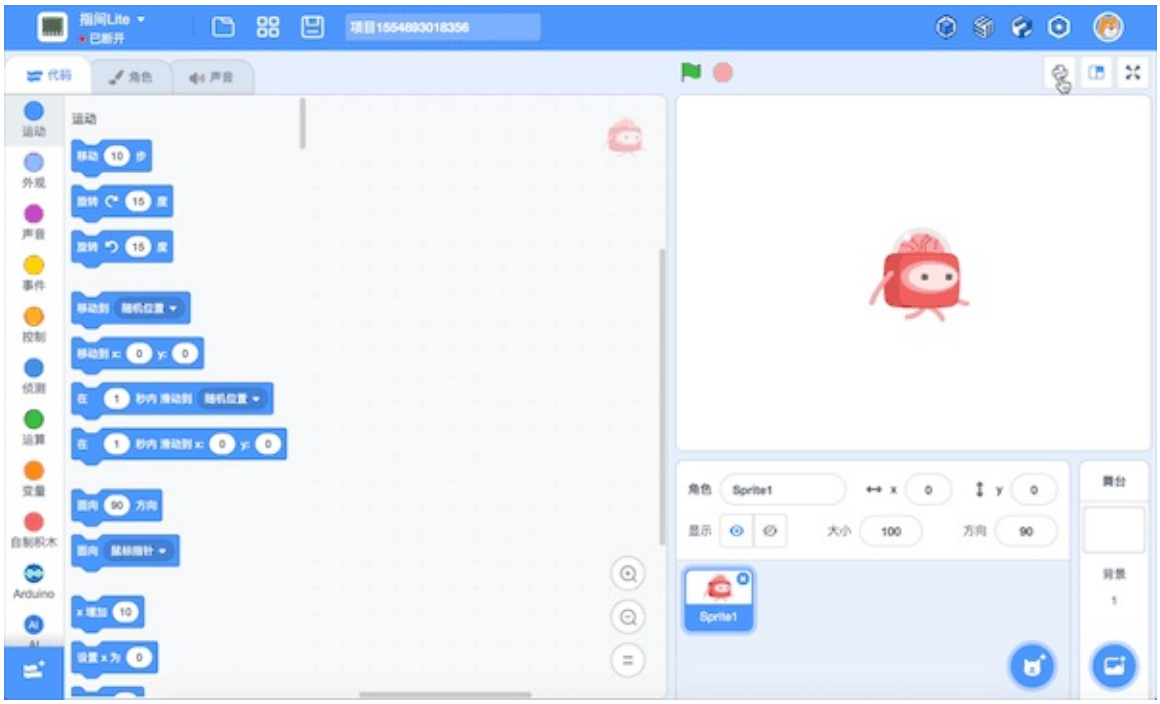
「上传模式」offline



- **上传程序**：把程序烧录到硬件设备中；
- **串口调试**：实时与硬件设备通信调试；

自定义布局

根据编程需要，拖拽改变窗口大小，适应各种屏幕



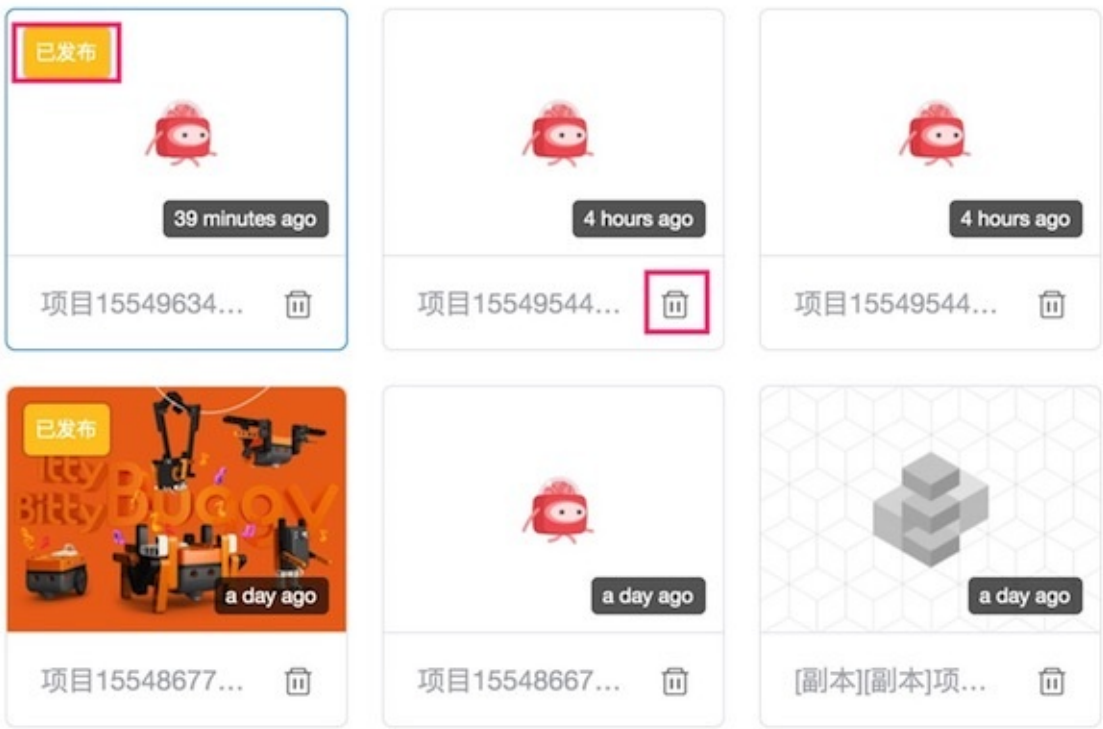
注册与登录

1. **注册：**点击右上角「头像」标志，选择「注册」，输入邮箱/手机号、收到的验证码并设置密码，就可以完成注册；
2. **登录：**点击右上角「头像」标志，输入邮箱/手机号、密码即可登录，也可以选择第三方账号登录。

注意： 中国版与国际版是不同的账号体系，请根据所在地区选择相应版本注册/登录

项目管理

同一用户在PC版、web版、Chromebook版、Pad版，存储项目会同步显示；



- **打开项目**：选择项目，单击即可打开该项目；
- **删除项目**：删除后，无法找回；
- **已发布**：表示该项目已经发布到ideaxlab平台，删除需要先到平台取消发布；
- **蓝色框**：表示当前打开的项目；

设备连接

1.选择套件:

默认勾选上一次所选的套件，初始为mCookie。

2.选择连接方式：

Windows电脑只支持USB连接，Mac电脑支持蓝牙和USB连接。

注意：连接之前，检查设备开关是否打开，电量是否充足。

3.选择设备：

选择要连接的设备，点击「连接」，如果固件不对会自动烧录相应的固件。

4.连接成功：

连接成功并固件正确，显示「就绪」。下拉可选择

- **断开连接：**断开设备连接
- **重置设备：**清空online模式/调试模式传入设备中的指令

web版连接注意事项

初次使用，会提示下载美科硬件助手，安装并启动后，再次连接即可发现设备

创作模式

新建项目，可以选择「实时模式」或「上传模式」；

实时模式(online)

指的是烧录一次实时执行的固件，点击「绿旗」标志，即可执行所编写的程序。（支持**Scratch**与**Python**语言）

特点：

1. 只需要烧录一次实时执行的固件；
2. 可以与软件舞台交互（比如控制精灵的运动）；
3. 不可脱离电脑运行；

注意：需要烧录相应产品固件，如果没有固件会自动烧录

上传模式(offline)

指的是将用户编写好的程序上传到硬件主控板中，之后硬件脱离电脑也能执行程序。（支持**Scratch**与**Arduino C**语言）

特点：

1. 每个新程序都要重新上传固件；
2. 不可以与软件舞台交互；
3. 可以脱离电脑运行；

程序块介绍

6种程序块类型：

1、**启动程序块**：是启动每个脚本，它们有圆形的顶部和凸起的底部，你只能在它们的下面放置块。

2、**执行程序块**：是执行命令的模块，它们在顶部有一个缺口，底部有一个凸起，可以在它们的上面和下面放置块。

3、**C型程序块**：这些程序块也被成为“包块”，这些C形块可以循环执行其内部的程序或检查条件是否为真。

4、**逻辑程序块**：是条件语句，它们不是真的就是假的，这好比问你的朋友：“2+2=4吗？”他们会告诉你“是”或“否”。

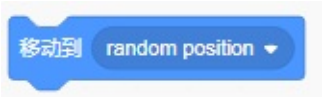
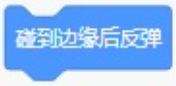
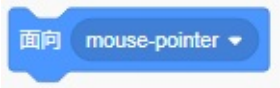
5、**报告程序块**：是数值，它们是圆形变边的，报告程序块可以保存数字和字符串。比如问一个朋友，“2+2是什么？”他们会回答“4”。它不只可以回答方程式，它还可以回复一个变量，例如“你的年龄是多少？”他们可能会回答：“15。”

6、**停止程序块**：是结束脚本的程序块，它们有有凹槽的顶部和平的底部，所以你不能在它们下面放置任何块。

8种基本Scratch程序块

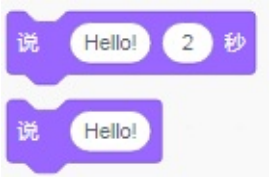
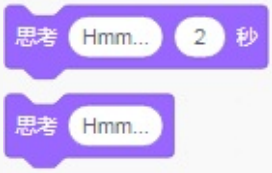
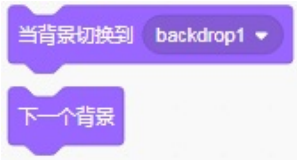
1、**运动**

功能	截图	说明
移		

移动		让角色在舞台上移动一定距离，数值越大，移动越远
移动		让角色移到对应位置
移动		让角色在规定时间内滑行到对应位置
移动		通过x、y坐标变化更简便地设置运动轨迹
移动		移到对象位置
移动		角色在运动时如果碰到舞台边缘则反弹，限定角色运动在舞台中
方向		旋转角度
方向		规定角色面向，与移动指令结合使用可以让角色朝着指定方向移动
方向		设定角色面朝着对象

2、外观

功能	截图	说明
说		让角色说话，有时间限定则限定时长，无时间则一直

话		说
说话		让角色思考，有时间限定则限定时长，无时间则一直思 考
外观		让角色显示或者隐藏
造型		角色可以有很多造型，造型切换让角色做出很多不同的动作
造型		结合循环与等待使用，让角色流畅运动
特效		给角色增加视觉特效
背景		一个程序可以有很多背景，通过切换背景让角色处于不同的场景中

3、**声音**

播放/停止声音	播放声音 Meow 播放声音 Meow 直到播放完毕 停止所有声音	播放/停止背景声音和角色音效
音效	将音效 pitch 增加 10 设定音效 pitch 为 100 清除音效	增加声音特效
音量	将音量增加 -10 将音量设置为 100 % 音量	设置音量大小

4、**事件**

功能	截图	说明
发生	当 被点击	直接开始，点击绿色旗子就可以开始下面的代码
发生	当 空格 键被按下时	当按下某个指定按键的时候开始下面的代码，默认是空格键。还可以使用下拉菜单中的其他按键控制程序。
发生	当角色被点击时	点击舞台中的角色，会运行下面的程序
发生	当背景切换到 backdrop1	当舞台背景被切换到制定背景时，执行下面程序
广播	当 loudness > 10 ✓ loudness 计时器	当声音或计时器大于指定值的时候，执行下面程序
广播	当接收到 message1 广播 message1	程序运行到广播模块时，会同时执行一列以“当接收到”开头的新程序

广播	广播 message1 并等待	程序运行到广播并等待模块时，会执行一列以“当接收到”开头的新程序，新程序运行完，才能执行广播并等待模块后面的程序

5、**控制**

功能	截图	说明
等待	等待 1 秒	等待相应时间再继续执行下面程序，或持续以上程序相应时间
等待	等待直到	菱形块中为执行条件，当满足条件时，执行下面程序，否则一直等待
循环	重复执行直到	菱形块中为执行条件，当满足条件时，执行下面程序，否则重复执行内部程序
循环	重复 10 次	执行内部程序相应次数，然后跳出，执行下面程序
循环	重复执行	一直重复执行内部程序
判断	如果 那么	菱形块中为条件，当满足条件时执行内部程序，之后再执行下面程序，不满足条件直接跳过，执行下面程序
判断	如果 那么 否则	菱形块中为条件，当满足条件时执行那么中的内部程序，不满足条件时执行否则中的内部程序
终止	停止 全部 ✓ 全部 当前脚本 角色的其他脚本	停止全部：停止所有的程序；停止当前脚本：停止当前成序列；停止角色的其他脚本：停止当前程序列以外的所有程序
克隆	克隆 myself	克隆当前角色或其他角色
克隆	删除本克隆体	删除所有克隆体

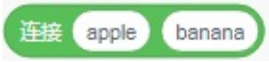
克隆	当作为克隆体启动时	控制克隆体执行下面的程序

6、侦测

功能	截图	说明
询问	问 What's your name? 并等待	可以结合使用 连接 apple banana 一起
条件	碰到 mouse-pointer ? 碰到颜色 ? 颜色 碰到 ? 按键 空格 被按下? 鼠标被按下?	程序之间侦测到的事件
条件	Stage 的 x 坐标	反映某一个角色的xy坐标
计时器	计时器 重置计时器	计时功能

7、运算

功能	截图	说明
运算	+ - * /	实现加减乘除运算
比较	> 100 < 100 = 100	实现比较大小的功能
与		

或非		立时，程序不执行；
与或非		逻辑与模块：只有当菱形模块中条件都满足时，程序执行，否则不执行；
与或非		不是模块：当不满足条件时，程序执行，否则不执行
随机数		在指定范围内随机取一个整数，包括两端数
连接		将两个量进行连接
检测		检测一个量中的指定位置字符的值
检测		检测一个量的长度值
判断		判断一个量中是否包含另外一个量
求值		求数值相除的余数
求值		将数值的小数部分四舍五入

求值		对数值取绝对值、向下取整、向上取整、平方根等

8、**变量**

功能	截图	说明
定义		创建变量
定义		定义变量名，确定创建变量
数据		变量赋值
数据		将变量值加一
数据		调用变量值

显示/隐藏	显示变量 apple 隐藏变量 apple	在舞台中显示或隐藏变量
定义	新建数组	创建数组
定义		定义数组名，确定创建数组
数据	添加 thing 到 banana	添加一个量到数组里
数据	删除 banana 的第 1 项	删除数组中的第一项
数据	将 1 插入到 banana 的第 1 项	将一个量插入到数组中指定位置
数据	将 banana 的第 1 项替换为 1	将数组中指定项被替换为一个量
数据	banana 的第 1 项	获取数组中的指定项的值
数据	item # of thing in banana	判断一个量在数组中的位置
数据	banana 的长度	获取数组的长度值
数据	banana 包含 thing ?	判断数组中是否包含指定量

