

Gravity : Mind+趣味学习 套件教程(掌控板)



目录

第一章 有用的小玩意	5
项目一 电子蜡烛	5
核心知识点	5
规划方案	7
功能实现	9
评价与反思	13
项目二 自动门	14
核心知识点	14
规划方案	14
功能实现	16
评价与反思	20
项目三 音乐盒	21
核心知识点	21
规划方案	21
功能实现	23
评价与反思	26
项目四 自平衡仪	27
核心知识点	27
规划方案	28
功能实现	30
评价与反思	33
第二章 好玩的小游戏	34

项目五 投篮机	34
头脑风暴：如何构建一台好玩的投篮机？	35
规划方案	38
功能实现	40
评价与反思	45
附件-完整参考程序	46
项目六 火线冲击	47
头脑风暴：如何让火线冲击变得好玩？	47
核心知识点	48
规划方案	51
功能实现	53
评价与反思	60
项目七 答案之书	61
头脑风暴	61
规划方案	62
功能实现	64
评价与反思	69
项目八 坦克大战	70
头脑风暴	70
核心知识点	70
规划方案	73
功能实现	74

评价与反思.....	82
第三章 让生活更加智能.....	83
项目九 感应垃圾桶.....	83
头脑风暴：你的感应垃圾桶需要具备哪些功能？	83
核心知识点.....	84
规划方案.....	85
功能实现.....	87
评价与反思.....	92
项目十 AI 助力垃圾分类.....	93
核心知识点.....	94
功能实现.....	95
评价与反思.....	100
项目十一 宠物投食机.....	101
头脑风暴：宠物投食机需要具备哪些功能？	101
核心知识点.....	102
规划方案.....	103
功能实现.....	105
评价与反思.....	113
项目十二 植物监测仪.....	114
头脑风暴：植物监测仪需要解决植物生长过程中的哪些问题？	114
核心知识点.....	115
规划方案.....	115

功能实现·····	118
评价与反思·····	124
项目十三 智能婴儿床·····	125
核心知识点·····	125
规划方案·····	126
功能实现·····	127
第四章 小车跑啊跑·····	137
项目十四车流量统计·····	137
规划方案·····	138
功能实现·····	140
项目十五 风帆车·····	147
头脑风暴·····	147
核心知识点·····	147
规划方案·····	149
功能实现·····	151
评价与反思·····	156
项目十六 无人漫游车·····	157
头脑风暴·····	157
核心知识点·····	158
规划方案·····	159
功能实现·····	163
评价与反思·····	169

第一章 有用的小玩意

项目一 电子蜡烛

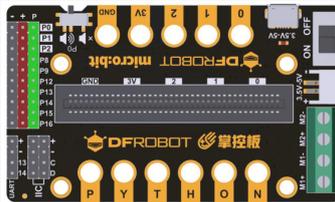
新闻上讲的温室效应好恐怖啊~原来人类的任何活动都可能造成碳排放，比如烧火做饭之类的...

过生日的时候，为了不污染环境，能不能制作一个电子蜡烛呢？

核心知识点

一、micro:bit 掌控两用扩展板

micro:bit 掌控两用扩展板完全兼容掌控板和 micro:bit 主板。



使用方法

掌控板	micro:bit 主板
OLED 屏朝扩展板“掌控板”图标方向	LED 点阵朝扩展板“micro:bit”图标方向
	

功能介绍

10 路 3Pin IO 口、两路 IIC 口（可外接颜色传感器等）、一路 UART 口（可外接 MP3 模块等）；

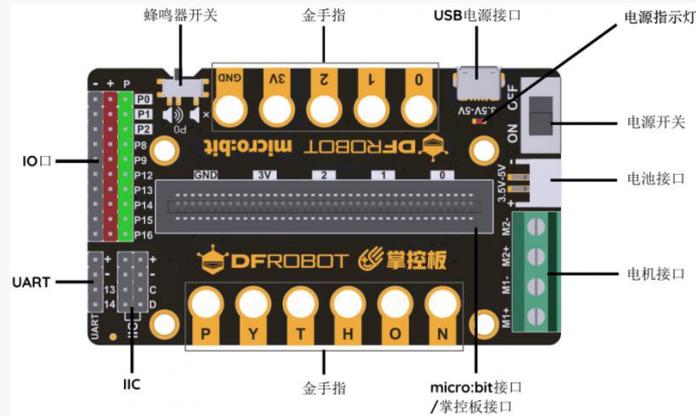
分别兼容掌控板金手指和 micro:bit 金手指；

板载两路电机接口，且不占用额外引脚；

板载 PH2.0 及 microUSB 两种供电口，既可以通过 usb 线供电，也可以通过电池盒或者锂电池，供电电压 3.5-5V，板载电源开关，可以打开或关闭供电电源；

板载一个带有开关的高品质蜂鸣器，使用掌控板的时候，不支持扩展板蜂鸣器，请将蜂鸣器关闭，此时可正常使用 P0 口。；

兼容乐高孔径，可以与乐高积木拼插结合。



供电方式

主板直接供电： 使用主板 USB 供电。此时，可以使用扩展板的各种扩展口及蜂鸣器。因主板驱动电流有限，无法使用电机驱动。

扩展板 USB 供电口供电： 使用电脑 USB 口或充电宝、手机充电头供电。此时，扩展板的所有功能均可使用。

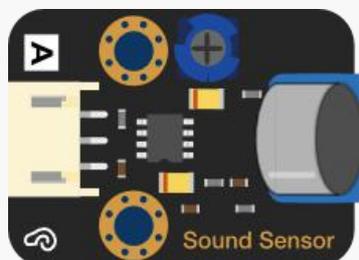
扩展板外接电池盒： 使用 PH2.0 接口的 3 节干电池盒或 3.7V 锂电池供电，输入电压为 3.5~5V。此时，扩展板的所有功能均可使用。

引脚说明

引脚	功能描述
P0	模拟/数字输入，模拟/数字输出
P1	模拟/数字输入，模拟/数字输出
P2	模拟/数字输入
P8	数字输入，模拟/数字输出
P9	数字输入，模拟/数字输出
P12	保留（掌控板不支持使用 P12 口）
P13	数字输入，模拟/数字输出
P14	数字输入，模拟/数字输出
P15	数字输入，模拟/数字输出
P16	数字输入，模拟/数字输出

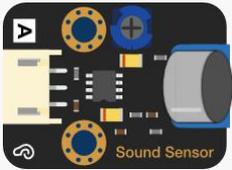
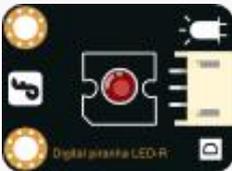
二、声音传感器

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。



规划方案

功能分解

分解功能	所需元件
<p>1、声音强度检测</p> <p>测量自己的声音多大,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~</p>	<p>声音传感器</p> 
<p>2、吹气蜡烛</p> <p>不吹气的时候小灯一直亮, 当用力吹小灯, 它就会灭, 就像蜡烛一样</p>	<p>LED 灯</p> 

构思外观

画出外观草图	所需耗材、工具
例如：	例如：牛皮纸、雪糕棒、透明胶等... 1、 2、 3、

问题记录

在后面项目实施过程中，可能会遇到各种各样的困难，尝试在下表中记录你遇到的问题和解决办法，便于以后出现类似问题时能更好的面对。

遇到的困难	你的解决方案
1、	1、
2、	2、
3、	3、
...	...

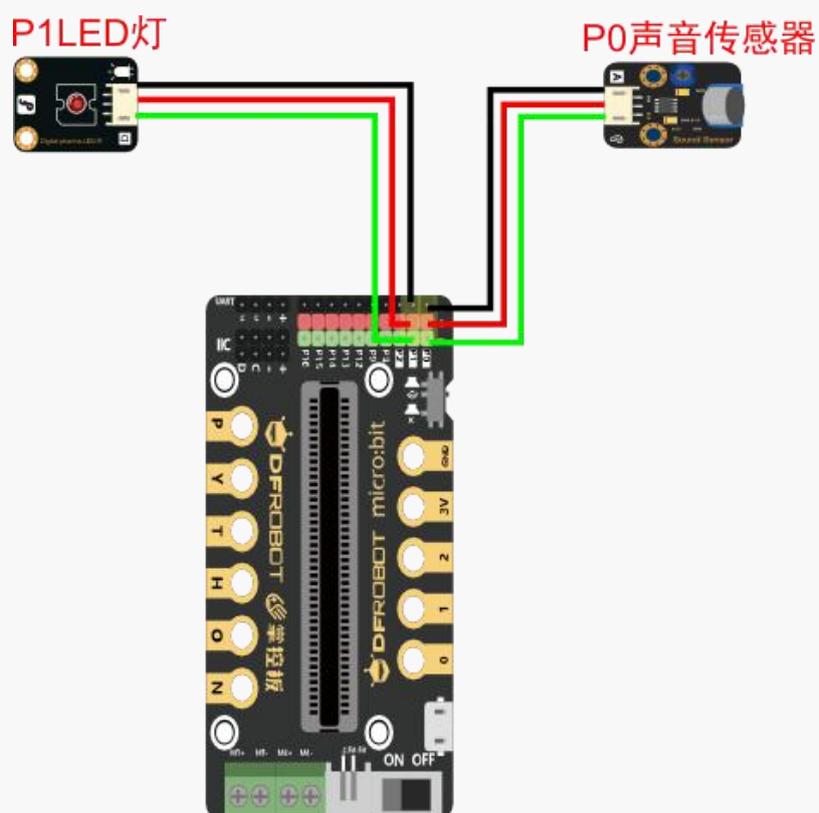
功能实现

功能 1：声音强度检测

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。

STEP1：连线图：

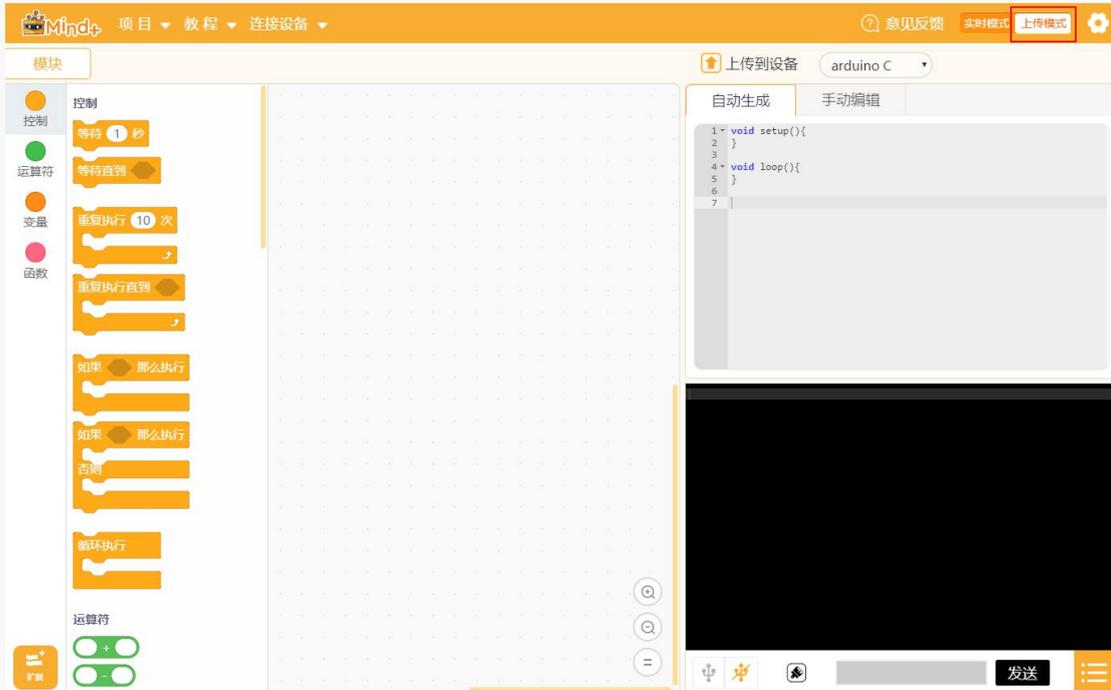


STEP2：编写程序：

作为课程第一个项目，需要进行一些软件的设置：打开 Mind+ 软件，选择“上传模式”，点击“扩展”，在“主控板”下，点击加载“掌控板”。

Mind+ 软件设置

- 1、打开 Mind+ 软件（1.5.5 及以上版本），选择“上传模式”。



2、打开“扩展”，在“主控板”下选择“掌控板”。



3、再次打开“扩展”，在“扩展板”下选择“micro:bit 掌控两用扩展板”。如果没有用到电机，这里可以不选择扩展板，用到电机的项目再选择即可。



* 每次编程时都要先完成以上软件设置，后面项目不再重复。



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。



①根据上一个小贴士的提示，我们可以进行初步的设计啦：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。



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（再次强调，模拟数值范围在 0—4095 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。

如下图，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。



功能 2：吹气蜡烛

不吹气的时候小灯一直亮，当用力吹小灯，它就会灭，就像蜡烛一样。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。

没有吹气时，也就是声音没有达到设定的强度，小灯呈现亮的状态，小灯对应的引脚应该为高电平；吹气时，如果声音达到了设定的强度，小灯呈现灭的状态，小灯对应的引脚应该为低电平。

编写程序：

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。



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。



更多创意

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~

评价与反思

与人分享快乐，能产生更多快乐，尝试与身边的人一起分享你的项目吧！你还可以在 DF 创客社区中分享作品，听听更多人的建议！社区网址：www.dfrobot.com.cn

问卷调查

尝试让体验过你的项目的人做下面的问卷表，看看在别人眼中这个作品的闪光点与不足。

你觉得我的作品好玩吗?	
你觉得最好玩的点在哪?	
你觉得哪里还有不足?	

自我反思

梳理自己设计项目的思维逻辑、项目在执行层面的问题和经验等等，可以帮助我们形成学习的闭环，加深在项目进行的过程中对知识点和内容的理解，找到可以改进和提高的地方。

优点与不足	1、
	2、
可以从项目中总结的经验	1、
	2、
学到的知识	1、
	2、

项目二 自动门

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！

核心知识点

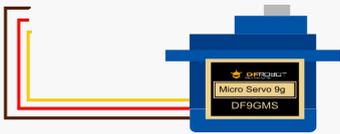
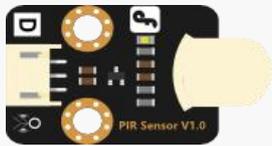
舵机

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~



规划方案

功能分解

分解功能	所需元件
<p>1、按钮控制舵机</p> <p>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 0°。</p>	<p>舵机</p> 
<p>2 运动传感器控制舵机</p> <p>当有人靠近时舵机转动一定角度，当人离开时舵机恢复之前的角度，实现开关门的效果。</p>	<p>运动传感器</p> 

构思外观

画出外观草图	所需耗材、工具
例如：	例如：牛皮纸、雪糕棒、透明胶等... 1、 2、 3、

问题记录

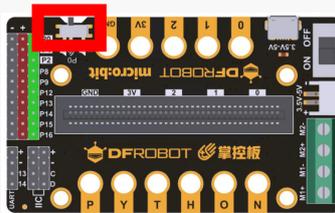
在后面项目实施过程中，可能会遇到各种各样的困难，尝试在下表中记录你遇到的问题和解决办法，便于以后出现类似问题时能更好的面对。

遇到的困难	你的解决方案
1、	1、
2、	2、
3、	3、
...	...

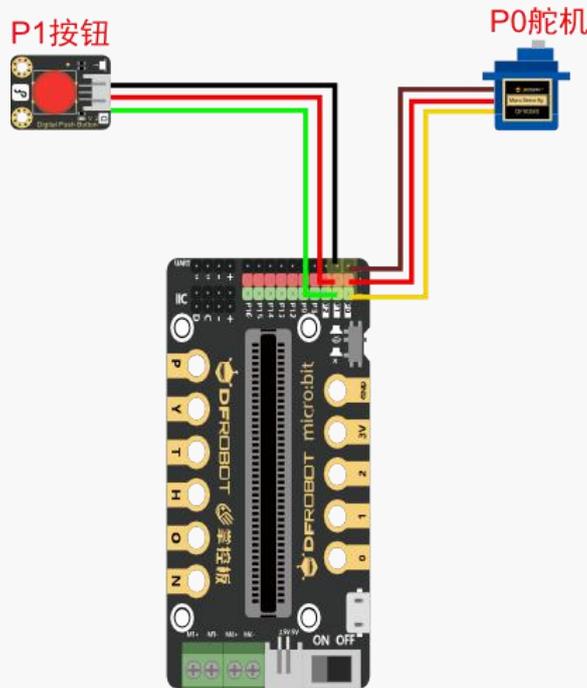
功能实现

功能 1：按钮控制舵机

我们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°(最好不要用 0 度，舵机会抖动，一般舵机用 10-170 度)。因为我们的扩展板 P0 引脚和蜂鸣器的控制是同一引脚，所以要用 P0 引脚做其它实验时，记得拨动开关关闭蜂鸣器声音，如下图红色标记处



STEP1：连线图



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？

如下图，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。



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。



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



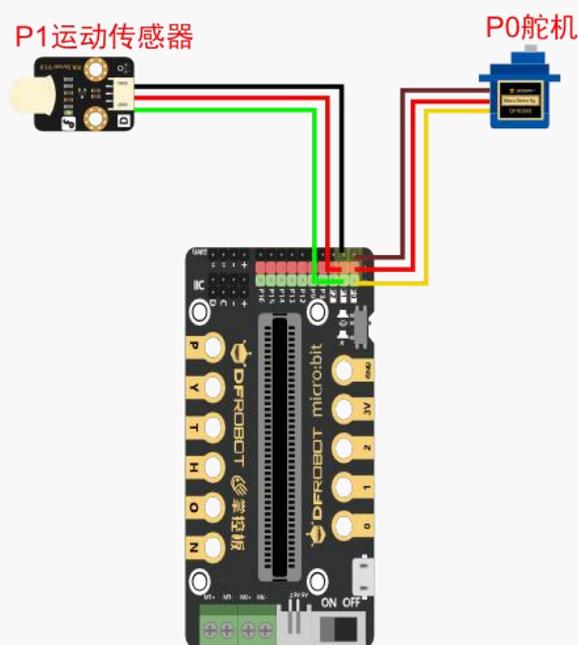
③最终程序:



功能 2：运动传感器控制舵机

当有人靠近时舵机转动一定角度，当人离开时舵机恢复之前的角度，实现开关门的效果。

STEP1：连线图



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。



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”。



③最终程序：



更多创意

除了做自动门，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

~

评价与反思

与人分享快乐，能产生更多快乐，尝试与身边的人一起分享你的项目吧！你还可以在 DF 创客社区中分享作品，听听更多人的建议！社区网址：www.dfrobot.com.cn

问卷调查

尝试让体验过你的项目的人做下面的问卷表，看看在别人眼中这个作品的闪光点与不足。

你觉得我的作品好玩吗？	
你觉得最好玩的点在哪？	
你觉得哪里还有不足？	

自我反思

梳理自己设计项目的思维逻辑、项目在执行层面的问题和经验等等，可以帮助我们形成学习的闭环，加深在项目进行的过程中对知识点和内容的理解，找到可以改进和提高的地方。

优点与不足	1、
	2、
可以从项目中总结的经验	1、
	2、
学到的知识	1、
	2、

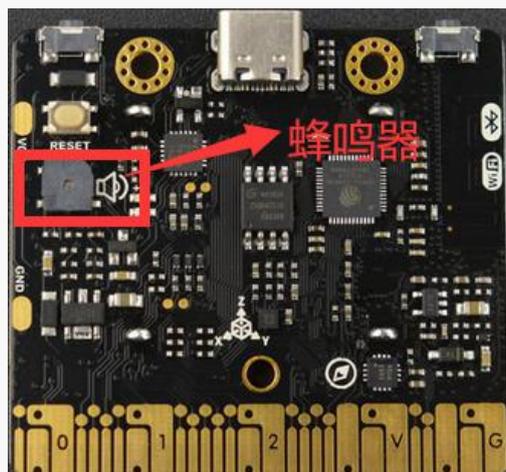
项目三 音乐盒

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。让我们来动手做一做吧！

核心知识点

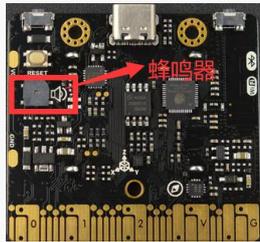
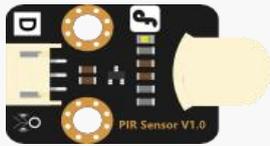
掌控板上的蜂鸣器

蜂鸣器是一种一体化结构的电子讯响器，可以发出不同音调的声音。



规划方案

功能分解

分解功能	所需元件
<p>1、播放音乐</p> <p>想要它播放出美妙的音乐！</p>	<p>掌控板上的蜂鸣器</p> 
<p>2、身体感应音乐</p> <p>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！</p>	<p>运动传感器</p> 

构思外观

画出外观草图	所需耗材、工具
例如：	例如：牛皮纸、雪糕棒、透明胶等... 1、 2、 3、

问题记录

在后面项目实施过程中，可能会遇到各种各样的困难，尝试在下表中记录你遇到的问题和解决办法，便于以后出现类似问题时能更好的面对。

遇到的困难	你的解决方案
1、	1、
2、	2、
3、	3、
...	...

功能实现

功能 1：播放音乐

STEP1: 连线，直接用 usb 对电脑和主板连接即可！



注意： 掌控板板载蜂鸣器使用 P6 引脚，使用蜂鸣器时，P6 引脚不能再用了哦。

STEP2: 编写程序

①将“掌控板”模块中的播放旋律模块“**播放音乐....重复播放一次**”，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**”。

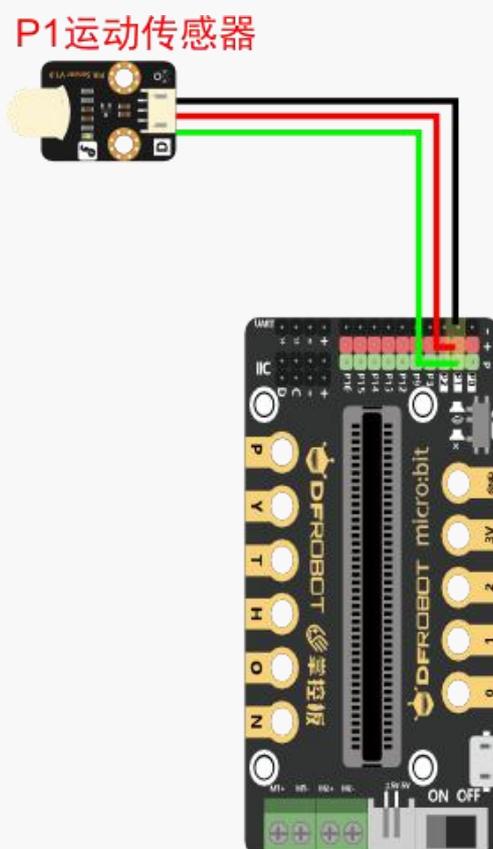


②最终程序。



功能 2：身体感应音乐

STEP1: 连线图



STEP2: 编写程序

- 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

— 闪 — 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“掌控板”模块集中的播放旋律模块“播放音符 1 低 C/C3 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）

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！



更多创意

除了做音乐盒，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

~

评价与反思

与人分享快乐，能产生更多快乐，尝试与身边的人一起分享你的项目吧！你还可以在 DF 创客社区中分享作品，听听更多人的建议！社区网址：www.dfrobot.com.cn

问卷调查

尝试让体验过你的项目的人做下面的问卷表，看看在别人眼中这个作品的闪光点与不足。

你觉得我的作品好玩吗？	
你觉得最好玩的点在哪？	
你觉得哪里还有不足？	

自我反思

梳理自己设计项目的思维逻辑、项目在执行层面的问题和经验等等，可以帮助我们形成学习的闭环，加深在项目进行的过程中对知识点和内容的理解，找到可以改进和提高的地方。

优点与不足	1、
	2、
可以从项目中总结的经验	1、
	2、
学到的知识	1、
	2、

项目四 自平衡仪

生活中使用手机时我们会发现，手机横着放在手里，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面有自平衡装置，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！

那我们一起用掌控板来探索这个神奇的功能吧！

核心知识点



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。

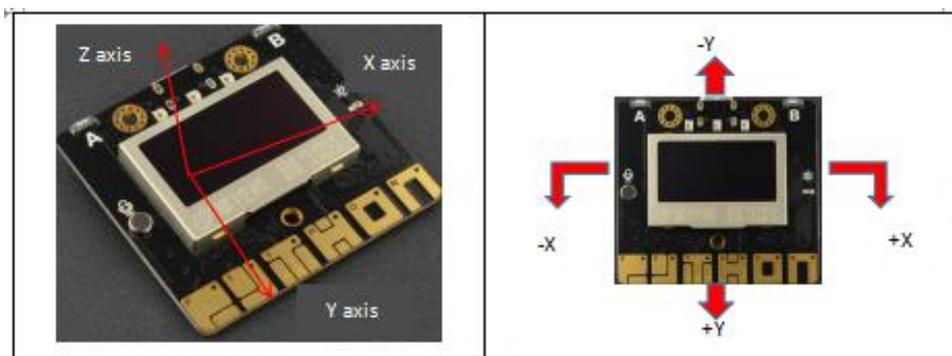
使用重力加速度可以做些什么呢？

掌控板自带加速度计，它能实时检测掌控板的姿态，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！



掌控板加速度计：

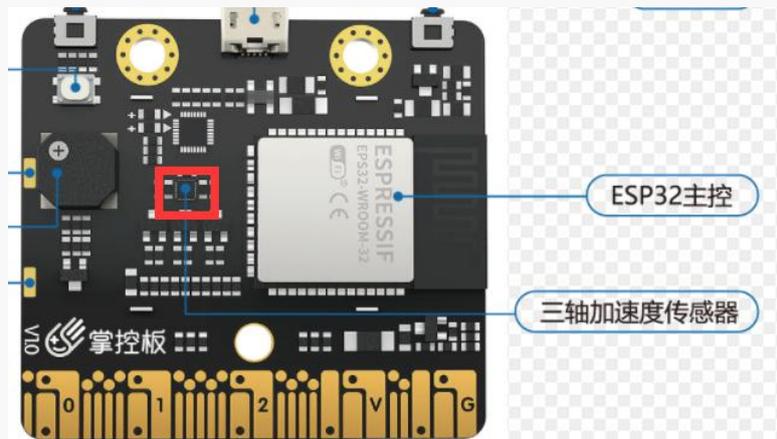
掌控板可以检测 X,Y,Z 三个方向的重力加速度，其中 X 沿着左右方向，Y 沿着前后方向，Z 垂直于板，沿着上下方向，如左下图。



三个方向

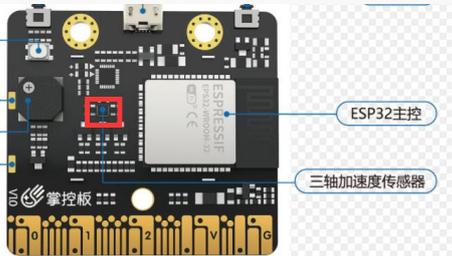
数值变化

掌控板上的三轴加速度传感



规划方案

功能分解

分解功能	所需元件
1、确定三轴加速度的方向 X,Y,Z 三个方向的重力加速度	掌控板上的三轴加速度 
2、三轴加速度控制舵机 我想要无论怎么旋转，小舵机上的指针永远垂直于上方~	舵机 

构思外观

画出外观草图	所需耗材、工具
例如：	例如：牛皮纸、雪糕棒、透明胶等... 1、 2、 3、

问题记录

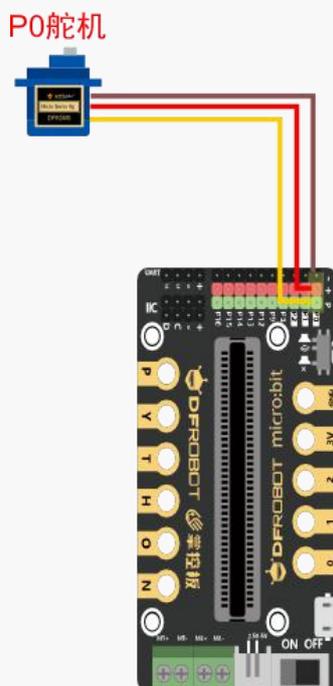
在后面项目实施过程中，可能会遇到各种各样的困难，尝试在下表中记录你遇到的问题和解决办法，便于以后出现类似问题时能更好的面对。

遇到的困难	你的解决方案
1、	1、
2、	2、
3、	3、
...	...

