

Gravity: Mind+趣味学习 套件教程(micro:bit)



DFROBOT
DRIVE THE FUTURE

目录

第一章：小试牛刀	7
项目一：电子蜡烛	7
1. 麦克的故事	7
2. 麦克的布兜	7
3. 小麦的愿望：声音强度检测	8
4. 小克的黑板	8
5. 小麦的愿望：吹气“蜡烛”	10
6. 小克的黑板	10
7. 麦克创造屋	12
项目二：自动门	13
1. 麦克的故事	13
2. 麦克的布兜	13
3. 小麦的愿望：按钮控制舵机	14
4. 小克的黑板	14
5. 小麦的愿望：运动传感器控制舵机	17
6. 小克的黑板	17
7. 麦克创造屋	19
项目三：音乐盒	20
1. 麦克的故事	20
2. 麦克的布兜	20
3. 小麦的愿望：播放音乐	21
4. 小克的黑板	21
5. 小麦的愿望：身体感应音乐	23
6. 小克的黑板	23
7. 麦克创造屋	25
项目四：炫彩灯带	26

1. 麦克的故事.....	26
2. 麦克的布兜.....	26
3. 小麦的愿望：点亮彩虹灯带.....	27
4. 小克的黑板.....	27
5. 小麦的愿望：逐一点亮彩虹灯.....	29
6. 小克的黑板.....	30
7. 小麦的愿望：声音控制彩虹灯带.....	33
8. 小克的黑板.....	33
9. 麦克创造屋.....	36
第二章：如虎添翼.....	37
项目一：自平衡仪.....	37
1. 麦克的故事.....	37
2. 麦克的布兜.....	37
3. 小麦的愿望.....	38
4. 小克的黑板.....	40
5. 麦克发明室.....	43
项目二：DJ 演奏台.....	44
1. 麦克的故事.....	44
2. 麦克的布兜.....	44
3. 小麦的愿望.....	45
4. 小克的黑板.....	46
5. 麦克发明室.....	49
项目三：可移动门铃.....	50
1. 麦克的故事.....	50
2. 麦克的布兜.....	50
3. 小麦的愿望.....	51
4. 小克的黑板.....	51
5. 麦克发明室.....	55

项目四：拆弹游戏	56
1. 麦克的故事	56
2. 麦克的布兜	56
3. 小麦的愿望	57
4. 小克的黑板	57
5. 麦克发明室	65
第三章：大展拳脚	66
项目一：投篮机	66
1. 麦克的故事	66
2. 麦克的布兜	66
3. 小麦的愿望：超声波判断篮球是否投中	67
4. 小克的黑板	67
5. 小麦的愿望：按下按钮，开始或结束游戏	69
6. 小克的黑板	69
7. 麦克小发明	71
项目二 火线冲击	72
1. 麦克的故事	72
2. 麦克的布兜	72
3. 小麦的愿望：判断是否导通	73
4. 小克的黑板	73
5. 小麦的愿望：按下按钮，开始或结束游戏	74
6. 小克的黑板	74
7. 小麦的愿望：碰触到火线时，发出声光警报	75
8. 小克的黑板	76
9. 麦克小发明	78
项目三 感应垃圾桶	79
1. 麦克的故事	79
2. 麦克的布兜	80

3. 小麦的愿望：舵机转动，带动垃圾桶盖子打开或关闭·····	80
4. 小克的黑板·····	80
5. 小麦的愿望：超声波传感器感应·····	81
6. 小克的黑板·····	82
7. 小麦的愿望：按钮控制垃圾桶盖常开·····	83
8. 小克的黑板·····	83
9. 麦克小发明·····	86
项目四 答案之书·····	87
1. 麦克的故事·····	87
2. 麦克的布兜·····	88
3. 小麦的愿望：按下按钮，显示答案·····	88
4. 小克的黑板·····	88
5. 小麦的愿望：神奇的答案之书·····	90
6. 小克的黑板·····	90
7. 麦克小发明·····	92
第四章 各显神通·····	93
项目一 坦克大战·····	93
1. 麦克的故事·····	93
2. 麦克的布兜·····	93
3. 小麦的愿望：左右移动的“坦克”·····	94
4. 小克的黑板·····	94
5. 小麦的愿望：按下按钮，发射炸弹·····	96
6. 小克的黑板·····	96
7. 小麦的愿望：飞机从舞台上落方落下，攻击坦克；被炸弹击中则爆炸·····	99
8. 小克的黑板·····	99
9. 麦克小发明·····	101
项目二 AI 助力垃圾分类·····	102
1. 麦克的故事·····	102

2. 麦克的布兜.....	103
3. 小麦的愿望：语音识别.....	103
4. 小克的黑板.....	103
5. 小麦的愿望：垃圾分类语音助手.....	106
6. 小克的黑板.....	106
7. 麦克小发明.....	109
项目三 风帆车.....	110
1. 麦克的故事.....	110
2. 麦克的布兜.....	110
3. 小麦的愿望：制作小车.....	111
4. 小克的黑板.....	111
5. 小麦的愿望：制作风帆车.....	112
6. 小克的黑板.....	112
7. 小麦的愿望：制作风扇.....	112
8. 小克的黑板.....	113
9. 小麦的愿望：“循规蹈矩”的风帆车.....	114
10. 小克的黑板.....	114
11. 麦克小发明.....	117
项目四 无人漫游车.....	118
1. 麦克的故事.....	118
2. 麦克的布兜.....	118
3. 小麦的愿望：雷达扫描.....	119
4. 小克的黑板.....	119
5. 小麦的愿望：“智能”规划路线.....	121
6. 小克的黑板.....	121
7. 麦克小发明.....	125

第一章：小试牛刀

凭着对 micro:bit 模糊的了解，你是不是跃跃欲试了呢？现在，我们就开始 micro:bit 的神奇之旅吧！

项目一：电子蜡烛

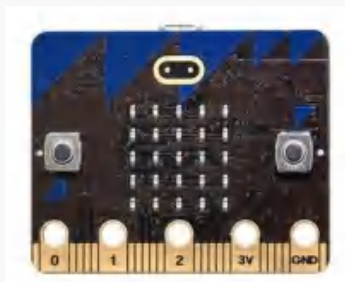
1. 麦克的故事

看完今天的新闻，发现温室效应与我们每个人密切相关，我们的任何活动都可能造成碳排放，比如烧火做饭之类的。

而我的好朋友就要过生日了，为了不污染环境，我想给她制作一个电子蜡烛，我们一起来学习吧！

先来看看要用到哪些硬件吧！

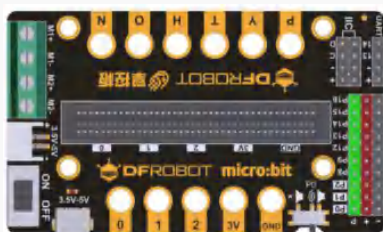
2. 麦克的布兜



1× micro:bit 主控板



1× micro USB 连接线



1× micro:bit 扩展板



1× LED 灯模块



1× 声音传感器模块



第一次看到声音传感器模块，快来了解一下吧：

声音传感器模块能够感知外界声音的强弱，经过编写代码后可以实现声音强弱控制 LED 小灯的亮灭哦。

3. 小麦的愿望：声音强度检测



我想知道自己的声音多大，比如如果 LED 板载小灯上面写 1，就表示声音小；写 2，就表示声音大~



如何通过学过的知识进行设计呢？

我们可以设置当声音强度模拟值小于 80 的时候，板载小灯显示数字“1”，表示声音较小，强度为 1 级；

否则，声音强度模拟值大于 80 的时候，板载小灯显示数字“2”，表示声音较大，强度为 2 级。

4. 小克的黑板

STEP1: 首先要把声音传感器模块接到扩展板 P0 号接口上；把 LED 灯模块接到扩展板 P1 号接口上。

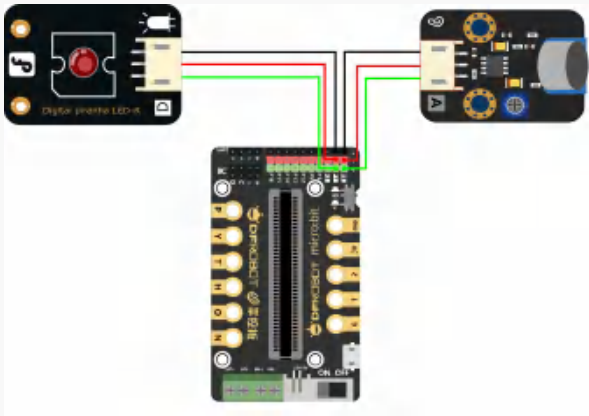


图 1.1.1 接线示意图

STEP2: 编写程序

①根据上一个小贴士的提示，我们可以进行初步的设计啦：单击指令区最下面的“micro: bit”，将“控制”代码模块中的条件语句“如果--否则”拖动至脚本区，将“运算符”中的“<”拖动至脚本区。



我们顺便来学习一下“如果--否则”语句吧：

条件语句代码“如果--否则”的功能：如果满足右边逻辑判断语句。则执行在“那么执行”下方的代码语句，反之，如果不满足逻辑判断语句，则执行的是在“否则执行”下方的代码。



图 1.1.2 “如果--否则”指令

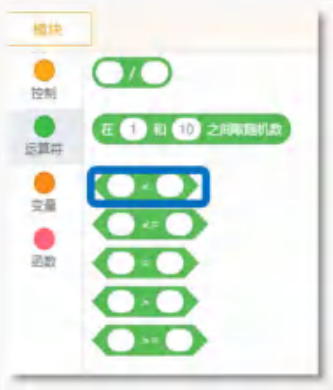


图 1.1.3 “<”指令

②通过“读取模拟引脚 P0”连接的声音传感器的模拟信号数值（模拟数值范围在 0—1023 之间哦）来检测声音的强度。

③设定“声音强度探测”的实现程序。逻辑指令中的“<”可以判断声音强度的模拟值是否达到设定的标准。

如图 1.1.4，读取模拟引脚 P0 的数值“<80”成立的时候，也就是声音强度为 1 级的时候，板载小灯显示数字“1”；

否则，模拟读取引脚 P0 的数值“≥80”时，也就是声音强度为 2 级，板载小灯显示数字“2”。在“否则”代码之后，可以增加“等待 1 秒”，防止数字闪烁。



图 1.1.4 执行程序

5. 小麦的愿望：吹气“蜡烛”



我想要不吹气的时候小灯一直亮，
当我用力吹小灯，它就会灭，就像
蜡烛一样

6. 小克的黑板

编程过程中，通过设置连接声音传感器的引脚值来控制小灯的亮暗哦。

①使用“设置数字引脚 P0 输出”这一模块，由于当前小灯连接的是 P1 引脚，将 P0 修改为 P1，拖至脚本区。



图 1.1.5 “设置数字引脚”指令

②设定“电子蜡烛”的实现程序。

当声音传感器模拟值小于设定值 80 的时候，小灯保持亮起的状态，反之，轻轻一吹，声音强度大于 80 时，小灯将保持 2 秒的熄灭状态。

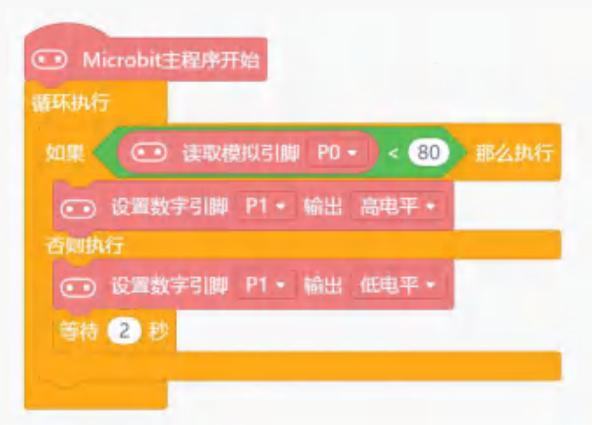


图 1.1.6 执行程序

7. 麦克创造屋

以后也可以提倡其他小伙伴制作这样既环保又有趣的生日蜡烛哦~除了做生日蜡烛，你还能用这节课布兜里面的东西做出其他好玩的东西吗？
快把你的好想法设计出来吧~



我的设计	我的程序
	

项目二：自动门

1. 麦克的故事

我发现咱们的生活中还有好多有趣的发明呀,上次我跟同学去图书馆借学习机器人的书,走到门口的时候发现不用自己开门,门自动开了,这是什么神奇的装置呢?

今天我们就来制作个可以自动开门的装置吧!

先来看看要用到哪些硬件吧!

2. 麦克的布兜



1× micro:bit 主控板



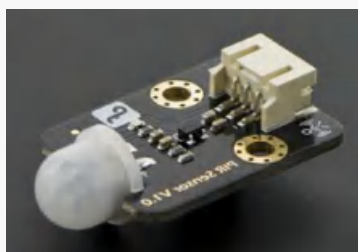
1× micro USB 连接线



1× micro:bit 扩展板



1× 按钮模块



1× 运动传感器模块



1× 舵机 (伺服机构)



第一次看到运动传感器模块和舵机模块，快来认识一下吧：

舵机：它能旋转到特定角度，根据型号不同，旋转范围有 0 到 180 度，0 到 360 度。它的特点是转速相对电机小，力气更大。一般用于机械臂完成更灵活更高难度的动作，也可以让机械装置更加精确地运动哦~

3. 小麦的愿望：按钮控制舵机

*由于舵机本身会存在死区问题，所以编程时建议使用 10-170 度，本项目中使用 10°。



我想要用按钮控制舵机，如果按下按钮，舵机转动 100°，不按按钮，舵机转 10°。

4. 小克的黑板

STEP1: 把按钮模块接到扩展板 P0 号接口上；把舵机接到扩展板 P1 号接口上。

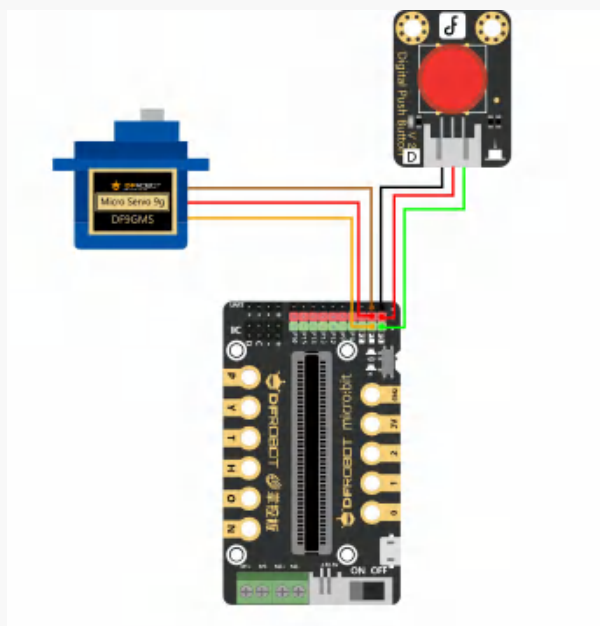


图 1.2.1 接线示意图



在编写程序之前需要增加控制器模块哦：

如果我们想要控制舵机的运行，那就需要增加一个控制舵机的指令模块，怎么调用呢？

如图 1.2.2 和 1.2.3，在左下角的“扩展”中找到“执行器”，选择“舵机模块”，点击舵机模块后再返回到编程界面即可。



图 1.2.2 “扩展” 模块图



图 1.2.3 执行器-舵机模块

STEP2: 编写程序

①点开执行器的模块可以控制舵机，通过设置 P0 号引脚可以控制舵机的值。



图 1.2.4 设置舵机值

②设置如果按钮按下，舵机转动 100°，LED 小灯显示 “”；否则舵机旋转至 10°，LED 小灯显示 “”。需要用到的指令为：“如果--否则”。

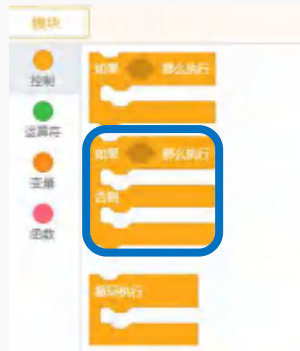


图 1.2.5 “如果-否则” 指令

③最终程序：



图 1.2.6 执行程序

实验效果：

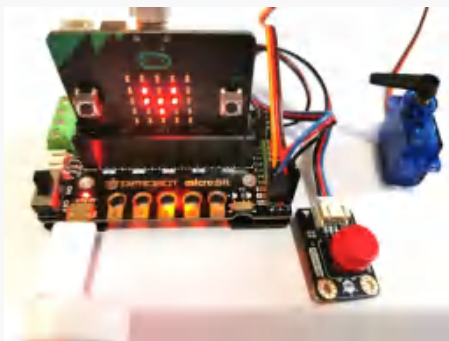
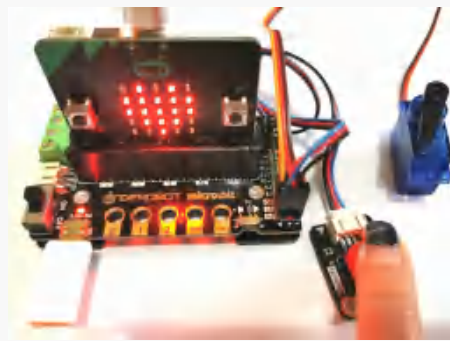


图 1.2.7 实验效果-未按按钮图



1.2.8 实验效果-按下按钮

5. 小麦的愿望：运动传感器控制舵机



我想要自动门的效果，只要有人经过，它（舵机）就转动 100°，否则就转 10°。

6. 小克的黑板

STEP1: 将舵机连接至 P1 引脚；将运动传感器连接至 P0 引脚。

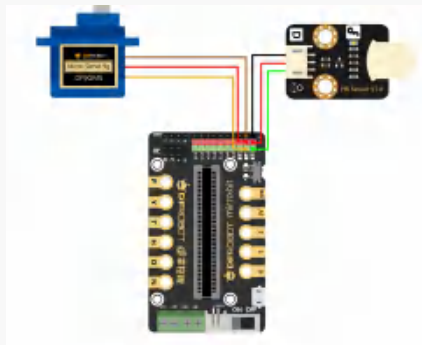


图 1.2.9 连线示意图

STEP2: 编写程序

① 通过读取数字引脚 P0 的值来判断运动传感器的数字信号数值。



图 1.2.10 “读取数字引脚” 指令

②当运动传感器检测到有人经过时，实现舵机转动 100°；没有人经过时，舵机旋转 10°。需要用到的指令为：“如果—否则”。

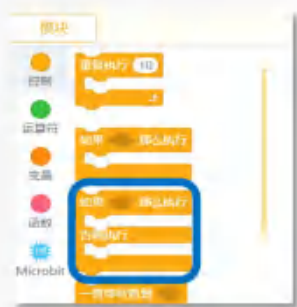


图 1.2.11 “如果-否则” 指令

③最终程序：



图 1.2.12 执行程序

实验效果：

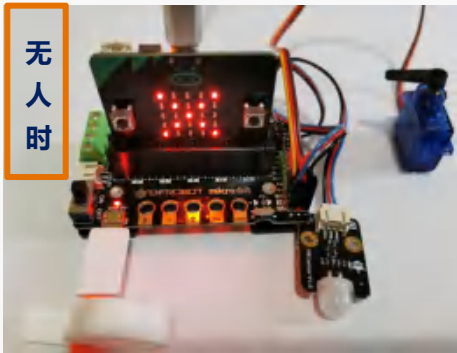


图 1.2.13 实验效果-无人经过

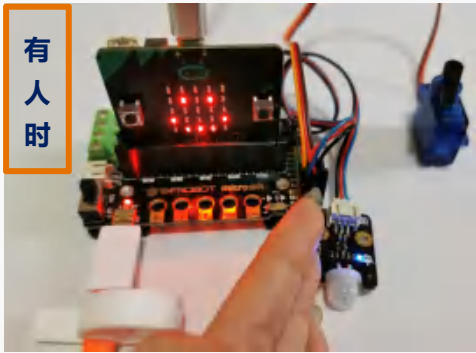


图 1.2.14 实验效果-有人经过

7. 麦克创造屋

你可以用这些有趣的硬件做很多好玩的东西哦！



我的设计	我的程序
	

项目三：音乐盒

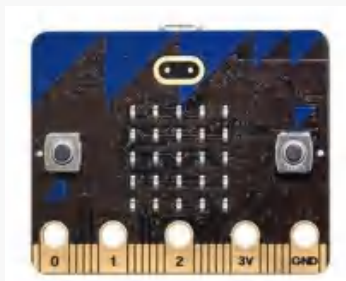
1. 麦克的故事

前几天音乐老师给我们演示了一个音乐盒，非常的漂亮，而且里面的旋律很好听，我也好想有一个自己的音乐盒呀！那我们来看看能不能用 micro:bit 做出好玩的音乐盒！



