



米思齐创意电子培训

北师大教育学部创客教育实验室

- 米思齐（Mixly）是由北京师范大学教育学部创客教育实验室开发的面向中小学创客教育（创意电子领域）的开源图形化编程软件，它用直观的图形化积木块堆叠方式代替了复杂的文本编辑，具有入门简单、使用方便、功能强大、应用广泛、易于扩展的优势。





米思齐入门套件

- 基于磁吸连接
- 安全性非常高
- 便于连接乐高
- 适合小学使用



米思齐标准套件

- 基于定制导线
- 安全性比较高
- 连接规律明显
- 适合初中使用



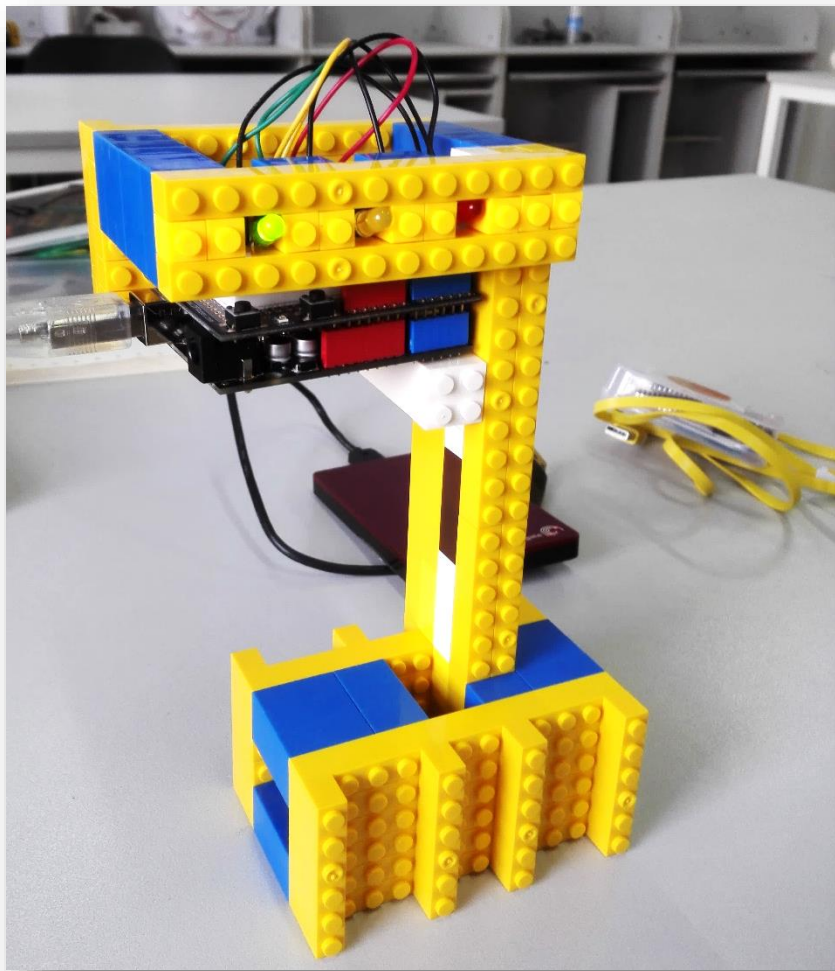
米思齐专业套件

- 基于面包板
- 灵活程度较高
- 连接较易出错
- 适合高中使用

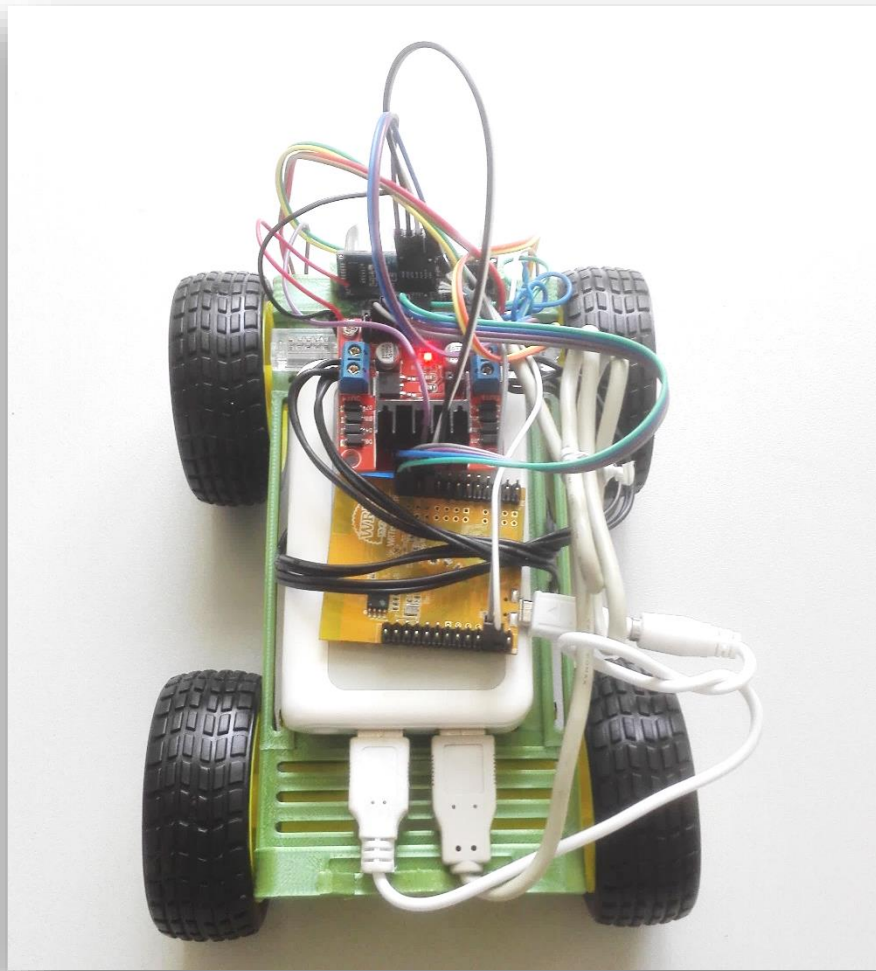


- Mixly官方交流入门群：199124078

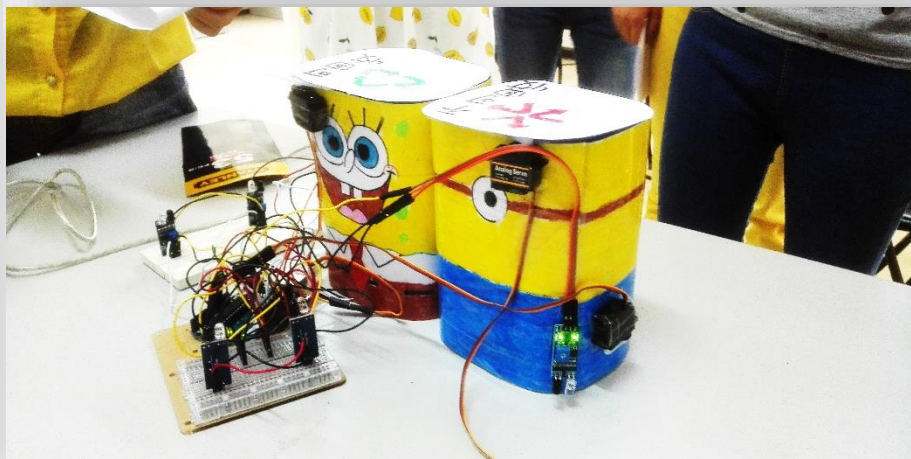
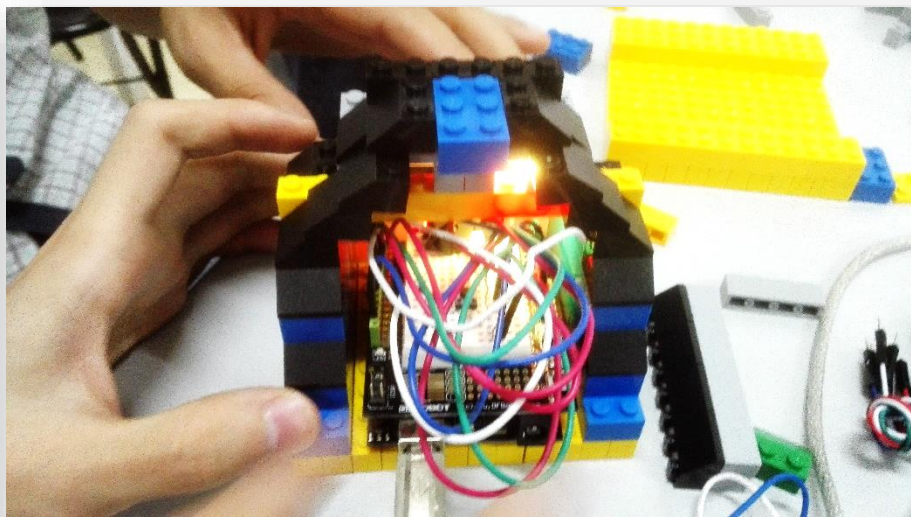
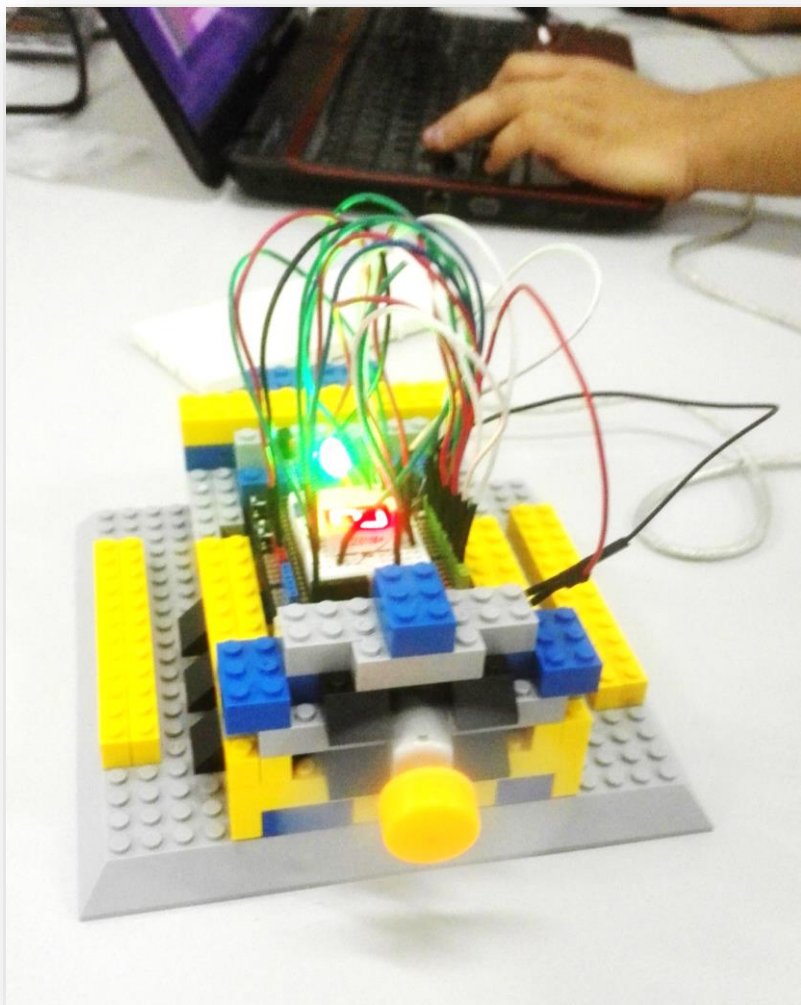
- Arduino是一个开放源码电子原型平台，拥有灵活、易用的硬件和软件(板子及在此之上的软件)。
- Arduino可以接收来自各种传感器的输入信号从而做到监测环境的效果，并通过控制光源，电机以及其他执行器来影响其周围环境(类似于人类的工作)。
- 硬件低廉，软件免费。
- 可以完成的例子：
 - 当咖啡煮好时，咖啡壶就发出“吱吱”声提醒
 - 当邮箱有新邮件时，电话就会发出铃声通知
 - 自制一个心率监测器，将每次骑脚踏车的记录存进存储卡
 - 复制一张门禁卡、饭卡
 - ...



