
MAX:BOT

机器人探索之旅



DFROBOT
DRIVE THE FUTURE

目录

目录.....	I
初识 Max:bot 机器人	1
什么是 Max:bot 机器人 ?	1
什么是机器人 ?	1
Max:bot 上的装备	1
初识 Micro:bit	3
什么是 Micro:bit ?	3
Micro:bit 的编程方式	3
Makecode 图形化编程	3
MicroPython 代码编程	4
怎么对 Micro:bit 进行 Makecode 图形化编程 ?	4
进入 Makecode 编程界面	4
界面认知	5
程序下载	5
第一章 小车快跑	8
学习目标	8
电子模块	8
1.1 小车动起来	8
知识要点	8
动手实践	9
探究思考	10
1.2 小车转圈圈	10
知识要点	10
动手实践	10
探究思考	12
1.3 小车走方形	12
知识要点	12
动手实践	13
探究思考	15
第二章 聪明的 Max:bot	15
学习目标	15

电子模块	16
2.1 害羞的 Max:bot	16
知识要点	16
动手实践	17
探究思考	20
2.2 迷宫游戏	20
知识要点	20
动手实践	21
探究思考	22
第三章 感光的眼睛	23
学习目标	23
电子模块	23
3.1 探秘光强的大小	23
知识要点	23
动手实践	24
探究思考	24
3.2 飞蛾机器人	25
知识要点	25
动手实践	25
探究思考	26
第四章 秘密武器超声波	27
学习目标	27
电子模块	27
4.1 超声波测距	28
知识要点	29
动手实践	29
探究思考	31
4.2 行车安全仪	31
知识要点	32
动手实践	32
探究思考	33
第五章 勇往直前的 Max:bot	35
学习目标	35

电子模块	35
5.1 悬崖勒马	35
知识要点	37
动手实践	37
探究思考	38
5.2 循“轨”蹈矩	39
知识要点	39
动手实践	39
探究思考	42
第六章 无线通信	43
学习目标	43
电子模块	43
6.1 Say Hi	43
知识要点	45
动手实践	45
探究思考	48
6.2 体感赛车	48
知识要点	50
动手实践	50
探究思考	54
第七章 声光互动	55
学习目标	55
电子模块	55
7.1 炫彩灯带	55
知识要点	56
动手实践	56
探究思考	57
7.2 演奏美妙音乐	58
知识要点	58
动手实践	58
探究思考	59
7.3 声光互动	60
知识要点	60

动手实践	61
探究思考	65

*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

什么是 Max:bot 机器人？

Max:bot 机器人是一款低门槛，功能丰富的移动平台，设计感的接口，可以完美的与 Micro: bit 搭配使用。集成了机器人的基础功能，具有易组装、一体成型的全金属闪亮外壳。

- Micro:bit 的完美搭档
- 上手轻松便捷，快速安装
- 采用太空铝材料，结实耐用
- 支持拓展进阶，可拓展 100+Gravity 系列电子模块

什么是机器人？

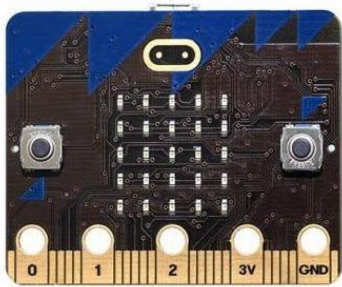
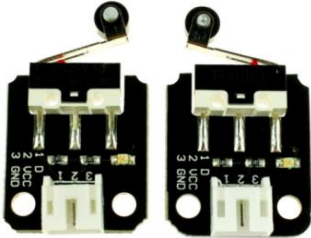
机器人是具备一定运算能力的自动化机器。类似于生物机体，机器人一般由以下几大“器官”构成：

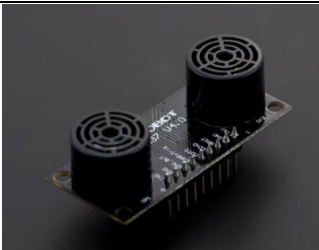
大脑：Micro:bit 作为整个机器人的大脑，具有逻辑能力，用于处理接收到的信息与发送控制指令。

感知系统：Max:bot 上扩展了有很多传感器，担任了机器人的感知系统，如眼睛（智能灰度传感器）与手（碰撞传感器），用于接收外部信息

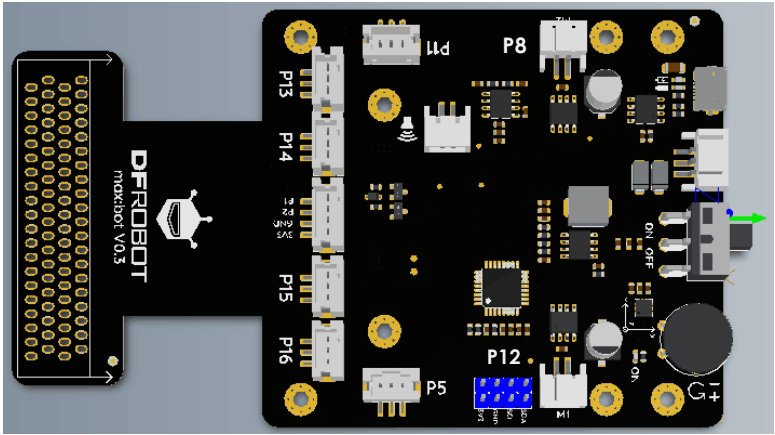
执行系统：如肢体动作，语言，表情等用于改造外部环境或向外部发送信息。Max:bot 上面搭载了灯带和喇叭，并且 Micro:bit 上有 25 颗红色 LED，用以显示信息和图案。

Max:bot 上的装备

元件名称	图片	引脚
Micro:bit		
碰撞开关		P16 (左)、P13 (右)

超声波传感器		P1、P2
巡线传感器		P15 (左)、P14 (右)
灯带		P5 (左)、P11 (右)
电机		P12 (左)、P8 (右)

以上这些设备，全部通过以下板子，装备在 max:bot 机器人上。

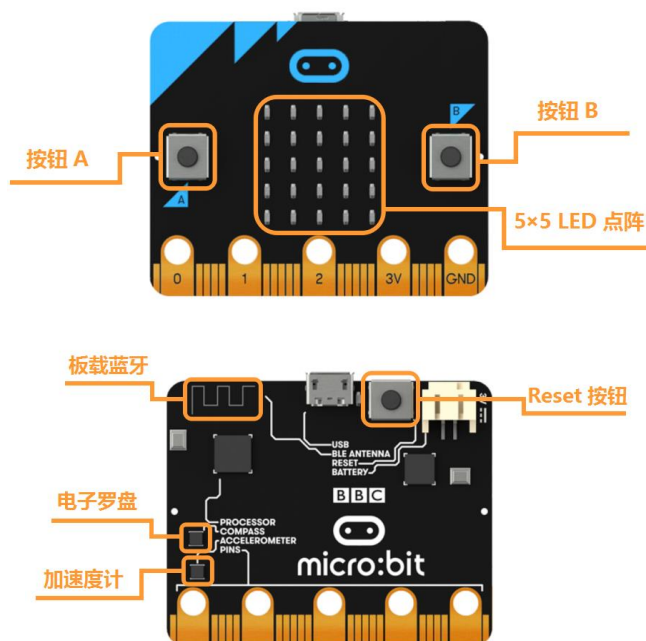


P12

初识 Micro:bit

什么是 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。



得益于其丰富的 I/O 扩展口以及硬件支持，Micro:bit 能很好地胜任各种机器人相关学习和开发场景。

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

Micro:bit 搭载了 ARM 的 M0 处理器，运算性能满足绝大多数的机器人入门场景。

Micro:bit 的编程方式

Makecode 图形化编程

对于初学者来说，Micro:bit 的优势更多在于其丰富的编程语言支持和友善的交互界面。

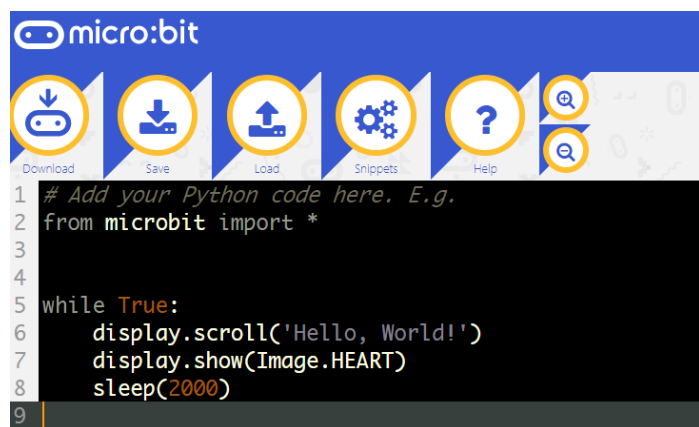


以微软的 MakeCode 在线编程平台为例

用户可以在网页中拖拽相关模块来获取传感器的数据，通过一定的循环，判断等逻辑功能快速地对数据进行处理。micro:bit 的出现大大降低编程的入门成本，使学习的重心能更好地放在机器人控制上。

MicroPython 代码编程

对于有编程基础的高级用户来说，Micro:bit 可以使用 Python 进行编程，它功能强大，应用广泛。MicroPython 是 Python 语言的子集，延续使用 Python 语言的规范，使用上功能更加强大。



本教程中，使用的是 makecode 的图形化编程。运用 Python 进行编辑的教程，请等待后续更新。

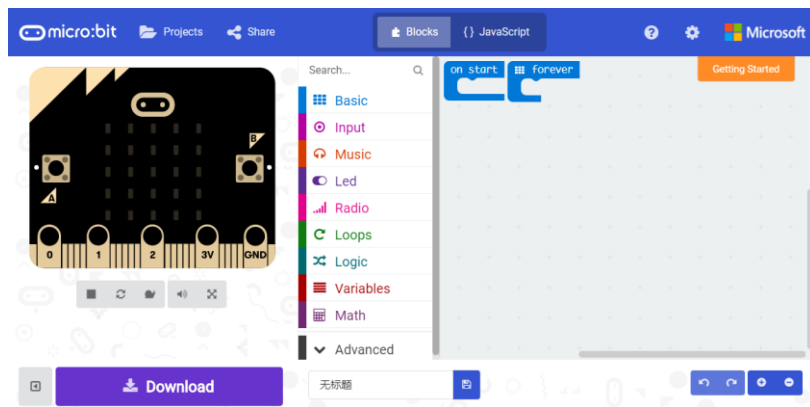
怎么对 Micro:bit 进行 Makecode 图形化编程？

进入 Makecode 编程界面

在浏览器中输入以下网址，即可进入编程界面，注意必须在联网的电脑上登录，网址为：

<https://makecode.microbit.org/#>

第一次使用时会显示空白界面，以及英文界面。



Makecode 界面具有记忆功能，以后每次登陆会保持上一次使用的最终保存界面以及使用语言。

界面认知

Makecode 界面，按照功能区域，清晰明了，分为以下几个部分



模拟窗口：模拟 Micro:bit 工作的状态，在编程过程中，我们可以通过观察模拟窗口，观看初步的运行结果。

指令区：拖动指令区的指令可以对 Micro:bit 进行编程控制。

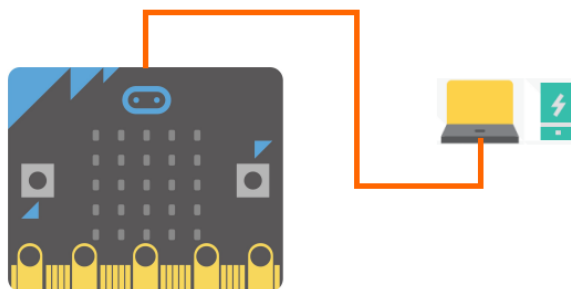
脚本区：程序的编写区域，拖拽指令区的指令在此编写程序；在删除程序模块时，直接拖动模块到指令区。

