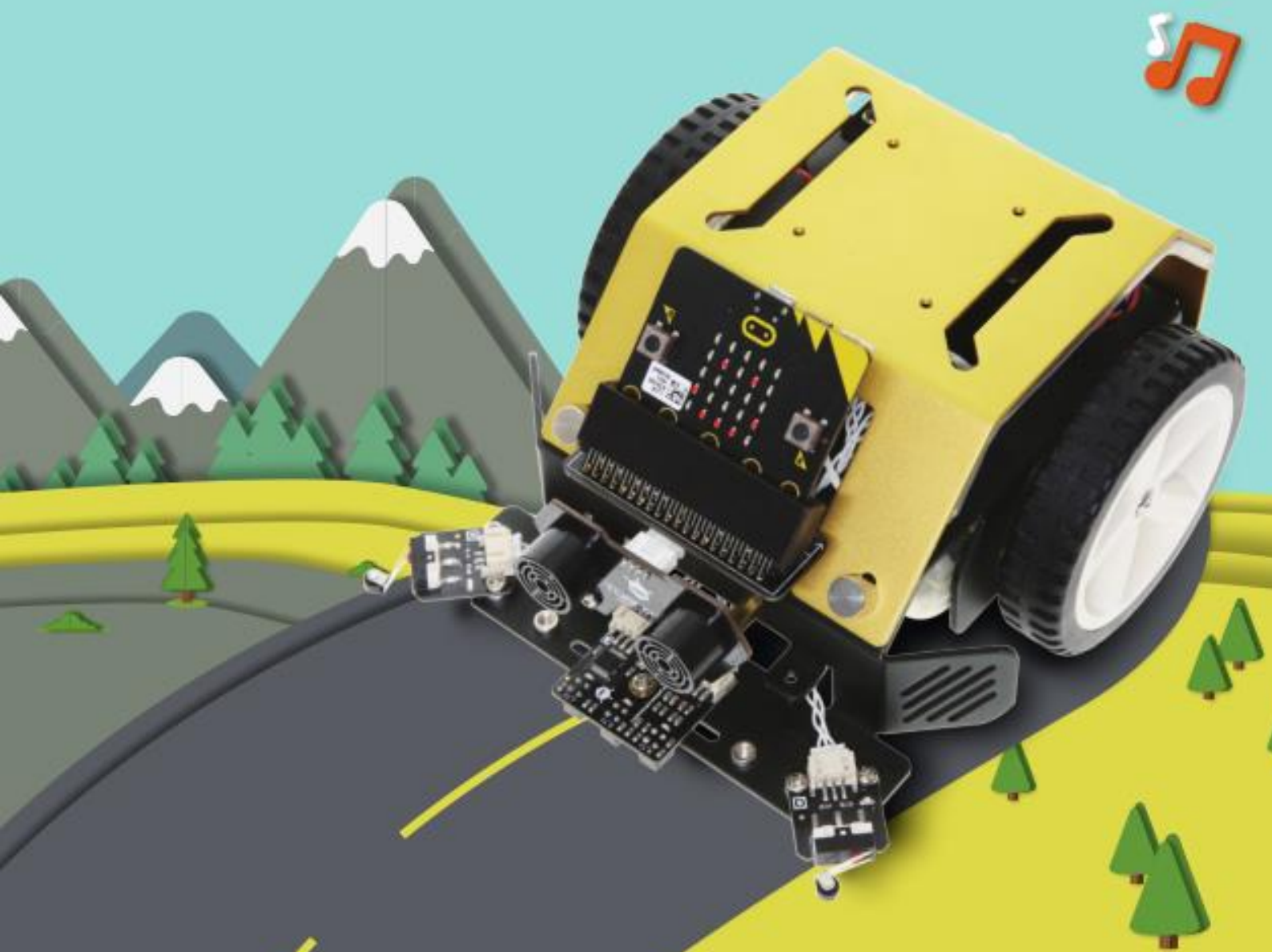




DFROBOT
DRIVE THE FUTURE

MAX:BOT

机器人探索之旅



目录

初识 Max:bot 机器人	1
机器人	1
Max:bot 机器人	1
Max:bot 的装备	1
初识 Micro:bit	3
Micro:bit	3
Micro:bit 的编程方式	4
Micro:bit 的 Mind+图形化编程	5
软件下载	5
界面认知	5
安装驱动及程序上传	6
第一章 小车快跑	9
学习目标	9
电子模块	9
1.1 小车动起来	9
知识要点	9
动手实践	10
探究思考	12
1.2 小车走圆形	12
知识要点	12
动手实践	12
探究思考	14
1.3 小车走方形	14
知识要点	14
动手实践	14
探究思考	17
第二章 感光的眼晴	17
学习目标	17
电子模块	18
2.1 探秘光强的大小	18
知识要点	18

动手实践.....	18
探究思考.....	19
2.2 飞蛾机器人.....	19
知识要点.....	19
动手实践.....	20
探究思考.....	21
第三章 聪明的 Max:bot.....	21
学习目标.....	21
电子模块.....	22
3.1 害羞的 Max:bot	22
知识要点.....	22
动手实践.....	23
探究思考.....	25
3.2 迷宫游戏.....	25
知识要点.....	25
动手实践.....	26
探究思考.....	27
第四章 秘密武器超声波.....	27
学习目标.....	27
电子模块.....	28
4.1 超声波测距.....	28
知识要点.....	29
动手实践.....	29
探究思考.....	30
4.2 行车安全仪.....	31
知识要点.....	31
动手实践.....	31
探究思考.....	33
第五章 勇往直前的 Max:bot.....	33
学习目标.....	33
电子模块.....	34
5.1 悬崖勒马.....	34
知识要点.....	35

动手实践.....	36
探究思考.....	38
5.2 循“轨”蹈矩.....	38
知识要点.....	39
动手实践.....	39
探究思考.....	41
第六章 无线通信	41
学习目标.....	42
电子模块.....	42
6.1 Say Hi.....	42
知识要点.....	42
动手实践.....	43
探究思考.....	44
6.2 体感赛车.....	45
知识要点.....	45
动手实践.....	46
探究思考.....	48
第七章 声光互动	49
学习目标.....	49
电子模块.....	49
7.1 炫彩灯带	49
知识要点.....	50
动手实践.....	50
探究思考.....	51
7.2 演奏美妙音乐	52
知识要点.....	52
动手实践.....	52
探究思考.....	52
7.3 圣诞机器人.....	53
知识要点.....	53
动手实践.....	54
探究思考.....	56

*点击页面左上角 DF 创客社区，有意外惊喜哦！

初识 Max:bot 机器人

机器人

在介绍新朋友 Max:bot 机器人之前，先来了解一下什么是机器人吧！

机器人是具备一些与人或生物相似的智能能力的自动化机器。和人体一样，机器人也由几大“器官”构成：

机器人的大脑——主控制器：主控制器是机器人的核心部件，具有逻辑能力，用于处理接收到的信息与发送控制指令。

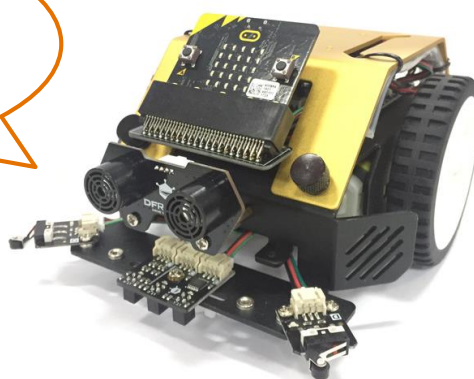
机器人的感知系统——传感器：就像人可以通过眼睛和耳朵感受世界，机器人也有自己的“眼睛和耳朵”，各种不同的传感器可以接收不同的外部信息，例如光线传感器可以感知环境光强度。

机器人的执行系统——驱动器和执行器：机器人的执行系统就像人体的手足，可以用于改造外部环境或向外部发送信息。例如喇叭模块可以发出声音。

Max:bot 机器人

Max:bot 机器人是一款与 Micro: bit 搭配使用的低门槛，功能丰富的移动平台，它的外形是如同金色甲壳虫的酷炫小车。通过编程它可以实现很多功能哦，例如，光线感知，躲避障碍等等。从现在开始，和 Max:bot 成为好朋友，一起来学习吧！

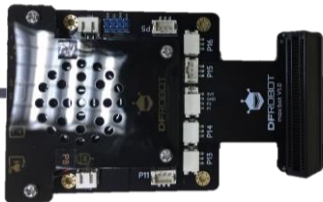


大家好！我是 Max:bot!
和我成为好朋友吧！



Max:bot 的装备

元件名称	图片	功能说明及引脚
------	----	---------

Micro:bit		Micro:bit 是一款由英国 BBC 设计的 ARM 架构的单片机，板载蓝牙，加速度计，电子罗盘，三个按钮，5×5 LED 点阵。
电路板		电路板可以搭载 Max:bot 的超声波传感器、喇叭等多种传感器和驱动器。
碰撞传感器		碰撞传感器类似于按钮，可以检测到是否碰撞到障碍物。左侧碰撞传感器引脚为 P16、右侧碰撞传感器引脚为 P13。
超声波传感器		发射端发射出超声波，超声波遇到障碍物反射回接收端。超声波传感器发射端引脚为 P1、接收端引脚为 P2。
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，也可以检测黑背景的白线。共三个检测头，但只用到了左右两个，左侧引脚为 P15、右侧为 P14。

喇叭		喇叭模块是输出模块，可以播放出音乐，引脚为 P0。
灯带		灯带模块可以设置灯珠的颜色，亮度以及动画效果，左侧灯带引脚为 P5，右侧灯带引脚为 P11。
电机		电机是可以带动小车运动的，实现转弯或直行，左侧电机引脚为 P12，右侧电机引脚为 P8。

P8

初识 Micro:bit

Micro:bit

Micro:bit 是一款操作简单，功能强大，价格低廉，为初学者设计的微型电脑。



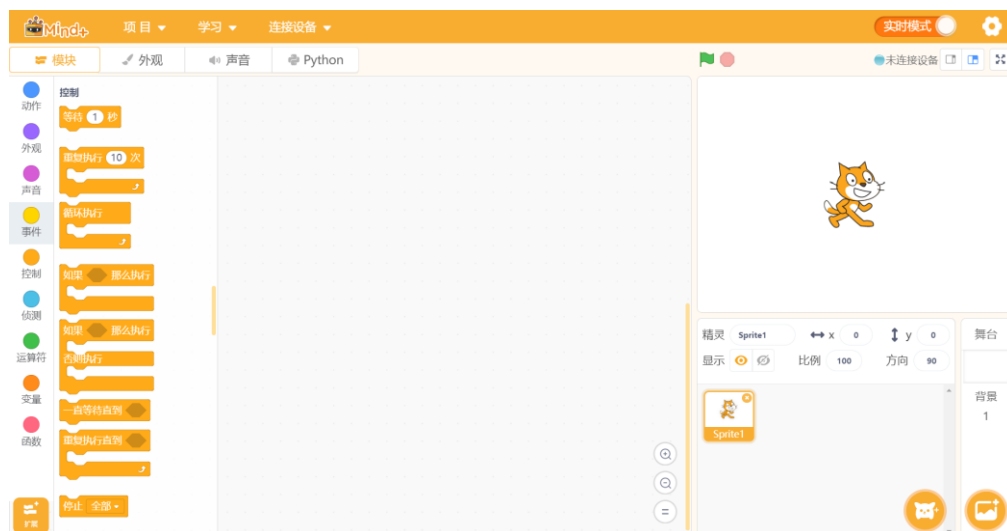
- 一块 4cm x 5cm 的小板子
- 可以塞进口袋的超迷你计算机
- 内嵌的加速度计、磁力计和低功耗蓝牙
- 集成了 5X5 的 LED 灯点阵
- 集成了温度和光敏传感器

Micro:bit 搭载的处理器能满足绝大多数机器人相关学习和开发场景。

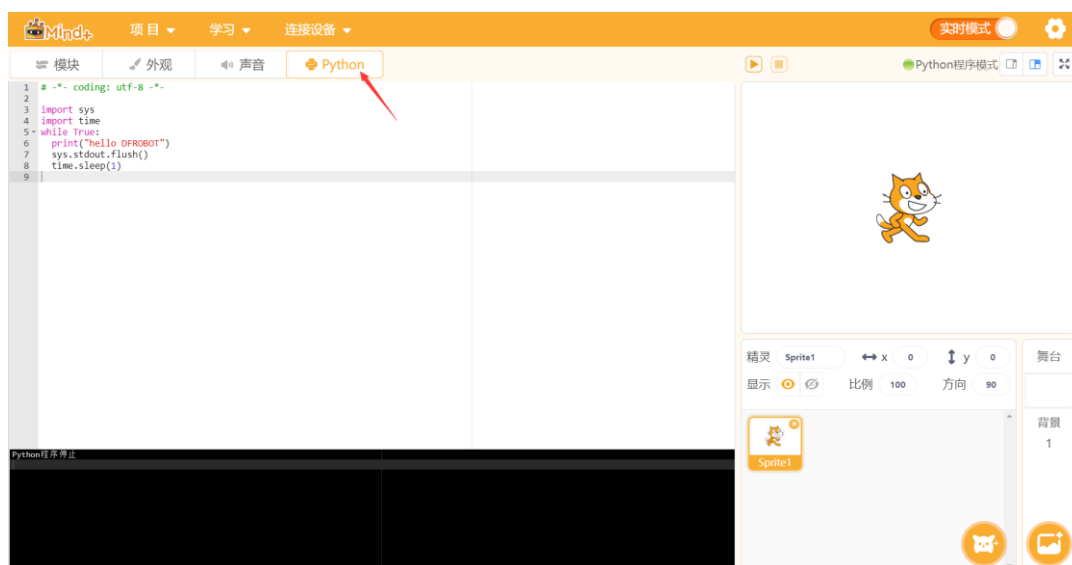
Micro:bit 的编程方式

Mind+是一款基于 Scratch 3.0 开发的图形化编程软件，不仅适合中小学生的学习，而且还能为提高自身技能的“创客”提供 python、c、c++等高级编程语言的学习。支持 Micro:bit 等各种开源硬件，只需要拖动程序块即可完成编程，让大家轻松体验创造的乐趣。

