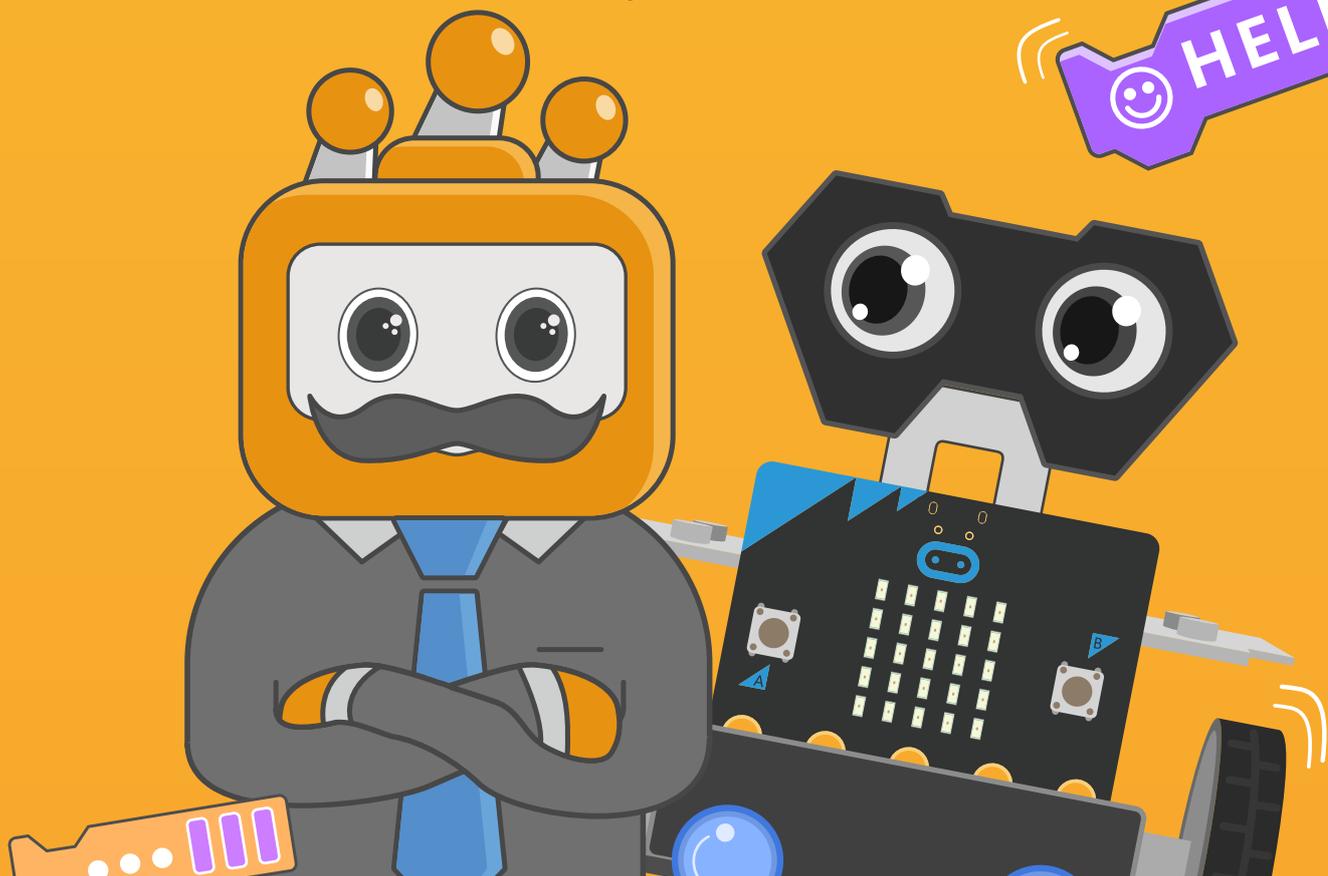


PLUS

HELLO



# 第一章

认识麦昆 Plus

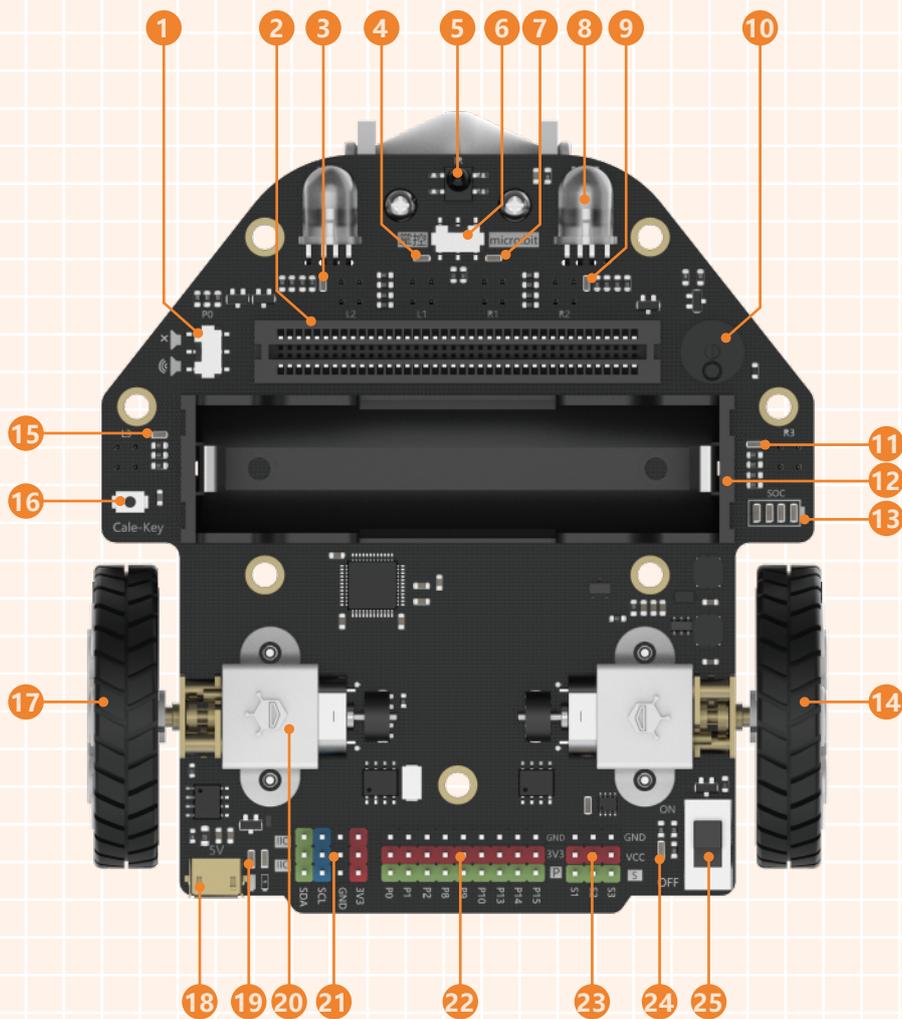


## 什么是麦昆 Plus ?

麦昆 Plus 是精心为初学者设计的一款可编程的智能教育机器人。支持 Mind+ 和 MakeCode 图形化编程平台，其编程语言和编程环境简单、易理解。可以让我们快速的使用麦昆 Plus 做出许多有趣的东西。麦昆 Plus 本身就自带了多种炫酷的技能，不仅能够带领我们走进机器人的世界，了解机器人原理，而且还可以在我们玩游戏的过程中轻松的学习编程知识。

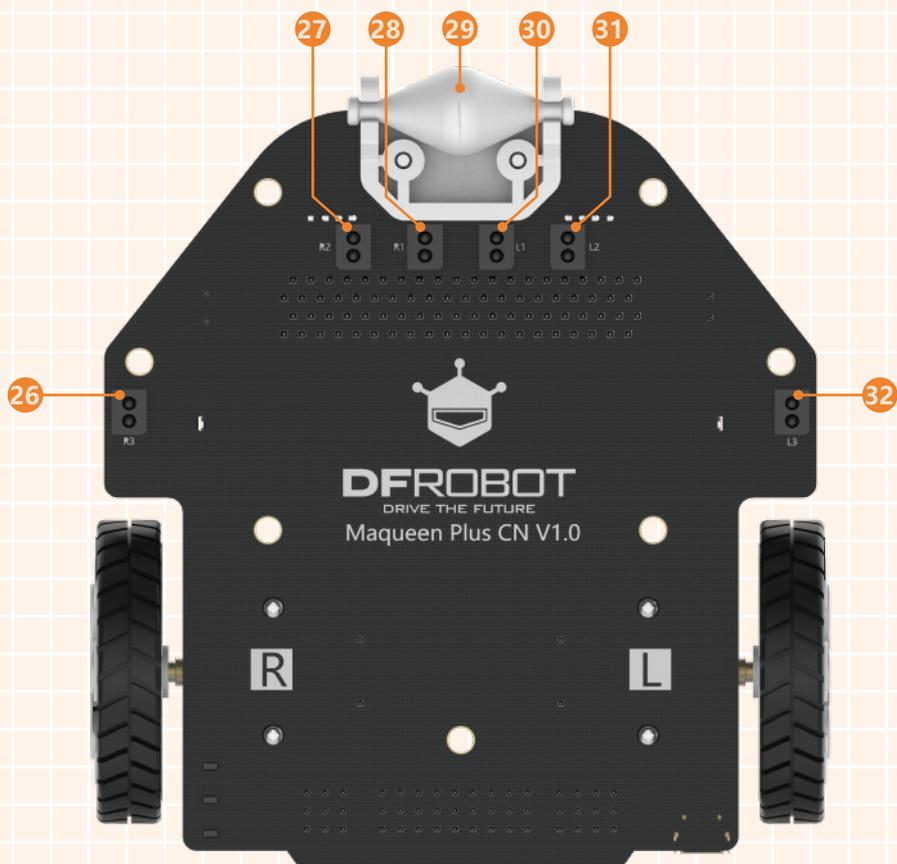
在开始学习之前，先用一张图来简单的了解一下麦昆 Plus 具有哪些功能。在后面的使用过程中，再来对各个功能做具体的了解。

Plus 主板正面指示图



- |         |               |         |          |         |          |
|---------|---------------|---------|----------|---------|----------|
| ① 声音开关  | ② micro:bit插槽 | ③ L2指示灯 | ④ L1指示灯  | ⑤ 红外接收  | ⑥ 主板切换开关 |
| ⑦ R1指示灯 | ⑧ 车灯          | ⑨ R2指示灯 | ⑩ 蜂鸣器    | ⑪ R3指示灯 | ⑫ 电池盒    |
| ⑬ 电量指示灯 | ⑭ 右轮胎         | ⑮ L3指示灯 | ⑯ 巡线复位开关 | ⑰ 左轮胎   | ⑱ 充电口    |
| ⑲ 充电指示灯 | ⑳ 金属电机        | ㉑ IIC扩展 | ㉒ 扩展口    | ㉓ 舵机扩展口 | ㉔ 电源指示灯  |
| ㉕ 电源开关  |               |         |          |         |          |

## Plus 主板背面指示图



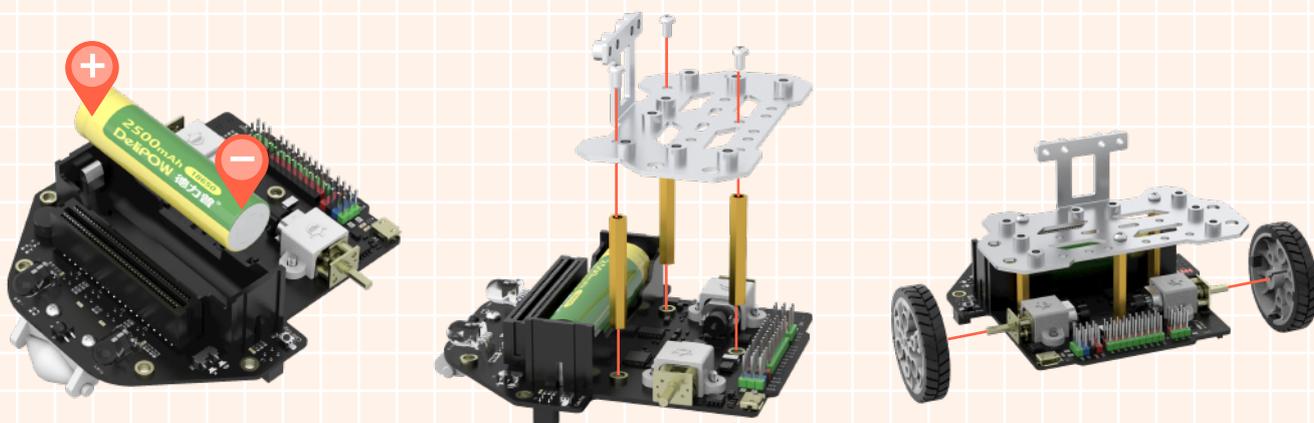
- 26 R3巡线    27 R2巡线    28 R1巡线    29 支撑轮    30 L1巡线    31 L2巡线    32 L3巡线

从图中可以看出,麦昆 Plus 具有很多的功能。那如何才能将这些功能运用起来呢? 所以接下来的学习就尤为重要了。

## 组装教程



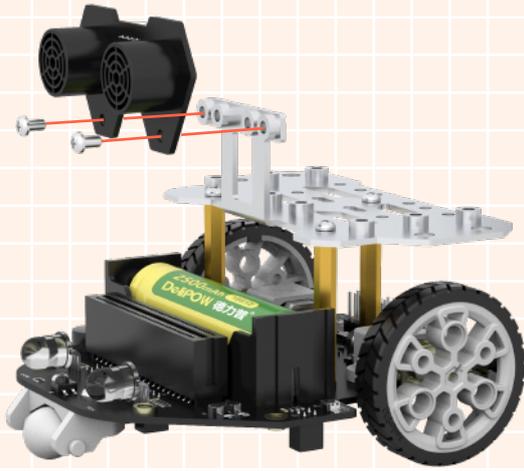
## Plus 组装指示图



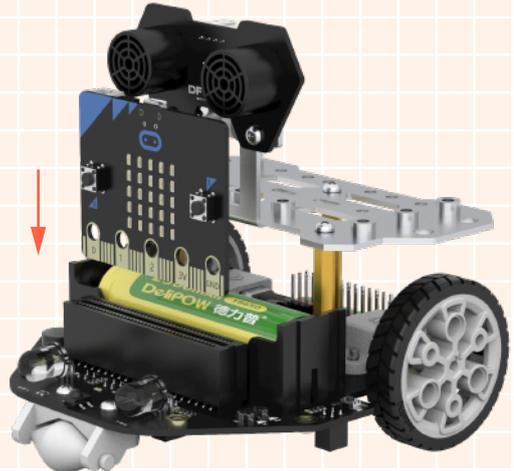
1 安装 18650 电池

2 安装支架

3 安装轮子



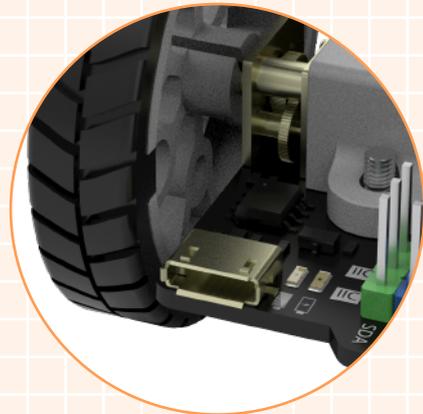
4 安装超声波



5 插上 micro:bit 主板



电量显示



充电口

注意：当电池电量充足的时候，LED 全亮，当电量逐渐降低后，LED 灯会逐个熄灭。当 LED 都熄灭后，就表示我们应该给电池充电了。

麦昆 Plus 组装完成之后，先将它放在一边。接下来我们要认识一个非常最重要的主控设备——micro:bit 主控板。micro:bit 相当于电脑的芯片，同样也是麦昆 Plus 的“大脑”，它能对数据进行处理和存储，是让麦昆 Plus“活”过来的关键。

## 认识 micro:bit 主控板



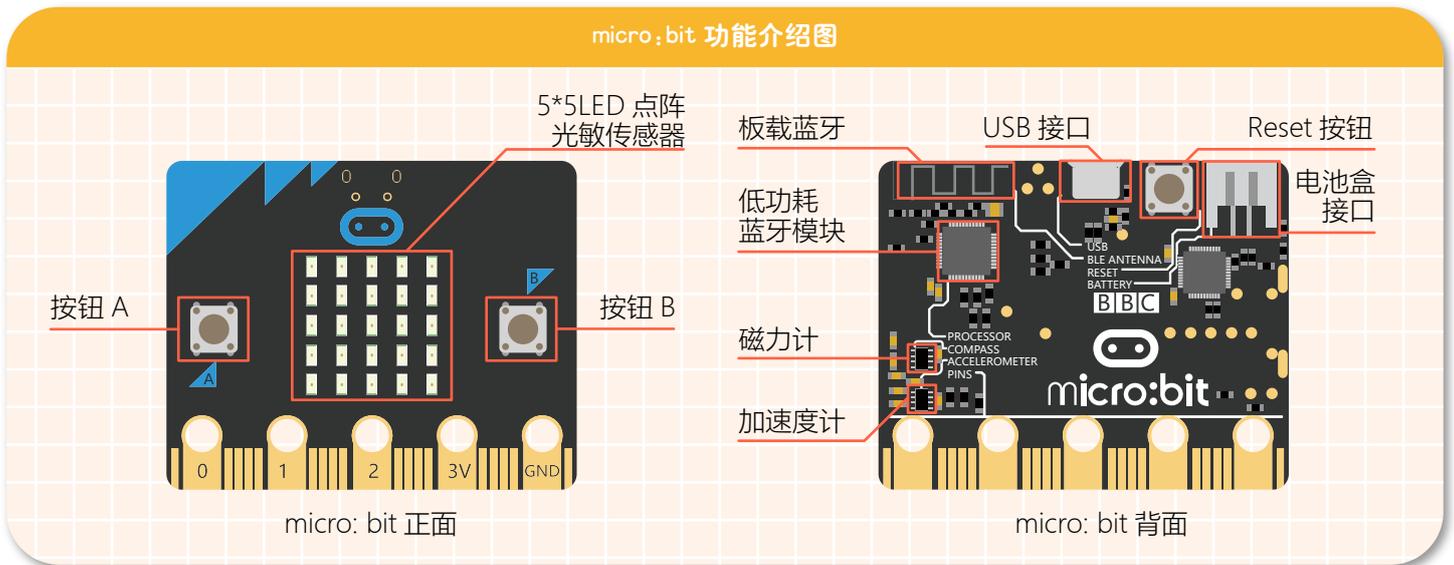
### micro:bit 能用来干什么？

micro:bit 集成了很多基本的传感器，可以使用它制作一些有趣的小发明：板载蓝牙可以和手机互联，实现通过手机发送指令控制单片机；无线通信功能可以实现多个 micro:bit 之间的远程通信，完成一些需要远距离控制的任务；电子罗盘可以感知上下左右和东西南北，利用它可以制作不会迷路的装置；micro:bit 板中间的 5×5 LED 点阵显示屏，可以创作有趣的表情包和小动画；点阵显示屏两边有两个可编程按钮，可以控制游戏操作或暂停 / 播放一首音乐……

micro:bit 还自带常见的感应装置，例如光线传感器，它在与光线有关的项目中可以派上大用处；温度传感器能感知环境的温度。因此 micro:bit 本身就可以实现很多生活中常见的智能化功能。

## micro:bit 功能介绍

micro:bit 只有半张信用卡大小(长 5cm\* 宽 4cm), 搭载的 5×5 可编程 LED 点阵、2 颗可编程按键、光敏传感器、加速度计、电子罗盘、温度计、蓝牙等电子模块。



5×5 可编程 LED 点阵	micro:bit 有 25 颗可编程 LED 灯, 可为我们显示文字, 数字和图标。
2 颗可编程按键	micro:bit 正面有 A, B 两颗可编程按钮, 可以用来触发事件。例如: 当按下 A 键, 显示爱心。
光敏传感器	用 LED 作为基本的光敏传感器, 可以用它检测环境光线。
加速度计和电子罗盘	可以测量当 micro:bit 被移动时的加速度, 能检测到一些动作。例如: 摇晃、倾斜、自由落体。
温度计	可以检测当前的环境温度。
蓝牙、无线通信	无线通信功能可以让两块及以上的 micro:bit 之间进行无线通信, 允许 micro:bit 发送和接收蓝牙信号。

## micro:bit 的编程方式

micro:bit 有很多支持的图形化编程软件, 例如: Mind+、MakeCode、python、Scratch 等编辑器编写代码。而 Mind+ 软件本身很多传感器, 只需要拖动图形化程序块就可以完成编程。

本教程主要使用 Mind+ 进行编程, 接下来我们就来学习 Mind+ 的使用方法。

## 认识 Mind+ 图形化编程



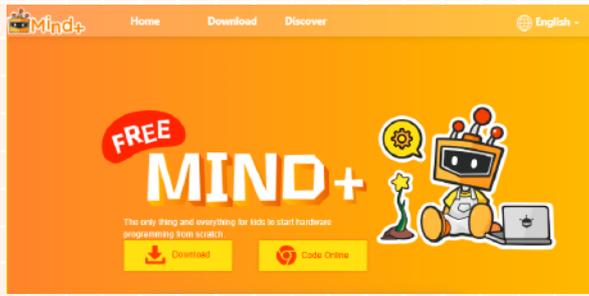
### Mind+ 介绍

Mind+ 是一款基于 Scratch3.0 开发的青少年编程软件, 支持 arduino、micro:bit、掌控板等各种开源硬件, 只需要拖动图形化程序块即可完成编程, 还可以使用 python/c/c++ 等高级编程语言, 让大家轻松体验创造的乐趣。

### 软件下载

在浏览器中输入以下网址, 进入下载页面, 点击立即下载则会跳转至 MIND+ 客户端下载页面, 选择适合自己电脑系统的版本下载即可。

下载地址: <http://mindplus.cc>



## 搭建编程环境

按照功能划分, Mind+ 界面可分为以下几个部分(上传模式):



类别	功能
菜单栏	项目：可以新建项目、打开项目、保存项目等。 教程：在这里能找到很多使用教程和示例程序哦！ 连接设备：能检测到连接的设备，显示串口，可以选择连接或是断开设备。
上传模式 / 实时模式	“上传模式 / 实时模式”按钮：切换程序执行的模式。 上传模式：将程序上传到硬件设备中执行； 实时模式：将脚本区可执行的程序在硬件和 Mind+ 舞台中实时执行。
设置区	设置：用于设置软件主题、语言、学习基本案例，在线或加群寻求帮助。
指令区	指令模块：这里包括编程需要的许多模块和指令，将不同指令按一定逻辑结合就能实现各种各样的功能。
扩展	扩展按钮：可以选择更多额外的道具，支持各种硬件编程。
编辑区	程序编辑区：组合指令进行图形化编程的区域。通过鼠标拖拽就可以将需要的指令移动到脚本区适当位置进行组合。
代码查看区	自动生成：显示图形化指令相对应的代码。 手动编辑：可自己在此区域输入代码编程。
串口区	串口区：可打开 / 关闭串口开关、滚屏开关、清出输出、波特率口、串口输入框、输出格式控制等。 黑色区域：可显示下载状况，比如可以看到程序运行状况，显示串口通信数据等。

有了以上作为基础,下面我们就要开始实际操作了。下面用一个简单的实例程序来具体了解在 Mind+ 中如何编写和上传程序。下载完成并且安装成功后,先打开软件进行如下操作,具体步骤如下:

## 1、安装驱动

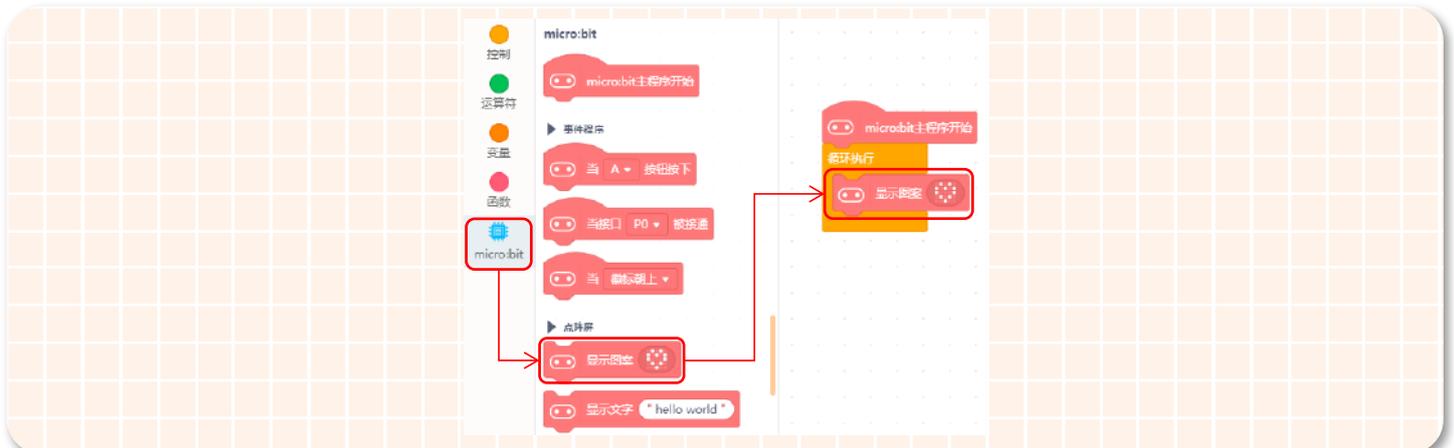


## 2、程序编写

操作一: 点击左下角“扩展”--“主控板”--“micro:bit”

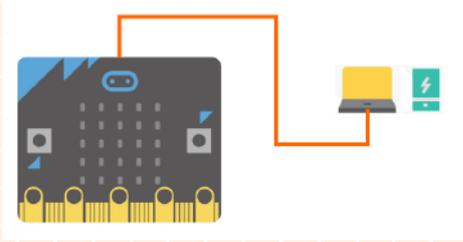


操作二: 将需要的指令模块从指令区拖放至编辑区。



### 3、上传程序到设备

操作一：通过 USB 连接线将 Micro:bit 和电脑相连接。

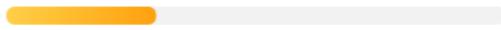


操作二：点击菜单栏上“连接设备”，找到并且点击串口号“COM-Microbit”，软件连接到设备。



编程完成后，再点击“上传到设备”（如果此时未连接设备，会提示“没有连接到设备”），上传过程中会显示上传进度，上传成功后，提示框自动消失。

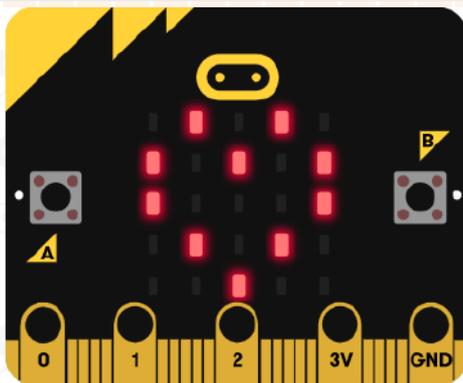
正在上传  
上传进度:30%



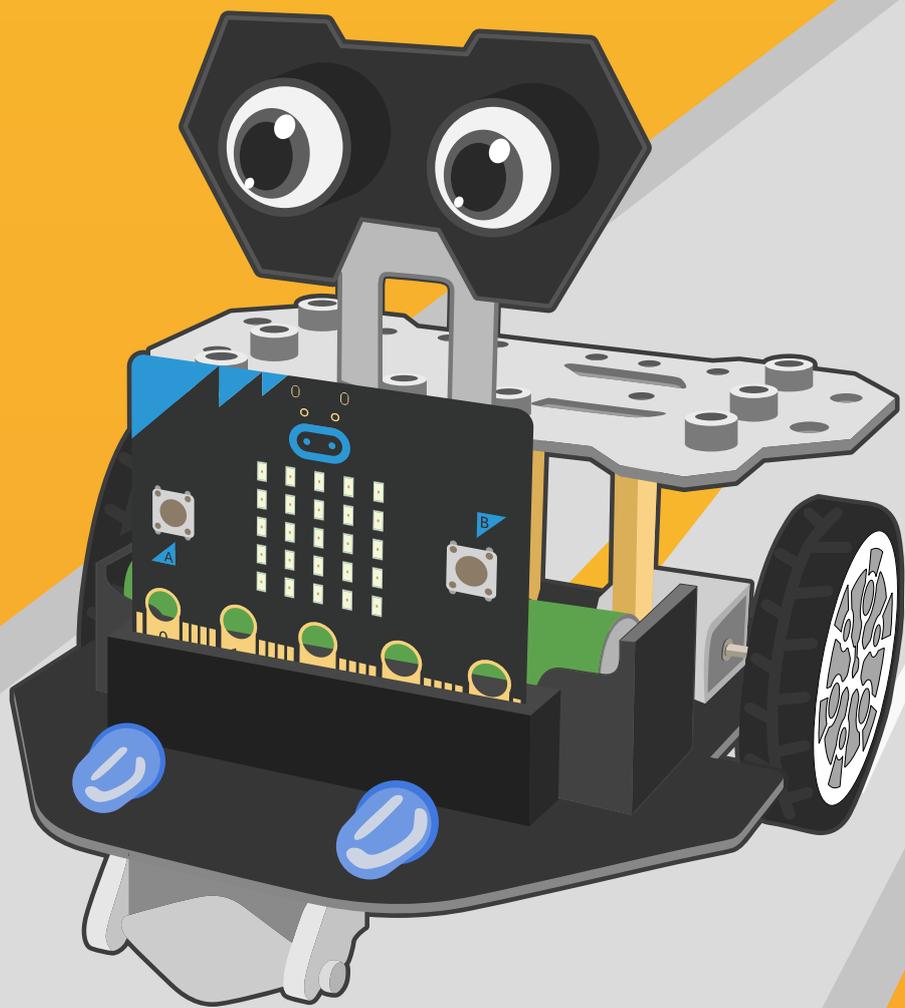
注意：当数据写入时，micro:bit 背面的电源指示灯会不停的闪烁，此时一定不能断开 USB 数据线哦。

### 4、下载完成

程序下载完成之后，不要将 USB 线与 micro:bit 和电脑断开，这个时候就可以看到我们的 micro:bit 点阵屏上显示着心形图案哦！



经过前面的学习，我们已经了解了什么是麦昆 Plus，什么是 micro:bit 以及 Mind+ 编程方式。那么接下来我们就用一些实例程序来进一步学习如何用 micro:bit 驱动麦昆 Plus。



# 第二章

## 小车动起来

酷酷的麦昆 Plus 机器人,好神奇呀!  
是不是已经迫不及待的想和麦昆 Plus 玩游戏了  
那么我们就先让麦昆 Plus 小车动起来吧!