设置：常用功能为切换语言  Language 。

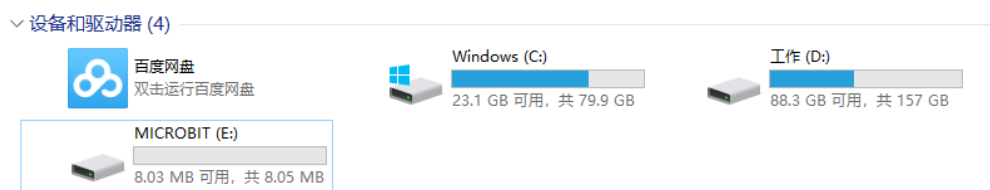
命名：对当前编辑的程序进行命名，默认为“无标题”。

程序下载

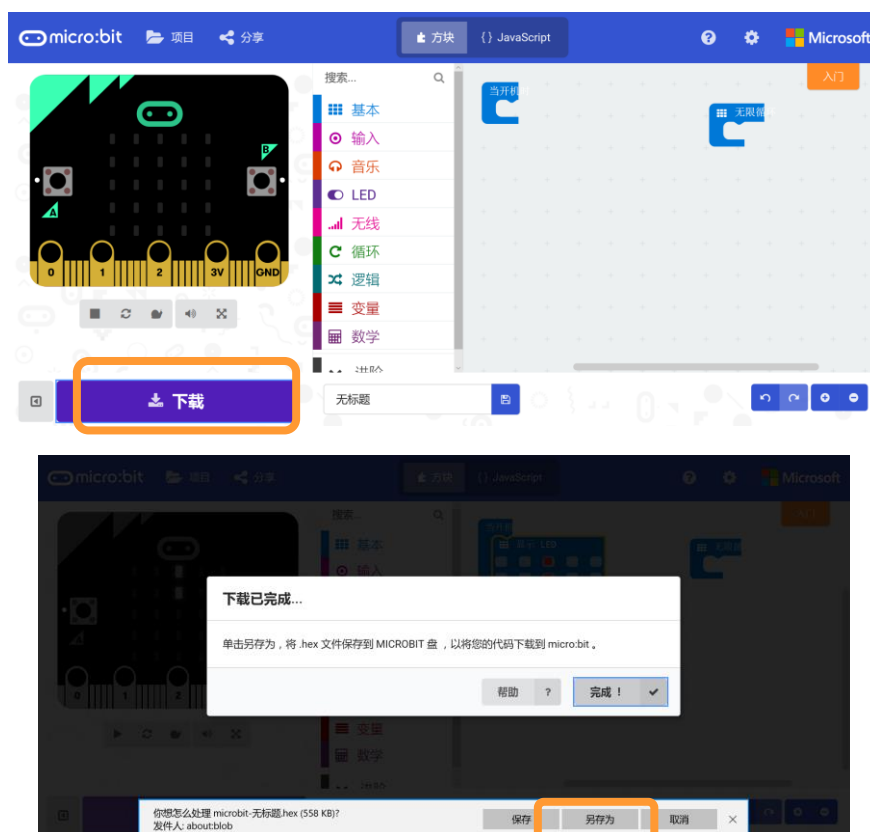
1. 通过 USB 连接线将 micro:bit 和电脑相连接。



2. 打开“我的电脑”，在“我的电脑”中会出现一个硬盘----“micro:bit”，说明板子可以正常使用。

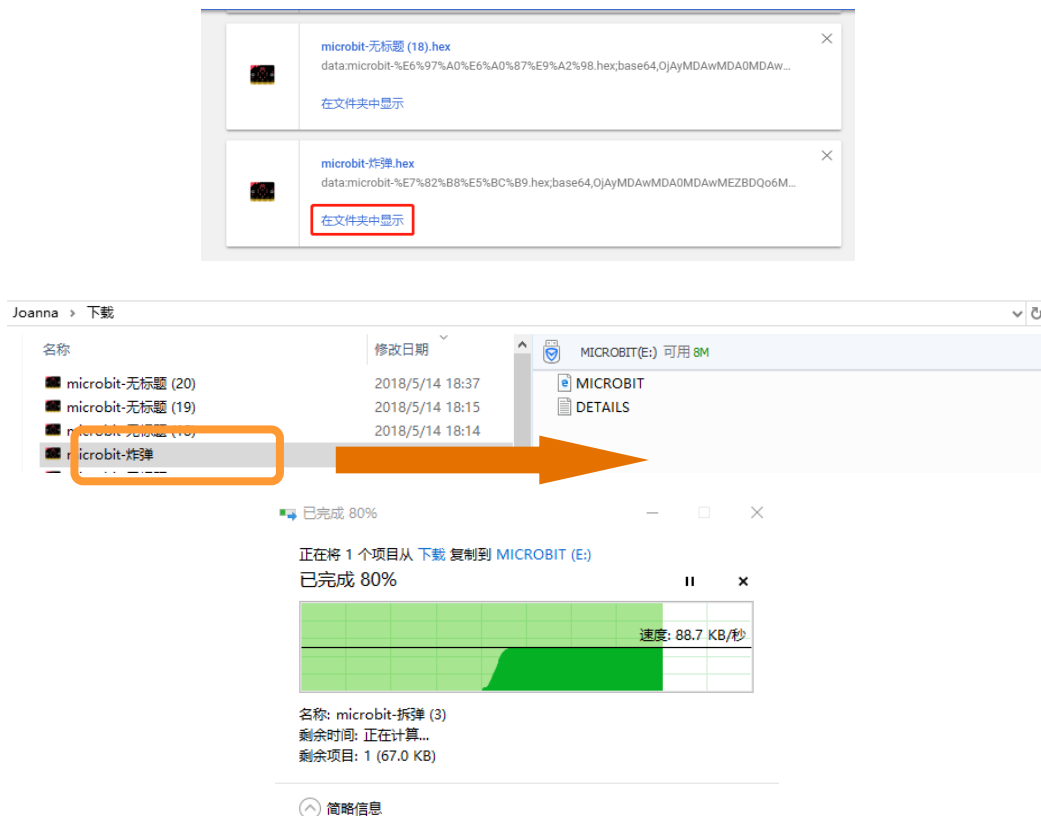


3. 点击模拟窗口正下方的“下载”，弹出的对话框下选择“另存为”。

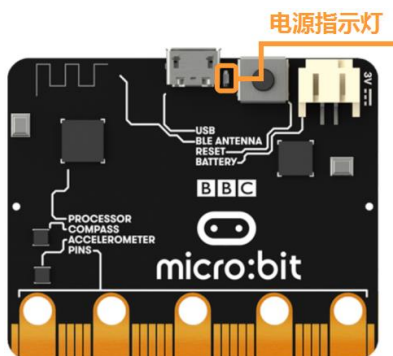


在弹出的对话框中，选择把.hex 文件另存到“MICROBIT 磁盘”，再点击“保存”。

如果没有出现另存为，需要你找到下载的.hex 文件，复制或者拖入到“MICROBIT 磁盘”。

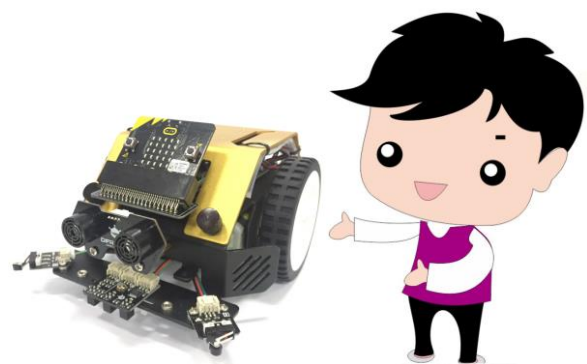


下载过程中 micro:bit 主板背面的电源信号灯会闪烁，当下载完成后电源信号灯停止闪烁，保持长亮。



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

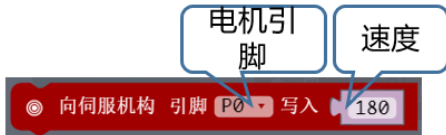
- 1. 学习怎么让轮子动起来
- 2. 学习差速转向的原理
- 3. 学习创建子函数

电子模块

电子模块	图片	功能说明
电机		电机是可以带动平台运动的，实现旋转运动

1.1 小车动起来

知识要点

积木	用法说明	功能说明								
电机速度控制		控制电机的速度及方向 <table><tr><th>数字</th><th>0</th><th>90</th><th>180</th></tr><tr><td>速度与方向</td><td> “后退” 最大速度</td><td>0 停止</td><td> “前进” 最大速度</td></tr></table>	数字	0	90	180	速度与方向	 “后退” 最大速度	0 停止	 “前进” 最大速度
数字	0	90	180							
速度与方向	 “后退” 最大速度	0 停止	 “前进” 最大速度							
主循环模块		需要重复执行的动作，都要放在该模块中								

动手实践

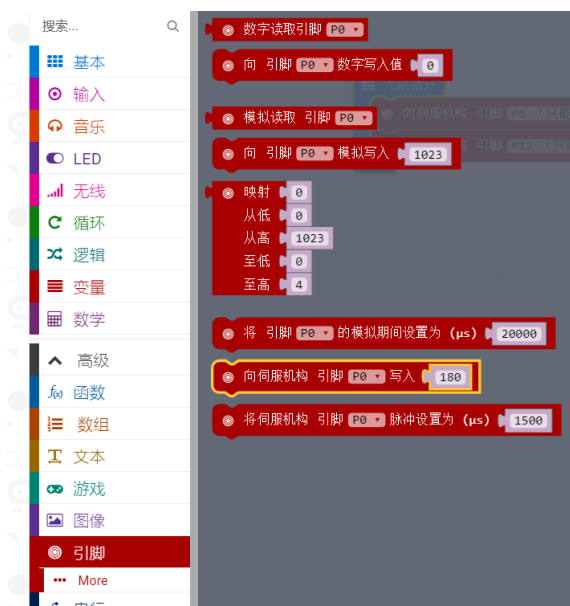
(1) 进入网址: <https://makecode.microbit.org/#>

(2) 删掉不需要的模块“当开机时”。

左键拖动, 就可以轻松删除啦。



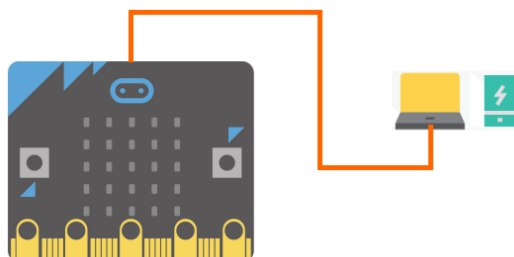
(3) 找到电机速度控制模块。



(4) 拖入到循环模块中, 注意左轮对应引脚为 P12, 右轮对应引脚为 P8。



(5) 将程序下载到 Max:bot 中, 小车就向前跑起来了。



下载程序完成后，就可以拔掉接在电脑上数据线，将电池的线接好。

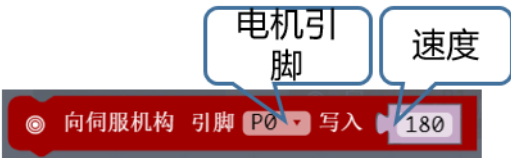
注意：开关一定要打开，小车才会跑哦。



探究思考

怎么让小车一直后退呢？试一试吧。

提示：需要改变电机速度控制模块的数字，可参考知识要点中对于该积木功能说明。



1.2 小车转圈圈

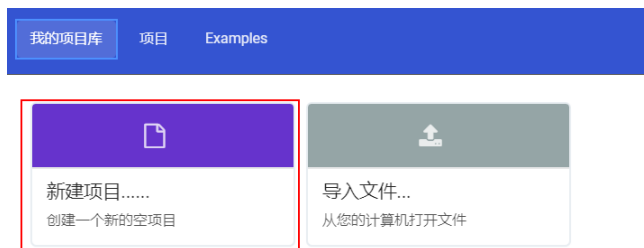
知识要点

积木	用法说明	功能说明
定义子函数		当一个连续的动作，需要在循环中出现多次，为了使得主循环更加清楚，需要定义子函数。
调用子函数		在主循环中，需要使用子函数这一系列动作时，拖入到主循环中。 需要与“定义子函数”对应使用。