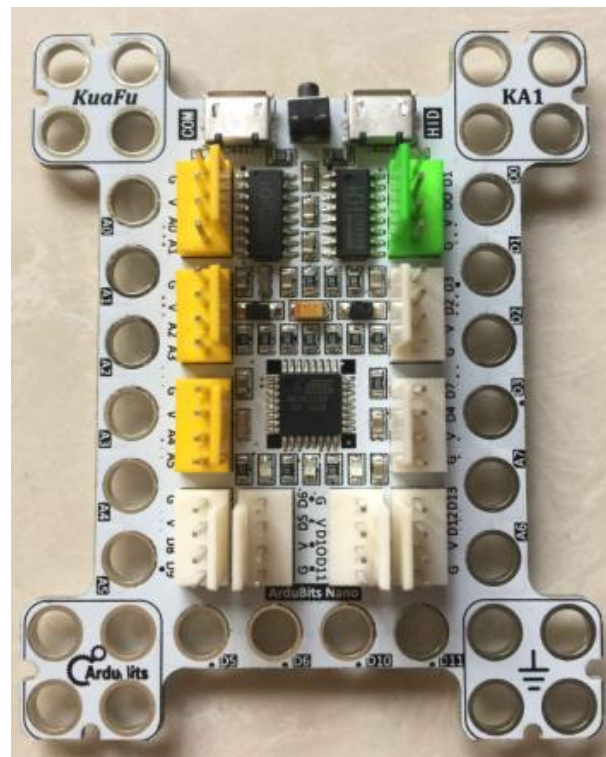
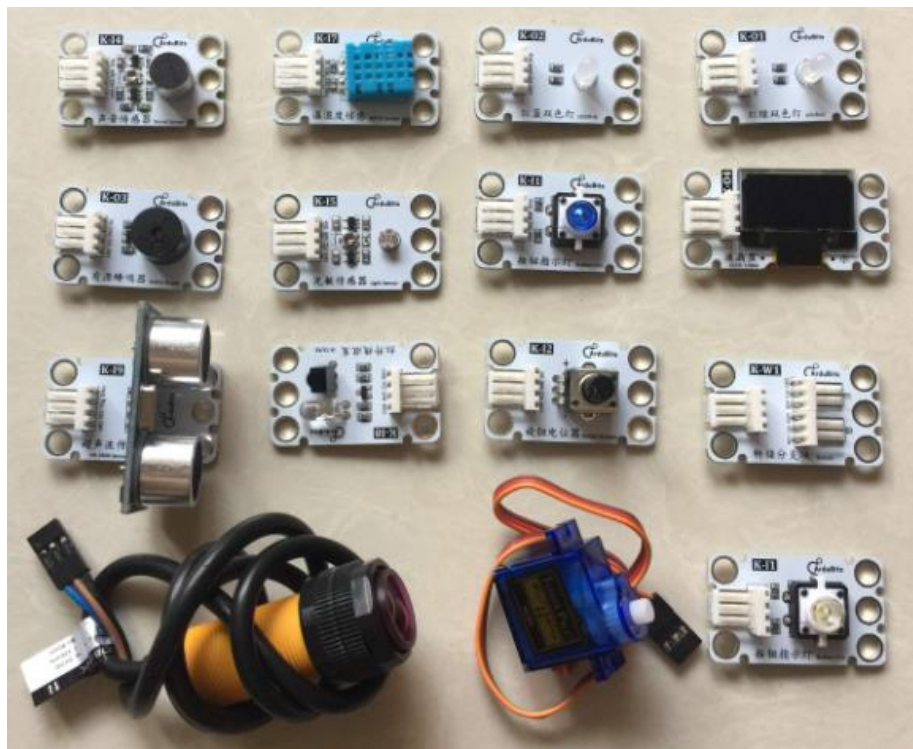
学生作品：交通灯



学生作品：智能小车



创意电子与外壳结构搭配完成成型的创意作品



名称	数量	名称	数量	名称	数量	名称	数量
主控板	1	按钮指示灯	2	声音传感器	1	USB线	1
LED模块	2	舵机	1	旋钮电位器	1	鳄鱼夹线	8
温湿度传感器	1	蜂鸣器	1	光敏传感器	1	4pin端子线	6
超声波传感器	1	红外遥控	1	液晶显示屏	1	转接分支模块	1

◆ Arduino Nano

Arduino Nano是Arduino USB接口的微型版本，最大的不同是没有电源插座以及USB接口是Mini-B型插座。Arduino Nano处理器核心是[ATmega168](#)(Nano2.x)和[ATmega328](#)(Nano3.0)。

- 14路数字I/O口
- 6路PWM输出
- 8路模拟输入

- 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://mixly.org/> → 软件平台 → Mixly 官方版

爱上米思齐 (Mixly)

[首页](#)[探索](#)[课程](#)[套件](#)[控制板](#)[硬件模块](#)[软件模块](#)[软件平台](#)[应用](#)[米思齐讲师](#)[注册](#)[登录](#)

[首页](#) / [软件平台](#) / [Mixly 官方版](#)

[更新时间](#)[发布时间](#)[评价](#)

1/1



Mixly For Arduino (Mac)

米思齐图形化编程工具 Mac 版

最近更新: 2017-04-21 | 最新版本: 0.995

软件平台 - Mixly 官方版

★ 0



Mixly For Arduino (Windows)

米思齐图形化编程工具 Windows 版

最近更新: 2017-04-21 | 最新版本: 0.995

软件平台 - Mixly 官方版

★ 3



Mixly For Arduino (python-web)

米思齐图形化编程工具 Python 版

软件平台 - Mixly 官方版

★ 0

- 下载后，右键解压到Mixly目录下

–  arduino-1.7.10	← Arduino官方案序
–  blockly	← Mixly主要图形代码
–  company	← 公司扩展库
–  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.exe 类型: 应用程序	← .exe文件，Win用户双击该文件运行Mixly
–  Mixly.jar 类型: Executable Jar File	← .jar文件，Mac用户双击该文件运行Mixly

• 物理连接

– 通过USB线把Arduino UNO和电脑相连

• 驱动安装

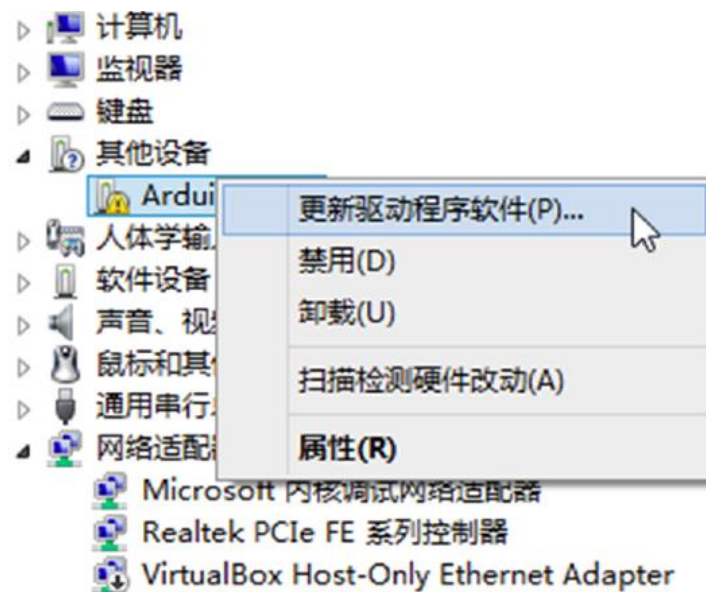
– 驱动程序在arduino-1.7.10\drivers中

– 安装成功后会出现一个串口

– 驱动修复：FixDriverTool (.Net 4.0)

