

micro:bit 物联网主题教程

序

物联网，它是继电脑，互联网之后的第三次世界信息产业浪潮，它被称为下一个万亿级的信息技术产业。在很多人的意识中，物联网听起来就是一个比较抽象的概念。其实也不一定，现在很多学校已经开设了关于物联网的课程了。在未来，我们的孩子只会更加习惯这个美好的物联网时代，

micro:bit 物联网入门套件是一款基于 DFRobot 的物联网平台——“Easy IoT”的物联网套件，它搭载着时下流行青少年编程教育的 ARM 开发板——micro:bit，在只有半张信用卡大小的体积上集成了蓝牙，加速度计，电子罗盘，按钮，LED 点阵等功能。套件中含有 OBLOQ 物联网模块和多款 Gravity 传感器，接线方便简单。

编程软件我们使用 Mind+，是一款基于 Scratch3.0 开发的优秀青少年编程软件，支持 Arduino、Micro:bit 等各种开源硬件，只需要拖动图形化程序块即可完成编程，还可以使用 Python/C/C++ 等高级编程语言，让大家轻松体验编程的乐趣。

本教程一共通过智能生活、智慧交通、智慧农业三大板块来学习 micro:bit 物联网入门套件，学生们可以在大量的实践案例中学会编程以及物联网基础知识，在学习的过程中展现他们更多的创造力。

目录

micro:bit 物联网主题教程	1
序	1
第一章 认识物联网	6
一、什么是物联网	6
二、物联网的基本框架	7
三、物联网是如何工作的	9
三、物联网的实际应用	9
第二章 物联网环境搭建	11
学前提示	11
装备认知	11
第三章 智慧生活	26
第一课 心情表达器	26
学前提示	26
硬件清单及连接	27
知识点解析	28
指令提前知	30
程序编写	30
参考程序	34
拓展阅读	36
第二课 云养猫之猫粮机	38
学前提示	38
硬件清单及连接	39
知识点解析	39
指令提前知	40
程序编写	40
参考程序	43
第三课 智能晾衣杆	45
学前提示	45
硬件清单及连接	46
知识点解析	47
指令提前知	48
程序编写	48
参考程序	53
拓展阅读	55
第四课 快递质量抽测	56
学前提示	56
硬件清单及连接	57

知识点解析	57
指令提前知	60
程序编写	61
参考程序	65
第五课 多功能鞋子	67
学前提示	67
硬件清单及连接	68
知识点解析	68
指令提前知	70
程序编写	70
参考程序	75
第六课 智能婴儿摇篮	78
学前提示	78
硬件清单及连接	79
知识点解析	80
指令提前知	81
程序编写	82
参考程序	86
第七课 智能门锁	88
学前提示	88
硬件清单及连接	89
知识点解析	90
指令提前知	91
程序编写	91
参考程序 1	95
参考程序 2	99
拓展阅读	102
第八课 智能温控系统	103
学前提示	103
硬件清单及连接	104
知识点解析	105
程序编写	106
参考程序	111
拓展阅读	115
第九课 竞选投票器	117
学前提示	117
硬件清单及连接	118
知识点解析	118

指令提前知	119
程序编写	119
参考程序	129
第十课 综合项目	133
学前提示	133
基础任务一 智能家居系统	134
基础任务二 远程电梯预约	136
基础任务三 车内儿童滞留报警器	140
第四章 智慧交通	142
第一课 车流量统计	142
学前提示	143
硬件清单及连接	143
知识点解析	144
指令提前知	146
程序编写	146
参考程序	150
拓展训练	151
第二课 停车场剩余车位统计	152
学前提示	152
硬件清单及连接	153
指令提前知	154
程序编写	154
参考程序	160
拓展训练	161
拓展阅读	161
第三课 超速报警器	163
学前提示	163
硬件清单及连接	164
知识点解析	166
指令提前知	166
程序编写	166
参考程序	171
拓展阅读	174
第四课 综合项目	176
学前提示	176
基础任务一 闯红灯电子警察系统	176
基础任务二 不停车收费系统	179
第五章 智慧农业	182

第一课 智能灌溉系统 182

 学前提示 183

 硬件清单及连接 184

 知识点解析 185

 指令提前知 186

 程序编写 186

 参考程序 191

 拓展训练 195

 拓展阅读 199

第二课 智能水稻种植 200

 学前提示 200

 硬件清单及连接 201

 知识点解析 202

 程序编写 203

 参考程序 206

 拓展阅读 209

第一章 认识物联网

在过去的几个世纪中，人类经历了一系列的技术革命，而每次革命都是由某种主流技术引导，例如：18 世纪的工业革命开启机械时代；19 世纪的第二次工业革命迎来了电气时代；20 世纪开启了信息化时代，网络和电脑成了这个时代的标志；而进入 21 世纪后，随着传感设备，嵌入式系统和互联网的普及后，物联网被称为是继电脑，互联网之后的第三次世界信息产业浪潮，物联网作为当下智能家居开发以及城市建设的中坚力量，它将应用于各个领域，并引领人们进入更加智能化的时代。

一、什么是物联网

某天早上，你从梦中醒来，发现已经 7 点 40 了。明明调好 7 点的闹钟确没有响！上班要迟到了，你匆忙的从床上起来，跑进卫生间刷牙洗脸，这时乱糟糟的头发也没有时间去管它了，穿好衣服后，你跑进厨房，打开冰箱，发现里面什么也没有，因为昨天你忘记采购新的食物了，没办法只能饿着肚子去上班了。你抓起钥匙，拿上桌上的资料来到了地下车库。当你将车开出小区之后，发现今天因为下雨，通往公司路上堵车了。面对眼前长长的车队，你却没有丝毫的办法.....在生活中，你是否遇见这些窘境呢？

接下来，我们设想另外一种情况，早上七点，阳光透过窗帘，房间的窗帘自动打开，阳光洒了进来，天气预报自动播报天气情况并将数据传递到衣柜，衣柜根据今天的天气为你挑选衣服和鞋子。你去洗漱的同时，厨房里的机械手臂夹起烤好的面包，一杯咖啡，放在餐桌上。吃过早餐后，你便开车出门了，这时你拿出手机，输入起始点，手机已经告诉你走那条路线最节省时间，到达公司的时候，手机也已经推送了最佳停车位置。

到了中午，你的手机向你发出了警报，告诉你家可能有生物入侵，你立即点开实时监控，发现只是一直小猫在你家门口徘徊，然后你发现家里的插座和门窗没有关好，于是你轻点手机，家里的插座和门窗自动关闭。当门关闭的一瞬间，小猫咪被吓了一跳，感觉好像是对你家“会思考的家居”表示震惊。

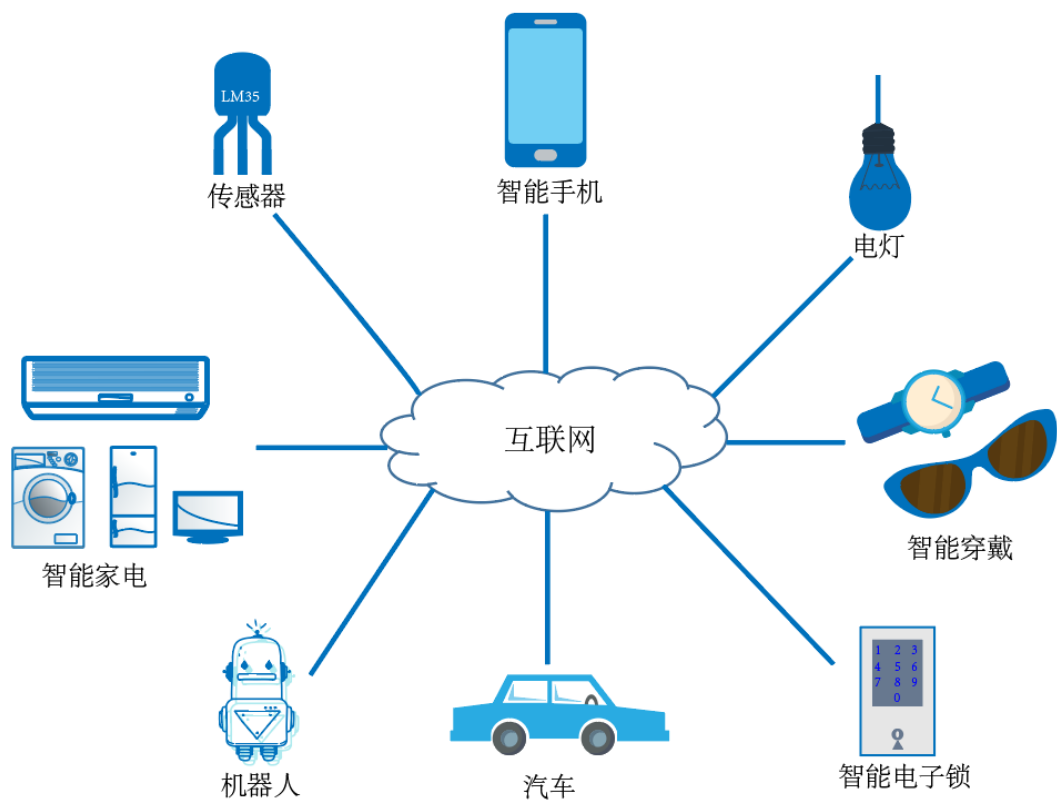
到了下午，在老家的爷爷打来电话，说家里种植的蔬菜，不知怎么的下午全焉了，这时你打开手机发现土壤湿度过低，棚内的温度过高了，你轻点手机，大棚里的喷水头开始喷水，自动降温送风系统启动。

一天工作结束后，你想回到家就能舒舒服服的在家里躺着了。于是你在下班的路上，通过一键启动居家环境，这时空调自动打开，电饭煲开始工作，扫地机器人开始打扫。当你回到家的时候，你想到的一切智能家居都帮你完成了。

看到这里，你会想，这些不是在科幻电影里面看见的场景吗？对！在未来，通过物联网这些都能变成现实。那么什么是物联网呢？

物联网（IoT，Internet of things）里面所说的“物”，是包含你身边所有的物品，日常生活常见的有：洗衣机、冰箱灯家电；汽车、房子等固定资产；就连桌子、椅子都包含其中。非日常生活中的物品有：温度计、测速器等。以此得知，所谓的物联网就是“万物皆可相连”的世界。在物联网里，你生活中所有的物品都可以用电子标签上网联结，然后彼此进行信息交换和通信，以实现对物品的智能化识别、定位、跟踪、

监控和管理。



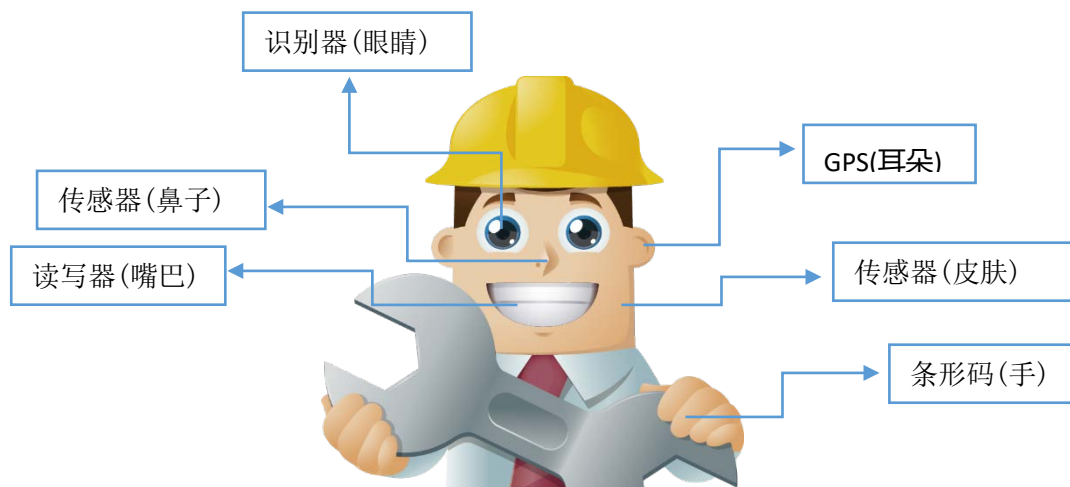
物联网，就是所有的物品都与互联网相连

二、物联网的基本框架

物联网，它让每件物品都具备和人一样的“思维能力”和“执行能力”。而实现这些功能主要靠它的三个基本架构：感知层，网络层，应用层。如下图所示。

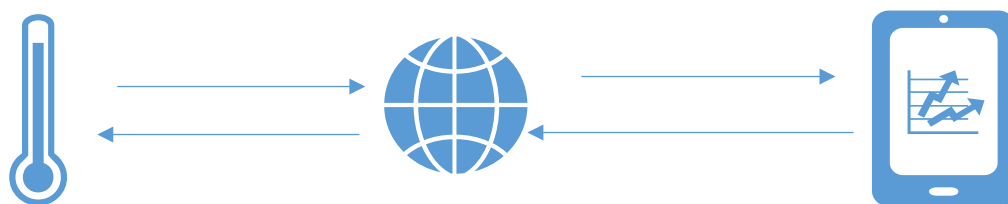


感知层是物联网的底层，由各种传感器以及传感器网关构成，包括二氧化碳浓度传感器、温度传感器、湿度传感器、二维码标签、条形码、RFID 标签和读写器、摄像头、GPS 等感知终端。感知层相当于人的眼耳鼻和皮肤，它的主要功能是识别物体，信息采集。如下图所示。



感知层和五官

网络层是物联网的中间层，它由各种私有网络、互联网、有线网、无线网、网络管理系统以及云计算平台组成。相当于人的神经中枢和大脑，主要负责传递和处理感知层获取到的信息。网络层信息的传递，主要是通过因特网和各种网络的结合，对接收到各种感知信息进行传送，实现两个端系统之间的数据透明传送。具体的功能包括寻址、路由选择，以及连接的建立、保持和终止。如下图所示。



网络传输数据

应用层是物联网和用户（人、组织、其他系统）的接口，它与行业需求结合，实现物联网的智能应用。目前物联网的行业特性主要体现在其应用领域内，目前绿色农业、工业监控、公共安全、城市管理、远程医疗、智能家居、智能交通、自然检测等各行各业都有物联网的应用。



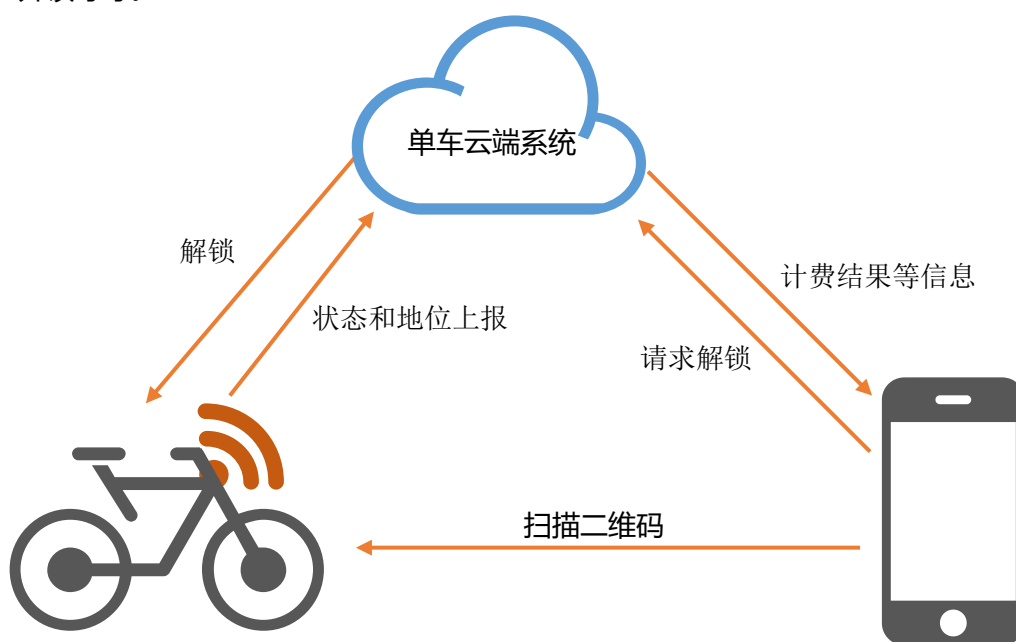
物联网应用领域

三、物联网是如何工作的

物联网日益改变着我们的生活，从驾车出行，购物，乃至房供暖都深受影响。通过前面对物联网的基本框架的了解，你清楚物联网是如何工作的嘛？

我们将周边的客观物体都植入精密的传感器和芯片，它们将不断的收集并发送有价值的信息。通过这些数据，让我们能更深入的了解这些物体是如何协同工作。但这些设备应该如何分享如此大规模的数据，同时我们应该怎样利用这些数据呢？不管是提高工厂的生产效率，为城市居民实时更新最佳停车点，还是检测个人健康状况，这些都需要一个物联网平台来整合信息，并且为设备和应用软件之间的通信提供一种通用的语言，整个流程始于设备自身，他们与物联网平台进行安全通信，平台将所有的数据进行整合分析，提取最有价值的部分，再发送给特定工业需求的应用程序。

我们举一个简单的例子：共享单车，它使用起来很方便，它的特点就是扫描二维码既可以解锁骑车，所以共享单车的物联网原理主要是采用了“手机端—云端—单车端”的架构。我们通过手机端的 App 可以查看附近的单车，充值、预约开锁等等。云端就是服务器端，是整个共享单车系统的控制台，它可以与所有的单车进行数据通讯，收集信息指令，响应用户和管理员的操作。单车端是收集信息与执行命令的一端，比如卫星定位、开锁等等。



三、物联网的实际应用

物联网的应用领域涉及到方方面面，除了上面我们举例的共享单车，在工业、农业、环境、交通、物流、安保等基础设施领域的应用，下面我们就一起来看看在现实生活中，物联网都运用在哪些地方了。

1、物联网下的便利生活

想必现在每个小区都会有速易递，丰巢等智能包裹箱，它们的自助收投服务系统被认为快递物流行业的

一个新想法。

智能包裹箱不需要人工把守，送件员把包裹放置柜重，收件人便会收到消息通知（短信或者微信），消息中有取货码，取件人只需要输入取货码或者微信扫取件二维码，就能打开箱口，非常的便利。

2、物联网下的服装行业

近两年 RFID 系统集成商开始和商家一起共同探索 RFID 系统的整体应用，不仅用于仓库管理，还应用于前端门店的销售过程，以便在销售数据上报之前及时了解热销款式，灵活调整生产计划，避免大量库存产生，这一点对于时尚快销服装类尤其重要。

它们将服装上安装射频识别电子标签，每一件衣服对应唯一标码，一件衣服从服装生产厂家到上架，工作人员不再需要一一验查、清点。因为，衣服会将自己的身份信息“告诉”接货员。到了销售门店，有多少顾客从货架上将衣服取下，又有多少人试穿过，衣服都可以将这些消息实时反馈。

目前，海澜之家、迪卡侬、H&M，ZARA 等品牌都已开始部署 RFID 系统。会“说话”的衣服，带来的是更加智能化的生产，节省大量的人力成本。

3、物联网下的医疗健康

想必现在医疗健康也是人们比较关系的一大类，以前去医院就诊，需要现场挂号，等报告。而现在，我只需要在网上预约挂号，做完检查后，只需要在手机上查看自己的报告。除此之外，我们还能在线就诊，卫生系统的资源也得到了充分的利用。

在生活中，不止以上 3 种的实际应用，还有很多覆盖在其他领域，比如教育、金融、餐饮、农业、影视等。随着物联网的发展，我们在科幻电影中看见的那些神奇场景都有可能现实生活中实现。

第二章 物联网环境搭建

在科学技术日渐发达的今天，物联网技术已经逐渐运用到生活中。那我们可不可以自己搭建一个属于自己的物联网环境呢？又需要用到哪些“装备”呢？接下来我们就一起来学习如何搭建一个完整的物联网环境吧！

学前提示

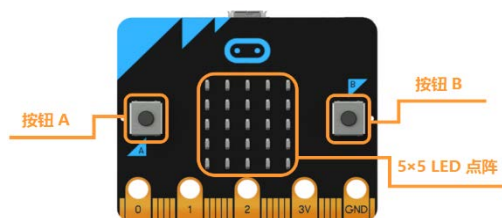
建一座房子，需要先打地基，只有地基打好了，房子才能牢固。同理，我们想要搭建出物联网环境，首先要了解装备，以及它们的基础使用方法。所以本小节我们将认识 micro:bit 和 Micro:mate，了解 mind+ 的基础操作。

装备认知

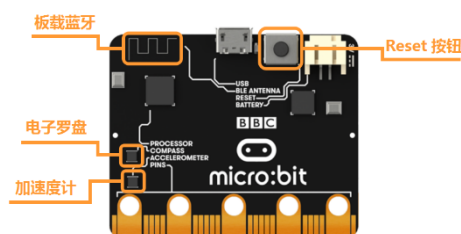
一、认识 micro:bit

micro:bit 是一款由英国广播电视公司（BBC）为青少年编程教育设计，并由微软、三星、ABM、英国兰卡斯特大学等合作伙伴共同完成开发的微型电脑。micro:bit 用于各种酷炫的创作，从机器人到乐器，可能性是无止境的。

1、micro:bit 功能介绍



micro: bit 正面



micro: bit 背面

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

2、micro:bit 能用来干什么？

micro:bit 集成了很多基本的传感器，LED 点阵可以显示基本的英文字符串、数字或者图形）通过图形化或者代码编程，可以完成很多有意思的项目。micro:bit 还可以通过鳄鱼夹或者扩展板与电子元件互动，支持读取传感器，控制舵机和 RGB 灯带。能够轻松胜任各种与编程环境相关的教学与开发环境，例如：编写电子游戏、声光互动、机器人控制、科学实验、可穿戴装置开发。

3、micro:bit 的编程方式

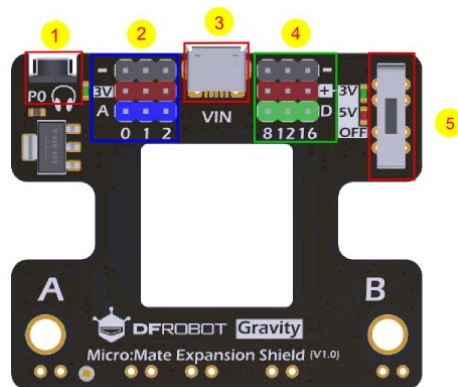
micro: bit 有一个特点就是可以不用安装任何软件，直接在网页上进行编程，可以通过任意一个浏览器在 Blocks, javascript, python, scratch 等编辑器编写代码，网页上可以登陆 micro:bit 的官网 <http://micro:bit.org> 进行编程。

除了登录官方网站进行编程外，还用很多支持 micro:bit 的图形化编程软件，而 Mind+就是其中的一款，Mind+ 不仅适用于中小学生的学习，还为想要提高自身技能的“创客”提供 Arduino、python/c/c++ 等高级编程语言的学习环境。它还支持很多传感器，只需要拖动图形化程序块就可以完成编程。

二、认识 Micro:mate

1、Micro:mate 的介绍

Micro:Mate 是一款为 micro:bit 设计的微型多功能 IO 传感器扩展板,其彩色 3Pin 接口支持 DFRobot 百款 Gravity 系列电子模块，可直插传感器或电子元件。

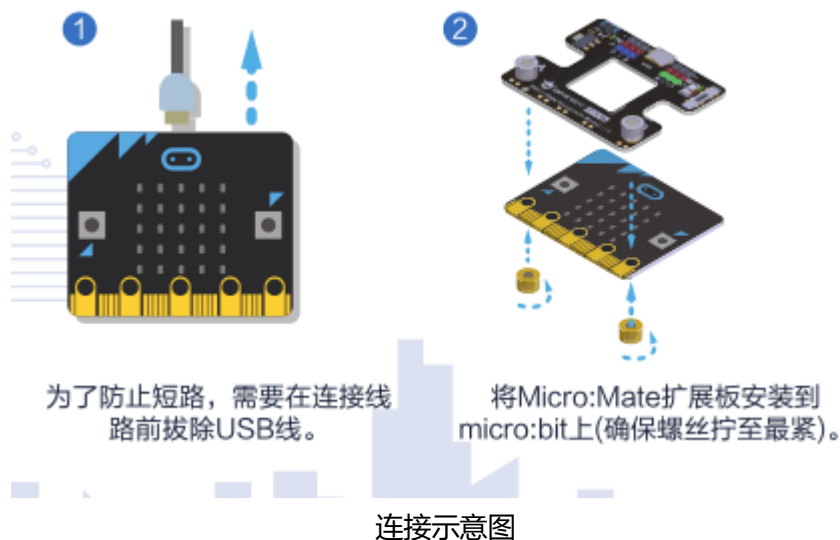


标号	名称	功能描述
1	耳机接口	用于直插 3.5mm 耳机
2	模拟输出口	兼容 Gravity 系列数字与模拟模块 (仅兼容 3V)
3	USB 外接电源口	用于外接 5V 电源 (不可用于上传程序)
4	数字口	兼容 Gravity 系列数字模块 (兼容 3V 与 5V)
5	数字口电源切换开关	用于切换 3V,5V 与 OFF

Micro:Mate 扩展板体积小巧，完成后仅增加 micro:bit 板 5mm 的厚度，最大限度地集成了 micro:bit 的常用功能，板载 3.5mm 耳机插口，支持 3 路模拟输入和 3 路数字输入输出，支持 3V 与 5V 两种驱动电压，使 micro:bit 兼容 5V 数字输入输出元件。扩展板 USB 口可为大功率设备提供外接供电，保证舵机、电机等设备能够正常运行。

2、连接 Micro:Mate 与 micro:bit

Micro:Mate 扩展板通过弹簧弹针连接 micro:bit 控制器，在保证接触良好的情况下，避免了损伤主板金手指。预留的开口刚好可以露出两个按键以及中间的 LED 矩阵。



三、认识 mind+

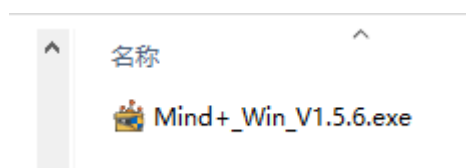
Mind+是一款基于全新的版本是基于 Scratch3.0 进行开发的, 同时也支持使用 python/c/c++等高级编程语言进行控制。它, 可以一键安装常用的硬件驱动; 它, 支持与 Scratch 舞台角色的交互、实时和脱机运行。在 Mind+ 里, 编程的过程就像拼图一样有趣。孩子们通过拖动图形化的功能模块就可以学习编程了。

1. mind+下载与安装

- Mind+客户端下载, 下载地址: <http://mindplus.cc/>



- 下载完成后双击文件安装。



注意: 本教程是在 Mind+_Win_V1.5.6 的版本下完成的。

2. 安装驱动

- 软件成功之后点击“教程”——“视频教程”按钮打开入门教程，学习驱动安装。



点击安装驱动，观看教程

- 根据视频教程，一键安装驱动（注意：此步骤需要以管理员权限运行 Mind+）



3.Mind+界面介绍

Mind+共有两种操控模式：

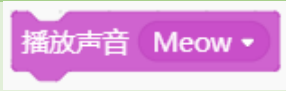
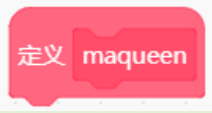
- 实时模式下，需要将设备一直连接电脑，可以和 mind+舞台互动。



- 上传模式下，上传程序给设备之后，设备可以脱离电脑运行。



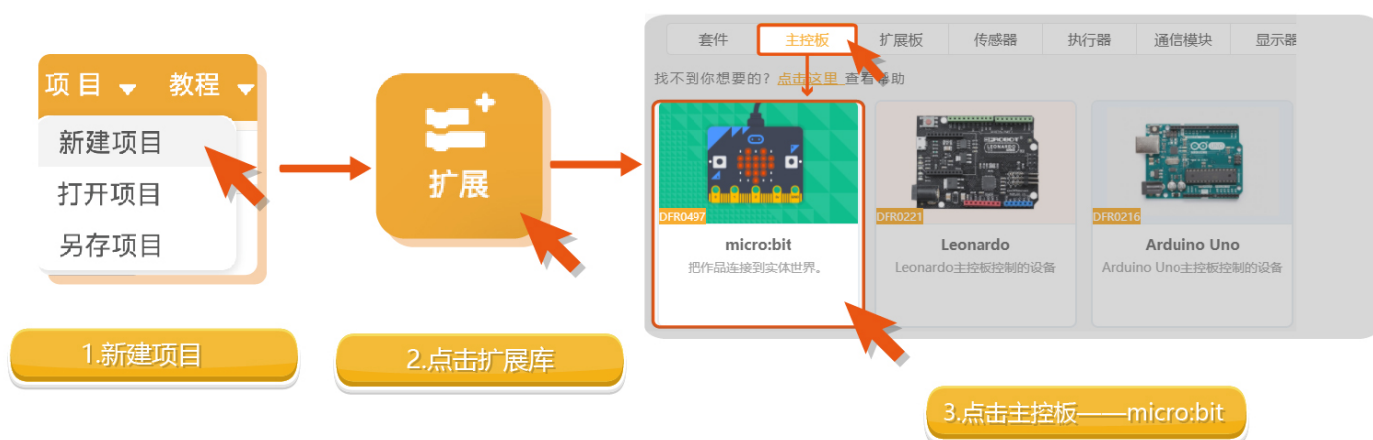
● 指令区模块类别

类别（根据颜色区分）	功能	示例	所属模式
 运动	主要针对角色设定的，用来控制角色活动。		实时模式
 外观	控制角色以及舞台的外观，主要包括说话效果，角色造型，背景切换等功能。		实时模式
 声音	该模块可以为角色或场景添加音乐效果。		实时模式
 事件	用来检测一些来自于其他模块的事件操作，一般用于启动程序的基础。		实时模式
 控制	主要是用来控制某个功能或者动作的运行效果；例如，重复执行某一个步骤，或者根据条件来决定是否执行接下来的动作。		实时模式 上传模式
 侦测	此模块是用来检测角色，鼠标，外部环境的变化。		实时模式
 运算符	运算符模块里面包括数字运算和逻辑运算。 数字运算：加减乘除、取余、平方根、比较大小等。 逻辑运算：与，或，非。		实时模式 上传模式
 变量	表示各种变化的量，可自由创建数字变量、字符串变量、列表。		实时模式 上传模式
 函数	可自己创建函数。名字自己取。		实时模式 上传模式
 micro:bit	micro:bit 函数模块（在扩展区添加）用来操作 micro:bit。		实时模式 上传模式

注意：micro:bit 函数模块，需要另外添加。

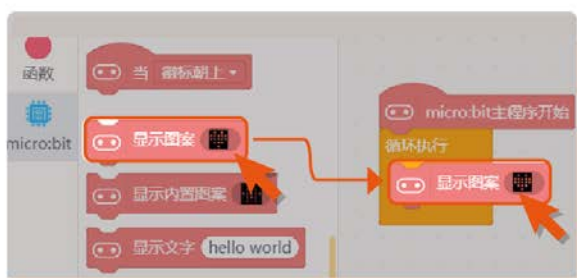
4.项目实例

- 添加主控板，选择 micro:bit。(根据自己使用的主板选择型号)

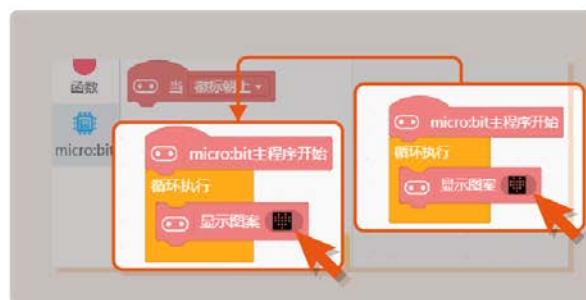


- 程序编写

1: 将需要的指令从左边指令区移动到右边编辑区，以此对主板下达“指令”。



2: 如需删除不需要或者错误的指令，将指令从右边编辑区，拖回到指令区，或右键选择直接删除即可。



- 上传程序

每次完成指令编辑后即可将新的指令按照以下步骤传送给 micro:bit。

1. 用USB线将麦昆的大脑（micro:bit）和电脑连接在一起。



2. 让Mind+连接麦昆



*打开Mind+中的“连接设备，打开连接“通道”。

3. 上传指令



*当进度条达到100%时，指令就传送完毕了。

四、认识物联网模块——OBLOQ

1.介绍 OBLOQ

- OBLOQ 物联网模块，主要功能是可以使得设备可以连接互联网，物联网设备设计提供基础技术。具备的基础功能是，连接 wifi，发送数据到物联网和接收物联网数据。



- 简单认识一下物联网模块的接口、指示灯及按键的作用。

接口：OBLOQ 物联网模块需要通过两根线与 microbit 主控板进行连接，通过两根线与 microbit 设备进行信息的交互，即接收和发送端。

状态指示灯：在连接 wifi 的过程中，因为网络的因素很难排查，所以需要按照指示灯的颜色进行问题查找。

颜色	指示状态
红色	没有正常运行
蓝色	正在连接wifi
绿色	正常工作
紫色	Mqtt连接断开
白色	Obloq固件升级中

按键 RESET：复位键，当模块使用中出现了不可预期的问题时，可以使用复位键，对模块进行重新刷新配置。

1 TX
2 RX
3 GND
4 VCC

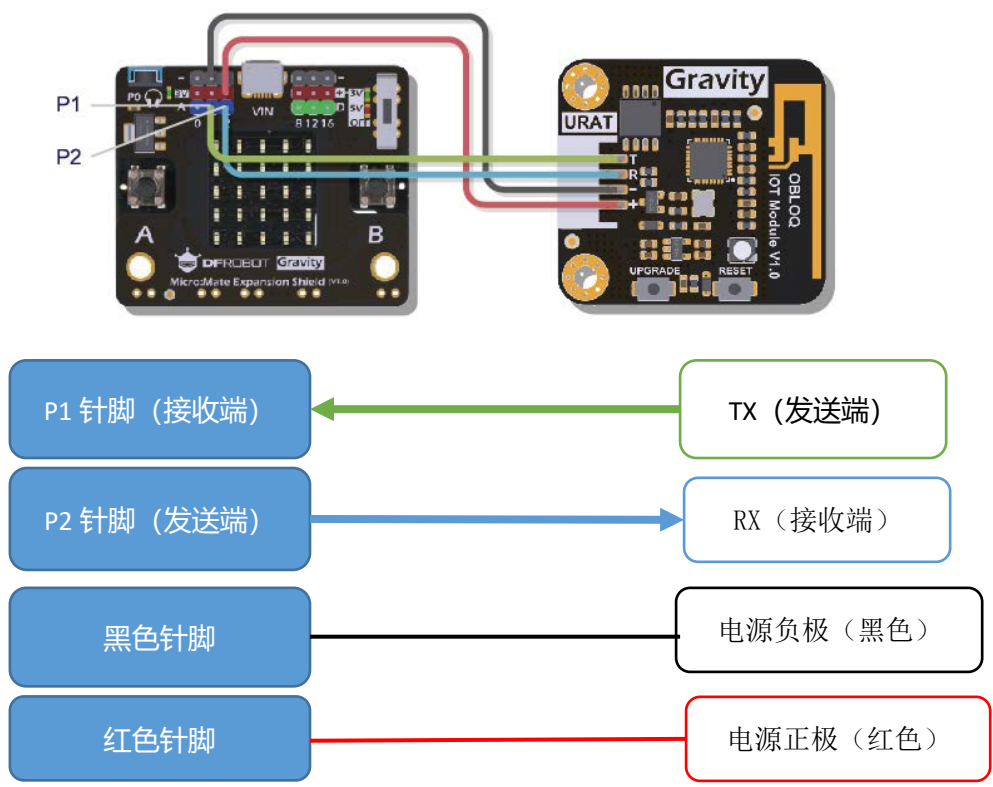
状态指示灯

固件更新按钮 复位键

标号	名称	功能描述
1	TX	串口发送端
2	RX	串口接收端
3	GND	地
4	VCC	电源

2.连接设备

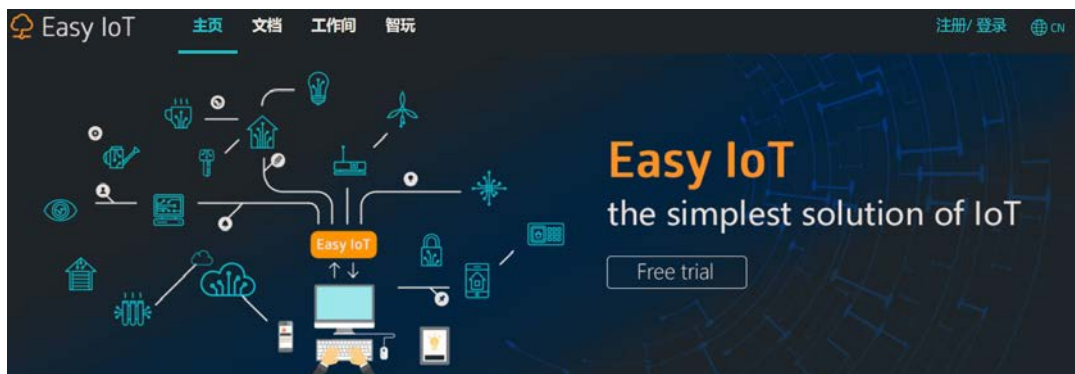
在连接设备的时候，需要注意一个概念，接收与发送，microbit 的接收端（P1）要与物联网模块的发射端（TX）连接，因为物联网模块通过 wifi 接收到了数据信号，然后发送（TX）出来，到达 microbit 的接收端（P1）。就好像是，聊天的时候，我们通过嘴巴（TX 发射端）说出来的声音，被对方用耳朵（RX 接收端）听到，经过大脑处理之后，再通过嘴巴回复。



五、Easy IoT 登陆与设置

接下来，我们需要对 microbit 设备进行编程，配置上网的密码等等，程序设置好就可以上网了。但是，我们这些信息传到互联网上之后，这些信息是怎么被查看的呢？这个信息到底发送给谁呢？

打开网址：iot.dfrobot.com.cn，可以进入终端网页 Easy IoT。



Micro:bit 主控板、物联网模块配合 Easy IoT 终端网页，是一整套简单的物联网系统。

1.注册登录

- 进入终端网页 Easy IoT 之前，需要注册一个个人后台监控账号。点击右上角的“注册/登录”。



- 填写个人信息，可以通过个人手机或者邮箱注册。

手机注册

邮箱注册

请输入手机号码

验证码

获取验证码

输入6-16位密码

再次输入6-16位密码

☐ 我已看过并接受[用户协议](#)

下一步

已有账号？立即[登录](#)

- 注册成功后，登录账号进入工作间，就是可以检测 microbit 设备的数据界面了。需要通过添加新设备，才能够和自己 microbit 设备进行配对。



2. Easy IoT 设置

- 新设备建立之后，会自动生成随机的设备的 Topic: eIJA7JHZg，每个人的都不一样；点击设备名称，可以进行重新命名。



点击发送消息，可以看见接收信息与发送信息的终端页面，详细的看到终端收到的数据，并且可生成图表。
([点击查看详情](#)，[可以查看生成的图表](#))。

Easy IoT 主页 文档 工作间 智玩 Joanna.li 注销 CN

←回到我的工作间

microbit设备A 对应设备名称

发送新消息 发送信息

请输入要发送的内容,长度限制128位。 **发送**

查看消息 查看信息的图表

Date range -

请输入消息包含的字符串 **查看**

最新消息 收到的消息列表

时间	消息	操作
上一页	下一页	

- 认识页面的其他信息

lot_id(user)

..... **IoT_id: 让micro:bit设备找到easy-IoT网页终端的确定账户地址代码**

lot_pwd(password)

..... **IoT_pwd: 允许micro:bit设备接入easy-IoT网页终端的密码钥匙**

重新生成 **重新生成: 重新生成之后会产生新的IoT_id与IoT_pwd, 会导致按照原IoT_id与终端进行信息交互的设备找不到终端。**

已分配: 1000/10000 **点击小眼睛, 查看和隐藏id与密码**

六、对 micro:bit 设备进行程序设计

当我们根据前面的步骤，将物联网的硬件环境都搭建好了，接下来我们就需要进行程序设计了，让

Easy IoT 终端与 micro:bit 之间通信。

1. 添加通信模块

- 打开软件 mind+, 选择主控板 microbit, 物联网属于高级功能, 需要在扩展中找到通信模块中的物联网模块。



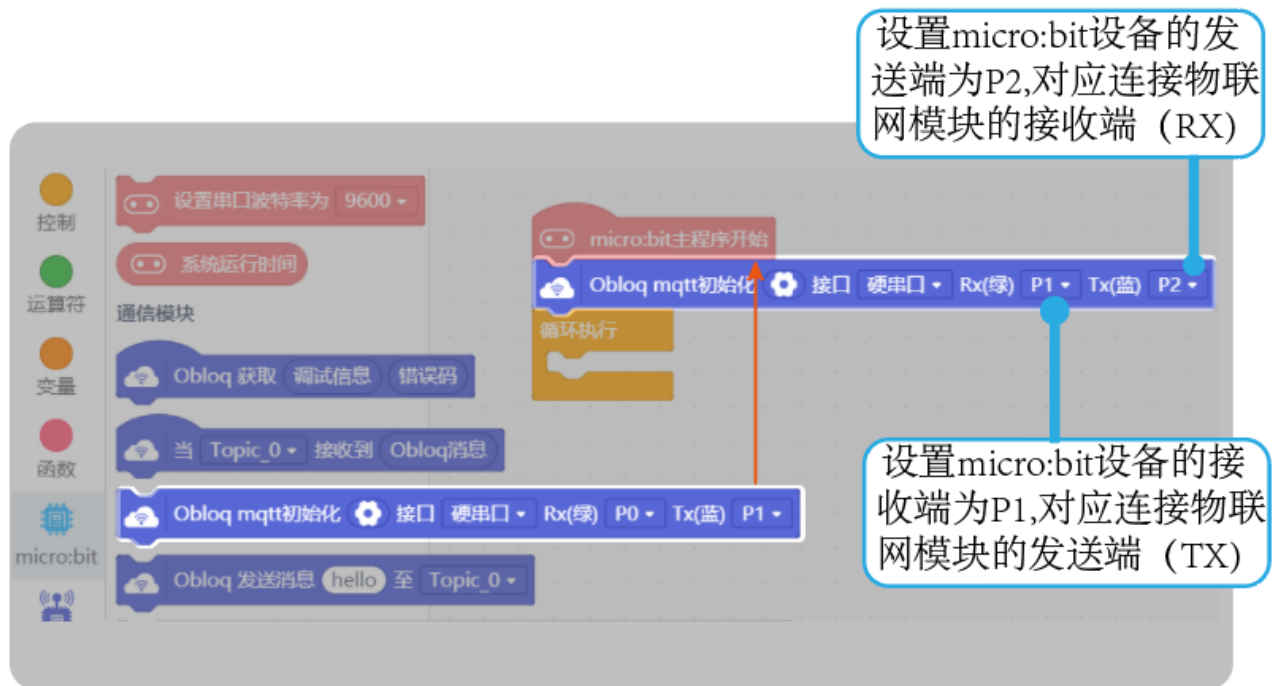
添加之后, 在指令区, 会多了一个版块——通信模块。



注意: 在添加通信模块前, 需要将主控板——micro:bit 添加好。

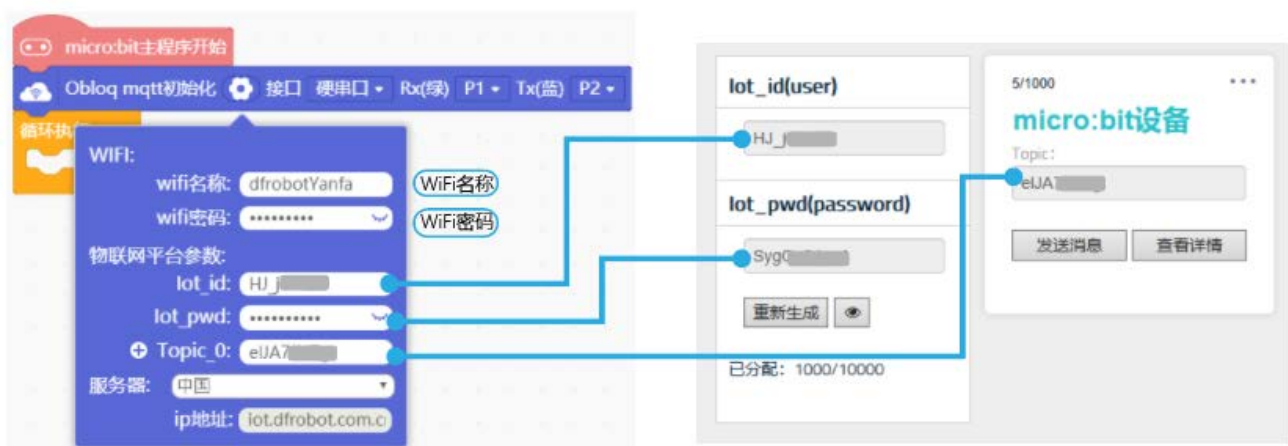
2. 物联网模块程序初始化设置

- 将初始化模块 “Obloq mqtt初始化 接口 硬串口 Rx(绿) P0 Tx(蓝) P1” 拖放至右边的编辑区。然后定义发送和接收的引脚, 要和实际连线一致。



注：绿色的线接 P1 口（micro:bit 设备的接收端），蓝色的线接 P2 口（micro:bit 设备的发送端）

- 对 microbit 设备设置 wifi 账号和密码和对应的网页终端的相关账号和密码（支持复制）。注意这里 Topic 可以设置多个，也就是说一个设备可以同时发给多个 Topic。



3.设置数据发送——温度数据

- 想要将获取到的数据发送到网页终端，需要调用发消息指令 “Obloq 发消息 hello 至 Topic_0”。注意设置 Topic。将下面完整的程序下载到 micro:bit 设备中，按下 A 键，就能将当前的温度发送至网页终端了。



实时温度信息。

最新消息	
时间	消息
2019/7/23 16:38:28	32
2019/7/23 16:38:25	32
2019/7/23 16:37:38	31
2019/7/23 16:37:38	31
2019/7/23 16:37:31	31

第三章 智慧生活

随着物联网的发展，对我们的生活也带来了巨大的改变，现在越来越多的人希望利用自己的手机或者其他的移动电子设备通过物联网技术与生活中一切物品连接，并进行全方位的智能控制，以此开启一个智慧生活时代。



第一课 心情表达器

聊天的时候，我们经常会发一些表情来表达自己当时的心情，如果我们希望知道远方的小伙伴今天是什么心情，我们应该怎么办呢？我们可以不可以制作一个这样的“设备——心情表达器”，让自己身边的显示屏被小伙伴控制，接收从远方发过来的各种信息呢？

学前提示

使用心情表达器能够表示你现在的状态，比如我现在心情不好，通过给物联网服务发送信息，物联网服务分析获取到的信息，然后把分析结果反馈给设备。也就是说，物联网服务平台会通过传感器或者其他设备感测你当时状态，然后以表情的形式将你现在的状态反馈出来。



心情表达器的项目结构分析图

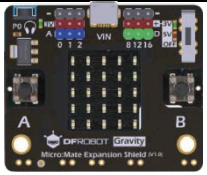
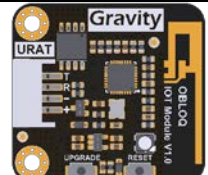
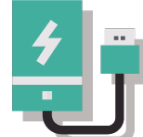
心情表达器的物联网原理和现在市面上有很多穿戴设备类似。只是穿戴设备可以将其看作能进行“感测”和“反馈”的设备。



穿戴设备的分析图

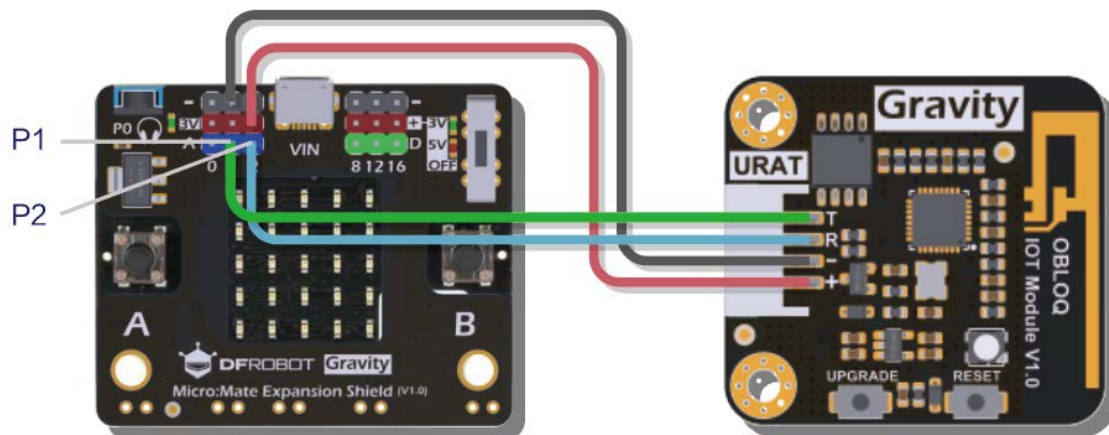
硬件清单及连接

硬件清单

			
Micro:bit+Micro:mate 扩展板	OBLOQ 模块	电脑 (需要接入互联网)	USB 充电宝 (备选供电)

硬件连接

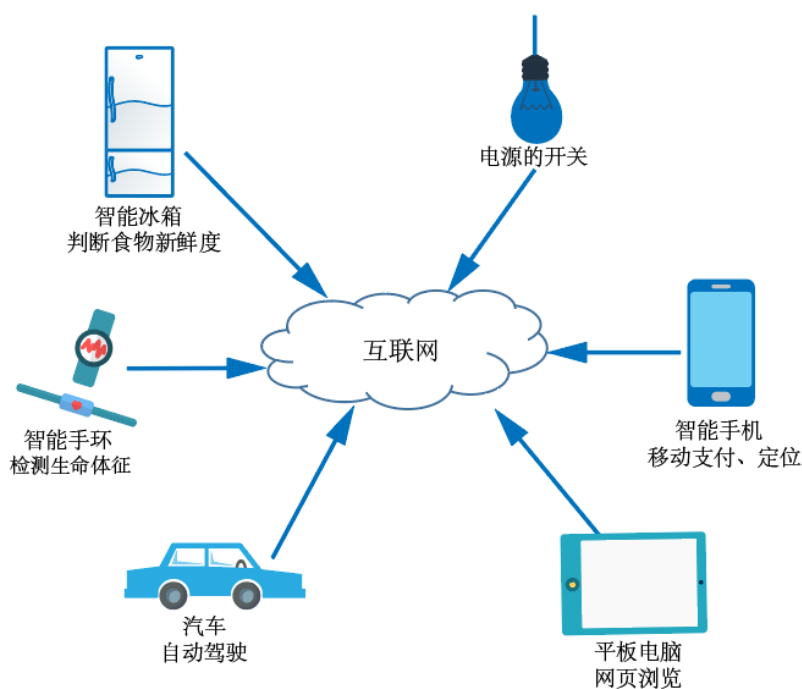
OBLOQ 模块中的 T 引脚（绿线）连接扩展板上的 P1 针脚（蓝色针脚），OBLOQ 模块中的 R 引脚（蓝线）连接扩展板上的 P2 针脚（蓝色针脚），OBLOQ 模块中的“-”引脚（黑线）连接扩展板上的黑色针脚，OBLOQ 模块中的“+”引脚（红线）连接扩展板上的红色针脚。



知识点解析

物联网的设备

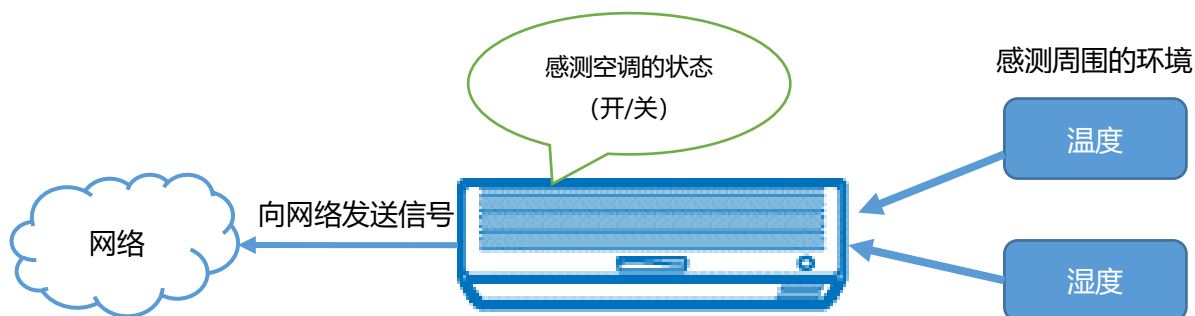
我们都知道物联网中的“物”是指我们身边的物品，而设备就是能联网的“物品”，除了能联网外一般都会安装传感器这类电子元件。常见设备有：智能手机、平板电脑、家用电器、手环等，只要满足条件，都可以是物联网的设备。这些设备都起着两个作用：感测和反馈。