动手实践

(1) 新建项目——小车转圈圈

点击 项目，需要创建新的项目，不然会直接使用之前的项目名称下保存，不利于后期的项目整理。



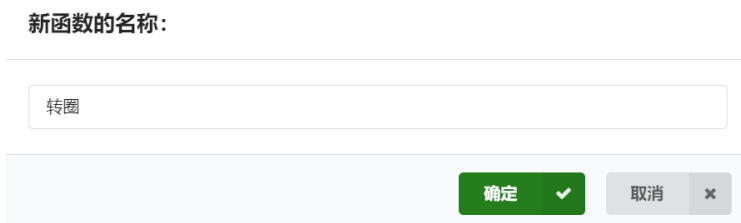
命名为“小车转圈圈”。



(2) 创建一个子函数，点击“创建一个函数”。



给子函数命名。



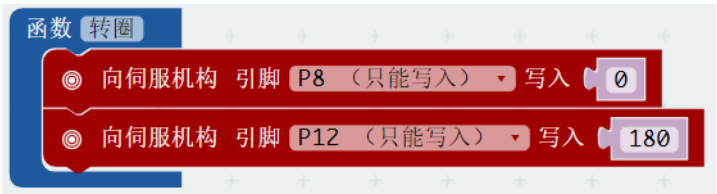
(3) 怎么实现小车的转向呢？

这里我们用到的是差速转向的原理。

在实际生活中，车辆通过控制左右两个驱动轮的转速实现转向。驱动轮转速的不同可以通过操作安装在左右半轴上的两个单独的离合器或制动装置来实现。

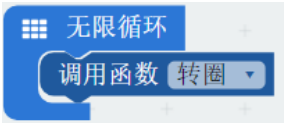
目前，几乎所有链轨（履带）车辆都采用这个方法实现转向，例如坦克。

需要考虑的是左右两个轮子具有不同的速度，就能成功转向了。

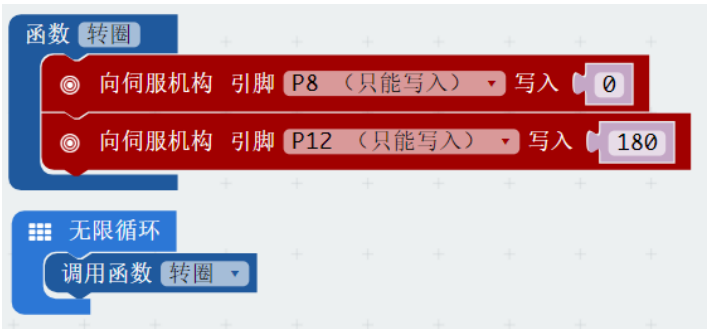


其中：“180”与“0”分别是前进最大速度和后退的最大速度。

在主循环中，调用“转圈”这个子函数。



(4) 将程序下载到 Max:bot。



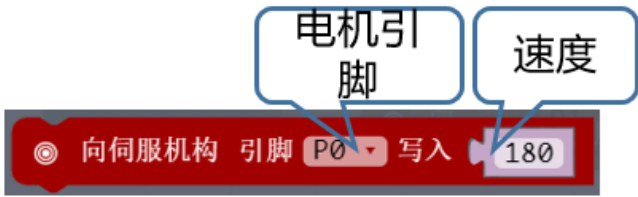
你会看到小车在顺时针原地转圈。

注意：拔掉下载程序数据线，打开 Max:bot 的开关。

探究思考

怎么让小车可以逆时针转圈呢？

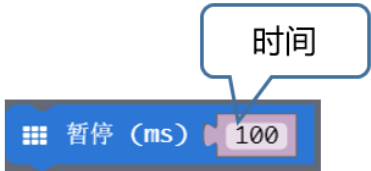
提示：需要改变电机速度控制模块的数字。



1.3 小车走方形

知识要点

积木	用法说明	功能说明
----	------	------

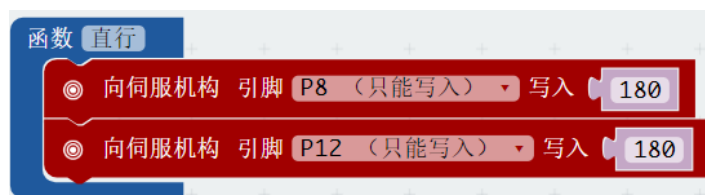
延时模块		某一个动作或状态需要在持续执行一段时间
------	---	---------------------

动手实践

- (1) 新建项目——小车走方形
- (2) 创建两个子函数，“直行”和“向右转”。



- (3) 编写直行的程序。



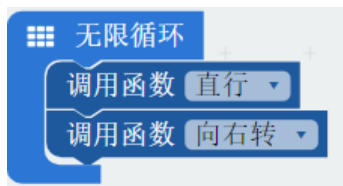
- (4) 编写转弯程序



与上面“转圈”程序比较：

“90” 代表轮子转速为 0，实现左侧轮子（P12）转，右侧轮子（P8）不转，小车向右转。

(5) 在主循环中，调用“直行”和“向右转”子函数。



这样就能实现走正方形了吗？

走正方形，需要设定直线和转向的运行时间。调用延时模块，设定直行和转向的时间。



完整程序如下：

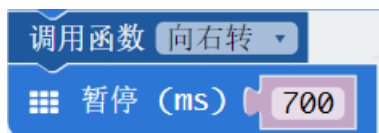


其中注意的是：

a. “2000” 影响的是正方形的边长的长度

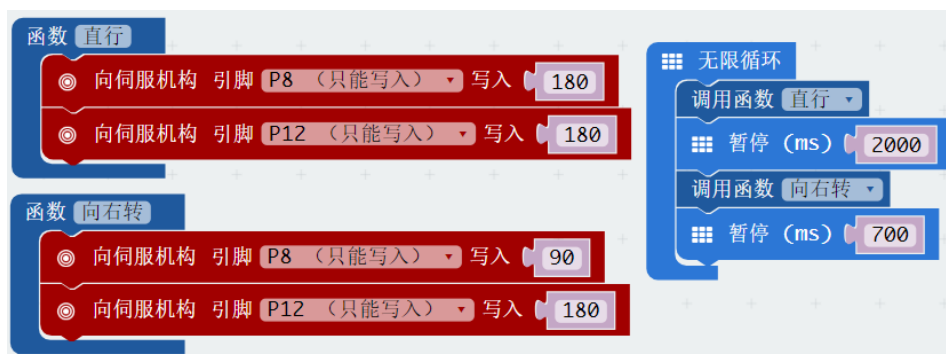


b. “700” 影响的是转向的角度大小。



由于每个小车的电池电压不同，需要根据实际情况来调整这两个数字，才可以实现走正方形。需要你耐心进行调试哦。

(6) 将程序下载到 Max:bot。



你会看到小车顺时针走正方形。

探究思考

怎么让小车可以走三角形呢？



第二章 聪明的 Max:bot

Max:bot 可以听话的跑起来了，但是很困扰的是，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？

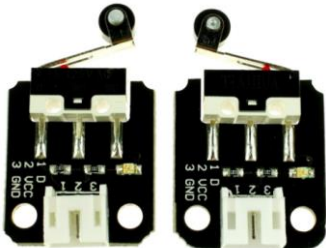


学习目标

1. 学习碰撞开关的使用
2. 运用条件判断模块

3. 运用流程图辅助编程

电子模块

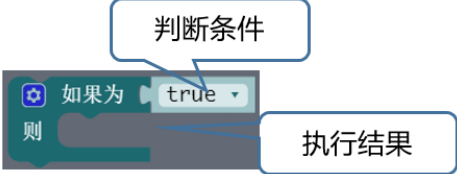
电子模块	图片	功能说明
碰撞开关		碰撞开关类似于按钮，可以检测到是否碰撞到障碍物

2.1 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。



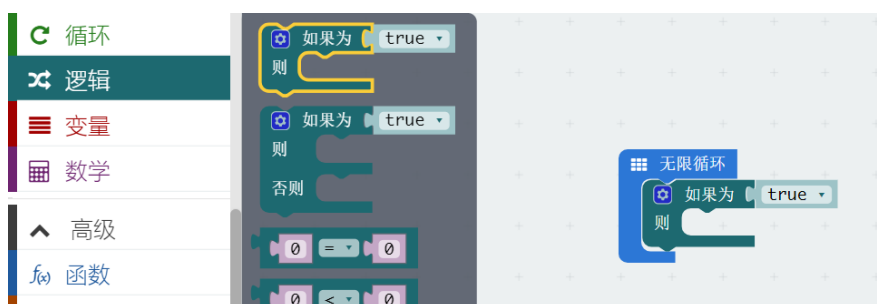
知识要点

积木	用法说明	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
条件判断		当判断条件满足时候，则会执行“则”后面的执行结果； 如果不满足直接进入下面的程序。

逻辑判断		通常用来作为判断条件，判断某一个值的范围 (“<” “>”) 或者状态 (“=”)
读取数字引脚		可以读取数字引脚的状态，“0”或“1”

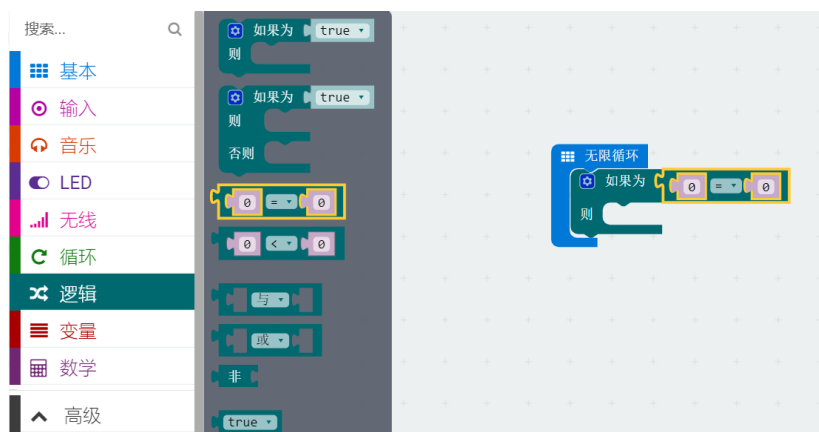
动手实践

- (1) 新建项目——害羞的 Max:bot
- (2) 将条件判断模块，拖入到主循环中。

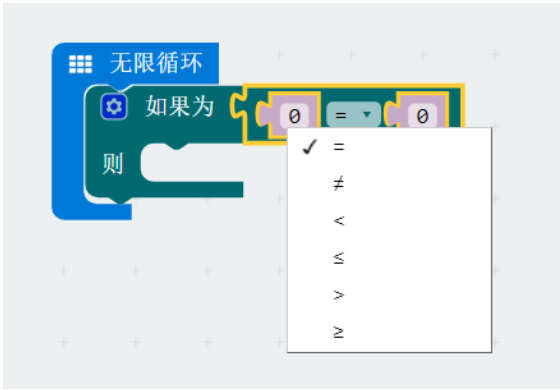


当判断条件成立（即为真）时候，则会执行“则”后面的执行结果；如果不满足直接进入下面的程序。判断条件只有两种状态，真或假。

- (3) 找到逻辑判断模块

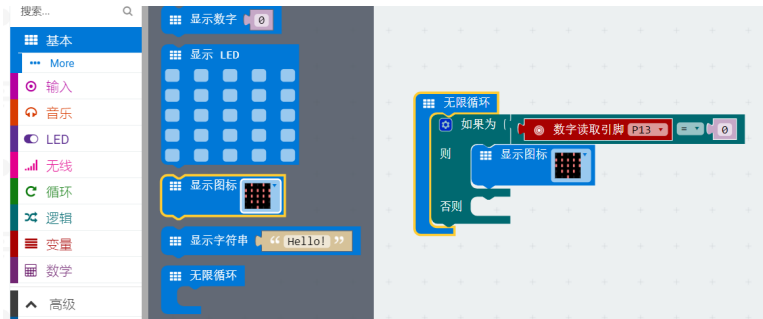


逻辑判断模块的下拉选项中，有不同的判断符号。



(4) 判断碰撞开关的状态

当 Max:bot 受到触碰的时候，他会紧张害羞，也就是需要判断右侧碰撞开关（接在引脚 P13 上）的状态。



那么如果触碰到碰撞开关的时候，碰撞开关会发出信号“0”吗？

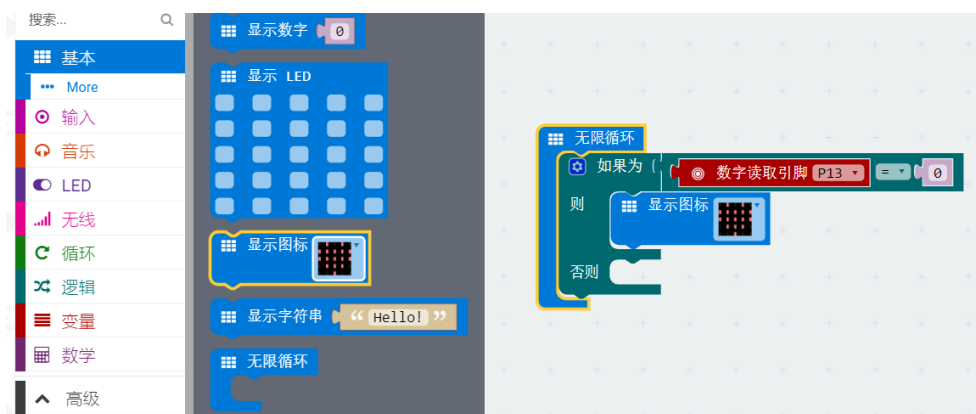
碰撞开关的输出状态对应如下：

状态	值
有东西触碰（指示灯亮）	0
无东西触碰（指示灯灭）	1

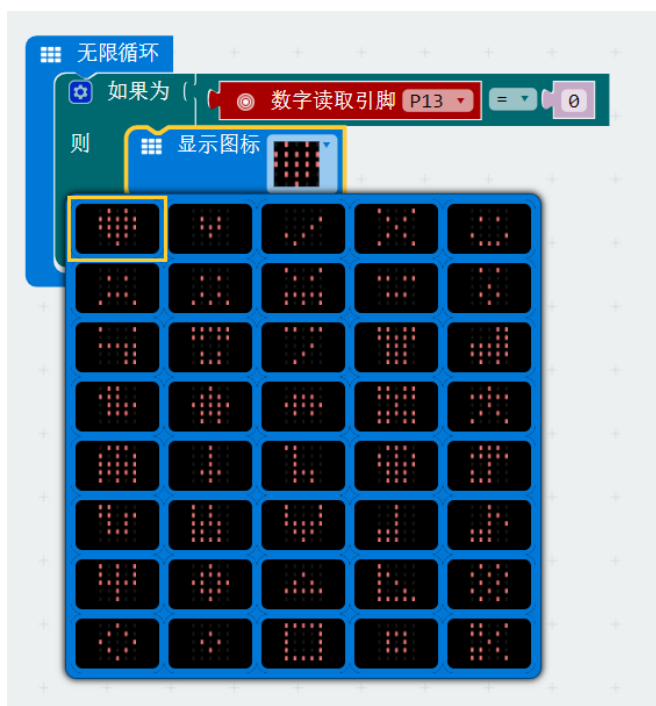
(5) 制作心跳的显示效果

当 Max:bot 接收到触碰的时候，就会害羞紧张，心跳加速。

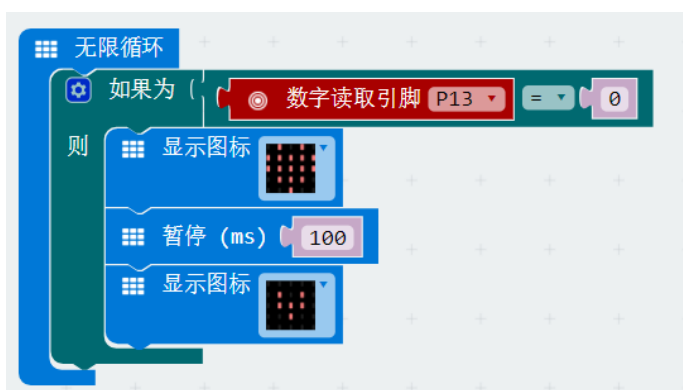
运用图案显示模块，制作一个心跳的效果吧。



下拉可以选择一个“小心形”



