

www.M5Stack.com

M5StickC
ESP32物联网开发
套件入门教程

Revolutionary All-in-One Compact IoT Solution

www.M5Stack.com

深圳市明栈信息科技有限公司
Shenzhen Mingzhan Information Technology Co.,Ltd
深圳市南山区西丽桃园街道珠光北路88号明光科技园1栋106
邮箱: Support@m5stack.com

M5STICK



CONTENTS

开始之前

UIFlow编程器	05
UIFlow布局	07
功能菜单栏	09
程序结构	10
程序的保存与打开	11

课程

HelloWorld	15
DIY手表	21
闪现的小球	27
加速度计应用	31
电子名片	35
躲避方块	38
超级音乐家	44
电子温湿度计	52

附录

配置WIFI	59
更新固件	63

[了解更多](#)

BEFORE WE START:

FUNCTIONAL

UIFLOW MENU BAR

UIFLOW PROGRAMMER

BEFORE WE START

INTRODUCTION

「开始之前」

BEFORE WE START

M5StickC

M5StickC



开始之前
Preface



M5StickC 功能菜单栏
Operations Menu



UIFlow 编程器
UIFlow IDE



UIFlow介绍
UIFlow Introduction

M5StickC

M5StickC

M5StickC

开始之前

一、M5StickC是什么？

M5StickC 是一款可编程的控制器。内部集成芯片ESP32、提供WIFI、蓝牙模块，6轴IMU运动传感器（MPU6886）、单色LED灯、红外发射管等功能硬件。

提供两个物理按键以及多个拓展接口，无论是编程简单的开关控制或是复杂的高阶功能，这款精致迷你的开发板都能够为你的编程学习提供完善的硬件基础。

硬件构造



电脑的基本硬件组成

CPU处理器

电脑内部的一块集成芯片，电脑会使用CPU处理器进行数据计算，以及信息处理，它在电脑中担任着大脑的角色。

内存空间

内存空间是电脑中存取数据资料的部分，它有两种类型。

- 1、RAM（运行内存），我们可以将它理解为一个高速读写的暂存空间，运行中的程序将会被加载到RAM中，当程序被关闭或是设备关机时，RAM中的数据将被释放。
- 2、Flash（储存内存），电脑中的一个长期存储空间，即使是关闭了设备的电源，Flash中存储的数据也可以保存。

输入设备

电脑接受外部控制信号的一个部分，例如键盘，鼠标，触摸屏，麦克风等。

输出设备

电脑向外传递信息的一个部分，如显示器/屏幕，耳机/扬声器，打印机等。

二、M5STICKC具有与电脑相同的四个组件

与电脑相同的四个组件		
中央处理器	✓	✓
记忆体	✓	✓
输入设备	例如键盘、鼠标	例如按钮、加速度计、麦克风
输出设备	例如显示器、扬声器	LED、LCD屏幕、红外发射管



UIFlow是一款专为M5硬件体系设计的编程工具，采用Blockly图形化编程与Python代码编程方式，带给编程初学者友好的编程体验的同时，传递更为专业与硬核的编程知识。在本章节中将向你介绍如何去配置与使用UIFlow，并编程控制LED灯的点亮与熄灭，完成一个简单的程序。

01 下载UIFlow Desktop IDE

访问<https://m5stack.com/>，选择SUPPORT>Download选项，根据操作系统选择对应版本的编程工具UIFlow-Desktop-IDE，点击下载。



02 解压、运行程序

将下载完成的压缩包解压至文件夹中进行存放，双击打开UIFlow应用程序。



03 安装驱动

当应用程序检测到你的电脑还未安装USB驱动时，将提示你进行安装：点击Install，开始安装驱动，按照提示依次点击下一步，直到安装完成。



04 连接设备

安装完成后将自动进入UIFlow编程器，此时将M5StickC设备通过Type-C数据线连接到电脑。



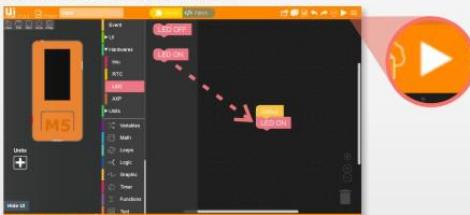
05 USB编程模式

长按机身左侧的电源键1秒进行开机，在出现UIFlow Logo后，快速单击Home键（中心M5按键），进入配置页面。按机身右侧按钮将选项切换至Setting，按下Home键确认。按右侧按键切换选项至USB mode，按下Home键确认，在USB编程模式，在IDE中选择相应的COM口与设备，点击连接。



06 运行程序

点击程序块列表中的Hardware选项，选择LED ON程序块并拖拽到编程区域连接SETUP，待程序编辑完成后，点击右上角的运行按钮，推送程序。



07 内容补充

M5StickC搭载了一个与UIFlow相对应的操作系统，为了获取更加丰富的功能与体验，你需要将固件更新至最新版本，更新固件的具体操作请查看附录“[更新固件](#)”。



程序块列表

程序块列表，是所有Block程序的一个集合列表，在这个菜单中你可以找到各种类型的程序，包括驱动类型的硬件控制类“Hardwares”，与数学运算型的“Math”，或是用于逻辑判断的“Logic”。

所有的程序块的使用与搭建与实际代码编程，在语法与执行逻辑上高度一致。



功能菜单栏

功能菜单栏

提供了论坛，帮助文档以及编程案例等相关连接，程序的运行与下载等基本操作选项、程序的本地保存与打开、与配置功能。



程序结构

Setup

"Setup"是程序中必须存在的一个块，程序是从"Setup"块开始运行的，并且只运行一次。



Loop

"Loop"是一个无限循环，当执行到它时，将会无限循环运行包含在块内的程序。



Wait

"Wait"是一个延时程序，当执行到它时，程序将会进行一段时间的延时。



程序运行流程



程序的搭建方式

UIFlow使用blockly可视化的图形编程语言，编辑程序的方式就像拼图一样；当两个程序块靠近时，可连接处就会被突出显示，此时松开鼠标，即可完成连接；我们只需要按照我们想要的执行顺序，将程序块连接起来，就可以完成构建。



UIFlow操作

Block复制操作: 右键选择要复制的块点击Duplicate或左键快速双击，进行复制。

Block删除操作: 将要删除的程序拖动至编程区域右下方的垃圾桶，或是程序块列表即可完成删除。

程序的保存与打开

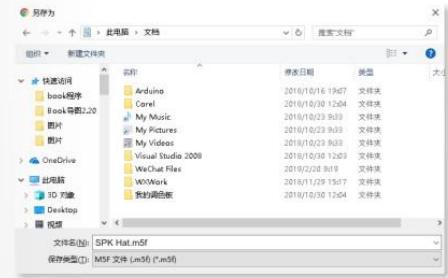
在以后的学习中，我们可能会编程一些功能复杂的程序，这使得我们并不能一次性将程序编写完成。当我们将工作进行了一部分的时候，我们可以将程序先保存到电脑本地，等到下次继续工作的时候再次打开；或是在工作中时不时的进行一些保存用作备份，这些时候我们都会使用到保存与打开功能。点击“Save”，选择要保存到的目录，点击保存。当需要打开程序时，则点击“Open”找到已经保存的以“.m5f”为后缀的程序文件，点击打开，UIFlow就会将文件中的程序重新加载到编程区域。



