

创意电子设计



Hello World

——向世界问好

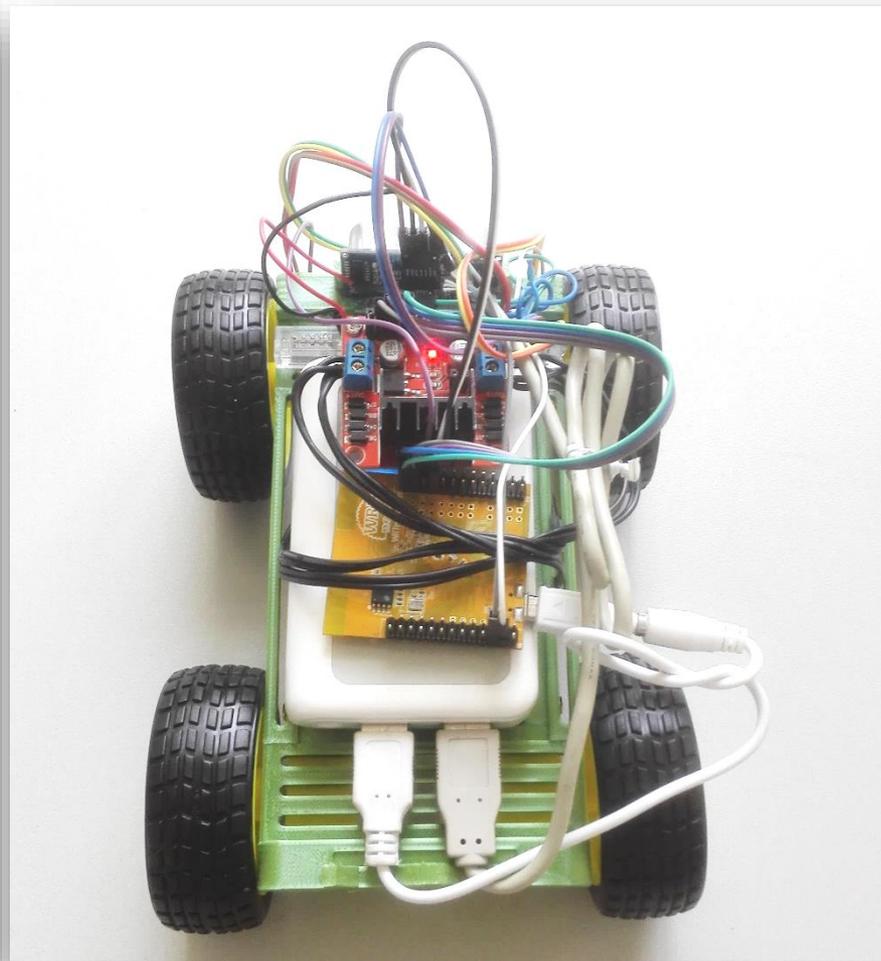
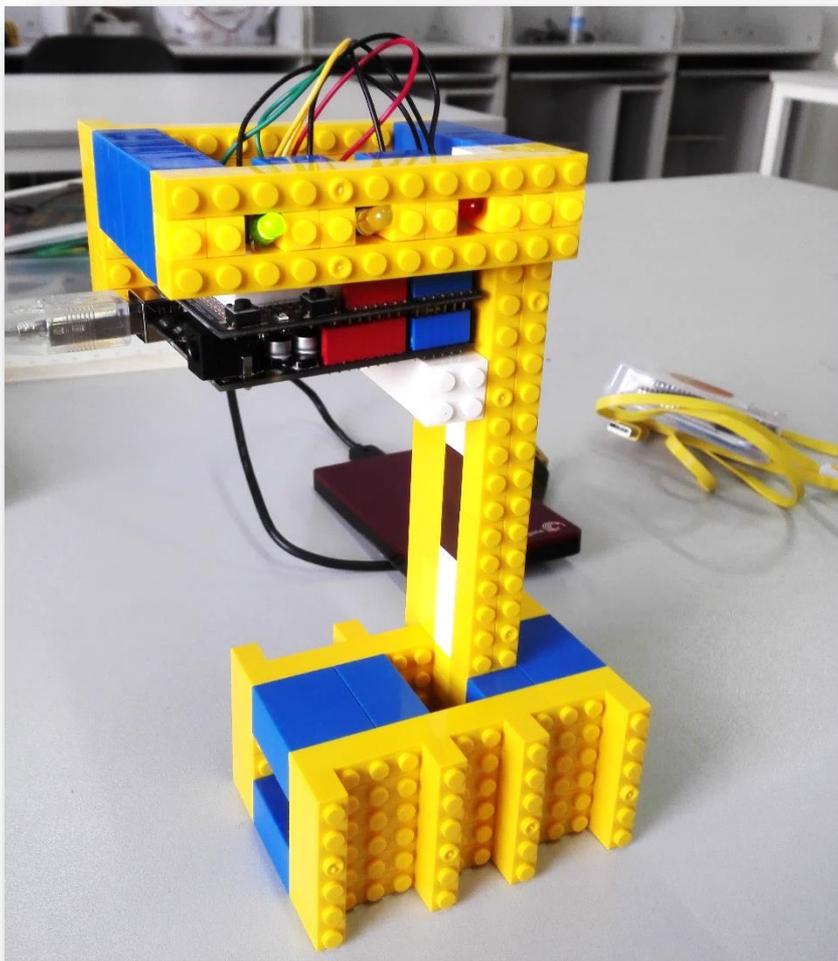


Arduino开源系统简介

- Arduino是一个开放源码电子原型平台，拥有灵活、易用的硬件和软件(板子及在此之上的软件)。
- Arduino可以接收来自各种传感器的输入信号从而做到监测环境的效果，并通过控制光源，电机以及其他执行器来影响其周围环境(类似于人类的工作)。
- 硬件低廉，软件免费。
- 可以完成的例子：
 - 当咖啡煮好时，咖啡壶就发出“吱吱”声提醒
 - 当邮箱有新邮件时，电话就会发出铃声通知
 - 自制一个心率监测器，将每次骑脚踏车的记录存进存储卡
 - 复制一张门禁卡、饭卡
 - ...

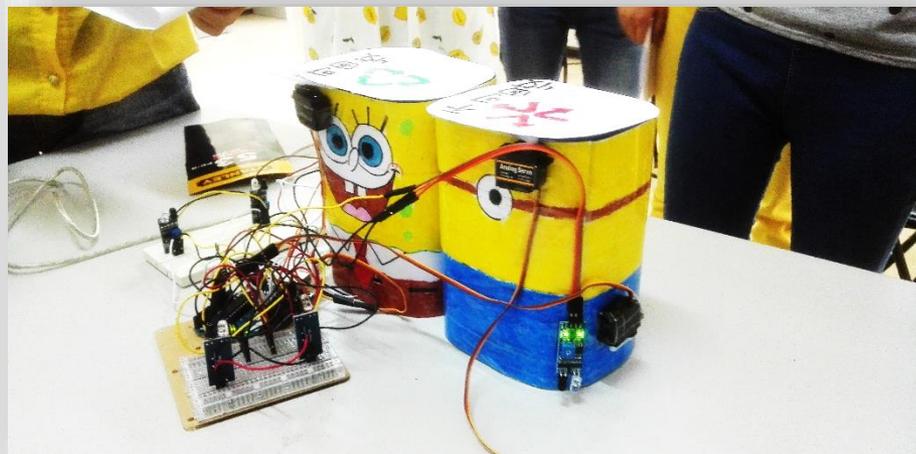
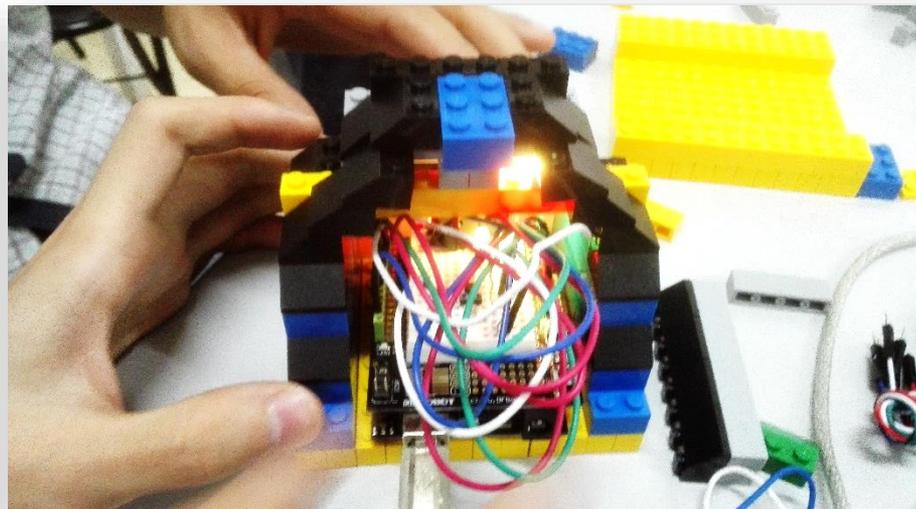
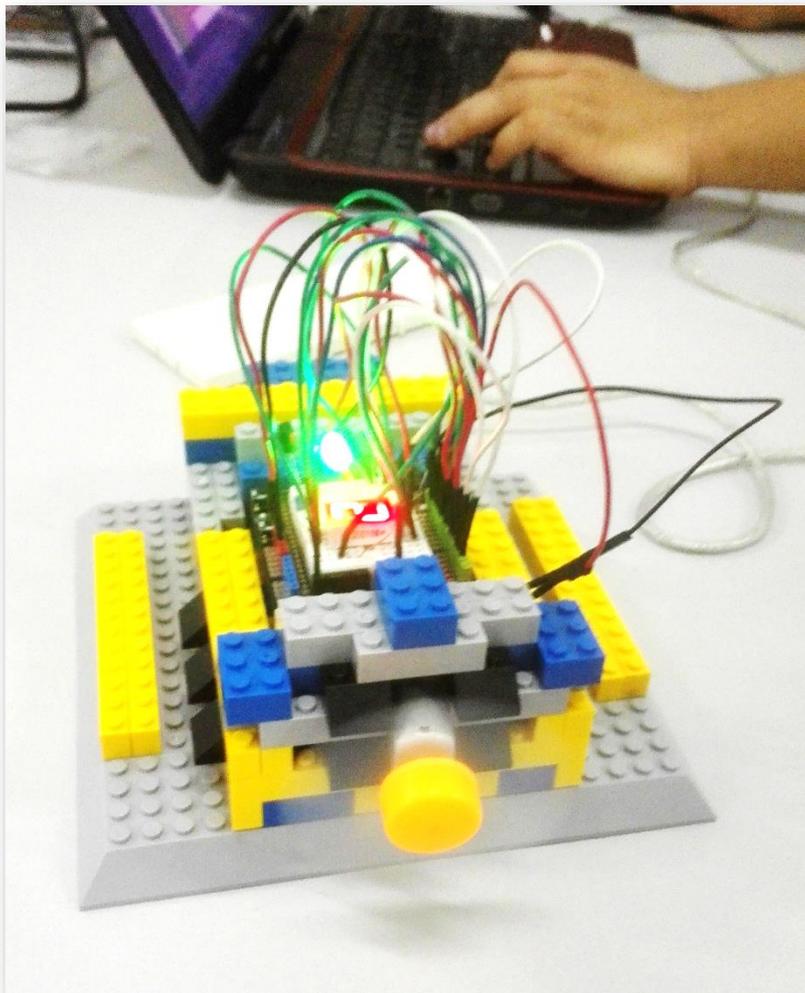
1 Hello World