功能实现

功能：三轴加速度控制舵机

STEP1: 连线图



STEP2: 编写程序

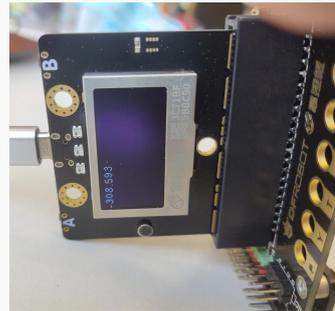
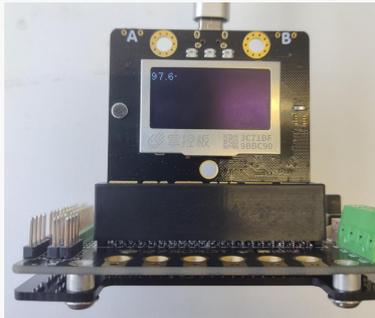
①通过“读取加速度的值”模块实现获取 X,Y,Z 轴的加速度数值。



可以通过下拉列表选择 X,Y,Z 轴哦~



②加速度传感器的数值范围在-1023~1023 之间变化，可以通过“屏幕显示文字”模块来查询变化范围哦，因为掌控板插上扩展板后是立起来的，所以我们选择 y 方向进行使用，更为方便。



③当掌控板向左倾斜时，y 方向会产生一个值，这个时候就需要舵机向右转动来抵消这个角度啦！

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°)。不同板子可能有些差异，本次实验的具体数据如下表所示，映射之后，当掌控板放在水平桌面时，舵机的角度就为 90 度啦。

舵机角度	0 度	90 度	180 度
掌控板加速度	1000	-27	-1000

在引脚中找到映射模块

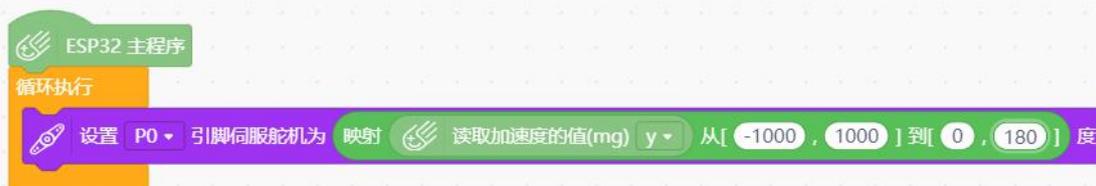


④在“扩展”中找到“执行器”模块，再找到“舵机模块”。

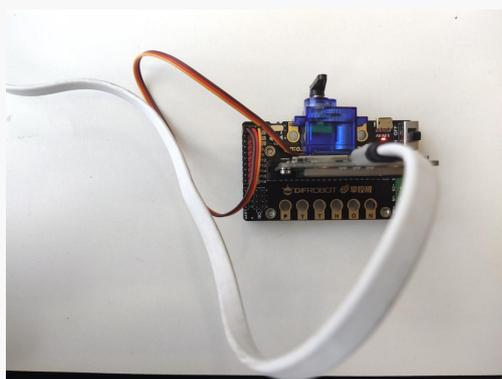


舵机模块

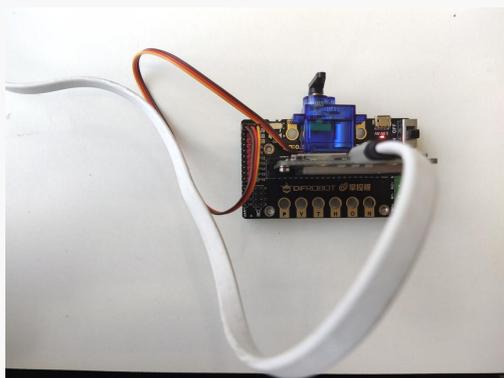
⑤选中“舵机模块”放入“循环执行”中，与“映射模块”嵌套如下：



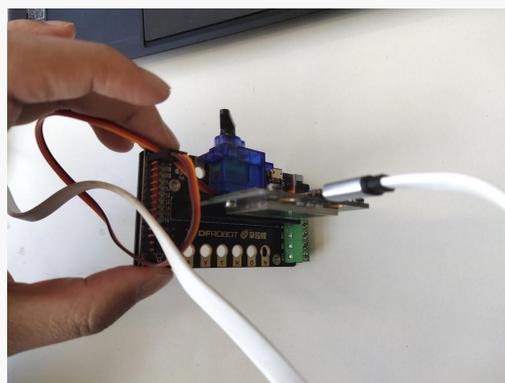
⑥最后进行结构制作，用双面胶将舵机和扩展板固定起来，使两者保持相对静止。



固定结构



水平静止



向右倾斜

更多创意

除了做自平衡仪，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~

评价与反思

与人分享快乐，能产生更多快乐，尝试与身边的人一起分享你的项目吧！你还可以在 DF 创客社区中分享作品，听听更多人的建议！社区网址：www.dfrobot.com.cn

问卷调查

尝试让体验过你的项目的人做下面的问卷表，看看在别人眼中这个作品的闪光点与不足。

你觉得我的作品好玩吗？	
你觉得最好玩的点在哪？	
你觉得哪里还有不足？	

自我反思

梳理自己设计项目的思维逻辑、项目在执行层面的问题和经验等等，可以帮助我们形成学习的闭环，加深在项目进行的过程中对知识点和内容的理解，找到可以改进和提高的地方。

优点与不足	1、
	2、
可以从项目中总结的经验	1、
	2、
学到的知识	1、
	2、

第二章 好玩的小游戏

项目五 投篮机

你最喜欢玩的游戏是什么？

相信每个人都有自己的答案，不管是小时候和大人去赶集玩过的套环、飞镖射气球，还是学生时代的篮球、足球，又或者是现在风靡的各类电子游戏、电玩城游戏。在玩游戏的时候，时间总是过得那么快，又那么快乐。

你有没有想过：你为什么会觉得这个游戏好玩？如果让你设计一款游戏，你要如何让它变得好玩呢？

其实**游戏设计的秘诀**主要有两点：

- 1、**简单的反馈机制**：玩家行为可以快速获得反馈。
- 2、**成就感**：设定目标，玩家在达成目标时会获得成就感、满足感。

比如扔沙包游戏，扔中敌方就是进攻成功，接到敌方的沙包就是防守成功，这就是最简单的反馈；躲过敌方的沙包攻击或者用沙包打中敌方，都会获得成功的快乐，这就是成就感。

从这两点出发，思考一下你喜欢玩的游戏在这两个特征上是不是有特别的优势呢？尝试将它总结出来。

在这个项目中，我们将从上面两点秘诀去理解游戏设计，并设计一款好玩的投篮机。



头脑风暴：如何构建一台好玩的投篮机？



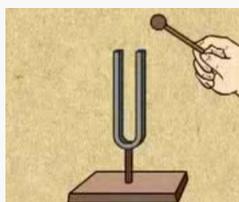
问题清单

- 1、投篮机是怎么玩的？
- 2、投篮机好玩的点在哪里？尝试从反馈机制和成就感两个方向各举例说明。
- 3、关于投篮机，你还有更好玩的想法吗？
- 4、你的想法具备可实现性吗？需要哪些传感器或执行器？
- 5、你的投篮机有怎样的外观？

超声波传感器

什么是超声波？（结合物理学科）

物理课堂上讲过声音是由物体振动产生的。比如说话时声带会动，打鼓时鼓面在振动。发声物体在一秒钟之内振动的次数叫做声音的频率，单位是赫兹（Hz）。

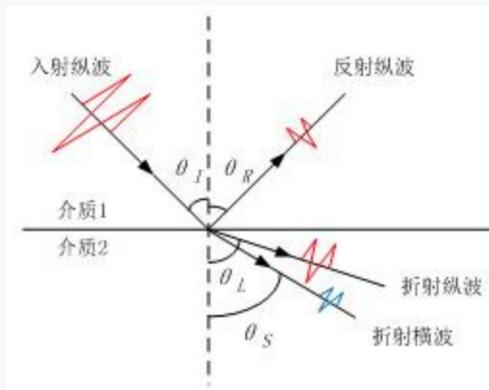


声音作为一种波，人耳可以听到频率为 20HZ-20KHZ，超过 20KHZ 称为超声波，低于 20HZ 的称为次声波。

声波类型	频率
次声波（超低声）	小于20Hz
可听声	20Hz~20kHz
超声波	20kHz~1GHz
特超声（微波超声）	大于1GHz

超声波的折射与反射

超声波可以在气体、液体及固体中传播，传播速度依次加快。超声波有折射和反射现象，并且在传播过程中有衰减。



超声波优点

- 频率高、波长短、绕射现象小；
- 方向性好、能够定向传播；
- 碰到杂质或分界面会产生显著反射、形成反射波的特点。

超声波传感器

利用超声波的优点，可做成各种超声传感器，在通讯、医疗、家电等领域得到广泛应用。

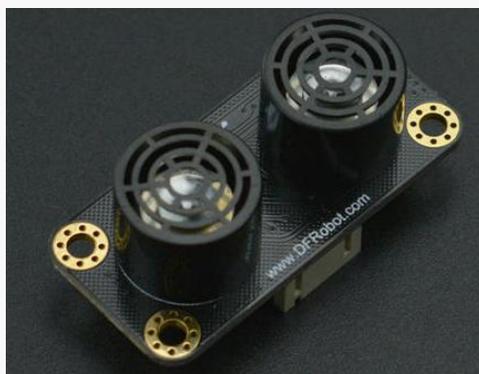
超声波传感器一般由发送传感器(或称波发送器)、接收传感器(或称波接收器)、控制部分与电源部分组成，有的超声波传感器既作发送、也能作接收。

超声波测距工作原理：发射器发出的超声波，遇到障碍物发生反射，接收器接收到反射波，从而使传感器检测到障碍物。



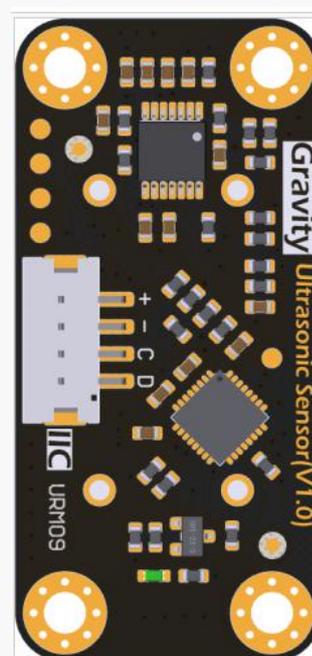
URM09-I²C 超声波测距传感器

在掌控板入门套件中，我们使用了 Urm10 超声波，探测距离为 2cm ~ 500cm (可设置，最大量程 500CM)。



通过此模块，可以获得在探测范围内的确切的障碍物的距离。

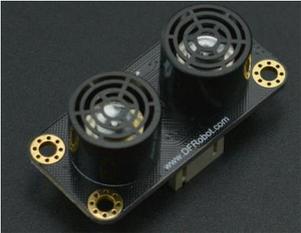
引脚	引脚说明
VCC	电源输入 (3.3V-5.5V)
GND	电源地
C	I2C 时钟线 SCL
D	I2C 数据线 SDA



思考一下：利用超声波可以实现投篮机的什么功能呢？

规划方案

功能分解

分解功能	所需元件
<p>1、判断篮球是否投中</p> <p>将超声波传感器安装在篮球框附近，将篮球作为障碍物，每当篮球投中篮筐，超声波传感器检测到障碍物，则判断篮球投中。</p>	<p>超声波(探测距离)</p> 
<p>2、判断游戏是否开始</p> <p>就像电玩城需要投币才能开始游戏一样，我们还需要一个开关装置来判断游戏是否开始，当开关被触发时，开始游戏。</p> <p>这个开关可以是按钮、角度传感器，甚至是遥控器！</p>	<p>Eg1.按钮 (数字开关，按下为 1，松开为 0)</p>  <p>Eg2.模拟角度传感器 (模拟开关,)</p>  <p>Eg3.红外遥控器 (遥控设备，与红外接收管配合使用) ...</p> 
<p>3、更多好玩的功能：例如倒计时闯关功能，在规定时间内，投中数量要达到最低要求，才能继续游戏！</p>	
<p>4、你的创意：</p>	

构思外观

画出外观草图	所需耗材、工具
例如：	例如：牛皮纸、雪糕棒、透明胶等... 1、 2、 3、

问题记录

在后面项目实施过程中，可能会遇到各种各样的困难，尝试在下表中记录你遇到的问题和解决办法，便于以后出现类似问题时能更好的面对。

遇到的困难	你的解决方案
1、	1、
2、	2、
3、	3、
...	...

功能实现

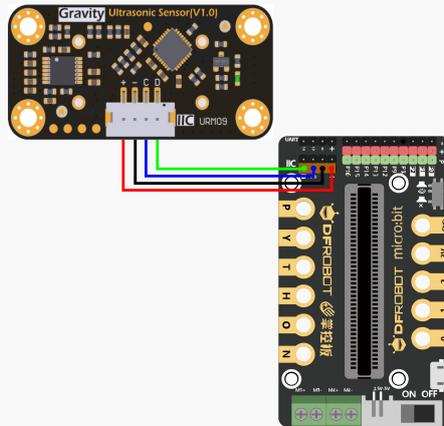
功能 1：超声波判断篮球是否投中

连线说明：扩展板上的 10 路 IO 口分为黑、红、绿 3 种颜色，功能如下表。

黑色	GND 引脚
红色	电源引脚
绿色	信号引脚

* 所有 GND 引脚或电源引脚可互相通用，信号引脚与标注的 IO 号一一对应。

超声波连线图：C→C、D→D、+→+、-→-引脚



* 这里要将掌控板插入扩展板中（带有 OLED 屏的一面，朝向扩展板上掌控板图标），为了便于看清连线，在图中画未出掌控板。后面连线图都默认连接掌控板，不再重复说明。

Mind+ 软件设置

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”。



测试程序：串口读取超声波距离,注意超声波传感器在使用前需要初始化。



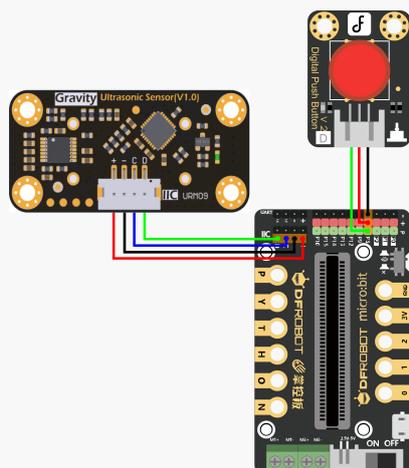
功能程序：超声波每次检测到障碍物，算作进球一次，将得分实时显示在屏幕上



功能 2：按下按钮，开始或结束游戏

这里以按钮为例，作为游戏开关。

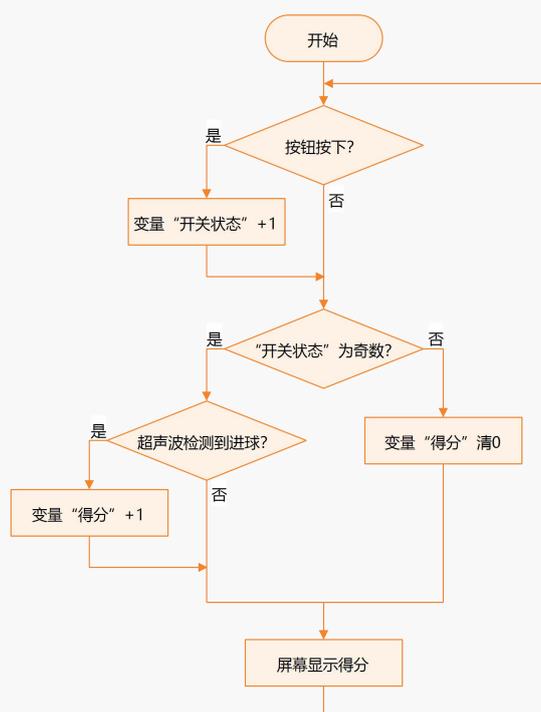
连线图：按钮→P8



按钮作为数字传感器，按下为 1、松开为 0，将按钮作为游戏开关时，需每按下一次按钮就切换游戏开始或结束的状态。如何实现呢？

根据掌控板裸板课程项目五 心情灯的学习，我们知道只需要在编程时加入一个变量记录按钮按下的次数即可，按下次数为奇数表示游戏开始，为偶数表示游戏结束。对奇数和偶数的判断可以通过将该数除以 2 求取余数的办法，余数为 1 则为奇数，余数为 0 则为偶数。

流程图分析：



功能程序：

ESP32 主程序

初始化I2C超声波传感器地址为 0x20

设置 开关状态 的值为 0 初始化超声波及变量

设置 得分 的值为 0

循环执行

如果 读取数字引脚 P8 那么执行 判断按钮是否按下

将 开关状态 增加 1

等待 1 秒 按钮消抖

如果 变量 开关状态 除以 2 的余数 = 1 那么执行

设置 超声波 的值为 读取I2C超声波传感器距离(cm)

如果 变量 超声波 > 2 与 变量 超声波 < 10 那么执行

将 得分 增加 1 按钮按下基数次表示开始游戏

等待 0.5 秒

否则

设置 得分 的值为 0 按钮按下偶数次表示结束游戏

屏幕显示文字 得分: 在坐标 X: 0 Y: 24 预览

屏幕显示文字 变量 得分 在坐标 X: 48 Y: 24 预览 可以加几个空格刷新分数

构造外观

示例图：

外观材料：牛皮纸、一次性纸杯、废纸

工具：白乳胶、透明胶

安装在篮筐下方的超声波



纸杯做的篮筐



掌控板显示屏



废纸团成的篮球



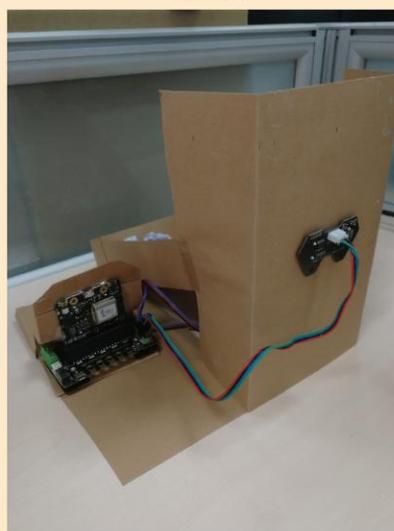
按钮开关



侧面图



背面图



更多创意

通过上面两个案例，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！

在前面提到的“倒计时闯关功能，在规定时间内，投中数量要达到最低要求，才能继续游戏”也是一个不错的想法，实现程序可以参考掌控板裸板课程项目十二 电子秒表，通过获得系统时间进行倒计时判断。

评价与反思

与人分享快乐，能产生更多快乐，尝试与身边的人一起分享你的投篮机吧！你还可以在 DF 创客社区中分享作品，听听更多人的建议！社区网址：www.dfrobot.com.cn

问卷调查

尝试让玩过你投篮机的人做下面的问卷表，看看在别人眼中这个作品的闪光点与不足。

你觉得我的投篮机好玩吗？（1-5 分）	
你觉得最好玩的点在哪？	
你喜欢我搭建的外观吗？（1-5 分）	
你觉得哪里还有不足？	

自我反思

梳理自己设计项目的思维逻辑、项目在执行层面的问题和经验等等，可以帮助我们形成学习的闭环，加深在项目进行的过程中对知识点和内容的理解，找到可以改进和提高的地方。

你的投篮机的优点与不足	1、
	2、
可以从项目中总结的经验	1、
	2、
学到的知识	1、
	2、

附件-完整参考程序

在上面项目中未实现投篮机中屏幕文字提示部分，比如游戏开始、游戏结束等。

完整参考程序如下。

The image displays four blocks of code for an ESP32 program, likely using a visual programming language like Scratch. The code is organized into several sections:

- ESP32 主程序 (Main Program):**
 - 初始化 (Initialization):** A loop that reads digital pin P8. If it is high, it increments the switch state, sets a flag to 1, and waits 1 second.
 - 循环执行 (Loop Execution):** Checks if the switch state divided by 2 has a remainder of 1. If true, it starts the game. Otherwise, it ends the game and reinitializes.
- 定义 初始化 (Define Initialization):**
 - Initializes the I2C ultrasonic sensor address to 0x20.
 - Sets switch state, score, and flag to 0.
 - Clears the screen and displays "按下按钮开始游戏" (Press button to start game) at X:0, Y:22.
 - Waits until pin P8 is read.
- 定义 游戏开始 (Define Game Start):**
 - Clears the screen and displays "快速投篮" (Quick shot) at X:32, Y:24.
 - Sets the flag to 0 and waits 1 second.
- 定义 超声波计数 (Define Ultrasonic Counting):**
 - Reads the I2C ultrasonic sensor distance (cm).
 - If the distance is between 2 and 10 cm, it increments the score and waits 0.5 seconds.
 - Displays the current score at X:48, Y:22.
 - Displays "得分:" (Score:) at X:0, Y:24.
 - Displays the variable score at X:48, Y:24.
- 定义 游戏结束 (Define Game End):**
 - Repeats the following actions 3 times:
 - Displays "得分:" (Score:) at X:0, Y:24.
 - Displays the variable score at X:48, Y:24.
 - Waits 0.5 seconds.
 - Clears the screen.
 - Waits 0.3 seconds.
 - Displays "游戏结束" (Game over) at X:32, Y:22.
 - Waits 3 seconds.

项目六 火线冲击

在投篮机项目中，我们知道了让游戏设计的两点秘诀：玩家快速获得**反馈**、完成**目标**获得**成就感**。

请你想一想，不管是剪刀石头布，还是超级马里奥，是不是几乎所有的游戏都具备这两点特征？这说明具备这两点还不能让游戏一定变得好玩，那让游戏变好玩的秘诀又是什么呢？

秘诀就是基于这两点去**丰富游戏机制**的设计，从多方面提升玩家的游戏体验。

从**反馈设计**来说，如何让玩家在游戏中获得反馈？什么时候让玩家获得反馈更好？作为游戏设计师，在设计玩家体验时，这些都是必须要考虑到的。比如简单的剪刀石头布游戏，如果给输的一方加入一些惩罚反馈，我们是不是觉得就更刺激了！



从**目标设计**来说，如何让玩家始终处于努力去达到某个目标的状态下？如何将游戏的终极目标拆分成若干个里程碑目标，引导玩家一步步达成？在游戏过程中，让玩家体验到强烈的参与和代入感，当目标完成时自然会强烈的成就感。

在这个项目中，我们将从游戏设计者的角度出发，从以上两点出发设计一款好玩的火线冲击游戏。

在火线冲击游戏中，玩家手持**电环**进入**火线冲击通道**，电环在通道中运行时不能碰触中间的火线，一旦碰触则游戏失败，一路顺利地走完火线到达终点即为胜利者。



头脑风暴：如何让火线冲击变得好玩？

问题清单

- 1、火线冲击游戏是怎么玩的，有哪些**基本功能**？
- 2、从**反馈设计**角度，你有什么可以让游戏变好玩的点子？
- 3、从**目标设计**角度呢？
- 4、你的想法具备可实现性吗？需要哪些**传感器或执行器**？
- 5、描述出在你的想法中，火线冲击游戏的**外观设计**。
- 6、你学过**初中电路知识**吗？尝试用电路知识解释火线冲击游戏的原理。

核心知识点

一、电路（结合物理学科）

我们都知道，生活中有各种各样的电器，比如冰箱、洗衣机、电灯等。但是为什么有了电，这些电器就可以运转呢？

那是因为有**电路**的存在！

电路组成

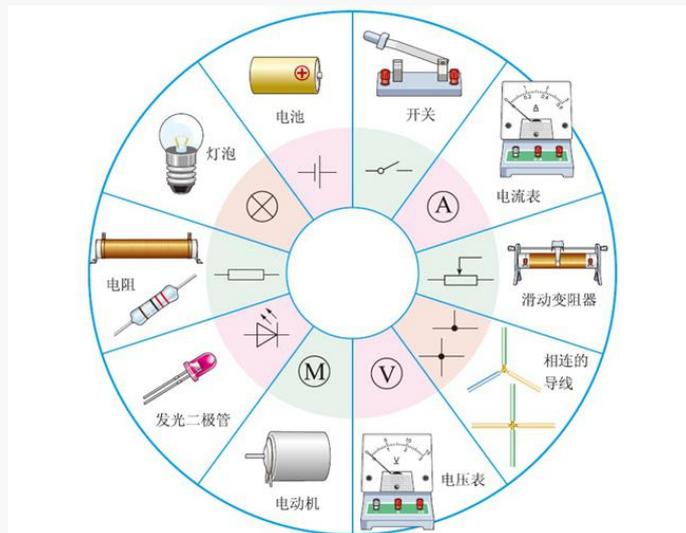
初中物理课上讲过最简单的电路可由**电源、开关、导线、用电器**组成。

如右图，在手电筒中，电池是**电源**，灯泡是**用电器**，电线就是**导线**，开关按下就可以打开电灯。

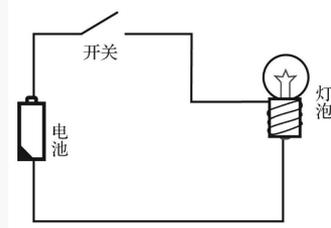
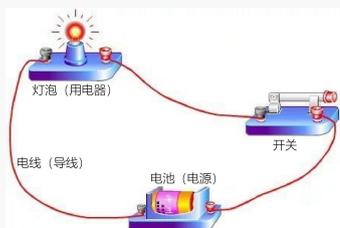


电路图

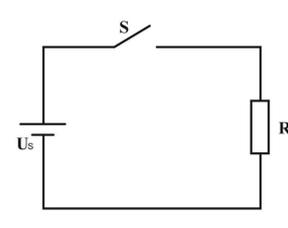
我们知道地图中的道路、森林等可以通过简单的符号来清晰的表述。类似的，在物理电学中也有**标准的符号和图标**表示电路的元件，如下图。



用简化后的元件图标就可以**快速绘制电路图**。如下图为简化的手电筒电路图。



手电筒电路示意图

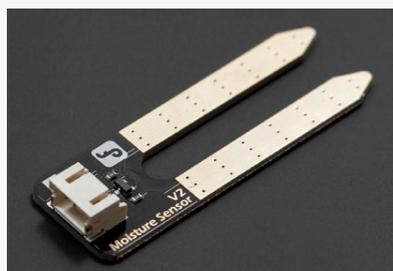


手电筒电路模型

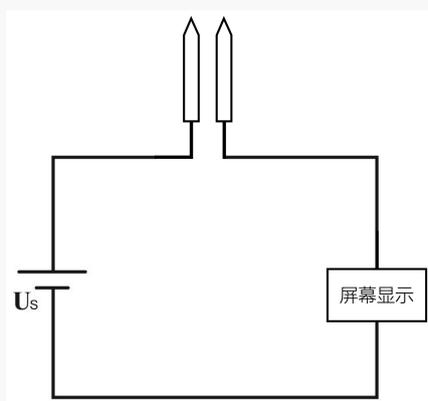
二、土壤湿度传感器

土壤湿度传感器

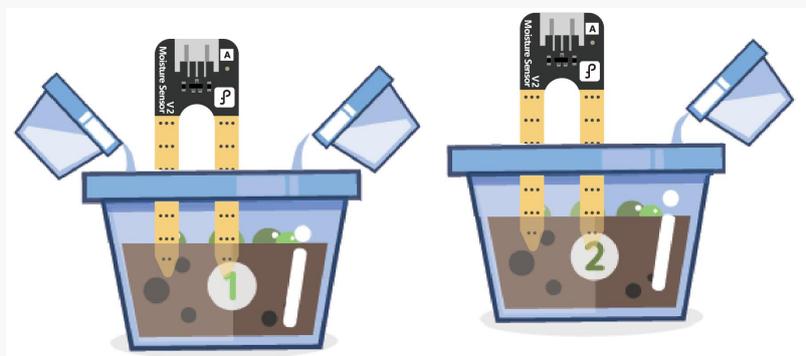
土壤湿度传感器主要用来检测土壤含水量。使用时同时将两金属条插入土壤中，当土壤缺水时，传感器输出值将减小，反之将增大。



例如使用掌控板、土壤湿度传感器可以做一个简易土壤湿度显示器，结合电路知识，电路图简化如下。从电路图可以看出，这里的土壤湿度传感器相当于开关，当两金属条插入到土壤中后，相当于电路导通。



思考一下：为什么在不同湿度的土壤中，传感器的输出值不一样呢？为什么湿度越大，输出值越高呢？



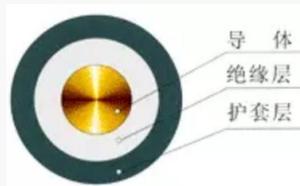
因为不同物体的导电能力是不一样的，土壤的湿度会影响它的导电能力！根据是否容易导电，可以将物体分为导体和绝缘体。

什么是导体？（结合物理学科）

在生活中，有的东西是导电的，有的却不能导电。以电线为例，线芯是导电的，包裹线芯的塑料皮是不导电的。

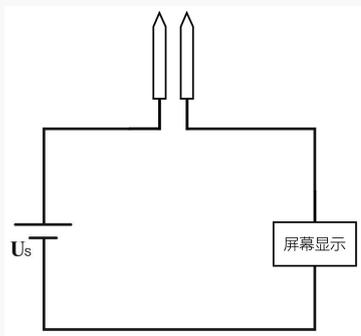


在初中物理中，把容易导电的物体叫导体，不易导电的物体叫做绝缘体。如下为电线截面图，其中电线芯是导体，塑料皮是绝缘体。



常见的导体	铜芯、有杂质的溶液（比如土壤中的水）、保险丝、金属、人体.....一般金属都是导体。
常见的绝缘体	空气、陶瓷、橡胶、干木头、塑料制品、玻璃.....

