

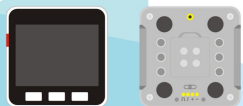
www.M5Stack.com

M5GO 编程入门 手册



专为中小學生编程设计的创客教育利器

LET'S BUILD OUR FIRST IoT APPLICATION IN 60 SECONDS
深圳市明线信息科技有限公司



深圳市明线信息科技有限公司
Shenzhen Mingzhan Information Technology Co.,Ltd
深圳市南山区西丽桃源街道珠光北路88号明线科技园1栋106
邮箱: Support@m5stack.com





M5Stack 致力于提供全方位基于ESP32平台的便捷式开发套件，硬件提供灵活可堆叠的扩展功能模块。软件提供便捷低门槛的图形化blockly语言的软件平台及教程。产品基于Micropython的编程语言并且兼容Arduino开发环境。涵盖了物联网应用、工程测量、机械控制、STEM教育等领域。

M5Stack



M5STACK

从入门到高手
M5Stack

从陌生到喜爱
M5Stack

从相识到需要
M5Stack



给未来的创造者





Contents

目录

开始之前

准备工作

UIFlow编辑器	13
UIFlow介绍	15

课程

显示文字及图形	23
Emoji动画与游戏	29
超级音乐家	35
迷你灯光秀	41
感应报警器	45
植物管家	49
智能气象站	53
DIY手表	55
加速度的应用	61
旋钮控制	67
交通红绿灯	72
光线感应与函数	77
电子通讯录	80
红外遥控	85
游戏工程师	91
远程控制	99

附录

配置WiFi	105
更新固件	109
了解更多	113

BEFORE WE START:

HARDWARE

CONSTRUCTION

UNIT INTRODUCTION

BEFORE WE START

BRING IN TO USE

「开始之前」
BEFORE WE START

M5GO

M5GO

GO

5GO



开始之前
Before We Start



硬件构造
Hardware Construction



UNIT介绍
Unit Introduction



开始使用
Bring In To Use

M5G

M5GO

M5GO

M5GO

开始之前

一、M5GO是什么？

M5GO是一个可编程的控制器,采用芯片ESP32、提供WIFI、蓝牙模块、MPU9250九轴姿态模块、扬声器、RGB bar等功能硬件。

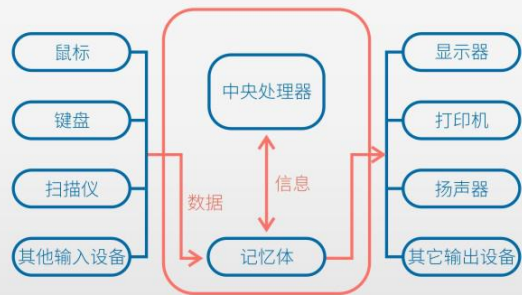
提供三个按键以及三个拓展模块接口,无论是简单开关控制,还是复杂的高阶功能, M5GO都能为你的编程学习提供强大且完善的硬件基础。

二、M5GO是一台微型电脑



三、什么是电脑？

电脑由四个主要组件组成



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。

四、M5GO具有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	例如RGB灯、LCD屏幕、扬声器

硬件构造

PORT B
端口B支持ADC输入，检测模拟量输入的Unit需要连接端口B

PORT C
端口C除了能够实现基本输入输出功能以外还支持串口通信

LCD显示屏
一块分辨率为320x240的彩色LCD显示屏，可以用它显示文本，图片等内容

物理按键
提供了三个物件按键接口A,B,C用作输入以及人机交互

PORT A
端口A能够实现基本的输入输出功能，使用I2C协议的Unit需要连接端口A

TYPE-C充电/数据接口
可以通过Type-C接口进行串口通信以及数据传输

电源/复位按键
单击 = 开机/复位
双击 = 关机




GROVE PORT	CABLE
A I2C	GND 5V SDA SCL
B I/O	GND 5V G26 G36 (In/Out Input)
C UART	GND 5V TXD RXD

LED Bar
机身两侧带有LED灯条，其内部为10个独立的可控LED灯，左右各5个

磁铁x4
可以把M5GO吸附在一些金属物体表面上用作固定

1W 扬声器
驱动发出声音，用做制作音乐以及提示音

TF卡拓展插槽
用作内存的拓展，最大支持拓展16G的TF卡

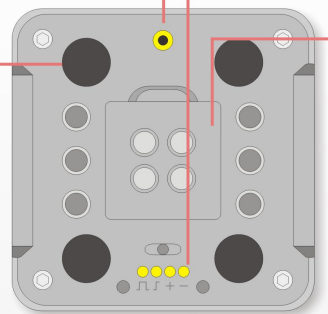
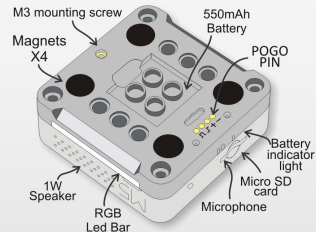
M3螺丝孔
通过M3螺丝进行结构固定

POGO-PIN
连接充电底座即可进行充电

550mAh电池
脱离电源线束缚，提供一段时间的续航

充电指示灯
在充电时会进行闪烁提示，当充电完成则常亮

麦克风
用作环境声音模拟量的输入与各种声控应用场景

开始使用

一、开关机、复位

"单击"机身左侧红色的电源键为"开机",运行中再次按下为"复位",快速"双击"电源键为"关机"



二、

当开机后, 首先会出现一个主菜单界面, 菜单提供了三个的功能选项来进入不同的功能页面。

编程模式

选择"PROGRAM", 将进入可编程模式, 等待程序推送 (编程模式提供了USB, internet 两种模式)

程序库

选择"APP.LIST"将进入程序库列表, 程序库中存放着已经下载的程序, 以及演示程序 "M5GO", 使用按键可以上下移动光标或选择程序运行, 长按右键还可以删除程序 (系统会记录选择的程序, 并在之后开机默认运行)

配置

选择"SETUP"将进入连接设置列表, 在该选项可以选择M5GO的两种连接模式, USB 连接或网络连接, 使用按键可以上下移动光标选择连接方式 (详细配置WIFI方法, 请查看附录)

小贴士

首次开机, 没有做任何的操作时, 默认状态下系统会进入"M5GO"演示程序, 演示程序会向你展示一系列有趣的功能, 能够更好的启发你编程的思维, 给我们更多的灵感去编程更多的功能。

三、Demo演示程序



Demo1 开关机

开关机及复位操作提示,单击为开机与复位, 快速双击为关机。

Demo2 扬声器

为你展示M5GO的扬声器, 不需要外接其他硬件, 按下中间按钮使M5GO播放一小段音乐。

Demo3 麦克风

尝试向着麦克风说话, 它将采集你的声音信号, 看看音频线条是如何随着你的声音变化的。

Demo4 重力感应器

使用M5GO的陀螺仪控制小球, 试着以各种角度倾斜M5GO, 看看屏幕上的小球是怎么运动的。

Demo5 表情灯

左右两侧红色与蓝色的呼吸灯, 像警车灯一样闪烁。

Demo6 扩展端口

提供了三个拓展端口, 可以将对应接口颜色的Unit通过GROVE线连接到M5GO上进行功能拓展。



Demo7 环境监测

连接ENV Unit到A端口,就能获取到当前环境的温度,湿度及大气压,尝试对着传感器呼气,看看温度与湿度数值有什么变化。



Demo8 人体感应

连接PIR Unit到B端口,当模块感应到人体时,屏幕上的红色圆形就会变成红色提示,静置一段时间后恢复灰色。



Demo9 RGB彩灯

连接RGB Unit到B端口,在该程序中M5GO就能够驱动模块上的LED灯点亮,如果你有多个RGB灯还可以尝试将它们串联起来,注意:RGB Unit需要连接正确的一端才能点亮。



Demo10 红外遥控

连接IR Unit到B端口, M5GO就能够通过IR Unit发射与接收红外信息,按下中间按键将执行发射,两个M5GO可以通过对准IR Unit,远程控制点亮RGB。



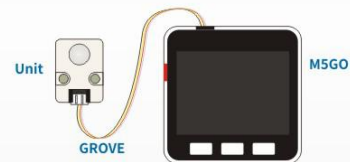
Demo11 角度传感器

连接Angle Unit到B端口,旋转Unit上的旋钮,能够实现灯光亮度的控制。

Unit介绍

一、什么是Unit?

Unit指的是功能拓展单元,在M5GO的体系中有着各种各样的Unit,不同的Unit拥有着不同功能;将Unit通过GROVE线连接至M5GO上对应的端口,通过编辑程序进行使用,能够实现更多丰富的功能。

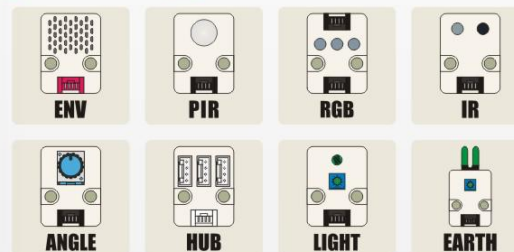


二、Unit的分类

Unit的分类,主要可以分为两大类,输入型与输出型。

输入型的Unit主要以采集数据的形式,将外界的一些环境信息以数据的形式传输给M5GO。

输出型的Unit主要以驱动的形式,以M5GO为控制核心,驱动外部连接的一些硬件。



三、如何连接Unit

M5GO有三个标签为A,B,C的端口,分别标有不同的颜色,这使得你更容易的知道使用哪个端口,例如,带有红色端口的Unit接到M5GO上的红色端口。

连接器上这些突起的脊必须对应都端口上槽。



PREPARATORY WORK:

INTRODUCTION

UIFLOW PROGRAMME

PROGRAMMING

STRUCTURE:

ABRIR AND GUARDAR

「准备工作」
PREPARATORY WORK

M5GO

M5GO



UIFlow布局
Uiflow Introduction



UIFlow编程器
Uiflow Programme



保存与打开
Abrir and Guardar



程序结构
Programming Structure

M5GO

M5GO

M5GO
M5GO

UIFlow 编辑器

UIFlow 是一个能够支持Blockly 与 Python 编程的一个编辑器

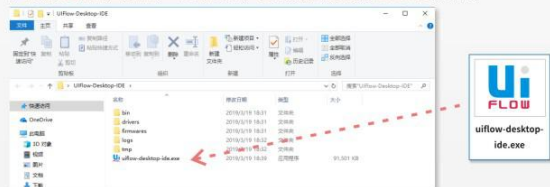
01 下载压缩包

访问 <https://m5stack.com/>, 点击 "SUPPORT", 选择 "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。



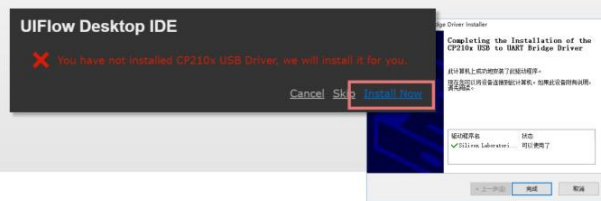
02 解压压缩包

将下载的 IDE 压缩包解压到一个文件夹中, 进行存放, 双击打开 UIFlow 应用程序。



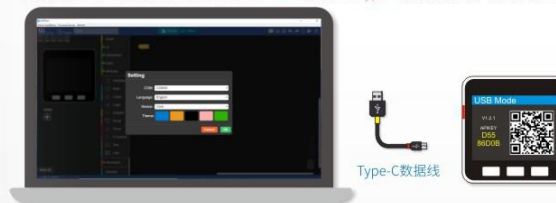
03 安装 CP210X 驱动

当应用程序监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, 直到完成安装。



04 连接设备

安装完成后将自动进入 UIFlow 编辑器, 将 MSGO 设备通过 Type-C 数据线连接到电脑, 选择使用的 COM 端口。



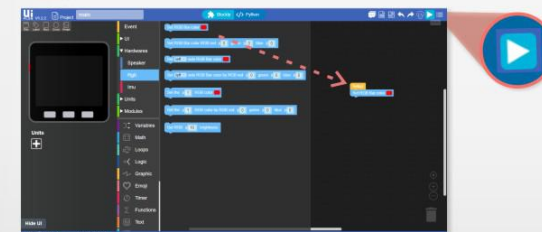
05 进入 USB 编程模式

单击电源键, 将 MSGO 开机, 进入菜单后选择 "SETUP", 进入配置页面, 选择 "USB mode"。



06 运行第一个程序

点击 "Hardware" 选项中的 "RGB" 功能, 拖动 "RGB bar" 连接到 "Setup" 上, 点击运行程序, 点亮 MSGO 两侧的 RGB 灯。



07 内容补充

MSGO 搭载了一个与 UIFlow 相对应的操作系统, 为了获取更多强大的功能, 你需要将固件版本更新至最新版本, 更新固件的具体操作请查看附录 "更新固件"。

UIFlow布局

Blockly </> Python
 随时进行Blockly转换Python代码，逐渐从图形化编程进阶到代码编程

项目命名名称
 在输入框中，可以给你的程序项目命名，方便你再次打开时寻找，或是说明程序功能

功能菜单栏
 提供一些功能操作，与相关链接，如程序的运行与下载，本地保存与打开，撤销等基本操作，以及帮助文档论坛链接

UI模拟器
 编辑文本与图像等元素的显示，在程序运行后它们将会在M5GO屏幕上对应位置显示

程序块列表
 搭建程序所需要的各种程序块

UI隐藏/显示
 将UI模拟器及其他左侧部分隐藏，以提供更大的编程区域面积

Units模块拓展
 添加各种各样的Unit功能模块

编程区域
 编程区域是构建程序的地方

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。

Event
 程序结构部分，按键与无限循环

Hardwares
 基础的一些硬件控制程序，如RGB, Speaker, 以及陀螺仪IMU

Units
 进行Units模块添加后，在该选项下会出现相对应的程序块

Math
 一系列数学运算程序，随机数以及数据类型处理

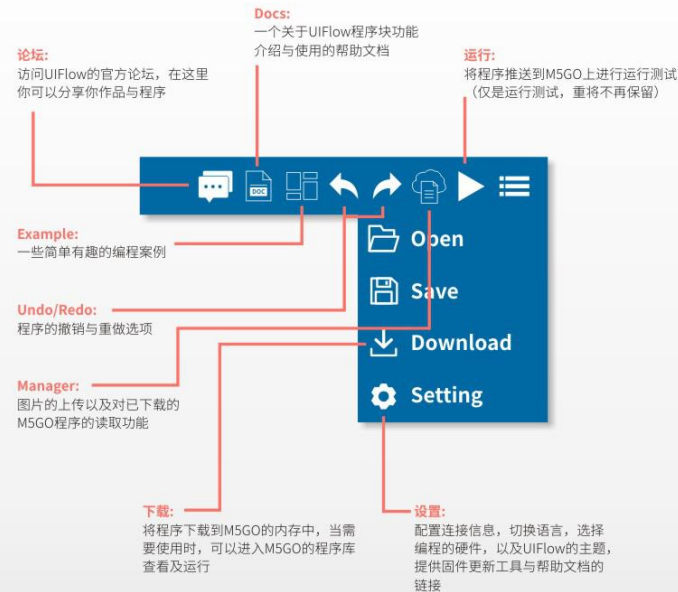
Logic
 逻辑判断，与逻辑运算程序

Advanced
 高级功能程序，涉及网络通讯及硬件的引脚控制与协议

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。



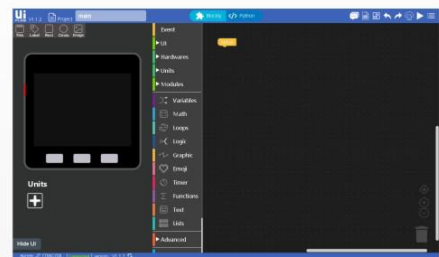
UIFlow操作

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

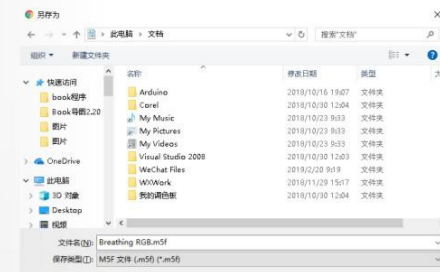
在以后的学习中，我们可能会编程一些功能复杂的程序，这使得我们并不能一次性将程序编写完成。当我们将工作进行了一部分的时候，我们可以将程序先保存到电脑本地，等到下次继续工作的时候再次打开；或是在工作途中时不时的进行一次保存用作备份，这些时候我们都会使用到保存与打开功能。点击“Save”，选择要保存到的目录，点击保存。当需要打开程序时，则点击“Open”找到已经保存的，以“.m5f”为后缀的程序文件，点击打开，UIFlow就会将文件中的程序重新加载到编程区域。