## 学习目标



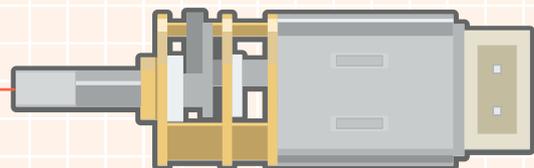
学习如何驱动电机

## 电子模块



### 电机介绍图

电机



电机的主要功能是控制麦昆 Plus 左转、右转、前进、后退。

## 指令学习



### 模块介绍图

无限循环程序模块

循环执行

把程序放在这个模块中,就会一直重复的执行模块中的程序。

电机控制模块

小车前进方向左侧车轮

小车运动方向

控制电机的速度及方向。



小车速度最高为 255

## 动手实践



### STEP 1 新建项目

打开 Mind+ 软件,点击“项目”--“新建项目”新建一个项目。



### STEP 2 添加麦昆 Plus

1、选择“扩展”--“主控板”--“micro:bit”主控板,选择成功如下图:



2、继续点击“扩展板”--“麦昆 Plus”,选择成功如下图:



麦昆 Plus 添加成功后,指令区会显示“ 扩展板”,点击扩展板,可以查看所有对应的指令模块。指令模块的具体功能见附件 1。

### STEP 3 程序编写

1、如何删除不需要的程序模块。如下图所示。



#### 知识要点

##### 如何删除模块

将鼠标指针移到到想要删除的模块上,鼠标指针变为“”; 按下鼠标左键选中该模块,鼠标指针变为“”,此时拖拽模块至指令区,当鼠标指针变为“”,松开鼠标左键,这样就可以轻松删除啦!

2、将电机控制模块拖至“循环执行”模块中。如下图所示。



3、将电机模块中的“左侧”更改为“全部”,这样麦昆 Plus 左右两侧的轮子就能以 200 的速度前进。如下图所示。



#### 知识要点

同一个模块是可以被多次拖拽使用的, 模块中带有下拉箭头“”都是可以按下进行选择的哦! 像“”是可以使用键盘输入数字或者通过拖动滑动模块来改变填入的数字!

4、完整程序如下图所示。

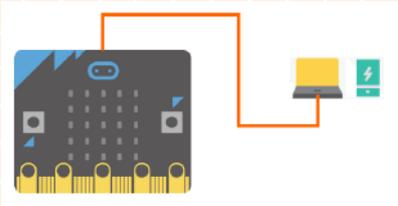


5、最终程序如上图所示,保存项目,命名改为“小车动起来”。如下图所示。



#### STEP 4 下载程序

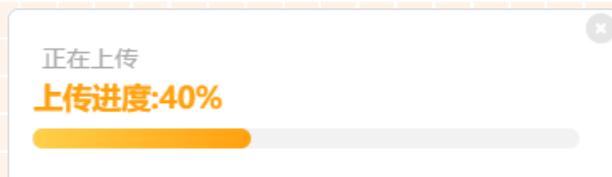
1、连接电脑: 下载程序之前,首先需要用 USB 线将 micro:bit 与电脑连接在一起。



2、下载程序: 点击菜单栏上“连接设备”,找到并且点击串口号“COM-Microbit”,软件连接到设备。

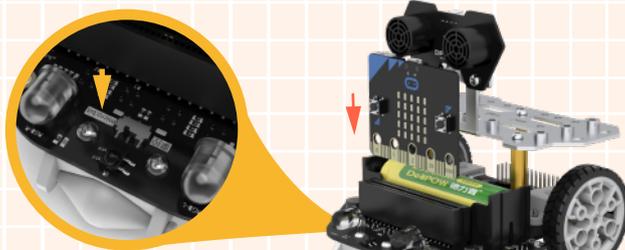


编程完成后,再点击“上传到设备”(如果此时未连接设备,会提示“没有连接到设备”),上传过程中会显示上传进度,上传成功后,提示框自动消失。



3、安装主控板: 程序下载完成后拔掉 USB 数据线,将带有程序的 micro:bit 主控板安装到麦昆 Plus 上。将 micro:bit 主控板的点阵面朝前,插入插槽,如图所示:

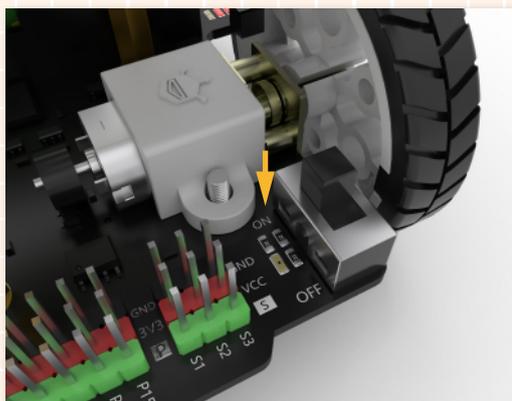
主控板安装好之后,一定要记得将前面的切换开关拨到 micro:bit 侧。



注: 国内版的麦昆 Plus 是有主控板的切换开关的,国外版没有。

## STEP 5 效果展示

完成这些步骤后,打开麦昆 Plus 的电源开关后,麦昆 Plus 就能欢快的动起来吧!



将电源开关拨到 ON  
开启麦昆 Plus

## 探究思考



前面学习了如何让麦昆 Plus 前进,可是大家想一想实际生活中,小车除了能够前进以外还可以后退,那么现在我们想让麦昆 Plus 后退该怎样来实现呢?你是否能够结合我们所学过的知识让麦昆 Plus 以 100 的速度后退呢?

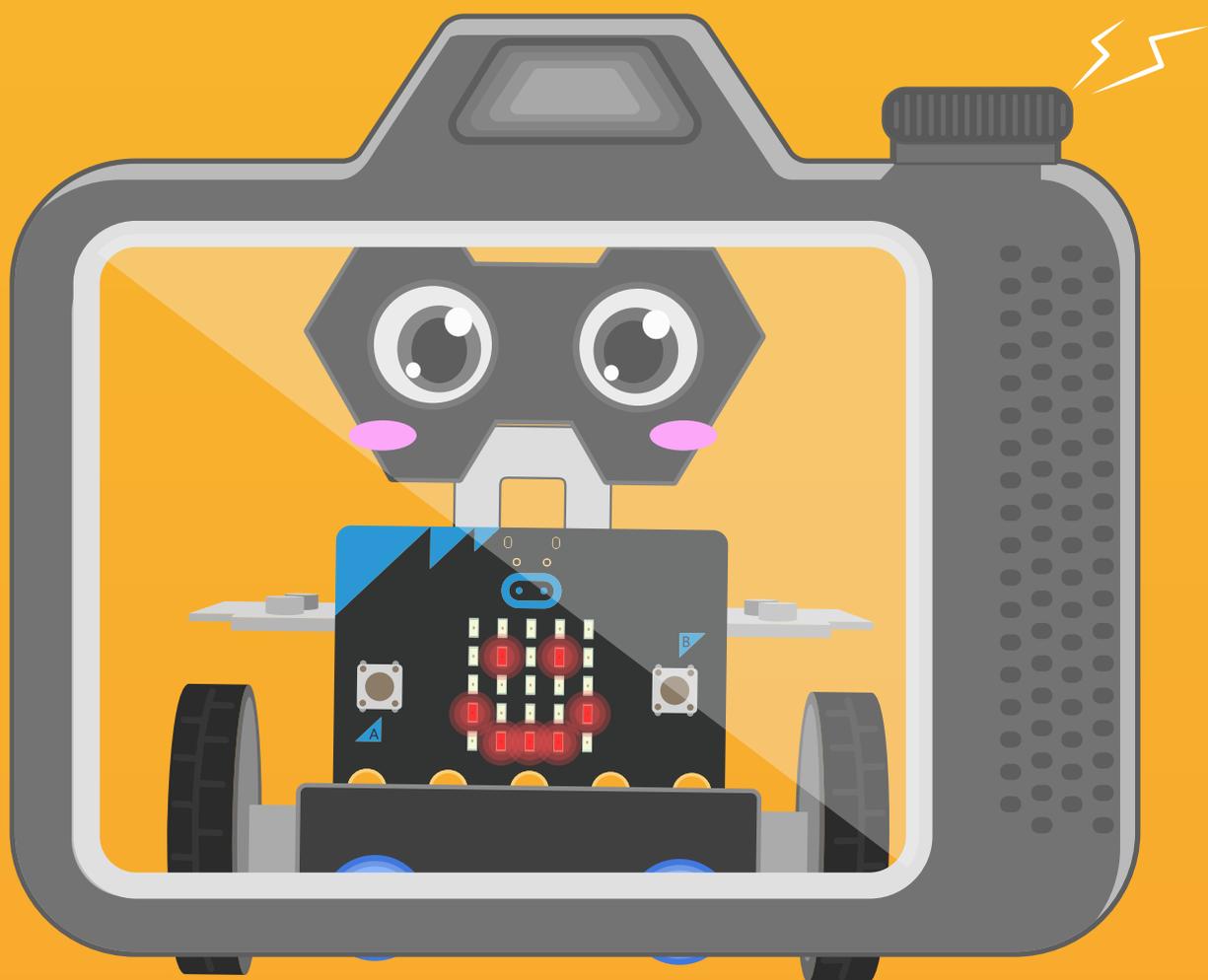
提示: 改变电机控制模块的方向和速度就可以了哦!

小车前进方向左侧车轮

小车运动方向



小车速度最高为 255



# 第三章

## 行走的表情包

麦昆 Plus 在行走的过程中, 可以变换出许多的表情, 我们将麦昆 Plus 的表情拍照截图, 效果近似于网络上流行的表情包素材, 现在麦昆 Plus 已经成为 DF 界的表情包啦。下面就来一起学习并制作一个会走圆形路线的表情包。

## 学习目标



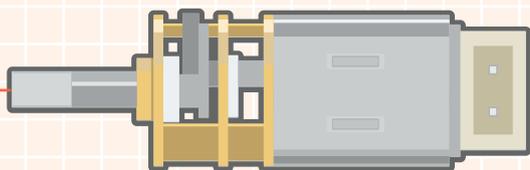
- 1、学习差速转向原理
- 2、了解延时模块的作用

## 电子模块



### 电机介绍图

电机



电机的主要功能是控制麦昆 Plus 左转、右转、前进、后退。

## 指令学习



### 模块介绍图

显示图标模块



显示图标模块可以让 micro:bit 点阵屏上显示各种各样的内置图案

电机停止控制模块



电机停止模块可以让麦昆 Plus 停止不动

延时模块



延时模块可以让程序暂停一段时间后继续执行, 暂停时间可以自己调整

## 动手实践



### STEP 1 新建项目

打开 Mind+ 软件, 点击“项目”--“新建项目”新建一个项目。



### STEP 2 添加麦昆 Plus

1、选择“扩展”--“主控板”--“micro:bit”主控板, 选择成功如下图:



2、继续点击“扩展板”--“麦昆 Plus”, 选择成功如下图:



麦昆 Plus 添加成功后,指令区会显示“



扩展板

”,点击扩展板,可以查看所有对应的指令模块。指令模块的具体功能

见附件 1。

### STEP 3 程序编写

#### 1. 显示表情

将显示图案模块拖动到如图所示位置。当主程序开始时, micro:bit 的点阵屏上就会显示心形图案。如下图所示。



注：点击 还可以选择其他的图案。

#### 2. 小车顺时针运动

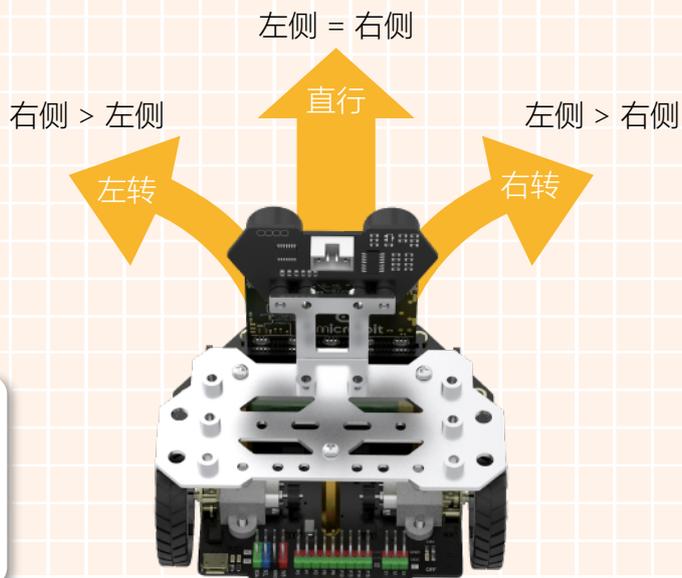
小车现在已经可以显示表情了,那接下来就应该让他顺时针转圈圈了。根据差速转向原理,设置左侧电机以 200 的速度正转; 右侧的电机以 50 的速度正转。如下图所示。



#### 知识要点

##### 差速转向原理:

当左右两轮的速度和方向一致时,小车可以前进或者后退。如果左右两轮的方向一样,速度不一样,小车是不是就能实现左右转向呢。



### 3. 小车走圆形

让小车走圆形,主要是通过延时模块设置等待时间,使得小车行进路线为一个完整的圆形。如下图所示。



#### 知识要点

注意: 小车走出一个完整的圆形路线所需的时间长短与地面的摩擦力、电池电量等因素有关, 因此, 要根据实际情况进行调整哦!

当 micro:bit 的点阵屏上显示笑脸图案时, 表示麦昆 Plus 已经走完一个圆形的路线了。如下图所示。

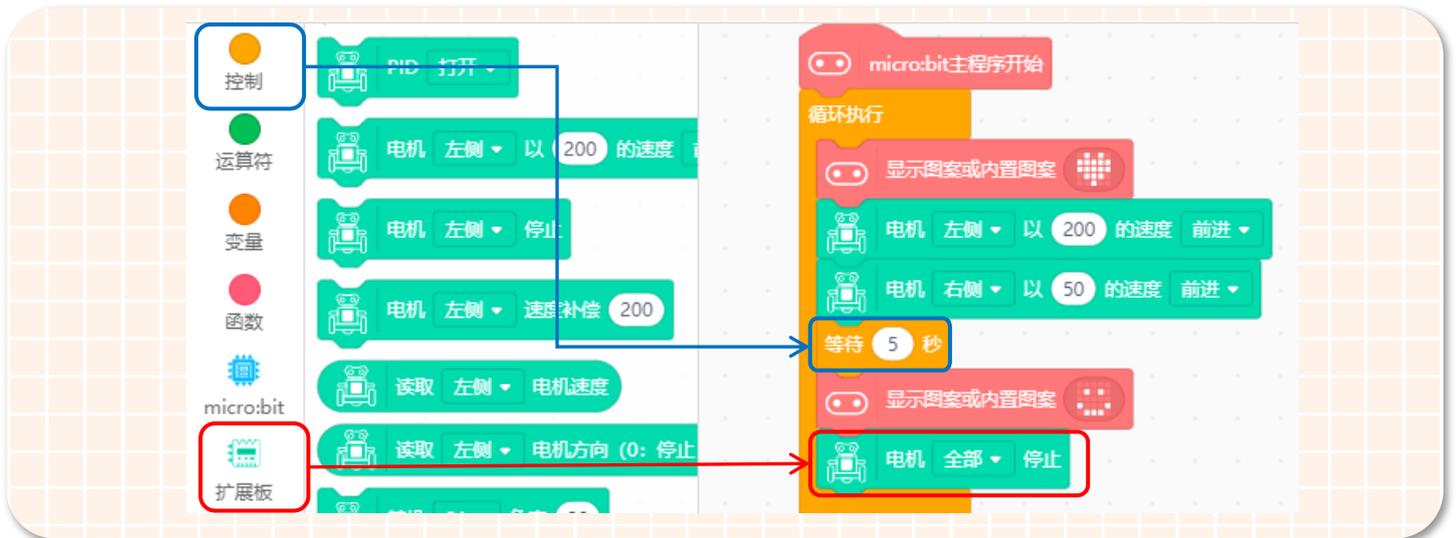


### 4. 小车停止

现在小车已经能完整的行走出一个圆形路线, 如果不想让它一直走下去, 就需要通过电机停止控制模块让小车停下来。将电机停止命令的“左侧”改为“全部”。这样左右两侧的电机会停止转动。如下图所示。



5、完整程序如下图所示。

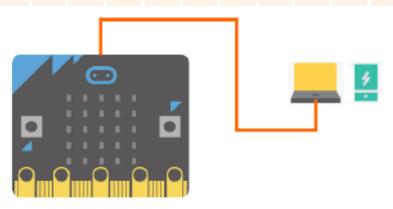


6、保存项目，命名为“行走的表情包”。如下图所示。



#### STEP 4 下载程序

1、连接电脑：下载程序之前，首先需要用 USB 线将 micro:bit 与电脑连接在一起。



2、下载程序：点击菜单栏上“连接设备”，找到并且点击串口号“COM-Microbit”，软件连接到设备。

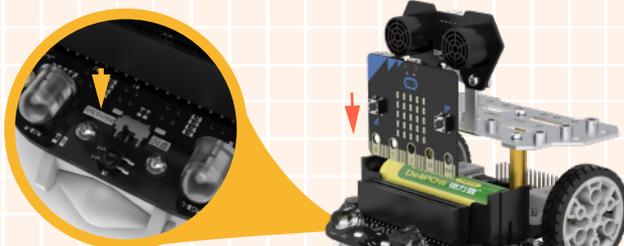


编程完成后,再点击“上传到设备”(如果此时未连接设备,会提示“没有连接到设备”),上传过程中会显示上传进度,上传成功后,提示框自动消失。



**3、安装主控板:**程序下载完成后拔掉 USB 数据线,将带有程序的 micro:bit 主控板安装到麦昆 Plus 上。将 micro:bit 主控板的点阵面朝前,插入插槽,如图所示:

主控板安装好之后,一定要记得将前面的切换开关拨到 micro:bit 侧。

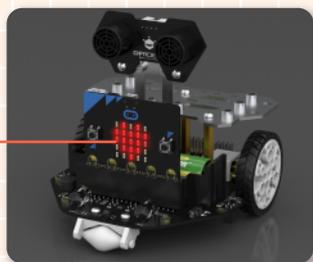


注:国内版的麦昆 Plus 是有主控板的切换开关的,国外版没有。

### STEP 5 效果展示

完成上面的步骤之后,打开电源开关,我们的麦昆 Plus 就变成行走的表情包啦!而且我们可以清楚的看到麦昆 Plus 在行走的过程中 micro:bit 显示的是心形的表情,停止行走后 micro:bit 显示笑脸的表情。

行走途中:心形



行走完成:笑脸



### 探究思考



小车的左右两个电机在不同的速度和方向下,小车的运动状态应该怎样的呢?按照第一组的例子,动手编写程序并完成下面的表格吧!提示:改变电机控制模块的方向和速度就可以了哦!

组别	左侧电机		右侧电机		小车运动状态
	速度	方向	速度	方向	
1	200	前进	200	前进	小车前进
2	200	前进	50	前进	
3	50	前进	200	前进	
4	200	后退	200	后退	
5	200	后退	200	前进	



# 第四章

## 城市守护者——警车

不知道大家有没有看过经典动画片《黑猫警长》？黑猫警长每次出警都会开着帅气的警车去抓不法分子。当警笛鸣响，警灯闪起的时候，帅气的黑猫警长就变得更加高大了。那么现在就来制作一款属于自己的城市守护者——警车吧！

## 学习目标



1. 学习蜂鸣器模块的使用
2. 学习 RGB 灯模块的使用

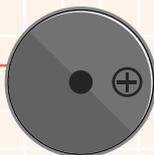
## 电子模块



### 蜂鸣器、RGB 灯介绍图

蜂鸣器

蜂鸣器,做为输出模块,可以用来播放音乐



RGB 灯

RGB 灯,可以设置灯珠颜色和灯光效果

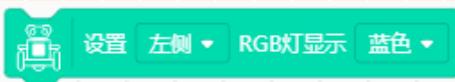


## 指令学习



### 模块介绍图

RGB 灯控制模块



控制 RGB 灯的颜色

播放旋律模块



调用 makecode 内置的音乐

## 动手实践



### STEP 1 新建项目

打开 Mind+ 软件,点击“项目”--“新建项目”新建一个项目。



### STEP 2 添加麦昆 Plus

1、选择“扩展”--“主控板”--“micro:bit”主控板,选择成功如下图:



2、继续点击“扩展板”--“麦昆 Plus”,选择成功如下图:



麦昆 Plus 添加成功后,指令区会显示“ 扩展板”,点击扩展板,可以查看所有对应的指令模块。指令模块的具体功能见附件 1。

### STEP 3 程序编写

#### 1、警车行驶的程序编写

设置警车以 200 速度前进,并将电机的“左侧”更改为“全部”,如下图所示。



#### 2、警灯的程序编写

首先我们让麦昆 Plus 的左侧 RGB 灯显示红色,右侧的 RGB 灯显示蓝色。如下图所示。



因为警车的警灯是交替显示的,所以接下来我们需要将麦昆 Plus 的左侧 RGB 灯显示蓝色,右侧的 RGB 灯显示红色。如下图所示。



#### 知识要点

大家知道为什么警灯是红色和蓝色的吗?

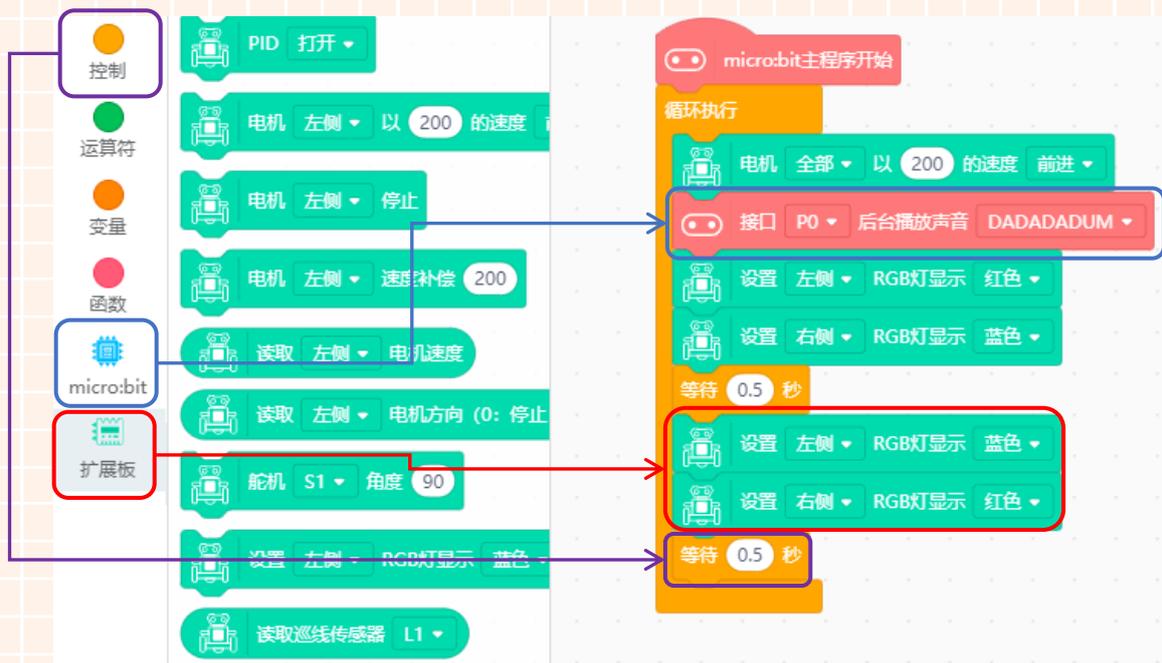
1. 从警灯的作用上来说,警灯的主要目的是为了告诫行人让道,提醒人们注意和警觉,以防发生事故。
2. 红色是暖色系中的原色,具有警示作用,蓝色是冷色系中的原色,代表戒备与秩序。

### 3、警笛的程序编写

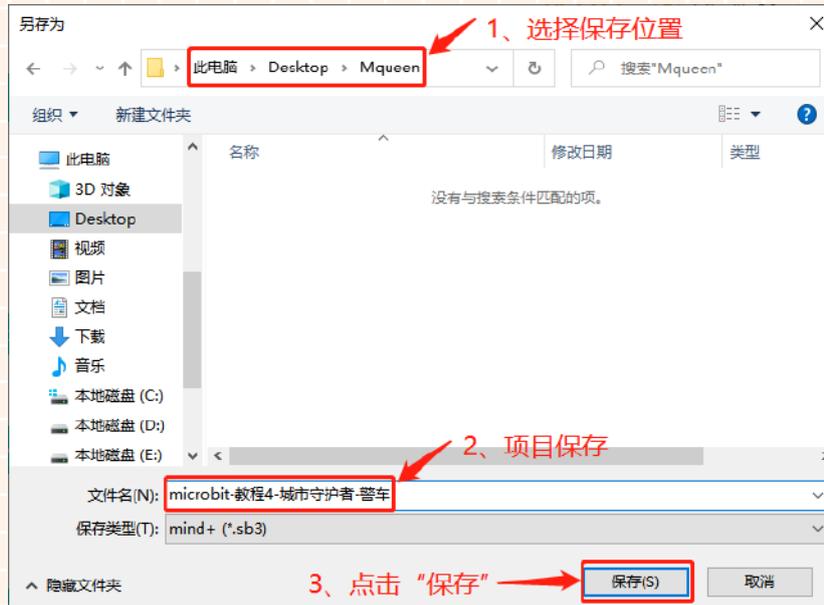
这里我们用“播放旋律”模块中的内置旋律“dadadum”来模拟警笛的声音。如下图所示。



### 4、完整程序如下图所示。

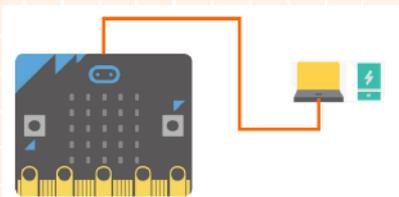


### 5、保存项目，命名为“城市守护者—警车”。如下图所示。



## STEP 4 下载程序

**1、连接电脑：**下载程序之前，首先需要用 USB 线将 micro:bit 与电脑连接在一起。



**2、下载程序：**点击菜单栏上“连接设备”，找到并且点击串口号“COM-Microbit”，软件连接到设备。

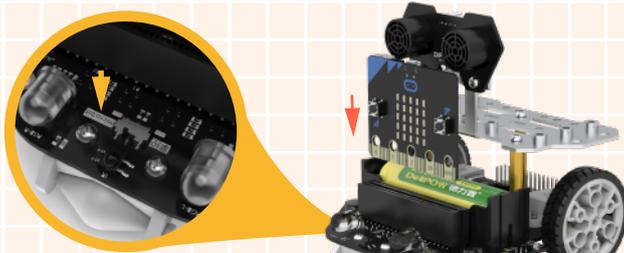


编程完成后，再点击“上传到设备”（如果此时未连接设备，会提示“没有连接到设备”），上传过程中会显示上传进度，上传成功后，提示框自动消失。

正在上传  
上传进度:40%

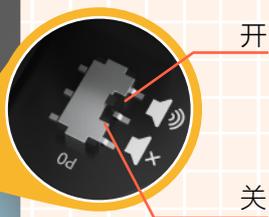
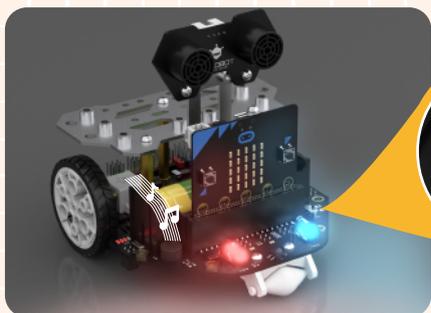
**3、安装主控板：**程序下载完成后拔掉 USB 数据线，将带有程序的 micro:bit 主控板安装到麦昆 Plus 上。将 micro:bit 主控板的点阵面朝前，插入插槽，如图所示：

主控板安装好之后，一定要记得将前面的切换开关拨到 micro:bit 侧。



注：国内版的麦昆 Plus 是有主控板的切换开关的，国外版没有。

## STEP 5 效果展示



注意：

1. 在麦昆 Plus 的左侧，设置了蜂鸣器开关，在使用蜂鸣器的时候需要打开蜂鸣器开关。
2. 蜂鸣器和 P0 口共用的，如果在后面的使用过程中需要用 P0 口作为输入输出时，只需要将蜂鸣器开关关闭就可以了。

## 探究思考



上面我们学习了如何用 makecode 内置的音乐模块来模拟警车的警笛声音，那你想不想自己制作美妙的旋律呢？如果想让麦昆 Plus 也能帮你实现哦。主要用如下图所示的“播放音调”模块来实现。

单击此处会出现一个键盘，用来选择音调

选择音调的节拍

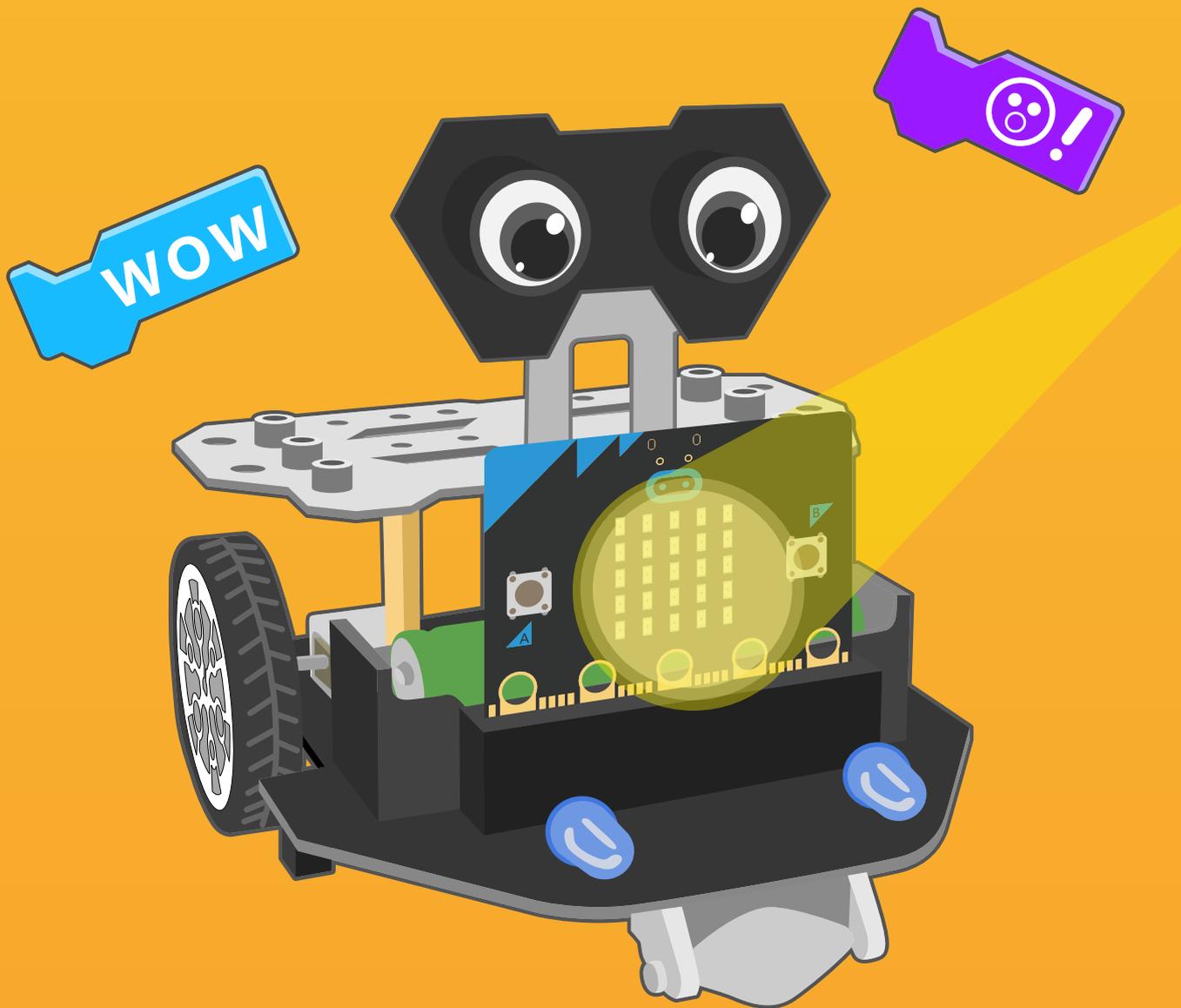


“C”是音符 do(即音符 1), 字母前面的“中”是音符的音高, 即“中”表示中音, 那么“中 C”就表示中音 do; “1”是节拍, 节拍关系到曲子的快慢和节奏。音调与简谱的对应关系如下图:

C	D	E	F	G	A	B
1	2	3	4	5	6	7

下面我们就根据曲谱, 自己制作一首简单的歌曲吧!