首次打开 Mind+软件界面如下图所示（实时模式下）：



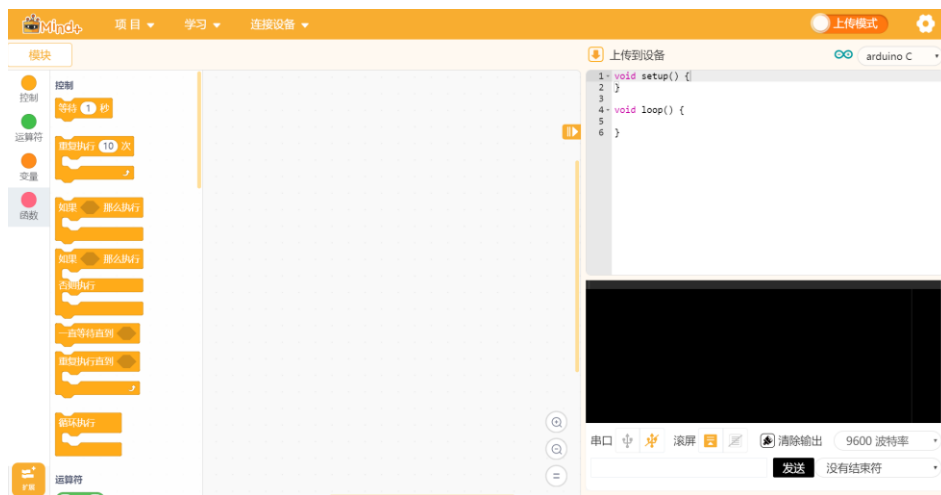
Mind+软件同样支持使用 Python 语言编程。



本教程均是在“上传模式”下进行的，点击按钮，切换上传模式。



本教程中，使用的是 Mind+的图形化编程。



Micro:bit 的 Mind+图形化编程

软件下载

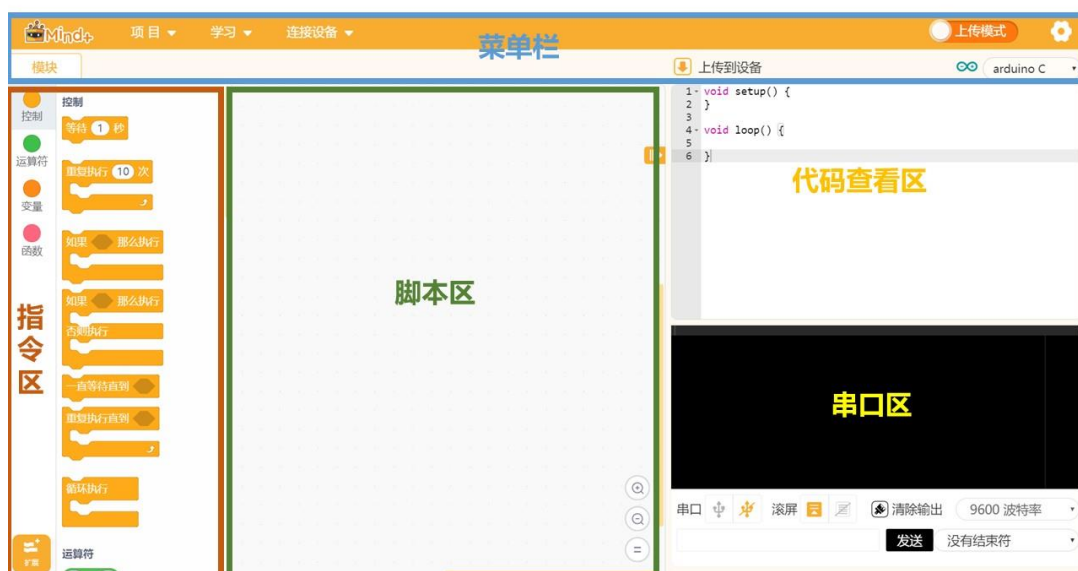
在浏览器中输入以下网址，进入下载界面，点击下载即可。

下载地址：<http://mindplus.cc>



界面认知

按照功能划分，Mind+界面可分为以下几个部分（上传模式）：



菜单栏：包括设置软件的各项功能，学习资料等。

“项目”菜单：可以新建项目、打开项目、保存项目等。

“学习”菜单：在这里能找到很多使用教程和示例程序哦！

“连接设备”菜单：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”按钮：切换程序执行的模式。“上传模式”是将程序上传到硬件设备中执行；“实时模式”是将脚本区可执行的程序在硬件和 Mind+舞台中实时执行。

本教材都是在“上传模式”下进行的。



“（设置）”按钮：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：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。在“扩展”里，可以选择更多额外的道具，支持各种硬件编程。

脚本区：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：显示图形化指令相对应的代码。

串口区：这里能显示下载状况，比如可以看到程序运行状况，显示串口通信数据等。这里还有：串口开关、滚屏开关、清出输出、波特率□、串口输入框、输出格式控制。

安装驱动及程序上传

1、安装驱动

点击“学习”菜单下的“视频教程”。

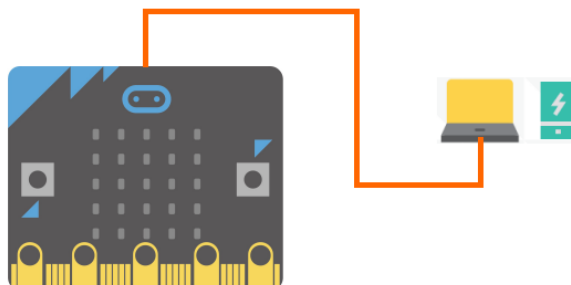


点击“安装驱动”教程视频，打开课程视频悬浮窗，根据课程视频一步一步完成驱动安装，只需安装一次，再次打开 Mind+ 无需安装。



2、上传程序到设备

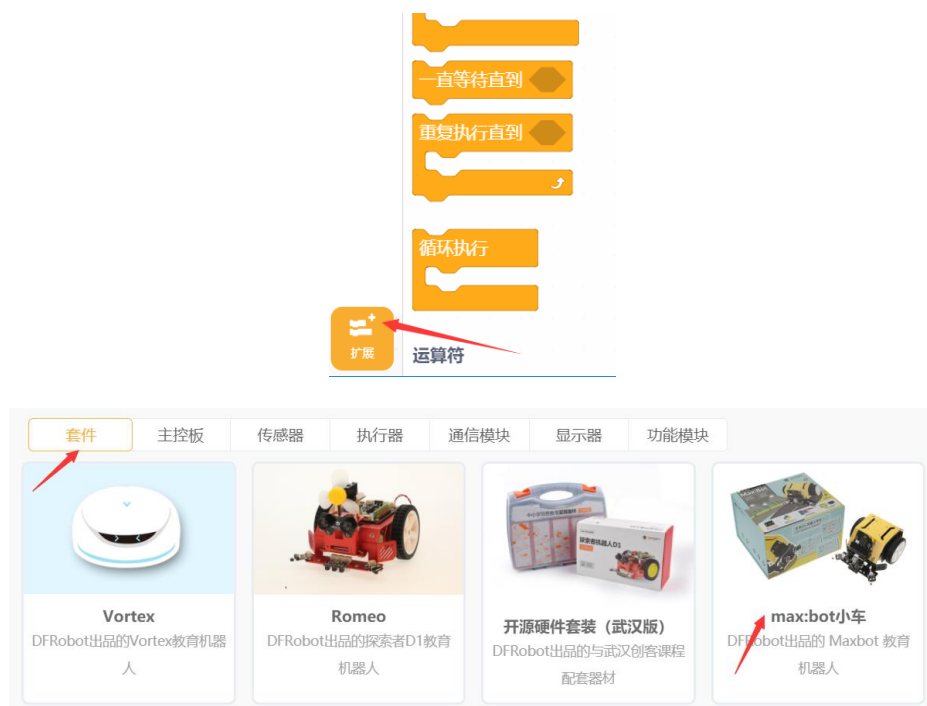
通过 USB 连接线将 Micro:bit 和电脑相连接。



点击菜单栏“项目”菜单——新建项目



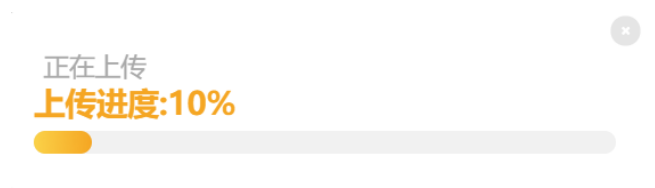
选择界面左下角“扩展”——“套件”——“Max:bot 小车”



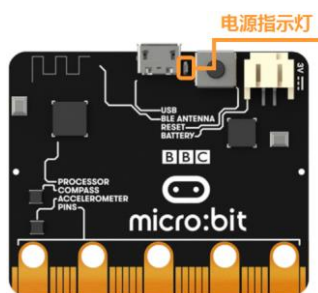
点击菜单栏上“连接设备”，找到并且点击串口号“COM-Microbit”，软件连接到设备。