- 01 打开UIFLOW网站
<http://flow.m5stack.com/>



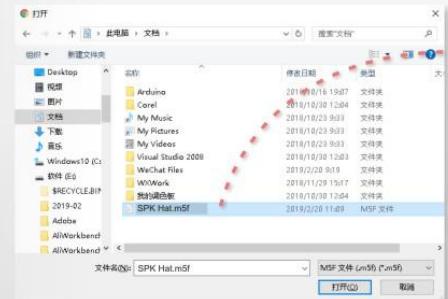
- 02 编程完你的程序后，
点击Save



- 03 将程序另存为后缀为
".m5f"的程序文件



- 04 当需要打开保存的
程序文件时，点击Open



- 05 找到保存后缀为
".m5f"的程序文件
点击打开即可



CURRICULUM:
SUPER MUSICIAN
「
FLASHING BALLS
CURRICULUM
课 程
TRAFFICLIGHT
」
PLANT STEWARD:
AVOIDANCE CUBES



1 HELLO WORLD

简介：使用UIFlow中的UI模拟器功能，编程实现文字与图片的显示。
目标：了解程序的编写流程，在M5StickC上运行你的第一个程序。

1-1 第一个挑战

"Hello World"程序通常是每个新程序员都会编写的第一个程序，它会将 "Hello" 和 "world" 两个词打印到屏幕上；这听起来十分的简单，但即使是一些技术高超的程序员在开始学习新语言或在新设备上测试一些代码时也会经常使用它，这是我们必须面对的第一个挑战；一旦你成功地运行了Hello world程序，它将使你有信心学习更高级更复杂的程序。



一、添加Label标签

在UI模拟器中，使用Label标签能够帮助我们快速的进行一些文本的显示，添加方式也十分的简单，只需要点击UI模拟器上方的Label元素选项，并将其拖动到屏幕上即可。



向屏幕添加Label标签后，在程序块菜单的UI选项将出现相应的Label控制程序块，接下来使用程序对标签显示的内容进行修改。



这时候，一个简单的Hello World程序就编写完成了，接下来就是要让这个程序运行起来，在运行程序前将M5StickC通过数据线连接到电脑上，并设置进入USB编程模式。



二、元素属性配置

在UI模拟器中，点击已有的元素将会弹出属性配置框，参考下方属性简介，修改标签的颜色、角度、以及默认显示内容。



三、添加/移除元素

UI模拟器上方有一些不同的元素选项，你可以将它们添加到屏幕上丰富显示内容。



同样的，在往屏幕上添加了元素后，程序块列表的UI选项将出现于其相对应的控制程序块。

一些不需要的元素，可以通过点击、拖动至UI模拟器右上方的垃圾桶进行删除。

四、控制元素

通过相应的UI程序块，我们能够通过程序对标签或其他的元素进行动态控制。

01 添加两个Label2控制程序，用来改变标签所显示的内容



03 为了使程序不断运行，添加"Loop"
02 添加两个Wait延时程序，用做控制内容变化的间隔时间

五、项目名称

程序编辑完成后，点击页面左上角的输入框可以给这个程序设置名称，例如：Hello M5。



六、下载程序/运行程序

在一般的程序测试阶段，我们常常会使用右上角的运行程序功能（三角形按钮）。但这仅仅适用于程序的测试，当设备重启，程序则会丢失。

如果我们想要程序永久性的保留在设备中且可以随时打开运行，则需要用到下载功能。

点击页面右上角菜单栏中的
"Download" 选项，开始下载程序。



当M5StickC屏幕显示由Uploading变为Resting，则表示程序已经下载完成，设备将自动重启并默认运行我们所下载的程序。



七、程序库

所有下载的程序，将被统一放到程序库中，在设备开机出现Logo后快速按下HOME键能够进入功能菜单，按下车机右侧的按钮切换功能至"APP.LIST"，按下HOME键进入程序库。



提示：切换与选择的按键方式与功能菜单一致（右侧按键切换，HOME键选择），长按右侧按键约3秒松开将提示删除程序，相同名字的程序在进行下载的时候将自动覆盖，重启将默认运行上次执行的程序，若想下载其他程序可以在开机时长按右侧按键2秒左右进入编程模式，或是进入功能菜单选择Program、或Setup功能，进行模式切换。



2 DIY手表

简介：使用变量与逻辑判断以及按键编程，用M5StickC制作一个可以调节时间的手表。

目标：学习变量概念以及创建与使用，学习逻辑判断的嵌套使用。

1-1 变量的理解

(1) 我们可以把变量看作一个瓶子，我们可以给变量取一个名字，例如A，往变量里存放一些数值，或是字符串内容。



(2) 当我们往变量里放置新的数据，旧的数据就会被覆盖。



(3) 可以把一个变量的内容赋值给另一个变量，这相当于把瓶子里的东西复制了一份，然后放到了另一个瓶子里。



01 点击程序块列表中的
"Variables"选项



创建完成后，在“Variables”选项中就会出现与这个变量相关的一些程序块。

Create variable...

```
set Count to [ ]  
change Count by [ ]  
Count
```



一、计时器

使用变量“Count”编程一个计时器，每经过一秒钟将其加一，并通过标签，将数值显示在屏幕上。

设置“Count”的初始值为“0”，将变量“Count”通过“Label show”显示，经过一秒的延时使“Count”加一。



问题：在进行手表程序编辑前，你是否知道，时分秒的概念。

解答：60秒=1分钟，60分钟=1小时，一天为24小时。

1-2 DIY手表

一、显示UI部分

首先我们需要添加三个标签用作“时”、“分”、“秒”的显示。

可以先预先输入数字，预览看看效果，同时将标签摆放整齐，点击标签选择不同的字体来修改大小，以及修改成你喜欢的颜色，可以再两个标签，内容为“：“分号，放在数字之间用作间隔，更方便我们查看时间。



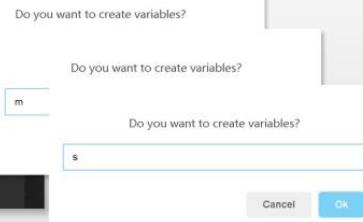
点击赋值块上的选项框，可以进行变量的选择以及重命名，删除操作。



- (1) 程序是由“Setup”开始运行的，在它的下方我们先进行程序一些数据的初始化，将“s”“m”“h”三个变量全部赋值“0”。
- (2) 使用三个“Label show”块将三个变量分别与标签上的时分秒位置一一对应的连接显示。
- (3) 将代表“秒”的变量“s”自身加一，“change s by 1”，添加一个1秒的延时。
- (4) 由于程序的运行速度极快，我们可以粗略的认为，每延时一秒钟，我们将变量“s”增加1。

二、程序部分

- ① 点击“Create variable”
创建三个变量
- ② 输入创建变量的名称“h”“m”“s”，
分别用来存放“时”“分”“秒”