先来看看要用到哪些硬件吧！

2. 麦克的布兜



1× micro:bit 主控板



1× micro USB 连接线



1× micro:bit 扩展板



1× 运动传感器模块

3. 小麦的愿望：播放音乐



我的愿望很简单，想要它播放出美妙的音乐！

4. 小克的黑板

STEP1: 用数据线将 micro:bit 主板与电脑连接起来，将 micro:bit 扩展板中的蜂鸣器开关打开；将运动传感器模块接在扩展板的 P1 号接口。



注意：播放音频的蜂鸣器占用了 P0 端口，因此播放声音的时候，P0 端口不能再用了哦。



图 1.3.1 蜂鸣器开关打开示意图

STEP2: 编写程序

①将“micro:bit”模块中的播放旋律模块“**接口 P0 播放声音--直到结束**”，拖动到脚本区。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“**DADADADUM**”。



图 1.3.2 “播放声音” 直到结束指令

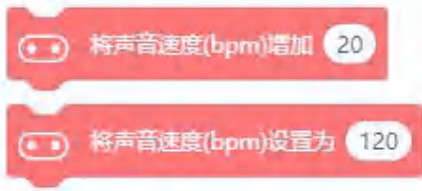
②最终程序。



图 1.3.3 执行程序



注意：在程序中可添加对应的声音速度程序，可以调节 micro:bit 扩展板播放音乐的速度哦。



5. 小麦的愿望：身体感应音乐



我想让它播放自己编的音乐！
要是当我靠近时播放，没有人靠近时不播放就更好了！

6. 小克的黑板

① 编写一段歌曲《小星星》的音乐。



先看看《小星星》的简谱是怎样的吧！音调不同，音符不同，需要的发声频率也不同哦！

小星星

调号 1=C 1 1 5 5 6 6 5 — 4 4 3 3 2 2 1 音符

一闪一闪亮晶晶，满天都是小星星。

怎么用 Mind+ 把简谱呈现出来以便播放好听的音乐呢？

首先要学会把简谱中的音符和字母对应起来哦，见下表：

音符	1	2	3	4	5	6	7
	(do)	(re)	(mi)	(fa)	(sol)	(la)	(si)
字母	C	D	E	F	G	A	B

② 将“micro:bit”模块集中的播放旋律模块 **“接口 P0 播放音符 1 中 C/C4 拍”**

拖至脚本区，这个模块可以通过调节参数 **“C4”** 演奏不同的音符，可以通过调节参数 **“1/2”** 调整节拍。（C4 是指在中音状态下演奏，其中 4 表示音符的音高，也可以换成 D4、G4、A4，变为中音演奏）



图 1.3.5 “播放音符” 指令



图 1.3.6 编写音符指令

③ 运动传感器连接的是 P1 引脚，如果有人经过就相当于 P1 引脚的值为 1，没有人则为 0，因此添加一个已经学过的条件判断语句“**如果-那么执行**”就可以实现智能音乐盒啦！



图 1.3.7 执行程序

7. 麦克创造屋

小麦同学，你想设计一个属于自己的音乐盒吗？
开动脑筋创造起来吧~

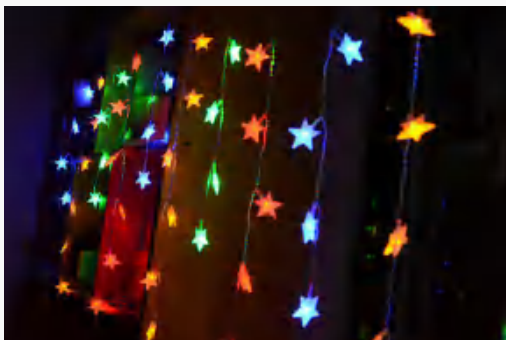


我的设计	我的程序
	

项目四：炫彩灯带

1. 麦克的故事

想给音乐盒做一个好看的外壳，想要它有炫彩的灯光，这样看起来就更梦幻了！



先来看看要用到哪些硬件吧！

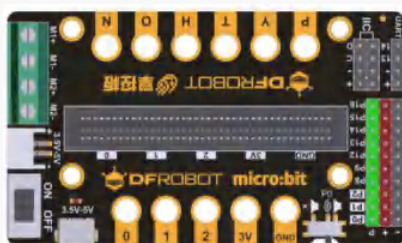
2. 麦克的布兜



1× micro:bit 主控板



1× micro USB 连接线



1× micro:bit 扩展板



1× 声音传感器模块



1× 彩虹灯带



第一次看到彩虹灯带，快来认识一下吧：

彩虹灯带可以通过调节色彩参数调出不同颜色的灯效哦！

3. 小麦的愿望：点亮彩虹灯带



我想点亮全部的彩虹灯，让它发出七彩的光！

4. 小克的黑板

STEP1: 将彩虹灯带接在扩展板的 P0 号接口。

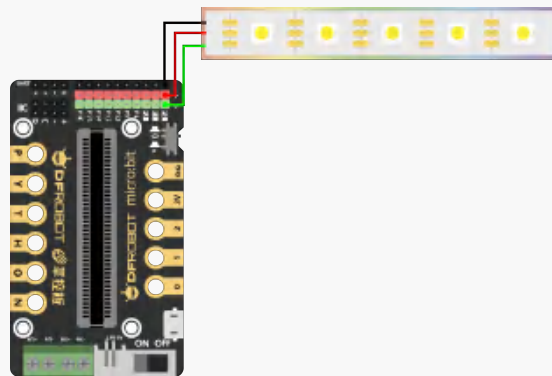


图 1.4.1 接线示意图

STEP2: 增加扩展灯带的指令库。使用灯带需要调用“扩展”功能中的“显示器”模块，选中“Ws2812 RGB 灯”。



图 1.4.2 “扩展”指令



图 1.4.3 添加 “Ws2812 RGB 灯” 显示器

STEP3: 编写程序

①灯带常用指令：在指令区左侧点击 **“显示器”** 模块，所有灯带指令就会显示出来啦。

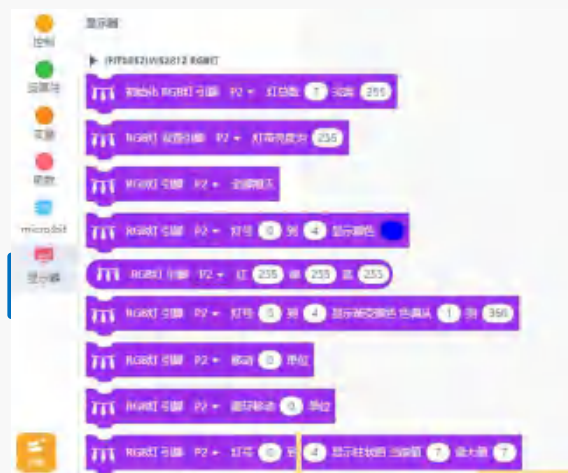


图 1.4.4 “扩展” 指令



如何使用常用的彩虹灯带指令呢？

在使用彩虹灯带的过程中,首先需要选定灯带连接 micro:bit 的引脚、灯带 LED 灯的数目及颜色模式,需要调用的指令如下:

①如果你要确定 LED 灯使用个数和亮度,需要调用的指令是:



(其中亮度最高是 255, 灯总数是实际灯带的灯珠总个数哦)

②如果你想确定彩虹灯的色调范围,需要调用的指令是:



(灯号 0 代表第一个灯, 色调表示 360 种不同的颜色)

② 灯带连接引脚为 P0，包含 7 个 LED 小灯，设置彩虹灯效色彩范围 1-360（红色--绿色--蓝色），彩灯就会呈现七彩效果啦。

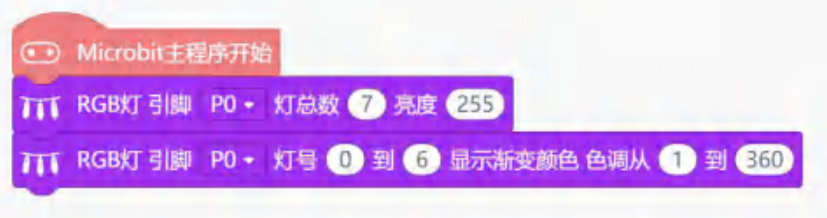


图 1.4.5 执行程序

实验效果：

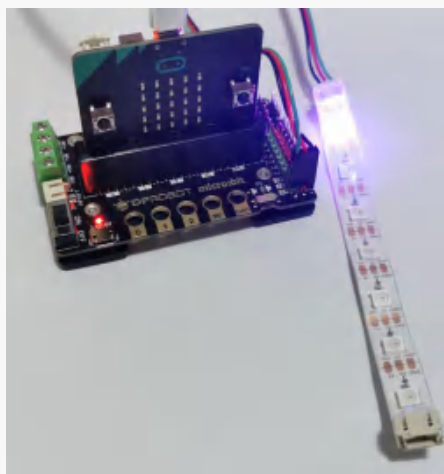


图 1.4.6 实验效果

5. 小麦的愿望：逐一点亮彩虹灯



可不可以每隔一秒钟就点亮一盏灯呢？

6. 小克的黑板

① 如果想要控制彩虹灯带的 LED 小灯每隔一秒点亮一盏, 那么就需要运用一个新的东西: **变量**。

需要调用指令的流程为: “**变量**----**新建数字类型变量**”, 任意设定变量名 (如: 点亮小灯数), 再单击确定。



图 1.4.7 新建变量图



图 1.4.8 设置变量名称



初次新建变量, 来认识认识吧?

新建变量之后, 在变量指令区会出现常用的 “**变量+变量名称的模块**” 和 “**设置变量的值**” 的模块:



当你需要使用该变量的时候, 直接拖动模块到脚本区就可以进行编辑啦!

②要对小灯的引脚和总灯数进行初始化，并设置亮度。“亮灯数量”（也可以简称为 α ）最初设定为 0（因为最初没有小灯被点亮哦）。

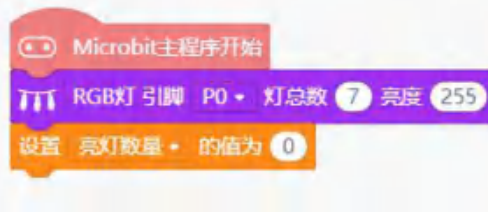


图 1.4.9 初始化引脚和亮灯数

③当“亮灯数量”（ α ）小于等于 6 的时候，需要点亮 α 盏灯，等待 1 秒之后让 α 加 1，如果 α 超过 6 的话就要将灯设置为全部熄灭，再重新开始，逻辑图如图 1.4.10。

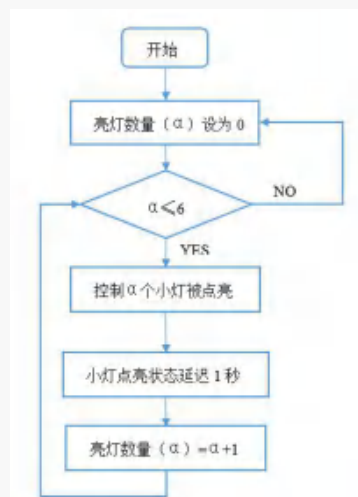


图 1.4.10 逻辑图

④按照流程进行程序的编写，首先设置条件，需要加入“如果-那么执行”指令和“循环执行”指令。



图 1.4.11 部分指令

⑤其次向里面添加条件：如果：“亮灯数量” (α) 小于等于 6，那么执行：点亮 α 盏灯，等待 1 秒之后让 α 加 1，也就是再多亮起一盏灯。



图 1.4.12 部分指令

⑥直到亮灯数量大于 6，在“否则执行”指令下，将变量“亮灯数量”的值设置成 0，并且让所有小灯熄灭。

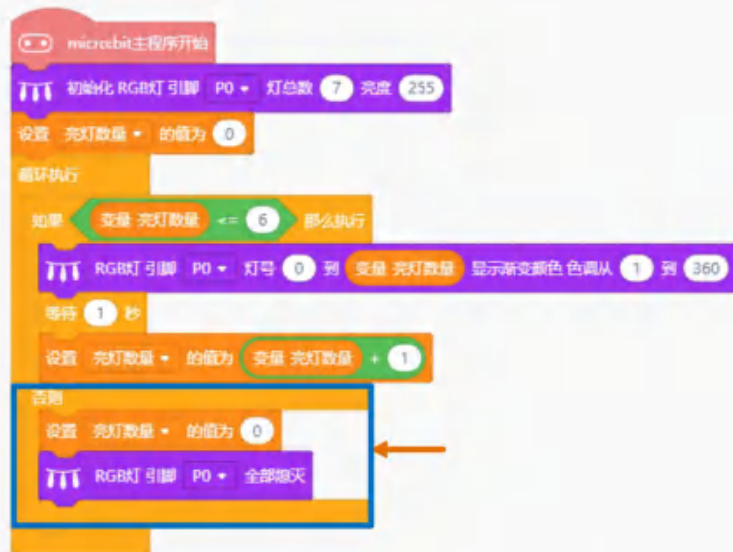


图 1.4.13 部分指令

7. 小麦的愿望：声音控制彩虹灯带



哎？可不可以每次发出声音都能点亮一盏灯，直到七盏小灯全部点亮之后再重新开始玩呢？

8. 小克的黑板

STEP1: 彩灯连接引脚保持不变，将声音传感器连接到 P1 引脚。

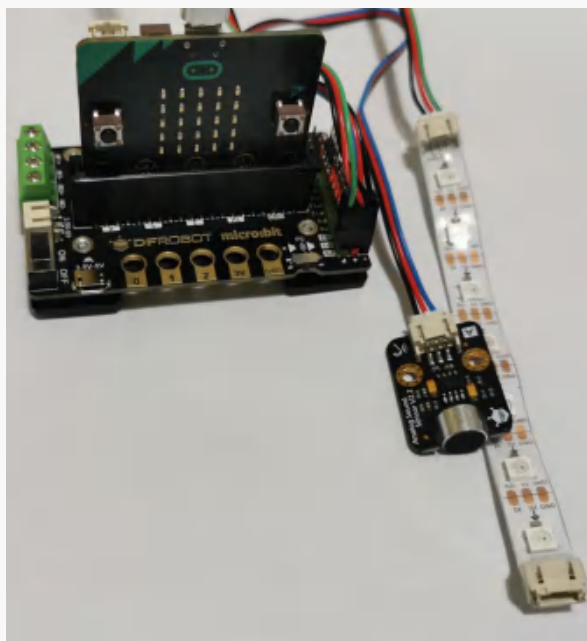


图 1.4.14 连线示意图

STEP2: 单击 “micro:bit” 模块，选择 “读取模拟引脚 P0” 指令，将 P0 改为 P1，并拖到脚本区。

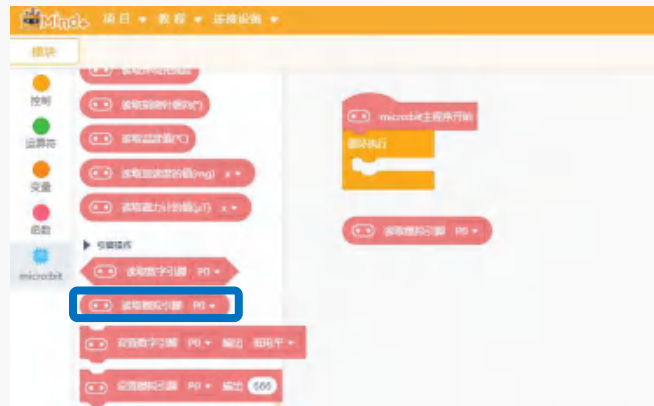


图 1.4.15 “读取模拟引脚” 指令

STEP3: 当声音传感器检测到声音 > 40 且 “亮灯数量 α ” 不超过 6 时，控制灯带点亮 α 盏 LED 灯，之后持续不断检测声音是否超过 40。

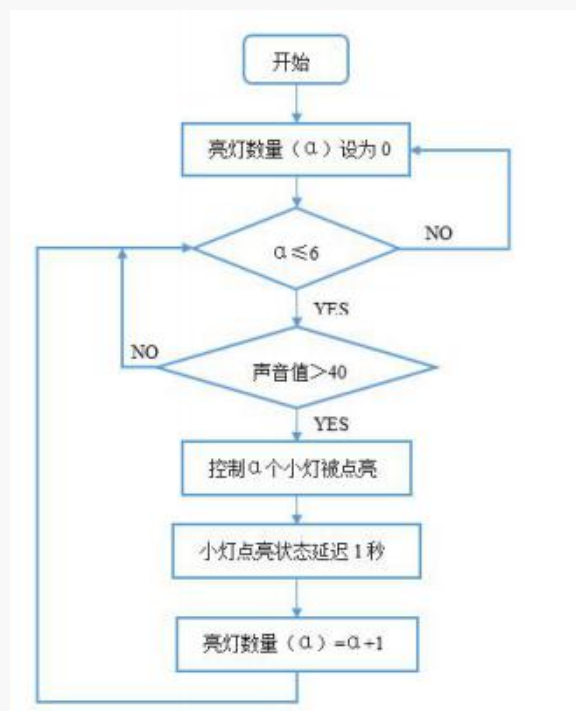


图 1.4.16 逻辑图

STEP4: 编写程序

在上一个愿望的执行程序的基础上，只要再增加一个声音强度的判断条件就可以了哦，也就是说，在“亮灯数量 α ” ≤ 6 时，如果模拟引脚 P1 的值 > 40 ，那么执行点亮 α 盏灯，等待 1 秒之后让 α 加 1，并持续不断检测声音是否超过 40。直到“亮灯数量 α ” > 6 ，灯全部熄灭，重新开始。