Arduino开源系统简介



1 Hello World

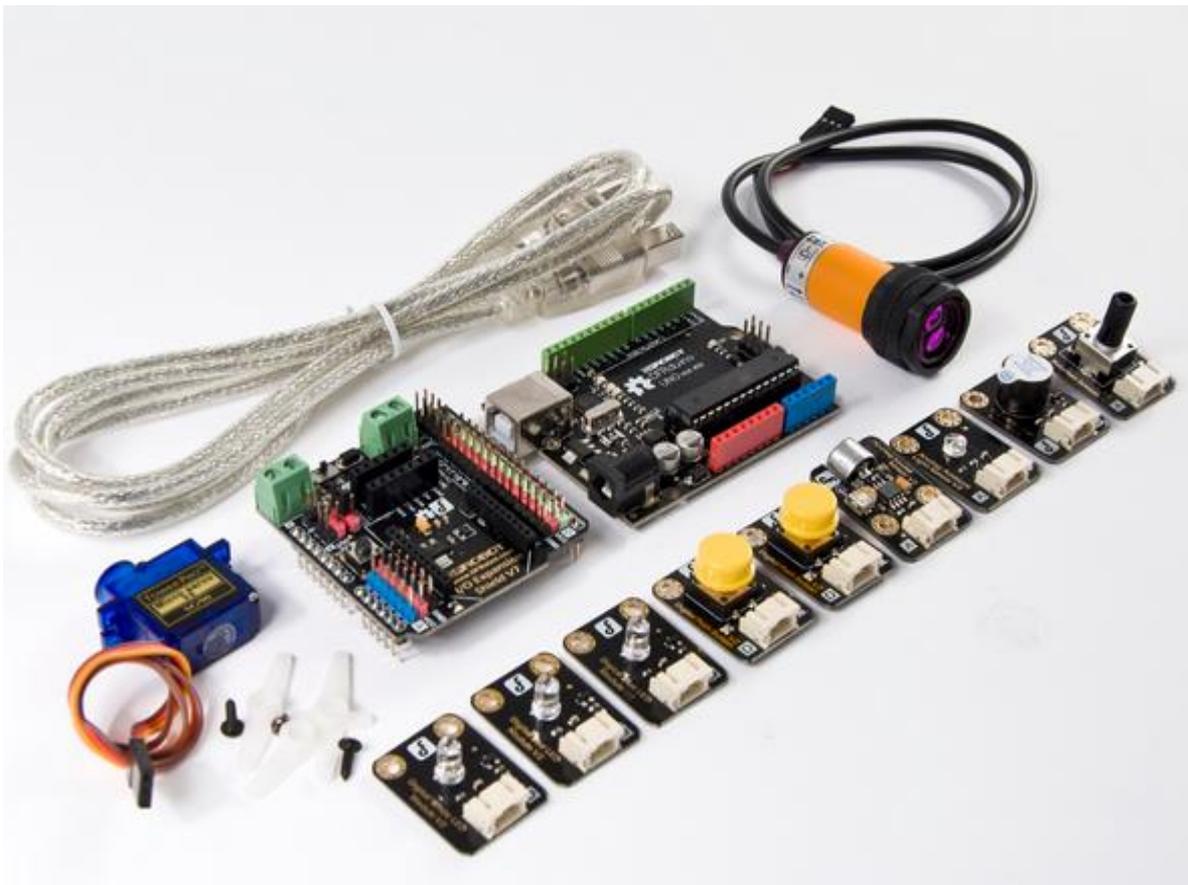
Arduino开源系统简介



1 Hello World

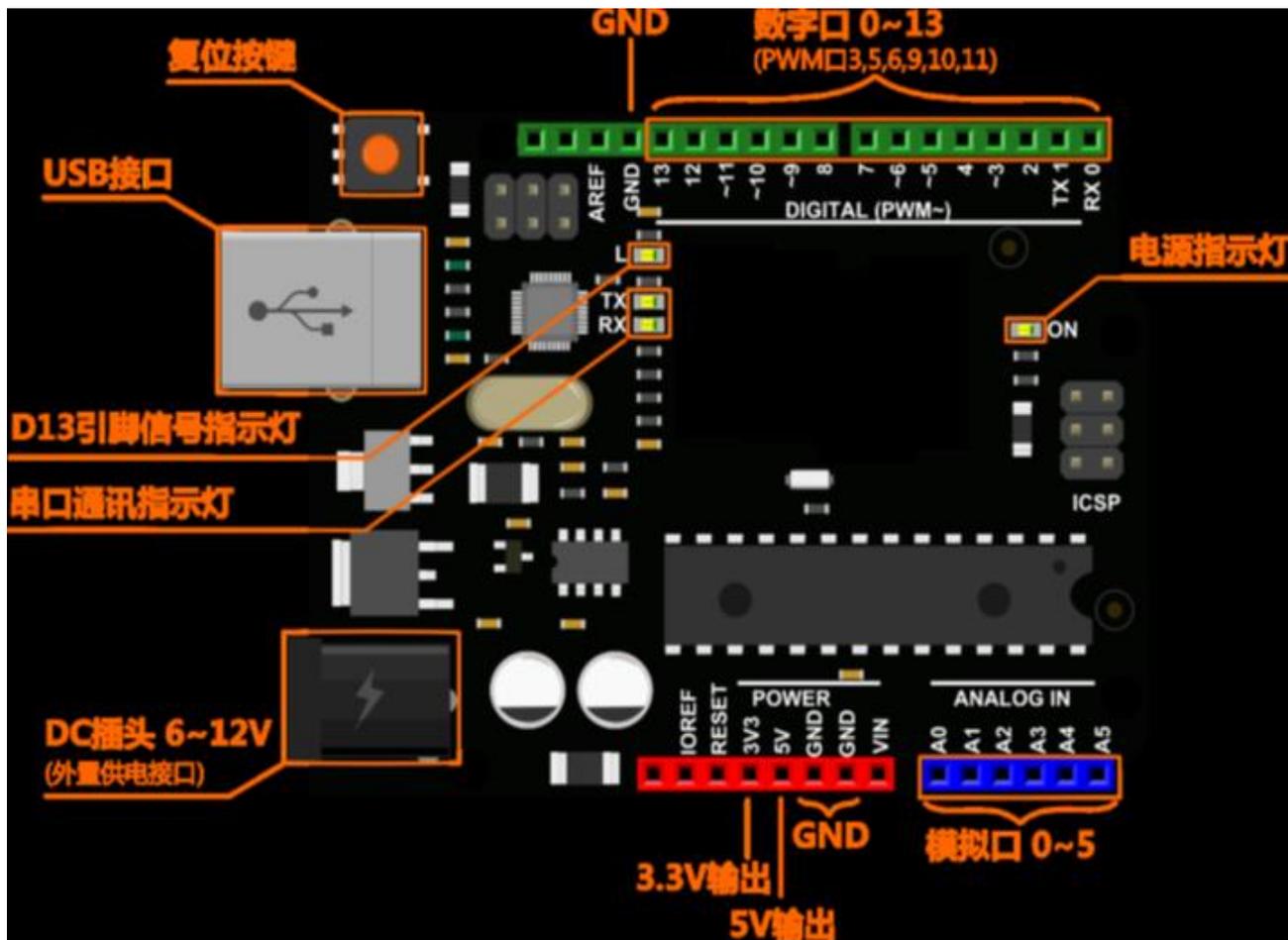
Arduino UNO介绍(套件)

- 3-80cm红外接近开关
- 模拟角度电位器
- 数字蜂鸣器模块
- 模拟环境光线传感器
- 模拟声音传感器
- 数字大按钮模块
- 数字白色LED发光模块
- 数字红色LED发光模块
- 数字蓝色LED发光模块
- TowerPro SG90舵机
- 红外接收模块+遥控器
- 传感器IO扩展板V7
- DFRduino UNO R3
- USB电缆



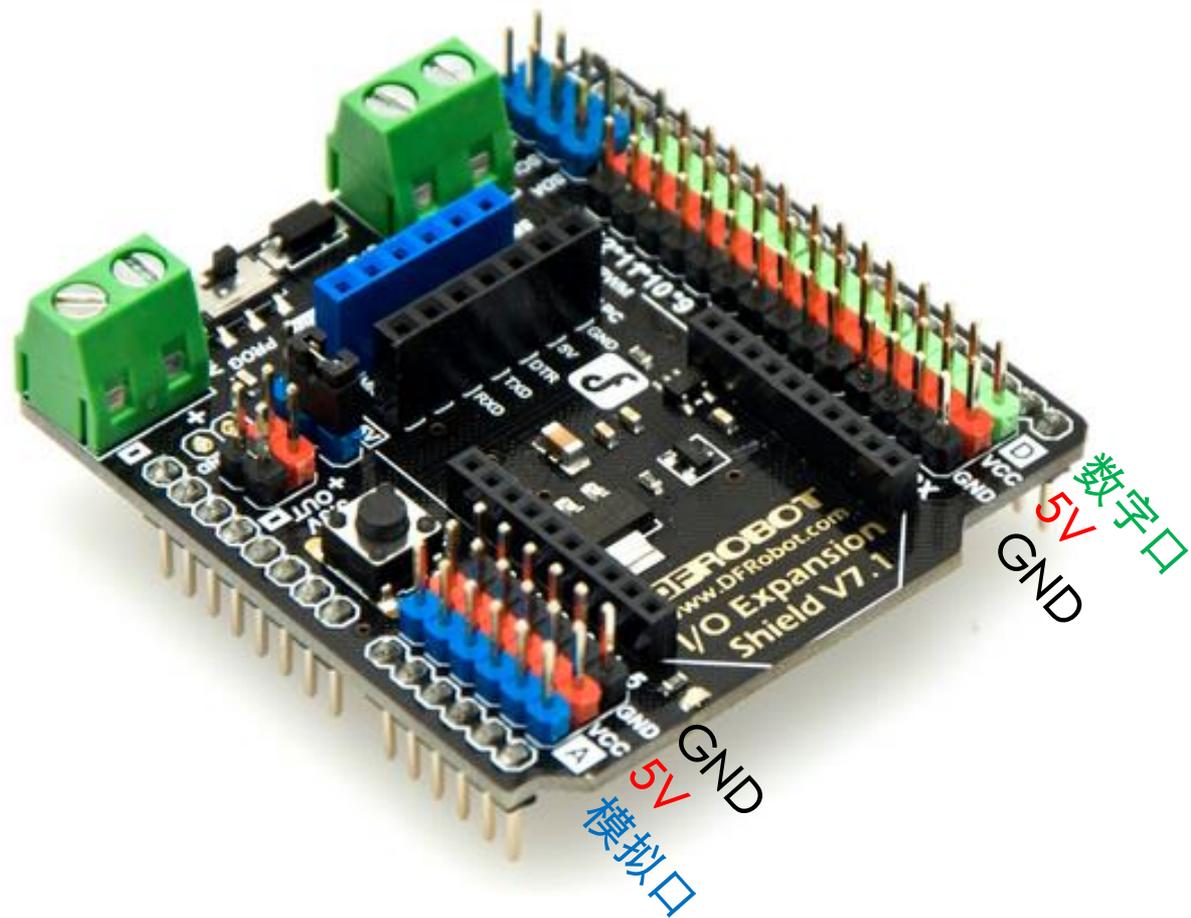
1 Hello World

Arduino UNO介绍



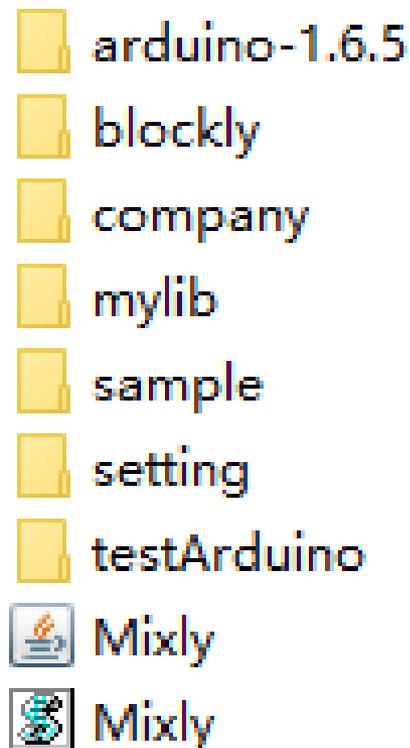
1 Hello World

Arduino UNO扩展板介绍



Arduino入门——Mixly安装使用

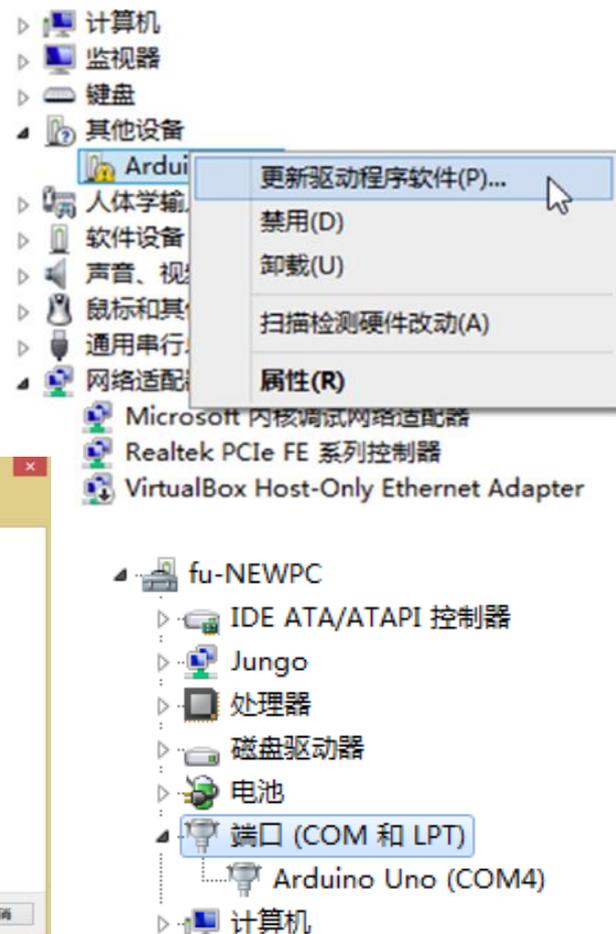
- Mixly是北师大教育学部创客教育实验室提供的免费工具
 - 下载地址：<http://pan.baidu.com/s/1gdjTwLt>
- 把它复制到**F:\创意电子**，按右键，解压到Mixly目录下
 - Arduino-1.xx Arduino官方案序
 - Blockly Mixly主要图形代码
 - Company 公司的第三方扩展库
 - Mylib 自定义扩展库
 - Sample 系统自带例子
 - Setting 系统配置文件
 - testArduino 系统临时文件
 - **Mixly.vbs**是可执行程序，双击执行即可



1 Hello World

Arduino入门——Mixly安装使用

- 物理连接
 - 通过USB线把Arduino UNO和电脑相连
- 驱动安装
 - 驱动程序在arduino-1.x.x\drivers中
 - 安装成功后会出现一个串口
- 启动mixly
 - 双击mixly目录下的mixly.vbs



1 Hello World

Arduino入门——Mixly安装使用

- 请同学们打开**Mixly** 文件夹
- 右键单击 Mixly.vbs → 发送到 → 桌面快捷方式
- 显示桌面
- 今后，每次双击桌面上的Mixly即可打开软件界面



1 Hello World

Why “HELLO WORLD”?

- Hello World的中文意思是：你好，世界。
- 它是世界上的第一个程序。
- 代表**程序员**向世界发出的第一声问候。

做个程序员并不困难!

1 Hello World

任务1：点亮板载LED灯

➤ 代码编写

- 如图编写代码，此代码将板载LED灯闪烁

➤ 上传

- 选择对应串口，点击**上传**按钮，如果没有问题的话，最终会显示上传成功
- 如果只想查看代码是否有错，则可以直接使用**编译**按钮



此时你会发现UNO板上的灯开始闪烁，尝试改变延时中的时间间隔观察效果。

1 Hello World

任务1：点亮板载LED灯



数字输出：

- 支持管脚：0~13、A0~A5（不用0、1）
- 两种状态：高（亮灯）、低（灭灯）

延时：

- 可以选择毫秒（1000毫秒=1秒）或微秒

说一说

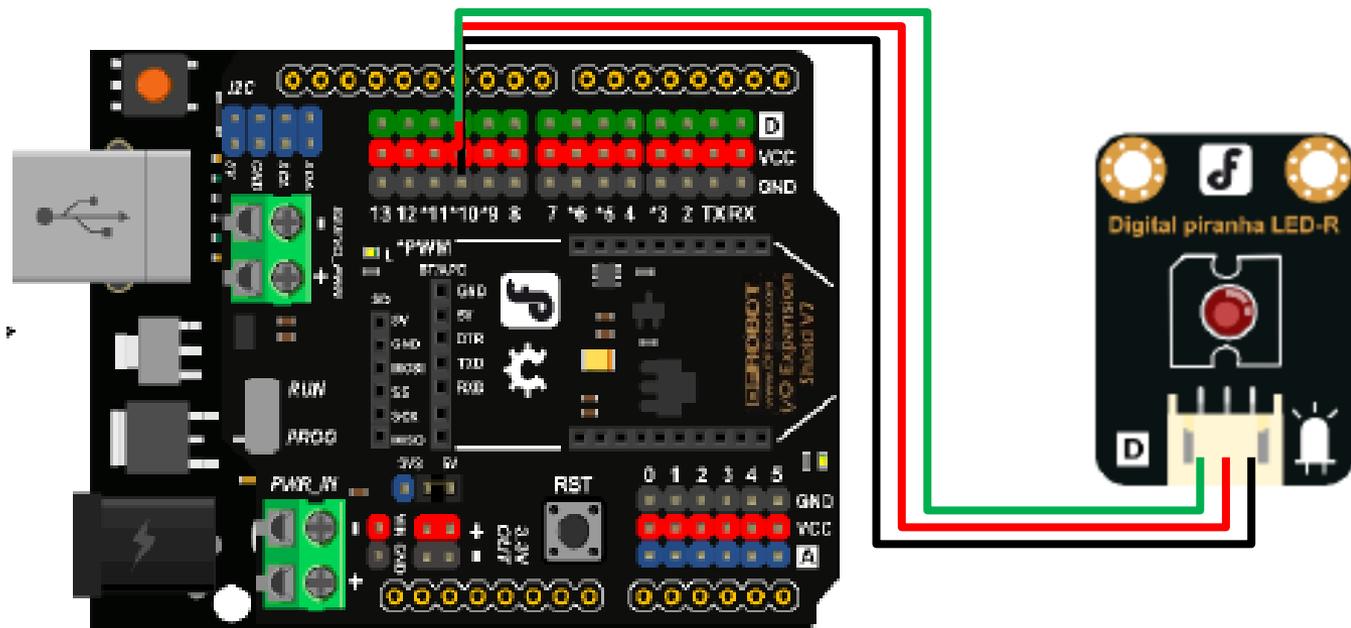
你能用语言描述一下程序的工作过程吗？

1 Hello World

任务2：点亮真实的LED灯

➤ 物理连接

- 将扩展板和主控板的针脚位置做好对应，并插好
- 取出一个LED模块，用连接线将其与**10号管脚**对应的三个管脚相连（**注意插线时颜色的对应**）



1 Hello World

任务2：点亮真实的LED灯

➤ 物理连接

- 将扩展板和主控板的针脚位置做好对应，并插好
- 取出一个LED模块，用连接线将其与**10号管脚**对应的三个管脚相连（**注意插线时颜色的对应**）



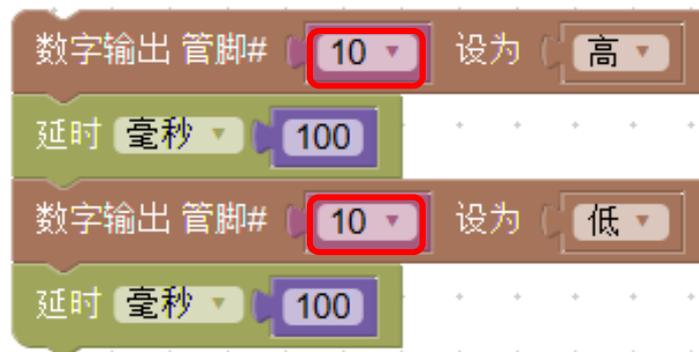
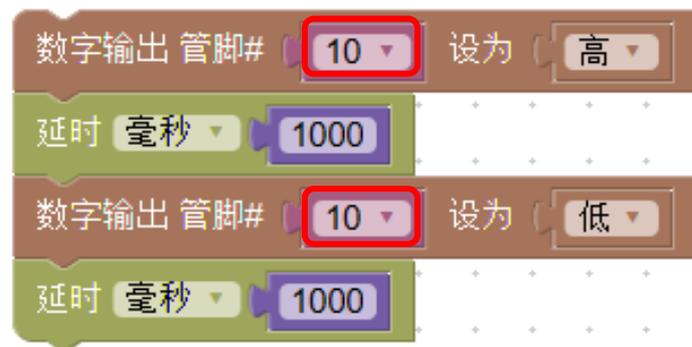
想一想