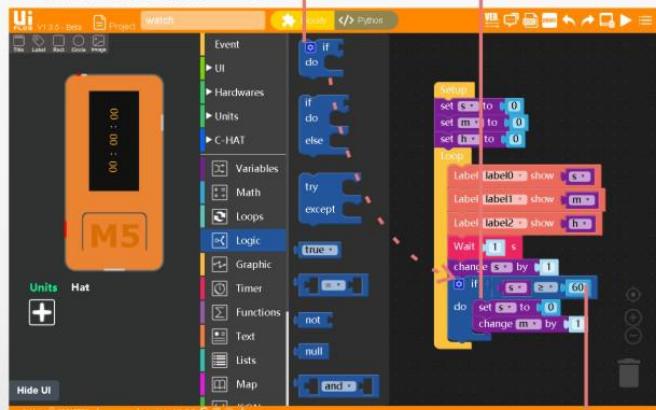


我们通过标签实现了各个时间变量的显示，对于时间的“分”、“秒”、“时”概念，或许大家并不陌生。(60秒=1分钟，60分钟=1小时，1天=24小时)但通过程序，我们要如何去编程实现，使得“秒”积累到一定数值能够自动的“进位”，使分钟加1呢？此时我们就需要学习一个新的程序块“if”。

“if”块的作用是，我们可以设定一个判断条件，当运行到if块时它会进行一次判断，当判断成立则执行do包含在内的程序，不成立则执行else包含的程序。



我们只需要控制满足条件时，将时间“进位”，并不需要使用到else判断。所以可以添加一个仅有条件成立执行的“if”块



变量“`s`”与常量数字块组合形成一个判断关系式， $s \geq 60$ ，将其连接到if块的右侧用作条件判断

三、逻辑判断

我们根据秒的变化实现分秒时的转换

使用循环判断，当“秒”大于等于60时，将“分”加一，且将“秒清零。

然后进行判断“分”是否大于等于60，如果是则将“时”加一，且将“分清零。

然后进行判断“时”是否大于等于24，如果是则将“时”清零。



添加两个按键检测模块分别用来调整分钟和小时。

3 闪现的小球

简介：在屏幕上绘制小球，理解坐标。

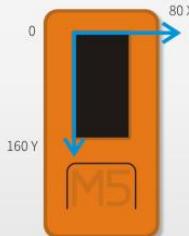
目标：认识坐标，学会基本的图形绘制，了解Random随机模块随机生成坐标。

3-1 认识坐标

坐标能够帮助我们准确无误的确定某个地方的位置，或是计算两个地方之间的距离。生活中坐标无处不在，世界地图就是其中一个应用案例，地图上的每一处角落都有着自己独立的坐标。而屏幕也是类似的，屏幕的显示部分由一个个的像素点均匀排列的组成，就像一张地图一样，每一个像素有着自己的坐标。



M5StickC 分辨率为 80*160，这意味着屏幕上有着 80*160 个像素，X 轴坐标范围为 0~80，Y 轴坐标范围为 0~160。当我们添加标签或是图形到屏幕上时，都会有一个坐标信息。需要注意的是，当元素坐标超出了屏幕所能显示的坐标范围，则无法完整显示。

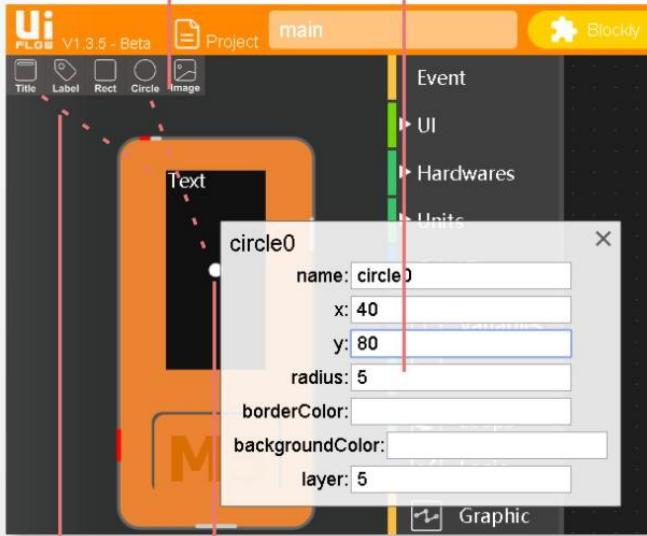


3-2 绘制小球

本节内容我们通过UI模拟器在屏幕上绘制一个小球，观察小球在屏幕坐标上的位置。

01 拖动 Circle 到屏幕上

03 修改 circle1 的参数，修改半径，能够调整圆形的大小



02 点击圆形，弹出设置窗口

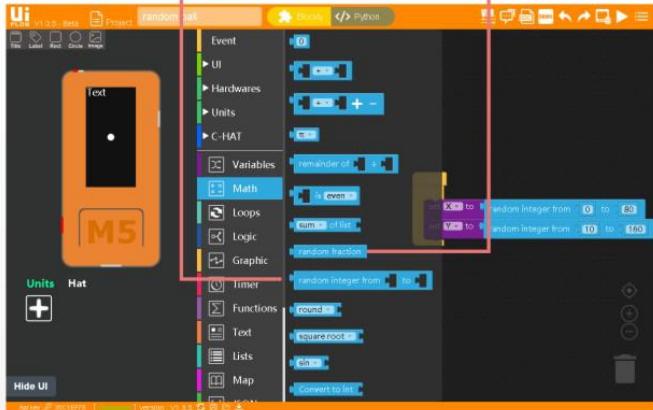
04 添加 Label 到屏幕上
为了使我们更加直观的了解坐标与位置的关系。
我们可以添加一个标签置于屏幕顶部，用来显示小球每次运动的坐标。

Random随机数

在“Math”选项中提供了两种能够帮我们随机生成数字的程序块

“random fraction”，随机生成一个分数

“random integer from a to b”，在我们指定的范围内(a~b)，随机生成一个整数



01 创建变量X,Y 储存随机生成屏幕坐标数值

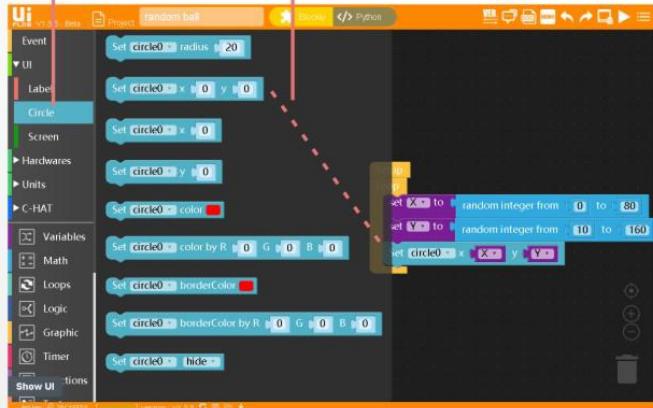
02 将X坐标随机范围设置为0~80

```
set [X v] to [random integer from 0 to 80]
set [Y v] to [random integer from 10 to 160]
```

03 为了使得小球不会出现遮挡标签信息的状况，我们需要将Y坐标随机范围设置为10~160

04 点击UI, Circle选项

05 将Circle坐标程序块中
的坐标信息设置为变量X,Y



完成上面的程序，就已经成功实现随机闪现小球动画了，接下来要做的就是将小球每一次变化的坐标信息都显示出来。我们只在屏幕上放置了一个Label标签，可是我们却有两个信息(X,Y坐标)需要显示，为了在一个标签上显示多个信息。