1=C 1 1 5 5 6 6 5— 4 4 3 3 2 2 1—  
一 闪 一 闪 亮 晶 晶, 满 天 都 是 小 星 星,



# 第五章

## 感光机器人

人类所看见的物体，是光线照射在该物体上，经过反射和折射后的光线就会形成我们眼中的图案。其实眼睛是非常灵敏的光接收器，不论光线太强还是太弱对眼睛都是有伤害的，光线太强的时候会刺眼，我们会通过遮挡眼睛和遮挡光线的方式来保护眼睛。光线太弱的时候会看不清东西，我们会通过调亮灯光的方式来保护眼睛。那么我们如何才能知道当前光线的亮度值变化呢？这就需要麦昆 Plus 来帮忙了。

## 学习目标



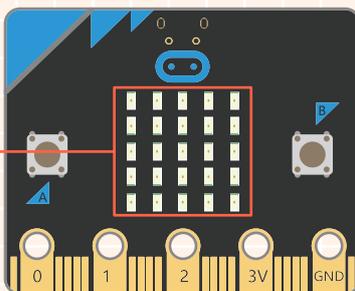
1. 学习光线传感器
2. 运用数字显示模块

## 电子模块



### 光线传感器介绍图

光线传感器



Micro:bit 的 LED 点阵屏具有感光的元件,它可以感知光线的亮度值,可以直接将光线的亮度值转化为电信号输出。

## 指令学习



### 模块介绍图

读取光线的亮度值模块

读取环境光强度

读取光线的亮度值,范围为“0~255”,表示光的亮度由弱至强

显示文字模块

显示文字 hello world

显示文字模块是将文字或数字显示在 micro:bit 点阵屏上

## 动手实践



### STEP 1 新建项目

1. 打开 Mind+ 软件,点击“项目”--“新建项目”新建一个项目。



2. 选择“扩展”--“主控板”--“micro:bit”主控板, 选择成功如下图:



3. 继续点击“扩展板”--“麦昆 Plus”, 选择成功如下图:



麦昆 Plus 添加成功后, 指令区会显示“ 扩展板”。

## STEP 2 程序编写

1. 将显示模块放入无限循环模块中, 这样感光机器人的眼睛就可以一直读取光线的亮度值啦。如下图所示。

注: 显示文字模块默认显示“hello world”。

2. 因为 micro:bit 的点阵屏上会实时显示当前环境光亮度的值, 所以需要将“读取环境光强度”模块放入到显示模块中。如下图所示。

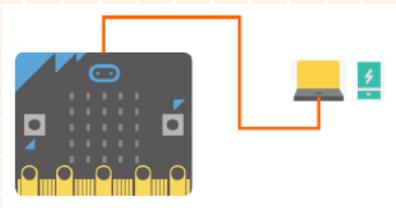
3. 完整程序如下图所示。

4. 保存项目, 在项目命名区将项目名称改为“感光机器人”。



### STEP 3 下载程序

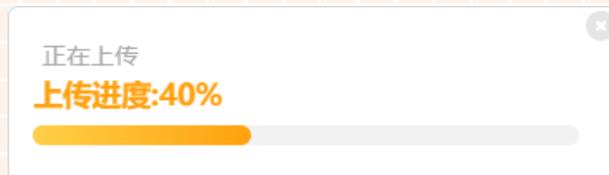
**1、连接电脑：**下载程序之前，首先需要用 USB 线将 micro:bit 与电脑连接在一起。



**2、下载程序：**点击菜单栏上“连接设备”，找到并且点击串口号“COM+Microbit”，软件连接到设备。

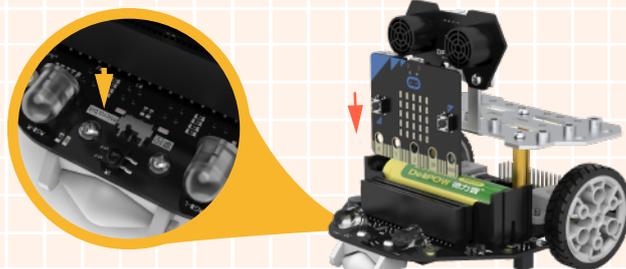


编程完成后，再点击“上传到设备”（如果此时未连接设备，会提示“没有连接到设备”），上传过程中会显示上传进度，上传成功后，提示框自动消失。



**3、安装主控板：**程序下载完成后拔掉 USB 数据线，将带有程序的 micro:bit 主控板安装到麦昆 Plus 上。将 micro:bit 主控板的点阵面朝前，插入插槽，如图所示：

主控板安装好之后，一定要记得将前面的切换开关拨到 micro:bit 侧。



注：国内版的麦昆 Plus 是有主控板的切换开关的，国外版没有。

## STEP 4 效果展示

上面的步骤都完成后,打开麦昆 Plus 的电源开关,这时候就可以通过 micro:bit 的点阵屏观察到当前光线的亮度值大小啦!下面我对办公室和摄影棚这两个不同的地点进行了光线的亮度值测试。

地点	光线亮度值
办公室	125
摄影棚	255

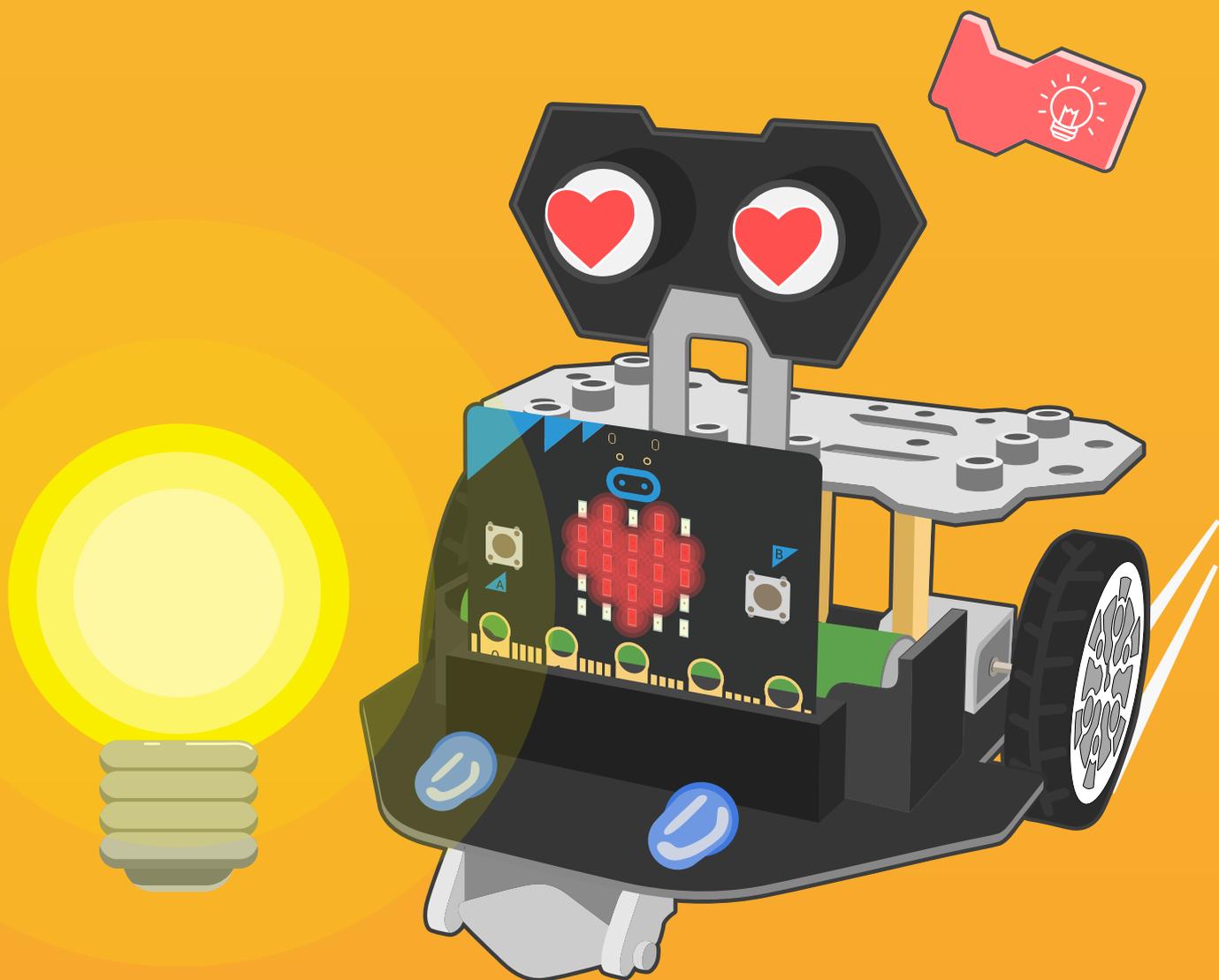
注:实验数据仅供参考。

## 探究思考



在生活中,不同的生活地点对应的光线亮度也是不相同的,那么下面就去探究一下这些地点所对应的具体光线亮度是多少吧!思考,每个环境的光线亮度设计合理吗?

地点	光线亮度值
厨房	
客厅	
卧室	
书房	



# 第六章

## 飞蛾机器人

大家听说过飞蛾扑火的故事吗？可是为什么飞蛾会朝着有火光的地方飞呢？经过科学家们长期观察和实验，发现飞蛾扑火是一种“天文导航”。飞蛾在夜间飞行时，依靠月光来判定方向。飞蛾看到灯光，将灯光误认为月光。因此，飞蛾是在用这个假“月光”来进行方向辨别的。下面就来利用麦昆 Plus 可以感知光线亮度的“眼睛”来制作一只飞蛾机器人吧！

## 学习目标



1. 学习条件判断模块
2. 运用流程图辅助编程

## 指令学习



### 模块介绍图

#### 双分支判断模块

当满足条件时,执行语句 1;  
如果不满足,执行语句 2。



#### 比较运算符">"模块

比较左右两边的大小



## 动手实践



### STEP 1 新建项目

打开 Mind+ 软件,点击“项目”--“新建项目”新建一个项目。



### STEP 2 添加麦昆 Plus

1、选择“扩展”--“主控板”--“micro:bit”主控板,选择成功如下图:



2、继续点击“扩展板”--“麦昆 Plus”,选择成功如下图:



麦昆 Plus 添加成功后,指令区会显示“

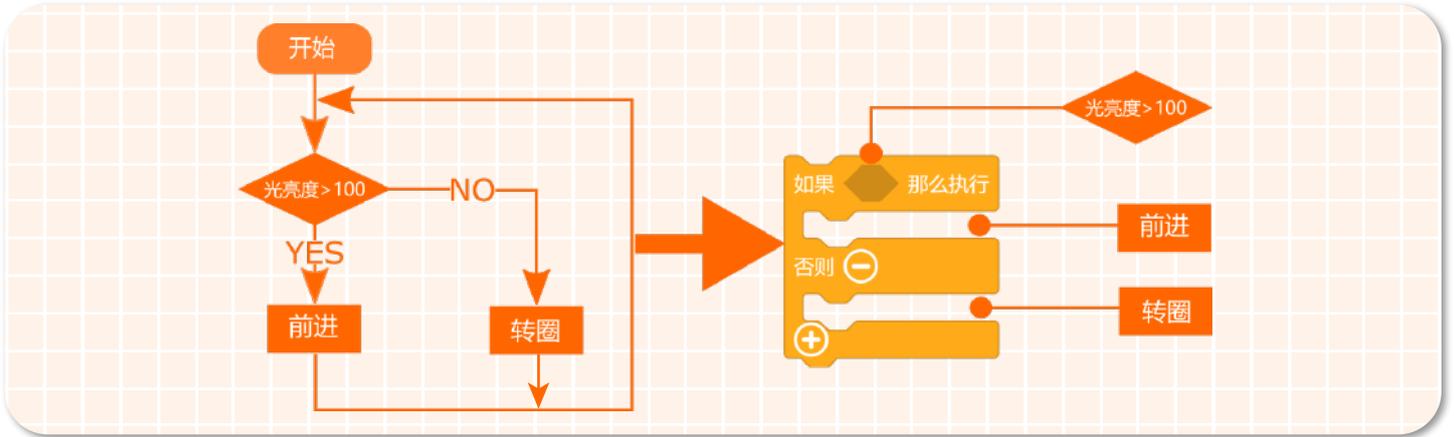


”,点击扩展板,可以查看所有对应的指令模块。指令模块的具体功能

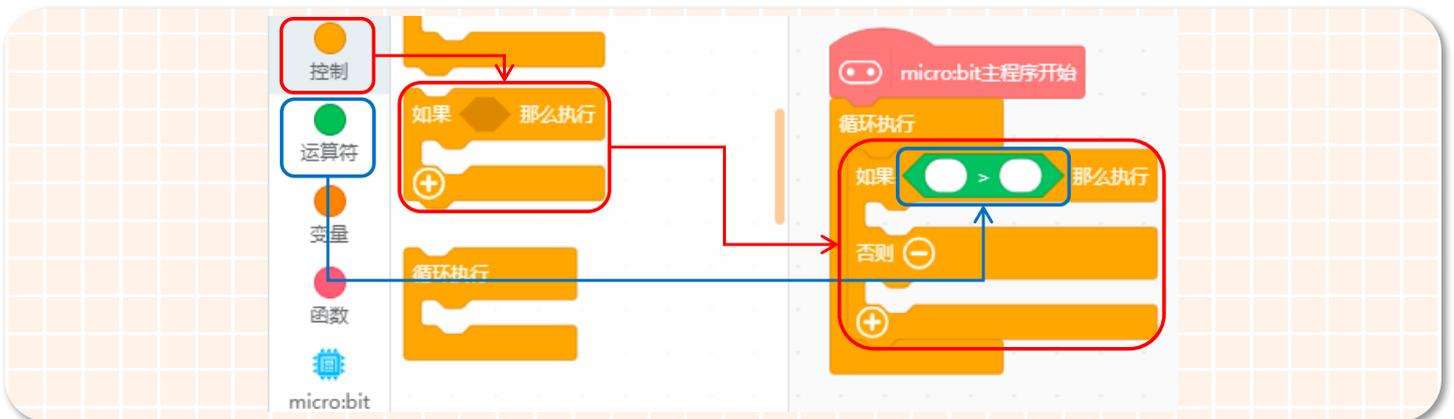
见附件 1。

### STEP 3 程序编写

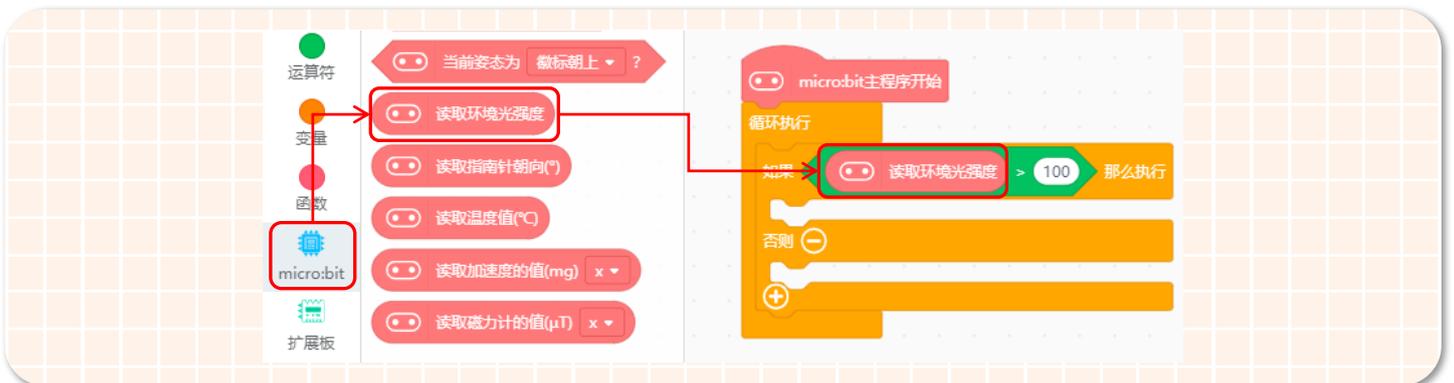
1、当光的亮度达到一定值(样例中为 100)的时候,飞蛾机器人就可以向光的方向前进;当光的亮度小于一定值(100)的时候,飞蛾机器人就在原地旋转。根据上面所描述的功能就可以画出对应的流程图来帮助编程哦!如下图所示。



2、将条件判断模块放入循环执行中,这样就可以实时处理条件判断模块了。如下图所示。

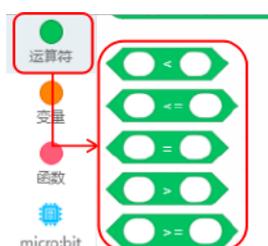


3、判定条件语句“读取环境光强度 > 100”,是整个程序的关键,在程序执行的过程中判断该条件是否成立,根据判断的结果执行不同的操作。如下图所示。



#### 知识要点

1、条件判断模块中的“”,可以按下添加条件判断的分支语句,“”可以按下删除分支语句。添加分支条件后的模块如下图:

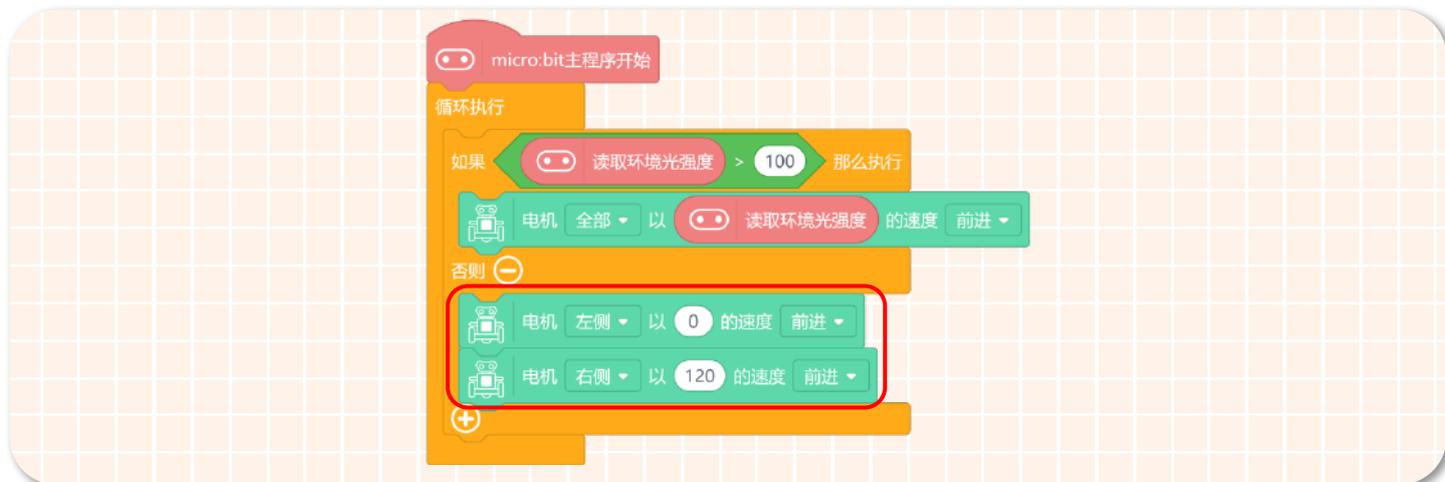


2、点击运算符,可以选择不同的比较运算符。

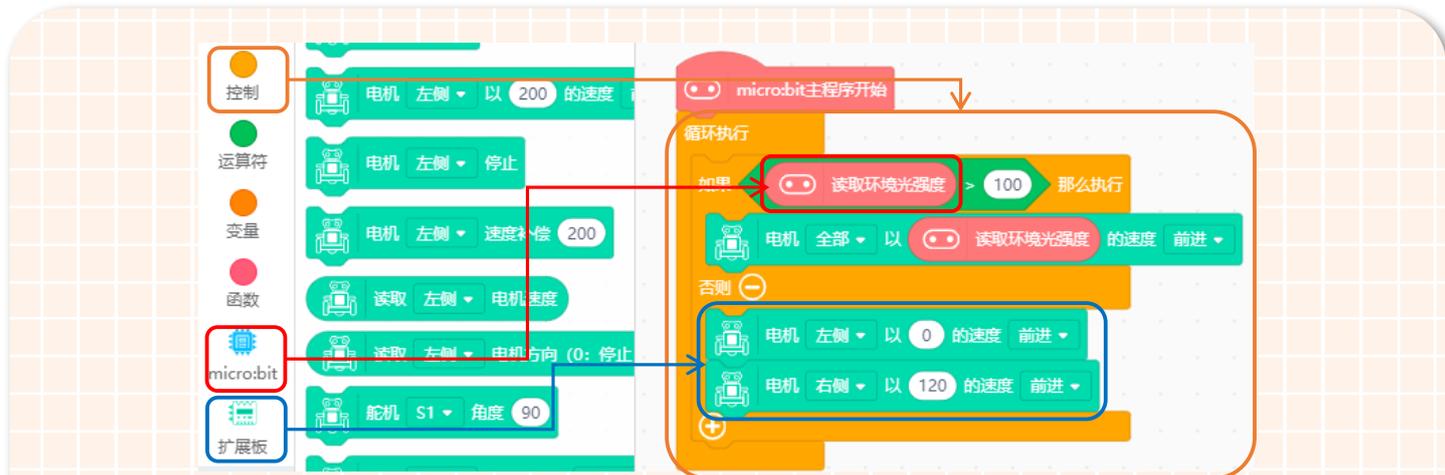
4、当“读取环境光强度 >100”这个判定条件成立时,执行前进语句并且设置前进的速度为当前的亮度级别。如下图所示。



5、当“读取环境光强度 >100”这个判定条件不成立时,就执行转圈语句。如下图所示。



## 6. 完整程序



### 知识要点

判定条件“  ”

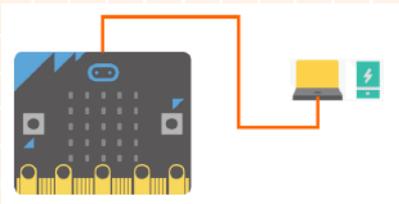
需要设置一个合适的光线亮度临界值,如果临界值太大,则需要很强的光线飞蛾机器人才能向前运动;如果临界值太小,飞蛾机器人就会一直向前运动,不会执行原地转圈的程序。所以需要大家根据所在的环境,选择合适的临界值。

7. 保存,在命名保存区将项目名称改为“飞蛾机器人”。



## STEP 4 下载程序

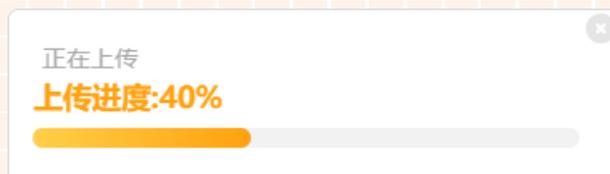
**1、连接电脑：**下载程序之前，首先需要用 USB 线将 micro:bit 与电脑连接在一起。



**2、下载程序：**点击菜单栏上“连接设备”，找到并且点击串口号“COM-Microbit”，软件连接到设备。

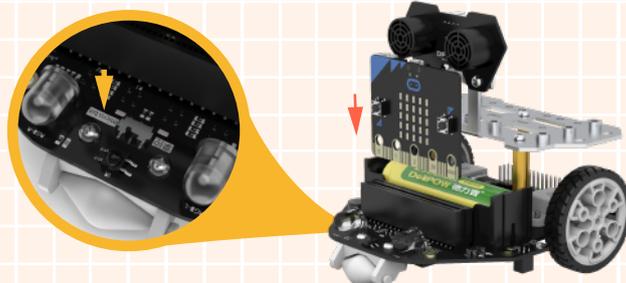


编程完成后，再点击“上传到设备”（如果此时未连接设备，会提示“没有连接到设备”），上传过程中会显示上传进度，上传成功后，提示框自动消失。



**3、安装主控板：**程序下载完成后拔掉 USB 数据线，将带有程序的 micro:bit 主控板安装到麦昆 Plus 上。将 micro:bit 主控板的点阵面朝前，插入插槽，如图所示：

主控板安装好之后，一定要记得将前面的切换开关拨到 micro:bit 侧。



注：国内版的麦昆 Plus 是有主控板的切换开关的，国外版没有。

## STEP 5 效果展示

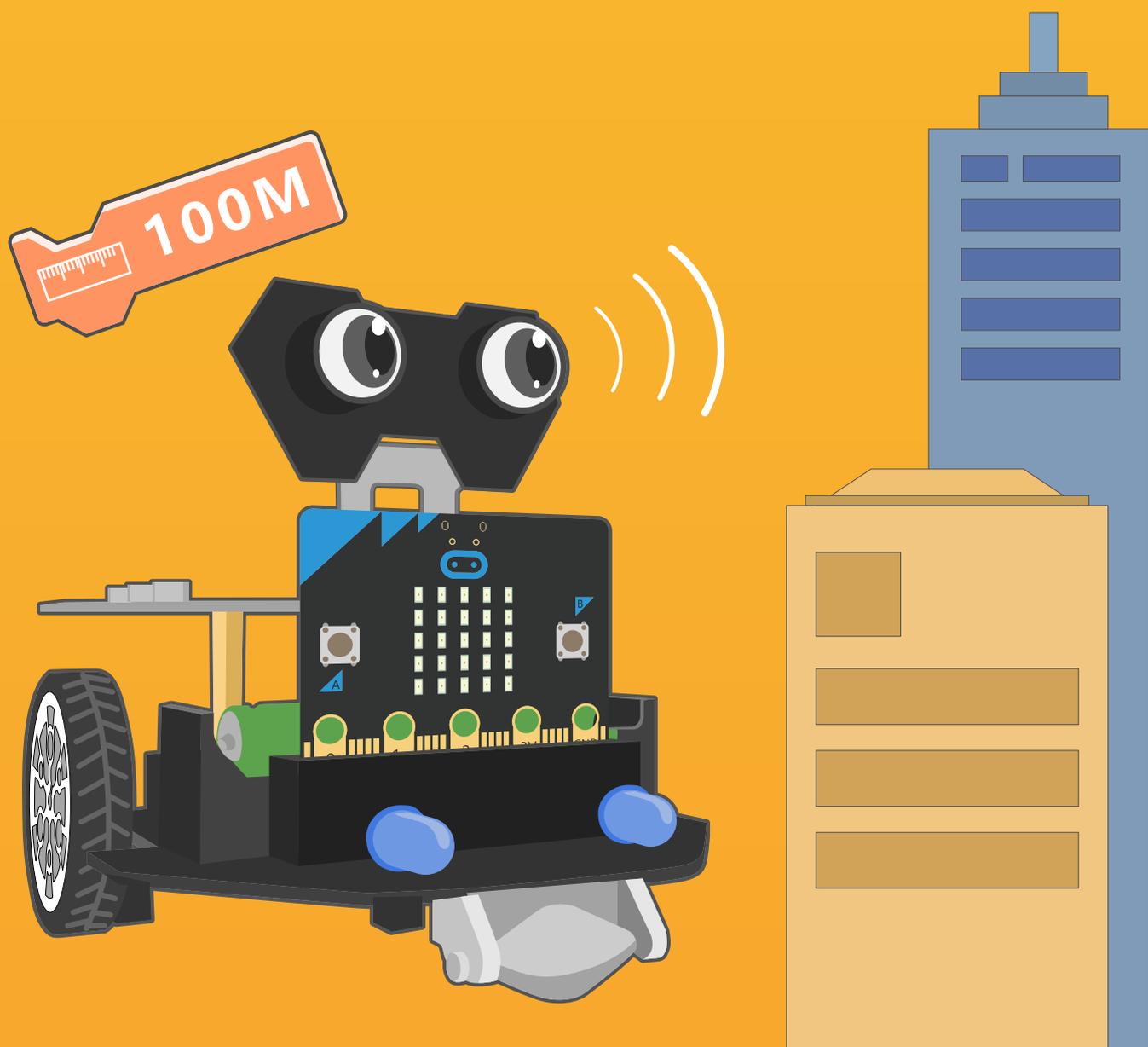
完成上面的步骤后,打开飞蛾机器人的电源开关,这时候我们的麦昆 Plus 就变成了一只向着光前进的飞蛾机器人了。当光的亮度值大于 100 时,飞蛾机器人前进的速度随着亮度值的增加而变快;当亮度值小于 100 时,飞蛾机器人就失去辨别的方向,所以只能原地转圈。是不是很有趣,快来一起和飞蛾机器人玩游戏吧!

### 探究思考



一起玩飞蛾赛跑吧!和小伙伴比一比看谁的飞蛾机器人更厉害。比赛规则:用手电筒引导飞蛾机器人前进,从起点出发绕过前方的障碍物到达终点,用时最短者获胜。不许犯规哦!

提示:如果飞蛾机器人在转圈的过程中速度很快,那么就很难控制飞蛾机器人听我们的指挥。所以调节转圈的速度是让飞蛾机器人听话的关键哦!