生活中常见的设备

感测的作用

感测（感知+测量），就是设备利用传感器将周边环境的状态转换成电信号并通知给系统，比如空调的开关，房间内的温湿度之类。

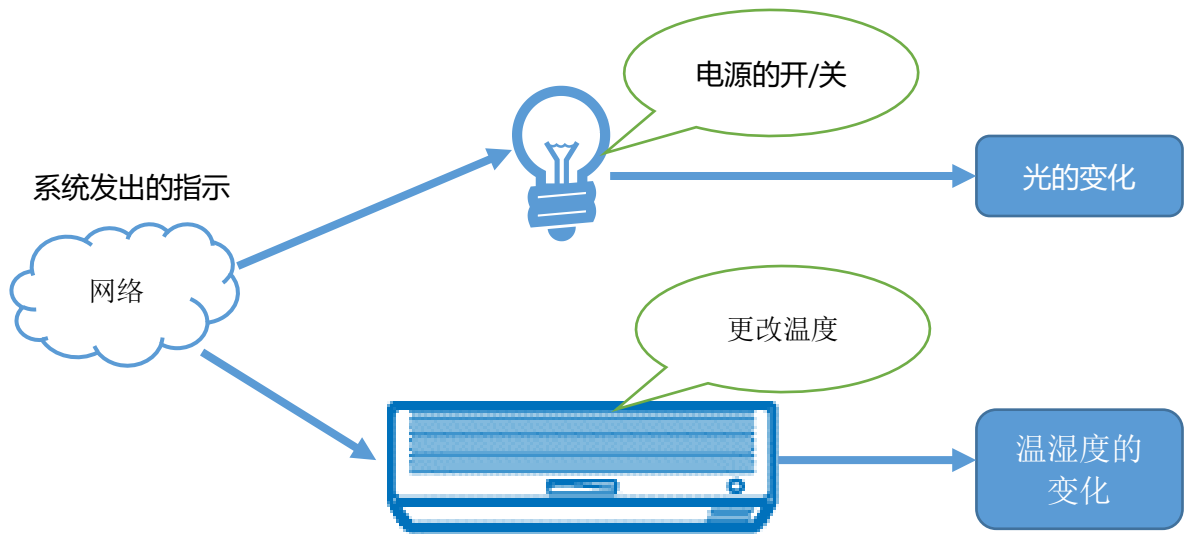


感测的作用

反馈的作用

反馈，就是设备接收从系统发出来的通知，显示信息或者执行指令操作。系统会基于从传感器处搜集

到的信息进行一些反馈。



反馈的作用

反馈一共分为 3 种类型：可视化，通知，控制。接下来我们举个例子来理解这 3 种类型。

“可视化”能使用电脑和智能手机上的浏览器浏览物联网终端搜集到的信息。例如温湿度检测器，把房间的当前温度和湿度“可视化”，人就能将环境控制在最适宜的条件下。

“通知”系统能检测到“物”的状态和某些活动，并通知给设备，近几年来，微信，微博，facebook，抖音等社交软件通过智能手机贴心的向我们推送朋友们吃饭和旅行的消息。

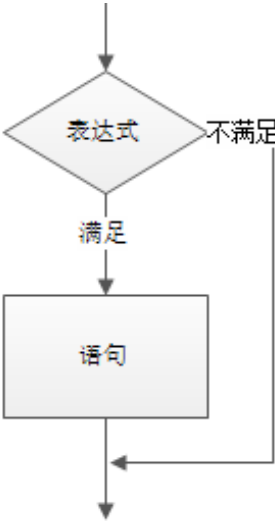
“控制”系统是可以直接控制设备运转的，不需要人工操作。例如，你下班回家，你的手机会根据你的位置和距离，然后利用传感器将你的步速通知给物联网终端。这样终端就能分析你到家的时间，然后就可以发送指令给家里的家电。这样你回到家时，饭可能已经煮好了，温度也达到最舒适的温度了。

条件判断结构——如果

如果我们需要判断某些条件来决定是否执行这个结果，条件判断就是一个很方便的结构了，在 mind+ 中条件判断结构有如下图两种。在程序中，如果有条件判断结构，程序就会根据不同的条件选择不同的程序语句，来实现不同的功能。

模块	类型	语义
	单分支条件	判断表达式 () 中的值，如果条件满足，就执行模块里包含的语句 1。执行结束接着执行模块后面的其他语句，如果不满足，直接跳过语句 1，执行模块后面的其他语句。
	双分支条件	判断表达式 () 中的值，如果条件满足，就执行模块里包含的语句 1，否则就执行语句 2。

本节课我们会学习“单分支条件判断”模块。其过程图如下，其中判断条件可以是关系运算/逻辑运算。



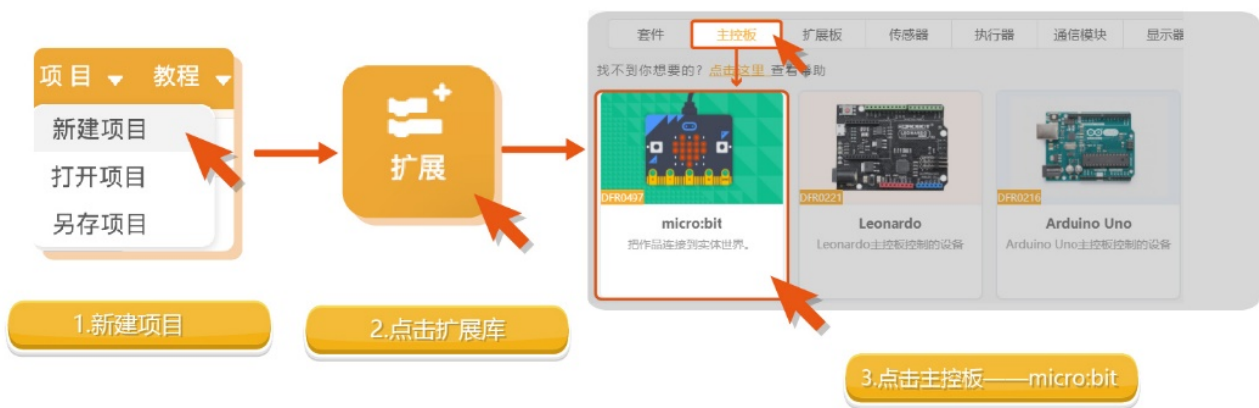
指令提前知

模块	功能说明
	物联网初始化模块，用来定义发送和接收的引脚以及网页终端相关账号和密码的设置。
	发送消息指令，终端发送消息给设备
	判断设置好的 Topic 是否收到信息。
	等于运算符，判断左边的值等于右边
	显示 micro:bit 主板库里面内置的图案
	单分支条件判断，如果条件成立就执行其中包含的脚本

程序编写

1.添加主控板和物联网通信模块

根据下图的步骤添加主板——micro:bit 和物联网模块——OBLOQ（[详细步骤请阅读第一章相关内容](#)）



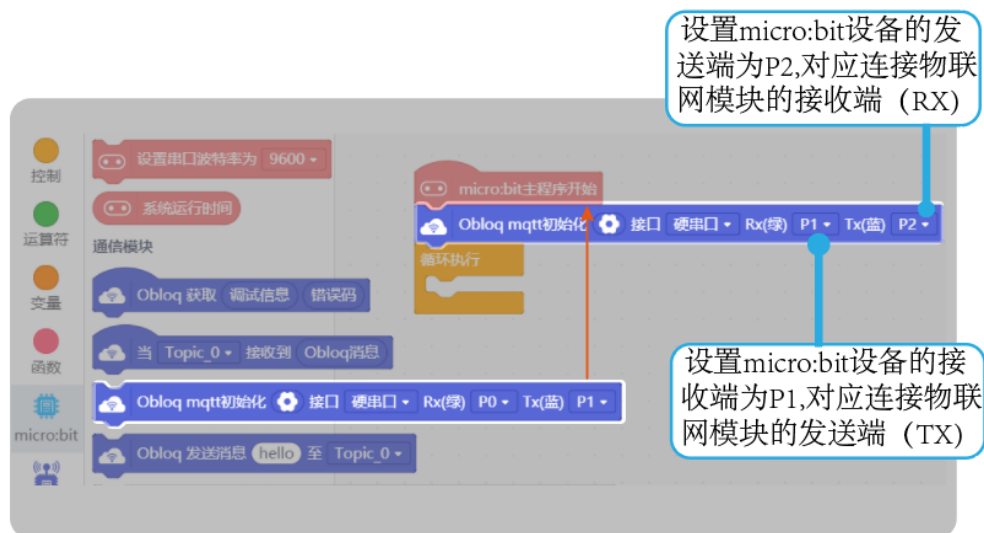
添加主板



添加物联网模块

2.物联网模块初始化及联网测试

- 将初始化模块 “Obloq mqtt初始化 接口 硬串口 Rx(绿) P0 Tx(蓝) P1” 拖放至右边的编辑区。然后定义发送和接收的引脚（要和实际连线一致）以及设置网页终端的相关账号与密码。



注：绿色的线接 P1 口（micro:bit 设备的接收端），蓝色的线接 P2 口（micro:bit 设备的发送端）。



设置 WiFi 密码和物联网平台参数

- 设置好物联网的相关信息后，将发送消息指令 “Obloq 发送消息 hello 至 Topic_0” 拖动到如图所示位置，帮助确认是否联网成功。



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。



3.表情设置

- 将接收消息指令 “当 Topic_0 接收到 Obloq消息” 拖放到如图所示位置，判断 Topic 是不是已经收到了消息。



- 通过网页终端发送指令“Y”则显示笑脸，表示心情很好。发送“N”则显示哭脸，表示心情不是很好。实现这样的功能就需要使用条件判断指令，判断接收到的指令。

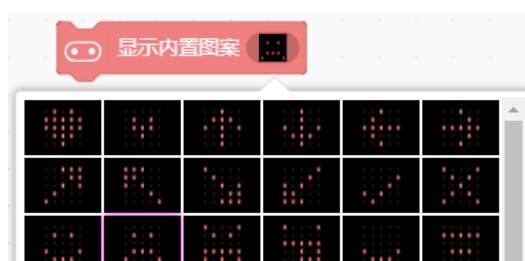


提示：在物联网通信模块的指令区没有该模块“Obloq消息”，我们需要从模块

“当 Topic_0 接收到 Obloq消息”中，将它拖出来，然后放在需要用到它的位置，如下图。



micro:bit 库中有很多表情选择，通过点击表情框下拉选择其他表情。



注意，这里不需要循环执行指令，因为只有判断 Easy IoT 收到数据，该程序才会被执行，否则，不会执行，所以不需要循环执行。

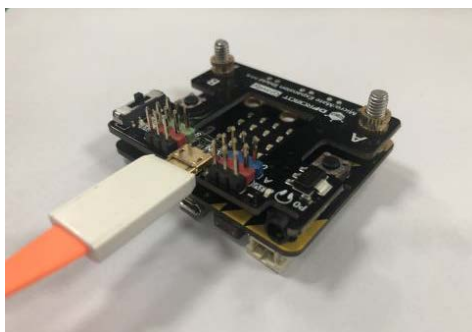
参考程序

程序展示

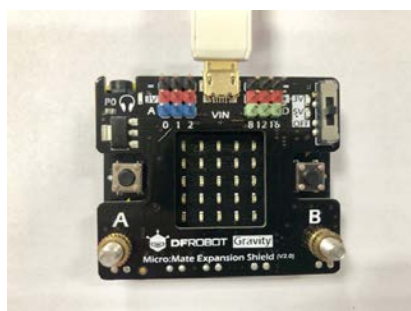
将下面的程序下载到 micro:bit 设备中。



注意：程序上传完毕后，将连在主控上面的 USB 数据线拔下，连接到扩展板上；也可以在不改变主控板连接 USB 数据线的基础上，多用一根 USB 线将扩展板上的 VIN 接口连接外接电源上。



当板子额外供电后，需将板载上调节按钮的电源调至 5v。



效果展示

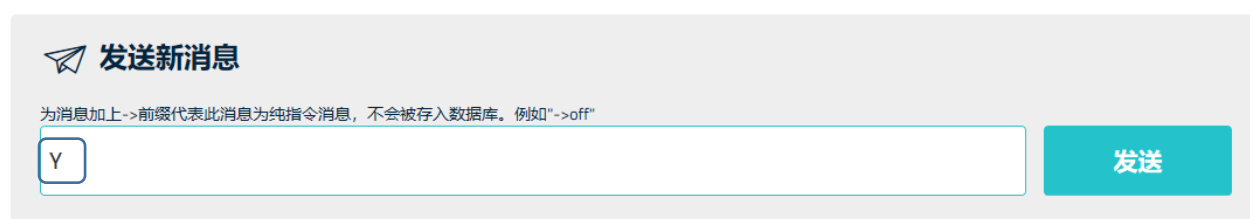
- 进入 Easy IoT 网页发送信息给 micro:bit 设备。找到相应的 Topic，点击“发送消息”。



- 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。

[←回到我的工作间](#)

micro:bit设备



- 可以在下面“最新消息”板块看见刚刚发出去的消息记录。

最新消息

时间 消息

2019/7/25 17:9:47 Y

- 当消息“Y”发送出去后，就能在 micro:bit 上看见一个笑脸。



拓展阅读

可穿戴设备

可穿戴设备即直接穿在身上，或是整合到用户的衣服或配件的一种便携式设备。可穿戴设备不仅仅是一种硬件设备，更是通过软件支持以及数据交互、云端交互来实现强大的功能，可穿戴设备将会对我们的生活、感知带来很大的转变。



在生活中比较常见穿戴设备类型有：手表型，HMD 型，饰品型。

手表型设备是一种佩戴在手腕上使用的可穿戴设备。该设备跟手表形状相同，其显示屏有呈四边形的也有呈圆形的。比如智能手环，Apple Watch。

HMD 型是一种像眼镜或者护目镜可以架在眼前的设备。其中眼镜型主要是为穿戴着它进行操作或是来

回走动时设计的，而护目镜型会完全遮挡住穿戴者的视野，所以多用于娱乐和游戏场合。比如谷歌眼镜，VR眼镜。

饰品型多数是缠在手臂上使用的腕带型设备，这种设备的传感器体积小、重量轻，能够轻松测量穿戴者的运动状况和睡眠状况。比如指环型的手势识别器。

可穿戴设备的种类不一样，功能也不一样。比如我们前面讲的三种类型，它们分别用了摄像头，语音识别，手势控制 3 种功能。那除这三种还有什么功能呢？

作用	功能	能做到的事
向设备输入信息	摄像头	拍摄图像和影像，识别图像
	语音识别	使用语音进行输入和控制
	手势控制	使用手势操作各种设备
接收设备发来的反馈	显示信息	将文本、图像、影像等显示在显示屏上
	通知	利用语音或者显示屏通知穿戴者
其他	上网	能够连接到网络，传输数据，在云端实施对数据的加工等
	储存	离线存储数据

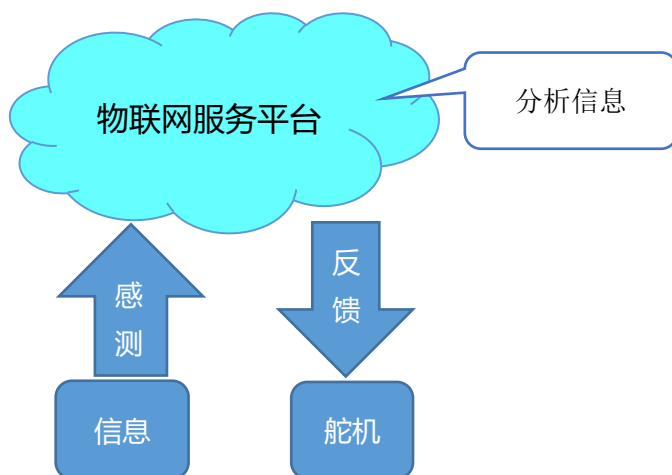
第二课 云养猫之猫粮机

现如今有很多人喜欢养猫，猫是一个让人感觉柔软的动物，也是现在比较常见的宠物。在养猫的过程中也会遇到很多困难，比如：中午不在家的时候，或是准备外出旅游的时候，如果有个可以远程控制的猫粮机诞生，将为我们的生活提供很多便利，接下来，我们一起尝试制作一套可以远程控制的猫粮机！



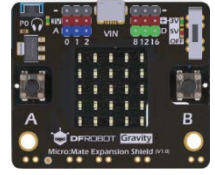
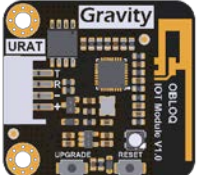
学前提示

制作远程控制的猫粮机，通过物联网服务平台远程发送控制信息，当接收到控制信息时，可以控制硬件装置旋转不同的角度，该硬件装置可以选择为舵机，从而控制猫粮正常下落或停止下落。



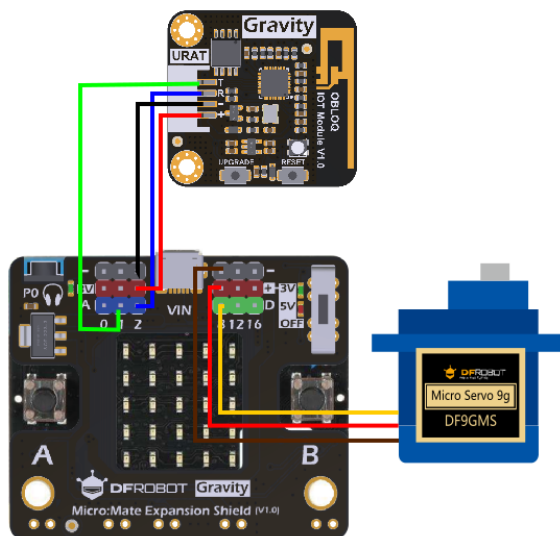
硬件清单及连接

硬件清单

				
Micro:bit+Micro:mate 扩展板	OBLOQ 模块	电脑 (需要接入互联网)	USB 充电宝 (备选供电)	金属 9g 舵机 (舵机模块)

硬件连接

OBLOQ 模块中的 T 引脚（绿线）连接扩展板上的 P1 针脚（蓝色针脚），OBLOQ 模块中的 R 引脚（蓝线）连接扩展板上的 P2 针脚（蓝色针脚），OBLOQ 模块中的“-”引脚（黑线）连接扩展板上的黑色针脚，OBLOQ 模块中的“+”引脚（红线）连接扩展板上的红色针脚。舵机模块连接扩展板上的 P8 针脚，其中黄色的线连接绿色针脚，红色的线连接红色针脚，棕色的线连接黑色针脚。



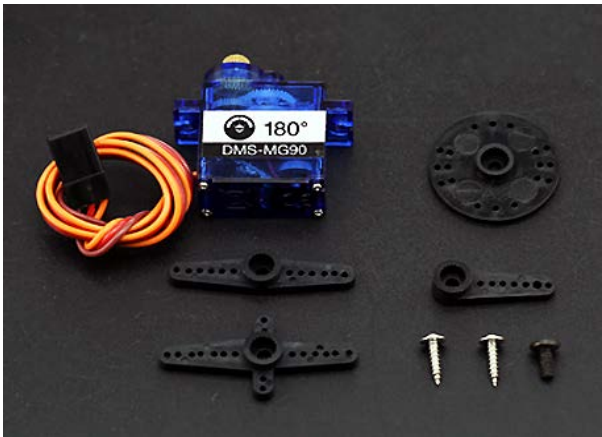
知识点解析

舵机

舵机是一种位置（角度）伺服的驱动器，适用于那些需要角度不断变化并可以保持的控制系统。舵机有塑料齿、金属齿、小尺寸、标准尺寸、大尺寸，另外还有薄的标准尺寸的舵机。小舵机一般称为微型舵机，扭力都比较小，市面上 2.5g、3.7g、4.4g、7g、9g 等舵机指的是舵机的重量分别是多少克，体积和扭力也是逐渐增大。微型舵机内部多数都是塑料齿，9g 舵机有金属齿的型号，扭力也比塑料齿的要大些。

本节课所使用的舵机是金属 9g 舵机，该舵机采用高强度 ABS 透明外壳配以内部高精度金属齿轮组，加上精准的控制电路、高档轻量化空心杯电机使该微型舵机的输出力矩达到了惊人的 1.8kg*cm。在使用过

程中也许注意：尽量不要将角度旋转至 0 度或 180 度，当舵机承载较重的物体时，易发生其他特殊情况。使用时，可将舵盘固定在舵机上，由舵机的动力来带动舵盘转动。



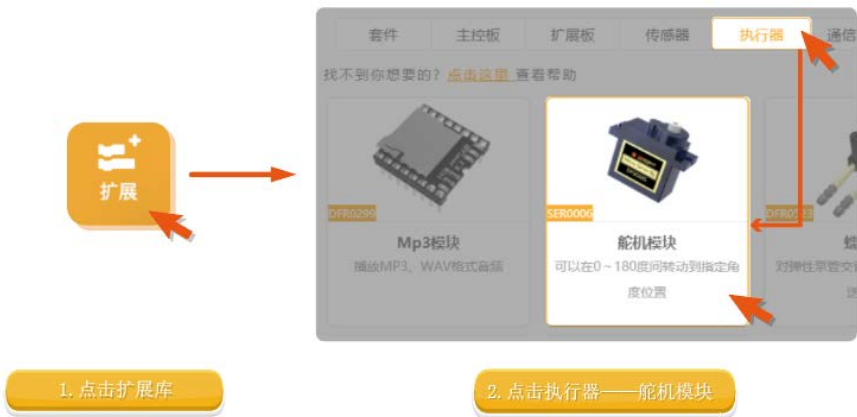
指令提前知

模块	功能说明
 设置 P9 ▾ 引脚伺服舵机为 90 度	控制舵机模块旋转一定角度，可以选择范围为 0-180 度，由于 0 度或者 180 度为极限角度，一般常用设置角度 10-170 度。

程序编写

1.添加模块

按步骤添加程序编写过程中需要运动的模块，添加主板——micro:bit、物联网模块——OBLOQ (详细步骤请阅读第一章相关内容)、执行器——舵机模块。



添加舵机模块

2.物联网模块初始化及联网测试

- 将初始化模块 “ Obloq mqtt初始化 接口 硬串口 ▾ Rx(绿) P0 ▾ Tx(蓝) P1 ▾” 拖放至右边的编辑区。然后定义发送和接收的引脚（要和实际连线一致）以及设置网页终端的相关账号与密码。

注：绿色的线接 P1 口
(micro:bit 设备的接收端)，蓝色的线接 P2 口
(micro:bit 设备的发送端)。



设置 WiFi 密码和物联网平台参数

- 设置好物联网的相关信息后，将显示内置图案指令 “显示内置图案” 拖动到如图所示位置。当 OBLOQ 模块板载上的灯由红色变成——蓝色变成——绿色，同时板载显示设置的自定义图案时，说明物联网模块（OBLOQ 模块）联网成功。



3.舵机角度控制

- 将接收消息指令 “当 Topic_0 接收到 Obloq消息” 拖放到如图所示位置，判断 Topic 是不是已经收到了消息。
- 通过网页终端发送指令，发送消息的方法已在上一节课程中讲解。当网页终端发送“开”时，显示对号；发送“关”时，显示错号，说明网页终端发送信息时，可以正常显示相应的运行效果。



- 本节课围绕制作远程猫粮机为主题，接下来需设置舵机的初始化状态，假设舵机的起始角度为 10 度。



- 当猫粮机关闭时（同舵机的起始角度），舵机的起始角度为 10 度；打开猫粮机时，旋转 90 度，此时，舵机角度为 100 度。关闭猫粮机时，则又将舵机的设置为初始角度 10 度。



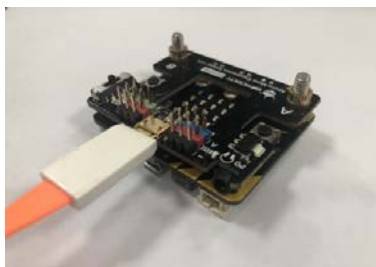
参考程序

程序展示

将下面的程序下载到 micro:bit 设备中。



提示：由于舵机耗电量大，需要外接电源。由于舵机耗电量较大，可断开舵机与板子的连接，再烧录代码，烧录代码后，可以将连在主控上面的 USB 数据线拔下，连接到扩展板上；也可以在不改变主控板连接 USB 数据线的基础上，用另外一根 USB 线将扩展板上的 VIN 接口连接外接电源。



当板子额外供电后，需将板载上调节按钮的电源调至 5v。



效果展示

- 进入 Easy IoT 网页发送信息给 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，当打开猫粮机是，发送指令“开”。

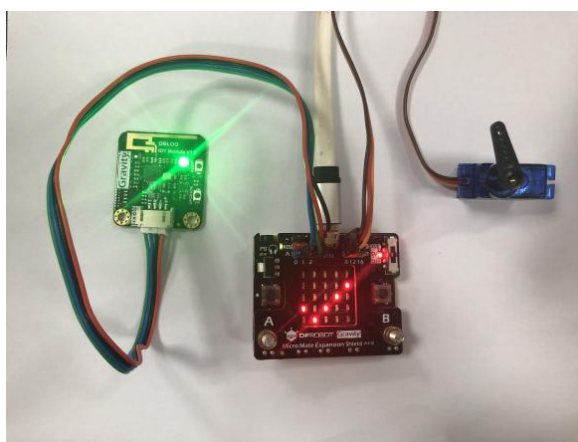
 **发送新消息**

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”

×

发送

- 当接收到“开”的指令时，可以在下面“最新消息”板块看见新消息“开”，同时板子显示“√”，舵机旋转 90 度，当前度数为 100 度。



- 当打开猫粮机是，发送指令“关”。

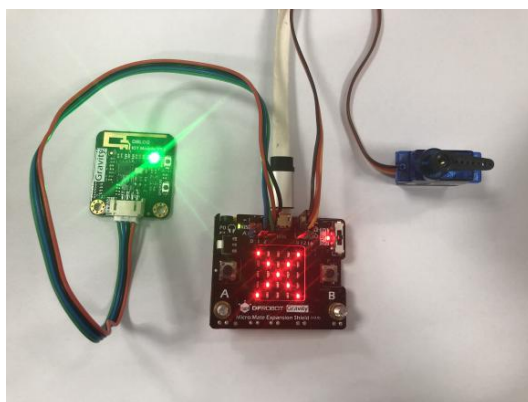
 **发送新消息**

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”

×

发送

- 当接收到“关”的指令时，可以在下面“最新消息”板块看见新消息“开”，同时板子显示“×”，舵机回到起始值。



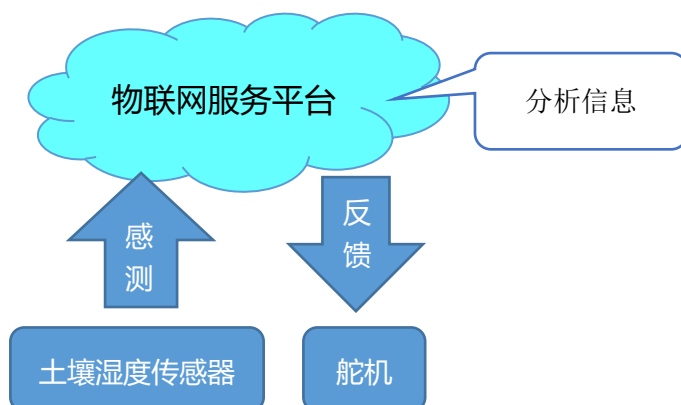
第三课 智能晾衣杆

大家洗完衣物，通常会将它们晾晒在阳光之下。上班的时候，每当遇到天气异常，心里就会记挂衣物，担心会被雨水打湿，或希望天黑时及时收取衣物。我们可以尝试制作智能晾衣杆，当检测到有雨水的时候，晾衣杆会自动收取衣物；或远程发送收衣服的信息，当晾衣杆接收到信息时，会执行收缩功能。接下来，让我们尝试制作吧！



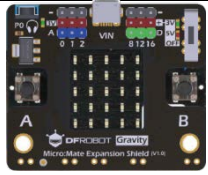
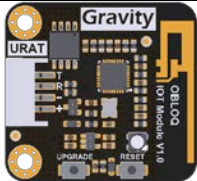
学前提示

制作远程控制的晾衣杆，可将土壤湿度传感器检测到的湿度值传递值至物联网服务器，接收到当前湿度值时，可向 OBLOQ 模块发送信息，根据获取的湿度值进行判断，当接收到“收”的信息时，控制硬件装置旋转一定的角度，实现收衣服的功能，当接收到“晾”的信息时，控制硬件装置旋转不同的角度，实现晾衣服的功能，该硬件装置可以选择为舵机，从而实现收衣服和晾衣服的功能。



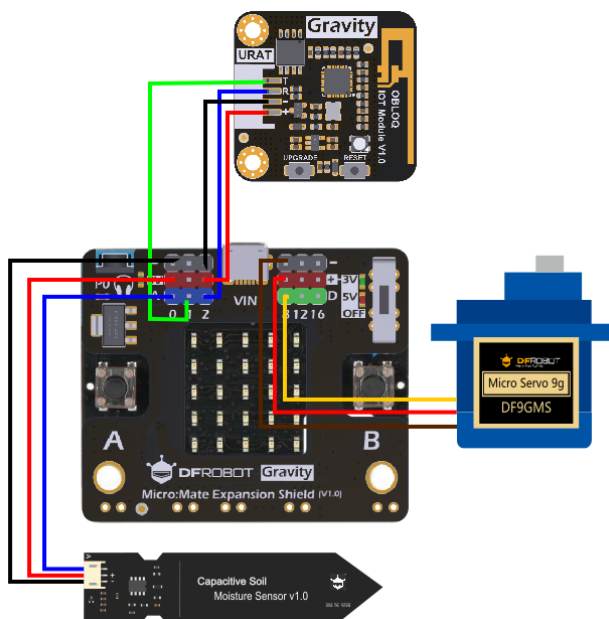
硬件清单及连接

硬件清单

		
Micro:bit+Micro:mate 扩展板	OBLOQ 模块	电脑 (需要接入互联网)
		
USB 充电宝 (备选供电)	金属 9g 舵机 (舵机模块)	电容式 土壤湿度传感器

硬件连接

OBLOQ 模块中的 T 引脚（绿线）连接扩展板上的 P1 针脚（蓝色针脚），OBLOQ 模块中的 R 引脚（蓝线）连接扩展板上的 P2 针脚（蓝色针脚），OBLOQ 模块中的“-”引脚（黑线）连接扩展板上的黑色针脚，OBLOQ 模块中的“+”引脚（红线）连接扩展板上的红色针脚。舵机模块连接扩展板上的 P8 针脚，其中黄色的线连接绿色针脚，红色的线连接红色针脚，棕色的线连接黑色针脚。土壤湿度传感器连接扩展板上的 P0 针脚，其中传感器上的 A 线连接蓝色针脚，传感器上的“+”连接红色针脚，传感器上的“-”连接黑色针脚。



知识点解析

串口监视器

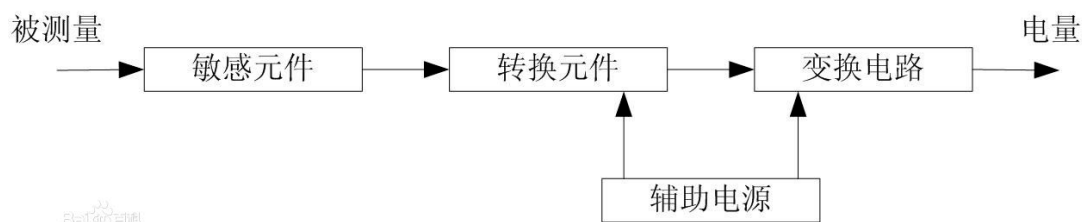
串行接口，简称串口，也就是 COM 接口，是采用串行通信协议的扩展接口。在 Mind+ 中可以用来接收传感器的信号以及数值。软件右下角黑色区域即是串口打印功能区（如下图）。电脑与主控板之间可以使用串口通讯，可以通过串口监视器显示或者发送数据。

在编程调试过程中，通过串口监视器监测变量数值变换，可以提高程序设计尤其是程序排错的效率。点击第一个按钮“打开串口”，即可看到从传感器返回的数值。如需关闭串口打印数值，点击第二个按钮“关闭串口”。



传感器

传感器是一种物理装置，能够探测，感受外界的信号、物理条件（如光照、强度、温湿度）或者化学组成，并将探知到的信息传递给其他装置。传感器一般由敏感元件、转换元件、变换电路 3 个部分组成，通过敏感元件获取外界的信息并转换成电信号输出，然后由控制器进行分析处理，如下图。



传感器的结构

传感器可以从不同角度进行分类，如果根据输出类型大致可以分为以下几类：

模拟传感器——将被测量的非电学量转换成模拟电信号，如声音传感器，光线传感器。

数字传感器——将被测量的非电学量转换成数字电信号，如人体红外传感器。

开关传感器——当一个被测量的信号达到某个特定的阈值时，传感器相应地输出一个设定的高电平或者低电平，如开关。

指令提前知

模块	功能说明
	使用串口读取数据，串口部分可以打印出数据的值。
	读取该引脚获取的模拟值
	比较符号左右两边内容的大小，判断结果为真或假
	能够合并两个字符串

程序编写

实验一：读取土壤湿度传感器的值

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——micro:bit。

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。该模块能够读取主控板连接的 P0 接口的当前值。
- 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。



代码可优化为：



在使用串口时，需初始设定串口波特率，没有设定初始串口波特率时，默认 9600。

3.烧录代码并打开串口

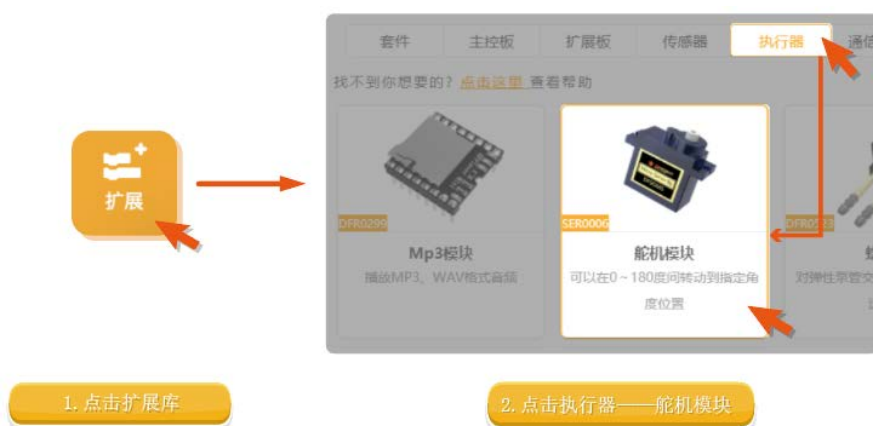
- 单击“上传至设备”，并打开串口。当土壤湿度传感器置于空气之中时，读取的引脚值在 800 以上，当向土壤湿度传感器表面附着水时，读取值会下降。



实验二：自动控制晾衣杆

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——micro:bit、执行器——舵机模块。



添加舵机模块

2.初始化舵机值

- 将“设置 P9 引脚伺服舵机为 90 度”模块拖放至右边的编辑区。该模块可以控制舵机旋转的角度，将舵机初始角度设为 100 度，表示开始操作时，衣物处于晾晒状态。



设置舵机初始状态

3.控制舵机角度

- 由实验一可以看出，土壤湿度传感器表面无水时，读取值较大，有水时，读取值较小。通过实验，可

设定临界值为 650，当读取值小于 650 时，默认当前下雨，舵机角度旋转至 10 度，模拟收取衣物；当读取值大于或等于 650 时，默认当前没有下雨，舵机角度恢复初始值 100 度，模拟晾晒衣物。

- 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。判断是否下雨，需要连续判断，运用 “循环执行” 模块。代码设计如下：



实验三：远程控制晾衣架

制作远程控制晾衣架，将读取到的土壤湿度传感器的值转递至物联网服务器，通过观测数据值，判断是否下雨，从而控制舵机旋转不同的角度。天黑或其他场景，也可通过物联网服务器发送信号控制舵机角度。

1. 添加模块

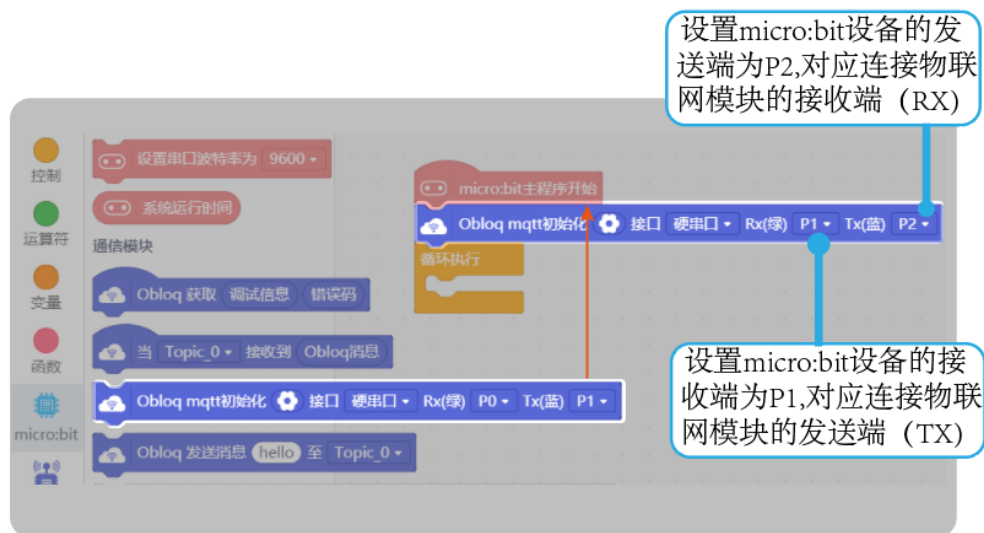
按步骤添加程序编写过程中需要的模块，添加主板——micro:bit、物联网模块——OBLOQ（[详细步骤请阅读第一章相关内容](#)）、执行器——舵机模块。



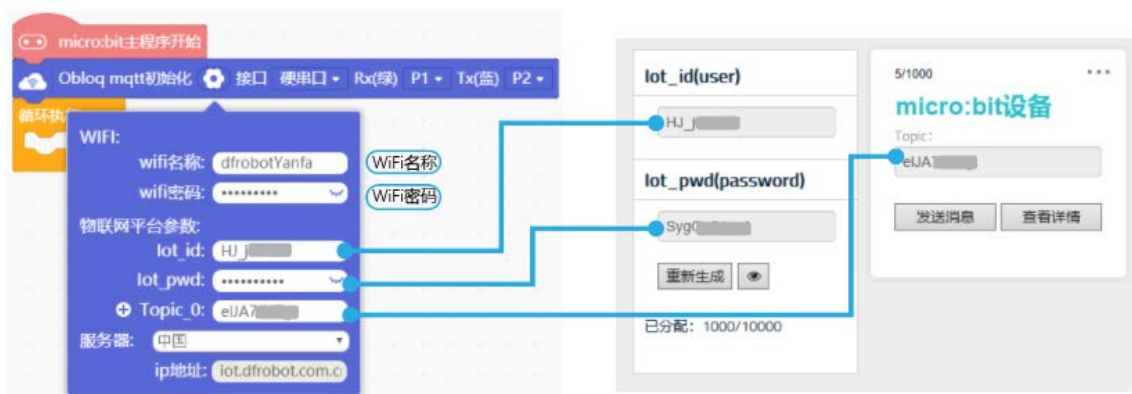
添加舵机模块

2. 物联网模块初始化及联网测试

- 将初始化模块 “ ” 拖放至右边的编辑区。然后定义发送和接收的引脚（要和实际连线一致）以及设置网页终端的相关账号与密码。



注：绿色的线接 P1 口（micro:bit 设备的接收端），蓝色的线接 P2 口（micro:bit 设备的发送端）。



设置 WiFi 密码和物联网平台参数

- 设置好物联网的相关信息后，将 “设置 P9 引脚伺服舵机为 90 度” 模块拖动到如图所示位置，控制舵机初始角度为 100 度。



3.舵机角度控制

- 当读取的湿度值小于一定范围（小于 650）时，说明当前湿度较大或有可能下雨，传递读取值至物联网终端，该读取过程是连续的，故将“循环执行”模块拖动至脚本区。传递信息时，可运用 “合并 hello world” 模块，使物联网服务器获取的信息值更加清晰。具体代码如下：



- 当接收到物联网服务器发送的“收”的信号时，舵机旋转至 10 度，达到收衣服的效果；当接收到物联网服务器发送的“晾”的信号时，舵机旋转至 100 度，达到晾晒衣服的效果。



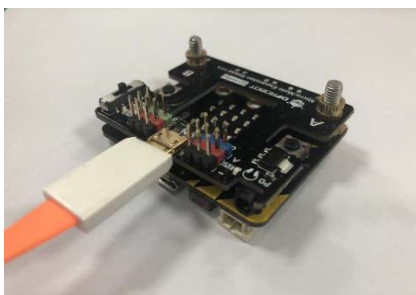
参考程序

程序展示

将下面的程序下载到 micro:bit 设备中。



注意：由于舵机耗电量大，需要外接电源。由于舵机耗电量较大，可断开舵机与板子的连接，再烧录代码，烧录代码后，可以将连在主控上面的 USB 数据线拔下，连接到扩展板上；也在不改变主控板连接 USB 数据线的基础上，用另外一根 USB 线将扩展板上的 VIN 接口连接外接电源。



当板子额外供电后，需将板载上调节按钮的电源调至 5v。



效果展示

- 进入 Easy IoT 网页发送信息给 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，当收取衣服时，发送指令“收”。

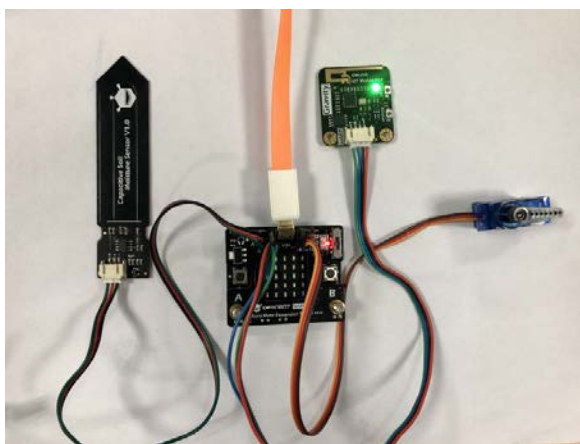
 **发送新消息**

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”

×

发送

- 当接收到“收”的指令时，可以在下面“最新消息”板块看见新消息“收”，旋转 90 度（初始值为 100 度），当前度数为 10 度。



- 当晾衣服时，发送指令“晾”。

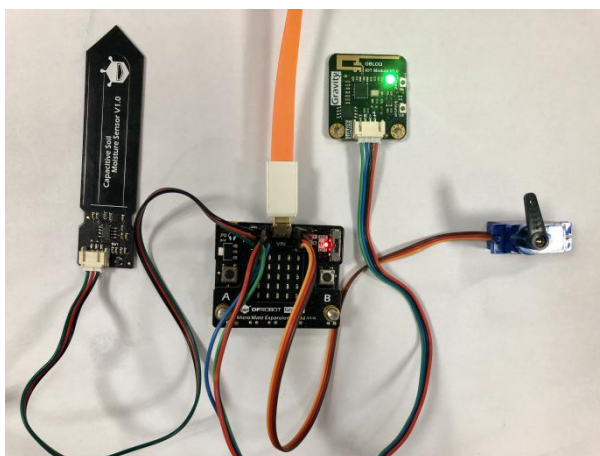
 **发送新消息**

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”

×

发送

- 当接收到“晾”的指令时，可以在下面“最新消息”板块看见新消息“晾”，旋转 90 度，当前度数为 100 度，舵机回到起始值。



拓展阅读

软串口与硬串口的区别

在物联网初始化模块中，我们发现接口有：软串口和硬串口两种。这时小朋友们可能会有疑问，这两个串口有什么区别吗？那接下来我们就分析一下这两者的区别，和在什么时候使用吧。

软串口：软串口稳定性差。但是它可以使用串口打印数据。

硬串口：硬串口稳定性好。但是它会占用到打印，如果我们需要用串口打印数据，就不能使用硬串口。



项目中，我们用串口输出模块获取土壤湿度传感器在空气之中的值。



如果我们需要将土壤湿度的数据在串口区和物联网平台（Easy IoT）上同时显示，这时 Obloq 的接口就需要选择软串口。



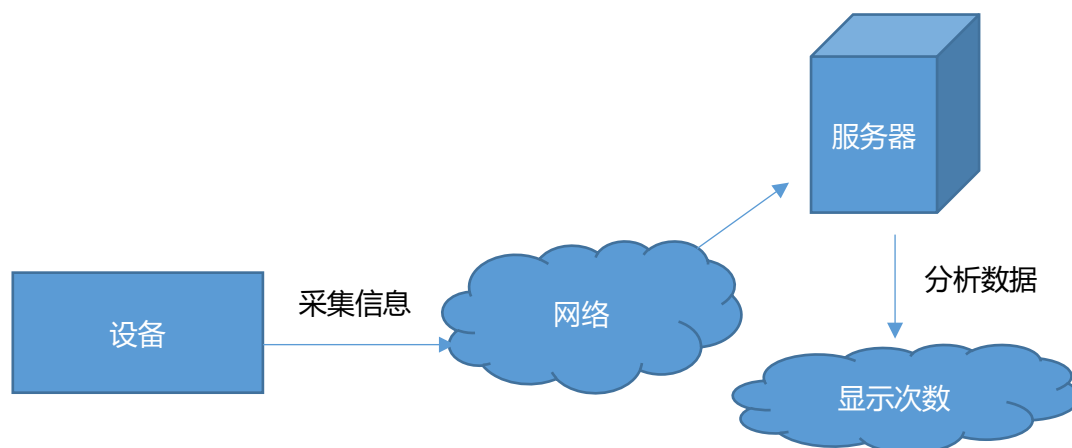
第四课 快递质量抽测

近几年，随着电子商务的快速发展。越来越多的人喜欢网购。每当快递员将包裹完美无瑕的送到我们的手中时，心情可以说是无比美丽。但是偶尔我们也会收到一些外包装破损的包裹。如果里面装的是贵重物品，可怎么办呢？那我们可不可以做一个快递质量抽测器，检测快递被暴力拿放的次数，从而判断物品在运输途中的破坏程度。



学前提示

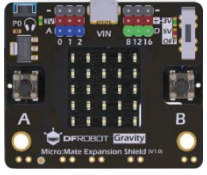
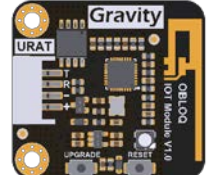
从前几章我们明白，只要将配有传感器的设备连接到网络，就能将所有的信息采集到物联网服务中。从快递质量抽测这个角度来讲，如果我们给快递植入一个加速度传感器，快递在发货后，通过收集和分析传感器数据来判断快递被暴力拿放的次数。



快递质量抽测的项目结构分析图

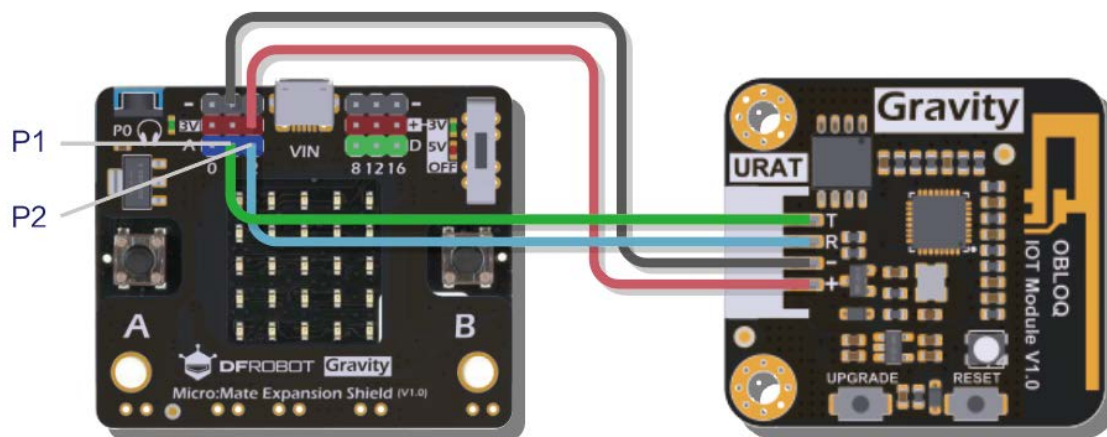
硬件清单及连接

硬件清单

			
Micro:bit+Micro:mate 扩展板	OBLOQ 模块	电脑 (需要接入互联网)	USB 充电宝 (备选供电)

硬件连接

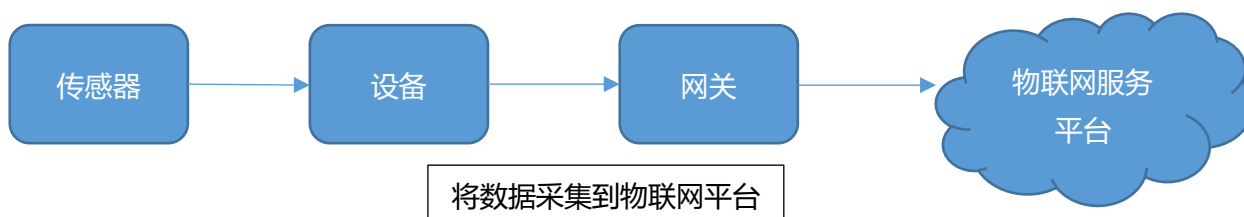
OBLOQ 模块中的 T 引脚（绿线）连接扩展板上的 P1 针脚（蓝色针脚），OBLOQ 模块中的 R 引脚（蓝线）连接扩展板上的 P2 针脚（蓝色针脚），OBLOQ 模块中的“-”引脚（黑线）连接扩展板上的黑色针脚，OBLOQ 模块中的“+”引脚（红线）连接扩展板上的红色针脚。



知识点解析

物联网数据分析

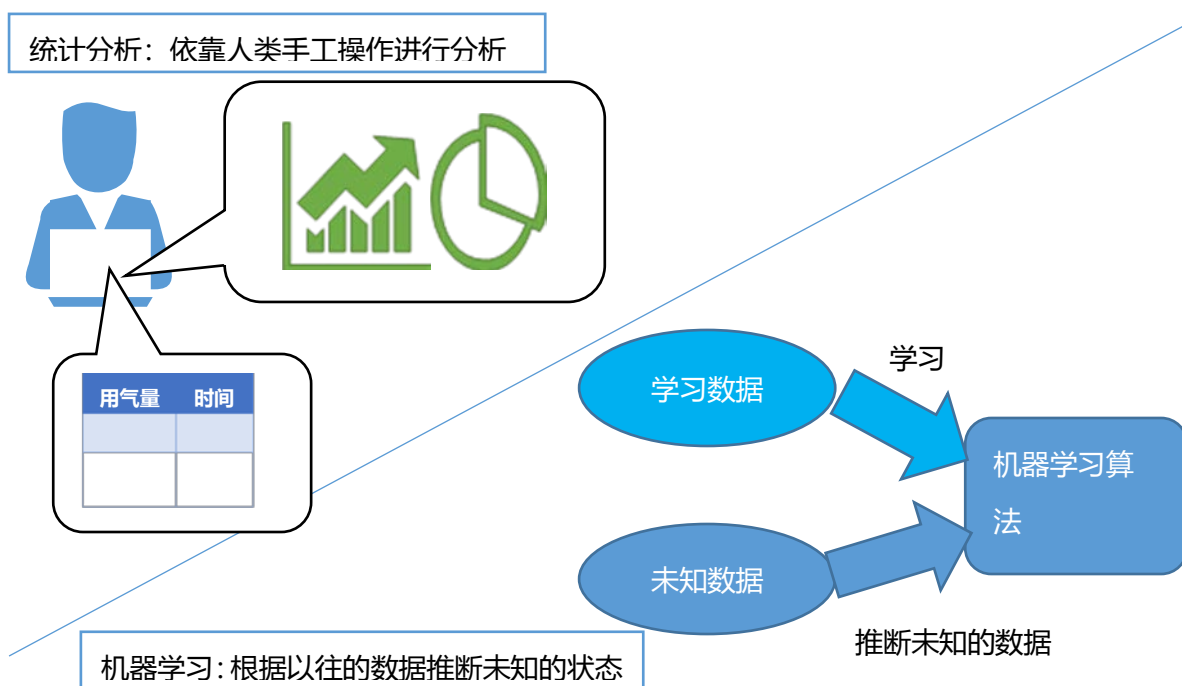
基于物联网服务构建的基础，我们知道物联网数据只要是指传感器和设备发过来的数据，这些数据一部分是对现实环境参数的采集值，一部分是设备的一些常规信息值，比如：状态、故障信息、错误代码、运行情况等。



但是，光是采集传感器和设备发来的数据，那就只不过是堆庞大的数据聚在一起而已，很难直接应用这些数据。为了实现服务，需要从采集到的数据中分析出有价值的信息。只有通过对数据进行分析，才有可能掌握机器的运转情况。

比如：利用物联网的数据可以有效预判机械的故障，减少因为设备故障而带来的工作耽误。根据设备上传的设备状态信息值的变化，以及特定传感器采集的参数，就可以监测预判发生故障的概率，同时提前做出响应，这样就能够把一个物联网服务从一个单纯的数据采集，上升到一个能够不断创造附加值的服