对于土壤来说，土壤的含水量会影响它的导电性，所以下面这个电路可做如下分析：



物质	导体/绝缘体	结果
空气	绝缘体	电路不通, 读值为 0
含水少的土壤	导体, 导电性弱	电路导通, 读值小
含水多的土壤	导体, 导电性强	电路导通, 读值大

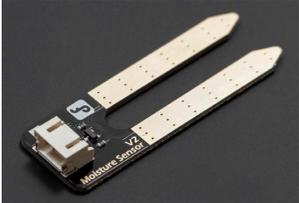
这里的土壤湿度传感器的工作原理可以理解为连接在两金属条之间的物体的导电性。导电性越好，测量值越大；导电性越差，测量值越小；不导电，测量值为 0。

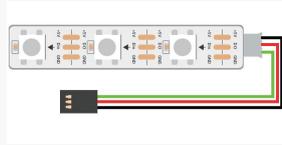
思考一下：在火线冲击项目中，如何利用土壤湿度传感器呢？

规划方案

功能分解

在本项目中，为了能从游戏设计者的角度去优化游戏设计，我们将游戏分为**基础功能**和**加分功能**。其中基础功能是这个游戏必不可少的，而加分功能就是游戏的亮点，按照前面所学，我们将从**反馈设计**和**目标设计**两个角度作为加分功能去增强游戏机制。

基础功能	所需元件
<p>2、基本功能：判断火线是否被套环触碰</p> <p>将土壤湿度传感器的一端金属条连接火线，另一端连接套环，当套环碰到火线，相当于两金属条导通。</p>	<p>土壤湿度传感器(检测土壤湿度)</p> 
<p>3、基本功能：判断游戏是否开始</p> <p>我们需要一个开关装置来判断游戏是否开始。</p> <p>这个开关可以是按钮、角度传感器，甚至是遥控器！</p>	<p>Eg1.按钮 (数字开关，按下为 1，松开为 0)</p>  <p>Eg2.模拟角度传感器 (模拟开关)</p>  <p>Eg3.红外遥控器 (遥控设备，与红外接收管配合使用) ...</p> 

加分功能	所需元件
1、从反馈设计角度：碰触到火线时，发出 声光警报 ，增加游戏刺激感。	RGB 灯带（含 7 个 RGB 灯，可显示任意颜色）  掌控板自带蜂鸣器
2、从目标设计角度：设定火线碰触次数，超过碰触次数超限则 游戏失败 。	编程实现即可

构思外观

画出外观草图	所需耗材、工具
例如：	例如：导电铜丝、绝缘胶带、杜邦线、泡沫垫、剪刀、胶带等... 1、 2、 3、

问题记录

在后面项目实现过程中，可能会遇到各种各样的困难，尝试在下表中记录你遇到的问题和解决办法，便于以后出现类似问题时能更好的面对。

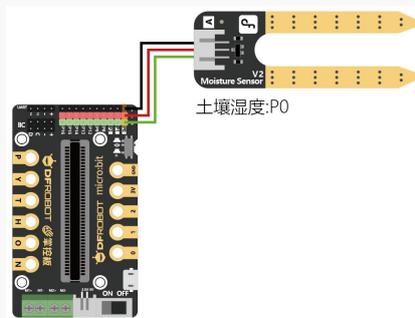
遇到的困难	你的解决方案
1、	1、
2、	2、
3、	3、
...	...

功能实现

功能 1：土壤湿度传感器判断火线是否与套环导通，并记录导通次数

这里可以选用导电铜丝制作出火线和套环。

连线图：



测试程序：将土壤湿度传感器测量的湿度显示在屏幕上。



测试方法：在两个金属条之间放不同的物体，记录测量数值。例如下表：

物体	空气	双手	铜线	杜邦线外皮	杜邦线线芯
测量值	0	100-900	3000-3100	0	3000-3100

* 用剪刀轻轻剪掉杜邦线外皮即可看到线芯。

功能程序：

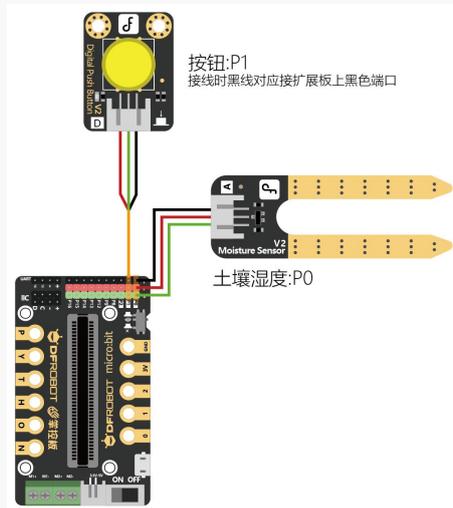


运行结果：两个金属条每导通一次，屏幕显示的数值加 1。

功能 2：按下按钮，开始或结束游戏

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。

连线图：



功能程序：

ESP32 主程序

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

等待 0.5 秒

判断按钮是否按下

如果 变量 开关状态 除以 2 的余数 = 1 那么执行

导通检测

按钮按下奇数次表示开始

否则

设置 导通次数 的值为 0

屏幕显示为 全黑

按钮按下偶数次表示结束

定义 导通检测 定义函数检测导通次数，简化主程序，便于理解

如果 读取模拟引脚 P0 > 1000 那么执行

将 导通次数 增加 1

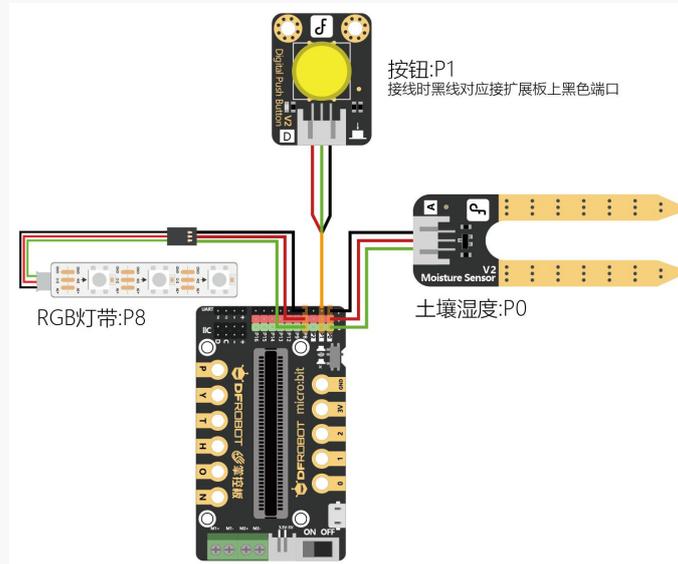
等待 0.5 秒

屏幕显示文字 变量 导通次数 在坐标 X: 0 Y: 22 预览

功能 3：碰触到火线时，发出声光警报（从反馈设计角度）

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

连线图：



Mind+软件设置：使用 RGB 灯带需要点击“扩展”，在“显示器”下选择“RGB 灯”。



这里为了营造丰富的视觉效果，在游戏开始前，灯带显示**渐变色**；游戏开始后，灯带为**渐变色流水灯**；当套环碰到火线时的**灯带显示红色**，并发出**声音警报**；套环离开火线后，灯带继续为**渐变色流水灯**。

功能程序：在功能 2 程序基础上作如下修改

ESP32 主程序

灯带初始化指令，用于设置引脚、灯总数和亮度

初始化 RGB 灯 引脚 P8 灯总数 7 亮度 255

RGB 灯 引脚 P8 灯号 0 到 6 显示渐变颜色 色调从 1 到 360

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

等待 0.5 秒

灯带循环移动指令

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

等待 0.1 秒

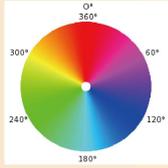
游戏过程中产生流水灯效果

否则

设置 导通次数 的值为 0

等待 0.1 秒，即流水灯移动速度

屏幕显示为 全黑



定义 导通检测

如果 读取模拟引脚 P0 > 1000 那么执行

将 导通次数 增加 1

导通时灯带显示红色

RGB 灯 引脚 P8 灯号 0 到 6 显示颜色

播放音符 1 低 C/C3 1/4 拍

声光报警

RGB 灯 引脚 P8 灯号 0 到 6 显示渐变颜色 色调从 1 到 360

屏幕显示文字 变量 导通次数 在坐标 X: 0 Y: 22 预览

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

功能 4：设定导通次数上限，导通次数超限则游戏失败（从目标设计角度）

在功能 3 程序基础上，主程序不变，在函数“导通检测”中添加判断导通次数是否达到上限的功能。程序修改如下：

```

    定义 导通检测
    如果 读取模拟引脚 P0 > 1000 那么执行
        将 导通次数 增加 1
        RGB灯 引脚 P8 灯号 0 到 6 显示颜色 红色
        播放音符 1 低 C/C3 1/4 拍
        RGB灯 引脚 P8 灯号 0 到 6 显示渐变颜色 色调从 1 到 360
        屏幕显示文字 变量 导通次数 在坐标 X: 0 Y: 22 预览
    如果 变量 导通次数 >= 10 那么执行
        RGB灯 引脚 P8 灯号 0 到 6 显示颜色 蓝色
        屏幕显示文字 挑战失败 在坐标 X: 32 Y: 22 预览
        播放音符 1 高 C/C5 1/4 拍
        等待 3 秒
        屏幕显示为 全黑
        设置 开关状态 的值为 0
        RGB灯 引脚 P8 灯号 0 到 6 显示渐变颜色 色调从 1 到 360
    
```

挑战过程中，碰触次数超过上限，则挑战失败，发出声光提示和文字显示

重置“开关状态”为0，结束游戏

运行结果：按下按钮，开始游戏，游戏过程中如果导通次数超过上限，则显示挑战失败，再次按下按钮可重新开始游戏。

构造外观

示例图:

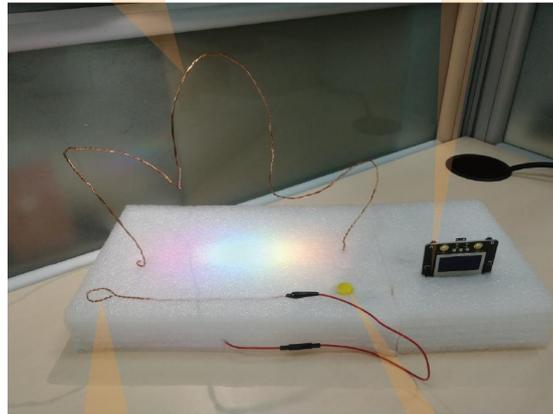
外观材料：导电铜丝、绝缘胶带、杜邦线、泡沫垫

工具：剪刀、胶带

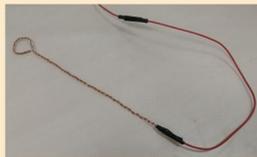
铜丝卷曲出火线模型



只露出显示屏的掌控板

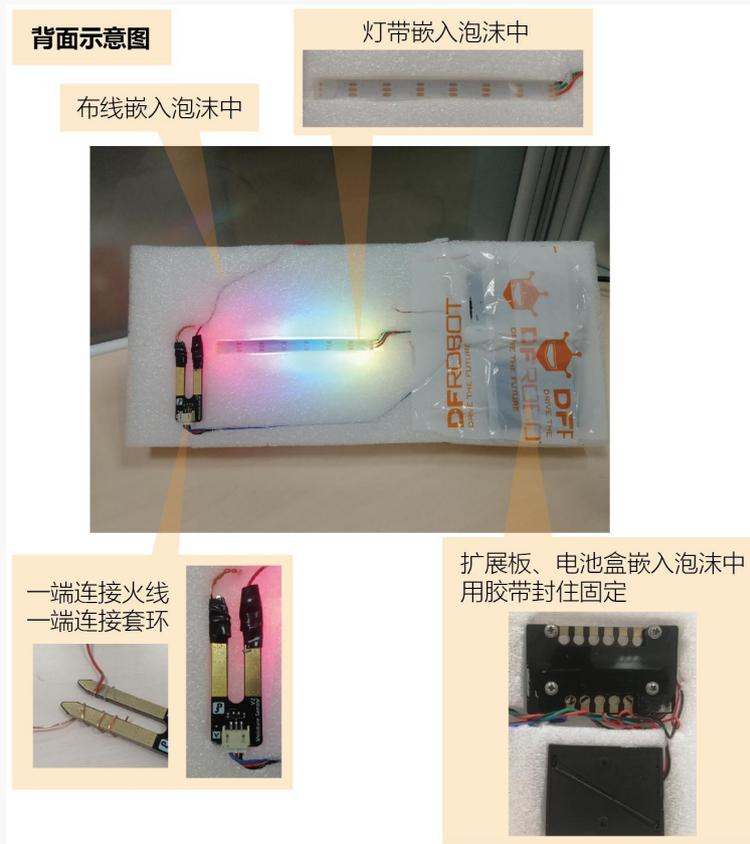


铜丝结合杜邦线的套环模型



只露出按钮的开关





更多创意

在添加了功能 3、4 之后，我们的游戏是不是更加好玩了？从设计游戏的角度来说，不管游戏本身是简单还是容易，只要能够设计出足够好玩的游戏机制，都可以变得充满趣味。

在项目的开头，我们讲到设计游戏的两点秘诀，你还记得是什么吗？尝试使用秘诀优化你的火线冲击，设计出让人欲罢不能的游戏！

评价与反思

与人分享快乐，能产生更多快乐，尝试与身边的人一起来挑战火线冲击！你还可以在 DF 创客社区中分享作品，听听更多人的建议！社区网址：www.dfrobot.com.cn

问卷调查

尝试让玩过你作品的人做下面的问卷表，看看在别人眼中这个作品的闪光点与不足。

你觉得我的火线冲击好玩吗？（1-5 分）	
你觉得最好玩的点在哪？	
你喜欢我搭建的外观吗？（1-5 分）	
你觉得哪里还有不足？	

自我反思

梳理自己设计项目的思维逻辑、项目在执行层面的问题和经验等等，可以帮助我们形成学习的闭环，加深在项目进行的过程中对知识点和内容的理解，找到可以改进和提高的地方。

优点与不足	1、
	2、
可以从项目中总结的经验	1、
	2、
学到的知识	1、
	2、

项目七 答案之书

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。

那么《答案之书》或者说类似的占卜真的有那么神奇吗？

其实不是，这只能起到给人一种心理暗示的作用。在真正有问题时候，要知道所有预测都不能真的得到结局，但是可以通过这样的心理暗示，赋予自己迎接结局的勇气。

话说回来，掌控板如何能做一个《答案之书》呢？如果结合求签活动中的一些好玩的点子，能不能做出更有趣的应用呢？一起来头脑风暴一下吧！

头脑风暴

问题清单

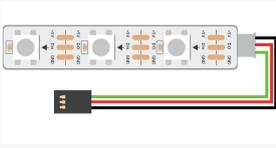
- 1、真实的《答案之书》通过随意翻页得到一个随机的一句话作为答案，掌控板如何显示随机的一句话呢？如何模拟翻页这个动作呢？
- 2、能否结合求签活动中将求签目的分为求财、问事、姻缘等等这些，开发更好玩的应用方式呢？
- 3、如何设计人与答案之书的硬件交互呢？是否可以加入一些灯光增加神秘色彩呢？
- 4、你还有其他好玩的想法吗？尝试评估它的可实现性，然后动手实现吧！

规划方案

功能分解

在这个项目中，需要实现的基本功能是像《答案之书》一样，让寻求答案的人在做出某种触发后，得到一句答案。

这里选择按钮作为触发，在掌控板上随机显示一句话作为答案。结合求签活动的启发，添加手动选择目的的功能，使用旋钮配合舵机、灯带实现功能。

分解功能	所需元件
<p>1、按下按钮，显示答案</p> <ul style="list-style-type: none"> 触发答案之书的方法有很多，按钮、旋钮、土壤湿度、颜色传感器、超声波等各种传感器几乎都可以做到。 这里为了操作简单，选用了按钮。大家在实际做项目时，可以开发其他好玩的方式。 	<p>按钮（数字开关，按下为 1，松开为 0）</p> 
<p>2、神奇的答案之书</p> <ul style="list-style-type: none"> 通过旋钮控制舵机的转动角度，当舵机在不同的角度时，分别代表求财、问事、姻缘等等，即可让人通控制旋钮自由选择目的。 加入 RGB 灯带增加更多灯光互动，这里笔者的想法是用灯带的 7 颗灯珠代表 7 个不同的目的，与舵机结合后，将舵机的 180 度转动范围均匀分为 7 个范围，每个范围对应一颗灯。 当旋钮转动时，舵机转动，灯带上对应的灯珠亮。 选好目的后，按下按钮，屏幕显示选择的目的地和答案。 	<p>模拟角度传感器（模拟开关）</p>  <p>舵机（在 0-180 度之间转动）</p>  <p>RGB 灯带（含 7 个 RGB 灯，可显示任意颜色）</p> 
<p>3、更多创意</p>	

构造外观

画出外观草图	所需耗材、工具
例如：	例如：塑料泡沫、废纸盒、雪糕棒、双面胶等... 1、 2、 3、

问题记录

在后面项目实施过程中，可能会遇到各种各样的困难，尝试在下表中记录你遇到的问题和解决办法，便于以后出现类似问题时能更好的面对。

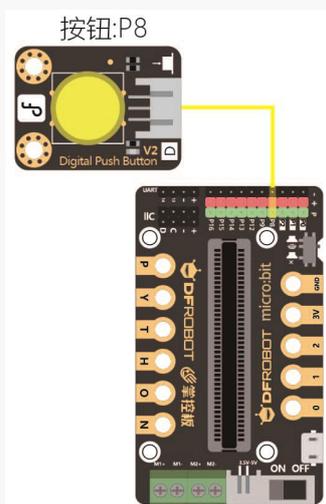
遇到的困难	你的解决方案
1、	1、
2、	2、
3、	3、
...	...

功能实现

功能 1：按下按钮，显示答案

通过按下按钮，在掌控板屏幕上随机显示一句话作为答案。

连线图：



程序：

```

    ESP32 主程序
    屏幕显示文字 答案之书 在坐标 X: 32 Y: 14 预览
    等待 3 秒
    屏幕显示文字 默念问题按下按钮 在坐标 X: 0 Y: 34 预览

    循环执行
    如果 读取数字引脚 P8 那么执行
        回答 按下按钮，显示一个回答
        等待 0.5 秒
        等待直到 读取数字引脚 P8
        屏幕显示为 全黑
        屏幕显示文字 默念问题按下按钮 在坐标 X: 0 Y: 24 预览
        等待 0.5 秒
    
```



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！

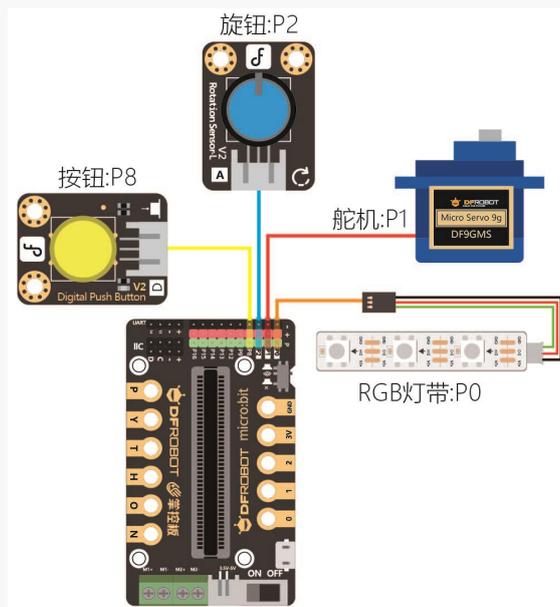
功能 2：神奇的答案之书

使用旋钮控制舵机，舵机 0-180 度转动范围与灯带的 0-6 号灯珠——对应。

这里用灯带的 7 颗灯珠代表 7 个不同的目的，例如求签中常用的平安、姻缘、占病、问事、寻人、功名、求财。

当旋钮转动时，舵机转动，对应的灯珠亮。这样寻求答案的人可以在手动选好目的后，按下按键，即可在屏幕上显示选择的目的地和随机出现的答案。

连线图：



程序:

ESP32 主程序

屏幕显示文字 答案之书 在坐标 X: 32 Y: 24 预览

等待 3 秒

初始化 RGB灯 引脚 P0 灯总数 7 亮度 255

通过“映射”，将旋转旋钮的模拟引脚读值0-4095与舵机角度0-180度对应

循环执行

RGB灯 引脚 P0 全部熄灭

旋钮控制舵机角度和灯带

设置 P1 引脚伺服舵机为 映射 读取模拟引脚 P2 从 [0, 4095] 到 [0, 180] 度

设置 选择 的值为 映射 读取模拟引脚 P2 从 [0, 4095] 到 [0, 6]

RGB灯 引脚 P0 灯号 变量 选择 到 变量 选择 显示颜色

如果 读取数字引脚 P8 那么执行

回答 按下按钮, 显示一个答案

等待 0.5 秒

等待直到 读取数字引脚 P8

再次按下按钮, 清除答案

屏幕显示为 全黑

屏幕显示文字 缘主心中求何? 在坐标 X: 8 Y: 24 预览

等待 0.5 秒

定义 回答

这里用了5句随机的话作为答案

屏幕显示为 全黑

设置 结果 的值为 在 1 和 5 之间取随机数

如果 变量 结果 = 1 那么执行

答案1

屏幕显示文字 一切都会很不错 在坐标 X: 0 Y: 32 预览

否则

如果 变量 结果 = 2 那么执行

答案2

屏幕显示文字 还需要一点耐心 在坐标 X: 0 Y: 32 预览

否则

如果 变量 结果 = 3 那么执行

答案3

屏幕显示文字 你在开玩笑吗 在坐标 X: 0 Y: 32 预览

否则

如果 变量 结果 = 4 那么执行

答案4

屏幕显示文字 也许会很难, 但是值得 在坐标 X: 0 Y: 32 预览

否则

如果 变量 结果 = 5 那么执行

答案5

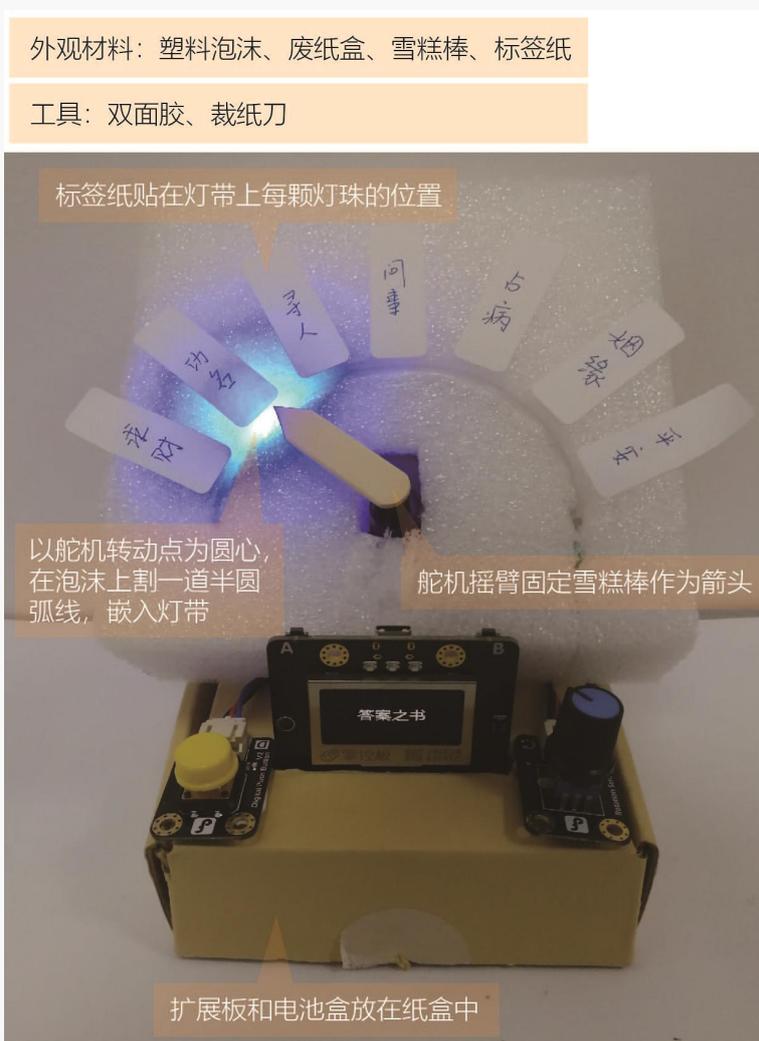
屏幕显示文字 不要指望它 在坐标 X: 0 Y: 32 预览



运行结果：先转动旋钮，舵机随之转动，对应灯带上的灯珠亮；再按下按钮，屏幕上第一行显示旋钮选择的目的，第二行显示一句话作为答案。

构造外观

外观结构有多种实现方法，下图仅作参考：



评价与反思

与人分享快乐，能产生更多快乐，尝试让身边有困扰的亲朋好友来体验一下你的答案之书，看看能否为他们心中的困扰找到一丝方向呢！你还可以在 DF 创客社区中分享作品，听听更多人的建议！社区网址：www.dfrobot.com.cn