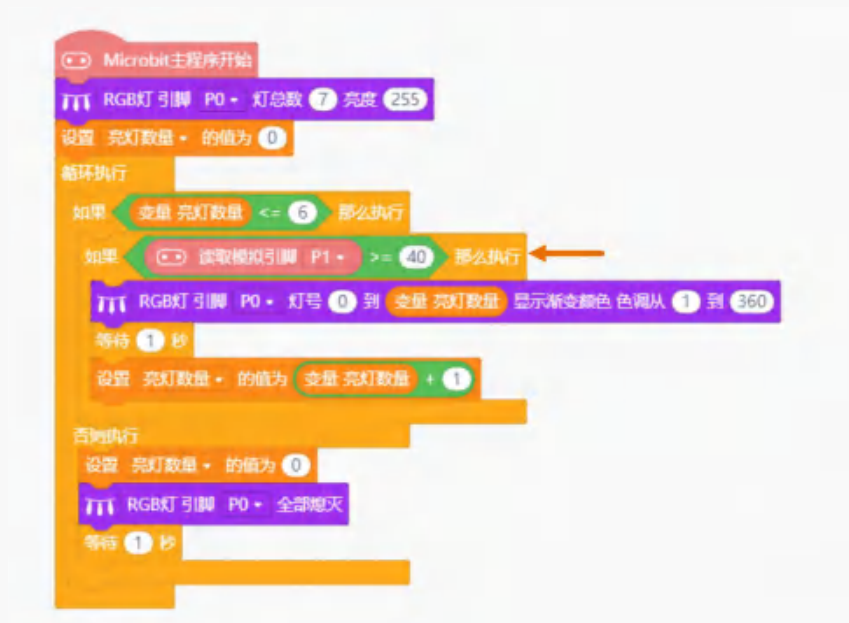


图 1.4.17 执行程序

9. 麦克创造屋



哇！这节课的程序好难呀！看来我要好好的练习一下了，
不如回去做一个好玩的声控灯吧！谁和我一起设计一下呢？

我的设计	我的程序
	

第二章：如虎添翼

除了前面学习的按钮模块、旋钮模块，还有很多常用的、有趣的传感器，使用他们可以制作出更加智能化、好玩和实用的作品，快来认识它们吧！

项目一：自平衡仪

1. 麦克的故事

生活中使用手机时我们会发现，手机横放在手里，但是当我手向左转的时候，屏幕里的页面就神奇的向右转动，这是为什么呢？其实原因是因为手机里有自平衡装置，为了方便观看，屏幕里的页面要始终保持水平向上，所以根据开发者的设置，它会感应手机的方向，如果你把手机向左旋转，屏幕里的页面就向右转啦！

那我们一起用 micro:bit 来探索这个神奇的功能吧！

先来看看要用到哪些硬件吧！

2. 麦克的布兜



1× micro:bit 主控板



1× micro USB 连接线



1× micro:bit 扩展板



1× 舵机（伺服机构）

3. 小麦的愿望



我想要无论怎么旋转，小舵机上的指针永远垂直于上方~

理解重力加速度有助于理解自平衡仪的原理哦~
快去小贴士学习一下吧~



什么是重力呢？

我们都生活在地球上，而地球是有吸引力的~由于它的吸引而使物体受到的力就是重力。

什么是重力加速度呢？

由重力产生的加速度就叫做重力加速度，也叫自由落体加速度，用 g 表示。

使用重力加速度可以做些什么呢？

micro:bit 自带加速度计，它能实时检测 micro:bit 的姿态，通过重力加速度的感应和程序设计就可以控制舵机指针一直指向正上方啦！



这么看来，我们是要将舵机和板子连在一起，然后通过板子的重力加速度的感应来控制舵机的旋转方向喽？

理解的很对哦！不过重力加速度还有不同的方向呢！去下面的小贴士里学学我们今天要用哪个方向的重力加速度吧！



micro:bit 加速度计：

micro:bit 可以检测 X,Y,Z 三个方向的重力加速度，其中 X 沿着左右方向，Y 沿着前后方向，Z 垂直于板，沿着上下方向，如图 2.1.1。

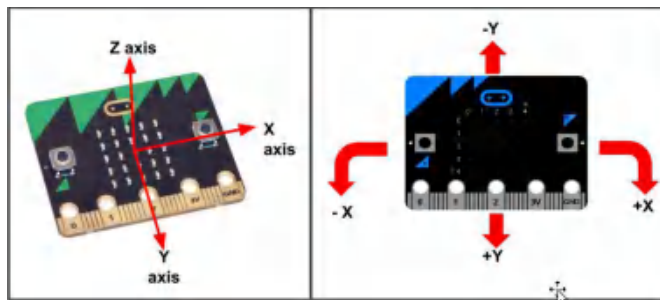


图 2.1.1 三个方向

图 2.1.2 数值变化



当主板平放在桌面上时，micro:bit 只受到竖直方向上的重力（Z 轴），所以 X,Y 都接近于 0。

而当板子产生左右倾斜时，X 轴的数值会发生变化。向左侧倾斜则 X 为负值，反之为正值，如图 2.1.2。

4. 小克的黑板

STEP1: 将舵机与 micro:bit 的 P0 号接口相连接。

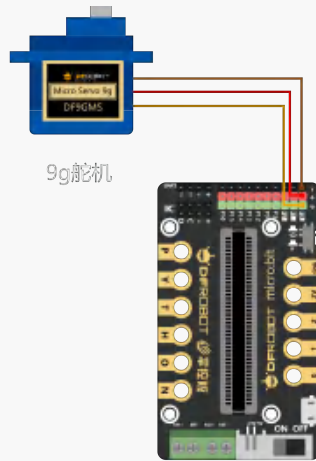


图 2.1.3 连接示意图

STEP2: 编写程序

①通过“读取加速度的值”模块实现获取 X,Y,Z 轴的加速度数值。



可以通过下拉列表选择 X,Y,Z 轴哦~



图 2.1.4 读取加速度的值

②加速度传感器的数值范围在-1023~1023 之间变化，可以通过“显示”模块来查询变化范围哦。



图 2.1.5 显示加速度的值



图 2.1.6 实现效果

③当 micro:bit 板子向左倾斜时，X 方向会产生一个负值，这个时候就需要舵机向右转动来抵消这个角度啦！

那 micro:bit 板子的加速度与舵机旋转的角度是怎么对应的呢？

这里需要用到“映射模块”，将加速度 X 轴的值（-1023~1023），映射到舵机的转动角度（0°~180°）。映射之后，当 micro:bit 板子放在水平桌面时，舵机的角度就为 90 度啦，对应关系如表 2.1.1。

表 2.1.1

舵机角度	0 度	90 度	180 度
micro:bit 加速度	-1023	0	1023

在引脚中找到映射模块



图 2.1.7 映射模块

④在“扩展”中找到“执行器”模块，再找到“舵机模块”。



图 2.1.8 舵机模块

⑤选中“舵机模块”放入“循环执行”中，与“映射模块”嵌套如下：

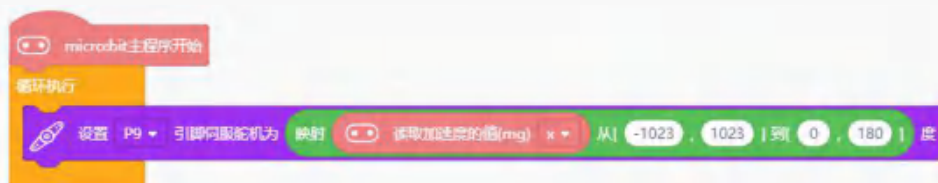


图 2.1.9 执行程序

⑥最后进行结构制作，用双面胶将舵机和扩展板固定起来，使两者保持相对静止。



图 2.1.10 结构固定

5. 麦克发明室

micro:bit 在 Mind+ 里面也有“指南针模块”，那么就用自带的电子罗盘功能制作一个指南针吧！



由于 micro:bit 自带的舵机模块的旋转范围是 $0^{\circ}\sim 180^{\circ}$ ，而指南针需要 360° 旋转，因此你可以不使用舵机，而采用在 micro:bit LED 点阵上“显示”的“指南针朝向度数”来判断方位哦！仔细阅读 LED 点阵上的提示操作，你可以做到的哦！

我的设计

我的程序



项目二：DJ 演奏台

1. 麦克的故事

音乐节目看多了，我也想要像 DJ 一样呈现出炫彩的灯光，演奏出动感的音符。我要用 micro:bit 做一个像模像样的 DJ 演奏台。



先来看看要用到哪些硬件吧！

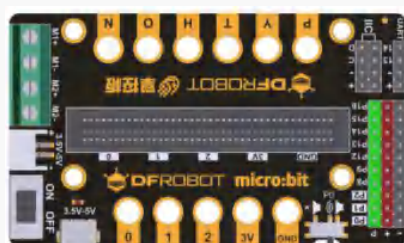
2. 麦克的布兜



1× micro:bit 主控板



1× micro USB 连接线



1× micro:bit 扩展板



1× 旋钮模块



1× 彩虹灯带

3. 小麦的愿望



我想要通过调整旋钮的转动角度，实现演奏不同的音乐，同时灯带闪烁不一样的色彩！



当用手拨动旋钮时，旋钮对应的引脚读出的模拟值在 0~1023 之间变化，我们可以将它分隔成 8 个小范围，对应音乐中的一个八度 “Do~Do’ ”，并且对应小灯的变化，对应图见表 2.2.1。

表 2.2.1

旋钮（P1）的输入值	声音	灯带
0~128	Do	1 颗灯亮
128~256	Re	2 颗灯亮
256~384	Mi	3 颗灯亮
384~512	Fa	4 颗灯亮
512~640	So	5 颗灯亮
640~768	La	6 颗灯亮
768~996	Xi	7 颗灯亮
996~1023	Do’	7 颗灯亮，均为蓝色

4. 小克的黑板

STEP1: 将旋钮模块接在扩展板的 P1 号接口；将彩虹灯带接在扩展板的 P2 号接口。

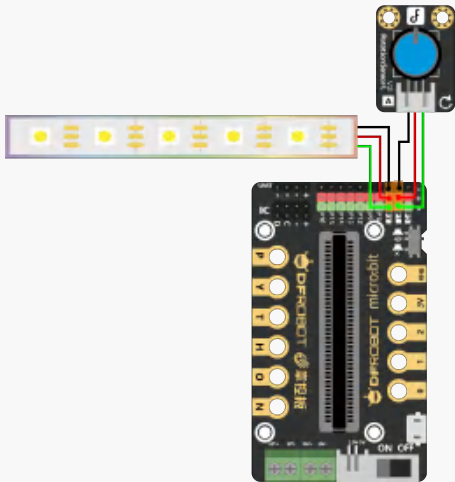


图 2.2.1 连线示意图

STEP2: 想要完成愿望，首先要绘制出 DJ 演奏台的实现逻辑图哦。

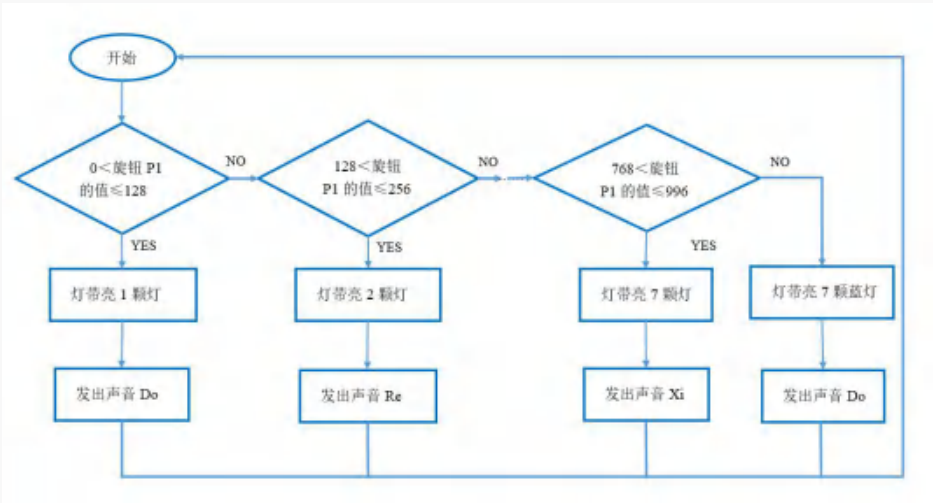


图 2.2.2 逻辑图

STEP3: 编写程序

①以“ $0 < \text{旋钮 P1} < 128$ ”为例，通过“运算符”模块中的“与”模块设置旋钮 P1 值的控制范围，将范围放入“如果...那么执行”的条件中，在“那么执行”下嵌入需要的灯带与音效，并加入“等待...秒”。下图是第 1 个区域的控制程序，后 6 个控制程序以此类推。



图 2.2.3 程序示例 1

②由于只有 7 盏灯，在第 8 个区域中，可以通过不同的灯带表现形式来表达“Do”，程序如下图。



图 2.2.4 程序示例 2



图 2.2.5 执行程序

③为了可以方便进行演奏，需要给 DJ 演奏台做一个便于操作的外观哦。给旋钮做个手柄，方便进行操作，将旋钮固定好之后，将每个音的范围标注出来吧！



图 2.2.6 结构改变 1

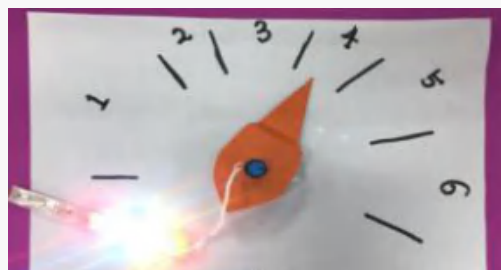


图 2.2.7 结构改变 2

5. 麦克发明室

想不想让你的演奏台更完善呢？把 micro:bit 主板的 5×5LED 点阵也利用起来，试试能不能当演奏出不同的音节时，点阵对应显示该音节？快尝试制作一下吧！



我的设计	我的程序



项目三：可移动门铃

1. 麦克的故事

在家里我们有可能在卧室认真的听音乐或者看电视，并未被外界的声音所打扰。那么你想在卧室就听到是否有人在门外敲门吗？可以制作一个可以远程遥控的门铃哦，就不用担心听不见门铃的声音了！