如何改写程序，才能让LED灯闪烁起来呢？

1 Hello World

任务2：点亮真实的LED灯

➤ 程序编写



此时你会发现LED灯开始闪烁，尝试改变延时的数值观察效果。

任务3：让LED灯越闪越快

➤ 设计思路

- 让LED灯依次以1000、900、……、100的速度闪烁
- 你需要完成的工作只是重复编写之前的代码并做一些简单的修改
- **注意：在这个阶段，所有的代码块都要按先后顺序连接在一起**



任务3：让LED灯越闪越快

成功后感觉怎么样？

1 Hello World

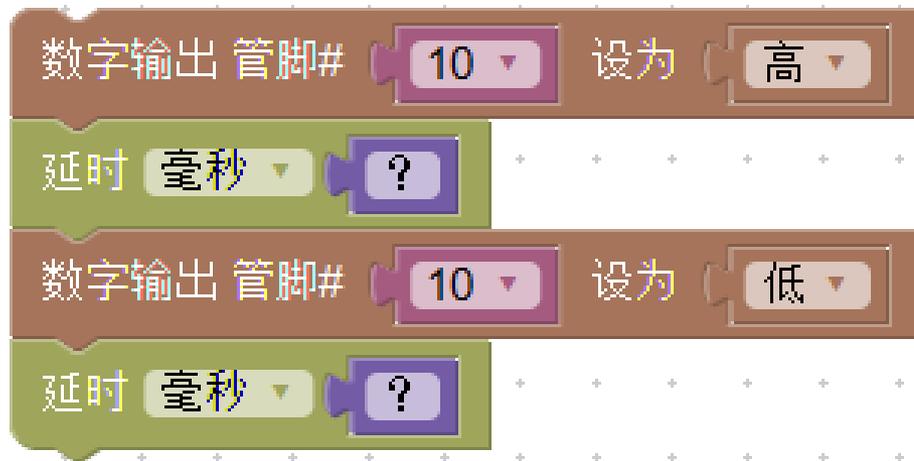
任务3：让LED灯越闪越快

让我们来重新审视一下自己的代码——

你发现了什么？



任务3：让LED灯越闪越快



让我们来做一些大胆的假设：

- ① “？”处的数值可以自动变化
- ②这段代码可以重复执行

任务4：让LED灯越闪越快



如何理解“使用 i 从 1000 到 100 步长为 -100”？

- i 的值第一次为多少？第二次呢？第三次呢？
- i 的值一共会有多少个？最后一次 i 的值是多少？

任务4：让LED灯越闪越快

第一次：i=1000



如何理解“使用 i 从 1000 到 100 步长为 -100”？

- i 的值第一次为多少？第二次呢？第三次呢？
- i 的值一共会有多少个？最后一次 i 的值是多少？

任务4：让LED灯越闪越快

第二次： $i=1000-100=900$



如何理解“使用 i 从 1000 到 100 步长为 -100”？

- i 的值第一次为多少？第二次呢？第三次呢？
- i 的值一共会有多少个？最后一次 i 的值是多少？

任务4：让LED灯越闪越快

第三次： $i=900-100=800$



如何理解“使用 i 从 1000 到 100 步长为 -100”？

- i 的值第一次为多少？第二次呢？第三次呢？
- i 的值一共会有多少个？最后一次 i 的值是多少？

1 Hello World

任务4：让LED灯越闪越快

最后一次（第十次）：i=100



i
1000
900
800
700
600
500
400
300
200
100

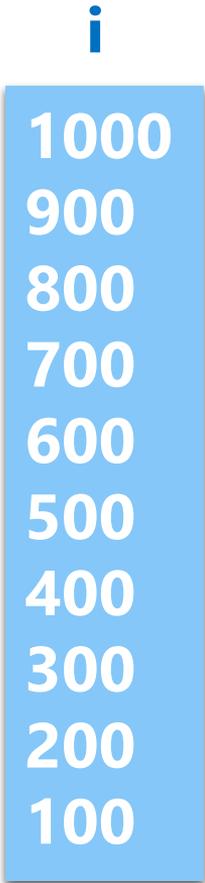
如何理解“使用 i 从 1000 到 100 步长为 -100”？

- i 的值第一次为多少？第二次呢？第三次呢？
- i 的值一共会有多少个？最后一次 i 的值是多少？

1 Hello World

任务4：让LED灯越闪越快

```
使用 i 从 1000 到 100 步长为 -100
执行
  数字输出 管脚# 10 设为 高
  延时 毫秒 i
  数字输出 管脚# 10 设为 低
  延时 毫秒 i
```



变量好比一个盒子，可以用来存放东西(如数值)。

盒子有大小 \leftrightarrow 变量有取值范围
盒子里的东西可以变化 \leftrightarrow 变量的值可以变化

此外，变量有名字，可以用字母*i, j, k*或单词*red, state*等表示。

任务4：让LED灯越闪越快



如何理解“使用 i 从 1 到 15 步长为 2”？

- i 的值第一次为多少？第二次呢？第三次呢？
- i 的值一共会有多少个？最后一次 i 的值是多少？

本节小结

➤ 元件

- LED模块

➤ Mixly块

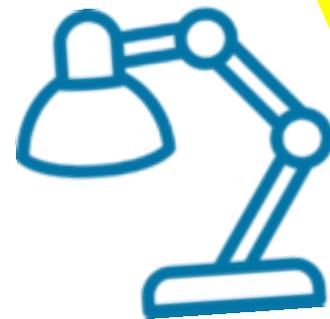
- 数字输出
- 延时
- 循环

➤ 程序设计

- 顺序结构
- 循环结构
- 变量



创意电子设计



Hello World

——向世界问好



创意电子设计



S. O. S.



Save Our Souls (拯救我们的灵魂)

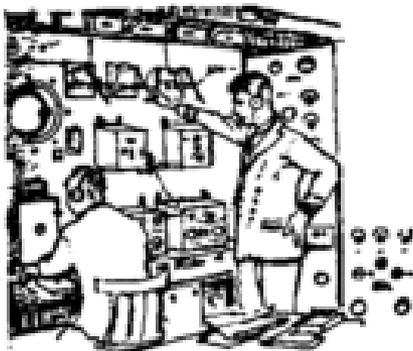
Save Our Ship (拯救我们的船)

Send Our Succour (速来援助)

Saving Of Soul (救命)

摩尔斯电码中，用...---...（**三短三长三短**）来表示S.O.S.
求救信号。

用LED发出摩尔斯电码中的S.O.S.求救信号



.. --- -.- - - . -.- -
A B C D E F G

.... .. -.- - - - -
H I J K L M

.. --- -.- - - - - - - -
N O P Q R S T

.... -.- - - - - - - -
U V W X Y Z



1 .- - - - 2 .- - - - 3 .- - - - 4 .- - - - 5 .- - - -

6 .- - - - 7 .- - - - 8 .- - - - 9 .- - - - 0 .- - - -

摩尔斯电码是美国人萨缪尔·摩尔斯于1844年发明的，它是一种时通时断的信号代码，通过不同的排列顺序来表达不同的英文字母、数字和标点符号。

用LED发出摩尔斯电码中的S.O.S.求救信号

想一想

如何用程序控制小灯按“三短三长三短”的方式闪烁？

试一试

不妨简化问题：先做一组“三短”

提示：每组“三短”之间用一个较长的延时隔开，方便看效果

用LED发出摩尔斯电码中的S.O.S.求救信号



试一试

请你在此基础上完成S. O. S. 的开发

本节小结

- 元件
 - LED模块
- Mixly块
 - 数字输出
 - 延时
 - 循环

创意电子设计



S. O. S.



创意电子设计



神奇的按钮



- 按钮是我们接触的**第一个输入设备**。
- 具有**按下**（高）和**抬起**（低）两种状态。
- 默认状态为**抬起**。

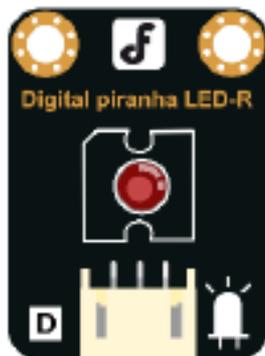
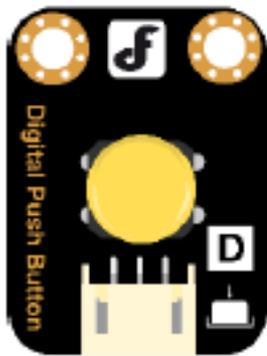


3 神奇的按钮

任务1：让按钮简单工作——按下灯亮，抬起灯灭

➤ 物理连接

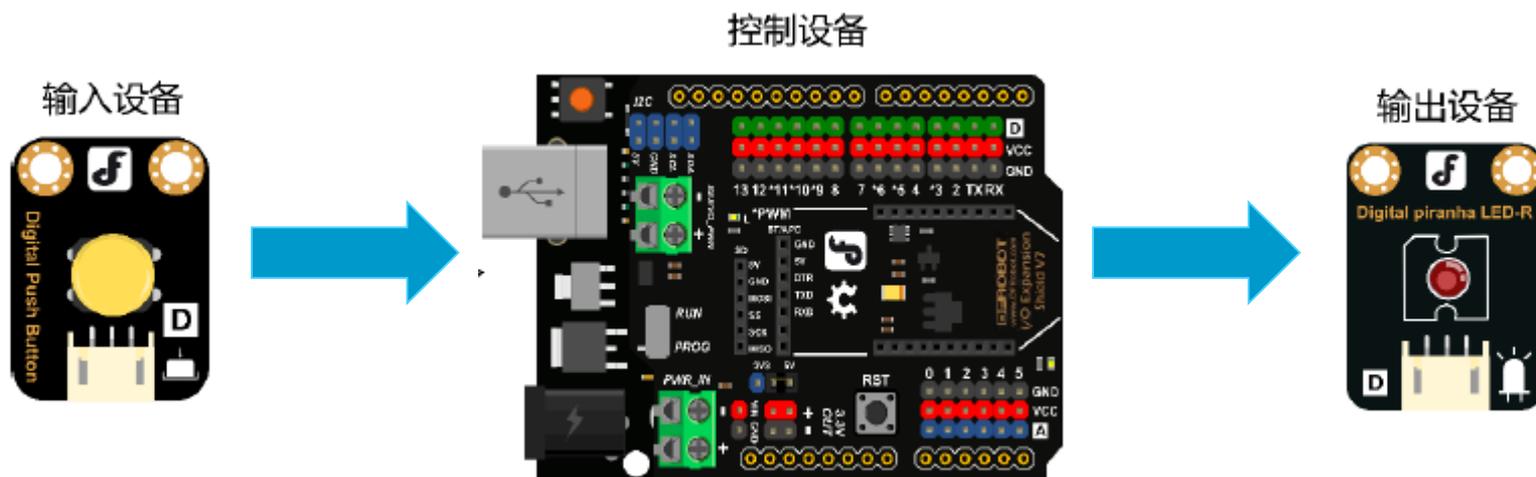
- 取出一个LED模块，用连接线将其与**10号管脚**对应的三个管脚相连（注意插线时颜色的对应）
- 取出一个按钮模块，用连接线将其与**2号管脚**对应的三个管脚相连（注意插线时颜色的对应）



3 神奇的按钮

任务1：让按钮简单工作——按下灯亮，抬起灯灭

➤ 思路分析

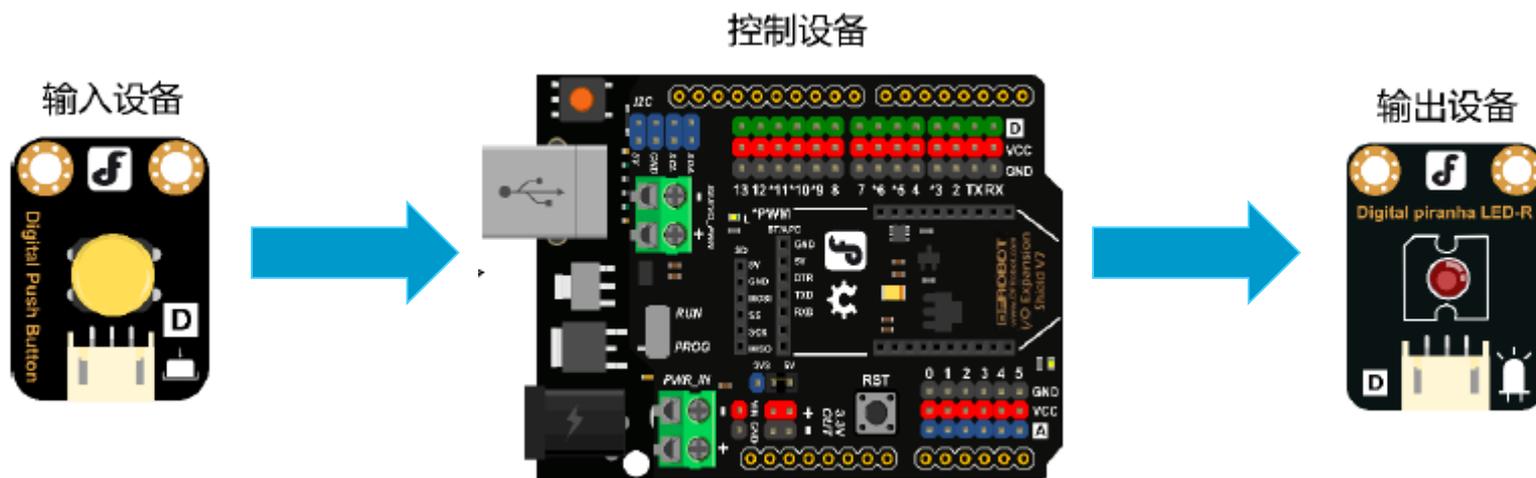


当按钮被按下时，向2号管脚输入__电平，此时10号管脚应输出__电平；
当按钮被抬起时，向2号管脚输入__电平，此时10号管脚应输出__电平。

3 神奇的按钮

任务1：让按钮简单工作——按下灯亮，抬起灯灭

➤ 程序编写



数字输出 管脚# 设为 数字输入 管脚#

数字输入：支持管脚：0~13、A0~A5（不用0、1）；
两种状态：高（按钮按下）、低（按钮抬起）

3 神奇的按钮

任务2：简易延时灯——按下按钮亮，三秒后灭



试一试

这样的思路可以完成任务吗？
如果不行，问题出在哪里？

3 神奇的按钮

任务2：简易延时灯——按下按钮亮，三秒后灭

➤ 程序编写



如果：

- 上口（向右开）插入判断条件，结果有“真”、“假”两种可能。
- 下口（上下开）插入当判断条件为“真”时执行的代码，判断条件的结果为“假”时，则不执行这段代码。