这里我们用到了两个新的模块：

Convert to str

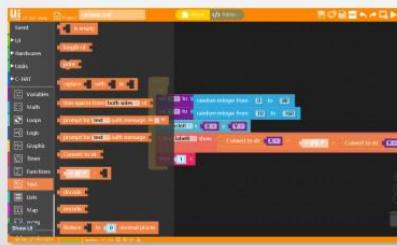
字符串转换

将数据类型转化为字符串类型，能够用于标签显示，或是字符串拼接。

String + String

字符串拼接

将两个或多个字符串按照顺序拼接成一个字符串。



放入字符串转化模块

加入空格和逗号

为了方便观察，可以添加一个1秒延时程序，减缓小球变化的速度

4 加速度计应用

简介：通过摇一摇，改变屏幕颜色。

目标：了解加速度计，学习逻辑判断的应用。

加速度是什么？

加速度指的是单位时间内速度改变程度，一般情况下，加速度是个瞬时概念，它的常用单位是米/秒²等。在最简单的匀加速直线运动中，加速度的大小等于单位时间内速度的增量。

例如，一辆正在进行匀加速直线运动的汽车从“20”m/s经过一秒加速到达了“30”m/s的速度，那么它的加速度就为10m/s²，加速度能够传达给我们“速度变化的快慢”这一信息。

问题：有两辆汽车都有静止状态，启动后均匀加速到40m/s,汽车A用了10s,而汽车B只用了6秒，请问它们的加速度一样吗？

解答：很显然，它们二者的加速度是不一样的，花费时间更少的汽车B的加速度要大于汽车A的加速度。



加速度在生活中无处不在，即便是一次简单的挥手动作，其中有着速度变化的过程，下面将以加速度这一主题，将你讲解两个有趣的功能应用。



接下来，我们将利用M5StickC甩动时所产生的加速度，对屏幕颜色切换进行控制。

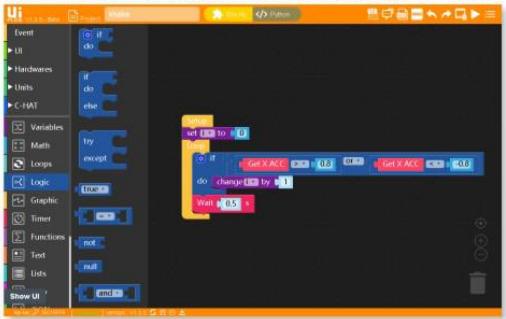
- (1) 创建变量“i”，用来标记屏幕切换的颜色
- (2) 在Setup初始化程序时，赋值变量“i”的初始值为“0”



4-2 逻辑判断

通过检测加速度，我们可以判断M5StickC是否被摇晃，如果摇晃改变屏幕颜色。每当摇晃一次都会按照预先设定的顺序去显示。

添加变量记录晃动的次数，晃动次数决定了显示的颜色，初始值为0。在loop中判断有没有晃动发生。



根据实验测得大幅度晃动X方向的加速度为 ± 0.8

逻辑运算有三种，and（与），or（或），not（非）

- (1) **and**，左右两边的条件都满足结果为真，有一个不满足条件为假。
- (2) **or**，左右两边的条件至少有一个满足结果就为真，都不满足结果为假。
- (3) **not**，条件取反，如果原来结果为真，则**not**为假，如果原来结果为假，则**not**为真。

根据变量*i*的变化切换不同的颜色，如果*i*的值超过4，重新赋值为0。



5 电子名片

简介：制作一张电子名片，按下按钮时循环显示个人信息。

目标：认识列表。

一、列表的理解

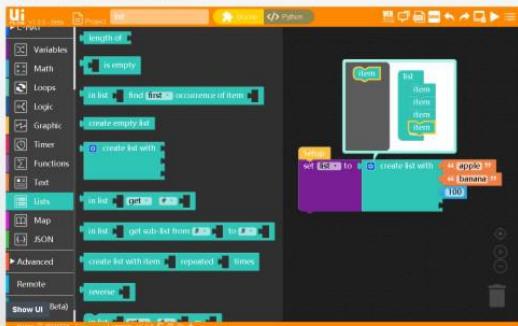
什么是列表？我们可以把列表理解为多个变量的集合，如果把变量比喻作一个瓶子，那么列表就一个可以放置多个变量的一个柜子，这些变量存放在列表中会被排序，通过这些序号我们可以访问它们，或是改变它们的值。



在列表的创建时，我们会用一个变量去存储它，在之后的使用中，只需要访问这个变量并指定列表中元素的序号，就可以访问到列表中的数据。

二、列表的创建

我们需要创建一个变量用做整个列表的存放；点击程序块列表中的“Lists”选项，选择“Create List with”块；我们可以看到该块的右侧，有着许多嵌口，这些是用让我们存放数据用的，我们可以存放数值或字符串，变量等数据到列表中；这些数据会由上而下，以 $1 \sim N$ 的顺序编号，当通过列表调用这些数据的时候则需要使用正确且对应的编号每一个嵌口连接的数据都将成为数组的一部分，点击块上的小齿轮，添加“item”到list中，可以创建更多的数据接口用来存放数据。



三、数组的使用

当我们在数组中存放了一系列的数据后，要学习的就是如何去按我们的需求去调用数组中的任意数据。



03 在list部分放置需要使用的数组，#部分放置调用数据的编号

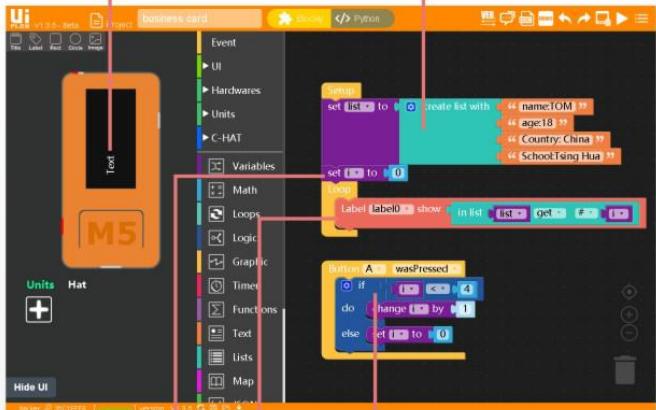
在UIFlow中，数组的数据是以 $1 \sim N$ 的顺序编号的，例如我们想要调用数组中的“MS”数据，它对应的编号为“1”，那么在调用时，就需要将编号“1”放置到#位置。

问题：可以使用一个变量表示编号去调用数组的数据吗？

解答：是可以的，这使得同一条程序的调用结果有着更多的可能性。

5-2 制作电子名片

- 01 添加标签，用于显示名片内容
由于名片内容较长，为了更完整的显示内容，我们可以将标签旋转270°
- 02 创建一个列表用来存放名片的个人信息



- 03 设置一个变量*i*用于索引列表，初始值为1
- 04 使用Label show程序将名片信息显示到屏幕上
- 05 使用按键程序块，去控制变量的增加并同时切换对列表内容的索引
(由于所创建的列表含有4个元素，其索引范围为1-4，为了防止变量*i*超出索引范围4，我们需要通过判断程序对其进行限制)