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。

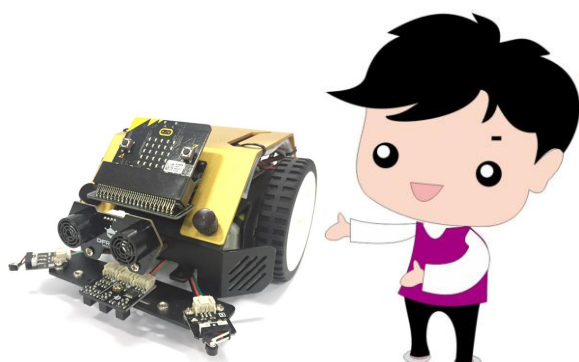


上传过程中 Micro:bit 主板背面的电源信号灯会闪烁，当上传完成后电源信号灯停止闪烁，保持常亮。



第一章 小车快跑


金灿灿的 Max:bot 机器人，好神奇呀！是不是已经迫不及待的要和 Max:bot 玩游戏了，那么先让 Max:bot 动起来！



学习目标

- 1.学习让轮子动起来
- 2.学习差速转向的原理

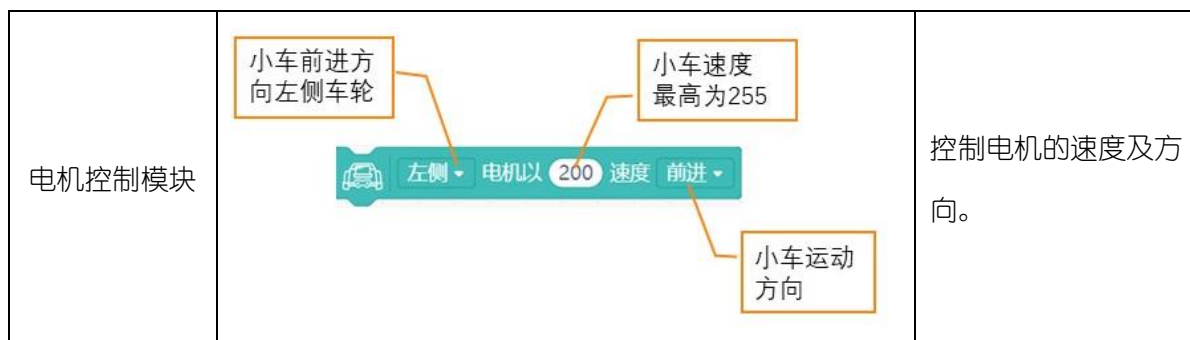
电子模块

电子模块	图片	功能说明
电机		电机是可以带动小车运动的，实现转弯或直行，左侧电机引脚为 P12，右侧电机引脚为 P8。

1.1 小车动起来

知识要点




名称	模块	功能说明
Maxbot 主程序开始模块		部分程序只有放在该模块下才能上传执行。



动手实践

- (1) 打开 Mind+ 软件，新建一个项目。
- (2) 删掉不需要的“循环执行”模块。

小贴士

将鼠标指针移动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！



- (3) 将电机控制模块拖至“Maxbot 主程序开始”下。



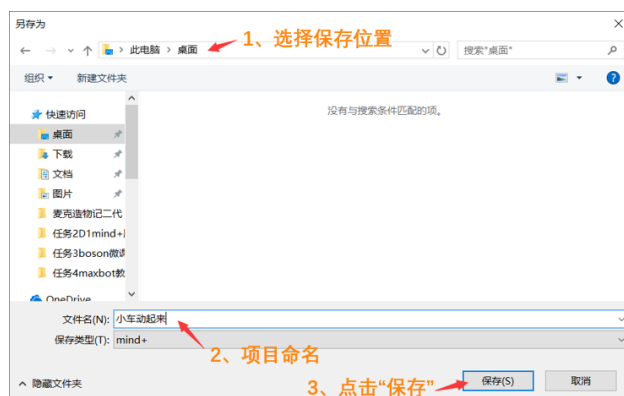
(4) 重复步骤 (3)，将“左侧”改为“右侧”。

小贴士

同一个模块是可以被多次拖拽使用的呢，模块中带有“▼”都是可以按下进行选择的哦！

而像“200”是可以使用键盘输入数字或者通过拖动滑块来改变填入的数字哦！

(5) 最终程序如下图所示，保存项目，命名为“小车动起来”。

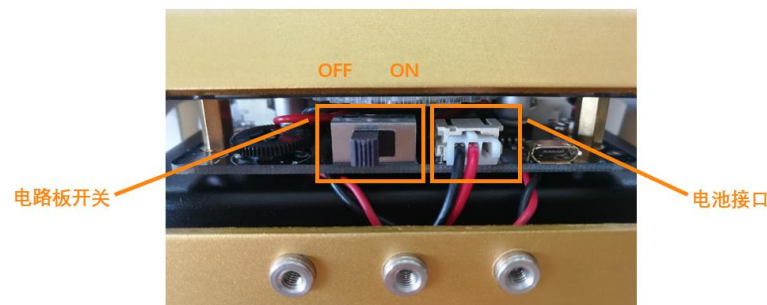


(7) 将程序上传到 Max:bot 中。

上传程序完成后拔掉接在电脑上数据线，将电池的线接好，打开电路板开关。

小贴士

为了防止小车上传程序后突然开始运动，要记得先关闭小车后面的电路板开关再进行程序上传哦！如果小车的前进方向不对，那就检查一下电机的引脚有没有插反吧！



探究思考

怎么让小车一直低速后退呢？试一试吧。

提示：需要改变电机控制模块的方向和速度，可参考知识要点中对于该名称功能说明。



1.2 小车走圆形

知识要点

名称	模块	功能说明
延时模块		持续执行上一个动作或状态。

动手实践

- (1) 打开 Mind+软件，新建项目。
- (2) 小车顺时针移动

根据差速转向原理，设置左侧电机速度为 200，方向为前进；右侧电机速度为 100，方向为前进。

小贴士

差速转向原理 通过控制左右两个驱动轮的转速实现小车转向。当小车左右两个车轮的转速与转动方向均相同时，小车能够前进或者后退；两个车轮的转速相同而方向相反时，小车就会原地转圈；两个车轮转动方向相同转速不同时，小车的行驶方向就会发生变化，当某一侧车轮的转动速度比另一侧的慢时，小车的行驶方向就会偏向这一侧。

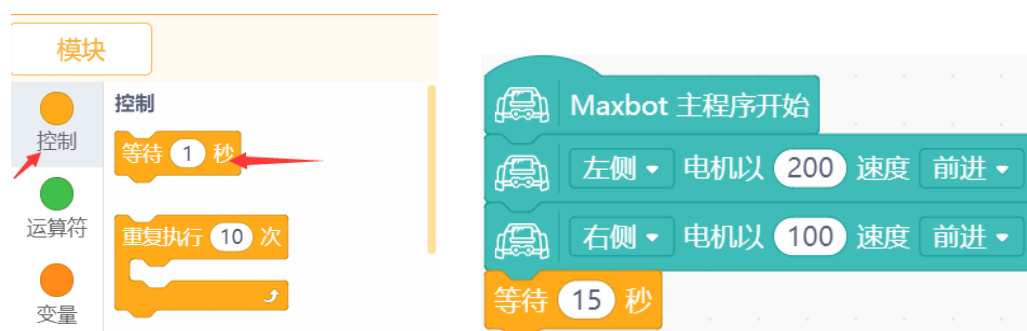


(3) 小车走圆形

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。

小贴士

小车走一个完整的圆形路线所需的时间长短，即等待时间长短，与地面摩擦力、电池电量等因素有关，因此，要根据实际情况进行调整哦！



(4) 小车停止

设置左侧电机速度为 0，方向为前进；右侧电机速度为 0，方向为前进。完整程序如下。



(5) 保存项目，命名为“小车走圆形”，将程序下载到 Max:bot。

探究思考

小车的左右两个电机在不同速度差和不同方向下，小车的运动状态应该如何呢？不同速度差下的小车行进路线是否有变化呢？按照第 1 组的例子，动手编写程序完成下面表格吧！

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	100	前进	顺时针走圆形
2	200	前进	50	前进	
3	100	前进	200	前进	
4	50	前进	200	前进	
5	200	后退	200	前进	

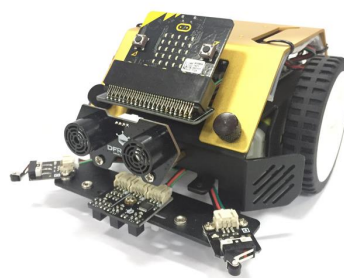
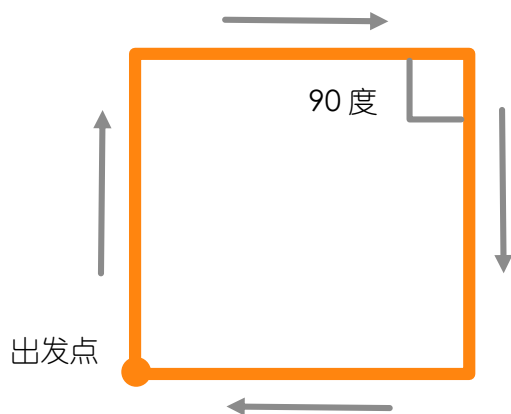
1.3 小车走方形

知识要点

名称	模块	功能说明
循环模块		无限次重复执行模块中的动作。
重复执行模块		有限次重复执行模块中的动作，可自定义重复执行次数。

动手实践

- (1) 打开 Mind+ 软件，新建项目。
- (2) 下图是小车的行进路线，我们可以将小车走方形的过程分成两部分：小车直行和小车转 90 度角，通过观察路线我们可以发现小车只要完成 4 次直行转 90 度就可以走一个完整的方形啦！



小车直行的程序在前面已经介绍过了，大家有没有写出来呢？根据差速转向原理，要想让小车原地顺时针转过 90 度角，需要小车**右轮不转**，**左轮转动**，小车实现转弯。



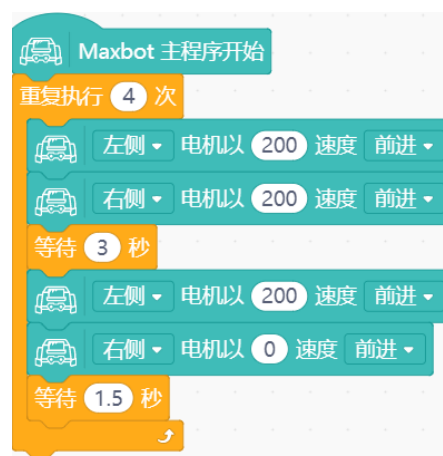
(3) 添加**延时模块**使小车实现直行转弯。



小贴士

第一个延时模块是设置小车直行的时间，调整等待时间可以控制小车前进的距离；第二个延时模块是设置小车转过 90 度，调整等待时间可以控制小车转过的角度。小车转过 90 度所需的时间长短，即等待时间长短，与地面摩擦力、电池电量等因素有关，因此，要根据实际情况进行调整哦！

(4) 添加**重复执行模块**，重复执行 4 次上面的直行转弯程序。



(5) 在重复执行模块外加入使左轮右轮停止的模块指令。

小贴士

为什么要加入使左轮右轮停止的模块？这是因为在重复 4 次直行转弯后，将跳出重复执行模块，而重复执行模块外面没有其他模块指令，所以小车状态保持在最后执行的转弯命令，这时上传程序我们会看到小车在最后一一直原地转圈哦！

最终完整程序如下：



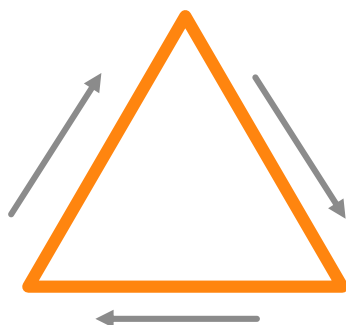
(6) 保存项目，命名为“小车走方形”，将程序下载到 Max:bot。

尝试一下使用循环模块代替重复执行模块后，观察小车是怎么运动的呢？



探究思考

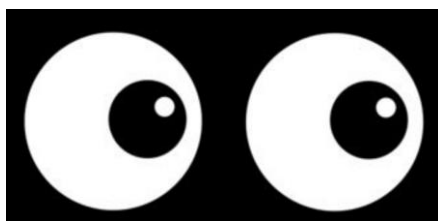
怎么让小车可以走三角形呢？



第二章 感光的眼睛

眼睛是心灵的窗户，眼睛不仅可以让我们看清这个美好的世界，还可以帮助我们表达喜怒哀乐。

Max:bot 也有一双眼睛，但是和我们的有一些不同，那么它的这双眼睛到底有什么神奇之处呢？

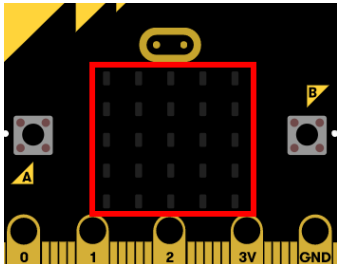


Max:bot 的眼睛可以感知光，并且可以准确知道光的强度，一起来探索这个功能吧！

学习目标

1. 学习光线传感器
2. 运用数字显示模块
3. 运用流程图辅助编程

电子模块

电子模块	图片	功能说明
光线传感器		Micro:bit 的图案显示部分还可以感知光的强度值。具有感光的元件，可以直接将光的强度转化为电信号输出。

2.1 探秘光强的大小

人类的眼睛在很强的光照下，会通过眯眼来保护眼睛，而在光照比较弱的情况下，会看不清东西。那么我们怎么知道光的强度的变化呢？



这就需要 Max:bot 来帮忙了，先来试试使用 Micro:bit 把光强用数字表达出来吧！

知识要点

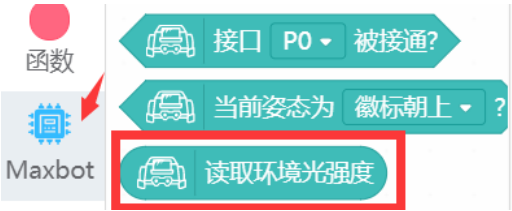
名称	模块	功能说明
读取环境光强度 模块		读取环境光的强度值，范围为“0~255”，表示光的强度由弱至强。
显示模块		显示英文和数字

动手实践

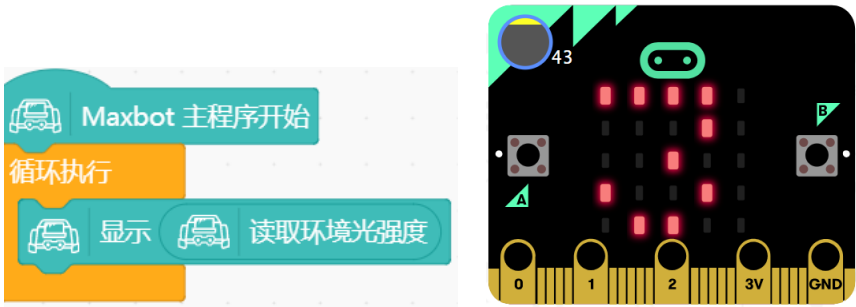
- (1) 打开 Mind+软件，新建项目。
- (2) 将显示模块放入循环模块中，这样 Max:bot 的眼睛就可以一直读取光强值啦，显示模块默认显示 “hello world”。



(3) 获取环境光强度的值



(4) 保存项目，命名为“感光眼睛”，将程序下载到 Max:bot，就可以观察到光强的大小啦！



探究思考

去探究一下每个房间的光照强度是多少呢？思考，各个环境的光强设计合适吗？

环境	光强
卧室	
厨房	
客厅	

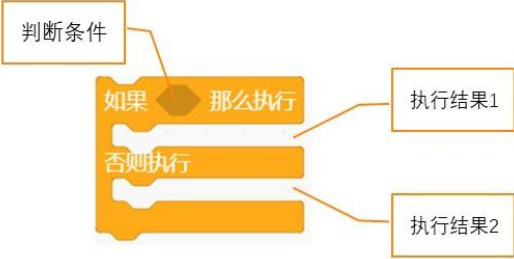

2.2 飞蛾机器人

你听过飞蛾扑火的故事吗？飞蛾是一种喜光的昆虫，它在夜晚会聚集在光亮的地方。利用 Max:bot 可以感知光强大小的“眼睛”来制作一个飞蛾机器人吧！



知识要点

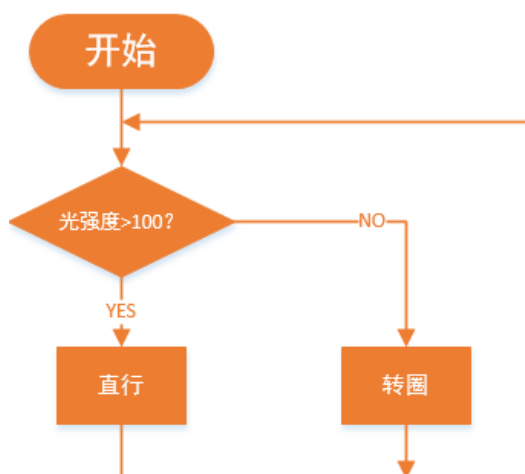
名称	模块	功能说明
----	----	------

<p>“如果否则”条件判断模块</p>		<p>当判断条件满足时，执行“那么执行”下的执行结果 1；如果不满足，执行“否则执行”下的执行结果 2。</p>
<p>逻辑判断“>”模块</p>		<p>判断条件 1 是否大于条件 2</p>