• 启动Mixly

– 双击Mixly目录下的mixly.exe





Mixly 1.0.0

模块

- 输入/输出
- 控制
- 数学
- 文本
- 数组
- 逻辑
- 串口
- 通信
- 传感器
- 执行器
- 显示器
- 变量
- 函数

代码

Copyright © 北师大教育学部创客教育实验室 maker.bnu.edu.cn

普通视图 高级视图 简体中文

Arduino Micro

COM6

串口监视器

上方出现的板卡型号和串口号均与Arduino主控板对应一致即成功



让我们开始吧！



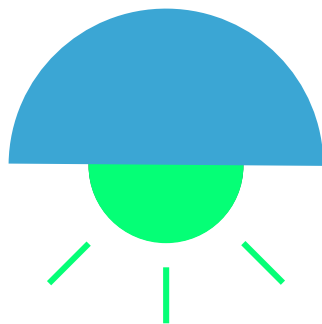
第1课

Lesson 1

Hello World

任务发布

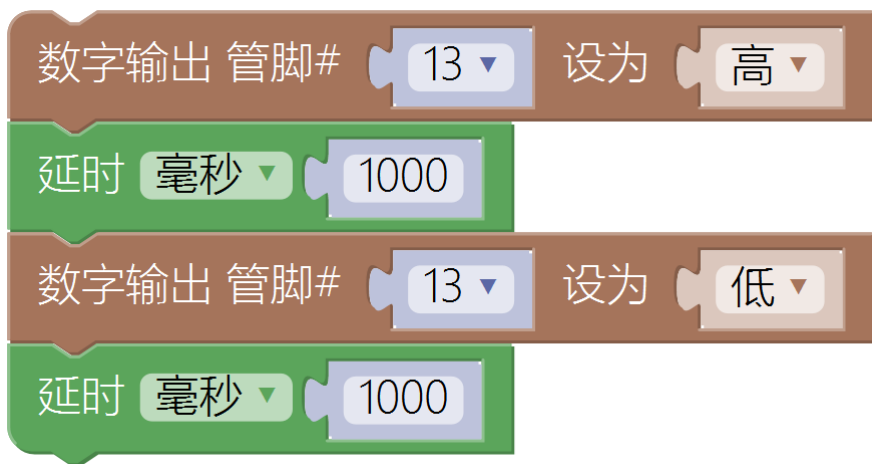
- 点亮板载LED灯（13号管脚），亮1秒灭1秒



- 打开Mixly后，单击**打开**，找到“**01闪烁LED.xml**”并打开。
- 检查板卡型号与串口号无误后**上传**。



程序解读



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：

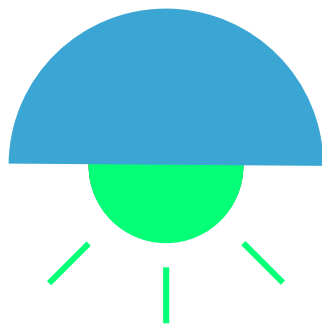
- 可以选择毫秒（1/1000秒）或微秒

说一说

您能简单描述一下程序的工作过程吗？

任务发布

- 点亮LED灯（10号管脚），亮1秒灭1秒

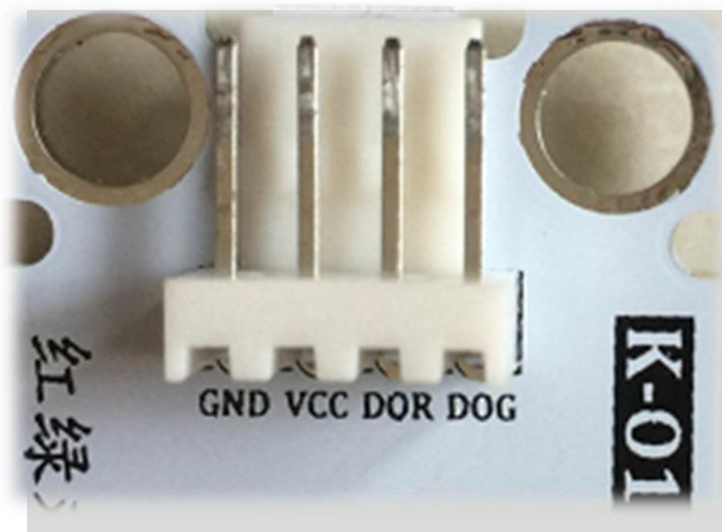


认识LED

- LED即发光二极管（Light Emitting Diode）
- LED灯模块，可发出光亮，LED灯有很多种颜色，比如蓝色、黄色、绿色、红色
- 优点：体积小、耗电量低、使用寿命长、高亮度、低热量等

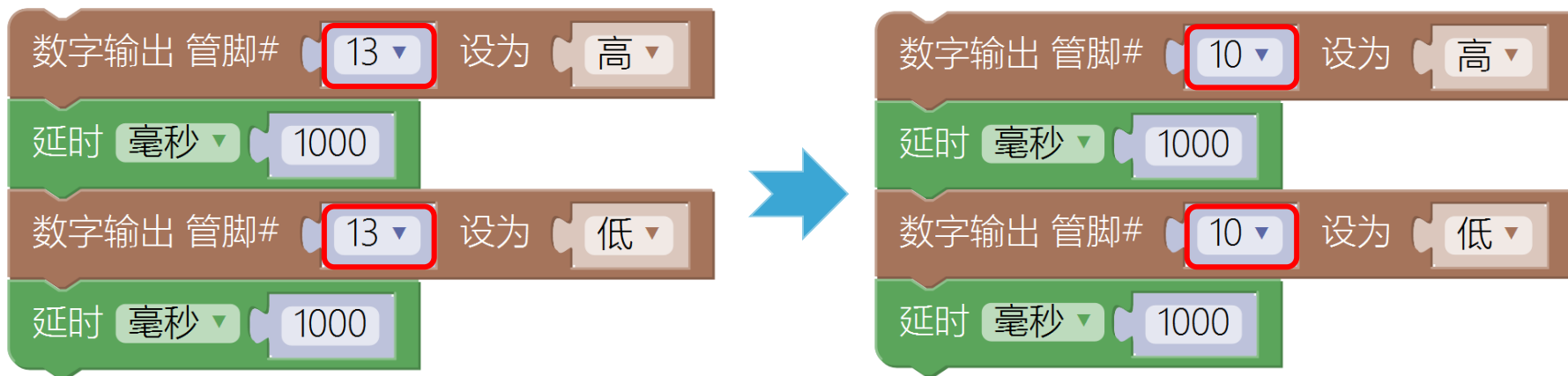


认识LED



- 第一个口为：电源-GND
 - GND：指的是电线接地端Ground的简写。代表地线或零线。
- 第二个口为：信号针脚 OUT
 - 表示输出信号，可能是高电平、低电平，信号输出情况由主控板控制。
- 第三个口和第四个口为信号线，所以连接10号管脚应接在**G V 10 11**上。
-

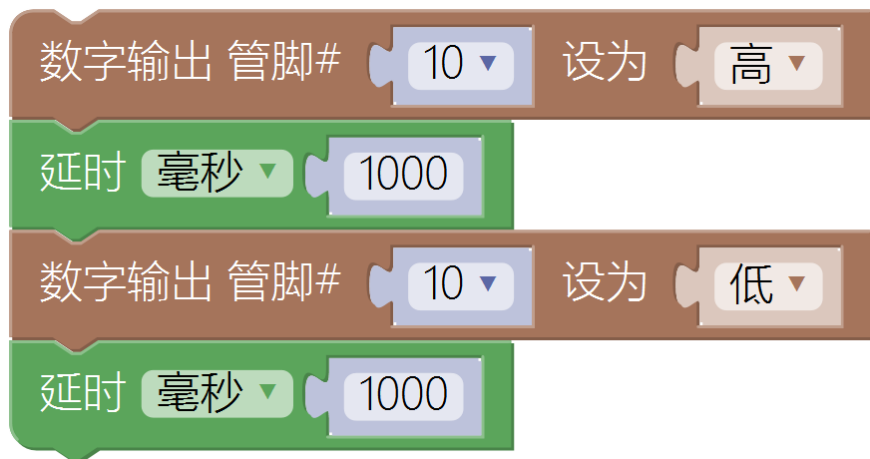
- 打开Mixly后，单击**打开**，找到**“01闪烁LED.xml”**并打开。
- 将**数字输出管脚#13**改为**10**。
- 检查板卡型号与串口号无误后**上传**。



试一试

将延时分别改为100、10、1，会有什么效果？

程序解读



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）

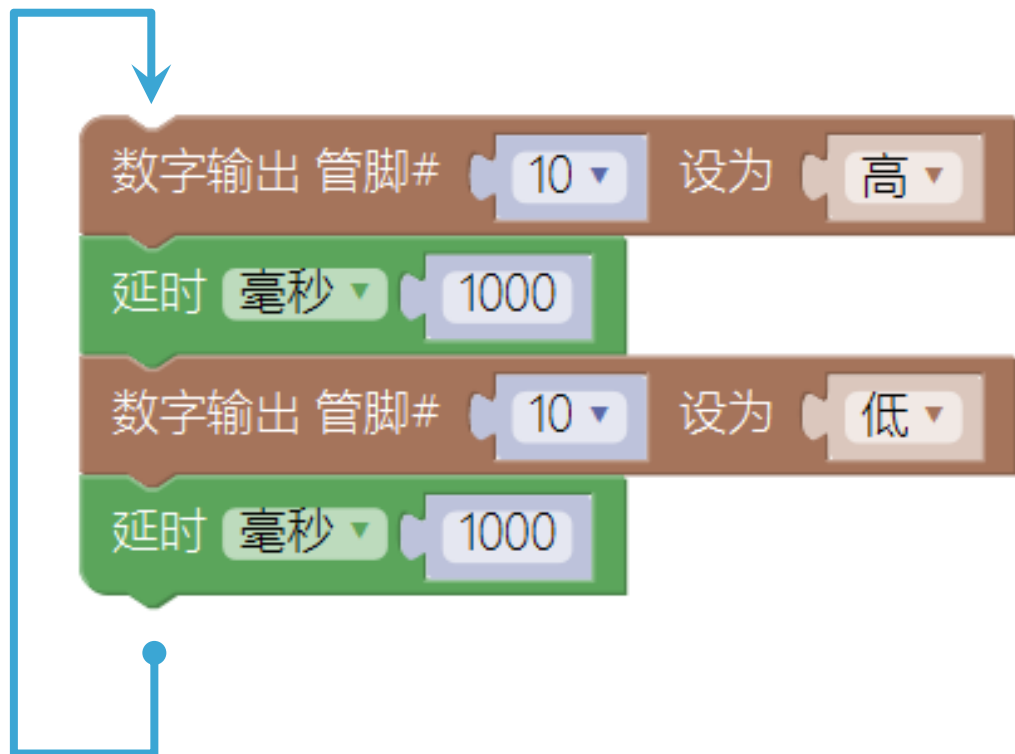
延时：

- 可以选择毫秒（1/1000秒）或微秒

说一说

您能简单描述一下程序的工作过程吗？

程序解读

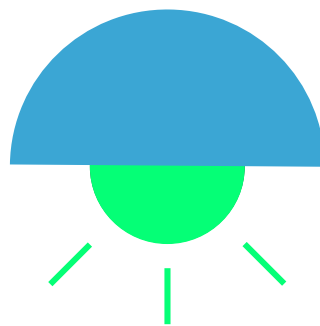
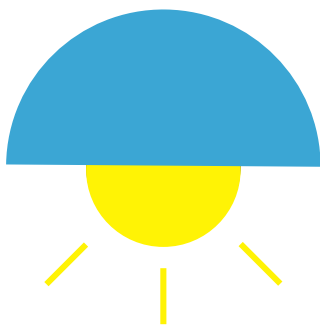


```
void setup()
{
  pinMode(10, OUTPUT);
}

void loop()
{
  digitalWrite(10, HIGH);
  delay(1000);
  digitalWrite(10, LOW);
  delay(1000);
}
```