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



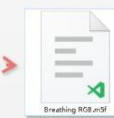
02 编程完你的程序后，点击Save



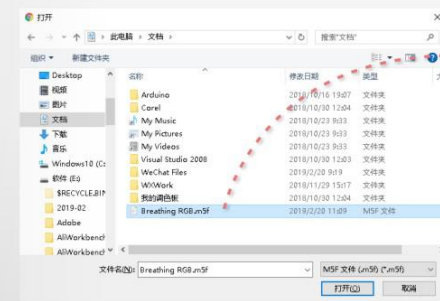
03 将程序另存为后缀为“.m5f”程序文件



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

INDUCTION ALARM

TRAFFIC LIGHT

PLANT STEWARD:

GAME ENGINEER

CURRICULUM

「课程」

M5GO

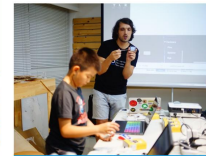
M5GO



感应报警器
Induction Alarm



超级音乐家
Super Musician



游戏工程师
Game Engineer



交通红绿灯
Traffic Light



M5GO

M5GO

M5GO

M5GO

1 显示文字及图片

简介: 使用UIFlow中的UI模拟器功能, 编程实现文字与图形的显示。

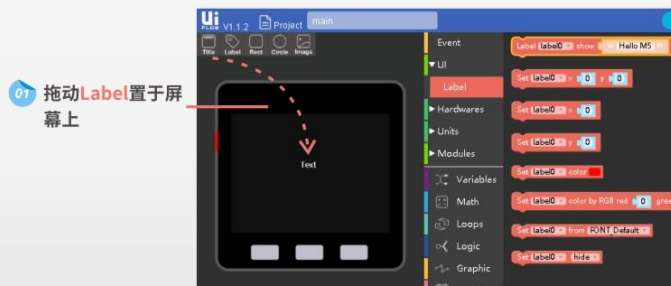
目标: 了解程序开发流程, 并在MSGO上运行你的第一个程序, 最后制作一张电子个人名片。

1-1 程序员的第一个程序

"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, 它将使你更有信心学习更高级更复杂的程序。



所以我们将学习如何编程实现, 在MSGO的屏幕上显示"Hello World"。



点击程序块菜单中的UI选项, 选择Label功能, 选中"Label show"块, 拖动到编程区域, 并连接到"Setup"块修改显示内容为"Hello World"。

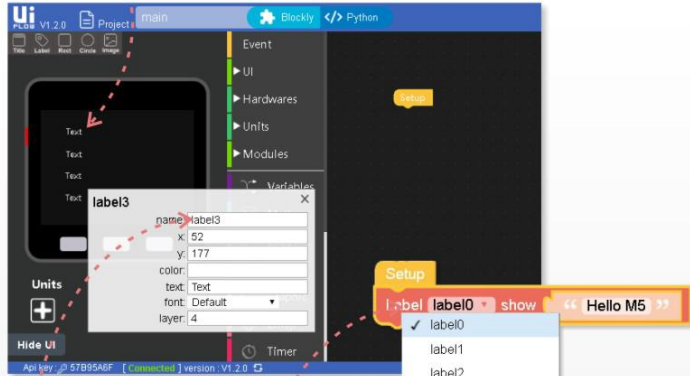


这时候, 就完成了你的第一个程序, 接下来就是要让这个程序运行起来, 在运行程序前将MSGO通过数据线连接到电脑上, 并设置进入USB编程模式。



1-2 制作电子个人名片

1、我们可以尝试添加多个标签，然后显示不同的内容，输入名字，性别，爱好等信息制作一个电子名片。



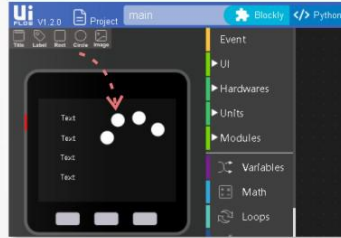
注意：当添加多个标签时，需要点击“Label show”块上的标签选项，选择为不同的标签，否则只有一个标签有显示内容。



2、右键点击程序，选择“Duplicate”，复制多个“Label show”块。



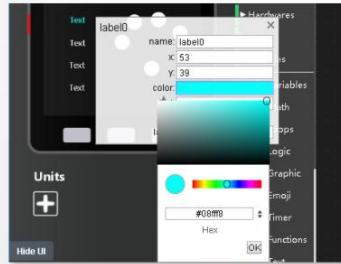
3、修改对应的标签，将内容修改成名字以及其他个人信息。



4、添加几何图形

可以添加一些几何图形到“名片”中，点击UI模拟器上方的元素选项，选择圆形或是矩形拖动放置到屏幕上。

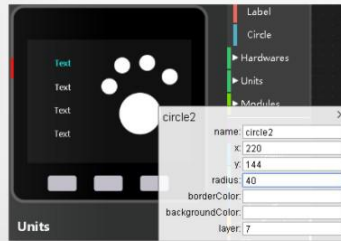
当我们运行程序时，这些几何图形就会在屏幕上对应的位置显示。



5、为电子名片添加色彩

我们可以自定义文字或是图形的颜色，以及其他参数，单击已经添加到屏幕上的标签或图形，在弹出属性框中，我们可以修改这个元素的初始值。

例如：点击标签Label,选择属性框中的“Color”属性，修改它的颜色为蓝色，这样文字的颜色就被修改成了蓝色。



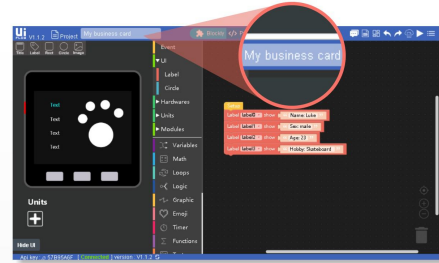
6、修改图形参数

我们可以修改图形的颜色，长宽，半径等参数，还可以更改文本的字体，让我们设计一个属于自己的电子名片吧。



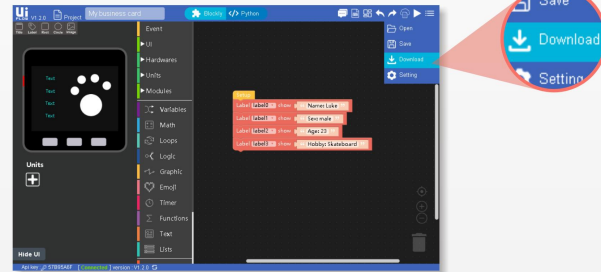
7、项目名称

制作完成后，可以点击UIFlow页面左上方的项目名称输入框，给这个程序取一个名字，例如“My business card”。



8、下载程序

点击右上角菜单中的下载按钮，将程序下载到M5GO的程序库中，这样当我们下次想要使用这个程序的时候，只需要进入程序库中找到它，并点击运行就可以了。

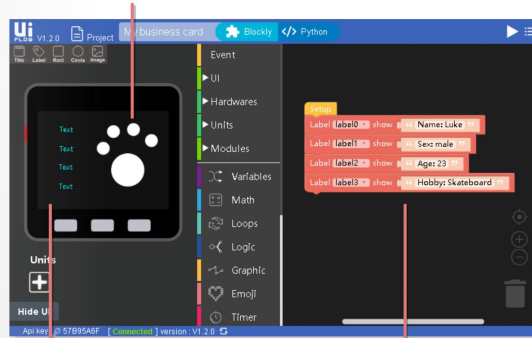


9、程序库

开机出现主菜单后，选择“APPLIST”将进入程序库列表，程序库中存放着已经下载的程序，点击“RUN”即可运行。



"circle"是我们添加在屏幕上的一些圆形，不同的序号后缀将它们进行区分



屏幕的背景颜色，你可以尝试改变它的数值为0xff0000,并运行程序看看背景是如何改变的

"Label"是我们的标签显示，对应不同的标签序号，我们所编辑的信息，也出现在了程序中



点击Python代码切换，查看我们目前程序的代码内容

2 Emoji动画与游戏

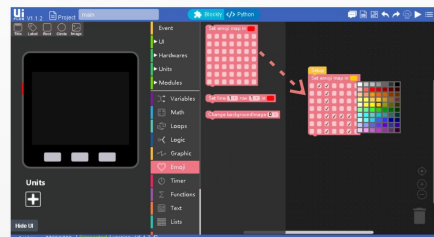
简介: 使用Emoji块与延时块编辑一个小动画, 使用Emoji与按键块制作一个剪刀石头布小游戏

目标: 学习延时块Wait, 与按键控制的使用

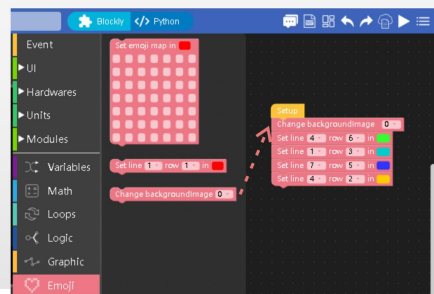
1-1 制作Emoji动画

我们将学习如何编程实现, 在M5GO的屏幕上显示Emoji像素图, 然后尝试将几张图连续滚动播放, 形成一个小动画首先, 我们先尝试使用一下Emoji块。

点击程序块菜单中的Emoji选项, 将“Emoji方阵”块拖动到编程区域, 并连接到“Setup”, 单击Emoji方阵块上的小方格, 可以激活像素点, 再次点击可以取消像素点, 单击块上的颜色选项, 可以修改显示时像素点的颜色尝试勾选几个小方格, 然后点击运行程序按钮, 查看效果。

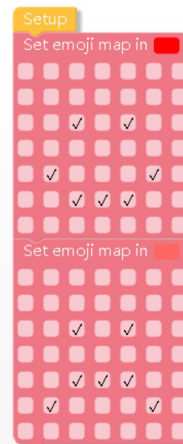


假如你想要在同一张Emoji图像中有不同颜色的像素点, 你可以选择“Emoji行列”块, 根据选择的行和列, 指定的激活某一个像素点并设置颜色, 使用“Emoji背景”块, 修改显示的背景序号, 还可以切换Emoji图像的背景图片。



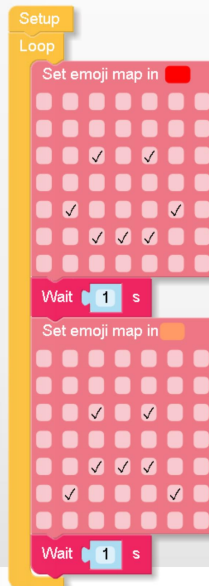
学习了Emoji图像的三种块, 接下来, 我们要学习如何制作动画了。在实际生活我们所看到的动画, 其实都是由一张张图片顺序切换形成的, 越流畅的动画, 其中包含的图片就越多张, 单位时间内所显示的图片张数, 就是我们所说的“帧数”。

于是我们知道, 如果我们要制作动画, 我们最少需要两张Emoji图, 我们尝试将添加两个Emoji方阵到程序中, 就像右图一样将它们连接, 看上去并没有什么, 可当我们实际运行的时候却发现, 好像只是显示了放在后面的Emoji图像, 这是为什么呢?



一、产生原因

其实两张图片都被正常的显示了, 但是程序运行的速度实在是太快了, 从第一张Emoji图像运行到第二张的时间很短很短, 以致于我们只能看到第二张Emoji图像。



二、解决方法

我们可以在两张Emoji图片的中间添加一个延时块Wait, 使第一张图像间隔一段时间再切换到第二张图像点击程序菜单中的Timer选项, 将Wait延时块拖动到编程区域, 放置在两张Emoji图像中间作间隔。

我们发现动画实现了但是只动作了一次, 如果我们想要使动画不断的运行, 这时候我们就需要用到无限循环Loop。

点击程序菜单中的Event选项, 将Loop无限循环块加入到程序中, 把需要循环的程序放入Loop, 且再添加一个延时用来间隔第二个图像切换到第一个图像。

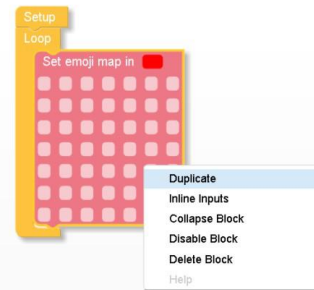
这样一个简单的动画就制作完成了, 你可以发挥想象, 用更多的Emoji图像, 去制作一个长的动画, 或是描述一个故事。

1-2 烟花动画

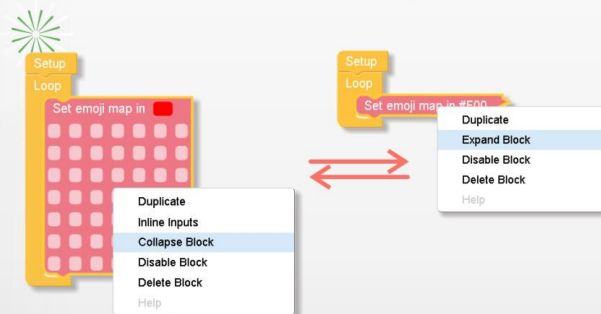
使用Emoji制作一个循环播放的烟花动画。



当有多个相同类似程序的时候，我们可以右键点击程序块选择“Duplicate”或左键快速双击进行复制将其复制出来的Emoji块，修改成其他的图案，像烟火一样逐渐展开。



当你认为程序太长了，我们可以右键点击程序块选择“Collapse Block”，将程序块进行收缩使得程序更加简洁，当需要查看程序的详细参数的时候，同样的方式右键点击程序块，选择“Expand Block”则可以再次展开。



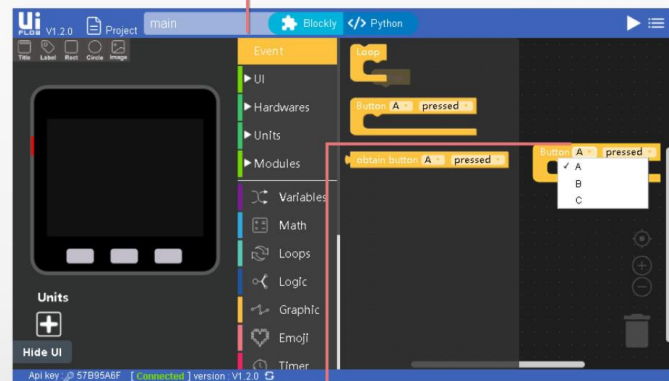
1-3 剪刀石头布游戏

我们除了使用Emoji图像制作动画以外，还可以用它去制作游戏，下面就来学习如何配合按键控制，编程实现一个通过按下不同按键，显示不同Emoji图像的剪刀石头布游戏吧。