基于数据应用的用途考虑，我们将数据分析方法暂时可分为统计分析和机器学习法。



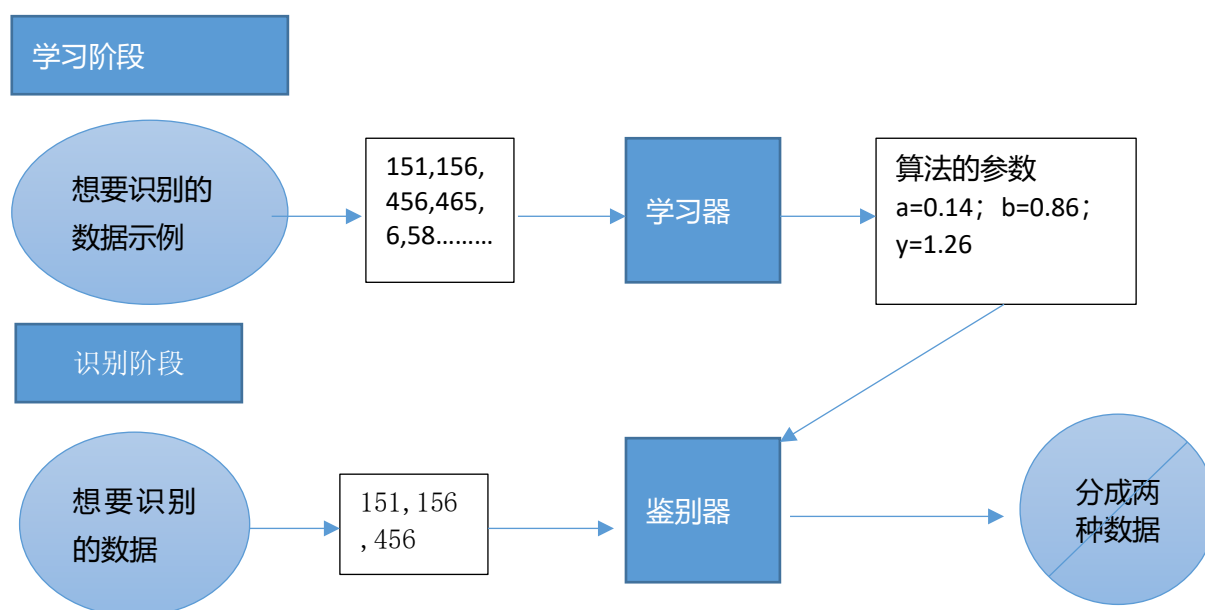
数据分析的两种方法

统计分析法主要是依靠分析师手工进行分析，基于一定的目的提取数据，并进而以各种形式进行展示。比如家里面的智能燃气表采集了用气的数据，那么我们可以用图形表示一定时间内用气量的变化，从而直观的掌握用气数据的变化趋势。



用气量和时间的变化

机器学习法，就是说的比较多的人工智能，主要依据对以往数据的学习，来推断未知的数据或者状态。机器学习主要分为两个阶段：学习阶段和识别阶段。在学习阶段，一个名为“学习器”的程序会基于一些训练数据，机械的掌握这些数据之间的联系。作为学习阶段的结果，计算机会根据机器学习的算法输出参数，然后以这个参数为基础创建叫作“鉴别器”的程序。只要把未知的数据给这个鉴别器，就能输出最适合这个值的结果。



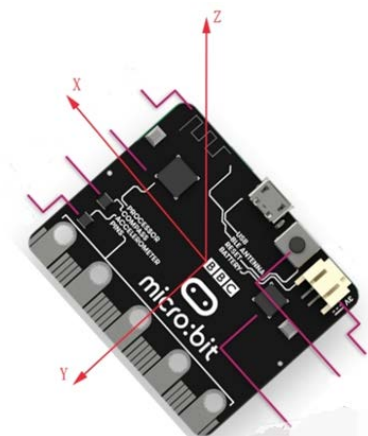
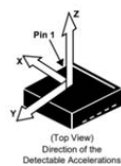
机器学习示例

举个例子，假设我们想利用传感器来识别房间里有没有人。这种情况下，就要准备两组数据——房间里有人时传感器的数据和房间里没人时传感器的数据。计算机通过把这两组数据分别交给学习器，来获取鉴别器用的参数，以参数为基准制作鉴别器，只要输入从各个感测设备接收到的数据，鉴别器就能输出结果，告诉我们现在房间里是否有人。

什么是加速度传感器

加速度传感器是一种能够测量加速力，将加速度转换为电信号的电子设备。加速力就是当物体在加速过程中作用在物体上的力，就好比物体下坠时，受到重力作用。加速度力可以是个常量，比如 g ，也可以是个变量，其范围比重力感应器要大。

micro:bit 自带一个三轴加速度传感器，可以检测 x,y,z 三个方向的重力加速度，每个轴测量的值是正数或者负数，当读数为 0 时，表示加速度传感器沿着该特定轴“水平”位置。加速度值的强度表示 x,y,z 的矢量和。矢量表示带有方向的量，矢量和指方向与大小的和。在高中和大学的数学课中会讲到。



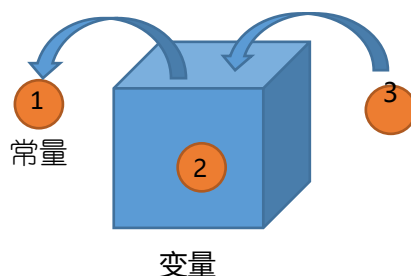
什么是变量和常量

变量是在某一变化过程中，可以取不同数值的量。

常量是在某一变化中，始终保持不变的量。

在这里，我们可以将变量理解成一个用来存储数据的盒子，盒子里面可以暂时存放各种各样的数据。比如我们用来存放整数，装了 1 之后，再装 2 的话，1 会被 2 替换掉，去盒子里只会拿到 2 这个数字。同理，如果装 3,4,5.....都会依次被新的数据替换。

常量可以理解成盒子里的某一个数据。例如，其中的数字 1，这个数字就是一个常量。



指令提前知

模块	功能说明
	数字型变量
	设置变量的初始值
	让变量以 1 个步长增减
	大于运算符，判断左边的值大于右边
	使用串口读取数据，下拉窗口可选择输出方式。
	读取三轴加速度传感器的值，下拉窗口中可选择 X、Y、Z、强度。

程序编写

1.添加主控板和物联网通信模块

根据下图的步骤添加主板——micro:bit 和物联网模块——OBLOQ（详细步骤请阅读第一章相关内容）



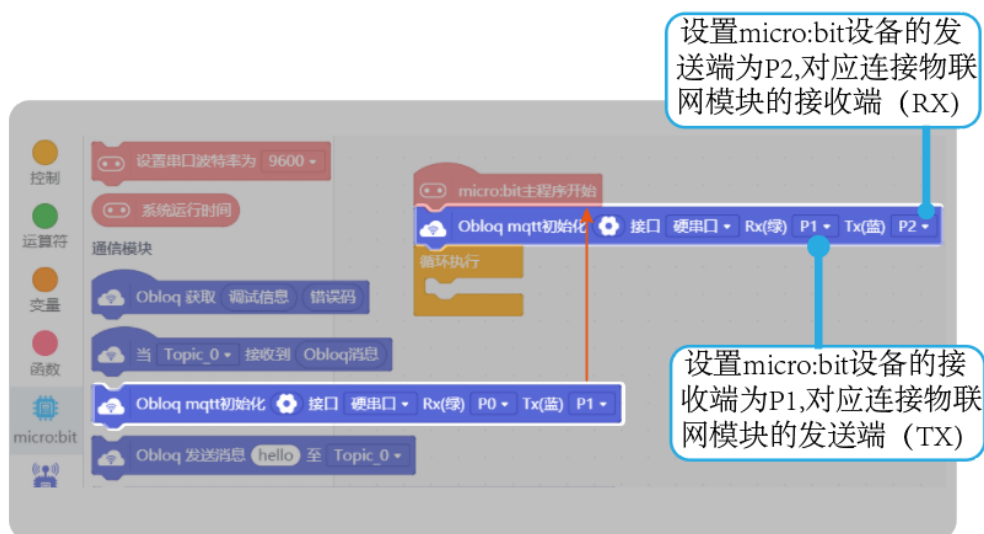
添加主板



添加物联网模块

2.物联网模块初始化及联网测试

- 将初始化模块 “Obloq mqtt初始化 接口 硬串口 • Rx(绿) P0 • Tx(蓝) P1 •” 拖放至右边的编辑区。然后定义发送和接收的引脚（要和实际连线一致）以及设置网页终端的相关账号与密码。



注：绿色的线接 P1 口（micro:bit 设备的接收端），蓝色的线接 P2 口（micro:bit 设备的发送端）。



设置 WiFi 密码和物联网平台参数

- 设置好物联网的相关信息后，将发送消息指令 “Obloq 发送消息 hello 至 Topic_0” 拖动到如图所示位置，帮助确认是否联网成功。



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。



3.设置变量

- 首先找到变量模块点击 “新建数字类型变量” 创建变量 “被摔次数”，其步骤如下图。



- 然后找到 “设置 被摔次数 的值为 0” 模块，将它拖放到如图所示位置。



- 将条件判断指令 “如果 那么执行” 拖放到如图所示位置，判断加速度的综合强度值。



根据测试，2200 是一个比较合适的值，小朋友们可以根据后期实际测试情况调整；当加速度强度 > 2200 时，被摔次数 + 1。



提示：

什么是步长？步长就是每次递增或者递减的幅度；步长为 1，每次执行一次加 1；步长为-1，每执行一次减。

在程序中我设置的初始值为 0，因为在项目中是用于计数，所以需要以 1 为步长逐渐递增。点击读取加速度值的下拉箭头，还可以选择 x,y,z 三轴的方向。



x 沿着左右方向加速度值；

y 沿着前后方向加速度值；

z 沿着上下方向加速度值；

强度：3 个方向加速度值的矢量和。

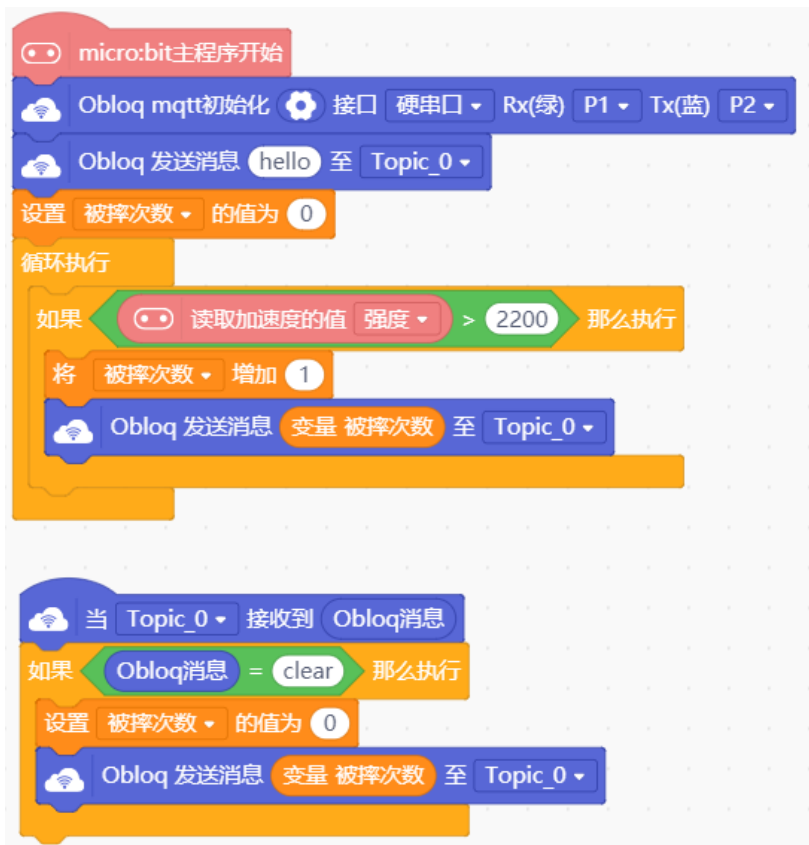
- 重置快递质量抽测器，通过网页终端发送指令“clear”，将变量“被摔次数”设置为 0。可以重新开始记录被摔次数。



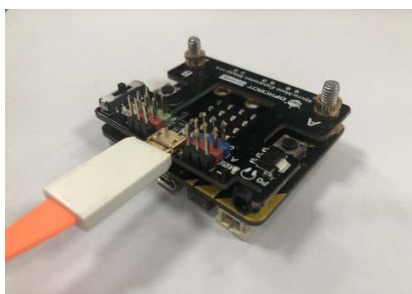
参考程序

程序展示

将下面的程序下载到 micro:bit 设备中。



注意：程序上传完毕后，将连在主控上面的 USB 数据线拔下，连接到扩展板上；也可以在不改变主控板连接 USB 数据线的基础上，用另一根 USB 线将扩展板上的 VIN 接口连接外接电源上。



当板子额外供电后，需将板载上调节按钮的电源调至 5v。



效果展示

- 进入 Easy IoT 网页发送信息给 micro:bit 设备。找到相应的 Topic，点击“发送消息”。

24/1000

micro:bit设备

Topic:

[发送消息](#) [查看详情](#)

- 当你用力晃动 micro:bit 主板时，刷新网页，就能看见被摔的次数。

最新消息	
时间	消息
2019/7/31 10:4:35	3
2019/7/31 10:4:30	2
2019/7/31 10:4:27	1
2019/7/31 10:3:38	hello

- 将清屏指令“clear”，填写在对话框中，点击发送，即可将消息发送出去。

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"

×

发送

最新消息

时间	消息
2019/7/31 10:21:29	clear

- 当消息“clear”发送出去后，刷新网页，被摔次数归 0，这时重新计数。

最新消息	
时间	消息
2019/7/31 10:21:31	0
2019/7/31 10:21:29	clear

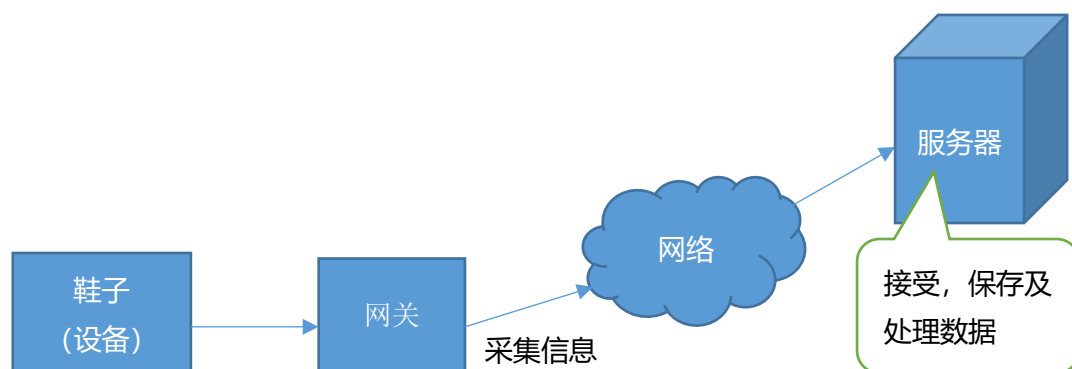
第五课 多功能鞋子

不知道大家是否还记得 2016 年 nike 公司推出了一款天价鞋子 Nike Mag。相比其他的鞋子，它可是酷出了天际，自动化绑鞋带、酷炫的灯光效果，浑身都充满着科技元素。这样的鞋子，我们买不起，但是我们可以自己制作一款多功能鞋子，让它在记录步数的同时，也能提醒我们刚刚的动作很危险，以后需要注意。



学前提示

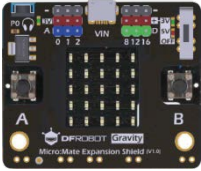
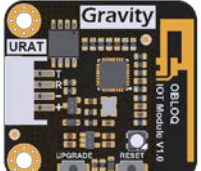
多功能鞋子的原理是在鞋子里装入加速度传感器，通过网络将采集到的数据上传至物联网服务，然后分析数据的变化，实现计步和危险报警。



多功能鞋子的项目结构分析图

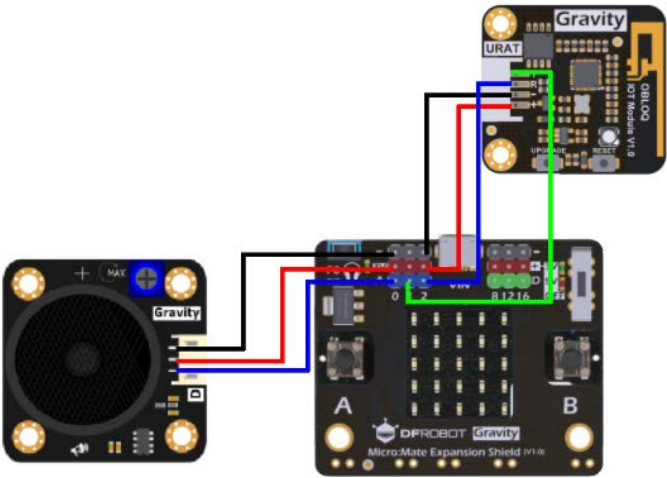
硬件清单及连接

硬件清单

				
OBLOQ 模块	Micro:bit+Micro:mate 扩展板	电脑 (需要接入互联网)	USB 充电宝 (备选供电)	带功放喇叭模块

硬件连接

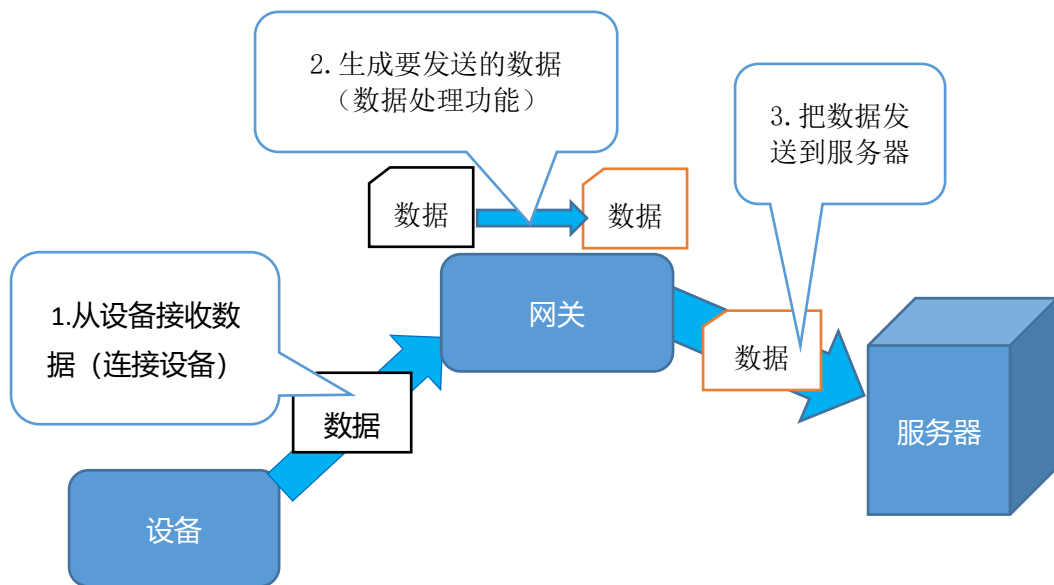
OBLOQ 模块中的 T 引脚 (绿线) 连接扩展板上的 P1 针脚 (蓝色针脚), OBLOQ 模块中的 R 引脚 (蓝线) 连接扩展板上的 P2 针脚 (蓝色针脚), OBLOQ 模块中的 “-” 引脚 (黑线) 连接扩展板上的黑色针脚, OBLOQ 模块中的 “+” 引脚 (红线) 连接扩展板上的红色针脚。喇叭模块连接扩展板上的 P0 针脚, 根据对应的颜色接线。



知识点解析

网关的作用

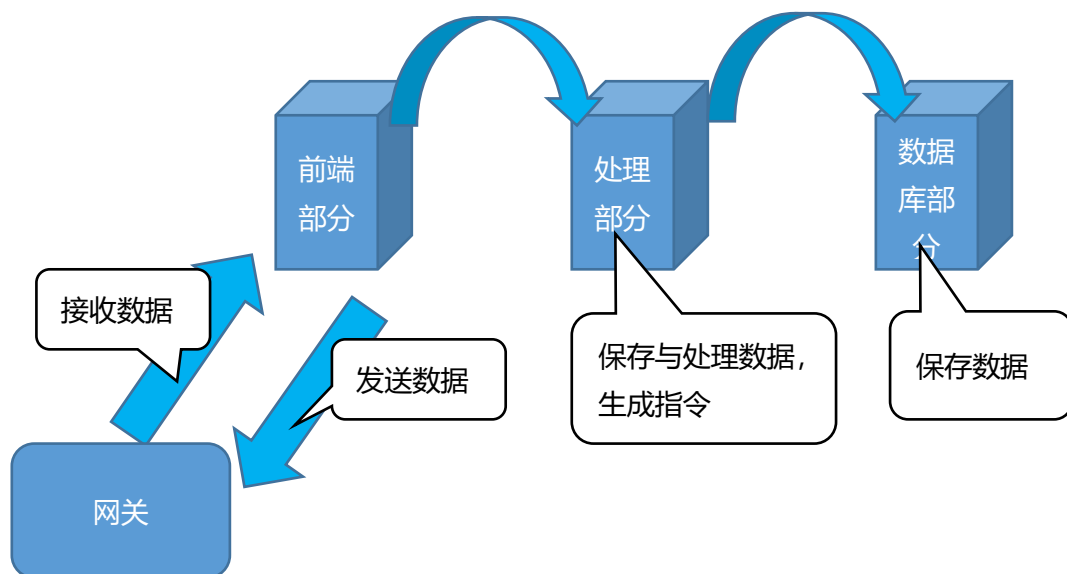
在物联网使用的设备中, 有一些是不能直接连接到互联网的, 需要通过网关把这些设备转发到互联网。网关分别由连接设备功能, 数据处理功能和向服务器发送数据的功能构成。



网关的功能

物联网服务

物联网服务可以分为 3 个部分：前端部分，处理部分，数据库部分。一般前段部分包括数据接收服务器和数据发送服务器，数据接收服务器接收到设备和网关发送来的数据后，将它转给处理部分“处理”。数据发送服务器则负责把从处理服务器接收到的内容发送给设备。数据库部分是用来保存数据。



物联网服务的功能

指令提前知

模块	功能说明
	减法运算符，运算左右两个值的差值。
	绝对值运算符
	播放音符模块
	显示文字模块，显示英文和数字。
	按键触发指令，可选择 A，B，A+B 三种模式

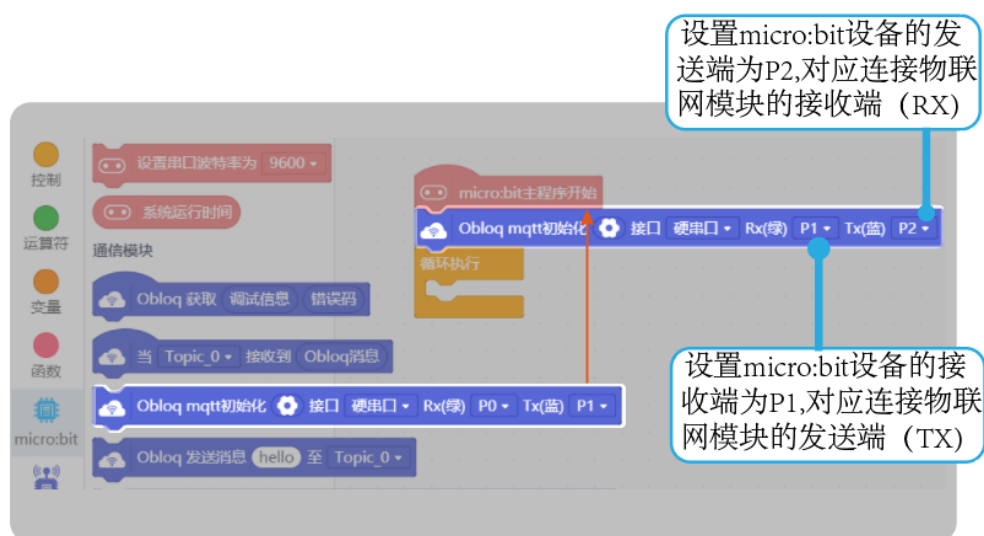
程序编写

1.添加主控板和物联网通信模块

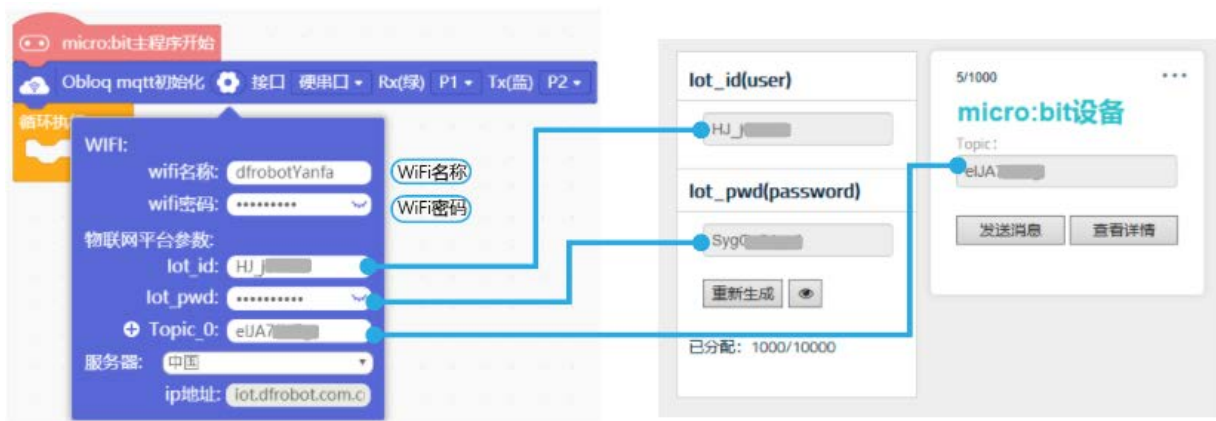
根据第一章相关内容添加主板——micro:bit 和物联网模块——OBLOQ。

2.物联网模块初始化及联网测试

- 将初始化模块 “Obloq mqtt初始化 接口 硬串口 Rx(绿) P0 Tx(蓝) P1” 拖放至右边的编辑区。然后定义发送和接收的引脚（要和实际连线一致）以及设置网页终端的相关账号与密码。



注：绿色的线接 P1 口（micro:bit 设备的接收端），蓝色的线接 P2 口（micro:bit 设备的发送端）。



设置 WiFi 密码和物联网平台参数

- 设置好物联网的相关信息后，将发送消息指令 “Obloq 发送消息 hello 至 Topic_0” 拖动到如图所示位置，帮助确认是否联网成功。



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。



3.设置变量

- 首先找到变量模块点击 “新建数字类型变量” 创建 3 个变量 “步数, X1, X2”。



- 然后找到 “设置 步数 的值为 0” 和 “设置 X1 的值为 0” 模块，将它拖放到如图所示位置。



- 将模块 “设置 X2 的值为 0” 拖放到如图所示位置。



- 将条件判断指令 “如果 那么执行” 拖放到如图所示位置，用来判断每走一步，X 轴方向加速度值的变换。



- 将大于运算符 “>” 和绝对值运算符 “绝对值” 拖放到如图所示位置。





根据测试，在走路的时候，腿甩动的时候，X1 和 X2 的绝对值大于 300（数值根据自己的实际情况设置），这时步数就+1。判断完之后，将当前的加速度值（X2）赋值给 X1，用于计算接下来这一步的变化。



提示：

什么运算符？运算符是说明特定操作的符号，在 mind+中一般使用常见的三大类：算术运算符、关系运算符、逻辑运算符。

在程序中，我们运用了绝对值来计算加速度计 x 轴前后数据值的变化。点击该模块的下拉箭头，可以选择更多的数学符号。



- 经过测试，当运动幅度过大时，加速度计的强度>2200（数值根据自己的实际情况设置）。所以我将程序设置如下，当强度>2200，喇叭播放提示音，并向 Easy IoT 平台发送危险的信号。



- 设置事件触发指令，当 A 键被按下时，步数清零。当 B 键按下，显示步数，并向 Easy IoT 发送步数。

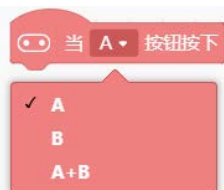


提示 1:

播放音符模块的使用方法如下:



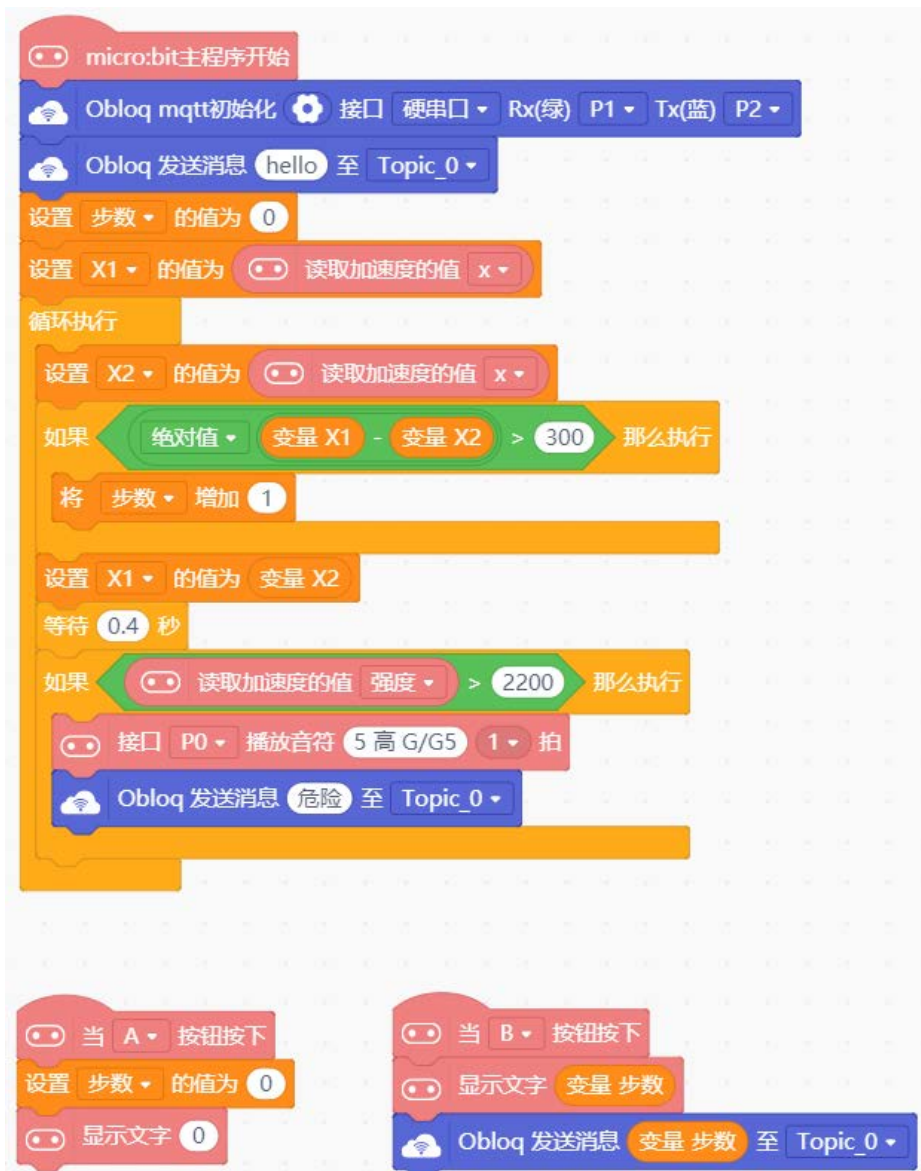
事件触发指令——按键触发，点击下拉箭头可以选择三种按键方式:



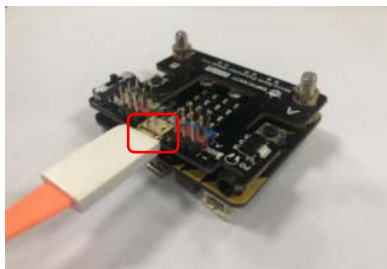
参考程序

程序展示

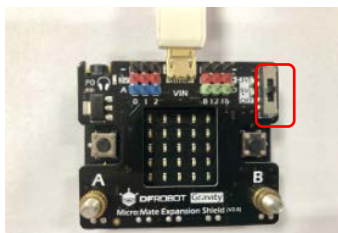
将下面的程序下载到 micro:bit 设备中。



注意：由于多连接了喇叭，需要外接电源。可以将连在主控上面的 USB 数据线拔下，连接到扩展板上的电源接口（VIN 口）。也可以在不改变主控板上的 USB 线，用另外一根 USB 线连接在 VIN 接口。



当板子额外供电后，需将板载上调节按钮的电源调至 5v。



效果展示

- 进入 Easy IoT 网页发送信息给 micro:bit 设备。找到相应的 Topic，点击“发送消息”。
- 如果我们在行走的过程幅度过大，喇叭会响，Easy IoT 会收到危险的提示。

查询结果

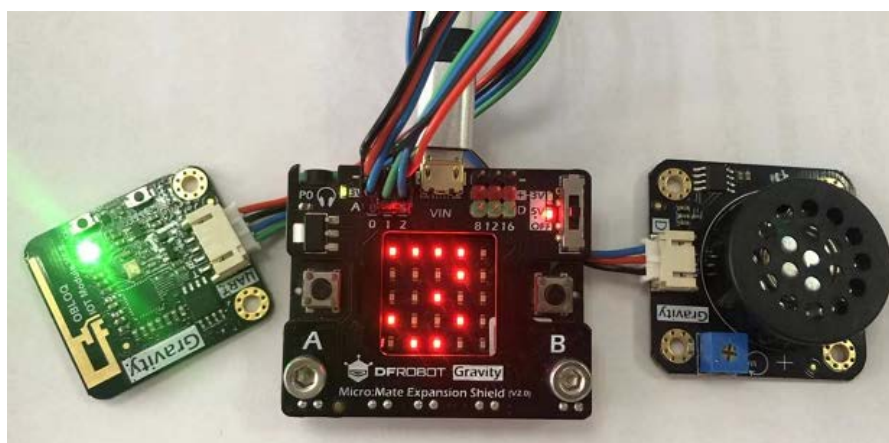
时间	消息	操作
2019/8/8 15:17:22	危险	
2019/8/8 8:27:35	hello	

这样就可以记录一天中，我们有多少次是处于在一个比较危险的状态。

- 当我们想查看已经走了多少步时，按下按键 B 就可以获取步数。同时将数据发送给 Easy IoT

查询结果

时间	消息	操作
2019/8/8 15:52:9	3	



- 按下 A 键，重置设备，开始新一轮计步和测险。



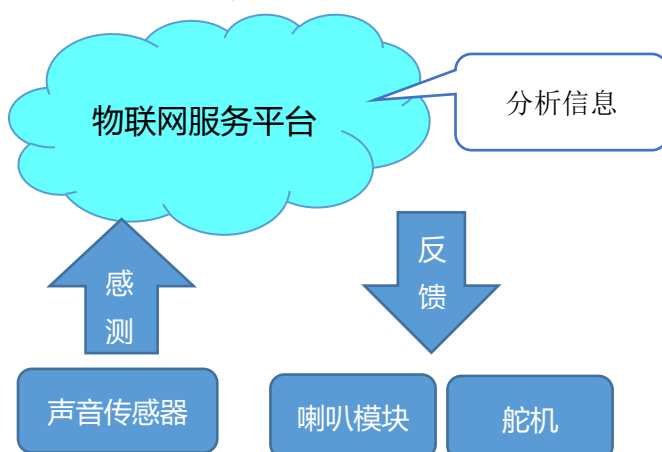
第六课 智能婴儿摇篮

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，它不固定的作息時間、半夜的哭鬧又給父母們帶來極大的痛苦。如果能夠遠程哄娃，將會給家長們帶來更多的便利。接下來，我們嘗試制作智能嬰兒搖籃吧！



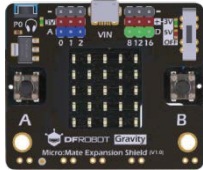
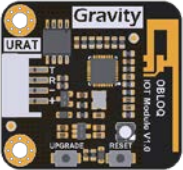
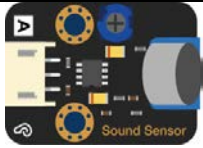
学前提示

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 OBLOQ 模块发送信息，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。



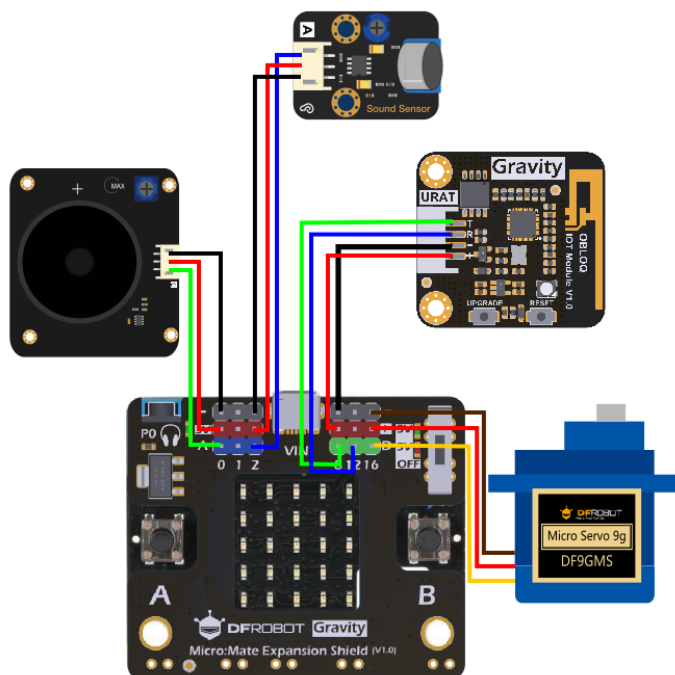
硬件清单及连接

硬件清单

			
OBLOQ 模块	Micro:bit+Micro:mate 扩展板	电脑 (需要接入互联网)	USB 充电宝 (备选供电)
			
模拟声音传感器	带功放喇叭模块	金属 9g 舵机	

硬件连接

OBLOQ 模块中的 T 引脚（绿线）连接扩展板上的 P8 针脚（绿色针脚），OBLOQ 模块中的 R 引脚（蓝线）连接扩展板上的 P12 针脚（绿色针脚），OBLOQ 模块中的“-”引脚（黑线）连接扩展板上的黑色针脚，OBLOQ 模块中的“+”引脚（红线）连接扩展板上的红色针脚。声音传感器连接扩展板上的 P2 针脚，喇叭模块连接扩展板上的 P0 针脚。舵机模块连接扩展板上的 P16 针脚，其中黄色的线连接绿色针脚，红色的线连接红色针脚，棕色的线连接黑色针脚。



知识点解析

模拟声音传感器

物理老师告诉我们，震动产生声音，声音的传播形式是声波。雷声轰鸣可以劈开树杈，雪山大声吼会引起雪崩，这些现象都可以证明声音的能量。



首先我们先认识一下，我们人类是怎么听到声音的呢？声音通过耳道传到鼓膜，引起鼓膜振动，振动传到听小骨，听小骨跳动再通过耳蜗将振动传递给大脑，这时就是我们能听到的声音了。

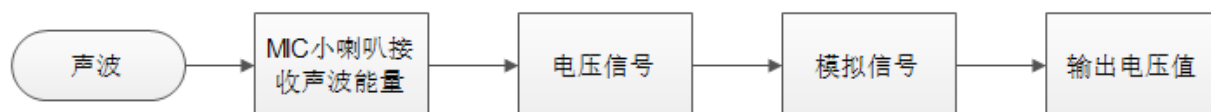


耳朵内部结构

模拟声音传感器可以比喻成设备的“电子耳朵”，它根据传感器上的 MIC 小喇叭接收声波的能量，并将声波能量大小转化为不同的电压信号。然后结合其他电路元件将电压信号转换成模拟信号，再反馈出电压值来表达检测到的声音大小，这就是模拟声音传感器的原理了。



MIC 小喇叭



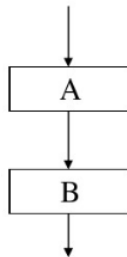
模拟声音传感器原理图

程序的三种基本结构

1996 年，计算机科学家 Bohm 和 Jacopini 证明了：任何简单或复杂的算法都可以由顺序结构、选择结构和循环结构这三种基本结构组合而成。

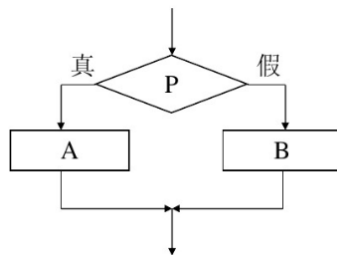
顺序结构：

- 1) 程序中的各个操作是按照它们在源代码中的排列顺序依次执行的；
- 2) 流程图如下：



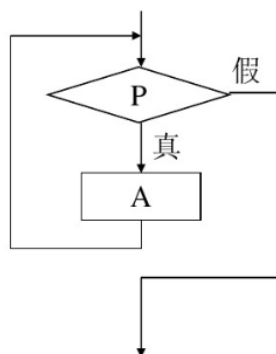
选择结构：

- 1) 根据某个特定的条件进行判断后，选择其中一支执行；
- 2) 流程图如下（其中一种情况）：



循环结构：

- 1) 反复执行某个或某些操作，知道条件为假或为真时才停止循环；
- 2) 流程图如下（其中一种情况）：



指令提前知

模块	功能说明
 接口 P0 ▾ 播放声音 BIRTHDAY ▾ 直到结束	连接 P0 引脚的传感器播放音乐。

 停止后台播放	停止播放音乐
--	--------

程序编写

实验一：读取声音传感器的值

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——micro:bit。

2.读取串口值

- 将 “ 串口 字符串输出 hello 换行 ” 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 “ 读取模拟引脚 P0 ” 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 “ 等待 0.5 秒 ” 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

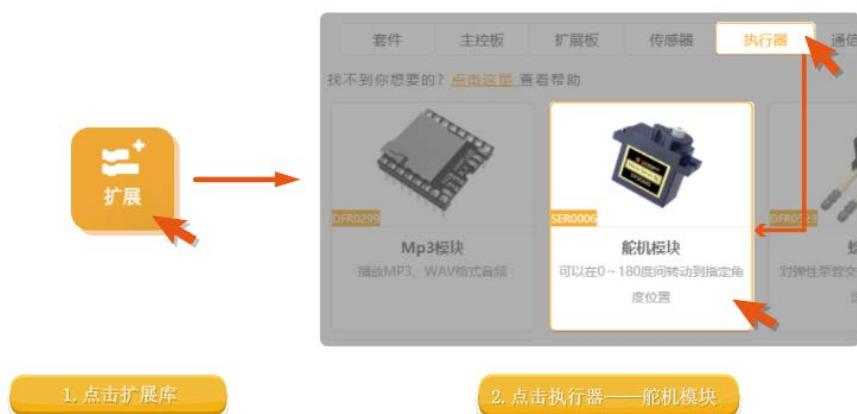
- 单机 “上传至设备”，并打开串口。可显示当前声音传感器读取的声音强度，不同环境下读取值有所差别。写该教程时，当前安静环境下，声音传感器读取的值在 10~20 之间，有较大噪音时，声音传感器读取的值在 150 以上。



实验二：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——micro:bit、执行器——舵机模块。



添加舵机模块

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：

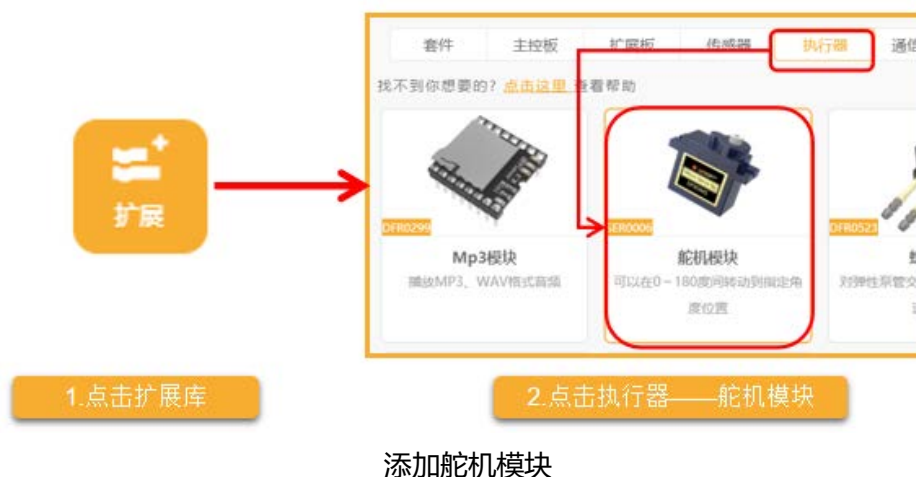


实验三：远程控制摇篮

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。

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——micro:bit、物联网模块——OBLOQ（[详细步骤请阅读第一章相关内容](#)）、执行器——舵机模块。



2.连接物联网并发送数据

- 将初始化模块 “Obloq mqtt初始化 接口 硬串口 Rx(绿) P0 Tx(蓝) P1” 拖放至右边的编辑区。然后定义发送和接收的引脚（要和实际连线一致）以及设置网页终端的相关账号与密码。OBLOQ 模块中的 T 引脚（绿线）连接扩展板上的 P8 针脚（绿色针脚），OBLOQ 模块中的 R 引脚（蓝线）连接扩展板上的 P12 针脚（绿色针脚）。
- 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。



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：



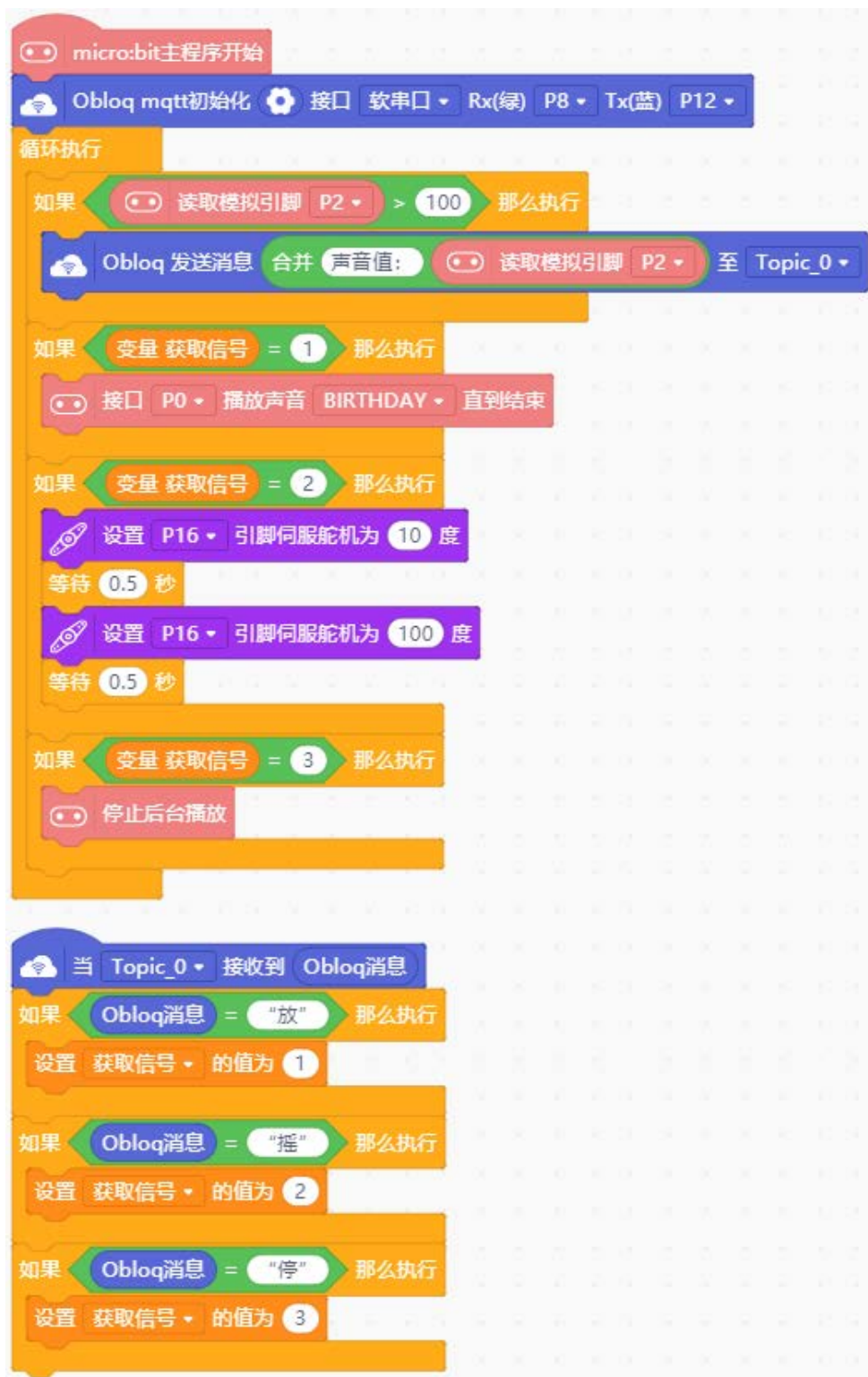
- 当接收到信号值为“1”时，控制 P0 连接的喇叭模块播放音乐；当接收到的信号值为“2”时，控制舵机 10 度-100 度之间来回切换，形成摇篮摇摆的状态；当接收到的信号值为“3”时，停止播放音乐和舵机角度切换。



参考程序

程序展示

将下面的程序下载到 micro:bit 设备中。



注意:将程序上传至设备后, 需要将连在主控上面的 USB 数据线拔下, 连接到扩展板上的电源接口 (VIN 口)。额外供电后, 将板载上的调节开关调至 5V。

效果展示

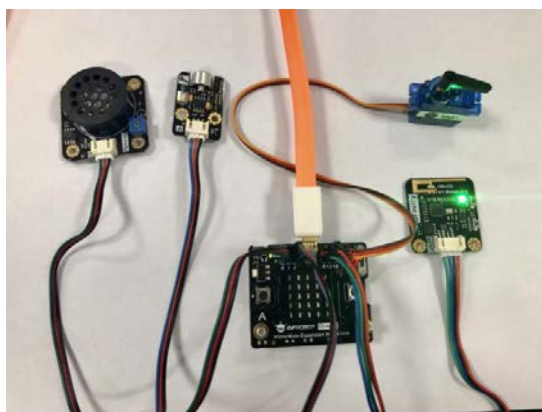
- 进入 Easy IoT 平台发送信息给 micro:bit 设备。找到相应的 Topic，点击“发送消息”。
- 将想要发送的指令，填写在对话框中，点击发送，即可将消息发送出去，发送指令“放”，实现播放音乐。

 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"

发送

- 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”，喇叭模块开始播放音乐。



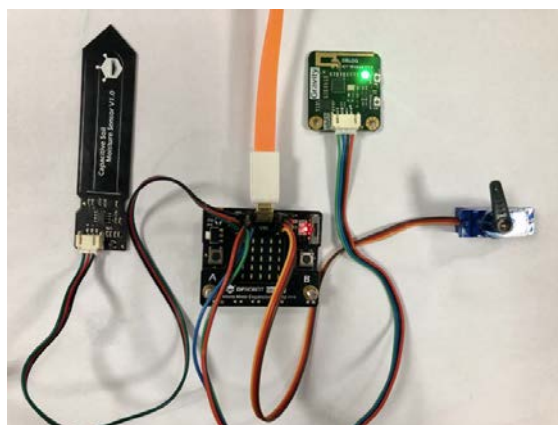
- 发送指令“摇”，实现摇篮摇摆。

 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"

发送

- 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。

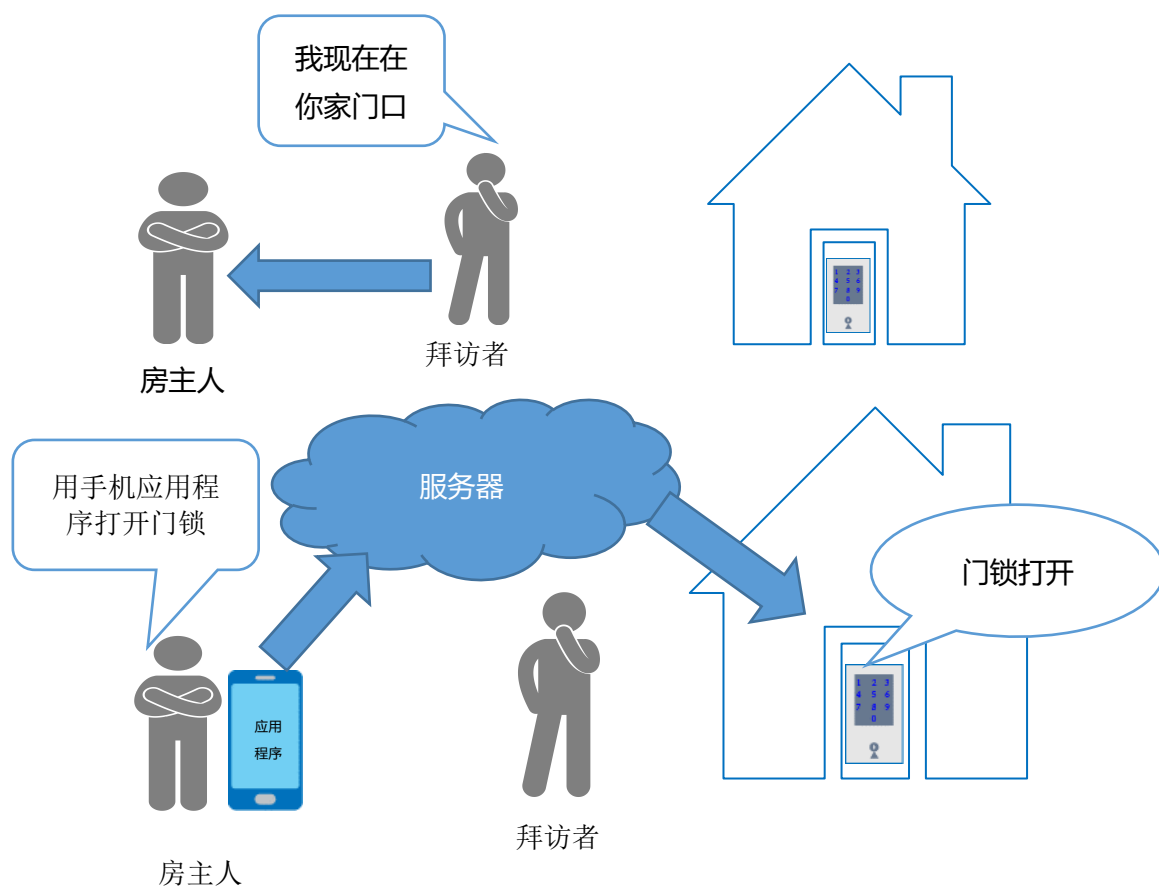
第七课 智能门锁

以前我们出门，钥匙都是放在钥匙包里面随身携带，如果遇到你不在家的时候，朋友突然拜访，这时你只能自己赶回去帮她开门。现在随着智能门锁的普及，远程开锁已经不是难题了。如果你用的是智能门锁，即使你出门忘记带钥匙，或者亲朋好友来家里拜访时，就算你远在外地。你也可以利用手机上的应用程序将房门打开。