6 躲避方块

简介：屏幕上方不断下落方块，利用陀螺仪控制小球躲避方块。

目标：学习循环迭代，了解循环条件的设定，了解游戏的设计流程。

2-1 认识迭代

我们可以将迭代程序理解为，将指定的一组数据按顺序，赋值给一个变量，并且在每次赋值后，执行一次我们自定义的程序。

我们可以简单的将迭代过程，比喻成一次课堂点名，老师按照点名册上的名字顺序的喊出学生名字，如果把每次被点的名字当作一个变量，点名册上的名字当作一组数据。

按照点名册（数据）顺序的喊出学生名字，被叫到的名称（变量）随着点名的进行，按顺序的发生变化，这样的一个过程，可以称为迭代。



问题：为什么要使用迭代？迭代对程序能产生什么样的优化？

我们可以通过一个简单的例子说明。

例如：我们需要将变量*i*按顺序赋值从0到100，每隔一秒增加1，并且不断循环这个过程，我们尝试使用迭代，与不使用迭代去写这个程序，看看它们有什么样的区别。



方法一

当“*i*”小于100时；
执行“change *i* by 1”；
当“*i*”大于等于100时；
则执行将“*i*”重新赋值为“0”；
每次判断运行结束后进行一次1秒延时；
通过Loop无限循环这个过程。



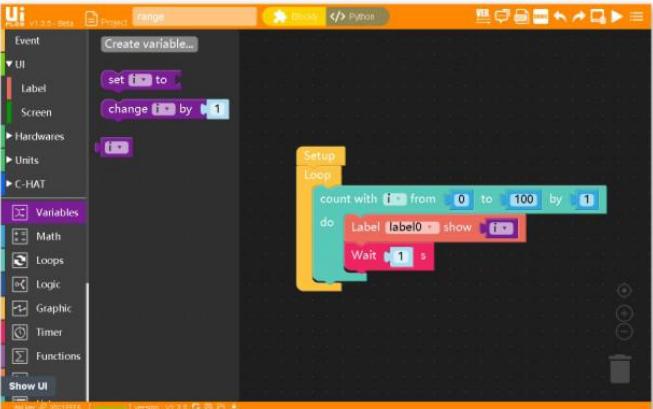
方法二

使用迭代“count with i from”；
填写迭代的参数：将“i”的初始值赋值为“0”，上限值为“100”，每次增加的量为“1”；
在每次执行变量“i”加1后，都会执行一次do右侧的程序，延时一秒；
通过Loop无限循环这个过程。

使用迭代程序很多时候都能够帮我们精简程序，提升程序的可读性。
迭代的对象并不是唯一的，这使得迭代程序能够帮助我们处理更复杂的程序场景。

补充

在UIFlow中，当我们将迭代程序添加到编程区域后，会自动生成一个提供你迭代的变量，你可以在“Variables”选项项中找到。



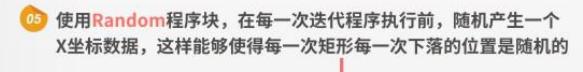
6-2 利用迭代控制坐标



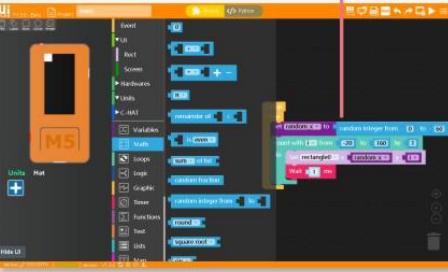
- ① 添加矩形图案
② 通过迭代程序，不断的更新方块的坐标，使其产生向下运动的效果

由于矩形图案的坐标点位于它的左上角，且当前矩形的宽度为20，为了实现矩形“从无到有”的下落过程，我们应该设置迭代的范围为-20 ~ 160

- ③ 通过调整延时的时间，能够调整方块运动的快慢



- ④ 使用Random程序块，在每一次迭代程序执行前，随机产生一个X坐标数据，这样能够使得每一次矩形每一次下落的位置是随机的



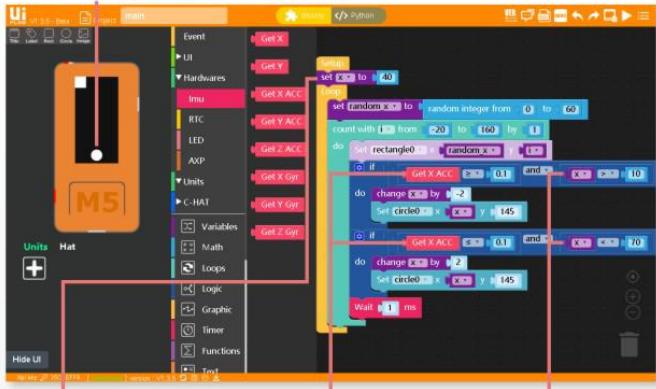
- ⑤ 将随机生成的数值，用于控制矩形的X坐标
⑥ 同样由于矩形图案的坐标点位置是位于左上角，因此在我们应当将随机范围设置在0 ~ 60，避免出现矩形位置离开屏幕范围

- ⑦ 将随机生成的数值，用于控制矩形的X坐标

6-3 控制小球

现在我们需要绘制小球作为操纵的角色，小球左右方向的控制使用加速度计实现

添加圆形图案，将其放在屏幕底部



添加变量X控制小球坐标，初始值为40让小球出现底部在中间位置

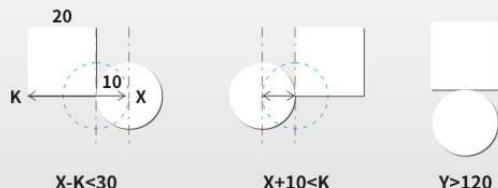
通过加速计的变化判定倾斜方向控制变量X的增减从而控制小球滚动

判定小球坐标防止超出屏幕范围

6-4 游戏判定

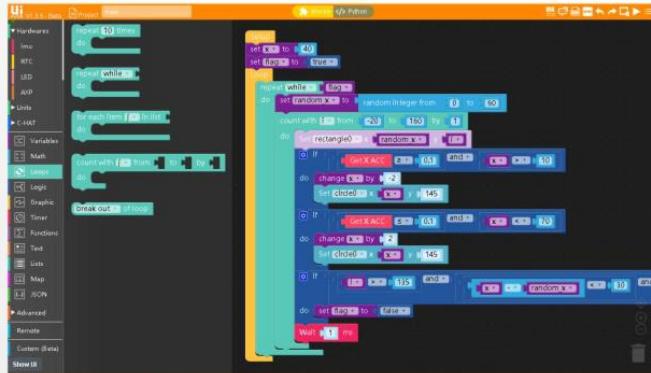
作为游戏，不可避免的会有游戏失败的判定和游戏得分，接下来我们分析一下如何判定。

这里简单的分为三种情况进行讨论，方块的右侧与小球左侧接触，方块的左侧与小球右侧接触，方块下方与小球接触。只有三种同时满足才意味着游戏失败。



游戏运行

可以发现，当点击运行程序后，游戏就自动开始了，而我们并没有控制游戏的开始。在游戏中，当矩形和小球发生碰撞后，意味着游戏结束，我们需要使游戏终止。等待下一局的开始，可是目前的程序运行结果，即便发生碰撞还是不停有矩形在下落，我们并没有控制游戏的结束，我们可以使用一个“标志”来判断当前的游戏状况是结束，还是正在进行，使用这个标志来控制游戏的进行与结束。