扩展任务

- 请使用两个LED灯，编写程序，完成如下的实验效果：两个LED灯交替亮灭，每2秒钟完成一组交替。

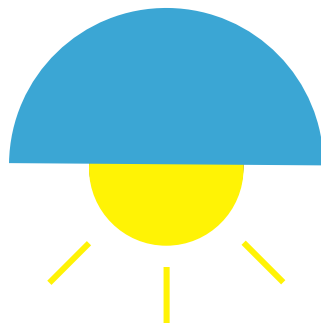


参考代码



扩展任务

- 使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**





让我们一起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



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？



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。

i

1000
900
800
700
600
500
400
300
200
100

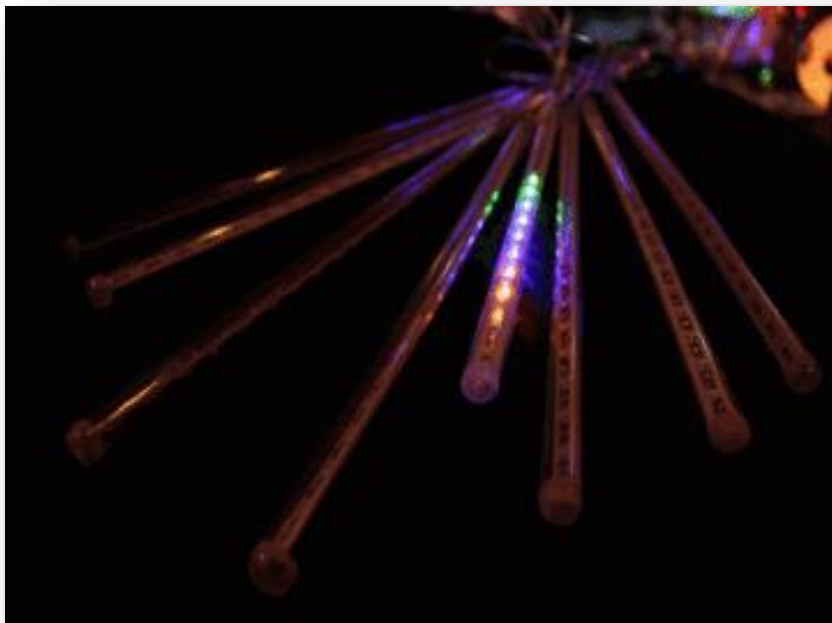


如何理解“使用 i 从 1 到 15 步长为 2”？

- 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？

思考

- 如何让更多的灯呈现更多样的效果？





第2课

Lesson 2

神奇的开关

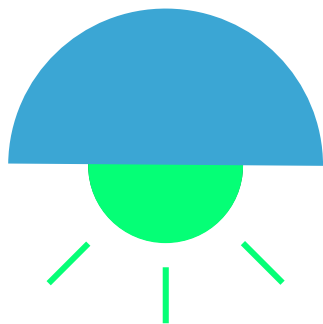
第2课 神奇的开关

- 开关是我们接触的**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**。



任务发布

- 程序上传后，按下开关时，LED灯点亮；松开开关时，LED灯熄灭。

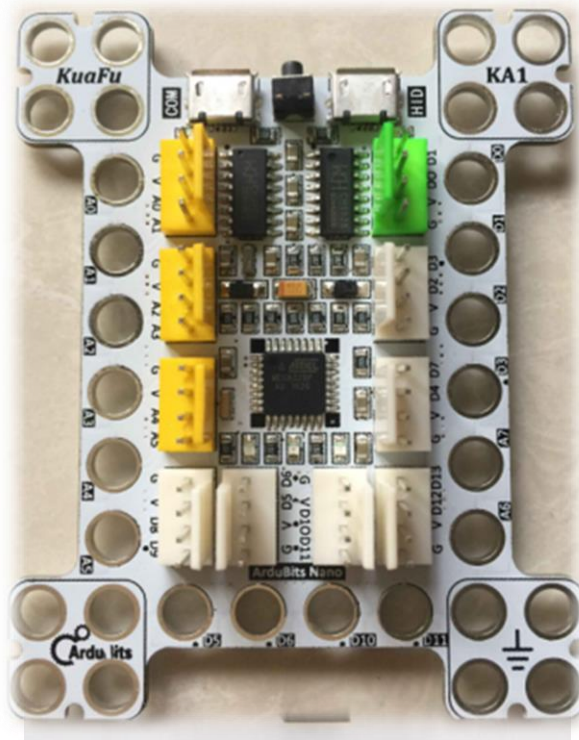


第2课

神奇的开关

让开关简单工作——按下灯亮，抬起灯灭

思路分析



当开关被按下时，向2号管脚输入___电平，此时10号管脚应输出___电平；
当开关被抬起时，向2号管脚输入___电平，此时10号管脚应输出___电平。

软件编写

数字输出 管脚#

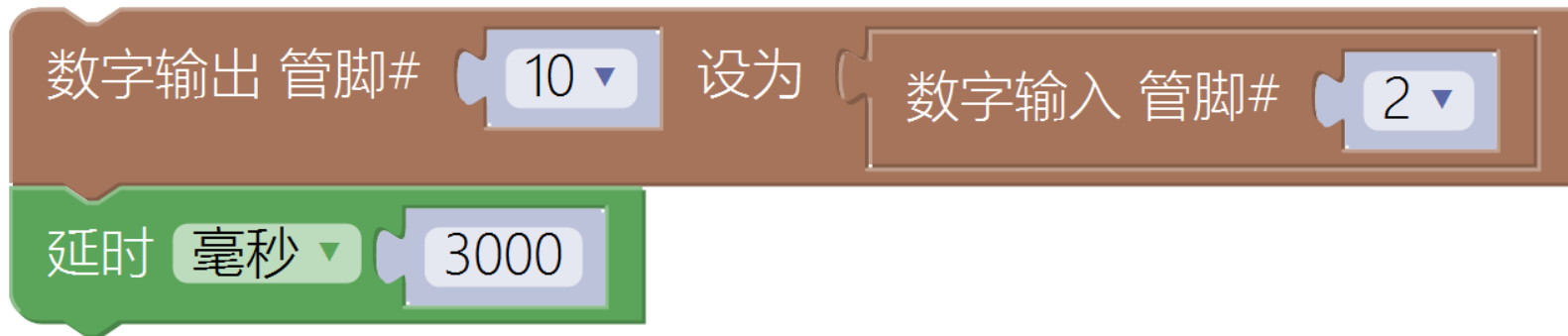
10 ▼

设为

数字输入 管脚#

2 ▼

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

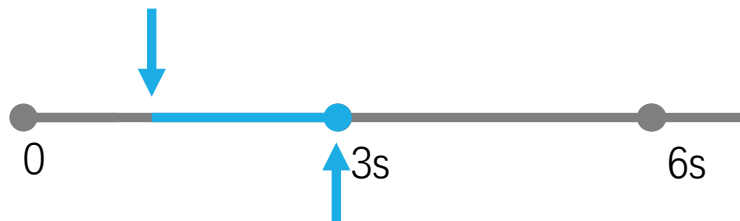


试一试

这样的思路可以完成任务吗？
如果不行，问题出在哪里？

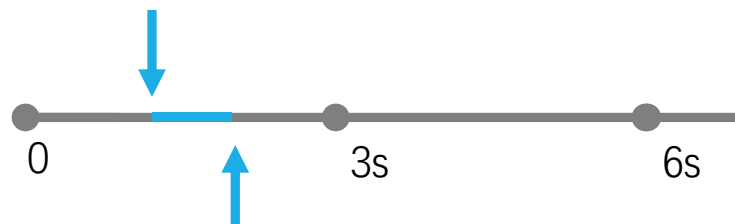
- 在Arduino中，程序会被循环执行。
- 除了数字输出语句执行的那一时刻外，其余的时间程序都被延时语句“锁住”了。
- 因此，按下开关后灯不会立刻被点亮！