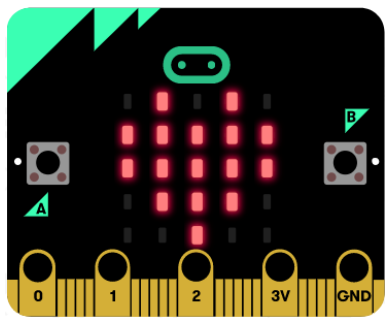
加入一个延时模块，就可以有心跳的效果啦。



(6) 将程序下载到 Max:bot

当你碰触到碰撞开关（右侧）就会发现 Max:bot 就心跳加速了。

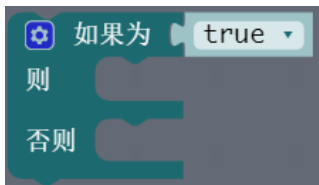
真的是个可爱的机器人呢。



探究思考

Max:bot 是个很友好的小伙伴，很当你不触碰 Max:bot 的时候，它会对你露出笑脸。

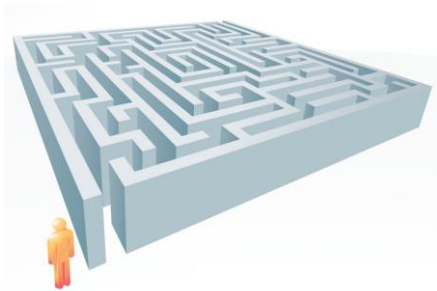
该怎么实现呢？需要用到以下条件判断模块。



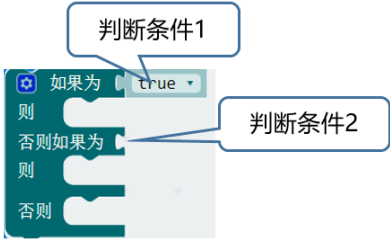
“否则”就是当判断条件不成立时，就会执行“否则”中的程序模块。

2.2 迷宫游戏

搭配了碰撞开关的 Max:bot 机器人，可以感知触碰了。那么，如果把 Max:bot 放在迷宫中，它能顺利的走出来吗？



知识要点

积木	用法说明	功能说明
多条件判断		1. 当判断条件 1 满足时候，则会执行对应的“则”后面的执行结果； 2. 当判断条件 2 满足时候，则会执行对应的“则”后面的执行结果； 3. 当条件 1 和条件 2 都不满足，会执行对应的“否则”后面的结果。

动手实践

(1) 创建新项目——迷宫游戏。

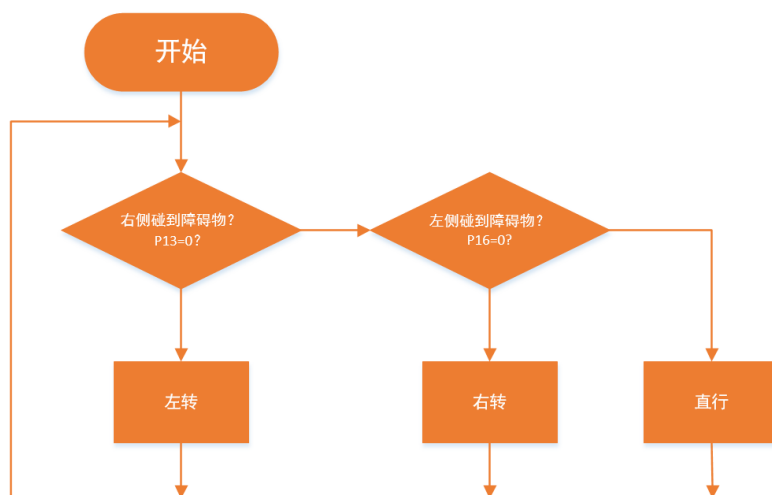
(2) Max:bot 走迷宫

Max:bot 需要怎样走出迷宫呢？要帮助 Max:bot 思考一下，怎么来躲开障碍物

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，试着去编写一下程序吧！

(3) 将程序下载到 Max:bot。

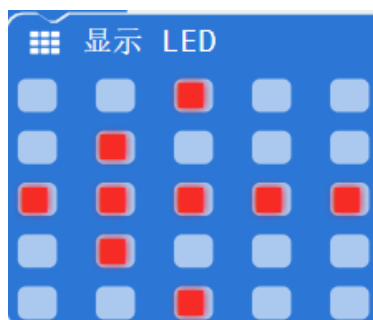
由于电池电压不同，所以 Max:bot 的转向的时间需要调整，才能够实现完美的躲开障碍物。



探究思考

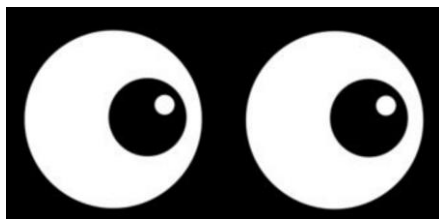
Max:bot 能够聪明的走出了迷宫，怎样来同步显示 Max:bot 的状态呢。

需要用到图案显示模块。



第三章 感光的眼晴

Max:bot 可以聪明的躲开障碍物啦，就像眼睛一样，那么 Max:bot 还有哪些灵敏的感知系统呢？

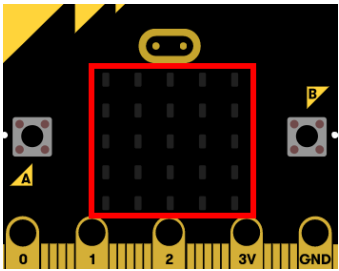


Max:bot 还可以感知光，并且可以准确的知光的强度，一起来探索这个功能吧。

学习目标

1. 学习光敏传感器
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
光敏传感器		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。

3.1 探秘光强的大小

人类的眼睛会在很强的光照下，会眯眼来保护眼睛，在光强比较弱的情况下，会看不清东西，那么我们怎么知道光的强度到底是怎么变化的？不同颜色的光照射，强度由什么变化呢？



那么需要 Max:bot 来帮忙了。先来试试，怎么用 Micro:bit 来把光强用数字表达出来吧！

知识要点

积木	用法说明	功能说明
----	------	------

读取光强值		读取光的强度值，范围为“0~255”，表示光的强度由弱变强。
显示数字		显示数字

动手实践

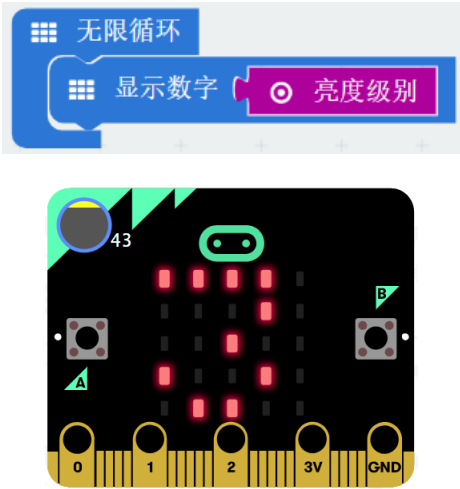
- (1) 创建新的项目——探秘光强的大小
 - (2) 需要显示数字模块来帮忙
- 默认是显示数字“0”。



- (3) 获取光强的值



- (4) 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灯光效果，光强的大小是多少呢？思考，各个环境的光强设计合适吗？

环境	光强
卧室	
厨房	
客厅	

3.2 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功能，来制作一个飞蛾机器人吧。



知识要点

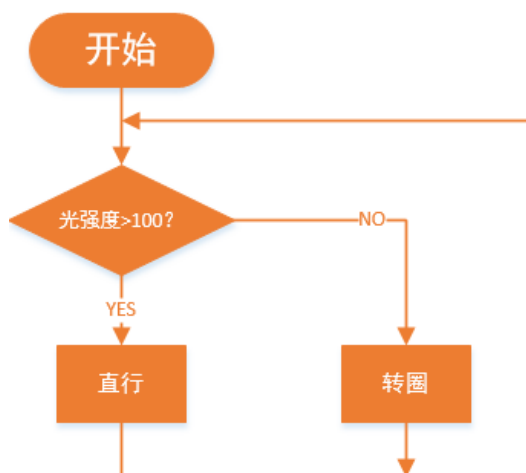
积木	用法说明	功能说明
条件判断		当判断条件满足时候，则会执行“则”后面的执行结果 1； 如果不满足，执行“否则”后面的执行结果 2.

动手实践

(1) 创建新的项目——飞蛾机器人

当光强度到一定值（样例中为 100）的时候，飞蛾机器人就可以向光运动（向前）；当光强度小于一定值（100）的时候，飞蛾机器人原地旋转。

怎样实现这个功能呢？请参考以下流程图进行程序编写吧。



(2) 用到条件判断模块和逻辑判断模块。



其中，光强的值，需要设置一个合适的大小



如果太大，则需要很强的光，飞蛾机器人才能向前运动；

如果太小，则会在环境光强下，飞蛾机器人会直接向前一直运动；

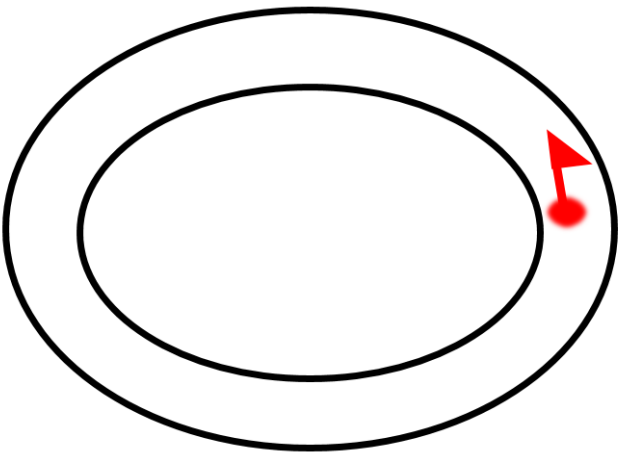
请根据你自己的环境，选择合适的值。

(3) 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小红旗出发，跑完一圈吧！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，你很难控制你的 Max:bot 听你指挥。

第四章 秘密武器超声波

看到 Max:bot 前面的大眼睛，你是不是在想，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。

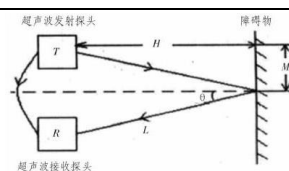
学习目标

- 1. 学习超声波的工作原理
- 2. 添加扩展包功能
- 3. 运用流程图辅助编程

电子模块

电子模块	图片	功能说明
------	----	------

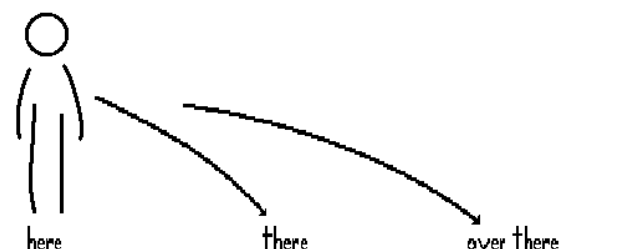
超声波传感器



发射端发出超声波之后，遇到障碍物会反弹回去，被接收端接收，从而得到障碍物的相关信息

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出距离了。



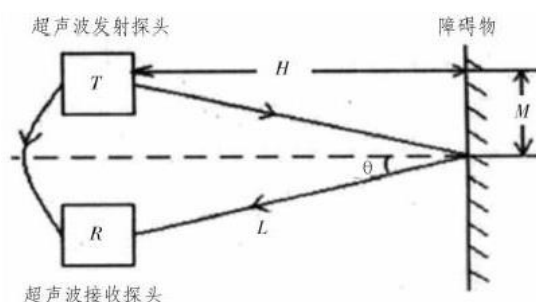
什么是超声波呢？

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”。

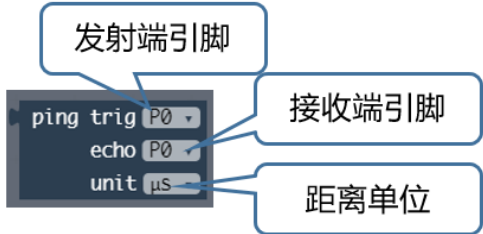
超声波怎么测距？



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，就等到了距离 H。

有没有跃跃欲试，想试试怎么轻松获得距离？

知识要点

积木	用法说明	功能说明
读取超声波距离值		读取超声波距离值，需要设定好超声波传感器上发射端和接收端的引脚以及距离单位。

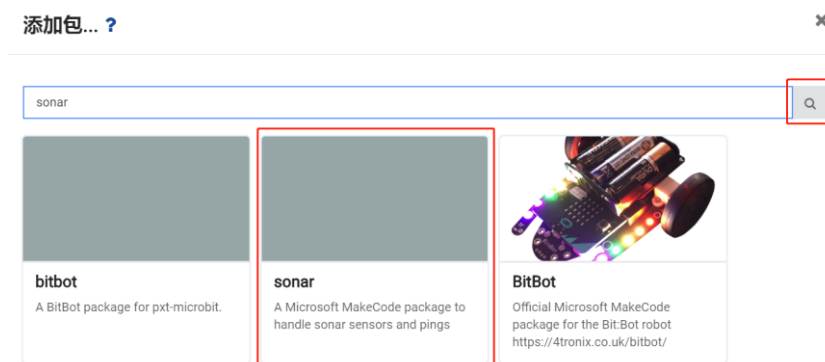
动手实践

- (1) 创建新的项目——超声波测距
- (2) 需要添加超声波相关模块

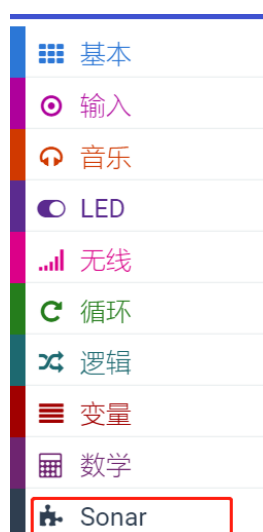
在 Makecode 的界面上，是一些常用的简单功能模块，如果你需要一些特殊功能，比如超声波功能，需要添加对应的功能库。