按下按键A显示“石头”，按下按键B显示“布”，按下按键C显示“剪刀”



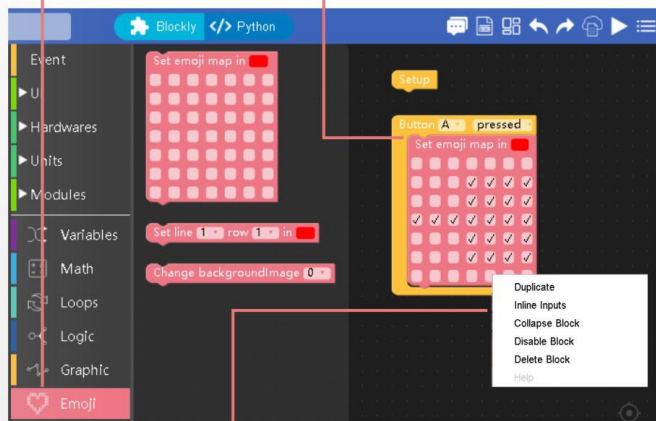
01 点击选择Event选项
Button按键块



02 点击Button按键块上的选项框，可以更改对应的控制按键A,B或C，当我们按下对应的按键时，就会执行Button按键块中所包含的程序

01 点击选择Emoji选项

02 添加Emoji方阵图案，绘制成表示“石头”的图案，我们可以用一个锤子表示



03 右键选择Button块，点击“Duplicate”，会一同复制Button块，以及它包含的程序



同样的方式绘制出“剪刀”，与“布”，并将修改它们所对应的按键。

运行程序，和小伙伴一块来一场剪刀石头布对决吧。



3 超级音乐家

简介：使用Speaker块去驱动M5GO发出声音，制作一段音乐，使用按键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。

1-1 制作个人音乐

我们将学习如何编程驱动M5GO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

一、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M5GO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。

Setup

Speaker.beep freq: 1600 duration: 200 ms

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。

Setup

play tone Low A for 1 beat

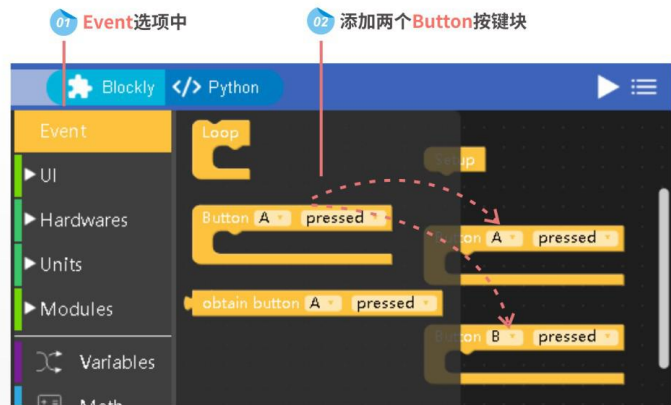
- ✓ Low A
- Low B
- Low C

Setup

play tone Low A for 1 beat

- ✓ 1
- 1/2
- 1/4

我们尝试编辑一个程序，来测试频率与音调的关系。



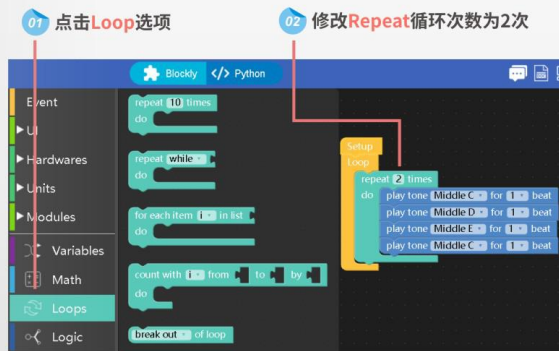
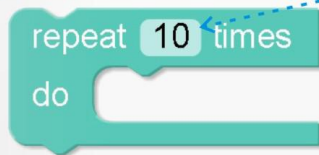
05 运行程序，分别按下按键A与按键B测试，你会发现，它们发出的声音是一样的

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"



例如下面的两个程序，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。

The image shows two code snippets side-by-side. The left snippet is a long, linear sequence of blocks: 'play tone' (Middle C, D, E, C, D, E, C, D, E, C, G, A, G, F, E, C), 'wait' (0.2 s), 'play tone' (Middle E, F, G), 'wait' (0.2 s), 'play tone' (Middle E, F, G), 'wait' (0.2 s), 'play tone' (Middle G, A, G, F, E, C), 'wait' (0.2 s), 'play tone' (Middle G, A, G, F, E, C), 'wait' (0.2 s), 'play tone' (Middle G, A, G, F, E, C), 'wait' (0.2 s), 'play tone' (Middle C, Low G, Middle C), 'wait' (0.2 s), 'play tone' (Middle C, Low G, Middle C).

The right snippet is more concise, using 'repeat' blocks to group identical sequences of 'play tone' and 'wait' blocks. It uses 'repeat 2 times' for the first three groups, 'repeat 2 times' for the fourth group, and 'repeat 2 times' for the fifth group.

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。



4 迷你灯光秀

简介：控制M5GO上的RGB bar发出各种颜色的光芒，并通过延时功能与次数循环制作一场迷你灯光秀。
目标：学习大自然"三原色"原理，加深循环与延时功能的概念。

1-1 点亮与熄灭RGB灯

01 点击程序块菜单中的**Hardware**s选择**RGB**选项

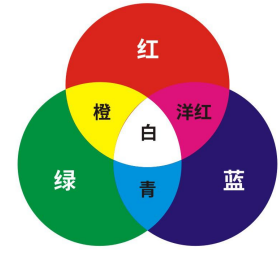
02 拖动"Set RGB bar"连接到"Setup"上

03 单击块上的颜色选项可以选择不同的灯光颜色

04 点击运行程序

除了直接选择颜色以外，你也可以使用可以输入"RGB值"的控制块进行颜色设定

问题：为什么RGB灯要做RGB灯？RGB灯为什么能发出各种颜色的光芒？
解答："R","G","B"分别表示的是红(Red)，绿(Green)，蓝(Blue)三种颜色，在大自然中，几乎所有的颜色都可以由这三种颜色通过不同的比例混合而成，这也就是我们所说的"光的三原色"。



RGB灯就是通过三原色原理，将三色灯光混合形成各种各样的颜色，三种色的色值范围为"0-255"

问题：如果我们将颜色设置为黑色？RGB灯会点亮吗？黑色的RGB值为多少？

解答：是不会点亮的，黑色会吸收所有的可见光，但没有反射，黑色的RGB值为 (0, 0, 0)。

除了同时设置M5GO左右两侧颜色的"Set RGB bar"以外，我们还可以实现更加精确的控制，像是控制左或右一侧的颜色"Set left side"，或是控制单个RGB灯的颜色"Set the N RGB"，M5GO的RGB bar共有10个独立的RGB灯，左右两侧各五个，由机身右侧灯条开始以顺时针方向命名序号"1-10"，还可以"Set RGB N brightness"设置灯的亮度,理论の設定值范围为"0-255"。

1-2 警车灯光

左右灯光切换的同时，驱动Speaker发出警车警报声。

```

Setup
Loop
  Set left side RGB Bar color [Red]
  Set right side RGB Bar color [Black]
  play tone High C# for 1 beat
  Set left side RGB Bar color [Black]
  Set right side RGB Bar color [Blue]
  play tone Middle G# for 1 beat

```

一、制作迷你灯光秀

使用RGB不同的控制方式，结合已经学习过的延时与循环程序制作一个炫酷的灯光秀，其中包含不同的特效且自动切换，循环运行。

第一种效果

将RGB bar左右两侧同时点亮，经过0.2秒的延时后熄灭，再次进行延时0.2秒，使用了次数循环，这段程序会执行两次。

```

Setup
Loop
  repeat 2 times
  do
    Set RGB Bar color [Orange]
    Wait 0.2 s
    Set RGB Bar color [Black]
    Wait 0.2 s
  repeat 2 times
  do
    Set left side RGB Bar color [Yellow]
    Set right side RGB Bar color [Black]
    Wait 0.2 s
    Set left side RGB Bar color [Black]
    Set right side RGB Bar color [Purple]
    Wait 0.2 s

```

第二种效果

将RGB bar的“一侧”点亮，另一侧同时熄灭，经过0.2秒的延时后，交换状态，使得每次只有一侧点亮，使用了次数循环，这段程序会执行两次，将整个程序放入一个无限循环Loop中，使它不断的循环运行。



二、功能拓展

编程实现当按下不同按键时，执行不同灯光效果，经过一段时间后熄灭。

```

Blockly Python
Event
  UI
  Hardware
    Speaker
    Rgb
    IMU
  Units
  Modules
    Variables
    Math
    Loops
    Logic
    Graphic

Setup
Loop
  Set RGB Bar color [Black]

Button A pressed
  repeat 10 times
  do
    Set RGB Bar color [Green]
    Wait 0.2 s
    Set RGB Bar color [Black]
    Wait 0.2 s

Button B pressed
  repeat 5 times
  do
    Set left side RGB Bar color [Red]
    Set right side RGB Bar color [Black]
    play tone High C# for 1 beat
    Set left side RGB Bar color [Black]
    Set right side RGB Bar color [Blue]
    play tone Middle G# for 1 beat

Button C pressed
  Set the 1 RGB color [Red]
  Set the 2 RGB color [Orange]
  Set the 3 RGB color [Yellow]
  Set the 4 RGB color [Green]
  Set the 5 RGB color [Cyan]
  Set the 6 RGB color [Blue]
  Set the 7 RGB color [Purple]
  Set the 8 RGB color [Pink]
  Set the 9 RGB color [Magenta]
  Set the 10 RGB color [Black]
  Wait 1 s

```

执行10次的绿色灯光闪烁

执行5次的警车灯效果

将10个RGB灯设置为不同颜色

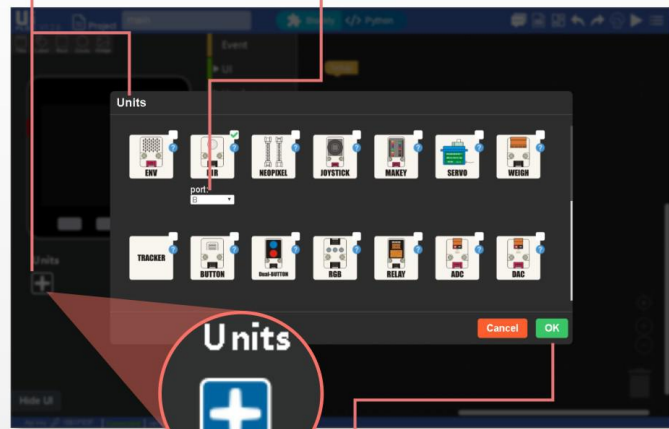
5 感应报警器

简介: 使用PIR人体感应Unit与M5GO编程制作一个感应报警器。

目标: 了解红外热释原理, 学习逻辑判断基础内容, 以及Unit拓展编程的方法。

一、Unit的添加

- 01 点击UIFlow页面左侧的Unit添加按钮, Unit菜单中提供了许多Unit功能模块选项
- 02 选择"PIR"人体感应模块 在模块的下方可以指定要使用的端口



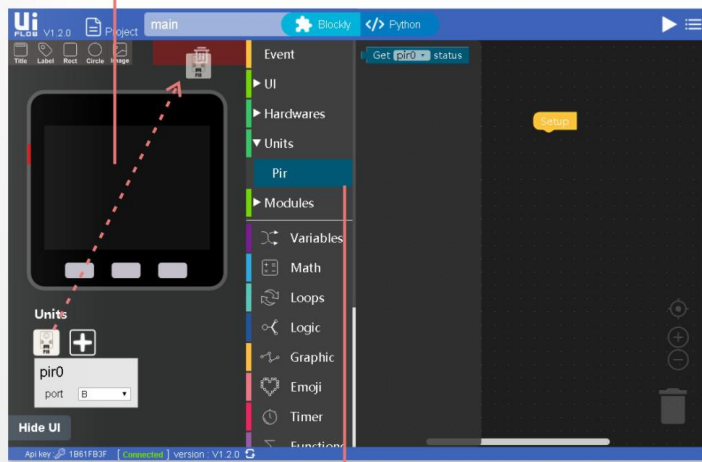
- 03 选择完成后, 点击OK保存, 就完成添加了

一些特殊的Unit还可以设定一些其他参数。

如:Neopixel灯条 可以设定控制的灯的总个数, 已经添加的Unit会显示在添加选项的旁边, 单击它还可以在弹出的属性框中更改使用端口或其他参数。

二、Unit的移除

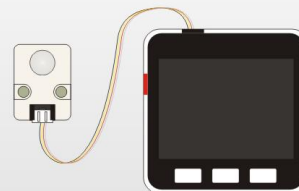
假如有错误添加的Unit, 或是已经添加的Unit我们不需要使用, 我们可以把它移除, 移除的方法也非常简单我们只需要点击已经添加的Unit, 这时候你会发现页面上方出现了一个“垃圾桶”, 将Unit拖动到垃圾桶中松开, 即可完成删除。



三、Unit的使用

当我们成功添加了一个Unit功能模块, 在UIFlow的程序块列表中的Units, 就会出现与它对应的程序块, 不同的Unit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Unit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Unit模块通过GROVE线, 提前连接到M5GO相应的端口上”, 完成连接后, 运行程序, 即可使用。



四、PIR模块测试

01 首先，我们在UIFlow中添加"PIR"Unit

02 点击PIR

03 PIR Unit提供了一个程序块"Get PIR"，我们并不知道它的实际功能，所以我们可以对它进行测试

04 从UI模拟器中添加一个标签Label

05 将"Get PIR"块放入到"Label show"块的显示内容中

06 将PIR Unit连接到MSGO对应端口上，我们这里选择的是"B端口"

07 点击运行

五、模块测试

当我们静置传感器且人体没有靠近时，MSGO屏幕上的标签显示0，当我们用手去靠近传感器时，标签显示为1

分析结论

当没有感应到人体时，PIR Unit的程序块"Get PIR"为数值"0"，当感应到人体时，数值为"1"。

问题：为什么PIR人体感应Unit可以检测人体？

解答：人体能力的热释放有一种特定波长的红外线，PIR人体感应Unit就是通过检测这种红外线变化来实现感应人体。

1-1 感应报警器

利用"Get PIR"根据是否感应到人体而表示不同数值这一功能，我们可以制作一个感应报警器。编程实现，当"Get PIR"为"1"时，驱动Speaker扬声器发出声音报警且点亮RGB灯，当"Get PIR"为"0"时，不发声且熄灭RGB灯，为了实现这样的一个功能，我们需要学习一个新的程序块"if"逻辑判断。

"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。



01 点击Logic选项

02 添加一个带有"else"的"if"块，我们需要编辑一个关系式来作为判断条件，所以要使用到关系式块

03 点击Math

04 常量数字块组合形成一个"Get PIR"="1"的式子，然后连接到if的右侧用作判断关系式

05 为条件成立与不成立两种情况添加执行功能，当条件成立时执行"Speaker beep freq"，且"Set RGB bar"为红色，当条件不成立时执行"Set RGB bar"为黑色