3 神奇的按钮

任务3：使用按钮模拟开关——每次按下，LED灯切换亮灭

试一试

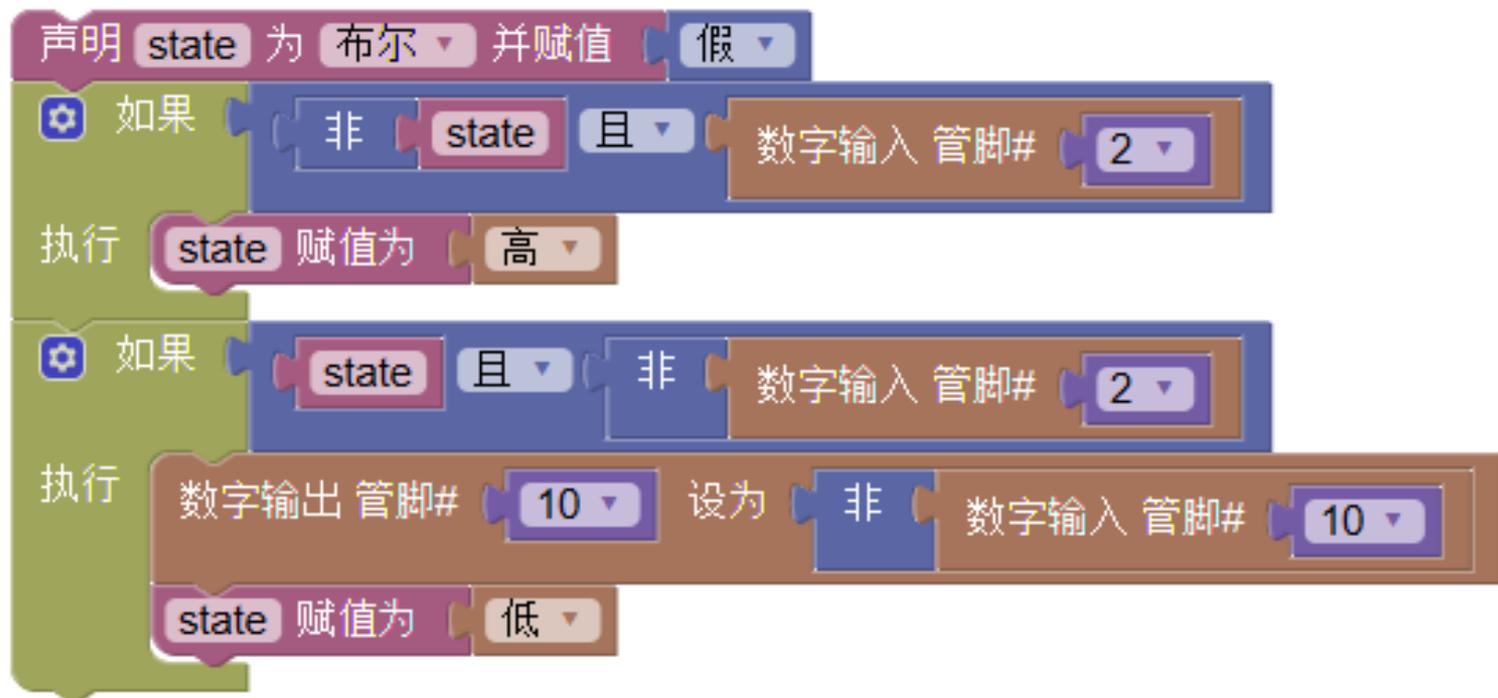
你有什么好思路？不妨尝试动手编一编！

想一想

如果没有成功，问题出在哪里？

3 神奇的按钮

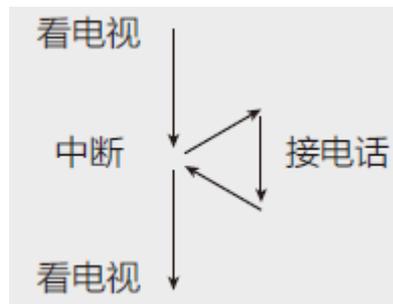
任务3：使用按钮模拟开关——每次按下，LED灯切换亮灭



3 神奇的按钮

任务4：使用按钮模拟开关——按下长亮，再按下一秒后灭

什么是中断？打个比方吧，比如你在家看电视，突然家里电话铃响了，那么你不得不停下看电视先去接电话，等接完电话后，你又可以继续看电视啦！在整个过程中接电话就是一个中断过程，电话铃响就是中断的标志，即触发中断的条件。



→中断被触发的四种条件：

- ①LOW（低）当引脚为低电平时，触发中断。
- ②CHANGE（改变）当引脚电平发生改变时，触发中断。
- ③RISING（上升）当引脚由低电平变为高电平时，触发中断。
- ④FALLING（下降）当引脚由高电平变为低电平时，触发中断。

→中断可以使用的管脚：

Arduino Uno：管脚#2（对应中断0）和管脚#3（对应中断1）。

3 神奇的按钮

任务4：使用按钮模拟开关——按下长亮，再按下一秒后灭



创意电子设计



神奇的按钮





简易入侵检测仪



4 简易入侵检测仪

- 红外接近开关是一种**数字输入**设备。
- 具有未入侵（高）和入侵（低）两种状态。
- 默认状态为未入侵（高）。



4 简易入侵检测仪

任务1：让入侵检测仪工作起来：入侵灯亮，人走灯灭

➤ 物理连接

- 取出一个LED模块，用连接线将其与**10号管脚**对应的三个管脚相连（注意插线时颜色的对应）
- 取出一个红外接近开关，用连接线将其与**3号管脚**对应的三个管脚相连（注意插线：**黄接绿，红接红，绿接黑**）

想一想

红外接近开关在使用上与按钮有什么区别？

4 简易入侵检测仪

任务1：让入侵检测仪工作起来：入侵灯亮，人走灯灭

➤ 软件编写



4 简易入侵检测仪

任务2：如何知道是否有人入侵过？

➤ 软件编写





简易入侵检测仪



创意电子设计



可调灯



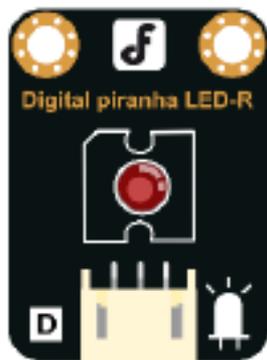
- 灯光在微电脑控制之下完成由暗到亮再由亮到暗的逐渐变化，感觉像是在呼吸，所以称为呼吸灯。
- 呼吸灯广泛应用于手机、无线路由器之上，如你的手机里面有未处理的通知，比如说未接来电，未查收的短信等等，呼吸灯就会由暗到亮的变化，像呼吸一样那么有节奏，起到一个通知提醒的作用。

5 可调灯

任务1：制作一个简易呼吸灯

➤ 物理连接

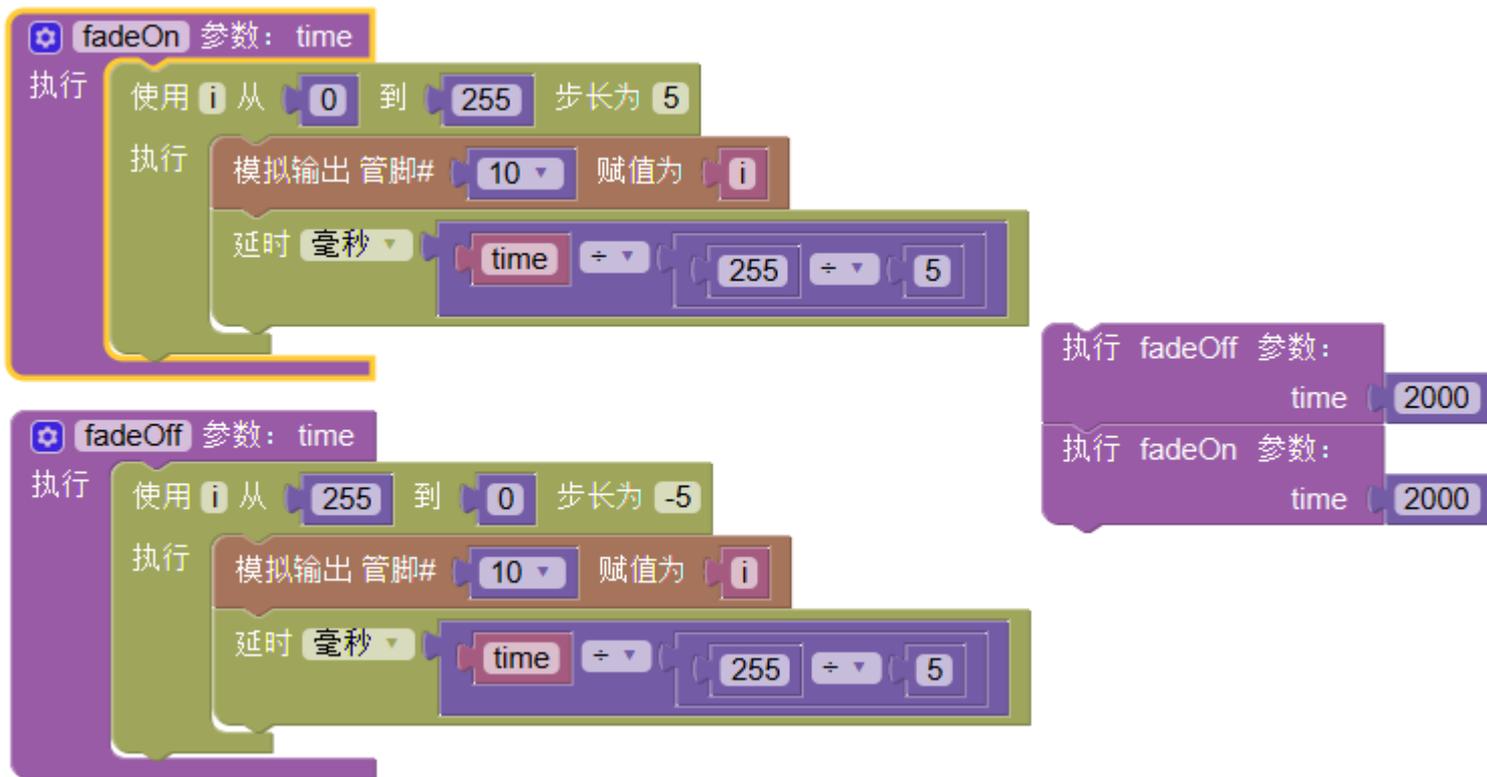
- 取出一个LED模块，用连接线将其与**10号管脚**对应的三个管脚相连（注意插线时颜色的对应）



5 可调灯

任务1：制作一个简易呼吸灯

➤ 软件编写

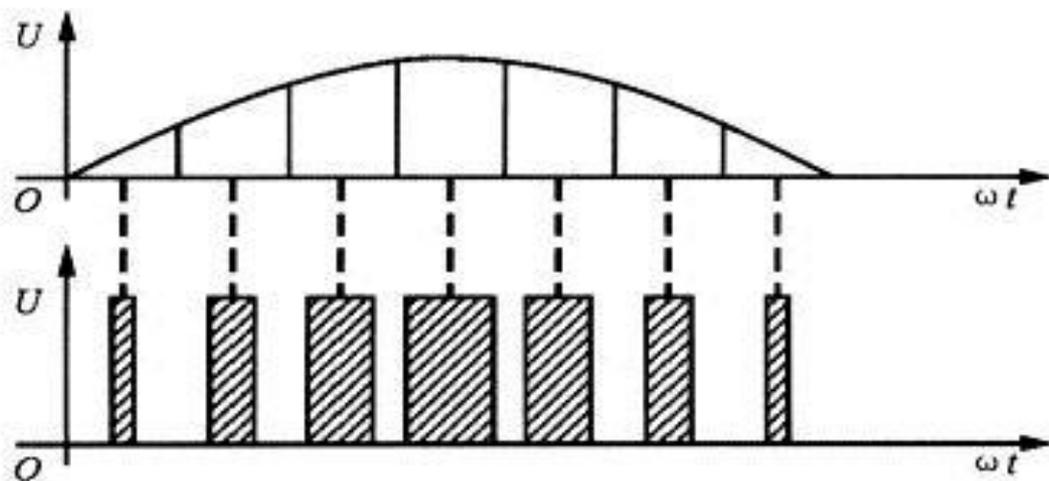


5 可调灯

任务1：制作一个简易呼吸灯

PWM基础知识

- **脉宽调制 (PWM: Pulse Width Modulation) 输出**：它是一种对模拟信号电平进行数字编码的方法，简单来说就是通过一个时钟周期内高低电平的不同占空比来表征模拟信号，如下边就是一个具体的编码样例。