动手实践

(1) 打开 Mind+软件，新建项目。

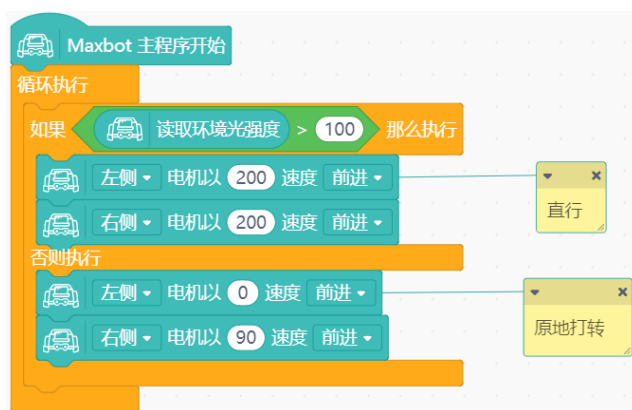
当光强度达到一定值（样例中为 100）的时候，飞蛾机器人就可以向光运动（前进）；当光强度小于一定值（100）的时候，飞蛾机器人原地旋转。根据上面所描述的功能可以画出对应的流程图来帮助我们进行编程哦！



(2) 需要用到条件判断模块和逻辑判断“>”模块。



完整程序如下图所示：



小贴士

如果 读取环境光强度 > 100 那么执行

判断条件：

需要设置一个合适的光强临界值，如果临界值太大，则需要很强的光飞蛾机器人才能向前运动；如果太小，则在环境光强下，飞蛾机器人就会直接向前一直运动。请大家根据所在的环境，选择合适的光强临界值。

(3) 保存项目，命名为“飞蛾机器人”，将程序下载到 Max:bot，来和飞蛾机器人一起来玩个游戏吧。

探究思考

一起玩飞蛾竞赛跑吧！和小伙伴比一比，用手电筒引导 Max:bot 前进，谁能在最短的时间从起点出发到达终点吧！不允许压线哦！

提示：调节速度，会让你的飞蛾机器人更加听话哦。如果转圈的速度很快，就很难控制 Max:bot 听你指挥哦！

第三章 聪明的 Max:bot

Max:bot 虽然有一双感光的“眼睛”但是这双“眼睛”不能让它躲开障碍物，Max:bot 一直会撞在桌子上或者衣橱上，这可怎么办呢？有没有什么办法帮助一下 Max:bot 呢？

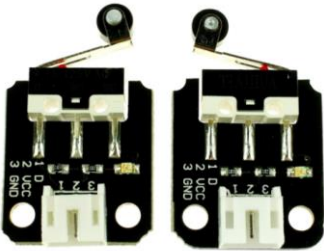


学习目标

1. 学习碰撞传感器的使用

- 2. 运用条件判断模块
- 3. 运用流程图辅助编程

电子模块


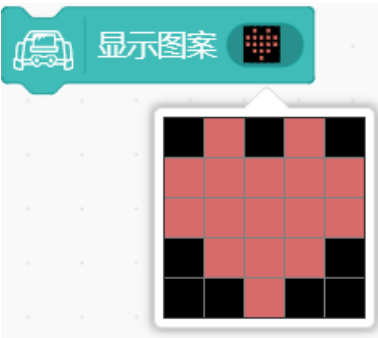
电子模块	图片	功能说明
碰撞传感器		碰撞传感器类似于按钮，可以检测到是否碰撞到障碍物。左侧碰撞传感器引脚为 P16、右侧碰撞传感器引脚为 P13。

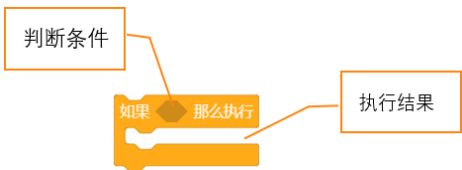

3.1 害羞的 Max:bot

Max:bot 是个很可爱的机器人，它也是可以感受到你的触碰的。



知识要点

名称	模块	功能说明
显示图案选择模块		可以选择想要的图案
显示图案制作模块		可以鼠标点击，制作显示图案

“如果” 条件判断模块		当判断条件满足时，则会执行“那么执行”下的执行结果；如果不满足直接进入下面的程序。
碰撞模块		判断左侧（右侧）碰撞传感器是否被触碰。

动手实践

- (1) 打开 Mind+软件，新建项目。
- (2) 将“如果”条件判断模块，拖入到循环模块中。



小贴士

判断条件只有两种状态：真或假。当判断条件成立（即为真）时候，则会执行“那么执行”下的执行结果；如果不满足，直接进入下面的程序。

- (3) 判断碰撞传感器的状态

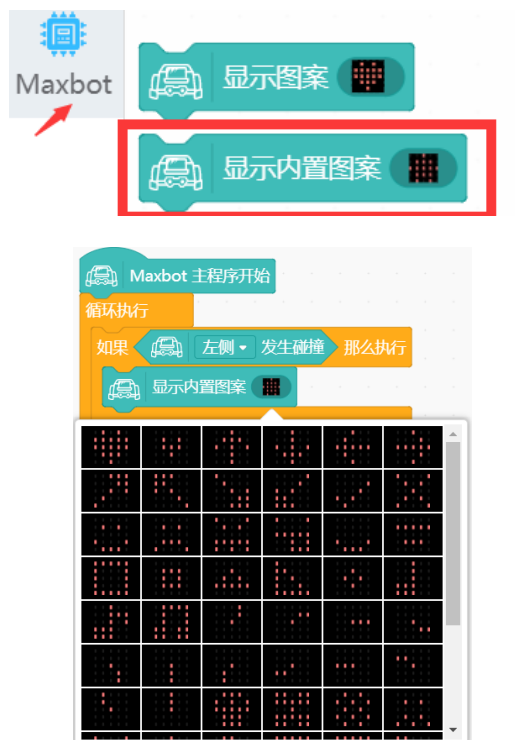
当 Max:bot 受到触碰的时候，它会紧张害羞，也就是需要判断碰撞传感器的状态，以左侧碰撞传感器为例。



(4) 制作心跳的显示效果

当 Max:bot 接收到触碰的时候，就会害羞紧张，心跳加速，运用图案显示模块，制作一个心跳的效果吧！

点击图案后会出现一个下拉菜单，选择两个大小不同的心形。

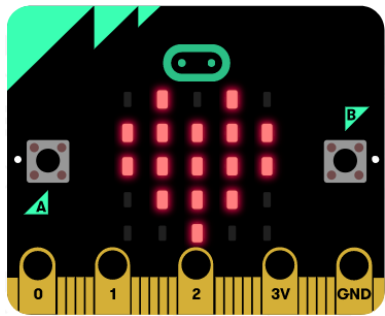


加入一个延时模块，就可以有心跳的效果啦。



(5) 保存项目，命名为“害羞的 Max:bot”，将程序下载到 Max:bot。

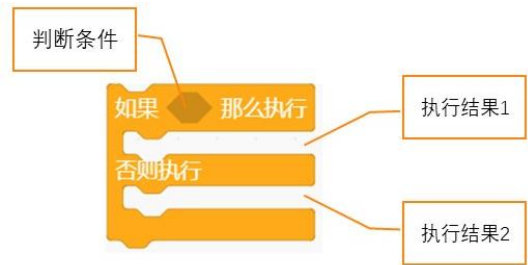
当你碰触到左侧碰撞传感器就会发现 Max:bot 心跳加速啦！Max:bot 是不是一个非常可爱的机器人呢？



探究思考

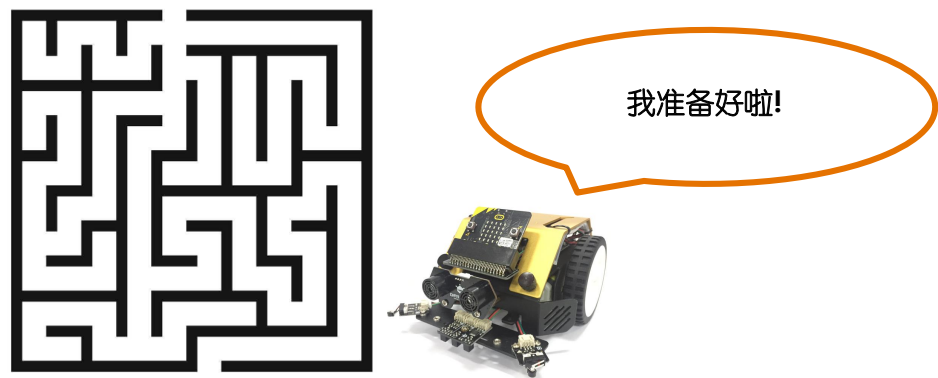
Max:bot 是个很友好的小伙伴，当你不触碰 Max:bot 的时候，它会对你露出笑脸。这个功能该怎么实现呢？

提示：需要用到“**如果否则**”条件判断模块



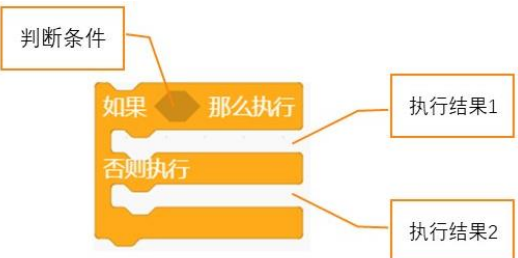
3.2 迷宫游戏

搭配了碰撞传感器的 Max:bot 机器人可以感知触碰了，如果把 Max:bot 放在迷宫中，它能顺利的走出来吗？



知识要点

名称	模块	功能说明
----	----	------

<p>“如果否则”条件判断模块</p>		<p>当判断条件满足时，则会执行“那么执行”下的执行结果1；如果不满足，执行“否则执行”下的执行结果2。</p>
---------------------	---	--

动手实践

(1) 打开 Mind+软件，新建项目。

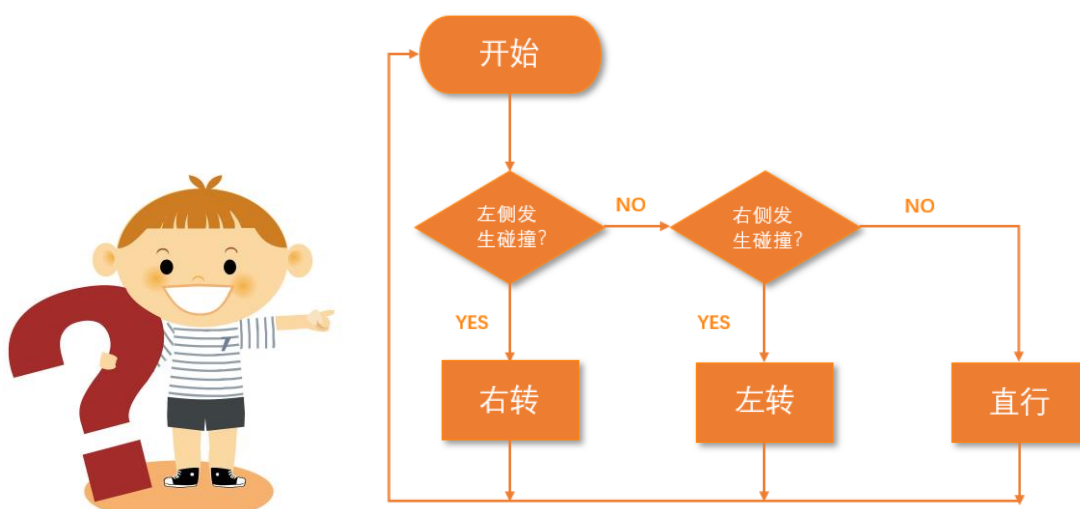
(2) Max:bot 走迷宫

Max:bot 要怎样才能迷宫中躲开障碍物成功走出去呢？让我们和 Max:bot 一起来思考：

如果 Max:bot 左侧遇到障碍物的话，需要向右侧转向，躲开障碍物；

如果 Max:bot 右侧遇到障碍物的话，需要向左侧转向，躲开障碍物；

程序编写流程图如下，根据流程图，同学们试一试编写对应的程序吧！



(3) 保存项目，命名为“害羞的 Max:bot”，将程序下载到 Max:bot。完整程序如下图所示。



探究思考

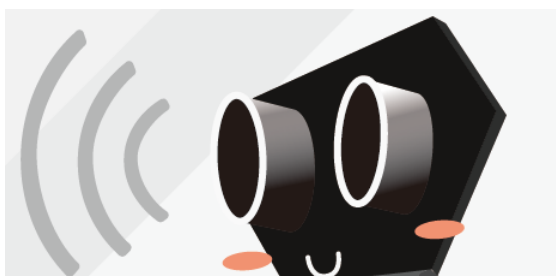
Max:bot 已经聪明地走出了迷宫，那么怎样来同步显示 Max:bot 的运动方向呢？