# 第七章

## 测距小达人

当我们还在拿着尺子测量距离的时候，测距小达人已经可以用它的“大眼睛”——超声波来帮助我们测量距离了。

## 学习目标



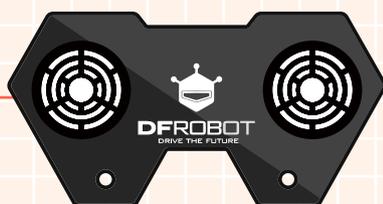
- 1、了解什么是超声波
- 2、学习超声波工作原理

## 电子模块



### 超声波介绍图

超声波传感器



超声波的发射端发射处超声波,超声波遇到障碍物反射回接收端。

## 指令学习



### 模块介绍图

读取超声波传感器模块



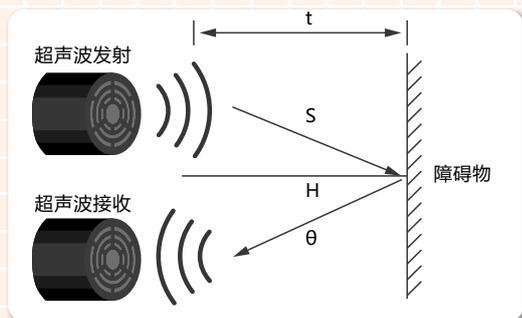
读取超声波传感器测量的距离值,需要设定好超声波传感器上发射端和接收端。距离单位为 cm/inch。

## 什么是超声波呢?

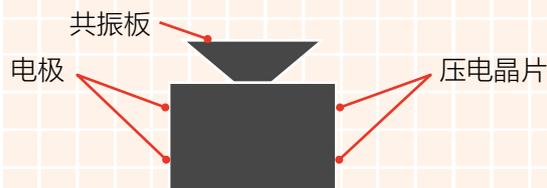
我们知道,当物体振动时会发出声音。科学家们将每秒钟振动的次数称为声音的频率,它的单位是赫兹。我们耳朵能听得到的声波频率为 20Hz-20000Hz。当声波的振动频率大于 20000Hz 或小于 20Hz 时,便听不见了。因此,把高于 20000Hz 的声波称为“超声波”。

## 超声波如何测距?

超声波之所以能够测距,是因为超声波发射器向某一方向发射超声波,在发射时刻的同时开始计时,超声波在空气中传播时碰到障碍物就立即返回来,超声波接收器收到反射波就立即停止计时。



超声波传感器在把接收到的声波信号转换成电信号,一般常用的超声波传感器由压电晶片组成的。内部结构如下图所示。

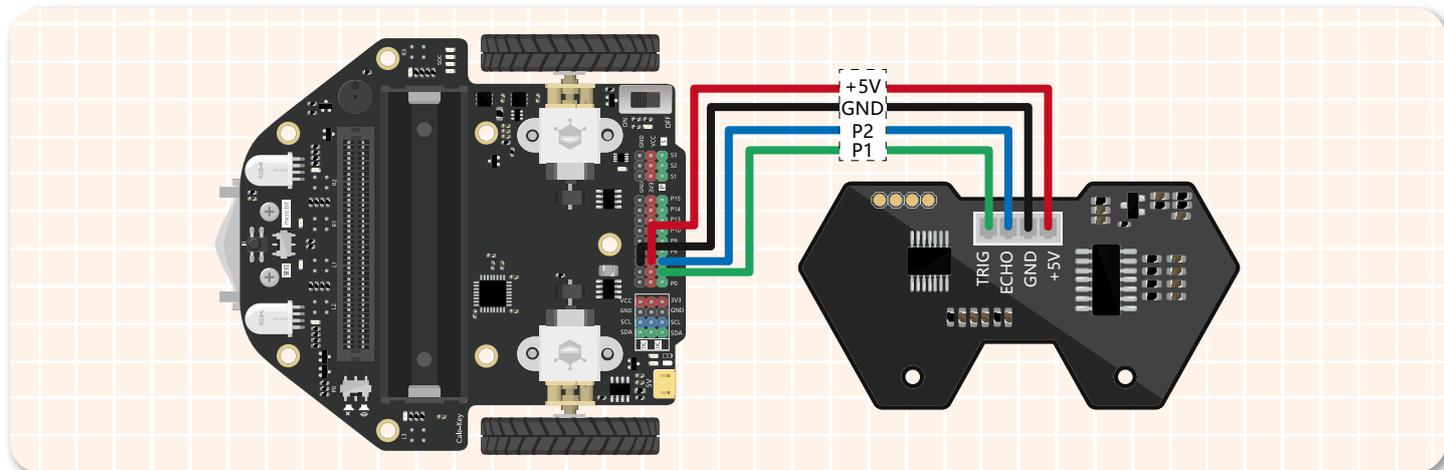


## 动手实践



### 硬件连接

超声波引脚连线方式如下图：



### STEP 1 新建项目

打开 Mind+ 软件, 点击“项目”--“新建项目”新建一个项目。



### STEP 2 添加麦昆 Plus

1、选择“扩展”--“主控板”--“micro:bit”主控板, 选择成功如下图：



2、继续点击“扩展板”--“麦昆 Plus”, 选择成功如下图：

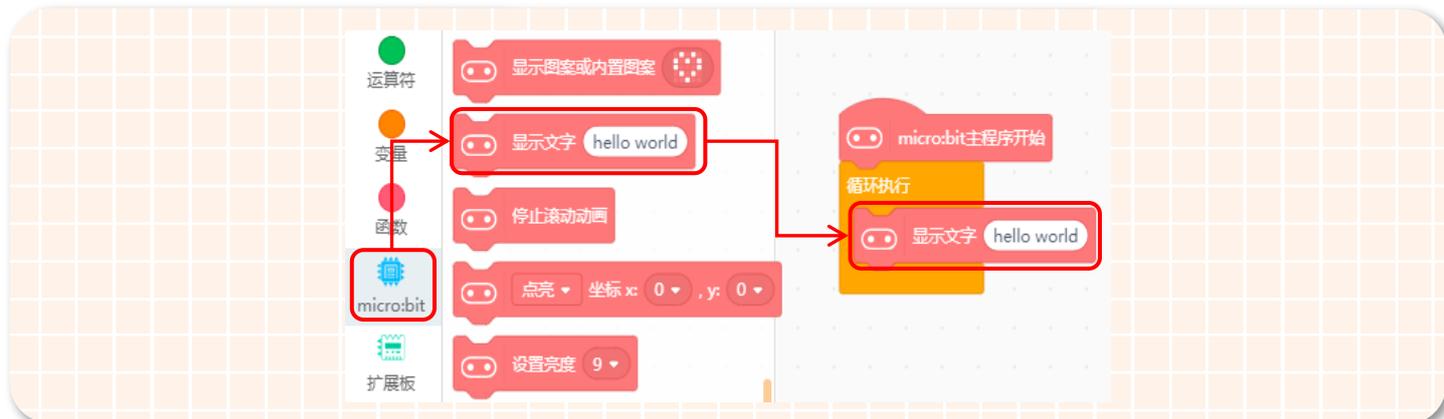


麦昆 Plus 添加成功后, 指令区会显示“



### STEP 3 程序编写

1、将显示文字模块放入在循环执行模块中, 这样测距小达人的“大眼睛”就可以一直测量距离了。如下图所示。



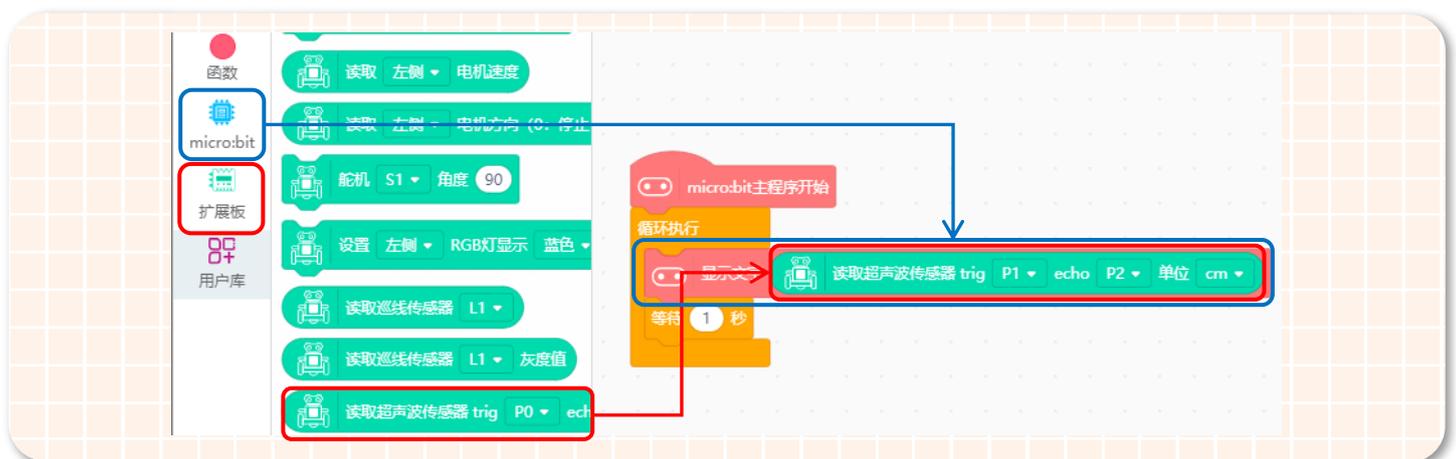
2、因为 micro:bit 的点阵屏上会实时显示当前超声波测量的距离值，所以需要将读取到的距离值放入到显示模块中，为防止读值过快，需要加上延时模块延长显示时间。如下图所示。



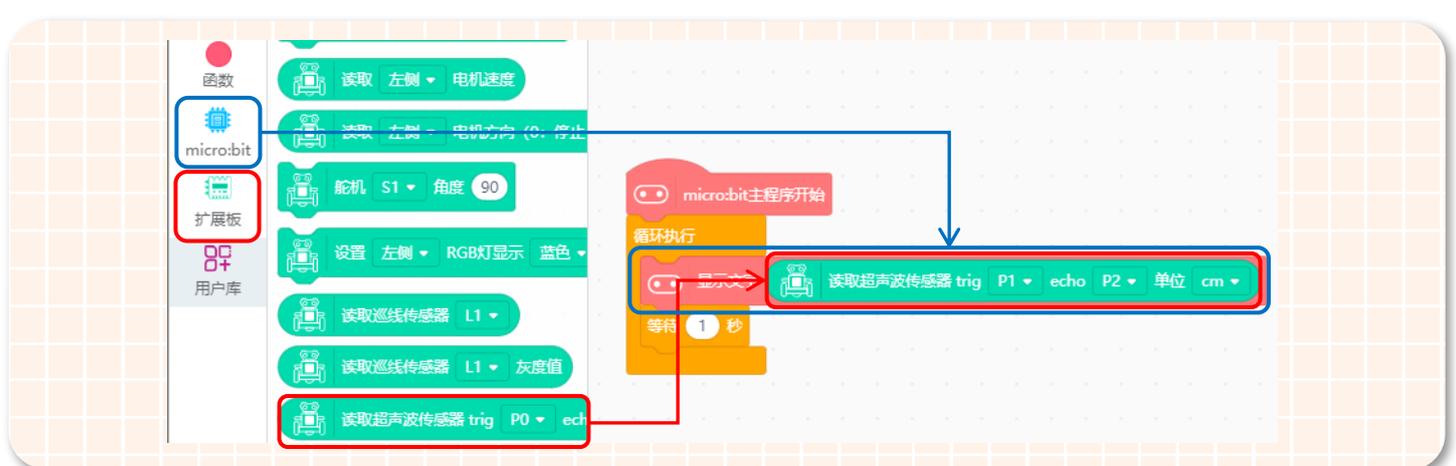
**知识要点**

“T”是超声波控制端，在超声波传感器上对应的“trig”；“E”是超声波的返回端，在超声波超感器上对应的“echo”。在麦昆 Plus 上，“trig”对应的是“P1”引脚，“echo”对应的是“P2”引脚，所以我们需要将读取超声波传感器模块中的引脚口设置成对应的“P1”、“P2”。

3、完整程序如下图所示。

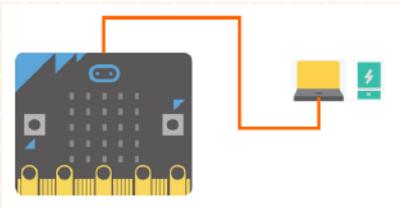


4、保存项目，命名为“测距小达人”。



## STEP 4 下载程序

**1、连接电脑：**下载程序之前，首先需要用 USB 线将 micro:bit 与电脑连接在一起。



**2、下载程序：**点击菜单栏上“连接设备”，找到并且点击串口号“COM-Microbit”，软件连接到设备。

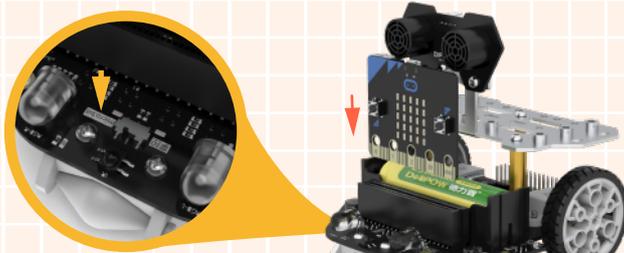


编程完成后，再点击“上传到设备”（如果此时未连接设备，会提示“没有连接到设备”），上传过程中会显示上传进度，上传成功后，提示框自动消失。

正在上传  
上传进度:40%

**3、安装主控板：**程序下载完成后拔掉 USB 数据线，将带有程序的 micro:bit 主控板安装到麦昆 Plus 上。将 micro:bit 主控板的点阵面朝前，插入插槽，如图所示：

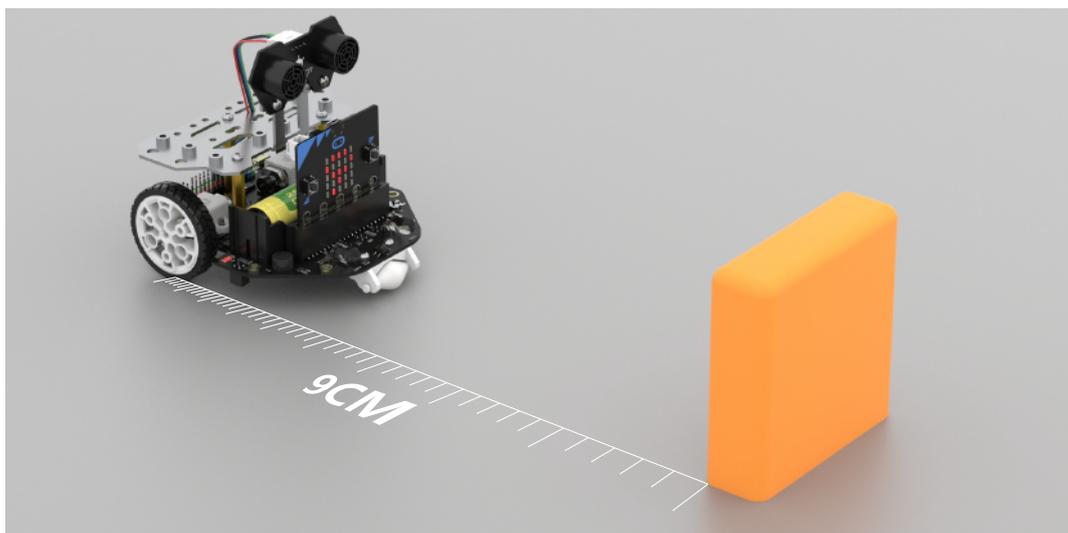
主控板安装好之后，一定要记得将前面的切换开关拨到 micro:bit 侧。



注：国内版的麦昆 Plus 是有主控板的切换开关的，国外版没有。

## STEP 4 效果展示

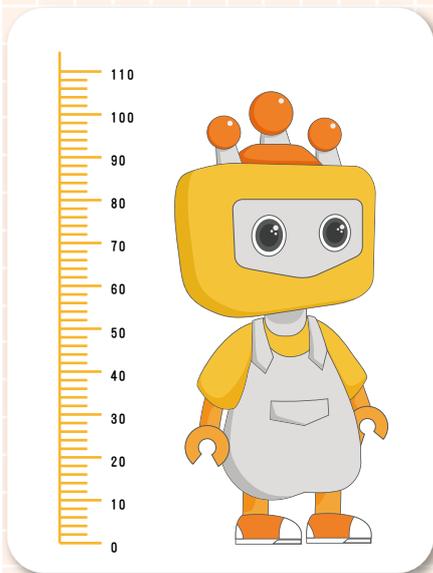
完成了上面的操作后，开启电源开关，就可以拿着测距小达人去测量距离了。测出来的距离会在 micro:bit 上显示出来哦！



## 探究思考



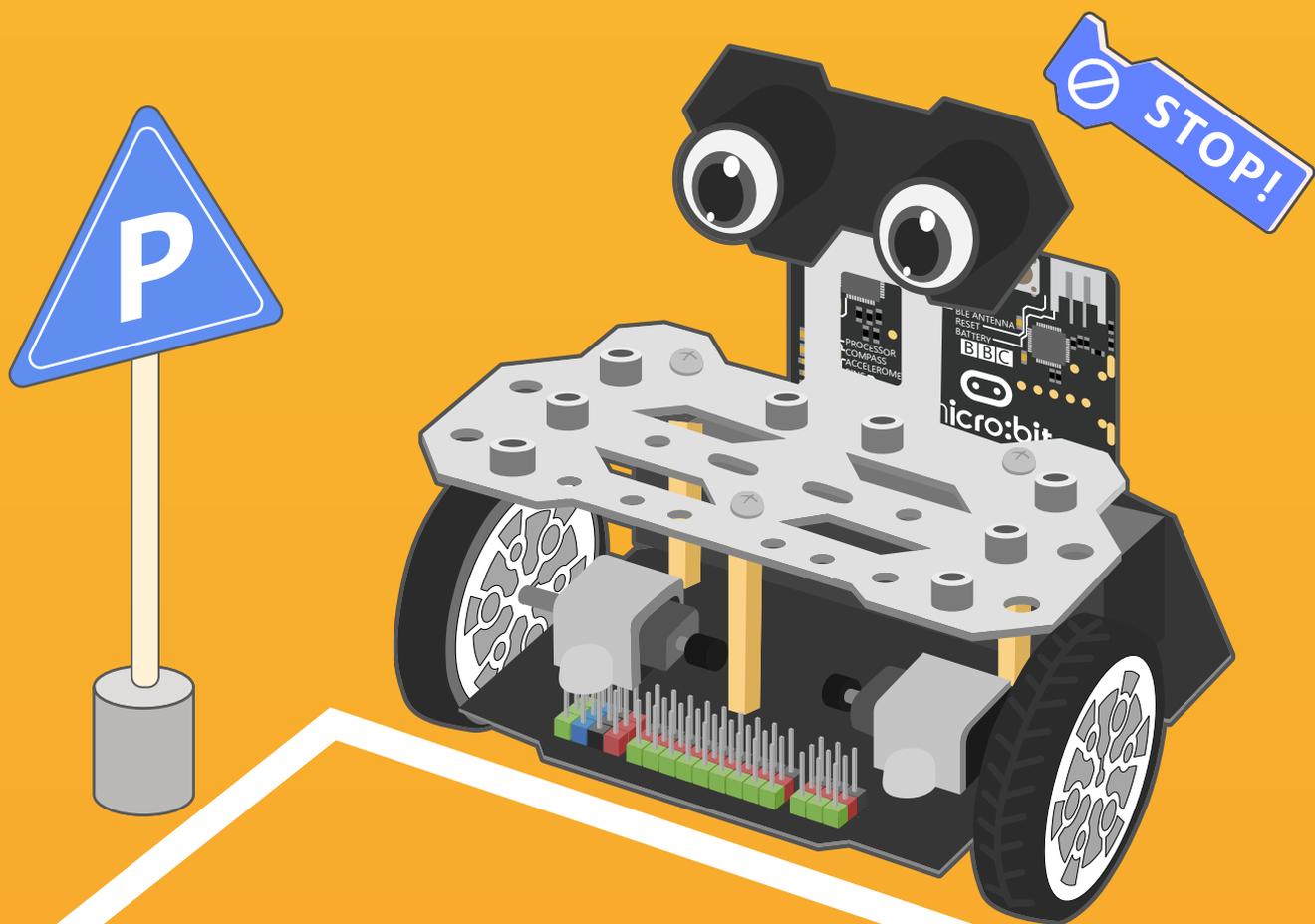
怎样利用测距小达人来测量身高呢？



想要获得精确的身高,你可能需要注意:

固定超声波的位置和方向

在 10cm 范围进行校准,防止出现大的偏差



# 第八章

## 倒车机器人

随着科技的发展，现在很多汽车上都装有倒车雷达。倒车时，倒车雷达利用超声波原理，计算出车体与障碍物之间的距离，然后提示给司机，从而协助司机进行倒车操作。倒车雷达的出现降低了司机在倒车时的风险，对一些新手司机来说也是非常的便利。倒车雷达究竟是怎样实现的呢？下面就用麦昆 Plus 结合超声波传感器制作出一个能实现倒车雷达功能的“倒车机器人”吧！

## 学习目标



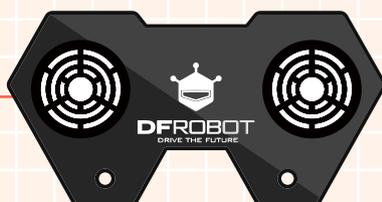
1. 学习使用变量模块
2. 学习如何在条件模块中嵌套一个条件模块

## 电子模块



### 超声波介绍图

超声波传感器



超声波的发射端发射处超声波,超声波遇到障碍物反射回接收端。

## 指令学习



### 模块介绍图

变量模块

变量 distance

表示各种变化的量,可自由创建数字变量、字符串变量、列表。

变量赋值模块

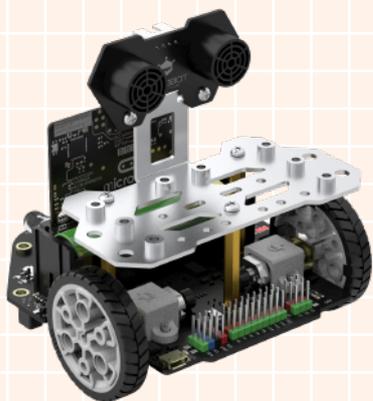
设置 distance 的值为 0

给变量 distance 赋值,设置变量成功后,模块将默认为: distance=0。

## 动手实践



### 反装超声波



在倒车机器人这个项目中,需要超声波传感器实时检测机器人尾部与障碍物之间的距离。所以下面我们就对机器人做一些小改造,将超声波拆下来反着安装在扩展支架上。

## STEP 1 新建项目

打开 Mind+ 软件,点击“项目”--“新建项目”新建一个项目。



## STEP 2 添加麦昆 Plus

1、选择“扩展”--“主控板”--“micro:bit”主控板,选择成功如下图:



2、继续点击“扩展板”--“麦昆 Plus”,选择成功如下图:



麦昆 Plus 添加成功后,指令区会显示“ 扩展板”。

## STEP 3 程序编写

1、新建一个名为“distance”的变量。如下图所示。



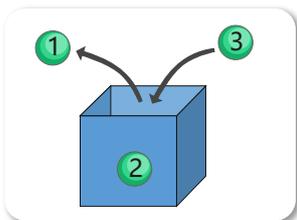
①在指令区用鼠标左键点击“变量”;

③输入新变量名“distance”;

②在指令区右边的变量下点击“新建数字类型变量”;

④变量名输入完成后,点击“确定”就可以了;

### 知识要点



#### 什么是变量

变量是指在某一变化过程中,可以取不同数值的量。

在这里,我们可以将变量理解成一个用来存储数据的盒子,盒子里面可以暂时存放各种各样的数据。比如我们用变量来存放整数,装了1之后,再装2的话,1会被2替换掉,去盒子里只会拿到2这个数字。装入的新数据会替换盒子里面的旧数据。

2、在倒车机器人项目中,我们需要实时检测超声波传感器测量的距离值,将超声波传感器测量的距离值赋值给变量“distance”。如下图所示。

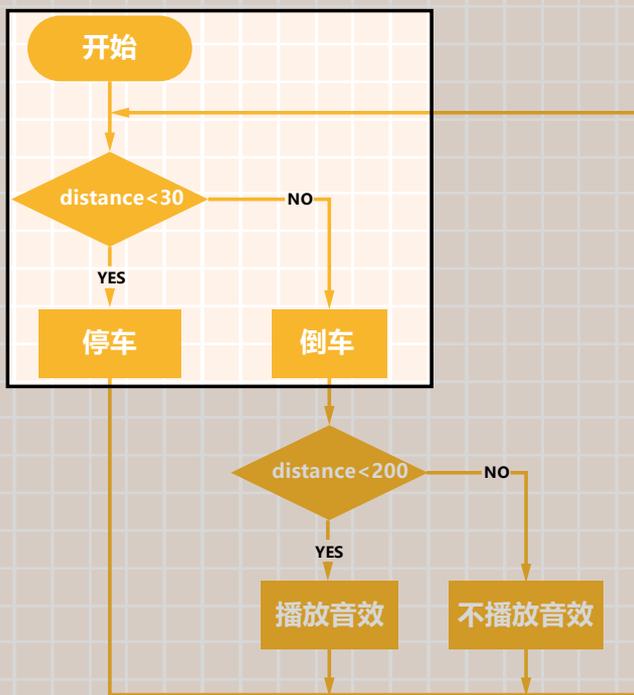
设置 distance 的值为 读取超声波传感器 trig P1 echo P2 单位 cm

当需要调用超声波距离值时,直接使用变量“distance”就可以了。如下图所示。

变量 distance

3、倒车机器人会自动根据超声波测出的距离来判断是否停车,当倒车机器人与障碍的距离比较近时(样例中为:  $distance < 30\text{cm}$ ),倒车机器人自动停车;当倒车机器人与障碍物比较远时,倒车机器人就开始后退;为了与现实生活中的汽车泊车程序相似,我们在倒车过程中加上蜂鸣器,起到倒车警示作用。

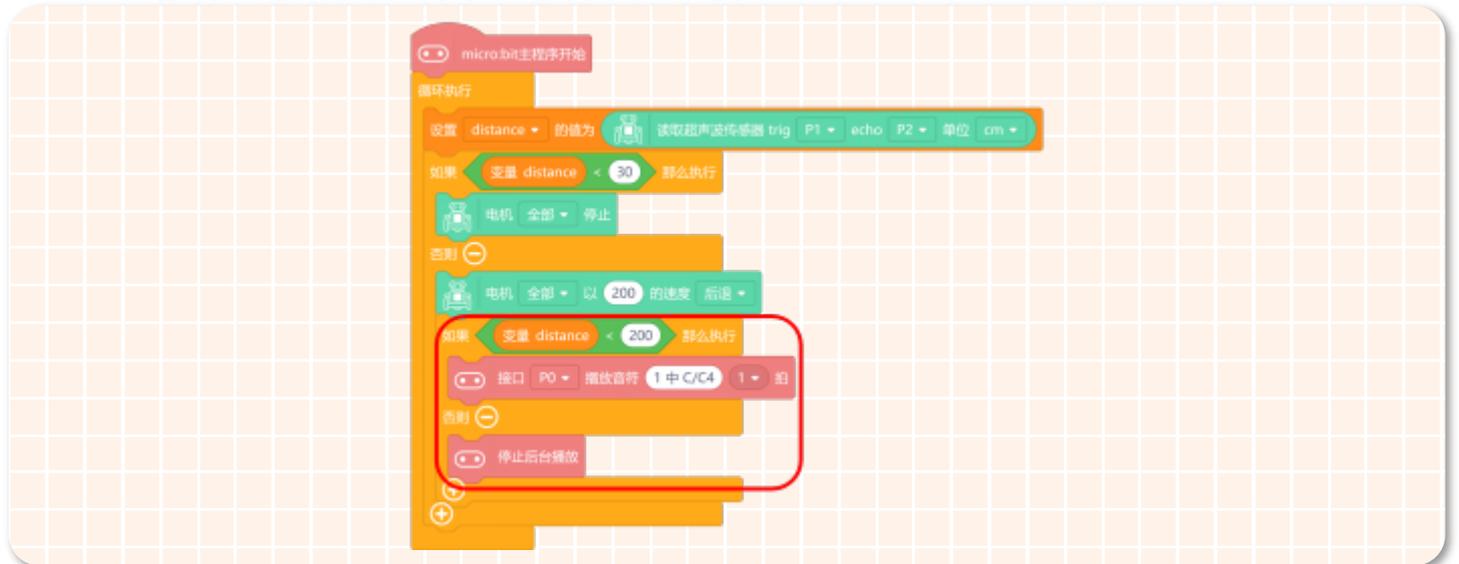
具体要怎样实现停车和倒车的功能呢?为了方便理解,下面我们用流程图来帮助大家梳理一下知识点吧!