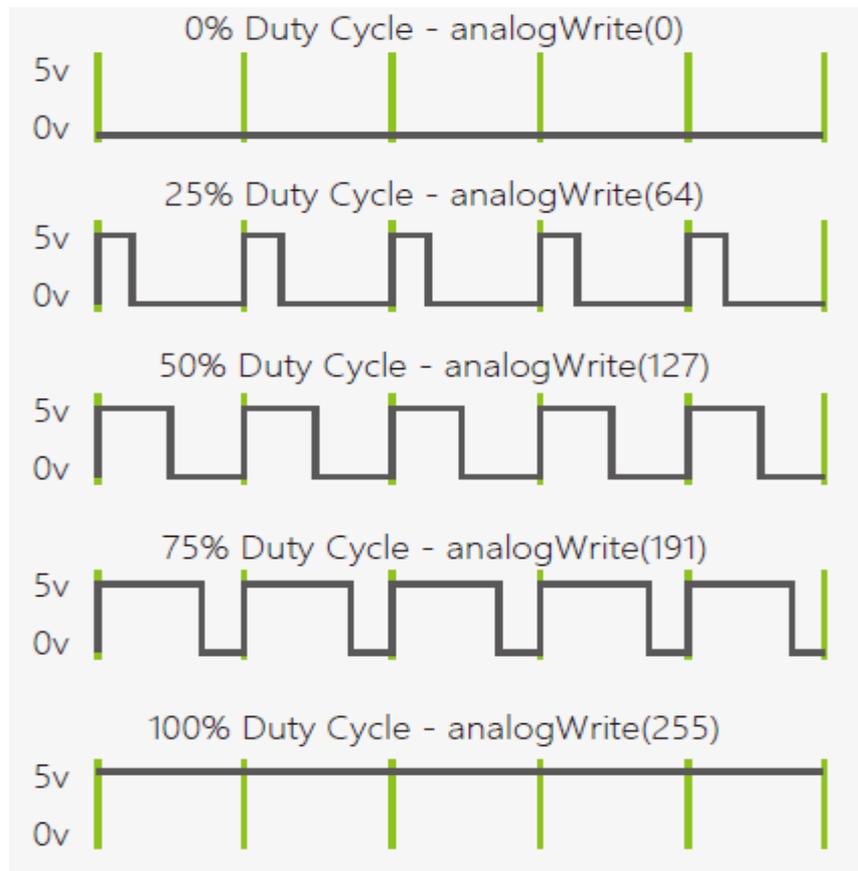


5 可调灯

任务1：制作一个简易呼吸灯

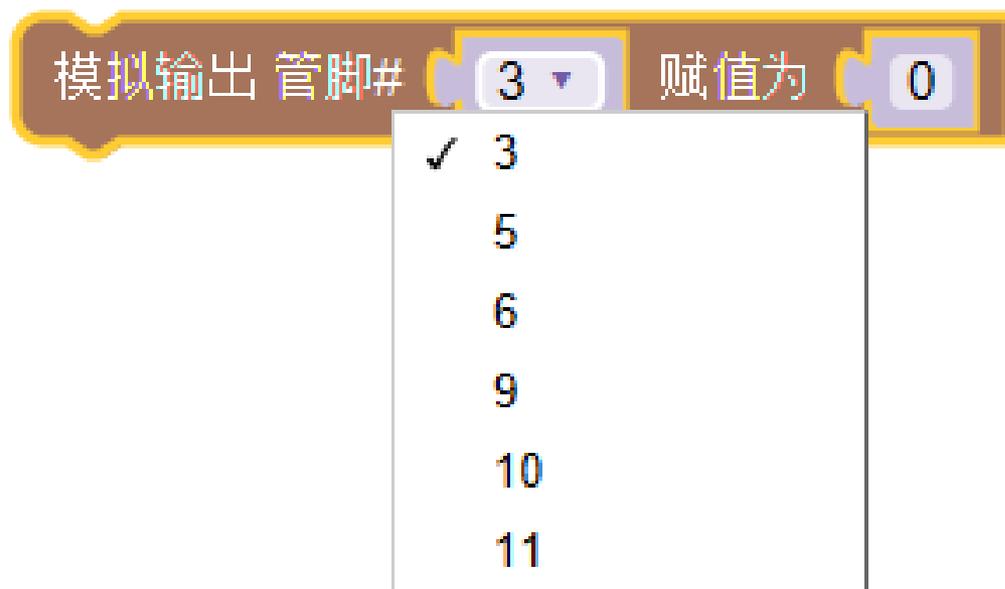
PWM基础知识

- Arduino使用`analogWrite(int value)`输出PWM信号，其中的`value`取值范围是0-255，效果如右图所示。
- Arduino主控板只有有限个GPIO管脚支持PWM。
- 观察一下Arduino板，查看数字引脚，你会发现其中6个引脚（**3、5、6、9、10、11**）旁标有“~”，这些引脚不同于其他引脚，因为它们可以输出PWM信号。



5 可调灯

任务1：制作一个简易呼吸灯



模拟输出：

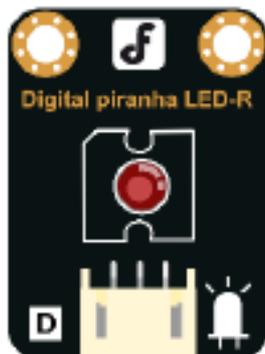
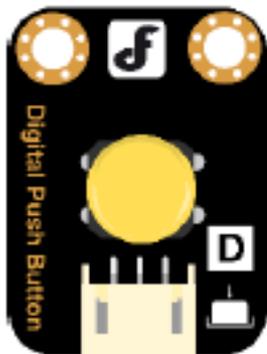
- 支持管脚（UNO）：3、5、6、9、10、11
- 取值范围：0~255

5 可调灯

任务2：制作五档可调灯

➤ 物理连接

- 取出一个LED模块，用连接线将其与**10号管脚**对应的三个管脚相连（注意插线时颜色的对应）
- 取出一个按钮模块，用连接线将其与**2号管脚**对应的三个管脚相连（注意插线时颜色的对应）

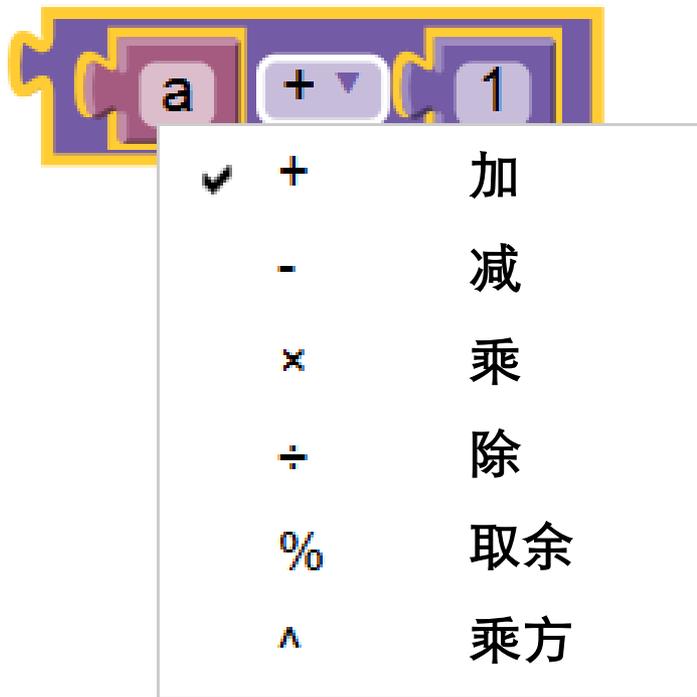


5 可调灯

任务2：制作五档可调灯

➤ 软件编写

档位切换如何实现？



5 可调灯

任务2：制作五档可调灯

➤ 软件编写



想一想

Brightness的值如何变化？

5 可调灯

任务2：制作五档可调灯

➤ 软件编写

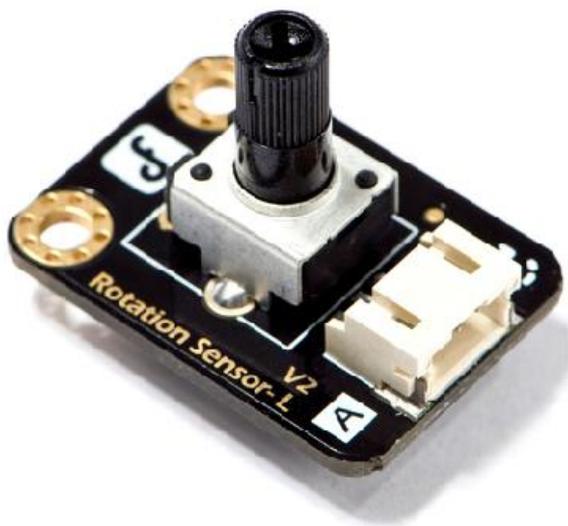
The image shows a sequence of Scratch code blocks for a 5-level adjustable lamp project. The blocks are as follows:

- 初始化 (Initialization):**
 - 声明 `brightness` 为 `整数` 并赋值
 - 声明 `state` 为 `布尔` 并赋值
 - 中断 管脚# `2` 模式 `上升`
 - 执行 `brightness` 赋值为 `brightness + 1 % 6`
- 模拟输出 (Analog Output):**
 - 管脚# `10` 赋值为 `brightness × 51`

任务3：制作旋钮可调灯

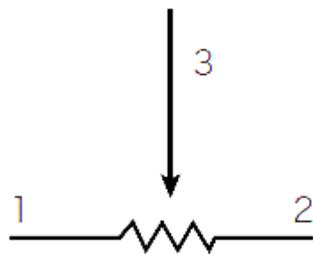
➤ 物理连接

- 取出一个LED模块，用连接线将其与**10号管脚**对应的三个管脚相连（注意插线时颜色的对应）
- 取出一个**模拟**角度电位器模块，用连接线将其与**A0号管脚**对应的三个管脚相连（注意插线时颜色的对应）



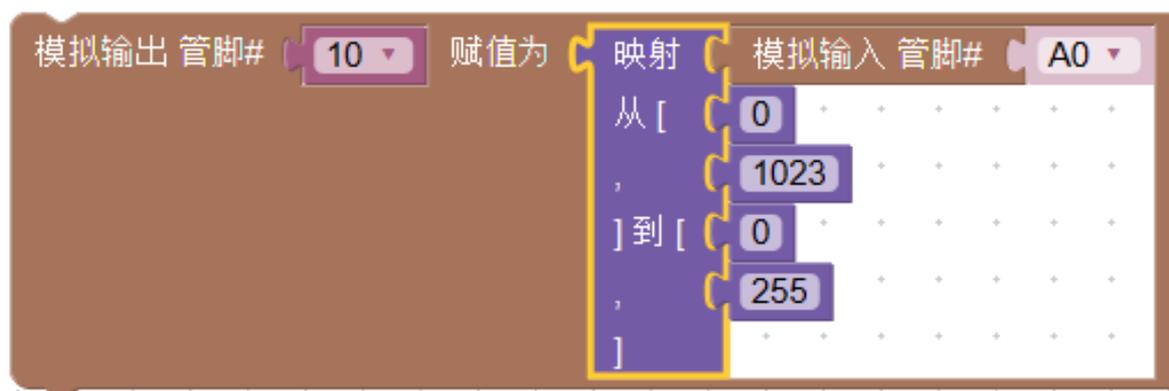
电位器：

- 通过旋转旋钮变化阻值
- 使用模拟输入管脚读取



任务3：制作旋钮可调灯

- 软件编写
(选择外部输入)



- 映射：从[a,b]映射到[c,d]，进行线性变换。C不一定大于d。
- 模拟输入：
 - 支持管脚：**A0~A5**
 - 取值范围：0~1023

想一想

按键和旋钮同时工作，只有开灯后才能调光

5 可调灯

任务3：制作旋钮可调灯

➤ 软件编写

The image shows a Scratch script for controlling a rotary knob dimmer. It is divided into two main sections: initialization and a conditional loop.

初始化 (Initialization):

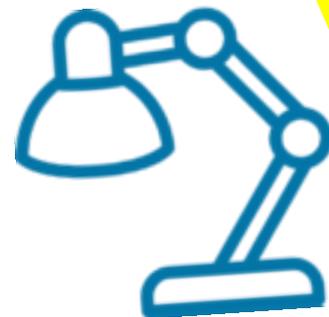
- 声明 `state` 为 布尔 并赋值
- 中断 管脚# `2` 模式 上升
- 执行 `state` 赋值为 非 `state`

主循环 (Main Loop):

- 如果 `state` 为真 (If `state` is true):
 - 执行 模拟输出 管脚# `10` 赋值为 映射 模拟输入 管脚# `A0` 从 [`0` , `1023`] 到 [`0` , `255`]
- 否则 (Else):
 - 数字输出 管脚# `10` 设为 低

扩展任务：制作带开关的可调灯

创意电子设计



可调灯



创意电子设计



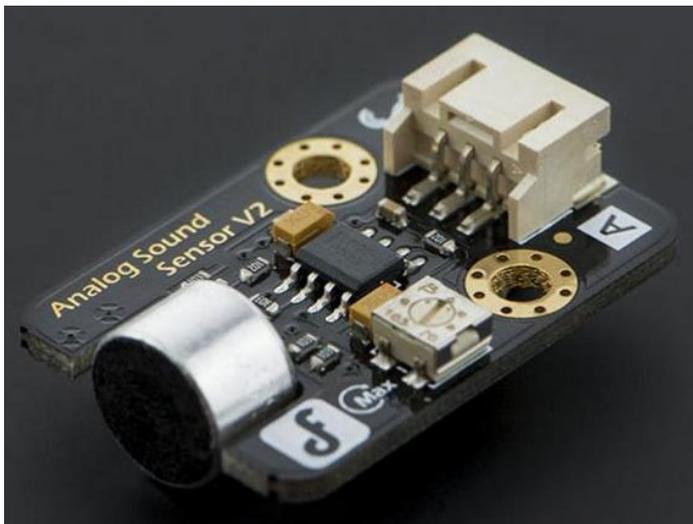
智能灯



任务1：制作声控灯

➤ 物理连接

- 取出一个LED模块，用连接线将其与**10号管脚**对应的三个管脚相连（注意插线时颜色的对应）
- 取出一个**模拟声音传感器**模块，用连接线将其与**A1号管脚**对应的三个管脚相连（注意插线时颜色的对应）



声音传感器：

- 探测声音变化
- 使用模拟输入管脚读取

6 智能灯

任务1：制作声控灯

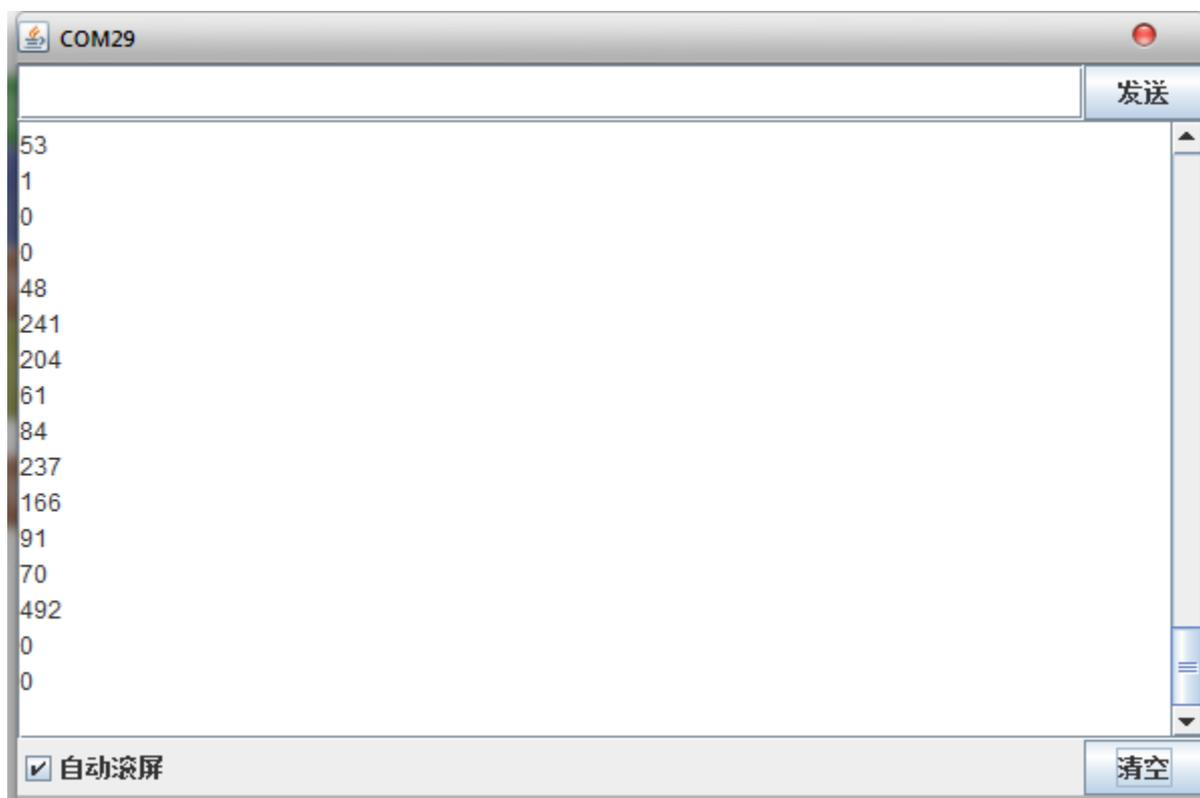
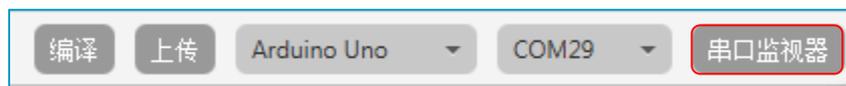
➤ 软件编写