先来看看要用到哪些硬件吧！

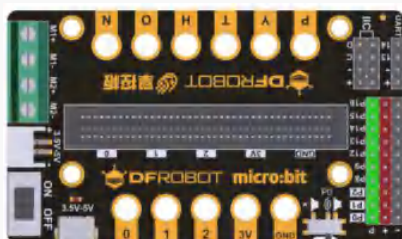
2. 麦克的布兜



2× micro:bit 主控板



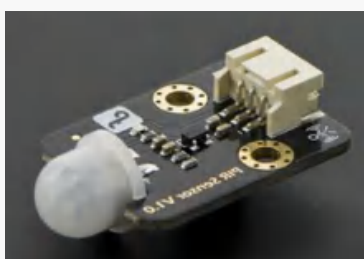
1× micro USB 连接线



2× micro:bit 扩展板



1× 按钮模块

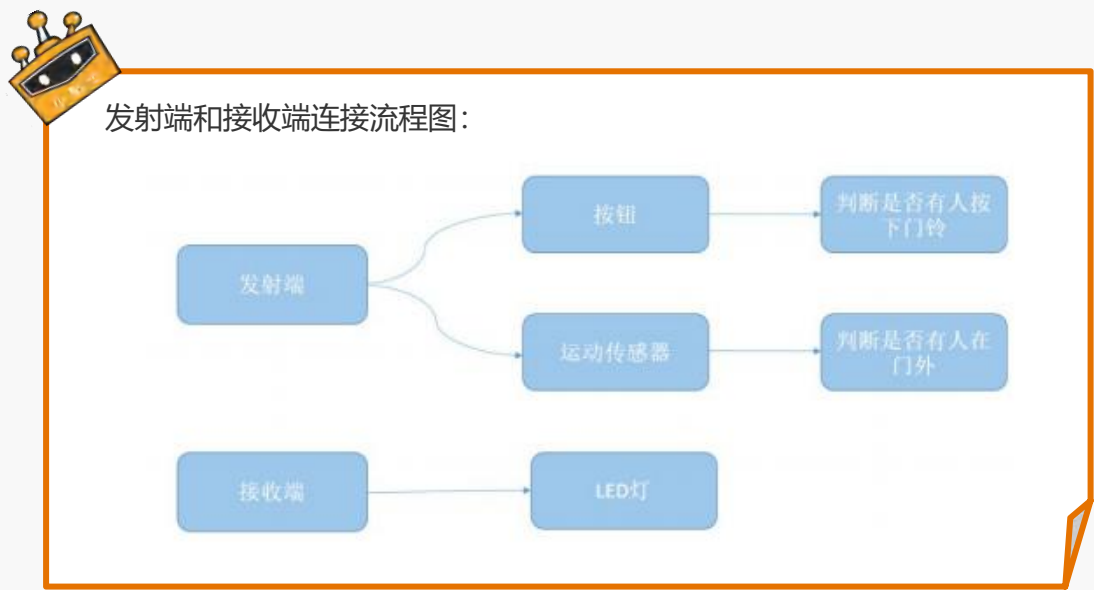


1× 运动传感器模块



1× LED 灯模块

3. 小麦的愿望



4. 小克的黑板

STEP1: 发射端硬件连接：

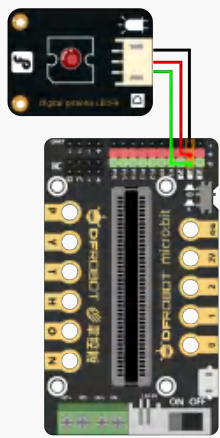


图 2.3.1 发射端连接示意图

接收端硬件连接：

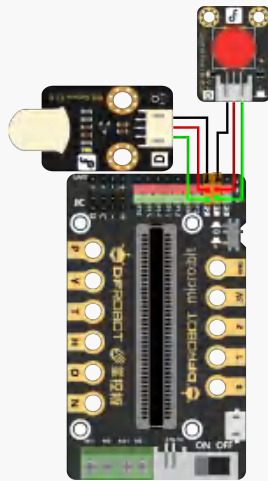


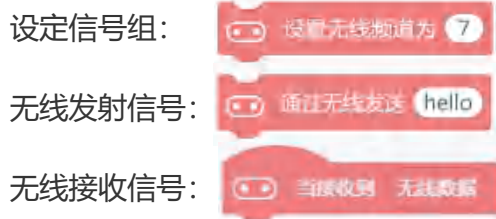
图 2.3.2 接收端连接示意图

STEP2: 了解无线传输模块

这就需要学习一个新的知识：**无线传输模块**啦！当发射端接收到信号之后，通过无线传输模块将信号传到接收端，所以需要**分别对发射端和接收端进行程序编写**哦！



常用的无线传输模块：



注意：接收端需要识别接收到的信号，从而做出对应的动作，这里对应发射端有数字和文字两种情况哦~

STEP3: 设置发射端程序

①首先对发射端进行程序编辑，编写逻辑图。

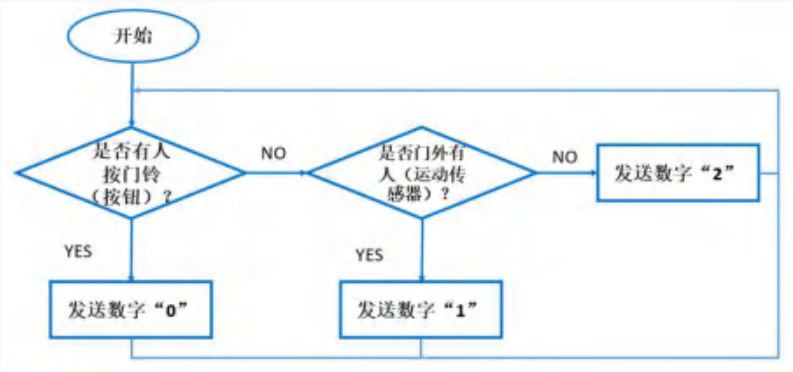


图 2.3.3 发射端逻辑图

②根据逻辑图编写发射端完整程序

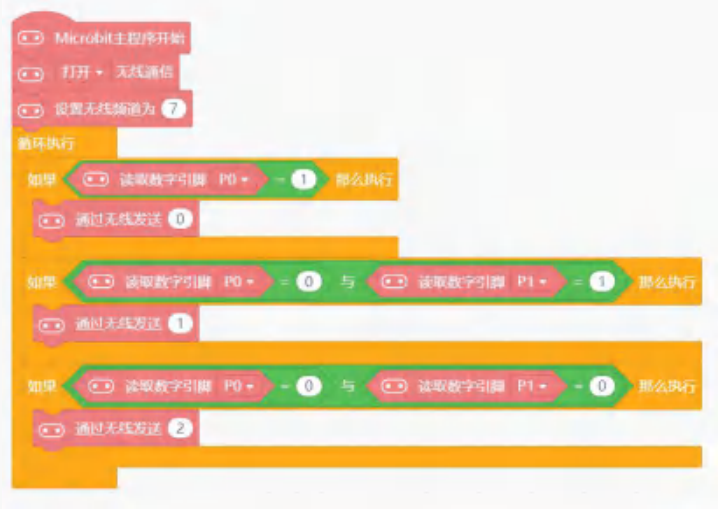


图 2.3.4 发射端执行程序

STEP4: 设置接收端程序

①对接收端端进行程序编辑，编程逻辑图。

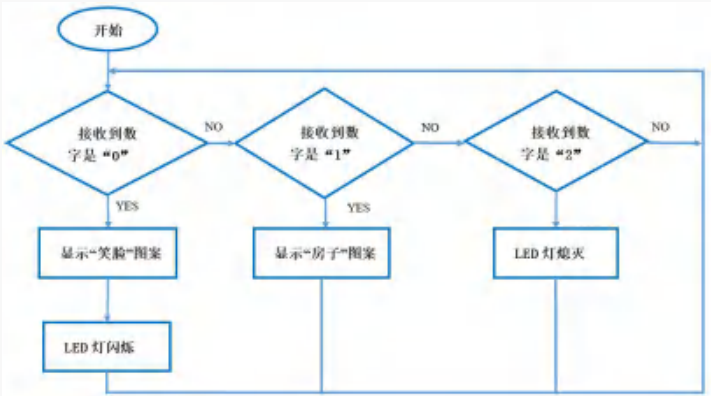


图 2.3.5 接收端逻辑图

②根据逻辑图编写接收端完整程序。

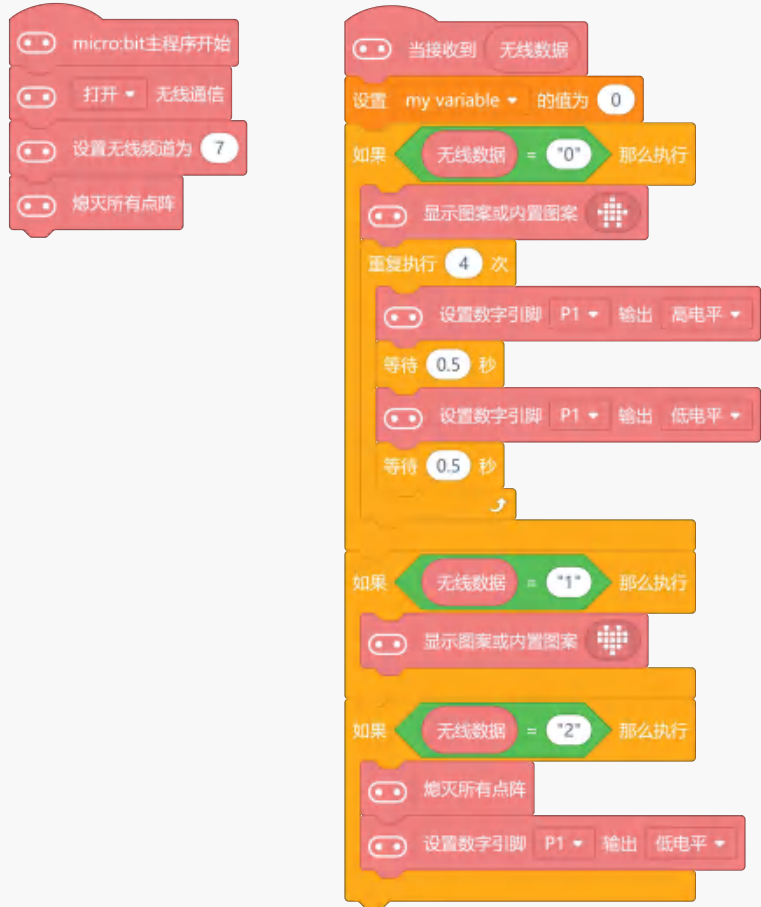


图 2.3.6 接收端执行程序

实验效果：

(1) 当门外有人经过但是没有按门铃时：

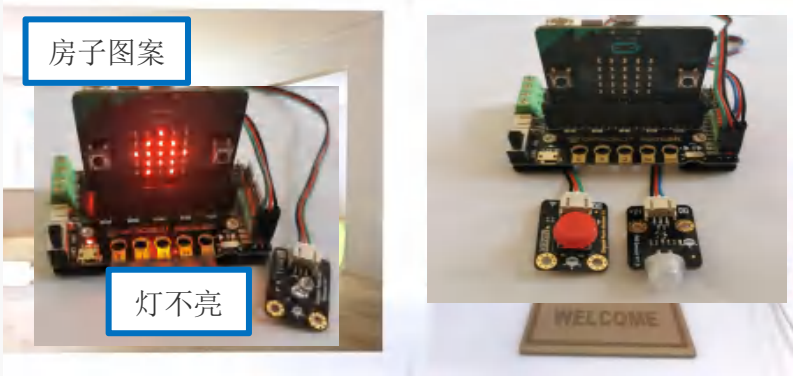


图 2.3.7 实验效果 1

(2) 当门外有人按门铃时

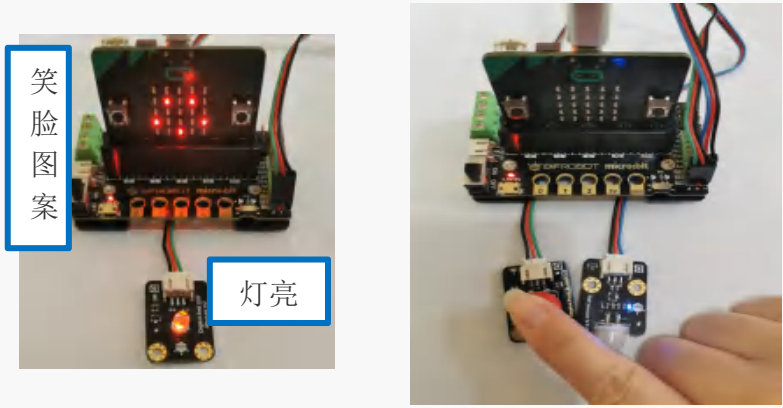


图 2.3.8 实验效果 2

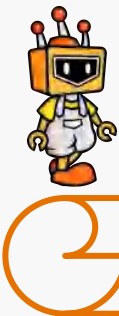
5. 麦克发明室

在门外按门铃的人一定很焦急, 请为门外的门铃设定专属铃声吧~



我的设计

我的程序



项目四：拆弹游戏

1. 麦克的故事

遥控汽车、遥控飞机我们平常经常玩，但是现在我想玩一些新鲜的、有趣的、酷炫的玩具，重点是可以考验反应能力哦！那我们就用布兜里的工具制作一个考验反应能力的拆弹游戏！



先来看看要用到哪些硬件吧！

2. 麦克的布兜



1× micro:bit 主控板



1× micro USB 连接线



1× micro:bit 扩展板



1× 彩虹灯带

3. 小麦的愿望



我想要做一个炸弹游戏，启动游戏之后，LED 灯开始进入倒计时模式，灯全部熄灭也就是倒计时结束，炸弹会被引爆~

如果想在炸弹引爆之前解锁，需要把 micro:bit 板向 LED 灯提示的方向倾斜~

如果在 LED 灯熄灭之前完成了 20 个方向的倾斜，那么就是拆弹成功啦！



这样的拆弹装置由以下三个部分组成：

1. 动作识别：用来识别 micro:bit 板的倾斜方向是否与 LED 灯指示的方向一致。
2. 彩灯倒计时：为 LED 灯设置倒计时的效果，每隔 10s 亮灯数量减 1。
3. 计分系统：在 LED 灯全部熄灭之前完成 20 个动作则拆弹成功；若 LED 灯全部熄灭仍然没有完成 20 个动作，则游戏结束。

4. 小克的黑板

STEP1: 将彩虹灯带接在扩展板的 P0 号接口。

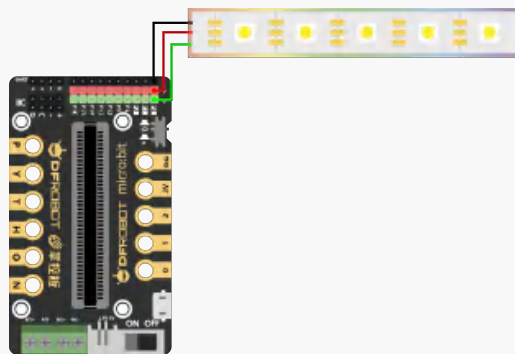


图 2.4.1 连线示意图

STEP2: 动作识别部分

1. 相关指令学习

首先我们需要让 micro:bit 板随机产生并显示一个方向，如果我们的动作与该方向

一致，那么再随机产生一个方向；如果我们的动作与该方向不一致，那么一直保持当前的箭头方向，程序的流程图大致如下。

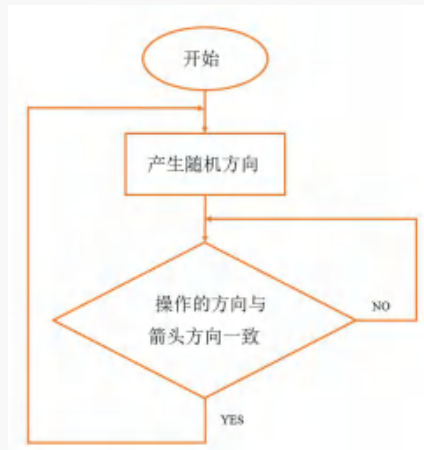


图 2.4.2 动作识别部分逻辑图



选取随机数模块：

通过运算符中的“选取随机数”模块来随机选取指定范围中的数字，比如上下左右有 4 个方向，我们就可以指定随机数从 0 到 3 之间随机选取啦！



构建函数模块：

通过函数中的“自定义模块”可以构建函数，构建函数之后，如果需要使用到该函数，直接将指令拖到对应位置就好啦！

比如我们现在需要将“0, 1, 2, 3”四个数字和“上下左右”四个方向对应起来，那么就可以构建“产生随机方向”的函数，在构建之后就会出现“产生随机方向”指令啦！



函数

自定义模块...