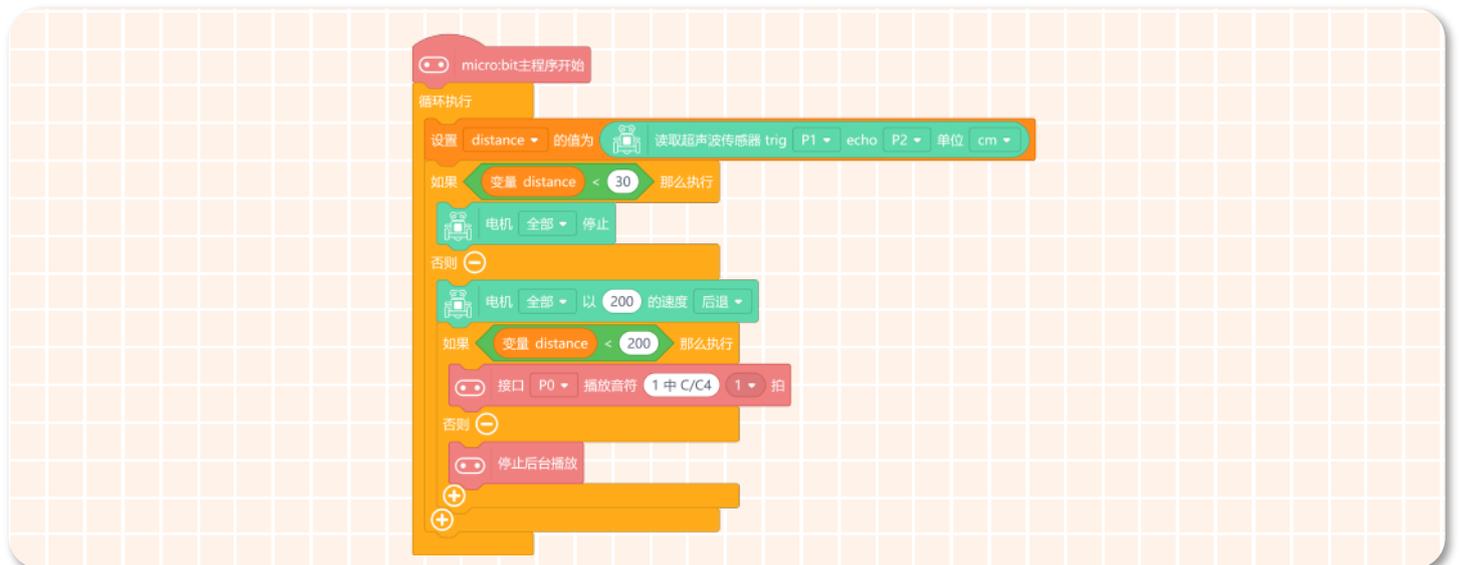
4、根据流程图,我们先来完成停车和倒车的部分。如下图所示。



5、蜂鸣器的警示声在倒车状态下才有效,所以蜂鸣器启动的条件判断是建立在倒车的基础下的,我们应该将“distance<200”这个条件判断模块嵌套到倒车状态下。如下图所示。

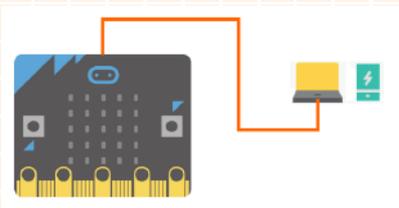


6、完整程序如下图所示。



## STEP 4 下载程序

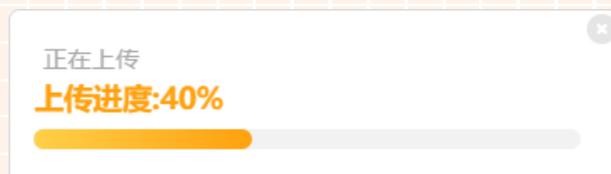
1、连接电脑：下载程序之前,首先需要用 USB 线将 micro:bit 与电脑连接在一起。



2、下载程序：点击菜单栏上“连接设备”，找到并且点击串口号“COM-Microbit”，软件连接到设备。

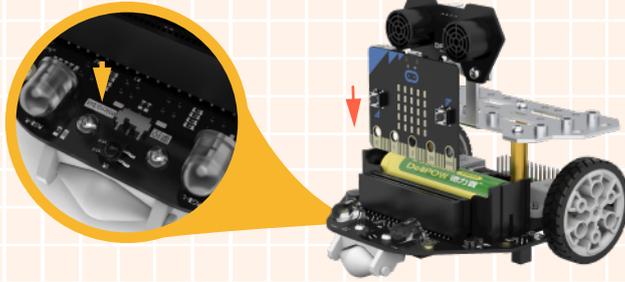


编程完成后,再点击“上传到设备”(如果此时未连接设备,会提示“没有连接到设备”),上传过程中会显示上传进度,上传成功后,提示框自动消失。



**3、安装主控板：**程序下载完成后拔掉 USB 数据线，将带有程序的 micro:bit 主控板安装到麦昆 Plus 上。将 micro:bit 主控板的点阵面朝前，插入插槽，如图所示：

主控板安装好之后，一定要记得将前面的切换开关拨到 micro:bit 侧。



注：国内版的麦昆 Plus 是有主控板的切换开关的，国外版没有。

### STEP 4 效果展示

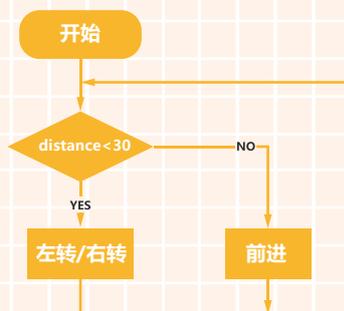
完成上面的步骤后，我们在倒车机器人的尾部放一个障碍物，打开倒车机器人的电源开关，这时候可以看到，当倒车机器人的超声波传感器检测到的距离大于 200 时，执行倒车；检测到的距离大于等于 30 小于等于 200 时，倒车过程中会有警报提示；检测到的距离小于 30 时，停车。



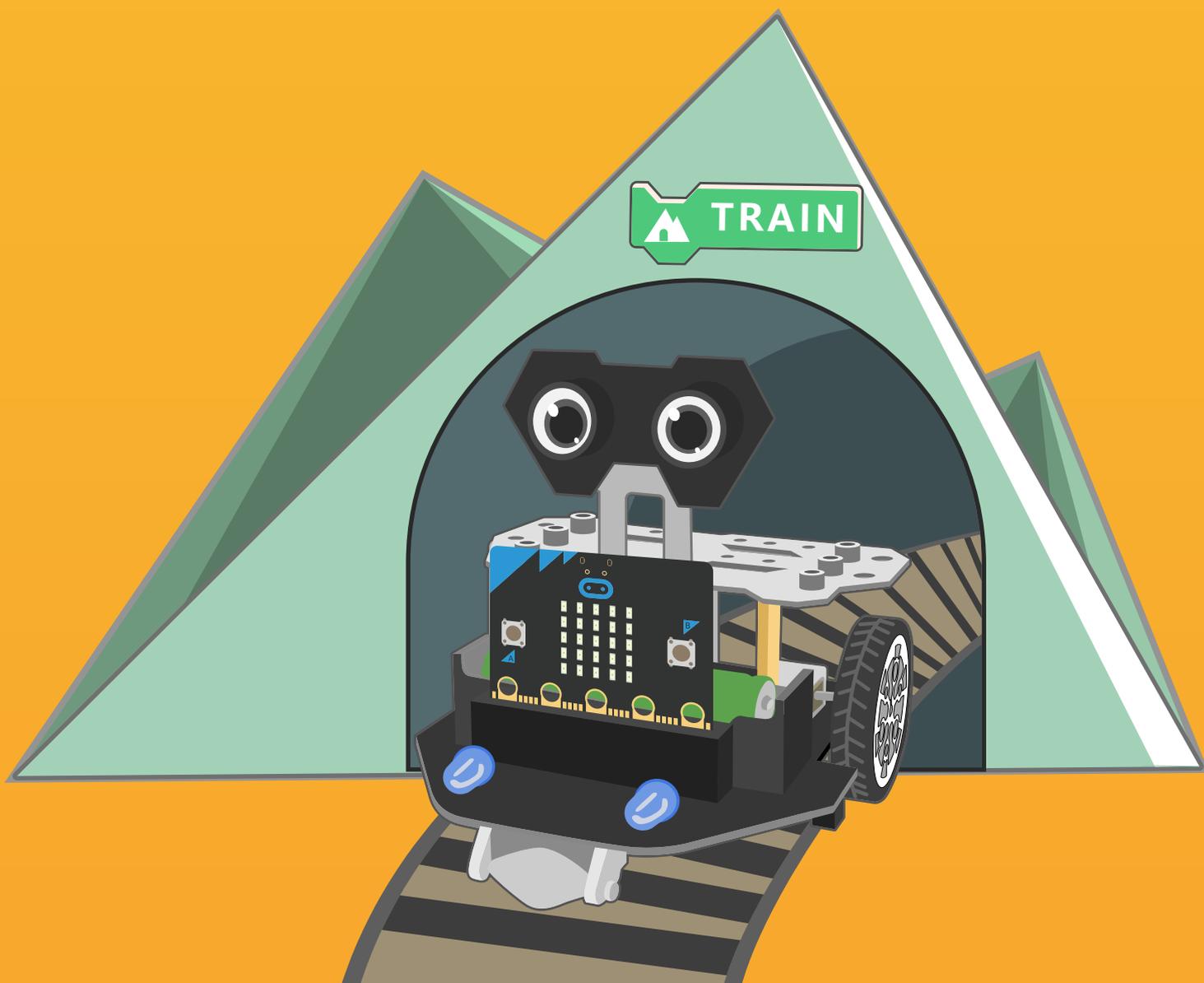
### 探究思考



上面学习了如何制作倒车机器人，倒车机器人是在倒车的过程中检测车位上的障碍物，当离障碍物很近的时候小车停止运动，那现在想让机器人避开障碍物又该怎么实现呢？我们可以根据下面的流程图做一个避障机器人吗？



小提示：超声波的朝向和小车前进的方向一致。



# 第九章

## 循“轨”蹈矩机器人

循“轨”蹈矩机器人喜欢按照地图去寻宝，不管地形多么复杂，它都能够沿着轨迹去探知未知的世界。下面就带着机器人一起去寻宝吧！

## 学习目标



1. 学习巡线传感器的工作原理
2. 理解逻辑“与”

## 电子模块



### 巡线传感器介绍图

巡线传感器



巡线传感器帮助机器人进行白线或者黑线的跟踪,可以检测白背景中的黑线,也可以检测黑背景中的白线。

## 指令学习



### 模块介绍图

巡线传感器模块



读取巡线传感器 L1、L2、L3、R1、R2、R3 的输出值。

比较运算符“等于”模块



比较左右两个值是否相等,相等则为“真”,不等则为“假”

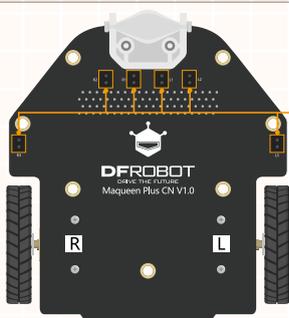
逻辑“与”模块



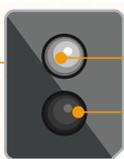
两个条件同时成立(都为真)的条件下“逻辑与”的运算结果为“真”,否则为“假”

巡线传感器是怎样工作的呢?

### 巡线传感器工作原理



麦昆 Plus 的主板上集成了六个巡线传感器的检测头,每个检测头都有两个红外探头。



红外发射器

红外接收器

当巡线传感器检测到地图上的黑线时,输出“1”。  
当巡线传感器没有检测到地图上的黑线时,输出“0”。

L1 没检测黑色  
指示灯灭  
输出 0

R1 检测黑色  
指示灯亮  
输出 1



注意:

1. 巡线传感器检测到的颜色为深色时,则吸收所有的颜色光(包括红外线)而导致红外发射器发出的红外光线不能被反射回来。
2. 这里输出值“0”和“1”不是指高低电平,而是将读取的灰度值进行处理后得到的值。

## 动手实践



### STEP 1 新建项目

打开 Mind+ 软件, 点击“项目”--“新建项目”新建一个项目。



### STEP 2 添加麦昆 Plus

1、选择“扩展”--“主控板”--“micro:bit”主控板, 选择成功如下图:



2、继续点击“扩展板”--“麦昆 Plus”, 选择成功如下图:



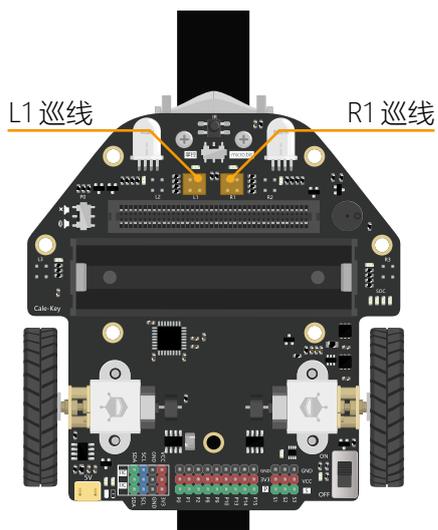
麦昆 Plus 添加成功后, 指令区会显示“



”。

### STEP 3 程序编写

我们的机器人可以沿着地图上的黑色轨迹行走, 如果没有地图可以自己绘制黑色轨迹或者使用黑色胶带代替地图(本项目中应用到 R1、L1 这两个巡线探头)。



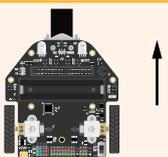
#### 知识要点

必须在黑线内

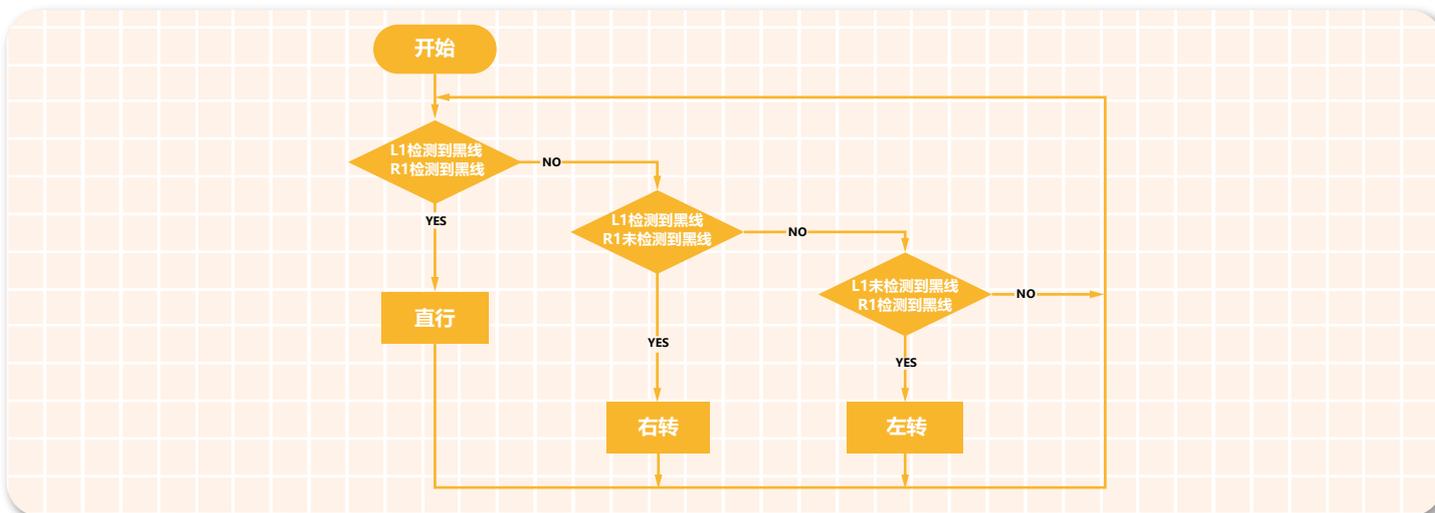


自己绘制黑色轨迹时, L1、R1 这两个传感器需要同时保持在黑色轨迹上, 地图上的黑线宽度就可以保证 L1、R1 传感器同时在黑线上。

1、机器人沿着黑线轨迹行进的过程中会遇到以下情况：

状态示意图	状态描述	巡线传感器状态	输出值	电机运行状态
	L1、R1 两个传感器都在黑线上	左侧传感器检测到黑线 右侧传感器检测到黑线	L1 = 1 R1 = 1	直行
	L1 传感器在黑线上 R1 传感器不在黑线上	左侧传感器检测到黑线 右侧传感器未检测到黑线	L1 = 1 R1 = 0	左转
	L1 传感器不在黑线上 R1 传感器在黑线上	左侧传感器未检测到黑线 右侧传感器检测到黑线	L1 = 0 R1 = 1	右转

2、根据上述机器人可能遇到的情况制作出对应的流程图。



3、在这个项目中,判断条件有两个,R1传感器检测的结果与L1传感器检测的结果,所以需要逻辑“与”模块将这两个判断条件组合起来。

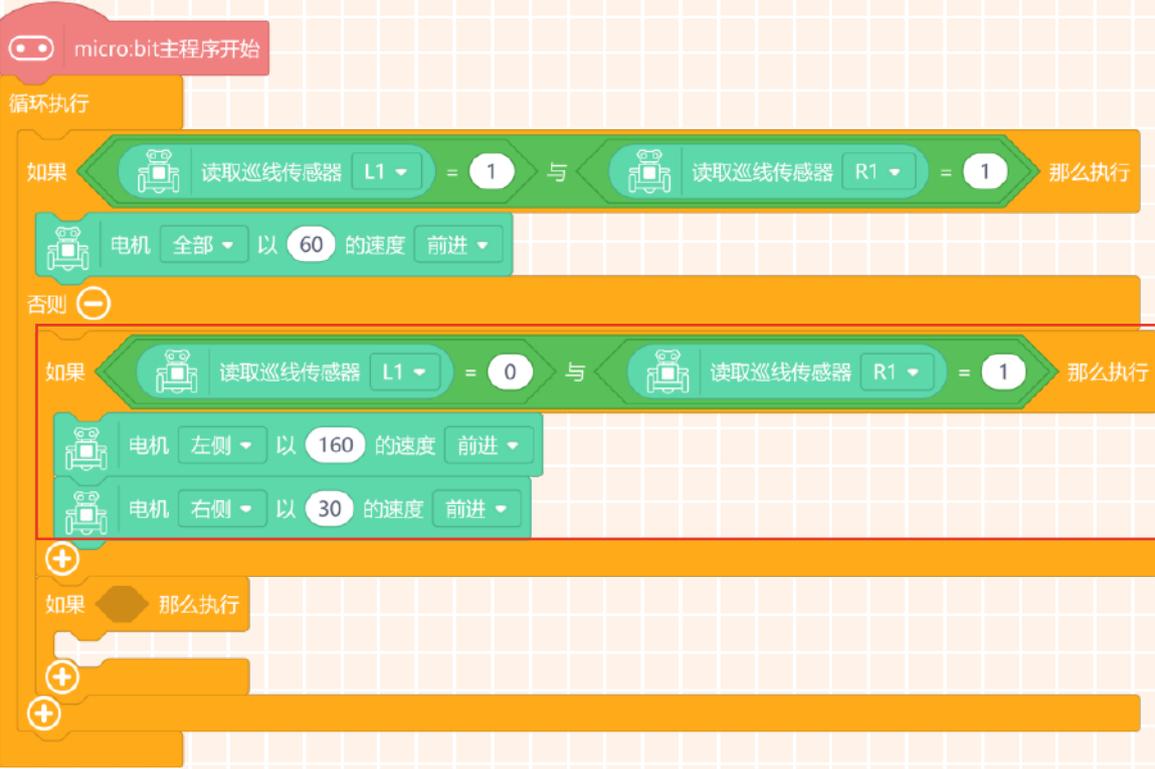
注意：这个程序就表示：L1 巡线传感器检测到黑线,R1 巡线传感器也检测到黑线。

4、根据流程图,先来完成直行的程序,直行的状态是：左侧传感器(L1)检测到黑线,右侧传感器(R1)检测到黑线。

```

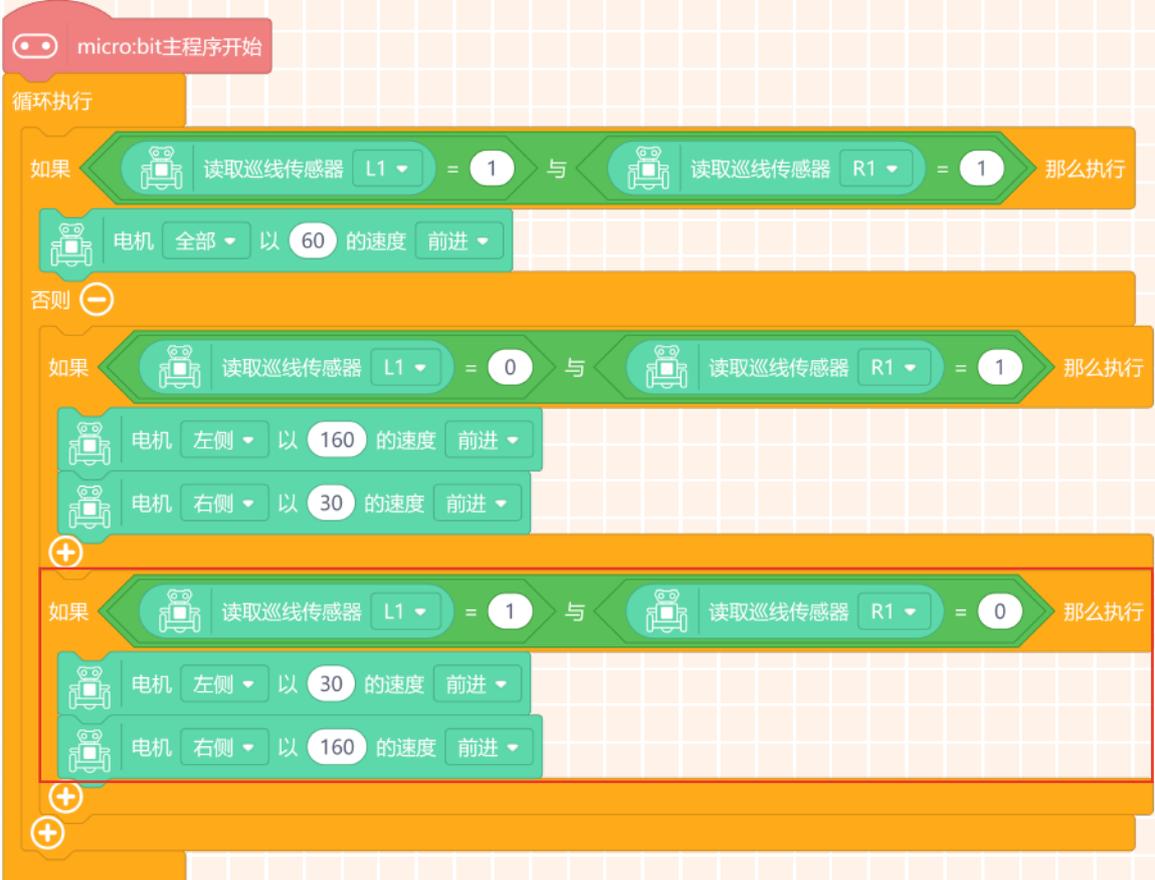
    micro:bit主程序开始
    循环执行
      如果 读取巡线传感器 L1 = 1 与 读取巡线传感器 R1 = 1 那么执行
        电机 全部 以 60 的速度 前进
      否则
  
```

5、然后完成机器人右转的程序,右转的状态是:左侧传感器(L1)没有检测到黑线,右侧传感器(R1)检测到黑线。



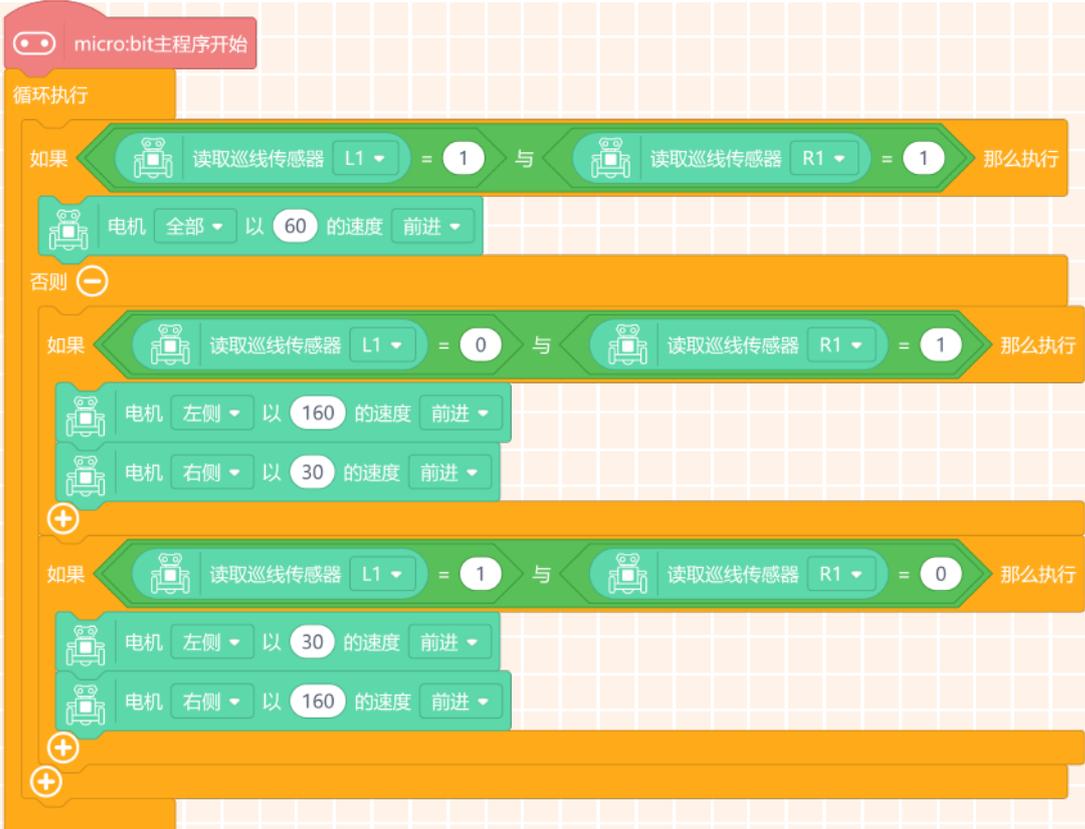
The image shows a Scratch script for a micro:bit robot. It starts with a 'micro:bit 主程序开始' block, followed by a '循环执行' block. Inside the loop, there is an '如果' (if) block with the condition '读取巡线传感器 L1 = 1 与 读取巡线传感器 R1 = 1'. The '那么执行' (then) block contains '电机 全部 以 60 的速度 前进'. The '否则' (else) block contains another '如果' (if) block with the condition '读取巡线传感器 L1 = 0 与 读取巡线传感器 R1 = 1'. This inner 'if' block's '那么执行' block contains two '电机' (motor) blocks: '电机 左侧 以 160 的速度 前进' and '电机 右侧 以 30 的速度 前进'. This entire right-turn logic is highlighted with a red border. Below the 'else' block, there are several empty '那么执行' and '否则' blocks, indicating the script continues.

6、最后完成机器人左转的程序,左转的状态是:左侧传感器(L1)检测到黑线,右侧传感器(R1)没有检测到黑线。



The image shows a Scratch script for a micro:bit robot, similar to the one above. It starts with a 'micro:bit 主程序开始' block, followed by a '循环执行' block. Inside the loop, there is an '如果' (if) block with the condition '读取巡线传感器 L1 = 1 与 读取巡线传感器 R1 = 1'. The '那么执行' (then) block contains '电机 全部 以 60 的速度 前进'. The '否则' (else) block contains another '如果' (if) block with the condition '读取巡线传感器 L1 = 0 与 读取巡线传感器 R1 = 1'. This inner 'if' block's '那么执行' block contains two '电机' (motor) blocks: '电机 左侧 以 160 的速度 前进' and '电机 右侧 以 30 的速度 前进'. Below this, there is a third '如果' (if) block with the condition '读取巡线传感器 L1 = 1 与 读取巡线传感器 R1 = 0'. This block's '那么执行' block contains two '电机' (motor) blocks: '电机 左侧 以 30 的速度 前进' and '电机 右侧 以 160 的速度 前进'. This entire left-turn logic is highlighted with a red border. Below the third 'if' block, there are several empty '那么执行' and '否则' blocks, indicating the script continues.

7、完整程序如下图所示。



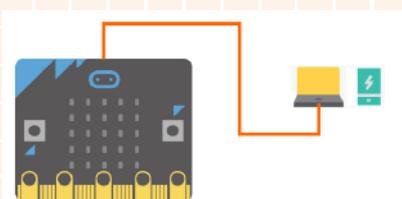
The image shows a Scratch script for a micro:bit robot. It starts with a 'micro:bit 主程序开始' block. A '循环执行' (Loop) block contains three conditional blocks. The first '如果' (If) block checks if both L1 and R1 line sensors are 1; if true, it sets all motors to move forward at speed 60. The second '否则' (Else) block contains two '如果' blocks: the first checks if L1 is 0 and R1 is 1, setting the left motor to speed 160 and the right to 30; the second checks if L1 is 1 and R1 is 0, setting the left motor to speed 30 and the right to 160. There are also two empty '如果' blocks at the bottom of the loop.

注意：如果循轨蹈矩机器人在巡线的过程中，出现转弯太小或转弯太大的情况，可以通过调节电机的速度来改善。

8、保存，在命名保存区将项目名称改为“循轨蹈矩机器人”。

### STEP 4 下载程序

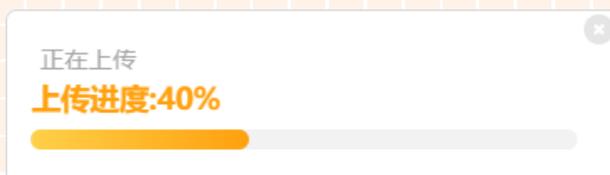
1、连接电脑：下载程序之前，首先需要用 USB 线将 micro:bit 与电脑连接在一起。



2、下载程序：点击菜单栏上“连接设备”，找到并且点击串口号“COM40-Microbit”，软件连接到设备。

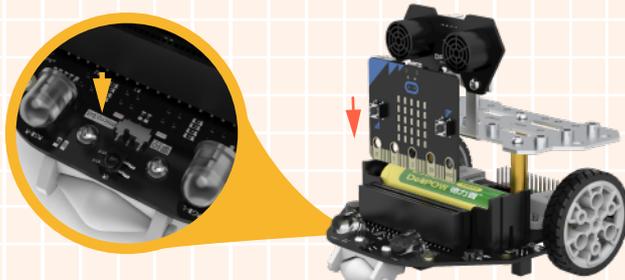


编程完成后，再点击“上传到设备”（如果此时未连接设备，会提示“没有连接到设备”），上传过程中会显示上传进度，上传成功后，提示框自动消失。



**3、安装主控板：**程序下载完成后拔掉 USB 数据线，将带有程序的 micro:bit 主控板安装到麦昆 Plus 上。将 micro:bit 主控板的点阵面朝前，插入插槽，如图所示：

主控板安装好之后，一定要记得将前面的切换开关拨到 micro:bit 侧。



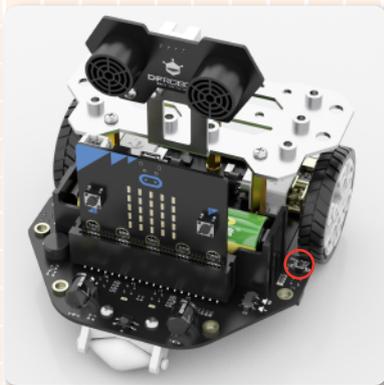
注：国内版的麦昆 Plus 是有主控板的切换开关的，国外版没有。

## 扩展



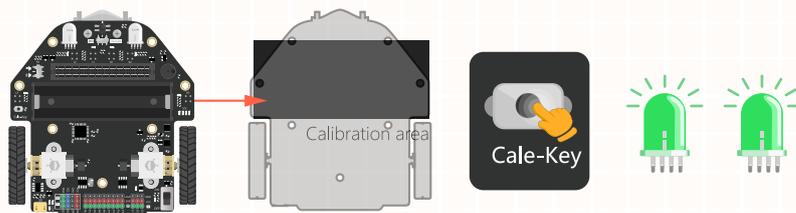
麦昆 Plus 的巡线传感器在出厂的时候就已经对它进行过校准了，都可以正常使用。如果我们手中拿到的产品不能准确的识别黑色线条，或者受光线影响，识别率差，那么可以尝试做一下巡线传感器的校准。

下图红圈内的按钮即为校准按键



将小车放到地图上的黑色校准区域内，确保所有的巡线传感器都在黑色校准区域内，按下校准按键 1 秒钟，车前的两个 RGB 灯闪烁绿色，表示校准完成，松开按键就可以了。

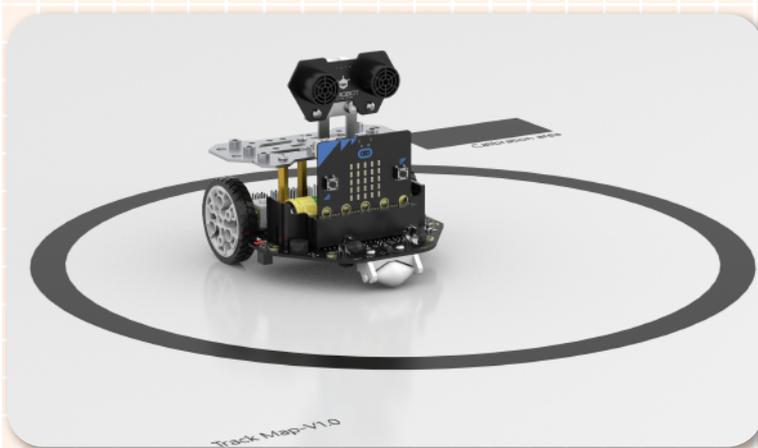
注意：打印机打印的黑色是无法正确识别为黑色，黑色胶带、印刷地图可以正常使用。



## 探究思考



在生活中，扫地机器人逐渐成为我们的居家必备品，他可以自主充电，按时对地面进行清洁，但是为了防止它在扫地的过程中跌下台阶，所以它需要具备防止跌落的功能。防跌落功能的实现是需要依靠扫地机器人底部的多个探测器，麦昆 Plus 具有 6 个巡线传感器探头，刚好满足这个条件。下面就来制作一款扫地机器人吧！用地图中的黑线来模拟台阶的边缘，让扫地机器人只能在黑线内行驶哦！





# 第十章

## 道路选择

当小车行至十字路口时，它应该作何选择呢？每条路都不相同，如果小车需要将每条路都走一次，并且只走一次又该怎样实现呢？下面就一起来做一个可以进行道路选择的小车吧！

## 学习目标



学习使用多个巡线探头配合

## 动手实践



### STEP 1 新建项目

打开 Mind+ 软件, 点击“项目”--“新建项目”新建一个项目。



### STEP 2 添加麦昆 Plus

1、选择“扩展”--“主控板”--“micro:bit”主控板, 选择成功如下图:



2、继续点击“扩展板”--“麦昆 Plus”, 选择成功如下图:



麦昆 Plus 添加成功后, 指令区会显示“

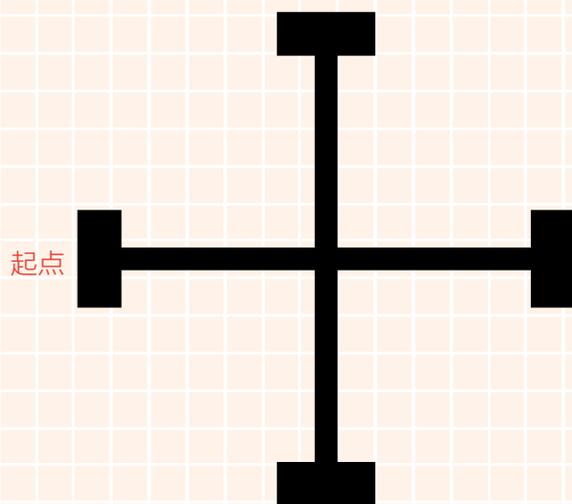


扩展板

”。

### STEP 3 程序编写

1、在这个项目中, 我们需要使用到地图 2。让小车从起点出发, 走遍地图上所有的黑线轨迹, 应该怎样做呢? 只要小车在巡线的过程中, 到达十字路口处转弯, 路口尽头处调头。一直重复这个步骤就可以了。



2、首先完成巡线部分的程序。

The code starts with a 'micro:bit 主程序开始' block. It enters a '循环执行' (Loop) block. The first '如果' (If) block checks '读取巡线传感器 L1 = 1' and '读取巡线传感器 R1 = 1'. If true, it sets '电机 全部' to '以 40 的速度 前进'. The '否则' (Else) block contains two '如果' blocks. The first checks 'L1 = 1' and 'R1 = 0', setting '电机 左侧' to '以 0 的速度 前进' and '电机 右侧' to '以 40 的速度 前进'. The second checks 'L1 = 0' and 'R1 = 1', setting '电机 左侧' to '以 40 的速度 前进' and '电机 右侧' to '以 0 的速度 前进'.