问卷调查

尝试让体验过你作品的人做下面的问卷表，看看在别人眼中这个作品的闪光点与不足。

你觉得我设计的《答案之书》对你有帮助吗？（1-5 分）	
你觉得最特别的设计点在哪？	
你觉得还有更好的建议吗？	

自我反思

梳理自己设计项目的思维逻辑、项目在执行层面的问题和经验等等，可以帮助我们形成学习的闭环，加深在项目进行的过程中对知识点和内容的理解，找到可以改进和提高的地方。

优点与不足	1、
	2、
可以从项目中总结的经验	1、
	2、
学到的知识	1、
	2、

项目八 坦克大战

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**。



头脑风暴

问题清单

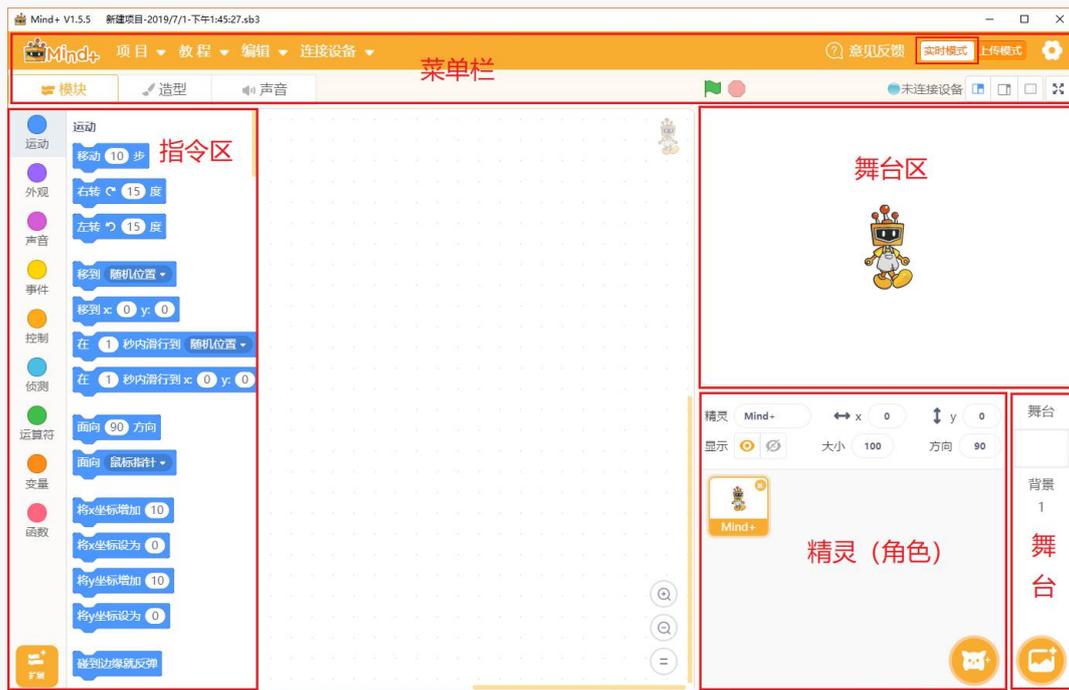
- 1、在坦克大战中，**玩家需要如何操作**？
- 2、玩家需要遵守哪些**规则**？
- 3、玩家想要达成的**目标**是什么？
- 4、玩家在游戏中有哪些**交互的对象**？
- 5、需要哪些**硬件**实现玩家的操作？
- 6、尝试从游戏中几个基本的交互对象**分解游戏功能**。

核心知识点

一、Mind+实时模式

简介

打开 Mind+软件，切换为“**实时模式**”。实时模式下，软件界面介绍如下。



操作说明

1、删除/新建角色

删除角色	新建角色
点击角色右上角“x”，即可删除	如下图，有多种方法新建角色
	

2、编写/运行程序

实时模式下，每个角色要**分别编程**。单击角色对应程序块即可运行程序，运行中的程序有黄色底边。



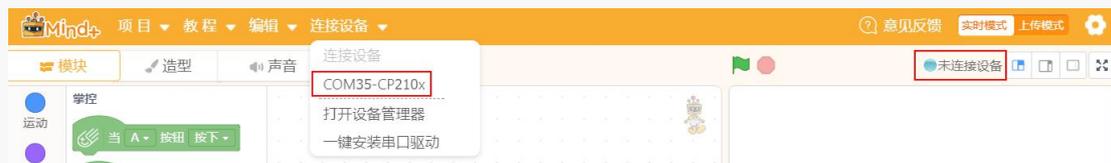
连接掌控板

实时模式下，通过以下三步可连接掌控板：

- 1、用 USB 线将掌控板连接到电脑；
- 2、在扩展中，选择掌控板；



3、选择对应 COM 口。



连接成功后，显示端口信息和“连接设备成功”。



二、模拟角度传感器

模拟角度传感器是一款非常基本的模拟信号输入设备。上面有一个旋钮，用来改变输入信号的大小。



在生活中就有很多地方用到了角度传感器，比如在音箱上旋转旋钮可以改变声音大小。



规划方案

功能分解

在实时模式下，需要对**每个角色分别编程**，最终所有角色在舞台上产生**交互效果**。所以在这里，我们将对坦克大战中几个基本角色进行功能分析。

角色	功能分析	所需元件
坦克 	玩家控制坦克在舞台底部 左右平移 可通过角度传感器控制。	模拟角度传感器（模拟开关） 
炸弹 	玩家控制炸弹发射 按下按钮，触发一次炸弹发射。	按钮（数字开关，按下即发射炸弹） 
飞机 	1、从舞台顶部随机位置出现，向下掉落，攻击坦克 2、被坦克发射的炸弹击中则爆炸	编程实现

为了使游戏更加人性化，还要**完善游戏机制**。

完善游戏机制	1、 得分机制 ：炸弹击中飞机，得分+1，否则得分-1 2、 失败机制 ：超过游戏时间或炸弹用光，则挑战失败 3、 成功机制 ：在游戏失败前，得分达到设定目标，则挑战成功	编程实现
--------	--	------

构思舞台

在本项目中，玩家与游戏的主要交互是**舞台**，就像坦克大战的游戏界面一样。

尝试勾勒出你设计的**坦克大战游戏**的交互界面，画出各个角色在舞台中的简单**位置示意图**。



画出舞台草图

例如：

问题记录

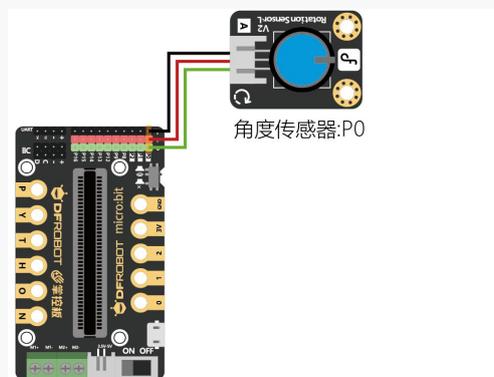
在后面项目实现过程中，可能会遇到各种各样的困难，尝试在下表中记录你遇到的问题和解决办法，便于以后出现类似问题时能更好的面对。

遇到的困难	你的解决方案
1、	1、
2、	2、
3、	3、
...	...

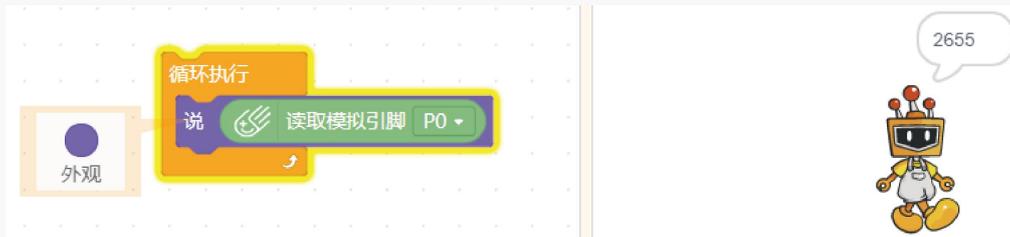
功能实现

功能 1：角度传感器控制“坦克”在舞台底部左右移动

连线图：



测试程序：角色读取角度传感器值。



运行程序，转动旋钮，读值为 0-4095。

接下来，需要新建一个“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）。



新建“坦克”角色：点击“上传角色”，选择要导入的图片。



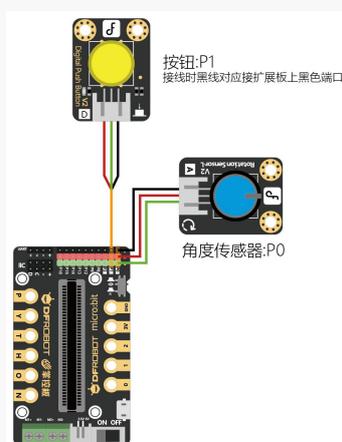
“坦克”角色功能程序：角度传感器控制“坦克”x坐标变化，使其随旋钮转动左右移动。



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。

功能 2：按下按钮，发射炸弹

连线图：



接下来新建“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。



“炸弹”角色功能程序：选中“炸弹”角色，编写下面程序

点击绿旗运行程序。

运行结果：转动旋钮，“坦克”随转动方向左右平移；按下按钮，“坦克”发射“炸弹”。



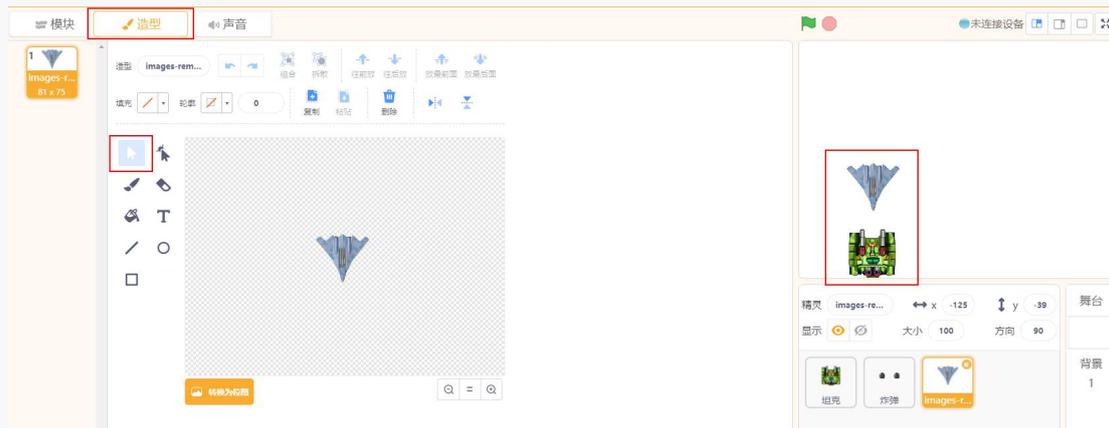
功能 3：飞机从舞台上空落下，攻击坦克；被炸弹击中则爆炸

这里需要新建“飞机”角色。

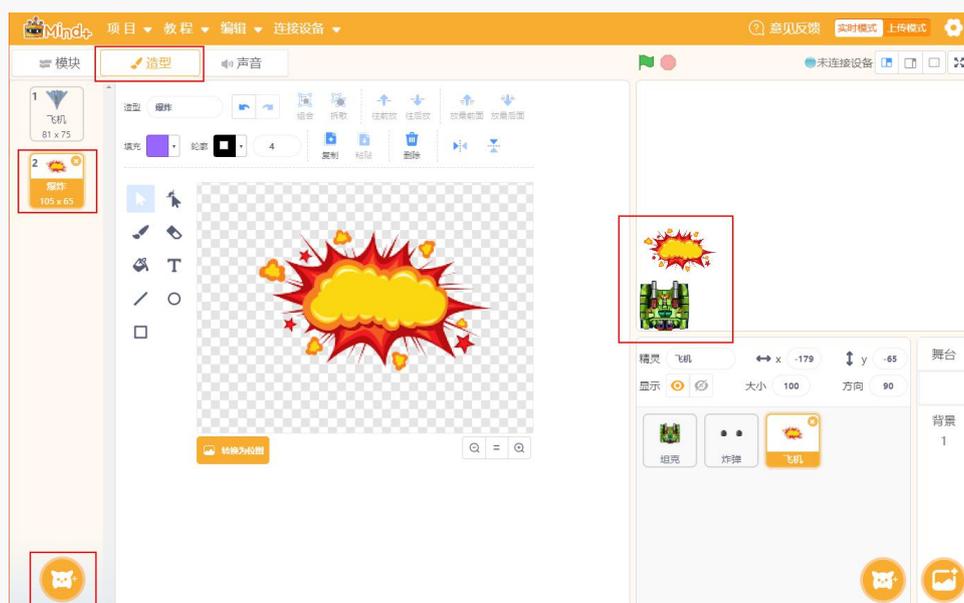
“飞机”角色图片：



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。



功能程序：选中“飞机”角色，编写下面程序

点击绿旗，运行程序。

运行结果：飞机从舞台顶部随机位置下落；

转动旋钮，“坦克”随转动方向左右平移；

按下按钮，“坦克”发射“炸弹”；

当“炸弹”碰到“飞机”，显示“爆炸”造型。

功能 4：完善游戏机制

游戏机制	功能描述	编程思路
得分机制	炸弹击中飞机，得分+1，否则得分-1	<ul style="list-style-type: none"> ▪ 设定得分变量 ▪ 在“飞机”角色下，当“炸弹”触碰到“飞机”，得分+1 ▪ 当“飞机”垂直降落到底部，得分-1
游戏结束	超过游戏时间或炸弹用光，游戏结束	<ul style="list-style-type: none"> ▪ 设定时间变量，记录游戏时间 ▪ 设定炸弹变量，记录游戏中剩余炸弹数量 ▪ 在“炸弹”角色下，当发射一次炸弹，炸弹数量-1 ▪ 当时间大于设定值或炸弹数量小于 0，游戏结束
成功机制	得分达到设定目标，挑战成功，否则挑战失败	<ul style="list-style-type: none"> ▪ 需要新建一个角色，包括挑战成功、挑战失败两个造型 ▪ 游戏结束后判断得分是否达到目标值，然后显示对应造型

编程实现步骤为 STEP1-STEP4。

STEP1：新建 3 个适用于所有角色的变量。



STEP2：修改“飞机”角色程序，加入得分判断



STEP3：修改“炸弹”角色程序，加入炸弹数量判断



STEP4: 新建“成功失败”角色，包含“挑战成功”、“挑战失败”两个造型。这里可自己绘制造型。



“成功失败”角色程序:



构造外观

本项目的视觉交互主要在于软件中的舞台设计，硬件部分只需要固定角度传感器和按钮即可，这里不再作示例图。

更多创意

想一想，在真实的电子游戏中，还有哪些设计可以提升玩家体验？比如好看的**背景设计**，更丰富的**角色交互**，加入**背景音乐**等。尝试从自己玩游戏的感受出发，设计出更加好玩的坦克大战！

评价与反思

与人分享快乐，能产生更多快乐，尝试与身边的人一起来玩坦克大战！你还可以在 DF 创客社区中分享作品，听听更多人的建议！社区网址：www.dfrobot.com.cn

问卷调查

尝试让玩过你作品的人做下面的问卷表，看看在别人眼中这个作品的闪光点与不足。

你觉得我的坦克大战好玩吗？（1-5分）	
你觉得最好玩的点在哪？	
你喜欢我设计的游戏机制吗？（1-5分）	
你觉得哪里还有不足？	

自我反思

梳理自己设计项目的思维逻辑、项目在执行层面的问题和经验等等，可以帮助我们形成学习的闭环，加深在项目进行的过程中对知识点和内容的理解，找到可以改进和提高的地方。

优点与不足	1、
	2、
可以从项目中总结的经验	1、
	2、
学到的知识	1、
	2、

第三章 让生活更加智能

项目九 感应垃圾桶

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许多方便。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！



头脑风暴：你的感应垃圾桶需要具备哪些功能？

问题清单

- 1、没有盖子的垃圾桶容易产生蚊虫和异味，有盖子的垃圾桶还要手动打开很不卫生，如何解决呢？
- 2、如果制作一个可以通过感应自动开关盖的垃圾桶，通过什么方式控制它自动开关盖子呢？
- 3、有的时候我们希望垃圾桶在感应状态的时候可以及时开关，但是有的时候，比如打扫卫生时需要垃圾桶的盖子一直打开，不受感应开关的影响，如何实现呢？
- 4、垃圾桶在每个人的生活中都会使用到，设想一下人们在使用感应垃圾桶的时候可能还会遇到的其他问题。

核心知识点

舵机

舵机是一种电机，可以控制**旋转角度**。一般的舵机旋转范围是**0-180度**，也有一些能旋转更大的角度，甚至360度。套件中的9g金属小舵机是180度的。

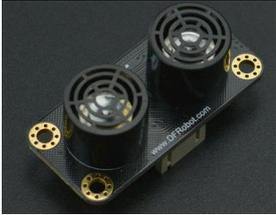


舵机一般用在人形机器人、云台网络摄像头等智能设备中。



规划方案

功能分解

分解功能	所需元件
<p>1、垃圾桶自动开盖、关盖</p> <p>垃圾桶的盖子打开或关闭都有固定的角度，所以可以通过舵机转动，带动垃圾桶盖子开关。</p>	<p>舵机 (在 0-180 度之间转动)</p> 
<p>2、有物体或人靠近垃圾桶时会感应开盖，丢垃圾后会自动关盖</p> <p>用超声波传感器进行判断，有物体或人靠近垃圾桶时，超声波检测到障碍物距离较近，启动舵机，打开盖子。</p>	<p>超声波(探测距离)</p> 
<p>3、通过触发垃圾桶盖子常开，方便长时间扔垃圾</p> <p>这里可以使用按钮作为触发开关。</p> <p>按下按钮，盖子常开，此时超声波不起作用。</p> <p>再次按下按钮，盖子关闭，超声波恢复检测。</p>	<p>按钮 (数字开关，按下为 1，松开为 0)</p> 
<p>4、你还有更多创意吗?</p>	

构造外观

画出外观草图	所需耗材、工具
例如：	例如：硬纸、牛皮纸、双面胶、透明胶等... 1、 2、 3、

问题记录

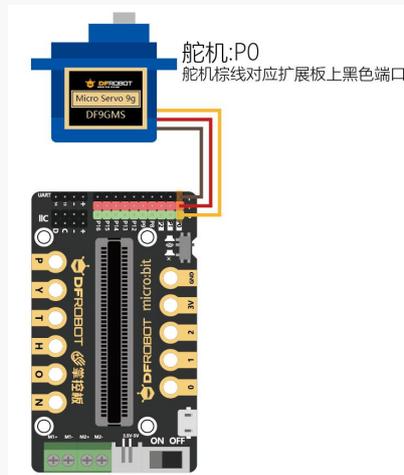
在后面项目实施过程中，可能会遇到各种各样的困难，尝试在下表中记录你遇到的问题和解决办法，便于以后出现类似问题时能更好的面对。

遇到的困难	你的解决方案
1、	1、
2、	2、
3、	3、
...	...

功能实现

功能 1：舵机转动，带动垃圾桶盖子打开或关闭

连线图：使用 P0 引脚注意关闭扩展板上蜂鸣器开关



软件设置：使用舵机需要打开 Mind+ 软件的“扩展”，在“执行器”下点击加载“舵机模块”。



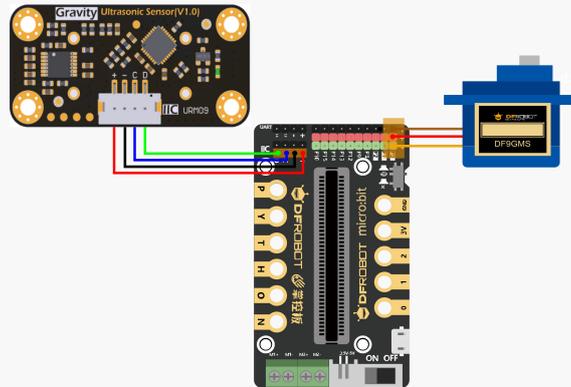
程序：舵机转动

* 运行程序前，可以将舵机摇臂安装在舵机上，便于查看转动过程。



功能 2：使用超声波传感器感应开盖，丢垃圾后自动关盖

连线图：



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”。



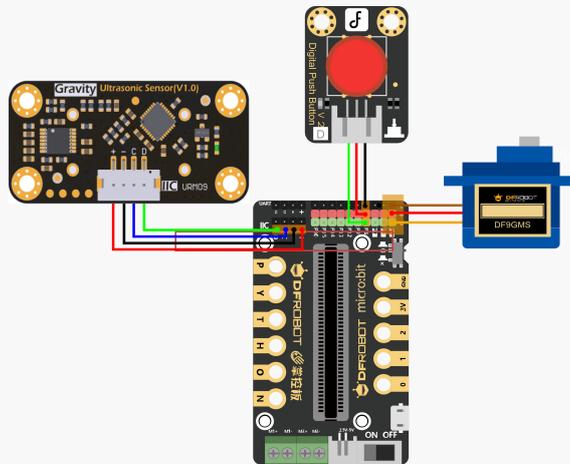
程序：



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。

功能 3：按钮控制垃圾桶盖常开

连线图：

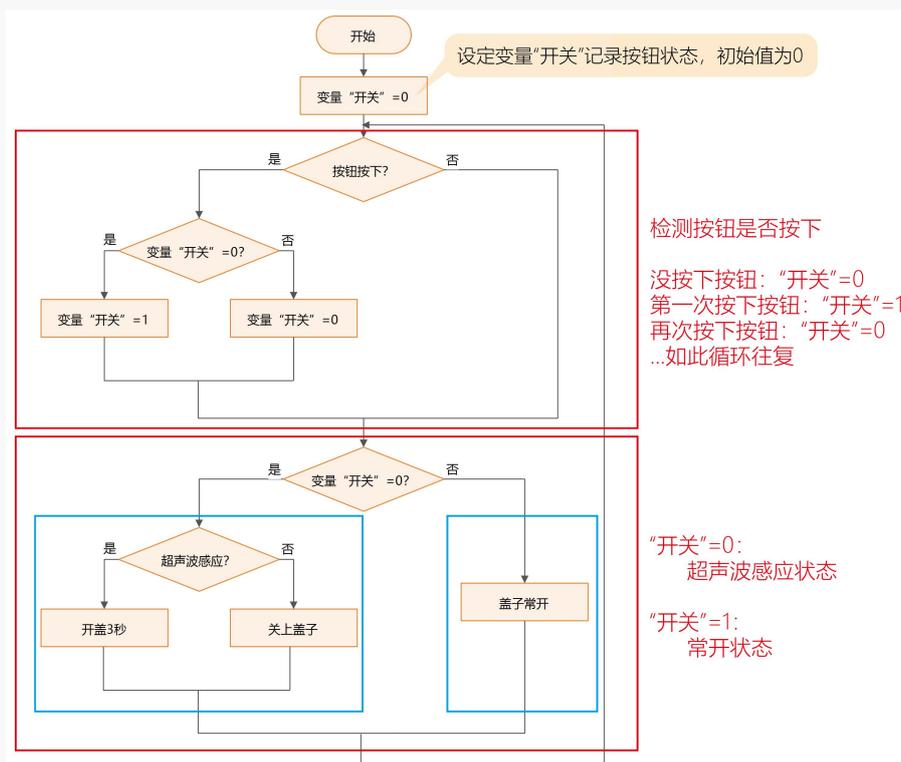


这里需要实现的功能是：

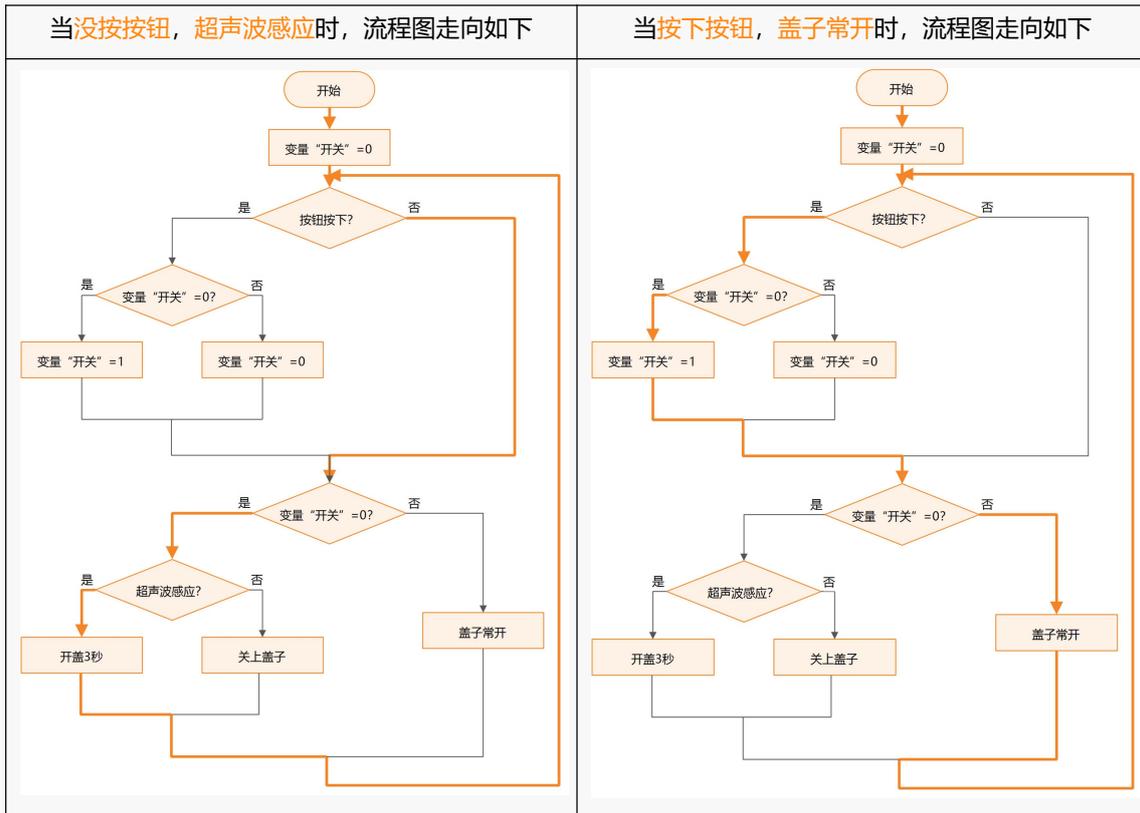
- 程序运行时为**超声波感应状态**；
- **按下按钮**，**盖子常开**，此时超声波不起作用；
- **再次按下按钮**，**盖子关闭**，超声波恢复检测。

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。



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。



程序:

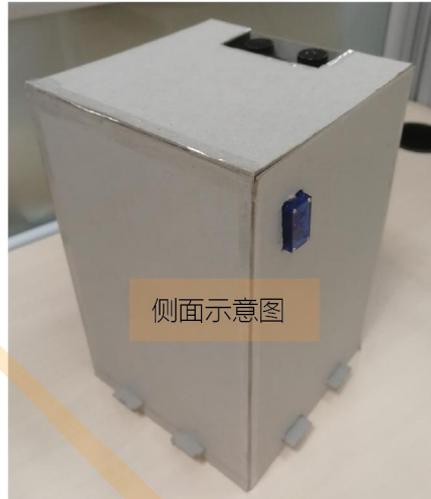


构造外观

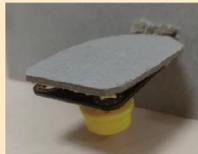
示例图:

外观材料：硬纸、牛皮纸

工具：双面胶、透明胶



按钮安装在底部，触发常开



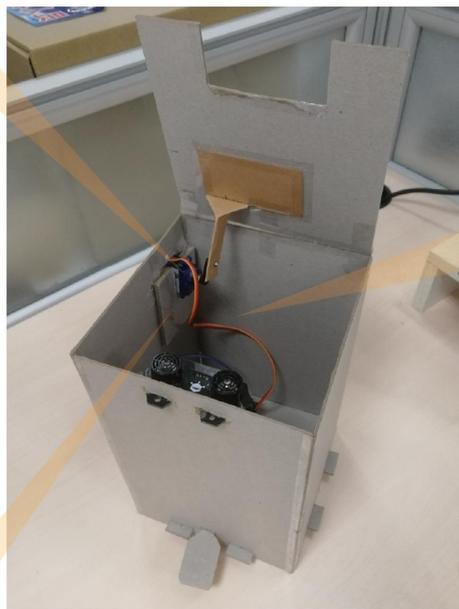
顶部超声波感应



舵机安装在侧面，
通过摇臂和牛皮
纸组成连杆结构，
带动垃圾盖子



硬纸剪出舵机凹
槽，将3-4个凹
槽纸片粘贴在一
起，固定舵机



顶部示意图



评价与反思

与人分享快乐，能产生更多快乐，尝试与身边的人一起来体验你的感应垃圾桶！你还可以在 DF 创客社区中分享作品，听听更多人的建议！社区网址：www.dfrobot.com.cn