数据		在舞台中显示或隐藏数组
数据		在舞台中增加或删除数组中的项

舞台和角色

舞台：Scratch可以理解为是一种校内戏剧，是观众看戏的地方。

角色：可以理解为是剧中的演员或人物，背景是舞台背景，他们一起登上舞台。

下面是对于一些名词的解释：

- 背景：这是舞台的背景。舞台中只有一个背景。它可以是改变的，你也可以自己设计。
- 角色：角色可以理解为是演员或人物。舞台上可以有多个角色，你也可以自己设计角色形象。
- 绿色旗帜：该旗帜被点击时，所有“当绿色旗帜被点击”下的模块启动。
- 停止信号：被点击时，停止所有运行中的程序。

角色和背景： 这一区域你可以选择、编辑和创建角色和背景。点击它可以选择一个角色或是背景。

声音窗格:

造型窗口： 当选择当前角色时，跳转到造型面板。你可以在该区域为当前选定的角色设计造型。造型是一个角色不同的形象/外观。你可以使用造型窗口创建角色动画或改变你的角色形象。

代码
角色
声音
造型名称
costume1
编辑工具
当前选择的
角色的造型
造型样子
为当前选定的角色
添加新的造型
当角色被选择
Sprite1

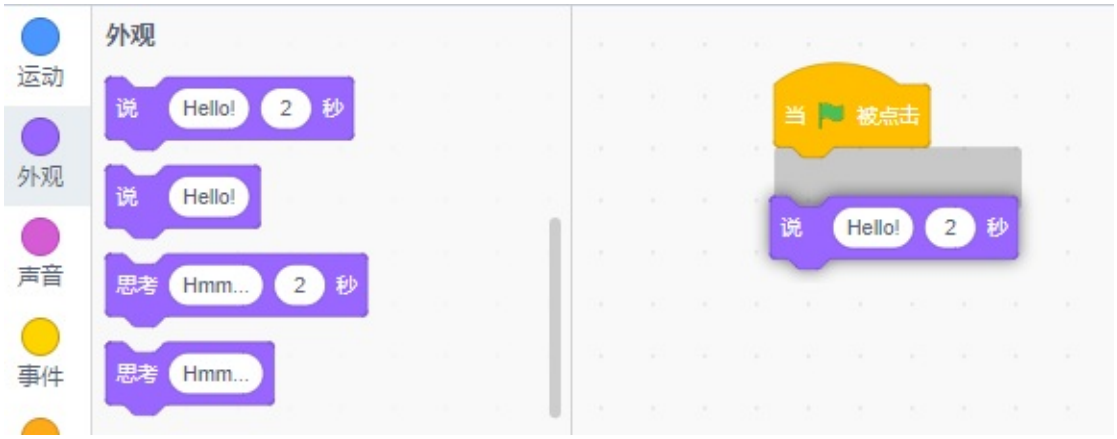
Scratch代码编写

1、加入

1. 按住鼠标左键选择代码块
2. 拖动代码块到编辑区

2、组合

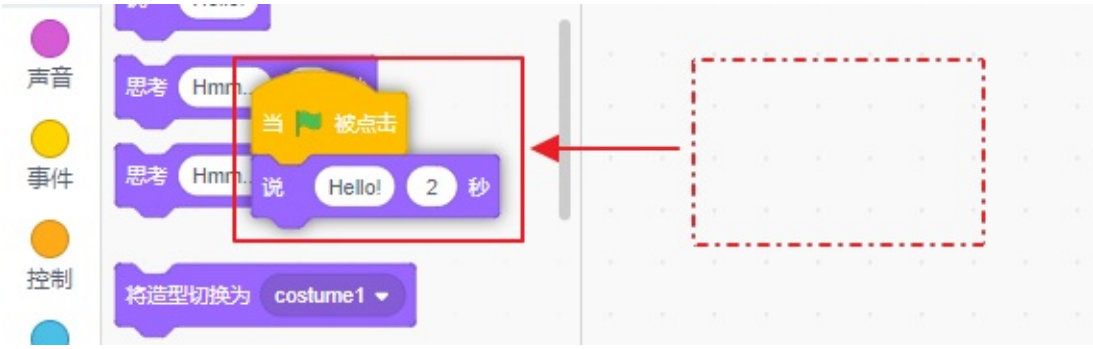
1. 拖动要组合的代码块到要组合的位置
2. 出现灰色连接图后放下代码块即可



3、删除

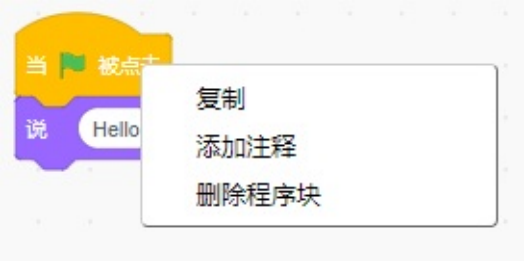
方法一：

1. 按住鼠标左键选择要删除的代码块
2. 拖动代码块到脚本（代码）区



方法二：

1. 右键点击要删除的代码
2. 左键点击“删除代码块”



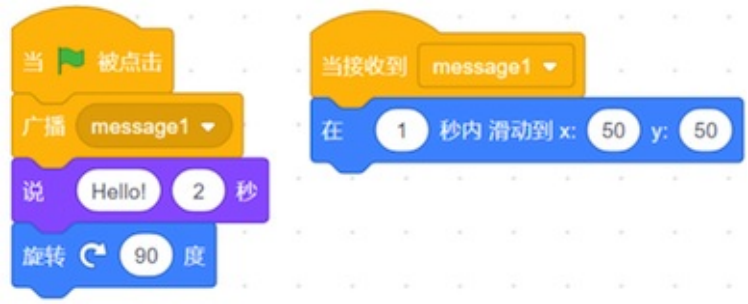
4、多组代码的关系

多线程关系：即在同一时间执行多个程序的关系

4、广播关系

当程序执行到广播模块时，开启另一列接收广播程序。此时，广播模块所在程序和接收到广播的程序同时执行。

程序既可以控制舞台角色又可以控制硬件，每个角色中都可以编写程序，多个角色之间通过广播、变量和侦测模块进行联系，可以实现多线程控制。硬件只可以用单线程程序控制。



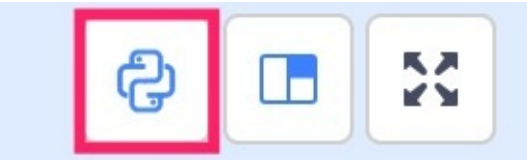
Python控制舞台角色

通过使用Python操控舞台角色，能达到和使用Scratch程序块一样的效果，您可以：

- 在Scratch的基础上学习Python；
- 比拖放语句块更方便地编写有一定规模的代码，比如实现算法；

开始使用Python

点击右上角Python标志即可切换到Python编程区



程序块转换为Python

拖动程序块会生成相对应的Python代码

运动

移动：move(距离)

```
sprite.move(10)
```

右转：right(角度)

```
sprite.right(15)
```

左转：left(角度)

```
sprite.left(15)
```

让角色移动到屏幕上指定的x,y坐标

```
sprite.gotoxy(10,10) #移动到(10,10)坐标处
```

让角色移动到屏幕上指定位置

```
sprite.moveto('random') #移动到随机位置
sprite.moveto('mouse') #移动到鼠标处
```

让角色在规定时间内滑动到屏幕上指定的x,y坐标

```
sprite.glide(9,-110,1) #在1秒内滑动到(9, -110)坐标处
```

让角色在规定时间内滑动到屏幕上指定位置

```
sprite.glideto('random',1) #在1秒内滑动到随机位置
sprite.glideto('mouse',1) #在1秒内滑动到鼠标处
```

设置和获取角色朝向的角度

```
sprite.direction(105) #设置角色的朝向角度是105度
print(sprite.get_direction()) #输出角度
```

让角色朝向指定的角色

```
sprite.towards('mouse') #朝向鼠标指针
sprite.towards('Sprite1') #朝向Sprite1
```

设置和获取角色在舞台上的x, y坐标

```
sprite.set_x(98)  #设置x坐标
sprite.set_y(-40) #设置y坐标
sprite.x_add(10)  #x坐标增加10
sprite.y_add(10)  #y坐标增加10
print(sprite.get_x()) #获取x坐标
print(sprite.get_y()) #获取x坐标
```

设置角色的旋转模式

```
sprite.rotation_mode('left-right') #左右翻转
sprite.rotation_mode('all-around') #任意旋转
sprite.rotation_mode("none")       #不能旋转
```

角色碰到舞台边缘则反弹

```
sprite.bounce()
```

外观

让角色在屏幕上「说出」或「思考」指定的文字

```
sprite.say('Hello!',2) #2秒后消失
sprite.say('Hello!') #一直显示

sprite.think('Hmm...',2) #2秒后消失
sprite.think('Hmm...') #一直显示
```

设置和切换造型

```
sprite.set_costume('avery-b') #改变角色的造型，名称必须是该角色已经有的造型名称
sprite.next_costume() #将角色的造型切换为下一个造型
```

设置和切换舞台背景

```
sprite.set_backdrop('backdrop1')  #改变背景，名称必须是已经有的背景名称
sprite.set_backdrop('next-backdrop') #切换到下一个背景
sprite.set_backdrop('previous-backdrop') #切换到上一个背景
sprite.set_backdrop('random-backdrop')   #切换到随机
sprite.next_backdrop()  #切换到下一个背景
```

角色的「显示」与「隐藏」

```
sprite.show() #显示
sprite.hide() #隐藏
```

常用内置模块

常用内置模块主要有：

```
import time
import random
import datetime
import re
import json
import sprite
```

> 使用前，请先导入相应的模块，本篇教程及之前函数库的教程，皆是以Python 3.x为基础版本进行编写

一、 time

时间模块，用于时间的获取与转换

1 time.sleep()

time.sleep()用于在程序执行时进行指定时间的休息，单位为秒。例如：

```
for i in range(5):
    print(i)
    time.sleep(1)
```

运行结果：

```
0
1
2
3
4
```

> 这段代码的功能为：每隔1秒打印一个数字，其中range(5)用于生成一个迭代对象，此处可理解为是生成了一个列表range(5) => [0, 1, 2, 3, 4]，range(5)默认从0开始，共产生5次。可以指明开始点，例如rang(1, 6) = [1, 2, 3, 4, 4](左包括，右不包括); for .. in ..是用于遍历可迭代对象

2 time.localtime()

time.localtime()用于获取本地时间并生成为一个时间元组

```
print(time.localtime())
```

运行结果：

```
time.struct_time(tm_year=2018, tm_mon=8, tm_mday=20, tm_hour=18, tm_min=3, tm_sec=18, tm_wday=0, tm_yday=232, t
m_isdst=0)
```

索引值(index)	属性(Attribute)	值(Values)
0	tm_year(年)	(例如:2015)

1	tm_mon(月)	1-12
2	tm_mday(日)	1-31
3	tm_hour(时)	0-23
4	tm_min(分)	0-59
5	tm_sec(秒)	0-61(60代表闰秒,61是基于历史原因保留)
6	tm_wday(星期几)	0-6(0表示星期一)
7	tm_yday(一年中的第几天)	1-366
8	tm_isdst(是否为夏令时)	0,1,-1(-1代表夏令时)

3 time.time()

返回当前时间的时间戳(1970元年后的浮点秒数)，可用于测出某段代码的运行耗时

```
start_time = time.time()

a = 0
for i in range(1, 101):
    a += i

end_time = time.time()

print('a的值为:%s' %(a))
print('运行for循环共耗时%s' %(end_time - start_time))
```

运行结果:

```
a的值为:5050
运行for循环,共循环100次，总耗时为:0.000190258026123
```

> a += i 是一种简化写法，相当于 a = a + i，类似的用法也可适用于减法(a -= i)、乘法(a *= i)、除法(a /= i)等

二、 randome

主要拥有随机数生成、指定范围内随机数的生成、随机选择等功能

1 random.random()

生成0-1之内的随机浮点数

```
for i in range(5):
    print(random.random())
```

运行结果:

```
0.0692932045127
0.00866250612738
0.223208213488
0.215825885145
0.815881998416
```

> random.uniform()用于生成指定范围内的随机数，例如random.uniform(1, 10)，生成1-10之内的随机的数

2 random.randint()

生成指定范围内的随机整数

```
for i in range(5):
    print(random.randin(10, 100))
```

运行结果：

```
23
67
60
34
83
```

> random.randint()填写范围时，第一个参数为下限，第二个参数为上限，下限参数应小于上限

3 random.choice()

随机选取可迭代对象中的某一元素

```
a = 'Kuture'
b = ['Mr.Li', 'Keven', 'Lily', 'Potter', 'Marry']
c = (128, 64, 1024, 512, 16)


# 定义方法
def rand_method(num):

    for i in range(num):

        rand_str = random.choice(a)
        rand_list = random.choice(b)
        rand_tuple = random.choice(c)

        print('Str: ', rand_str)
        print('List: ', rand_list)
        print('Tuple: ', rand_tuple)
        print('-' * 15)

# 调用方法
rand_method(3)
```

运行结果：

```
Str:  K
List:  Mr.Li
Tuple:  16
---------------
Str:  t
List:  Marry
Tuple:  128
---------------
Str:  t
List:  Lily
Tuple:  1024
---------------
```

> 在Python中常见的迭代对象有字符串(string)、列表(list)、元组(tuple)、字典(dict)等。此段代码中使用了方法，方法相当于是将一段代码放在了一个代码块里，利于减少对于常用功能的重复编写。此方法的功能是根据传入的参数重复相应数量的循环，其中a、b、c分别是字符串(string)、列表(list)、元组(tuple)类型，在for循环中，使用random.choice()随机从其中选择出一个元素。print()在终端打印相应的结果，print('-' * 15)中'-' * 15表示字符串重复显示多少次。

三、datetime

datetime 常用来获取日期、日期间隔、时区等信息，此处介绍常用的两个模块date与datetime中的date.today()与datetime.now()