学前提示

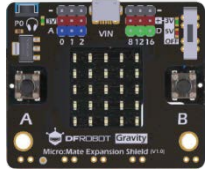
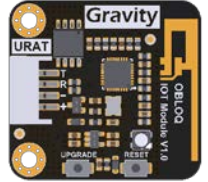
智能门锁，设备将自己感测到的信息发送给物联网服务器提醒主人，主人根据是否接到电话来判断是否开门？



智能门锁的项目结构分析图

硬件清单及连接

硬件清单

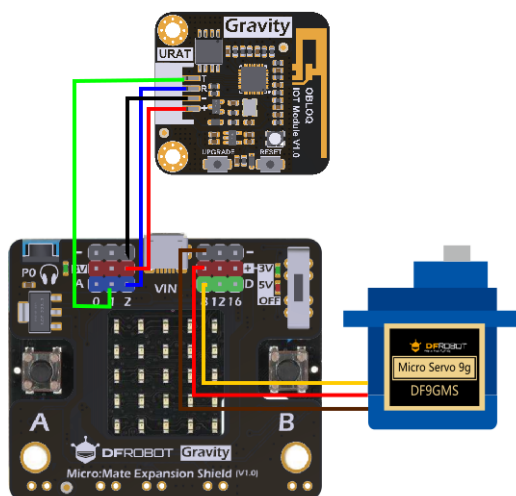
		
2 块 Micro:bit 主板 +2 块 Micro:mate 扩展板	OBLOQ 模块	电脑 (需要接入互联网)
		
USB 充电宝 (备选供电)	金属 9g 舵机 (舵机模块)	带功放喇叭模块

硬件连接

在这个项目中，我们需要用到无线电收发，所以需要两块 micro:bit 主板和扩展板，分别作为发射端和接收端。

发射端电路连接

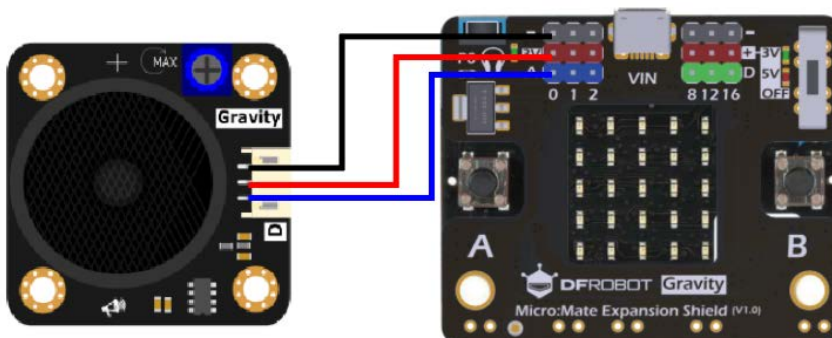
OBLOQ 模块中的 T 引脚（绿线）连接扩展板上的 P1 针脚（蓝色针脚），OBLOQ 模块中的 R 引脚（蓝线）连接扩展板上的 P2 针脚（蓝色针脚），OBLOQ 模块中的“-”引脚（黑线）连接扩展板上的黑色针脚，OBLOQ 模块中的“+”引脚（红线）连接扩展板上的红色针脚。舵机模块连接扩展板上的 P8 针脚，其中黄色的线连接绿色针脚，红色的线连接红色针脚，棕色的线连接黑色针脚。



发射端电路连接

接收端电路连接

将喇叭模块连接在另外一块扩展板上的 P0 针脚。

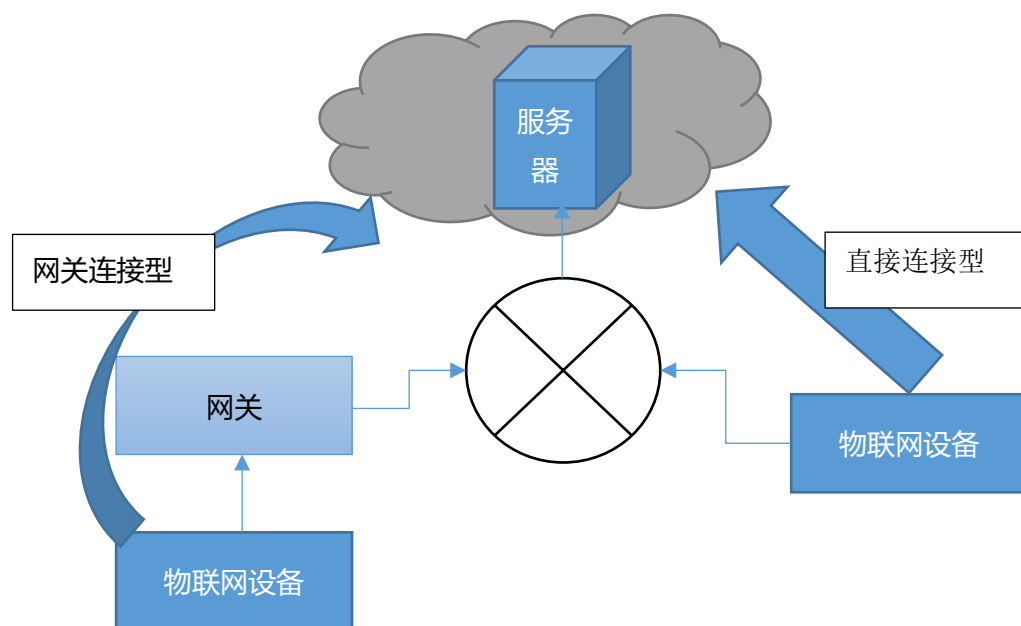


接收端电路连接

知识点解析

物联网设备与网络相连接的方式

设备连接到网络的方式有两种，一种是设备本身直接连接到网络，另一种是在本地区域使用网关连接网络。近年来，“生活记录”型的设备越来越多，其结构更接近第二种连接方式，例如通过蓝牙将可穿戴设备和智能手机配对，通过智能手机向服务器发送数据。网络的连接形式如下图。

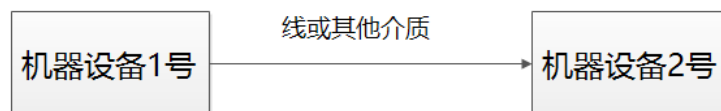


网络连接形式

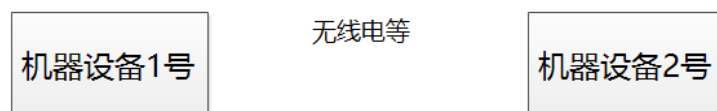
与网关设备的通信方式

前面我们介绍了物联网设备与网络连接的两种方式，与物联网设备相比较，网关设备的硬件结构都比较丰富，他们之间的通信方式也有很多，既有有线的也有无线的。

有线通信：是指传输媒质为导线、电缆、光缆、波导、纳米材料等形式的通信，其特点是媒质能看得见，摸得着的。例如：以太网，串行通信，USB 等。



无线通信：是指的不用导线、电缆、光纤等有线介质，而是用空间来传递信号的通信方式。例如:wifi,4G/LTF, 3G/LTF, 蓝牙等。



指令提前知

模块	功能说明
	打开无线通信
	设置无线频道
	检测按钮被按下
	无线发送字符

程序编写

实验一:远程门铃

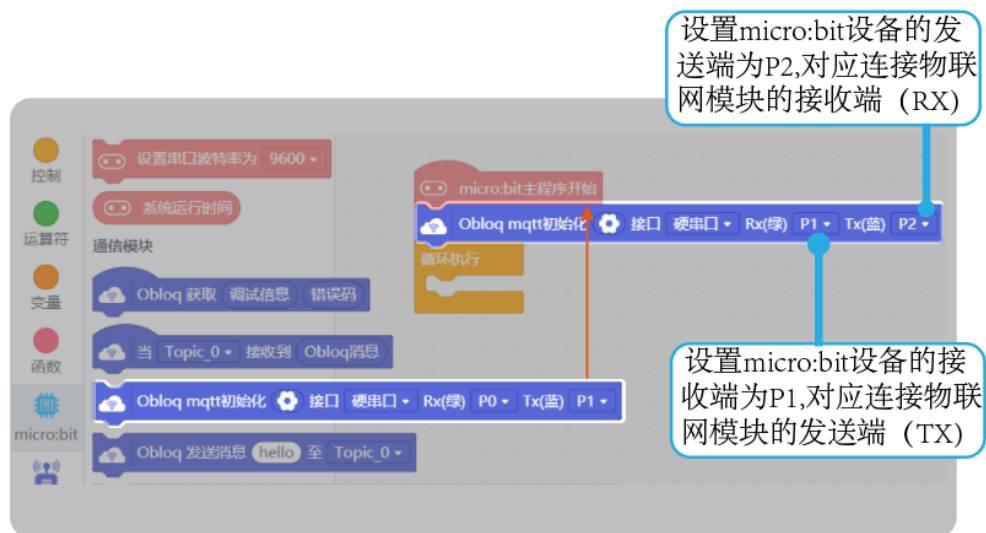
远程门铃的程序分为两个部分——发送端和接收端。首先我们先设计发送端的程序。

1.添加主控板和物联网通信模块

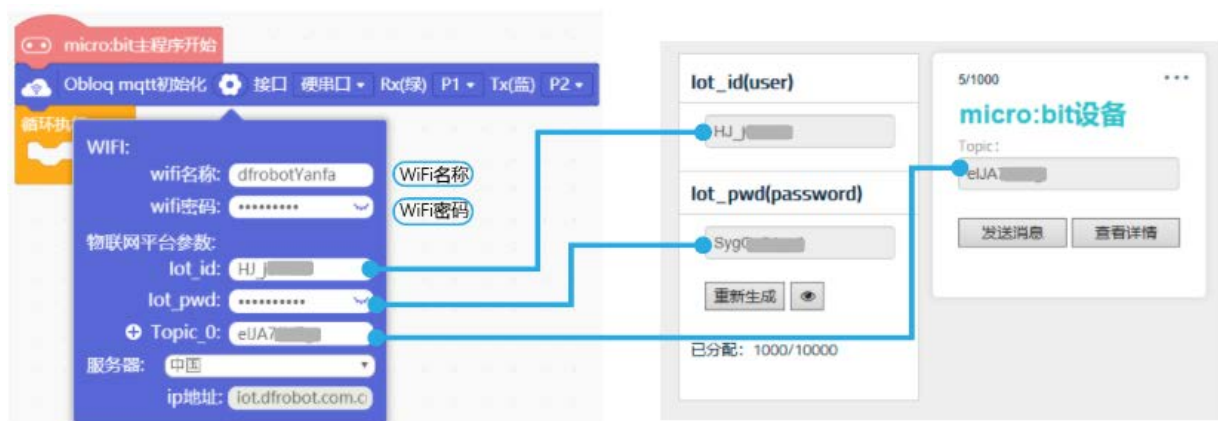
根据第一章相关内容添加主板——micro:bit 和物联网模块——OBLOQ。

2.物联网模块初始化及联网测试

- 将初始化模块 “” 拖放至右边的编辑区。然后定义发送和接收的引脚（要和实际连线一致）以及设置网页终端的相关账号与密码。



注：绿色的线接 P1 口（micro:bit 设备的接收端），蓝色的线接 P2 口（micro:bit 设备的发送端）。



设置 WiFi 密码和物联网平台参数

- 设置好物联网的相关信息后，将发送消息指令 “Obloq 发送消息 hello 至 Topic_0” 拖动到如图所示位置，帮助确认是否联网成功。



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。

最新消息	
时间	消息
2019/7/25 15:50:51	hello

3.设置无线通信频道

- 将 “” 和 “” 模块拖放到右边的编辑区。用来打开无线通信和设置无线通信频道。



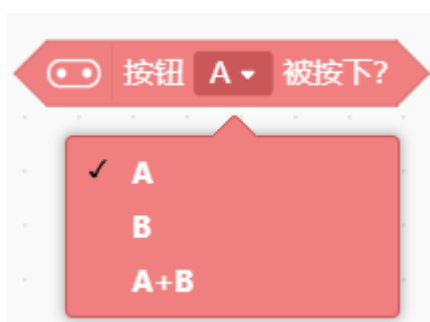
4.设置事件触发指令

- 设置事件触发指令，当按钮 A 被按下，显示主人的电话号码，无线发送指令 “A” 给主人端（接收段）的系统，并发送消息给 Easy IoT，提示主人有人按门铃。



提示：

事件触发指令——检测按钮被按下，点击下拉箭头可以选择三种按键方式：



前面第 5 章我们也学习了这种事件触发指令 “当 A 按钮按下” 这两者实现的结果都是一样。但是

他们之间的触发方式不一样，“当 A 按钮按下” 表示时间触发，用于管理中断触发；

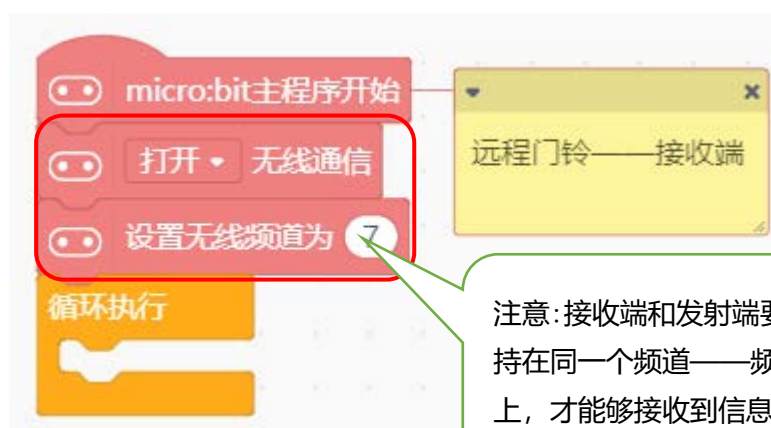
“按钮 A 被按下？” 表示查询触发，用于读取状态。

将发送端的程序命名并保存，并将它发送给连接有 OBLOQ 的主板上。接着我们新建项目，编写接收端的程序（即主人端的设备程序）。

5.设计接收端程序

- 新建项目，根据第一章相关内容添加主控板——micro:bit。

- 将 “打开 无线通信” 和 “设置无线频道为 7” 模块拖放到右边的编辑区。打开无线通信和设置无线频道。



- 将 “当接收到 无线数据” 拖放到右边的编辑区，用于判断是不是接收到无线数据。



- 使用条件判断指令，判断接收端接收到的指令是什么？如果接收到的数据信息为“A”，播放音效。



提示 1：在 micro:bit 模块的指令区没有该模块 “无线数据”，我们需要从模块

“当接收到 无线数据”中，将它拖出来，然后放在需要用到它的位置，如下图。



注意，这里不需要循环执行指令，因为只有判断无线接收到数据，该程序才会被执行，否则，不会执行，所以不需要循环执行。

参考程序 1

程序展示

将下面的程序分别下载到发射端和接收端的设备中。

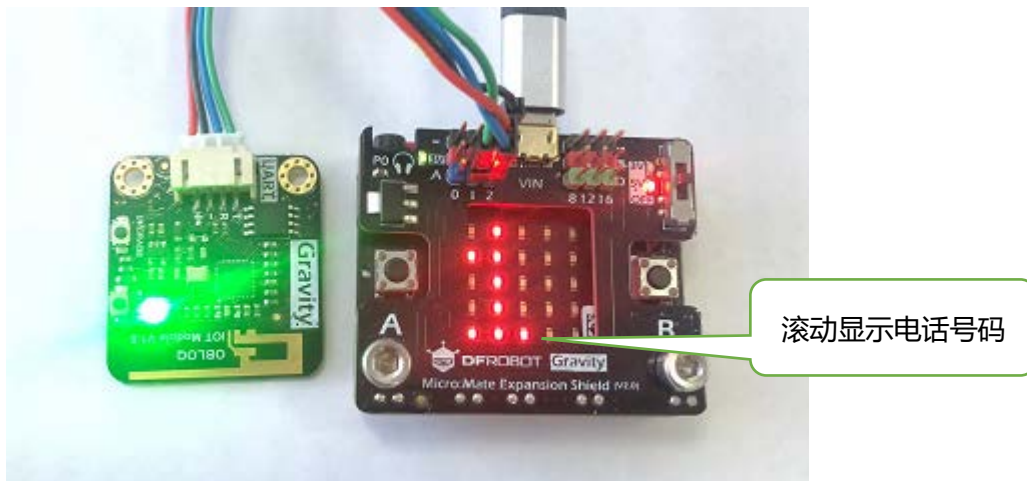


注意:将程序上传至设备后, 需要将连在主控上面的 USB 数据线拔下, 连接到扩展板上的电源接口 (VIN 口)。额外供电后, 将板载上的调节开关调至 5V。



效果展示

- 进入 Easy IoT 网页发送信息给 micro:bit 设备。找到相应的 Topic，点击“发送消息”。
如果我们按下发送设备的 A 键（连接 Obloq 的设备），点阵屏会显示电话号码，Easy IoT 也会收到提示语：有人按下门铃。



最新消息

时间	消息
2019/8/14 14:0:3	有人按门铃
2019/8/14 13:59:26	hello

- 如果接收设备接收到发送设备发出的信息，喇叭就开始响。提示主人有客来访，如果主人在家就可以去开门了。

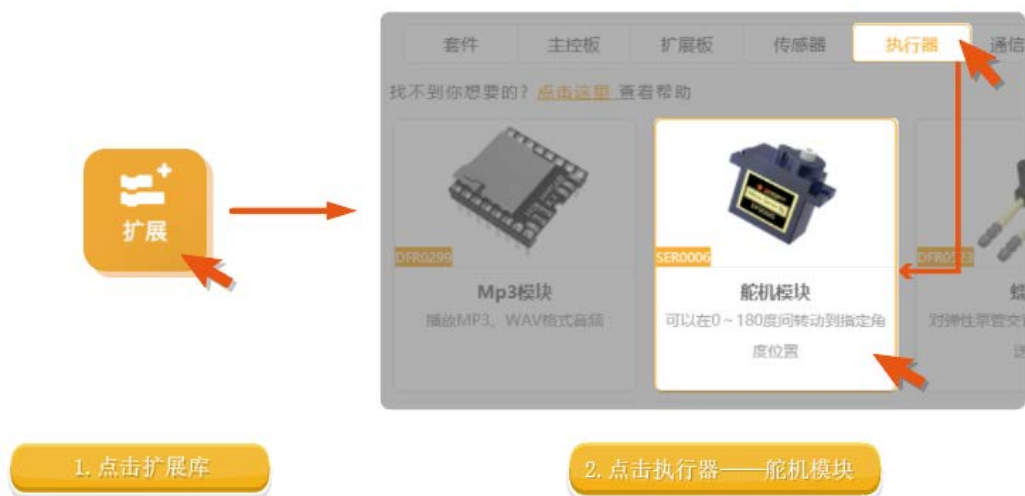


实验二:远程门禁

假如有一天你外出办事，突然朋友打电话说在你家门口。你又不能马上到家，于是你利用手机远程开锁，让他先进去等你。如果我们自己想制作一个可以远程控制的门锁，应该怎么做呢？其实很简单，只需要修改发送端的程序。

1. 添加模块

- 打开远程门铃的程序，添加执行器——舵机模块。



- 将标签修改成“远程门禁——发送端”。然后另存一个远程门禁的项目文件。



2、初始化舵机角度

- 因为我们利用舵机代表门锁，所以我们需要设置门锁的打开和关闭状态，接下来需设置舵机的初始化状态为关闭状态，假设舵机的起始角度为 0 度。

在“远程门铃”的基础上，将 “ 设置 P9 引脚伺服舵机为 90 度 ” 模块拖放至右边的编辑区，将引脚修改成 P8 脚，并设置舵机的初始化角度为 0 度。



3、发送控制信号

- 当我们接到朋友打来的电话后，通过 Easy IoT 平台发送指令“开”，控制舵机转动到 90 度，表示是开锁的状态。

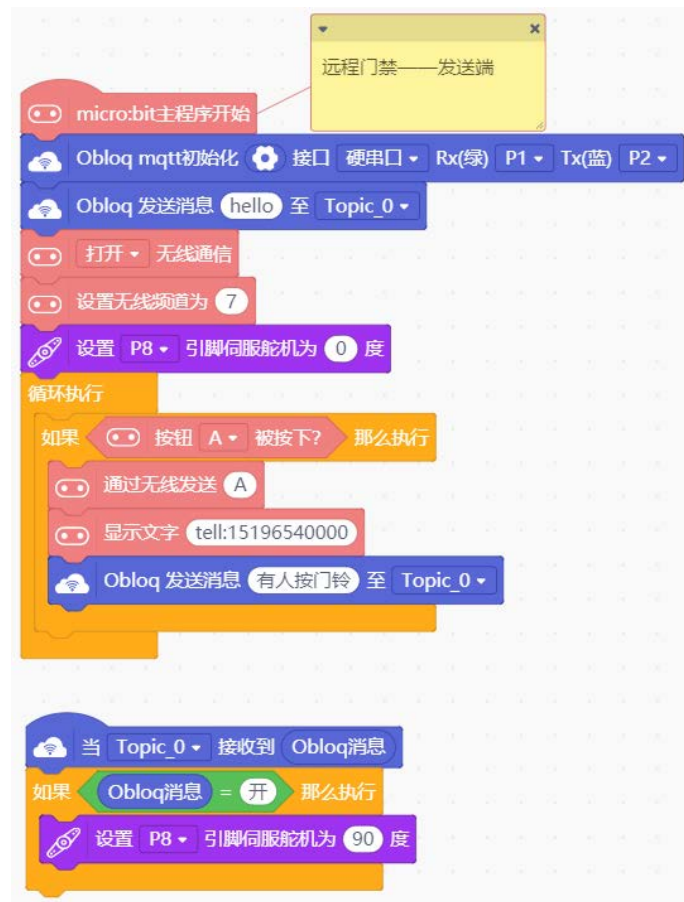


因为接收端的程序和远程门铃的一样，所以就不介绍它的编写过程了。

参考程序 2

程序展示

将下面的程序分别下载到发射端和接收端的设备中。

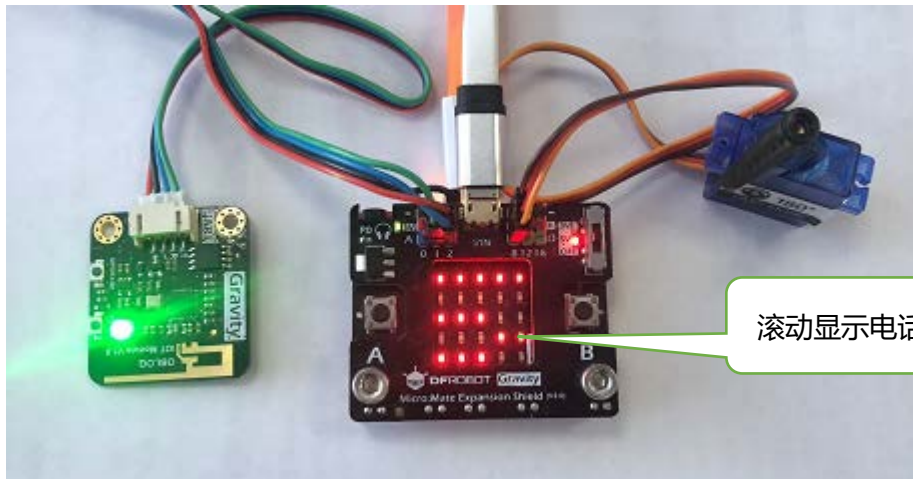


注意:将程序上传至设备后,需要将连在主控上面的 USB 数据线拔下,连接到扩展板上的电源接口(VIN 口)。额外供电后,将板载上的调节开关调至 5V。



效果展示

- 进入 Easy IoT 网页发送信息给 micro:bit 设备。找到相应的 Topic，点击“发送消息”。
如果我们按发送设备的 A 键（连接 Obloq 的设备），点阵屏会显示电话号码，Easy IoT 也会收到提示语：有人按下门铃。



最新消息

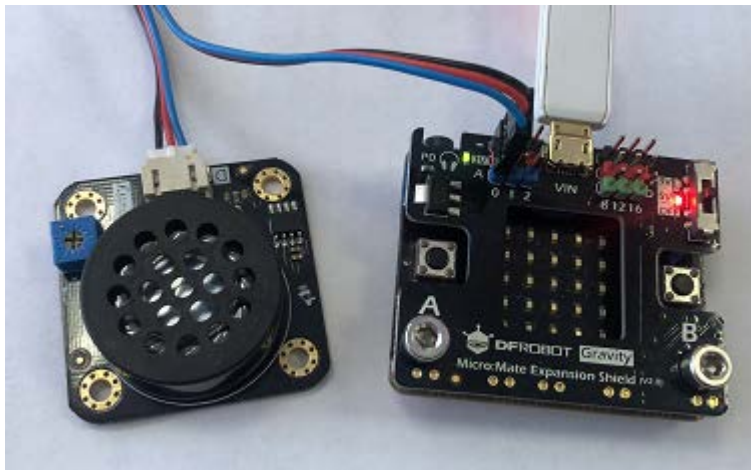
时间

消息

2019/8/14 17:41:8

有人按门铃

- 如果接收设备接收到发送设备发出的信息，喇叭就开始响。提示主人有客来访。



- 当你收到提示后，通过 Easy IoT 平台发送“开”的指令。



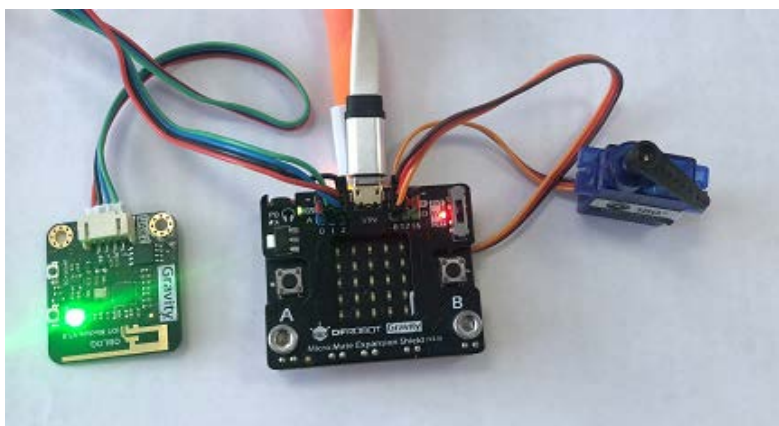
发送新消息

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"

开

发送

- 当远程门禁接收端接收到“开”的指令后，舵机转动到 90 度，实现开门。



拓展阅读

有线通信和无线通信的区别

有线通信需要通过实体导线进行数据传输，而无线通信是通过虚拟微波来进行数据传输，因此二者的主要区别是外观。

从发明技术来看，无线设备的发明，是建立在有线设备技术基础上的，因此无线通信设备较有线通信设备而言，技术含量更高、更先进；

从实际使用的角度来看，无线设备摆脱了实体介质导线的束缚，比有线设备更方便。目前在人们的日常工作和生活中，使用最为广泛的无线通信产品有：手机、无线电对讲机、无线鼠标、蓝牙耳机等。

在当前通信设备市场上，仍然还存在着众多有线通信产品，这是因为有线通信设备具有传输稳定、安全、快速等优点，并且传输的信息量更为庞大。比如，目前我们所使用的宽带网线，也有无线网络和有线网络两种。从当前的情况看，电子计算机连接有线网络的信号交换效率仍然比无线网络要高得多。电子计算机要实现无线网络传输，必须安装无线网卡，在运行网页或电脑游戏时，会产生很大的数据流量，通过有线网络进行数据传递要比无线网络的传递速度快得多，并且无线网络的信号也比有线网络稳定性更高。

无线通信怎么传播信号

小到手机、wifi，大到卫星通信，我们日常所见到的这些无线的通信形式都是通过无线电波来传播的。micro:bit 的无线通信原理就是基于无线通信方法中的无线电波。

无线电波（一种类似于可见光的电磁辐射波）具有某种属性（例如它们的幅度，相位或脉冲宽度），这些属性由发射机调制，使得信息可以被编码，从而可以携带信息进行传播，这是无线电信号的发射的过程。

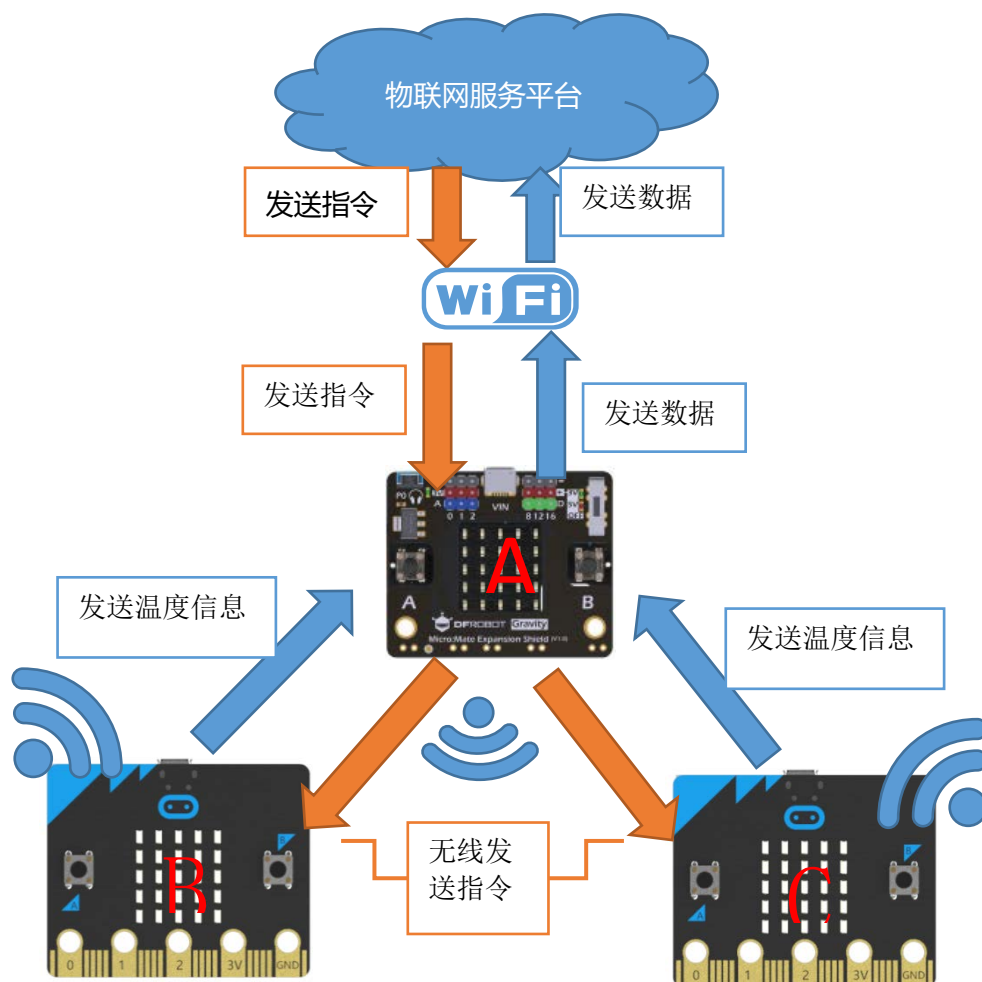
第八课 智能温控系统

以前，夏天回到家中，打开门迎来的是一股热气，这时就算你打开空调也不能马上让房间的温度降下来。现在，我们可以在回家前的 20 分钟，远程打开空调，回到家中迎接我们的将是一阵凉风。



学前提示

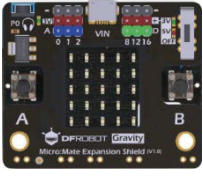
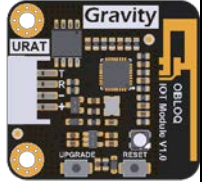
智能温控系统，采用的是多点采集温度，设备 B 与设备 C 通过无线通信将温度发送给设备 A，然后设备 A 将采集到的数据通过 WiFi 发送给物联网服务平台。



智能温控系统的项目结构分析图

硬件清单及连接

硬件清单

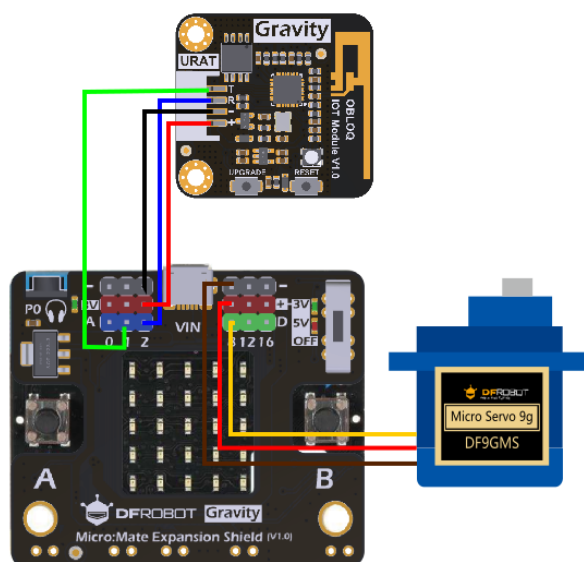
				
3 块 Micro:bit 主板 +1 块 Micro:mate 扩展板	OBLOQ 模块	电脑 (需要接入互联网)	USB 充电宝 (备选供电)	金属 9g 舵机 (舵机模块)

硬件连接

在项目中采用了多点采集温度，所以需要用到 3 块 micro:bit 主板和 1 块扩展板，其中 A 主控板连接 OBLOQ 模块和舵机为主设备，作为物联网模块数据接口端，也是局域网数据的接收和发送端。B，C 主控板（裸板）为从设备，为局域网中的相关测试节点，主要作用是收到主设备的无线数据后，发送反馈数据给主设备。

主设备电路连接

OBLOQ 模块中的 T 引脚（绿线）连接扩展板上的 P1 针脚（蓝色针脚），OBLOQ 模块中的 R 引脚（蓝线）连接扩展板上的 P2 针脚（蓝色针脚），OBLOQ 模块中的“-”引脚（黑线）连接扩展板上的黑色针脚，OBLOQ 模块中的“+”引脚（红线）连接扩展板上的红色针脚。舵机模块连接扩展板上的 P8 针脚，其中黄色的线连接绿色针脚，红色的线连接红色针脚，棕色的线连接黑色针脚。



主设备电路连接

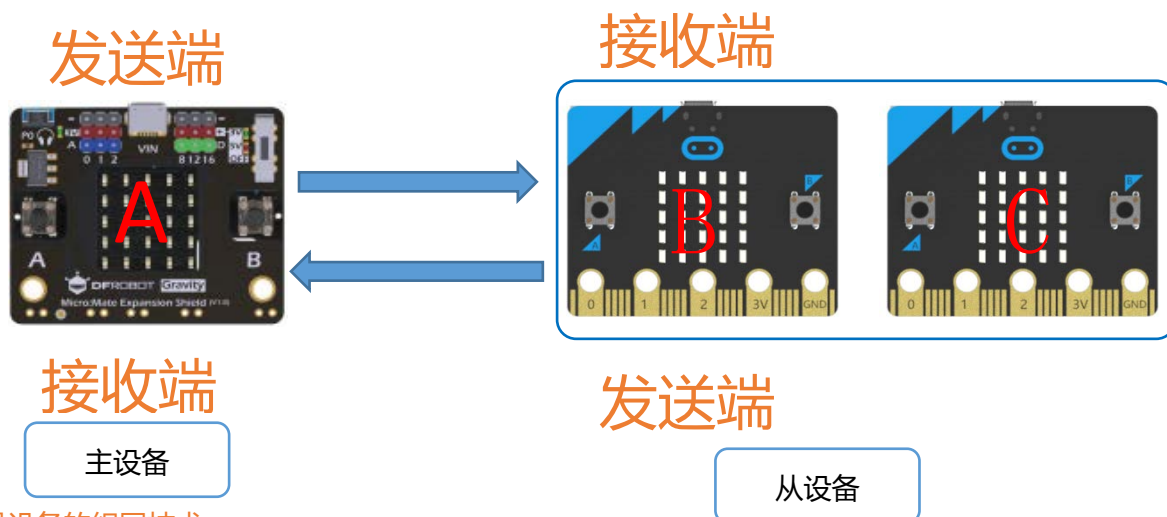
提示：从设备不需要连接硬件，所以准备两块裸板就行了。

知识点解析

什么是接收端（接收器）和发送端（发射器）

在通信中，通常有两个角色，发射和接收端，在第 7 课我们已经对两个设备之间进行了单工通信，当门铃被按下，发送消息给主人端的设备。此时，门铃相当于发送端。主人端相当于接收端。

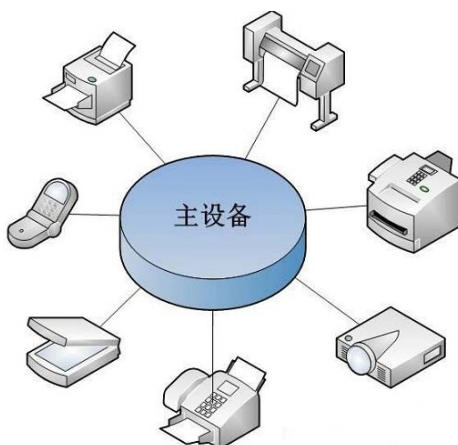
在本课的任务中，实现了双工通信，这个时候，主设备和从设备都在不停的扮演接收端和发送端的角色，也就是主设备和从设备都可以发送和接收信息。



什么是设备的组网技术

在第七课我们熟悉了无线通信中的一对一通信，即一个主设备对应一个从设备。但是有时候需要一个设备控制多个设备的情况，这时我们就需要考虑到组网技术了。

什么是组网技术？简单的理解，组网就是让各个设备之间能相互通讯的方式。比如你的一台手机连接上了你家的路由器，此时手机和路由器之间产生了数据传输，这就是组网，下面图中，是我们家里目前的状态，大部分设备都可以通过连接主设备（路由器）而可以上网，然后主设备去进行外界的信息交互。



局域网和广域网

上面说了“组网”概念中的“组”即设备连接的方式；接下来我们就了解一下“网”的概念，“网”大

致可以分为局域网和广域网，比如你家的三台手机和路由器之间就组成了一个局域网，那么这四台设备就可以互相通信了，局域网的优点是，在设备之间传输一些巨大的数据时，会非常快捷，通常用在企业或一些小组织内部；缺点是，只在局域网内通讯的应用范围较窄。本课的智能温控系统采用的就是局域网。

至于广域网，是一种跨地区的数据通讯网络。比如你家的路由器或者网关，通过上级设备，与其他省地区的设备产生了通讯，这就可以理解成一个广域网。通常跨接很大的物理范围，所覆盖的范围从几十公里到几千公里，它能连接多个地区、城市和国家，或横跨几个洲并能提供远距离通信，形成国际性的远程网络。一般所指的互联网是属于一种公共型的广域网。

但其实局域网与广域网之间并没有严格的区分界限。比如你可以把整个中国的网络当成一个局域网。只不过它比较大而已。所有在中国的联网设备，都可以在这个局域网内进行通讯。



程序编写

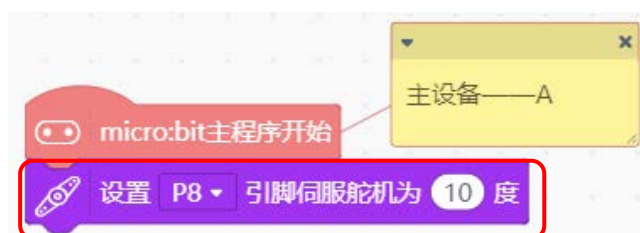
智能温控系统的程序分为三个部分——主设备 A 和从设备 B、C 的程序。首先我们先设计主设备 A 的程序。

1. 添加主控板及物联网模块

根据第一章相关内容添加主板——micro:bit 和物联网模块——OBLOQ。

2. 添加执行器——舵机及初始化舵机角度

- 根据前面学习到的内容添加执行器——舵机。
- 因为舵机表示空调开关，所以我们需要设置开关的打开和关闭状态，接下来需设置舵机的初始化状态为关闭状态，设舵机的起始角度为 10 度。



3. 物联网模块初始化及联网测试

- 将初始化模块 “Obloq mqtt 初始化 接口 硬串口 Rx(绿) P0 Tx(蓝) P1” 拖放至右边的编辑区。根据前面学习的内

容定义发送和接收的引脚（要和实际连线一致）以及设置网页终端的相关账号与密码。

- 设置好物联网的相关信息后，将发送消息指令 “Obloq 发送消息 hello 至 Topic_0” 拖动到如图所示位置，帮助确认是否联网成功。



注：为了避免收发信息有冲突，将接口选为软串口比较好。

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。



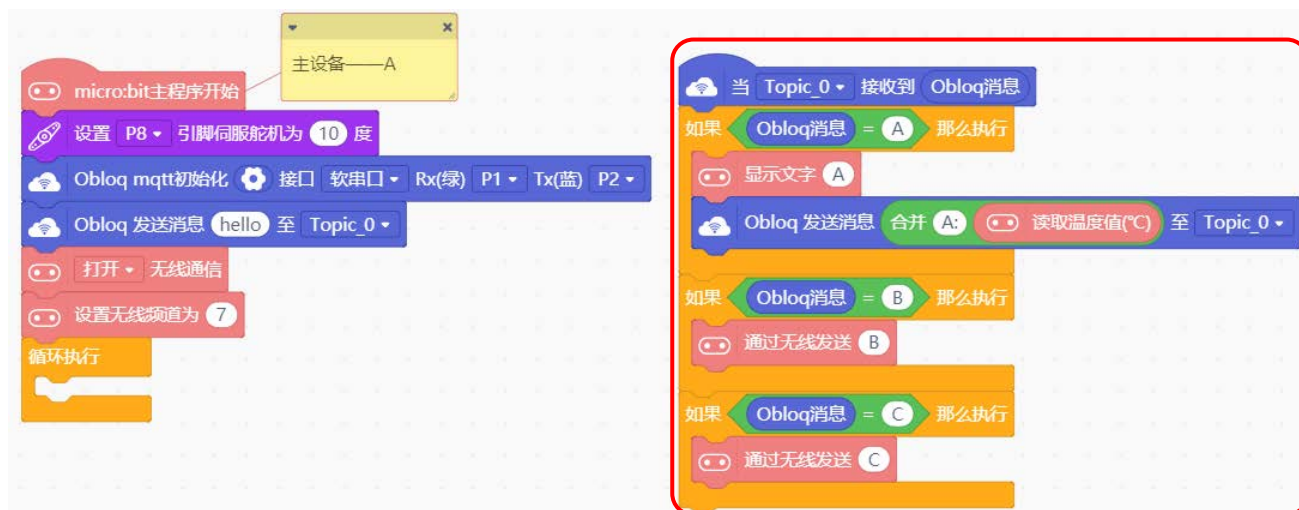
4.设置无线通信频道

- 将 “打开 无线通信” 和 “设置无线频道为 7” 模块拖放到右边的编辑区。用来打开无线通信和设置无线通信频道。



5.设置触发指令

- 设置事件触发指令，通过 Easy IoT 平台发送指令 A，设备 A 显示文字 A，并获取设备 A 的温度数据。Easy IoT 平台发送指令 B，通过无线发送数据 “B” 给设备 B，当 Easy IoT 平台发送指令 C，通过无线发送数据 “C” 给设备 C。



- 当我们将指令发送给设备 B, C 后，设备 A 如何采集它们的数据呢？这时就需要用到模块

“”。当设备 A 接收到设备 B,C 的无线数据后，将收到的信息发送给 Easy IoT 平台。



- 当我们收到了温度数据后，通过 Easy IoT 平台发送指令打开或者关闭空调。发送指令 “开”，控制空调打开（即舵机转动到 100 度）；发送指令 “关”，控制空调关闭（即舵机转动到 10 度）。



6.设计设备 B,C 端的程序

- 新建项目，根据第一章相关内容添加主控板——micro:bit。

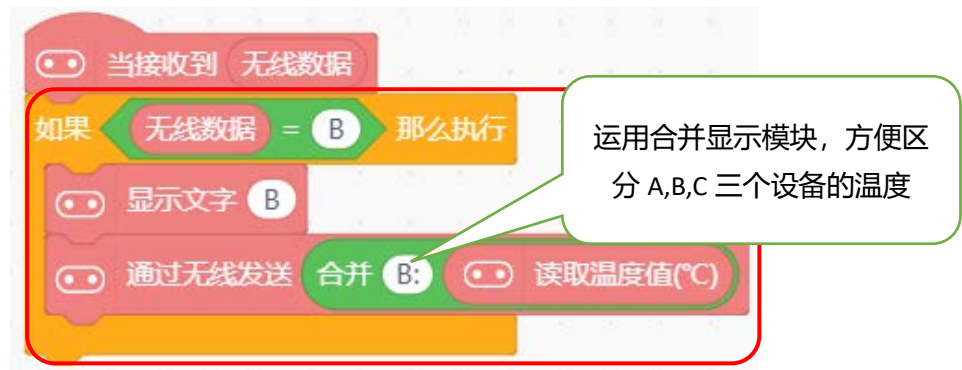
- 将 “打开 无线通信” 和 “设置无线频道为 7” 模块拖放到右边的编辑区。打开无线通信和设置无线频道。



- 将 “当接收到 无线数据” 拖放到右边的编辑区，用于判断是不是接收到无线数据。



- 使用条件判断指令，判断接收到的指令是什么？如果接收到的数据信息为“B”，主板显示文字“B”，并通过无线发送该设备的当前温度。



- 做如下图所示的修改，设备 C 的程序就完成了。

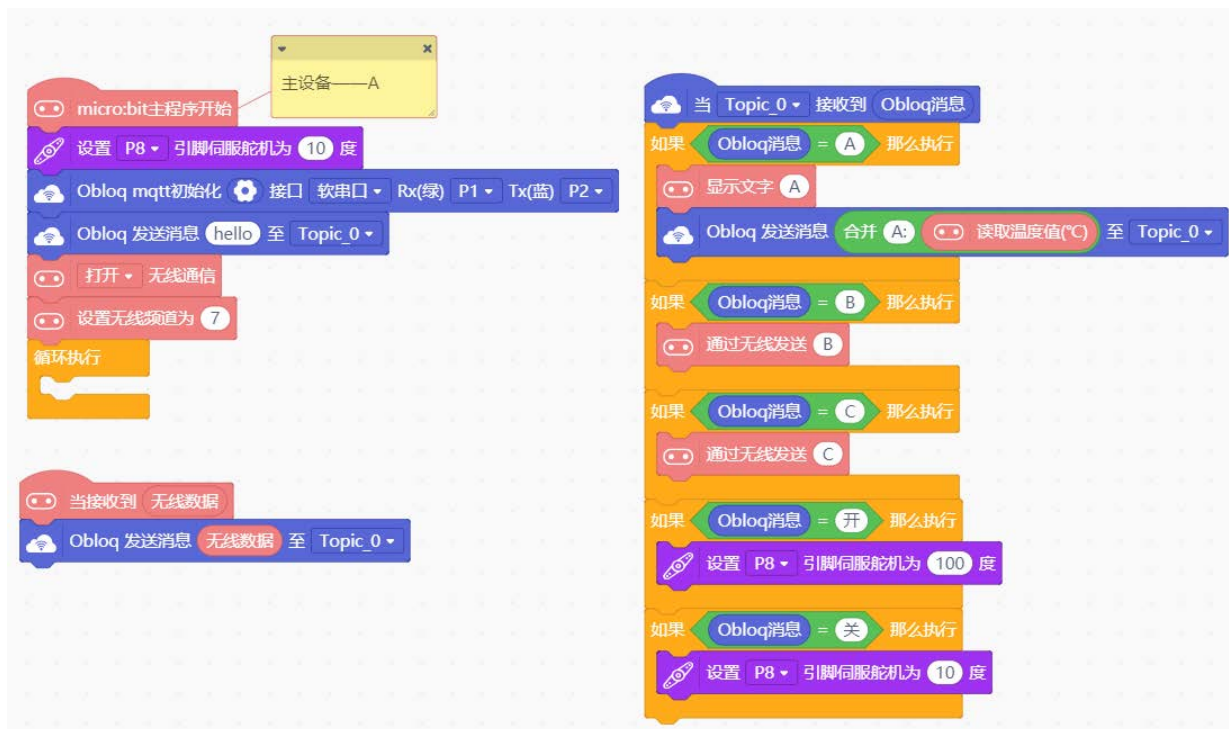


参考程序

程序展示

将下面的程序分别下载到设备 A、设备 B 和设备 C 中。

设备 A 的程序：



注意:将程序上传至设备后, 需要将连在主控上面的 USB 数据线拔下, 连接到扩展板上的电源接口 (VIN 口)。额外供电后, 将板载上的调节开关调至 5V。

设备 B 的程序：



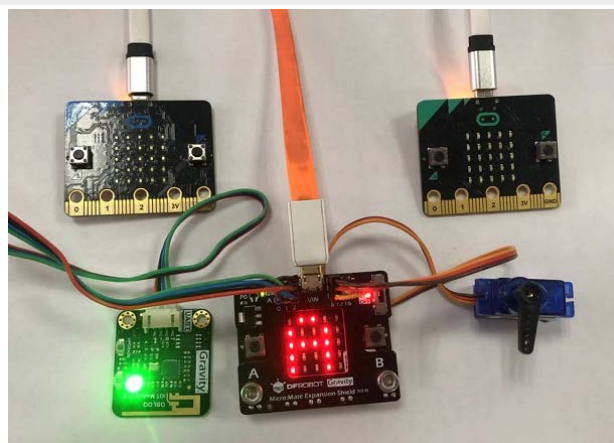
设备 C 的程序:



效果展示

- 进入 Easy IoT 网页发送信息给 micro:bit 设备。找到相应的 Topic, 点击 “发送消息” ,发送指令 “A”