问卷调查

尝试让玩过你作品的人做下面的问卷表，看看在别人眼中这个作品的闪光点与不足。

你觉得我设计的垃圾桶好用吗？（1-5 分）	
你觉得最特别的功能点在哪？	
你觉得还有更好的建议吗？	

自我反思

梳理自己设计项目的思维逻辑、项目在执行层面的问题和经验等等，可以帮助我们形成学习的闭环，加深在项目进行的过程中对知识点和内容的理解，找到可以改进和提高的地方。

优点与不足	1、
	2、
可以从项目中总结的经验	1、
	2、
学到的知识	1、
	2、

项目十 AI 助力垃圾分类

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。

核心知识点

人工智能

人工智能 (Artificial Intelligence)，英文缩写为 AI。它是研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门新的技术科学。



人工智能技术应用的细分领域：深度学习、计算机视觉、智能机器人、虚拟个人助理、语音识别、实时语音翻译、情境感知计算、手势控制、视觉内容自动识别、推荐引擎等。

垃圾分类

目前各地实行的垃圾分类法则略有不同，上海市主要将生活垃圾分为 4 类，可参考下图。

湿垃圾 HOUSEHOLD FOOD WASTE	干垃圾 RESIDUAL WASTE	有害垃圾 HAZARDOUS WASTE	可回收物 RECYCLABLE WASTE
菜叶	旧浴缸	漆桶	塑料瓶
橙皮	盆子	电池	食品罐头
葱	坏马桶	打火机	玻璃瓶
饼干	旧水槽	创口贴	易拉罐
番茄酱	贝壳	酒精	报纸
蛋壳	化妆刷	调色板	旧书包
西瓜皮	坛子	油漆	旧手提包
马铃薯	海绵	过期的胶囊药物	旧鞋子
鱼骨	花生壳	温度计	牛奶盒
甘蔗	菜板	过期药片	旧塑料篮子
玉米	砖块	荧光灯	旧玩偶
骨头 (鸡鸭鹅)	卫生纸	蓄电池	玻璃壶
虾壳	篮球	医用棉签	旧铁锅
蛋糕	桃核	杀虫剂	垃圾桶
面包	杯子	水彩笔	旧镜子
草莓	陶瓷碗	农药瓶	牙刷
西红柿	一次性筷子	医用纱布	塑料梳子
梨	西梅核	口服液瓶	旧帽子
蟹壳	坏的花盆	香水瓶	旧夹子
香蕉皮	木质梳子	荧光棒	废锁头
辣椒	脏污衣服	过期化妆品	牙膏皮
巧克力	烟蒂	发胶	雨伞骨架
茄子	渣土	注射器	旧纸袋
豌豆皮	湿垃圾袋	废弃灯泡	纸盒
苹果	瓦片	煤气罐	旧玩具
树叶	扫把	医用手套	

功能实现

功能 1：语音识别

在 Mind+实时模式下，提供了语音识别功能，通过向电脑的麦克风说出内容，利用“语音识别”模块可以将语音转换为文字。

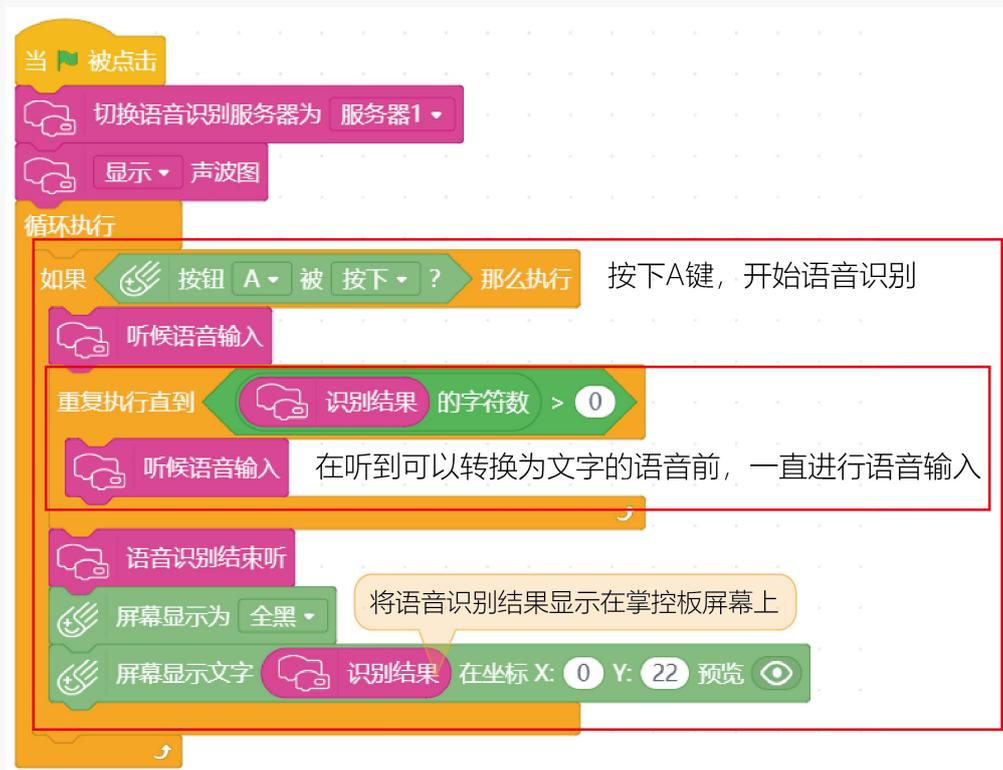
软件设置：使用语音识别功能需要在 Mind+的实时模式下，打开“扩展”，在“功能模块”下点击加载“语音识别”。



接下来将以一个简单的应用为例，说明语音识别的应用方法。程序功能：录入语音，掌控板显示语音转换成的文字。

* 编程时，需在“扩展”中，选择“掌控板”为主控板。

在“实时模式”下，新建任一角色，输入下面程序：



下面的程序图与上面一样，旨在说明程序中**语音识别相关指令**的作用。

设置语音识别服务器，作用是把输入的声音上传到对应服务器进行文字转换

服务器1: MIT服务器
服务器2: 百度服务器

当 被点击

切换语音识别服务器为 服务器1

显示 声波图

循环执行

如果 按钮 A 被按下? 那么执行

听候语音输入 (语音输入过程)

重复执行直到 识别结果的字符数 > 0 (语音转成的文字)

听候语音输入

语音识别结束听 (结束语音输入)

屏幕显示为 全黑

屏幕显示文字 识别结果 在坐标 X: 0 Y: 22 预览

显示 隐藏

运行结果：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，掌控板**屏幕显示**语音转换成的文字。

功能模块

- 当听到 开始
- 听候语音输入
- 识别结果
- 设置每次听 10 秒
- 语音识别结束听
- 显示 声波图
- 切换语音识别服务器为 服务

语音识别: 识别结果

勾选后可在舞台上看到识别结果

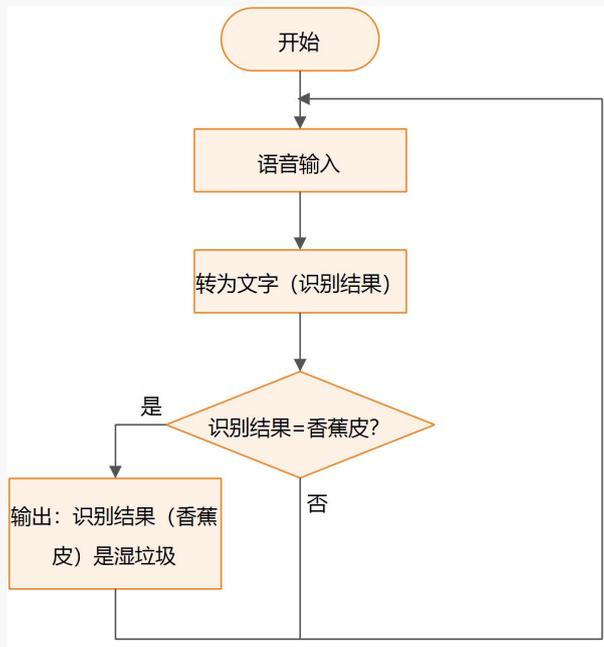
执行“听语音输入”指令时，舞台默认显示声波图

* 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。

功能 2：垃圾分类语音助手

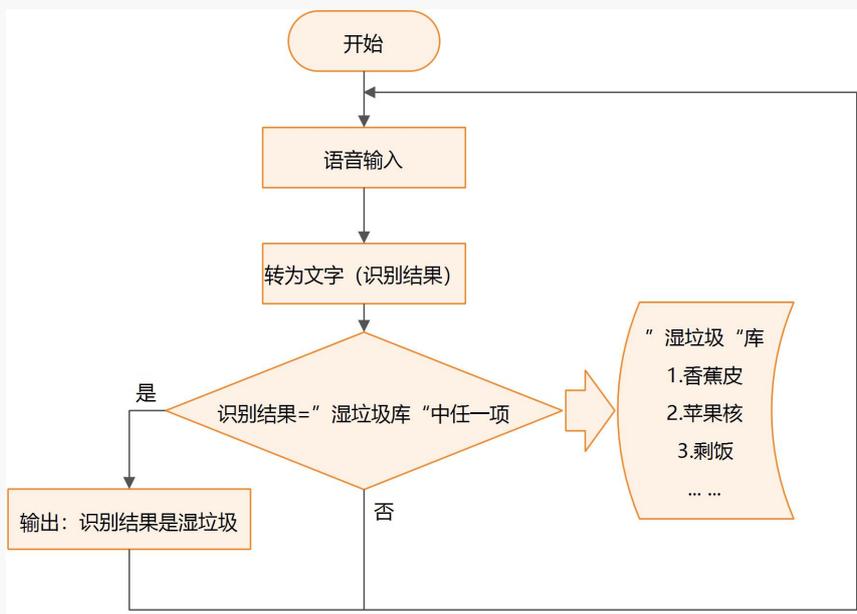
垃圾分类语音助手的作用是人们在说出垃圾名字后，就能知道垃圾是哪一类，想一想，如何用程序实现呢？

在功能 1 程序中，我们知道，**语音**可以转换为**文字**，你可能会想，如果对**文字**做出判断，比如听到“香蕉皮”时，在程序中**判断**“香蕉皮”为“湿垃圾”，然后输出“香蕉皮是湿垃圾”的结果，是不是就满足需求了呢？



其实这样是可以的，但是由于有各种各样的垃圾，如果都要一一判断，程序会非常冗长，这显然是不合理的。

那么试想一下，能不能按照垃圾的分类建立多个“垃圾库”，比如所有的湿垃圾都属于同一个“湿垃圾库”，当外部输入一个垃圾名字时，判断“湿垃圾库”中是否包含这个名字，如果有则说明这个垃圾是湿垃圾。



如何建立这样的“垃圾库”呢？其实可以通过列表来实现。

可将角色造型替换为垃圾桶图片。



程序如下：

```

当 被点击
  屏幕显示为 全黑
  屏幕显示文字 垃圾分类语音助手 在坐标 X: 0 Y: 22 预览
  切换语音识别服务器为 服务器1
  显示 声波图
  循环执行
    如果 按钮 A 被按下? 那么执行 按下A键, 开始语音识别
      屏幕显示为 全黑
      屏幕显示文字 你是什么垃圾? 在坐标 X: 0 Y: 22 预览
      听候语音输入
      重复执行直到 识别结果 的字符数 > 0
      听候语音输入
      语音识别结束听
      屏幕显示为 全黑
      识别垃圾
  
```

程序接下一页。

新建4个列表，在每个列表中分别输入多个垃圾名，勾选显示在舞台上
 当识别结果为对应列表中的某一项时，屏幕显示识别结果，对应颜色灯亮

新建列表

- 列表 可回收物
- 列表 干垃圾
- 列表 有害垃圾
- 列表 湿垃圾

定义 识别垃圾

如果 **干垃圾** 包含 **识别结果 ?** 那么执行 **识别结果为列表“干垃圾”中的某一项时**

屏幕显示文字 合并 **识别结果 是干垃圾** 在坐标 X: 0 Y: 22 预览

灯号 全部 显示颜色 ●

否则

如果 **湿垃圾** 包含 **识别结果 ?** 那么执行 **识别结果在列表“湿垃圾”中**

屏幕显示文字 合并 **识别结果 是湿垃圾** 在坐标 X: 0 Y: 22 预览

灯号 全部 显示颜色 ●

否则

如果 **可回收物** 包含 **识别结果 ?** 那么执行 **识别结果在列表“可回收物”中**

屏幕显示文字 合并 **识别结果 是可回收物** 在坐标 X: 0 Y: 22 预览

灯号 全部 显示颜色 ●

否则

如果 **有害垃圾** 包含 **识别结果 ?** 那么执行 **识别结果在列表“有害垃圾”中**

屏幕显示文字 合并 **识别结果 是有害垃圾** 在坐标 X: 0 Y: 22 预览

灯号 全部 显示颜色 ●

否则

屏幕显示文字 合并 **识别结果 无法判别** 在坐标 X: 0 Y: 22 预览

灯号 全部 显示颜色 ● **识别结果不在上面任一列表时，显示无法判别**

列表“干垃圾”

干垃圾	
1	餐巾纸
2	塑料袋
3	快餐盒

列表“湿垃圾”

湿垃圾	
1	剩菜
2	鸡蛋壳
3	果皮

列表“可回收物”

可回收物	
1	饮料瓶
2	废纸
3	玻璃瓶

列表“有害垃圾”

有害垃圾	
1	电池
2	灯泡
3	药品

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：

语音识别: 识别结果 识别结果 **将4个列表平铺在舞台上**

湿垃圾	干垃圾	可回收物	有害垃圾
1 剩菜	1 餐巾纸	1 饮料瓶	1 电池
2 鸡蛋壳	2 塑料袋	2 废纸	2 灯泡
3 果皮	3 快餐盒	3 玻璃瓶	3 药品
+ length 3 =			

点击“+”，可向列表中插入新项

将角色造型更改为垃圾桶，更加生动

空出此区域，显示声波图



试想一下，如果将**语音助手**和**垃圾桶**结合起来，是否可以做一个更加智能的**语音识别垃圾桶**呢？比如有“湿垃圾桶”、“干垃圾桶”，根据语音识别结果，打开对应垃圾桶的盖子。

但目前语音识别模块需要在**实时模式**下使用，但是扩展板需要在**上传模式**下使用，如何将这二者结合呢？

能否用**物联网**去解决问题呢？大家不妨动手试一试。

评价与反思

与人分享快乐，能产生更多快乐，尝试与身边的人一起来体验你的垃圾分类语音助手！你还可以在 DF 创客社区中分享作品，听听更多人的建议！社区网址：www.dfrobot.com.cn

问卷调查

尝试让体验过你作品的人做下面的问卷表，看看在别人眼中这个作品的闪光点与不足。

你觉得我设计的语音助手好用吗？（1-5 分）	
你觉得最特别的功能点在哪？	
你觉得还有更好的建议吗？	

自我反思

梳理自己设计项目的思维逻辑、项目在执行层面的问题和经验等等，可以帮助我们形成学习的闭环，加深在项目进行的过程中对知识点和内容的理解，找到可以改进和提高的地方。

优点与不足	1、
	2、
可以从项目中总结的经验	1、
	2、
学到的知识	1、
	2、

项目十一 宠物投食机

养宠物的朋友们，有没有过这样的苦恼，每次出门一整天不在家，没有办法及时给宠物喂食，就会担心它们会不会饿到。要是出差或者是旅行最先考虑的不是把宠物寄养在宠物医院会花多少钱，而是宠物换了环境会不适应、食欲不振，让人非常心疼。



那有没办法让独自在家的宠物也能按时“吃饭”呢？接下来，就让我们想想办法，一起来做一个能让独自在家的宠物也能被好好照顾的宠物投食机！



头脑风暴：宠物投食机需要具备哪些功能？

问题清单

- 1、主人不在家时，如何保证宠物**按时**有粮吃？
- 2、主人在家时，如何**随时**用投食机喂粮？
- 3、当主人不在家，但是想要**远程**给宠物加餐，怎么办呢？
- 4、主人如何知道宠物一天吃了几顿呢？
- 5、实现这些设想需要哪些**硬件**支持？
- 6、如何构造**外观**？

核心知识点

电机

电机（英文：Electric machinery，俗称“马达”）的主要作用是作为用电器或各种机械的动力源，在电路中是一般用字母 M 表示。



物联网

物联网就是将现实世界中的物体连到互联网上，使得物与物、人与物可以很方便的互相沟通。举一个简单的例子，比如**共享单车**，单车上面的锁连上网后就不再需要钥匙开锁，只需要用手机扫码，锁就能通过网络收到开锁的命令自己开锁。

物联网的本质还是互联网，只不过终端不再是计算机，而是嵌入式计算机系统及其配套的传感器。只要有硬件或产品连上网，发生数据交互，就叫物联网。



规划方案

功能分解

在宠物投食机中最重要的功能就是**投放粮食**，这里假定我们使用**电机**完成投放粮食的动作，电机转动即可投放，当然需要配合外观搭建才能实现功能。

从前面的头脑风暴中，可以对宠物投食机作功能分解如下。

分解功能	所需元件
<p>1、定时投喂</p> <p>指定间隔时间使电机转动，实现定时投喂。</p>	<p>电机</p> 
<p>2、触发投喂</p> <p>为了随时给宠物喂粮，增加一个外部触发按钮，主人或宠物随时按下按钮，触发投喂。</p>	<p>按钮（数字开关，按下为 1，松开为 0）</p> 
<p>3、远程投喂</p> <p>当主人在外想要给宠物加餐时，通过物联网在电脑或手机端发送指定消息控制实现远程投喂。</p>	<p>使用物联网平台编程实现</p>
<p>4、记录喂食次数</p> <p>每喂食一次，向物联网平台自动发送投喂次数，避免喂食过多，宠物吃撑。</p>	<p>使用物联网平台编程实现</p>
<p>5、更多创意</p> <p>例如：检测投食机的余粮，提醒主人及时补粮。如果加入摄像头，还可以实时观测宠物吃食情况。</p>	<p>超声波(探测距离)</p> 

构造外观

画出外观草图	所需耗材、工具
例如：	例如：硬纸、牛皮纸、双面胶、透明胶等... 1、 2、 3、

问题记录

在后面项目实施过程中，可能会遇到各种各样的困难，尝试在下表中记录你遇到的问题和解决办法，便于以后出现类似问题时能更好的面对。

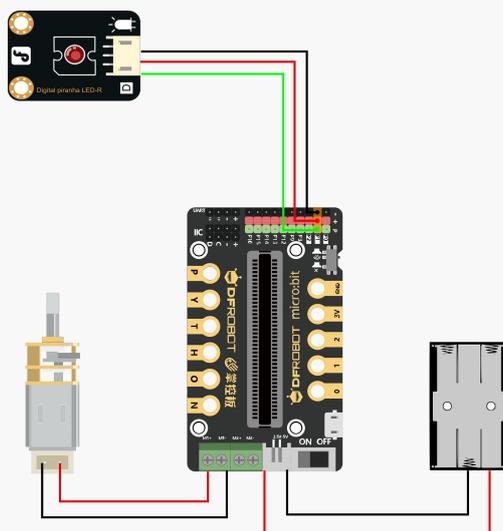
遇到的困难	你的解决方案
1、	1、
2、	2、
3、	3、
...	...

功能实现

功能 1: 定时投喂

这里通过电机的转动与否表示是否投喂，当电机转动时，表示宠物投食机投放粮食；电机停转则停止投放。加入一个 LED 灯作为喂食指示灯，喂食时灯亮，否则灯灭。

连线图：



* 使用电机时，扩展板外需接电池盒。

软件设置：使用电机需要打开 Mind+ 软件的“扩展”，在“扩展板”下点击加载“掌控扩展板”。



程序：

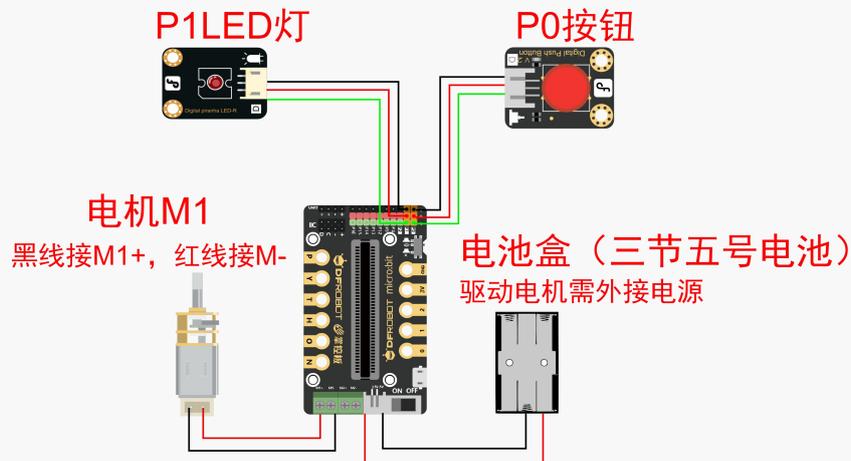


运行结果：每隔 5 秒，电机转动 1 次。“喂食”函数中定义电机以 50 的速度转动 1 秒，后面项目实现过程中可根据实际需求进行修改，结合外观构建达到最佳的效果。

功能 2：触发投喂

这里需要实现的功能是在**定时投喂**基础上额外增加**按钮触发投喂**。并且定时投喂与触发投喂**互不干扰**，即定时投喂过程中，按下按钮随时触发投喂；触发投喂时也不会影响定时投喂的时间间隔。

连线图：

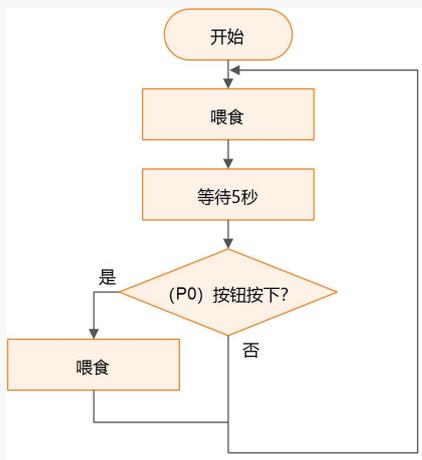


思考一下，如何实现呢？让我们来写几个程序试验一下。

试验程序 1: 如果在功能 1 程序基础上, 加入**对按钮的判断**是否能实现功能呢?



运行结果: 定时投喂功能正常, 按钮几乎不起作用, 无法实现触发投喂。这是为什么呢? 让我们一起画个流程图, 分析一下!



我们知道, 程序运行顺序是**自上而下, 逐条执行**。

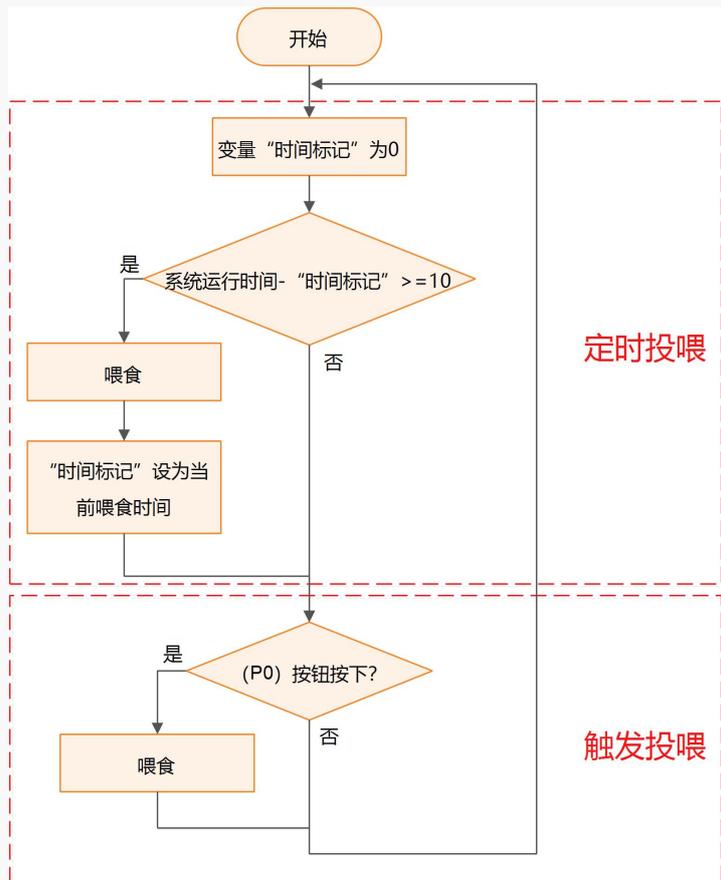
如左流程图, 程序开始后, 先执行“喂食”, 然后“等待 5 秒”, 等待的这段时间什么都不干, 等待结束后, 立马判断“按钮按下?”, 然后根据结果迅速回到“喂食”, 如此循环往复。

在“等待 5 秒”期间, **无法判断按钮状态**, 所以无法实现**触发投喂**。并且由于程序在运行每条指令时速度非常快, 所以在等待结束后判断按钮阶段, 往来不及按下按钮, 程序就立马又进入喂食、等待 5 秒了。所以无法实现预期功能。

思考一下, 既然“等待 5 秒”时**无法判断按钮状态**, 那不用等待可以吗? 如果不用等待, 又如何实现计时呢?

其实在程序中, 需要实现让两次喂食的**时间间隔 5 秒**, 思考一下, 如果我们可以获得**第一次喂食的时间点**, 然后不断判断当下时间与**第一次喂食的时间点**的间隔是否达到 5 秒, 进而决定是否触发**下一次喂食**, 是否可以实现功能呢? 那么如何获得时间呢? 那就是使用 **系统运行时间!**

通过使用**变量**获取 记录**第一次喂食时间点**, 随着时间推移, 不断判断与第一次喂食时间点的**差值**是否达到指定时间, 当达到指定时间则开始下一次定时喂食。在此期间, 几乎没有任何等待时间, 所以加入触发投喂的判断后, 两者互不干扰。流程图分析如下。



试验程序 2: 使用  标记喂食时间点

运行结果：程序中使用串口方便查看变量“时间标记”，可以发现每隔 11 秒左右会喂食一次，其中 10 秒为等待时间，1 秒为喂食时间。在此期间，随时按下按钮，即可实现触发投喂。

通过这样的**标记时间法**，即可实现定时投喂与触发投喂互不干扰！

功能 3: 远程投喂

1、进入 esayiot 平台注册自己的账号，并创建设备。

中国服务器：<http://iot.dfrobot.com.cn>

全球服务器：<http://iot.dfrobot.com>

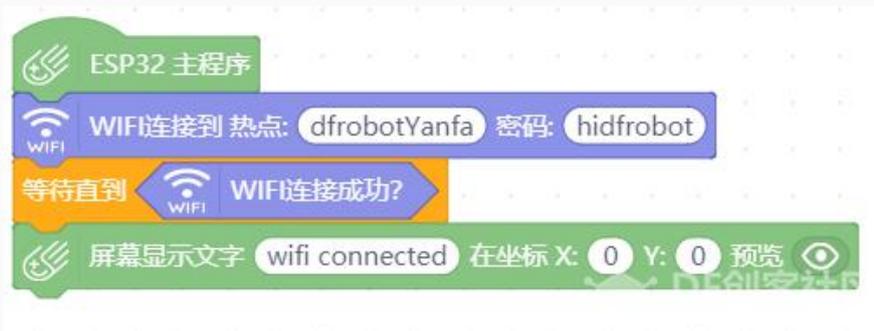
注册完成登陆自己账号进入物联网平台，进入物联网平台后，点击顶部菜单的“工作间”，在工作间里即可新建自己的设备。可以新建多个设备，但是 Mind+ 最多只支持同时对 5 个设备发送和接收消息。



(图中标记的参数需要对应填入到 mind+ 的 MQTT 初始化 block 中。)

2、打开 mind+ 编写程序 (使用版本 V1.5.2 及以上)

(1) 连接 wifi, 无论使用哪一个物联网平台都需要先连接 wifi。



(2) 初始化 MQTT



3、通过使用物联网，主人可在电脑或手机端发送指定消息，实现远程投喂，这里我们选择 Easy IoT，注意关于 WiFi 和 MQTT 具体参数的设置按照自己本身的进行设置。

* 关于 Easy IoT 的介绍和用法详见掌控板入门教程 14 智能物联手表。

在功能 2 程序基础上，程序修改如下：

The image shows a Scratch-style code editor for an ESP32 microcontroller. The code is organized into two main sections:

- ESP32 主程序 (Main Program):**
 - Starts with a block: `屏幕显示文字` (Screen display text) with the text "宠物喂食机" (Pet Feeder) and coordinates X: 24, Y: 22.
 - Followed by a `等待 3 秒` (Wait 3 seconds) block.
 - Then a `屏幕显示为 全黑` (Screen display as all black) block.
 - Next is the `物联网初始化` (IoT initialization) block.
 - Then a `设置 时间标记 的值为 系统运行时间` (Set time marker value to system runtime time) block.
 - A `循环执行` (Loop execution) block contains:
 - `定时投喂` (Scheduled feeding)
 - `触发投喂` (Trigger feeding)
 - `远程投喂` (Remote feeding)
- 定义 远程投喂 (Define Remote Feeding):**
 - Starts with an `如果 变量 消息 = feed 那么执行` (If variable message = feed then execute) block.
 - Inside the if-block, there is a `喂食` (Feed) block.
 - Followed by a `设置 消息 的值为` (Set message value to) block with a dropdown menu.