1 datetime模块简介

datetime模块中包含如下类：

类名	功能说明
date	日期对象,常用的属性有year, month, day
time	时间对象
datetime	日期时间对象,常用的属性有hour, minute, second, microsecond
datetime_CAPI	日期时间对象C语言接口
timedelta	时间间隔，即两个时间点之间的长度
tzinfo	时区信息对象

datetime模块中包含的常量：

常量	功能说明	用法	返回值
MAXYEAR	返回能表示的最大年份	datetime.MAXYEAR	9999
MINYEAR	返回能表示的最小年份	datetime.MINYEAR	1

2 datetime.date.today()

获取当前日期

```
# 创建实例化对象
a = datetime.date.today()

b = a.year
c = a.month
d = a.day

print('Date: ', a)
print('Year: ', b)
print('Month: ', c)
print('Day: ', d)
```

运行结果：

```
Date: 2018-08-21
Year: 2018
Month: 8
Day: 21
```

> a 为由datetime.date.today()创建的实例化对象

3 datetime.datetime.now()

获取当前日期时间，包含年月日时分秒

```
a = datetime.datetime.now()
time.sleep(random.randint(1, 10))
b = datetime.datetime.now()

print('时间差: ', b - a)
```

运行结果：

```
时间差:  0:00:03.005850
```

> 随机产生时间休息时间，b - a 表示a点与b点之间的时间差,如果不想显示格式可以：(b-a).seconds => 3

四、re

re是python中的正则表达式模块，常用于匹配查询

> 正则表达式本身是一种小型的、高度专业化的编程语言，而在python中，通过内嵌集成re模块，可以直接用来实现正则匹配。正则表达式模式被编译成一系列的字节码，由用C编写的匹配引擎执行。

re.match() 与 re.search()

```
a = 'Kuture is a Developer for Microduino'

result_search = re.search(r'for(.*)', a)
result_match = re.match(r'Kuture is a Developer for(.*)', a)

if result_search:

    print('Search: ', result_search.group(1))
if result_match:

    print('Match: ', result_match.group(1))
```

运行结果：

```
Search: Microduino
Match: Microduino
```

> result__search 使用的是re.search(),其中'for(.*)'代表从a字符串中查找到包含for开始的字符串，并截取for后面的任何元素，'.'代表匹配任何元素，'*'代表匹配数量为0-∞。result_match 使用的是re.match()，match需要从头开始匹配，匹配符功能与search()相同。_mathc()是从字符串开始位置进行匹配，search()会在字符串内查找匹配

普通字符和11个元字符

普通字符	**匹配自身**	**abc**	**abc**

.	匹配任意除换行符"\n"外的字符(在DOTALL模式中也能匹配换行符	a.c	abc				
\	转义字符，使后一个字符改变原来的意思	a.c;a\c	a.c;a\c				
*	匹配前一个字符0或多次	abc*	ab;abccc				
+	匹配前一个字符1次或无限次	abc+	abc;abccc				
?	匹配一个字符0次或1次	abc?	ab;abc				
^	匹配字符串开头。在多行模式中匹配每一行的开头	^abc	abc				
$	匹配字符串末尾，在多行模式中匹配每一行的末尾	abc$	abc				
\		或。匹配\	左右表达式任意一个，从左到右匹配，如果\	没有包括在()中，则它的范围是整个正则表达式	abc\	def	abcdef
{}	{m}匹配前一个字符m次，{m,n}匹配前一个字符m至n次，若省略n，则匹配m至无限次	ab{1,2}c	abcabbc				
[]	字符集。对应的位置可以是字符集中任意字符。字符集中的字符可以逐个列出，也可以给出范围，如[abc]或[a-c]。[^abc]表示取反，即非abc。 所有特殊字符在字符集中都失去其原有的特殊含义。用\反斜杠转义恢复特殊字符的特殊含义。	a[bcd]e	abeaceade				
()	被括起来的表达式将作为分组，从表达式左边开始没遇到一个分组的左括号“（”，编号+1. 分组表达式作为一个整体，可以后接数量词。表达式中的\	仅在该组中有效。	(abc){2} a(123\	456)c	abcabca456c		

这里需要强调一下反斜杠\的作用：

- 反斜杠后边跟元字符去除特殊功能；（即将特殊字符转义成普通字符）
- 反斜杠后边跟普通字符实现特殊功能；（即预定义字符）
- 引用序号对应的字组所匹配的字符串。

预定义字符集

\d	**数字:[0-9]**	**a\bc**	**a1c**
\D	非数字:[^\d]	a\Dc	abc

\s	匹配任何空白字符:[<空格>\t\r\n\f\v]	a\sc	a c
\S	非空白字符:[^\s]	a\Sc	abc
\w	匹配包括下划线在内的任何字字符:[A-Za-z0-9_]	a\wc	abc
\W	匹配非字母字符，即匹配特殊字符	a\Wc	a c
\A	仅匹配字符串开头,同^	\Aabc	abc
\Z	仅匹配字符串结尾，同$	abc\Z	abc
\b	匹配\w和\W之间，即匹配单词边界匹配一个单词边界，也就是指单词和空格间的位置。例如， 'er\b' 可以匹配"never" 中的 'er'，但不能匹配 "verb" 中的 'er'。	\babc\b a\b!bc	空格abc 空格a!bc
\B	[^\b]	a\Bbc	abc

五、json

json常用来进行JSON对象的编码与解码，此处仅讲解对JOSN字符串解码为Python对象

> 目前mDesigner中尚不支持将Python对象编码成JSON字符串

```
datas = "[{'name': 'Mr.Zhang', 'age': 18, 'gender': 'male'},{'name': 'Lily', 'age': 17, 'gender': 'famale'},{'name': 'Maly', 'age': 19, 'gender': 'famale'}]"

print('loads_before: ', type(datas))

datas = datas.replace("'", '"')
datas_loads = json.loads(datas)

print('dumps_after: ', type(datas_loads))
```

> json.loads()不支持单引号，所以使用前需将单引号转换为双引号。datas.replace("'",'")用于替换单引号

运行结果：

```
loads_before:  <class 'str'>
dumps_after:  <class 'list'>
```

> 从结果中可以看到，datas转换前是一个字符串，转换后成了一个列表对象

画笔

画笔库中所有扩展方法及属性皆依赖于pen扩展包，使用前请导入pen

```
import pen
```

1 全部擦除

```
pen.clear()
```

2 图章

```
pen.stamp()
```

3 落笔

```
pen.pen_down()
```

4 抬笔

```
pen.pen_up()
```

5 设置颜色

```
pen.set_pen_color_to_color('#3da149')
```

6 画笔加粗

```
pen.change_pen_size_by(1)
```

7 设置粗细

```
pen.set_pen_size_to(1)
```

Arduino

arduino库中所有扩展方法及属性皆依赖于arduino扩展包，使用前请导入arduino

```
import arduino
```

> arduino 支持烧录固件‘mCookie’与‘mCookie扩展’。文中凡支持float类型或有效字符串float类型的参数，在实际运行中都是将其直接转换成int型。例如 2.3 等价于 2 、'2.6' 等价于 2

1、设置引脚高低电平

```
arduino.digitalWrite(port, type)
"""
    port ： 要设置引脚状态的端口，默认为2/3， 参数为2
    type ： 引脚电平状态，包含0(Low)、1(High)
    功能 ： 用于设置指定端口的电平状态
    参数类型 ： port、type皆适用于int、float及纯数字型字符串
    适用模块 ： 传感器扩展板
"""
```

2、读取数字引脚

```
arduino.digitalRead(port)
"""
    port ： 要获取引脚状态的端口，默认为2/3， 参数为2
    功能 ： 读取指定引脚当前的开关状态
    返回值类型 ： 返回值类型包含0(未工作状态)、1(工作状态)
    参数类型 ： port 适用于int、 float及纯数字型字符串
"""
```

3、判定引脚高低电平

```
arduino.digitalRead(port) == type
"""
    port ： 要获取引脚状态的端口，默认为2/3， 参数为2
```

```
    type : 引脚电平状态, 包含0(Low)、1(High)
    功能 : 读取指定引脚当前的开关状态
    返回值类型 : 返回值类型包含0(未工作状态)、1(工作状态)
    参数类型 : port 适用于int、 float及纯数字型字符串
    适用模块 : 传感器扩展板
"""
```

> 这是一个组合型的指令，由引脚电平状态读取指令与判断指令相组合而生成。使用时可以参考并修改判断指令以达到实现特殊需求的效果

4、读取模拟引脚

```
arduino.analogRead(port)
"""
    port: 模拟引脚端口, 默认为A0, 参数为0
    功能 : 读取指定模拟引脚的输入值
    返回值类型 : 返回值类型为int型
    参数类型 : port 适用于int、 float及纯数字型字符串
"""
```

5、设置引脚PWM值

```
arduino.analogWrite(port, pwm)
"""
    port : 引脚端口, 默认为3,  参数为3
    pwm : 输出pwm值
    功能 : 向指定引脚输出特定频率的pwm值
    参数类型 : port、pwm皆适用于int、 float及纯数字型字符串
"""
```

6、映射

```
arduino.map(num, snum1, snum2, enum1, enum2)
"""
    num : 目标值, 属于snum1到snum2范围内的数据
    snum1 : 映射前的起始值
    snum2 : 映射前的终止值
    enum1 : 要映射的起始值
    enum2 : 要映射的终止值
```

```
    功能 ： 将snum1至snum2范围内的数值num映射到enum1至enum2范围内
    返回值类型 ： 返回值类型为int型
    参数类型 ： num、snum1、snum2、enum1、enum2皆适用于int、 float及纯数字型字符串
"""
```

> 此函数主要起到一个对数值进行映射的作用，以便于将数值限制在一个有效的或实际需要范围内，例如，将数值1-1000范围内的500映射到1-100的范围内就是50.此处的500即为num,1-1000为snum1-snum2,1-100为enum1-enum2,50为映射后的返回值

7、约束

```
max(min(max_num, num), min_num)
"""
    num ：  要进行约束的数值
    min_num ： 约束范围最小值
    max_num ： 约束范围最大值
    功能 ： 将要约束的值，约束在指定的范围内
    返回值类型 ： 返回值类型为int
    参数类型 ： num、min_num、max_num
"""
```

> 此为一个由min()与max()组合的复合函数，主要作用是将目标值限定在规定的范围内，当目标值小于限定范围内最小值时取限定范围的最小值，反之亦然

AI

ai库中所有扩展方法及属性皆依赖于ai扩展包，使用前请导入ai

```
import ai
```

> ai支持烧录固件‘mCookie’与‘mCookie扩展’。文中凡支持float类型或有效字符串float类型的参数，在实际运行中都是将其直接转换成int型。例如 2.3 等价于 2 、'2.6' 等价于 2

1、图像识别

```
ai.cameraRead(modes,type)
"""
    modes : 开启模式，默认为手动，参数为'-1',包含'1'(1s后)、'2'(2s后)、'3'(3s后)、'-1'(手动)
    type : 选择要识别的类型，默认为表情，参数为‘emotion'，包含表情('emotion')、性别('gender')、年龄('age')、肤色('skin')、是否戴眼镜('glasses')、颜值('beauty')、车牌('carLicense')、文字ocr('ocr')
    返回值类型 : 返回值为字符串类型
    功能 : 通过摄像头，识别表情、性别、年龄、肤色、是否戴眼镜、颜值、车牌、文字ocr
    参数类型 : modes、type仅支持字符串类型
"""
```

2、识别植物

```
ai.animalAndPlantRecognition(modes, type, num)
"""
    modes : 开启模式，默认为手动，参数为'-1',包含'1'(1s后)、'2'(2s后)、'3'(3s后)、'-1'(手动)
    type : 选择要识别的类型，默认为植物，参数为'0'，包含 '0'(植物)'1'(动物)
    num : 选择要识别的对象，默认为多项，参数为'0'，包含 '0'(多项)、'1'(第一个)、'2'(第二个)、'3'(第三个)、'4'(第四个)、'5'(第五个)
    返回值类型 : 返回值为字符串类型
    功能 : 通过摄像头，识别动物各植物
    参数类型 : modes、type、num仅支持字符串类型
"""
```

3、语音输入

```
ai.voiceInput(language, second)
"""
    language : 选择语言，默认为中文，参数为'cn'，包含'cn'(中文)、'en'(英文)
    second :  语言输入持续时间，默认为2s,参数为'2',包含'2'(2s)、'5'(5s)、'10'(10s)、'60'(60s)
```

```
    功能 ：  用于语音输入
    参数类型 ： language、second 仅支持字符串类型
"""
```

4、朗读

```
ai.voiceOutput(gender,str(input_str))
"""
    gender ： 朗读声类型选择，默认为女声，参数为2，包含 1(男声)、2(女声)
    input_str ： 要进行朗读的字符串，默认为空
    功能 ： 将输入的字符串，转换为语言进行朗读
    参数类型 ： input_str 仅支持字符串类型，2 支持int、float及纯数字型字符串
"""
```

5、翻译

```
ai.translate(input_str,type)
"""
    input_str ： 要进行翻译的字符串，默认为空
    type ： 要翻译的类型，默认为英文，参数为’en',包含'zh'(中文)、'en'(英文)
    功能 ： 将输入的语言字符串，翻译为指定的语言
    返回值类型 ： 返回值类型为字符串
    参数类型 ： input_str、type仅支持字符串类型
"""
```

6、对话

```
ai.chat(input_str)
"""
    input_str ： 输入内容，默认为空可以输入英文或中文字符串
    功能 ： 输入对敌字符串，返回对话内容
    返回值类型 ： 返回值类型为字符串
    参数类型 ： input_str仅支持字符串类型
"""
```

IBB

buggy库中所有扩展方法及属性皆依赖于buggy扩展包，使用前请导入buggy

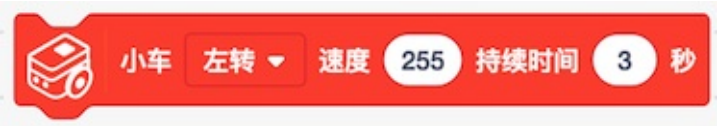
```
import buggy
```

> buggy支持烧录固件为‘buggy’。文中凡支持float类型或有效字符串float类型的参数，在实际运行中都是将其直接转换成int型。例如 2.3 等价于 2 、'2.6' 等价于 2。