提示：需要用到显示图案制作模块或者显示图案选择模块



第四章 秘密武器超声波

Max:bot 可不只有感光的“眼睛”哦，你是不是在想小车前面这对“大眼睛”能够帮助 Max:bot 做些什么呢？



这可是 Max:bot 的秘密武器，一起来探索这个功能吧！

学习目标

1. 学习超声波的工作原理

- 2. 学习使用串口读值
- 3. 学习使用变量模块

电子模块

电子模块	图片	功能说明
超声波传感器		发射端发射出超声波，超声波遇到障碍物反射回接收端。超声波传感器发射端引脚为 P1、接收端引脚为 P2。

4.1 超声波测距

Max:bot 的秘密武器超声波能够帮助解决一些实际问题。当你还在拿尺子测量距离的时候，Max:bot 已经可以轻松帮你测量距离了。

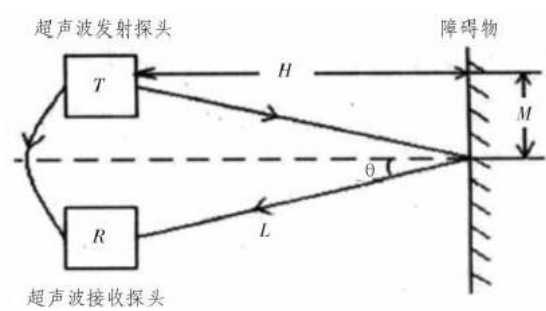
什么是超声波呢？

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。我们人类耳朵能听到的声波频率为 20~2000 赫兹。



当声波的振动频率大于 20000 赫兹或小于 20 赫兹时，我们便听不见了。因此，我们把频率高于 20000 赫兹的声波称为“超声波”。



超声波怎么测距？



超声波发射器向某一方向发射超声波，超声波在空气中传播，途中碰到障碍物就立即返回来，超声波接收器收到反射波。通过转换计算，就得到了距离 H。

同学们，想不想试试使用超声波传感器轻松测量距离？

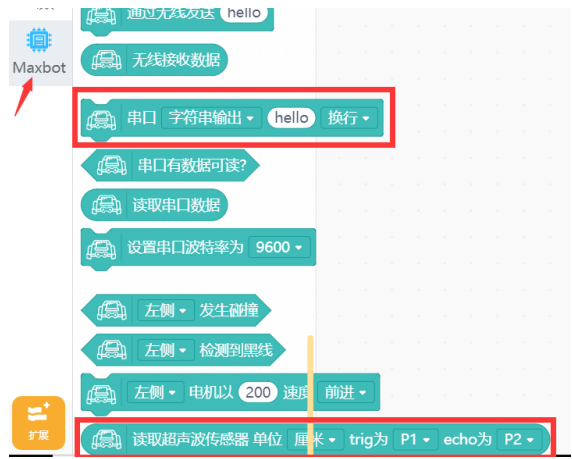
知识要点

名称	模块	功能说明
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(P1)和接收端的引脚(P2)以及距离单位。
串口输出模块		在串口区输出读取的值

动手实践

- (1) 打开 Mind+软件，新建项目。
- (2) 读取超声波传感器测量的距离值。

在 2.1 探秘光强的大小一节中我们使用显示模块在 Micro:bit 的点阵屏上读取到了光强值，其实通过串口也可以进行读值。以超声波传感器为例，选择读取超声波传感器模块和串口输出模块组合后放入循环模块中。



加入延时模块防止串口区读值过快。

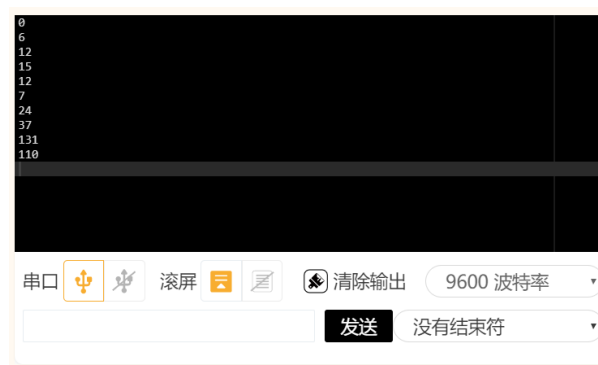


小贴士

“trig”对应的是超声波发射端引脚，“echo”对应的是超声波接收端引脚。这两个引脚也与超声波传感器上的标志统一，在 Max:bot 上，“trig”对应的是“P1”，“echo”对应的是“P2”，这里显示的单位（unit）为距离单位“cm”。注意使用 Max:bot 上的超声波传感器测距时要打开开关，前方距障碍物 5cm~300cm 范围内的测距结果比较准确。

(3) 保存项目，命名为“超声波测距”，将程序下载到 Max:bot。

(4) 打开串口开关查看超声波传感器测量的距离值。



探究思考

利用超声波传感器怎么测量身高呢？



想获得精确身高，你可能需要注意：



- 超声波的固定位置和方向
- 在 10cm 范围进行校准，防止出现大的偏差

4.2 行车安全仪

在川流不息的马路上，交通事故的发生总是那么的不可预测，如果可以帮助汽车在行驶过程中判断危险的存在，就会有效地避免事故，保证人身安全。



知识要点

名称	模块	功能说明
变量模块		建立一个变化的量，可以实现动态的效果。
变量赋值模块		给变量“my variable”赋值，默认模块意义为：my variable = 0。
逻辑“与”模块		两个条件都同时成立的情况下“逻辑与”的运算结果才为“真”。

动手实践

- (1) 打开 Mind+ 软件，新建项目。
- (2) 使用变量

感知系统可以实时地检测外界环境，如环境光传感器会实时监测到变化的光的强度值。在程序编写过程中我们可能要多次用到这一值，因此，通常把这类经常需要存储和调用的值赋给一个变量。“my variable”是变量名，为了方便区分不同的变量，我们可以用不同的名字为变量命名。

鼠标右击选中变量模块，选择“重命名变量”，将变量名改为“distance”。



在行车安全仪项目中，我们需要实时监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”。

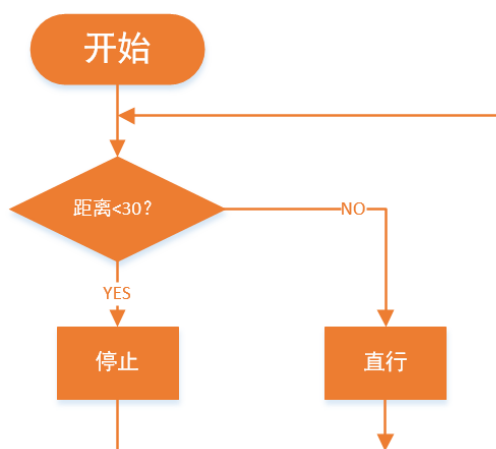


当需要调用超声波距离值时，直接使用变量“distance”就可以了。

变量 distance

- (3) 当小车前方的距离障碍物（前车）比较近时，小车会自动停下来，防止追尾；如果前方有足够的距离，那么就可以放心的向前行驶啦！

要怎么实现功能呢？编辑程序之前，先来写个流程图，梳理下思路吧。

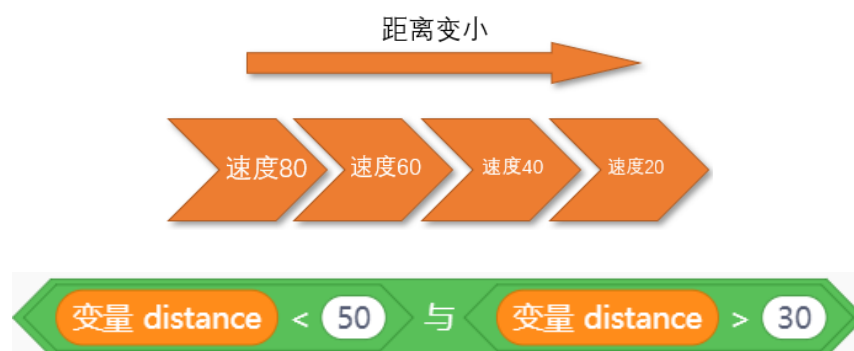


- (4) 保存项目，命名为“行车安全仪”，将程序下载到 Max:bot，小车就可以安全的向前行驶啦！



探究思考

在实际生活中，正在行驶的车辆是不会一下子就停下的，而是会慢慢地降低速度直到停下。也就是说随着距离变小，速度也慢慢降低。你能实现吗？



第五章 勇往直前的 Max:bot

Max:bot 机器人是个勇往直前的勇者，除了能够及时的判断前面是否有障碍物，它还有个神奇的技能：其实它有个朝向地面的“眼睛”，可以帮助它避免从高处跌落，实现沿着固定轨迹行走。快了解一下吧。

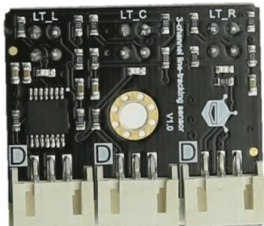


学习目标

1. 学习巡线传感器的工作原理
2. 学习使用函数模块

3. 逻辑“与”和逻辑“非”的理解与运用

电子模块

电子模块	图片	功能说明
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，也可以检测黑背景的白线。共三个检测头，但只用到了左右两个，左侧引脚为 P15、右侧为 P14。

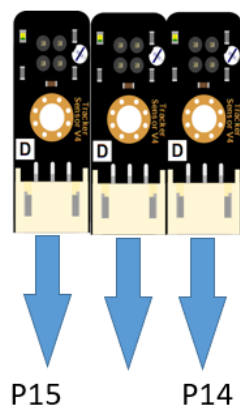
5.1 悬崖勒马

看！前面有个悬崖，可是 Max:bot 并没有掉下去，它可以自动后退，成功的避免“机”毁“车”亡，这多亏了巡线传感器！



巡线传感器是怎么工作的呢？

巡线传感器上面有三个检测头，但是只有左右两个被启用了，分别连接在接口 P15、P14。



每个检测头是成对出现的红外探头。

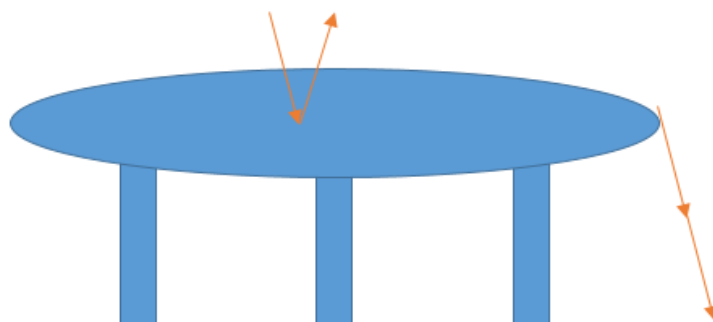


其中一个红外发射传感器发射出红外线，另外一个接收红外线。巡线传感器工作原理与超声波传感器类似，但其得到的结果并不会转化成距离，只能判断有没有障碍物。

巡线传感器工作原理

当安装了巡线传感器的 Max:bot 在桌面上时，红外线发射端发射出的红外线被桌面反射回来，红外线接收端接收到信号，指示灯亮，返回高电平 “1” ；

当安装了巡线传感器的 Max:bot 在桌面边缘时，红外线发射端发射出的红外线不能被桌面反射回来，红外线接收端接收不到信号，指示灯灭，返回低电平 “0” 。



小贴士

注意，这里的桌面颜色为浅色，如果为深色，则能够吸收所有颜色的光（包含红外线）而导致基本上不反射光线，所以深色桌面不能实现以上功能。





知识要点

名称	模块	功能说明
逻辑 “与”		两个条件都同时成立的情况下 “逻辑与” 的运算结果才为 “真”。
逻辑 “非”		本来值的反。即本来是 “真” 经过逻辑非后变为 “假”。

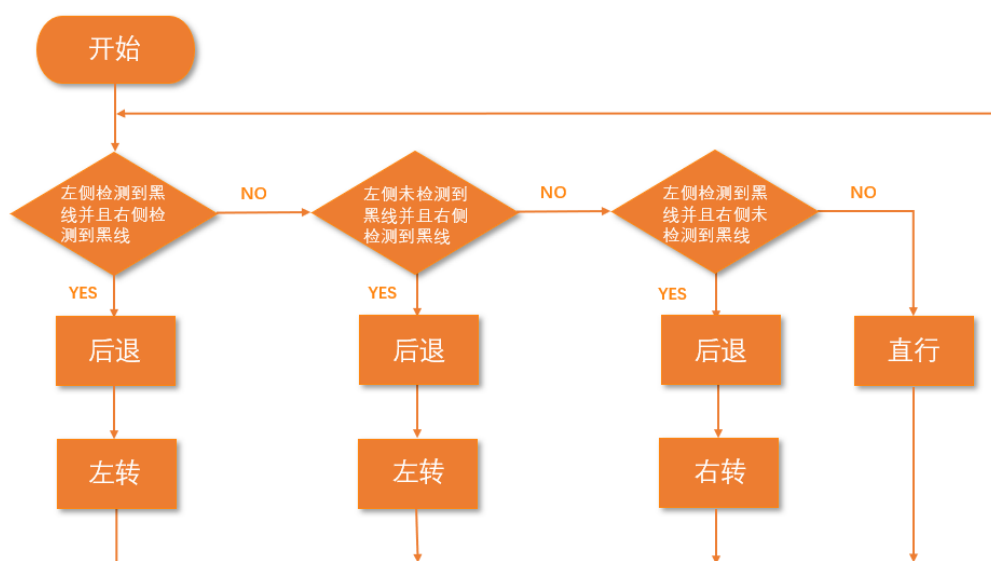
定义子函数		当一个连续的动作，需要在程序中出现多次时，为了使得程序更加清楚，需要定义子函数。
调用子函数		在程序中，需要使用子函数这一系列动作时，拖入到程序中。需要与“定义子函数”对应使用。

动手实践

- (1) 打开 Mind+软件，新建项目。
- (2) 在悬崖勒马项目中，需要用到两个巡线传感器，左侧传感器连接在 P15，右侧传感器连接在 P14。Max:bot 在桌面上的运动状态有如下几种：