将PIR Unit硬件连接完成后，点击运行程序，来看看感应报警器是如何工作的吧。

6 植物管家

简介: 使用Earth土壤湿度检测模块，制作一个植物管家，实时监控土壤水分。
目标: 深入学习逻辑判断，接触逻辑运算，学习布尔值。

一、布尔值

什么是布尔值？还记得在上一节中我们学习了，逻辑知识中的if判断，以及根据需要建立了判断条件的关系式吗？关系式运算“成立”与“不成立”这两种结果的值，就称为布尔值，布尔值只有“True”（真）与“False”（假）两种值以数字的形式来表达，False=0，True=除了0以外的其他数字，布尔值(True或False)有助于程序进行一些“决策”通过判断，决定某些功能和程序的执行或是不执行，在Logic中也提供了布尔值的程序块，你可以将它用在逻辑判断中充当条件，或是进行逻辑运算。

1+1=3
False

2+3=5
True

二、逻辑运算

什么是逻辑运算？我们为什么要学习逻辑运算？

在实际编程中，如果要实现一些复杂的功能，简单的逻辑判断条件就不能满足我们的需求；逻辑运算在我们的生活中无处不在，比如我们可以举例在餐厅点一杯可乐来比喻这样的过程。

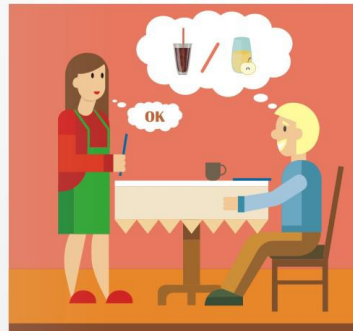


“我要一杯可乐，加冰”“可乐”+“加冰”，商家只有提供同时符合这两个要求的饮料的时候才是正确的，否则就是错误的，这就是逻辑运算中的“and”与运算。

and:
 要求进行运算的两个对象都是成立的，结果才是“True”，否则“False”。

“我要一杯可乐，如果没有可乐，那就换成果汁”，商家可以提供“可乐”或者“果汁”，只需符合其中一个，就是正确的，如果两者都不符合时就是错误的，这就是逻辑运算中的“or”或运算。

or:
 进行运算的两个对象只要有一个是成立的，结果就是“True”，否则“False”。



将一开始的符合条件，经过了否定变成了不符合，“将正确的变成不正确的”，“将不正确的变成正确”，这就是逻辑运算中的“not”非运算。

not:
 将运算对象反转，当对象为“True”，经过not的运算后为“False”，同理，“False”经过not的运算后为“True”。



三、Earth模块测试

Earth是一个能够检测土壤湿度的模块。



将Earth Unit连接到M5GO对应端口上，我们这里选择的是“B端口”。

当我们把Earth的两个引脚插入土壤中，“Get Analog Value”所连接的标签显示的是一个土壤湿度的模拟值。

“Get Digital Value”显示出来的为一个数字量0或1。

分析结论：土壤湿度的模拟值“Get Analog Value”能够较为具体形容湿度情况，“Get Digital Value”则是作用于检测土壤水分湿度的存在，根据需求我们要用到的程序块为“Get Analog Value”。

问题：为什么Earth Unit可以检测土壤湿度？

解答：Earth利用水的导电性原理，将引脚插入土壤后通过检测两个引脚之间的电势差，从而检测土壤的湿度。

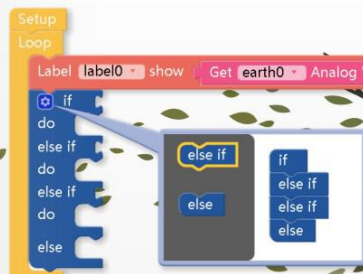
1-1 植物管家

为了实时监控植物土壤的水分，我们除了需要将湿度数据显示在屏幕上以外，还希望根据湿度的不同将土壤分为几个等级如“水分过多”“水分正常”“水分较少”“水分不足”。

问题：我们如何做才能使用湿度数值，作出四种不同情况的判断？

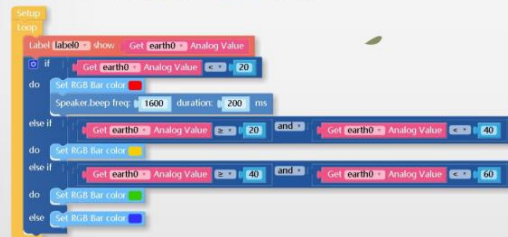
解答：使用逻辑判断，与逻辑运算，将湿度值设置四个范围。

首先，添加一个标签用来显示湿度值；添加“if”判断，并冲突的判断条件关系式，点击块上的齿轮修改逻辑判断结构，添加“else if”。



针对不同湿度值范围编辑相应的执行功能，如：

- (1) 当湿度值大于等于20时，点亮RGB bar为红色，且驱动Speaker发出声音提示。
- (2) 当湿度值大于等于20且小于40时，点亮RGB bar为黄色。
- (3) 当湿度值大于等于40且小于60时，点亮RGB bar为绿色。
- (4) 当湿度值大于等于60时，点亮RGB bar为蓝色。



将Earth Unit硬件连接完成后，点击运行程序，将Earth的引脚插入土壤中是如何工作的吧。

7 智能气象站

简介: 使用ENV环境温湿度大气压检测模块, 制作一个智能天气站, 实时监控环境温湿度且根据大气压判断所处的高度, 并执行不同的功能。

目标: 学习使用变量进行运算, 加深学习逻辑运算与逻辑判断知识, 多种功能的综合使用。

一、ENV模块测试

ENV是一个能够检测环境温湿度大气压的模块。



01 在UIFlow中添加"ENV"Unit

02 点击程序块列表中的Units选项, 找到ENV的程序块

03 添加三个标签Label, 三个程序块都通过"Label show"显示出来看看具体的内容



二、"温度" "湿度" "大气压"

经过测试我们发现这三个程序块分别提供了"温度""湿度""大气压"数据, 此时的显示的大气压数值, 表示我们所在位置高度的大气压。

问题: 什么是大气压?

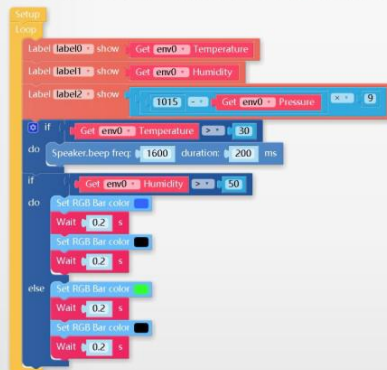
解答: 大气压可以简单理解为作用在单位面积上的大气压力, 也就是单位面积所受到的垂直空气柱压力, 气压的大小与温度, 高度等条件有关, 通常伴随着高度的增加而减小。

我们可以进行粗略的计算, 根据每上升9m, 大气压降低1hpa, 只要选定一个地面的参考点, 通过计算, 就可以知道我们当前高地的高度。

气压值的常用单位为"千帕"或是"百帕", ENV Unit为我们采集的气压值单位为hpa"百帕"。假设我们测试得出, 当前参考地面的大气压为"1015hpa", 那么基于这个参考值, 我们只需要对检测到的大气压值进行计算, 1015hpa减去检测值, 然后乘上9m, 就能得到我们基于参考地面上的高度。

1-1 智能气象站

- (1) 当温度大于30度时, 驱动Speaker发出声音提示。
- (2) 当湿度大于50时, 执行RGB灯闪烁蓝色灯光, 低于等于时闪烁绿色灯光。
- (3) 将"温度""湿度"以及当前气压值运算得到的"高度值"显示在屏幕上。
- (4) 参考点的气压值减去检测点的气压值, 乘以9得到当前位置的高度的估值。



将ENV Unit硬件连接完成后, 点击运行程序进行功能测试, 尝试向着ENV传感器呼气, 看看温度与湿度是怎么变化的? 当我们走到高的地方, 高度值是否也在跟着变化。

8 DIY手表

简介: 使用变量与逻辑判断以及按键编程, 用M5GO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) 我们可以简单的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

02 点击 "Create variable" 创建一个变量

03 输入创建变量的名称, 点击确定完成变量的创建

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。



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。

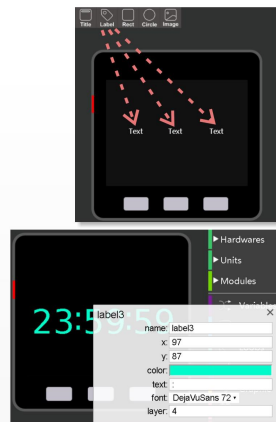
问题: 在进行手表程序编辑前, 你是否知道, 时分秒的概念。
 解答: 60秒=1分钟, 60分钟=1小时, 一天为24小时。

1-2 DIY手表

一、显示UI部分

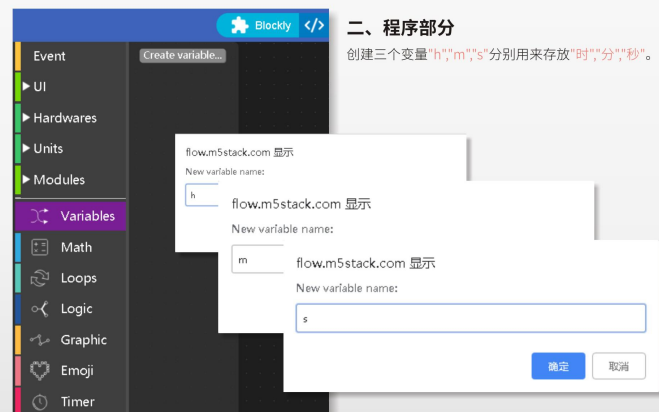
首先我们需要添加三个标签用作“时”“分”“秒”的显示。

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。

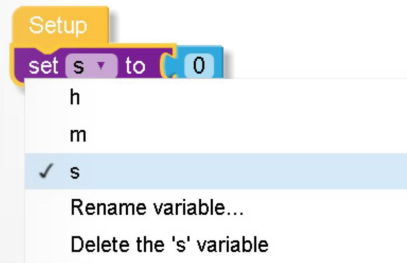


二、程序部分

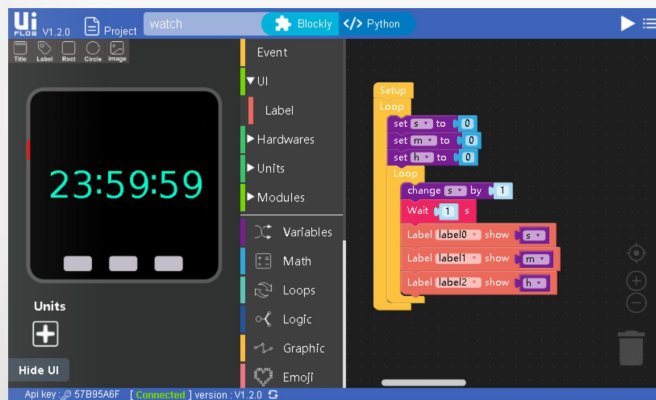
创建三个变量“h”“m”“s”分别用来存放“时”“分”“秒”。



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——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。




```

Setup
set s to 0
set m to 0
set h to 0

Loop
change s by 1
Wait 1 s
Label label0 show s
Label label1 show m
Label label2 show h

if s = 60
do
set s to 0
change m by 1

if m = 60
do
set m to 0
change h by 1

if h = 24
do
set h to 0
    
```

三、逻辑嵌套

我们根据秒的变化实现分秒时的转换使用循环判断，当“秒”等于60时，将“分”加一，且将“秒清零”。

然后进行判断“分”是否等于60，如果是则将“时”加一，且将“分清零”。

然后进行判断“时”是否等于24，如果是则将“时”清零。

这种在判断中再进行判断的逻辑判断，我们将它称为逻辑嵌套

四、功能拓展

使用逻辑运算与if判断，为手表增加一个定时闹钟功能。在程序中添加判断，当“时”、“分”、“秒”等于某一个时刻时，执行闹钟功能。

使用次数循环，每次循环执行“Speaker beep freq”块，每次执行时间为1秒，1s=1000ms。为了防止因为执行闹钟而导致秒数没能正常的增加，所以再每一次响铃一秒后，使用“change s by 1”，增加秒数通过“Label show”，将循环过程中的秒数变化显示出来。

```

Setup
set s to 0
set m to 0
set h to 0

Loop
change s by 1
Wait 1 s
Label label0 show s
Label label1 show m
Label label2 show h

if s = 60
do
set s to 0
change m by 1

if m = 60
do
set m to 0
change h by 1

if h = 24
do
set h to 0

if (h = 0 and m = 0 and s = 5)
repeat 10 times
do
Speaker beep freq: 1600 duration: 1000 ms
change s by 1
Label label0 show s
Label label1 show m
Label label2 show h

Button A pressed
change h by 1

Button B pressed
change m by 1

Button C pressed
change s by 1
    
```

按键调时我们可以使用按键控制，为手表添加一个调节时间的功能。

- (1) 按下按键A将变量“h”加一，即小时加一
- (2) 按下按键B将变量“m”加一，即分钟加一
- (3) 按下按键C将变量“s”加一，即秒钟加一

9 加速度的应用

简介: 使用M5GO内置的陀螺仪加速度计, 运用随机数编程, 制作一个摇摇骰子以及一个计步器。

目标: 学习使用陀螺仪的加速度程序, 掌握随机数与概率知识。

一、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动手动作, 其实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。

使用加速度计功能制作一个摇摇骰子。



- (1) 创建一个变量“Dice”, 用来表示骰子的点数
- (2) 单击块上的选项框, 选择“Rename variable”可更改变量的名称
- (3) 在Setup初始化程序时, 赋值变量“Dice”的初始值为“1”

Setup

set Dice to 1

- ✓ Dice
- Rename variable...
- Delete the 'Dice' variable

01 点击Hardware选项IMU

02 在这里我们将使用“Get X ACC”, 获取X轴方向的加速度

The screenshot shows the M5Stack IDE interface. On the left, the 'Hardware' menu is expanded to show 'IMU'. On the right, a code block is visible with the following structure:

```

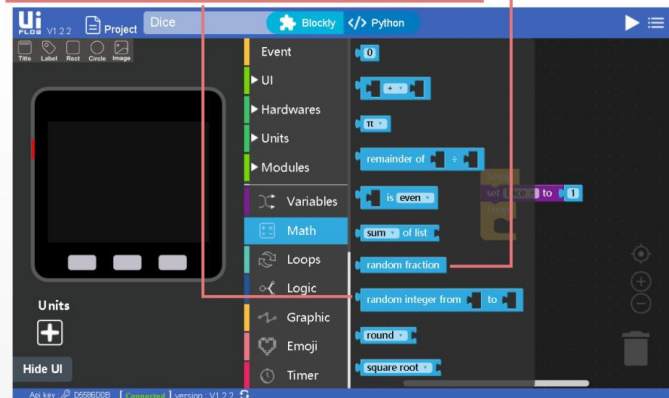
    [Get X ACC]
    [set (LCD) to 1]
    [loop]
  
```

二、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



该程序块则表示, 当每次执行到这个块的时候, 在1-6之间随机抽取一个整数, 并赋值给变量"Dice".