产生随机方向



获取当前姿态模块：

可以通过“当前姿态为”指令获取当前 micro:bit 板的姿态：

2.程序编写

①设置变量“目标方向”，它表示 LED 灯显示的方向，需要将“目标方向”的值设置为随机数 0~3。



图 2.4.3 设置目标方向的值



如果你忘记了如何设置变量，去“项目四-点亮彩虹灯带”复习一下

②建立一个函数“产生随机方向”，使“0123”4个数字分别对应 LED 点阵上不同方向的箭头。



图 2.4.4 “产生随机方向”函数

③设置变量“操作方向”，它表示 micro:bit 板的当前的倾斜方向，并通过“当前姿态为”指令将“操作方向”与“0123”4个数字对应起来。

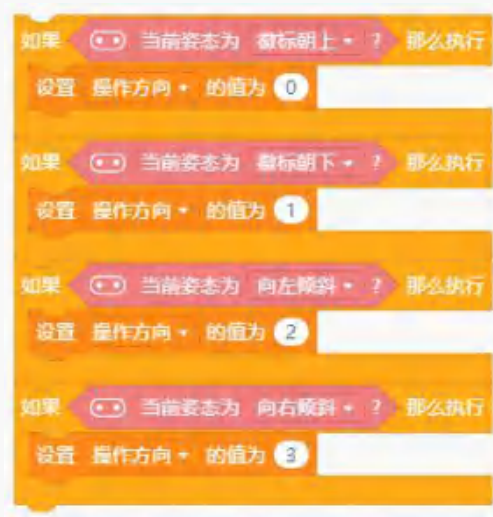


图 2.4.5 获取当前姿态指令

④通过一个“如果-那么执行”指令进行条件判断，如果操作方向与目标方向一致，那么继续产生随机方向。

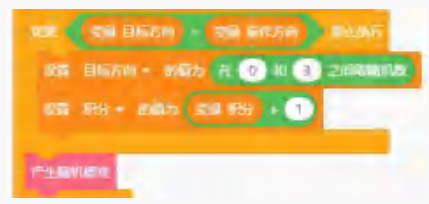


图 2.4.6 条件判断程序

⑤这个部分的完整程序为：

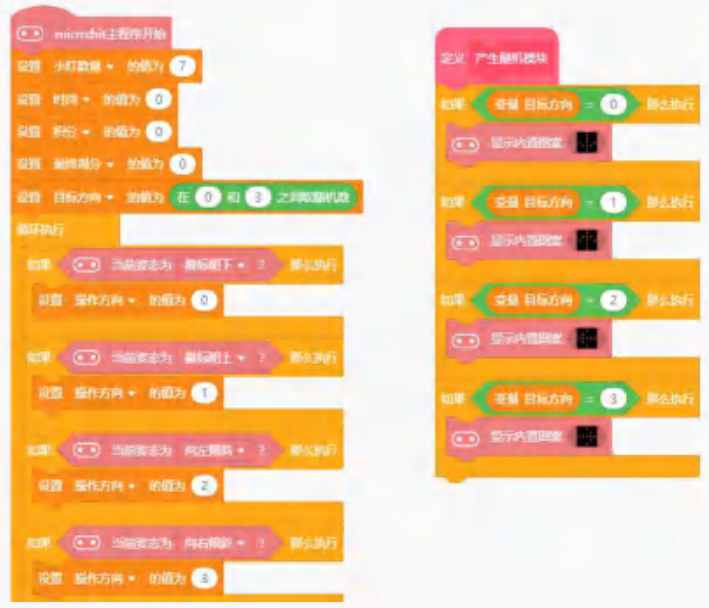


图 2.4.7 动作识别部分完整程序

STEP3: 彩灯倒计时部分

1.逻辑图绘制

随着时间的推移，led 灯珠将会以稳定的速度相继熄灭。所以反应速度越快，小灯剩余数量越多。

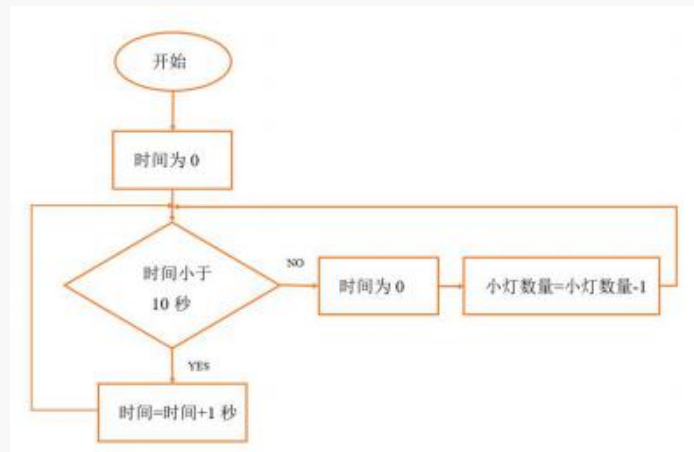


图 2.4.8 彩灯倒计时部分逻辑图

2.程序编写

①构建变量“时间”和“小灯数量”，并将“小灯数量”设置为 7（也就是 7 盏灯都亮起），将“时间”设置为 0，通过“如果-否则执行”实现逻辑图的要求。



注意：因为 micro:bit 在处理数据时会消耗掉一定的时间，所以倒计时模块实际运行时走得慢一些哦~



图 2.4.9 “如果-那么执行”指令

②在倒计时的同时，不要忘记设置小灯的亮灯程序哦：

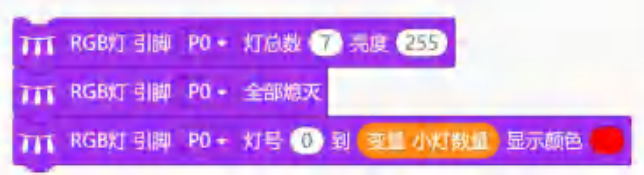


图 2.4.10 LED 灯显示指令

③前两部分的完整程序为：

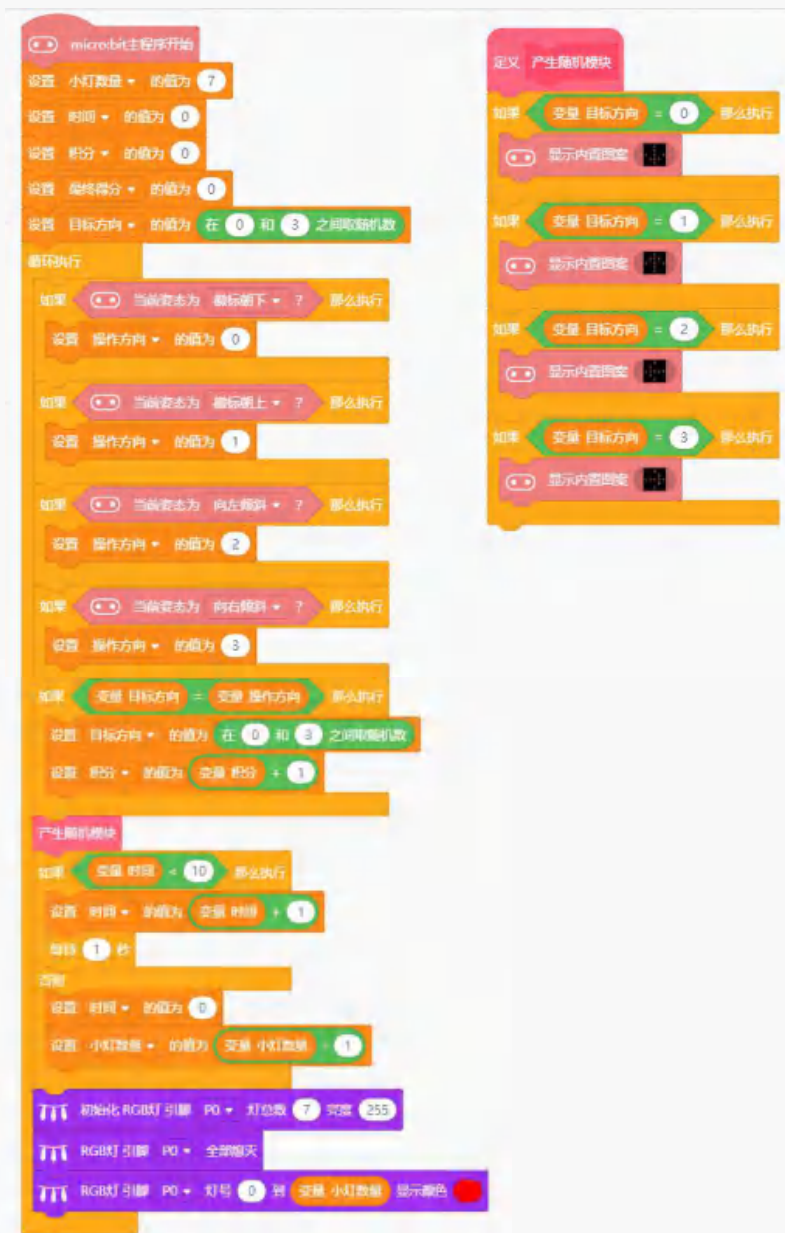


图 2.4.11 前两个部分完整程序

STEP4: 计分部分

1.逻辑图绘制

如果每次操作方向和目标方向一致，则积一分，如果积满 20 分，亮灯的数量还大于 0，则挑战成功，即拆弹成功，最终得分为 LED 灯的剩余个数；否则，拆弹失败。

流程图大致如下：

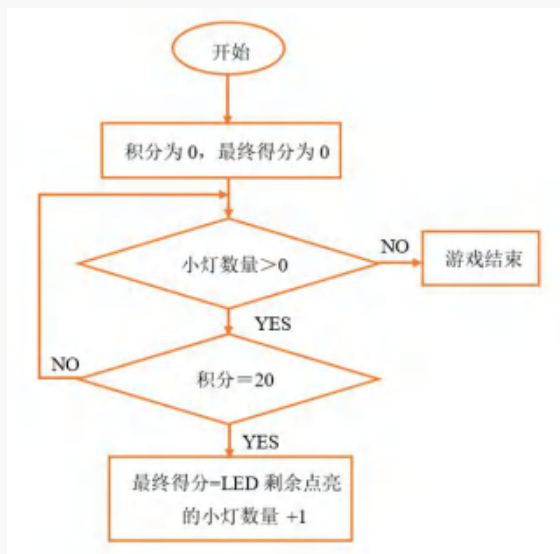


图 2.4.12 计分部分逻辑图

2.程序编写

①构建变量“积分”和“最终得分”，并将“积分”设置为 0，将“最终得分”设置为 0，通过“如果-那么执行”实现逻辑图的要求。



图 2.4.13 部分程序

②炸弹游戏完成程序如下：



图 2.4.14 完整执行程序

5. 麦克发明室

如果你想要它更完整更有趣，可以加一个游戏结束的标志哦，快去和小伙伴们探索一下如何实施吧！



我的设计

我的程序



第三章：大展拳脚

前面学习了很多传感器模块，是不是已经都掌握使用方法了呢？现在到大展拳脚的时候啦！

快使用它们做出与我们的生活密切相关且能在家里玩的项目吧！

项目一：投篮机

1. 麦克的故事

在家想投篮怎么办，那就来设计一个好玩的投篮机吧。



先来看看要用到哪些硬件吧！

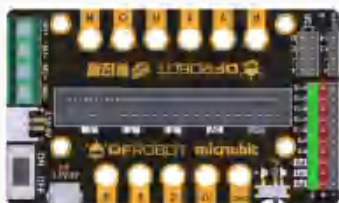
2. 麦克的布兜



1× micro:bit 主控板



1× micro USB 连接线



1× 扩展板



1× 超声波传感器



1× 按钮

3. 小麦的愿望：超声波判断篮球是否投中



我想用超声波传感器做一个简易投篮机，在家也可以玩好玩的投篮游戏~

4. 小克的黑板

STEP1: 将传感器连接到扩展板上。

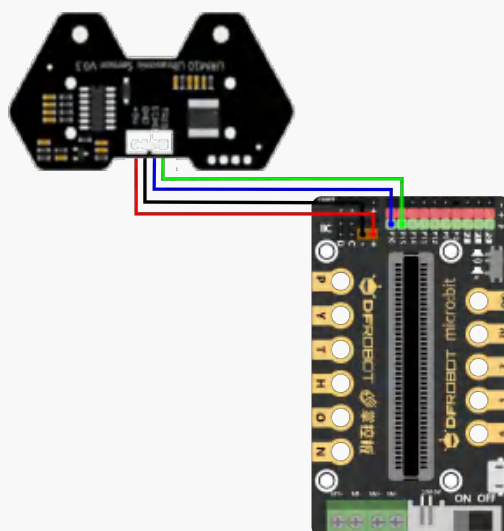


图 3.1.1 接线示意图

STEP2: 编写程序

①软件设置：使用超声波需要打开 Mind+软件的“扩展”，在“传感器”下点击加载“超声波测距传感器”。



图 3.1.2 超声波选择

②测试程序：串口读取超声波距离



图 3.1.3 串口读取超声波距离

③功能程序：超声波每次检测到障碍物，算作进球一次，将得分实时显示在屏幕上



图 3.1.4 超声波检测障碍物

5. 小麦的愿望：按下按钮，开始或结束游戏



6. 小克的黑板

STEP1: 将传感器连接到扩展板上。

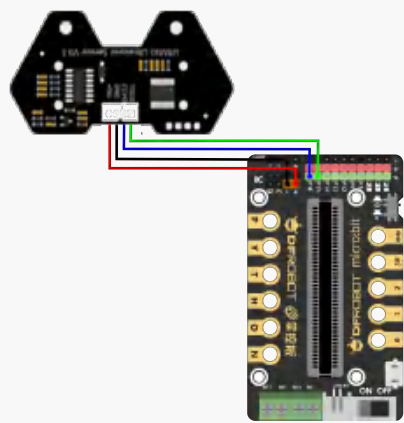


图 3.1.5 连线示意图

STEP2: 功能程序。

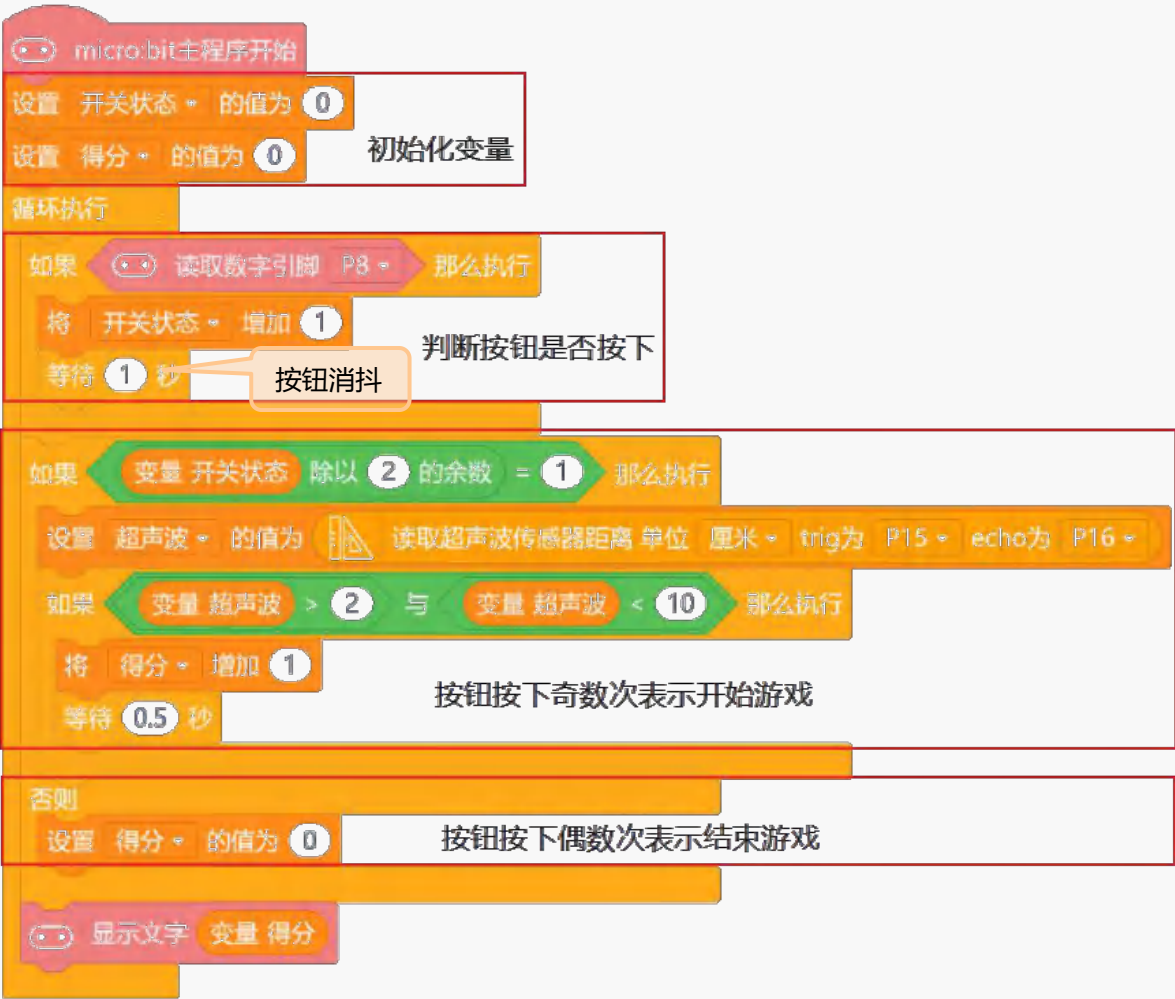


图 3.1.6 完整程序

7. 麦克小发明

这里只是实现投篮机的基本功能，如果你有更多的创意，赶紧动手尝试吧！



我的设计	我的程序
	

项目二 火线冲击

1. 麦克的故事

火线冲击游戏这么吸引人的眼球，在这个项目中，我们将从游戏设计者的角度出发，设计一款好玩的火线冲击游戏，让自己在家也可以玩。



先来看看要用到哪些硬件吧！

2. 麦克的布兜



1× micro:bit 主控板



1× micro USB 连接线



1× micro:bit 扩展板



1× 土壤湿度传感器



1× 按钮



1× 彩虹灯带

3. 小麦的愿望：判断是否导通



我想做一个简易火线冲击设备，并能够在判断火线与套环是否导通时，记录导通次数~

4. 小克的黑板

STEP1: 将传感器连接到扩展板上。

这里可以选用导电铜丝制作出火线和套环。

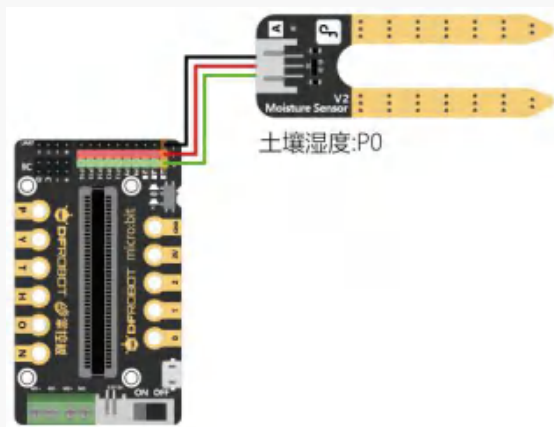


图 3.2.1 连线示意图

STEP2: 编写程序

①测试程序：将土壤湿度传感器测量的湿度显示在屏幕上。



图 3.2.2 读取湿度值

②功能程序:

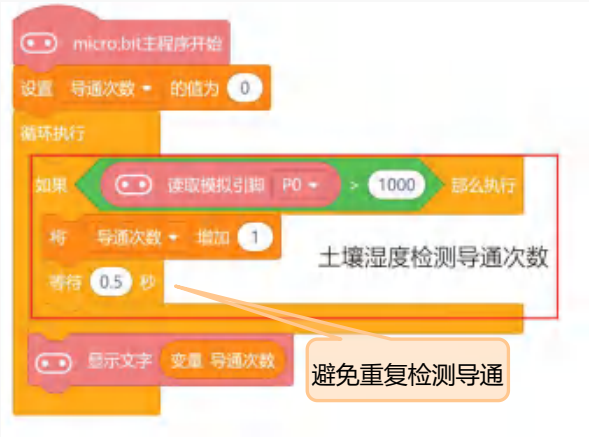
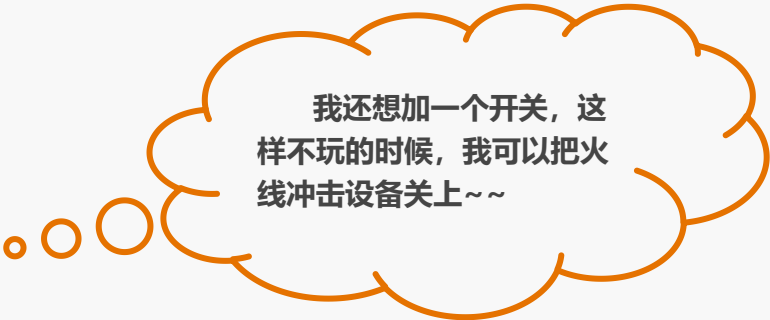


图 3.2.3 检测导通次数

5. 小麦的愿望：按下按钮，开始或结束游戏



6. 小克的黑板

STEP1：将传感器连接到扩展板上。

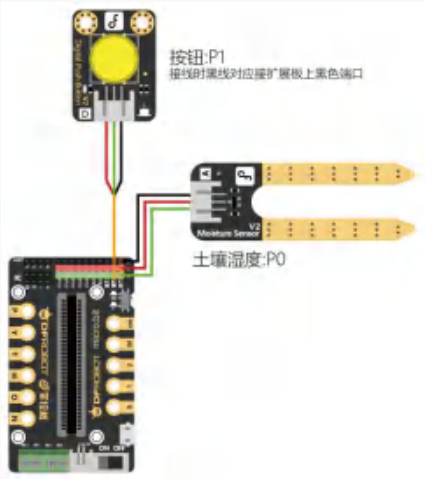


图 3.2.4 连线示意图

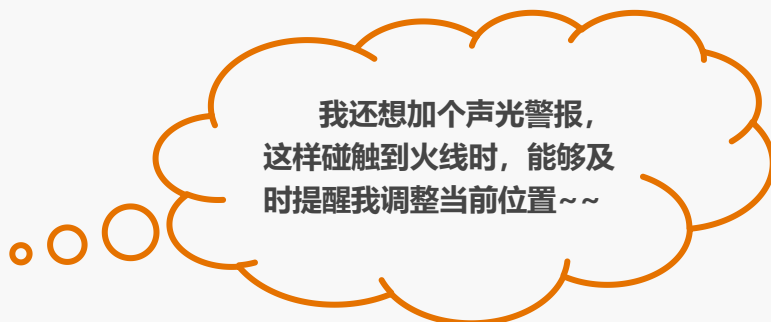
STEP2: 功能程序。

这里以按钮作为游戏开关，与投篮机项目相同，我们可以通过判断按下次数的奇、偶性，实现每按下一次按钮就切换游戏的开始、结束。



图 3.2.5 功能程序

7. 小麦的愿望：碰触到火线时，发出声光警报



8. 小克的黑板

STEP1: 将传感器连接到扩展板上。

加入 RGB 灯带可以实现炫酷的灯光效果，micro:bit 扩展板自带的蜂鸣器即可实现声音警报！

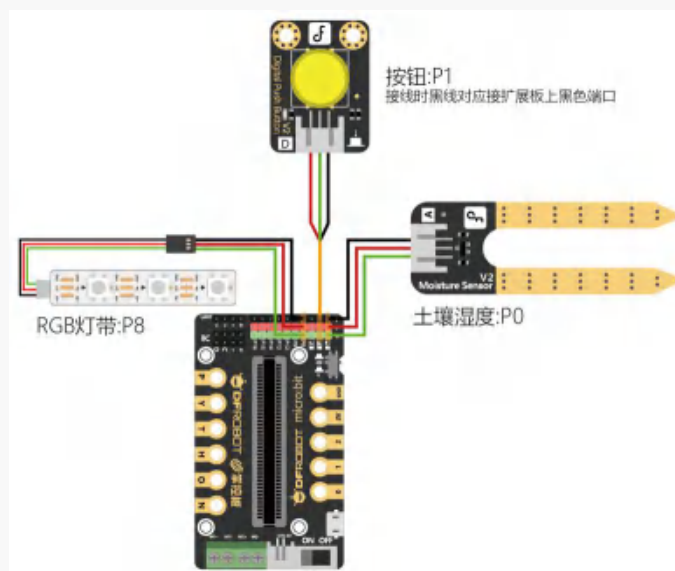


图 3.2.6 连线示意图

STEP2: 功能程序

在小麦的愿望 5 的程序基础上作如下修改。

micro:bit 主程序开始

- 初始化 RGB 灯 引脚 P8 灯总数 7**
灯带初始化指令, 用于设置引脚、灯总数和亮度
- RGB 灯 设置引脚 P8 灯带亮度为 255**
- RGB 灯 引脚 P8 灯号 0 到 4 显示渐变颜色 色调从 1 到 360**
设置灯带色调指令, 色调设置范围对应如下色环 1-360 度

设置 导通次数 的值为 0
设置 开关状态 的值为 0

循环执行

- 如果 读取数字引脚 P1 那么执行**
 - 将 开关状态 增加 1**
灯带循环移动指令
 - 等待 0.5 秒**
- 如果 变量 开关状态 除以 2 的余数 = 1 那么执行**
 - 导通检测**
 - RGB 灯 引脚 P8 循环移动 1 单位**
游戏中产生流水灯效果
 - 等待 0.1 秒**
等待 0.1 秒, 即流水灯移动速度
- 否则**
 - 设置 导通次数 的值为 0**
 - 熄灭所有点阵**

定义 导通检测

- 如果 读取模拟引脚 P0 > 1000 那么执行**
导通时灯带显示红色
- 将 导通次数 增加 1**
- RGB 灯 引脚 P8 灯号 0 到 6 显示颜色 红色**
声光报警
- 接口 P0 播放音符 1 低 C/C3 1/4 拍**
播放音符 1/4 拍指令在这里相当于延时 0.25 秒, 可用于避免重复导通检测
- RGB 灯 引脚 P8 灯号 0 到 6 显示渐变颜色 色调从 1 到 360**
- 显示文字 变量 导通次数**

色环图:

9. 麦克小发明

这里只是实现火线冲击的基本功能。那么学完这节课你能否自己设定导通次数上限呢？

例如：导通次数超限则游戏失败。

如果你还有更多的创意，赶紧动手尝试吧！



外观设计

程序设计



项目三 感应垃圾桶

1. 麦克的故事

随着技术的发展，在**家居生活**方面，智能电饭煲、扫地机器人、空气净化器等各类**智能家电**的出现为我们的生活带来了很多方便。甚至在**扔垃圾**的问题上，垃圾桶都有了一些非常大的改变，**自动感应垃圾桶**成为了人们新的挑选，让我们在扔垃圾的时候特别的方便，而且特别的干净卫生。



在这个项目中，我们就将制作一个感应垃圾桶。在进入项目学习前，请思考一下，传统的垃圾桶有哪些缺点呢？解决这些缺点，那就是我们要制作的感应垃圾桶！



先来看看要用到哪些硬件吧！

2. 麦克的布兜



1× micro:bit 主控板



1× micro USB 连接线



1× micro:bit 扩展板



1× 超声波传感器



1× 按钮



1× 舵机

3. 小麦的愿望：舵机转动，带动垃圾桶盖子打开或关闭



我想做一个开关可以打开或者关闭垃圾桶盖子的装置~

4. 小克的黑板

STEP1: 将传感器连接到扩展板上。

使用 P0 引脚注意关闭扩展板上蜂鸣器开关

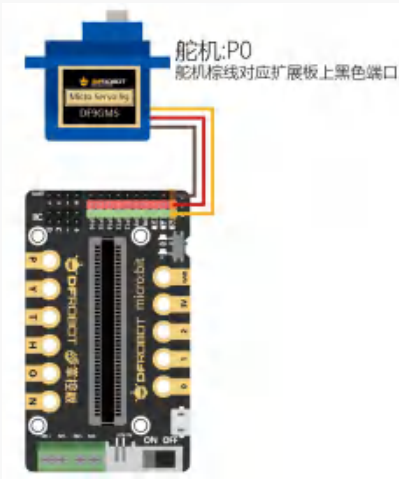


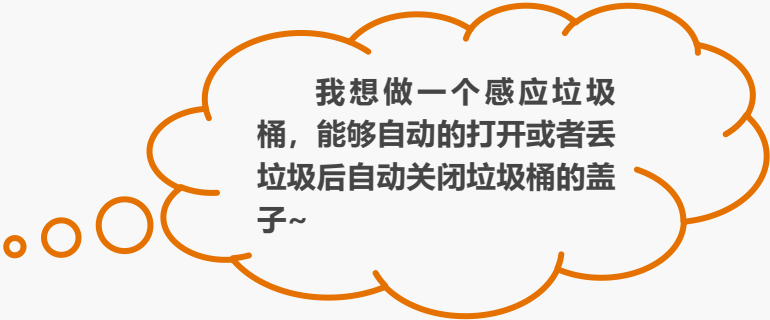
图 3.3.1 连线示意图

STEP2: 舵机转动



图 3.3.2 舵机转动

5. 小麦的愿望：超声波传感器感应



6. 小克的黑板

STEP1: 将传感器连接到扩展板上。

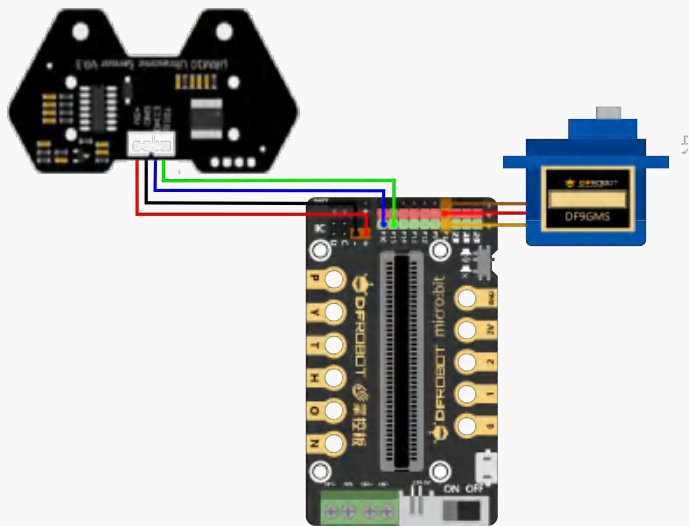


图 3.3.3 连线示意图

STEP2: 编写程序

图 3.3.4 超声波感应



运行结果：超声波感应到 10cm 以内的障碍物时，舵机转到 180 度，3 秒后回转到 0 度。

7. 小麦的愿望：按钮控制垃圾桶盖常开



我想做一个可以控制垃圾桶常开的按钮，这样就不怕要一直扔垃圾啦~

8. 小克的黑板

STEP1: 将传感器连接到扩展板上。

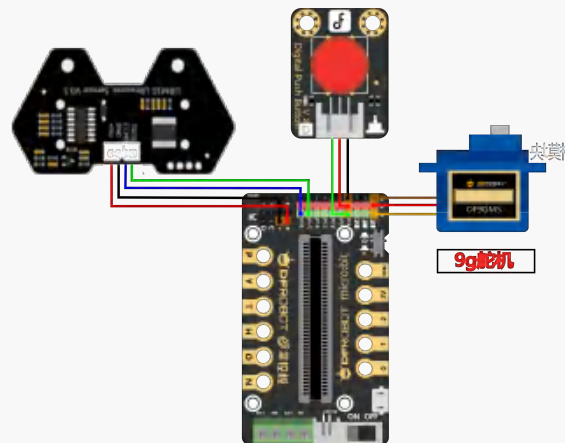


图 3.3.5 连线示意图

这里需要实现的功能是：

- 程序运行时为**超声波感应状态**；
- **按下按钮**，**盖子常开**，此时超声波不起作用；
- **再次按下按钮**，**盖子关闭**，超声波恢复检测。

这里用程序实现的难点是要记录**按钮的输出状态**，就像家里的台灯一样，第一次按下开关，台灯打开，再次按下台灯关闭...如此循环往复。假如将**灯灭记录为 0**，**灯亮记录为 1**，那么每一次按钮按下，灯都会发生一次**0、1 变换**。所以解决方案可以是：按下按钮后首先判断灯为 0 还是 1，如果为 0 就将其置 1，如果为 1 就将其置 0。

在程序中，使用了**变量“开关”**来记录按钮的输出状态。流程图如下。

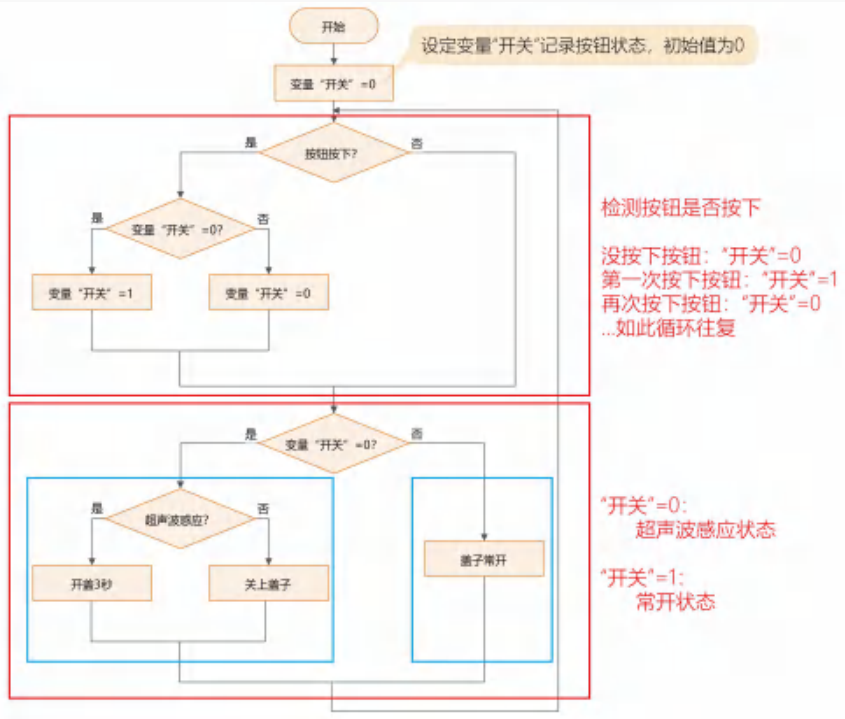
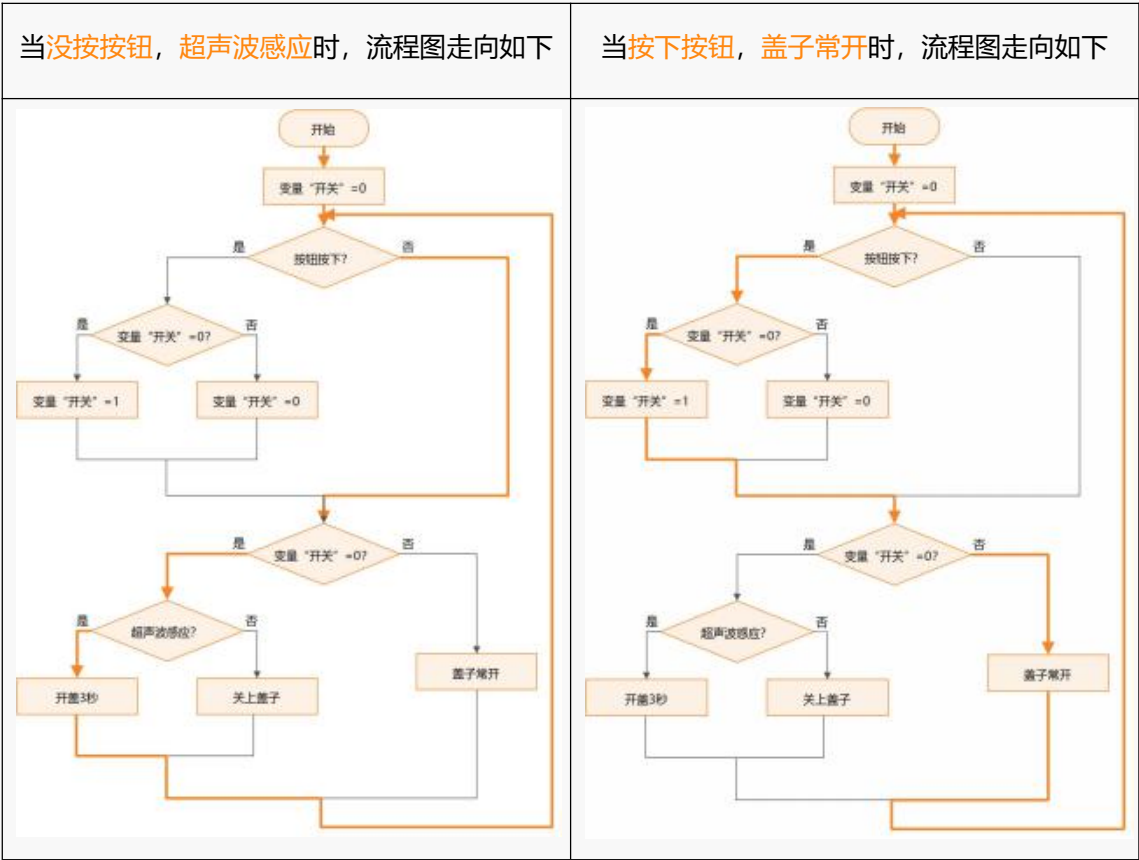


图 3.3.6 流程图

为了更好的理解上面流程图，我们可以根据不同现实情况分析流程图走向。



STEP2: 完整程序



图 3.3.7 执行程序

9. 麦克小发明

这里只是实现感应垃圾桶的基本功能。
学完这个项目，是否对你有新的启发呢？如
果你还有更多的创意，赶紧动手尝试吧！



外观设计

程序设计



项目四 答案之书

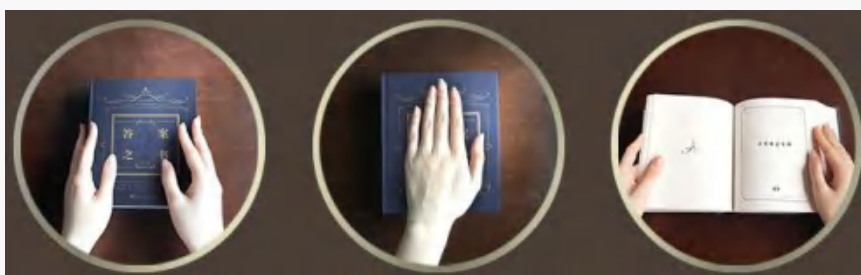
1. 麦克的故事

面对生活中的各种选择,我们时常难以抉择,如果这个时候得到一些提示,那该多好!

于是真的有人写了这样一本《答案之书》,就像下图这样,你见过吗?



在这本书里,每一页都写有一句有关选择或行动的答案,当你为生活中的小事犹豫不决的时候,随意翻开其中一页,《答案之书》就会给你一个回答。



《答案之书》有点类似于占卜,就像在一些宗教文化中常看到的求签活动一样,可以帮助人们在心理上预示一件事的结果。

先来看看要用到哪些硬件吧!

2. 麦克的布兜



1× micro:bit 主控板



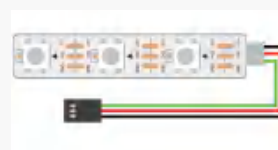
1× micro USB 连接线



1× micro:bit 扩展板



1× 按钮



1× 彩虹灯带



1× 模拟角度传感器



1× 舵机

3. 小麦的愿望：按下按钮，显示答案



我想通过按下按钮，在
micro:bit 点阵屏上随机显
示一句话作为答案~~

4. 小克的黑板

STEP1: 将传感器连接到扩展板上。

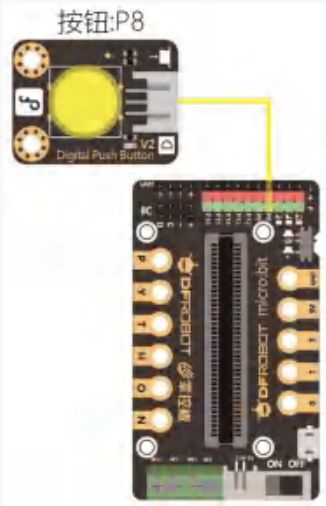


图 3.4.1 连线示意图

STEP2: 程序

使用 micro:bit 显示文字时请注意，micro:bit 板子本身不能显示中文，在程序中请使用英文、数字或者拼音（此项目中使用英文）

定义函数“回答”，显示随机一句话作为答案

micro:bit主程序开始

显示文字 ANSWER → 答案

等待 3 秒

显示文字 PRESS THE BUTTON → 按下按钮

循环执行

如果 读取数字引脚 P8 那么执行

回答

等待 0.5 秒

等待直到 读取数字引脚 P8

停止滚动动画 再次按下按钮，清除回答

显示文字 PRESS THE BUTTON

等待 0.5 秒

定义 回答

熄灭所有点阵

设置 结果 的值为 在 1 和 3 之间取随机数

如果 变量 结果 = 1 那么执行

显示文字 EVERYTHING WILL BE FINE 答案1

否则

如果 变量 结果 = 2 那么执行

显示文字 PLEASE BE PATIENT 答案2

否则

显示文字 ARE YOU JOKING 答案3

■ 设置变量“结果”为一个随机数,每个数对应一句话

■ 随机数范围越大,答案种类越多

大家可以随意修改文字内容

图 3.4.2 执行程序

运行结果：按下按钮时，屏幕上随机显示一句话。

大家可以在实际编程时，多加几句答案，这样操作会更加好玩哦！

5. 小麦的愿望：神奇的答案之书



我想做一个可以自己求签的答案之书，这样我就可以随时解疑了~

6. 小克的黑板

STEP1: 将传感器连接到扩展板上。

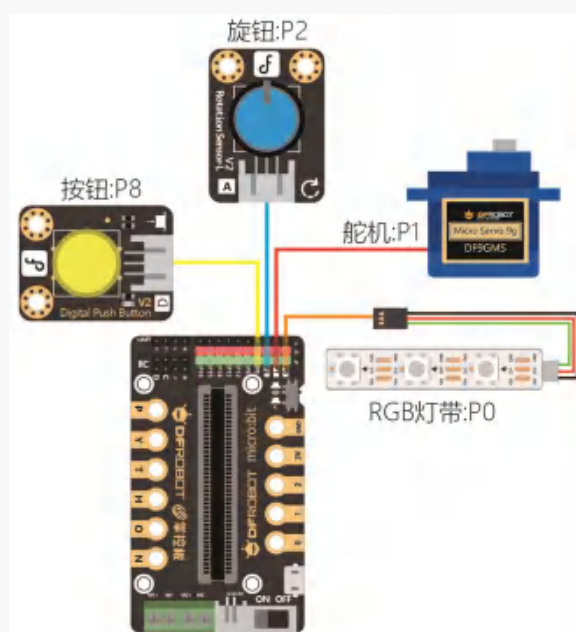


图 3.4.3 连线示意图

STEP2: 完整程序



图 3.4.4 完整程序

7. 麦克小发明

这里只是答案之书的基本功能，简单的实现求签的过程，如果你还有更多的创意，赶紧动手尝试吧！



外观设计

程序设计



第四章 各显神通

前面这么多愿望都实现了，是不是已经和 micro:bit 成为好朋友了呢？现在到各显神通的时候啦！快使用 micro:bit 主控板和各种传感器，制作出炫酷好玩的作品吧

项目一 坦克大战

1. 麦克的故事

伴随着计算机技术的发展，**电子游戏**进入了人们的视野，并迅速风靡全球。与传统游戏不同，电子游戏借助计算机营造出更丰富的视听感，进一步提升了玩家的游戏体验。

在这个项目中，我们将利用 Mind+实时模式下丰富的舞台交互复刻一款经典电子游戏——**坦克大战**。



先来看看要用到哪些硬件吧！

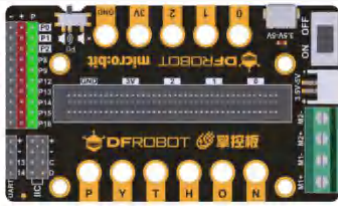
2. 麦克的布兜



1× micro:bit 主控板



1× micro USB 连接线



1× 扩展板



1× 土壤湿度



1× 按钮

3. 小麦的愿望：左右移动的“坦克”



我想通过角度传感器来控制坦“坦克”在舞台底部左右移动~~

4. 小克的黑板

STEP1: 将传感器连接到扩展板上。

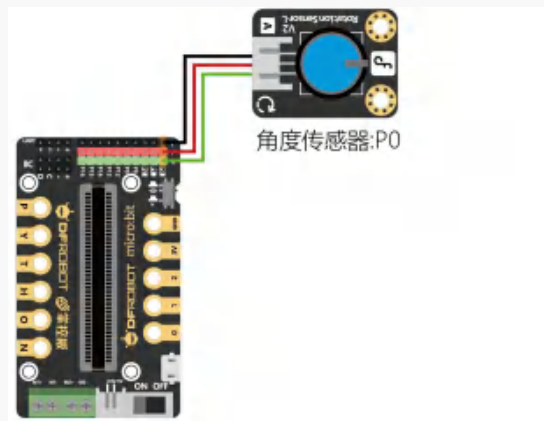


图 4.1.1 连线示意图

STEP2: 编写程序

①测试程序：角色读取角度传感器值。



图 4.1.2 读取角度传感器数值

②新建 “坦克” 角色：

“坦克” 角色图片：因为软件角色库中没有合适的图片，所以我们可以使用自己喜欢的坦克图片，下面图片可做参考（后面学习中，选择了绿色坦克，因为有双发射炮口，效果更加炫酷）。