此时按下按钮（保持按下）



此时命令才会被触发

此时按下按钮（保持按下）



此时松开 命令不会被触发

软件编写

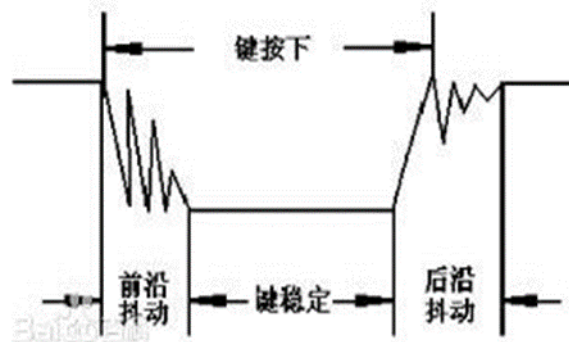
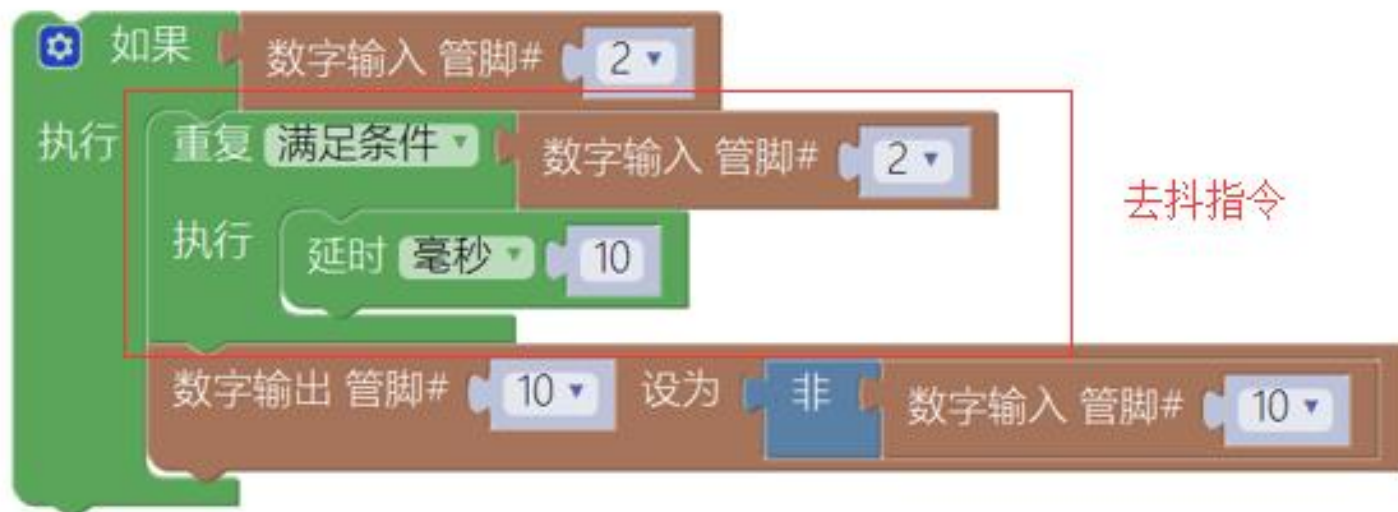
**如果：**

- 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。

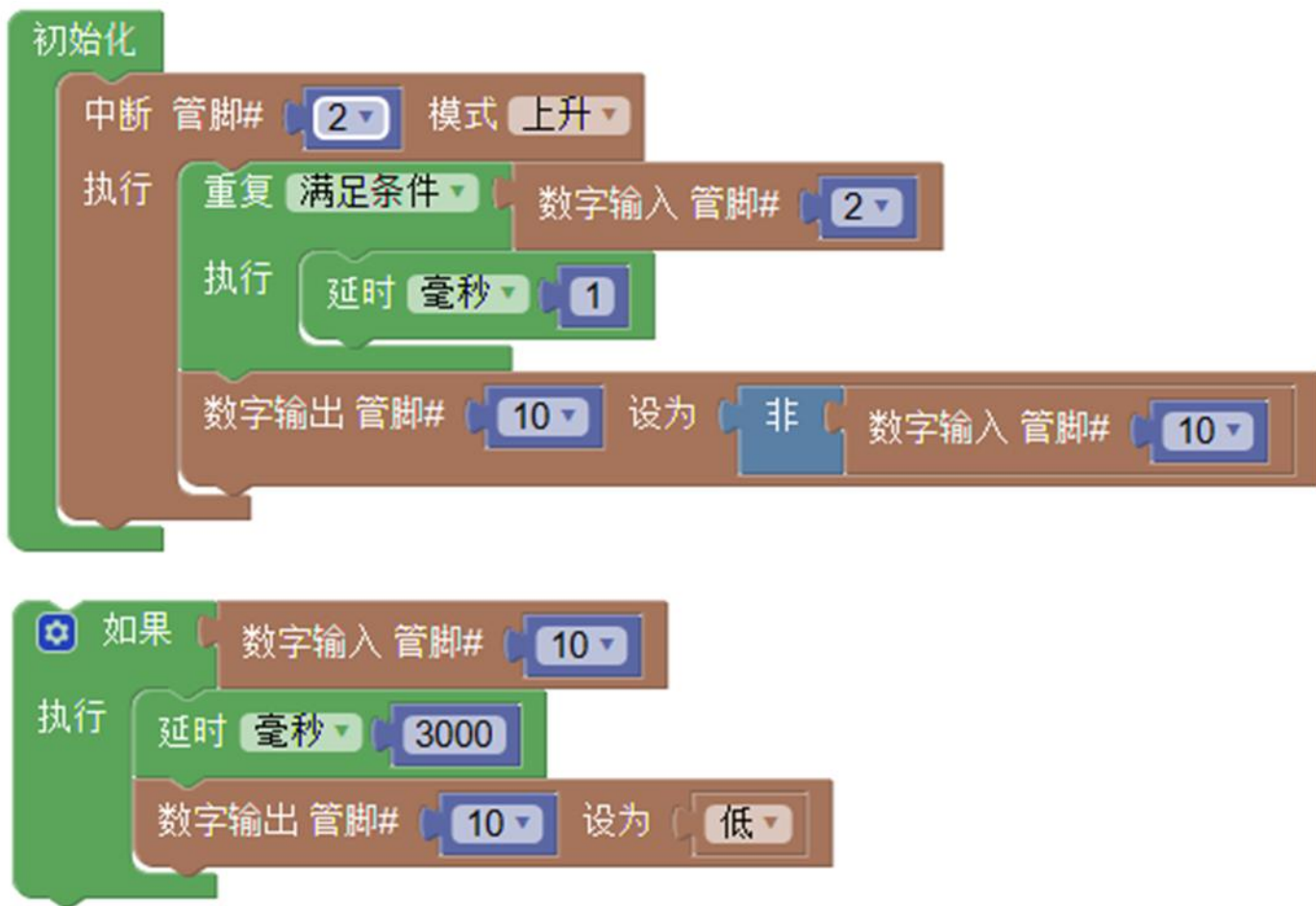
任务发布

- 程序上传后，按下开关时，LED灯点亮；再次按下开关时，LED灯熄灭。

软件编写



软件编写





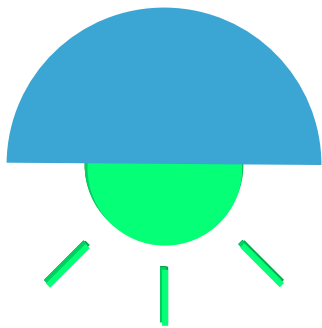
第3课

Lesson 3

呼吸灯

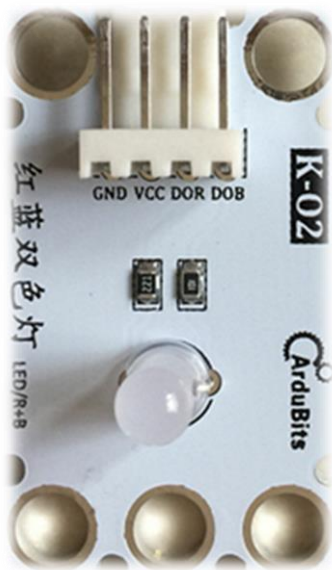
任务发布

- 呼吸灯的灯光在微电脑的控制下，可以完成由暗到亮再由亮到暗的逐渐变化的过程，感觉像是在呼吸。
- 请使用LED灯，编写程序完成如下的实验效果：程序上传后，LED灯先逐渐变亮再逐渐变暗。



硬件连接

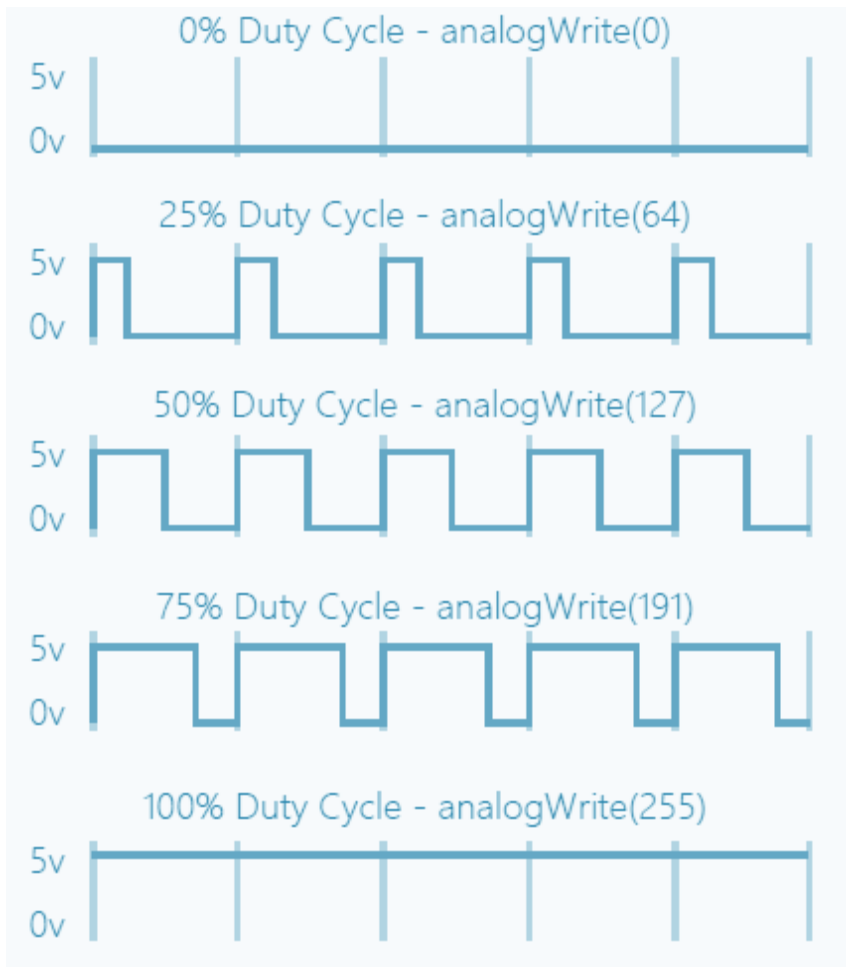
- 取出一个LED模块，用连接线将其与**10号管脚**对应的那组管脚相连（注意插线时颜色的对应）