输入“sonar”进行搜索，点击“Sonar”模块就可以添加进去啦。



你就可以看到指令区，出现了“Sonar”区域。



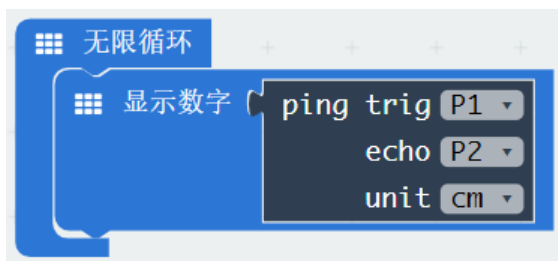
(3) 读取并显示超声波的距离值。

将读取超声波距离值模块拖入到数字显示模块中。



其中“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”

(4) 将程序下载到 Max:bot，就可以观察到与前方障碍物的距离值啦。



注意：要打开开关，前方有障碍物 5cm~300cm 的范围比较准确。

探究思考

利用超声波传感器，怎么去测量身高呢？



Tips: 想获得精确身高，你可能需要注意：

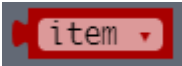
1. 超声波的固定位置和方向
2. 在 10cm 范围进行校准，防止出现大的偏差

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；如果可以通过技术，帮助汽车在行驶过程中，去判断危险的存在，则会有效的避免一些事故，保证人身安全。



知识要点

积木	用法说明	功能说明
变量		建立一个变化的量，可以实现动态的效果
变量赋值		给变量“item”赋值，默认模块意义为：item=0
逻辑“与”		需要逻辑“与”，左右两个条件必须同时满足

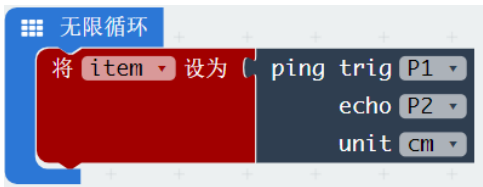
动手实践

- (1) 创建新的项目——行车安全仪
- (2) 使用变量 item

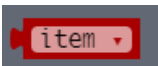
感知系统可以实时的检测外界环境。例如光敏的值，会实时监测到不同的变化的光的强度值，但是在程序中我们可能要多次用到这一值，通常把需要存储和调用的值，命名为“item”。



在行车安全仪中，我们实时监测的数据，就是超声波距离值。将这一值，赋值给变量“item”。

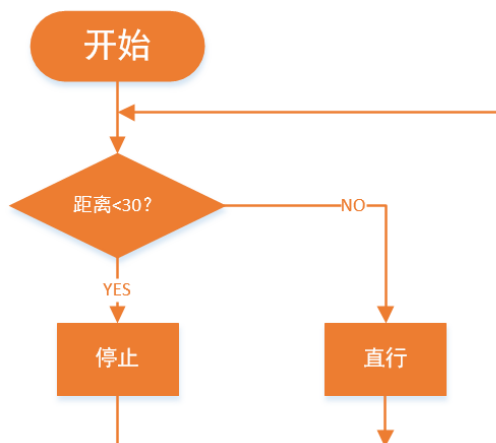


当需要调用超声波距离值时，直接使用变量“item”就可以了。

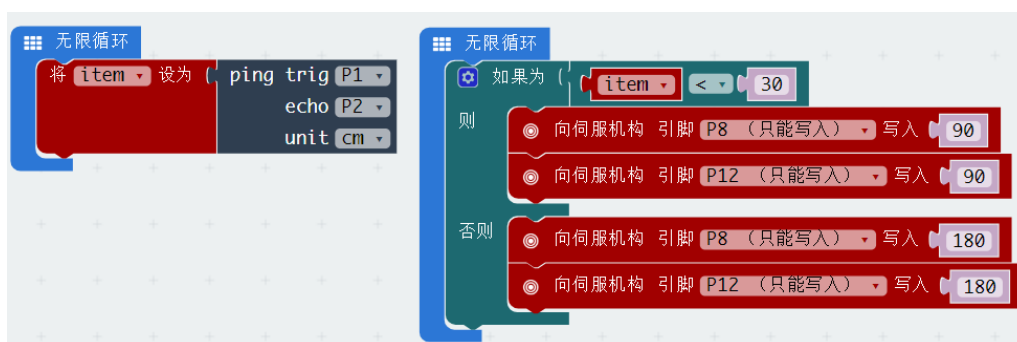


- (3) 当小车前方的距离（前车）障碍物比较近时，那么小车会自动停下来，防止追尾；如果前方有足够的距离，可以放心的向前行驶啦。

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) 将程序下载到 Max:bot，小车就可以安全的向前行驶啦。



为什么需要两个无限循环模块呢？



当需要两个部分同时进行时，可以调用两个主循环模块来实现，这两个部分是同时进行，不会干扰的。

探究思考

在正常的行驶中，不会一下子就停车的，会慢慢的降低速度。距离变小，速度也是慢慢降低的。

你能实现吗？



建议运用逻辑“与”模块，来多重判断距离。



第五章 勇往直前的 Max:bot

Max:bot 机器人是个勇往直前的勇者，能够智慧判断前面是否有障碍物，同时它还有个神奇的技能，还没有告诉你，它有个朝向地面的眼睛，可以帮助它避免从高处跌落，可以实现沿着固定轨迹行走。快来了解一下吧。



学习目标

- 1. 学习寻线传感器的工作原理
- 2. 逻辑“与”的理解与运用

电子模块

电子模块	图片	功能说明
寻线传感器		寻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（三个检测头，只用到了左右两个）

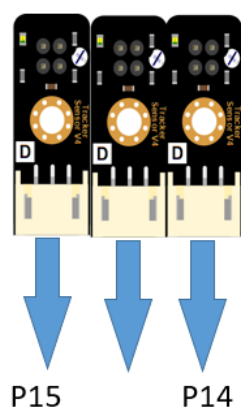
5.1 悬崖勒马

看，前面有个悬崖，Max:bot 并没有掉下去，它可以自动后退。成功的避免你的 Max:bot 小伙伴“机”毁“车”亡。多亏了寻线传感器。



寻线传感器怎么工作的呢？

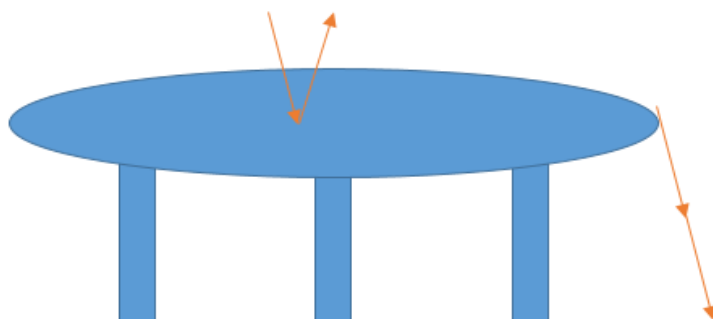
寻线传感器上面有三个检测头，分别连接在 P15、P14。



每个探头是成对出现的红外探头。



其中一个发射红外传感器发出红外线，另外一个接收。工作原理与超声波传感器类似，但是得到的结果并不会转化成距离，只能判断有没有障碍物。



工作原理：

当在桌面上时，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”；

当在桌面边缘，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”。

注意，这里的桌面为浅色，如果为深色，能够吸收所有颜色的光（包含红外线）而基本上不反射光线，所以深色桌面不能实现。

知识要点

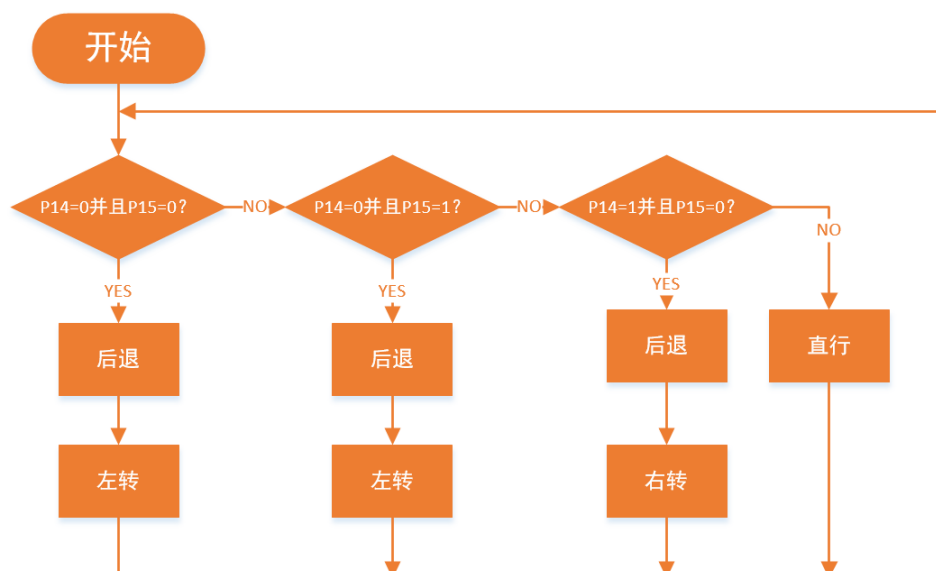
积木	用法说明	功能说明
逻辑“与”		需要逻辑“与”，左右两个条件必须同时满足

动手实践

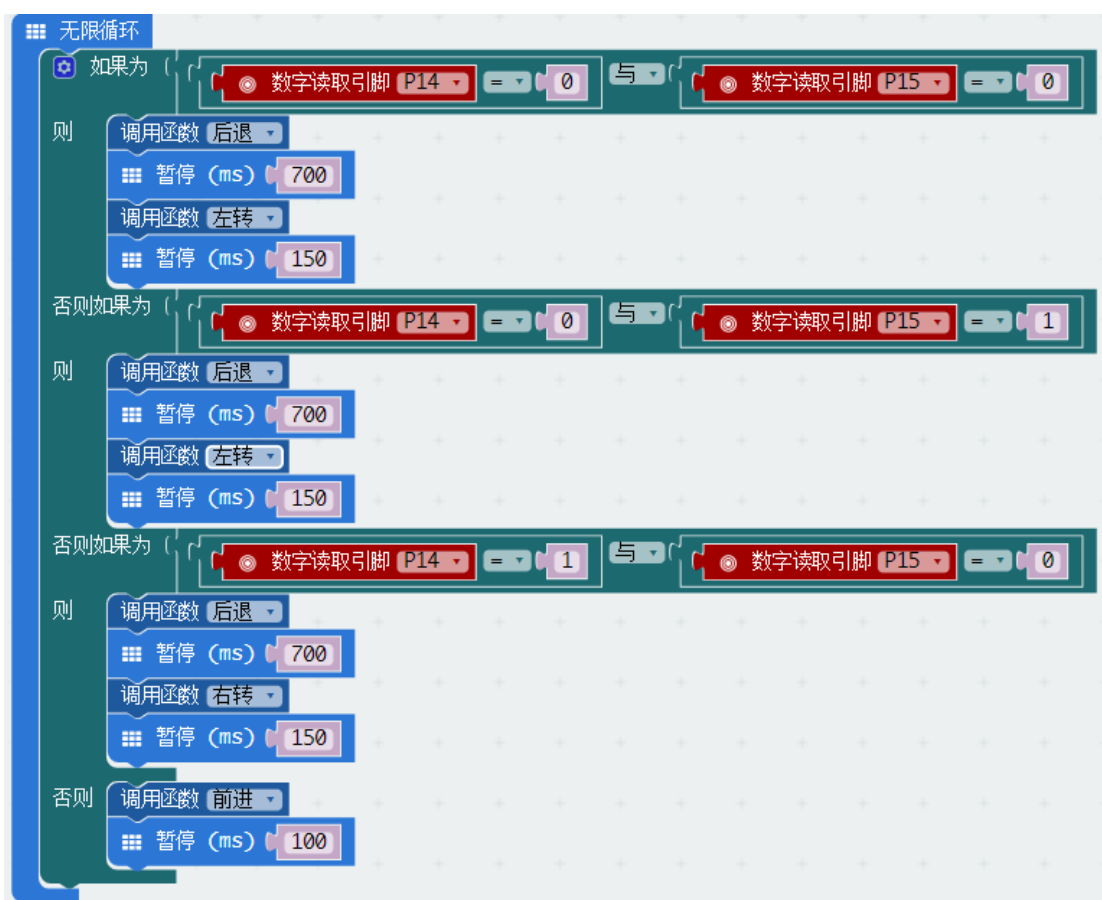
- (1) 创建新的项目——悬崖勒马
- (2) 在悬崖勒马项目中，需要用到两个寻线传感器，分别连接在 P15、P14。Max:bot 在桌面运动的时候有以下几种状态：