最新消息	
时间	消息
2019/8/20 17:9:2	A



- 刷新网页可以查看设备 A 的温度。

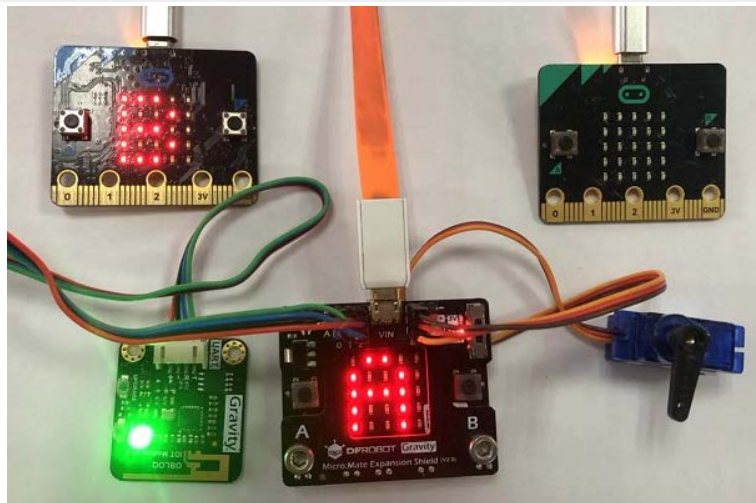
最新消息

时间	消息
2019/8/20 17:9:2	A:42
2019/8/20 17:9:2	A

- 发送指令 “B”，获取设备 B 的温度数据。

最新消息

时间	消息
2019/8/20 17:9:45	B



- 刷新网页，查看设备 B 的温度数据。

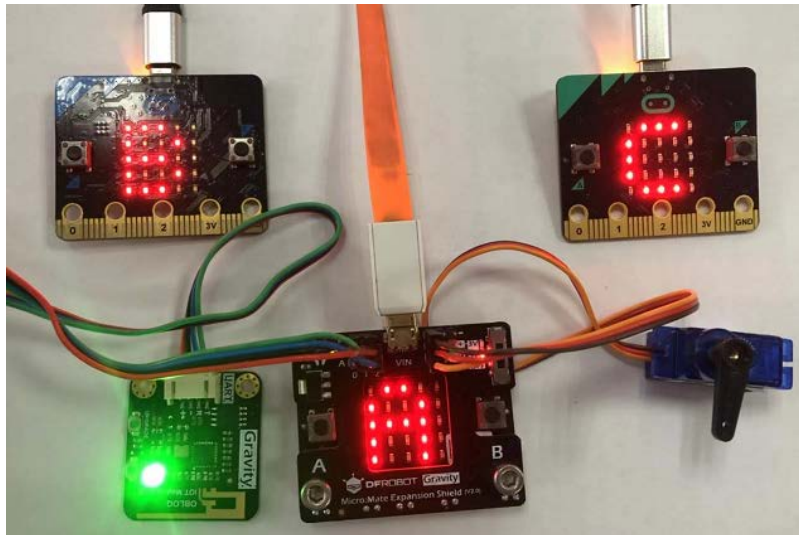
最新消息

时间	消息
2019/8/20 17:9:45	B:33
2019/8/20 17:9:45	B

- 发送指令 “C”，获取设备 C 的温度数据。

最新消息

时间	消息
2019/8/20 17:12:51	C



- 刷新网页，查看设备 C 的温度数据。



最新消息

时间	消息
2019/8/20 17:12:51	C:36
2019/8/20 17:12:51	C

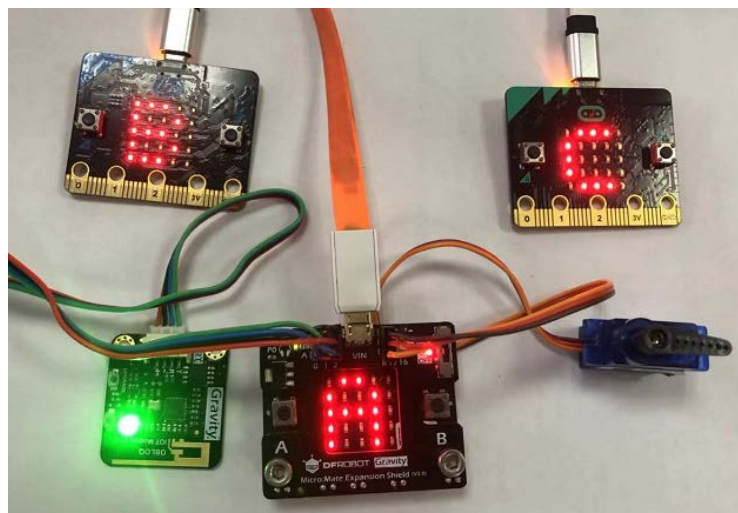
- 通过观察设备检测到的温度，发现家里面的温度已经这么高了，是时候打开空调了。通过 Easy IoT 发送“开”的指令。



最新消息

时间	消息
2019/8/20 17:24:46	开

- 当设备 A 接收到“开”的指令后，舵机转动到 100 度，实现开空调的功能。



- 如果我们想关掉空调，通过 Easy IoT 发送指令“关”。舵机转动到 10 度，又回到初始化状态。

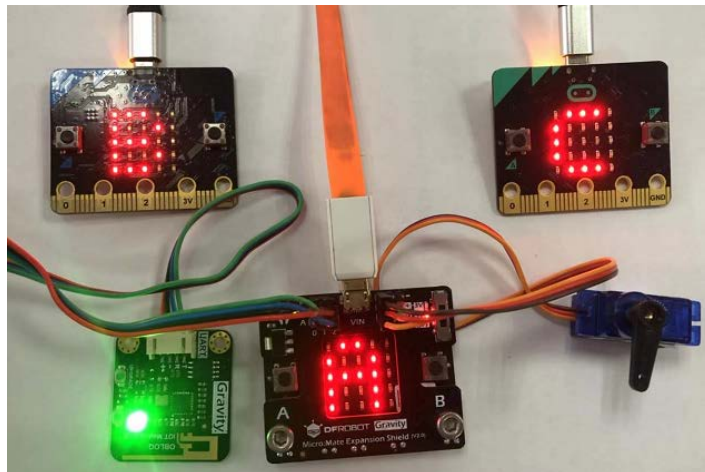


最新消息

时间

消息

2019/8/20 17:34:58 关



拓展阅读

通信方式

从不同的角度考虑，常用的通信工作方式可以分为以下几种。

1. 消息传送的方向和时间

对于点与点之间的通信，按照消息传送的方向与时间关系，通信方式可分为单工通信、半双工通信及全双工通信三种。



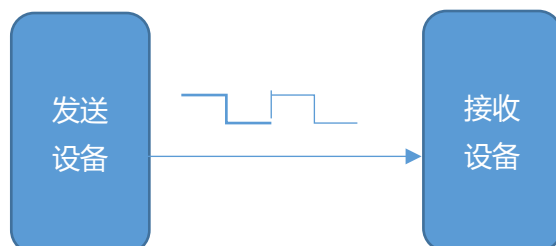
单工通信，是指消息只能单方向进行传输的一种通信工作方式，单工通信的例子有很多，比如广播，遥控，无线寻呼等。在这些产品上，消息只从广播发射台，遥控器，无线寻呼中心分别传到收音机，遥控对象和 BP 机上。

半双工通信，是指双方都能收发消息，但不能同时进行收和发的工作方式。对讲机、收发报机等都是这种通信方式。

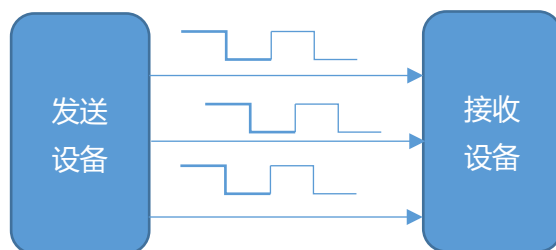
全双工通信，是指通信双方可同时进行双向传输消息的工作方式，在这种方式下，双方都可以同时进行收发消息。它们通信的信道是双向信道。生活中全双工的例子非常多，比如座机，手机等。

2. 数字信号排序方式

在数字通信中，按照数字信号代码排列顺序的方式不同，可以将通信方式分为串序传输和并序传输。串序传输，是将代表信息的数字信号序列按时间顺序一个接着一个的在信道中传输。



如果将代表信息的数字信号序列分割成两路或两路以上的数字信号序列同时在信道上传输，则称为并序传输通信方式。



一般的数字通信方式大都采用串序传输，这种方式只需要占用一条通路，缺点就是时间长。并序传输通道的优点就是传输时间短，但是它占用多条通路。

3. 通信网络形式

通信的网络形式可分为三种：两点直通方式，分支方式和交换方式。

直通方式是通信网络中最简单的一种；终端 A 和终端 B 之间的线路是专用的；在分支方式中，它的每个终端 (A,B,C,D.....,N) 经过同一信道与转接站相连接，此时，终端之间不能直接通信，需要经过转接站。交换方式是终端之间通过交换设备进行线路交换的一种方式，把要求通信的两个终端之间的线路接通（自动），或者通过程序控制实现消息交换。

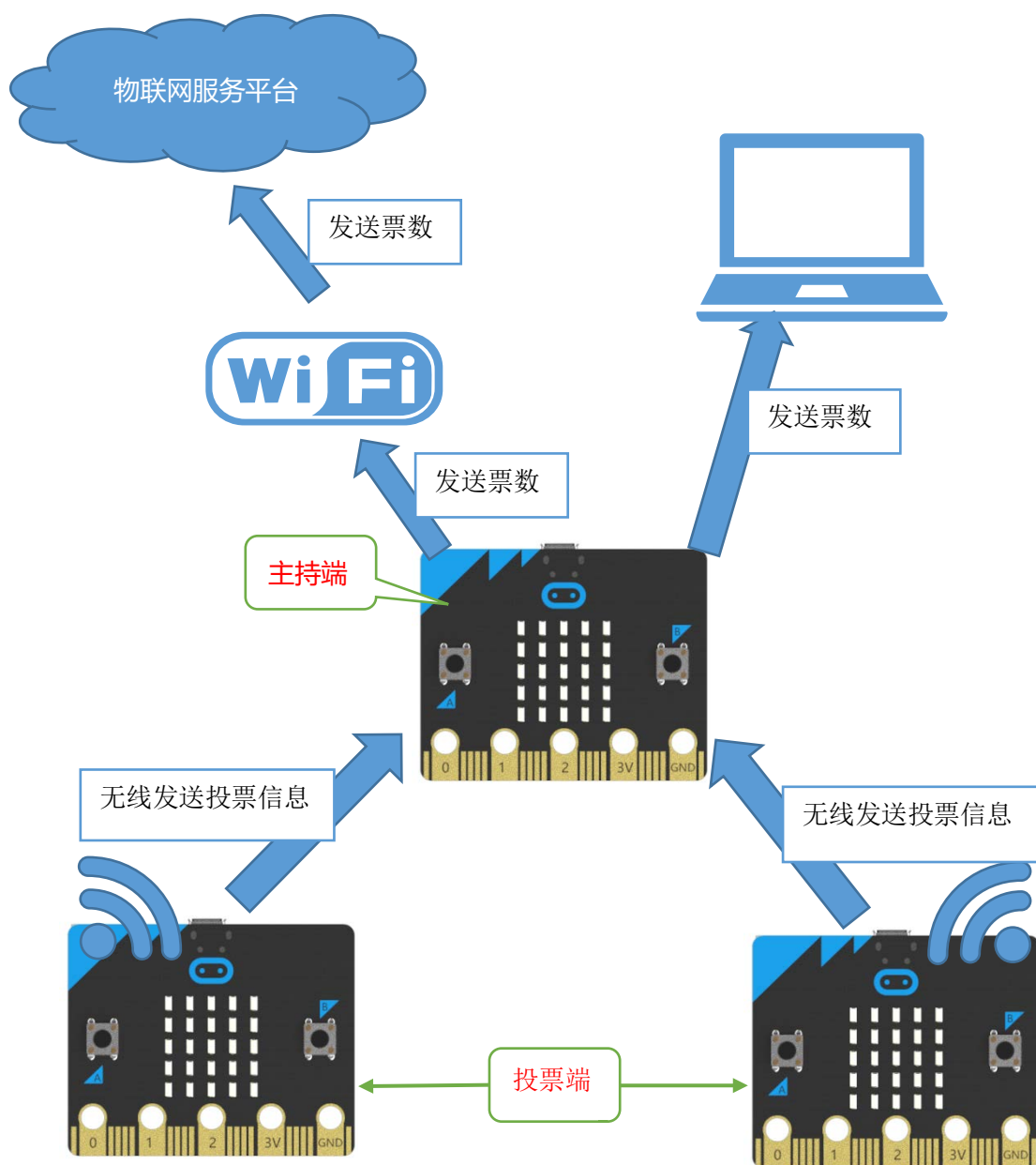
以上大致介绍了三种通信方式，对于其他方式感兴趣的同学，可以在网上查询更多的资料学习

第九课 竞选投票器

在家看比赛类的节目时，我们都会听见主持人说：“如果你支持红队，就请按红键，如果你支持蓝队，就请按蓝键进行投票”。现在随着投票器越来越多的被运用在会议、培训、年会活动，人们也享受到了观众互动投票带来的乐趣和便利：投票数据实时获取、节省会议时间、让现场互动变得容易了观众的参与度也提高了。那接下来我们就来制作一个投票器吧。

学前提示

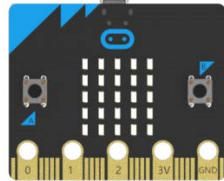
竞选投票器主要分为主持端和投票端（观众端），投票端（观众端）可以是无数个设备。其功能在使用时，投票者只需要按下相应的按键即可，相应的票数会在电脑屏幕上实时显示，同时，通过 WiFi 将票数发送给物联网平台。这样双方记录数据，保证数据的准确性。



竞选投票器的项目结构分析图

硬件清单及连接

硬件清单

		
3 块 Micro:bit 主板	电脑 (需要接入互联网)	USB 充电宝 (备选供电)

在项目中，用到了 3 块 micro:bit 主板，一块作为主持端，另外两块做投票端。

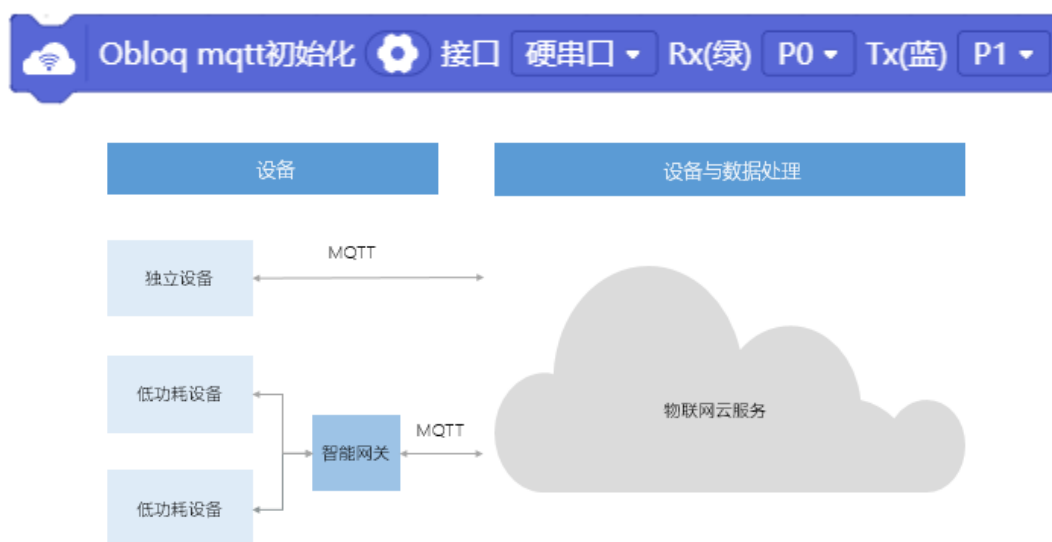
知识点解析

MQTT 的工作方式

在第五章的物联网服务中，我们了解了前段部分主要是用来收发数据，但是因为它要收发不同的数据，就需要一种可以互通的“语言”来进行交流，那就是通讯协议。

通信协议是指双方实体完成通信或服务所必须遵循的规则和约定。那么物联网都有哪些通信协议？目前在用的物联网七大通信协议有 MQTT、DDS、AMQP、XMPP、JMS、REST、CoAP。

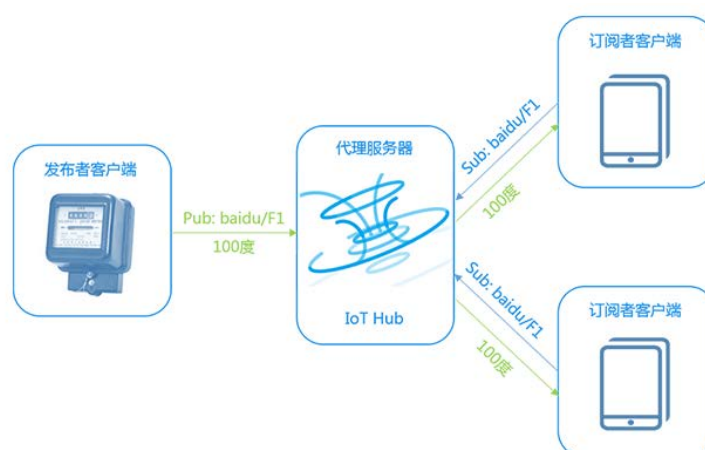
细心的同学，可能已经知道我们所用的协议是 MQTT 协议。通过 MQTT 协议，将终端设备的信息传送给服务器。



MQTT (Message Queuing Telemetry Transport)，是由 IBM 开发的即时通讯协议，相比来说比较适合物联网场景的通讯协议。

MQTT 是一种能实现一对多通信的协议（采用发布/订阅模式），所有的物联网终端都连接到云端，云端通过主题（Topic）的方式管理各个设备关注的通讯内容，负责将设备与设备之间消息的转发。

MQTT 有三个角色，包括代理服务器、订阅者和发布者。



代理服务器承担着转发 MQTT 通信的服务器的作用，发布者和订阅者起着客户端的作用，发布者是发送消息的客户端，订阅者是负责接收消息的客户端。MQTT 交换消息都是附带主题(Topic)地址的，客户端可以把这个“主题”视为收信件的地址，对其执行消息传输。

根据上图我们再来看一下 MQTT 的通信机制，首先代理服务器等待各个客户端的连接，订阅者连接了代理服务器，把自己订阅的主题名称 (baidu/F1) 告诉代理服务器，这个过程就叫订阅。然后发布者连接代理服务器，以主题 (baidu/F1) 为收信地址发送消息，这个过程就是发布。发布者和订阅者可以是多个。

MQTT 协议一般适用于设备数据采集到端(Device→Server, Device→Gateway)，集中星型网络架构(hub-and-spoke)，不适用设备与设备之间通信，设备控制能力弱，另外实时性较差，一般都在秒级。

指令提前知

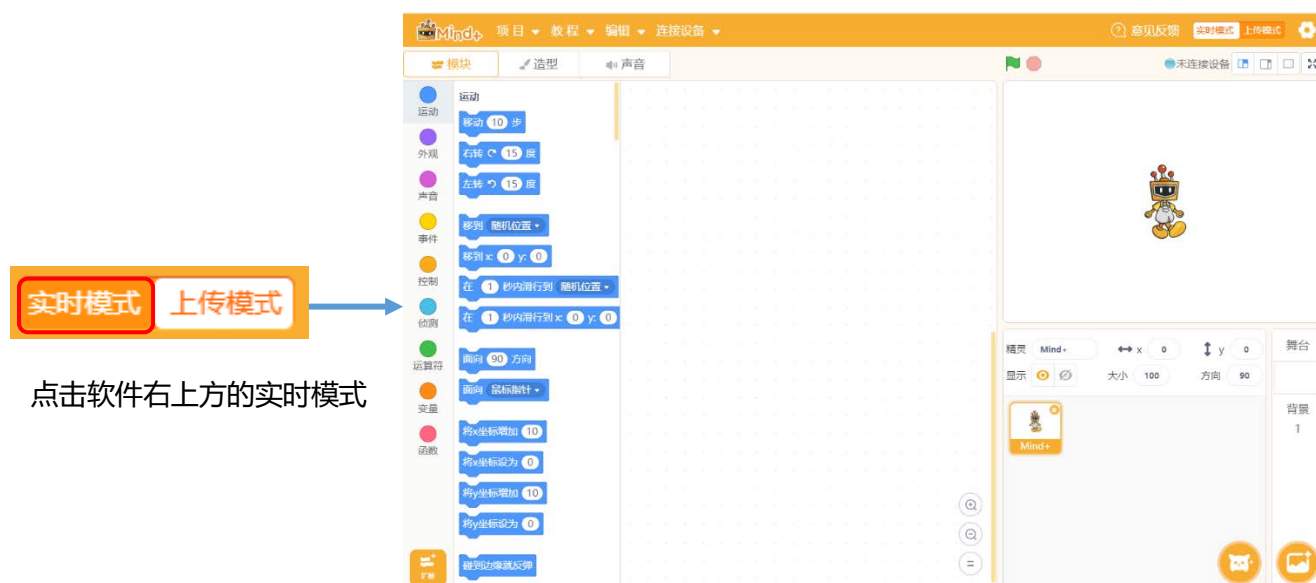
模块	功能说明
	初始化参数，添加 IoT 地址，订阅 topic
	MQTT 发起连接
	用于判断 MQTT 是否连接成功
	发送消息给 Topic
	让角色进行语言表达

程序编写

竞选投票器的程序分两个部分——主持人端和投票端（观众端）。首先我们先设计主持人端的程序

1.切换界面

将 Mind+ 界面切换到实时模式，具体操作如下图。

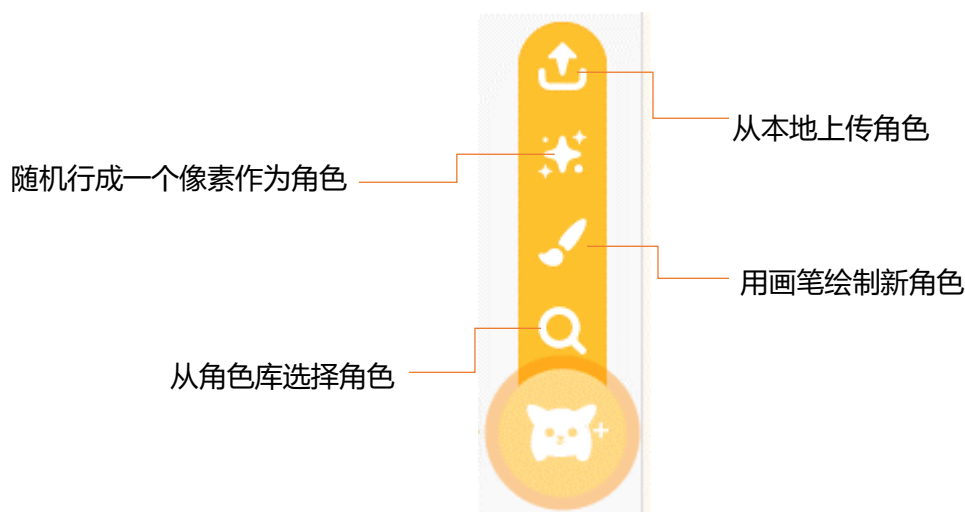


点击软件右上方的实时模式

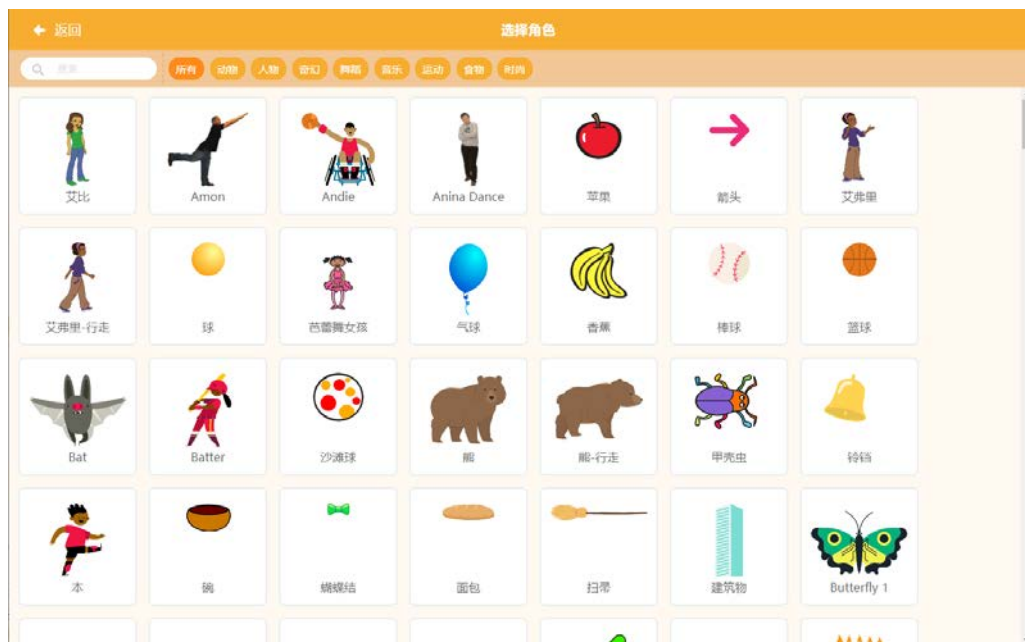
切换成实时模式界面

2.添加角色

- 新建项目时，角色区会有一位 Mind+ 先生作为默认角色，当然我们也可以添加其他的角色。接下来我们一起来学习一下吧。首先认识一下添加角色的工具，如下图所示。



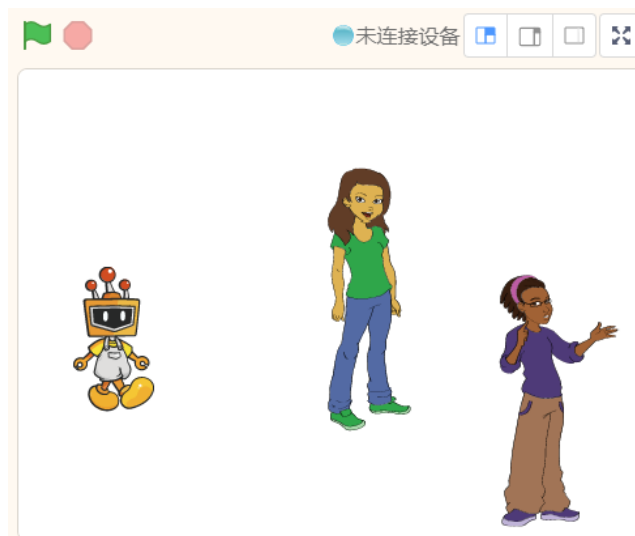
- 打开角色库，根据剧情的需要选择不同的角色，如下图所示。



- 在本项目中, 我们需要另外再添加两个角色——艾比和艾弗里。当然, 同学们也可以选择其他的角色。添加方法如下图所示。



- 用同样的方法添加艾弗里, 最后舞台区呈现的效果如下图所示。



- 有时候为了适应故事情节的发展，角色导入舞台后需要调节大小、编辑造型。比如艾比这个角色，她一共有 4 款默认造型，我们可以根据作品需求，选择合适的造型。



2.设置角色的触发事件

前面我们添加了 3 个角色，他们分别扮演的是主持人（Mind+先生）、选手（艾比和艾弗里）。接下来我们依次来设计他们相应的程序脚本吧！首先设计主持人 Mind+先生的程序。

- 点击“扩展”，选择主控板——micro:bit。
- 点击“扩展”，选择网络服务，添加 MQTT 协议。



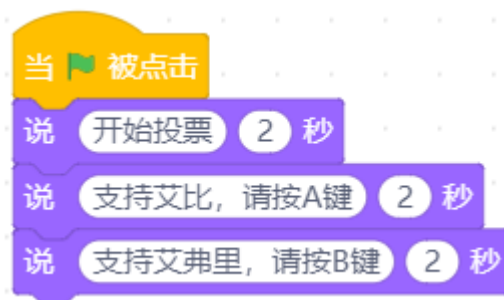
- 单击“Mind+先生”这个角色，在编辑区为该角色编写程序。



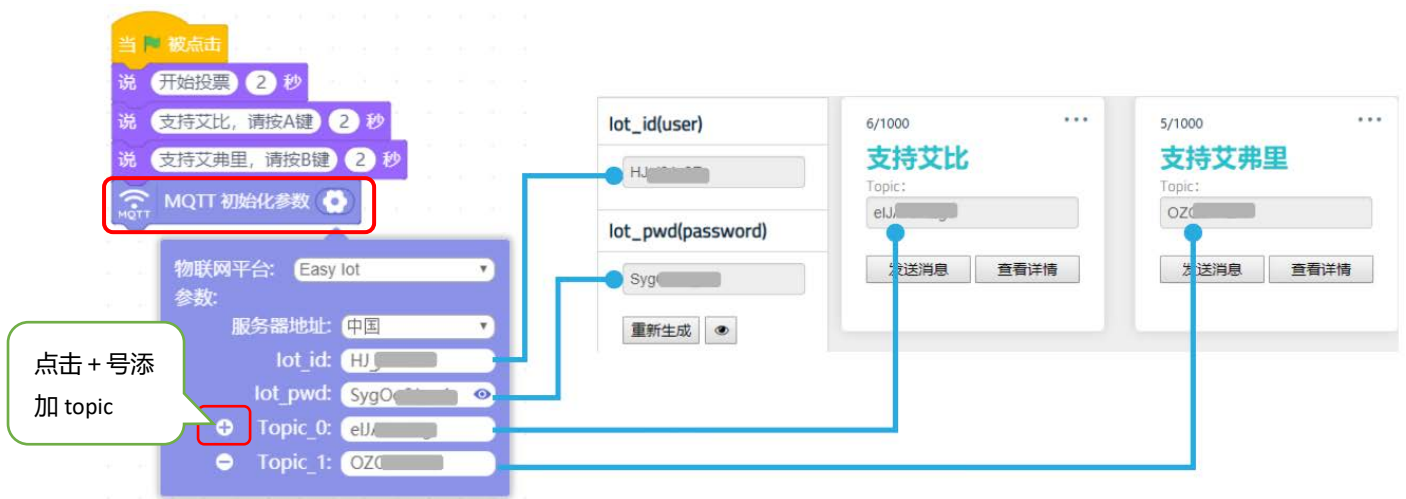
- 将“当被点击”模块拖动到右边的编辑区，用来检测其他模块的事件操作，主要作为启动程序的基础。



- 将“说 Hello! 2 秒”模块拖动到右边的编辑区，用来设置 Mind+ 的语言表达。



- 将“MQTT 初始化参数”模块拖动至右边的编辑区，点击设置“MQTT”图标，将 Easy IoT 中获取的 IoT 账号、密码和新设备 Topic 填写到对应位置（需要两个 Topic），完成 MQTT 参数初始化。



- 将 “MQTT 发起连接” 模块拖动至右边的编辑区，MQTT 发起连接。然后将事件模块里的 “等待直到” 模块拖动到右边的编辑区，再将 “MQTT 连接成功?” 拖放进直到等待的模块中，等待 MQTT 连接成功。



- 当 MQTT 连接成功后，打开无线通信，设置无线通信的频道为 7。



- 接着我们新建两个变量“艾比”和“艾弗里”，用来表示两位选手的得票数。设置它们的初始值为 0。



- 怎样才能知道 MQTT 是否联网成功呢？其实和前面学到的方法一样，给 Easy IoT 发送一条消息，比如发送“hello”。



- 因为我们采用的无线通信的功能，将观众手里投票设备与主持端的设备相连，主持端采集投票端的数据。所以需要将“当接收到 无线数据”模块拖动到右边的编辑区。用于接收投票端的反馈。
- 当主持端接收到的无线数据为“button_A”，表示支持“艾比”，当接收到的无线数据为“button_B”，表示支持“艾弗里”。所以程序设计如下：

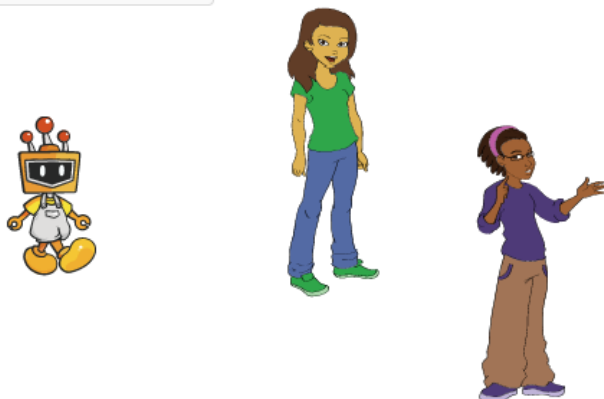
当观众按下 A 键，艾比的票数增加 1

当观众按下 B 键，艾弗里的票数增加 1



✓ 变量 支持艾弗里

提示：如果我们想在舞台区看见票数的变化，需要在变量里勾选 “☒ 变量 支持艾比 ” 这样就能在舞台区观察票数的变化了。如下图：



- 舞台区可以查看票数了，那接下来我们要做的就是将双方的得票数上传至 Easy IoT 平台。将艾比的票数上传到 Topic0，将艾弗里的票数上传到 Topic1。



提示：为了方便区分两个 Topic，我们可以将 Topic 的标题修改成“支持艾比”和“支持艾弗里”。如下图操作。



点击红框中的“铅笔”修改名称

修改名称后的 Topic0 和 Topic1

主持人 Mind+先生的程序已经设计完成了，接下来就是艾比和艾弗里。一般在投票前，选手都会自我介绍。

- 点击艾比这个角色，根据下图的内容在编辑区设计艾比的自我介绍。



- 点击艾弗里这个角色，根据下图的内容在编辑区设计艾弗里的自我介绍。



主持人端的程序设计好了，接下来我们来设计投票端（观众端）的程序。

1.新建项目

- 将 Mind+界面切换到上传模式，具体操作如下图。



切换成上传模式界面

- 点击扩展，添加主控板——micro:bit。



2.程序编写

- 将 “打开 无线通信” 和 “设置无线频道为 7” 模块拖动至右边的编辑区，打开无线通信并设置无线频道为 7。



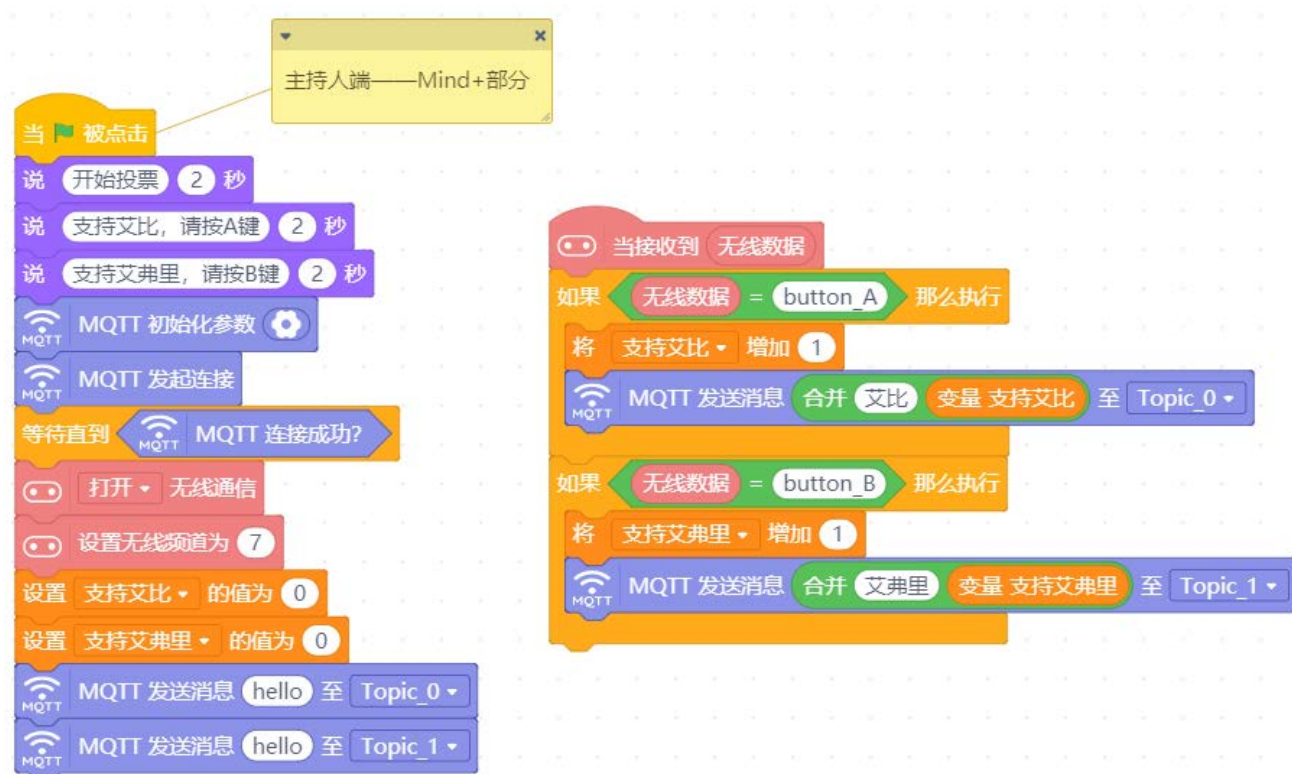
- 接下来设置事件触发，当按下 A 键的时候，无线发送 “button_A” 指令给主持端的设备；当按下 B 键的时候，无线发送 “button_B” 指令给主持端的设备。投票端的 LED 点阵屏分别显示 A,B。



参考程序

点击小绿旗，运行主持人端的程序。

主持人端的程序：



将下面的程序下载到投票端的 micro:bit 主板上，投票端可以是多个设备。

投票端的程序：

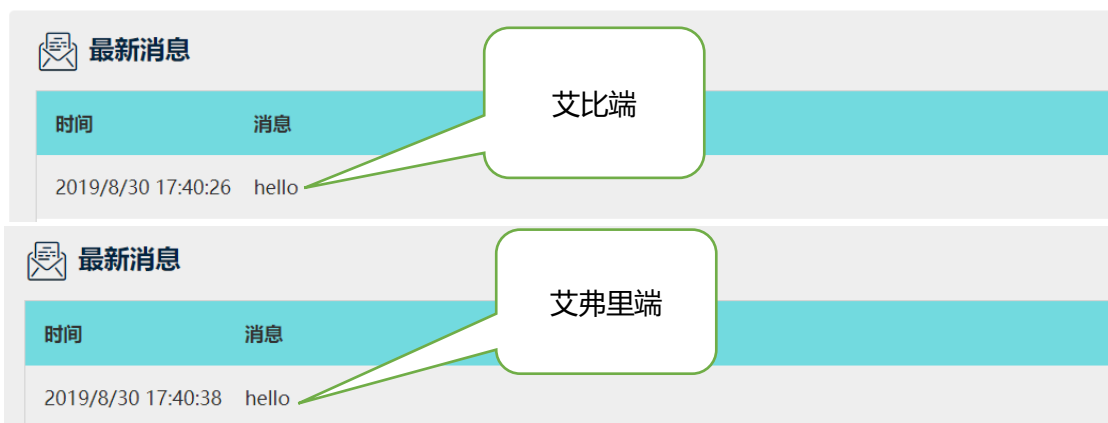


效果展示

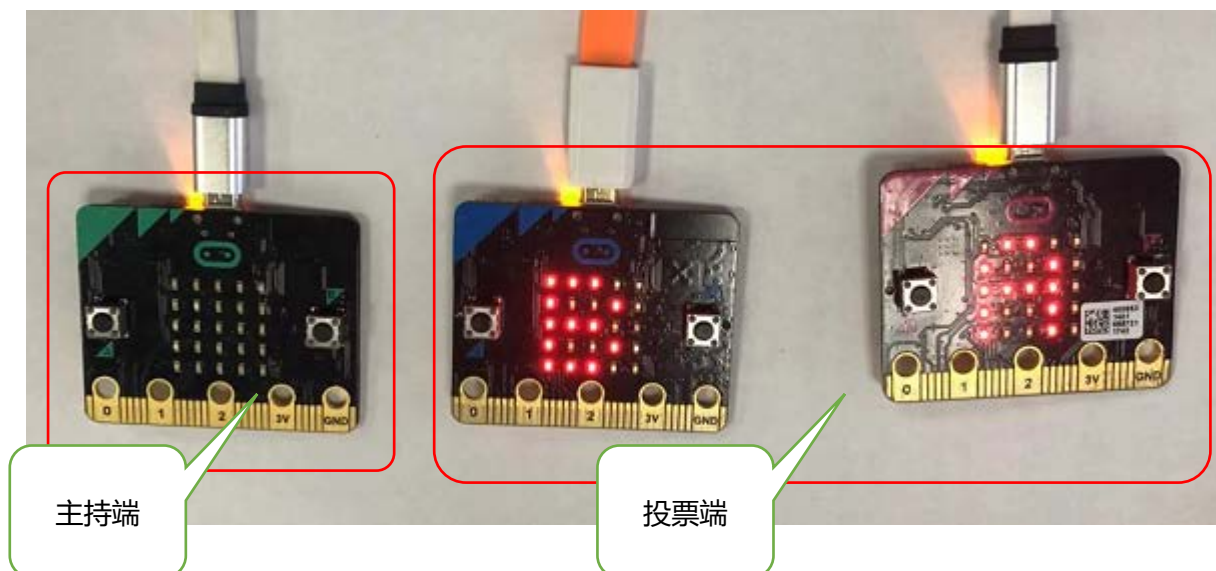
- 在实时模式中，点击右上角的小绿旗 “” 运行主持人端的程序。这时舞台区的 Mind+ 先生会介绍投票规则，艾比和艾弗里分别做自我介绍（运行结果如下图）。MQTT 也开始连接，并发送指令给 Easy IoT 平台。



- 进入 Easy IoT 网页，找到相应的 Topic，点击“发送消息”，查看 Topic 是否收到消息。



- 当我们按下投票端的 A、B 键，开始给选手投票（支持艾比按 A 键，支持艾弗里按 B 键）。按下按键后，会在 LED 点阵屏上显示 A 或者 B。



- 舞台区会显示实时的票数。

Mind+: 支持艾比 3

Mind+: 支持艾弗里 2



- 刷新网页，可以在 Easy IoT 平台查看艾比和艾弗里的票数。



最新消息

时间

消息

2019/8/30 17:42:56 艾比3



最新消息

时间

消息

2019/8/30 17:42:59 艾弗里2

2019/8/30 17:42:57 艾弗里1

第十课 综合项目

通过前面 9 个项目的学习，我们认识了许多有趣的传感器，同时也学习了物联网的基本框架、工作方式以及可应用的领域。相信学到这里，大家肯定有很多自己的想法，接下来我们就尝试着将这些想法变成现实，并做更多有趣的作品吧。

学前提示

随着物联网的兴起与发展，使得我们的生活充满了便利性，例如，在家里我们可以通过智能家居系统，控制热水器、安防监控、家庭影院、空调、窗帘、电视、冰箱、电饭煲、垃圾桶等等。如下图 1。在外面，我们可以通过物联网远程预约电梯。这样我们可以提前预约电梯，让电梯在相应的楼层等我们。

智能化家居无处不在

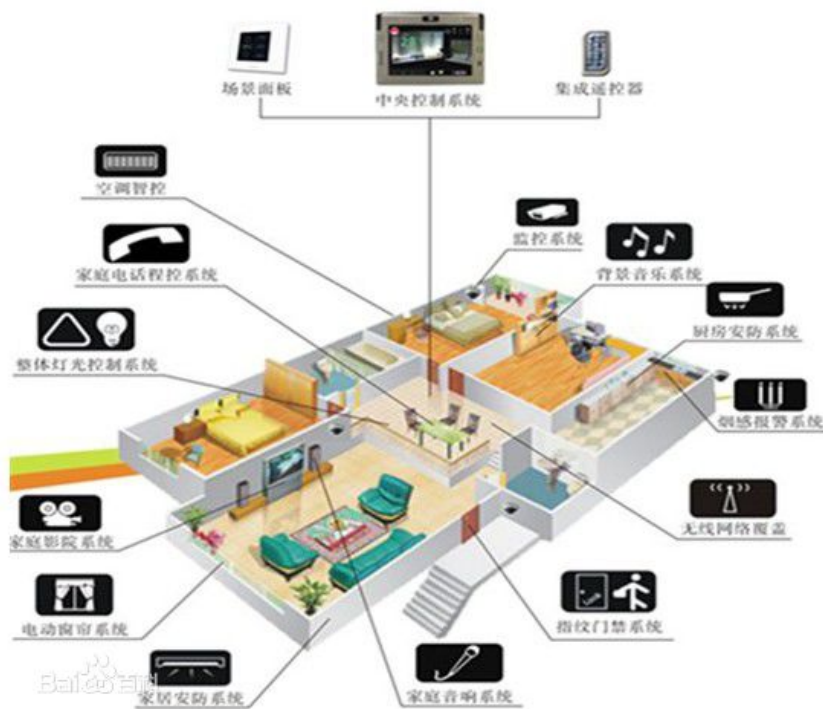
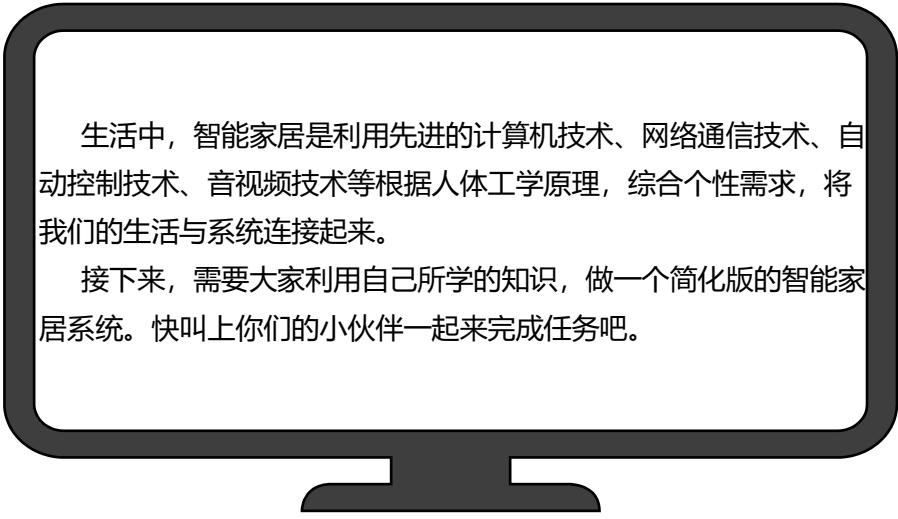


图 1

现在物联网正在各行业里快速展开，各种各样的畅想也铺天盖地的卷来了。接下来，你是否能根据任务书完成智慧生活中的物联网任务呢？

基础任务一 智能家居系统

任务发布



2 人为一组，自由分配任务，以套件里的传感器为基础（套件里没有的传感器，可以自己准备）完成任务。

任务背景	现在是一个智能化，信息化的时代。随着物联网与智能家居的发展，越来越多的人希望利用自己的手机或者其他移动设备通过物联网技术与家中的家电设备连接，并进行一个全方位的控制。从此开启一个便捷的智能生活。
问题	现在套件里的传感器有限，能实现的功能有限。
任务分解	任务 1、设计一个可行性方案，保证整个家居系统能够正常工作。 任务 2、结合物联网平台 Easy IoT。 任务 3、需要额外准备套件里没有的传感器。

技术手册

因为生活中的智能家居多种多样，同学们可能会遇见不知道做什么项目。那接下来，我列举几个例子：

1. 智能风扇——可以利用温度传感器检测环境温度，当温度达到某个值时，自动打开风扇。当温度降低到某个值时，风扇停止。这时就需要额外准备风扇模块或者电机模块。
2. LED 灯控——结合 Easy IoT 平台，创建一个通过物联网来控制 LED 灯，例如我们上班了，但是不确定家里的灯关没有，通过 Easy IoT 查看并开关灯。这时候就需要额外准备 LED 灯模块。
3. 自动垃圾桶——可以利用运动传感器检测是否有人在垃圾桶前方，如果有，就打开垃圾桶，等待几秒后，再关闭垃圾桶。
4. 智能门锁——有时候我们出门会想一想门是否关好，为了确保安全，基本上会折回去看一看，这样既费时间也费精力。如果我们制作一个智能门锁，通过物联网平台观察门是否关闭，如果没有关闭，可以通过物联网平台将门关上。这时候就需要额外准备继电器与阿里云 IoT 套件里的电磁锁。
5. 浴缸水位监测——有时候我们给浴缸放水时，后面会忘记关水龙头，如果我们设计一款水位监测器。

利用超声波监测水位的高度，当超声波与水面的距离值小于某个范围的时候，自动关闭水龙头。这时我们需要额外准备超声波。

除以上几个项目，我们还可以做智能插座、智能浇花系统、防盗报警器（可以参照远程门禁）等等。

以上是针对单个项目举例的，在项目中我们也可以将几个项目结合在一起。

其实在生活中，有很多地方都能看见智能家居的存在。只要善于观察与创新，相信你是可以完成任务的。

任务记录

任务已经明确好了，来开始完成任务吧。

STEP 1 召开团队会议，确定项目方案与硬件清单。并绘制硬件连线图

确定方案	智能家居方案（简单描述）：
硬件清单	所需要的硬件：
硬件连线图	简单绘制：

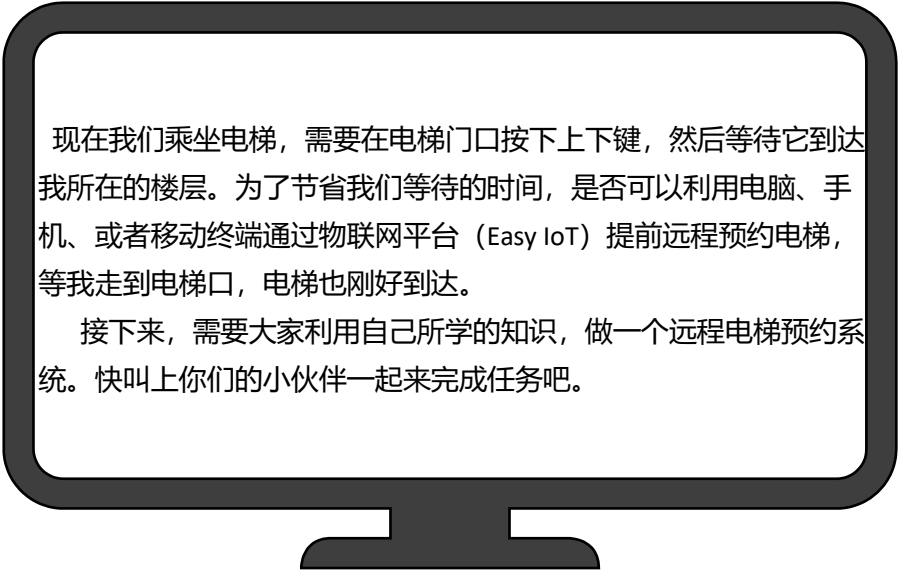
STEP 2 经过测试，请记录下，可实现的任务实现过程与实验数据和效果

实现过程	怎样实现这个方案，在此过程中遇见什么问题。
------	-----------------------

实验数据	主要填写 easy IoT 接收或者发送的数据。 发送数据： 接收数据：
效果展示	主要描述该项目最后的实验效果。

基础任务二 远程电梯预约

任务发布



2 人或者 3 人为一组，自由分配任务，以套件里的传感器为基础（套件里没有的传感器，可以自己准备）完成任务。

任务背景	节省等待电梯的时间，为人们提供更加高效节能、舒适的生活状态
问题	需要找一个代替电梯的产品（可以查阅技术手册）
任务分解	任务 1、当电梯停的楼层小于我所在的楼层时，电梯需要上升的层数。 任务 2、当电梯停的楼层大于我所在的楼层时，电梯需要下降的层数

技术手册

因为没有真正的电梯供我们做实验，所以我们可以采用 BOSON 套件里面的 RGB 灯带来代替。



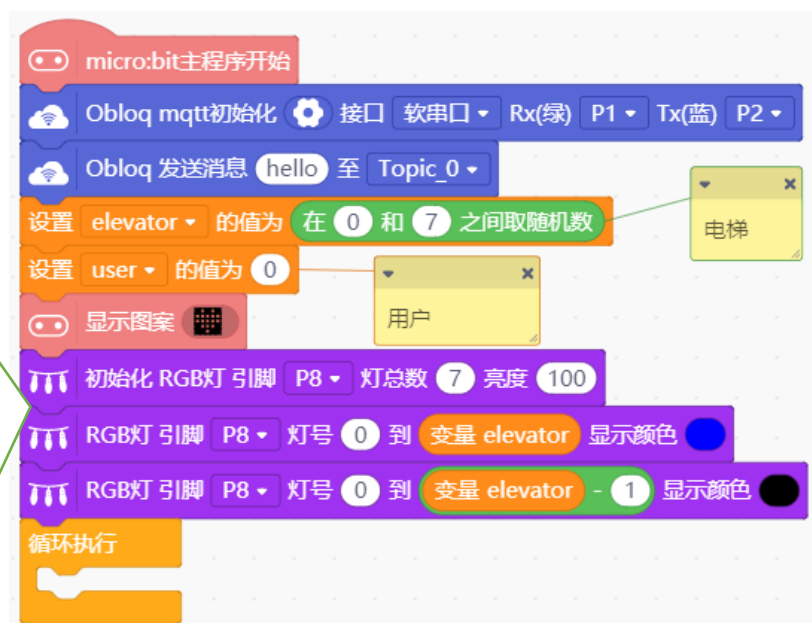
RGB 灯带

RGB 灯带一共有 7 颗灯，所以可以假设这栋楼一共有 7 层。首先需要获取电梯所在的楼层。在程序中我们可以利用随机数生成电梯当前所在的楼层数。



然后初始化 RGB 灯带，因为 RGB 灯带不能单独显示某一颗灯，所以我们需要将其他不需要亮的灯给熄灭。

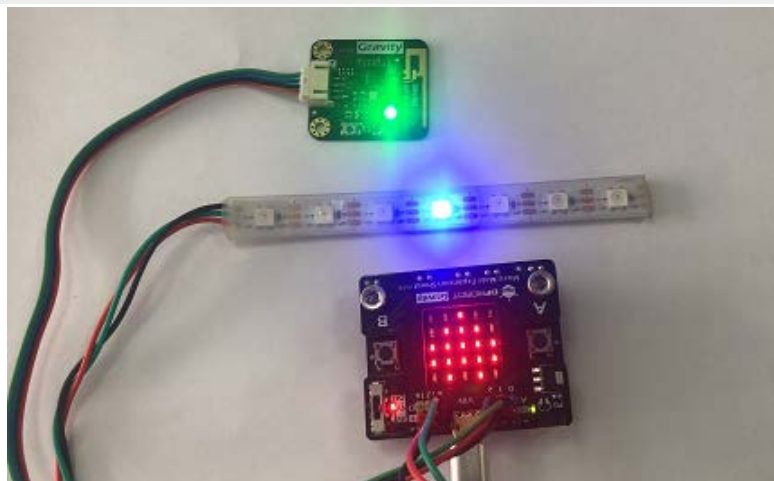
因为 RGB 灯点亮的方式是：如果我们需要点亮 0~5 号灯，则 0~5 号这 6 颗灯全部都会被点亮。如果只想点亮 5 号灯，就需要将 0~4 号这 5 颗灯的颜色设置成黑色（表示 RGB 灯熄灭）。