Two callout boxes provide additional context:

- A callout pointing to the IoT initialization block states: "由于物联网初始化耗时较长，这里将变量“时间标记”设为系统运行时间，相当于完成物联网连接后，定时投喂才启动" (Since IoT initialization takes a long time, here the variable "time marker" is set to system runtime time, equivalent to starting scheduled feeding after IoT connection is complete).
- A callout pointing to the remote feeding logic states: "当电脑或手机端在 Easy IoT 平台发送消息“feed”，则给宠物喂食" (When a computer or mobile terminal sends the message "feed" on the Easy IoT platform, then feed the pet).

对应填写自己的账号信息

物联网平台: Easy IoT

参数:

服务器地址: 中国

lot_id: BynXtXRv4

lot_pwd:

Topic_0: rkKMI9ZtN

定义 物联网初始化

WIFI 连接到 热点: DFRobot-guest 密码: dfrobot@2017

屏幕显示文字 WIFI等待连接 在坐标 X: 16 Y: 22 预览

等待直到 WIFI WIFI 连接成功?

屏幕显示文字 WIFI连接成功 在坐标 X: 16 Y: 22 预览

MQTT 初始化参数

MQTT 发起连接

屏幕显示文字 MQTT等待连接 在坐标 X: 16 Y: 22 预览

等待直到 MQTT MQTT 连接成功?

屏幕显示文字 MQTT连接成功 在坐标 X: 16 Y: 22 预览

等待 1 秒

屏幕显示为 全黑

当 Topic_0 接收到 MQTT消息

设置 消息 的值为 MQTT消息

“MQTT消息”的数据类型为字符串。在“变量”下，新建字符变量“消息”存储“MQTT消息”

运行结果：程序运行后，掌控板屏幕会显示文字“宠物投食机”，然后进行 WIFI 和 MQTT 连接，完成连接后，实现功能如下。

定时投喂	每隔固定时间，电机转动一次
触发投喂	当按下按钮，电机转动一次
远程投喂	在 Easy IoT 中，发送消息“feed”，电机转动一次

关于功能 4 使用物联网平台反馈记录喂食次数，程序实现较为简单，只需要新增一个变量在“喂食”函数中，每次触发“喂食”函数，变量加 1，然后将这个变量作为 MQTT 消息发送到 Easy IoT 即可。这里不再做程序介绍。

构造外观

构造外观有多种方法，下图仅作参考：

外观材料：空饮料瓶、牛皮纸、废纸、铜丝、雪糕棒

工具：热熔胶、透明胶



评价与反思

与人分享快乐，能产生更多快乐，尝试让亲朋好友的宠物都亲身体验一下你的投食机，看看能不能满足它们的需求！你还可以在 DF 创客社区中分享作品，听听更多人的建议！社区网址：www.dfrobot.com.cn

问卷调查

尝试让体验过你作品的人做下面的问卷表，看看在别人眼中这个作品的闪光点与不足。

你觉得我设计的宠物投食机好用吗？（1-5 分）	
你觉得最特别的功能点在哪？	
你觉得还有更好的建议吗？	

自我反思

梳理自己设计项目的思维逻辑、项目在执行层面的问题和经验等等，可以帮助我们形成学习的闭环，加深在项目进行的过程中对知识点和内容的理解，找到可以改进和提高的地方。

优点与不足	1、
	2、
可以从项目中总结的经验	1、
	2、
学到的知识	1、
	2、

项目十二 植物监测仪

随着社会的发展和水平的提高，人们逐渐追求高质量的生活，很多人会在家里或办公室种植一些花花草草以净化空气，陶冶情操。

但是很多养花的小伙伴会因为出差、旅游或者一些其他的原因，不能及时照料它们，短时间可能会影响其正常生长，长时间甚至会死亡。



基于上述情况，让我们制作一个**植物监测仪**，24 小时全方位照料花花草草茁壮成长！



在进行项目制作之前，首先想一想植物生长过程中可能会遇到哪些困难呢？

从解决这些困难入手，或许就是我们要做的植物监测仪！

头脑风暴：植物监测仪需要解决植物生长过程中的哪些问题？

问题清单

- 1、植物的生长离不开**水分**，如何确保植物有**充足的水分供给**呢？
- 2、**光合作用**帮助植物健康生长，如何在室内环境也让植物拥有**足够的光照**呢？
- 3、对于名贵的花卉，温度和空气湿度也会影响其生长，如何检测**空气温湿度**，并帮助植物营造合适的生长环境呢？
- 4、主人离家时，如何**远程监控**植物的土壤湿度、光照强度、空气温湿度等各项数据？
- 5、实现这些功能需要哪些**硬件**？
- 6、如何构造**外观**？

核心知识点

DHT11 温湿度传感器

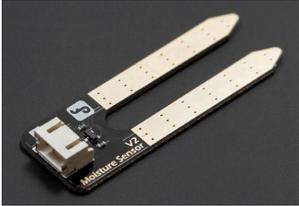
DHT11 温湿度传感器是将空气温湿度检测结合为一体的传感器。

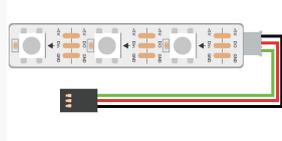
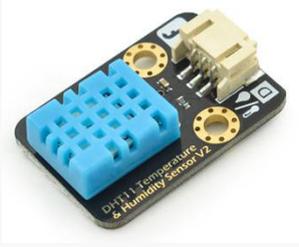


规划方案

功能分解

根据前面的头脑风暴，植物监测仪可以具备检测土壤水分、光照强度、空气温湿度的功能，并且能够根据各项数据对植物进行自动浇水、自动补光、降温或降温。

分解功能	所需元件
<p>1、自动浇水</p> <p>当土壤缺水时，自动浇水。</p> <p>利用土壤湿度传感器检测土壤水分，当水分不足时，通过转动舵机，触发浇水装置。</p>	<p>土壤湿度传感器(检测土壤湿度)</p>  <p>舵机 (在 0-180 度之间转动)</p> 
<p>2、自动补光</p> <p>当植物生长环境的光照不足时，自动补光。</p> <p>利用掌控板自带的环境光传感器检测光线强度，当光强较弱时，点亮 RGB 灯带。</p>	<p>环境光传感器 (监测光线强度)</p> 

	<p>RGB 灯带 (含 7 个 RGB 灯, 可显示任意颜色)</p> 
<p>3、自动通风</p> <p>当植物生长环境的温度或湿度比较高时, 自动通风。</p> <p>利用 DHT11 温湿度传感器检测空气温度、湿度, 当温度或湿度较高时, 启动电机带动风扇, 来降温或降湿。</p>	<p>DHT11 温湿度传感器 (检测空气温湿度)</p>  <p>电机</p> 
<p>4、数据监测</p> <p>主人远程查看各项数据。</p> <p>在 Easy IoT 中定义 4 个 Topic, 通过编程分别向 Topic 发送水分、光照、温度、湿度的数据。</p>	<p>使用物联网平台编程实现</p>
<p>5、更多创意</p> <p>例如: 使用超声波传感器检测浇水器的水位, 水位过低时提醒用户。</p>	<p>超声波(探测距离)</p> 

构造外观

画出外观草图	所需耗材、工具
例如：	例如：饮料瓶、吸管、雪糕棒、废纸盒、热熔胶等... 1、 2、 3、

问题记录

在后面项目实施过程中，可能会遇到各种各样的困难，尝试在下表中记录你遇到的问题和解决办法，便于以后出现类似问题时能更好的面对。

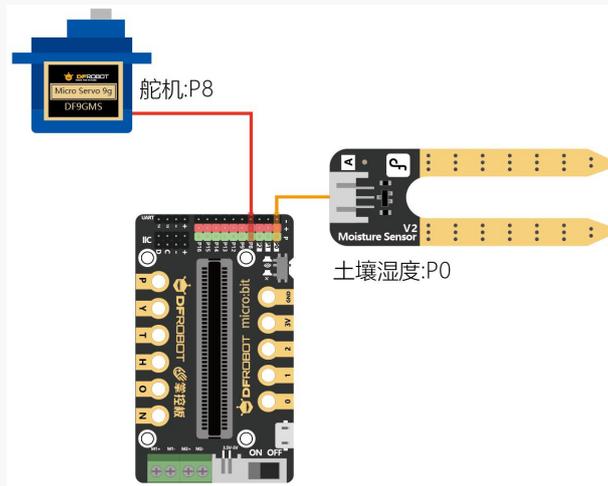
遇到的困难	你的解决方案
1、	1、
2、	2、
3、	3、
...	...

功能实现

功能 1：自动浇水

通过土壤湿度传感器检测植物是否缺水，通过转动舵机触发浇水装置。触发方式有多种，例如舵机摇臂带动浇水壶，摇臂向下时，使水壶向下浇水。

连线图：



* 使用舵机时，需在 Mind+ “扩展” 中，在 “执行器” 下，点击加载 “180 度舵机”。

程序：

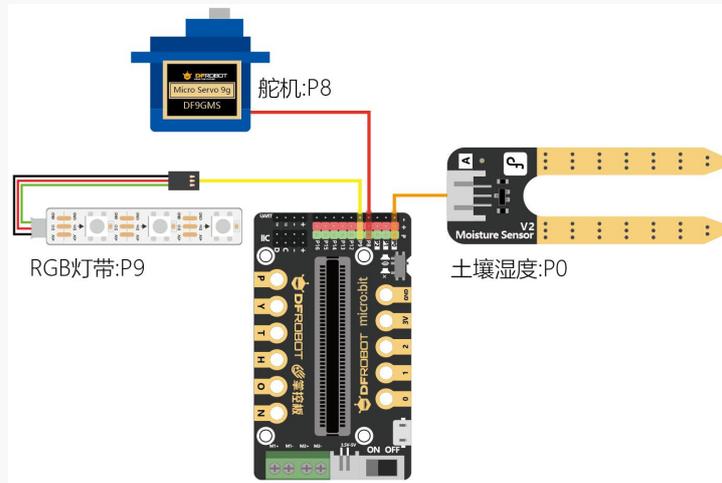
The screenshot shows a Mind+ program for an ESP32 microcontroller. The program starts with an "ESP32 主程序" block, followed by "屏幕显示文字 植物监测仪 在坐标 X: 24 Y: 22 预览". A "等待 3 秒" block is then used. The screen is set to "全黑" (All Black). A "循环执行" (Loop) block contains two "屏幕显示文字" blocks: one for "屏幕显示实时土壤湿度" at X: 48, Y: 0, and another for "合并 水分: 读取模拟引脚 P0" at X: 0, Y: 0. A callout bubble indicates "用6个空格刷新显示“引脚P0”的读值". Below the loop is an "自动浇水" block. A "定义 自动浇水" block contains an "如果" (If) condition: "读取模拟引脚 P0 < 200 那么执行". If true, it sets "设置 P8 引脚伺服舵机为 30 度" (soil moisture is low, servo turns to watering angle). Otherwise, it sets "设置 P8 引脚伺服舵机为 150 度" (soil moisture is normal, servo turns to other angle).

运行结果：屏幕显示土壤湿度值，当土壤湿度较小时，舵机转动。

功能 2：自动补光

利用掌控板自带的环境光传感器检测光线强度，当光强较弱时，点亮 RGB 灯带进行补光。

连线图：



在功能 1 程序基础上，“自动浇水”函数不变，其他程序修改如下：

ESP32 主程序

屏幕显示文字 植物监测仪 在坐标 X: 24 Y: 22 预览

等待 3 秒

屏幕显示为 全黑

初始化 RGB灯引脚 P9 灯总数 7 亮度 255

循环执行

屏幕显示文字 在坐标 X: 48 Y: 0 预览

屏幕显示文字 在坐标 X: 48 Y: 16 预览

屏幕显示文字 合并 水分: 读取模拟引脚 P0 在坐标 X: 0 Y: 0 预览

屏幕显示文字 合并 光强: 读取环境光强度 在坐标 X: 0 Y: 16 预览

自动浇水

自动补光

定义 自动补光

如果 读取环境光强度 < 50 那么执行

RGB灯引脚 P9 灯号 0 到 6 显示颜色

光线较弱时，点亮灯带，进行补光

否则

RGB灯引脚 P9 全部熄灭

光线较强时，熄灭灯带

运行结果：屏幕显示实测土壤湿度值、光线强度值；当光线较弱时，点亮 RGB 灯带。

功能 3: 自动通风

利用 DHT11 温湿度传感器检测空气温度、湿度，当温度或湿度较高时，启动电机带动风扇，来降温或降湿。

在功能 2 程序基础上，“自动浇水”、“自动补光”函数不变，其他程序修改如下：

The image shows a Scratch-style code editor for an ESP32 microcontroller. The main program starts with an 'ESP32 主程序' block, followed by '屏幕显示文字 植物监测仪' at coordinates (24, 22), a 3-second wait, and '屏幕显示为 全黑'. It then initializes RGB LEDs on pin P9 with 7 LEDs and a brightness of 255. A loop contains '屏幕显示', '自动浇水', '自动补光', and '自动通风' blocks. The '自动通风' block is defined as follows: an 'if' condition checks if temperature (read from P1 on dht11) is greater than 28 or humidity is greater than 70. If true, it sets motor M1 to 200 RPM clockwise. If false, it stops motor M1. A 0.1-second wait follows with a note: '使用温湿度传感器时，建议加延时，便于数据正确读取'. A separate '定义 屏幕显示' block defines a function to display data: '水分' (0,0), '光强' (0,16), '温度' (0,32), and '湿度' (0,48).

将屏幕显示部分定义为一个函数，简化主程序
 屏幕显示各类数据实测值：
 ■ 第一行：水分
 ■ 第二行：光强
 ■ 第三行：温度
 ■ 第四行：湿度

运行结果：屏幕显示实测土壤湿度、光线强度、空气温湿度；当温度或湿度较高，启动电机。

功能 4：数据监测

通过使用物联网，主人可在电脑或手机端远程查看植物的各项数据。

在 Easy IoT 中定义 4 个 Topic，分别向 Topic 发送水分、光照、温度、湿度的数据，这样在每个 Topic 中，可以对应查看各个数据的折线图和数值。

* 关于 Easy IoT 的介绍和用法详见掌控板入门教程 14 智能物联手表。

在功能 3 程序基础上，“自动浇水”、“自动补光”、“自动通风”、“屏幕显示”函数不变，其他程序修改如下：



定义 物联网初始化 WiFi和Easy IoT连接设置，注意对应填写自己的WiFi和Easy IoT信息



The Scratch code block includes the following steps:
1. WiFi 连接到 热点: DFRobot-guest 密码: dfrobot@2017
2. 屏幕显示文字: WiFi等待连接 在坐标 X: 16 Y: 22 预览
3. 等待直到: WiFi 连接成功?
4. 屏幕显示文字: WiFi连接成功 在坐标 X: 16 Y: 22 预览
5. MQTT 初始化参数
6. MQTT 发起连接
7. 屏幕显示文字: MQTT等待连接 在坐标 X: 16 Y: 22 预览
8. 等待直到: MQTT 连接成功?
9. 屏幕显示文字: MQTT连接成功 在坐标 X: 16 Y: 22 预览
10. 等待 1 秒
11. 屏幕显示为: 全黑

在Easy IoT中新建4个Topic，分别记录水分、光照、温度、湿度

物联网平台: Easy IoT
参数:
服务器地址: 中国
lot_id: BynXtXRv4
lot_pwd: *****
Topic_0: sjXas4HWR
Topic_1: rRWBs4NWR
Topic_2: e4aBs4NWR
Topic_3: nvQ712HWg

定义 物联网 设置每隔3秒，向Easy IoT发送一次各类数据，这里采用了**标记时间点**的方法，通过调用“系统运行时间”避免直接使用**延时3秒**，影响程序其他功能



The Scratch code block includes the following steps:
1. 如果: 系统运行时间 - 变量 时间标记 > 3000 那么执行
2. MQTT 发送消息: 读取模拟引脚 P0 至 Topic_0 Topic_0: 土壤湿度
3. MQTT 发送消息: 读取环境光强度 至 Topic_1 Topic_1: 光线强度
4. MQTT 发送消息: 读取 P1 引脚 dht11 温度 至 Topic_2 Topic_2: 温度
5. MQTT 发送消息: 读取 P1 引脚 dht11 湿度 至 Topic_3 Topic_3: 湿度
6. 设置 时间标记 的值为: 系统运行时间 每隔3秒，重置变量“时间标记”

运行结果：除了实现自动浇水、补光、通风以外，在 Easy IoT 中可以远程查看各项数据。如下图，4 个 Topic 分别记录水分、光照、温度、湿度的数据，点开 Topic 可以看到对应数据及折线图。



The image shows the Easy IoT web interface. At the top, there are navigation tabs: 主页, 文档, 工作间, 智玩. On the right, there are links for EVE, 注销, and CN. The main content area displays four data cards for different topics: 水分 (Water), 光照 (Light), 温度 (Temperature), and 湿度 (Humidity). Each card shows the lot_id (52/1000), the topic name, and the topic ID. Below each card are buttons for 发送消息 and 查看详情. A modal window is open for the 光照 (Light) topic, showing a date range selector and a line graph of the data over time.

构造外观

外观结构有多种实现方法，下图仅作参考：

外观材料：空饮料瓶、废纸盒、雪糕棒、吸管、彩纸

工具：热熔胶、裁纸刀



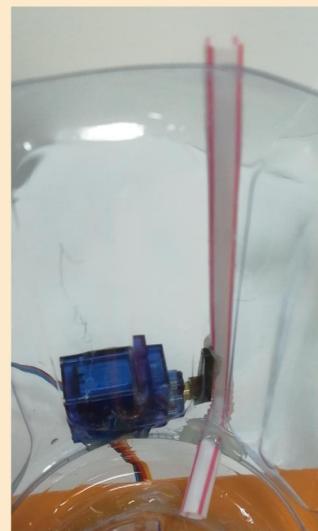
正面示意图



用热熔胶将舵机固定在瓶子上，摇臂固定吸管出水端



在塑料瓶底部侧边开一个洞，插入吸管，用热熔胶封口



更多创意

在这个项目中，为了能够照顾家里的花花草草，我们制作了能够浇水、补光、通风的植物监测仪，你还有其他更好的优化吗？比如监测浇水器的水位，水位过低时发出警报提示主人加水，或者将这个监测仪做成一个智能花盆等。

想一想，如果将植物监测仪改造一下，是否可以应用在大棚种植中呢？再进一步，如果应用在缺水的地方，那么对造林活动是不是也可以起到很大作用！你也可以尝试模拟出这些场景，做出一个更智能的沙盘，发挥无限创意吧！

评价与反思

与人分享快乐，能产生更多快乐，尝试让养花的亲朋好友试着体验一下你的植物监测仪，看看能不能满足需求！你还可以在 DF 创客社区中分享作品，听听更多人的建议！社区网址：www.dfrobot.com.cn

问卷调查

尝试让体验过你作品的人做下面的问卷表，看看在别人眼中这个作品的闪光点与不足。

你觉得我设计的植物检测仪好用吗？（1-5 分）	
你觉得最特别的功能点在哪？	
你觉得还有更好的建议吗？	

自我反思

梳理自己设计项目的思维逻辑、项目在执行层面的问题和经验等等，可以帮助我们形成学习的闭环，加深在项目进行的过程中对知识点和内容的理解，找到可以改进和提高的地方。

优点与不足	1、
	2、
可以从项目中总结的经验	1、
	2、
学到的知识	1、
	2、

项目十三 智能婴儿床

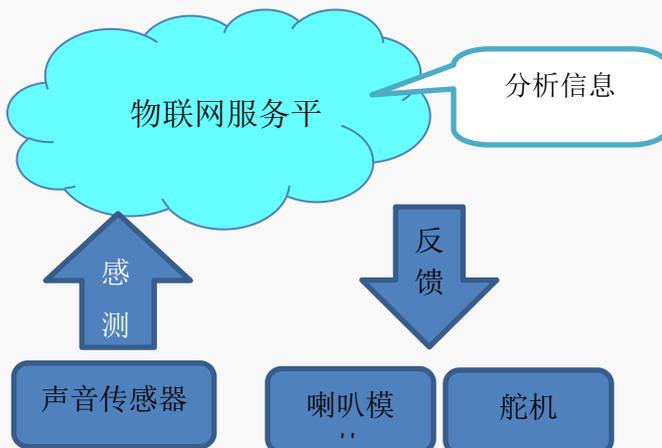
很多家庭迎来新生儿时，呆萌可爱的婴儿令人欣喜不已，同时，它不固定的作息时间和半夜的哭闹又给父母们带来极大的痛苦。如果能够远程哄娃，将会给家长们带来更多的便利。接下来，我们尝试制作智能婴儿摇篮吧！



核心知识点

一、物联网服务平台

制作智能婴儿摇篮，连接声音传感器，可将当前环境的声音强度值传递至物联网服务平台，当接收到声音强度值时，判断婴儿是否哭闹，向掌控板发送信息，发送控制信息令喇叭模块开始播放音乐或停止播放音乐，同时，也可以控制舵机旋转角度实现摇篮摇摆的效果。



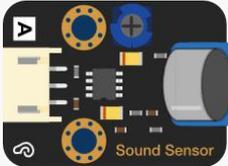
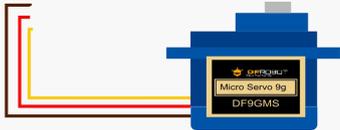
二、掌控板上的 wifi 模块

掌控板芯片自带 wifi 功能，因此不需要外界其他传感器即可实现物联网功能。



规划方案

功能分解

分解功能	所需元件
<p>1、读取声音传感器的值</p> <p>通过声音传感器获取外界声音大小数值</p>	<p>声音传感器</p> 
<p>2、声音控制摇篮</p> <p>由功能一可以看出，声音传感器周围环境较安静时，读取值较小，有较大声音时，读取值较大。通过实验，可设定临界值为 150，当读取值大于 150 时，当前的环境声音值较大，说明婴儿在哭闹。当婴儿哭闹时，可控制舵机多次摆动。</p>	<p>舵机</p> 
<p>3、远程控制摇篮</p> <p>制作远程控制摇篮，当声音传感器读取值大于一定值时，将读取到的声音传感器的值传递至物联网服务器。通过观测数据值，判断婴儿是否哭闹，从而控制播放音乐、摇篮摇摆等。</p>	<p>掌控板 wifi 功能</p> 

构造外观

画出外观草图	所需耗材、工具
例如：	例如：饮料瓶、吸管、雪糕棒、废纸盒、热熔胶等... 1、 2、 3、

问题记录

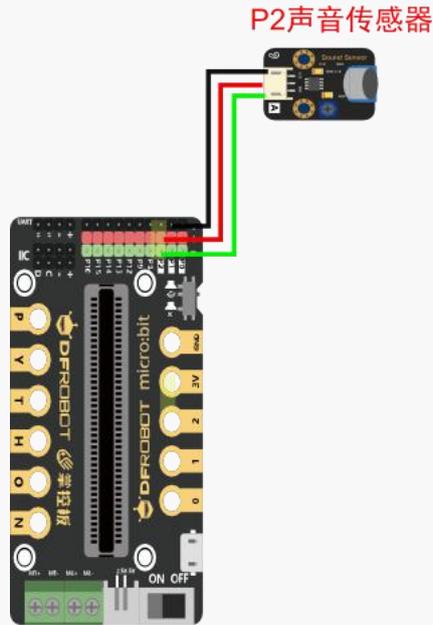
在后面项目实施过程中，可能会遇到各种各样的困难，尝试在下表中记录你遇到的问题和解决办法，便于以后出现类似问题时能更好的面对。

遇到的困难	你的解决方案
1、	1、
2、	2、
3、	3、
...	...