repeat until

不满足条件时循环执行内部的程序块，满足条件退出循环

repeat while

满足条件时循环执行内部的程序块，不满足条件退出循环

6-7 完善功能

现在我们已经实现了基本的游戏界面，并且游戏失败时画面会暂停。接下来我们需要加入计分和重新开始游戏的设置。

```

when green flag clicked
  set [score v] to [0]
  set [flag v] to [true]
  hide
  show
repeat
  set random x to random integer from [0] to [60]
  set random y to random integer from [20] to [160]
  if [x] <= [0] or [x] >= [60]
    change [x] by [-2]
    set [circle x] to [x]
    set [circle y] to [y]
  else if [Get X ACC v] <= [0.5] and [y] <= [70]
    change [x] by [2]
    set [circle x] to [x]
    set [circle y] to [y]
  else if [x] <= [35]
    if [random x] <= [30] and [x] >= [10]
      random x
    end
  do
    change [score] by [1]
    if [A] pressed
      show
    end
    if [Laser Launched] = [show]
      score + 1
    end
    if [A] pressed
      set [flag v] to [true]
    end
  end
end
when [A] key released
  hide
  show
end

```

变量score记录分数
隐藏分数，显示方块与小球
每次躲避完成一个方块记录一分
游戏结束隐藏方块和小球

在按键A按下之前始终显示分数，在游戏结束时坐标没有到达160，并不会直接跳出while循环，while内的程序会全部运行完一次才跳出，因此分数多比实际多1，这里我们需要进行处理，将分数减1，按下按键后flag重新设置为True

7 超级音乐家

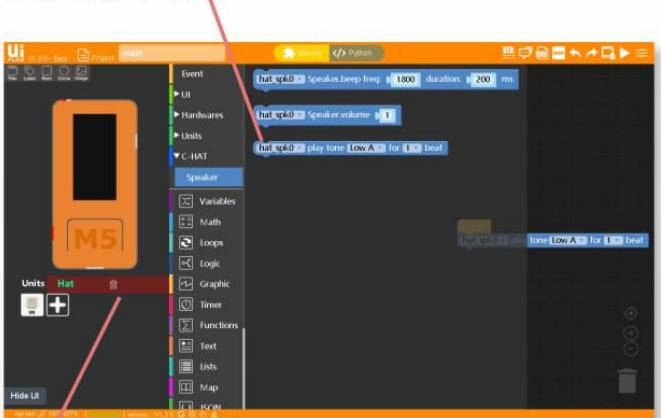
简介：使用Speaker块去驱动M5StickC发出声音，制作一段音乐，使用按钮控制与小伙伴们来一次大合奏游戏。
目标：了解声音的原理，加深程序运行结构的了解，学习次数循环Repeat。

HAT的添加



HAT的使用

当我们成功添加了一个HAT功能模块，在UiFlow的程序块列表中的C-HAT选项，就会出现与它对应的程序块，不同的HAT会有不同的程序块，根据功能的不同，提供的程序块个数也可能不同我们在使用HAT的程序块时，只需要把它按照需求添加到程序中去，“在运行程序前，将相应的HAT模块插接到M5StickC上方的拓展端口上”，完成连接后，运行程序，即可使用。



HAT的移除

假如有错误添加的HAT，或是已经添加的HAT我们不需要使用，我们可以把它移除，移除的方法也非常简单。我们只需要点击已经添加的HAT，这时候你会发现在其一旁出现了一个“垃圾桶”，将HAT拖动到垃圾桶中松开，即可完成删除。

1-1 制作个人音乐

我们将学习如何编程驱动M5StickC发出声音，并制作一些简单的音乐。
我们每天会听到很多很多的声音，你是否停下来思考过，为什么会有不同的声音？声音是怎么产生的？

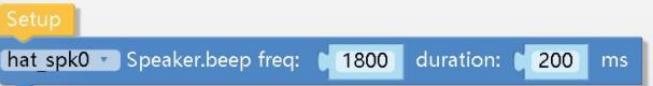
物理小实验：我们可以把尺子放在桌子并将一小段伸出桌外，用手去拨动尺子，当尺子伸出桌面的部分较多时，拨动后尺子摆动缓慢且发出的声音低沉，当尺子伸出桌面较少时，拨动后尺子摆动快速且发出声音高亮。



实验分析：通过拨动尺子使其震动，验证了“声音是由震动产生的”这一物理原理，尺子震动的速度也就是我们所说的“频率”，当频率越高，则音调越高，频率越低，音调越低，一般人的听力范围为20Hz到20KHz，如果大于，或小于这个范围的声音，我们则无法听到。

音调与频率

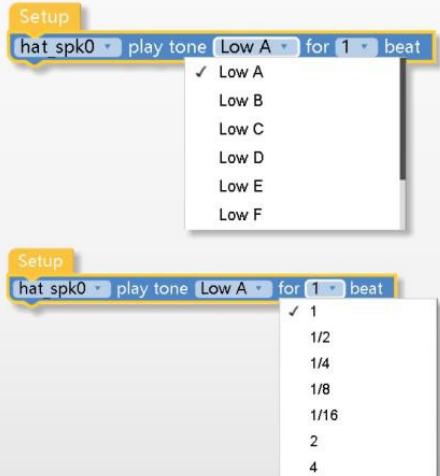
了解了声音的原理，我们开始使用一下Speaker块来驱动M5StickC发出声音，点击程序块菜单中的Speaker选项，将“Speaker beep freq”块拖动到编程区域，并连接到“Setup”，点击块上的参数框，可以修改声音的“频率”“持续时间”，尝试修改不同的数字并运行程序，听听声音有什么不同吧。



在一些乐谱上面，我们能够看到一些音符，像是“C-D-E-F-G-A-B”，事实上这些音符只是对不同频率的称呼，使得我们更容易记住和使用。

	C	D	E	F	G	A	B
Low	131	147	165	175	196	220	247
Middle	262	294	330	349	392	440	494
High	523	587	659	698	784	880	988

在Speaker选项中，也提供了发出指定音调的块“Play tone”，将它添加到程序中；点击块上的三角按钮选择不同的音调，你还可以设定它的节拍。



我们尝试编辑一个程序，来测试频率与音调的关系。



03 选择 Speaker 选项



04 分别添加“Speaker beep freq”，以及“Play tone”

05 运行程序，分别按下按键A与按键B测试，你会发现，它们发出的声音是一样的

使用“Play tone”尝试制作一小段音乐。



当我们想要编写一段长的音乐时，随着使用音调数量的增加你会发现程序变得越来越长，但其中却有很多重复的程序片段，我们有没有办法能够优化程序，使其变短的同时，又不影响原来的功能呢？