初始化完 RGB 灯带后，接下来我们就用它来模拟远程预约电梯的场景。通过 Easy IoT 输入我当前的楼层，让电梯到我所在楼层来等我。例如：我输入 3，电梯则停留在第三层，点亮第 4 颗 RGB 灯（RGB 灯带的第一颗灯的灯号是 0）。实验效果如下：

最新消息

时间	消息
2019/10/10 14:6:19	3



想要实现电梯上升与下降，我们需要对比电梯停留的楼层与我们自己所在的楼层。如果电梯停留的楼层大于我们所在的楼层，电梯应该需要下降；如果电梯停留的楼层小于我们所在的楼层，电梯应该上升。例如：电梯在第 6 层，我在第 2 层，电梯需要下降 4 层。则 RGB 灯需要向右移动-4 个单位；电梯在第 2 层，我在第 6 层，电梯则需要上升 4 层。则 RGB 灯需要向左移动 4 个单位。



同学们可以发挥自己的想象，怎样来完成这个任务，以上的例程可作为参考。

任务记录

任务已经明确好了，来开始完成任务吧。

STEP 1 召开团队会议，确定项目方案与硬件清单。并绘制硬件连线图

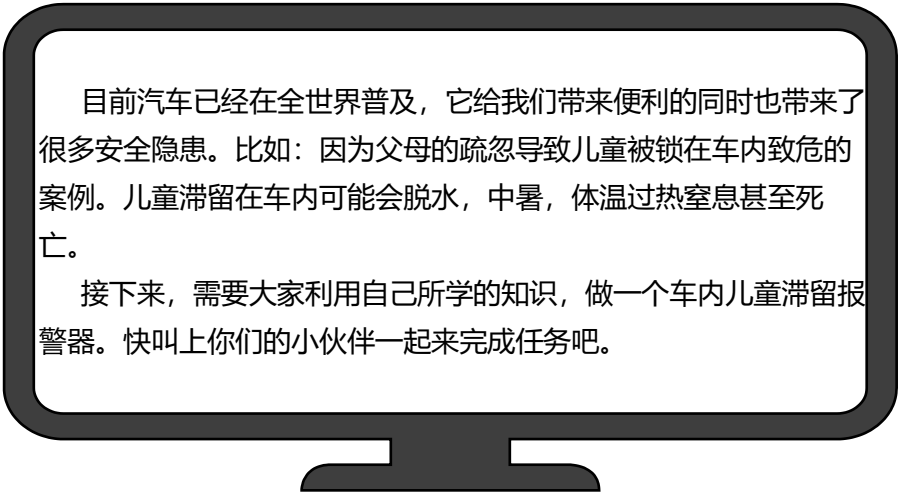
确定方案	确定远程预约电梯的方案（简单描述）：
硬件清单	所需要的硬件：
硬件连线图	简单绘制：

STEP 2 经过测试，请记录下，可实现的任务实现过程与实验数据和效果

实现过程	怎样实现这个方案，在此过程中遇见什么问题。
实验数据	Easy IoT 发送数据： 实验反馈（是否停留到相应的楼层）：
效果展示	主要描述该项目最后的实验效果。

基础任务三 车内儿童滞留报警器

任务发布



2 人或 3 人为一组，自由分配任务，以套件里的传感器为基础（套件里没有的传感器，可以自己准备）完成任务。

任务背景	根据国际上的统计，儿童滞留在车内的安全事故已经占车内非交通死亡相当大的比例，所以解决儿童滞留在车内的安全问题刻不容缓。
问题	现在套件里能实现的功能有限。
任务分解	任务 1、设计一个可行性方案，做一个车内儿童滞留报警器。 任务 2、结合物联网平台 Easy IoT。 任务 3、需要额外准备套件里没有的传感器。

技术手册

近年来，汽车内滞留儿童的事件屡次发生，严重的后果为儿童脱水死亡，这些发生的重要因素为儿童在车内睡着了，或者父母没有注意到孩子，将孩子反锁在汽车内，导致悲剧的发生，为此，我们可以设计一款儿童滞留报警器，在汽车不运行的时候检测儿童是否在汽车中。

针对儿童滞留车内的安全问题，我们可以根据以下几大类进行研究：

- 1、通过运动传感器检测车内是否有人运动，同时可以结合声音传感器判断车内是否有哭啼声，该设计需要根据车里的内部结构，考虑传感器摆放的位置。
- 2、通过压力传感器检测座椅上是否有压力，改设计需要考虑座椅上的杂物是否会引起误判。
- 3、通过二氧化碳、红外传感器检测儿童是否被滞留在车内，然后通过报警模块实现报警功能。然后我们可以通过物联网平台打开车门、天窗、通风系统。
- 4、可以将二氧化碳、压力、温度等多种传感器集成一体，通过这些传感器检测儿童滞留的情况，然后自动打开车门、通风系统实现自救。

综上所述，各种研究都是基于多种传感器的检测，同时可具备远程控制和自救功能。相信你可以根据提示完成任务。

任务记录

任务已经明确好了，来开始完成任务吧。

STEP 1 召开团队会议，确定项目方案与硬件清单。并绘制硬件连线图

确定方案	确定车内儿童滞留报警器的方案（简单描述）：
硬件清单	所需要的硬件：
硬件连线图	简单绘制：

STEP 2 经过测试，请记录下，可实现的任务实现过程与实验数据和效果

实现过程	怎样实现这个方案，在此过程中遇见什么问题。
实验数据	填写实验中测的得数据。
效果展示	主要描述该项目最后的实验效果。

第四章 智慧交通

交通运输业是指国民经济中专门从事运输的社会生产部门，其中包括铁路，公路，水路，航空，管道等多个运输部门。随着社会的进步，人们的生活水平提高了，随之而来的交通问题也越来越多的。例如：交通堵塞、交通事故、乘车不便。但是随着物联网的发展，产生了智慧交通，它不仅能够优化人们的出行体验也提高了出行的安全性。



第一课 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。



学前提示

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）。



车流量统计的项目结构分析图

硬件清单及连接

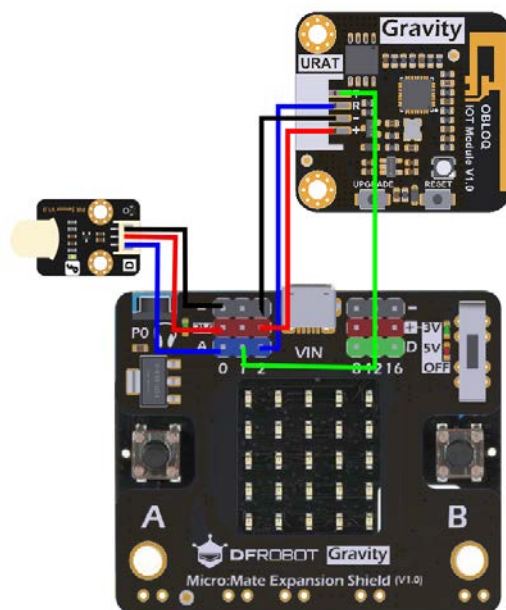
硬件清单

1 块 Micro:bit 主板 +1 块 Micro:mate 扩展板	OBLOQ 模块	电脑 (需要接入互联网)	USB 充电宝 (备选供电)	人体红外热释电运 动传感器

硬件连接

OBLOQ 模块中的 T 引脚（绿线）连接扩展板上的 P1 针脚（蓝色针脚），OBLOQ 模块中的 R 引脚

(蓝线) 连接扩展板上的 P2 针脚 (蓝色针脚), OBLOQ 模块中的 “-” 引脚 (黑线) 连接扩展板上的黑色针脚, OBLOQ 模块中的 “+” 引脚 (红线) 连接扩展板上的红色针脚。人体红外热释电运动传感器连接在扩展板上的 P0 针脚, 其中蓝色的线连接蓝色的针脚, 红色的线连接红色针脚, 黑色的线连接黑色针脚。



电路连接

知识点解析

数据采集的方式

经过第四章的学习, 我们知道了数据采集是利用一种装置, 将系统外部的数据采集并输入到系统内部的一个接口, 比如像温度传感器、光敏传感器等都是采集工具。将被采集的数据转换为电信号的各种物理量, 如温度、水位、风速、压力等。它们输出的值可以是模拟量也可以是数字量 (人体红外热释电运动传感器就是数字量)。哪在生活中采集数据的方式有哪些呢?

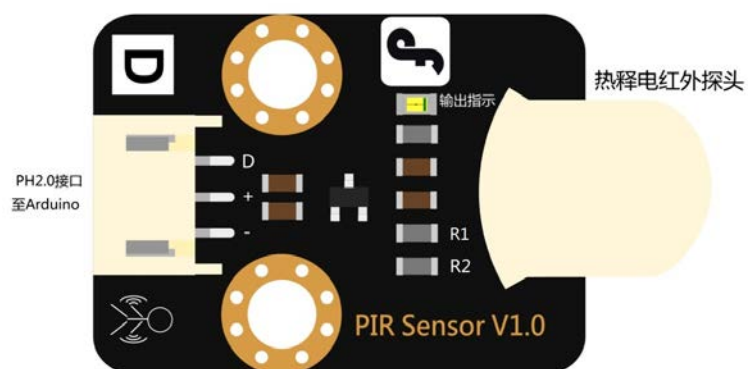
- 1、设备类: 是指从传感器或者其他设备待测设备等模拟和数字被测单元中自动采集信息的过程。常见的设备有: 条形码、扫描仪、麦克风、摄像头等。
- 2、网络类: 将批量采集的网页、论坛等内容直接保存到数据库或者发布到网络的一种信息化工具。



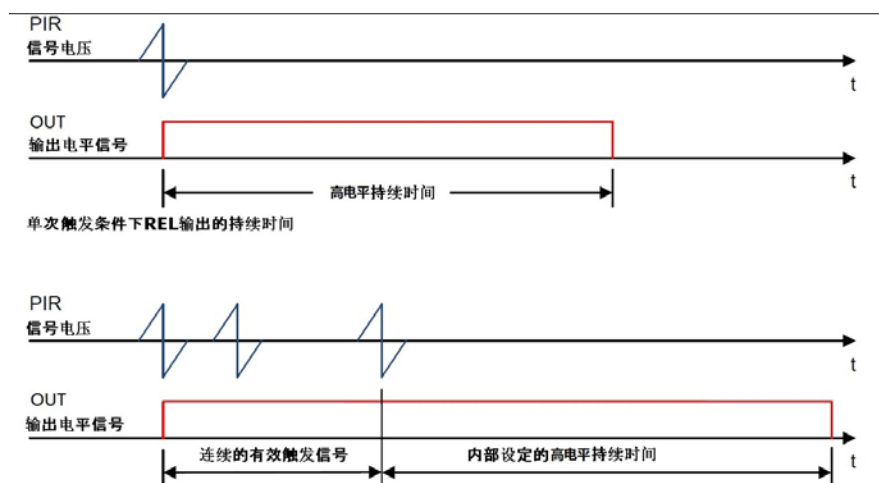
数据采集的方式

什么是人体红外热释电运动传感器

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。



它的触发模式是在检测的范围内当探头接收到的热释电红外信号超过探头内部的触发阈值之后, 会产生一个计数脉冲。当探头再次接收到这样的信号, 它会认为是接收到了第二个脉冲, 一旦在 4 秒钟之内接收到 2 个脉冲以后, 探头就会产生报警信号, 同时输出引脚输出高电平。另外, 只要接收到的信号幅值超过触发阈值的 5 倍以上, 那么只需要一个脉冲就能触发输出端的高电平输出。对于多次触发情况, 输出高电平的维持时间从最后一次有效脉冲开始计时。如下图。



注:项目中使用的红外热释电运动传感器默认的高电平延时时间为 2.3~3.5 秒

指令提前知

模块	功能说明
 系统运行时间	记录设备的运行时间。
	除以运算符。
 读取引脚 P0 人体红外热释电运动传感器	用来读取人体红外热释电运动传感器的状态两个状态 (0 或者 1)

程序编写

1.添加主控板及物联网模块

根据第一章相关内容添加主板——micro:bit 和物联网模块——OBLOQ。

2.添加传感器——人体红外热释电运动传感器

- 根据下图所示步骤添加人体红外热释电运动传感器



3.物联网模块初始化及联网测试

- 将初始化模块 “Obloq mqtt初始化 接口 硬串口 Rx(绿) P0 Tx(蓝) P1” 拖放至右边的编辑区。根据前面学习的内容定义发送和接收的引脚（要和实际连线一致）以及设置网页终端的相关账号与密码。
- 设置好物联网的相关信息后，将发送消息指令 “Obloq 发送消息 hello 至 Topic_0” 拖动到如图所示位置，帮助确认是否联网成功。



注：为了避免收发信息有冲突，将接口选为软串口比较好。

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
2019/10/15 8:31:58	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 Topic_0 接收到 Obloq消息” 模块拖放至编辑区。然后设置条件判断，当 Obloq 消息收到 “车流量” 这个消息就执行相应的内容。



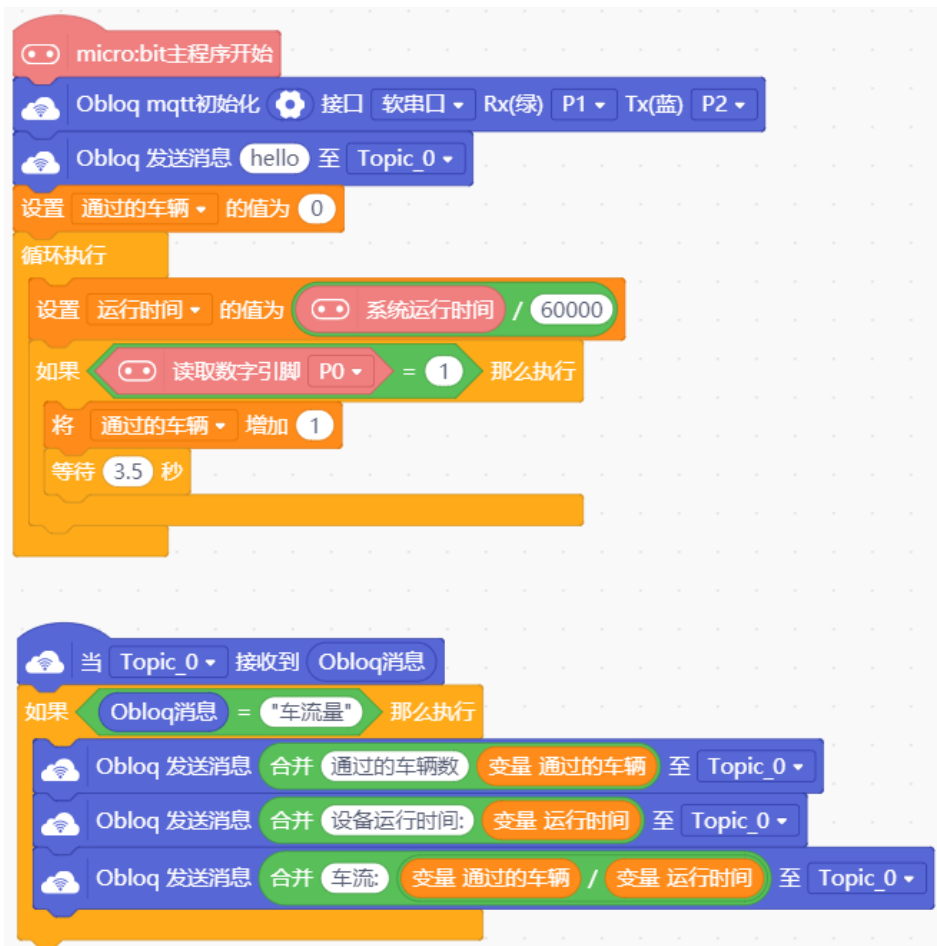
- 设置好条件后，就需要设置执行内容。因为车流量是在一定的时间内，某条公路点上所通过的车辆数，用公式表示为：车流量=通过车辆数/时间。 程序内容如下图所示：



参考程序

程序展示

将下面的程序下载到设备中。



注意:将程序上传至设备后, 需要将连在主控上面的 USB 数据线拔下, 连接到扩展板上的电源接口 (VIN 口)。额外供电后, 将板载上的调节开关调至 5V。

效果展示

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

最新消息	
时间	消息
2019/10/15 17:39:34	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。

2019/10/15 17:39:36 车流:2.00000

2019/10/15 17:39:35 设备运行时间:1.00000

2019/10/15 17:39:35 通过的车辆数2.00000

2019/10/15 17:39:34 车流量

拓展训练

现在我们学会统计了车流量，试着做一个统计人流量的程序吧。

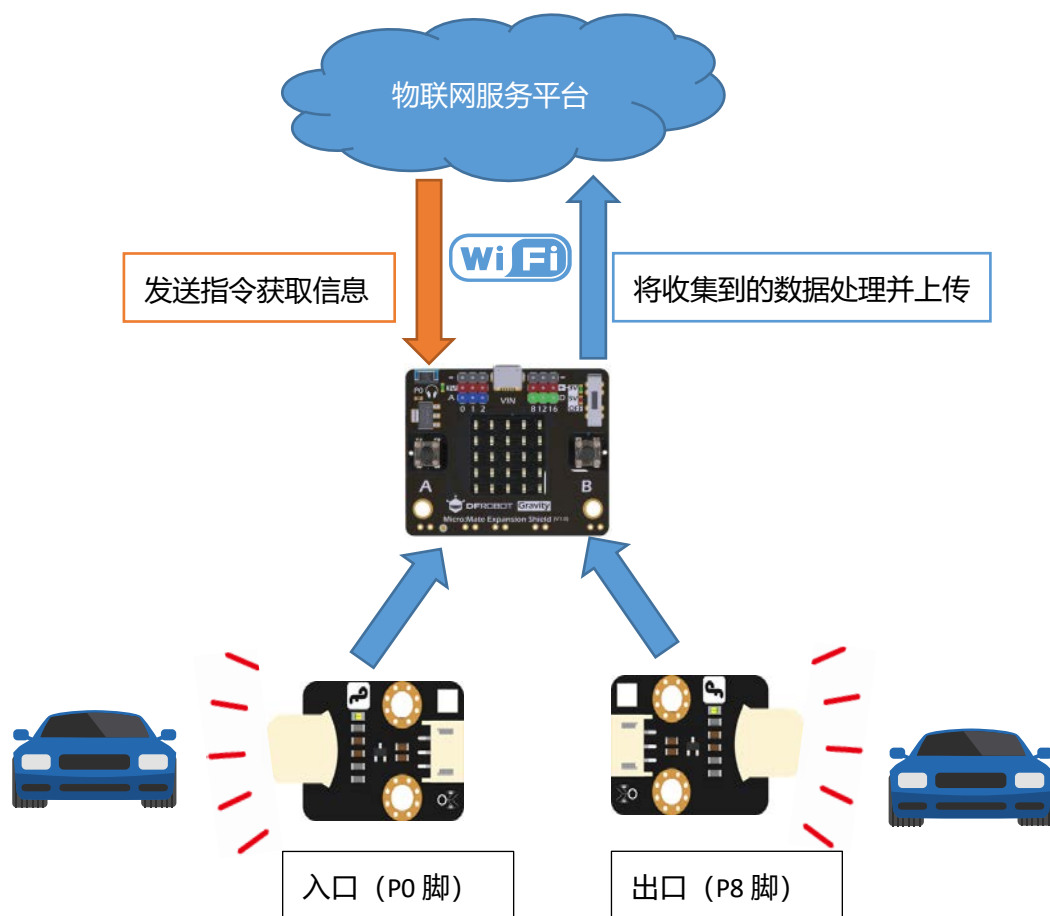
第二课 停车场剩余车位统计

有时候我们开车出去玩，首先要找的是附近有没有停车场，到了停车场我们可能会看见一块电子屏，显示当前剩余车位有多少？随着互联网与物联网的飞速发展，我们可以通过线上查看停车场的剩余车位。



学前提示

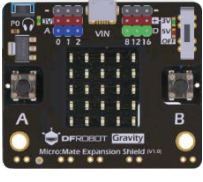
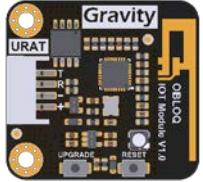
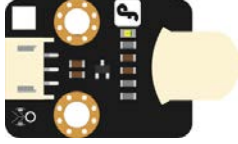
停车场剩余车位统计系统，分别在停车场的入口和出口安装传感器（运动传感器），统计车辆进出的数量。然后系统通过计算，计算出剩余的车位数。然后我们可以通过 LED 点阵屏或者物联网服务平台 (Easy IoT) 查看剩余车位。



停车场剩余车位统计的项目结构分析图

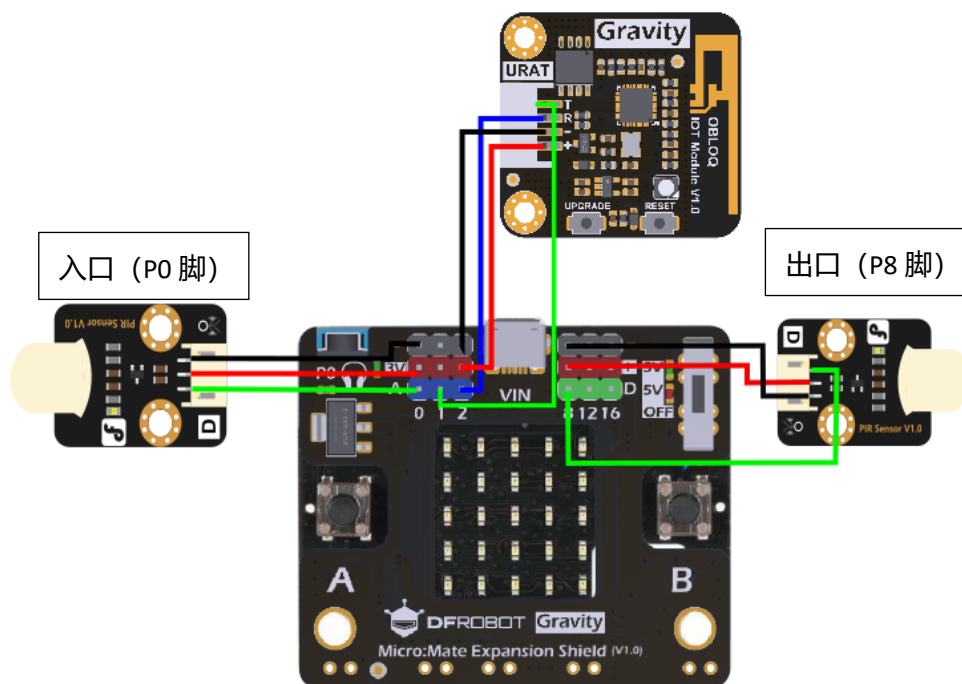
硬件清单及连接

硬件清单

				
1 块 Micro:bit 主 板+1 块 Micro:mate 扩展 板	OBLOQ 模块	电脑 (需要接入互联网)	USB 充电宝 (备选供电)	2 块人体红外热释 电运动传感器

硬件连接

OBLOQ 模块中的 T 引脚 (绿线) 连接扩展板上的 P1 针脚 (蓝色针脚), OBLOQ 模块中的 R 引脚 (蓝线) 连接扩展板上的 P2 针脚 (蓝色针脚), OBLOQ 模块中的 “-” 引脚 (黑线) 连接扩展板上的黑色针脚, OBLOQ 模块中的 “+” 引脚 (红线) 连接扩展板上的红色针脚。停车场入口的人体红外热释电运动传感器连接在扩展板上的 P0 针脚, 其中绿色的线连接蓝色的针脚, 红色的线连接红色针脚, 黑色的线连接黑色针脚。停车场入口的人体红外热释电运动传感器连接在扩展板上的 P8 针脚, 其中绿色的线连接绿色的针脚, 红色的线连接红色针脚, 黑色的线连接黑色针脚。



电路连接

指令提前知

模块	功能说明
	小于等于运算符。
	大于等于运算符。
	相加运算符，运算左右两个值的和。

程序编写

1. 添加主控板及物联网模块

根据第一章相关内容添加主板——micro:bit 和物联网模块——OBLOQ。

2. 添加传感器——人体红外热释电运动传感器

- 根据下图所示步骤添加人体红外热释电运动传感器



3. 物联网模块初始化及联网测试

- 将初始化模块 “ Obloq mqtt初始化 接口 硬件口 Rx(绿) P0 Tx(蓝) P1 ” 拖放至右边的编辑区。根据前面学习的内容定义发送和接收的引脚（要和实际连线一致）以及设置网页终端的相关账号与密码。
- 设置好物联网的相关信息后，将发送消息指令 “ Obloq 发送消息 hello 至 Topic_0 ” 拖动到如图所示位置，帮助确认是否联网成功。



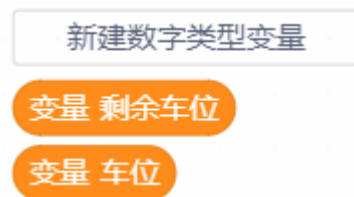
注：为了避免收发信息有冲突，将接口选为软串口比较好。

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
2019/10/15 8:31:58	hello

4.创建两个变量

- 创建两个变量，分别是 “变量 车位” 和 “变量 剩余车位”，都是用来记录车位信息。



- 将车位初始化的值设置为 100，表示该停车场的车位数量共 100 个。



5. 实时统计停车场的车位数量

由学前提示得知，P0 口为入口，P8 口为出口，所以我需要分别判断当入口有车进入与出口有车出去的两种情况。

- 当入口有车进入时的情况。**利用条件判断语句判断 P0 脚的人体红外热释电运动传感器的状态为 1。说明有车经过，则车位的数量减 1。

经过一辆车，车位余量： $100-1=99$ ；
经过第二辆车，
车位余量： $99-1=98$ ；
依次类推。



- 根据实际情况，有车进入，也可能有车出去。为了方便统计剩余车位，接下来我们还需要做一步，就是将现在监测到的车位数量赋值给“剩余车位”。



- 因为给停车场设置的是 100 个车位，如果 100 个停车位都停满了，我们应该怎么显示呢？利用判断条件，判断车位的数量是否小于等于 0。如果车位小于等于 0 时，显示车位数量为 0，屏幕显示“x”图案。



- 设计完入口的程序，接下来我们设计出口的程序，利用条件判断语句判断 P8 脚的人体红外热释电运动传感器的状态为 1。说明有车经过，则车位的数量加 1。



需要将当前的车位数再次赋值给剩余车辆。如果不添加这一步，车位数量的值只会加一次。例如，剩余车位为 89，经过一辆车 $89+1=90$ 。经过第二辆或者第三辆车后，还是 90。

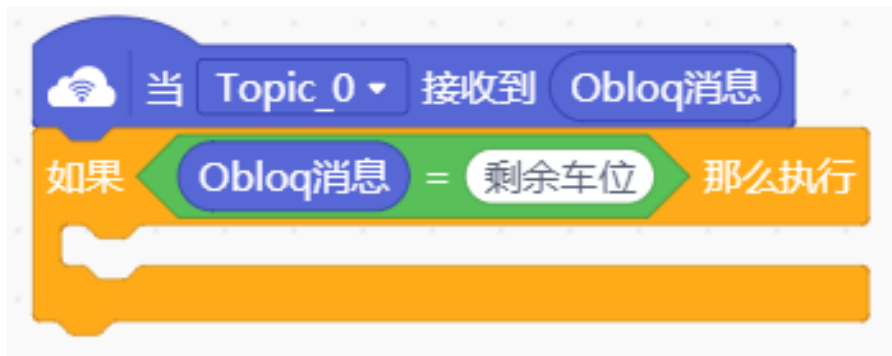
- 入口的问题是，如果车位已经停满了，怎么显示？那现在需要解决的问题是，如果车位都空了，又该怎么显示呢？利用判断条件，判断车位的数量是否大于等于 100。如果车位大于等于 100 时，显示车位数量为 100，屏幕显示“√”图案。



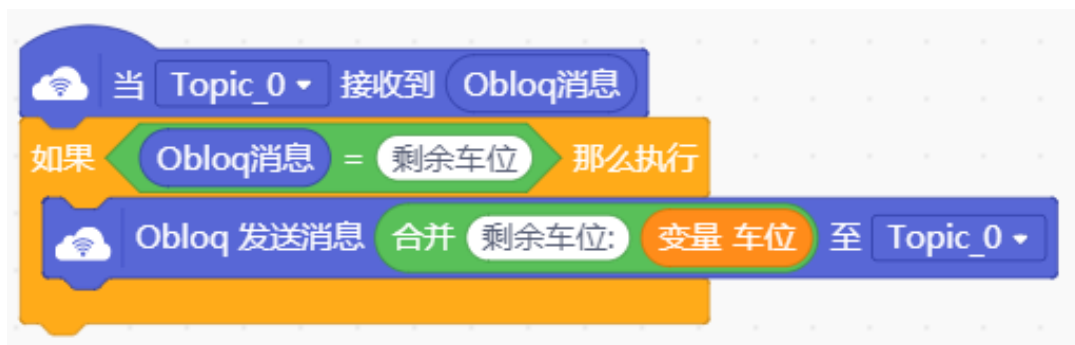
6.通过 Easy IoT 查看剩余车位

- 通过我们设计好的出入口程序计算出了剩余车位，那接下来我们就可以将计算后的数据上传到 Easy IoT 平台上。

- 将 “” 模块拖放至编辑区。然后设置条件判断，当 Obloq 消息收到 “剩余车位” 这个消息就执行相应的内容。



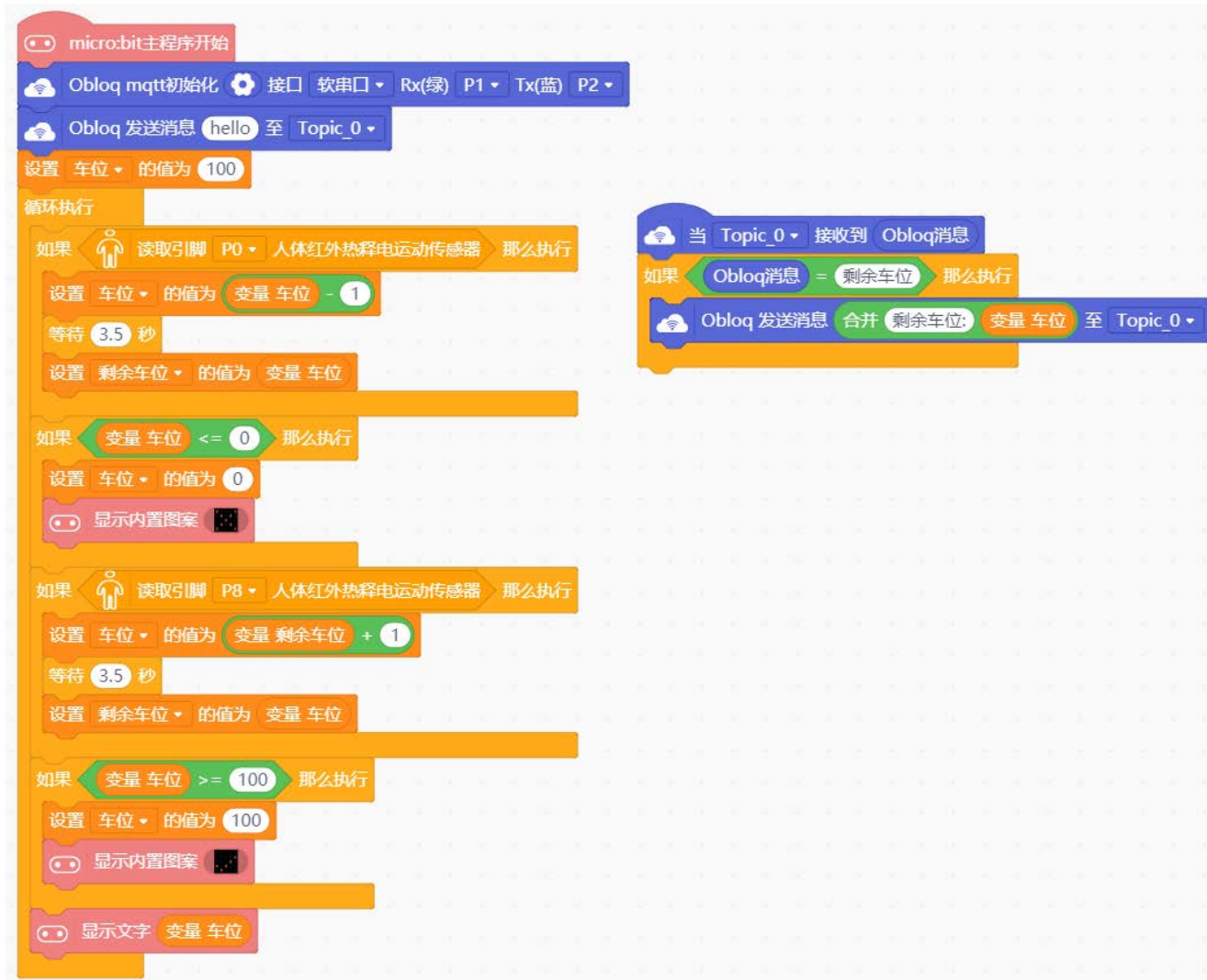
- 设置好条件后，就需要设置执行内容。程序内容如下图所示：



参考程序

程序展示

将下面的程序下载到设备中



效果展示

- 进入 Easy IoT 网页发送信息给 micro:bit 设备。找到相应的 Topic，点击“发送消息”，发送指令“剩余车位”

时间	消息
2019/10/18 17:31:8	剩余车位

- 刷新网页可以查看剩余车位。



最新消息

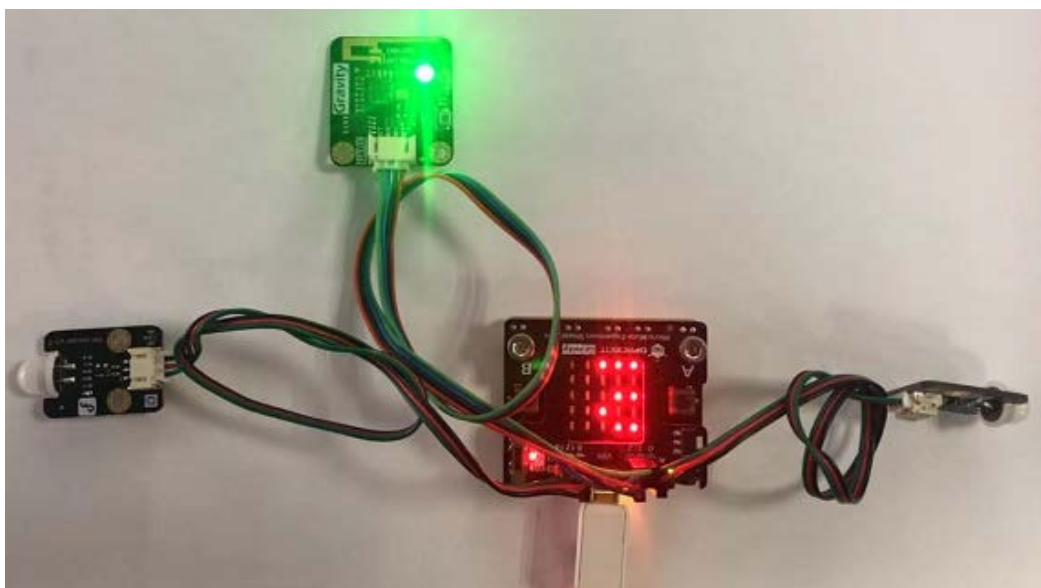
时间

消息

2019/10/18 17:33:27 剩余车位:11.00000

2019/10/18 17:33:27 剩余车位

注意：为了避免误差，不要将两个人体红外热释电运动传感器挨着太近。



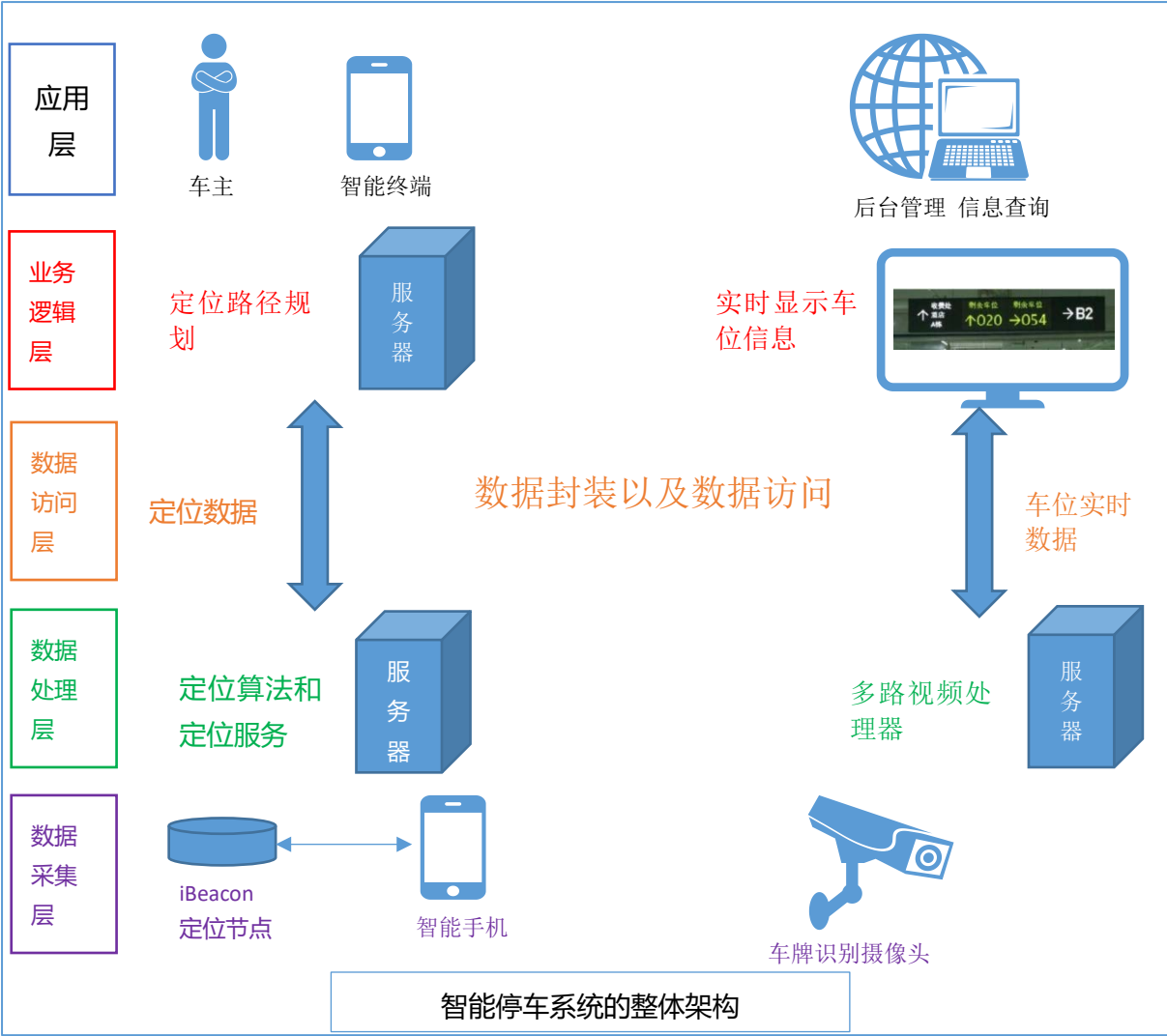
拓展训练

如果想分别知道从入口和出口经过多少辆车？应该如何设计程序？发挥你的想象力，试着设计一个出入口车辆统计系统。

拓展阅读

基于物联网的智能停车系统

智能停车系统充分运用了物联网识别技术，通过在停车场出入口安装的百万级高清的车牌识别摄像头抓拍车牌并快速识别车辆信息，这样免取卡便能进入停车场。车辆进入停车场内，车位引导系统实时显示停车位。此外还有反向寻车系统，只需在停车场内的查询机或者通过扫描二维码，然后输入车牌号码，反向寻车系统就会快速提供停车位的信息和电子地图显示。停车场缴费也很方便，在商场 pos 机、停车场自助缴费机、手机移动支付、二维码支付、停车场人工缴费都可以来完成的。



第三课 超速报警器

在高速路上行驶，经常会听见导航播报前方为测速路段请小心驾驶，或者是：前方为测速路段，您已超速，请减速。这就是高速路上的区间测速，在高速路上很多超速行驶的司机都是被区间测速检测抓拍出来的。接下来我们就根据区间测速的原理做一个超速报警器吧。

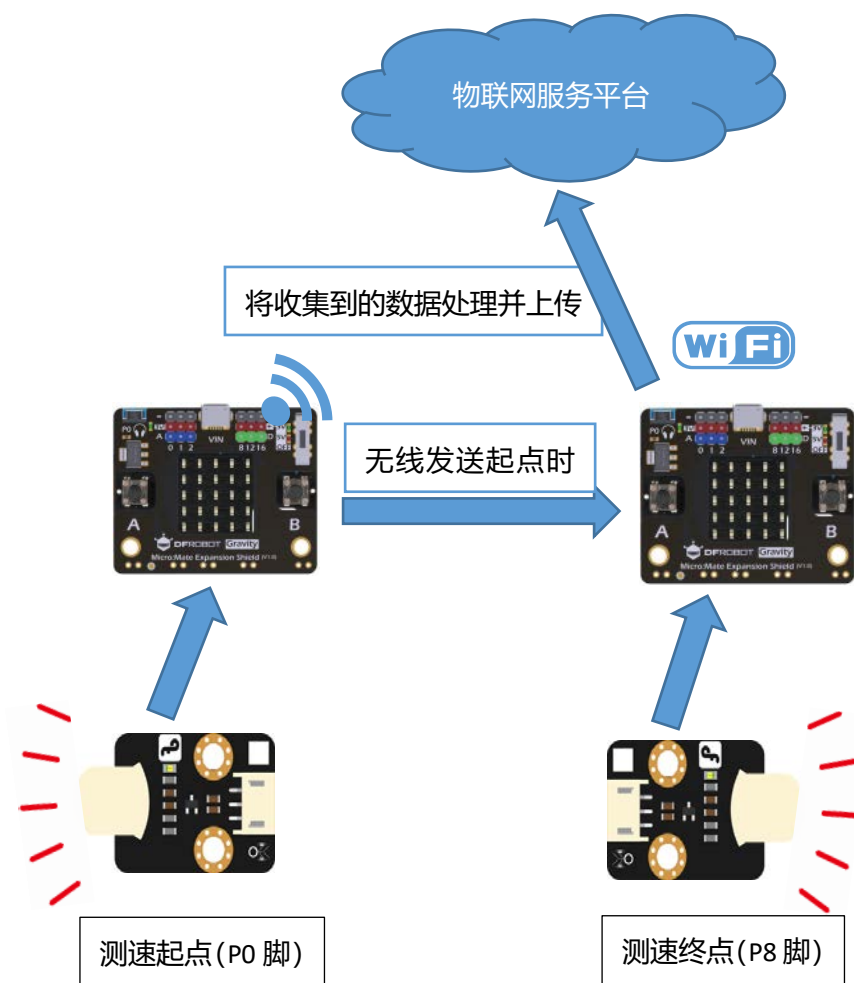


小贴士:

根据《道路交通安全法实施条例》第四十五条规定，在没有限速标志、标线的道路上，机动车不得超过下列最高行驶速度：没有道路中心线的道路，城市道路限速 30km/h，公路限速 40km/h，同方向只有一条机动车道的道路，城市道路限速 50km/h，公路限速 70km/h，高速限速 120km/h。

学前提示

超速报警器是分别在测速路段的起点和终点安装传感器（运动传感器），当在起点处的传感器检测到有车辆经过时，就将这个时间点作为测速的开始时间，当车辆经过终点处的传感器后，将这个时间点作为测速的结束时间，通过公式 $(v=s/t)$ 计算车辆通过这段路程的平均速度。然后我们可以通过物联网平台（Easy IoT）查看起始时间和平均速度。



超速报警器的项目结构分析图

硬件清单及连接

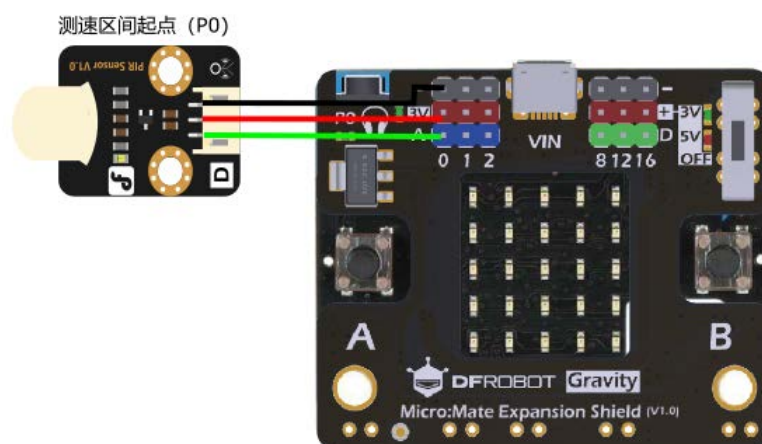
硬件清单

2 块 Micro:bit 主板+2 块 Micro:mate 扩展板	OBLOQ 模块	2 块人体红外热释电运动传感器
带功放喇叭模块	电脑 (需要接入互联网)	USB 充电宝 (备选供电)

硬件连接

测速区间起点电路

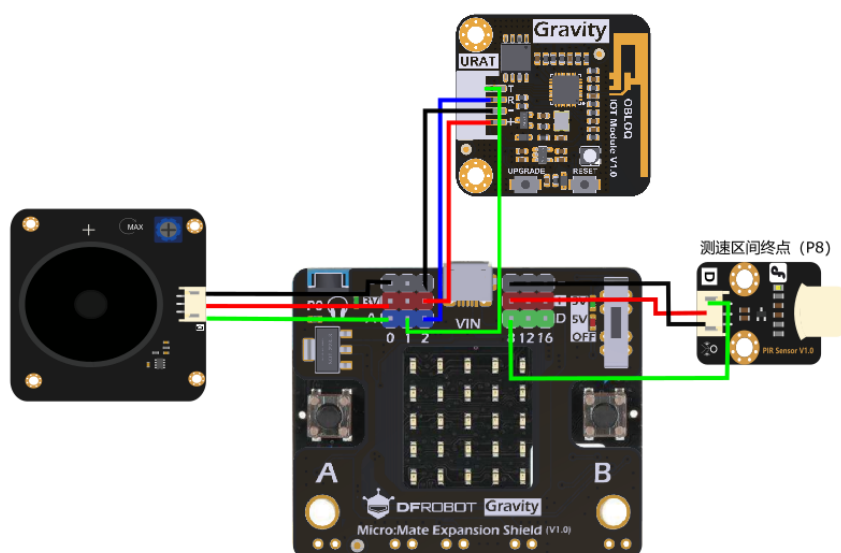
将测速区间起点的人体红外热释电运动传感器连接在扩展板上的 P0 针脚，其中绿色的线连接蓝色的针脚，红色的线连接红色针脚，黑色的线连接黑色针脚。



测速区间起点电路连接

测速区间终点电路

OBLOQ 模块中的 T 引脚（绿线）连接扩展板上的 P1 针脚（蓝色针脚），OBLOQ 模块中的 R 引脚（蓝线）连接扩展板上的 P2 针脚（蓝色针脚），OBLOQ 模块中的“-”引脚（黑线）连接扩展板上的黑色针脚，OBLOQ 模块中的“+”引脚（红线）连接扩展板上的红色针脚。将测速区间终点的人体红外热释电运动传感器连接在扩展板上的 P8 针脚，其中绿色的线连接绿色的针脚，红色的线连接红色针脚，黑色的线连接黑色针脚。将喇叭模块连接在扩展板上的 P0 脚，其中绿色的线连接蓝色针脚，红色的线连接红色针脚，黑色的线连接黑色针脚。



测速区间终点电路连接

知识点解析

区间测速

所谓区间测速是在同一路段上布设两个相邻的监控点，基于车辆通过前后两个监控点的时间来计算车辆在该路段上的平均行驶速度，并依据该路段上的限速标准判定车辆是否超速违章。

区间测速的优点：以前的单点测速仪，在司机熟知测速点的情况下，可以通过刹车降低车速逃避处罚，很容易造成追尾事故。而区间测速采取计算平均车速的方法来检测车辆是否超速，堵住了司机投机取巧的手段，更加科学公正。

区间测速的缺点：因为测的是平均速度 $v = \frac{s}{t}$ ，但事实上只要瞬时速度 $\lim_{\Delta t \rightarrow 0} \frac{\Delta x}{\Delta t}$ 超过规定值，就会有危险，所以区间测速只能查准（查到的一定是超速行驶），但是查不全（有人在某一瞬间超速行驶没有查出来）。

指令提前知

模块	功能说明
	转换字符串，可以将字符串转换成整数或者小数。
	播放音符，可以选择音符与节拍。

程序编写

1.添加主控板及物联网模块

根据第一章相关内容添加主板——micro:bit 和物联网模块——OBLOQ。

2.添加传感器——人体红外热释电运动传感器

- 根据前两课的步骤添加人体红外热释电运动传感器。

3.设计区间测速起点设备的程序

- 首先打开“无线通信”，设置无线频道为“7”（注意起点和终点的频道需要一致）。



- 然后利用条件判断，判断当有物体经过（即人体红外热释电运动传感器的状态为 1），通过无线发送“系统运行时间”给终点设备。



4.物联网模块初始化及联网测试

- 将初始化模块 “Obloq mqtt初始化 接口 硬串口 Rx(绿) P0 Tx(蓝) P1” 拖放至右边的编辑区。根据前面学习的内容定义发送和接收的引脚（要和实际连线一致）以及设置网页终端的相关账号与密码。
- 设置好物联网的相关信息后，将发送消息指令 “Obloq 发送消息 hello 至 Topic_0” 拖动到如图所示位置，帮助确认是否联网成功。



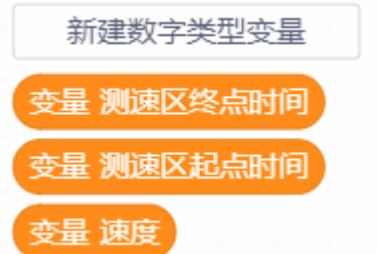
注：为了避免收发信息有冲突，将接口选为软串口比较好。

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
2019/10/15 8:31:58	hello

5. 设计区间测速终点设备的程序

- 创建三个变量，分别是 “变量 测速区起点时间”、“变量 测速区终点时间” 和 “变量 速度”，用它们记录测速区起点的时间、终点的时间和平均速度。



- 打开无线通信，设置无线频道为 “7” (和起点的频道一致)。



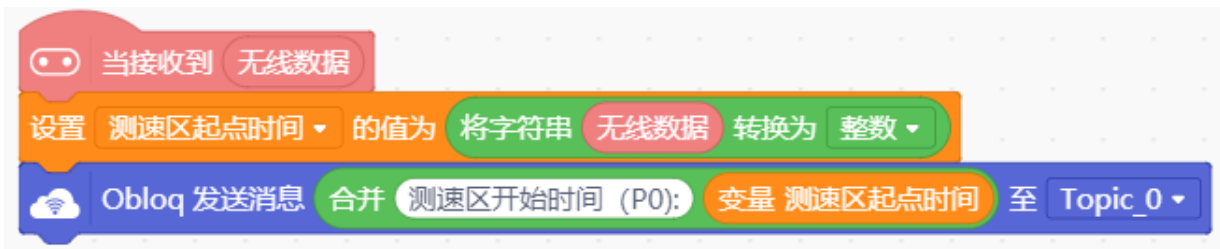
- 获取测速区终点时间的值。



小贴士:

因为项目实验测试，将时间缩短了 60 倍，用秒为单位计时。

- 将模块 “当接收到 无线数据” 拖放到如图所示位置，接收测速区起点时间。并将时间发送给物联网平台（Easy IoT）



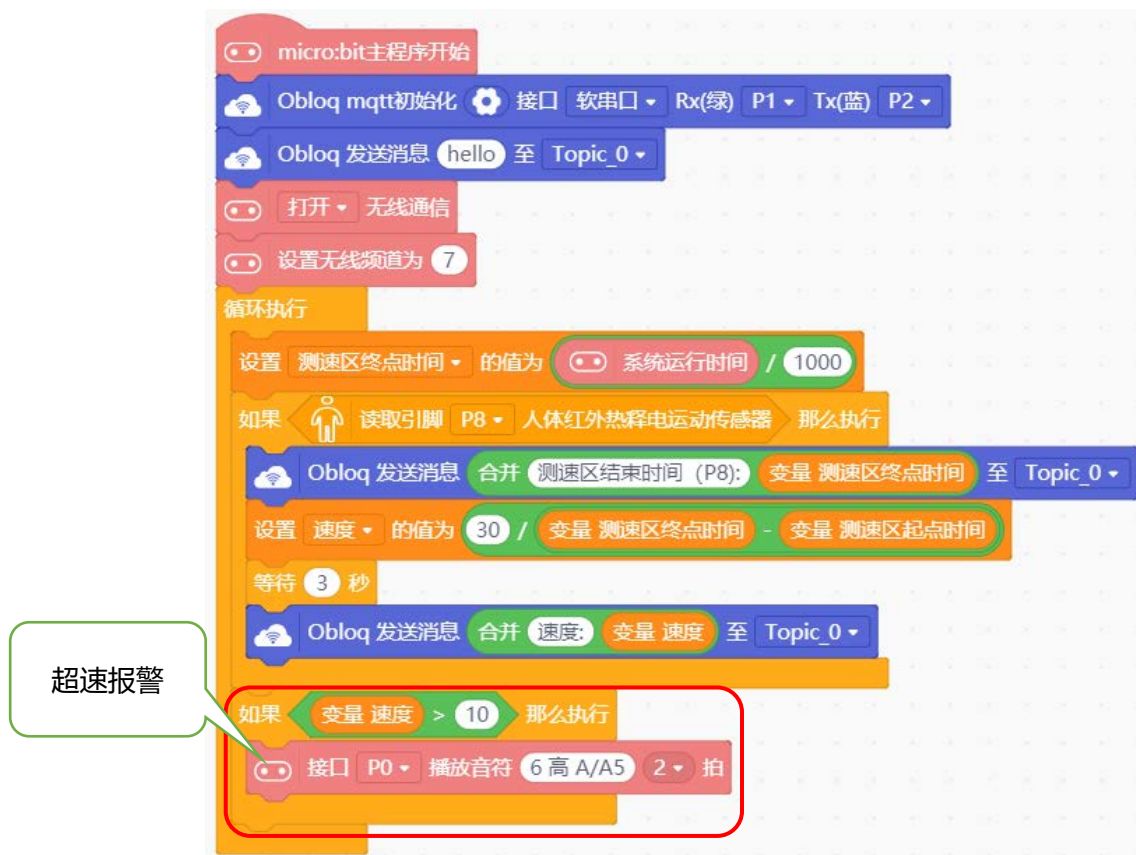
- 现在我们获取了起始点的时间，接下来就可以通过公式计算车辆通过这段路程的平均速度。
- 首先利用条件判断，判断当有物体经过（即人体红外热释电运动传感器的状态为 1），将测速区的终点时间发送给 Easy IoT 平台。