1 小车前进/后退

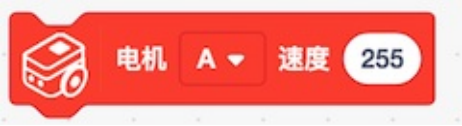
```
buggy.bot_car_forward(direction, speed)
time.sleep(delay)
buggy.bot_car_forward(direction, 0)
"""
    direction : 控制小车前进和后退，前进为1，后退为0
    speed : 默认转速为255r/min
    delay : 小车运动的时间，默认为3s
    参数类型 :  port，speed，delay支持int、float型和纯数字型字符串型
"""
```

2 小车左转/右转




```
buggy.bot_car_turn(direction, speed)
time.sleep(delay)
buggy.bot_car_forward(direction, 0)
"""
    direction : 控制小车左转和右转，左转为0，右转为1
    speed : 默认转速为255r/min
    delay : 小车运动的时间，默认为3s
    参数类型 :  port，speed，delay支持int、float型和纯数字型字符串型
"""
```

3 电机速度




```
buggy.bot_motor_control(ports, speed)
"""
    ports : 电机端口(包含3个参数，默认为1)
    speed : 默认转速为255r/min
    参数类型 :  port，speed支持int、float型和纯数字型字符串型
"""
```

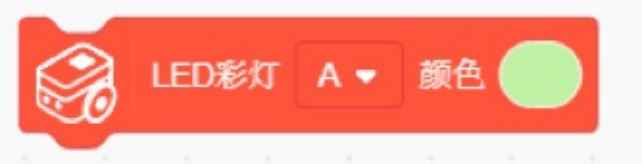
> A电机(1)、B电机(2)、AB电机(3)，AB代表同时控制A电机和B电机

4 电机停止

```
buggy.bot_motor_stop(ports)
"""
    ports : 电机端口(包含3个参数，默认为1)
    参数类型 :  port支持int、float型和纯数字型字符串
```

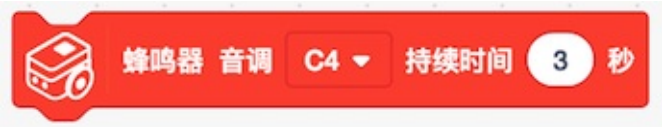
> A电机(1)、B电机(2)、AB电机(3)，AB代表同时控制A电机和B电机

5 LED彩灯




```
buggy.botLEDColor(ports, color)
"""
    ports : 彩灯端口(包含2个参数，A灯为1，B灯为2)
    color : 彩灯色彩值，默认为'#62f1e5'(色彩十六进制字符串)
    参数类型 :  port支持int、float型和纯数字型字符串，color为16进制色彩值字符串类型
"""
```

6 蜂鸣器播放音调




```
buggy.bot_buzzer_pitch(beat)
time.sleep(delay)
buggy.bot_buzzer_pitch(0)
"""
    beat : 音符节拍包含(C4、D4、E4、F4、G4、A4、B4、C5、D5、E5、F5、G5、A5、B5、C6、D6、E6、F6、G6、A6、B6)默认C4，参数为262
    delay : 蜂鸣器持续时间默认3s
    参数类型 :  beat，delay支持int、float型和纯数字型字符串
"""
```

> 音调对应的频率值为：C4(262)、D4(294)、E4(330)、F4(349)、G4(392)、A4(440)、B4(494)、C5(523)、D5(587)、E5(659)、F5(698)、G5(784)、A5(880)、B5(988)、C6(1047)、D6(1175)、E6(1319)、F6(1397)、G6(1568)、A6(1760)、B6(1976);

7 蜂鸣器播放歌曲

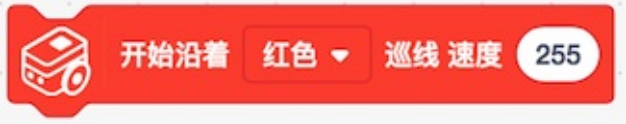
```
buggy.bot_buzzer_song(type)
"""
    type : 音乐类型, 默认为1
    参数类型 : type 支持int、float及纯数据型字符串
"""
```

> 音乐类型主要有: 1(丢手娟)、2(两只老虎)、3(多啦A梦)、4(小苹果)

8 蜂鸣器停止

```
buggy.bot_buzzer_stop()
```

9 小车开始巡线



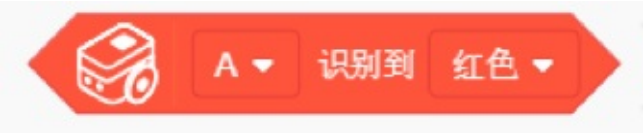

```
buggy.bot_find_line(color, speed)
"""
    color : 颜色类型, 默认为1
    speed : 默认转速为255r/min
    参数类型 : color, speed 支持int、float及纯数据型字符串
"""
```

> 颜色类型主要有: 红色(1)、绿色(2)、蓝色(3)、黑色(0)

10 小车停止巡线

```
buggy.bot_find_stop()
```

11 颜色识别状态




```
buggy.bot_judge_color(ports, color)
"""
```

```
    ports : 灰度传感器端口(包含2个参数, A为1, B为2)
    color : 颜色类型, 默认为1
    返回值类型 : 返回类型为bool类型, 状态为真时返回True否则False
    参数类型 : ports, color 支持int、float及纯数据型字符串
"""
```

> 颜色类型主要有: 红色(1)、绿色(2)、蓝色(3)、黄色(4)、紫色(5)、青色(6)、白色(7)、黑色(8)

12 获取灰度

```
buggy.bot_get_gray(ports, color)
"""
    ports : 灰色传感器端口(包含2个参数, A为1, B为2)
    color : 设置LED彩灯颜色, 默认为1
    返回值类型 : 返回值类型为float
    参数类型 : ports, color 支持int、float及纯数据型字符串
"""
```

> LED彩灯颜色主要有: 红色(1)、绿色(2)、蓝色(3)、关(0)

mCookie

mCookie库中所有扩展方法及属性皆依赖于mcookie扩展包，使用前请导入mcookie

```
import mcookie
```

> 除OLED模块外其他皆适用于'mCookie扩展'固件。文中凡支持float类型或有效字符串float类型的参数，在实际运行中都是将其直接转换成int型。例如 2.3 等价于2 、'2.6' 等价于 2

动作

1、电机(Motor)速度

```
mcookie.motorContro(ports, speed)
"""
    ports : 电机端口(包含两个端口, 1A1B与2A2B)默认1A1B, 参数为0和1
    speed : 电机转速(默认30r/min)
    功能 : 控制电机以指定速度运转
    参数类型: ports与speed皆支持int、float型和纯数字型字符串)
    适用模块 : 传感器扩展板
"""
```

> ports 为0 代表1A1B，1 代表2A2B;ports 为float类型或字符串float时会自动取整为int型

2、电机(Motor)停止

```
mcookie.motorBrake(ports)
"""
    ports : 电机端口(包含两个端口, 1A1B与2A2B), 参数为0和1
    功能 : 停止指定端口的电机
    参数类型: ports支持int、float型和纯数字型字符串
    适用模块 : 传感器扩展板
"""
```

> ports 为0 代表1A1B，1 代表2A2B;ports 为float类型或字符串float时会自动取整为int型

3、电机速度

```
mcookie.motorplusControl(ports, speed)
"""
    ports : 电机端口(包含两个端口, 1A1B与2A2B), 参数为0和2
    speed : 电机转速(默认255r/min)
    功能 : 控制电机以指定速度运转
    参数类型: ports与speed皆支持int、float型和纯数字型字符串
    适用模块 : 传感器扩展板
"""
```

> ports 为0 代表1A1B，2 代表2A2B;ports 为float类型或字符串float时会自动取整为int型

4、电机停止

```
mcookie.motorBrake(ports)
"""
    ports : 电机端口(包含两个端口, 1A1B与2A2B), 参数为0和2
    功能 : 停止指定端口的电机
    参数类型: ports支持int、float型和纯数字型字符串
    适用模块 : 传感器扩展板
"""
```

> ports 为0 代表1A1B，2 代表2A2B;ports 为float类型或字符串float时会自动取整为int型

左为电机(Motor)模块，右为电机模块

5、小车(Motor)控制

```
mcookie.carControl(port1, port2)
"""
    port1 ： 电机端口1A1B，参数默认为-150代表逆时针转速为150r/min
    port2 ： 电机端口2A2B，参数默认为150代表顺时针转速为150r/min
    功能 ： 控制两个端口的电机以相反方向进行旋转
    参数类型 ： port1与port2皆支持int、float型和纯数据型字符串
    适用模块 ： Motor模块
"""
```

> port1和port2 为float类型或字符串float时会自动取整为int型

6、小车控制

```
mcookie.carPlusControl(150, 150)

"""
    port1 ： 电机端口1A1B，参数默认为150代表逆时针转速为150r/min
    port2 ： 电机端口2A2B，参数默认为150代表顺时针转速为150r/min
    功能 ： 控制两个端口的电机以相反方向进行旋转
    参数类型 ： port1与port2皆支持int、float型和纯数字型字符串
    适用模块 ： MotorPlus模块
"""
```

> 小车控制中Motor模块与Motor+模块参数存在误解，Motor模块语句中默认‘-’号代表反转，Motor+模块语句中默认不加‘-’号2A2B端口也存在反转，电机转速较低时电机不能正常启动，此时须提高转速;port1和port2 为float类型或字符串float时会自动取整为int型

7、舵机旋转角度

```
arduino.servoWrite(port, angle)
"""
    port ： 舵机引脚端口，默认为2， 参数为2
    angle ： 舵机旋转角度
    功能 ： 将指定引脚下的舵机旋转到需要的角度
    参数类型 ： port、angle皆适用于int、 float及纯数字型字符串
    适用模块 ： 传感器扩展板
"""
```

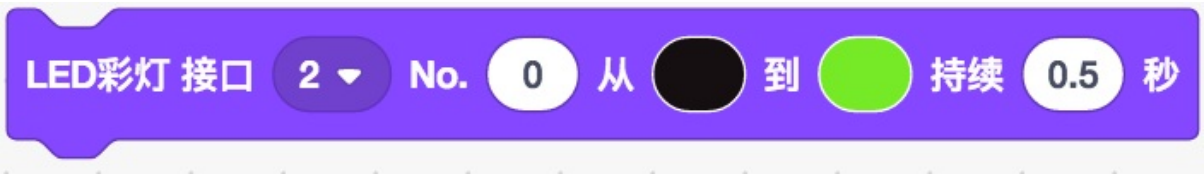
8、控制舵机速度

```
mcookie.servoControl(port, value1, value2, delay)
"""
    port : 舵机引脚端口，默认为2/3端口，参数为2
    value1 : 舵机启始转动值默认为30
    value2 : 舵机终止转动值默认为120
    delay :  舵机持续转动时间默认为500ms
    功能 : 控制舵机在规定时间内由转动值value1到value2
    参数类型 : port、speed1、speed2皆支持int、float型和纯数字型字符串
    适用模块 : 传感器扩展板
"""
```

> port为float类型或字符串float时会自动取整为int型

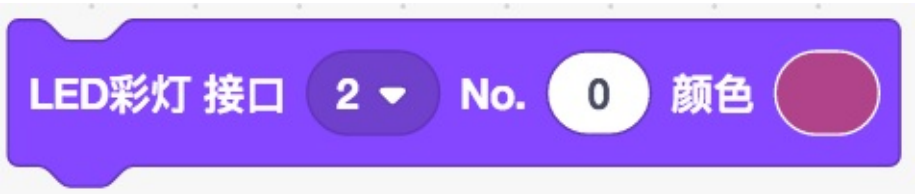
显示

9、LED彩灯渐变




```
mcookie.colorLEDControl(port, num, color1, color2, delay)
"""
    port : 彩灯引脚端口，默认为2/3端口，参数为2
    num : 彩灯索引值，彩灯允许串联使用，从连接至扩展口的第一个开始，索引值依次为0、1、2... 默认值为0，即第一个彩灯
    color1 : 彩灯起始色彩值，默认为'#b20862'(色彩十六进制字符串)
    color2 : 彩灯结束色彩值，默认为'#136e17'(色彩十六进制字符串)
    delay : 彩灯从起始颜色到结束颜色过渡时间，默认为500ms
    功能 : 控制彩灯在初始颜色与结束颜色之间在规定时间内进行渐变
    参数类型 :  port、num、delay皆支持int、float型和纯数字型字符串，color1、color2为16进制色彩值字符串类型
    适用模块 : 传感器扩展板
"""
```

10、LED彩灯单色

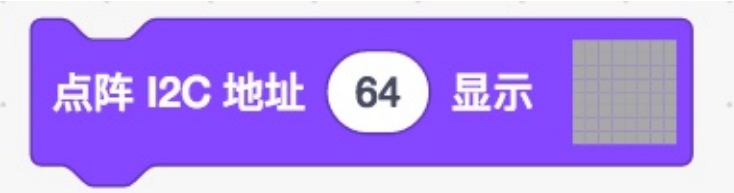



```
mcookie.colorLEDColor(port, num, color)
"""
    port : 彩灯引脚端口，默认为2/3端口，参数为2
    num : 彩灯索引值，彩灯允许串联使用，从连接至扩展口的第一个开始，索引值依次为0、1、2... 默认值为0，即第一个彩灯
    color : 彩灯色彩值，默认为'#62f1e5'(色彩十六进制字符串)
    功能 : 控制彩灯在初始颜色与结束颜色之间在规定时间内进行渐变
    参数类型 :  port、num皆支持int、float型和纯数字型字符串，color为16进制色彩值字符串类型
```

```
    适用模块 : 传感器扩展板
"""
```

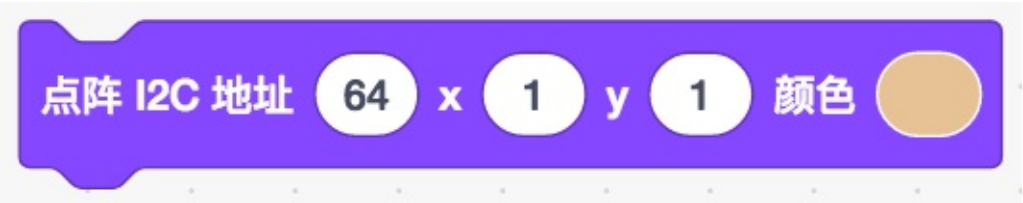
> 单色LED彩灯只有一个控制接口，所以只能接收一道命令;全彩LED灯有一个控制接口（IN），和一个传递接口（OUT)所以可以接收多道命令，并将命令传递给下一个全彩LED灯,使用时可以根据需要选择单色或者全彩LED