6 智能灯

任务1：制作声控灯

➤ 软件编写



任务2：制作楼道灯

➤ 物理连接

- 取出一个LED模块，用连接线将其与**10号管脚**对应的三个管脚相连（注意插线时颜色的对应）
- 取出一个**模拟声音传感器**模块，用连接线将其与**A1号管脚**对应的三个管脚相连（注意插线时颜色的对应）
- 取出一个**模拟光线传感器**模块，用连接线将其与**A2号管脚**对应的三个管脚相连（注意插线时颜色的对应）

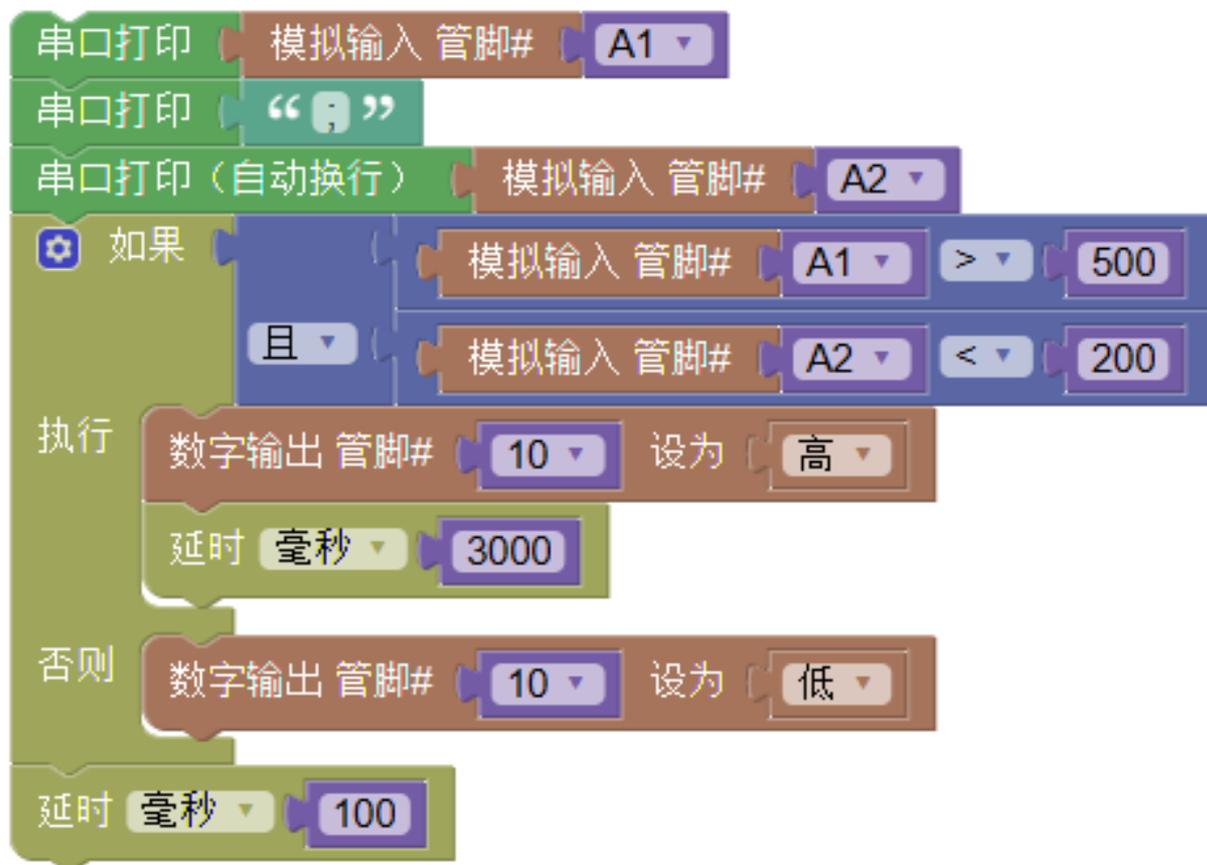


光线传感器：

- 探测声音变化
- 使用模拟输入管脚读取

任务2：制作楼道灯

➤ 软件编写



任务3：制作电子蜡烛

➤ 物理连接

- 取出一个LED模块，用连接线将其与**10号管脚**对应的三个管脚相连（注意插线时颜色的对应）
- 取出一个**模拟声音传感器**模块，用连接线将其与**A1号管脚**对应的三个管脚相连（注意插线时颜色的对应）
- 取出一个**模拟光线传感器**模块，用连接线将其与**A2号管脚**对应的三个管脚相连（注意插线时颜色的对应）



光线传感器：

- 探测声音变化
- 使用模拟输入管脚读取

6 智能灯

任务3：制作电子蜡烛

➤ 软件编写

想一想

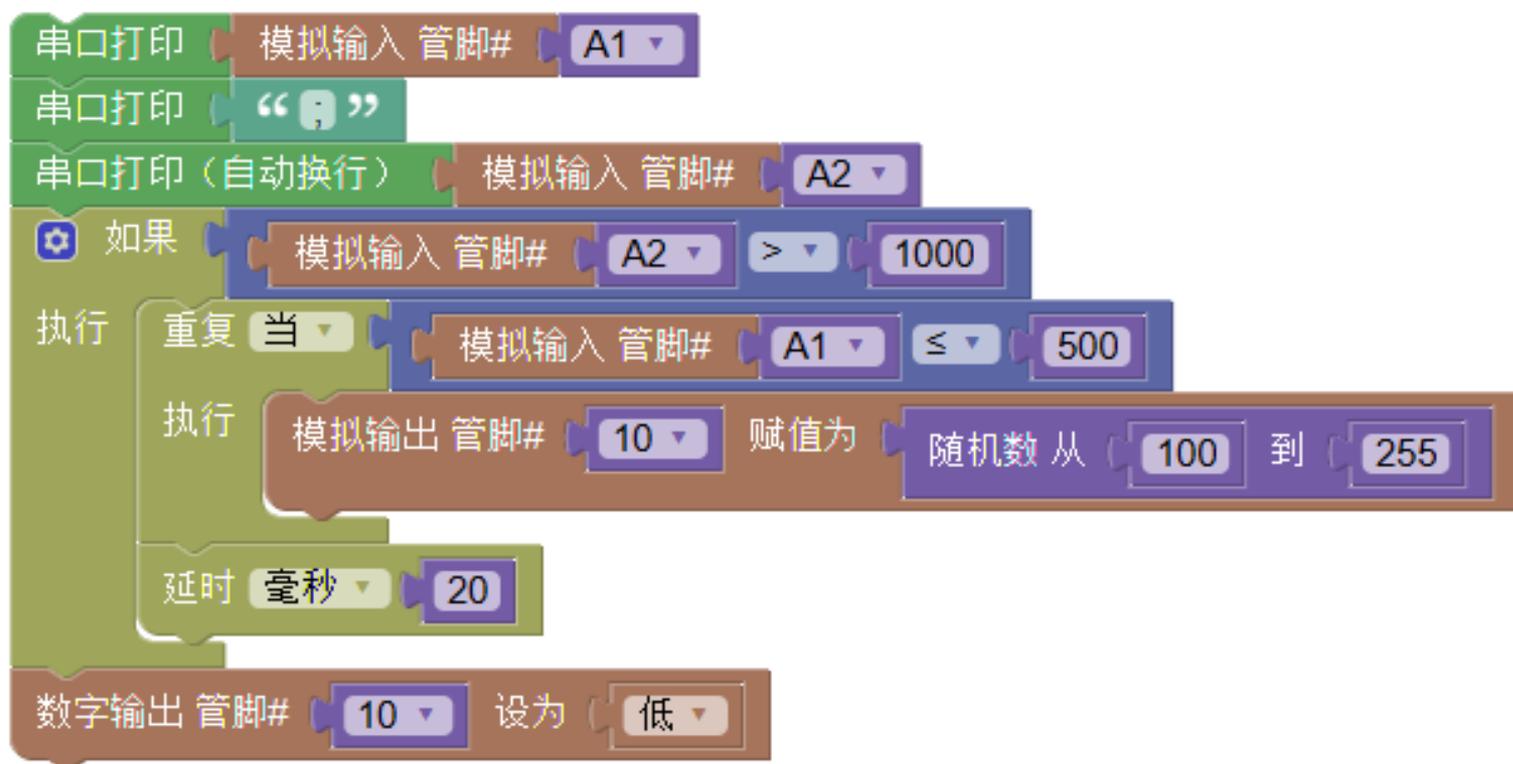
如何模拟蜡烛一晃一晃的效果？



6 智能灯

任务3：制作电子蜡烛

➤ 软件编写



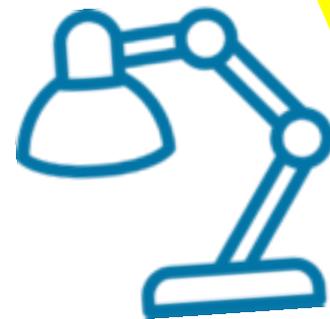
创意电子设计



智能灯



创意电子设计



创意门铃



7 创意门铃

任务：制作简易门铃：按钮按下后，发出“叮咚”的提示

➤ 物理连接

- 取出一个蜂鸣器模块，用连接线将其与**8号管脚**对应的三个管脚相连（注意插线时颜色的对应）
- 取出一个按钮模块，用连接线将其与**2号管脚**对应的三个管脚相连（注意插线时颜色的对应）



蜂鸣器：

- 根据频率输出声音
- 使用数字输出管脚

7 创意门铃

数组

- 有顺序的一列数。
- 两种定义方式：

小数 ▾ tonelist [7] 从字符串 “ 1046.5,1174.7,1318.5,1396.9,1568,1760,1975.5 ” 创建数组



英文逗号!

蜂鸣器相关的函数介绍

➤ tone(pin,frequency)

播放声音 管脚# 10 频率

- Pin都是指连接到蜂鸣器的数字引脚，frequency是以Hz为单位的频率值。

➤ tone(pin,frequency,duration)

播放声音 管脚# 10 频率 持续时间

- 加入duration参数，它是以毫秒为单位，表示声音长度的参数。像第一个函数，如果没有指定duration，声音将一直持续直到输出一个不同频率的声音产生。

➤ noTone(pin)

结束声音 管脚# 10

- noTone(pin)函数，结束该指定引脚上产生的声音。

7 创意门铃

任务：制作简易门铃：按钮按下后，发出“叮咚”的提示

➤ 软件编写

小数 | tonelist [7] 从字符串 “ 1046.5,1174.7,1318.5,1396.9,1568,1760,1975.5 ” 创建数组

ring

执行

- 使用 i 从 1 到 7 步长为 1
- 执行
 - 播放声音 管脚# 8 频率 获取第 3 项 (在数组 tonelist 中)
 - 延时 毫秒 500
 - 播放声音 管脚# 8 频率 获取第 1 项 (在数组 tonelist 中)
 - 延时 毫秒 500
 - 结束声音 管脚# 8

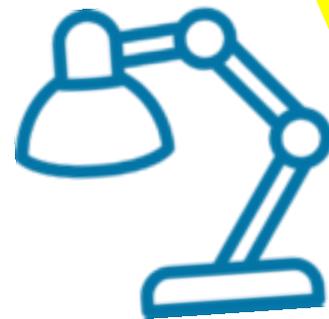
如果 数字输入 管脚# 2

- 执行 执行 ring
- 否则 结束声音 管脚# 8

扩展任务：

- ①每次按下按钮，播放一句旋律
- ②制作一个暗号门铃：用户依照特定方式按按钮（如一秒内按两次），门铃才会响

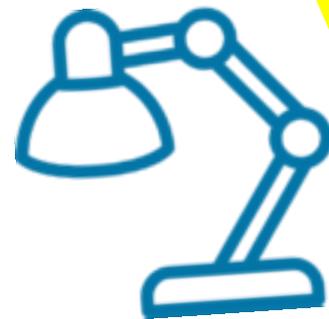
创意电子设计



创意门铃



创意电子设计



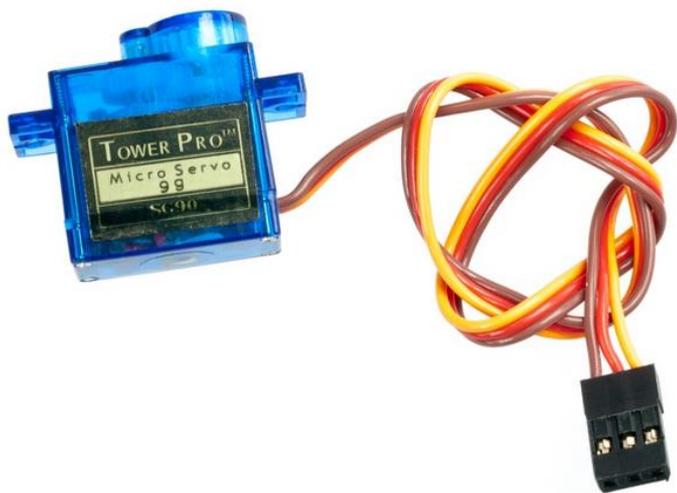
噪音计



基础任务：噪音计的开发

➤ 物理连接

- 取出舵机模块，用连接线将其与**9号管脚**对应的三个管脚相连（注意插线时颜色的对应）
- 取出一个**模拟声音传感器**模块，用连接线将其与**A1号管脚**对应的三个管脚相连（注意插线时颜色的对应）



舵机：

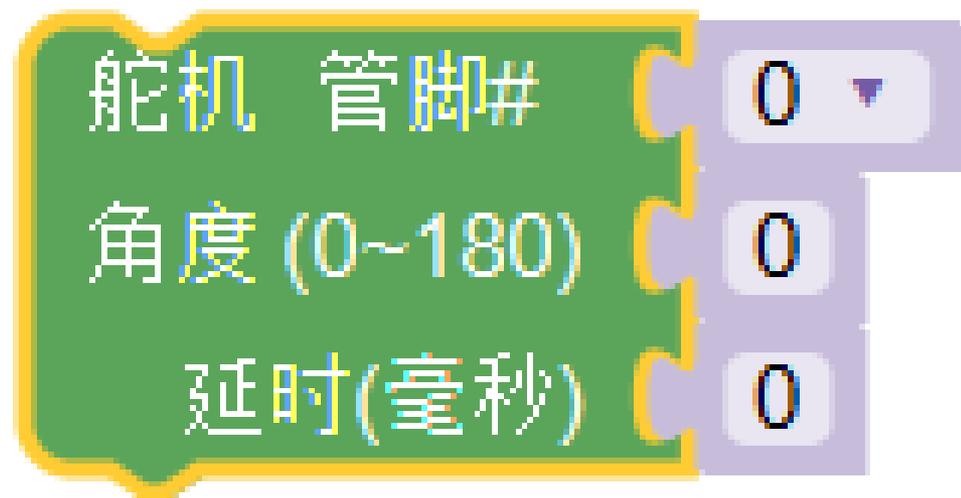
- 由直流电机、减速齿轮组、传感器和控制电路组成的一套自动控制系统。
- 通过发送信号，指定输出轴旋转角度。
- 套件中舵机最大旋转角度：180°