点击 “上传角色”，选择要导入的图片。



图 4.1.3 新建 “坦克” 角色

③功能程序：



图 4.1.4 执行程序

点击绿旗或程序块运行程序。

运行结果：转动旋钮，“坦克”随转动方向左右平移。

5. 小麦的愿望：按下按钮，发射炸弹



上一个愿望我们实现了左右移动，现在我们能在左右移动的时候发射炸弹吗
~~

6. 小克的黑板

STEP1: 将传感器连接到扩展板上。

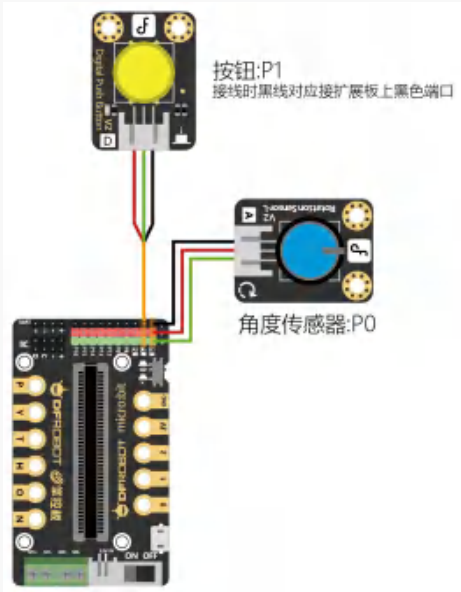


图 4.1.5 连线示意图

STEP2: 完整程序

①新建 “炸弹” 角色：

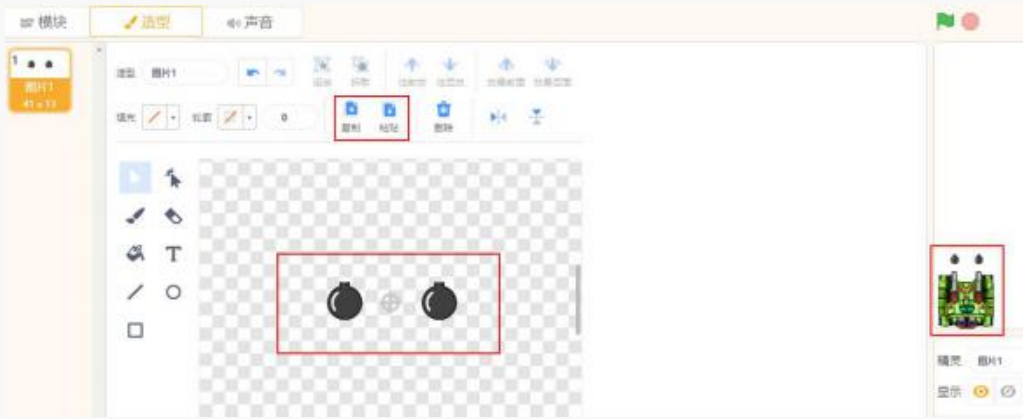
“炸弹” 角色图片：



新建 “炸弹” 角色：点击 “上传角色”，选择要导入的图片。导入后为了适应 “坦克” 的双炮口，需要绘制两个 “炸弹” 的造型。在菜单栏选择 “造型”，点击 “转换为矢量图”。



选中图案，调节大小，让舞台上的 “坦克” 和 “炸弹” 看起来大小相宜即可，然后复制 “炸弹” 造型，形成双炸弹。



③功能程序：

选中“炸弹”角色，编写下面程序



坦克



炸弹

当被点击

隐藏

循环执行

如果 读取数字引脚 P1 那么执行

等待 0.05 秒

按下按钮，产生一个炸弹

克隆 自己

通过克隆，产生多个炸弹，快速实现炸弹发射

当作为克隆体启动时

移到 坦克

显示

克隆的角色移动到“坦克”角色位置，模拟从坦克炮口发射炸弹

重复执行直到 碰到 舞台边缘

将y坐标增加 10

炸弹向上发射直到碰到舞台边缘

删除此克隆体

侦测

7. 小麦的愿望：飞机从舞台上空落下，攻击坦克；被炸弹击中则爆炸



当飞机从上方落下，攻击坦克时，若被炸弹击中则爆炸~~

8. 小克的黑板

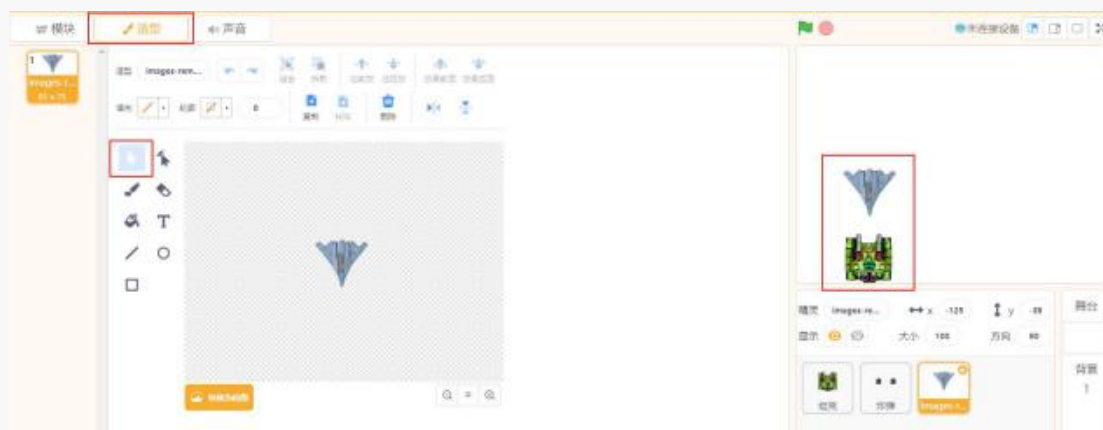
STEP1: 编写程序

①新建“飞机”角色：

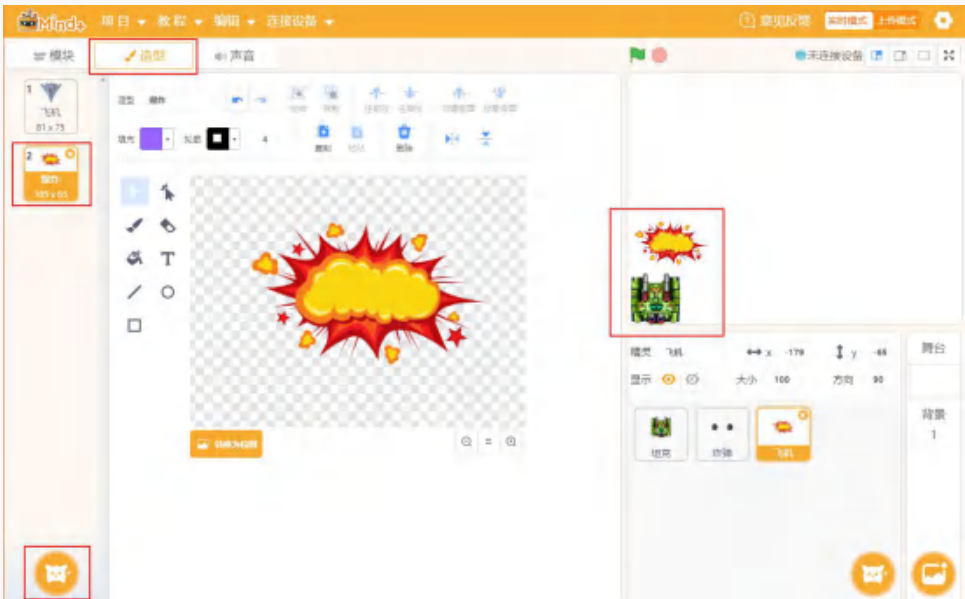
“飞机”角色图片：



新建“飞机”角色：点击“上传角色”，选择要导入的飞机图片。调节“飞机”造型大小，与舞台上的坦克看起来大小相宜即可。



在“飞机”角色下，加入“爆炸”造型，并调节造型大小。



②功能程序:

选中“飞机”角色，编写下面程序



9. 麦克小发明

这里只是坦克之战的基本功能，如果你还有更多的创意，赶紧动手尝试吧！



外观设计

程序设计



项目二 AI 助力垃圾分类

1. 麦克的故事

随着垃圾分类立法进程在全国不断推动，2019 年 7 月，上海作为第一站，正式开始实施垃圾分类最严政策《上海市生活垃圾管理条例》，「你是什么垃圾？」也成了很多上海人每天不得不面对的灵魂拷问。



可能有人会想：“总共才四个分类，看上去也不难啊。”虽说只分成了四大类，可是每个大类里面还有许多小类，每个小类还会有具体的划分。面对各种各样的垃圾到底属于哪一类，还真让居民们犯难！

为此，很多人工智能技术在垃圾分类中的应用应运而生。比如支付宝推出的“垃圾分类指南”小程序，输入垃圾名字后即可一键检索种类，还可以通过 AI 摄像头，智能识别垃圾。另外一些手机厂商也纷纷推出垃圾分类的语音助手。



在这个项目中，我们就将应用 AI 技术助力垃圾分类，利用 Mind+ 中自带的“语音识别”模块，做一个垃圾分类的语音助手，帮助居民更好的完成垃圾分类。

先来看看要用到哪些硬件吧！

2. 麦克的布兜



1× micro:bit 主控板



1× micro USB 连接线

3. 小麦的愿望：语音识别



我想做一个可以通过语音识别完成对应动作的装置

4. 小克的黑板

STEP1：软件设置：

使用语音识别功能需要在 Mind+ 的实时模式下，打开“扩展”，在“功能模块”下点击加载“语音识别”。



图 4.2.1 软件设置

STEP2: 编写程序。

①语音识别程序编写：



图 4.2.2 执行程序

②语音识别相关指令介绍：



图 4.2.3 执行程序说明

③连接设备，运行程序。

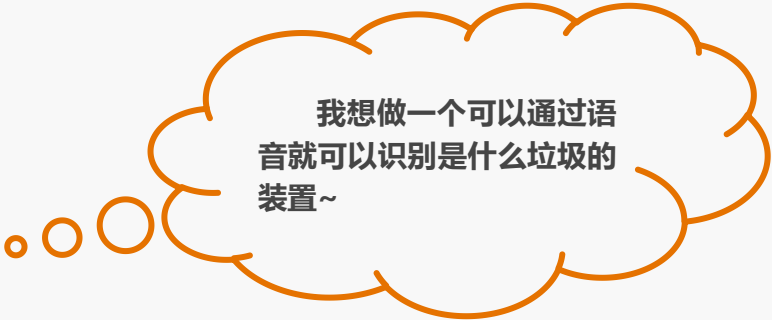
按下 **A 键**，当舞台出现**声波图**时，等待**波形图开始抖动**，对着**电脑麦克风**说话，停止说话后，声波图消失，Mind+将会显示语音转换成的文字。



图 4.2.4 运行程序

* Mind+中共有两个语音识别服务器，服务器 1 为实时识别语音，逐字翻译成文字；服务器 2 为录音一段时间后，将一句话翻译成文字。建议默认使用服务器 1，如果服务器 1 不稳定，可切换成服务器 2。

5. 小麦的愿望：垃圾分类语音助手



6. 小克的黑板

STEP1：流程图

①垃圾结果可分类：

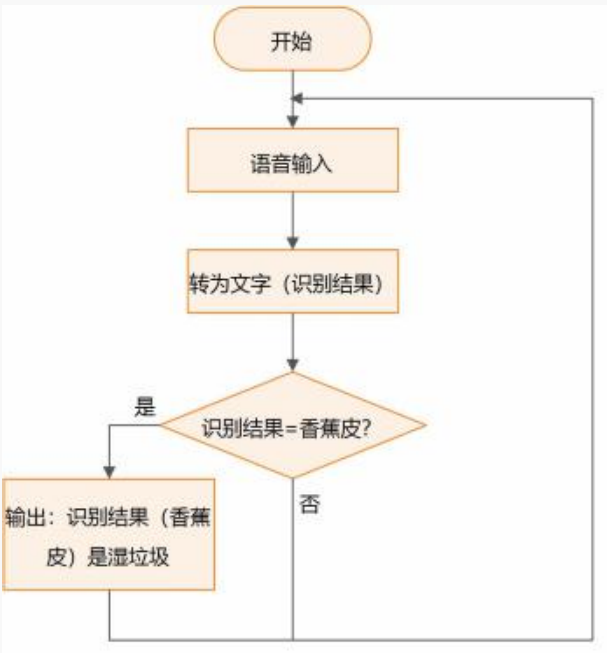


图 4.2.5 流程图 1

②垃圾按类区分并对应：

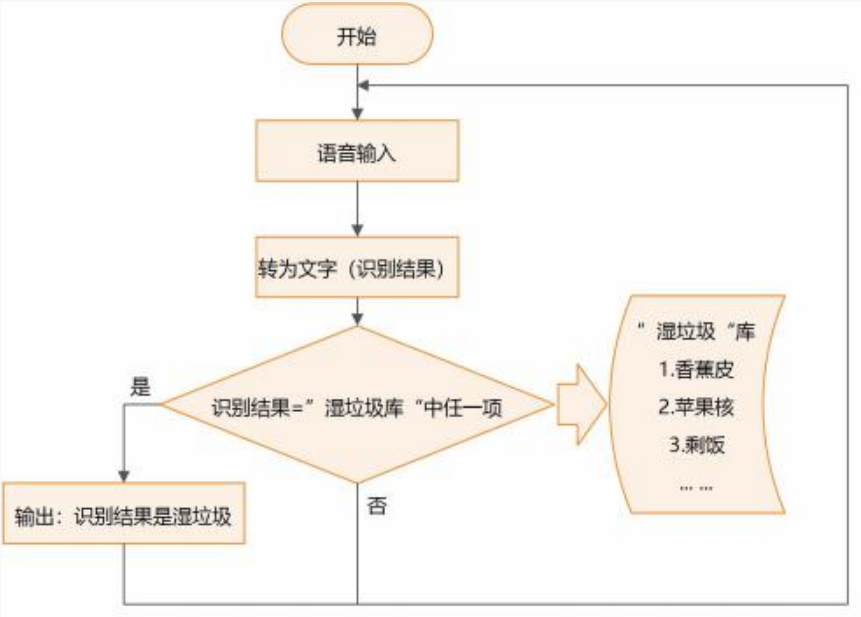


图 4.2.5 流程图 2

STEP2: 编写程序。



图 4.2.6 执行程序-1



图 4.2.6 执行程序-2

运行结果：按下 A 键，进行语音输入。根据语音转换结果，对应显示垃圾种类。

舞台界面：



图 4.2.7 舞台界面

7. 麦克小发明

这里只是 AI 助力垃圾分类的基本功能，还可以自己进行改进，如果你还有更多的创意，赶紧动手尝试吧！



外观设计

程序设计

项目三 风帆车

1. 麦克的故事

在新西兰的海边，有一项刺激的冒险运动——风帆车。风帆车实际上是一种卡丁车，在此基础上加上了一块帆，起风的时候，它可以借助风力极速前进，乐趣十足。



在这个项目中，将制作一辆风帆车，并探究如何让它跑的更远。

先来看看要用到哪些硬件吧！

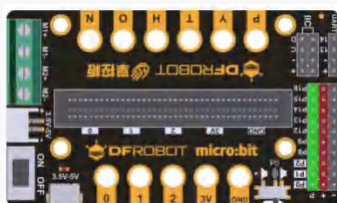
2. 麦克的布兜



1× micro:bit 主控板



1× micro USB 连接线



1× micro:bit 扩展板



2× 减速电机



1× 电池盒



2× 智能灰度

3. 小麦的愿望：制作小车



我想自己动手设计一辆
小车，请发挥自己的想象力
哦，下面会提供部分参考~

4. 小克的黑板

为风帆车制作一个小车底座，可以采用四轮车的结构，也可以是三轮车。



制作时可以使用平时生活中常见的各种材料，比如轮子可以用饮料瓶盖。



需要注意的是，车轮一定要可以顺滑的转动，否则车轮卡的比较紧的话，即使有风，小车也很难前进。

5. 小麦的愿望：制作风帆车



在做好的小车的基础上进行改造，期待制作好的风帆车哦~

6. 小克的黑板

为小车加上一面帆，小车就可以在风中行驶啦！

可以将纸或者布剪出三角形、四边形或者其他形状作为帆，然后固定在小车上，完成风帆车的制作。



7. 小麦的愿望：制作风扇



在做好的风帆小车上加上风扇，这样我们的风帆小车就完成了哦~

8. 小克的黑板

从前面的分析我们知道，在制作扇叶时，要让扇叶弯曲一定的弧度，这样才能产生向前的风！此项目中使用电机为风扇提供动力。

STEP1: 将传感器连接到扩展板上。

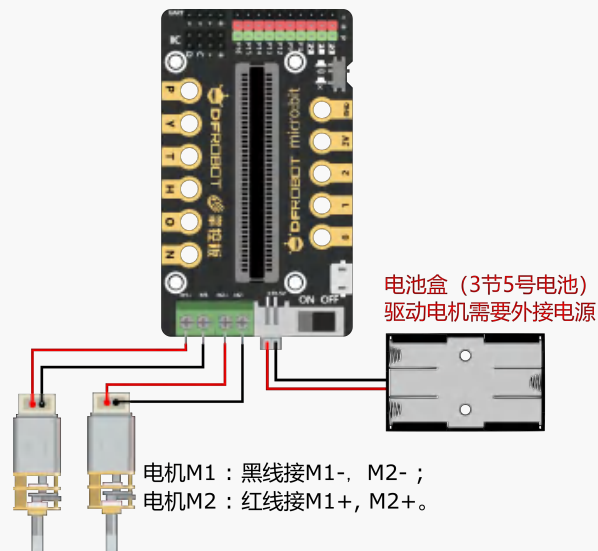
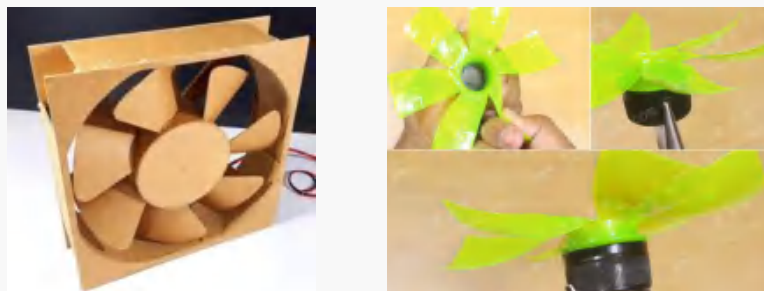