三、While条件循环

在"Loops"选项中, 提供了"while"条件循环, 它的功能是在判断条件成立时, 会无限循环包含在内的程序, 直到判断条件不成立时, 结束循环。



使用一个While条件循环, 判断条件, 当x轴方向的加速度大于1.5或小于-1.5时, 我们判断认为发生了MSGO晃动, 则不断运行包含其中的骰子点数生成程序"random"且点亮RGB bar, 直到停止晃动, 跳出While循环, 熄灭RGB bar, 我们也可以试着更改判断条件的数值, 这将改变晃动判断触发的灵敏度。

生成了骰子的点数, 接下来要做的就是将点数通过进行判断, 为接下来的点数显示做铺垫。



四、显示骰子点数

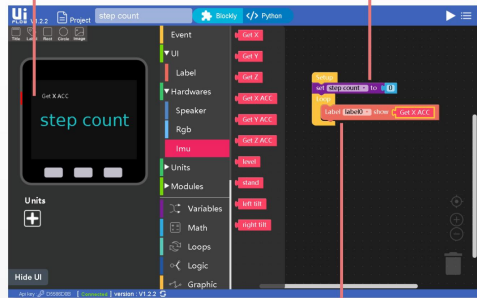
添加判断块, 添加条件关系式, 将变量"Dice"放入关系式中。

点击块上的小齿轮, 添加"else if"块到右侧的逻辑判断结构中去, 添加其他骰子点数的判断关系式, 使得程序进行多次判断。

使用Emoji图像绘制出不同的点数, 并放置到不同判断情况下的执行程序中。

计步器

- 01 添加两个标签 "Label" 放置在屏幕上，一个用作步数显示，另一个用作获取加速度值，创建一个变量 "step count"，这个变量将用来表示行走的步数
- 02 在 Setup 初始化程序时，赋值变量 "step count" 的初始值为 "0"



- 03 加速度值在不断的发生改变，所以我们需要使用程序不断的去对它进行访问，才能获取到最新的加速度值
- 04 在这里我们使用 "Label show" 去不断更新显示的加速度值，使得在循环中只要运行到了 "Label show Get X ACC"，我们就可以获得最新的加速度值

```

Setup
set count to 0
Loop
Label label0 show Get X ACC
if Get X ACC > 1.5
do
change count by 1
Label label1 show count
Wait 0.5 s
Button A pressed
set count to 0
Label label1 show count
  
```

- 05 对加速度值进行判断，当加速度值大于 "1.5" 时，我们认为是一次移动，所以我们将 "step count" 进行加一操作
- 06 将 "step count" 步数，通过 "Label show" 更新显示在屏幕上
- 07 进行 0.5 秒的延时
- 08 添加一个步数清零功能

由我们的程序运行速度实在太快了，很有可能当我们实际的走了一步，而程序就已经进行很多次的判断从而导致进行了很多次的步数增加。延时的目的在于缓解这种情况的发生，每进行一次步数增加后，需要经过 0.5 秒后，才允许有第二次的移动判断与步数增加。



10 旋钮控制

简介: 使用Angle角度输入模块, 进行RGB灯的亮度调节, 并将亮度百分比结合UI图形显示在M5GO屏幕上。

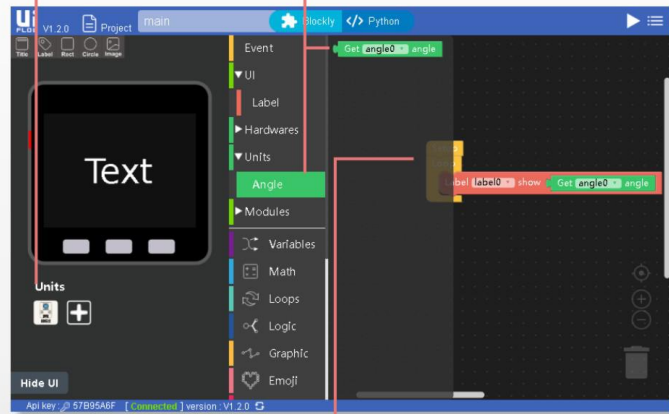
目标: 学习使用Angle角度传感器, 以及数据类型的转换, 了解数据映射, 以及图形UI的属性控制。

在我们的实际生活中, 有着各种各样的旋钮开关, 它们的控制着各式各样的功能。

一、Angle模块测试

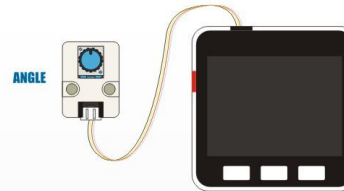
Angle是一个能够通过旋钮角度输入模拟量的模块。

- 01 首先, 我们在UIFlow中
添加"Angle"Unit
- 02 点击程序块列表中的Units选项
找到Angle的程序块"Get Angle"



- 03 添加一个标签Label, 将程序块通过
"Label show"显示出来看看具体的内容

将Angle Unit连接到M5GO对应端口上, 我们这里选择的是"B端口"
运行程序后, 当我们旋转传感器上的旋钮时, 将输入0-1024的模拟量



问题: Angle Unit是如何做到通过旋钮进行模拟输入的?

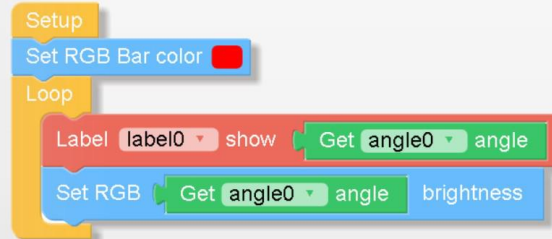
解答: Angle角度传感器也被称为"电位器", 它的硬件构造其实是一个"滑动变阻器", 在电路中, 有着一一种可以限制电路的电压的元器件, 它的名字叫做"电阻"。

而可以通过外界人为手动调节电阻值大小的这种电阻, 我们可以将它称为"滑动变阻器", 通过调整电阻值的方式, 输入不同的电压值, 从而实现模拟输入。

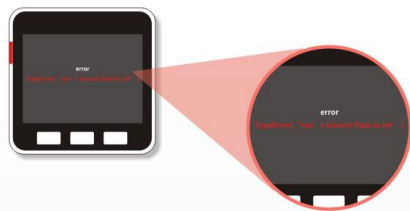
二、调光灯

了解了工作原理后, 我们来尝试使用Angle Unit控制RGB bar的亮度值

我们尝试将"Get Angle"块的数值直接用于控制的亮度



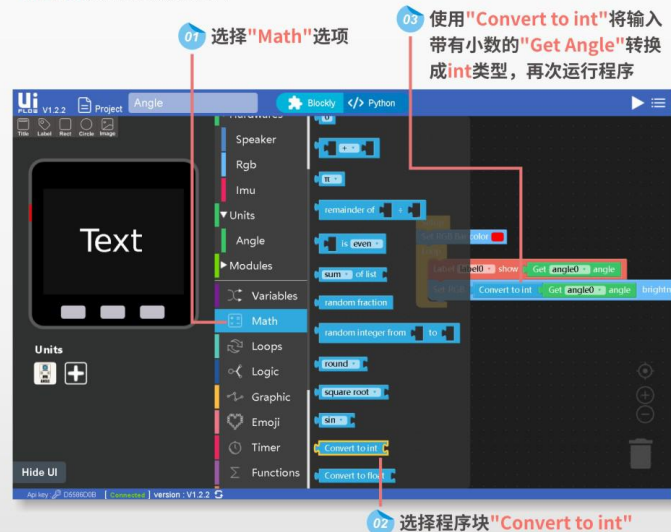
运行程序后，当我们旋转旋钮想要进行调节亮度的时候，我们发现有一个错误提示。



这是个错误的意思是，亮度值不可以是一个“float”型的数据，你需要将它转换成一个“int”型的数据。

三、“int”与“float”

“int”与“float”是两种常见的数据类型；其中“int”指的就是类似于“-4,-2,0,3,5”这样的整数；而“float”指的是类似于“-1.1,2.5,3.9”这样带有小数的数字。



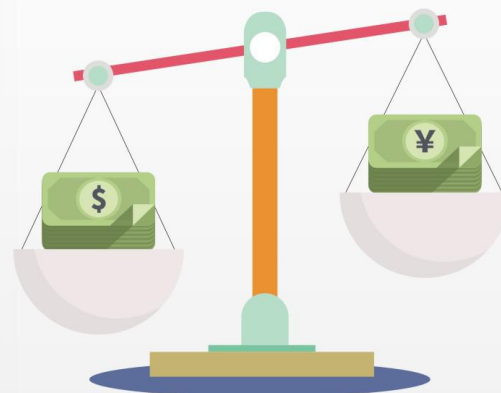
这时候我们发现，虽然调节灯光亮度的功能实现了，但是调节的灵敏度实在是太高了，AngleUnit输入值的范围为0~1024，而RGB bar的亮度值范围为0~255。当我们旋转旋钮的四分之一时，就已经将亮度从最低调节到最高。

那么我们有没有办法解决这个问题呢？

“我们可以使用数据映射去解决这个问题”

四、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美元兑换成了人民币，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。

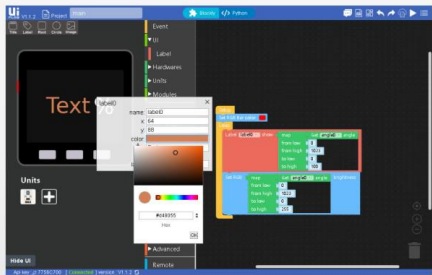


将需要进行映射的数据"Get Angle"，放置在map处，填写映射数据的范围，由"Get Angle"的0~1024，映射为0~255创建一个变量来存储映射后的新数据，在UIFlow中进行数据映射后，"map"会自动将映射创建出来的数据进行"int"处理，所以，我们可以直接将这个变量用做亮度值使用。



六、UI显示部分

为调光灯添加上显示部分，将调光的百分比显示在屏幕上。创建一个变量，用同样的方式将"Get Angle"数据映射为0~100，将变量值用做亮度百分比，通过"Label show"显示出来，并设置成自己喜欢的颜色。



运行程序，使用旋钮调节亮度吧

11 交通红绿灯

简介：使用RGB灯模块，运用变量与循环迭代编程，制作一个显示倒计时的交通红绿灯。
目标：学习变量的循环迭代。

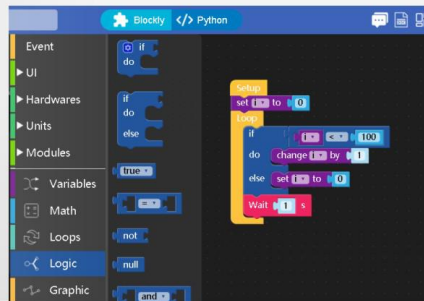
一、迭代的理解

我们可以将迭代程序理解为，将指定的一组数据按顺序的，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ATM机的取款排队，每一个人按照顺序，进行取款，如果把每个人从ATM取出的金额当作一个变量，把队伍中的人各自要取的金额当作一组数据。那么队伍（数据）按着顺序进行取款，每到下一个人取款金额（变量）跟着变化，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。

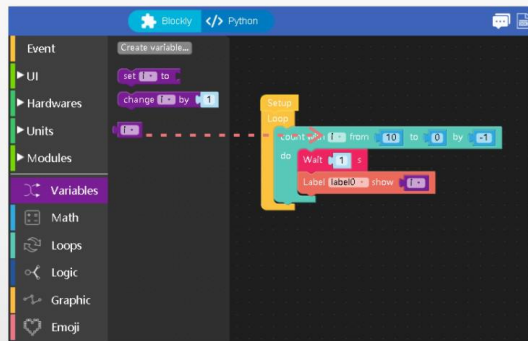


方法二
使用迭代“count with i from”；
填写迭代的参数：将“i”的初始值赋值为“0”，上限值为“100”，每次增加量为“1”；
在每次执行变量“i”加1后，都会执行一次do右侧的程序，延时一秒；
通过Loop无限循环这个过程。

使用迭代程序在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，当我们把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”选项中找到。



二、RGB灯模块测试



- 01 在UIFLOW中添加"RGB"Unit
- 02 点击Units选项选择RGB
- 03 拖动三个"Set RGB n index"到程序中
- 04 中间的数值框表示驱动的灯号 (1~3)
- 05 我们尝试将三个灯设置为不同的颜色
- 06 将RGB Unit连接到M5GO对应端口上，我们这里选择的是"B端口"，点击运行程序

三、交通红绿灯显示UI部分

首先我们需要添加两个标签；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先预先输入数字，预览看看效果，同时将标签摆放整齐另一个标签则固定显示“s”作为倒计时的秒单位。



四、程序部分

我们将整个程序分成三个部分：绿灯，黄灯，以及红灯。

1、绿灯

01 添加一个"Set RGB n index", 将控制灯号设置为"1", 并且将颜色设置为"绿色"

02 在其之后添加一个迭代"count with i from", 填写迭代的参数: 将"i"的初始值赋值为"6", 下限值为"0", 每次增加的量为"-1" (即每次进行减一)

03 每次赋值后执行的程序为显示最新的倒计时的"Label show", 并将变量"i"放置到显示内容中

04 添加一个1秒的wait延时

05 在迭代运行结束后, 使用"Set RGB all"将灯光设置为黑色使其"熄灭"



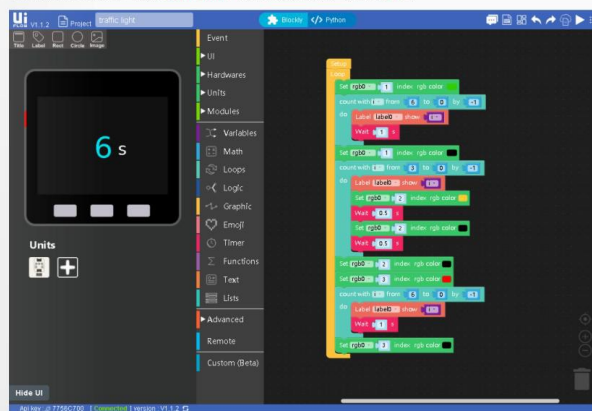
2、黄灯

填写迭代的参数: 将"i"的初始值赋值为"3", 下限值为"0", 每次增加的量为"-1" (即每次进行减一) 每次赋值后执行的程序为显示最新的倒计时的"Label show", 并将变量"i"放置到显示内容中; 使用"Set RGB n index", 将控制灯号设置为"2", 并且将颜色设置为"黄色"; 并且经过0.5秒的延时后颜色设置为"黑色"将其熄灭, 再经过0.5秒延时, 进入下一次迭代赋值。

3、红灯

添加一个"Set RGB n index", 将控制灯号设置为"3", 并且将颜色设置为"r"红色; 再次复制一个迭代"count with i from", 迭代的参数为: 将"i"的初始值赋值为"6", 下限值为"0", 每次增加的量为"-1"每次赋值后执行的程序为显示最新的倒计时的"Label show", 并将变量"i"放置到显示内容中, 以及添加一个1秒的wait延时。

在迭代运行结束后, 使用"Set RGB all"将灯光设置为黑色使其"熄灭"。



运行程序, 看看这个能够指挥交通的红绿灯是怎么工作的吧。

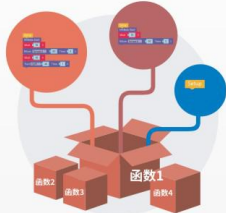
12 光线感应与函数

简介: 使用Light光线感应模块, 运用函数与循环迭代编程, 配合逻辑判断根据环境光线亮度而发出不同的灯光效果。

目标: 学习函数概念以及使用, 加深学习逻辑, 与灯光控制。

一、基础函数理解

函数就像一个包裹, 我们可以给函数取一个名字, 在函数里放入程序; 当函数被调用时, 它就会运行它包含在内的程序, 当程序中有多段程序重复的时候, 使用函数可以节省程序的长度, 将不同的功能分别编程成一个个独立的函数方便使用的同时使程序更加的简洁明了, 同时方便修改。