8 噪音计

基础任务：噪音计的开发

➤ 软件编写

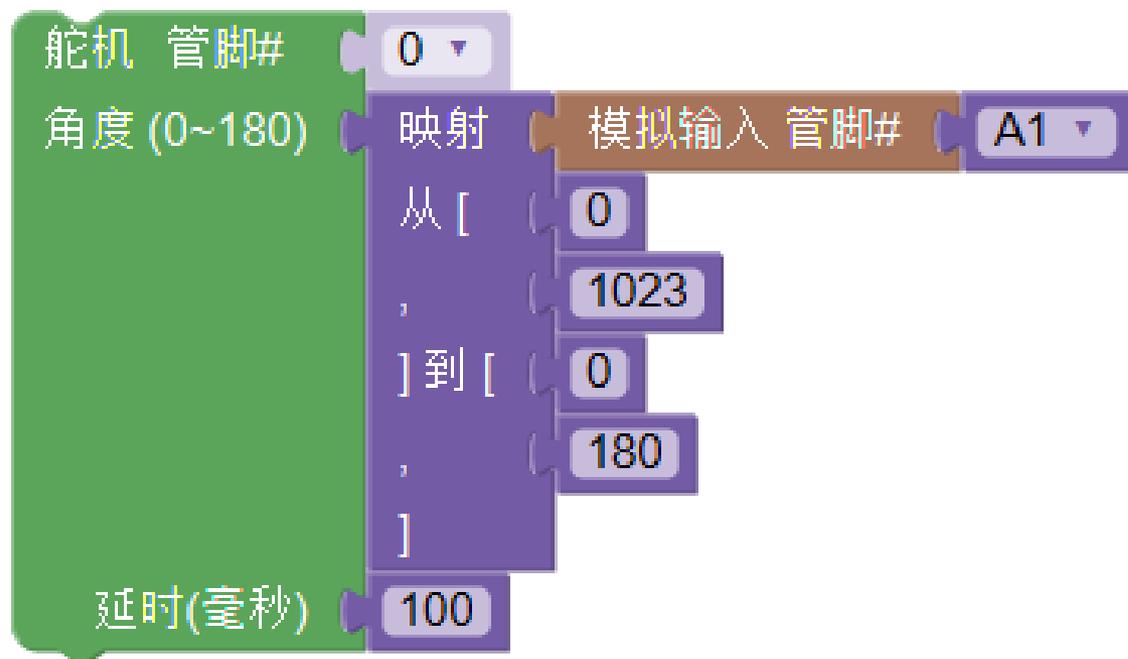
- 舵机模块的使用：



8 噪音计

基础任务：噪音计的开发

➤ 软件编写



扩展任务：噪音计的改进

➤ 任务要求

- 用户按下按键后开始工作，只记录最大值
- 再次按下按键停止记录，并回到初始位置

8 噪音计

扩展任务：噪音计的改进

➤ 软件编写

初始化

初始化操作(这里面的内容只执行一次)

声明 `maxSound` 为 整数 并赋值

中断 管脚# 2 模式 上升

执行 `state` 赋值为 非 `state`

如果 `state`

执行

数字输出 管脚# 10 设为 高

如果 `maxSound` < 模拟输入 管脚# A1

执行 `maxSound` 赋值为 模拟输入 管脚# A1

舵机 管脚# 9

角度 (0~180) 映射 `maxSound` 从 [0 , 1023] 到 [0 , 180]

延时(毫秒) 0

否则

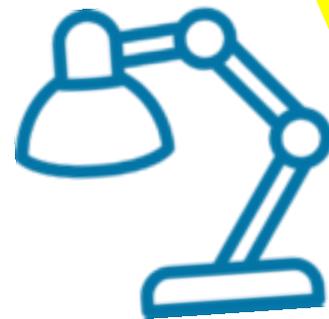
数字输出 管脚# 10 设为 低

舵机 管脚# 9

角度 (0~180) 0

延时(毫秒) 0

创意电子设计



噪音计



创意电子设计



遥控灯



基础任务：红外遥控灯的开发

➤ 物理连接

- 取出一个LED模块，用连接线将其与**10号管脚**对应的三个管脚相连（注意插线时颜色的对应）
- 取出一个红外遥控模块，用连接线将其与**5号管脚**对应的三个管脚相连（注意插线时颜色的对应）



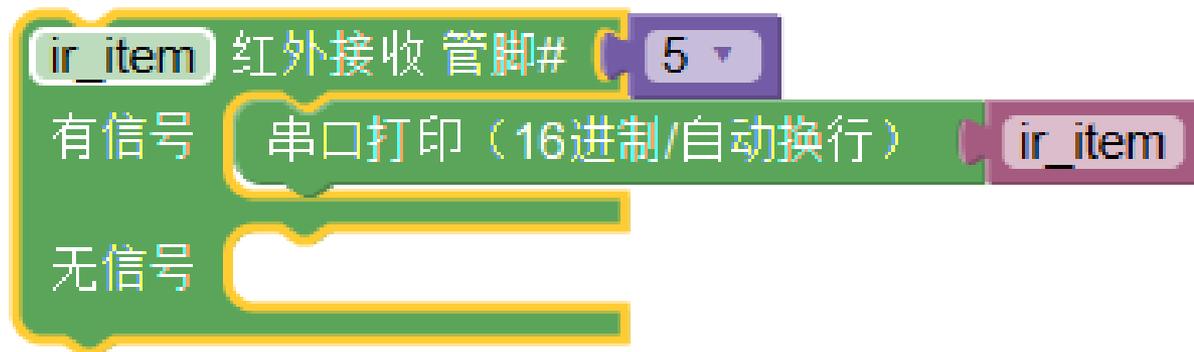
红外遥控器：

- 现实世界的大多数遥控器都是红外的，如电视机遥控器，机顶盒遥控器等。
- 任何一个遥控系统都由发射器和接收器两部分组成。

基础任务：红外遥控灯的开发

➤ 软件编写

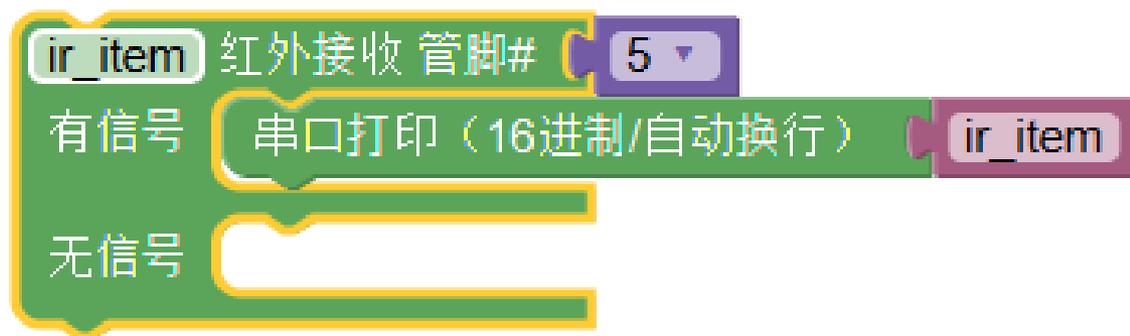
- 每个按钮都有一个特定的16进制代码，都以FD开头。
- 在接收过程中有可能出错。
- 读取按键对应的代码值：



基础任务：红外遥控灯的开发

➤ 软件编写

- 读取按键对应的代码值：



- 编译、上传后，打开“串口监视器”



9 遥控灯

基础任务：红外遥控灯的开发

➤ 软件编写



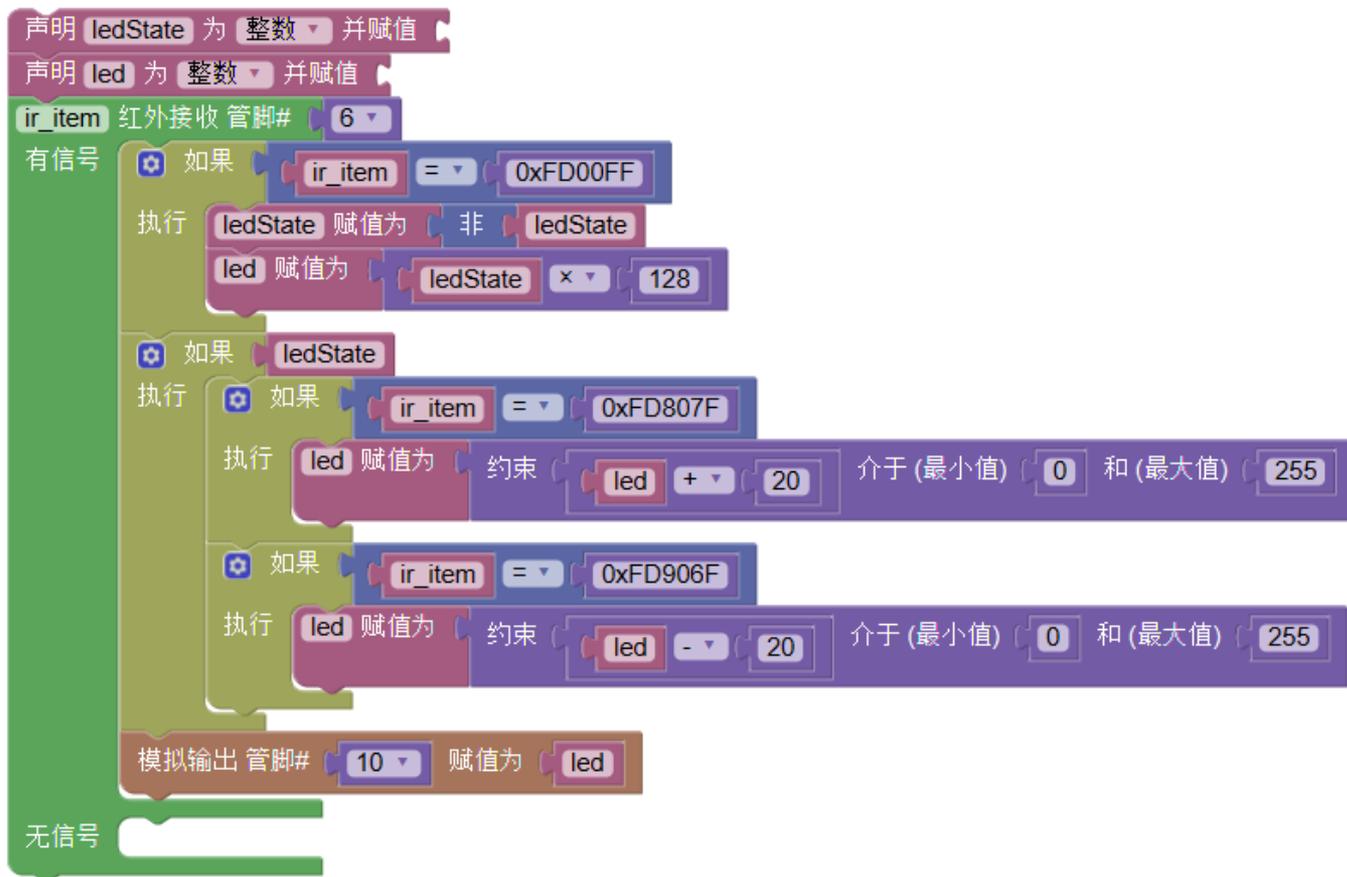
扩展任务：红外调光器的开发

➤ 软件编写

- 使用“+、-”对应光值的明暗变化
- 遥控多个LED灯
- 切换不同的闪灯模式
-

扩展任务：红外调光器的开发

➤ 软件编写——使用“+、-”对应光值的明暗变化



创意电子设计

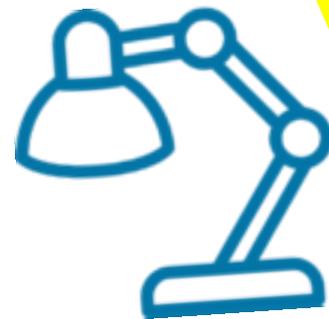


遥控灯



创意电子设计

第10课



另类电子琴

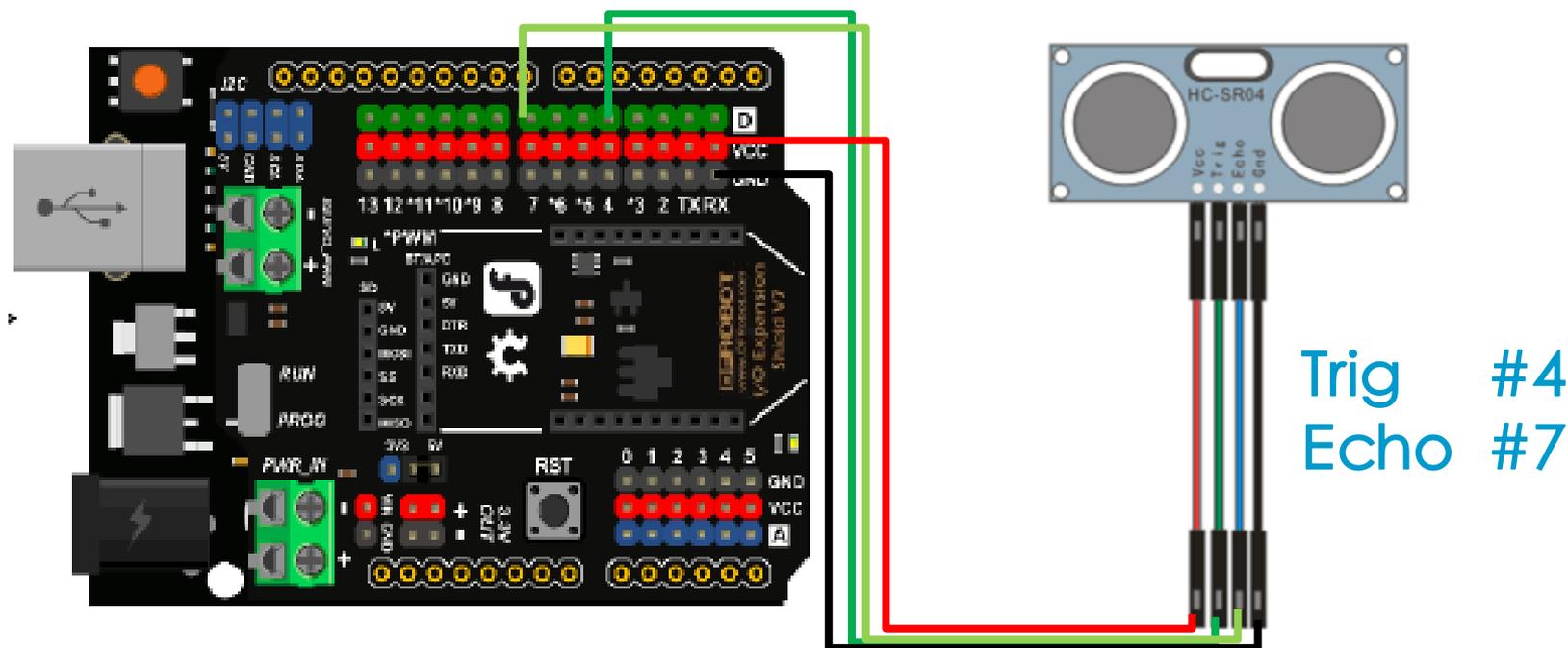


10 另类电子琴

任务：制作另类电子琴：根据距离变化发出不同的音

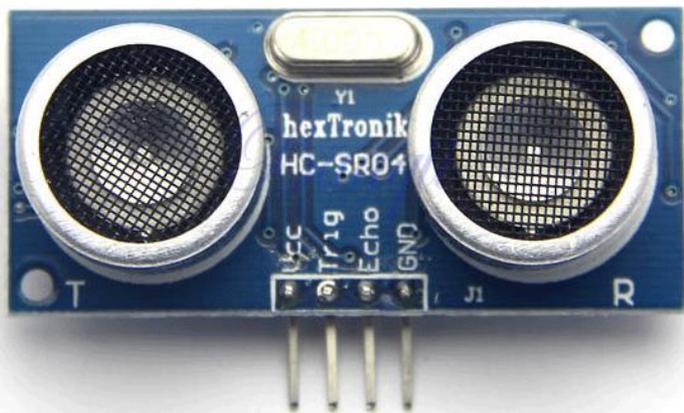
➤ 物理连接

- 取出一个蜂鸣器模块，用连接线将其与**8号管脚**对应的三个管脚相连（注意插线时颜色的对应）
- 取出一个超声波测距模块，用连接线按下图所示的方式相连



10 另类电子琴

任务：制作另类电子琴：根据距离变化发出不同的音



引脚	接线
VCC	5V
Trig	数字口#4（发射端）
Echo	数字口#7（接收端）
GND	GND

➤ 超声波传感器的具体工作原理为：

- 由超声波发射器向某一方向发射超声波，记发射时刻为 t_1 ，超声波在空气中传播，途中碰到障碍物就立即返回，记超声波接收器收到反射波的时刻为 t_2 ，则发射点距障碍物的距离 $S = 340 \times \frac{t_2 - t_1}{2}$ 。

10 另类电子琴

任务：制作另类电子琴：根据距离变化发出不同的音

➤ 软件编写

The image shows a Scratch code block for playing a sound. The block is titled "播放声音" (Play Sound) and has several fields: "管脚#" (Pin #) set to 6, "频率" (Frequency) set to "获取第" (Get Item) of an array named "tonelist" at index 1. The "tonelist" array is defined as a list of numbers: 1046.5, 1174.7, 1318.5, 1396.9, 1568, 1760, 1975.5. The "持续时间" (Duration) is set to 200 milliseconds. The "获取第" block is connected to a "约束" (Constraint) block. The "约束" block has "超声波测距(cm)" (Ultrasonic Distance in cm) as the variable, with "Trig#" set to 4 and "Echo#" set to 7. The "介于(最小值)" (Between) field is set to 0 and the "和(最大值)" (and Maximum) field is set to 69. The "约束" block is connected to a "除以" (Divide) block with the value 10, which is then connected to a "+" (Add) block with the value 1. The result of the addition is used to index into the "tonelist" array.