状态示意图	状态描述	传感器状态
	Max:bot在桌面上	左侧传感器未检测到黑线 右侧传感器未检测到黑线
	Max:bot左右两个巡线传感器在桌子边缘	左侧传感器检测到黑线 右侧传感器检测到黑线
	Max:bot右侧传感器在桌子边缘	左侧传感器未检测到黑线 右侧传感器检测到黑线
	Max:bot左侧传感器在桌子边缘	左侧传感器检测到黑线 右侧未传感器检测到黑线

- (3) 根据上表所分析的 Max:bot 可能出现的状态，需要为编程制作流程图。



- (4) 定义及调用子函数

通过流程图可以看出，在悬崖勒马项目中需要多次使用“后退”和“左转”，为简化主程序，添加“后退”和“左转”两个自定义模块。



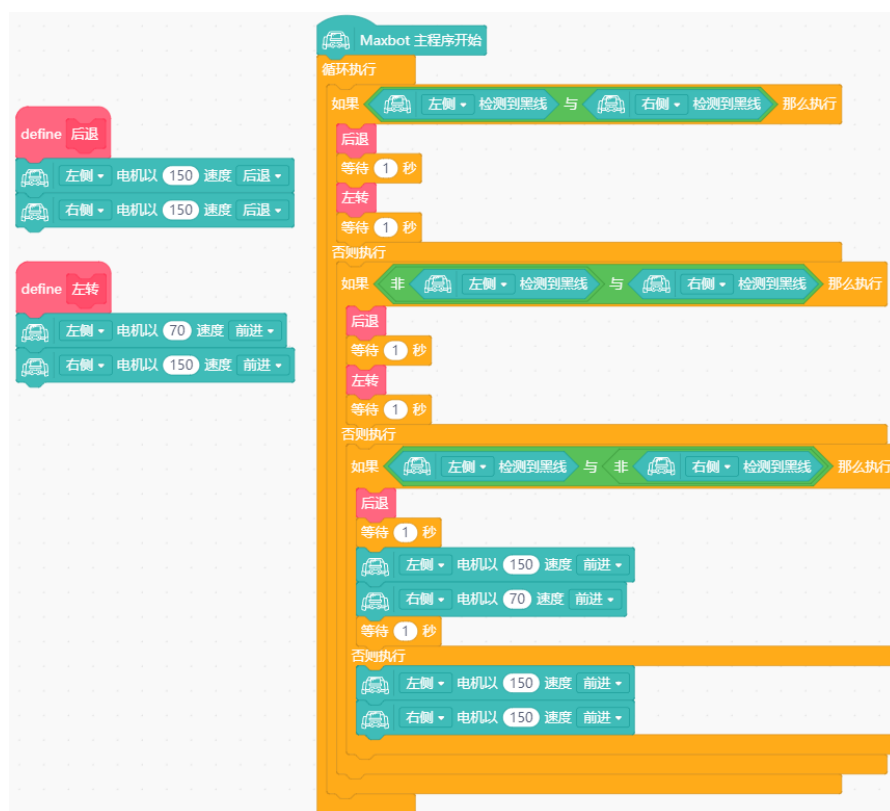
定义“后退”和“左转”两个函数。调用函数时只需将定义后的子函数模块拖入主程序中适当位置即可。



- (5) 根据流程图完善程序，保存项目，命名为“悬崖勒马”，将程序下载到 Max:bot 小车，小车就可以安全的在桌面上行驶啦！

小贴士

注意：为了防止小车失误掉落桌面，因此，小车速度不宜设置的太快。



探究思考

在生活中的哪些场景中，需要防跌落的功能呢？

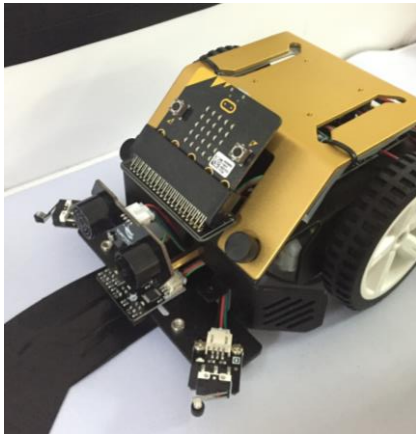
随着生活中科技化工具的丰富，扫地机器人逐渐成为居家必备，它可以自主充电，按时对地面进行清洁，但是为了防止在扫地过程中从楼梯上跌落，所以它需要具备防跌落功能。防跌落功能的实现需要依靠扫地机器人底部的多个探测器，通常在扫地机器人底部边缘安装一圈的探测器。



还有哪些场景需要这一功能呢？小朋友们，你想到了吗？


5.2 循“轨”蹈矩

Max:bot 最喜欢按照地图去寻宝啦，不管多么复杂地形，都能够沿着轨迹去探索未知世界。



带着 Max:bot 一起去玩寻宝游戏吧。

知识要点

名称	模块	功能说明
逻辑“与”		两个条件都同时成立的情况下“逻辑与”的运算结果才为“真”。
逻辑“非”		本来值的反。即本来是“真”经过逻辑非后变为“假”。

动手实践

- (1) 打开 Mind+软件，新建项目。
- (2) 在本项目中，需要让 Max:bot 沿着黑色的轨迹行走，可以绘制黑色轨迹或使用黑色胶带。

小贴士

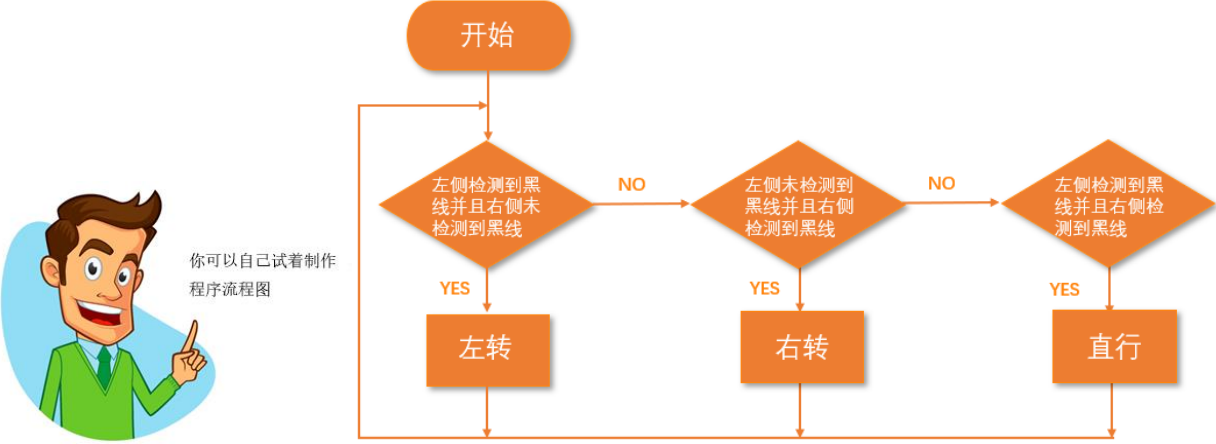
Max:bot 的左右两个巡线传感器（P14、P15）需要同时保持在黑色轨迹上，我们图纸上的黑线的宽度可以保证左右两个探测器同时放在黑线上。



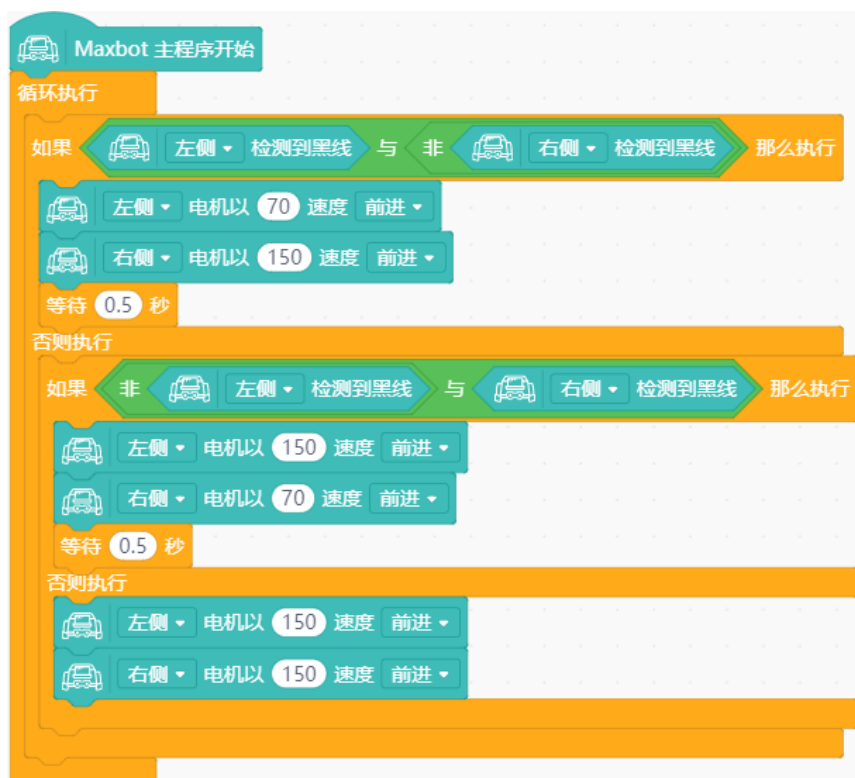
- (3) Max:bot 沿着直线轨迹行进的过程中会遇到以下情况：

状态示意图	状态描述	传感器状态
	Max:bot左右两个传感器在黑线上	左侧传感器检测到黑线 右侧传感器检测到黑线
	Max:bot右侧传感器在黑线外 左侧传感器在黑线上	左侧传感器检测到黑线 右侧传感器未检测到黑线
	Max:bot左侧传感器在黑线外 右侧传感器在黑线上	左侧传感器未检测到黑线 右侧传感器检测到黑线

(4) 根据上述分析 Max:bot 可能遇到的情况制作流程图。



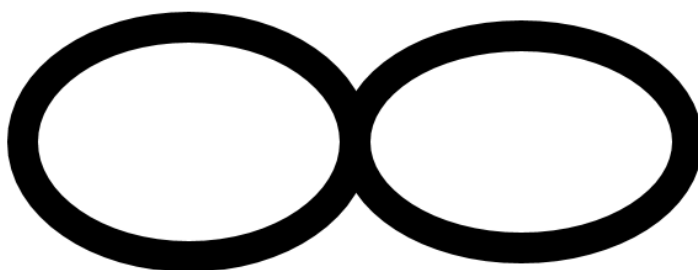
(5) 根据流程图完善程序，保存项目，命名为“循轨蹈矩”，将程序下载到 Max:bot 小车，小车就可以沿着轨迹行驶啦！



探究思考

Max:bot 可以沿着直线探索世界啦，给 Max:bot 出个难题吧。

下面的路线 Max:bot 可以巡线吗？你遇到了哪些问题呢？记录下来和其他小伙伴交流一下吧。



第六章 无线通信

如果拥有了两个 Max:bot 会发生什么样的效果呢？

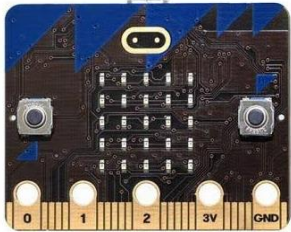


Max:bot 的大脑 Micro:bit 具有无线通信功能，可以实现更多的远程控制啦！

学习目标

- 1. 学习无线通信的基本用法
- 2. 学习加速度传感器的使用

电子模块

电子模块	图片	功能说明
Micro:bit		Micro:bit 上带有无线模块和加速度计。

6.1 Say Hi

当 Max:bot 遇到了另外一个 Max:bot 小伙伴，先来打个招呼吧！




当一个 Max:bot 通过 Micro:bit 发出一个信号时，另外一个 Max:bot 能够接收到信号，并做出反应，这里用到的就是 Micro:bit 的无线通信功能。

什么是无线通信功能？

在无线通信功能中，需要两个 Micro:bit，发出信号的叫做“发射端”，接收信号的叫做“接收端”。

知识要点

名称	模块	功能说明
打开（关闭）无线通信模块	 打开 ▾ 无线通信	打开或者关闭无线通信功能。
设置无线频道模块	 设置无线频道为 7	给数据传递的频道进行定义。
通过无线发送数据模块	 通过无线发送 hello	通过无线发射“hello”。
无线接收数据模块	 无线接收数据	识别接收到的数据，做出对应的动作。