图 4.3.1 连线示意图

STEP2: 扇叶制作

扇叶的制作方法有很多，比如使用纸片、塑料瓶等。注意将电机固定到风扇上，并搭建一个牢固的底座。

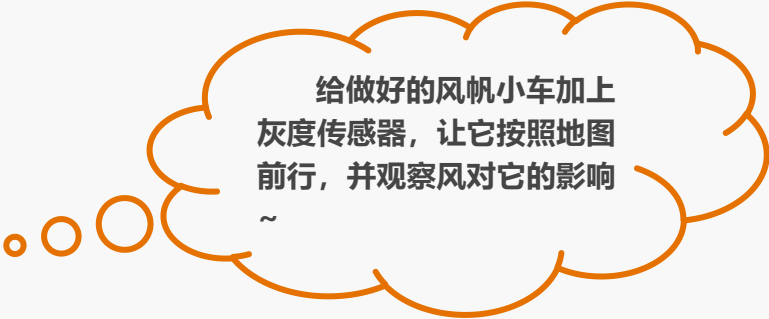


在实际操作时，由于电机转速不够，产生的风力会比较小，有什么办法可以加大风力吗？

或许可以考虑做一些提高风扇转速的装置，比如齿轮加速、滑轮加速等；或者改善扇叶，适当增加扇叶数量、增加弯曲程度。大家不妨动手试试吧！

打开做好的风扇，让你的风帆车在风中驰骋！注意调整帆的角度，获得更大的风力！

9. 小麦的愿望：“循规蹈矩”的风帆车



10. 小克的黑板

STEP1: 在上个愿望的基础上，将两个灰度传感器连接到扩展板上。

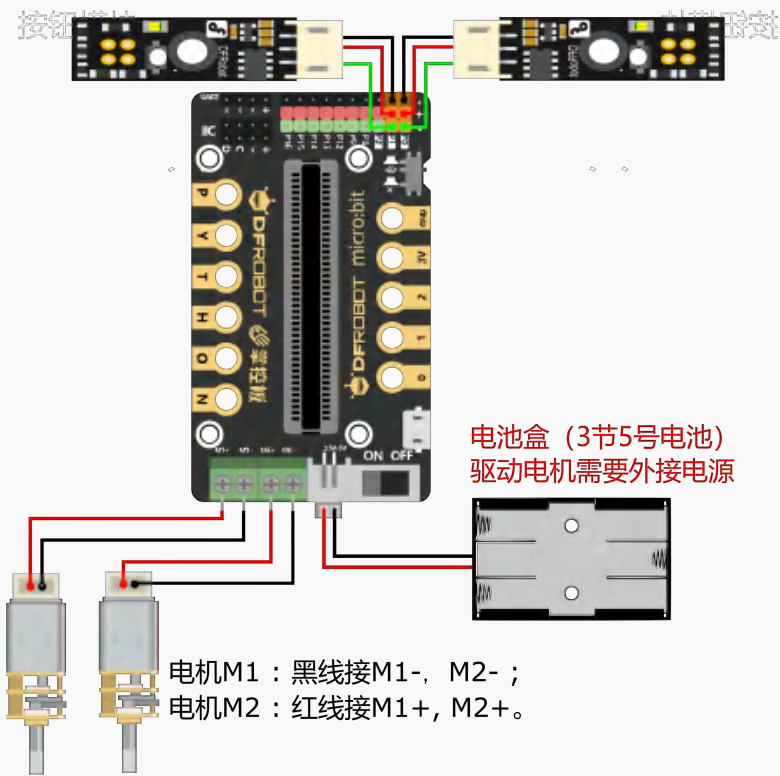


图 4.3.2 连线示意图

STEP2: 地图制作

地图的制作方法有很多，比如直接用图片打印出合适大小的纸张，也可以用黑胶带贴在白色的桌子上

进行物理识别运动



在实际操作时，由于光线以及地图材质的问题影响，可能会导致小车失去目标，这时请进行校准。

灰度传感器校准方法：

- 1.校准所要巡的线(比如黑线):将传感器探头放在黑线上方，**单击按键**后，待 LED 闪烁完毕后完成校准；
- 2.校准背景色(比如白色背景)：将传感器探头放在白色背景上方，**双击按键**后，待 LED 闪烁完毕后完成校准；
- 3.如果快速单击 3 次，LED 闪烁完毕后，**则会反转有效电平状态，即原来检测黑线输出为 0 现在变为输出为 1；**
- 4.快速单击 4 次，LED 反转亮和灭，原本检测黑线为亮变为检测白色背景为亮。

STEP3：读取灰度传感器数值

上面的校准方法会有用户产生疑问，LED 的颜色与电平都可进行状态反转，怎么判断呢？我们可以借助我们 Mind+ 软件的串口打印来进行实时判断！



图 4.3.3 灰度传感器当前状态

灰度传感器识别到的两种状态，具体实物如下：

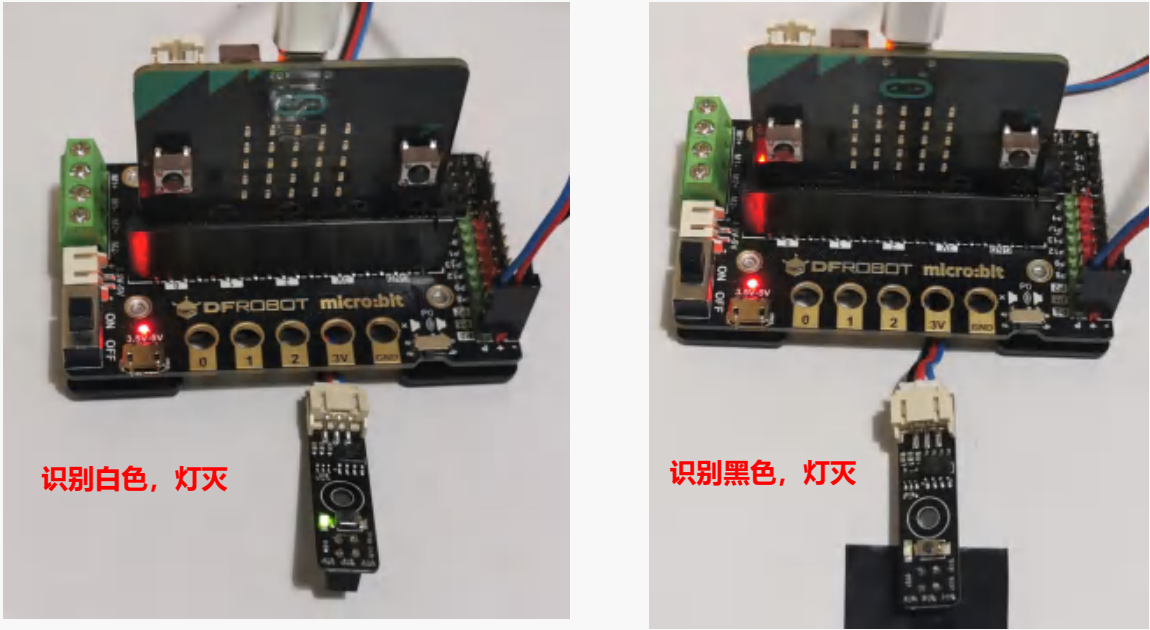


图 4.3.4 两种状态实物展示



图 4.3.5 识别到白色串口输出



图 4.3.6 识别到黑色串口输出

11. 麦克小发明

这个项目带给我们几个问题：

- 1、我们如何延长风帆车行驶的距离呢？
- 2、风帆车能否逆行航行呢？

这些问题都需要我们通过实验探究，如果你还有更多的创意与疑问，赶紧动手尝试吧！



外观设计	程序设计
	

项目四 无人漫游车

1. 麦克的故事

2011 年 11 月，总造价高达 25 亿美元的好奇号火星车发射升空。在砾石遍布的火星表面，带着探索火星生命元素的使命，好奇号无人漫游车每天都在缓慢移动着、寻找着。



在这个项目中，我们也将制作一台无人漫游车，并让它能够自行躲避障碍。

一般的避障小车会将超声波固定在车头检测障碍，当有障碍物的时候，就自动转左或转右，其实就是让小车一直向前“看”。在这个项目中，为了让小车更加“智能”，将会让小车向四周“看”，检测周围的障碍物，根据结果“智能”选择路线。

先来看看要用到哪些硬件吧！

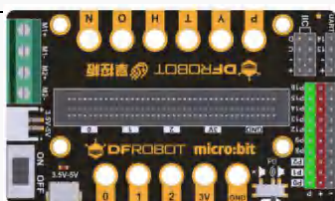
2. 麦克的布兜



1× micro:bit 主控板



1× micro USB 连接线



1× micro:bit 扩展板



2× 减速电机



1× 超声波传感器



1× 舵机

3. 小麦的愿望：雷达扫描



让小车在行驶的过程中
能够 180 度检测周围的障碍
物~

4. 小克的黑板

STEP1: 将传感器连接到扩展板上。

舵机带动超声波 0-180 转动，检测障碍。

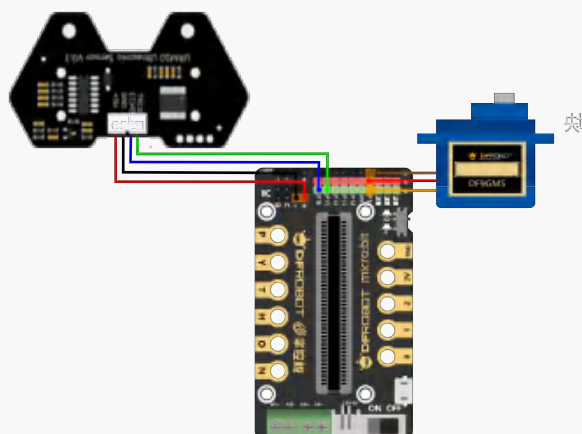


图 4.4.1 连线示意图

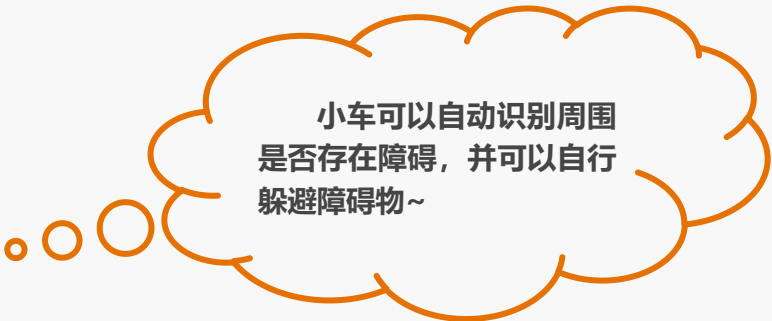
STEP2: 执行程序

将超声波固定在舵机上（可用胶带或胶枪固定），超声波随着舵机来回转动，在 5-60cm 范围内进行检测。



图 4.4.2 执行程序

5. 小麦的愿望：“智能”规划路线



6. 小克的黑板

STEP1: 将传感器连接到扩展板上。

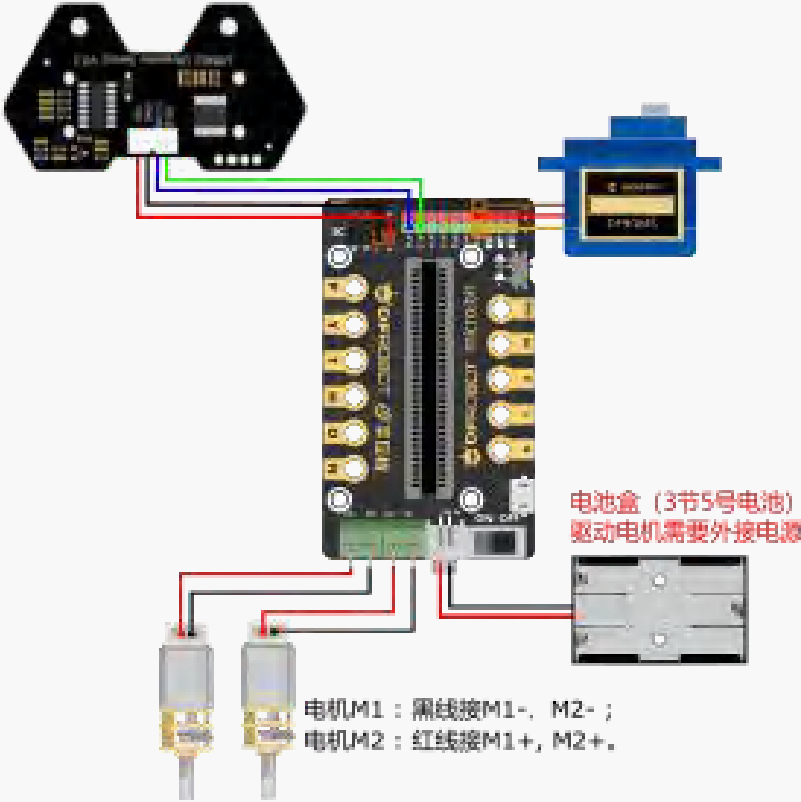


图 4.4.3 连线示意图

STEP2: 执行程序

在前一个愿望中，已经分析过小车行走的逻辑。也留下了一个问题：**如何判断区间 1、2、3 是否有障碍呢？**

从下面这张图可以看出,当 micro:bit 屏幕面向小车行驶前方时, **区间 1 对应舵机 0-60 度**, **区间 2 对应舵机 60-120 度**, **区间 3 对应舵机 120-180 度**。



因为在上面程序中设定**舵机每次转动 6 度**, 所以一个区间中, **舵机会转动 10 次**, 即**超声波进行 10 次检测**。为了让障碍判断的更加准确, 可以让**舵机进行一次来回扫描**。当在一个区间中, 超声波来回超过 10 次检测到有障碍, 可以大致确定这个方位一定是有障碍的。

将这种思路带到编程中, 可以设定 **3 个变量**, 分别累计 3 个区间超声波检测的情况, 当超声波检测到一次障碍, 变量加一, 来回扫描一遍后, 当这个变量大于 10, 则认为这个区间是有障碍的。

结合小车**优先转左**的行走逻辑, “智能” 选择路线程序如下:

* 注意在 Mind+ 中加载扩展板。



图 4.4.4 障碍检测



图 4.4.5 雷达扫描

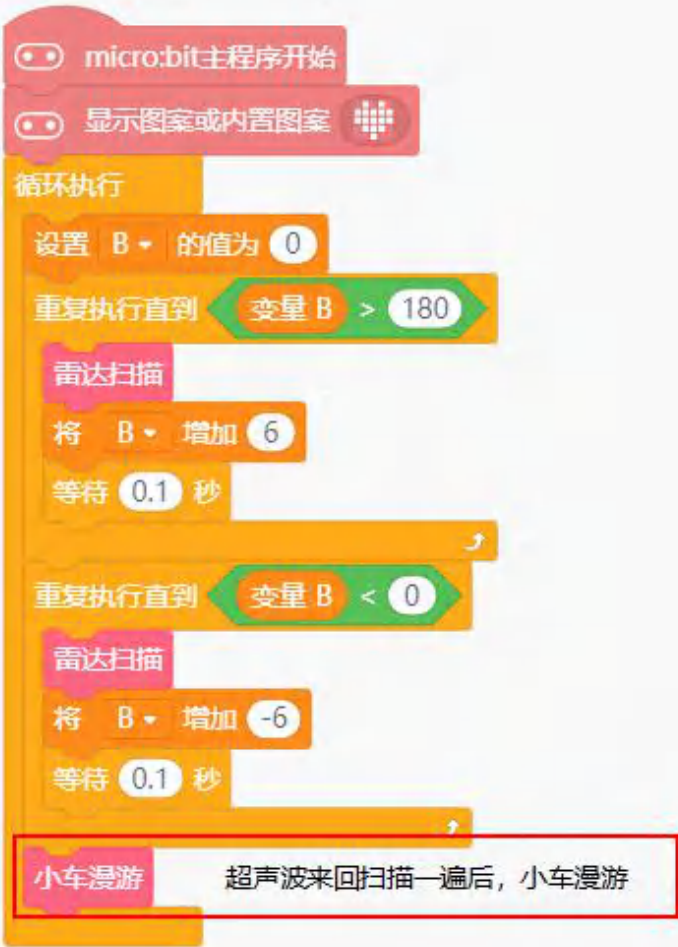


图 4.4.6 主程序

定义 小车漫游

串口 字符串输出 + 合并 0 区间1: 0 变量 区间1 不执行 +

串口 字符串输出 + 合并 0 区间2: 0 变量 区间2 不执行 +

串口 字符串输出 + 合并 0 区间3: 0 变量 区间3 执行 +

设置 P8 + 引脚伺服舵机为 90 度

等待 1 秒

如果 变量 区间1 < 10 那么执行

电机 M1 + 以 60 速度 反转 +

电机 M2 + 以 60 速度 正转 +

如果 变量 区间2 < 10 那么执行

电机 全部 + 以 60 速度 正转 +

等待 2 秒

如果 变量 区间3 < 10 那么执行

电机 全部 + 以 60 速度 反转 +

等待 2 秒

电机 M1 + 以 60 速度 正转 +

电机 M2 + 以 60 速度 反转 +

等待 2 秒

电机 全部 + 以 60 速度 正转 +

等待 4 秒

电机 M1 + 以 60 速度 反转 +

电机 M2 + 以 60 速度 正转 +

设置 超声波 - 的值为 15 读取超声波传感器距离 单位: 厘米 + trig为 P15 + echo为 P16 +

查看执行条件 变量 超声波 > 5 与 变量 超声波 < 80

设置 超声波 - 的值为 15 读取超声波传感器距离 单位: 厘米 + trig为 P15 + echo为 P16 +

串口 字符串输出 + 合并 0 超声波: 0 变量 超声波 执行 +

电机 全部 + 停止

接口 P9 + 播放音频 1 低C3 1 + 旧

显示图形或内置图案

设置 区间1 + 的值为 0

设置 区间2 + 的值为 0

设置 区间3 + 的值为 0

假如小车电机安装如下:



M2: 左轮 M1: 右轮

正转向前 反转向前

完成一次扫描后,舵机转到 90 度,超声波向前看

区间2无障碍, 小车直行

区间1无障碍, 小车右转

区间3无障碍, 小车左转

区间3有障碍, 小车后退调头

完成转向后, 小车直行

小车直行时, 超声波检测前方是否有障碍物, 没有则一直直行; 有则小车停止

图 4.4.7 小车漫游

7. 麦克小发明

这里只是无人漫游车的基本功能，在此基础上如果你还有更多的创意，赶紧动手尝试改进吧！



外观设计

程序设计





这些项目简直太有趣了! micro:bit 的功能真强大!

其实 micro:bit 和 Mind+ 还有很多很多功能, 就等着你慢慢去挖掘啦!

小克老师希望你的 micro:bit 之旅不会止步, 用你的奇思妙想, 玩出更多新颖有创意的作品。



我会一直保持着创造的热情继续探索好玩的游戏的!

可是如果以后我有关于 micro:bit 和 Mind+ 的问题怎么办呢?

如果以后你有什么疑问或者又有好的点子愿意和小克老师分享的话, 可以去 **DF 创客社区**: www.dfrobot.com.cn 里找到我哦! 那里还有很多和你兴趣相投的小伙伴呢! 希望可以在论坛里看到有关你成长的点点滴滴!



哇! 太棒了, 希望我能碰到更多有趣的同学, 我们论坛再见!