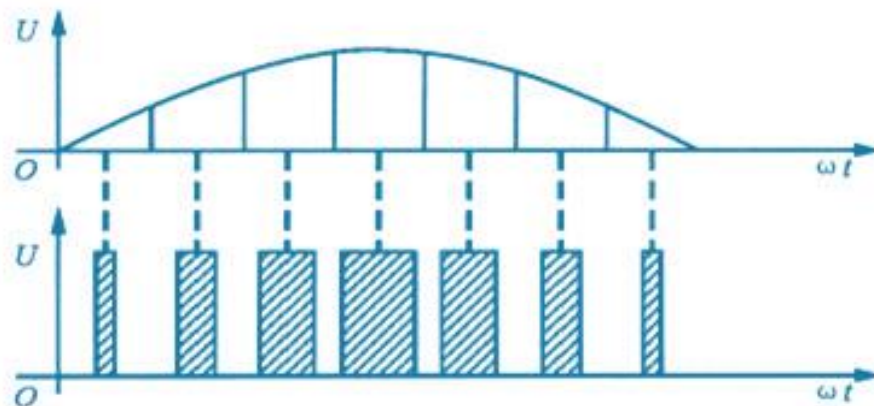
试一试

如何利用现有知识改变灯的亮度？

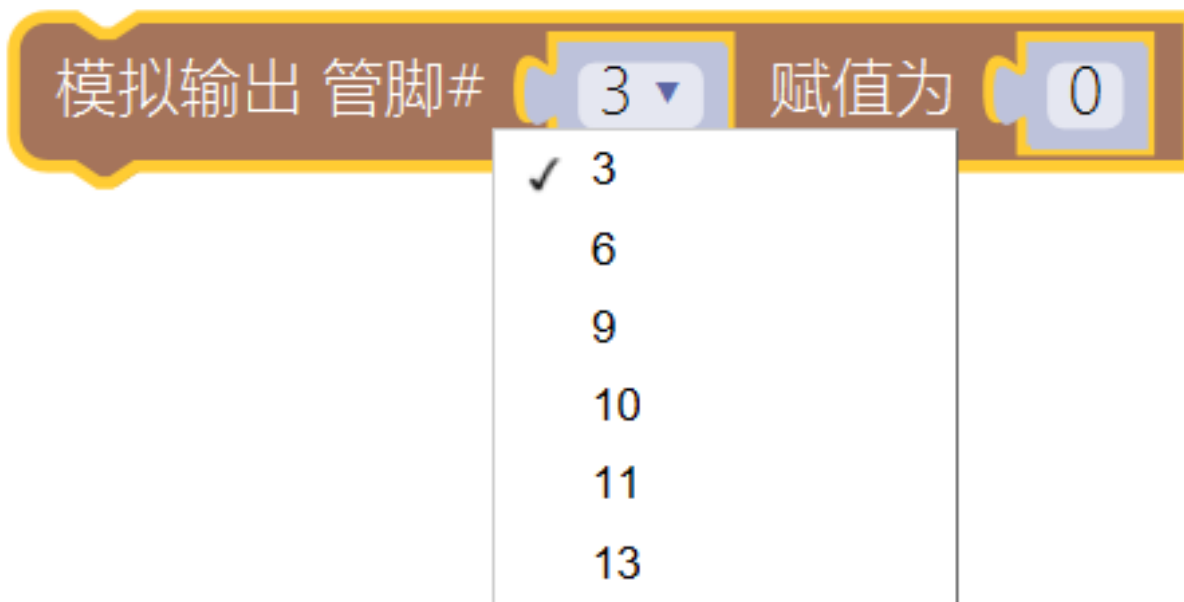
知识讲解：PWM



- Arduino模拟输出的取值范围是0-255(8位)，如左图。
- Arduino Uno主控板只有六个管脚（3、5、6、9、10、11）支持PWM。



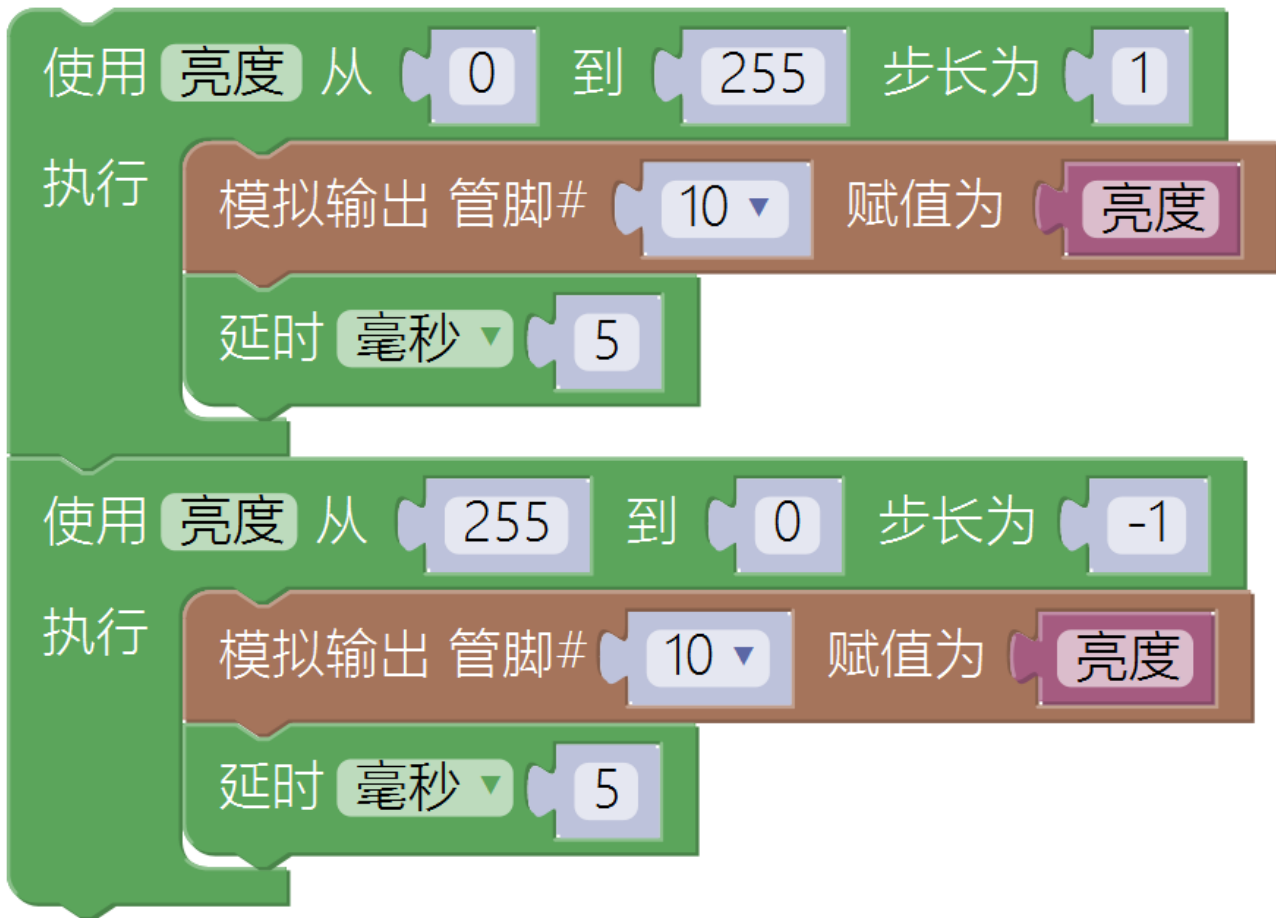
软件编写



模拟输出：

- 支持管脚：3、5、6、9、10、11、13（不用记住）
- 取值范围：0~255（需要记住）

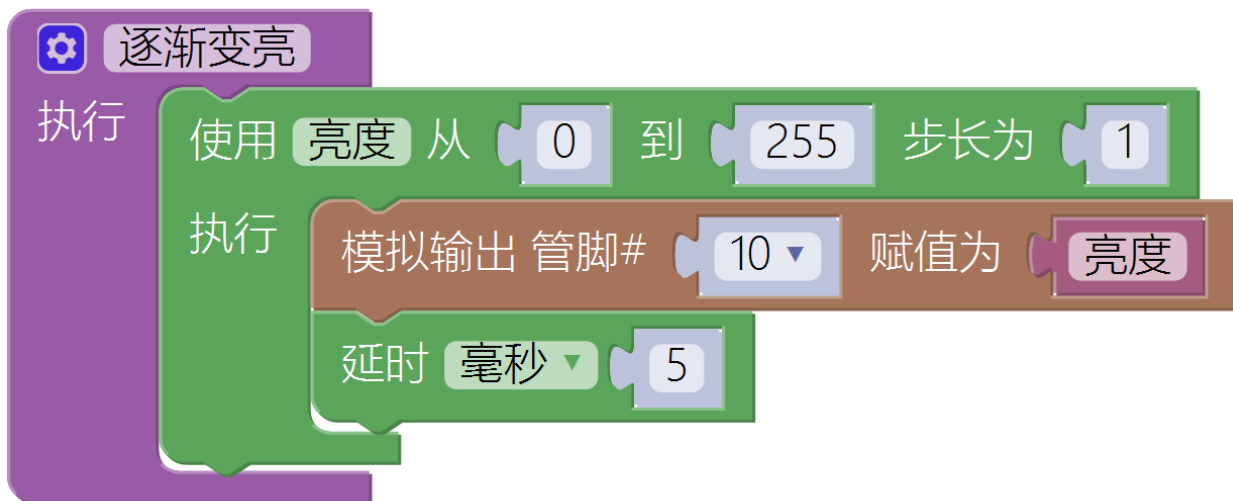
软件编写①



软件编写②

- **函数**是用户定义的一组代码块，使用时需**先定义再调用**。
- 编写函数的目的是为了便于今后重复使用，减少重复编写程序的工作量，提高程序的重用性和可读性。