3、当 L1、R1 两个传感器都没有检测到黑线,说明小车驶出黑色轨迹,让小车原地转圈直到再次找到黑色轨迹为止。

This code is identical to the previous one but includes a fourth '如果' block in the '否则' section. It checks 'L1 = 0' and 'R1 = 0'. If true, it sets '电机 左侧' to '以 40 的速度 前进' and '电机 右侧' to '以 40 的速度 后退', causing the robot to spin in place.

4、当 L2、R2 两个传感器都检测到黑线时,说明小车到达了十字路口或路口的尽头,小车需要原地转圈再次去找黑色轨迹。

micro:bit主程序开始

循环执行

如果 读取巡线传感器 L1 = 1 与 读取巡线传感器 R1 = 1 那么执行

电机 全部 以 40 的速度 前进

否则

如果 读取巡线传感器 L1 = 1 与 读取巡线传感器 R1 = 0 那么执行

电机 左侧 以 0 的速度 前进

电机 右侧 以 40 的速度 前进

如果 读取巡线传感器 L1 = 0 与 读取巡线传感器 R1 = 1 那么执行

电机 左侧 以 40 的速度 前进

电机 右侧 以 0 的速度 前进

如果 读取巡线传感器 L1 = 0 与 读取巡线传感器 R1 = 0 那么执行

电机 左侧 以 40 的速度 前进

电机 右侧 以 40 的速度 后退

+

如果 读取巡线传感器 L2 = 1 与 读取巡线传感器 R2 = 1 那么执行

电机 左侧 以 40 的速度 前进

电机 右侧 以 40 的速度 后退

5、完整程序如下图所示：

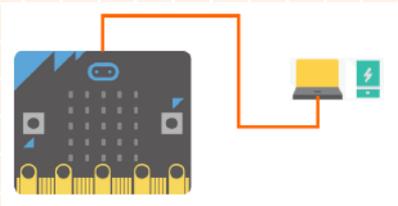
The image shows a Scratch script for a micro:bit robot. It starts with a 'micro:bit 主程序开始' block. A '循环执行' (Loop) block contains four conditional blocks based on sensor readings from L1, R1, L2, and R2. Each conditional block has specific motor actions: '全部' (all) at speed 40 for forward, '左侧' (left) at speed 0 for forward, '右侧' (right) at speed 40 for forward, '左侧' (left) at speed 40 for forward, and '右侧' (right) at speed 40 for backward.

```
micro:bit 主程序开始
循环执行
  如果 读取巡线传感器 L1 = 1 与 读取巡线传感器 R1 = 1 那么执行
    电机 全部 以 40 的速度 前进
  否则
    如果 读取巡线传感器 L1 = 1 与 读取巡线传感器 R1 = 0 那么执行
      电机 左侧 以 0 的速度 前进
      电机 右侧 以 40 的速度 前进
    如果 读取巡线传感器 L1 = 0 与 读取巡线传感器 R1 = 1 那么执行
      电机 左侧 以 40 的速度 前进
      电机 右侧 以 0 的速度 前进
    如果 读取巡线传感器 L1 = 0 与 读取巡线传感器 R1 = 0 那么执行
      电机 左侧 以 40 的速度 前进
      电机 右侧 以 40 的速度 后退
    如果 读取巡线传感器 L2 = 1 与 读取巡线传感器 R2 = 1 那么执行
      电机 左侧 以 40 的速度 前进
      电机 右侧 以 40 的速度 后退
```

注意：小车出现转弯太小或转弯太大的情况，可以自己对电机的速度进行调节。

## STEP 4 下载程序

1、连接电脑：下载程序之前，首先需要用 USB 线将 micro:bit 与电脑连接在一起。



2、下载程序：点击菜单栏上“连接设备”，找到并且点击串口号“COM40-Microbit”，软件连接到设备。

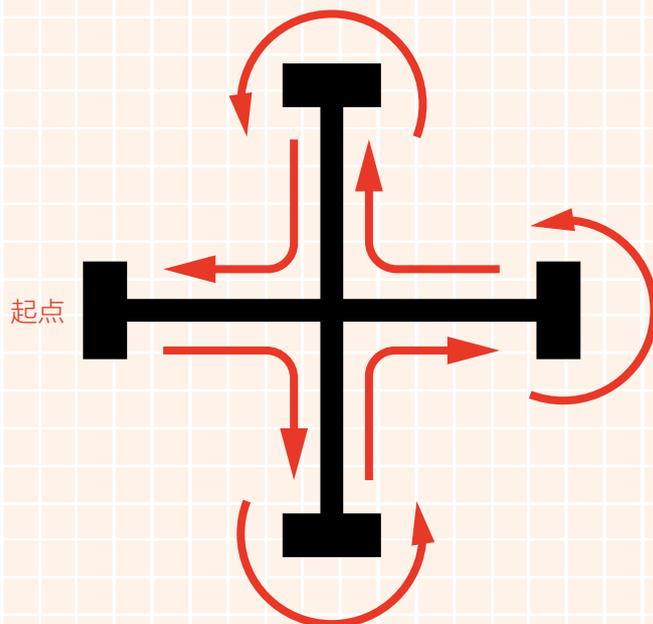


编程完成后,再点击“上传到设备”(如果此时未连接设备,会提示“没有连接到设备”),上传过程中会显示上传进度,上传成功后,提示框自动消失。

正在上传  
上传进度:40%

## STEP 5 效果展示

完成上面的步骤后,将小车放到准备好的地图 2 上,打开小车的电源开关。具体的执行效果如下图所示:

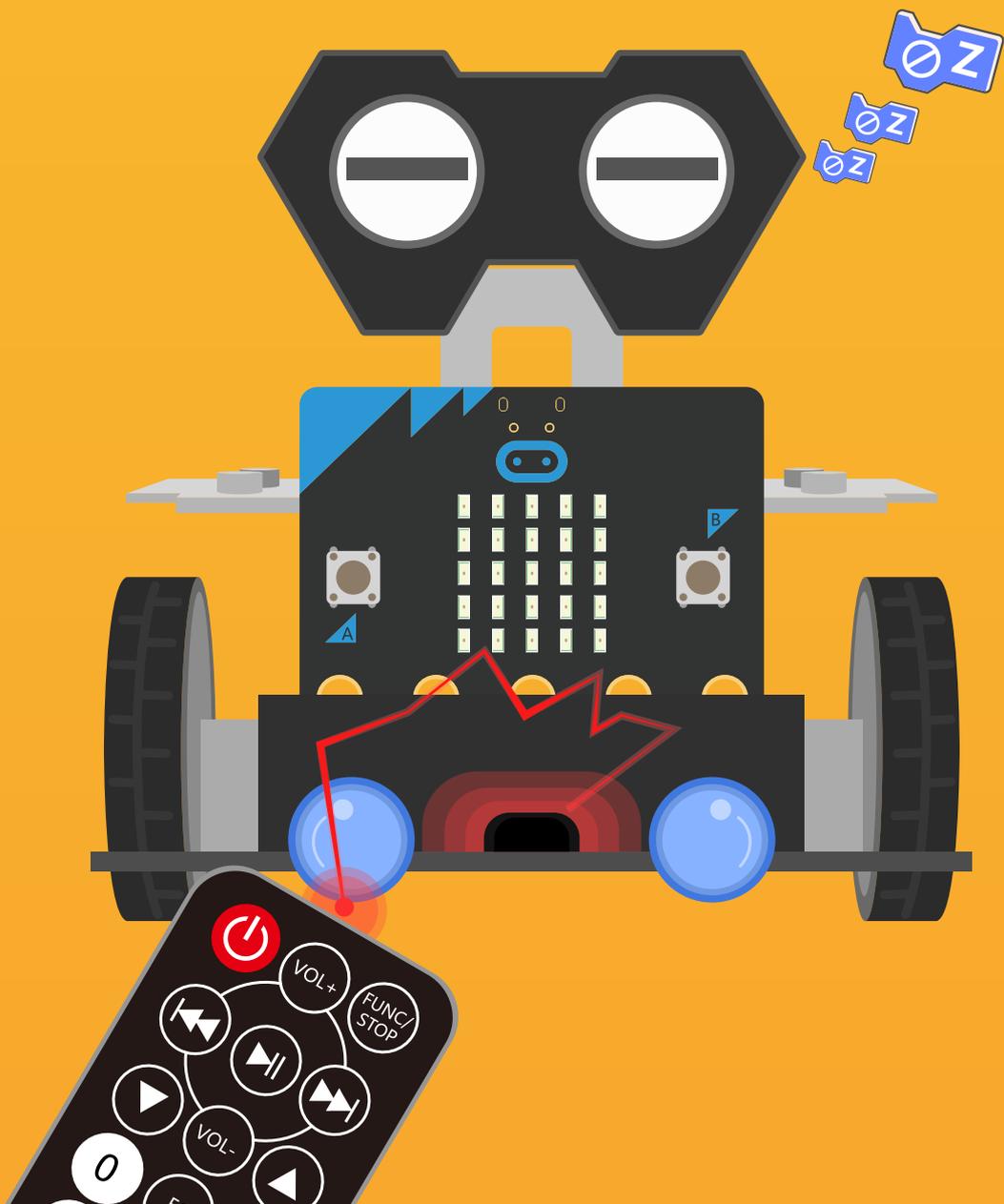


## 探究思考



这个项目中,小车行驶到十字路口的时候,自己优先选择的是右转,那我们是否可以在右转的过程中让右侧 RGB 灯闪烁一下呢?在路口尽头的时候让左右两侧的 RGB 灯同时闪烁一次再去完成调头操作。

小提示:要完成的功能很简单,只需要在上面的程序中加上设置 RGB 灯模块就可以了。



# 第十一章

## 红外遥控机器人

大家都知道，遥控器在生活中被广泛的应用。因为它可以使我们不接触设备，从而达到控制设备。这对我们的生活和工作带来很大的便捷，而最常使用的是红外遥控器。接下来就用红外遥控器结合麦昆 Plus 身上的红外接收器制作一个“红外遥控机器人”。

## 学习目标



1. 学习如何使用函数模块
2. 学习如何使用红外遥控器控制电机

## 电子模块



### 红外接收介绍图

红外接收器



将红外遥控器发来的红外光信号转换成电信号。

## 指令学习



### 模块介绍图

红外信息接收指令



接收和读取红外信号的值。

定义子函数



当一个连续的动作,需要在程序中多次出现时,为了使得程序更加清楚,需要定义子函数。

调用子函数



在程序中,需要使用到子函数的功能时,将调用模块拖入到程序中。

## 动手实践



这个项目中,要用到红外遥控器来控制机器人,所以需要了解红外遥控器上的按键所对应的码值,每个按键都对应了一个转换后的十进制数。

### 红外遥控对应码值



按键	码值
红色按钮	255
VOL+	127
FUNC/STOP	191
左箭头	223
暂停	95
右箭头	159
下箭头	239
VOL-	111

按键	码值
上箭头	175
0	207
EQ	79
ST/REST	143
1	247
2	119
3	183
4	215

按键	码值
5	87
6	151
7	231
8	103
9	167

## STEP 1 新建项目

打开 Mind+ 软件,点击“项目”--“新建项目”新建一个项目。



## STEP 2 添加麦昆 Plus

1、选择“扩展”--“主控板”--“micro:bit”主控板,选择成功如下图:



2、继续点击“扩展板”--“麦昆 Plus”,选择成功如下图:



麦昆 Plus 添加成功后,指令区会显示“ 扩展板”。

## Step 3 Programming

1、首先找到我们要使用的当“红外接收到数据”指令,并将当“红外接收到数据”拖到编辑区。如下图所示。



将红外信息接收指令拖到编辑区后,请在变量指令区会新建一个名叫“message”的变量,是用来存放红外遥控器的码值。如下图所示。



2. 创建一个名为“前进”的功能函数。如下图所示。



①在指令区用鼠标点击“函数”；

②在“函数”指令区点击“自定义模块”；

③在“添加一个自定义模块”中,将函数名“积木名称”改成“前进”；

④上面步骤完成之后,点击“完成”就可以了。

### 知识要点

#### 什么是函数？

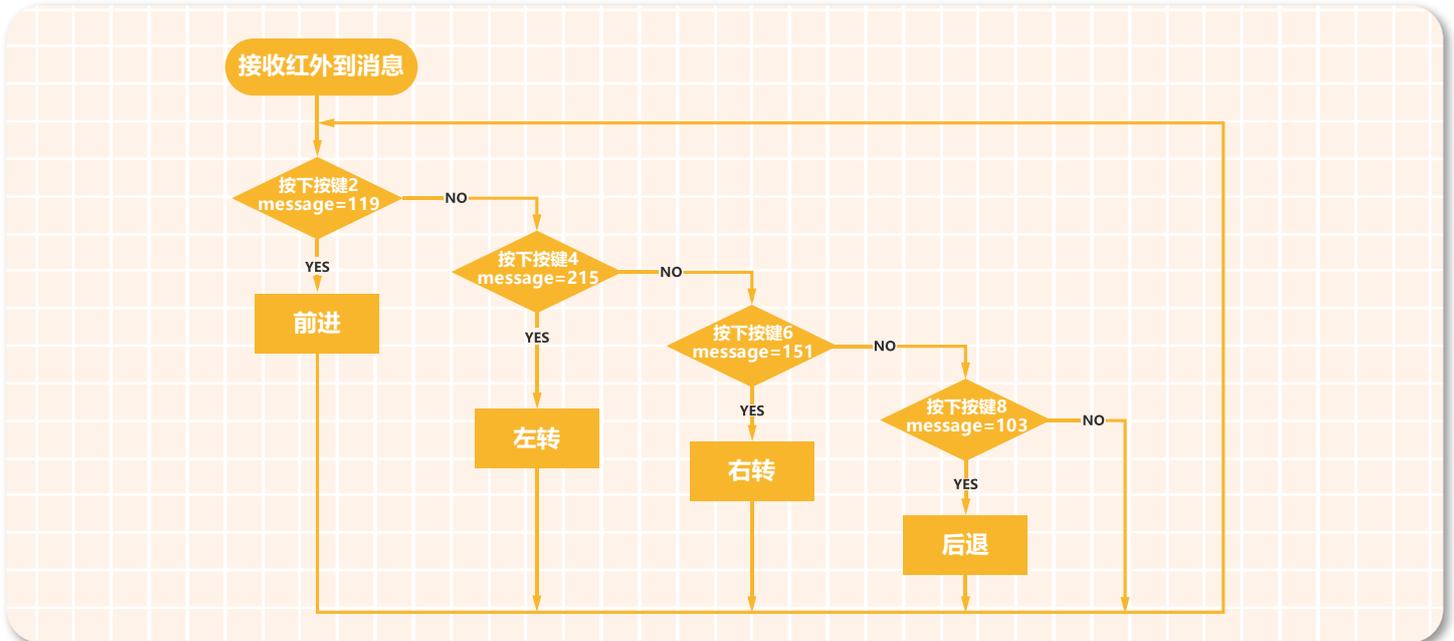
函数是指一段可以直接被另一段程序引用的程序,也可以叫做子程序或者方法。

通常我们将一段较大的程序分成多个程序块,每一个程序块都用来实现一个特定的功能。将经常使用的功能封装起来,在需要使用时可以由其他主函数或子函数直接调用,并且同一个函数可以被调用多次。使用函数不仅可以减少重复编写程序段的工作量,还能提高程序的利用率。

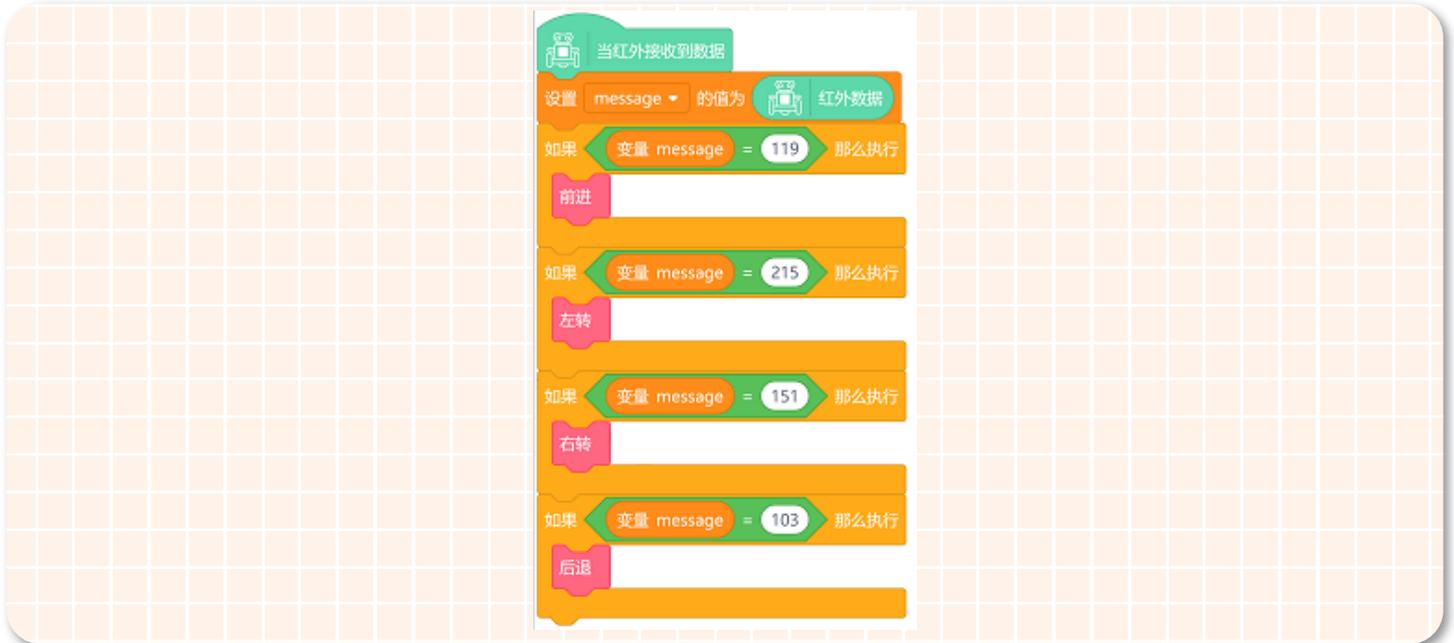
3. 再用同样的方法创建“后退”、“左转”、“右转”这几个功能函数。创建好自定义的功能函数后,对应的功能函数模块会自动出现在编辑区,在函数指令区也可以找到对应的调用子函数模块。所有功能函数创建成功后如下图所示：



4、我们使用红外遥控器上的按键 2、4、6、8 来控制小车完成指定的功能呢？下面就用一个流程图来分析一下吧！



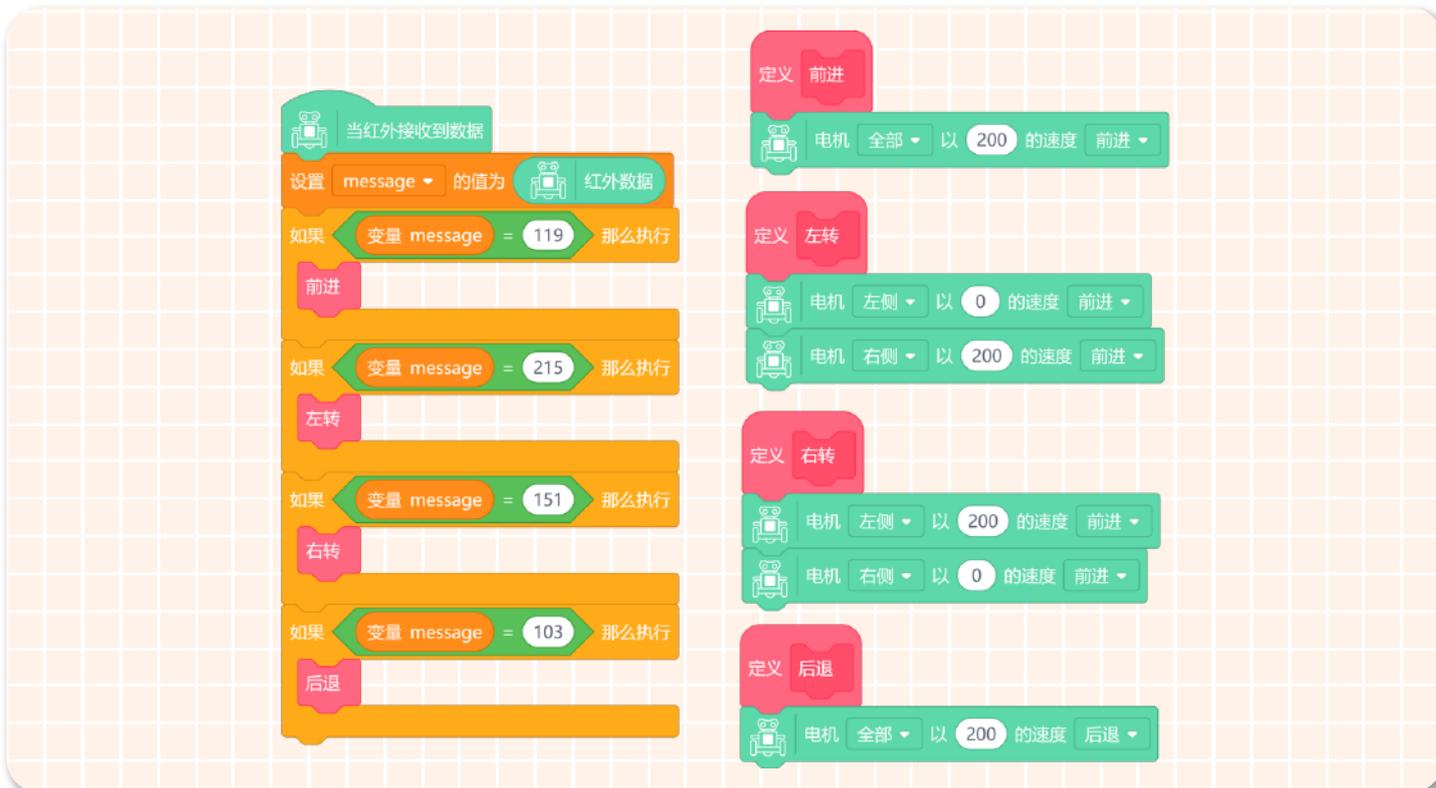
5、从流程图中可以看出，按下按键对应的的码值，然后判断该码值是否符合条件。当红外遥控器的码值“message=119”时，调用前进函数；红外遥控器的码值“message=215”时，调用左转函数；红外遥控器的码值“message=151”时，调用右转弯函数；红外遥控器的码值“message=103”时，调用后退函数。具体程序如下图所示：



6、上面的程序只是一个框架，是程序的一部分，具体的功能需要在功能函数中来实现。下面我们以前进功能函数为例，我们应该怎样实现按下按键 2，小车就前进呢？我们只需要在前进功能函数中加上一个电机驱动模块就可以了。如下图所示。

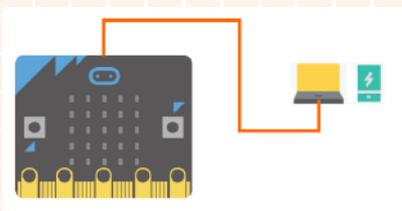


其实只要我们逻辑清晰,程序的实现是很简单的。接下来“后退”、“左转”、“右转”功能函数的实现,就大家自己完成吧!完整程序如下图所示。



7、保存项目,在项目命名区将命名改为“红外遥控机器人”。

1、连接电脑: 下载程序之前,首先需要用 USB 线将 micro:bit 与电脑连接在一起。



2、下载程序: 点击菜单栏上“连接设备”,找到并且点击串口号“COM-Microbit”,软件连接到设备。

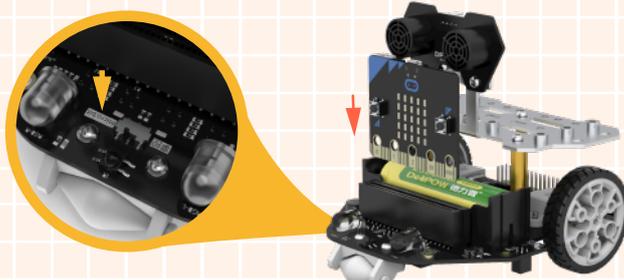


编程完成后,再点击“上传到设备”(如果此时未连接设备,会提示“没有连接到设备”),上传过程中会显示上传进度,上传成功后,提示框自动消失。



3、安装主控板: 程序下载完成后拔掉 USB 数据线,将带有程序的 micro:bit 主控板安装到麦昆 Plus 上。将 micro:bit 主控板的点阵面朝前,插入插槽,如图所示:

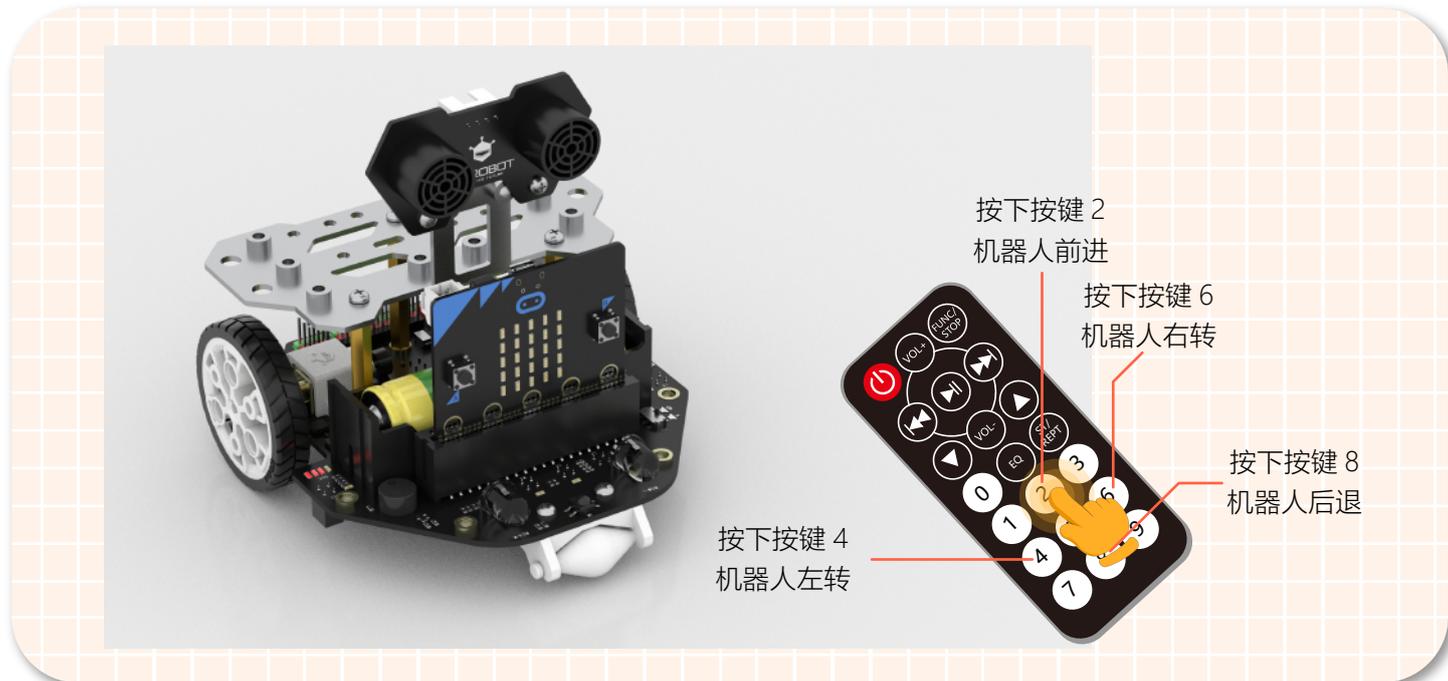
主控板安装好之后,一定要记得将前面的切换开关拨到 micro:bit 侧。



注: 国内版的麦昆 Plus 是有主控板的切换开关的,国外版没有。

## STEP 4 效果展示

完成上面的步骤之后,打开机器人的电源开关,用红外遥控器来遥控我们的机器人吧!