状态示意图	状态描述	传感器状态
	Maxbot在桌面	P14=1; P15=1;
	Maxbot两个探测器在桌子边缘	P14=0; P15=0;
	Maxbot右侧探测器在桌子边缘	P14=0; P15=1;
	Maxbot左侧探测器在桌子边缘	P14=1; P15=0;

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



(4) 将程序下载到 Max:bot，小车就可以安全的在桌面上行驶啦。



Tips: 运行速度不宜过快，防止失误掉落。

探究思考

在生活中的哪些场景中，需要防跌落的功能呢？



随着生活中多种科技化工具的丰富，一种扫地机器人也成为居家必备的一种工具，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。

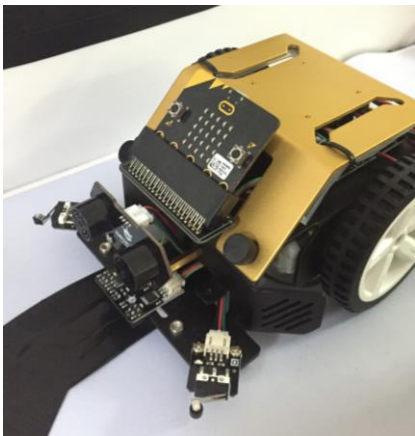
当然，防跌落功能，作为扫地机器人的一个自我保护功能，在底部具有多个探测器，才能确保避免跌落发生，通常在底部边缘有一圈的探测器。

还有哪些场景需要这一功能呢？小朋友们，你想到了吗？

- -
 -

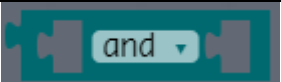
5.2 循“轨”蹈矩

Max:bot 最喜欢按照自己的地图去寻宝啦，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

积木	用法说明	功能说明
逻辑“与”		需要逻辑“与”，左右两个条件必须同时满足

动手实践

- (1)

创建新的项目——循轨蹈矩
- (2)

在本项目中，需要让 Max:bot 沿着黑色的轨迹行走，绘制黑色轨迹或使用黑色胶带（本样例中，黑线的宽度能够保证左右两侧的寻线传感器都在黑线上）。



需要注意的是，Max:bot 的左右两个探测器（P14、P15）可以同时保持在黑色轨迹上，我们图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。

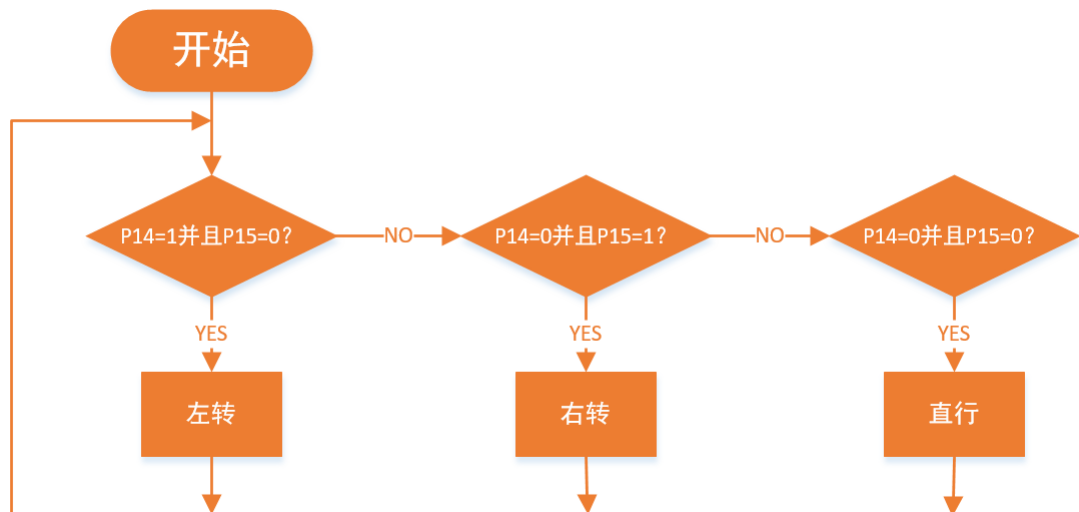
(3) 沿着直线行走，Max:bot 会遇到以下路况：

状态示意图	状态描述	传感器状态
	Maxbot两个探测器在桌子边缘	P14=0; P15=0;
	Maxbot右侧探测器在桌子边缘	P14=0; P15=1;
	Maxbot左侧探测器在桌子边缘	P14=1; P15=0;

(4) 根据上述分析的 Max:bot 可能出现的状态，需要为编程制作流程图。



你可以自己试着制作
程序流程图



(5) 将程序下载到 Max:bot，小车就可以安全的在桌面上行驶啦

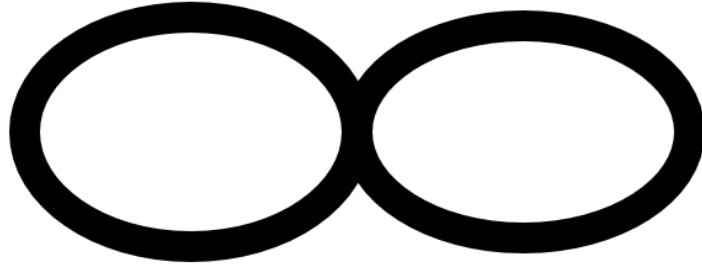


Tips: 合适的速度和运行时间才能使得 Max:bot 沿着直线行走。

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 可以寻线吗？



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？

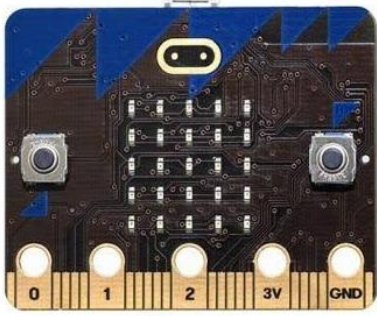


Max:bot 的大脑 Micro:bit 具有无线通信功能，是不是可以实现更多的远程控制啦？可以实现什么功能呢？

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	用法说明
Micro:bit		Micro:bit 上带有无线模块和加速度计

6.1 Say Hi

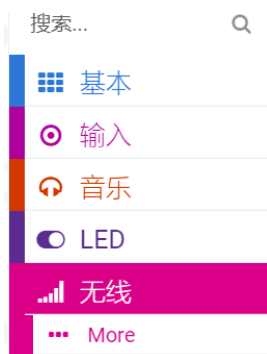
当 Max:bot 遇到了另外一个 Max:bot，先来打个招呼吧！



当一个 Max:bot 通过 Microbit 发出一个信号时，另外一个 Max:bot 能够接收到信号，并作出反应，需要 Micro:bit 的无线功能。

什么是无线功能？

在无线功能中，需要两个 Micro:bit，能够发出信号的一个叫做“发射端”，接收信号的叫做“接收端”。



无线功能的使用中有三个常见的模块：

1、设定信号组



设定信号组，就是给你的信号通道命名，需要发射端和接收端有相同的名字，才可以进行通讯，一共可以设置 0~255 个信号组。

注意，如果是多个小伙伴同时进行，需要设置不同的信号组，否则会出现接收信号混乱。

2、无线发射信号



无线功能，可以发送文字和数字，以上是发送数字命令。如果我们的变量也是数字

也可以把变量模块拖入。



以上是发送文字的命令。

3、无线接收信号



接收端需要识别接收到的信号，做出对应的动作，这里对应发射端发射出来的是数字。



以上是判断是否接收到文字的指令模块。

4、设置无线发射功率



无线发射功率有 1~7 等级，等级越高，说明能够传播的距离越远。

知识要点

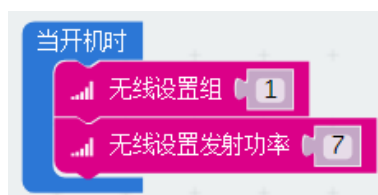
积木	用法说明	功能说明
初始化模块		在程序中，只需要执行一次
设定信号组		给信号传递的通道进行定义
无线发射信号		通过无线发射数字“0”
无线接收信号		判断，是否接收到数字信号
设置无线发射功率		设置无线发射的功率大小

动手实践

发射端

- (1) 新建项目——say hi 发射端
- (2) 对发射端进行初始设置

“当开机时”中的程序模块，只会在电源通电后，执行一次，不会重复执行。



(3) 设置需要发射的内容

先来打个招呼吧！



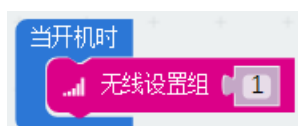
(4) 将程序下载到发射端



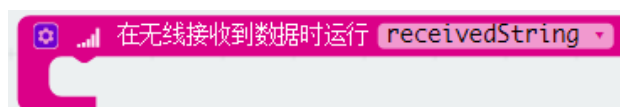
接收端

(1) 新建项目——say hi 接收端

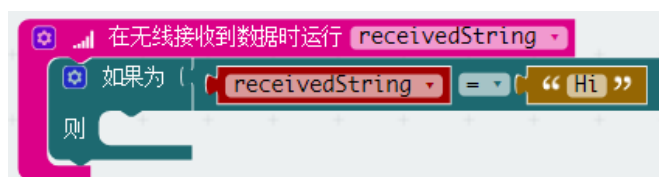
(2) 设置与发射端相同的组



(3) 判断是否接收到信号



(4) 判断是否是接收到正确的信号



注意，“Hi”是文字形式，需要用到以下模块



(5) 当收到 “Hi” ， 那么就会露出笑脸。



(6) 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？通过你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 上面集成了加速度计。

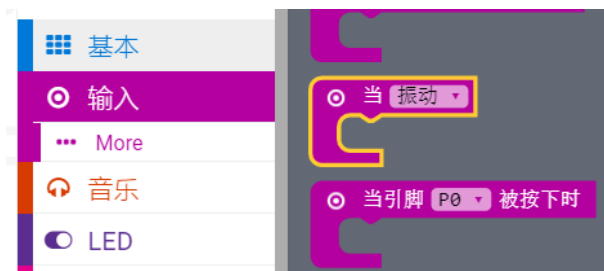
什么是加速度传感器？

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。



在指令部分，可以用以下指令来判断 Micro:bit 的状态。

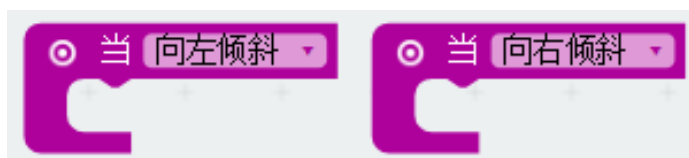


1、 手势判断

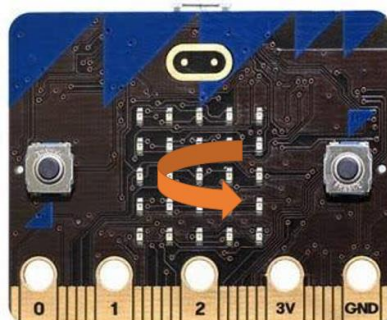


可以判断，是否进行晃动。这个功能和计步器类似。

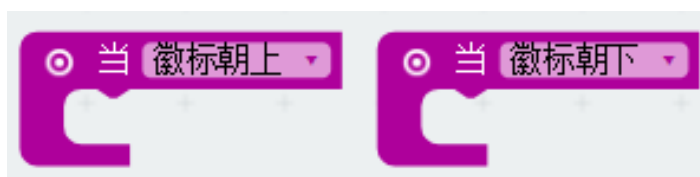
2、 左右倾斜判断



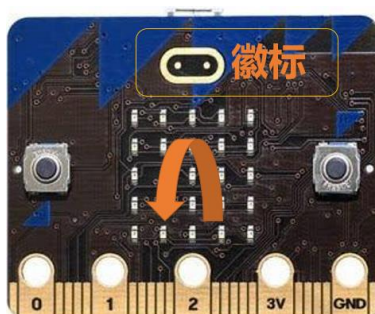
判断 Micro:bit 是否在进行向左或者向右倾斜。



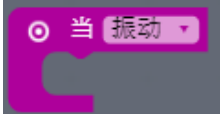
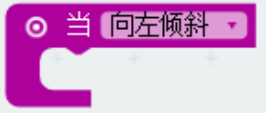
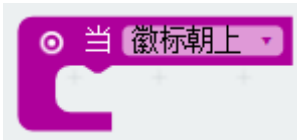
3、 前后倾斜判断