扩展任务：如何让电子琴发出更多的音？

创意电子设计

第10课



另类电子琴



创意电子设计



超声波测距仪

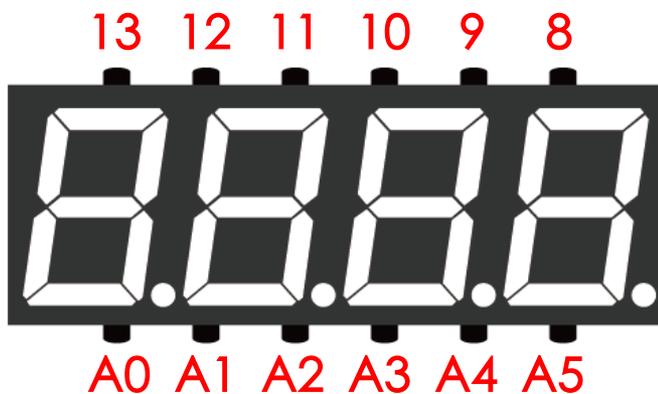


11 超声波测距仪

任务：制作超声波测距仪

➤ 物理连接

- 取出四位数码管模块，按下图所示的方式连接
- 取出一个超声波测距模块，依然采用第10课的连接方式

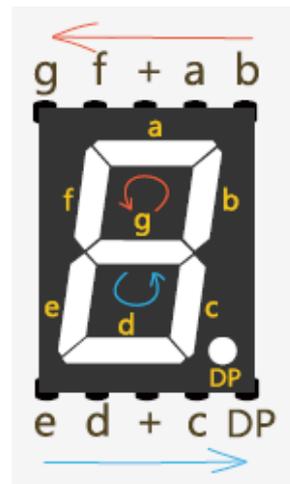


11 超声波测距仪

任务：制作超声波测距仪

➤ 八段数码管工作原理

- 数码管其实就是一个前面介绍的led的组合体，这个组合体包含8个led，所以也称之为八段数码管。说白了就八个灯。哪八段？不用多说了吧！a到g以及小数点DP。其实用法和前面说的LED也是一样的，每段都是一个发光二极管，分别用8个数字口来控制它们的亮灭，通过不同段的显示，就能组成0~9的数字。比如，我们让b, a, f, e, d, c亮起的话，就能显示一个数字“0”了。
- 数码管一共有10个引脚。a~DP 这8个引脚接到数字口，那还有两个引脚呢？这是公共端，LED只有一端是不能被点亮的。我们在RGB灯那章讲到过共阴共阳的问题，数码管也存在共阴共阳问题。所谓共阳就是公共端接+5V，共阴则是公共端接GND。

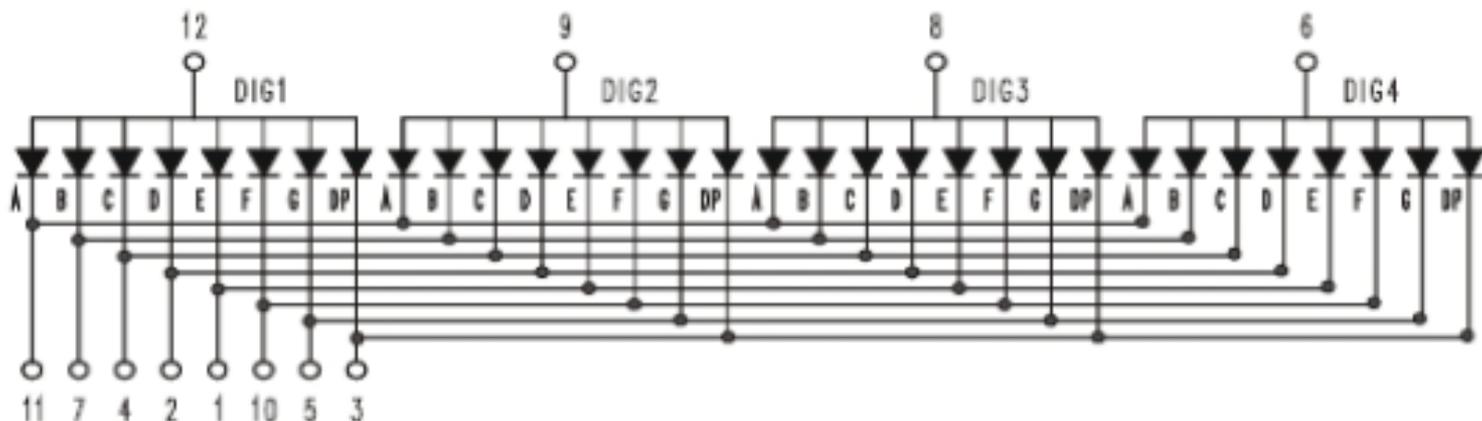


11 超声波测距仪

任务：制作超声波测距仪

➤ 四位数码管工作原理

- 四位数码管由四个八段数码管组成，其基本单元是发光二极管。
- 按发光二极管单元连接方式分为共阳极数码管和共阴极数码管。
- 左下角为1号脚，逆时针旋转编号。
- 8个管脚分别控制八段数码管的1段，4个管脚控制4位。



任务：制作超声波测距仪

- 软件编写
 - Mixly库功能的使用
 - 导入库



10 另类电子琴

任务：制作超声波测距仪

➤ 软件编写



扩展任务：你能看懂digit4函数吗？

能否修改函数，将输出的距离值精确到一位小数？

创意电子设计



超声波测距仪





公园人数计数系统



12 综合案例：公园人数计数系统

基础任务：制作公园人数计数系统

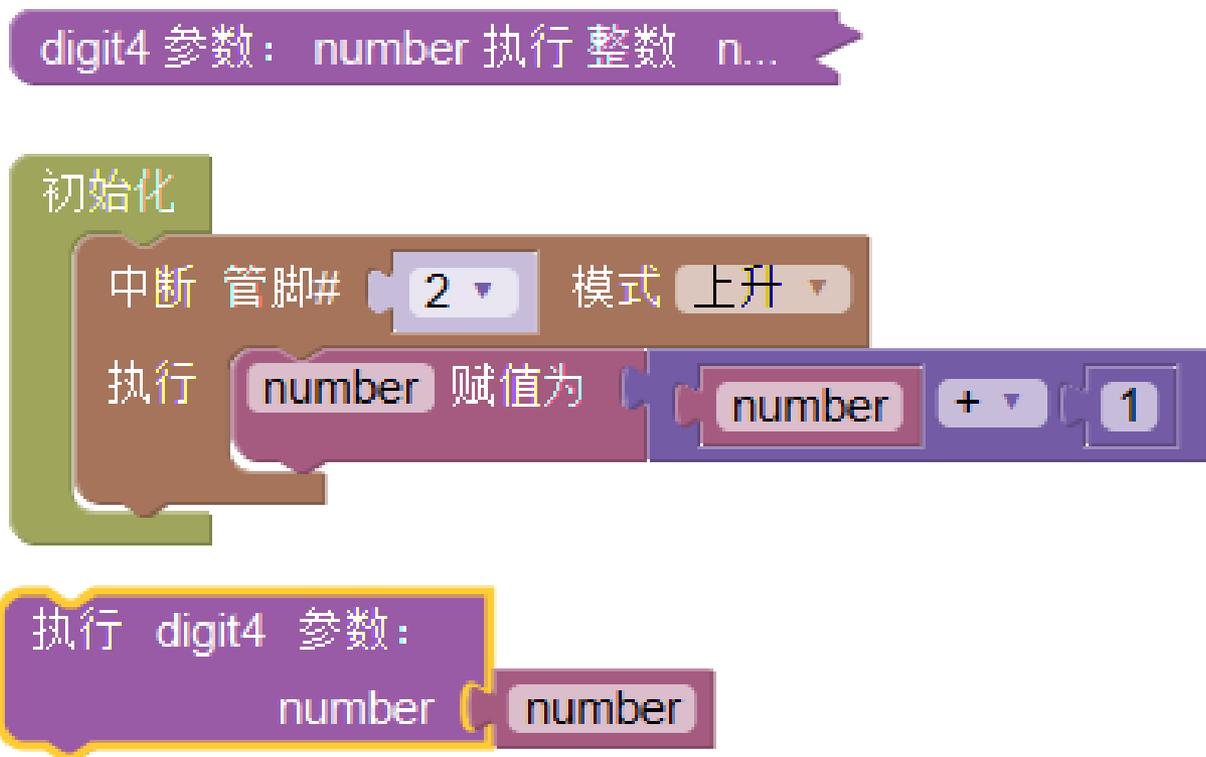
➤ 物理连接

- 取出**四位数码管**模块，依然采用第11课的连接方式
- 取出一个**红外接近开关**模块，连接**2号管脚**

12 综合案例：公园人数计数系统

基础任务：制作公园人数计数系统

➤ 软件编写



12 综合案例：公园人数计数系统

扩展任务：制作公园当前人数计数系统

➤ 物理连接

- 取出**四位数码管**模块，依然采用第11课的连接方式
- 两人一组，取出两个**红外接近开关**模块，分别连接**2、3号管脚**

12 综合案例：公园人数计数系统

扩展任务：制作公园当前人数计数系统

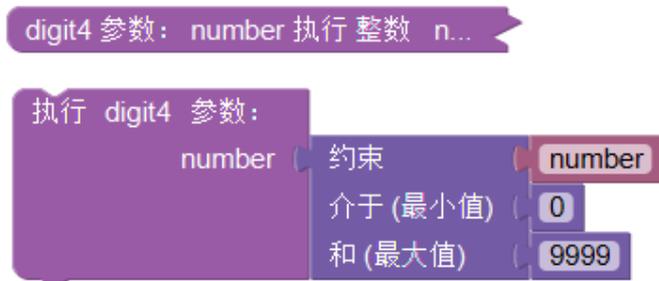
➤ 软件编写

- 2~3表示游客进入；3~2表示游客出去
 - 如何使用程序表示这一过程？

12 综合案例：公园人数计数系统

扩展任务：制作公园当前人数计数系统

➤ 软件编写



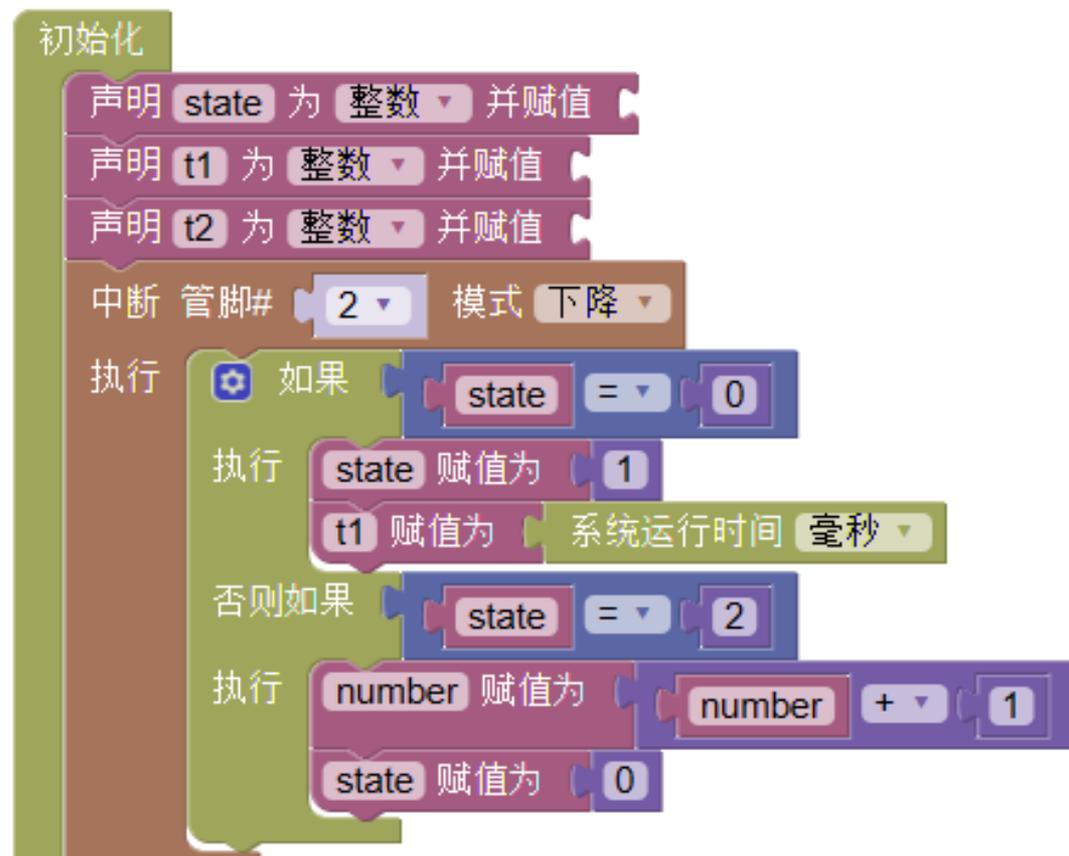
如果有人捣乱呢？



12 综合案例：公园人数计数系统

扩展任务：制作公园当前人数计数系统

➤ 软件编写

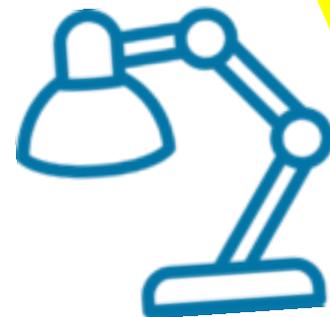


12 综合案例：公园人数计数系统

扩展任务：制作公园当前人数计数系统

➤ 软件编写





公园人数计数系统





创意电子任务设计与展示



13 创意电子综合任务设计与展示

- 两人一组（一位老师+一位小朋友）
- 设计并开发一个足够复杂的任务
 - 可参考公园人数计数系统、音乐盒等



创意电子任务设计与展示