功能实现

功能 1：读取声音传感器的值

通过声音传感器获取外界声音大小数值



1.添加模块

按步骤添加程序编写过程中需要的模块，添加主板——掌控板。

2.读取串口值

- 将 “” 模块拖放至右边的编辑区。该模块可以在串口部分打印输出内容。
- 将 “” 模块拖放至右边的编辑区。声音传感器连接的是 P2 引脚，故将 P0 修改为 P2。
- 将 “” 模块拖放至右边的编辑区。将读取的 P0 引脚值每隔 0.5s 打印一次。

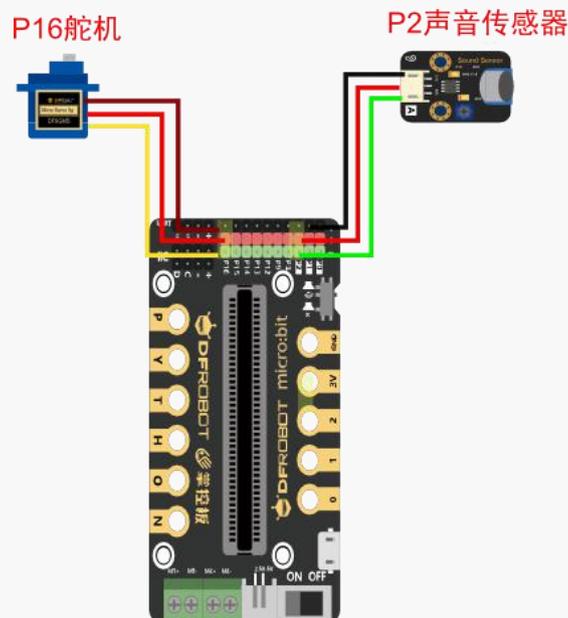


3.烧录代码并打开串口

- 单机 “上传至设备”，并打开串口。可显示当前声音传感器读取的声音强度，不同环境下读取值有所差别。写该教程时，当前安静环境下，声音传感器读取的值在 10~20 之间，有较大噪音时，声音传感器读取的值在 150 以上。



功能 2: 声音控制摇篮



1. 添加模块

按步骤添加程序编写过程中需要的模块，添加主板——掌控板、执行器——舵机模块。



2. 判断声音传感器的临界值

判断读取值和临界值的大小关系，可运用 “” 模块和 “如果...否则...” 模块。



3.控制摇篮摆动

由实验一可以看出，声音传感器周围环境较安静时，读取值较小，有较大声音时，读取值较大。通过实验，可设定临界值为 150，当读取值大于 150 时，当前的环境声音值较大，说明婴儿在哭闹。当婴儿哭闹时，可控制舵机多次摆动。代码设计如下：



功能 3：远程控制摇篮

制作远程控制摇篮，当声音传感器读取值大于一定值时，将读取到的声音传感器的值传递至物联网服务器。通过观测数据值，判断婴儿是否哭闹，从而控制播放音乐、摇篮摇摆等。保持功能 2 的连线方式。

1.添加模块

按步骤添加程序编写过程中需要的模块，添加主板——掌控板、网络服务——WIFI 和 MQTT、执行器——舵机模块。



2. 连接物联网并发送数据

获取的声音传感器的值大于临界值时，说明当前有较大声音，发送当前的声音值至物联网服务器。结合自己的网络，修改相应信息。



3.发送控制信号

- 创建变量“获取信号”，将获取的信号消息存放在该变量中，控制摇篮展示不同的效果。
- 通过 Easy IOT 平台发送指令，控制音乐响起时，发送指令“放”，用数字 1 代替信号值存放在变量中；控制摇篮摇摆时，发送指令“摇”，用数字 2 代替信号值存放在变量中；当需停止播放音乐和摇篮摇摆时，发送指令“停”，用数字 3 代替信号值存放在变量中。具体代码如下：



- 当接收到信号值为“1”时，控制 P0 连接的喇叭模块播放音乐；当接收到的信号值为“2”时，控制舵机 10 度-100 度之间来回切换，形成摇篮摇摆的状态；当接收到的信号值为“3”时，停止播放音乐和舵机角度切换。

ESP32 主程序

- Wi-Fi 连接到 热点: DFRobot-guest 密码: dfrobot@2017
- 等待直到 Wi-Fi 连接成功?
- 屏幕显示文字 wifi连接成功 在第 1 行
- MQTT 初始化参数
- MQTT 发起连接
- 等待直到 MQTT 连接成功?
- 屏幕显示文字 mqtt连接成功 在第 1 行
- 循环执行
 - 如果 读取模拟引脚 P2 > 100 那么执行
 - MQTT 发送消息 合并 声音值: 读取模拟引脚 P2 至 Topic_0
 - 如果 变量 my variable = 1 那么执行
 - 播放音乐 BIRTHDAY 重复 播放一次
 - 如果 变量 my variable = 2 那么执行
 - 设置 P16 引脚伺服舵机为 10 度
 - 等待 0.5 秒
 - 设置 P16 引脚伺服舵机为 100 度
 - 等待 0.5 秒
 - 如果 变量 my variable = 3 那么执行
 - 停止后台播放

参考程序

程序展示

将下面的程序下载到 micro:bit 设备中。

```

ESP32 主程序
Wi-Fi 连接到 热点: DFRobot-guest 密码: dfrobot@2017
等待直到 Wi-Fi 连接成功?
  屏幕显示文字 wifi连接成功 在第 1 行
MQTT 初始化参数
MQTT 发起连接
等待直到 MQTT 连接成功?
  屏幕显示文字 mqtt连接成功 在第 1 行
循环执行
  如果 读取模拟引脚 P2 > 100 那么执行
    MQTT 发送消息 合并 声音值: 读取模拟引脚 P2 至 Topic_0
  如果 变量 my variable = 1 那么执行
    播放音乐 BIRTHDAY 重复 播放一次
  如果 变量 my variable = 2 那么执行
    设置 P16 引脚伺服舵机为 10 度
    等待 0.5 秒
    设置 P16 引脚伺服舵机为 100 度
    等待 0.5 秒
  如果 变量 my variable = 3 那么执行
    停止后台播放
  
```

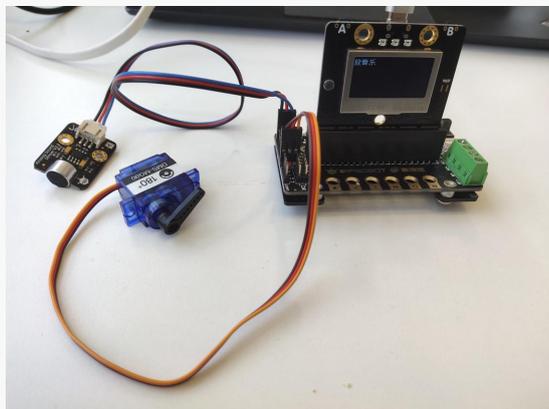


效果展示

- 进入 Easy IoT 平台发送信息给 micro:bit 设备。找到相应的 Topic，点击“发送消息”。
- 将想要发送的指令，填写在对话框中，点击发送，即可将消息发送出去，发送指令“放”，实现播放音乐。



- 当接收到“放”的指令时，可以在下面“最新消息”板块看见新消息“放”，掌控板开始播放音乐。



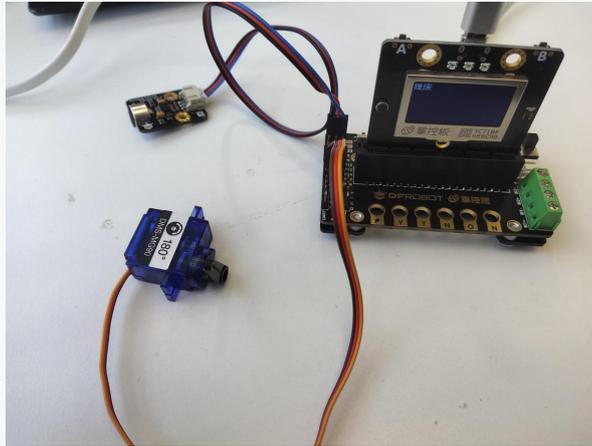
- 发送指令“摇”，实现摇篮摇摆。

发送新消息

为消息加上->前缀代表此消息为纯指令消息，不会被存入数据库。例如“->off”



- 当接收到“摇”的指令时，可以在下面“最新消息”板块看见新消息“摇”，舵机 10 度-100 度之间来回切换，形成摇篮摇摆的状态。



- 发送指令“停”，停止播放音乐和摇篮摇摆。

由于程序运行顺序是自上而下，逐条执行，故该程序中播放音乐和摇篮摇摆交替进行，无法同时进行。

第四章 小车跑啊跑

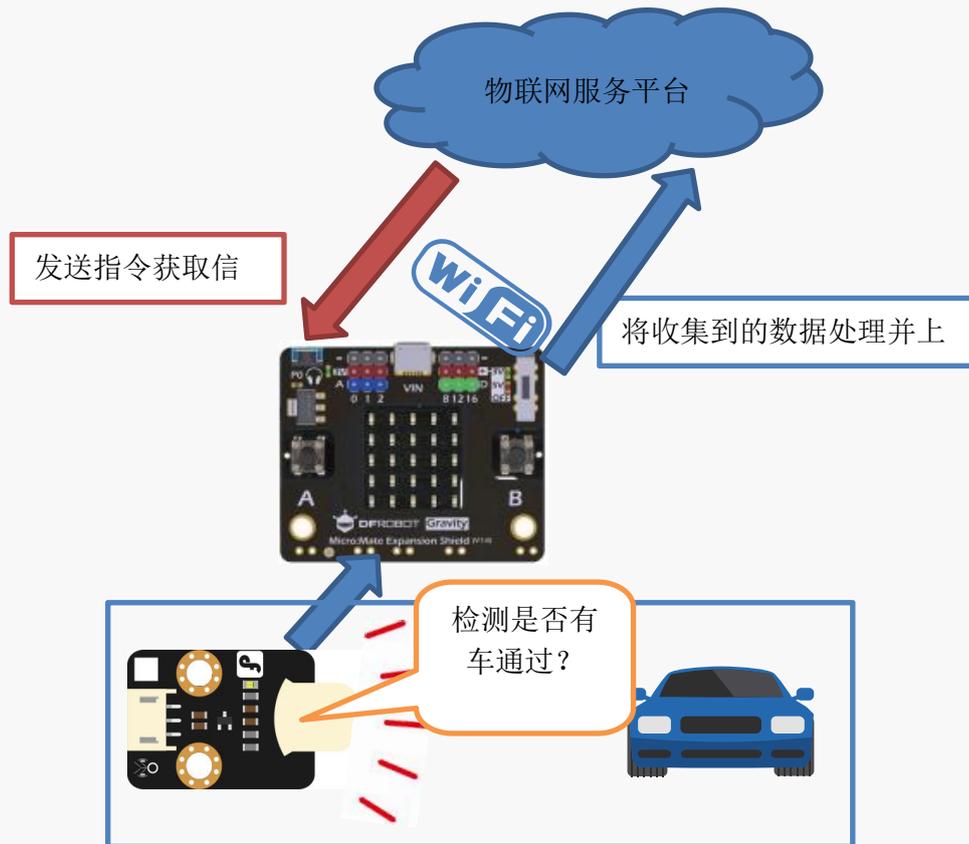
项目十四车流量统计

随着现代化建设的不断发展，人民的生活水平也不断的提高了，以前不敢想象的汽车现在已经进入每家每户了，随之而来的是道路上的车辆数量猛增，伴随而来的问题就是“堵”。所以现在很多地方都建立了自动交通流量观测仪来检测某路段的车流量。



一、物联网服务平台

车流量观测仪，主要利用运动传感器检测车辆是否有车辆通过，然后将处理后的数据通过 WiFi 上传至物联网平台（Easy IoT）。



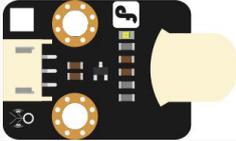
三、掌控板上的 wifi 模块

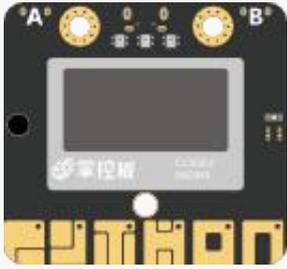
掌控板芯片自带 wifi 功能，因此不需要外界其他传感器即可实现物联网功能。



规划方案

功能分解

分解功能	所需元件
<p>1、热释电传感器功能</p> <p>热释电红外运动传感器能检测运动的人或动物身上发出的红外线，然后输出开关信号，它可以应用于各种需要检测人体运动的场合。</p>	<p>人体红外热释电运动传感器</p> 

<p>2、远程监测</p> <p>利用运动传感器检测车辆是否有车辆通过，然后将处理后的数据通过 WiFi 上传至物联网平台（Easy IoT）。</p>	<p>掌控板 wifi 功能</p> 

构造外观

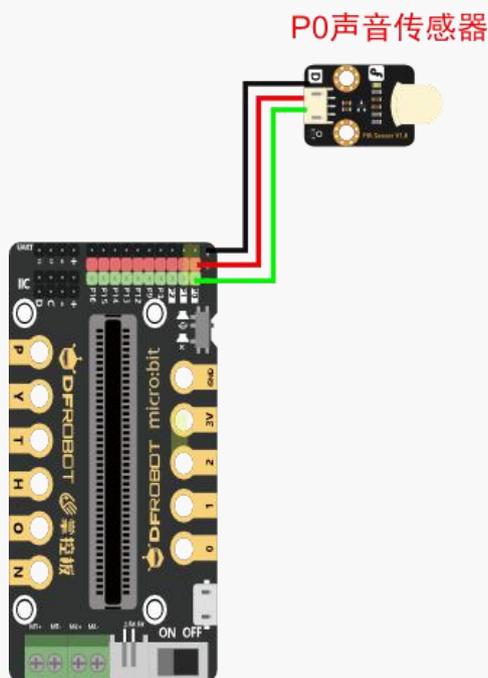
画出外观草图	所需耗材、工具
<p>例如：</p>	<p>例如：饮料瓶、吸管、雪糕棒、废纸盒、热熔胶等...</p> <p>1、</p> <p>2、</p> <p>3、</p>

问题记录

在后面项目实现过程中，可能会遇到各种各样的困难，尝试在下表中记录你遇到的问题和解决办法，便于以后出现类似问题时能更好的面对。

遇到的困难	你的解决方案
1、	1、
2、	2、
3、	3、
...	...

功能实现



1.添加主控板及物联网模块

根据上一节内容添加主板——掌控板、网络服务——WIFI 和 MQTT。

2.wifi 模块初始化及联网测试

Wifi 和 MQTT 的设置操作和项目十一类似，不会可以翻阅项目十一的相关内容进行学习



如果联网成功，在 Easy IoT 平台会收到一条“hello”的消息。如下图所示。

时间	消息
2020/3/17 11:46:27	hello

4.创建两个变量

- 创建两个变量，一个是 “**变量 运行时间**”，用来存放系统的运行时间；另一个是 “**变量 通过的车辆**”，用来记录通过的车辆数量。
- 初始化 “**变量 通过的车辆**”，将通过的车辆数设置为 0。



- 设置运行时间，因为系统运行时间的单位是 ms，我们需要将它换算成分钟或者小时。时间单位的关系：1s=1000ms, 1min=60000ms, 1h=3600000ms。



5.统计通过的车辆数量

- 人体红外热释电运动传感器有两个状态：输出 0 和 1，当有人或者有物体经过的时候，输出 1，没有人或者物体经过的时候就输出 0。利用条件判断语句，判断当人体红外热释电运动传感器的状态为 1 时，说明有车经过，则通过的车辆数加 1。



提示：以前我有遇见过这样的疑问，如果我们不用传感器固有的模块，可以做这个项目吗？是可以的，我们可以直接判断读取到的引脚状态。如下图所示：



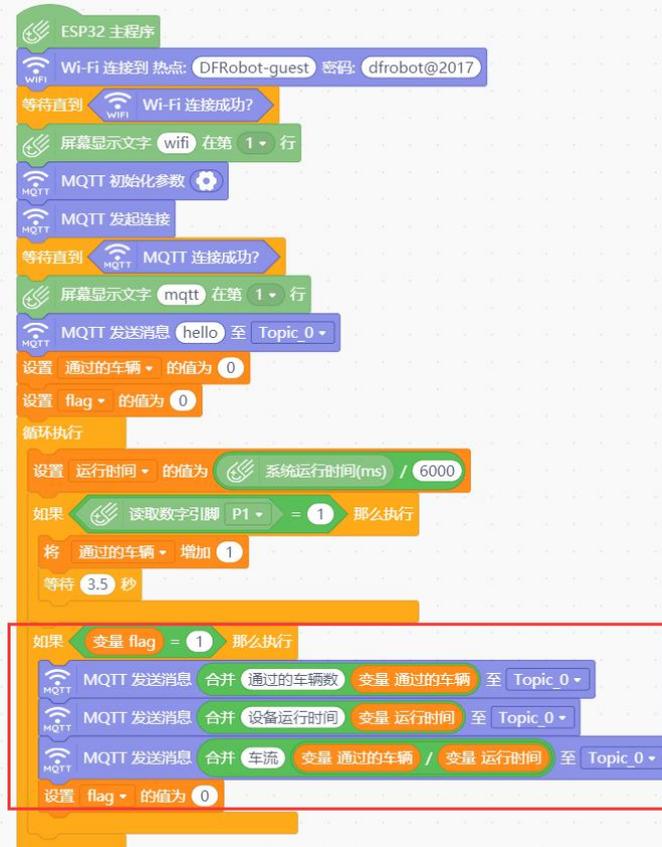
6.查看车流量

- 我们学习了如何统计通过的车辆数量，同时也获取了系统的运行时间，接下来我们就可以计算车流量。并将处理后的数据上传到 Easy IoT 平台上。

- 将 “当接收到 Topic_0 的 MQTT 消息” 模块拖放至编辑区。然后设置条件判断，当 MQTT 消息收到 “车流量” 这个消息就执行相应的内容。



- 设置好条件后，就需要设置执行内容。因为车流量是在一定的时间内，某条公路点上所通过的车辆数，用公式表示为：车流量=通过车辆数/时间。注意这里我们需要新建一个 flag 的变量，该变量用于触发网页端信息的打印，程序内容如下图所示：



参考程序

程序展示

将下面的程序下载到设备中。

```

ESP32 主程序
Wi-Fi 连接到 热点: DFRobot-guest 密码: dfrobot@2017
等待直到 Wi-Fi 连接成功?
  屏幕显示文字 wifi 在第 1 行
MQTT 初始化参数
MQTT 发起连接
等待直到 MQTT 连接成功?
  屏幕显示文字 mqtt 在第 1 行
  MQTT 发送消息 hello 至 Topic_0
  设置 通过的车辆 的值为 0
  设置 flag 的值为 0
  循环执行
    设置 运行时间 的值为 系统运行时间(ms) / 6000
    如果 读取数字引脚 P1 = 1 那么执行
      将 通过的车辆 增加 1
      等待 3.5 秒
    如果 变量 flag = 1 那么执行
      MQTT 发送消息 合并 通过的车辆数 变量 通过的车辆 至 Topic_0
      MQTT 发送消息 合并 设备运行时间 变量 运行时间 至 Topic_0
      MQTT 发送消息 合并 车流 变量 通过的车辆 / 变量 运行时间 至 Topic_0
      设置 flag 的值为 0
  当接收到 Topic_0 的 MQTT消息
    如果 MQTT消息 = "车流量" 那么执行
      设置 flag 的值为 1
    
```

效果展示

- 进入 Easy IoT 网页发送信息给 掌控板 设备。找到相应的 Topic，点击“发送消息”，发送指令“车流量”

时间	消息
2020/3/18 18:13:34	车流量

- 刷新网页可以查看通过的车辆数量、设备运行时间、车流。

2020/3/18 18:13:34	车流量
2020/3/18 17:2:18	车流1.29
2020/3/18 17:2:18	设备运行时间7.00
2020/3/18 17:2:18	通过的车辆数9.00
2020/3/18 17:2:18	车流量
2020/3/18 17:2:15	车流1.50
2020/3/18 17:2:15	设备运行时间6.00
2020/3/18 17:2:15	通过的车辆数9.00

项目十五 风帆车

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。

如何一步一步完成项目呢？一起来头脑风暴吧！

头脑风暴

问题清单

- 1、如何制作一辆车呢？
- 2、小车轮子是固定在车上的吗？如何让轮子转起来呢？
- 3、帆是风帆车捕捉风能的重要工具，怎么制作帆呢？
- 4、由于自然风时有时无，时大时小，且风向不定，用来驱动车辆难顺人意，怎么保证有稳定的风源呢？
- 5、在风力不变的情况下，猜想一下，影响风帆车行驶距离的因素会有哪些？是帆的大小、位置、形状吗？还是车的重量？
- 6、尝试就问题 5 提出一种假设，并通过实验的方法验证自己的假设。

核心知识点

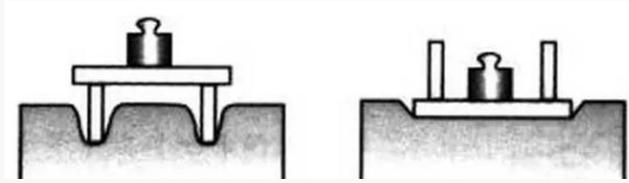
在这个项目中，为了便于了解风的产生和风推动风帆车前进的原理，先来简单学习几个概念。

压强（结合物理学科）

物理学中把垂直作用在物体表面上的力叫做压力。

压强是表示物体单位面积上所受力的大小的物理量。

压强是用来比较压力产生的效果，压强越大，压力的作用效果越明显。如下图，图左受力面积小，压强大，桌子陷的深；图右受力面积大，压强小，陷的浅。



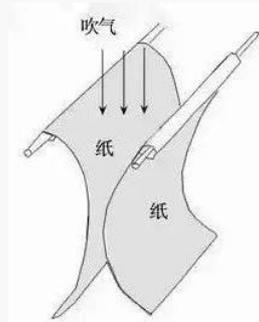
压强在固体、液体、气体中都存在，气体中的压强称为气压。

我们知道，风是由空气流动引起的一种自然现象。而空气的流动主要是由气压差引起的，这又是为什么呢？

伯努利原理

1726年，丹尼尔·伯努利首先提出：“在水流或气流里，如果速度小，压强就大；如果速度大，压强就小”，我们称之为“伯努利原理”。

比如拿起两张抽纸，往中间吹气，会发现纸不但不会向外飘去，反而会被一种力挤压在了一起。因为两张纸中间的空气被我们吹得流动的速度快，压力减小，而两张纸外面的空气没有流动，压力比内侧大，所以外面力量大的空气就把两张纸“压”在了一起。这就是“伯努利原理”原理的简单示范。



同样，在大气中，由于各种原因，不同地方的气压不相同，气压差会导致气体从压强大的地方流向压强小的地方，所以产生了风。

大自然的风总是飘忽不定，难以捉摸，所以聪明的人类发明了电风扇！

为什么风扇的叶片是斜的呢？

想一想，在闷热的夏天，打开电风扇，吹着悠悠的凉风，多么惬意！为什么风扇可以产生风呢？和扇叶的形状有关系吗？试想一下，如果风扇的叶面变成平的，还会产生风吗？



我们知道电风扇的扇叶不是平的，是有一定弯曲的，当风扇转动，叶面增强了风扇前端的空气流速，从而造成两边空气压强不同。根据伯努利原理，风扇前后压强不同，加快后面的空气向前流动，就会形成向前的风！

那为什么风扇前面有风、后面没风呢？那是因为前面的风比较集中，感受的风大，而后面流入的空气来源范围大，流速比较慢，人不太感觉得到。

规划方案

功能分解

在这个项目中，为了让风帆车可以顺利跑起来，需要制作一台风扇提供风源。然后再制作一台风帆车，借助风力行驶起来。在这个过程中，我们将会探索如何延长风帆车的行驶距离，并通过实践制作一辆能跑的最远的风帆车。

分解功能	所需元件
1、制作小车 小车的车轮要可以顺利的转动。	无
2、制作风帆车 为小车加上一面帆，完成风帆车。	无
3、制作风扇 手工制作一台电风扇，需要使用电机作为动力。根据前面的分析，我们知道，风扇的叶面需要做适当的弯曲。	电机 
4、制作一个“循规蹈矩”的风帆车	智能灰度传感器 
5、实验探究：如何延长风帆车行驶距离？ 风帆车的重量，还有帆的大小、位置、形状都有可能影响车的行驶距离，让我们一起通过实验来验证一下。	无

构造外观

小车外观草图	所需耗材、工具
例如:	例如: 塑料瓶盖、雪糕棒、一次性筷子等... 1、 2、 3、
风帆外观草图	所需耗材、工具
例如:	例如: 纸、布等... 1、 2、 3、
风扇外观草图	所需耗材、工具
例如:	例如: 塑料瓶、剪刀等... 1、 2、 3、

问题记录

在后面项目实现过程中，可能会遇到各种各样的困难，尝试在下表中记录你遇到的问题和解决办法，便于以后出现类似问题时能更好的面对。

遇到的困难	你的解决方案
1、	1、
2、	2、
3、	3、
...	...

功能实现

功能 1：制作小车

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。

功能 2：制作风帆车

为小车加上一面帆，小车就可以在风中行驶啦！

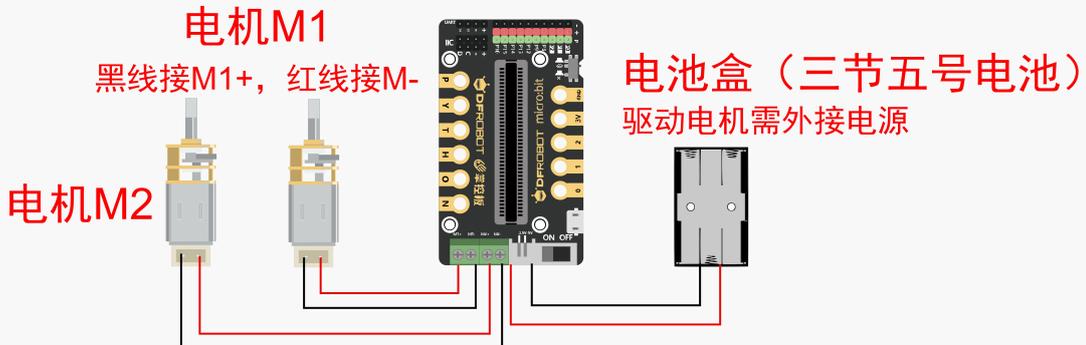
可以将纸或者布剪出三角形、四边形或者其他形状作为帆，然后固定在小车上，完成风帆车的制作。



功能 3：制作风扇

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！

使用电机为风扇提供动力。连线图：



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！

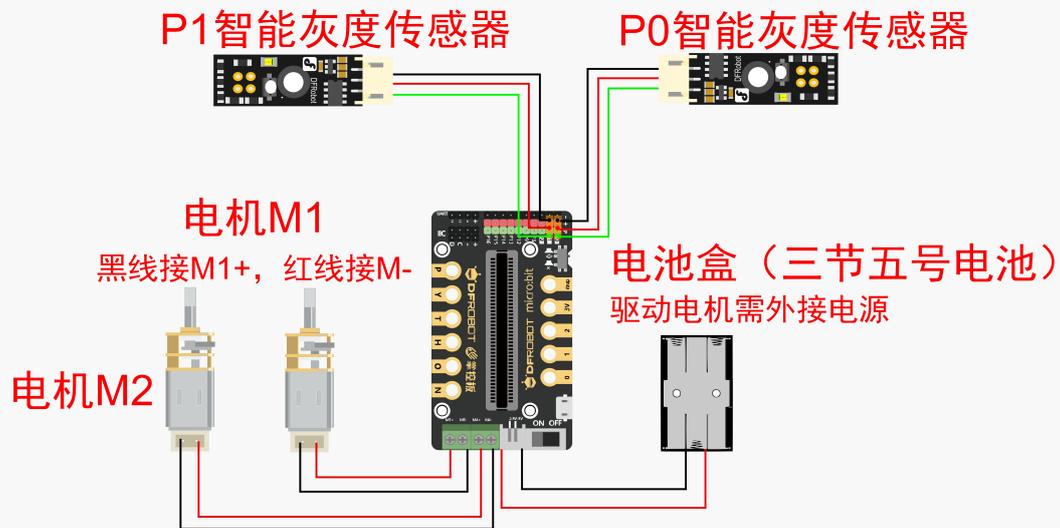
打开做好的风扇，让你的风帆车在风中驰骋！注意调整帆的角度，获得更大的风力！

参考程序



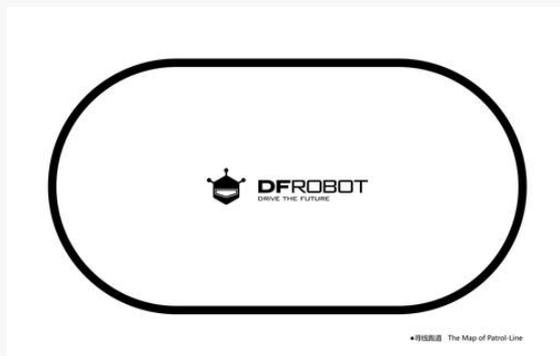
功能 4：“循规蹈矩”的风帆车

STEP1: 在功能 3 的基础上，将两个灰度传感器连接到扩展板上。



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：

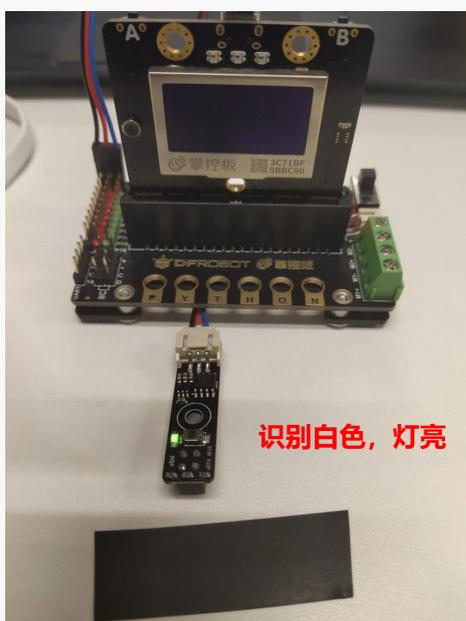
1. 校准所要巡的线(比如黑线):将传感器探头放在黑线上方，单击按键后，待 LED 闪烁完毕后完成校准；
2. 校准背景色(比如白色背景): 将传感器探头放在白色背景上方，双击按键后，待 LED 闪烁完毕后完成校准；
3. 如果快速单击 3 次，LED 闪烁完毕后，则会反转有效电平状态，即原来检测黑线输出为 0 现在变为输出为 1；
4. 快速单击 4 次，LED 反转亮和灭，原本检测黑线为亮变为检测白色背景为亮。