当无线接收到数据模块		无线接收到数据时运行。
------------	---	-------------

动手实践

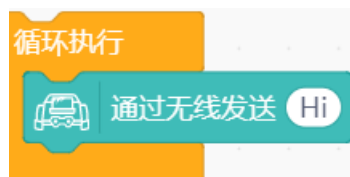
1、发射端

- (1) 打开 Mind+软件，新建项目。
- (2) 打开无线通信功能，设置无线频道为 1。



- (3) 设置需要发射的内容

先来打个招呼吧！



- (4) 保存项目，命名为“打招呼发射端”，将程序下载到发射端。

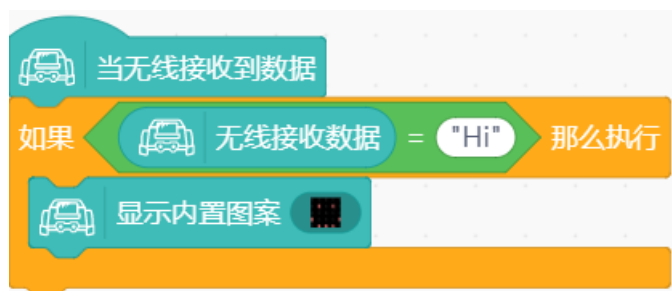


2、接收端

- (1) 打开 Mind+软件，新建项目。
- (2) 打开无线通信功能并设置与发射端相同的频道。



(3) 判断是否是接收到正确的数据，当收到“Hi”，那么就会露出笑脸。



小贴士

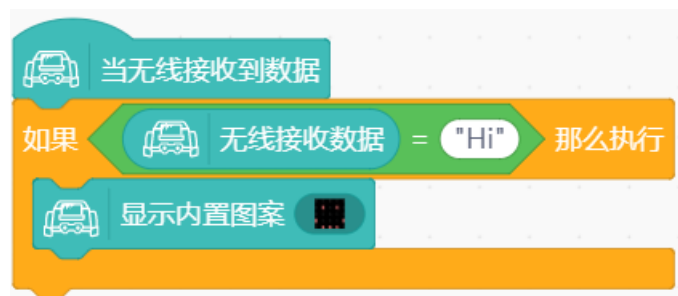


中的“Hi”在写入时不需要加入双引号“”，而



中的“Hi”在写入时必须加上双引号“”，这是因为英文单词是字符串形式，在运算模块需要使用英文输入法双引号“”进行引用。

(4) 保存项目，命名为“打招呼接收端”，将程序下载到接收端。



将接收端和发射端的开关都打开，接收端“露出”笑脸了吗？

探究思考

在一些神秘的场合中，需要发射一些数字暗号，来帮助双方来识别对方的身份，也就是在谍战剧中需要一些代码暗号。试着和小伙伴来一场谍战游戏吧。

请发送一串暗号，例如：1231；让你的小伙伴来说出（显示）对应的暗号，1321。



6.2 体感赛车

你有玩过体感赛车游戏吗？通过不停的变化遥控器的方向，来控制屏幕上的赛车急速奔跑和转弯，好刺激呀！



那么 Micro:bit 可以实现体感控制吗？Micro:bit 上面集成了加速度传感器。



什么是加速度传感器？

加速度传感器可以感受到物体的速度变化情况。Micro:bit 上具有加速度传感器，能判断 Micro:bit 的运动方向、角度、手势等场景。

手机上也有加速度传感器，可以玩一些体感互动游戏；手环上的加速度传感器，可以判读运动情况，进行计步。



知识要点

名称	模块	功能说明
倾斜振动判断	 当  徽标朝上 ▾	判断 Micro:bit 是否在进行向前、向后、向左、向右等方向的振动或倾斜。

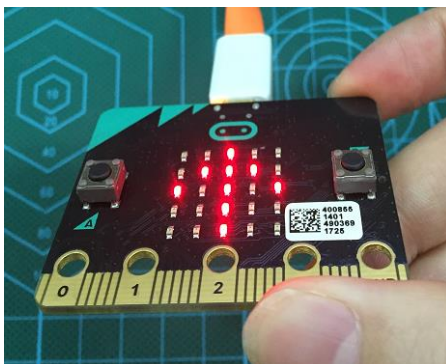
动手实践

1、发射端

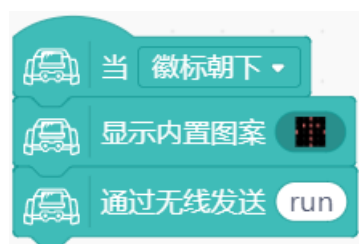
- (1) 打开 Mind+软件，新建项目。
- (2) 打开无线通信功能，设置无线频道为 1。



- (3) 当你转动 Micro:bit，使得徽标朝下时，对应控制 Max:bot 前进。

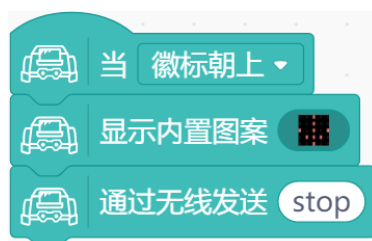


当检测到 Micro:bit 的徽标朝下时，用“向上箭头”显示出来当前的状态，发出控制另外一个 Max:bot 向前的指令。



- (4) 分别编辑其他三个控制状态。

停止：



向左转：



向右转：



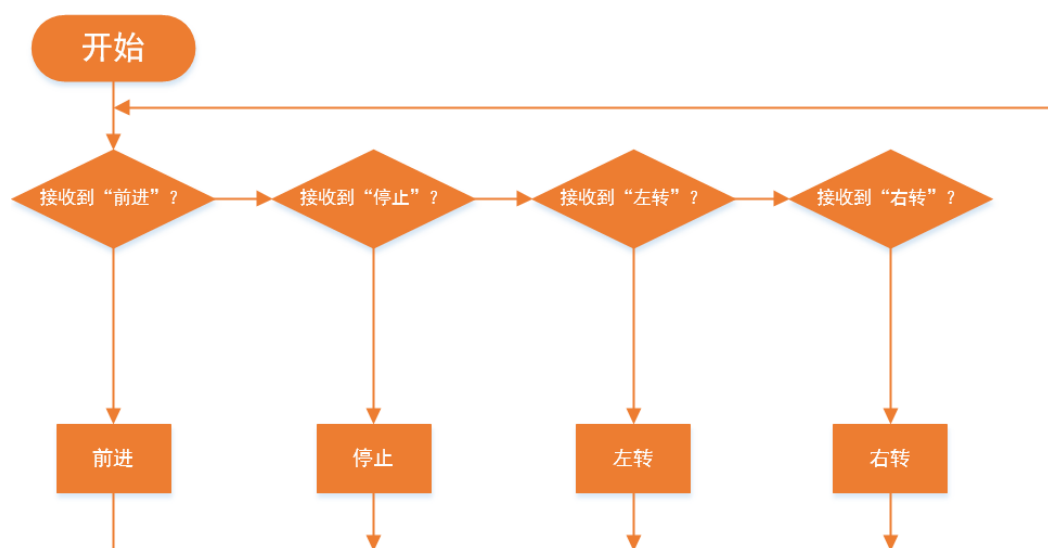
- (5) 保存项目，命名为“体感赛车发送端”，将程序下载到发送端。



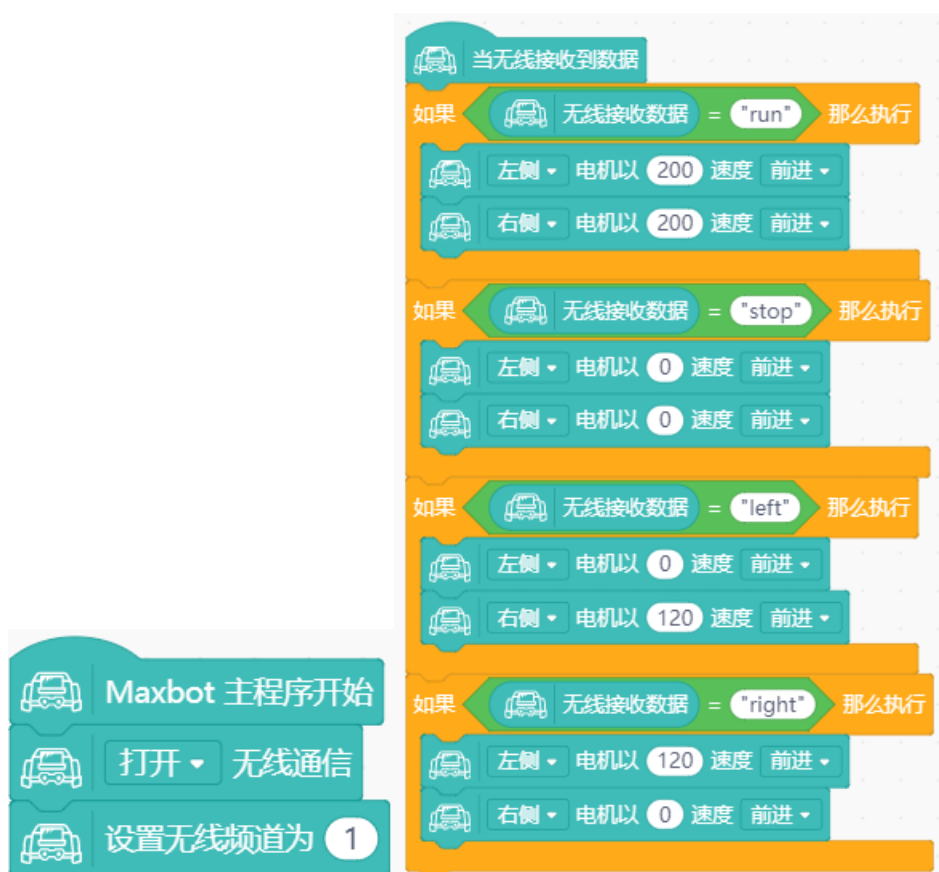
2、接收端

- (1) 打开 Mind+软件，新建项目。
- (2) 需要根据接收的信号指令，控制 Max:bot 动作。

请根据以下流程图，进行编程：



(3) 保存项目，命名为“体感赛车接收端”，将程序下载到接收端。来玩个赛车游戏吧！

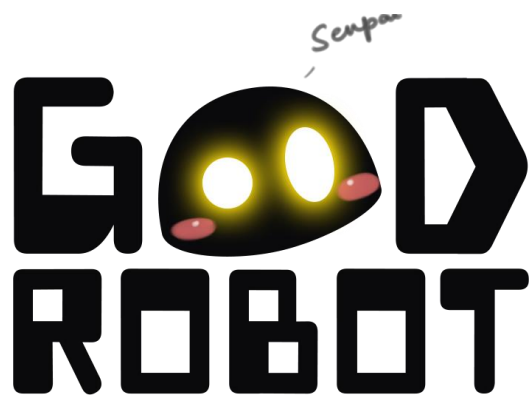


探究思考

在体感游戏中，如果你控制端的摆动的角度越大，赛车会转向的角度越大，我们的体感赛车 Max:bot 该怎样实现呢？

第七章 声光互动

Max:bot 完成了好多任务了，在迷宫中完成探索，可以帮你赢得了体感赛车游戏。


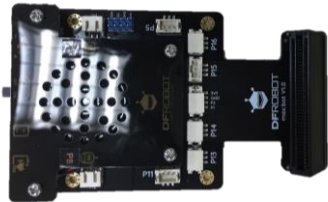


那么 Max:bot 的外观上可以有哪些变化呢？它可以发出声音吗？

学习目标

- 1. 学习灯带模块的使用
- 2. 学习喇叭模块的使用
- 3. 可自由编辑音乐
- 4. 制作流水灯效果

电子模块

电子模块	图片	功能说明
灯带模块		灯带模块，可以设置灯珠的颜色，亮度以及动画效果
喇叭模块		喇叭模块，作为输出模块，可以播放出音乐

7.1 炫彩灯带

在节日的时候，都会有好多漂亮的彩灯来装扮。



运用灯带，将 Max:bot 装扮起来吧！

知识要点

名称	模块	功能说明
灯带控制模块		设定可以控制灯带中灯珠的总个数和亮度值
灯带点亮顺序和色调范围控制		设定该模式下可以控制的灯珠和灯珠颜色范围

动手实践

- (1) 打开 Mind+软件，新建项目。
- (2) 添加灯带编程需要的模块

单击“扩展”，在“显示器”中选择“Ws2812 RGB 灯”。

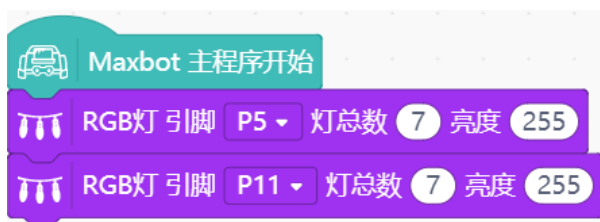


指令区出现显示器模块。



(3) 设置彩虹灯亮度

灯带连接引脚为 P5（左侧灯带）和 P11（右侧灯带），各包含 7 个 LED 小灯，设置彩虹灯亮度为 255。



(4) 设置彩虹灯效果

设置彩虹灯效的色彩范围 1-360（红色--绿色--蓝色-红色），彩灯呈现七彩效果。“0 到 6”表示灯带上的 7 个灯珠。



(5) 保存项目，命名为“炫彩灯带”，将程序下载到 Max:bot。

OK，灯带被点亮了，能够实现七彩灯带的效果啦！

探究思考

可不可以让灯带根据我们的喜好显示出不同的颜色呢？大家动手试一试吧！

提示：

RGB灯 引脚

P2 ▾

灯号

0 到

4

显示颜色

7.2 演奏美妙音乐

悦耳的音乐能够让人心情愉悦，如果在小游戏中加点音效进去，比如在胜利的时候有庆祝的音乐，Max:bot 会变得很有趣。

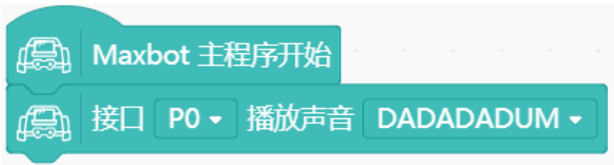


知识要点

名称	模块	功能说明
播放旋律	<div><div>喇叭连接引脚值</div><div>选择库中的旋律</div><div><div>接口</div><div>P0 ▾</div><div>播放声音</div><div>DADADADUM ▾</div></div></div>	调用 Mind+的自带的音乐
音调模块	<div><div>喇叭连接引脚值</div><div>音符音调</div><div><div>接口</div><div>P0 ▾</div><div>播放音符</div><div>C2 ▾</div><div>1 ▾</div><div>拍</div></div><div>节拍</div></div>	按照歌谱制作歌曲