- 然后利用公式 ($v=s/t$) 计算出平均速度, 并将计算出的结果上传到物联网平台 (Easy IoT)。



- 最后我们利用条件判断语句, 判断当速度超过 10cm/s 时, 小喇叭发声, 表示该车辆超速行驶了。



参考程序

程序展示

将下面的程序下载到设备中

超速报警器——起点设备程序



超速报警器——终点设备程序（见下一页）

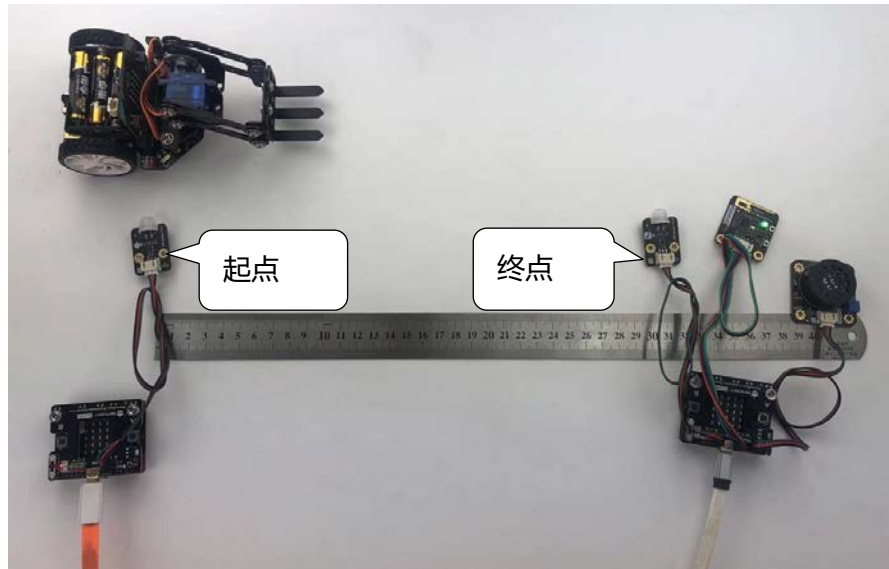


效果展示

- 同时启动测速区的起始设备。当有车经过测速区的起点时，Easy IoT 记录一条“测速区开始时间”。

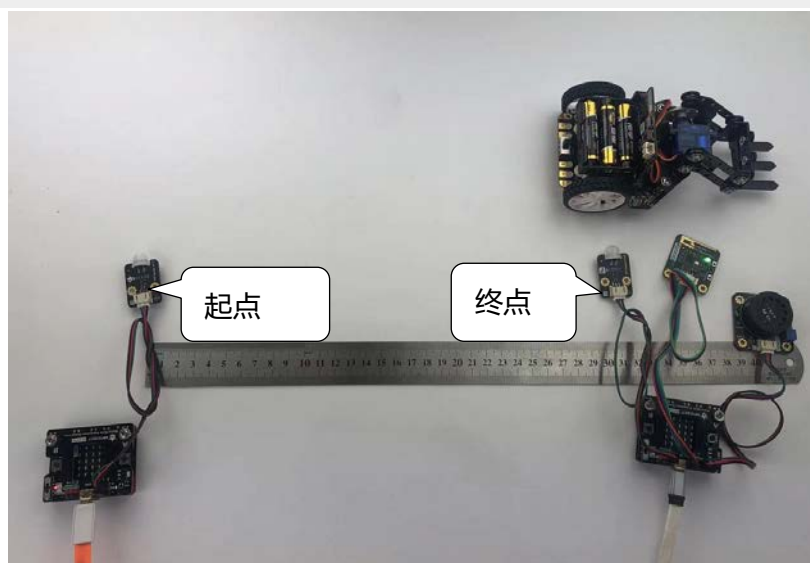
2019/10/21 15:39:54 测速区开始时间 (P0):11.00000

2019/10/21 15:39:47 hello



- 当小车经过终点时，Easy IoT 记录一条“测速区结束时间”，然后计算出平均速度。

最新消息	
时间	消息
2019/10/21 15:40:12	速度:1.42857
2019/10/21 15:40:11	测速区开始时间 (P0):27.00000
2019/10/21 15:40:9	测速区结束时间 (P8):32.00000
2019/10/21 15:40:2	速度:3.33333
2019/10/21 15:39:58	测速区结束时间 (P8):20.00000
2019/10/21 15:39:54	测速区开始时间 (P0):11.00000
2019/10/21 15:39:47	hello

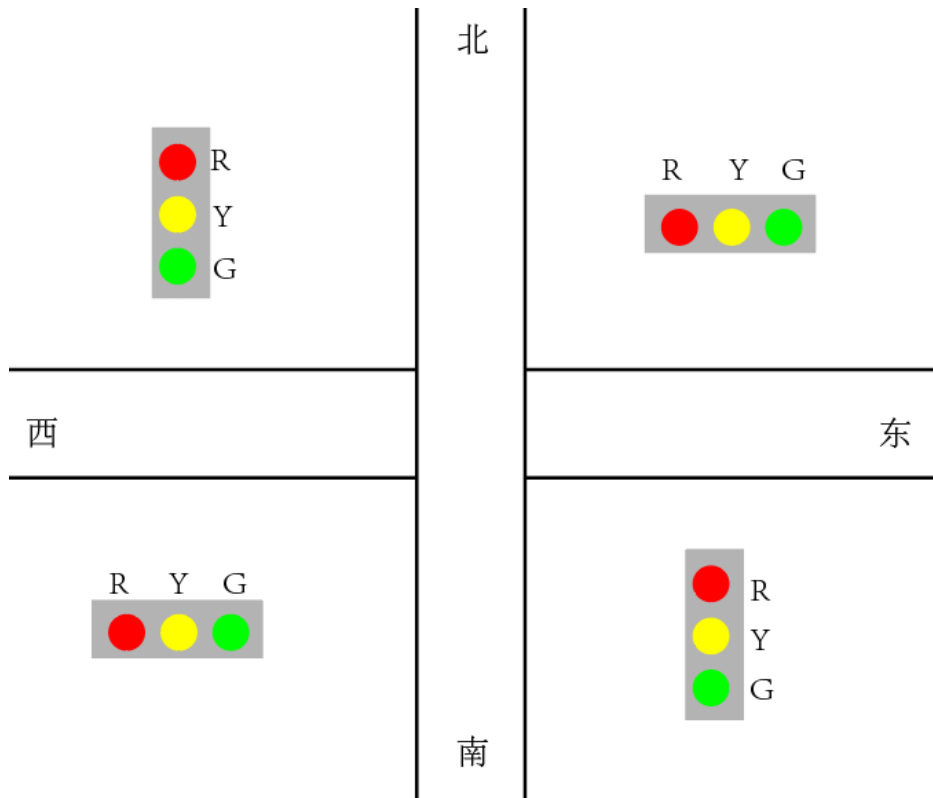


注意：在测试期间，因减少人员走动，这样可以避免传感器误判。

拓展阅读

基于物联网技术的智能交通

物联网时代的智能交通，通过对信息空间虚拟化车辆的智能管理控制，让交通事故从“事后处理”转化成“事前预判”。如虚拟十字路口和交通灯模拟。



虚拟十字路口和交通灯模拟

智能交通主要运用的技术有如下几种：

- 1、感知技术：感知技术是基于车辆和道路基础设施的网络系统，采用了先进的检测、感知、识别技术获取人与物的位置以及身份等信息。
- 2、无线通信技术：目前已经有多种无线通信解决方案应用在智能交通系统中，UHF 和 VHF 频段上的无线调制解调器通信，被广泛用于智能交通系统中。目前很多车辆都可以通过移动设备、无线通信方式与卫星、移动电话网络等进行通信，并且还可以利用 WiFi、移动电话网络接入互联网。
- 3、全球定位系统（GPS）：现在 GPS 是很多车内导航系统的核心技术，车辆中配备的嵌入式 GPS 接收器能够接收多个不同的卫星信号，并计算出车辆所在的位置，其误差也小，一般在几米的范围。
- 4、视频监测技术：利用视频摄像设备进行车流量统计和事故检测。
- 5、计算技术：实现海量交通信息的储存、传输、处理是智能交通管理系统建设的重点。在未来，车辆中将配备数量更少，功能更加强大的处理器，这时就需要强大的计算技术的支持。
- 6、基于物联网的路网分析技术：路网分析技术包括路网容量分析、路网脆弱性分析、路网广义费用优化等技术。如下表：

路网分析技术	概念	应用
路网容量分析技术	指在受交通控制的道路某点或断面处，给定的时间范围内，车辆或行人能合理的通过最大数量。	例如交通拥堵、自然灾害等现象都会造成道路通行能力急剧下降，在物联网的环境下，可以充分利用交通状态和车辆状态的实时数据，并在车辆和管理系统之间双向传输，即车辆在行驶的过程中向系统发送一些路况和车辆信息，同时管理系统也可以发一些预告信息。
路网脆弱性分析技术	指在受到随机事件的影响下，网络性能或服务水平下降，进而失去部分或者全部的连通能力	物联网技术为管理系统全方位提供信息来源，交通基础设施为物联网提供应用环境与平台，在该背景下，道路、车辆、交通工程设施都具备感知、计算、通信能力。因此道路设施能够将路面的完好情况，摩擦系数、温度、气象条件等性能参数和流量、速度、密度等交通状态参数实时发送给交通管理系统。同时也能将车辆的速度，加速度等运行参数发送给交通管理系统。
路网广义费用优化技术	加强公路网路管理，提高车辆运行效率，对于汽车节约燃油消耗、减少尾气排放有着重要的意义	在物联网背景下，能实时检测出每辆车的燃油消耗量和尾气排放量。同时结合道路交通个网络的实时数据，对路网和车辆性能特征进行多方面的分析计算。然后规划出消耗的最低费用。

- 7、基于车联网的车辆间协同运行技术：是指车与车、车与路、车与人、车与传感设备等，实现移动交互通信的系统。通俗的讲就是利用车载电子传感装置，通过汽车导航系统、移动通信技术、智能终端设备与信息网络平台，使人、车、路与城市之间实时联网，实现信息互通。从而对人、车、物、路进行有效的监控、管理、调度的网络系统。

第四课 综合项目

通过前面 3 个项目的学习，我们知道了物联网在交通方面的应用。然而随着科学技术的发展，物联网在交通上的应用领域也会越来越多。下面我们就一起来想想物联网在交通方面的典型案例，同时尝试着将它们变成现实吧。

学前提示

建设智能交通的目的是使人、车、路密切配合达到和谐统一，这样也能大幅度地提高交通安全、保障交通运输效率、改善交通运输环境以及提高能源利用效率。例如，电子警察的应用、ETC 的普及、智能车载交互系统、无人驾驶车辆与旅客自动传输系统等。



接下来，你是否能根据任务书完成智慧交通中的物联网任务呢？

基础任务一 闯红灯电子警察系统

任务发布

自 1997 年电子眼研发成功，“电子警察”系统就已经应用到我国各大城市。例如，陕西省就利用“闯红灯电子警察系统”来解决交通中存在的车辆流动、路面状况、违章行为和粉尘环境等诸多问题。

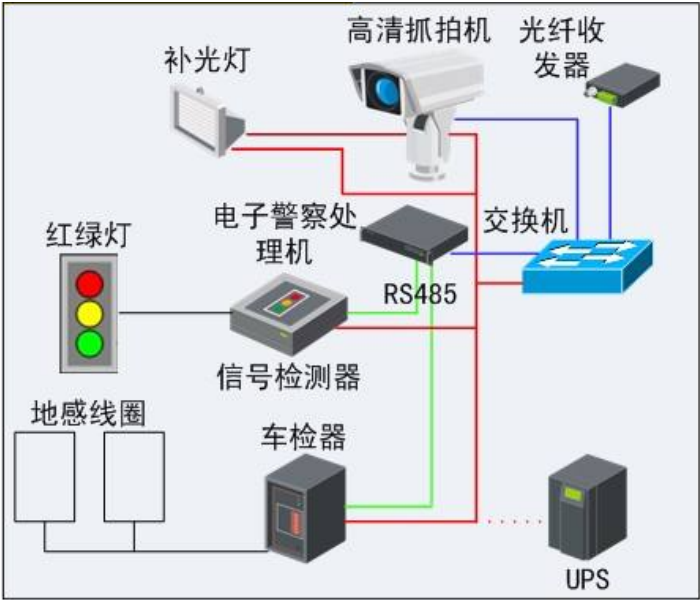
接下来，需要大家利用自己所学的知识，做一个简化版的闯红灯电子警察系统。快叫上你们的小伙伴一起来完成任务吧。

2 人或者 3 人为一组，自由分配任务，以套件里的传感器为基础（套件里没有的传感器，可以自己准备）完成任务。

任务背景	电子警察对事故的捕捉迅速，判断准确，在恶劣的环境下也能正常工作，它在维护良好的交通秩序、规范行车安全、增强安全驾驶意识、杜绝闯红灯现象和打击违法犯罪等方面具有重要意义。
问题	现在套件里的传感器有限，能实现的功能有限。
任务分解	任务 1、设计一个可行性方案，保证整个闯红灯电子警察系统能够正常工作。 任务 2、结合物联网平台 Easy IoT。 任务 3、需要额外准备套件里没有的传感器。

技术手册

“电子警察” 通常由图像检测、拍摄、采集、处理、传输与管理以及辅助光源、辅助支架和相关配套设备等组成。



目前，为了缓解交通拥堵和交通事故这些难题，“电子警察” 应用到各大城市，最常见的就是“闯红灯电子警察系统”。“闯红灯电子警察系统” 采用无人值守的方式，实现对违章车辆的全景及车牌特写记录，其具体功能如下：

- 1、利用动态视频检测触发技术，对闯红灯的车辆进行抓拍和车牌识别，准确的记录并储存违章车辆的时间、地点、行驶方向、违章车牌等信息 。
- 2、将抓拍到的违章车牌上传到指挥中心（物联网平台），存储到违章号码牌库中。供违章处理操作员进一步确认和处理。

可能需要的传感器：视觉传感器、RGB 灯模块、压力传感器。

任务记录

任务已经明确好了，来开始完成任务吧。

STEP 1 召开团队会议，确定项目方案与硬件清单。并绘制硬件连线图

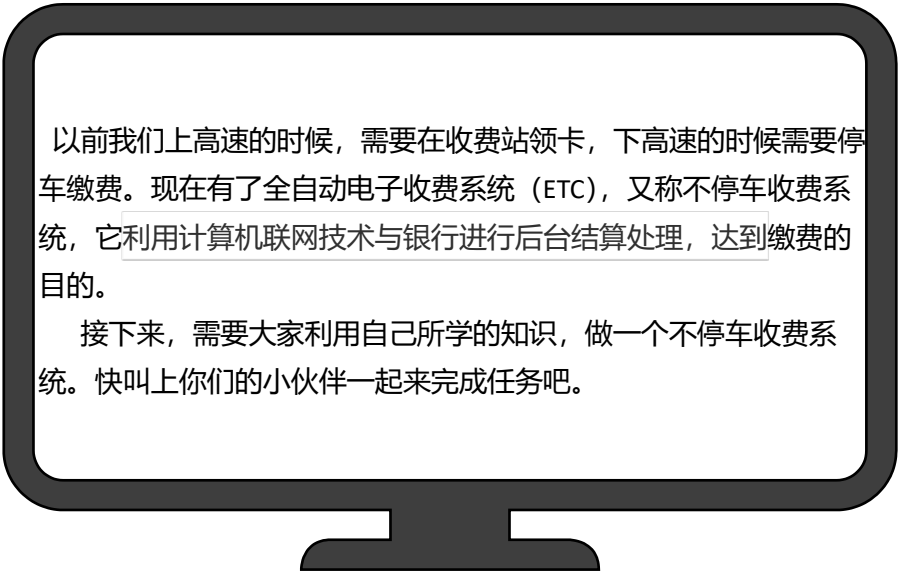
确定方案	简单描述：
硬件清单	所需要的硬件：
硬件连线图	简单绘制：

STEP 2 经过测试，请记录下，可实现的任务实现过程与实验数据和效果

实现过程	怎样实现这个方案，在此过程中遇见什么问题。
实验数据	主要填写 easy IoT 接收或者发送的数据。 违规车牌：
效果展示	主要描述该项目最后的实验效果（比如：共通过 10 辆车，检测出 3 辆车违规，车牌号是多少？其系统的准确率达到多少？）。

基础任务二 不停车收费系统

任务发布



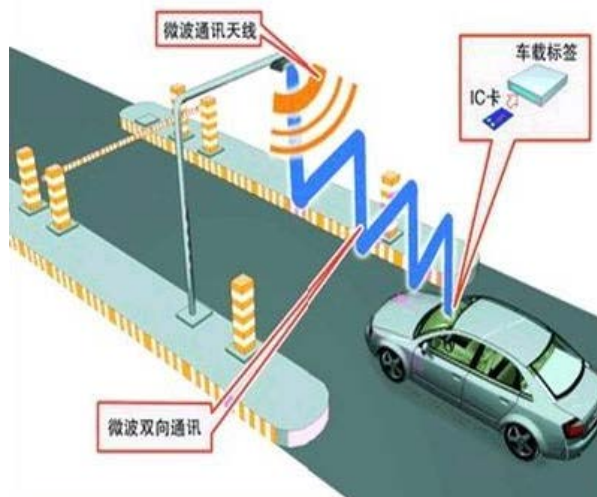
2 人或者 3 人为一组，自由分配任务，以套件里的传感器为基础（套件里没有的传感器，可以自己准备）完成任务。

任务背景	ETC 是国际上正在努力开发并推广的一种用于公路、大桥和隧道的电子自动收费系统。截至 2019 年 6 月 5 日 15 时，河北、广西、安徽等 29 个省区市已研究起草 ETC 电子不停车快捷收费系统推广发行方案，海南、西藏 2 个省份正在起草中，意味着 ETC 的推广应用正迈向全覆盖。
问题	现在套件里的传感器有限，能实现的功能有限。
任务分解	任务 1、设计一个可行性方案，保证不停车收费系统能够正常工作。 任务 2、结合物联网平台 Easy IoT。 任务 3、需要额外准备套件里没有的传感器。

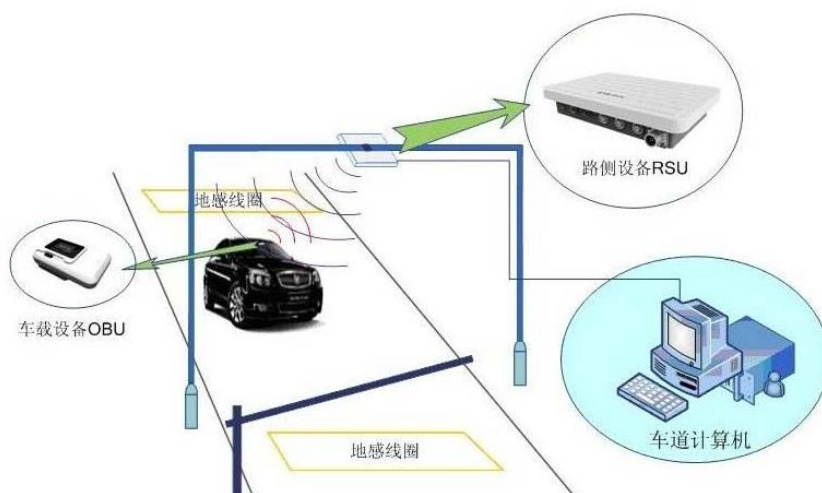
技术手册

ETC 系统是采用车辆自动识别技术完成车辆与收费站之间的无线数据通讯，进行车辆自动感应识别和相关收费数据的交换。采用计算机网络进行收费数据的处理，实现不停车、不设收费窗口也能实现全自动电子收费系统。

ETC 系统通过安装于车辆上的车载装置和安装在收费站车道上的天线之间进行无线通信和信息交换。主要由车辆自动识别系统、中心管理系统和其他辅助设施等组成。



车辆自动识别系统有车载单元又称应答器或电子标签、路边单元(Road side unit, RSU)、环路感应器等组成。OBU 中存有车辆的识别信息，一般安装于车辆前面的挡风玻璃上，RSU 安装于收费站旁边，环路感应器安装于车道地面下。中心管理系统有大型的数据库，存储大量注册车辆和用户的信息。当车辆通过收费站口时，环路感应器感知车辆，RSU 发出询问信号，OBU 做出响应，并进行双向通信和数据交换；中心管理系统获取车辆识别信息，如汽车 ID 号、车型等信息和数据库中相应信息进行比较判断，根据不同情况来控制管理系统产生不同的动作，如计算机收费管理系统从该车的预付款项账户中扣除此次应交的过路费，或送出指令给其它辅助设施工作。



可能需要的传感器：超声波传感器，舵机。

任务记录

任务已经明确好了，来开始完成任务吧。

STEP 1 召开团队会议，确定项目方案与硬件清单。并绘制硬件连线图

确定方案	确定不停车收费系统的方案（简单描述）：
------	---------------------

硬件清单	所需要的硬件：
硬件连线图	简单绘制：

STEP 2 经过测试，请记录下，可实现的任务实现过程与实验数据和效果

实现过程	怎样实现这个方案，在此过程中遇见什么问题。
实验数据	Easy IoT 发送数据： 实验反馈：
效果展示	主要描述该项目最后的实验效果。

第五章 智慧农业

在传统农业中，农民全靠经验来给农作物浇水、施肥、打药，如果不小心误判了，可能会导致颗粒无收。但是现在随着物联网的发展，产生了智慧农业。智慧农业的设备会用很精确的数据告诉我们农作物的浇水量，施肥、喷药的精确浓度，还有需要供给的温度、光照、二氧化碳浓度等信息。在以前用经验和感觉处理的问题，在现在都由信息化智能监控系统实时定量精确的把关。



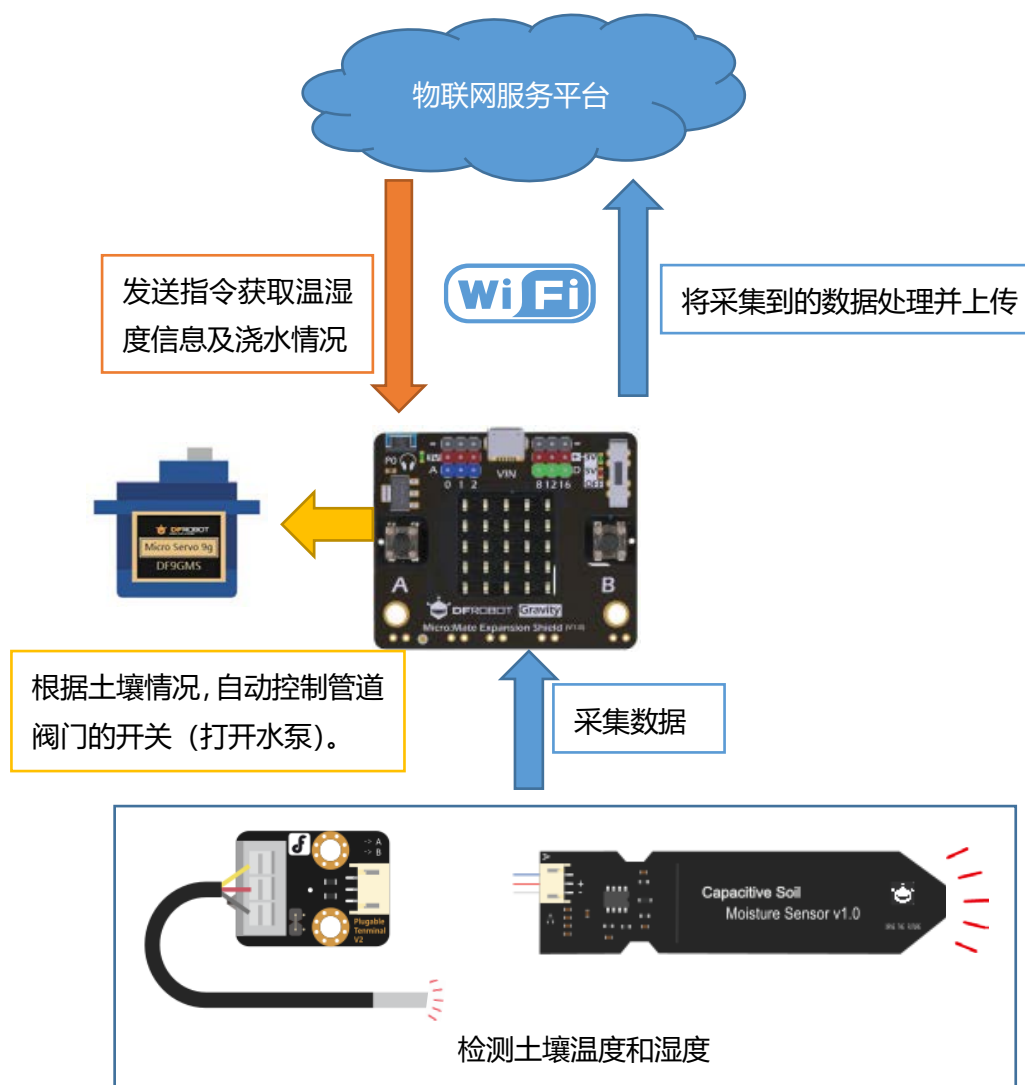
第一课 智能灌溉系统

智能灌溉系统是一种基于物联网的智能灌溉管理系统。我们都知道每种植物都有适合其生长的土壤湿度和土壤温度。湿度或者温度过高，植物的根系就会在土壤中腐烂；湿度过小，就不足以满足植物生长所需的水分。灌溉系统就是大幅度的满足了土壤的湿度在适宜植物生长的湿度范围之内。



学前提示

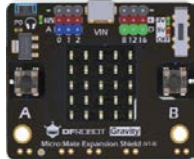
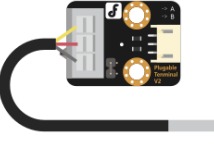
智能灌溉系统是通过土壤湿度传感器和防水温度传感器采集土壤的湿度和温度，并将数据传送给物联网平台（Easy IoT），我们可以发送指令查看数据。



智能灌溉系统的项目结构分析图

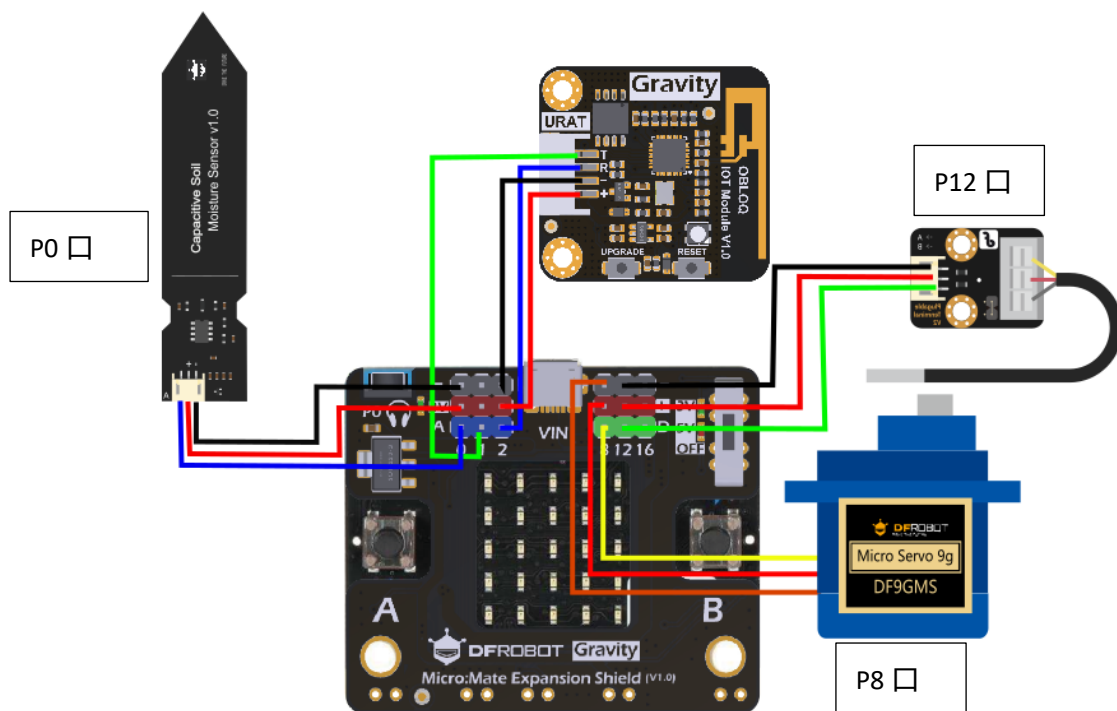
硬件清单及连接

硬件清单

			
1 块 Micro:bit 主板 +1 块 Micro:mate 扩展板	OBLOQ 模块	防水温度传感器	电容式土壤湿度传感器
			
舵机	电脑（需要接入 互联网）	USB 充电宝 （备选供电）	

硬件连接

OBLOQ 模块连接的方法与前面一样。将土壤湿度传感器连接在扩展板上的 P0 口，其中蓝色的线连接蓝色的针脚，红色的线连接红色针脚，黑色的线连接黑色针脚。将防水温度传感器连接在 P12 口，其中绿色的线连接绿色的针脚，红色的线连接红色针脚，黑色的线连接黑色针脚。将舵机连接在 P8 口，其中绿色的线连接绿色针脚，红色的线连接红色针脚，棕色的线连接黑色针脚。



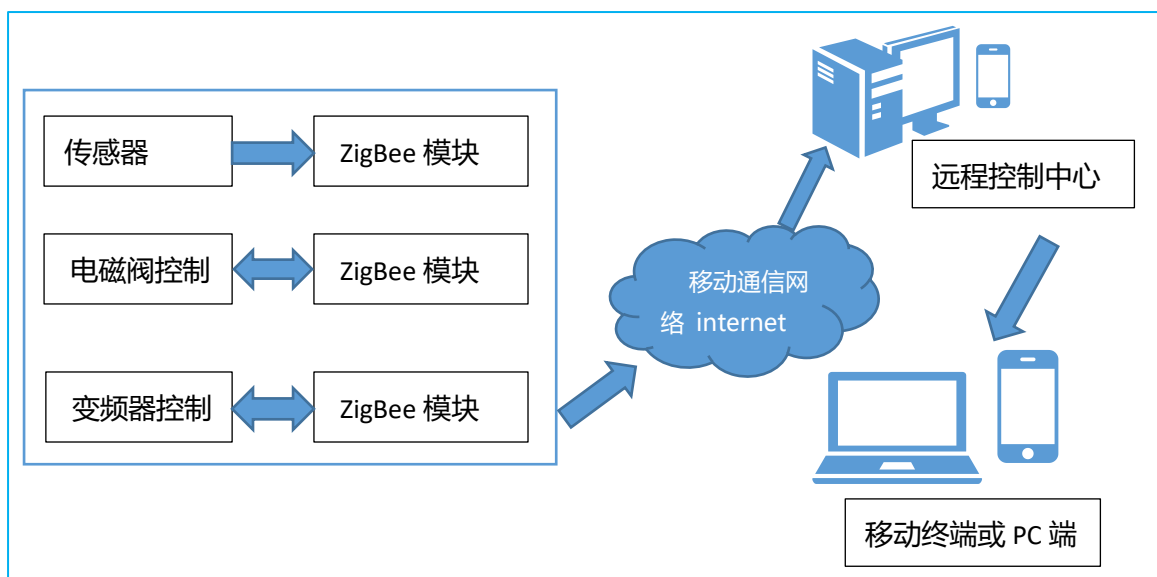
电路连接

知识点解析

基于物联网的智能灌溉系统

智能灌溉系统是国家农业信息化工程技术研究中心自主研发的集自动控制技术和专家系统技术，传感器技术、通讯技术、计算机技术等于一体的灌溉管理系统。

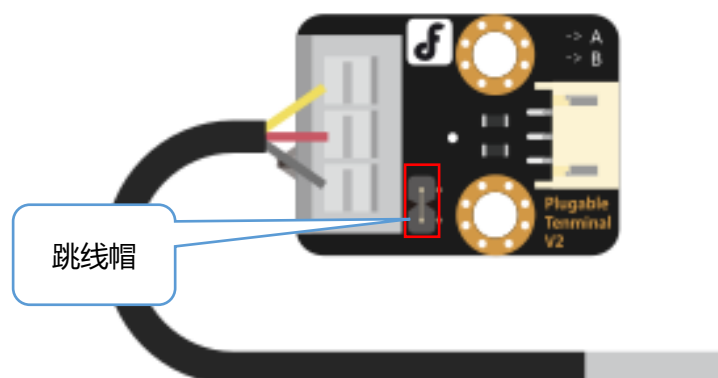
在灌溉系统中分布了多个传感器节点、电磁阀控制节点和变频器控制节点。土壤环境、空气环境参数都是有传感器节点采集，灌溉管道是由电磁阀控制节点实施开关控制。变频器控制节点则是用来实现全自动变频恒压供水。控制器节点通过 ZigBee 网络采集环境传感器节点、电磁阀控制节点和变频器控制节点信息，然后将现场采集的信息通过移动互联网和 internet 互联网有控制节点发送给远程监控中心，移动终端或者 PC 端接收远程监控中心传送过来的信息。这样就可以对智能灌溉系统进行实时监控。



智能灌溉系统架构图

防水温度传感器

防水温度传感器可以用到很多地方，比如土壤温度检测、热水箱温度控制等等。此款温度传感器，我们还额外设计了 Plugable Terminal 转接器，但想要运用此转接器，我们必须在温度传感器上加上拉电阻，使用跳线帽切换使用（将跳线帽放在 PULL UP 上）。



电压范围为 3.0 V 至 5.5 V，无需备用电源，其测量温度范围为 -55°C 至 $+125^{\circ}\text{C}$ ，显示为 -10°C 至 $+85^{\circ}\text{C}$ 的温度范围内精度更高达 $\pm 0.5^{\circ}\text{C}$ 。

3 线接口：分为 A 型和 B 型，接线之前请注意您的传感器接口。

类型 A：红线（VCC），黑线（GND），黄线（DATA）。

类型 B：红线（VCC），黄线（GND），绿线（DATA）。

注意：温度传感器具有负压特性，则是电源极性接反时，芯片不会因发热而烧毁，但不能正常工作。实现多点测温，最多只能并联 8 个，如果数量过多，会使供电电源电压过低，从而造成信号传输的不稳定。

指令提前知

模块	功能说明
	读取防水温度传感器采集到的温度
	读取模拟引脚（程序中用来读取土壤湿度的数据）
	与运算，左右两边的条件需要同时满足

程序编写

1. 添加主控板及物联网模块

根据第一章相关内容添加主板——micro:bit 和物联网模块——OBLOQ。

2. 添加传感器和执行器

- 根据下图所示步骤添加防水温度传感器

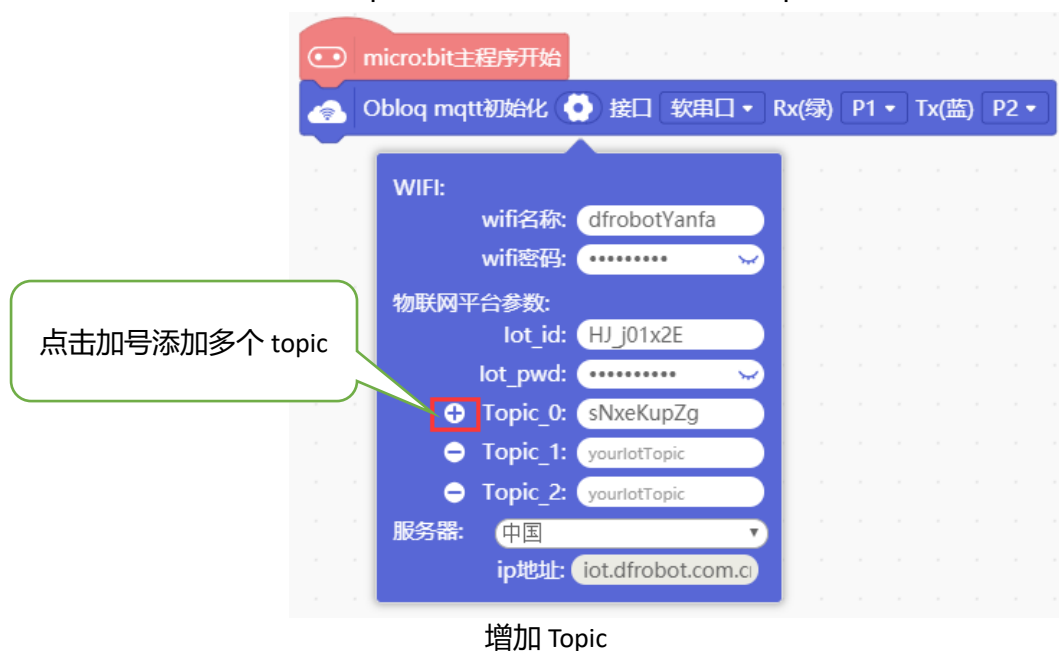


- 根据第二课云养猫之猫粮机的相关内容添加执行器——舵机。

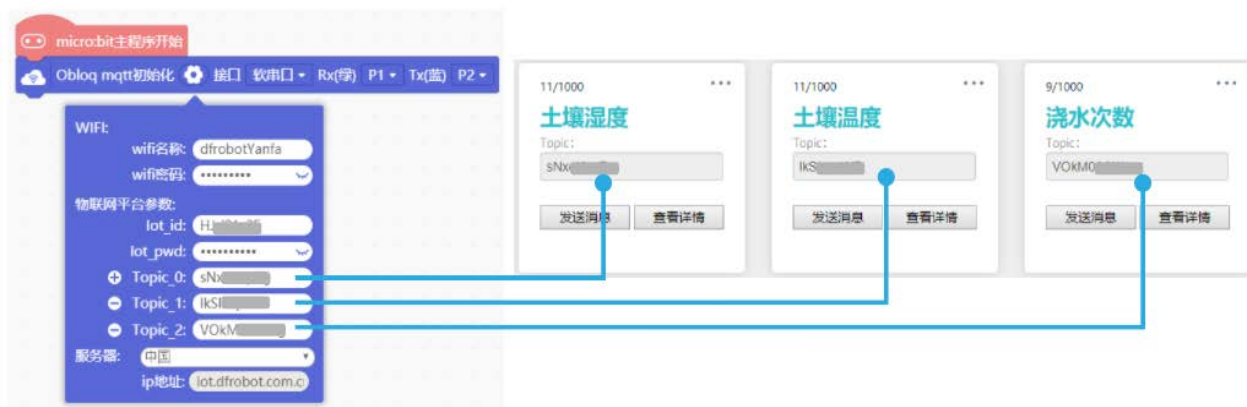
3. 物联网模块初始化及联网测试

- 将初始化模块 “Obloq mqtt 初始化 接口 硬件口 Rx(绿) P0 Tx(蓝) P1” 拖放至右边的编辑区。根据前面学习的内容定义发送和接收的引脚（要和实际连线一致）以及设置网页终端的相关账号与密码。

- 因为该项目我们需要用到 3 个 Topic，所以我们还需要增加两个 Topic。



- 添加 Easy IoT 中的 Topic 参数。



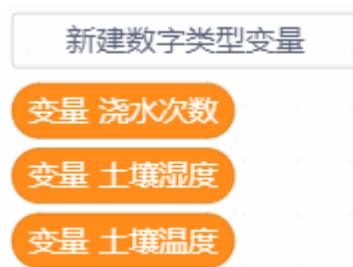
4.初始化舵机角度

- 设置好物联网的相关信息后，将舵机指令 “设置 P9 引脚伺服舵机为 90 度” 拖动到如图所示位置，初始化舵机。将舵机的引脚设置成 P8，初始化角度为 10 度，表示水阀的阀门处于关闭状态。



5.创建三个变量

- 创建三个变量，分别存放土壤湿度、土壤温度、浇水次数等数据。



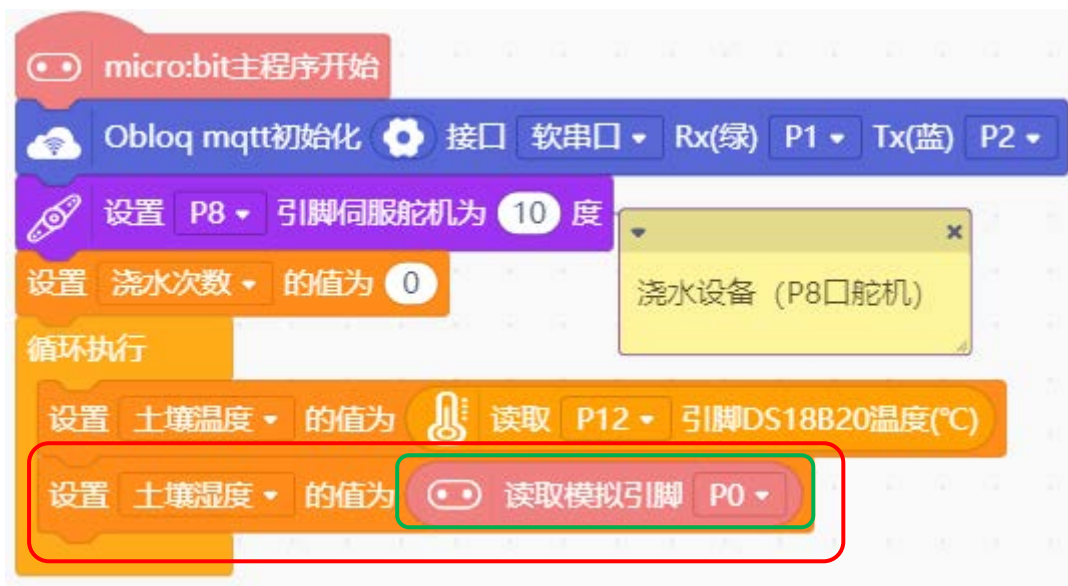
- 初始化“变量 浇水次数”，将浇水次数的值设置为 0。



- 设置变量“变量 土壤温度”，将传感器中的模块“读取 P3 引脚DS18B20温度(°C)”拖放到如图所示位置，这样就可以将读取到的温度数据存放在变量“变量 土壤温度”里。



- 设置变量“变量 土壤湿度”，将模块“读取模拟引脚 P0”拖放到如图所示位置，这样就可以将读取到的湿度数据存放在变量“变量 土壤湿度”里。。



6.控制舵机

- 现在我们已经获取到土壤的湿度和温度，接下来我们就可以利用条件判断语句控制舵，如果土壤温度 > 26 度，土壤湿度 > 520 时，说明土壤温度过高或者土壤干燥，这时舵机转动到 100 度打开阀门，实现灌溉的功能；否则舵机还是在初始化的角度，表示阀门还是在关闭状态。



提示：

土壤温度的范围可以根据实际情况设置，项目中是以玉米、棉花（适宜温度在 25℃左右）作为参照而设置的。

由于土壤湿度传感器数值会受到入土深度、土壤松紧度的影响，只能检测到土壤的相对湿度，我们把湿度的范围分为三等分，分别表示干燥、湿润、非常潮湿。例如：空气中读数为 520，水中读数为 260，这样就可以分为 (520,430],[430,350],[350,260]这三个区间分别代表干燥、湿润、非常潮湿。

7.查看相关数据

- 接下来我们就将获取到的数据上传到 Easy IoT 平台上，这样方便我们查看土壤的具体情况。

- 将 “” 模块拖放至编辑区。然后设置条件判断，当 Obloq 消息收到 “土壤湿度” 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，因为需要做数据分析，所以设置了当接收到消息后，发送 10 条土壤湿度的数据给 Easy IoT。程序内容如下图所示：

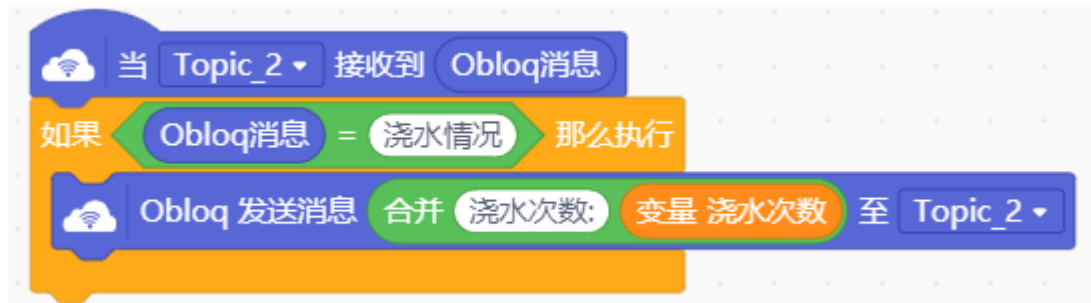


- 接下来我们继续将 “” 模块拖放至编辑区，将 “Topic_0” 修改成 “Topic_1”。用来反馈土壤温度的数据。当 Obloq 消息收到 “土壤温度” 这个消息，就发送温度数据到 Easy IoT。



- 智能灌溉系统除了可以利用 Easy IoT 查看土壤湿度和土壤温度。还可以用它查看浇水的次数。将

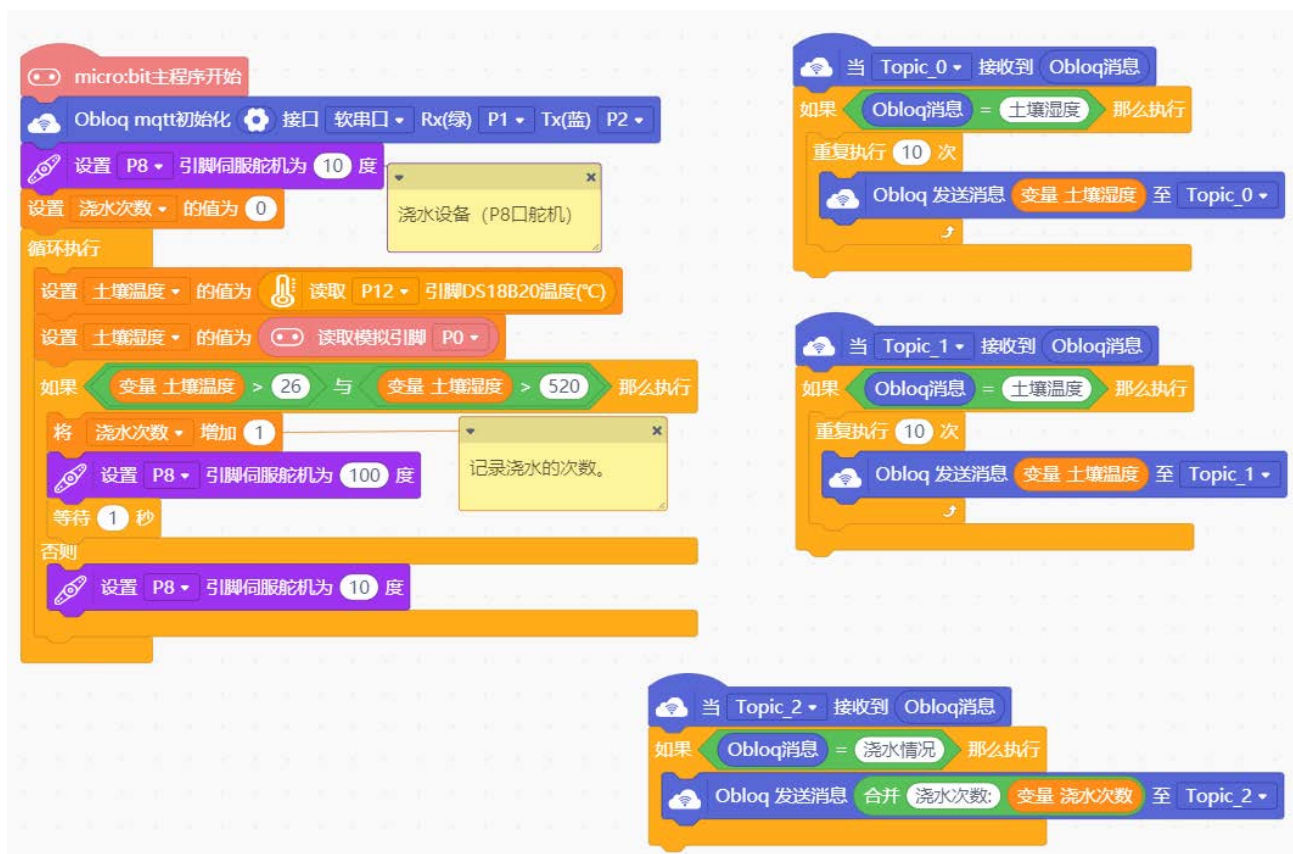
“”模块拖放至编辑区，将“Topic_0”修改成“Topic_2”。用来反馈浇水的次数。当 Obloq 消息收到“浇水情况”这个消息，就发送浇水的次数到 Easy IoT。



参考程序

程序展示

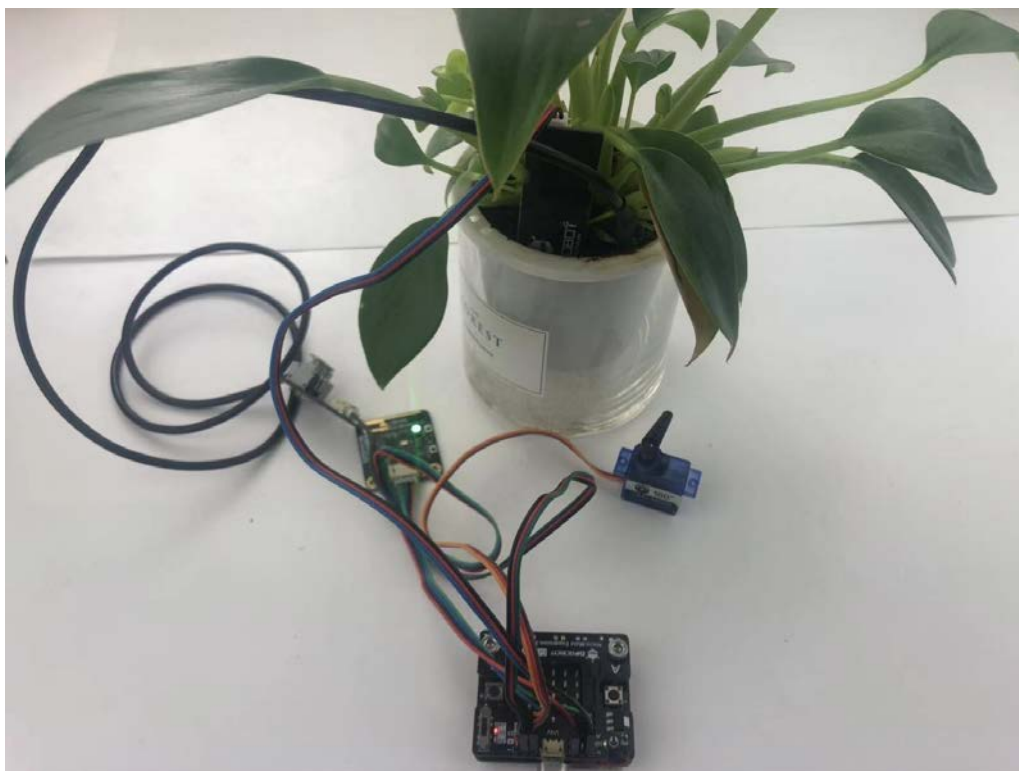
将下面的程序下载到设备中。



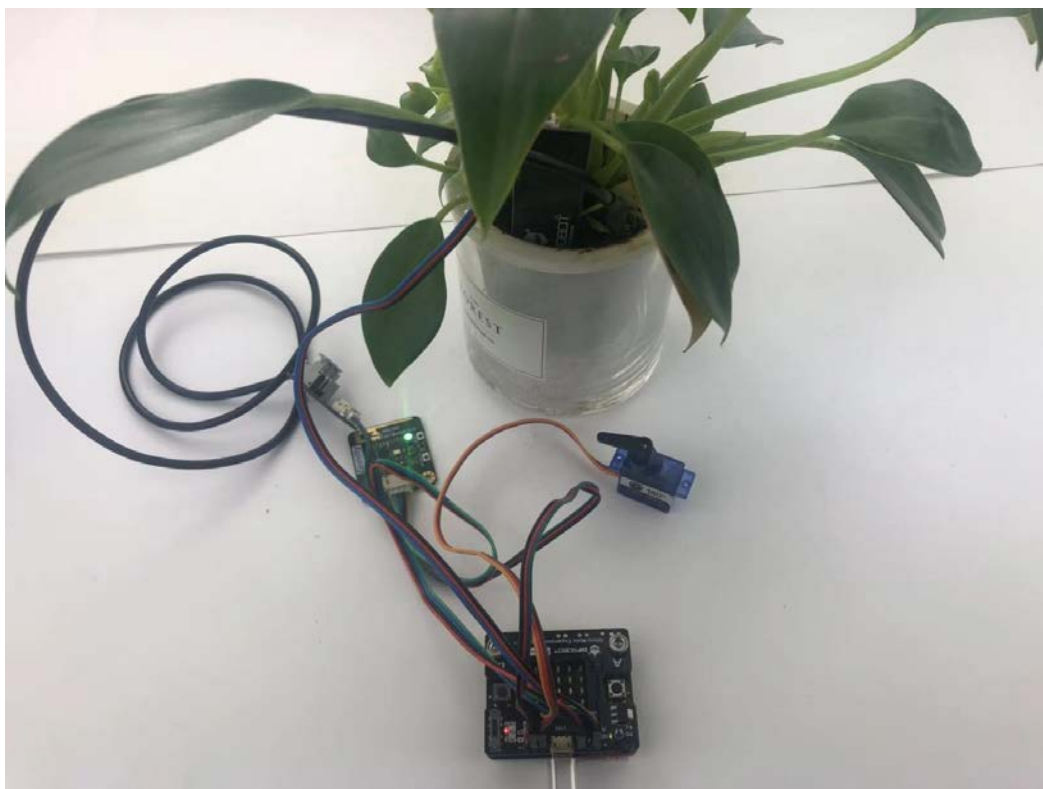
注意:将程序上传至设备后，需要将连在主控上面的 USB 数据线拔下，连接到扩展板上的电源接口（VIN 口）。额外供电后，将板载上的调节开关调至 5V。