定义→



调用→



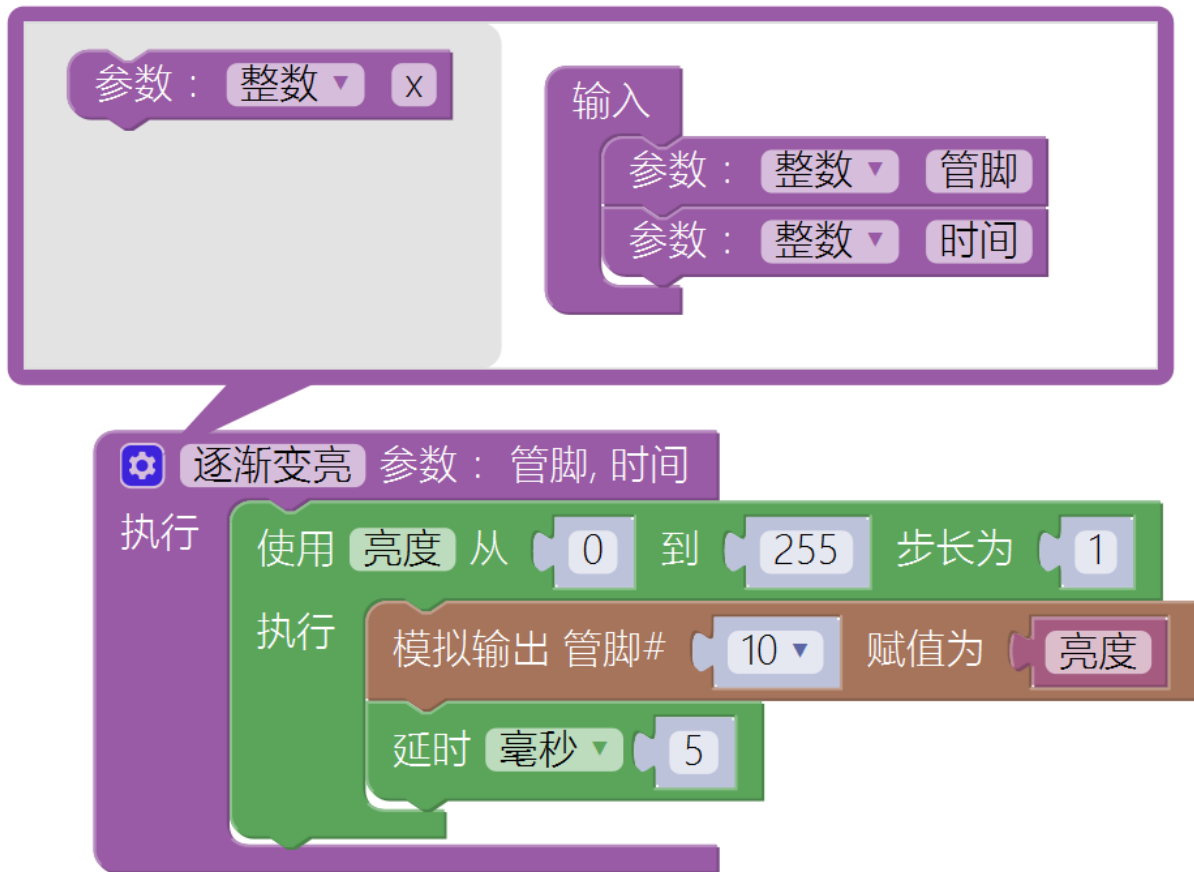
软件编写③

- 为函数添加**参数**



软件编写③

- 为函数添加**参数**



软件编写③

- 为函数添加**参数**

逐渐变亮 参数：管脚, 时间

执行

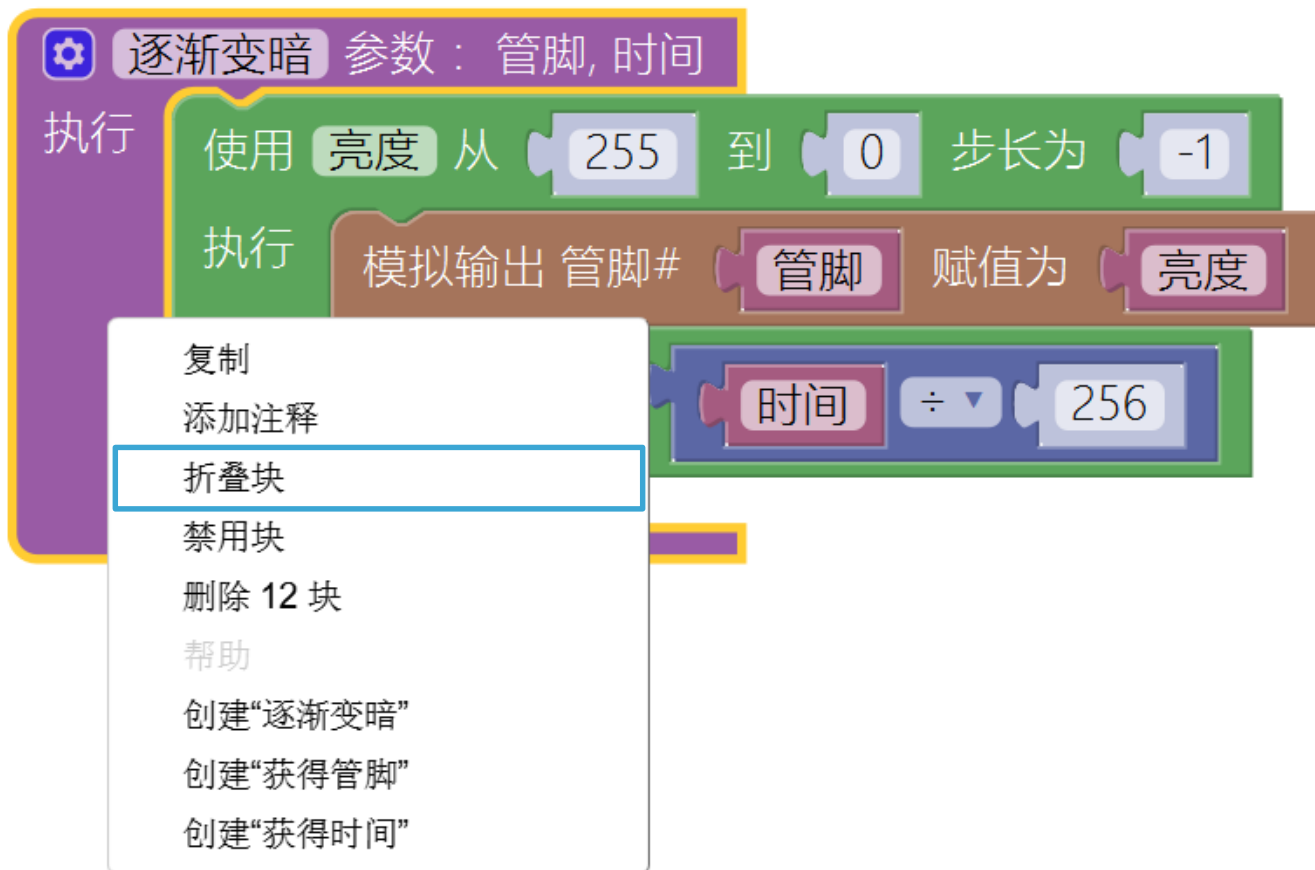
- 使用 亮度 从 0 到 255 步长为 1
- 执行 模拟输出 管脚# 管脚 赋值为 亮度
- 延时 毫秒 时间 ÷ 256

执行 逐渐变亮 参数：

- 管脚 10
- 时间 1000

软件编写③

• 折叠块



软件编写③

- 单行输入与外部输入



软件编写③

- 用户不必了解函数内部的操作过程，只需**执行（调用）函数**即可实现指定功能。

逐渐变亮 参数：管脚, 时间 执行 使用 亮度从 ...

逐渐变暗 参数：管脚, 时间 执行 使用 亮度从 ...

执行 逐渐变亮 参数：管脚 时间

执行 逐渐变暗 参数：管脚 时间



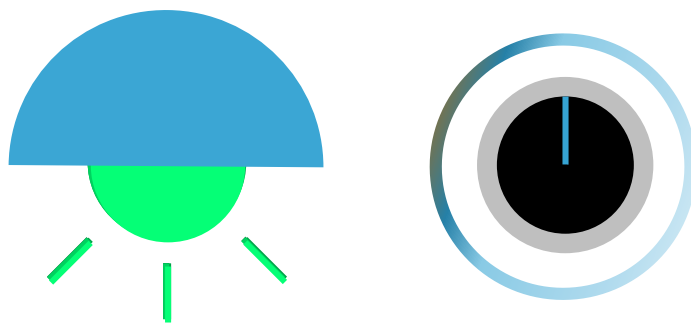
第4课

Lesson 4

旋钮可调灯

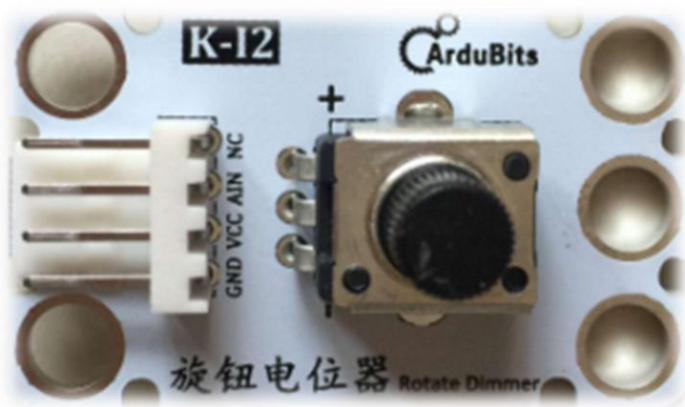
任务发布

- 使用电位器和LED灯，制作一个可调灯，编写程序实现以下效果：
- 程序上传后，通过旋转电位器的旋钮，改变LED灯的亮度。



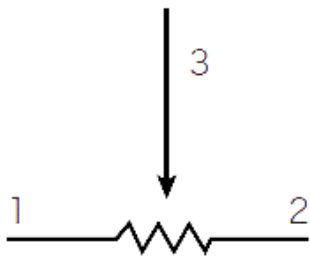
硬件连接

- 取出一个LED模块，用连接线将其与**10号管脚**对应的那组管脚相连（注意插线时颜色的对应）
- 取出一个**旋钮**电位器模块，用连接线将其与**A1号管脚**对应的那组管脚相连（注意插线时颜色的对应）