二、函数的创建

01 点击程序列表中的"Function"选项

02 拖动"to dosomething"块到编程区域中

03 点击输入框可以更改函数的名字

04 可以往函数内添加想要执行的程序, 这一部分被称为"函数体"

三、函数的调用

创建好了函数后, 在"Function"选项会自动出现, 与函数名一致的调用程序块, 将它放置到程序中, 当运行到该块时, 就会自动跳转到对应的函数中去, 并执行函数体中的函数。

四、Light模块测试

Light是一个能够通过感应环境亮度输入模拟量的模块。

01 在UIFlow中
添加"Light"Unit

02 点击程序列表中的Units选项,
找到Light的程序块"Get Light"

03 添加一个标签Label, 将程序块都通过
"Label show"显示出来看看具体的内容

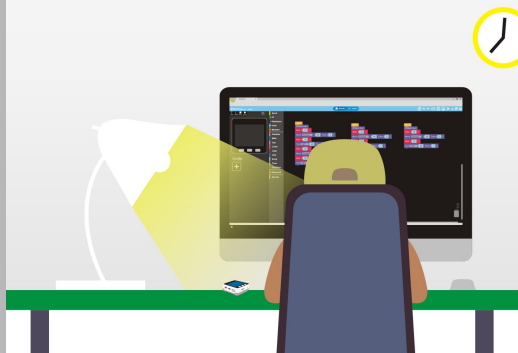
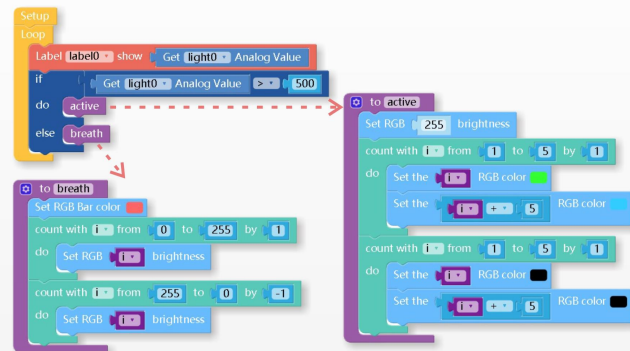
将Light Unit连接到MSGO对应端口上, 我们这里选择的是"B端口"
点击运行程序, 当我们用手去遮挡, 与用强光照射时, 将输入0-1024的模拟量。

问题: Light Unit是如何做到通过光线感应进行模拟输入的?

解答: Light光线感应模块也被称为“光敏传感器”,它的硬件构造其实是一个“光敏电阻”,与Angle Unit中的变动变阻器相似,不同之处在于,光敏电阻的电阻值不需要我们手动去调节,光敏电阻的电阻值随这光照强度的增加而减小,因此我们可以通过更改光照的方式,输入不同的电压值,从而实现模拟输入。

五、灯光功能函数

将不同的灯光效果放置到不同的函数中,不同的亮度执行不同的函数(即灯光效果),当白天环境光线良好,亮度值大于500时,执行函数“active”实现跑马灯效果,当夜晚环境光线较差,亮度值小于等于500时,执行函数“breath”实现呼吸灯效果,仿佛萤火虫的光芒般逐渐点亮又逐渐熄灭。



13 电子通讯录

简介: 使用数组与按键控制, 编程一个可以循环翻页的电子通讯录

目标: 学习数组概念以及创建与使用

一、数组的理解

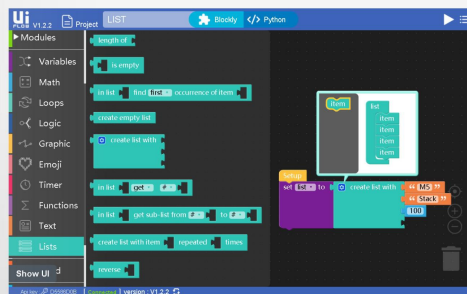
什么是数组? 我们可以把数组理解为多个变量的集合, 如果把变量比喻作一个瓶子, 那么数组就是一个可以放置多个变量的一个柜子, 这些变量存放在数组中会被排序, 通过这些序号我们可以访问它们, 或是改变它们的值。



在数组的创建时, 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数组中元素的序号, 就可以访问到数组中的数据。

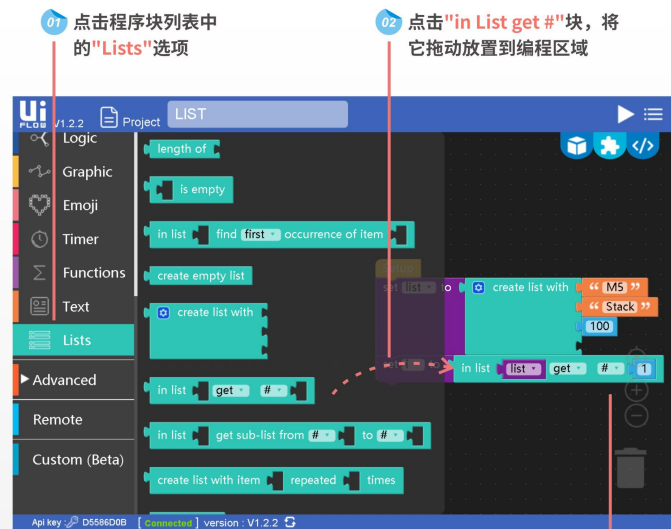
二、数组的创建

我们需要创建一个变量用做整个数组的存放; 点击程序列表中的“Lists”选项, 选择“Create List with”块; 我们可以看到该块的右侧, 有着许多接口, 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, 变量等数据到数组中; 这些数据会由上而下, 以1~N的顺序编号, 当通过数组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接口连接的数据都将为数组的一部分, 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

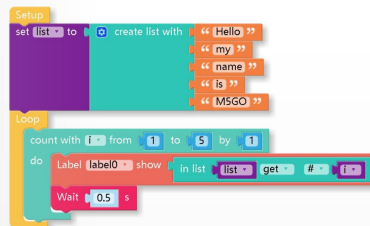
在UiFlow中，数组的数据是以1~N的顺序编号的，例如我们想要调用数组中的“M5”数据，它对应的编号为“1”，那么在调用时，就需要将编号“1”放置到#位置。

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。

小练习

使用数组编程实现一个自动切换滚动的信息显示。

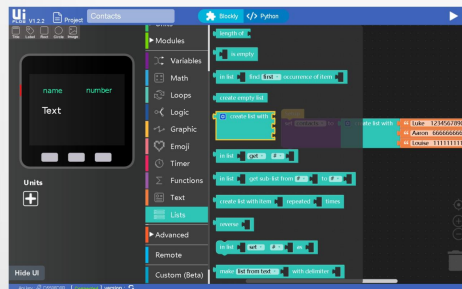


1-1 电子通讯录

编程实现，当程序运行时，M5GO屏幕显示联系人姓名，按下按键A与按键C可以进行左右循环翻页，切换联系人。

一、显示UI部分

首先我们需要添加三个标签分别表示：联系人标题，号码标题，以及联系人号码数据；可以先输入标题，与数据内容，预览看看效果，同时将标签摆放整齐。



二、数据部分

创建一个数组，并用变量“contacts”去存储这个数组

点击程序列表中的“Text”选项，点击“字符串”块，多次将它作为数据放置到数组上编辑字符串内容为联系人姓名与号码，并用空格隔开

三、程序部分

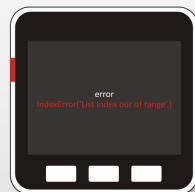
01 创建一个变量*i*作为访问数组的编号

02 使用"in List get #"块进行数据索引,并将索引结果通过"Label show"块显示出来

03 添加两个"Button"块到编程区域,添加"change i by -1"与"change i by 1"块到A按键与C按键中,我们可以通过按下按键,改变我们索引的编号,达到"翻页效果"

四、运行测试

我们发现在开始的正常翻页与使用时,通讯录能够正常运行,可一旦当我们向着同一个索引方向多次翻页到一定次数时,就会出现一个程序错误提示,这是为什么呢?



五、产生原因

当我们不断的按下按键A或是按键C时,用来索引数组数据的变量*i*就会超出数组数据能够访问的索引范围,目前我们所建立的数组中的数据一共有3个,所以它的索引范围为1~3。当变量*i*不在这个范围,因为找不到对应的数据,程序就会出错。

六、解决方法

运用逻辑判断if,我们可以避免这种状况的发生,在A与C的两个Button程序中分别修改程序。

按键A被按下后,判断当"*i*"是否大于"1",当条件符合时执行"change i by -1"进行减一,否则将变量"*i*"重新赋值为"3"

按键C被按下后,判断当"*i*"是否小于"3",当条件符合时执行"change i by 1"进行加一,否则将变量"*i*"重新赋值为"1"



14 红外遥控

简介：实现两个M5GO之间简单的红外通信，模拟红外遥控器与电视机的控制，远程实现切换节目。

目标：学习图片编程以及绘图工具的使用，了解红外控制原理，深入学习逻辑嵌套。

一、红外线是什么？

红外线是太阳光线中众多不可见光线中的一种，由英国科学家赫歇尔于1800年发现，又称为红外热辐射，热作用强。他将太阳光用三棱镜分解开，在各种不同颜色的色带位置上放置了温度计，试图测量各种颜色的光的加热效应。结果发现，位于红光外侧的那支温度计升温最快。因此得出结论：太阳光谱中，红光的外侧必定存在看不见的光线，这就是红外线。红外线除了是一种不可见光以外，还可以当作传输之媒介，红外线在生活中有着各式各样的应用，比如最常见的电视与空调的遥控器等。



在本节中，我们将学习如何通过M5GO，模拟红外遥控与电视机之间的通讯。我们需要使用两个M5GO，以及两个IR Unit，分别编程不同的程序，分别充当遥控与电视的角色。

二、遥控端程序

01 添加"IR Unit" 使用Button按键块

02 当按下按键时执行程序 "Set IR on"，通过IR Unit上的发射器向外发射红外信号

03 当松开按键时执行程序 "Set IR off"，停止发送

三、电视端程序

- 01 添加"IR Unit"
- 02 使用无限循环Loop，使用"Label show"块将"IR state"的接收情况显示在屏幕上



现象分析

将遥控端，电视端分别在两个M5GO上运行，并将遥控端的IR Unit对准电视端的IR Unit，然后按下按键A，进行信号发射。

- 当发射端的M5GO当按下按键A时，通过IR Unit向外发射红外信号。
- 当接收端接收到信号后，"IR state"的值将由"0"变为"1"。

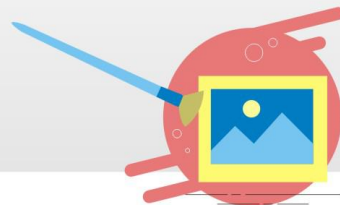
现在我们的电视端已经能够接收到红外信号，接下来我们将尝试着去丰富它的功能,使它像电视一样显示一些"节目"

1-1 绘制图片

除了显示基本的UI图形以及文本标签以外，UIFlow还支持使用一些自定义的图片进行编程，下面将讲解如何去使用这一功能，以及一些注意事项。

四、首先是图片的规格与属性

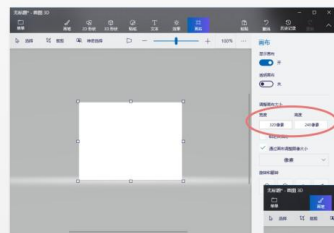
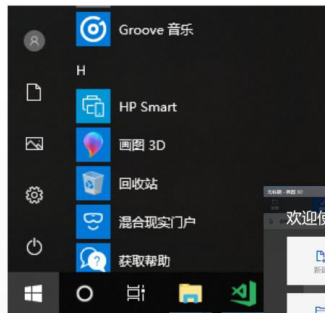
- 图片的像素尺寸应该控制在320x240以内，否则将无法完整显示
- 图片支持的图片格式有"bmp"，"jpg"
- 图片的大小应该控制在25kb以内



五、使用绘图工具绘制图案

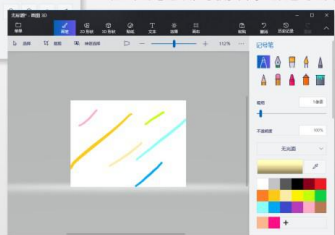
1. 在一般的电脑操作系统中一般都会提供了一些内置的绘图工具，我们可以使用绘图工具去绘制一些图片，并将它们显示在M5GO上，打开电脑中的绘图软件，进行图案绘制，下面演示使用的为"Windows10"自带的"画图3D工具"。

2. 新建一个画布

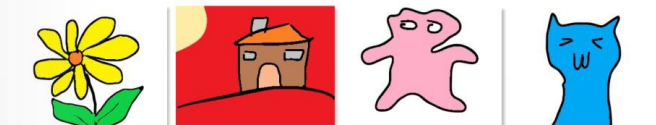


3. 将绘图的画布像素尺寸设置为"320x240"，与M5GO屏幕的分辨率一致，使得能够完整的铺满在屏幕上。

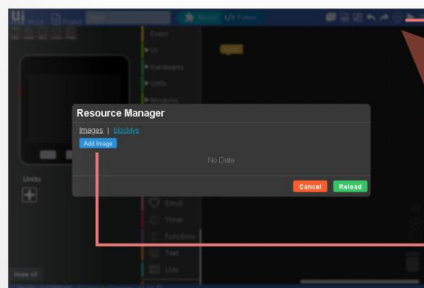
4. 点击画笔选项，使用不同的颜色与画笔开始创作吧。



5. 我们尝试绘画四张不同的图片，且分别进行保存，用来表示不同的电视节目，分别命名为"C1-C4"。

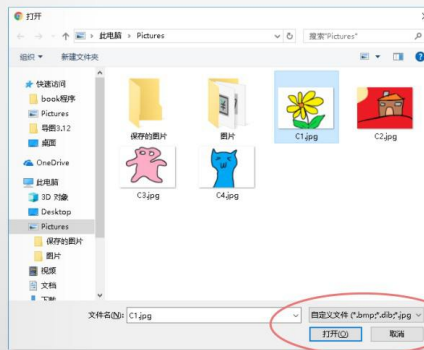


1-2 上传图片

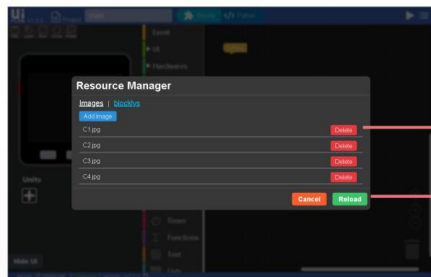


01 点击UIFlow菜单功能栏的"Manager"选项

02 点击"Add Image"



在本地目录中选择要上传的图片
(需符合图片格式与大小要求否则
将无法上传)



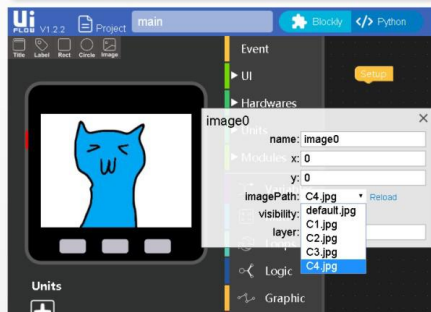
上传成功后，图片将被保存在MSGO的内部储存中，在image列表中就会显示相对应的图片文件，这时候就可以使用图片进行编程了。