效果展示

- 当土壤温度 <26 ,湿度 <520 时。设备情况如下,舵机角度为 10 度。表示水泵的阀门处于关闭状态。



- 当土壤温度 >26 ,湿度 >520 时。设备情况如下,舵机角度为 100 度。表示水泵的阀门处于打开状态。



- 进入 Easy IoT 网页发送信息给 micro:bit 设备。找到相应的 Topic, 如: 找到土壤湿度, 点击 “发送消息”, 发送指令 “土壤湿度”

←回到我的工作间

土壤湿度

✈ 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"

土壤湿度

发送

- 刷新网页可以查看土壤湿度。

时间	消息	操作
2019/10/30 15:16:45	土壤湿度	
2019/10/30 15:16:47	481	
2019/10/30 15:16:48	481	
2019/10/30 15:16:50	481	

上拉网页，找到查看消息项，点击“Date rang”，选择 1day。选择完毕，点击查看。

✉ 查看消息

Date range

1day

7day

30day

最新消息

✉ 查看消息

Date range

2019/10/29 20:00

-

2019/10/30 20:00

查看

就能看见土壤湿度的折线图表。



- 找到土壤温度这个 Topic，点击“发送消息”，发送指令“土壤温度”

土壤温度

发送新消息

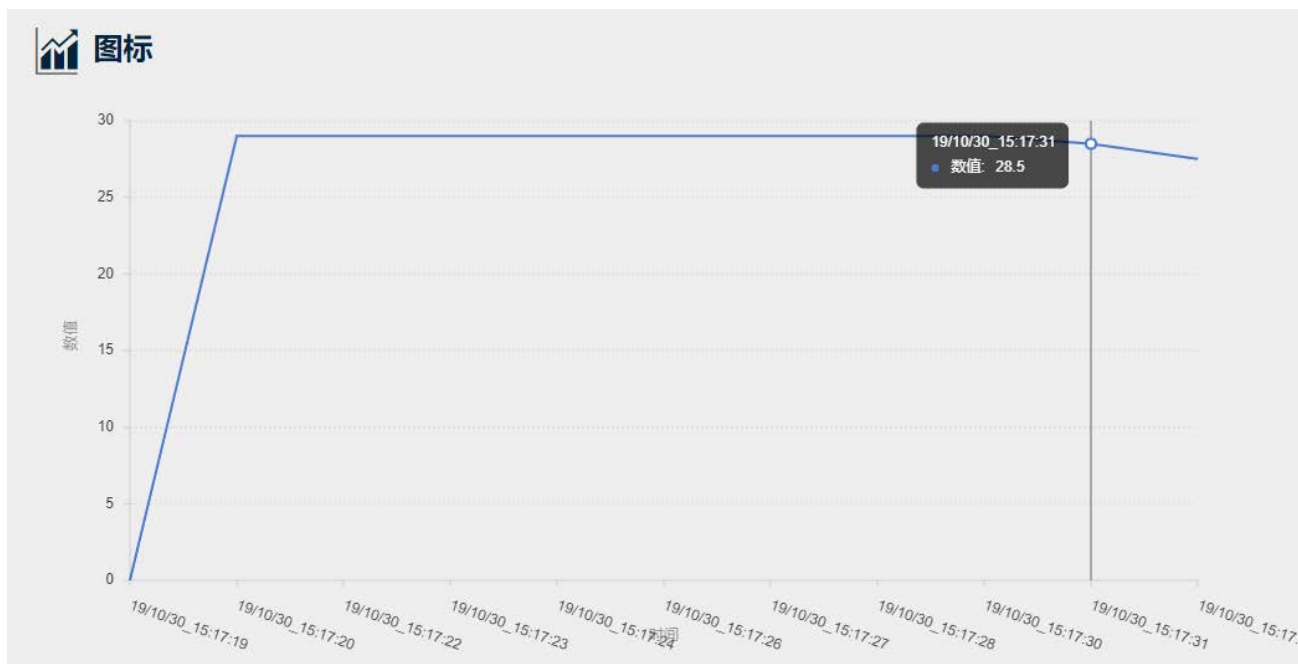
为消息加上->前缀代表此消息为纯指令消息，不会被存入数据库。例如"->off"

- 刷新网页可以查看土壤温度。

查询结果

时间	消息	操作
2019/10/30 15:17:19	土壤温度	
2019/10/30 15:17:20	29	
2019/10/30 15:17:22	29	
2019/10/30 15:17:23	29	

上拉网页，找到查看消息项，点击“Date rang”，选择 1day。选择完毕，点击查看（步骤和查看土壤湿度的折线图一样），就能看见土壤温度的折线图表。



- 找到浇水次数这个 Topic，点击“发送消息”，发送指令“浇水情况”

浇水次数

✈ 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”

浇水情况

发送

- 刷新网页可以查看浇水次数。

📧 最新消息

时间	消息
2019/10/30 15:26:15	浇水次数:2.00000
2019/10/30 15:26:15	浇水情况

拓展训练

在项目中，我们利用舵机当电磁阀门控制水泵的开关，那我们能不能升级一下硬件，做一个更完美的智能灌溉系统呢？能否根据我下面的提示完成系统升级呢？

提示：首先我们需要准备数字继电器，电磁阀门或者水泵。

数字继电器

在做项目的时候，很多大电流或高压的设备无法直接用数字 IO 口控制（如电磁阀、电机，电灯），这时我们就可以考虑用继电器来解决。

继电器模块接线端字符含义：

NC 表示常闭；

NO 表示常开；

N/A 表示空脚；

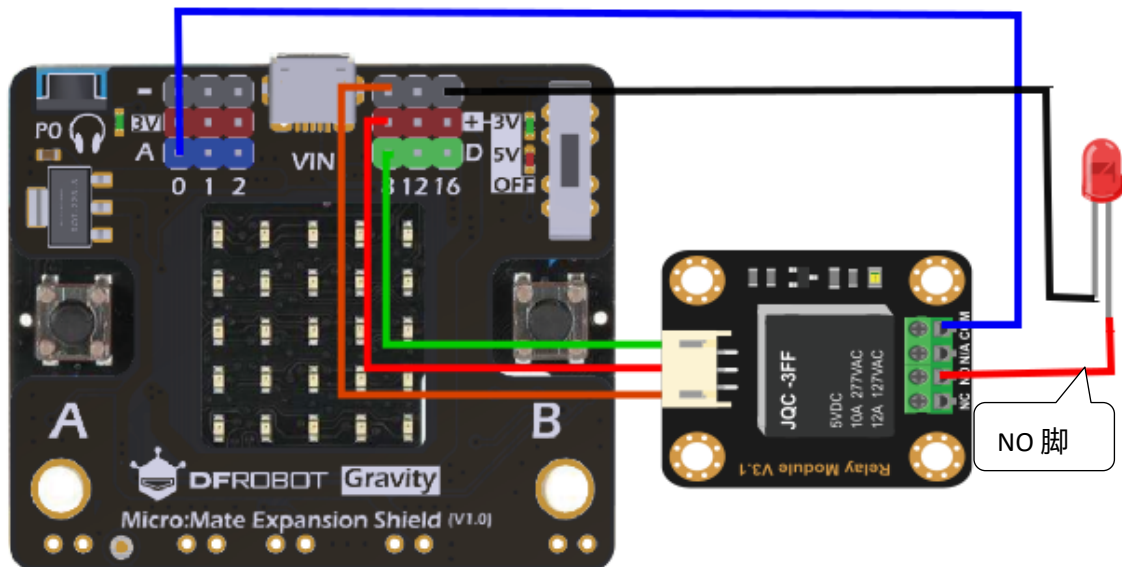
COM 表示公共端；

实例：用继电器控制 LED 灯。

继电器模块控制端接线：接 P8 引脚。

继电器模块继电器端接线：COM 端接 P0 蓝色针脚，NO 端接到 LED 的正极 ((LED 长脚)，， LED 的负极 (LED 短脚)接到 P0 脚的黑色针脚 (GND)。

工作原理：控制器没有给控制信号时，LED 灯呈熄灭状态，继电器触点处于常闭状态，NC 和 COM 保持导通，NO 断开状态，当控制器给高电平控制信号后，继电器触点吸合，NO 和 COM 保持导通，NC 断开状态，此时数字口 P0 脚输出高电平，然后通过 COM 口导通到 NO 口，给 LED 正极供电，LED 就亮了。硬件连线图如下：



代码如下：



提示：如何添加继电器模块



添加执行器——继电器

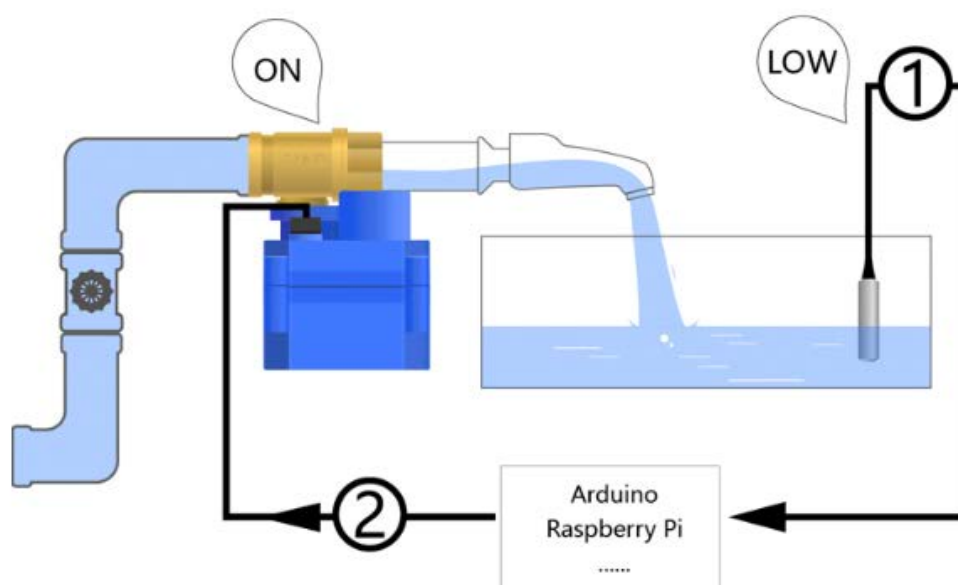
电磁水阀

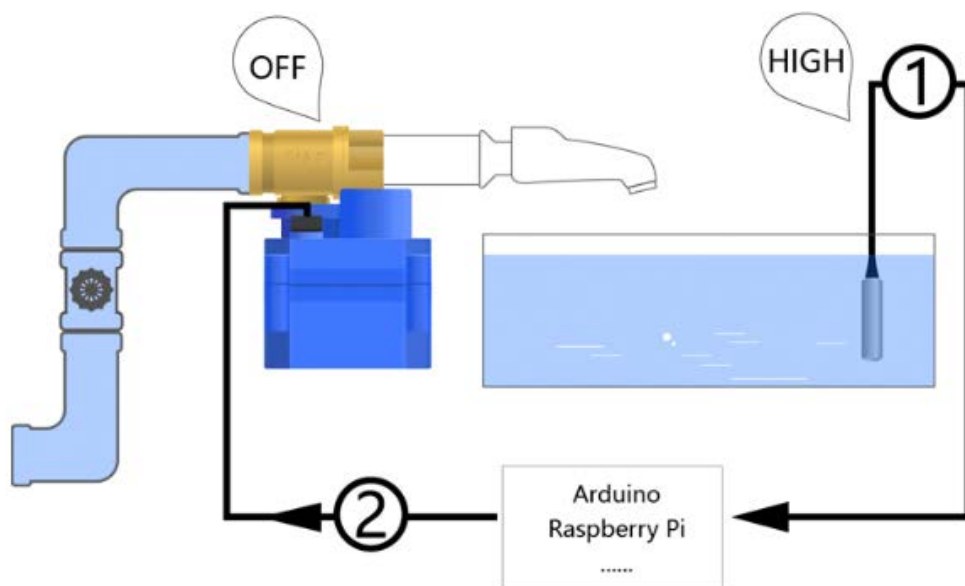
电磁水阀搭配继电器使用，可实现对阀门的控制。比如定时开关阀门，可以广泛应用于自动浇花、鱼缸养殖等需要对水量进行精确控制的项目以及智能家居

电磁水阀引线参数定义：

- 蓝线控制阀门打开
- 红线控制阀门关闭
- 黄线为共有地线

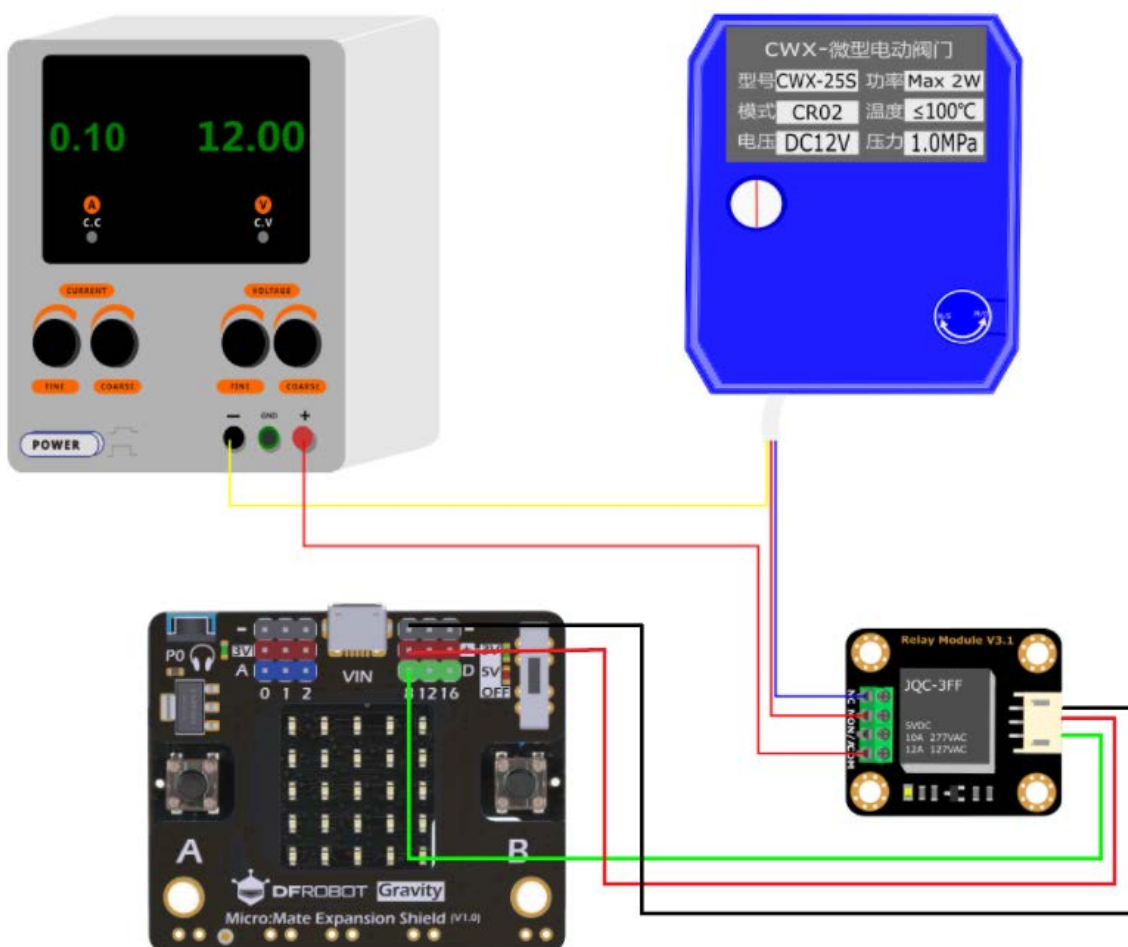
实例：自动蓄水系统





继电器模块控制端接线：接 P8 引脚。

继电器模块继电器端接线：COM 端接到外接电源正极，NO 端接到电磁阀的红线，NC 端接到电磁阀的蓝线；电磁阀的黄线与外接电源负极相连；阀门上方圆圈那个代表的是阀门开关状态的一个指示，竖着代表阀门关闭，横着代表阀门打开；下面圆圈代表的是阀门的一个手动开关旋钮，将其提起来旋转就能手动控制阀门开关。电磁阀与 micro:bit 的连线如下图：



根据上面的提示，你是否可以试一试利用继电器和电磁水阀升级我们的智能灌溉系统呢？
需要实现的效果是当土壤温度>26 度，土壤湿度>520 时，打开水阀，实现灌溉的功能；否则关闭阀门。

提示：需要修改关于舵机的程序。

```
if (variable: soil temperature > 26 and variable: soil humidity > 520) then execute:
  increase watering count by 1
  set servo motor P8 to 100 degrees
  wait 1 second
  record watering count.
else:
  set servo motor P8 to 10 degrees
```

拓展阅读

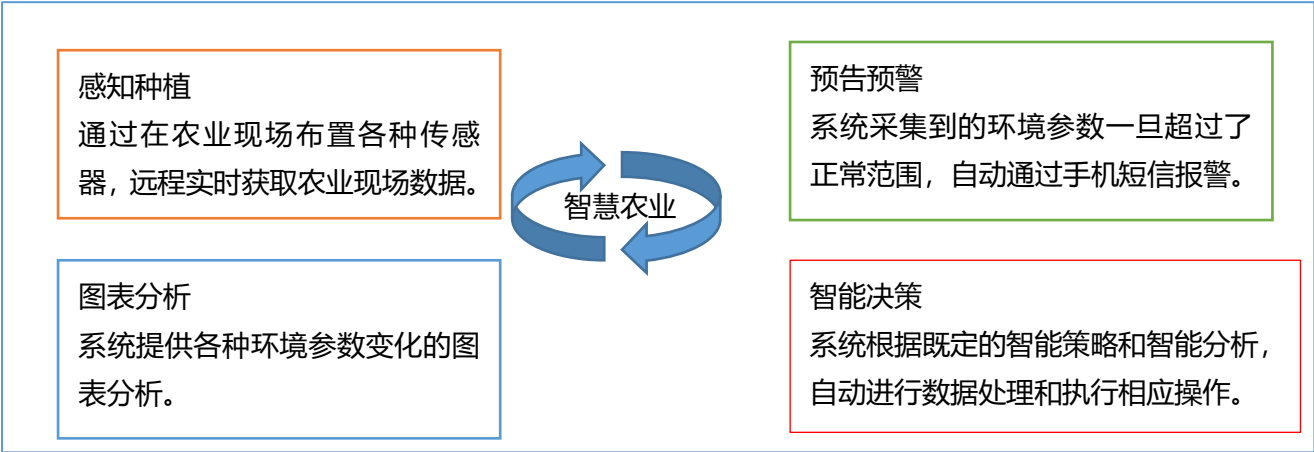
智能农业的特点

物联网在农业领域中有着广泛的应用，从农业产品生产的不同阶段来看，无论是从种植的培育阶段还是收获阶段，都可以用物联网技术来提高工作效率。

农作物生长阶段	运用简介
种植准备阶段	在温室里面布置很多传感器，通过分析实时的土壤信息来选择合适的农作物。
种植和培育阶段	利用物联网技术手段采集温度，湿度信息，进行高效的管理，从而应对环境的变化。
收获阶段	利用物联网信息，将各个阶段的性能进行采集，反馈到前段，从而在农作物种植收获阶段进行精准的测算。

物联网技术在农作物生长中的运用

在一块几千亩的农场里面，需要对大棚进行浇水，施肥，手工测温，手工卷帘，如果没有物联网技术的加入，就需要大量的时间和人员来操作。如果利用物联网技术，只需要几秒钟的鼠标操作就能完成繁琐的人工操作了。



智慧农业功能

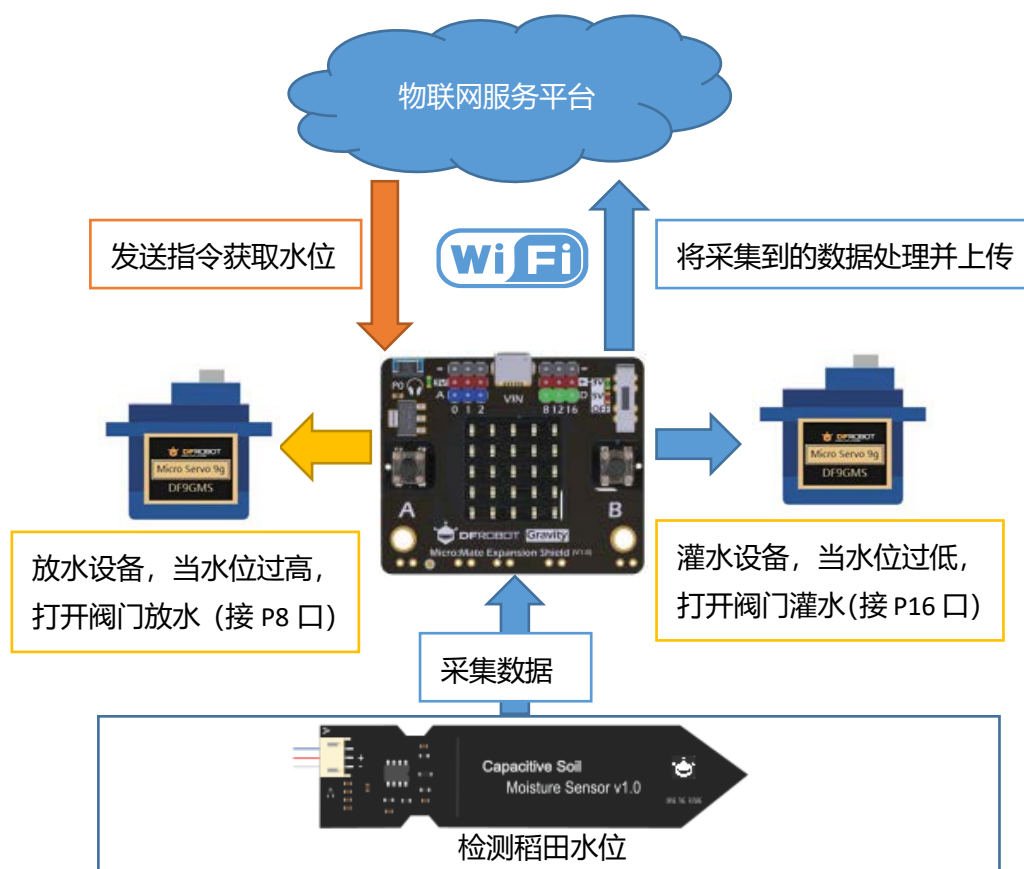
第二课 智能水稻种植

我国自古以来都是以种植水稻为主的农业大国，种植面积大约在 4.3~4.4 亿亩之间，水稻种植的方式也从原始人畜耕作到机械耕种，再到新型物联网种植。物联网新型种植方式利用先进的技术和现代化设施，人为控制农作物生长的环境条件，使农作物生长不受自然气候的影响，做到高效率、高产值和高效益的生产。



学前提示

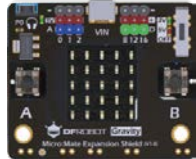
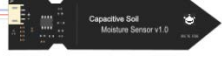
智能水稻养殖系统，根据水稻不同的时期，调节水位。比如水稻在生长期，需要高水位，水稻在成熟期，需要低水位。所以我们可以根据水稻的生长周期，调节水位。同时我们也能通过 Easy IoT 发送指令，获取稻田当前的水位。



智能水稻养殖的项目结构分析图

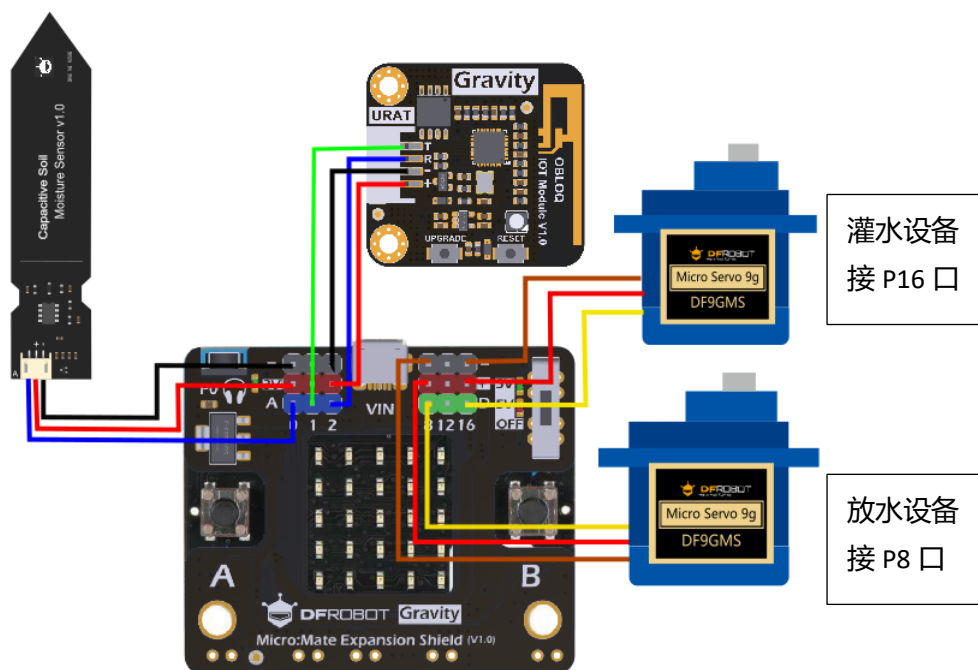
硬件清单及连接

硬件清单

		
1 块 Micro:bit 主板+1 块 Micro:mate 扩展板	OBLOQ 模块	电容式土壤湿度传感器
		
舵机	电脑（需要接入互联网）	USB 充电宝（备选供电）

硬件连接

OBLOQ 模块中的 T 引脚（绿线）连接扩展板上的 P1 针脚（蓝色针脚），OBLOQ 模块中的 R 引脚（蓝线）连接扩展板上的 P2 针脚（蓝色针脚），OBLOQ 模块中的“-”引脚（黑线）连接扩展板上的黑色针脚，OBLOQ 模块中的“+”引脚（红线）连接扩展板上的红色针脚。将土壤湿度传感器连接在扩展板上的 P0 针脚，其中蓝色的线连接蓝色的针脚，红色的线连接红色针脚，黑色的线连接黑色针脚。将灌水设备的舵机模块连接在扩展板上的 P16 脚，将放水设备的舵机模块连接在扩展板上的 P8 脚，其中棕色的线连接黑色针脚，红色的线连接红色针脚，黄色的线连接绿色针脚。



智能水稻养殖电路连接

水稻智慧化种植

1、种植系统功能

可根据不同区域的土壤类型、种植作物、灌溉水源及灌溉方式等划分。基于农业物联网的大田种植智能管理系统针对农业大田种植分布广、监测点多、布线和供电困难等特点，利用农业物联网技术，采用高精度土壤温湿度传感器和智能气象站，远程在线采集土壤湿度、酸碱度、养分、气象信息等，实现土壤湿度自动预报、灌溉用水量智能决策、远程、自动控制灌溉设备等功能，最终达到精耕细作、准确施肥、合理灌溉的目的。

该农业物联网种植系统根据不同地域的土壤类型、灌溉水源、灌溉方式、种植作物等划分不同类型区，在不同类型区内选择代表性的地块，建设具有土壤含水量，地下水位，降雨量等信息自动采集、传输功能的监测点。通过灌溉预报软件结合信息实时监测系统，获得作物最佳灌溉时间、灌溉水量及需采取的节水措施为主要内容的灌溉预报结果，定期向群众发布，科学指导农民实时实量灌溉，达到节水目的。

2、种植系统构架

1. 地面信息采集

1) 使用地面温度、湿度、光照、光合有效辐射传感器采集信息可以及时掌握水稻生长情况。

2) 使用雨量、风速、风向、气压传感器可收集大量气象信息，当这些信息超出正常值范围，用户可及时采取防范措施，减轻自然灾害带来的损失。如：强降雨来临前，打开稻田蓄水口。

2. 地下或水下信息采集

1) 可实现地下或水下土壤温度、水分、水位、氮磷钾、溶氧、PH 值的信息采集。

2) 检测土壤温度、水分、水位是为了实现合理灌溉，杜绝水源浪费和大量灌溉导致的土壤养分流失。

3) 检测氮磷钾、溶氧、PH 值信息，是为了全面检测土壤养分含量，准确指导水田合理施肥，提高产量，避免由于过量施肥导致的环境问题。

3. 视频监控

视频监控系统是指安装摄像机通过同轴视频电缆将图像传输到控制主机，实时得到植物生长信息，在监控中心或异地互联网上即可随时看到作物的生长情况。

4. 水稻智慧化种植系统报警系统

用户可在主机系统上对每一个传感器设配设定合理范围，当地面、地下或水下信息超出设定范围时，报警系统可将田间信息通过手机短息和弹出到主机界面两种方式告知用户。用户可通过视频监控查看田间情况，然后采取合理方式应对田间具体发生状况。

3、系统配置构成

1、信息采集系统：

1) 地面配置：温度、湿度、光照、光合有效辐射、雨量、风速、风向、气压等设备。

2) 地下或水下配置：土壤温度、水分、水位、氮磷钾、溶氧、PH 值等设备。

- 2、无线传输系统：用于远程无线传输采集数据。
- 3、视频监控系统：查看作物实时生长情况。
- 4、软件平台：远程数据实时查看功能；自动化控制功能；各类预警功能。



程序编写

1.添加主控板及物联网模块

根据第一章相关内容添加主板——micro:bit 和物联网模块——OBLOQ。

2.添加传感器和执行器

- 根据第二课云养猫之猫粮机的相关内容添加执行器——舵机。

3.物联网模块初始化及联网测试

- 将初始化模块 “Obloq mqtt初始化 接口 硬串口 Rx(绿) P0 Tx(蓝) P1” 拖放至右边的编辑区。根据前面学习的内容定义发送和接收的引脚（要和实际连线一致）以及设置网页终端的相关账号与密码。

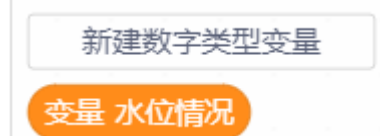
4.初始化舵机角度

- 设置好物联网的相关信息后，将舵机指令 “设置 P9 引脚伺服舵机为 90 度” 拖动到如图所示位置，初始化舵机。将舵机的引脚设置成 P8 和 P16，初始化角度为 10 度，表示放水设备和灌水设备处于关闭状态。



5.创建变量

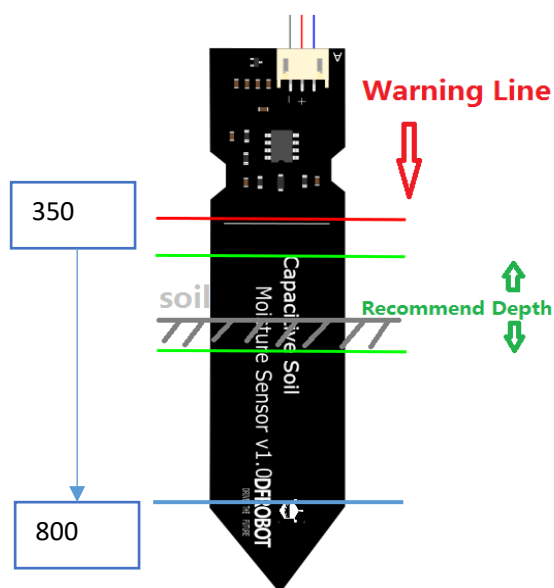
- 创建一个变量，用来表示水位情况。



- 设置变量 “变量 水位情况”，将模块 “读取模拟引脚 P0” 拖放到如图所示位置，这样就可以将读取到的数据存放在变量 “变量 水位情况” 里。。

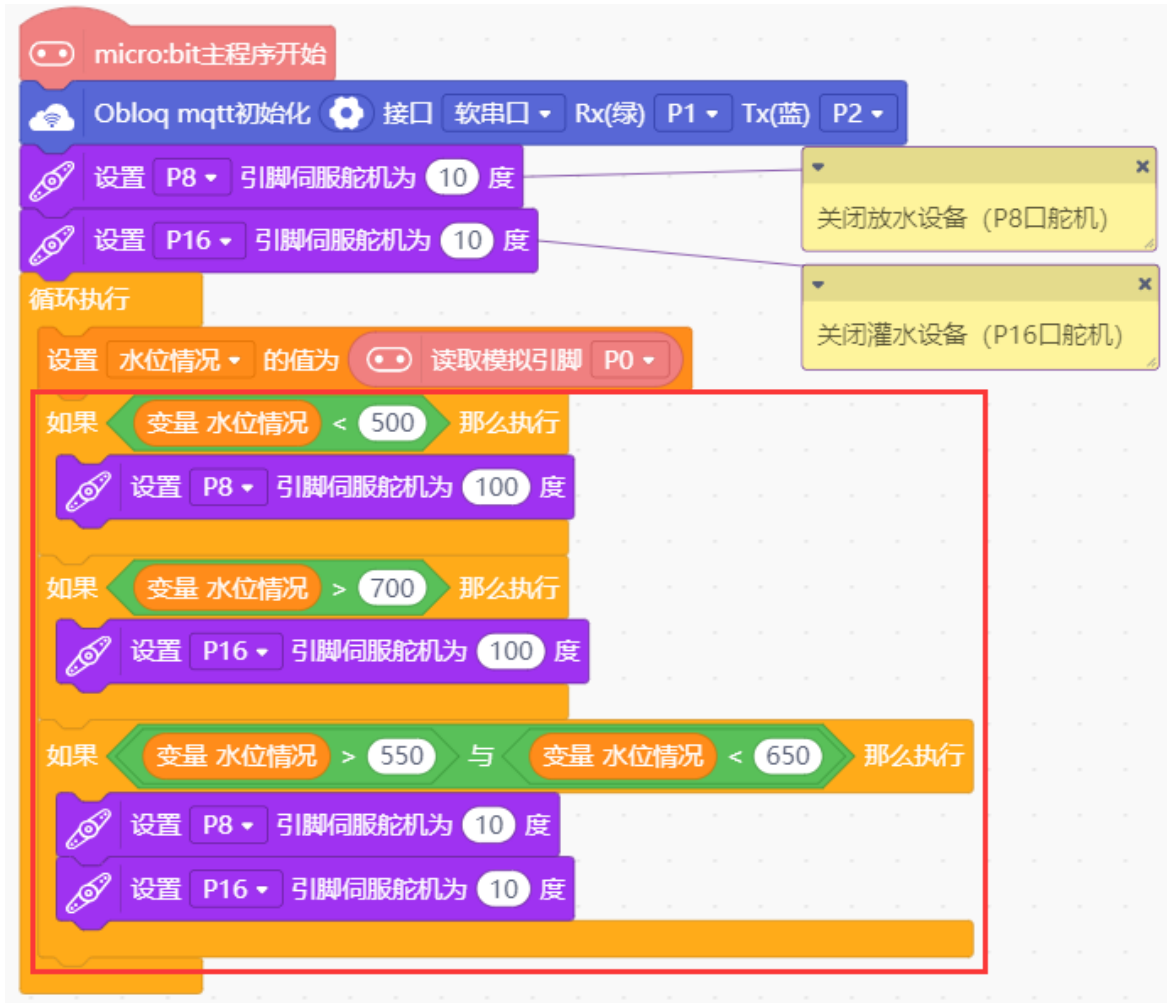


土壤湿度传感器输出的值与实际情况成反比，值越小表示水位越高（土壤湿润），值越大表示水位越低（土壤干燥）



6.控制舵机

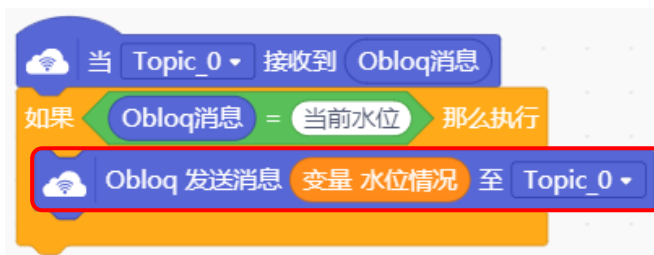
- 现在我们已经获取到水位情况，接下来我们就可以利用条件判断语句控制舵机，如果水位情况 < 500，说明水位过高，这时 P8 口的舵机转动到 100 度打开放水设备，将田里多余的水放出去；如果水位情况 > 700，说明水位过低，这时 P16 口的舵机转动到 100 度打开灌水设备，向田里灌水；如果 550 > 水位情况 < 650，说明水位合适，这时将舵机的角度都转动到 10 度，表示放水设备和灌水设备都处于关闭状态。



7.查看相关数据

- 接下来我们就将获取到的数据上传到 Easy IoT 平台上，这样方便我们查看水位的具体情况。

- 将 “当 Topic_0 接收到 Obloq 消息” 模块拖放至编辑区。然后设置条件判断，当 Obloq 消息收到 “当前水位” 这个消息，就将水位情况发送给物联网平台 (Easy IoT)。



参考程序

程序展示

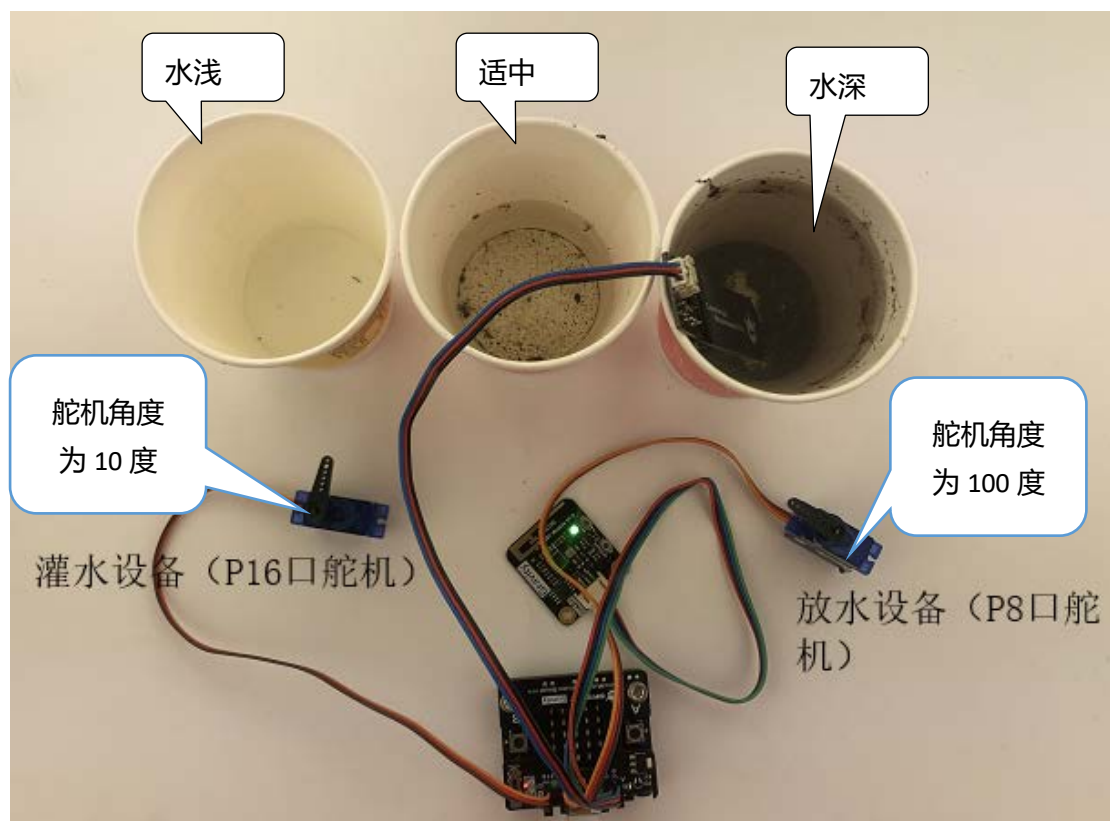
将下面的程序下载到设备中。

The program is written in a block-based language for a micro:bit. It starts with a 'micro:bit 主程序开始' (micro:bit Main Program Start) block. This is followed by an 'Obloq mqtt初始化' (Obloq mqtt Initialization) block, which is configured with '接口' (Interface) set to '软串口' (Soft Serial Port), 'Rx(绿)' (Rx) set to 'P1', and 'Tx(蓝)' (Tx) set to 'P2'. Below this, there are two '设置' (Set) blocks: '设置 P8 引脚伺服舵机为 10 度' (Set P8 pin servo motor to 10 degrees) and '设置 P16 引脚伺服舵机为 10 度' (Set P16 pin servo motor to 10 degrees). These two blocks are connected to two yellow callout boxes on the right: '关闭放水设备 (P8口舵机)' (Close water release device (P8 pin servo motor)) and '关闭灌水设备 (P16口舵机)' (Close water filling device (P16 pin servo motor)). The main logic is enclosed in a '循环执行' (Loop Execution) block. Inside the loop, the first block is '设置 水位情况 的值为 读取模拟引脚 P0' (Set water level status value to read analog pin P0). This is followed by three conditional blocks: 1. '如果 变量 水位情况 < 500 那么执行' (If variable water level status < 500 then execute), which contains '设置 P8 引脚伺服舵机为 100 度' (Set P8 pin servo motor to 100 degrees). 2. '如果 变量 水位情况 > 700 那么执行' (If variable water level status > 700 then execute), which contains '设置 P16 引脚伺服舵机为 100 度' (Set P16 pin servo motor to 100 degrees). 3. '如果 变量 水位情况 > 550 与 变量 水位情况 < 650 那么执行' (If variable water level status > 550 and variable water level status < 650 then execute), which contains two '设置' blocks: '设置 P8 引脚伺服舵机为 10 度' (Set P8 pin servo motor to 10 degrees) and '设置 P16 引脚伺服舵机为 10 度' (Set P16 pin servo motor to 10 degrees). After the loop, there is a '当 Topic_0 接收到 Obloq 消息' (When Topic_0 receives Obloq message) block. This is followed by an '如果 Obloq 消息 = 当前水位 那么执行' (If Obloq message = current water level then execute) block, which contains an 'Obloq 发送消息 变量 水位情况 至 Topic_0' (Obloq send message variable water level status to Topic_0) block.

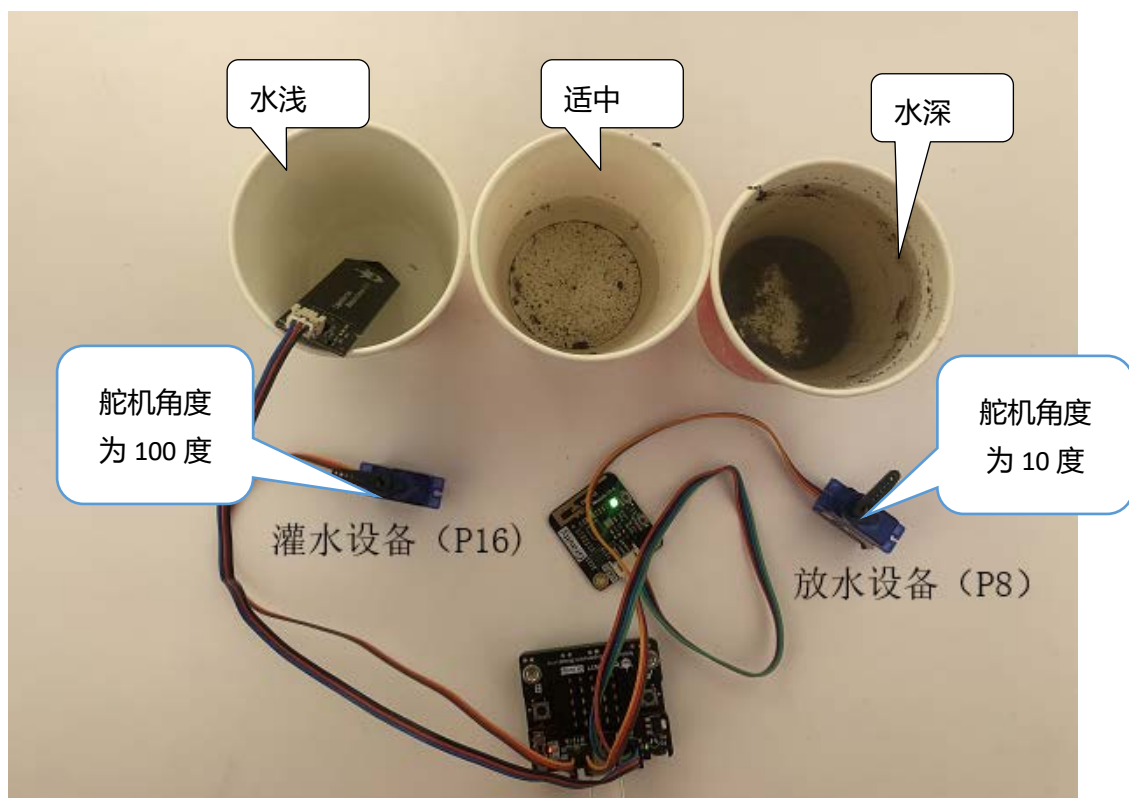
```
micro:bit 主程序开始
Obloq mqtt初始化 接口 软串口 Rx(绿) P1 Tx(蓝) P2
设置 P8 引脚伺服舵机为 10 度
设置 P16 引脚伺服舵机为 10 度
循环执行
  设置 水位情况 的值为 读取模拟引脚 P0
  如果 变量 水位情况 < 500 那么执行
    设置 P8 引脚伺服舵机为 100 度
  如果 变量 水位情况 > 700 那么执行
    设置 P16 引脚伺服舵机为 100 度
  如果 变量 水位情况 > 550 与 变量 水位情况 < 650 那么执行
    设置 P8 引脚伺服舵机为 10 度
    设置 P16 引脚伺服舵机为 10 度
当 Topic_0 接收到 Obloq 消息
  如果 Obloq 消息 = 当前水位 那么执行
    Obloq 发送消息 变量 水位情况 至 Topic_0
```

效果展示

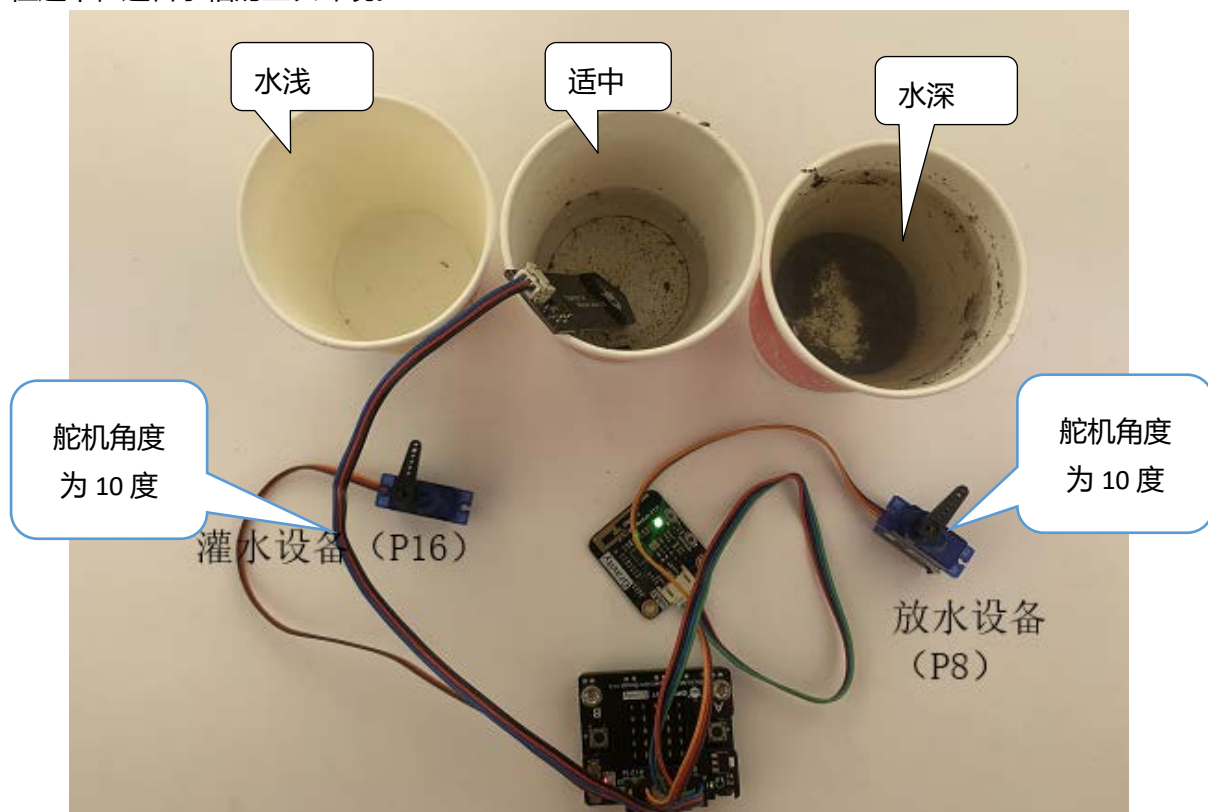
- 当水位情况 <500 时。设备情况如下，P8 口的舵机角度为 100° ，P16 口舵机为 10° 。表示打开放水设备，将田里的水放出去。



- 当水位情况 >700 时。设备情况如下，P16 口的舵机角度为 100° ，P8 口舵机为 10° 。表示打开灌水设备，开始向田里灌水。



- 当 $500 < \text{水位情况} < 700$ 时。设备情况如下，P16 口的舵机角度为 10 度，P8 口舵机为 10 度。表示田里的水位适中，适合水稻的生长环境。



- 进入 Easy IoT 网页发送信息给 micro:bit 设备。找到水位情况，点击“发送消息”，发送指令“当前水位”。

水位情况

 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"

当前水位

发送

- 刷新网页可以查看水位情况。

 最新消息

时间	消息
2019/11/4 9:52:26	469
2019/11/4 9:52:25	当前水位
2019/11/4 9:51:50	623
2019/11/4 9:51:49	当前水位

拓展阅读

物联网在农业领域的应用

物联网技术在农业中的应用，既能改变以前农业经营管理方式，也能提高动植物疫情疫病的防控能力，确保农产品质量安全，可以引领现代农业的发展。

现在传统农业已经不能适应农业发展的需要，产品质量问题，资源浪费，环境污染，产品种类需求等诸多问题，是传统农业发展陷入了恶性循环，而智能农业则是以高新技术和科学管理换取对资源的节约。

应用领域	简介
食品安全	建立了“养殖——屠宰——加工——交易——流通——消费”整套的产业链的全程信息追踪与溯源体系，实现信息汇集，构建“全网络，全过程”的监管
农业信息推送	将天气预报信息，施肥建议信息，病虫害信息及时的推送给农民。
智能化种植教育	对农业大田或者温室大棚里无线传感器或者其他智能控制系统，这样可以对整个作物生长的各种信息进行实时检测，从而及时掌握作物的生长信息，及时调整温湿度等参数。确保给农作物一个很好的生长环境。
水产养殖环境监测	对水中的溶氧量、水温、PH 值等参数进行自动检测和控制，提高水产养殖的自动化和智能化。
节水灌溉	物联网技术无线传感网具有实时性和灵活性布设等特点，它可以被用于各灌溉区域实时监控，从而很大效率的利用水量。

物联网在农业领域中的应用