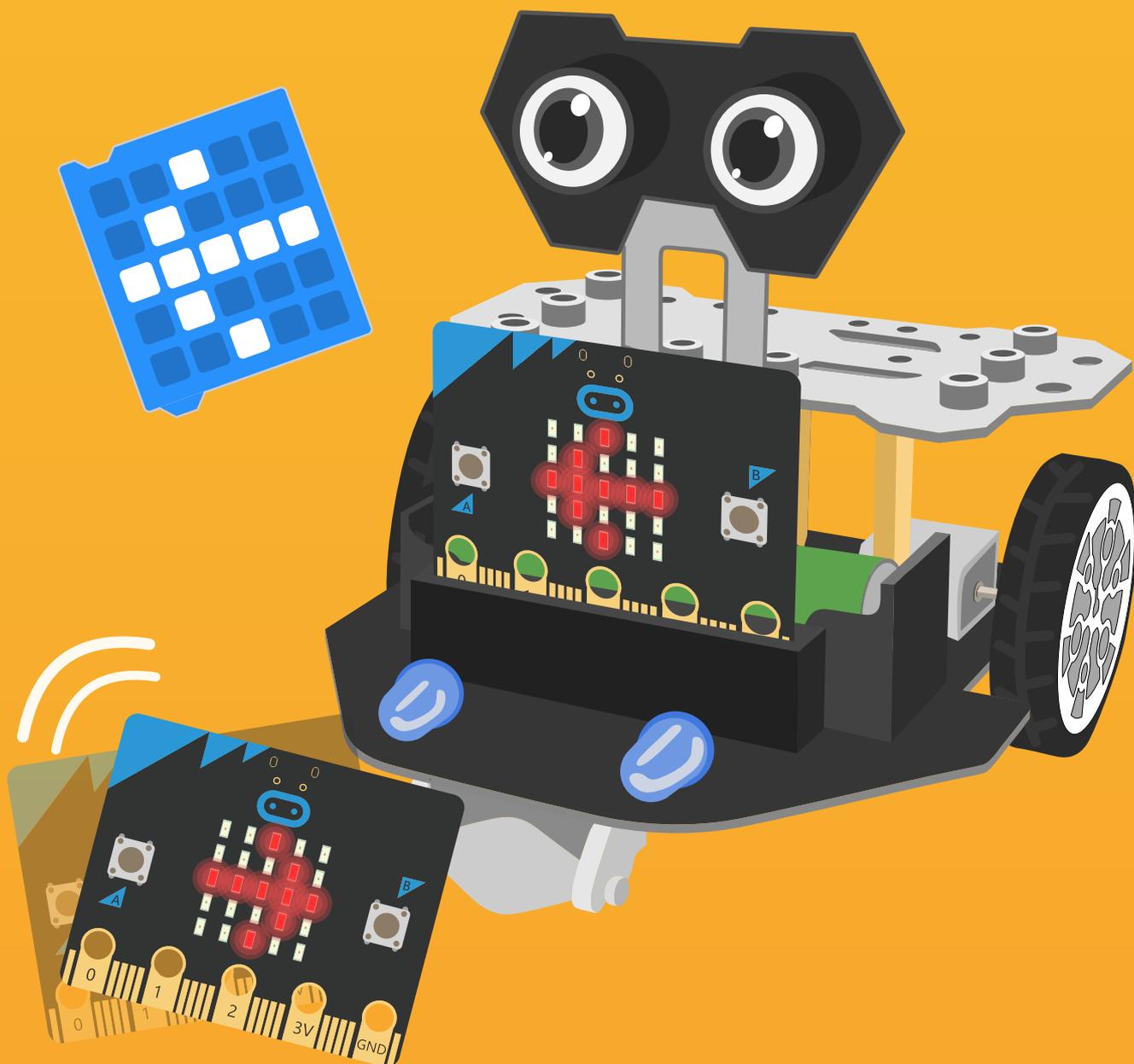
## 探究思考



上面程序已经可以实现用红外遥控器控制机器人了,可是在控制的过程中发现,一旦机器人运行起来它就会一直运行,只有关闭它的电源开关机器人才会停下来。这样对我们来说是非常不方便的,下面就留给大家一个任务吧!

**任务:** 通过红外遥控器上的红色按键控制机器人的电机停止运行。

**小提示:** 只需要在上面的程序中再加上一个停止的功能函数就可以了哦!



# 第十二章

## 体感机器人

你有玩过体感赛车游戏吗？通过不停的变化遥控器的方向，从而控制屏幕中的赛车奔跑和转弯，非常的刺激！而我们的车也可以实现体感控制，因为 micro:bit 上集成了加速度传感器，下面我们就利用 micro:bit 上的加速度传感器和无线通信来制作一辆“无线遥控机器人”吧！

## 学习目标



1. 学习无线通信的基本方法
2. 学习加速度传感器的使用

## 电子模块



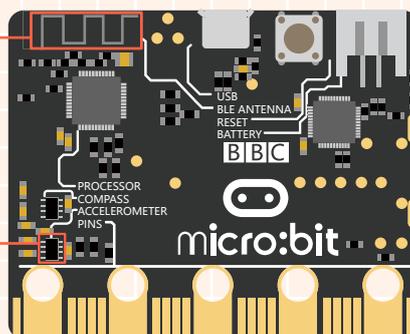
### 蓝牙与加速度传感器介绍

#### 蓝牙

板载蓝牙功能可以让两块及以上的 micro:bit 之间进行无线通信, 允许 micro:bit 发送和接收蓝牙信号。

#### 加速度传感器

micro:bit 自带一个三轴加速度传感器, 可以检测 x,y,z 三个方向的重力加速度



## 指令学习



### 模块介绍图

#### 设置无线频道模块



用于设置无线通讯频道, 让 micro:bit 与小车在同一个频道组里面, 可以进行收发数据。

#### 无线发送模块



通过无线通信向小车发送数字信息。

#### 无线接收数据模块



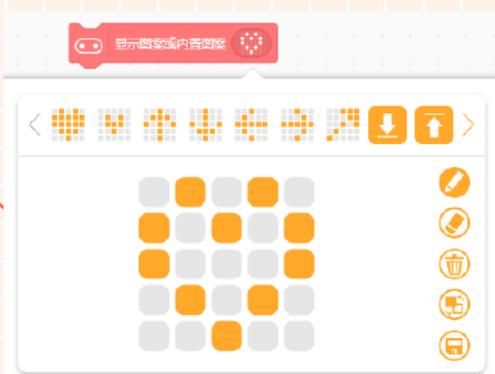
小车通过无线通信接收 micro:bit 发来的数据。

#### 姿态模块



判断 micro:bit 的姿态。共有 11 种姿态。

#### LED 显示模块



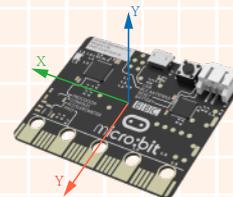
可以自主设置 micro:bit 点阵屏上的显示样式。

## 什么是加速度传感器?

加速度传感器可以感受物体的速度变化情况。micro:bit 上既有加速度传感器,能够判断 micro:bit 的运动方向、角度、手势场景等。

### 加速度传感器介绍

micro:bit 自带一个三轴加速度传感器,可以检测 x,y,z 三个方向的重力加速度,每个轴测量的值是正数或者负数,当读数为 0 时,表示加速度传感器沿着该特定轴“水平”位置。而 micro:bit 不同的姿态就是根据计算加速度传感器 x,y,z 这三个轴的矢量和来确定。矢量表示带有方向的量,矢量和指方向与大小的和。在高中和大学的数学课中会讲到。



## 动手实践



### 体感小车——控制端:

在开始编写程序之前,我们先来分析一下加速度传感器在项目中有什么作用?

体感机器人是通过改变 micro:bit 的姿态来控制小车前进、后退、左转、右转。当 micro:bit 向前倾斜时,点阵屏上出现向前的箭头,可控制小车前进;当 micro:bit 向后倾斜时,点阵屏上出现向后的箭头,可控制小车后退;当 micro:bit 向左倾斜时,点阵屏上出现向左的箭头,可控制小车左转;当 micro:bit 向右倾斜时,让点阵屏上出现向右的箭头,可控制小车右转。

**注意:** 根据上面的分析,在这个项目中需要两个 micro:bit 哦! 一个是控制端,通过无线通信发送数据;一个是接收端,通过无线通信接收数据。

### STEP 1 新建项目

打开 Mind+ 软件,点击“项目”--“新建项目”新建一个项目。



### STEP 2 添加麦昆 Plus

1、选择“扩展”--“主控板”--“micro:bit”主控板,选择成功如下图:



2、继续点击“扩展板”--“麦昆 Plus”,选择成功如下图:



麦昆 Plus 添加成功后,指令区会显示“



”。

### STEP 3 程序编写

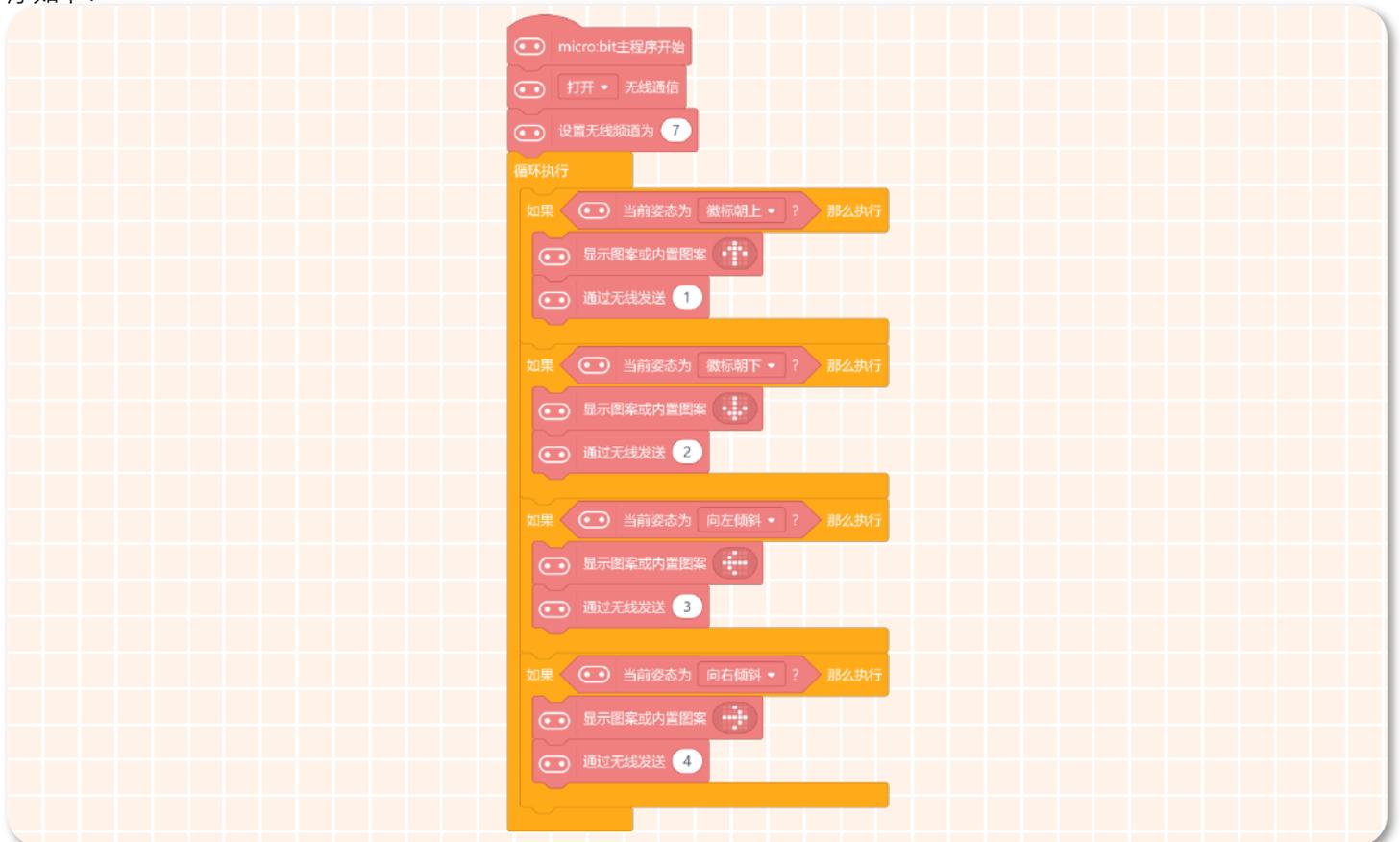
1. 打开无线通信功能, 在主程序中设置无线通信组为 7。如下图所示。



2. 当 micro:bit 向前倾斜时, 让 micro:bit 的点阵屏上显示向前的箭头, 并发送无线数据数字 1 给小车。程序如下:



3. micro:bit 向下、向左、向右转动的程序实现是一样的, 只需要将无线发送数字改成不同的就可以了。控制端的完整程序如下:



4. 保存项目,在命名保存区将项目名称改为“体感小车——控制端”,将程序下载到控制端的设备。

## 体感小车——接收端:

### STEP 1 新建项目

打开 Mind+ 软件,点击“项目”--“新建项目”新建一个项目。



### STEP 2 添加麦昆 Plus

1. 选择“扩展”--“主控板”--“micro:bit”主控板,选择成功如下图:



2. 继续点击“扩展板”--“麦昆 Plus”,选择成功如下图:



麦昆 Plus 添加成功后,指令区会显示“ 扩展板”。

### STEP 3 程序编写

1. 打开无线通信功能,在主程序中设置无线通信频道为“7”(控制端和接收端的通信组必须一致)。如下图所示。



2. 需要根据接收到的信号,控制小车做出对应的动作。请根据下面流程图进行编程。



3.分别创建“前进”、“后退”、“左转”、“右转”这四个功能函数，如下图所示。



4.从流程图可以看出，我们需要用到当接收到无线数据模块，根据接收的数据控制小车做不同的运行。具体如下图所示：



拖动方法如下：

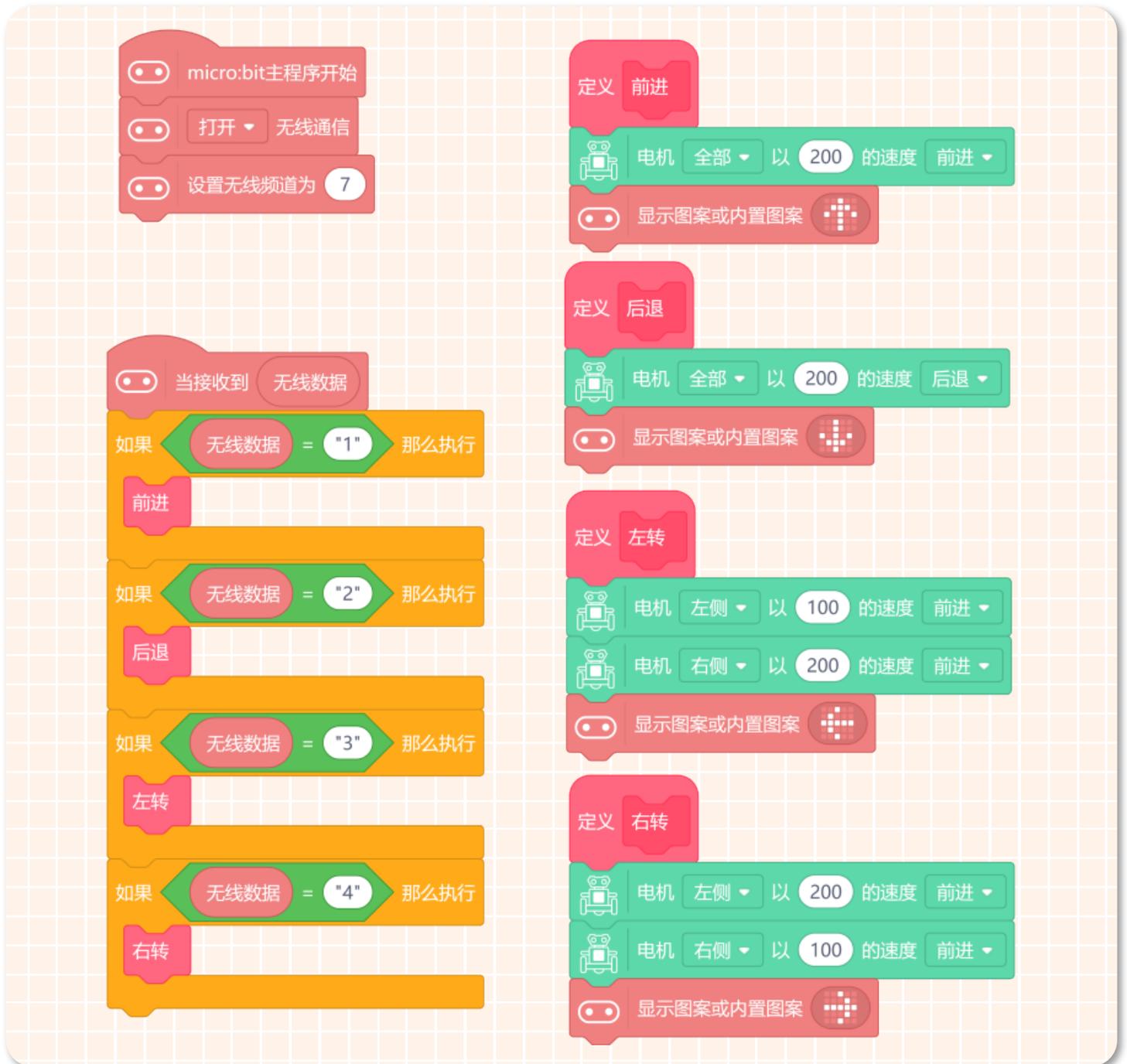


5、下面以前进功能函数为例,将具体的功能在对应的功能函数中实现。为了让运动方式看起来直观一些,当小车向前行驶时,我们加上一个 LED 显示模块,在 micro:bit 上显示向前的箭头。如下图所示。



接下来“后退”、“左转”、“右转”功能函数的实现,就由大家自己完成吧!

6、接收端的完整程序如下图所示:

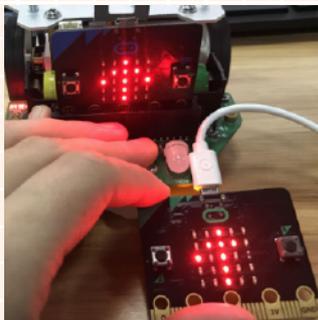


7. 保存,在命名保存区将项目名称改为“体感小车——接收端”,将程序下载到接收端的设备上(麦昆 Plus 小车)。

## 效果展示



完成上面的步骤之后,打开机器人的电源开关,这时候就可以用的控制端的 micro:bit 进行体感操控我们的小车了。快来一起试试吧!



小车前进!

注意: 因为 micro:bit 需要供电才能使用,所以控制端的 micro:bit 需要用 USB 线与电脑连接起来哦!

## 探究思考



在体感游戏中,如果你控制端摆动的角度越大,赛车转向的角度会相对的变大,在这里我们应该怎样实现呢? 这个问题留给大家思考吧!



# 第十三章

## 疯狂赛车

汽车拉力赛的道路状况十分复杂，每一段特殊路段为一个赛程，例如一个赛程全是曲折蜿蜒的山路，另一个赛程则是阴暗森林中的泥路。拉力赛的路线都是一致的，每一辆赛车在不同阶段都由裁判员记录下所需时间，总时间最短的便是胜利者。接下来我们就一起动手将麦昆 Plus 也变成赛车吧！

## 学习目标



了解什么是 PID 算法

## 指令学习



### 模块介绍图

PID 开关模块



维持电机转速平衡。

### 什么是 PID 算法?

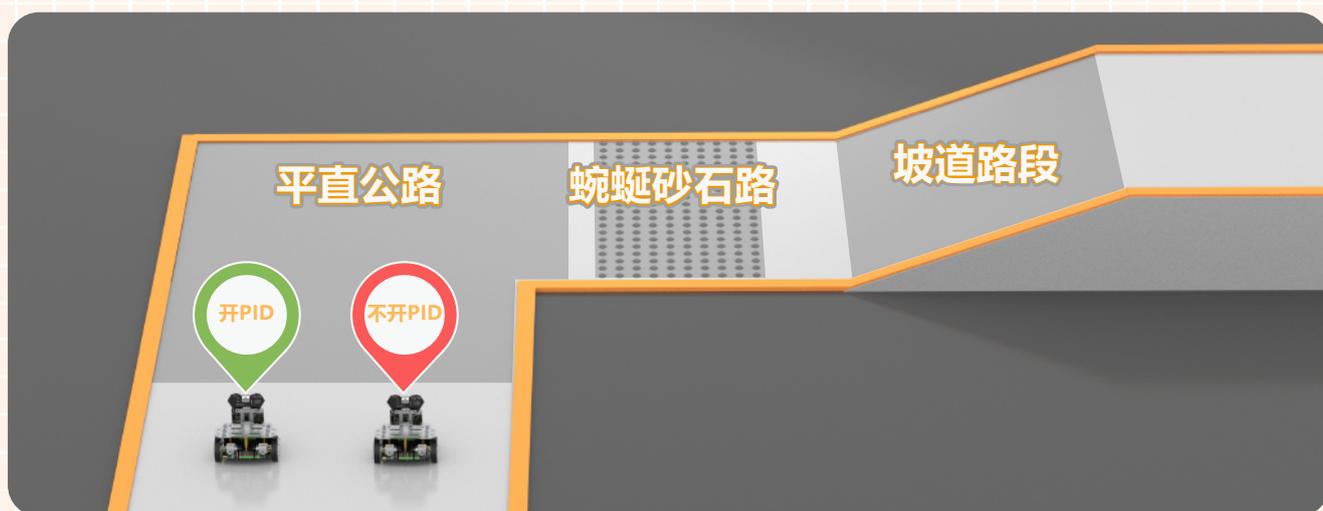
PID 是调速算法,是控制理论中最常用、最经典的算法,PID 算法在生活中的应用也很广泛。它并不是什么很神圣的东西,大家一定都见过 PID 的实际应用:比如四轴飞行器,再比如平衡小车,还有汽车的定速巡航、3D 打印机上的温度控制器、恒温热水器等。就是类似于这种:需要将某一个物理量“保持稳定”的场合(比如维持平衡,稳定温度、转速等),PID 都会派上大用场。

## 动手实践



### 比赛规则:

比赛全程车速规定为 30,赛道由三段不同的赛程组成:第一个赛程为普通公路;第二个赛程为蜿蜒砂石路;第三个赛程为坡道路段。穿过三段赛程先到达终点者获胜。下图是此次比赛的行进路线。



注意:

1. 比赛的赛道需要自己搭建。
2. 因为是赛车比赛,所以在这个项目中需要两个麦昆 Plus 哦!

下面将用两辆小车来学习 PID 的作用,看看开启 PID 和关闭 PID 有什么区别。

## 赛车 1—开启PID

PID  
ON

### STEP 1 新建项目

打开 Mind+ 软件,点击“项目”--“新建项目”新建一个项目。



### STEP 2 添加麦昆 Plus

1、选择“扩展”--“主控板”--“micro:bit”主控板,选择成功如下图:



2、继续点击“扩展板”--“麦昆 Plus”,选择成功如下图:



麦昆 Plus 添加成功后,指令区会显示“



”。

### STEP 3 程序编写

1、在开机程序中,加入 PID 算法并将 PID 的开关状态设置成“开”。如下图所示。



#### 知识要点

PID 算法中到 PID 具体代表什么呢?

P: 调节当前的误差;

D: 预测未来的误差;

I: 纠正过去的误差;

2. 第一个赛程,赛程以 30 的速度行驶 5 秒。



3. 开始第二段赛程之前, 赛车需要完成一个右转后继续直行直到终点。



4. 完整程序如下图所示:



5. 保存项目, 在命名保存区将项目名称改为“赛车 1- 开启 PID”, 然后将程序下载到第一个麦昆 Plus 中。

## 赛车 2—关闭PID



### STEP 1 新建项目

打开 Mind+ 软件, 点击“项目”--“新建项目”新建一个项目。



## STEP 2 添加麦昆 Plus

1、选择“扩展”--“主控板”--“micro:bit”主控板,选择成功如下图:



2、继续点击“扩展板”--“麦昆 Plus”,选择成功如下图:



麦昆 Plus 添加成功后,指令区会显示“



## STEP 3 程序编写

1、在主程序中,加入 PID 算法并将 PID 的开关状态设置成“关”。如下图所示。



2、第一个赛程,赛程以 30 的速度行驶 5 秒。如下图所示。



3、开始第二赛程之前,赛车需要完成一个右转后继续直行直到终点。如下图所示。



4.完整程序如下图所示：



5. 保存项目,在命名保存区将项目名称改为“赛车 2- 关闭 PID”,将程序下载到第二个麦昆 Plus 中。

## 效果展示



完后上面的步骤后,将赛车放在我们搭好的赛道上,同时打开两个赛车的开关,通过观察赛车的行驶情况,得出如下结论:

第一个赛程: 两辆赛车以缓慢的速度前前进,行驶的速度都差不多。

第二个赛程: 赛车转弯之后,我们可以明显的看到,在有砂石障碍物的赛道上,开了 PID 的赛车 1 还在行驶; 关了 PID 的赛车 2 遇到障碍物时停止行驶了。

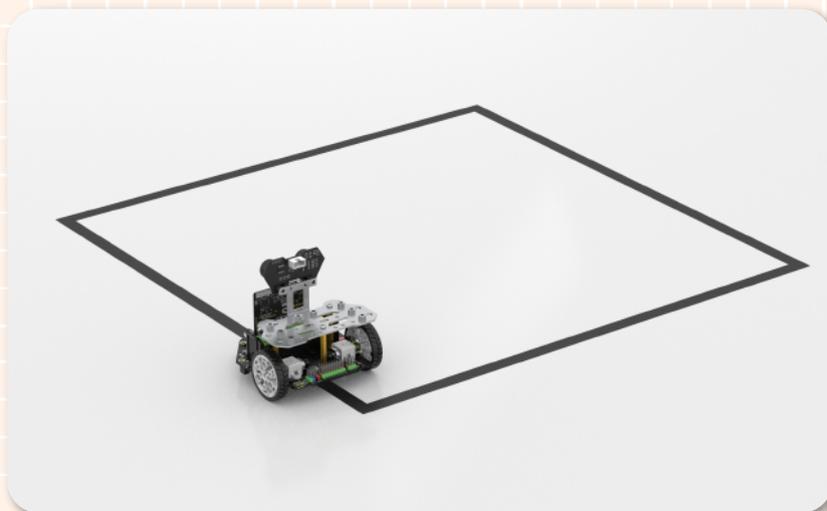
第三个赛程: 在坡道中,开了 PID 的赛车 1 还可以继续行驶; 关了 PID 的赛车 2 还停留在障碍物哪里。

最终开启 PID 的赛车 1 获得胜利。

## 探究思考



通过赛车比赛直观的看到了使用 PID 的优点,使用了 PID 模块,可以维持小车的电机平衡,让小车行驶更加稳定。接下来大家想一想,如何让赛车在正方形的赛道上进行比赛呢?