Delete

点击“Delete”可以删除已经添加的图片。

Reload

点击“Reload”可以将已经保存在MSGO内部储存中的图片加载出来。



一、图片编程

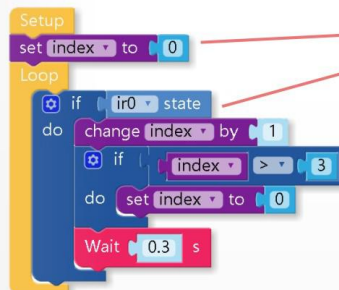
图片的编程方式与UI图形类似，从UI模拟器上方拖动“image”到屏幕上。单击元素，弹出属性框切换显示图片还可以进行预览。



添加图像到UI模拟器后，程序块列表中的“UI”选项里会出现“image”的一些操作程序块，提供进行编程。

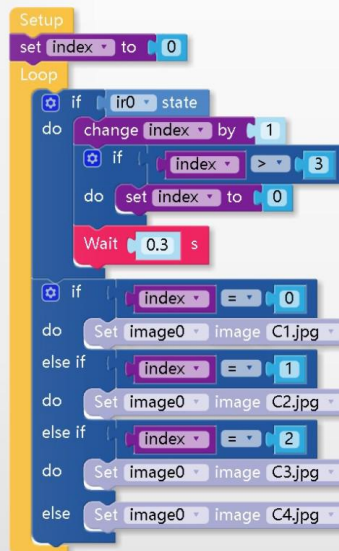
二、电视频道切换

创建一个变量“index”，用来索引不同的节目频道，使用if判断，IR的接收情况，当“IR state”等于“1”时（即接收到红外信号时），将变量“index”加“1”，然后进行0.3秒的延时。与之前的数组访问一样，我们的节目频道并不是无限的，所以我们需要添加一个if判断用作限制，使用if判断，当“index”等于“X”时（即已经超过了最后一个节目的索引号了），将“index”重新赋值为“0”，使得节目显示重新回到了最初的第一个频道。



三、电视节目显示

有了电视频道的索引号“index”，我们只需要判断当前“index”的值为多少，再对应的显示节目，就完成了节目与频道的关联。添加if判断，当“index”等于“1”时，执行显示第一张图片“Channel1”，点击if块上的齿轮添加“else if”与“else”到判断结构中，添加判断条件，当“index”等于“2”时，执行显示第一张图片“Channel2”，依次进行判断与执行。



思考：在实际生活中的红外遥控器有那么多按键，可是发射装置只有一个，是怎么实现多功能的呢？
解答：实际的红外遥控器发射的控制信号使用了一种特殊的脉冲编码，通过判断很短一段时间内红外信号开与关的时序识别出传递的信息。

15 游戏工程师

简介: 使用按键控制, 运用变量与循环迭代, 逻辑判断等综合知识与提示音结合UI图形控制, 编程制作游戏。

目标: 编程知识的综合使用, 对变量, 逻辑有更深入的理解。

1-1 炸弹传递游戏

制作一个带有提示音的“M5炸弹”, 随着时间的变化, 提示音越来越急促, 直到时间结束, 点亮RGB bar为红色, 表示引爆。

问题: 为了实现随着时间的变化, 实现倒计时的同时, 使灯光闪烁与提示音变得越来越急促, 我们需要更改wait延时块的延时时间我们要如何使得每一次的wait延时都不一样呢?

解答: 我们可以定义一个运算式来解决这个需求, 变量“time”随着每一次提示程序的执行都会进行减一, 于是我们可以用“time”去除以一个固定的数值用做延时时间, 这样随着time的减小, 延时时间也在减小。

添加一个标签用作提示, 表示当前炸弹的运行状态



01 创建一个变量“time”来做定时将初始值设定为50



02 判断“time”是否大于“0”, 当大于“0”时, 我们则认为炸弹还未引爆, 执行提示音与绿色灯光闪烁指示, 并将定时值“time”通过“change time by -1”, 自身数值减一

03 Wait延时的时间关联变量“time”

04 当“time”小于等于“0”时, 则表示已经达到定时时间, 执行红色灯光, 表示引爆

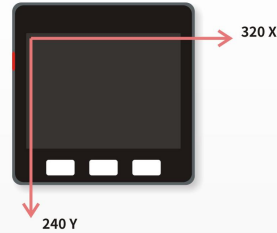
05 添加一个Button按键块, 当我们按下时, 将定时数据“time”重新设置为“50”, 这样新的一局游戏就开始了



1-2 躲避球游戏

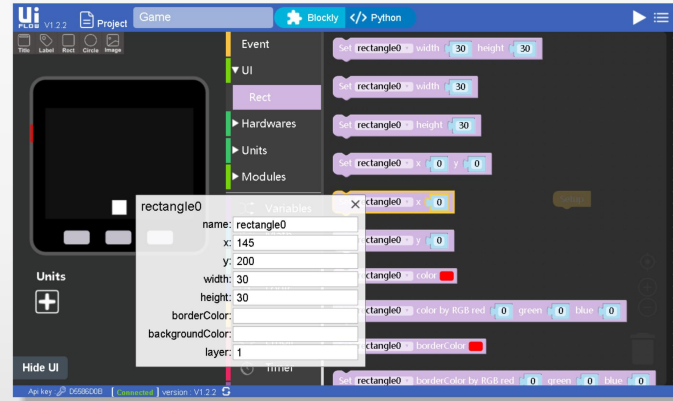
一、UI模拟器坐标系了解

UI模拟器中有着一个平面的X, Y坐标系, 屏幕上的元素能够根据坐标, 实现精准的位置, 坐标系与屏幕像素分辨率 (320x240) 是对应的, 这意味着如果你为元素设置了过大或过小的坐标值, 它超出屏幕的显示范围, 无法显示。



二、编程操作对象

添加一个矩形块到UI模拟器中, 放置在合适的位置, 单击矩形块在弹出的属性框中查看, 并记录下它的坐标信息



01 创建一个变量"r", 并在Setup下作为初始化程序将"r"赋值为当前矩形块的X初始坐标145

02 添加两个Button按键A,与C, 分别添加增减变量值的块

03 点击UI, 选择矩形Rect

04 并放置设置矩形块位置的X坐标控制块"Set rect X"

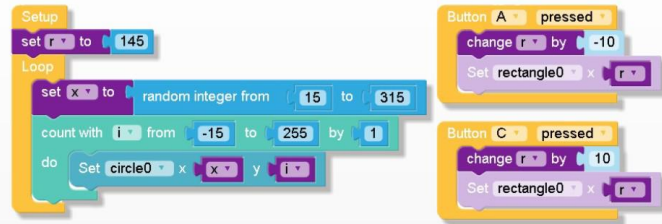
05 将变量"r"作为矩形的X坐标数值, 这样每当我们按下按键, 就能够通过改变变量"r"的值来改变矩形的坐标, 最终达到移动矩形的效果



三、随机下落的球

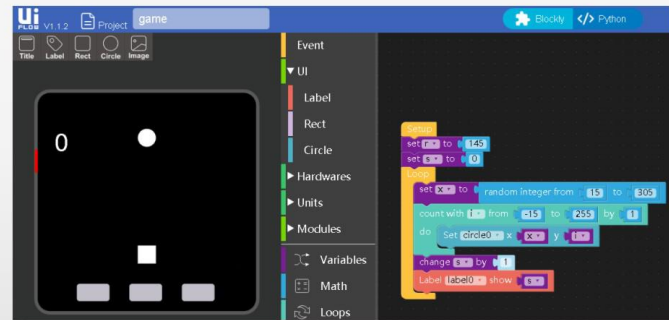
添加一个圆形到UI模拟器中，放在屏幕上方的位置，接下来需要做的就是如何让这个圆形随机的出现在上方，且由上而下的进行移动。

创建一个变量“x”，赋值为一个范围随机数，由于圆形的坐标点落在了圆心上且圆形的当前圆形的半径为15；所以我们设置随机生成15~315的数字并赋值给变量“x”，用于控制圆形的X轴坐标；使用循环迭代“count with i from”，填写迭代的参数：将“i”的初始值赋值为“-15”，上限值为“255”，每次增加的量为“1”使用循环迭代提供的变量“i”，对矩形的Y轴坐标进行迭代，使得矩形的Y轴坐标不断的变化，产生一种自动移动的效果。



四、分数统计

创建一个变量“s”，并在Setup下作为初始化程序将“s”赋值为“0”；变量“s”将用作分数统计，在每次小球掉落结束后，我们使用“change s by 1”使它加“1”，并使用标签将它显示在MSGO屏幕上；实现了操作对象的控制与随机下落的小球，接下来要做的就是进行这个游戏规则的制定了，“当下落的小球碰撞到我们操作的矩形的时候，就会被判定为失败，游戏结束”。

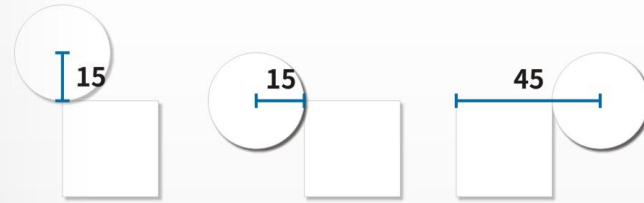


五、碰撞判断

问题：我们要怎么做才能够判断，小球与矩形是否发生了碰撞呢？

解答：通过逻辑判断以及对小球与矩形二者的坐标进行逻辑运算，由此判断碰撞是否发生。

判断圆形掉落接近矩形且发生碰撞，我们需要考虑以下因素，由于圆形的坐标点在圆心，且目前圆的半径为15，矩形的坐标点在矩形的左上角，且矩形的长宽都为30，因此对与判断坐标值的选择，我们有了下面几种场景。

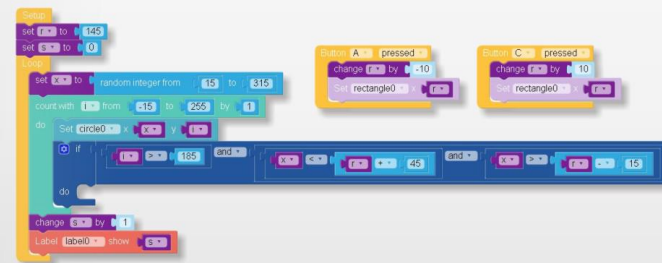


if的判断的第一个条件" $x > r - 185$ "，i在循环迭代中用作表示圆形的Y轴坐标，而矩形的Y轴坐标为"200"，当圆形下落到与矩形邻近时，才有可能发生碰撞。

" $x > r - 15$ "，当圆形从矩形的左边落下，两者之间的横向距离必须小于15才有可能发生碰撞。

" $x < r + 45$ "相对这矩形坐标，当圆形从矩形的右边落下，两者之间的横向距离必须小于45才有可能发生碰撞。

当三个条件同时满足时，表示小球与圆形发生了碰撞，我们还需要进一步的完善碰撞后的执行程序。



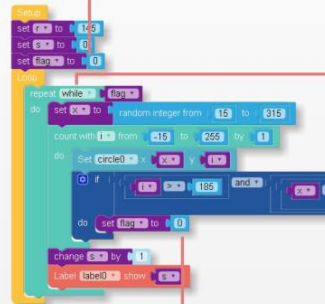
问题：碰撞判断程序为什么要放在循环迭代里？如果我们放在其他地方，还能判断到碰撞的发生吗？

解答：循环迭代的运行产生了小球由上而下的完整移动，我们需要在每一个小球移动的整个过程不断的判断是否有碰撞，如果将判断放置在其他地方，则是每一次小球完整的下落后才进行判断，即使发生碰撞，也会因为没有及时的进行判断而错过。

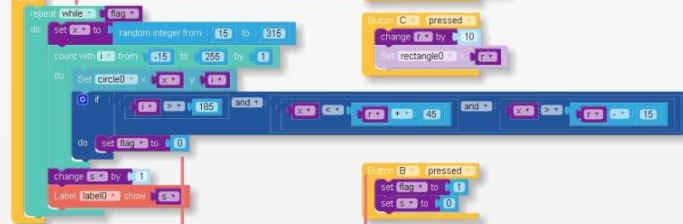
六、游戏运行

我们发现，当我们点击运行程序后，游戏就自动开始了，我们并没有控制游戏的开始。在游戏中，当发生碰撞后，意味着游戏结束，我们需要使游戏终止，等待下一局的开始，可还是不停有小球在下落，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。

01 创建一个变量“flag”，在Setup下作为初始化程序将“flag”赋值为“0”



02 使用条件循环“repeat while”判断“flag”，当标志“flag”为“1”时，无限循环执行游戏部分的程序



03 游戏执行时当发生“碰撞”，则执行将“flag”设置为“0”紧接着再下一次“while”判断时跳出游戏循环

04 于是我们使用一个Button按键块B，来执行将“flag”置“1”这个任务，同时将上一局获得的分数变量“s”清零

由于我们在初始化的时候将“flag”设置为“0”，所以在刚运行程序的时候，游戏是不会自动开始的，程序会不断的在Loop里面等待一个游戏开始的信号，这个信号就是“flag为1”。当需要再次开始游戏时，我们只需要将标志位“flag”重新置“1”，游戏部分的程序就会再次运行。

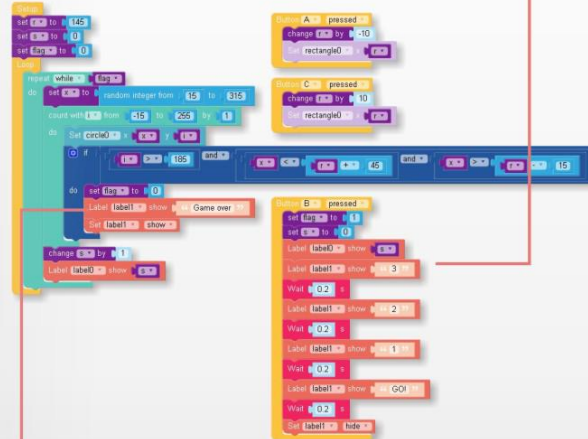
七、功能完善

目前已经实现了能够控制矩形移动，躲避下落的小球，当矩形与小球发生碰撞时，游戏结束，实现游戏的基本功能操作与规则；可在实际玩的时候，虽然游戏结束，屏幕却没有任何的提示，当我们按下按键B重新开始游戏的时候，在按下的一瞬间游戏就开始了，没有提供一个准备时间；在首次运行的时候，屏幕没有提示我们应该如何操作，来开始游戏。

思考一下，我们要如何修改程序，为游戏添加开始与结束提示，开始的倒计时准备时间。

我们可以使用一个标签来显示所有的提示信息向UI模拟器中再添加一个标签，单击打开属性框，将默认的文本设置为“Press button B”，这样当首次运行的时候屏幕上就会显示这个提示。

01 在Button按键块B中使用“Label show”块，配合Wait延时块，制作一个倒计时显示，在倒计时的最后，添加一个“Set Label1 hide”块，将整个标签隐藏起来，防止遮挡游戏的运行



02 在碰撞判断后的执行程序添加一个“Label show”块，显示“Game over”，提示游戏的结束，使用“Set Label1 show”块，将隐藏的标签再次显示出来，等到下一次“while”判断，游戏进入停止

16

远程控制

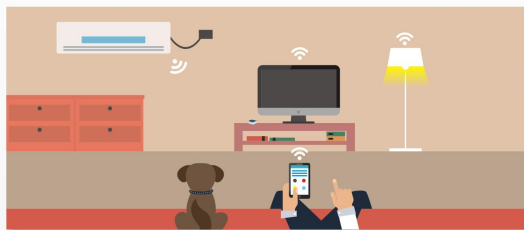
简介: 使用远程控制实现灯光控制与天气信息采集。

目标: 学习使用远程控制的按键, 开关, 滑动条功能, 学习变量的传递。

注意: Remote功能的使用需要在M5GO处于网络连接模式下, 详细网络配置方法请查看附录"配置WIFI"。

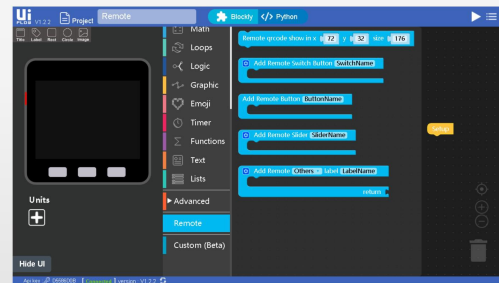
什么是远程控制, 为什么要学习远程控制? 我们可以使用远程控制做什么?