判断 Micro:bit 是否在进行向前或者向后倾斜，即徽标是朝上的状态，还是朝下的状态。



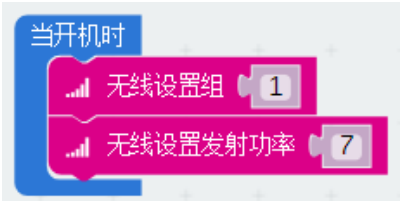
知识要点

积木	用法说明	功能说明
手势判断		可以判断，是否进行晃动。
左右倾斜判断		判断 Micro:bit 是否在进行向左或者向右倾斜。
前后倾斜判断		判断 Micro:bit 是否在进行向前或者向后倾斜，即徽标是朝上的状态，还是朝下的状态
箭头显示模块		显示箭头，下拉菜单可选不同的方向

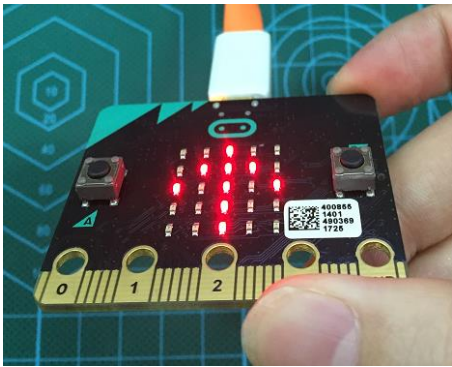
动手实践

发射端

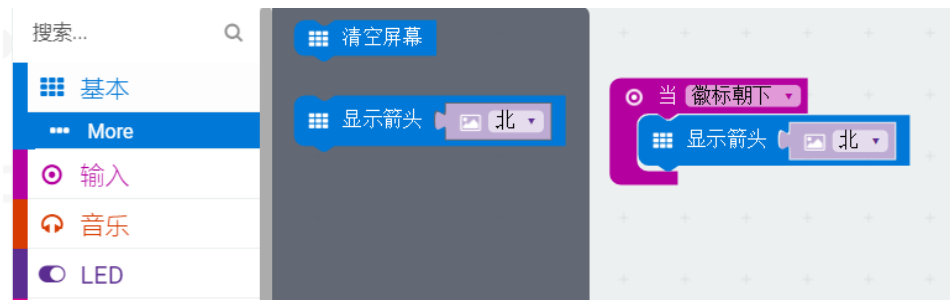
- (1) 新建项目——体感赛车发射端
- (2) 对发射端进行初始设置



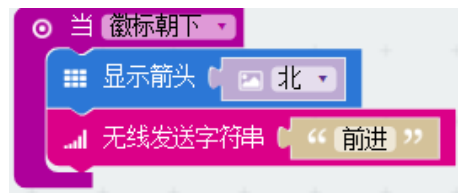
- (3) 当你转动 Micro:bit，使得 LOGO 朝下时，对应控制 Max:bot 前进。



当检测到 Micro:bit 的徽标朝下时，并且用箭头，显示出来当前的状态。

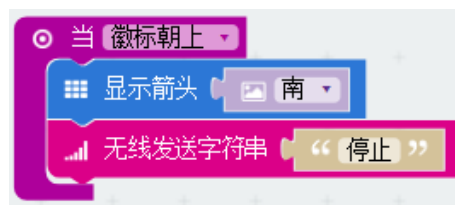


发出控制另外一个 Max:bot 向前的指令。

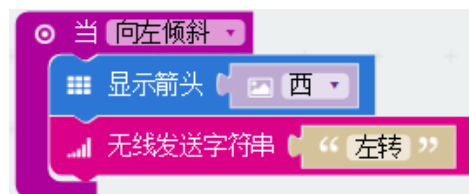


(4) 分别编辑其他三个控制状态。

停止：



向左转：



向右转：



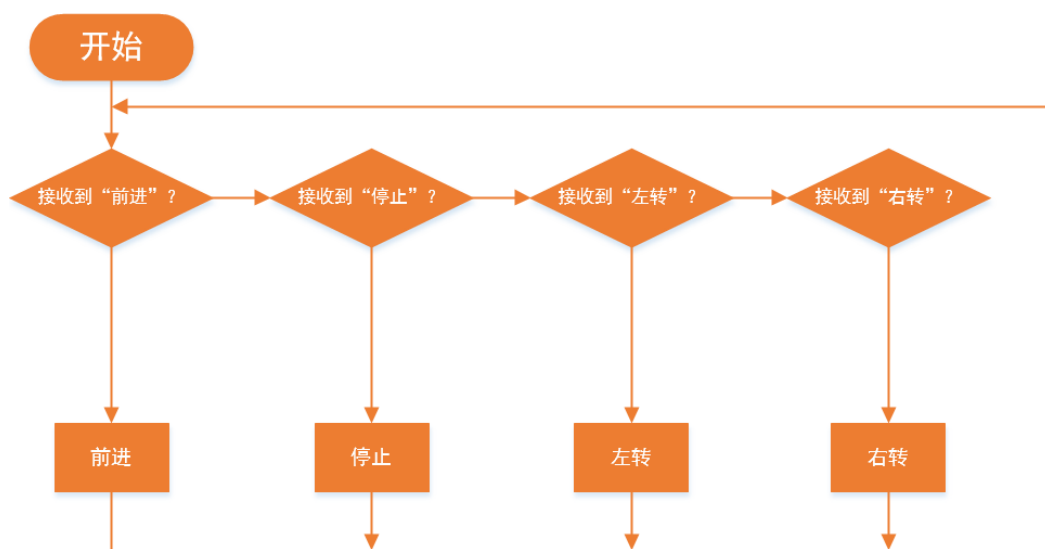
(5) 将整个程序，下载到发射端。



接收端

- (1) 新建项目——体感赛车接收端
- (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) 将程序下载到 Max:bot，来玩个赛车游戏吧。



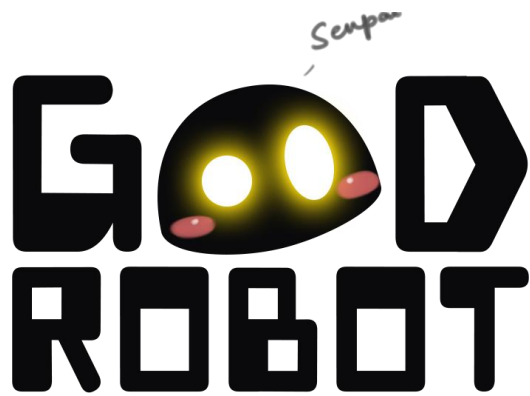
探究思考

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？画出你的制作流程图



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	用法说明
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

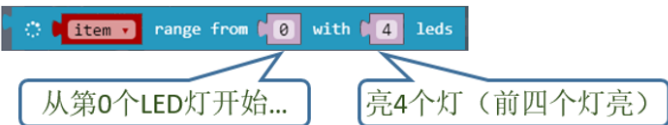
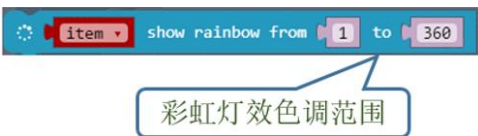
7.1 炫彩灯带

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！

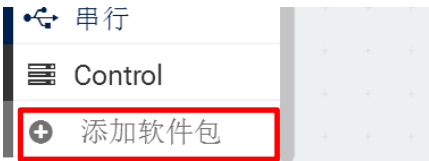
知识要点

积木	用法说明	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颜色模块
灯带点亮个数控制		设定该模式下可以控制的灯珠的范围
灯带色调范围		设定灯珠颜色范围

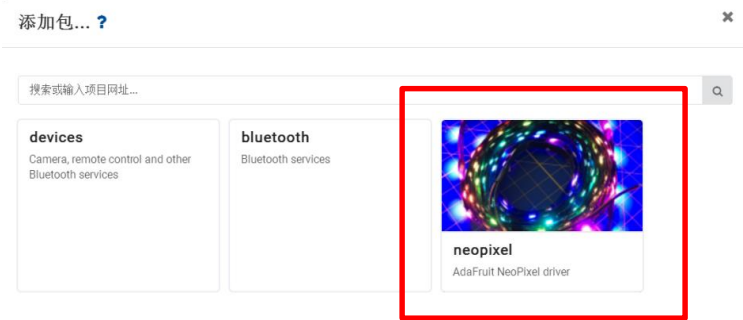
动手实践

(1) 添加灯带编程需要的 Neopixel 指令

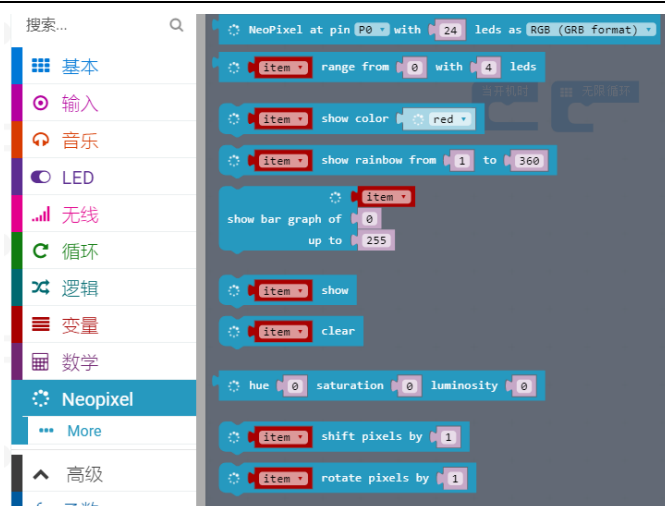
Neopixel 指令的调用需要在“进阶区”中，单击“添加软件包”，出现“添加包”的菜单栏。



单击 neopixel 会增加 neopixel 模块。

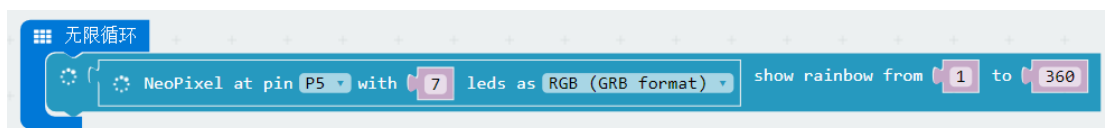


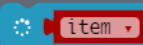
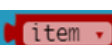
这时，指令区就有了可以随时调用的 neopixel 指令（该教程撰写时，灯带指令未汉化）。



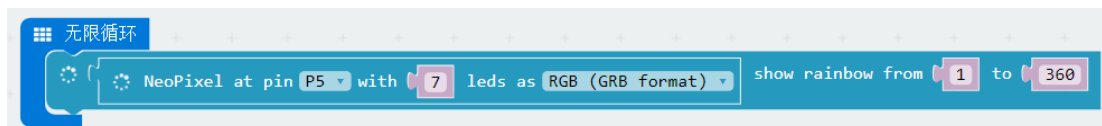
(2) 设置彩虹灯效

灯带连接引脚为 P5（左侧灯带），包含 7 个 LED 小灯，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。



其中， show rainbow from 1 to 360 模块中的  被灯带控制模块代替。

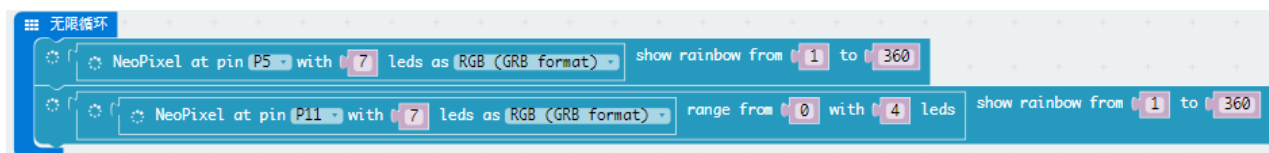
(3) 下载程序到 max:bot。



OK，灯带被点亮了，能够实现七彩灯带的效果啦！

探究思考

将以下程序上传到 Max:bot



左（P5）右（P11）两侧的灯带效果有什么区别呢？你得到了什么结论呢？分享给小伙伴吧。

-
-
-

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

积木	用法说明	功能说明
播放旋律	选择库中的旋律 播放方式、次数 播放旋律 dadadum 重复 once	调用 MakeCode 的自带的音乐
音调模块	音调 节拍 播放音调 中 C 持续 1 节拍	按照歌谱制作歌曲

动手实践

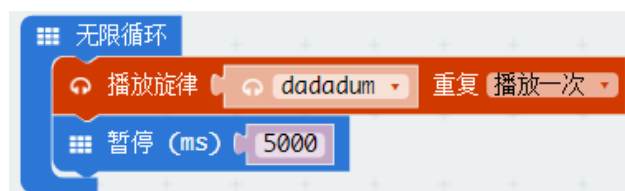
- (1) 新建项目——音乐播放
- (2) 将音乐模块集合中的播放旋律模块，拖动到无限循环中。