11、点阵屏图案




```
mcookie.dotMatrixColor(address,color)
"""
    address : 地址范围，默认为64
    color : 十六进制色彩值字符串,所有点的色彩值拼接为一个字符串，默认值全为#000000
    端口 : IIC
    适用模块 : 传感器扩展板
    参数类型 : address支持int、float及纯数据型字符串，color为字符型
"""
```

12、点阵屏坐标填色




```
mcookie.dotMatrixPoint(64, 1, 1, '#6d26ea')
"""
    address : 地址范围，默认为64
    x : x坐标值,默认值为1
    y : y坐标值,默认值为1
    color : 十六进制色彩值字符串,默认值为#6d26ea'
    端口 : IIC
    适用模块 : 传感器扩展板
    参数类型 : address、x 、y支持int、float及纯数据型字符串，color为字符型
"""
```

> 点阵的地址说明： 1、地址范围：1~64,超过64为64 2、地址表示方法：点阵上电后显示的绿点数量表示点阵地址 3、地址设置方法：点阵上电后的3秒内，按下背面按键可进入地址配置状态，此时点阵颜色变为白色 --慢按按键（2次按键间隔大于0.5s）会增加地址 --快按按键（2次按键间隔小于0.5s）会减小地址 点阵地址设定完毕后，长按设置键3秒，即可退出地址配置状态

13、RGB色值

```
combineRGB(R, G, B)
"""
    R : 三基色(红),  默认为255
    G : 三基色(绿),  默认为0
    B : 三基色(蓝),  默认为0
    功能 : 使用R、G、B组合出所需要的颜色
    参数类型 : R、G、B皆支持int、float型和纯数据字型字符串
"""
```

14、OLED清屏

```
mcookie.oledClear()
"""
    功能 : 用于清空OLED显示屏的当前显示内容
    适用模块 : OLED扩展模块
    固件选择 : 适用于烧录mCookie固件
    适用模块 : 传感器扩展板
"""
```

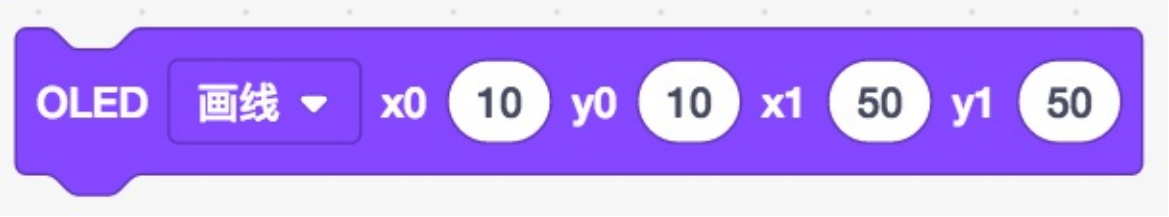
> 其他指令可以直接烧录‘mCookie扩展’固件即可使用，但是OLED模块需要烧录‘mCookie‘固件才可以正常使用，否则会出现OLED不能显示的故障

15、OLED信息

```
mcookie.oledMessage(x, y, msg)
"""
    x : x坐标值，默认值为10
    y : y坐标值，默认值为10
    msg : OLED展示屏要显示的信息
    功能 : 在OLED显示屏上显示指定信息
    固件选择 : 适用于烧录mCookie固件
    适用模块 : 传感器扩展板
"""
```

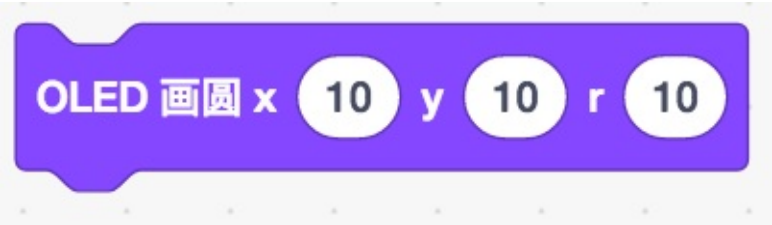
> OLED显示屏并不支持python语法中的转义符、特殊字符及汉字，例如转义符:'\n'、'\r'、'\t'...特殊字符:¥∆˚¬πœ∑†¥åßƒ˙∆˚©...否则会出现乱码或混合字符过长(特殊字符加汉字或单独特殊字符总长超过15个字符)会出现持续闪烁或偶发性死机的情况，偶尔出现死机时须断开并重新连接即可恢复正常

16、OLED画线 -画矩形



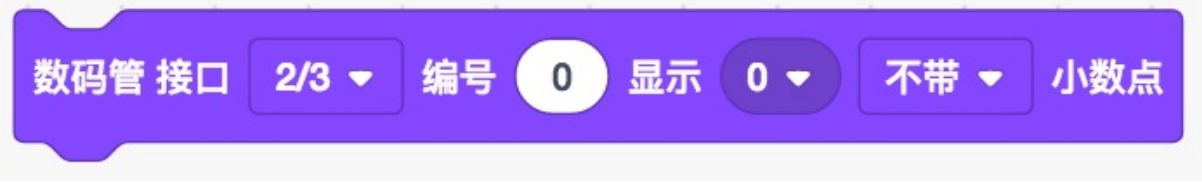

```
mcookie.oledDraw(type, x0, y0, x1, y1)
"""
    type : 绘图类型，默认为0，包含0(画线)、1(画矩形)
    x0 : 线条或矩形(矩形起始点为对角中的点)起始点x坐标y
    y0 : 线条或矩形(矩形起始点为对角中的点)起始点y坐标
    x1 : 线条或矩形(矩形终点点为对角中的点)起终点x坐标
    y1 : 线条或矩形(矩形终点点为对角中的点)起终点y坐标
    功能 : 绘制直线、斜线、矩形
    适用模块 : 传感器扩展板
    固件选择 : 适用于烧录mCookie固件
"""
```

17、OLED画圆




```
mcookie.oledCircle(x, y, r)
"""
    x : x坐标值，默认值为10
    y : y坐标值，默认值为10
    r : 圆的半径
    功能 : 绘制以指定坐标点为圆心r为半径的圆
    适用模块 : 传感器扩展板
    固件选择 : 适用于烧录mCookie固件
"""
```

18、数码管引脚




```
mcookie.setNixietube(port, num1, num2, point)
"""
    port : 数码管引脚端口，默认端口为2/3，参数为2
    num1 : 数码管索引值，允许数码管串联使用，从连接至扩展口的第一个开始，索引值依次为0、1、2... 默认值为0，即第一个数码管
    num2 : 选定数码管要显示的数字内容，须要关闭数码管时调用mcookie.sendNixieTube(port, num1, num2, point)，关闭时num1+=1
    point : 小数点，包含三个状态0(不带)、1(带)、3(用于关闭状态)
    功能 : 指定数码管并显示指定数字
    参数类型 : port、num1、num2、point皆支持int、float及纯数字型字符串
    适用模块 : 传感器扩展板
"""
```

播放

19、蜂鸣器音调

```
mcookie.buzzerNote(port, beat, delay1)
time.sleep(delay2)
"""
    port : 蜂鸣器引脚端口，默认为2/3端口，参数为2
    beat : 音符音调包含(C4、D4、E4、F4、G4、A4、B4、C5、D5、E5、F5、G5、A5、B5、C6、D6、E6、F6、G6、A6、B6)默认C4，参数为2
62
    delay1 : 蜂鸣器持续时间默认250ms
    delay2 : 蜂鸣器持续时间默认250ms
    适用模块 : 传感器扩展板
    参数类型 : port、beat、delay1、delay2 皆支持int、float及纯数据型字符串
"""
```

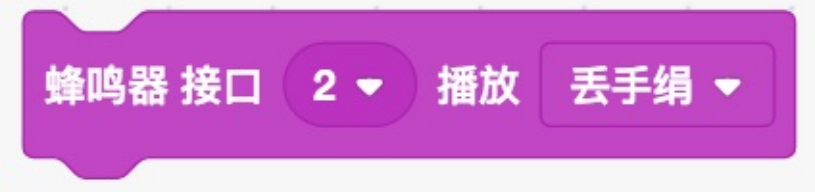
> 音调对应的频率值为：C4(262)、D4(294)、E4(330)、F4(349)、G4(392)、A4(440)、B4(494)、C5(523)、D5(587)、E5(659)、F5(698)、G5(784)、A5(880)、B5(988)、C6(1047)、D6(1175)、E6(1319)、F6(1397)、G6(1568)、A6(1760)、B6(1976); delay1为无效值，不能作为正常延时使用，延时操作需导入import time -> time.sleep(delay2） delay2为延时时间，单位ms

20、蜂鸣器频率

```
mcookie.buzzerControl(port, frequency, delay1)
time.sleep(delay2)
"""
    port : 蜂鸣器引脚端口，默认为2/3端口，参数为2
    frequency : 蜂鸣器振动频率
    delay1 : 鸣器持续时间默认250ms
    delay2 : 鸣器持续时间默认250ms
    适用模块 : 传感器扩展板
    参数类型 : port、beet、delay1、delay2 皆支持int、float及纯数据型字符串
"""
```

> delay1为无效值，不能作为正常延时使用，延时操作需导入import time -> time.sleep(delay2） delay2为延时时间，单位ms

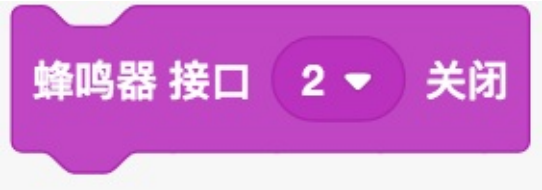
21、蜂鸣器播放音乐




```
mcookie.buzzerSong(port, type)
"""
    port : 蜂鸣器引脚端口，默认为2/3端口，参数为2
    type : 音乐类型，默认为1
    适用模块 : 传感器扩展板
    参数类型 : port、type 皆支持int、float及纯数据型字符串
"""
```

> 音乐类型主要有: 1(丢手娟)、2(两只老虎)、3(多啦A梦)、4(小苹果)

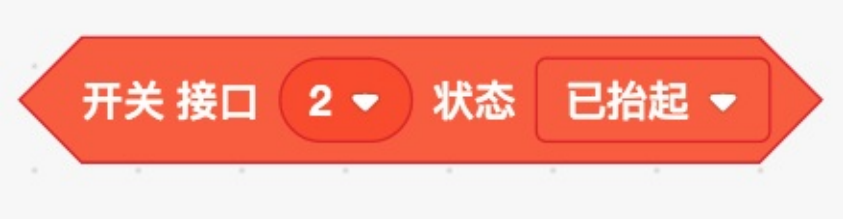
22、蜂鸣器关闭




```
mcookie.buzzerOff(port)
"""
    port : 蜂鸣器引脚端口，默认为2/3端口，参数为2
    功能 : 关闭指定端口的音乐播放
    适用模块 : 传感器扩展板
    参数类型 : port 支持int、float及纯数据型字符串
"""
```

感知

23、开关状态

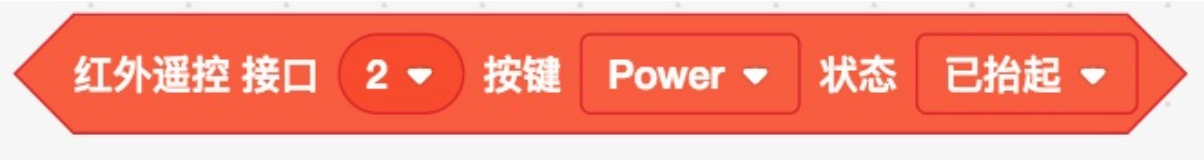



```
mcookie.touchKey(port, status)
"""
    port : 监听指定端口的开关状态，默认为2/3端口，参数为1
    status : 开关状态，主要有1(已抬起)、0(已按下)、2(长按)
    返回值类型 : 返回类型为bool类型，状态为真时返回True否则False
    参数类型 : port、status支持int、float及纯数据型字符串
    适用模块 : 传感器扩展板
"""
```

24、摇杆按键状态

```
mcookie.joystickKey(port, button, status)
"""
    port : 监听指定端口的摇杆按钮状态，默认为A0，参数为0，支持A0-A7端口，参数为0-7
    button : 摇杆的指定按钮默认为'上',参数为1，主要有'上'(1)、'下'(2)、'左'(3)、'右'(4)、'中央'(0)、
    status : 开关状态，主要有1(已抬起)、0(已按下)、2(长按)
    返回值类型 : 返回类型为bool类型，状态为真时返回True否则False
    参数类型 : port、button、status、支持int、float及纯数据型字符串
    适用模块 : 传感器扩展板
"""
```

25、红外摇控状态




```
mcookie.irKey(port, button, status)
"""
    port : 红外摇控引脚端口，默认为2/3端口，参数为2
    button : 摇控器状态指令，默认为Power，参数为 0x1FE48B7
    status : 开关状态，主要有1(已抬起)、0(已按下)、2(长按)
    返回值类型 : 返回类型为bool类型，状态为真时返回True否则False
    参数类型 : port、status、支持int、float及纯数据型字符串,button为标准十六进制
    适用模块 : 传感器扩展板
"""
```

> 摇控器的指令主要包括: Power(0x1FE48B7)、A(0x1FE807F)、B(0x1FE40BF)、C(0x1FEC03F)、D(0x1FE20DF)、E(0x1FE609F)、OK(0x1FE10EF)、上(0x1FEA05F)、下(0x1FED827)、左(0x1FEE01F)、右(0x1FE906F)、后退(0x1FE50AF)、播放/暂停(0x1FE30CF)、音量+(0x1FEF807)、音量-(0x1FE708F)、静音(0x1FEB04F)

26、姿态传感器-角度/偏航

```
mcookie.getMotionSTH2X('motion',angle, status, 106, 28)
"""
    angle : 角度选择，默认为角度，参数为‘0’，包含'0'(角度)、'2' (角速度)
    status : 状态选择，默认为偏航，参数为‘Z'，包含'Z'(偏航)、'Y'(俯仰)、'X'(翻滚)
    返回值类型 : 返回值类型为float
    参数类型 : angle，status皆为字符串类型
"""
```

> 参数'motion'、106，28为固定值，不可手动修改，其中angle为'0'时，最后一位参数为28，angle为'2'时，最后一位参数为16

27、姿态传感器-加速度

```
mcookie.getMotionSTH2X('motion','1',status, 106, 4)
"""
    status : 状态选择, 默认为'前后', 参数为'X', 包含'X'(前后)、'Y'(左右)、'Z'(上下)
    返回值类型 : 返回值类型为float
    参数类型 : status皆为字符串类型
"""
```