我们所说的远程控制是指通过网络或其他通讯手段实现远距离控制, 基于远程控制我们可以设想出各种各样的功能, 例如通过手机远程控制开关灯与调节亮度, 控制家用电器的打开与关闭, 远程监测家中的环境信息等等的智能家居应用场景, 充分利用远程控制将使我们的生活变得更加便捷与高效。



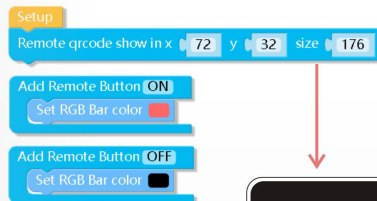
一、Remote

在"Remote"选项中提供了一系列的远程控制程序块, 使用它们可以使我们快速实现基于网络的远程控制, 下面将向你讲解如何去使用它们。

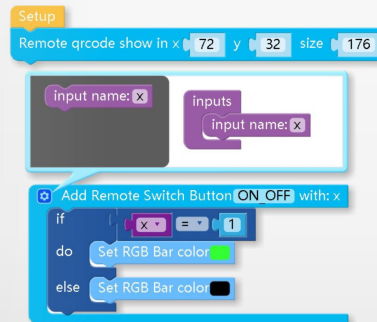
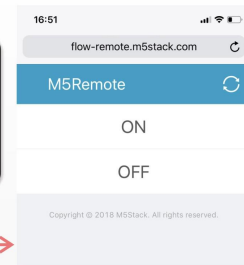


二、Remote Button

"Remote Button"能在控制页面生成一个与程序对应的按键, 使得我们可以通过按下按键去触发程序的执行。与之前学习的物理按键控制是类似的, 不同的地方在于, 这次我们可以在电脑或者手机上控制按键的按下。

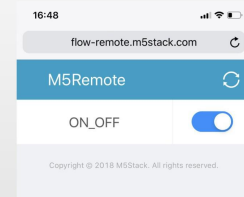


QR-code
生成一个能够访问控制页面的二维码, 并显示在M5GO的屏幕上, 我们可以使用手机扫描这个二维码, 或是在UIFlow上复制二维码的连接, 进行访问。



三、Remote Switch

"Remote Switch"能在控制页面生成一个开关, 这里涉及到了一个变量传递的知识。当我们在使用"Remote Switch"的时候, 首先需要为它添加一个"变量x"用来获取开关的状态, 当开关为"ON"时, 变量x会被赋值为"1", 当开关为"OFF"时, 变量x会被赋值为"0", 使用程序对变量x的值进行判断, 就是"Remote Switch"的主要使用方式。



Setup

Remote qrcode show in x: 72 y: 32 size: 176

input name: x

inputs input name: x

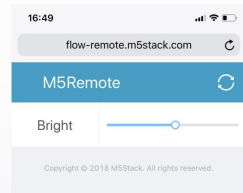
Add Remote Slider Bright

Set RGB Bar color

Set RGB	map	x	brightness
from low	0		
from high	100		
to low	0		
to high	255		

四、Remote Slider

"Remote Slider"能够在控制页面生成一个滑动条,同样的也是使用变量传递的形式。当我们在使用"Remote Slider"的时候,首先需要为它添加一个"变量x",用来获取滑动条的状态根据滑动条的滑动情况,变量x会被赋值为"0-100"。



Setup

Remote qrcode show in x: 72 y: 32 size: 176

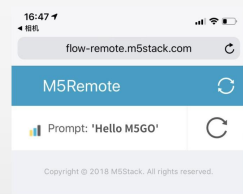
Add Remote Others label LabelName

- Others
- Temperature
- Humidity
- Pressure
- PIR Status
- Angle
- Weight

return: "Hello M5GO"

五、Remote Label

"Remote Label"的作用主要是在控制页面显示一些信息,可以是一些没有变化的静态提示文本,也可以是一些有着变化的数值,如温湿度值等,点击块上的标签选项还可以更改显示的图标。



六、小练习

编程实现通过远程控制,实时控制RGB bar的开关,调节RGB bar的亮度,控制灯光特效。使用ENV Unit将M5GO所处环境的温湿度气压值情况显示到手机或电脑上。

Setup

Remote qrcode show in x: 72 y: 32 size: 176

Add Remote Button Polce

repeat 10 times

do

- Set left side RGB Bar color
- Set right side RGB Bar color
- Wait 0.2
- Set left side RGB Bar color
- Set right side RGB Bar color
- Wait 0.2

Set RGB Bar color

Add Remote Temperature label Temperature

return: Get env0 Temperature

Add Remote Humidity label Humidity

return: Get env0 Humidity

Add Remote Switch Button ON/Off with x

If

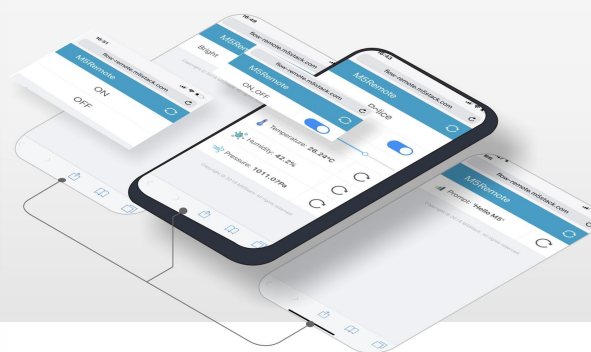
- Set RGB Bar color
- else Set RGB Bar color

Add Remote Slider Bright with x

Set RGB	map	x	brightness
from low	0		
from high	100		
to low	0		
to high	255		

Add Remote Pressure label Pressure

return: Get env0 Pressure



APPENDIX:

FIRMWARE UPGRADE

APPENDIX

WIFI CONFIGURATION

LEARN MORE:

FIRMWARE UPGRADE

「附录」
BEFORE WE START

M5GO

M5GO



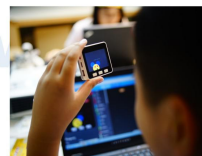
更新固件
Firmware Upgrade



配置WIFI
WIFI Configuration



附录
Appendix



了解更多
Learn more



M5GO

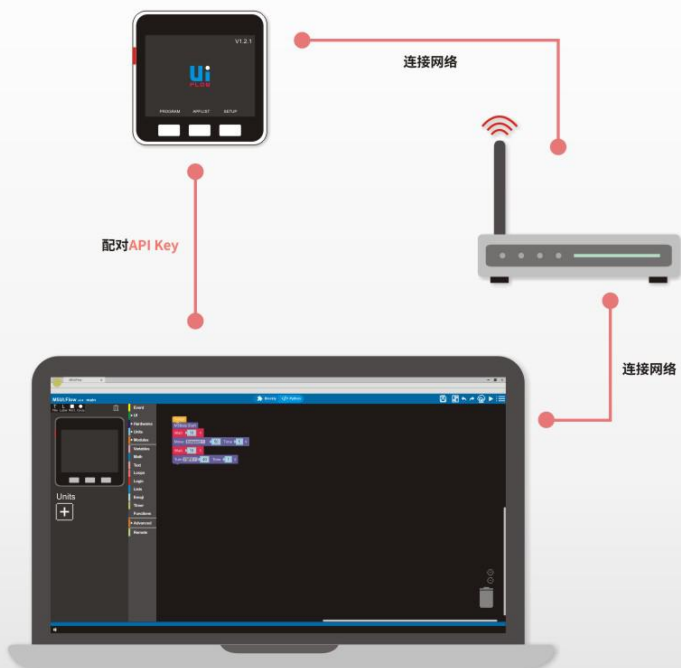
M5GO

M5GO
M5GO

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MSGO是两个独立的单元，在使用UIFlow进行编程前，我们需要将MSGO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

01 单击左侧电源键开机



02 在屏幕出现主菜单后，选择 setup 选项



03 进入 WiFi 选择页面，上下移动光标选择 change WiFi connect

04 选择 change wifi connect 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



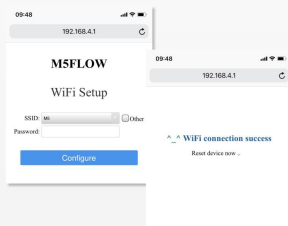
06 当有设备成功连接这个热点，屏幕上就会显示出一个二维码



07 打开手机浏览器扫描这个二维码，或是直接访问 **192.168.4.1** 这个地址，进入WiFi配置页面



IP:192.168.4.1



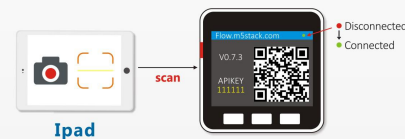
08 选择要连接的WiFi，然后填写密码，点击连接，配置成功将会自动重启

配对API Key

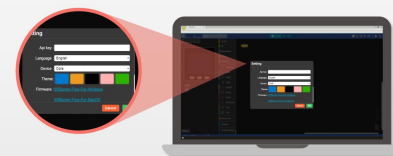
01 在开机进入主菜单后，选择"PROGRAM"进入编程模式



02 进入编程模式后，留意屏幕右上角的指示灯，当指示灯由红色变成绿色时，则表示已经成功连接上了网络



03 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



04 在API Key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

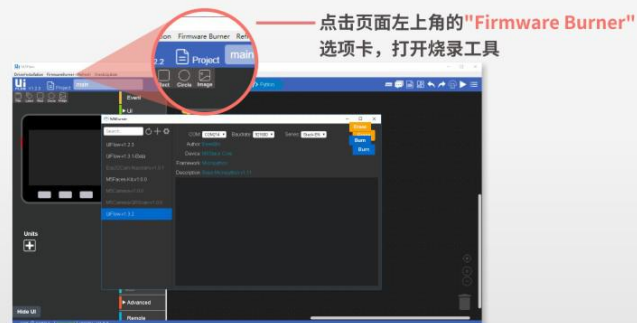
为了持续优化体验与增加更多强大的功能，我们会不断的推出新的固件版本，M5GO的固件版本与UIFlow的版本是同步更新的，你可以留意M5GO开机时，主菜单页面右上角的版本号，与UIFlow页面左上角的版本号是否一致，当你发现UIFlow的版本超前于M5GO的版本，则表示我们推出了新的固件版本，请及时更新。UIFlow-Desktop-IDE集成了M5GO的固件烧录功能，所以你无需下载其他的烧录工具就能进行固件的更新。



主机版本号

UIFlow版本号

提示：烧录固件与程序推送都需要使用到USB驱动，你可以在打开UIFlow编程器的时候根据提示进行安装，或是点击UIFlow页面的左上角的“Driver Installation”打开驱动安装程序。



点击页面左上角的“Firmware Burner”选项卡，打开烧录工具

一、COM

选择M5GO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M5GO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。

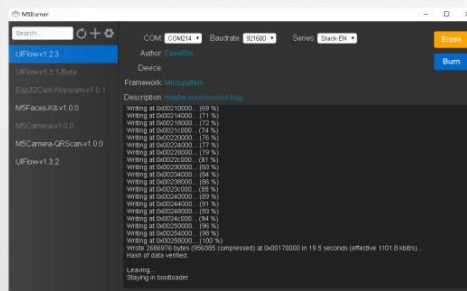


二、Baud

波特率可以简单的理解为烧录固件的速率，我们建议选择“115200”或“921600”。

三、Firmware

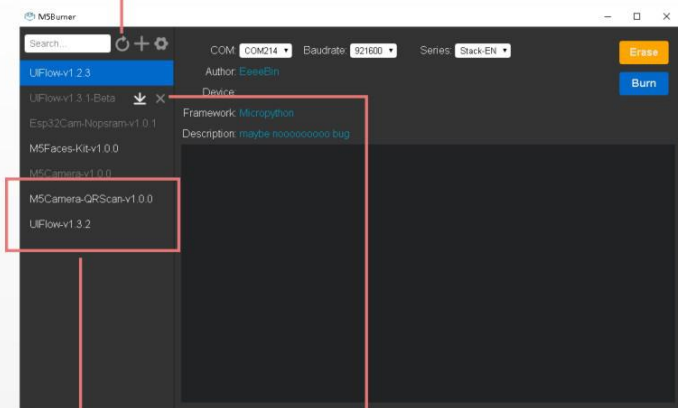
选择要更新的固件版本，除了最新版本外，还提供了几个先前版本选择，不同的设备需要烧录不同的固件，部分设备也可以兼容多种固件，使用UIFlow作为编程器的则需要烧录“UIFlow”系列的固件。



四、固件版本

获取更多的固件版本。

01 点击刷新按钮可以刷新获取最新的固件



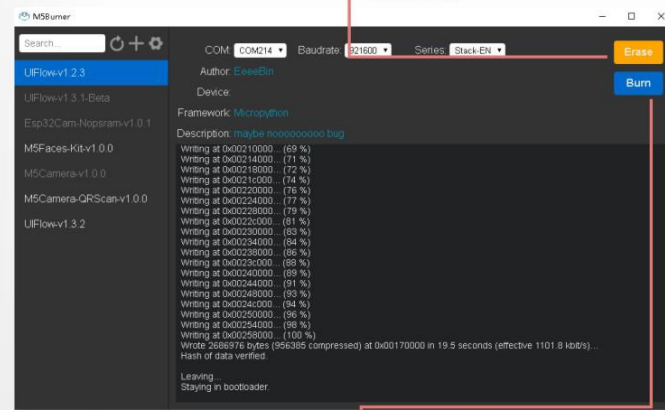
02 标题为白色的固件表示已经本地保存，可直接进行烧录。

03 标题为灰色的固件版本表示还未下载到本地，需点击下载按钮进行下载。

五、开始烧录

配置好烧录参数，就可以开始进行烧录操作了。

01 在首次更新时建议先点击"Erase"将内存清除一次，等待提示"finished"清除完成（在后续的更新中，则不需要清除，否则将清除已经保存的WIFI信息，以及更新API Key）



02 点击"Burn"开始烧录，等待提示"finished"烧录完成

M5GO



访问官方网站<https://m5stack.com/>获得更多有关M5的资讯与视频。

我们希望你在这本书中，能够充分掌握M5GO与UIFlow编程器的基本使用，在使用M5GO学习编程的同时收获快乐，并充分发挥你的创意，将M5GO运用到实际项目中，搭建出自己的物联网应用。

获取更多的Unit功能模块与M5GO配件，来拓展你的项目功能。



官网



速卖通



淘宝