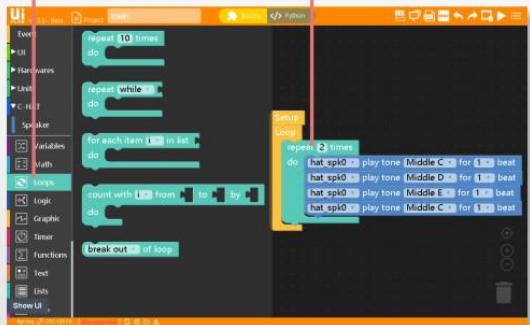
答案是有的！它就是“Repeat次数循环”。

当执行到“Repeat”时，其包含在内的程序块，就会按照我们指定的次数，循环执行。

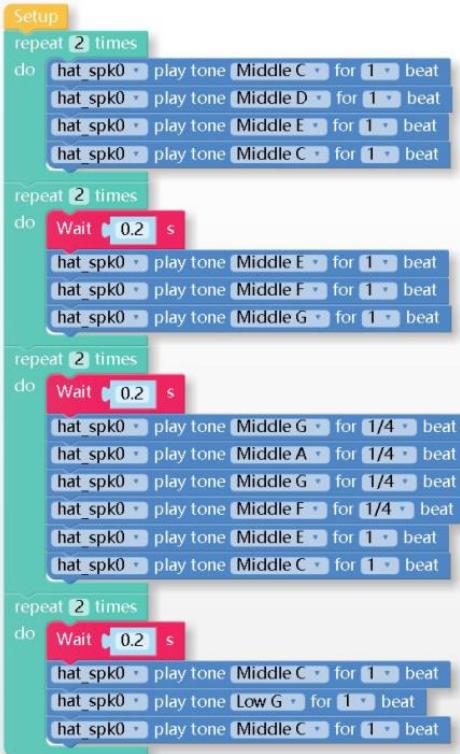


01 点击Loop选项

02 修改Repeat循环次数为2次



执行的效果是一样的，不同的是，使用了“Repeat”的程序，变得更加精简且易懂。



在日常编程中，我们要保持优化程序的习惯，除了实现功能以外，还要有清晰的结构与可读性，这样才称得上是一个好程序。

1-2 大合唱

使用按键控制与“Play tone”块，当按下不同按键时，发出不同的音调，你可以与几个小伙伴一起，每个人分别代表一个或者多个音符，配合的来一场大合奏。



8 电子温湿度计

简介：使用ENV模块制作一个显示温湿度的仪器。

目标：了解ENV模块的使用，强化UI设计。



8-1 认识ENV

ENV模块内置了温度、湿度、大气压强和罗盘，是一个集成度很高的传感器模块，温湿度的变化对植物的生长有着重要影响，利用ENV我们可以很方便的采集温湿度数据帮助我们了解环境变化。温度测量范围：温度：-20~60°C，湿度：20~95%RH，大气压：300~1100hPa。



8-2 读取数据

借助标签我们可以很方便的显示出当前的温湿度和气压



接下来我们设计一个UI界面，让数据显示更加直观一些。

8-2 数据图形化显示

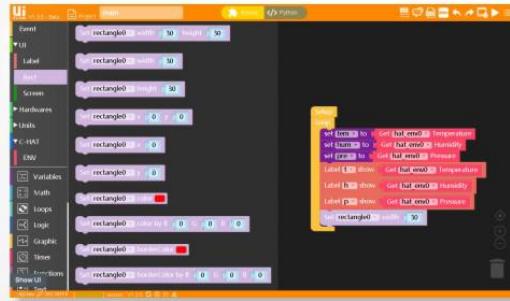
在UI模拟器中插入6个Label标签，三个用来显示文字，三个用来显示数据，同时我们添加Rect长方形做图形指示。



添加Rect，宽度为1，
设置边框颜色与背景
颜色

添加Label，显示数值

添加Label，显示名称



我们想要通过矩形条将温度、湿度、大气压值显示在屏幕上，但这将面临一个问题。M5StickC的屏幕分辨率为80x160，而传感器所检测的各项数值，都有着特定范围（温度：-20~60°C，湿度：20~95%RH，大气压：300~1100hPa）如果我们将这一数值直接用于改变矩形的宽度。在屏幕上它将没有一个合适的量程宽度，又或是直接超出屏幕所能显示的宽度，即便有数值很大的变化我们依然无法察觉。

因此，为了解决这一问题，我们需要学习一个新的程序块“map”（数据映射）



- 01 点击Advanced，
选项Easy IO
- 02 map 数据映射块

将数值范围为“-20 ~ 60”的温度值通过数据映射，转为0 ~ 30，使其能够合适的调整矩形宽度。



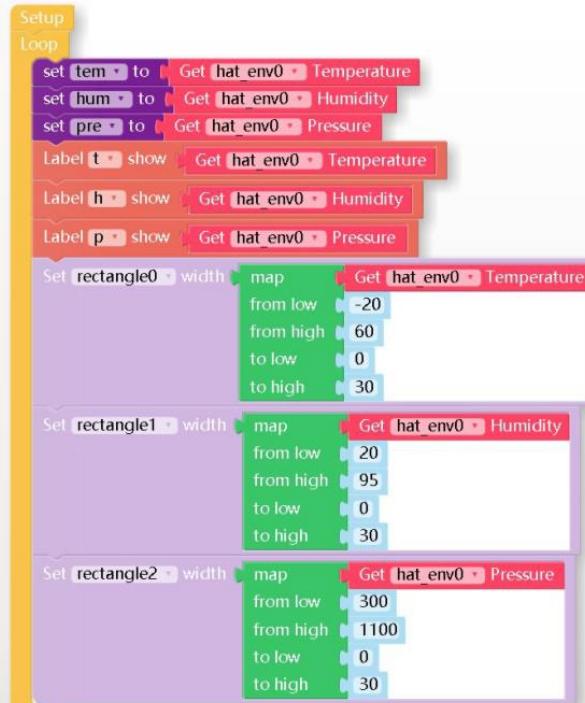
什么是数据映射

数据映射是指，将一组数据按一定的对应关系，建立起一组新的数据。我们可以用货币的转换来简单比喻数据映射的过程。假如你拥有100美元，你可以花这笔钱的范围就是在0~100美元；当你把100美元兑换了人民币，你可以花这笔钱的范围就是在0~670人民币（假设1美元=6.7元人民币）这两种货币在价值的对应关系上就是一种数据映射。



以相同的方式，将湿度、大气压值进行数据映射，并将其作为宽度值，应用到相应的矩形宽度控制块中。

完成程序编辑后，插接ENV-HAT模块，运行程序，查看效果吧。



APPENDIX:

FIRMWARE UPGRADE

APPENDIX
BEFORE WE START
附录

WIFI CONFIGURATION

LEARN MORE:

FIRMWARE UPGRADE

M5StickC

M5StickC



更新固件
Firmware Upgrade



M5StickC
了解更多
Learn more



配置WIFI
WIFI Configuration



附录
Appendix

M5StickC

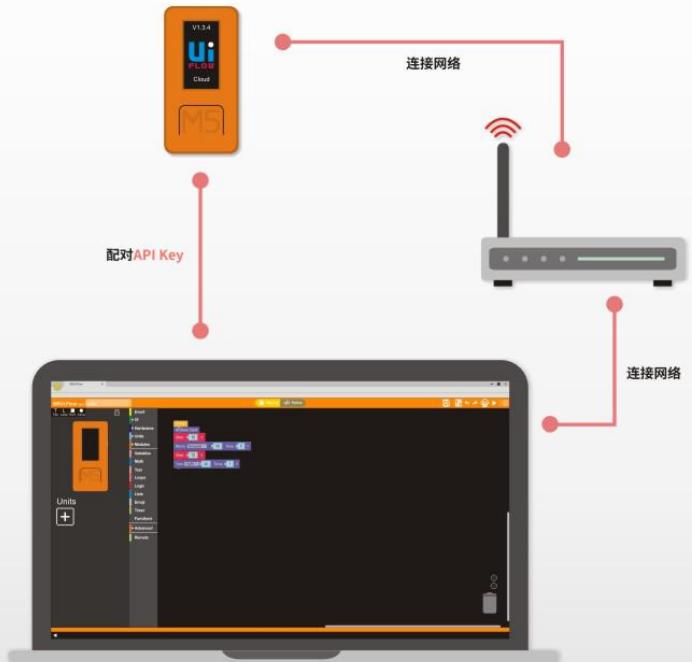
StickC

M5StickC

M5StickC

配置WIFI

除了支持本地USB连接编程，UIFlow还提供了web线上版本，与桌面IDE版本不同的地方在于，web版本支持程序的远程推送，无需数据线连接也能进行编程。
web UIFlow与M5StickC是两个独立的单元，在使用UIFlow进行编程前，我们需要将M5StickC通过WIFI连接网络，并配对API Key，以建立起它们之间的连接。



配置WIFI网络

- 长按电源键约2秒进行开机，开机后在屏幕出现Logo时，按下Home键进入功能选择菜单



- 按下机身右侧按键切换选择至Setup
- 按下Home键确定

04 选择Cloud Mode



该选项为WIFI信息管理，能够增加、删除已经保存的WIFI连接记录。

小贴士

在使用USB mode后，如需使用网络编程模式，需要在功能菜单中选择Cloud mode进行模式切换。

- 选择Cloud Mode后，屏幕上会显示一个WIFI热点的名称





05 使用手机WIFI，连接这个热点

06 打开手机浏览器直接访问192.168.4.1这个地址，进入WIFI配置页面



07 选择要连接的WIFI，然后填写密码，点击连接，配置成功将会自动重启

配对API Key

01 配置完成后将自动重启，进入网络编程模式，留意屏幕上的网络指示图标，当图标由红色变成绿色时，则表示已经成功连接上了网络



02 使用电脑或平板打开浏览器，访问flow.m5stack.com，进入UIFlow，点击右上角的菜单中的设置按钮



03 选择编程设备为M5StickC

04 在API Key选项中输入对应设备屏幕上显示的API Key，点击OK保存，等待提示连接成功，然后开始编程吧！

更新固件

为了持续优化体验与增加更多强大的功能，我们会不断的推出新的固件版本，M5StickC的固件版本与UIFlow的版本是同步更新的，你可以留意M5StickC开机时，主菜单页面上方的版本号，与UIFlow页面左上角的版本号是否一致，当你发现UIFlow的版本超前于M5StickC的版本，则表示我们推出了新的固件版本，请及时更新。UIFlow-Desktop-IDE集成了固件烧录功能，所以你无需下载其他的烧录工具就能进行固件的更新。



提示：M5StickC支持Win10/Linux/MAC操作系统免驱动连接，其余操作系统在进行固件更新或使用USB编程模式前，需要自行安装FTDI驱动程序。



一、COM

选择M5StickC连接电脑时所使用的端口，如果你并不知道使用的是哪一个端口，你可以尝试断开数据线连接，然后重新连接电脑，前后比较，新出现的端口则是M5StickC使用的端口，或是使用电脑自带的设备管理器查看端口使用情况。

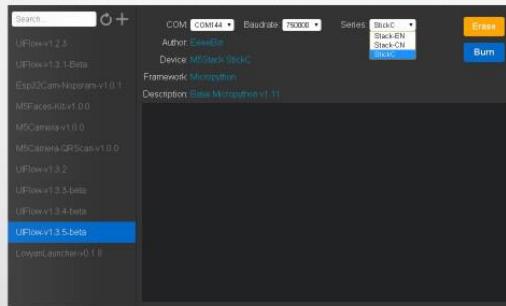


二、Baud

波特率可以简单的理解为烧录固件的速率，当我们为M5StickC烧录固件时建议选择“115200”或“750000”。

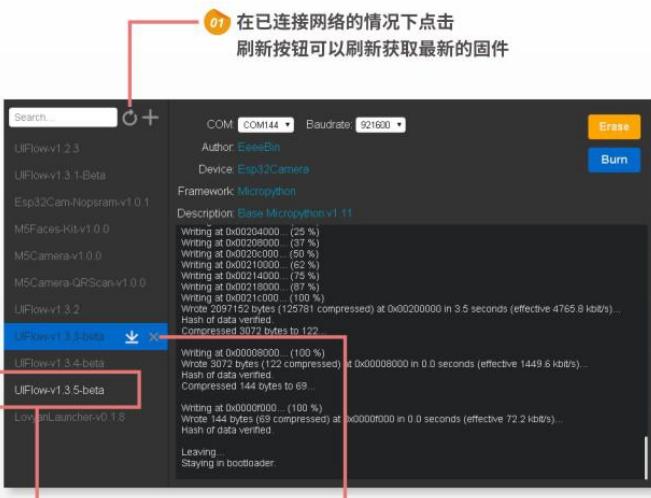
三、Series

选择烧录设备的类型，在选择固件后，我们需要将该选项设置为M5StickC。



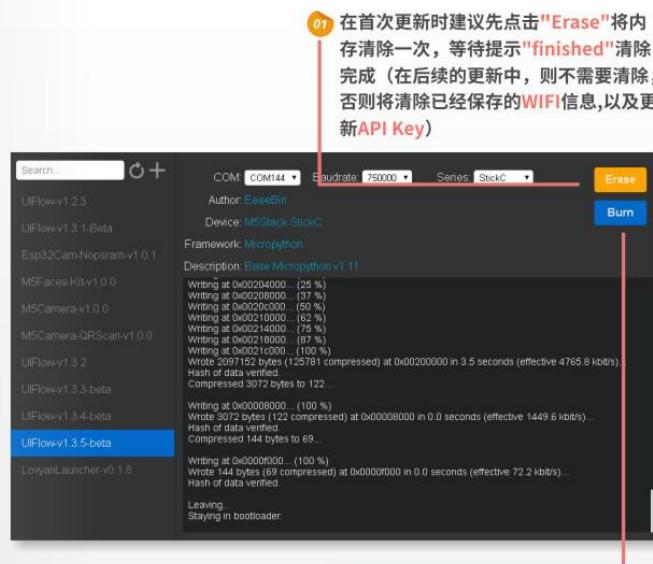
四、固件版本

选择要更新的固件版本，除了最新版本外，还提供了几个先前版本选择，不同的设备需要烧录不同的固件，部分设备也可以兼容多种固件。使用UIFlow作为编程器的则需要烧录“UIFlow”系列的固件。



五、开始烧录

配置好烧录参数，就可以开始进行烧录操作了。



M5StickC

了解更多

- 访问官方网址<https://m5stack.com> 获得更多有关M5的资讯与视频。
- 我们希望你在本书中，能够充分掌握M5StickC与UIFlow编程器的基本使用，在使用M5StickC学习编程的同时收获快乐，并充分发挥你的创意，将M5StickC运用到实际项目中，搭建出自己的物联网应用。
- 获取更多的功能模块与M5StickC配件，来拓展你的项目功能。

官网  速卖通  淘宝 