28、 温/湿度传感器

```
mcookie.temperatureSTH2X()
mcookie.humiditySTH2X()
"""
    端口 : IIC
    选择 : 检测温度接口(temperatureSTH2X())、检测湿度接口(humiditySTH2X())
    功能 : 用于检测温度和湿度
    返回值 : 返回值类型为float类型
    适用模块 : 传感器扩展板
"""
```

29、温度传感器

```
mcookie.temperatureLM75()
"""
    端口 : IIC
    功能 : 用于检测温度
    返回值 : 返回值类型为float类型
    适用模块 : 传感器扩展板
"""
```

30、手势距离

```
mcookie.gestureDistance()
"""
    端口 : IIC
```

```
    功能 ： 用于测量手势距离
    返回值 ： 返回值类型为int类型，表示实际距离
    适用模块 ： 传感器扩展板
"""
```

31、超声波测距

```
mcookie.ultrasonic()
"""
    端口 ： IIC
    功能 ： 用于测量距离
    返回值 ： 返回值类型为int类型,表示实际距离
    适用模块 ： 传感器扩展板
"""
```

ideaBox

ideaBox库中所有扩展方法及属性皆依赖于ideaBox扩展包，使用前请导入ideaBox

```
import ideaBox
```

> ideaBox支持烧录固件为‘ideaBox’(其中数码管显示及姿态需导入mCookie,同时需要专用的ideaBox模块来支持)。文中凡支持float类型或有效字符串float类型的参数，在实际运行中都是将其直接转换成int型。例如 2.3 等价于 2 、'2.6' 等价于 2。

1 数码管显示

```
mcookie.setNixietube(9, 0, num)
"""
    num : 要数码管上显示的内容，默认为5，参数为5，包含0-9
    功能 :  在数码管上显示指定的数字内容
    参数类型 : num 支持int、float及纯数字型字符串
    适用模块 : ideaBox
"""
```

> 9，0两个参数为固定参数，不可修改

2 姿态

```
mcookie.getMotion(8, 1, status)
"""
    status : 姿态状态，默认为翻滚，参数为'X',包含'X'(翻滚)、'Y'(俯仰)、'Z'(偏航)
    功能 : 用于判断当前姿态状态
    返回值类型 : 返回值类型为int
    参数类型 : status 仅支持字符串类型
"""
```

> 8, 1两个参数为固定参数，不可修改

3 音频播放

```
ideabox.audioPlay(song)
"""
    song : 要播放的音乐编号，默认为1，包含1-10
    功能 : 播放选择音乐曲目
    参数类型 : song 支持int、float及纯数字型字符串
    适用模块 : ideaBox
    硬件支持 : 需要ideaBox插入sdk卡，供音乐播放选择
"""
```

4 音频停止

```
ideabox.audioStop()
"""
    功能 : 停止音乐的播放
    适用模块 : ideaBox
"""
```

5 音频音量

```
ideabox.audioVolume(volume)
"""
    volume : 音频音量，默认为20%，可根据需要进行不同音量的输入
    功能 : 控制音频音量大小
    参数类型 : volume支持int 、float及纯数字型字符串
    适用模块 : ideaBox
"""
```

6 舵机旋转

```
ideabox.servoWrite(angle)
"""
    angle : 舵机旋转角度
    功能 : 控制旋转指定的角度
    参数类型 : angle 支持int、float及纯数字型字符串
    适用模块 : ideaBox
"""
```

姿态检测

PoseNet就是姿态估计,通过对图片视频进行一系列处理得到人的关键点,最基本的就是关节点。

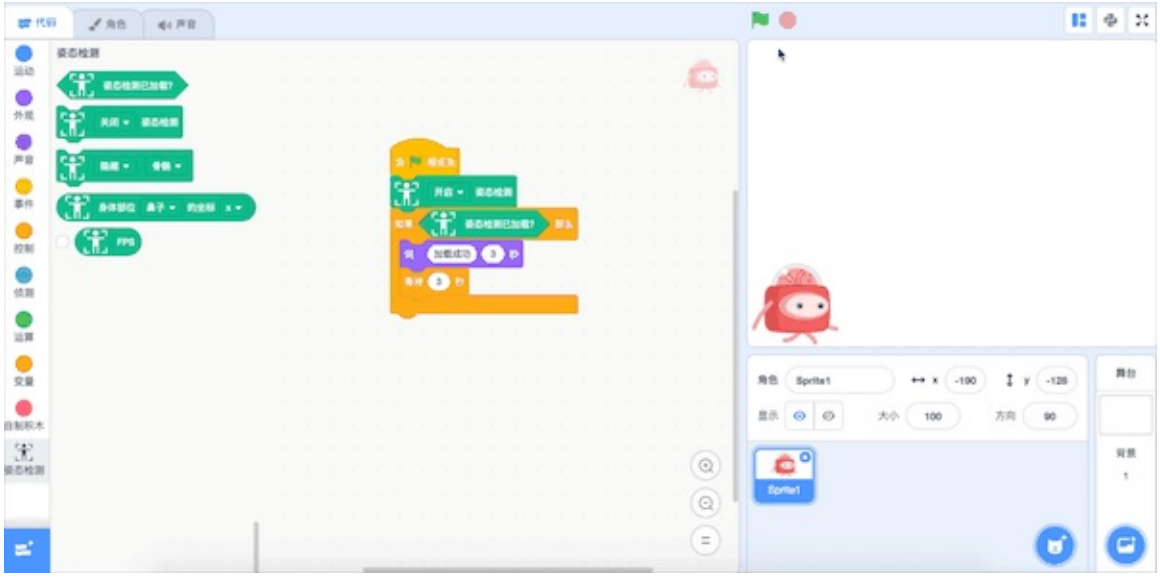
该扩展是基于TensorFlow.js 实时运行人体姿态检测。

1、加载模型

通过该程序块「打开」或「关闭」PoseNet模型，打开模型，需要打开摄像头；

2、检测模型是否加载成功

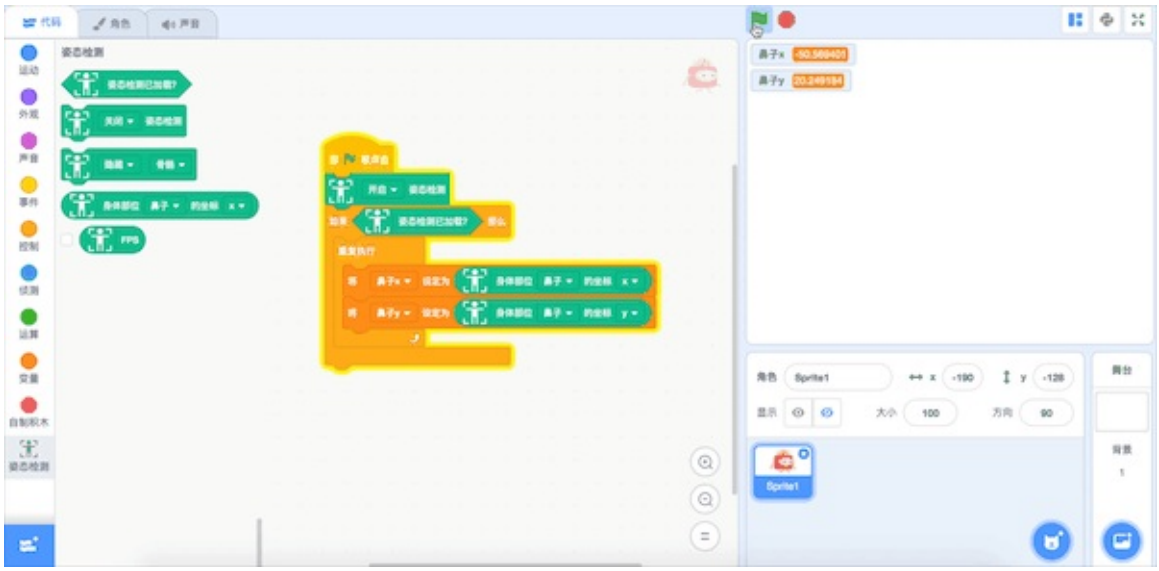
模型加载成功后，可以看到人体的关键点和骨骼；



3、人体关键点的坐标

该程序块会返回人体关键点在舞台中的x、y坐标值；

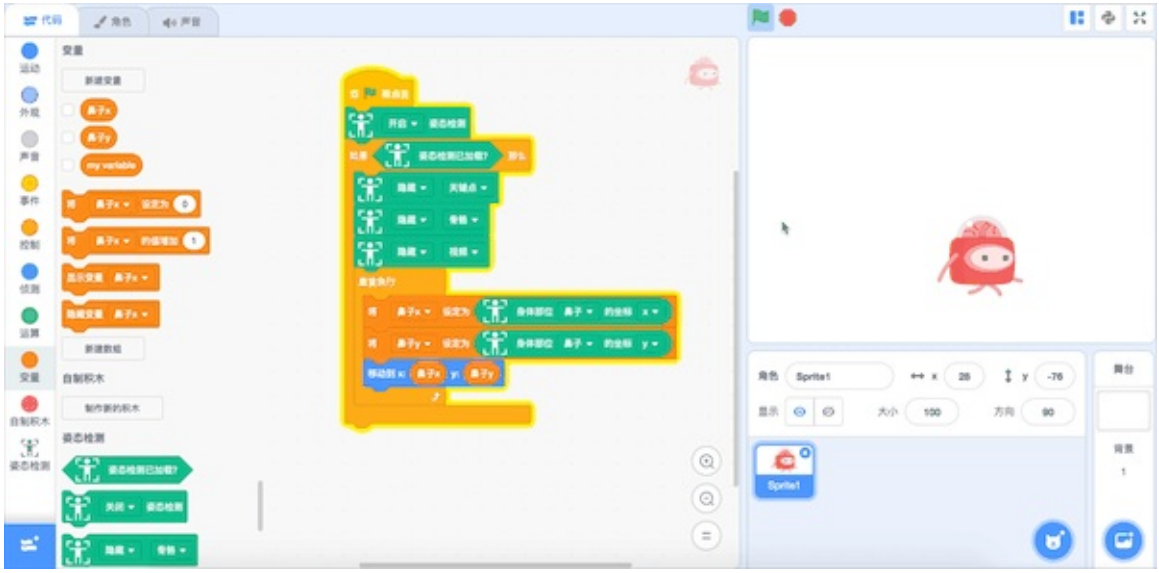
示例：实时返回鼻子的x、y坐标



4、隐藏关键点/骨骼/视频画面

注意：关闭视频，只是关闭视频画面，并没有关闭摄像头输入

示例：精灵跟随鼻子的坐标运动



连接mCenter硬件

硬件连接

堆叠模块：将模块简单堆叠在mCenter+上或者其他磁吸模块上。叠堆时确保插针正确对齐。

将模块连接到集线器：用集线器连接线将传感器、元件和其他模块链接到集线器上。将线的一段连接到模块，另一端连接到集线器上的端口。

故障排除

1. **电源问题**： 打开开关后电源指示灯不亮

***a.*电池电量过低**： 诊断及解决方案：关闭开关，连接电源，观察电源指示灯处于充电状态（红色呼吸灯），待充电完成（红灯亮）后再进行操作。

***b.*指示灯损坏**：诊断及解决方案：连接电脑，打开开关，如果可以正常识别接口即可初步判断为电源指示灯损坏，请联系经销商进行维修或更换。

2. **连接问题**

usb串口连接问题：遭遇连接问题时（包括但不限于：无法连接电脑、无法选取串口、传感器无法正常工作等）请首先尝试关闭硬件及相应软件，拔掉usb线，等待5秒后再次打开。

3. **连接电脑后无法识别端口：**

***a.*驱动工作异常**： 诊断及解决方案：依照提示卸载驱动，之后再进行驱动安装。

***b.USB*数据线工作异常**： 诊断及解决方案：使用另一条USB数据线，如果可以正常连接，即可判断是USB数据线工作异常。

4. **无法选取端口**

***a.USB*芯片工作异常**： 诊断及解决方案：关机后等待10秒，再次开机尝试连接。

***b.Core*模块吸附不牢**： 诊断及解决方案：关机，将顶盖打开，取下中央的红色Core模块，再次吸附上，再次开机尝试连接。

5. **传感器无法正常工作**： 传感器线损坏

诊断及解决方案：使用辅材中提供的备用传感器线进行替换，替换后工作正常。

mCookie模块使用教程

01 霓虹灯 全彩LED灯

02 多彩景观灯 触摸开关

03 音乐画 蜂鸣器

04 人体感应壁炉 温湿度传感器

05 能量护盾 灰度传感器

06 小夜灯 光敏传感器

07 强风指示器 声音强度传感器

08 肺活量大比拼 灯带

09 转速指示器 OLED

10 线控小车 motor 电机

11 摇摆机器人 舵机

12 表情大乱斗 彩色点阵

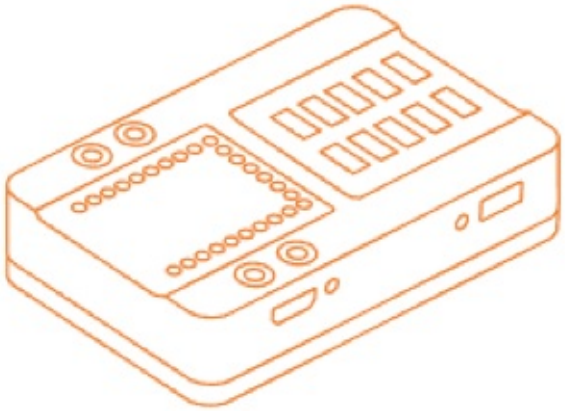
13 追踪神秘动物 手势传感器

14 无线灯光控制器 Wifi

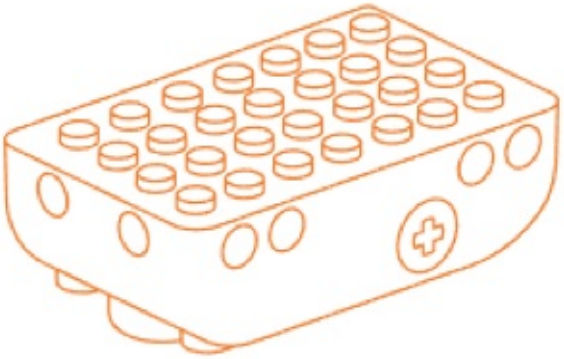
连接百变小强

认识小强

mCenter+： 核心处理，小强的大脑，可以处理和执行各种不同的指令。



动力底座：动力模块，为小强提供强劲的动力。底部还有一组神奇的传感器，可以识别各种颜色。



传感器线：连接mcenter+和动力底座，是小强的沟通神经。请将传感器线如图所示连接mcenter+和动力底座。

故障排除

1. **电源问题** ：打开开关后电源指示灯不亮

   a.**电池电量过低**： 诊断及解决方案：关闭开关，连接电源，观察电源指示灯处于充电状态（红色呼吸灯），待充电完成（红灯亮）后再进行操作。

   b.**指示灯损坏**：诊断及解决方案：连接电脑，打开开关，如果可以正常识别接口即可初步判断为电源指示灯损坏，请联系经销商进行维修或更换。

2. **连接问题**

   usb串口连接问题：遭遇连接问题时（包括但不限于：无法连接电脑、无法选取串口、传感器无法正常工作等）请首先尝试关闭硬件及相应软件，拔掉usb线，等待5秒后再次打开。