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！



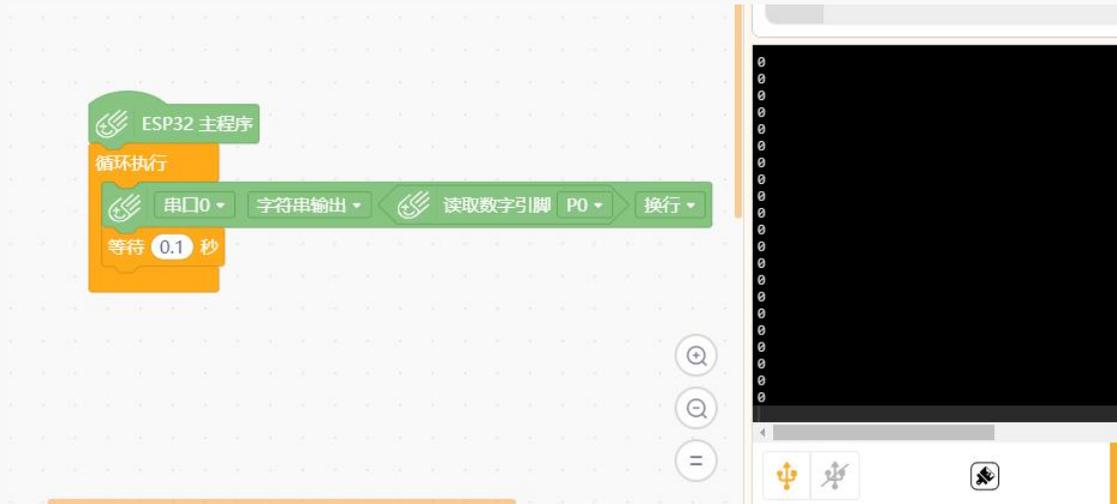
灰度传感器识别到的两种状态，具体实物如下：



识别到白色显示如下：



识别到黑色显示如下：



最终参考程序



功能 5：实验探究：如何延长风帆车行驶距离？

风帆车借助风力就可以行驶，如何延长它的行驶距离呢？

改变帆的大小、位置、形状或者车的重量会有影响吗？尝试提出一个假设，然后通过实验验证你的假设。

因为这里很多因素都会影响行驶距离，所以采用“唯一变量原则”，即只有一个变量而排除其他因素的干扰，从而验证这个唯一变量的作用。

比如，提出假设：帆变大可以延长行驶距离。

接下来，验证假设：制作大小不同的帆，安装在同一辆小车上，保证其他条件不变，然后多次实验，在表格中记录对应数据。

帆的面积 (cm ²)	行驶距离 (cm)

你也可以提出其他的猜想，然后通过这样的实验方法来做验证。

对于上面这个假设，在实际操作过程中会发现，帆的面积比较小的时候，行驶距离短，随着面积增大，行驶距离变大，但是当帆越来越大后，行驶距离又会变短。

思考一下，为什么积极的影响会变成消极的影响呢？

拓展研究：逆风航行

过去的帆船可以逆风航行吗？尝试查阅相关资料，了解帆船逆风航行的原理。

再思考一下，风帆车是否也可以逆风行驶呢？

评价与反思

与人分享快乐，能产生更多快乐，尝试与身边的朋友一起制作风帆车，看看谁得小车跑的最远！你还可以在 DF 创客社区中分享作品，听听更多人的建议！社区网址：www.dfrobot.com.cn

自我反思

梳理自己设计项目的思维逻辑、项目在执行层面的问题和经验等等，可以帮助我们形成学习的闭环，加深在项目进行的过程中对知识点和内容的理解，找到可以改进和提高的地方。

优点与不足	1、
	2、
可以从项目中总结的经验	1、
	2、
学到的知识	1、
	2、

项目十六 无人漫游车

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。

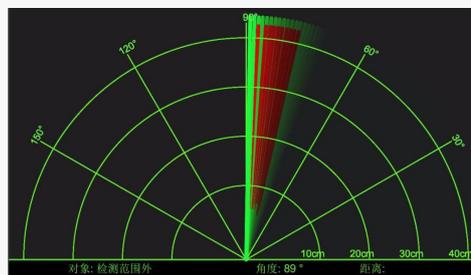
一般的**避障小车**会将**超声波**固定在车头检测障碍，当有障碍物的时候，就自动转左或转右，其实就是让小车一直**向前“看”**。在这个项目中，为了让小车更加“智能”，将会让小车**向四周“看”**，检测周围的障碍物，根据结果**“智能”选择路线**。

如何实现呢？一起来头脑风暴一下吧！

头脑风暴

问题清单

- 1、无人漫游车首先是一辆小车，参照“相扑机器人”，如何组装一辆车？
- 2、**超声波传感器**可以帮助小车**“看见”**障碍物，那么如何让它**往四周“看”**呢？借助什么硬件？
- 3、想一想，**往四周“看”**的**超声波**是不是有点像一个**雷达**？尝试将它简化为一个雷达，帮助小车判断周围路况，比如当小车前进方向的四周都被阻碍时，引导小车调头，原路返回！
- 4、**雷达**（详细介绍见核心知识点）可以通过 360 或 180 度扫描，反馈周围障碍物的**距离和方位**，并使用图形显示出来。刚好掌控板上有一块显示屏，能否在掌控板上做这样一个雷达网图呢？



5、如何确定障碍物的**方位**呢？能否借助**舵机**带动超声波看见四周？考虑到这里的舵机是 0-180 度转动，所以可以采用 180 度的雷达网图。

- 6、如何画出雷达网图？如何实时画出障碍物的位置？这可能需要一点数学知识喔！
- 7、如何“智能”选择路线？

核心知识点

雷达

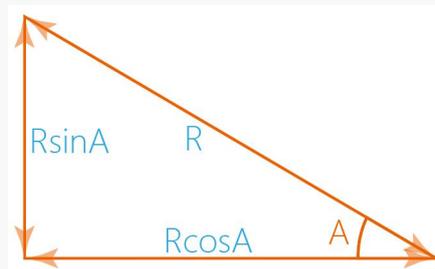
雷达，是英文 Radar 的音译，源于 radio detection and ranging 的缩写，意思为“无线电探测和测距”，即用无线电的方法发现目标并测定它们的空间位置。因此，雷达也被称为“无线电定位”。



有的雷达可以 360 度转动，用于定位附近的障碍物的距离和方位，并将数据传回电脑，通过雷达扫描图来实时显示。图中绿色圆点即为检测到的障碍物。

三角函数（结合初中数学）

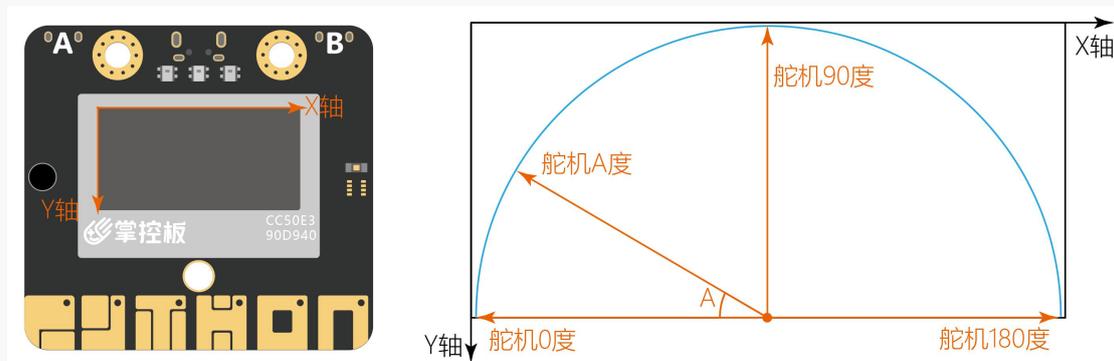
在初中数学中讲过，三角函数是将直角三角形的内角和它的两个边长的比值相关联，若已知斜边长 R 和其中一个内角的角度 A ，则两条直角边长可用三角函数表示，如下图。



掌控板屏幕的坐标系

在 Mind+ 中，掌控板屏幕上建立的坐标系如下左图。前面讲过，使用超声波可以确定障碍物的距离。那么如何确定方位呢？

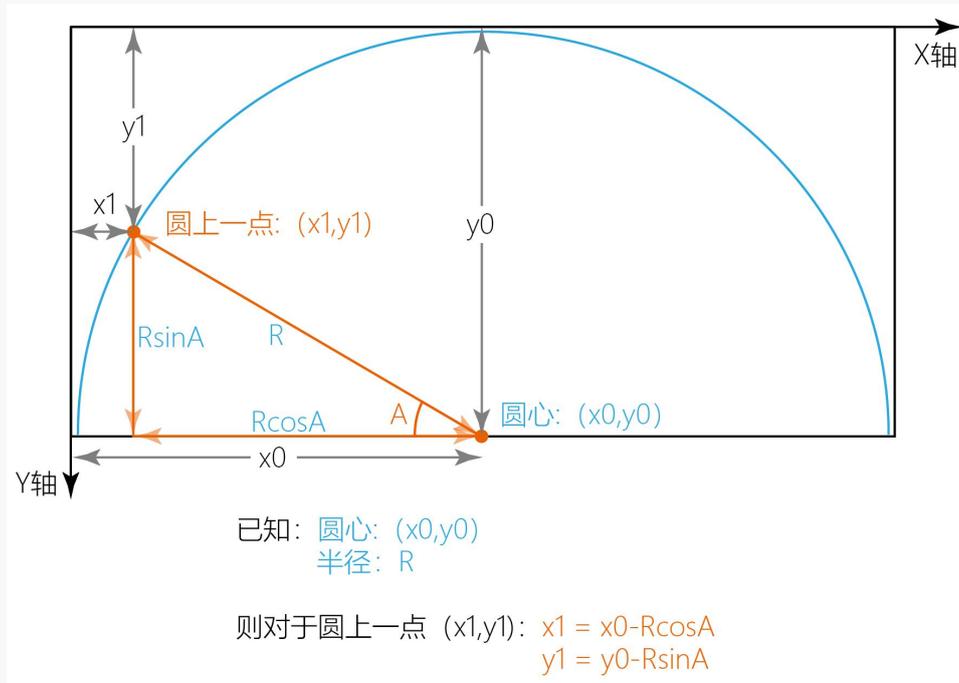
其实方位就是舵机转动的角度。舵机 0-180 度转动，在坐标系中可以表示为右图。



确定障碍物的坐标

将舵机转动的角度结合超声波探测的距离，即可在坐标系中定位障碍物的位置。比如在角度 A 时，超声波测得障碍物距离为 R ，通过一定数学运算即可获得该障碍在坐标系中的位置。

参照下图，在雷达网图中，圆心的位置 (x_0, y_0) 是确定的，如果知道舵机角度 A 和距离 R ，那么由三角函数的知识，即可求得障碍物位置坐标 (x_1, y_1) 的值。



当超声波探测距离 R 随舵机角度 A 实时变化，根据上面的数学关系，障碍物位置坐标 (x_1, y_1) 也会实时变化。

通过上面这个方法，可以在屏幕上画出简易的雷达网图，还可以实时画出超声波检测的障碍物的位置。一起来看看如何实现吧！

规划方案

功能分解

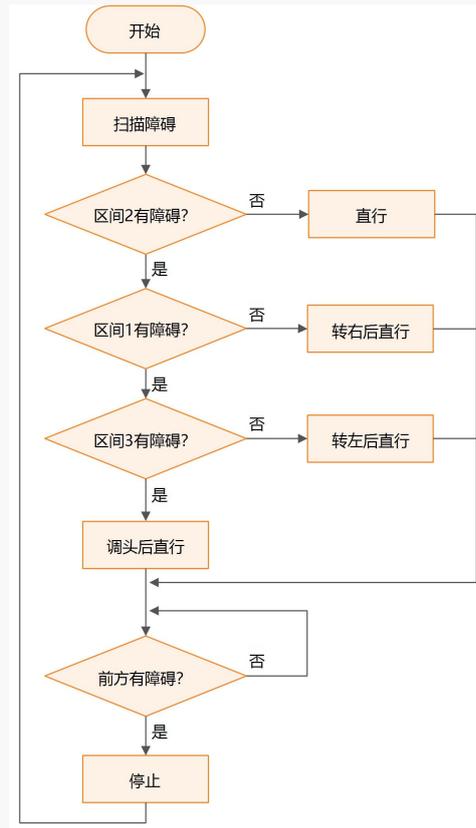
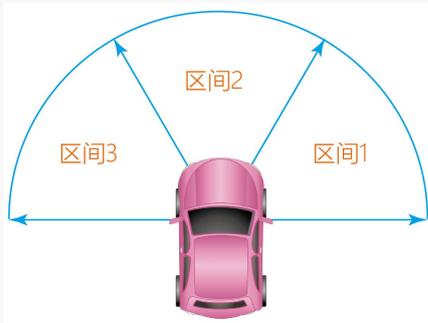
这个项目将制作一台无人漫游车，使用舵机带动超声波进行转动，模拟雷达功能，在屏幕上实时显示雷达扫描图，并根据扫描结果，“智能”选择没有障碍的路线。

在项目实现的过程中主要有两个难点：

- 1、如何实现雷达扫描图？
- 2、如何“智能”行走？

先来解决问题 2：如何“智能”行走？ 这里的“智能”其实是指小车行走的逻辑，在应对不同的路况时，做出合适的判断。那么行走的逻辑是什么呢？

为了简化分析，我们将小车的前进方向按照左、中、右划分为 3 个区间。如果采用优先转右的思路行走逻辑如流程图。

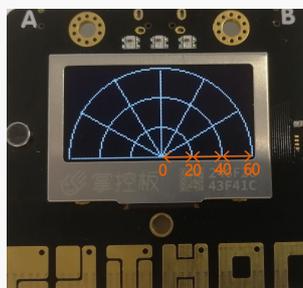
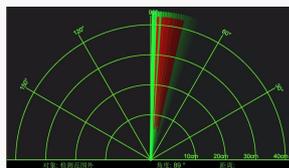


根据流程图可以看到，小车在行驶过程中，路线选择的优先级是 **区间 2→区间 1→区间 3**，分析如下：

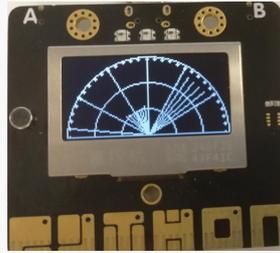
- 扫描过后，当前方无障碍，小车直行；
- 当前方有障碍，先判断右侧，无障碍，小车转右后直行；
- 当前方和右侧都有障碍，再判断左侧，无障碍，小车转左后直行；
- 当左、中、右都有障碍，说明小车前方道路不通，小车调头后直行；
- 小车启动后检测前方是否有障碍，无障碍则一直行走，有障碍则停止运动，再重新扫描。

* 优先转右是指当前方有障碍，优先判断右侧。大家也可以采用优先转左，只需要将判断的优先级改为 **区间 2→区间 3→区间 1**。

再来解决问题 1：如何实现雷达扫描图？ 超声波传感器的探测距离为 5cm-300cm，但是实际不用探测那么远，假设车身长 15cm，超声波探测的范围在 60cm 也就够用了。将 60cm 均匀分为 3 段，在屏幕上画出 3 个半圆，依次代表探测距离为 0-20cm、20-40cm、40-60cm。再画几根线区分 0-180 度的扫描区间，即可完成雷达网格。



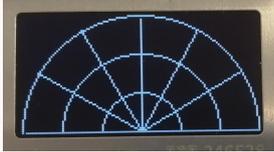
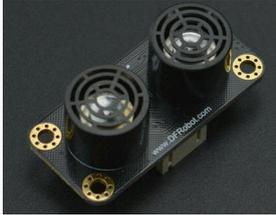
如何显示扫描的障碍呢？考虑到掌控板屏幕小、分辨率低、且只有黑白色，所以这里采用画线的方式显示障碍。如下图，无白线的区域表示没有障碍，有白线的表示有障碍，线的长短代表障碍距离的远近，线越短，说明距离越近！



这样如果扫描时，前方完全没有障碍，屏幕中将没有一点动态效果，为了显示扫描的过程，在雷达网图的最外侧弧线上，通过画点或画圆的方式，实时显示舵机的方位。

这样每次扫描时，可以根据外侧弧线的白点确定舵机方位，根据白线的长短判断障碍的距离。

解决了上面两个难题，一起来看看如何完整实现吧！首先，一步一步分解功能。

分解功能	所需元件
<p>1、画雷达网格</p> <p>根据前面的数学知识，编程实现。</p> 	<p>无</p>
<p>2、实现雷达扫描效果</p> <p>舵机带动超声波 0-180 转动，检测障碍。在屏幕上显示雷达扫描图。</p> 	<p>超声波(探测距离)</p>  <p>舵机 (在 0-180 度之间转动)</p> 
<p>3、“智能” 规划路线</p> <p>使用两个电机搭建一辆车。让小车能够根据雷达扫描的结果，“智能”选择路线。</p> <p>通过前面的分析，编程实现小车行走的逻辑。</p> <p>在这里思考一个问题：如何判断区间 1、2、3 是否有障碍呢？</p> <p>是超声波只要在这个区间中反馈一个比较小的值就让小车认为这个区间有障碍吗？</p> <p>还是可以通过一个区间范围内累计判断的方法呢？</p>	<p>电机</p> 

构造外观

无人漫游车外观草图	所需耗材、工具
例如：	例如：纸盒、剪刀等... 1、 2、 3、

问题记录

在后面项目实现过程中，可能会遇到各种各样的困难，尝试在下表中记录你遇到的问题和解决办法，便于以后出现类似问题时能更好的面对。

遇到的困难	你的解决方案
1、	1、
2、	2、
3、	3、
...	...

功能实现

功能 1：画雷达网格

程序及实现效果如下：

ESP32 主程序

雷达网图

定义 雷达网图

画圆 不填充 圆心 x: 64 y: 63 半径: 63

画圆 不填充 圆心 x: 64 y: 63 半径: 42

画圆 不填充 圆心 x: 64 y: 63 半径: 21

画3个半圆

设置 A 的值为 0

重复执行 7 次

画线 起点x1: 64 y1: 63 终点x2: $64 - 63 * \cos$ 变量 A y2: $63 - 63 * \sin$ 变量 A

将 A 增加 30

从圆心画出7根线，相邻两根线夹角为30度

起点为圆心

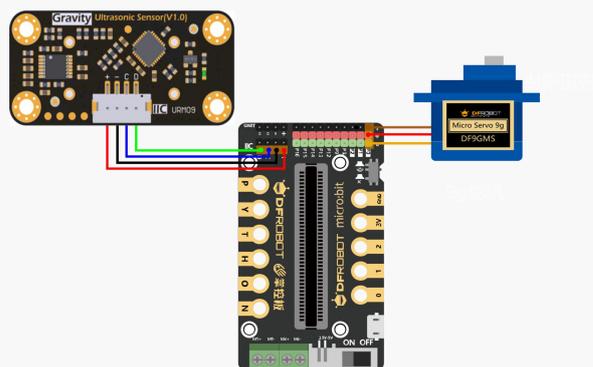
终点为最外侧半圆，半径R=63，角度A=0、30、60、90、120、150、180



功能 2：实现雷达扫描效果

舵机带动超声波 0-180 转动，检测障碍。在屏幕上显示雷达扫描图。

连线图：



* 注意在 Mind+ 中加载舵机和超声波。

在功能一程序基础上，程序修改如下：

```

定义 雷达扫描
    初始化I2C超声波传感器地址为 0x20
    设置 P0 引脚伺服舵机为 变量 B 度
    设置 超声波 的值为 读取I2C超声波传感器距离(cm)
    如果 变量 超声波 > 60 与 变量 超声波 < 5 那么执行
        设置 超声波 的值为 0
        将超声波检测在5-60cm定义为有效检测
    设置 超声波 的值为 约束 变量 超声波 介于(最小值) 0 和(最大值) 60 之间
    画图 填充 圆心 x: 64 - 61 * cos 变量 B y: 63 - 61 * sin 变量 B 半径: 1
    画线 起点x1: 64 y1: 63 终点x2: 64 - 变量 超声波 * cos 变量 B y2: 63 - 变量 超声波 * sin 变量 B
    
```

画圆表示舵机运动轨迹

画线表示检测到的障碍。超声波距离表示障碍的远近，舵机角度B表示方位。

```

ESP32 主程序
循环执行
    雷达网图
    设置 B 的值为 0
    重复执行直到 变量 B > 180
        雷达扫描
        将 B 增加 6
        等待 0.1 秒
    屏幕显示为 全黑

    雷达网图
    设置 B 的值为 180
    重复执行直到 变量 B < 0
        雷达扫描
        将 B 增加 -6
        等待 0.1 秒
    屏幕显示为 全黑
    
```

变量B表示舵机转动角度

舵机0-180度扫描

舵机转6度，超声波检测一次

舵机180-0度扫描

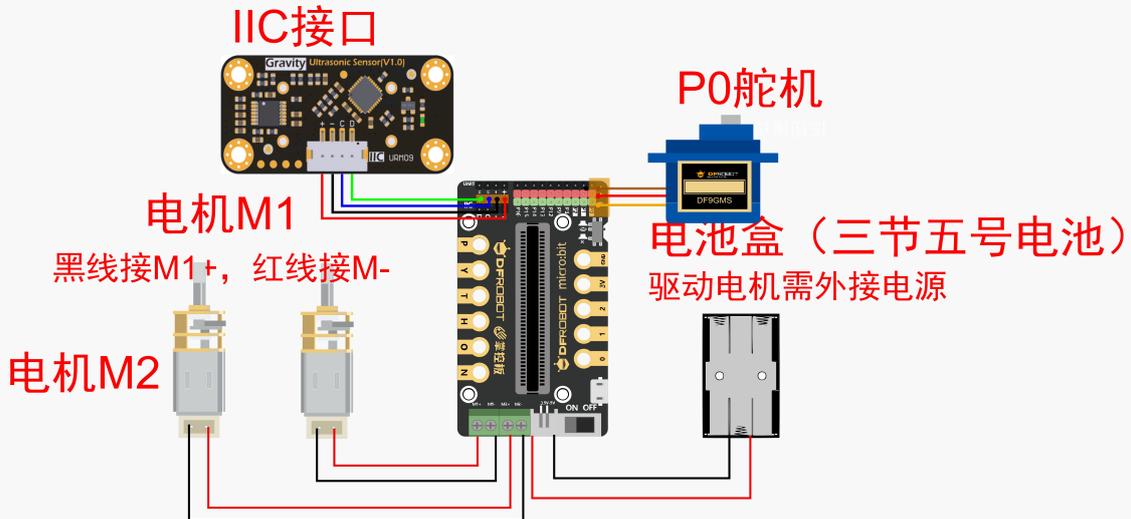
运行结果：将超声波固定在舵机上（可将胶带或胶枪固定），超声波随着舵机来回转动，在 5-60cm 范围内，如果有障碍，屏幕上将画出白线。



功能 3: “智能” 规划路线

使用两个电机搭建一辆车。

连线图:



在功能分解部分，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，当掌控板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。

定义 障碍检测

如果 **变量 B < 60** 那么执行

如果 **变量 超声波 > 5** 与 **变量 超声波 < 30** 那么执行
 将 **区间1** 增加 **1**

区间1: 舵机0-60度, 当障碍物距离小于30cm, 认为有障碍

如果 **变量 B >= 60** 与 **变量 B < 120** 那么执行

如果 **变量 超声波 > 5** 与 **变量 超声波 < 30** 那么执行
 将 **区间2** 增加 **1**

区间2: 舵机60-120度

如果 **变量 B >= 120** 与 **变量 B < 180** 那么执行

如果 **变量 超声波 > 5** 与 **变量 超声波 < 30** 那么执行
 将 **区间3** 增加 **1**

区间3: 舵机120-180度

定义 雷达扫描

初始化I2C超声波传感器地址为 **0x20**

设置 **P0** 引脚伺服舵机为 **变量 B** 度

设置 **超声波** 的值为 **读取I2C超声波传感器距离(cm)**

障碍检测 在扫描过程中进行障碍检测

如果 **变量 超声波 > 60** 与 **变量 超声波 < 5** 那么执行
 设置 **超声波** 的值为 **0**

设置 **超声波** 的值为 **约束 变量 超声波 介于(最小值) 0 和(最大值) 60 之间**

画圆 填充 圆心 x: $64 - 61 * \cos \text{变量 B}$ y: $63 - 61 * \sin \text{变量 B}$ 半径: **1**

画线 起点x1: **64** y1: **63** 终点x2: $64 - \text{变量 超声波} * \cos \text{变量 B}$ y2: $63 - \text{变量 超声波} * \sin \text{变量 B}$

定义 小车漫游

假设小车如下安装：
M2: 左轮，正传向前
M1: 右轮，反传向前

初始化I2C超声波传感器地址为 0x20

串口0 · 字符串输出 · 合并 区间1: 变量 区间1 不换行 ·
串口0 · 字符串输出 · 合并 区间2: 变量 区间2 不换行 ·
串口0 · 字符串输出 · 合并 区间3: 变量 区间3 换行 ·

设置 P0 · 引脚伺服舵机为 90 度 → 完成一次扫描后，舵机转到90°，超声波向前看

等待 1 秒

如果 变量 区间2 < 10 那么执行
电机 M1 · 以 60 速度 反转 ·
电机 M2 · 以 60 速度 正转 ·
区间2无障碍，小车直行

否则
如果 变量 区间1 < 10 那么执行
电机 全部 · 以 60 速度 正转 ·
等待 2 秒
区间1无障碍，小车右转

否则
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
区间3无障碍，小车左转

否则
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
区间3有障碍，小车后退，掉头

电机 M1 · 以 60 速度 反转 ·
电机 M2 · 以 60 速度 正转 ·
完成转向后，小车直行

设置 超声波 · 的值为 读取I2C超声波传感器距离(cm)

重复执行直到 变量 超声波 > 5 与 变量 超声波 < 30

设置 超声波 · 的值为 读取I2C超声波传感器距离(cm)

串口0 · 字符串输出 · 合并 超声波: 变量 超声波 换行 ·

电机 全部 · 停止

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

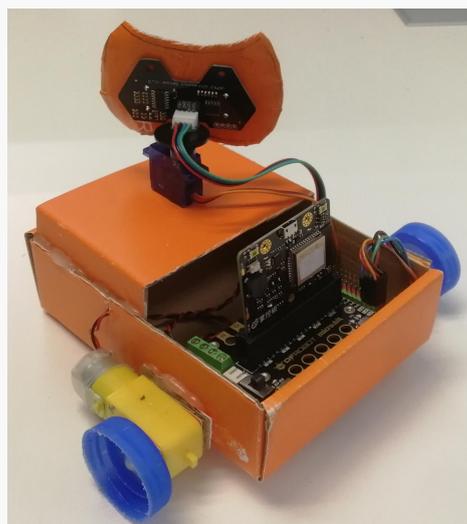
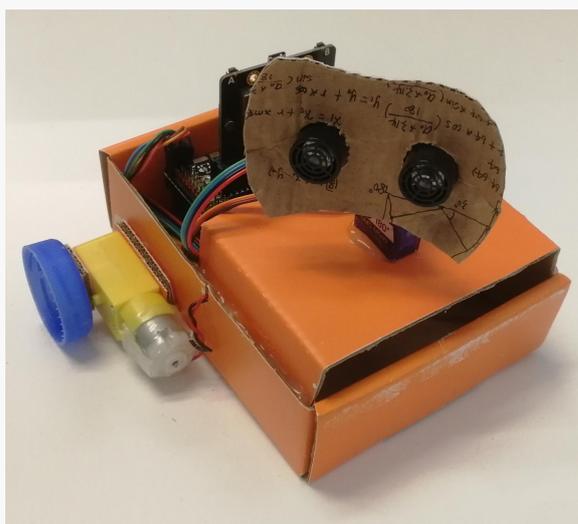
小车直行时，超声波检测前方是否有障碍，没有则一直直行；有则小车停止

```

ESP32 主程序
循环执行
  雷达网图
  设置 B 的值为 0
  重复执行直到 变量 B > 180
    雷达扫描
    将 B 增加 6
    等待 0.1 秒
  屏幕显示为 全黑
  雷达网图
  设置 B 的值为 180
  重复执行直到 变量 B < 0
    雷达扫描
    将 B 增加 -6
    等待 0.1 秒
  小车漫游 超声波来回扫描一遍后，小车漫游
  屏幕显示为 全黑
    
```

运行结果：上电后，舵机带动超声波 0-180 度扫描后，再 180-0 度扫描，同时在屏幕上显示雷达图。扫描一个来回后，小车“智能”行走，可以自行躲避障碍。

构造外观



评价与反思

与人分享快乐，能产生更多快乐，尝试与身边的朋友一起分享你的无人漫游车。你还可以在 DF 创客社区中分享作品，听听更多人的建议！社区网址：www.dfrobot.com.cn

自我反思

梳理自己设计项目的思维逻辑、项目在执行层面的问题和经验等等，可以帮助我们形成学习的闭环，加深在项目进行的过程中对知识点和内容的理解，找到可以改进和提高的地方。

优点与不足	1、
	2、
可以从项目中总结的经验	1、
	2、
学到的知识	1、
	2、