小提示: 赛道分成两部分: 赛车直行和赛车转 90 度角,不要忘记将 PID 算法模块加上哦!  
开启 PID 可以让小车走的路线更加笔直。



# 第十四章

## 车速记录仪

行车记录仪又叫行车记录器，它能够在司机驾驶的过程中客观和精准的记录全程的行驶状态，例如车子前进和后退、减速与加速、转弯变向等参数。能够将采集到的数据实时显示出来，供驾驶者掌握车况。那么如何才能将麦昆 Plus 的车速也显示出来呢？这就需要 OLED 显示屏来帮忙了。麦昆 Plus 拥有 3 路 I2C 接口，供我们外接 I2C 通信的 Gravity 模块。下面就一起来制作一个车速记录仪吧！

## 学习目标



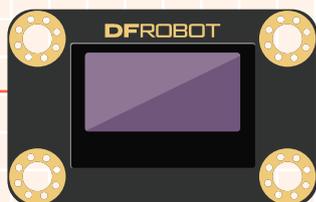
学习如何使用 OLED 显示屏

## 电子模块



### OLED 介绍图

OLED-2864 显示屏



OLED-2864 显示屏是一款无需背景光源,自发光式的显示模块。该模块支持通过 I2C 接口与控制器通信,支持高传输速率,能够实现 60Hz 的刷新频率。

## 指令学习



### 模块介绍图

读取电机速度模块



读取当前电机运行的速度。

初始化 OLED 模块



初始化 OLED 显示屏。

OLED 数字显示模块



将数字显示在 OLED 屏幕上。

## 动手实践



### STEP 1 新建项目

打开 Mind+ 软件,点击“项目”--“新建项目”新建一个项目。



## STEP 2 添加麦昆 Plus

1、选择“扩展”--“主控板”--“micro:bit”主控板,选择成功如下图:



2、继续点击“扩展板”--“麦昆 Plus”,选择成功如下图:



麦昆 Plus 添加成功后,指令区会显示“”。

## STEP 3 添加 OLED 显示屏

1、单击“扩展”,在“显示器”中选择“OLED-12864 显示屏”。



2、OLED 模块添加成功之后,指令区会出现对应的显示器指令模块。如下图所示。



## STEP 4 程序编写

1、因为要使用 OLED 显示屏来显示数据,所以需要在主程序模块中对 OLED 进行初始化。如下图所示。



2、关闭 PID,让小车以 30 的速度前进。如下图所示。



3、因为需要 OLED 显示屏上实时的显示数据,所以将 OLED 数字显示模块放入循环执行模块中。如下图所示。



4、OLED 屏上需要实时的显示当前小车的车速,所以需要将“读取电机速度”模块放在“OLED 显示数字”模块中,这样 OLED 屏上的第 1 行就会显示电机左侧的速度。如下图所示。



### 知识要点

你肯定会对“oled 128\*64 第 1 行显示 读取 左侧 电机速度”这个程序有疑问,在显示屏的第 0 行显示读取的电机速度,显示屏上会有数据显示吗? 第 0 行在哪里? 显示屏上肯定是有数据显示的,在程序中,计数是从 0 开始的,所以这里程序中的第 0 行对应的是 OLED 显示屏上的第 1 行。

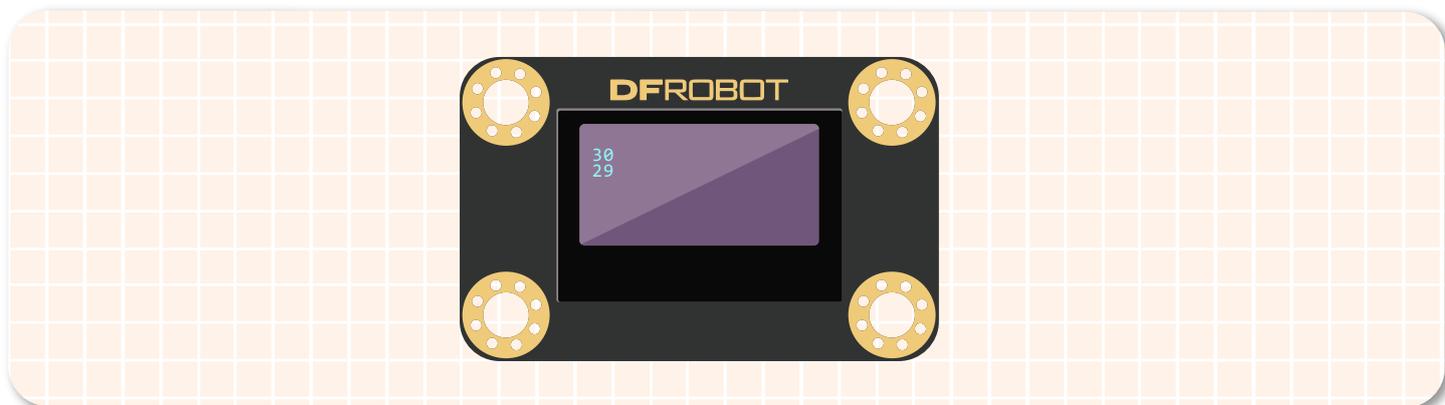
5、再用同样的方式将电机右侧的速度在 OLED 屏的第二行显示,每隔 1 秒刷新一次数据。完整程序入下图所示:



8. 保存项目,在命名保存区将项目名称改为“车速记录仪”,然后下载程序。

### STEP 5 效果展示

完成上面步骤后,打开小车电源开关。我们可以看到 OLED 显示屏的第 1 行显示电机左侧的速度,第二行显示电机右侧的速度,并且数据也在实时的刷新。



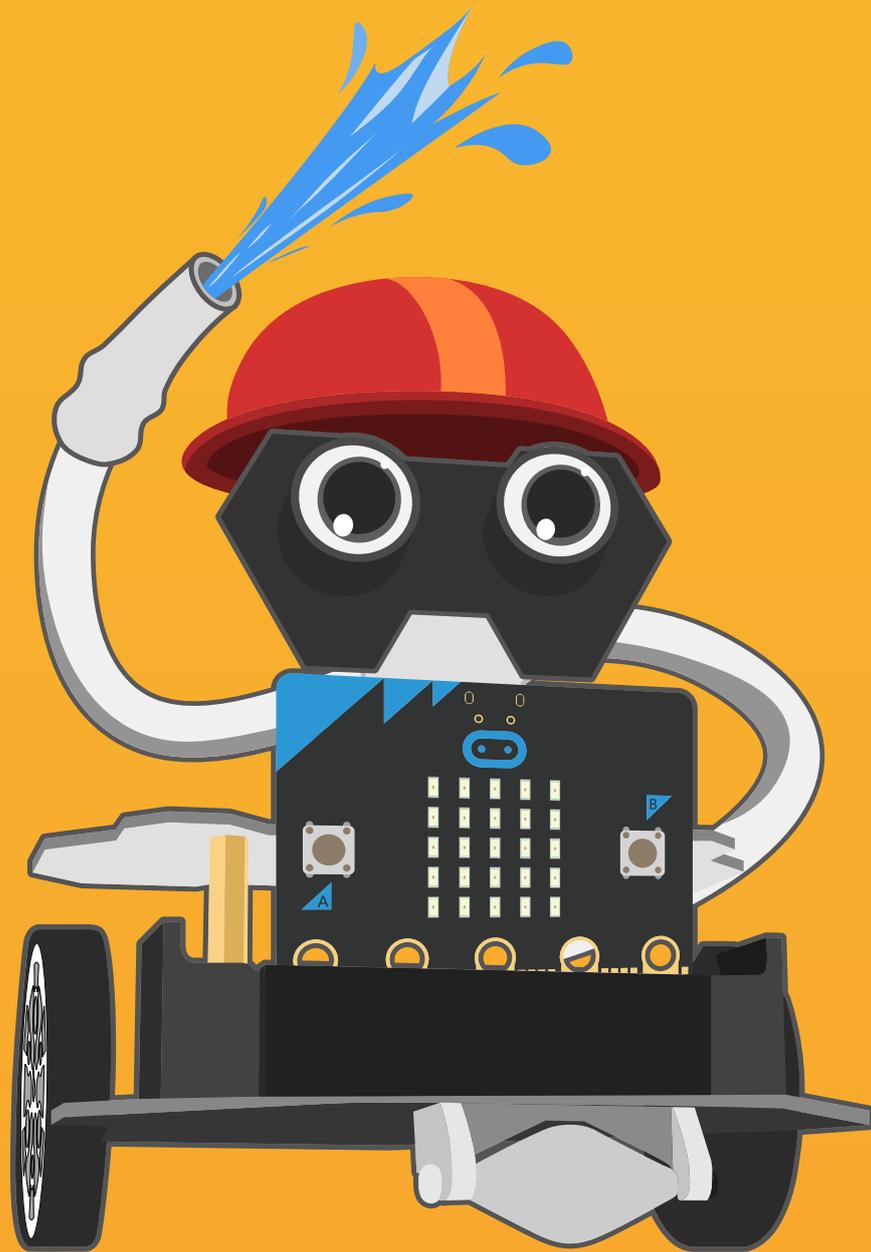
### 探究思考



1. 其实通过观察可以发现,虽然我们将左右两个电机的速度都设置成的 30,但是 OLED 显示屏的数据却有变化,这是为什么呢? 需要怎样做才能使数据变化变小?

小提示: 开启 PID 算法可以维持左右电机转速的平衡哦!

2. 将前面用 LED 点阵屏显示数据的项目,修改成用 OLED 显示屏显示(如感光机器人、测距小达人等项目)。



# 第十五章

## 消防机器人

消防员是特殊的救护职业，也是高危职业。他们主要以扑救火灾、抢险救灾为主。当消防员接到火警电话后，在最快的时间内整队出警到达火灾现场并快速救火。

同学们，消防员叔叔用生命保卫着我们的城市，国家。那我们能否通过自己所学的知识制作出消防机器人(完成“出警——灭火——收队”这三个动作)，帮助他们降低危险呢？消防机器人除了使用麦昆 Plus 外，还需要火焰传感器和舵机来帮忙哦！

麦昆 Plus 拥有 8 路 GPIO 口、3 路舵机口，供我们外接 Gravity 模块和舵机。下面一起来制作“消防机器人”吧！

## 学习目标

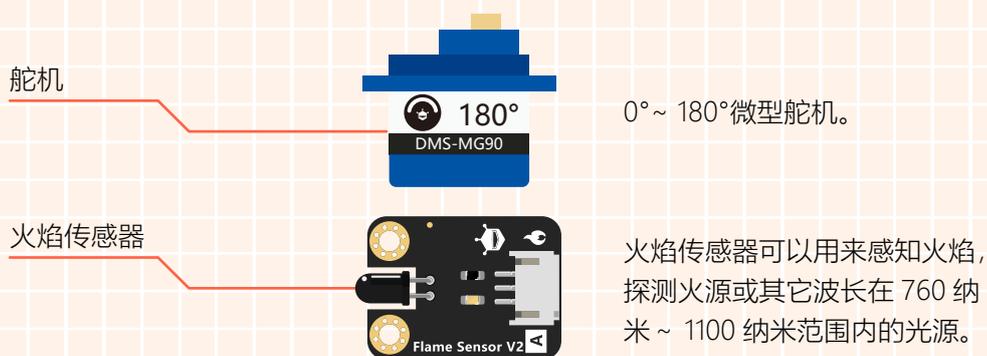


1. 学习如何驱动舵机
2. 学习如何使用火焰传感器

## 电子模块



### 舵机、火焰传感器介绍图



## 指令学习



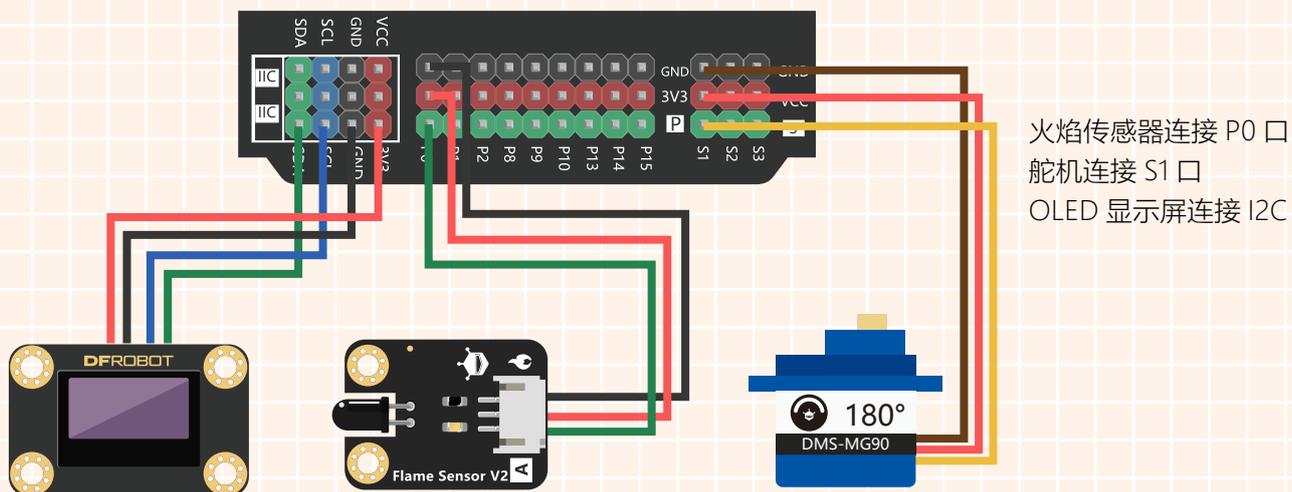
### 模块介绍图



## 动手实践



### 消防机器人硬件连接图



## STEP 2 添加麦昆 Plus

1、选择“扩展”--“主控板”--“micro:bit”主控板,选择成功如下图:



2、继续点击“扩展板”--“麦昆 Plus”,选择成功如下图:



麦昆 Plus 添加成功后,指令区会显示“



## STEP 3 添加 OLED 显示屏

1、单击“扩展”,在“显示器”中选择“OLED-12864 显示屏”。



2、OLED 模块添加成功之后,指令区会出现对应的显示器指令模块。如下图所示。



## STEP 4 程序编写

1、生活中,消防队员们救火出警的过程大概分为三个过程:

第一: 接到报警电话——出警前往火灾现场;

第二: 到达现场——进行灭火;

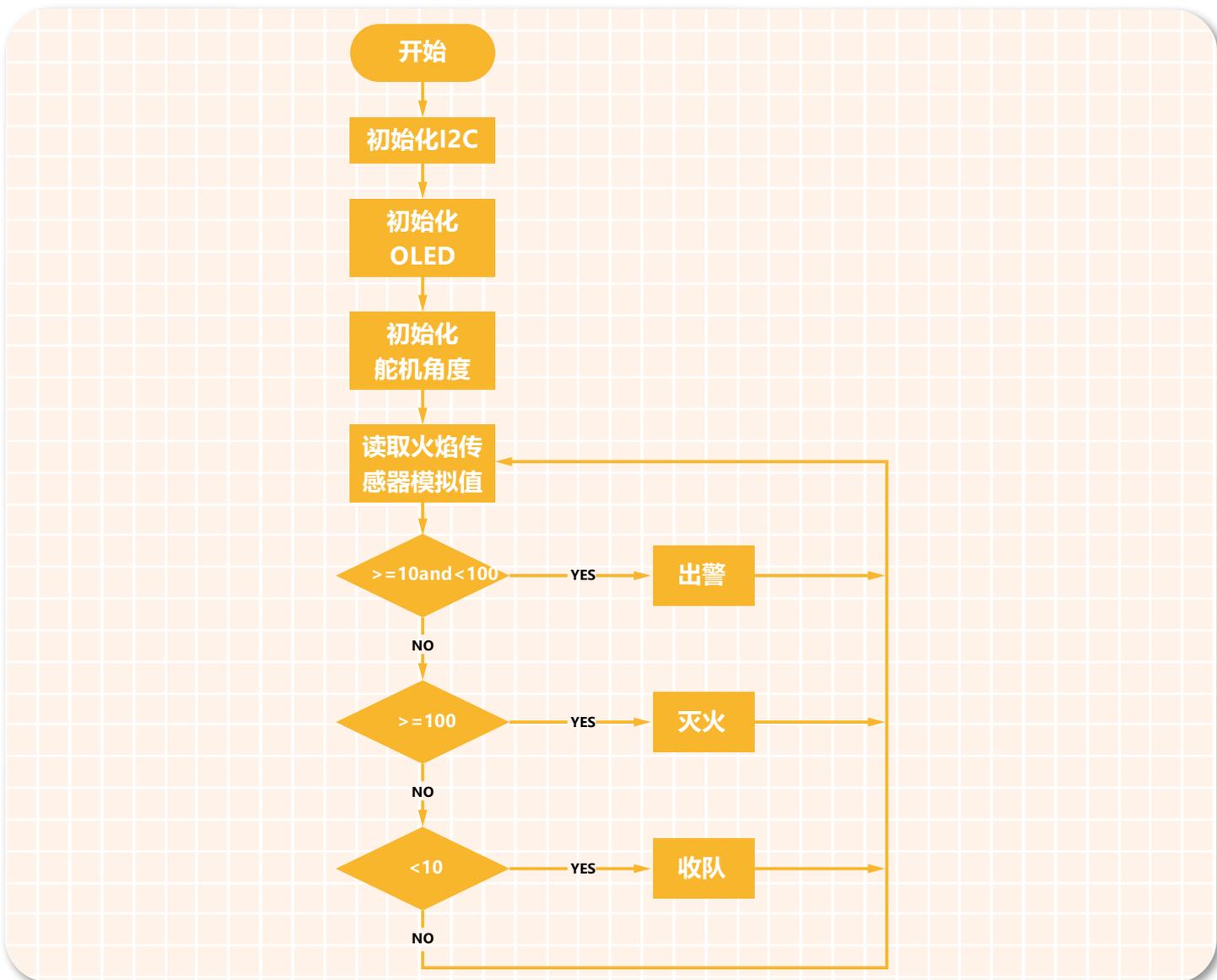
第三: 扑灭火源——收队;

因此我们将消防队员们出警的过程“出警”、“灭火”、“收队”封装成三个不同的功能函数,下面以“出警”功能函数为例。如下图所示。



- ①在指令区用鼠标点击“函数”;
- ②在“函数”指令区点击“自定义模块”;
- ③在“添加一个自定义模块”中,将函数名“积木名称”改成“出警”;
- ④上面步骤完成之后,点击“完成”就可以了。

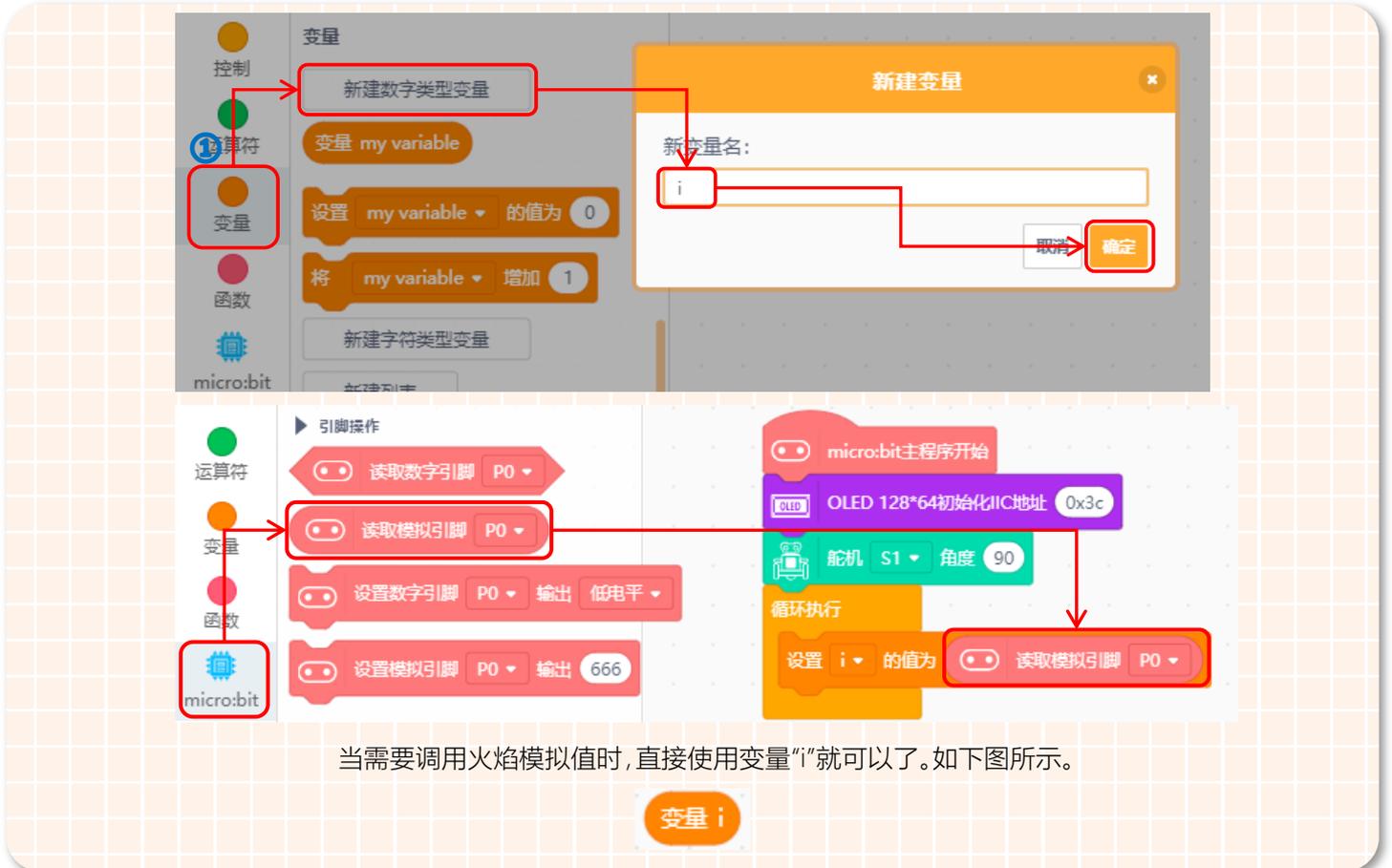
2、再用同样的方法创建“灭火”、“收警”这两个功能函数,功能函数创建成功之后,我们就可以进行条件判断了,具体流程图如下图所示。



3、基于流程图,我们需要将初始化 I2C、初始化 OLED、设置舵机角度的模块放入到开机程序中。如下图所示。



4、新建一个名为“i”的变量,因为后面的显示和条件判断中,需要多次用到火焰传感器读取的模拟值,所以我们将火焰传感器读取到的模拟值存放在变量“i”中。新建变量: i 如下图所示。

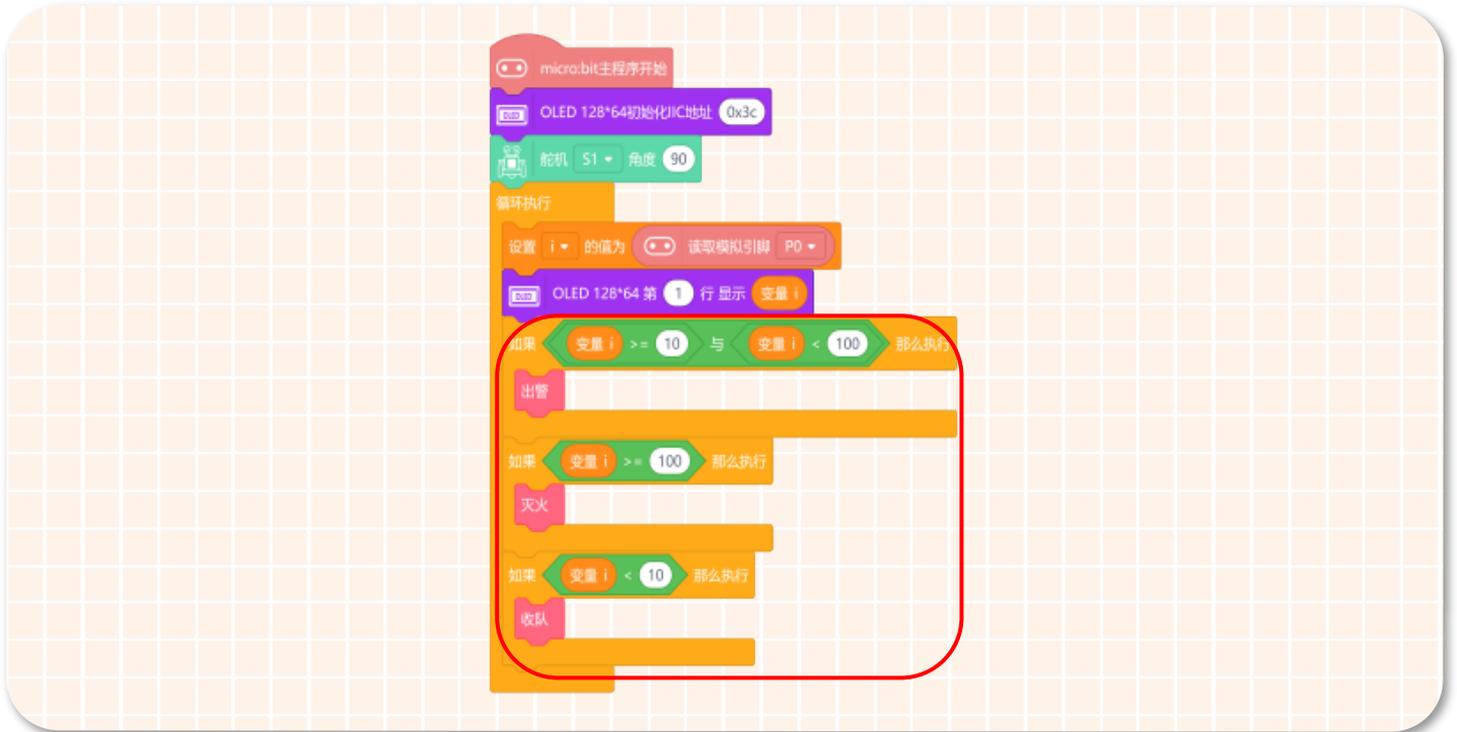


5、在 OLED 屏的第一行,显示火焰传感器检测到的模拟值。如下图所示。



(1) 根据流程图中的条件判断语句,调用对应的功能函数。如下图所示。

当消防机器人检测到的火焰模拟值在 10~100 之间时,说明离火灾现场还有一定距离,消防机器人就出警前往火灾现场;当消防机器人检测到火焰模拟值大于等于 100 时,说明消防机器人到达了火灾现场,消防机器人进行灭火;当火焰机器人检测到的火焰模拟值小于 10 的时候,说明火已经被灭了,消防机器人收队。



6、“出警”功能函数具体实现的功能: 消防车在前往火灾现场的过程中,警灯会红蓝的交替闪烁,同时还要响起警笛(这里我们用音乐“DADADAUM”来模拟消防车的警笛)。如下图所示。



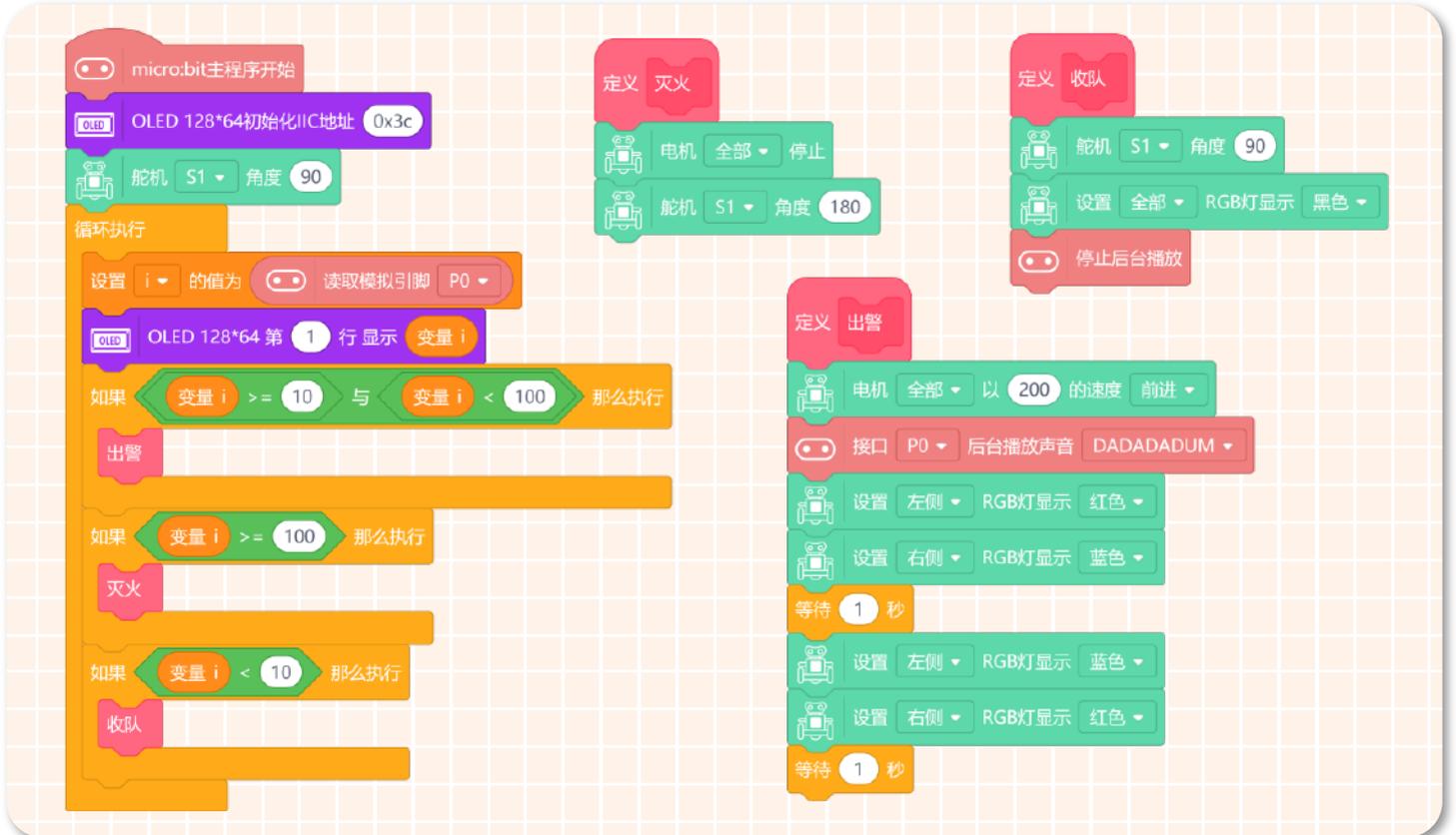
7、“灭火”功能函数具体实现的功能: 消防车到达火灾现场,打开消防水枪进行灭火(这里通过改变舵机的角度来模拟消防水枪的开/关)。如下图所示。



8、“收队”功能函数具体实现的功能：灭火完成之后,关闭消防水枪,同时关闭警灯和警笛。如下图所示。



9、消防机器人项目的完整程序如下图所示：



10、保存项目,在项目命名区将项目名称改为“消防机器人”,然后上传程序。

### STEP 5 效果展示

完成上面步骤之后,打开消防机器人的电源开关。可以看到火焰传感器采集的模拟值会实时的显示在 OLED 屏上,当模拟值大于 10 小于 100 的时,消防机器人以 200 的速度前进,在前进的过程中亮起警灯响起警笛;当模拟值大于 100 时,消防机器人会停止前进,舵机转动到 180 度;当模拟值小于 10 时,舵机转回到 90 度的状态并且停止播放音乐,关闭 RGB 灯。

注意：可以用打火机来模拟火灾现场,虽然火焰传感器可以检测火焰,但是它并不防火。因此使用时请与火焰保持距离,以免烧坏传感器。项目中存在危险动作,请监护人或老师协助完成。

### 探究思考



大家都知道吸烟不仅危害自己的身体健康还严重的危害他人身体健康,因此在一些公共场所是禁止吸烟的。那是否可以制作一个“禁烟机器人”呢?用来实时监测有没有人吸烟。

小提示：外接烟雾传感器就可以了哦!

## 附录1.麦昆plus模块说明



从技术层面来讲,麦昆 plus 就相当于我们的设备,例如家里的电视、空调。而指令模块就是让设备运作起来的工具,例如遥控器。如果没有下面这些指令模块,麦昆 plus 就是冷冰冰的铁块。所以我们需要合理运用下面的指令模块让麦昆“动起来”。那接下来让我们起来认识一下这些指令模块。



PID 算法,可以维持电机的转速平衡,在使用到电机时,都可以开启 PID 算法来维持电机的转速平衡。  
只有“打开 / 关闭”两种状态。



可以分别设置“左 / 右 / 全部”电机的速度和当前状态。  
速度范围: 0~255。  
当前状态: 前进,后退。



可以分别设置“左 / 右 / 全部”电机停止运动。



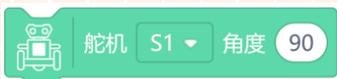
当两个电机存在转速差时,可以通过设置“左 / 右”电机补偿速度来调节转速差。  
可调节范围是 0~255。



读取“左 / 右”电机的当前速度是多少。



读取“左 / 右”电机的方向,  
0: 电机停止;  
1: 电机正转;  
2: 电机反转。



根据硬件连接设置“S1-S3”舵机转动的角度,  
可设置范围: 0°~ 180°



设置“左 / 右 / 全部”RGB 灯的显示颜色,可以设置 7 种颜色。也可以关闭 RGB 灯显示。



读取巡线传感器

L1 ▾

分别判断“L1、L2、L3、R1、R2、R3”巡线传感器返回的值是 1 还是 0。以地图黑线为例：检测到黑线返回 1，未检测到黑线返回 0。



读取巡线传感器

L1 ▾

灰度值

分别读取“L1、L2、L3、R1、R2、R3”巡线传感器检测到的灰度值。



获取版本

获取麦昆 plus 的当前版本号。



读取超声波传感器 trig

P0 ▾

echo

P1 ▾

单位

cm ▾

检测超声波与障碍物之间的距离。trig：发射端；echo：接收端。需要根据硬件连接设置 trig 和 echo 所对应的 GPIO 口。



当红外接收到数据

红外接收模块，接收来自外部的红外信号。



红外数据

读取接收到的红外信号。  
以十进制的码值展示。