- (3) 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”重复一次。



- (4) 需要给旋律 “dadadum” 播放设定播放时间，用到延时模块。如果不添加延时模块，那么旋律的播放没有足够的时间，不会进行播放。



- (5) 将程序上传到 Max:bot

能够听到播放旋律 “dadadum”。

探究思考

你想不想，自己制作想听的旋律呢？Max:bot 也能帮你实现哦。



其中音调与简谱的对应关系如下图：

Middle C	Middle D	Middle E	Middle F	Middle G	Middle A	Middle 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。

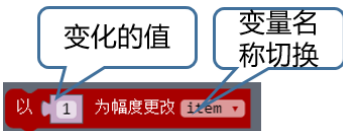
7.3 声光互动

每个圣诞来临的时候，能够看到很多漂亮的圣诞树。



下面制作一个圣诞机器人吧，可以一遍闪耀着灯光，一遍播放着圣诞歌曲。

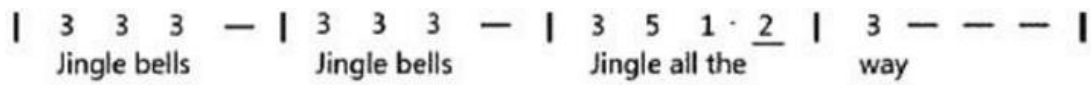
知识要点

积木	用法说明	功能说明
重复模块	 次数	在这个模块中的动作会重复执行
变量变化步长	 变化的值 变量名称切换	设定每次该变量变化的大小： Item → item+1 → (item+1)+1

条件判断		判断“判断条件”是否满足，如果满足则执行“执行”后面的动作。
灯带效果清除		关掉灯带的效果
灯带效果显示		显示灯带的效果

动手实践

- (1) 新建项目——声光互动
- (2) 编辑一段圣诞的音乐“Jingle Bell”，运用音调模块。

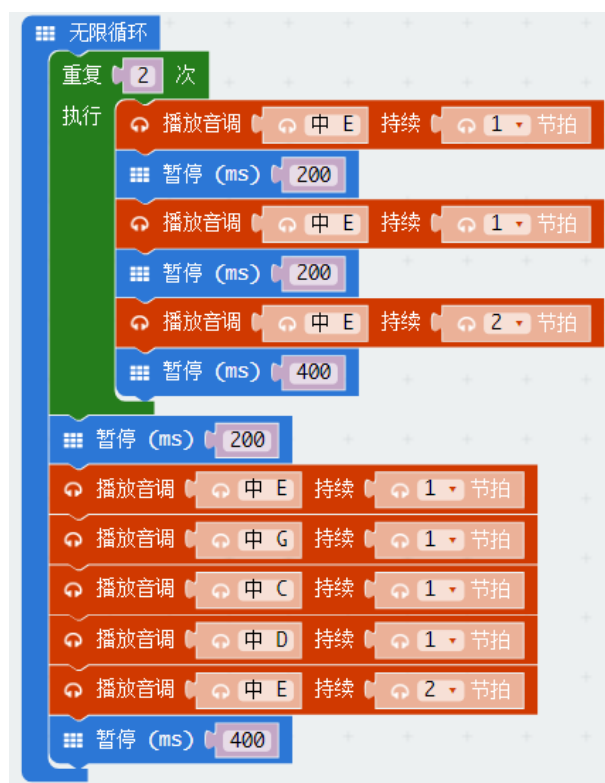


看到简谱上，前面两个短句是相同的音调和节拍。那么可以使用重复模块



在编辑的时候，注意节拍和加入延时模块，旋律听起来更好听。

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。



所以灯亮起的个数，是个变化的量，需要用到变量来控制。

创建一个变量“灯珠个数”



New variable name:

灯珠个数

确定

取消

重新拖入一个循环模块

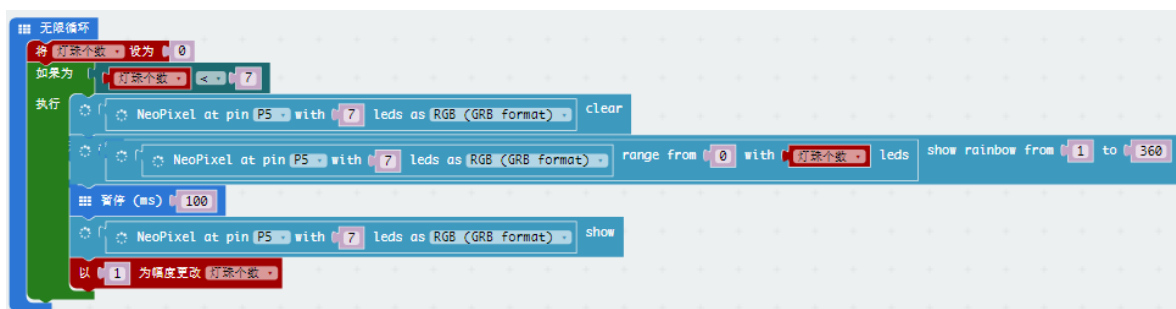


将创建好的变量，放到控制灯带点亮个数控制模块的位置，如下图：



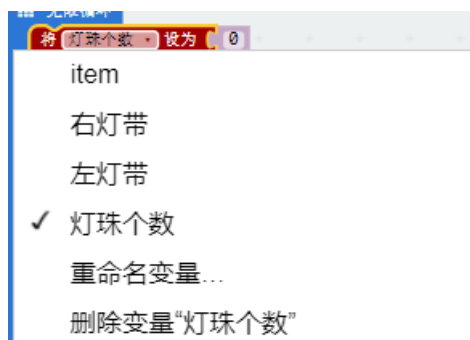
那么，左侧灯带的灯珠的个数的变化，就需要控制变量“灯珠个数”来控制了。

因为我们的灯带只有 7 颗灯珠，需要限制变量“灯珠个数”小于 7，并且初始值为 0



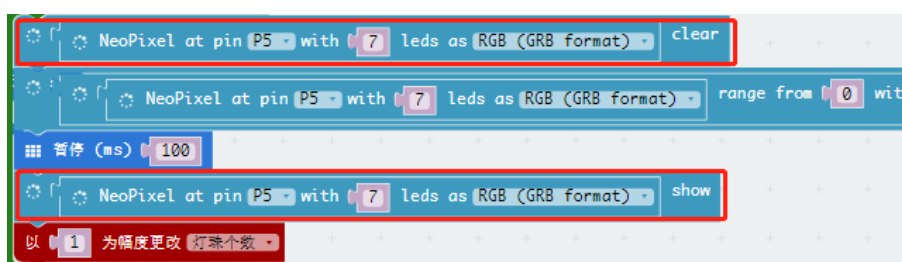
需要注意以下几点：

a. 变量的切换，在下拉菜单中切换



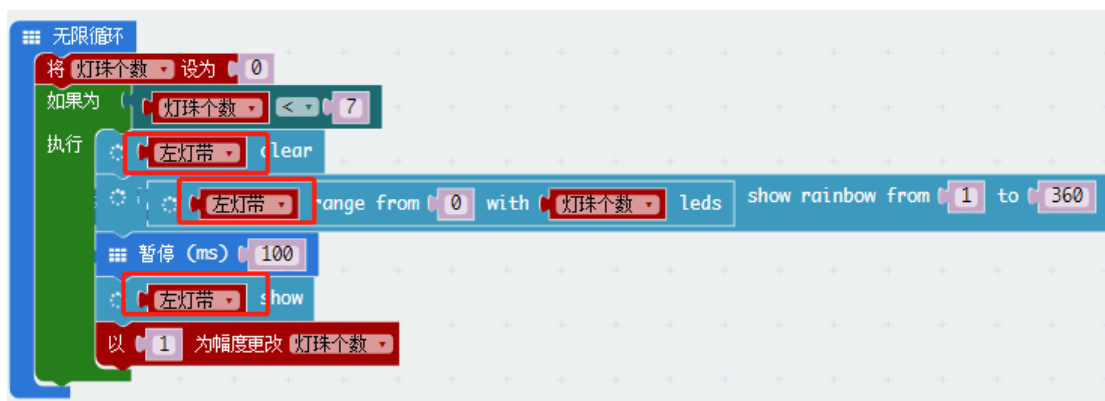
b. 灯带的引脚，要选择对应的引脚 P5（左侧）或 P11（右侧）

c. 灯带效果清除模块与灯带效果显示模块需要搭配使用才能展现出流水灯的效果

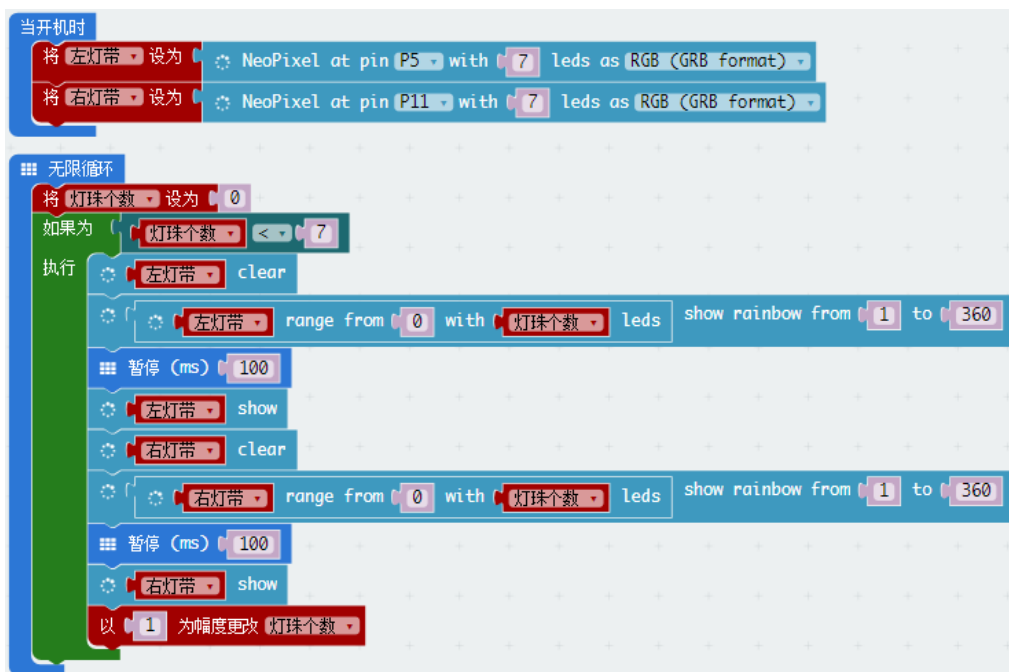


(4) 变量“左灯带”的运用

在灯带控制程序中，我们发现重复使用了灯带控制模块 3 次，那我们可以使用变量来代替，这样程序就变得清晰明了。

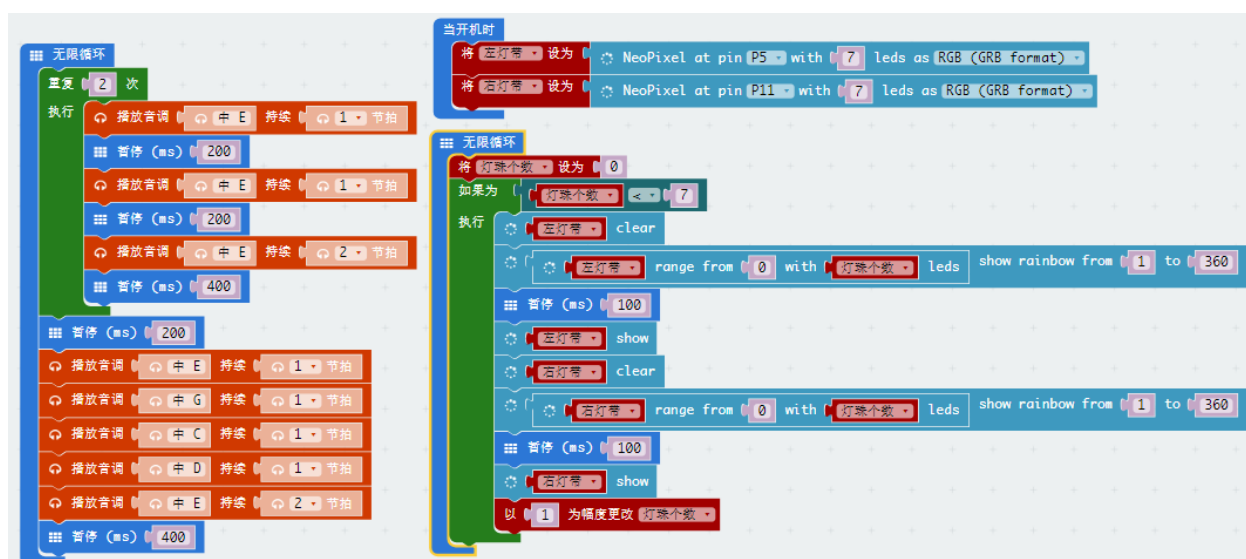


那么，同时控制两条灯带的程序就简单了。



(5) 下载程序到 max:bot

将完整程序下载到 Max:bot，看看效果吧。



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