动手实践

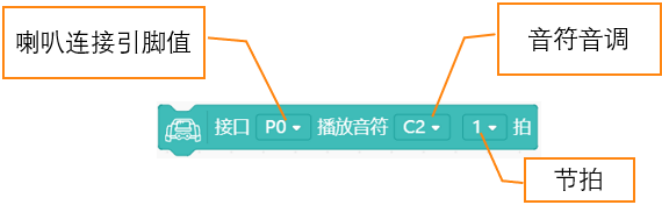
- (1) 打开 Mind+软件，新建项目。
- (2) **播放旋律模块**中已经有内置的很多旋律可供选择，我们在这里选择的是第一段旋律“dadadum”。



- (3) 保存项目，命名为“演奏美妙音乐”，将程序下载到 Max:bot。
- 能够听到播放旋律“dadadum”。

探究思考

你想不想自己制作美妙的旋律呢？Max:bot 也能帮你实现哦。



“C” 是音符 do(即音符 1)，字母后面的数字是音符的音高，即 “2” 表示低音，那么 “C2” 就表示低音 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

利用音调模块，制作一首简单的歌曲吧！

小星星

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，

5 5 4 4 3 3 2 — 5 5 4 4 3 3 2 —
挂 在 天 上 放 光 明， 它 是 我 们 的 小 眼 睛。

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。

7.3 圣诞机器人

每个圣诞来临的时候，能够看到很多漂亮的圣诞树。



下面我们和 Max:bot 一起来制作一个圣诞机器人吧，它可以一边闪耀着灯光，一边播放着圣诞歌曲哦！

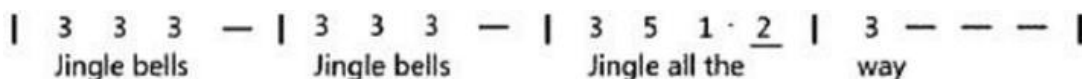
知识要点

名称	模块	功能说明
变量变化步长 模块		设定每次该变量变化的大小： 变量 → 变量+1 → (变量+1)+1

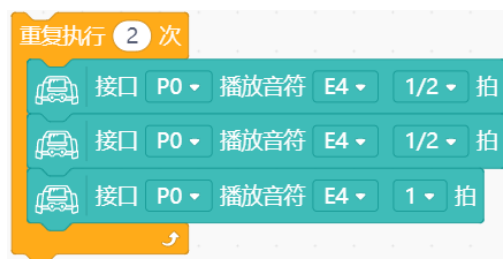
灯带效果清除 模块		关掉灯带的效果
--------------	---	---------

动手实践

- (1) 打开 Mind+软件，新建项目。
- (2) 编辑一段圣诞的音乐“Jingle Bell”，运用音调模块。



看到简谱上，前面两个短句是相同的音调和节拍。那么可以使用重复执行模块。



完整的音乐程序部分如下。



- (3) 编辑灯带的效果

下面来学习怎么把两条灯带做成“流水灯”的效果。

首先，“流水灯”的效果是每次增加一颗灯亮起。



所以两侧灯带灯亮起的个数，是个变化的量，需要用到变量来控制。

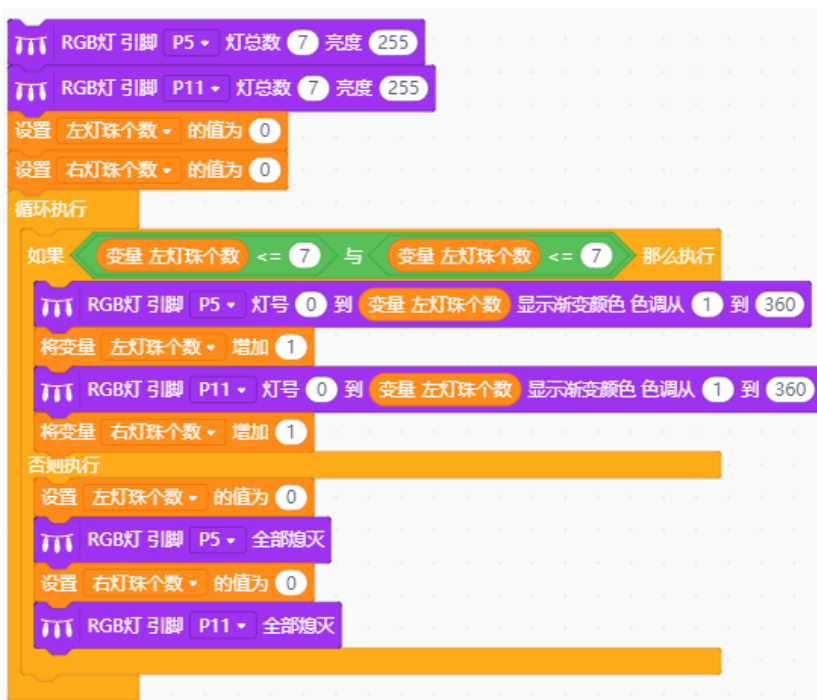
将已有变量名称更改为“左灯珠个数”，再新建一个变量“右灯珠个数”。



左侧灯带（P5）的灯珠的个数的变化，就需要控制变量“左灯珠个数”来控制了；右侧灯带（P11）的灯珠的个数的变化，就需要控制变量“右灯珠个数”来控制了。

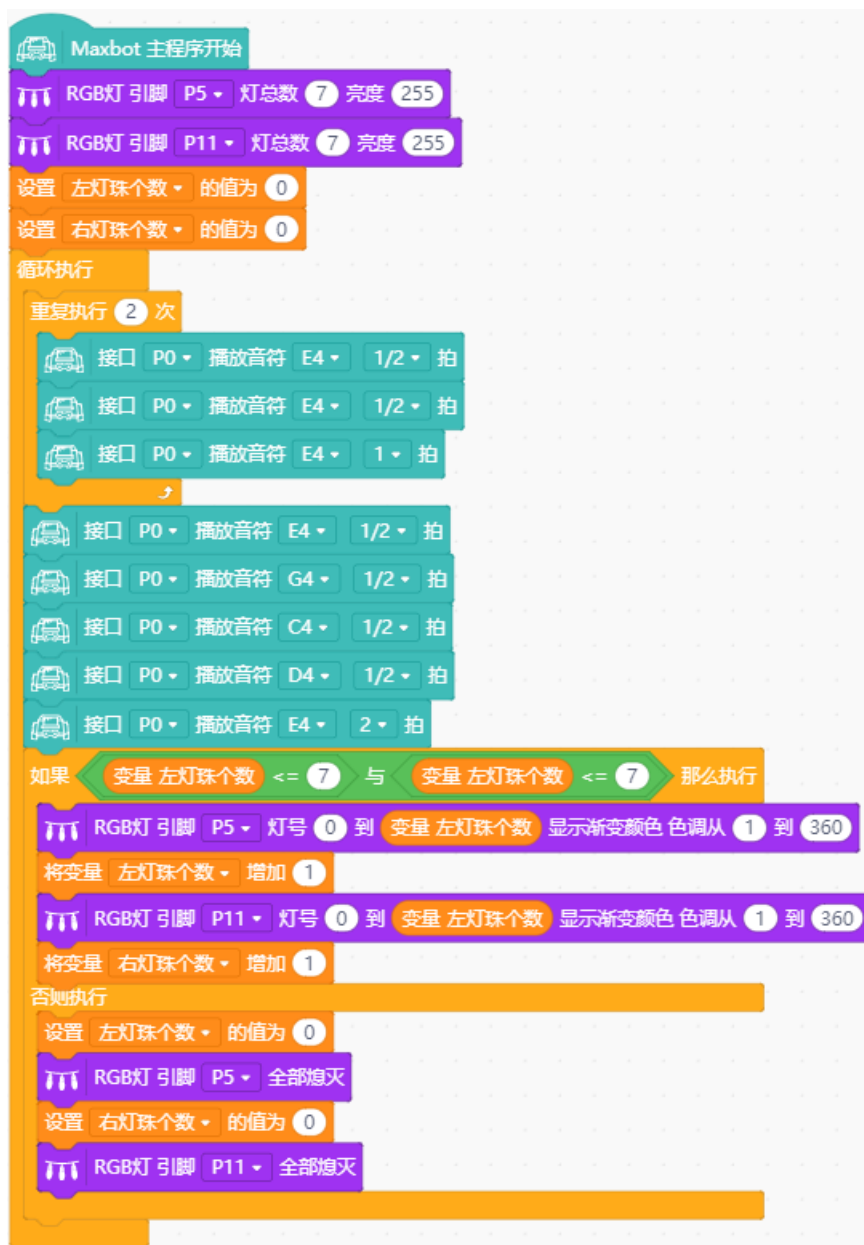
因为我们的灯带只有 7 颗灯珠，需要限制变量“左灯珠个数”和“右灯珠个数”分别同时小于 7，并且初始值为 0。

每次灯珠改变颜色亮起后，对变量“左灯珠个数”和“右灯珠个数”增加 1，使下一个灯珠亮起变色。



(4) 保存项目，命名为“圣诞机器人”，将程序下载到 Max:bot。

一个会唱歌会闪烁的圣诞机器人就完成啦！



探究思考

试着和体感赛车结合，让你的圣诞机器人动起来，帮你去派发礼物吧。

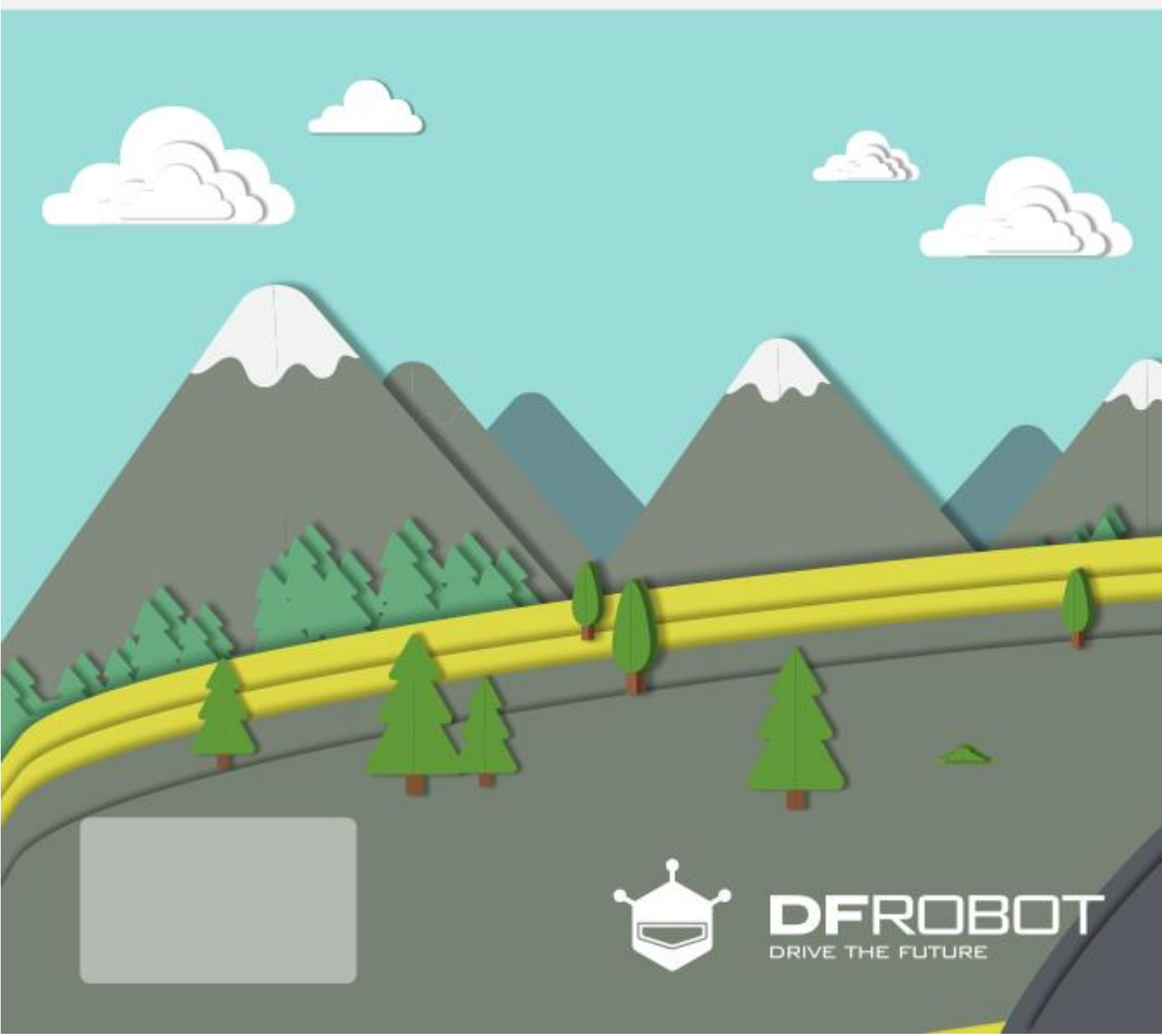
希望你的机器人探索兴趣一直延续，用你的奇思妙想，玩出更多新颖有创意的作品。如果你愿意与我们分享的话，也可以直接登陆我们的论坛，让我们的社区论坛记录下你的点点滴滴！

欢迎登陆 DFRobot 创客社区！

DFRobot 创客社区: www.dfrobot.com.cn



扫描二维码
关注DF创客社区
带你轻松玩转创客项目



DFROBOT
DRIVE THE FUTURE