3. **连接电脑后无法识别端口：**

   a.**驱动工作异常**： 诊断及解决方案：依照提示卸载驱动，之后再进行驱动安装。

   b.USB**数据线工作异常**： 诊断及解决方案：使用另一条USB数据线，如果可以正常连接，即可判断是USB数据线工作异常。

4. **无法选取端口**

   a.USB**芯片工作异常**： 诊断及解决方案：关机后等待10秒，再次开机尝试连接。

1. **传感器无法正常工作**： 传感器线损坏

   诊断及解决方案：使用辅材中提供的备用传感器线进行替换，替换后工作正常。

小强编程

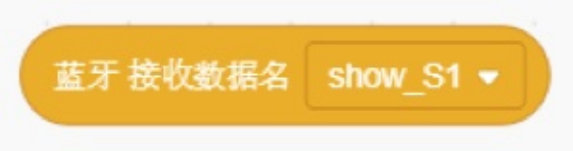
蓝牙遥控编程

连接好小强后，选择「上传模式」，选择「IoT」程序块

1、蓝牙与mDock通信

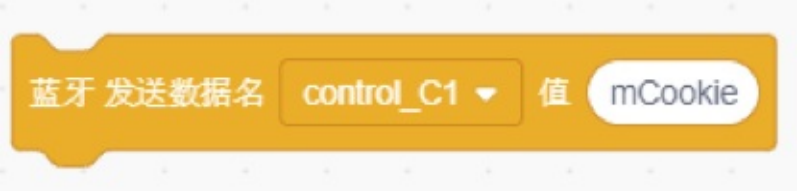
在程序块中可以设置蓝牙通信的接口和波特率，百变小强的蓝牙接口是0/1，波特率57600，与默认设置相同。传输数据并执行的程序都需要放在这个程序块中间。

2、蓝牙接收



从mDock APP蓝牙传回的数据中识别出数据名（键）为show_S1的数据的值。mDesigner编程支持接收四组键值对，分别是show_S1/show_S2/show_S3/show_S4，值的格式都是字符串格式。

3、蓝牙发送



向mDock APP发送数据名（键）为control_C1的数据的值。mDesigner编程支持发送四组键值对，分别是control_C1/control_C2/control_C3/control_C4，值的格式都是字符串格式。

典型案例——控制小强前进/后退:

小强控制编程

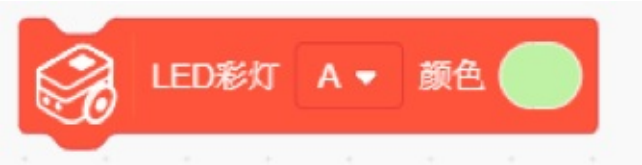
1、小车运动

设置小车运动，在持续时间内程序将停留在这个程序块不向后运行，无法执行其他命令。

2、电机控制

单独设置百变小强每一个电机的转速，设置后程序会继续向后执行，而转速会一直保持直到程序使它改变转速。

3、彩灯控制

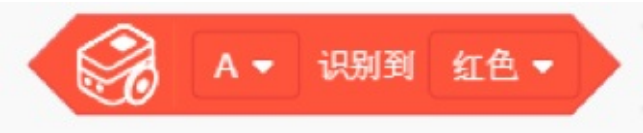


设置百变小强底部彩灯的颜色

4、蜂鸣器控制

让小强小车的蜂鸣器播放音调或者旋律，播放过程中程序将停留在这个程序块不向后运行，无法执行其他命令，可以用蜂鸣器停止的命令终止播放。

5、颜色识别



当颜色传感器A或B，识别到设定的颜色时成立。

6、小车自动巡线

让小强沿着某颜色巡线运动，设置后程序会继续向后执行，电机转速会一直保持直到程序使它停止巡线。

注意：使用巡线程序块，不能同时使用灰度、颜色识别、LED彩灯等程序块

7、获取灰度

可以通过A、B两边传感器来获取灰度值，同时获取灰度时可以设置灯的颜色。

灰度识别原理:LED灯发出不同颜色的光经过物体表面反色回来的光线强弱我们认为是灰度值。返回的光线越 强，灰度值越大，在250左右。返回的光线越弱，灰度值越小,在150左右。不同颜色的灯在不同背景下灰度值 不一样。所有颜色的灯在黑色下灰度值都是最小的，因为黑色把光给吸收了。当背景颜色与灯颜色一致时，光 基本不吸收，灰度值越大。

不同灯在不同背景下的灰度值:

LED灯-红色:在读取有红色的颜色(红、黄、紫)时，红色越饱和灰度值越高在250左右，黄、紫有其他颜色，灰度值会偏低。 无红色(绿、蓝、青)时，灰度值150左右。

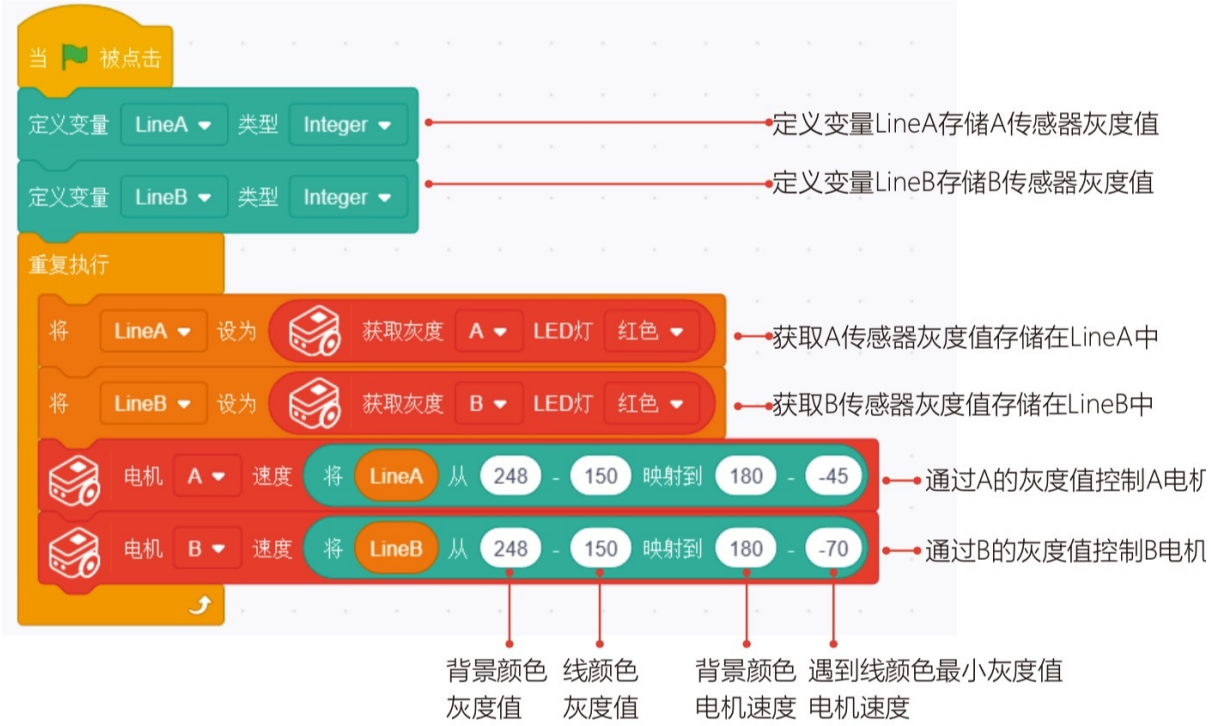
LED灯-绿色:在读取有绿色的颜色时，灰度值250左右，无绿色时，灰度值150左右。

LED灯-蓝色:在读取有蓝色的颜色时，灰度值250左右，无蓝色时，灰度值150左右。

LED灯-关: 灯关了就只能读取环境光返回的光线强度，环境光越强灰度值越大，在125左右，无环境光时，值为0。这个模式 适合在反光的屏幕上运行，如ipad、电脑屏幕。

典型案例——灰度巡线:

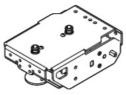
线和背景颜色读取的灰度值有差别时才可以巡线。例如线是黑色，背景是红色，用红灯来巡线。因为红灯 在红背景上灰度值最大，在黑色上灰度值最小。如果线是红色，背景是白色，这是需要用绿色和蓝色，因为 白色的灰度值最大，所有要找一个在红色上灰度最小，可选绿色、蓝色，推荐蓝色，蓝色在红色上最低。其 它颜色大家就举一反三了:


当 被点击
定义变量 LineA 类型 Integer
定义变量 LineB 类型 Integer
重复执行
将 LineA 设为 获取灰度 A LED灯 红色
将 LineB 设为 获取灰度 B LED灯 红色
电机 A 速度 将 LineA 从 248 - 150 映射到 180 - -45
电机 B 速度 将 LineB 从 248 - 150 映射到 180 - -70
定义变量LineA存储A传感器灰度值
定义变量LineB存储B传感器灰度值
获取A传感器灰度值存储在LineA中
获取B传感器灰度值存储在LineB中
通过A的灰度值控制A电机
通过B的灰度值控制B电机
背景颜色灰度值
线颜色灰度值
背景颜色电机速度
遇到线颜色最小灰度值电机速度

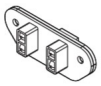
组装百变小钢

step1

用到的零件

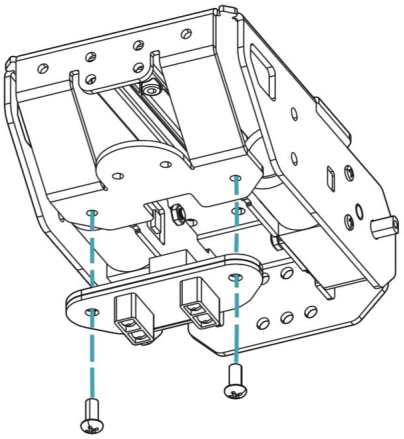


车体总成 X1



光电模块 X1

M4*8mm螺丝 X2



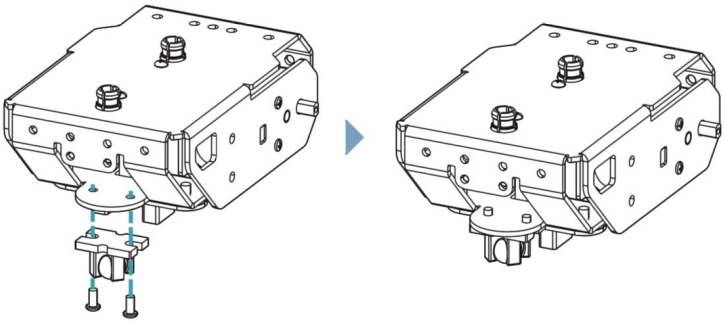
1.首先拿出总成车体和，将光电模块用螺丝拧到车体底部,如图。

Step2

用到的零件

白色万向轮 X1

M4*8mm螺丝 X2



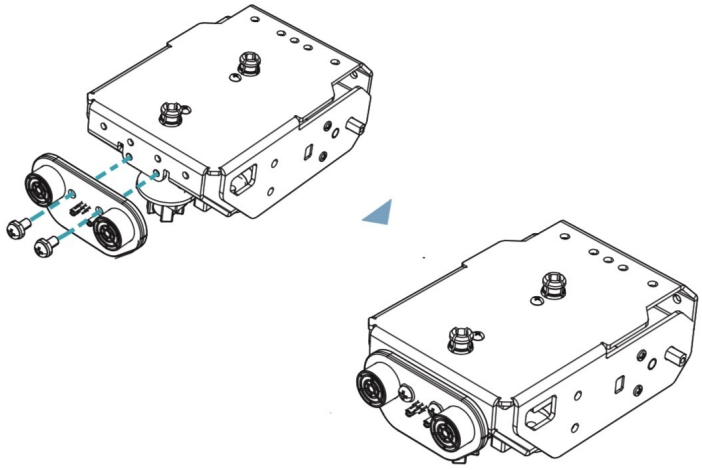
2.用螺丝将万向轮安装在车体底部的位置上，如图。

Step3

用到的零件

超声波模块

M4*8mm螺丝 X2

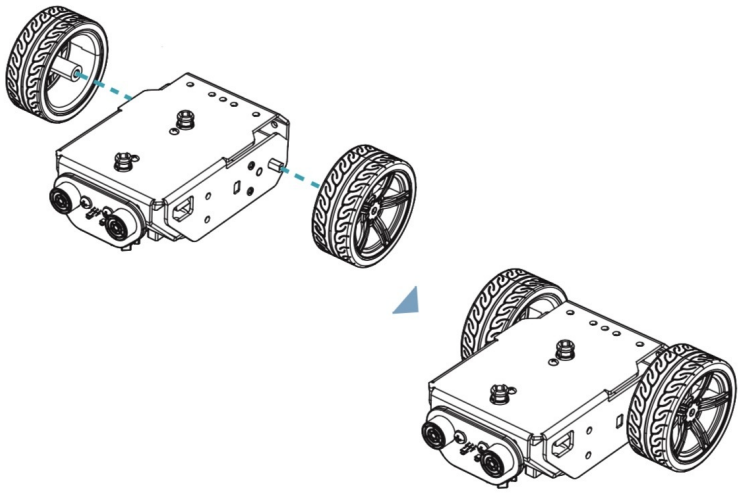


3.将超声波模块用螺丝装车身前方，如图所示。

Step4

用到的零件

车轮 X2

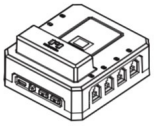


4.将两个车轮安装在总成车体上TT电机露出的地方，这里不需要螺钉固定，直接对齐插上即可。

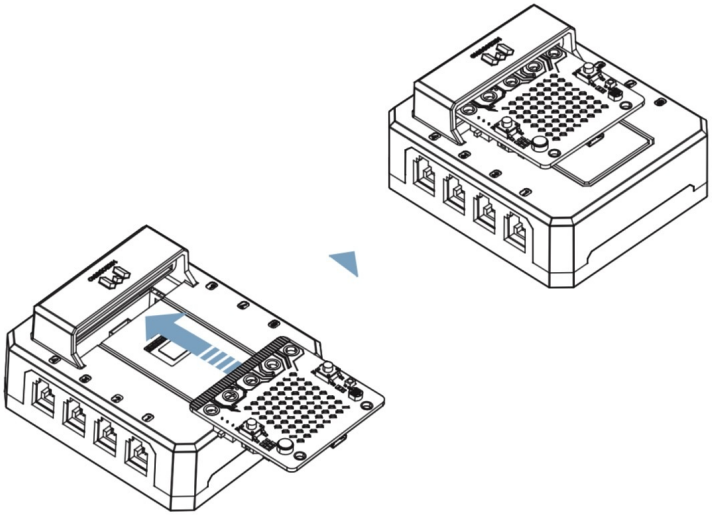
Step5

用到的零件

指间Matrix开发板 X1



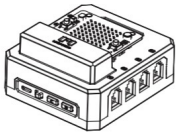
主控 X1

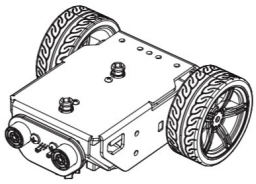


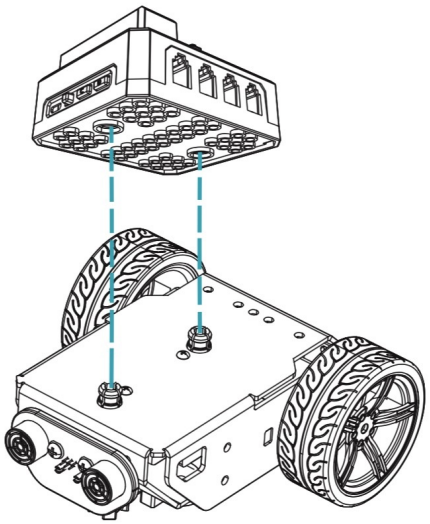
5.将指间Matrix开发板水平插入主控顶部的水平插槽中，让开发板屏幕朝上，如图。

Step6

用到的零件

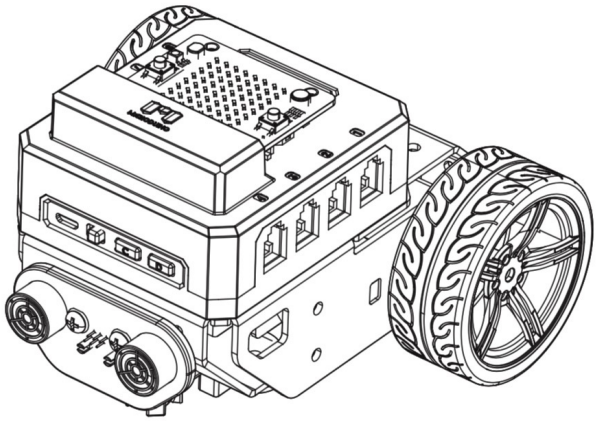






6.将刚才组装好的车体和主控拿来直接拼装到一起即可，不过注意安装方向并将孔位对齐。

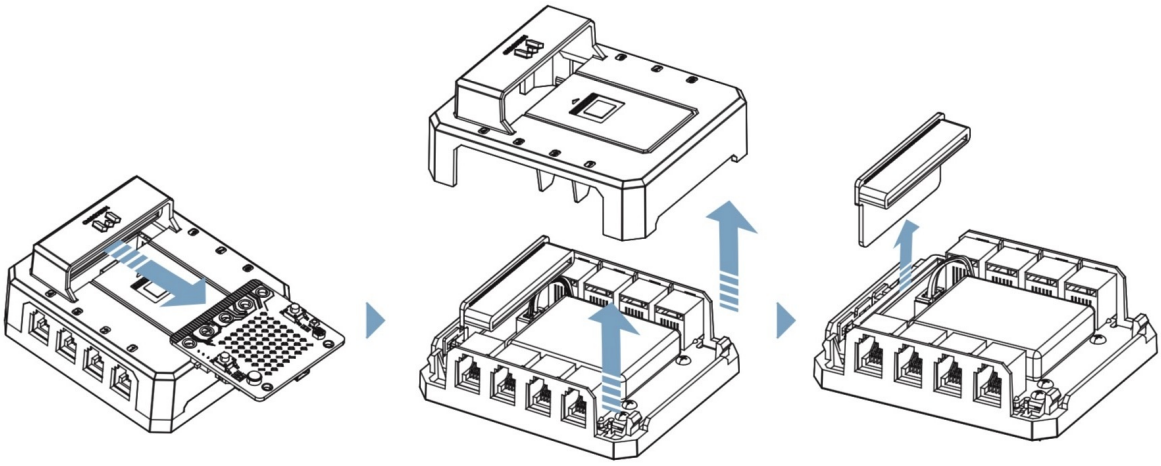
Step7



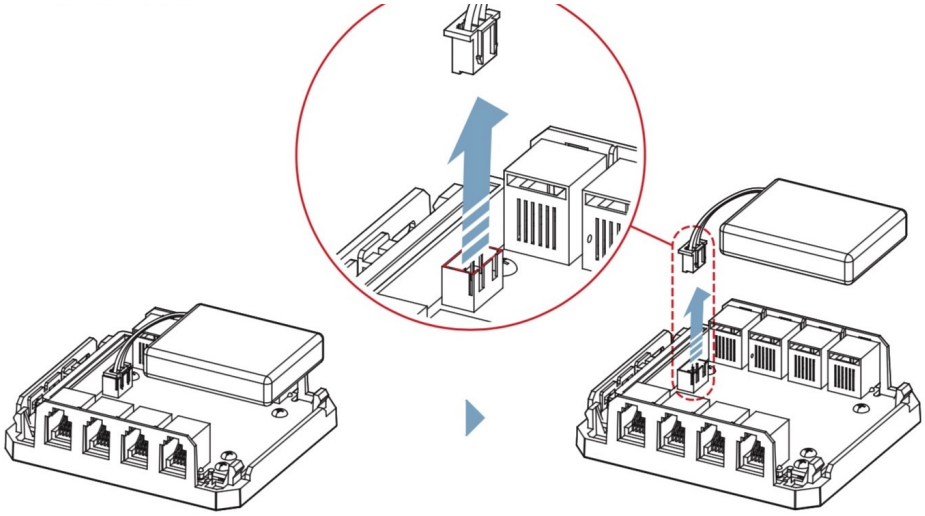
DONE! 完成

电池更换流程

1、将主控上的指间Matrix开发板、上盖、转接板按箭头方向分别依次拔出。

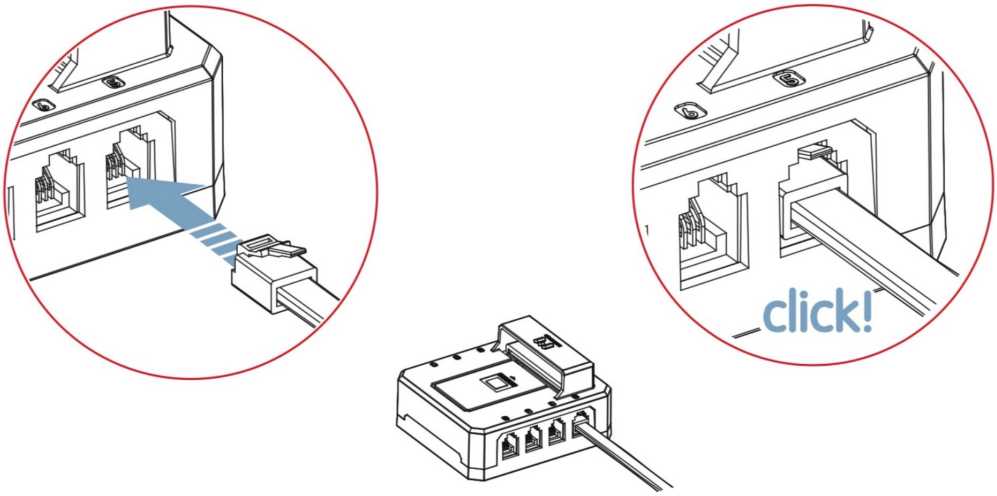


2、此时电池与pcb的连接部分就露出来了，只需要按图示拔出，换新电池即可。



数据线连接方式

插数据线时孔位选择根据实际用途适当调整，具体插接方式如图，听到嚓哒声后至牢固即可。



注意事项

1、只能用百变小钢专用电池，标准为7.4V，2S动力电池。

2、请勿带电插拔指间开发板，组装百变小钢时应关机。

3、电子产品，使用时请避免接触水或处于潮湿环境。

4、避免过冷或过热的环境中使用、存放百变小钢，可能会导致工作不正常。

5、请勿将百变小钢接入不支持的设备，或将其他设备接入百变小钢。

6、更换电池时请勿拉扯电源线，应夹住连接器小心拔插。

7、请确保电池插入方向正确(连接器有防反插设计)。

8、请勿短接电池正负极。

9、可以将microUSB插在在百变小钢上为电池充电。

小钢编程

1、自识别传感器概念

当在插入「自识别传感器」时，程序块列表会显示相应的程序块，其中数字代表接口编号。

2、自识别超声波传感器

利用测量超声波时差的方法，超声波传感器得以检测前方障碍物距离。

3、自识别光电传感器

此传感器可检测表面颜色及灰度，有两个程序块

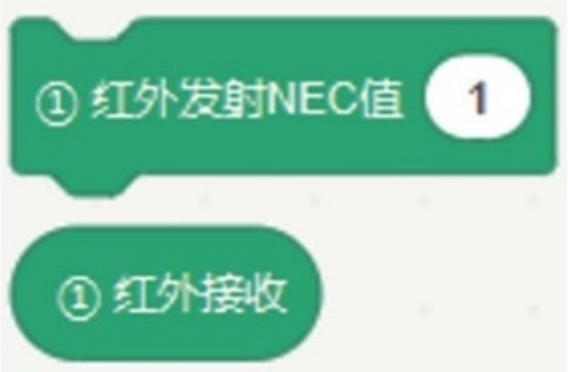
可以这样搭建程序块

4、自识别彩灯

此传感器有一个程序块，可以单独或整体控制板载4颗全彩灯的颜色和亮度

5、自识别红外遥控

红外遥控板载红外接收和红外发射两部分功能，支持NEC码，有两个程序块



可以配合其他传感器，实现远程遥控应用

6、自识别TT电机

TT电机是百变小钢的动力输出，有两个程序块，可以用来设置速度及检测速度

通过调整车身两侧TT电机的不同速度，可实现前进、后退和转向的操作

7、自识别手势传感器

手势传感器可以检测物体在表面的移动以及物体距离

有两个程序块可以作为控制输入，操作其他外接设备

V2.0.1 (2019/5/9)

- 新增功能引导；
- 新增「数组」扩展，数字光照程序块；
- 上传程序增加错误提示；
- 修复若干已知bug；

V2.0.0 (2019/4/12)

- 升级UI，PC版、web版、Chromebook版统一UI和交互；
- 自定义布局，根据编程需要，拖拽改变窗口大小，适应各种屏幕；
- 优化设备连接，可自动烧录固件，增加「重置设备」功能；
- 优化项目存储，统一各平台存储，项目文件升级为.mdx，兼容.md3、.sb3文件；
- 新增「百变小钢」、「指间Lite」产品套件；
- 新增「姿态检测」扩展；
- 优化Python功能，增加错误提示、新增多角色、广播、函数、画笔等功能；
- 重构Arduino代码生成，新增初始化、函数、for循环、间隔执行、RTC、EEPROM等程序块；
- 优化「串口调试」功能，调试无需重新连接设备；
- 新增软件自动更新功能；
- 修复若干已知bug；

V1.6.4 (2019/2/1)

- 合并最新官方Release代码
- IBB套件增加姿态传感器指南角度程序块
- 增加葡萄牙(巴西)语言
- 「IBB套件」新增电机模块

V1.6.3 (2018/12/26)

- 新增MicroRobot扩展，支持NOC中的电机、舵机等传感器
- IoT分类下，新增JoyPad与nRF通信程序块
- 扩展中心增加开关，可以控制扩展的显示与隐藏
- Buggy改名为「IBB」
- 修复若干已知bug

V1.6.2 （2018/11/30）

- 新增Buggy扩展，支持蓝牙/nRF连接、LED彩灯、OLED、点阵屏、超声波等传感器

- 新增百变小强连接和编程的使用教程
- 修复若干已知bug

V1.6.1 （2018/11/2）

- 新增Buggy扩展，支持舵机、乐高电机、姿态传感器
- 优化产品分类，「mCookie扩展」添加到扩展中心
- 优化硬件连接文案「已经连接」改为「等待烧录」
- IoT分类下，新增「nRF」程序块
- 新增「Big_F」「Conn」「Eddie」等6个角色
- 解决各版本文件的兼容性
- 修复若干已知bug

V1.6.0 （2018/10/10）

- 新增Buggy产品，支持scratch、Python、Arduino编程
- 新增扩展程序，支持Pen、Music等扩展
- 优化软件安装方式，一键安装软件和驱动程序
- 修改mCookie程序块分类「动作、显示、播放、感知」
- 优化IoT程序块名称
- 优化ideaBox程序块颜色和分类
- 新增「使用说明」和「报告问题」功能
- 修复若干已知bug

V1.5.0 （2018/8/9）

- 修改「实时online」与「上传offline」两种执行模式切换入口
- 「实时online」模式下支持Python控制舞台角色
- 优化AI模块，图像识别和语音识别交互，支持自动拍照和定时录音
- 更改套件选择与烧录固件UI
- 修复若干已知bug

V1.4.0 （2018/7/2）

- 增加了用户中心功能，支持中国版与国际版
- 增加了文件云端存储功能
- 统一保存工程文件为md3，包含sb3，py，ino文件
- 优化AI语音识别采集输入
- 修复若干已知bug

开源软件鸣谢

scratch-blocks

scratch-vm

scratch-gui

scratch-paint

scratch-render

scratch-storage

scratch-svg-renderer

node

electron

antd

react

tensorflow