电位器：

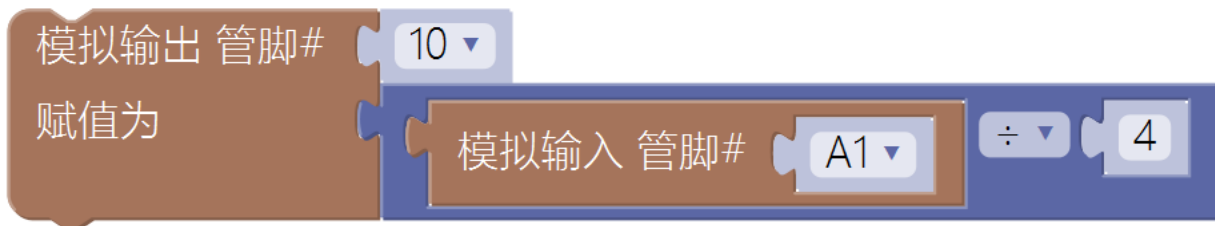
- 通过旋转旋钮变化阻值
- 使用模拟输入管脚读取



软件编写

• 模拟输入：

- 支持管脚：A0~A11（根据板子确定）
- 取值范围：0~1023



模拟输入 管脚#

A1 ▾

A0

✓ A1

A2

A3

A4

A5

A6

A7

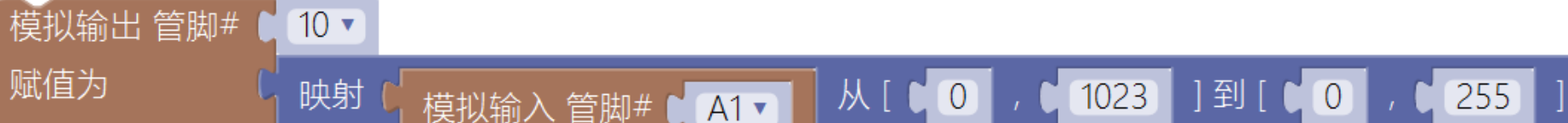
A8

A9

A10

A11

- **映射**：从[a,b]映射到[c,d]，进行线性变换，c不一定小于d。





第5课

Lesson 5

声控灯

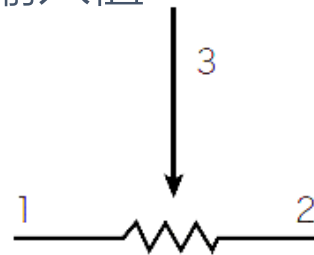
硬件连接

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



声音传感器：

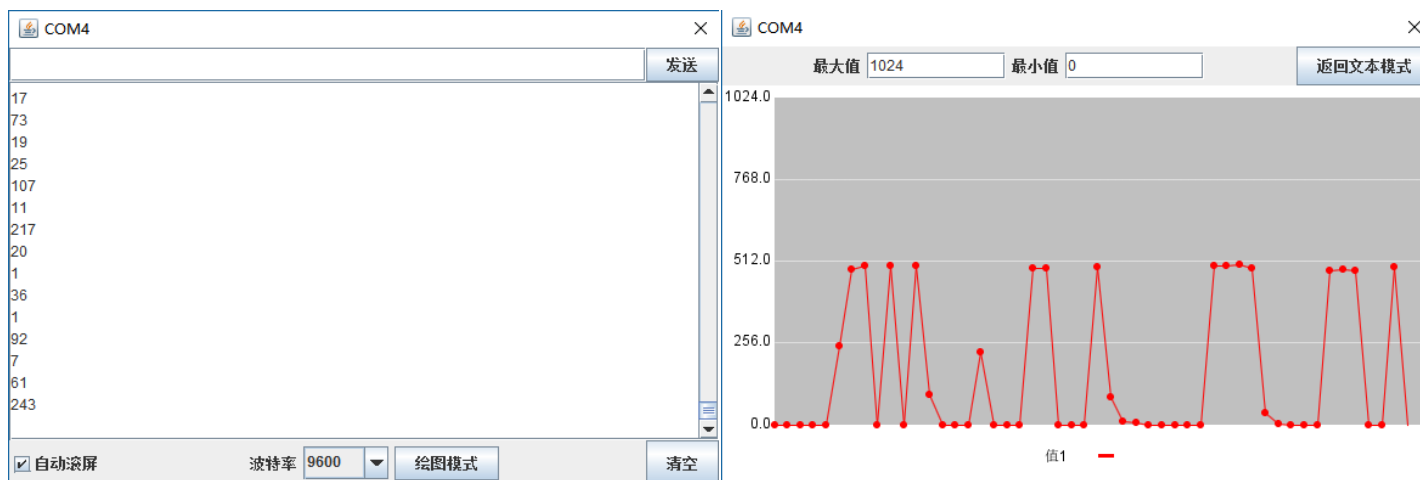
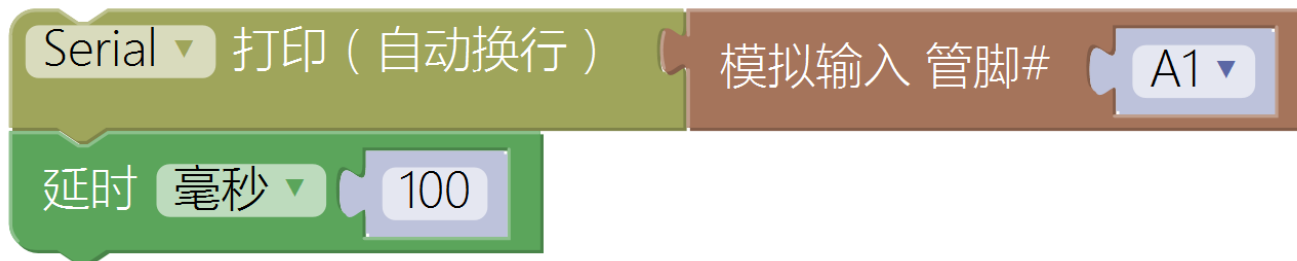
- 探测声音变化
- 使用模拟输入管脚读取
- 声音大小会影响模拟输入值



软件编写

• 串口打印

- 便于调试，经常应用于模拟输入传感器阈值的测试（但不是必须的）

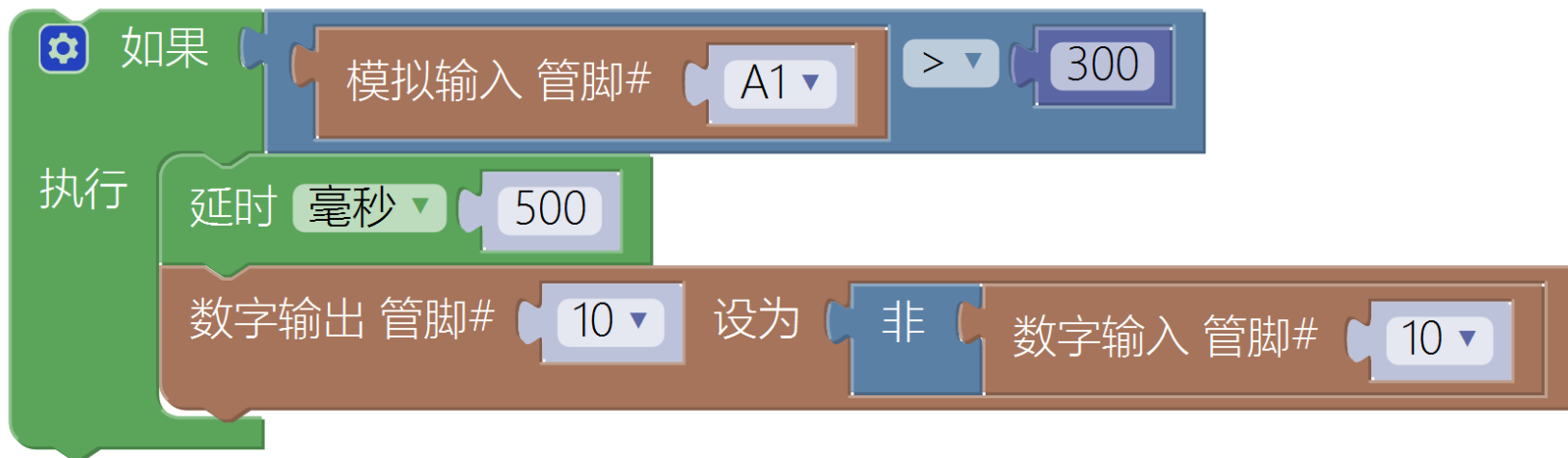


软件编写



- 思考：如何拍手控制LED灯的亮灭？拍一下亮，再拍一下灭

软件编写





第6课

Lesson 6

遥控灯

硬件连接

- 取出一个LED模块，用连接线将其与**10号管脚**对应的那组管脚相连（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**12号管脚**对应的那组管脚相连（注意插线时颜色的对应）

现实中的红外遥控器

- 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。



套件中的红外遥控器

- 每个开关都有一个特定的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：



数据存储基础

- 数据：计算机的加工对象
- 位：数据存储的最小单位
 - 在计算机中的二进制数系统中，位(bit)，简记为b，也称为比特，每个0或1就是一个位
- 位串：由若干位组合起来形成位串
- 流 (stream)：一个长的位串
 - 非常长
 - 人脑不容易理解
 - 如何简化这种位模式的表示方法？

数据存储基础

• 用一个符号表示位模式的4位

– 例如，一个16位串只需要_____个符号就可以表示

位模式	十六进制表示	位模式	十六进制表示
0000	0	1000	8
0001	1	1001	9
0010	2	1010	
0011	3	1011	
0100	4	1100	
0101	5	1101	
0110	6	1110	
0111	7	1111	

数据存储基础

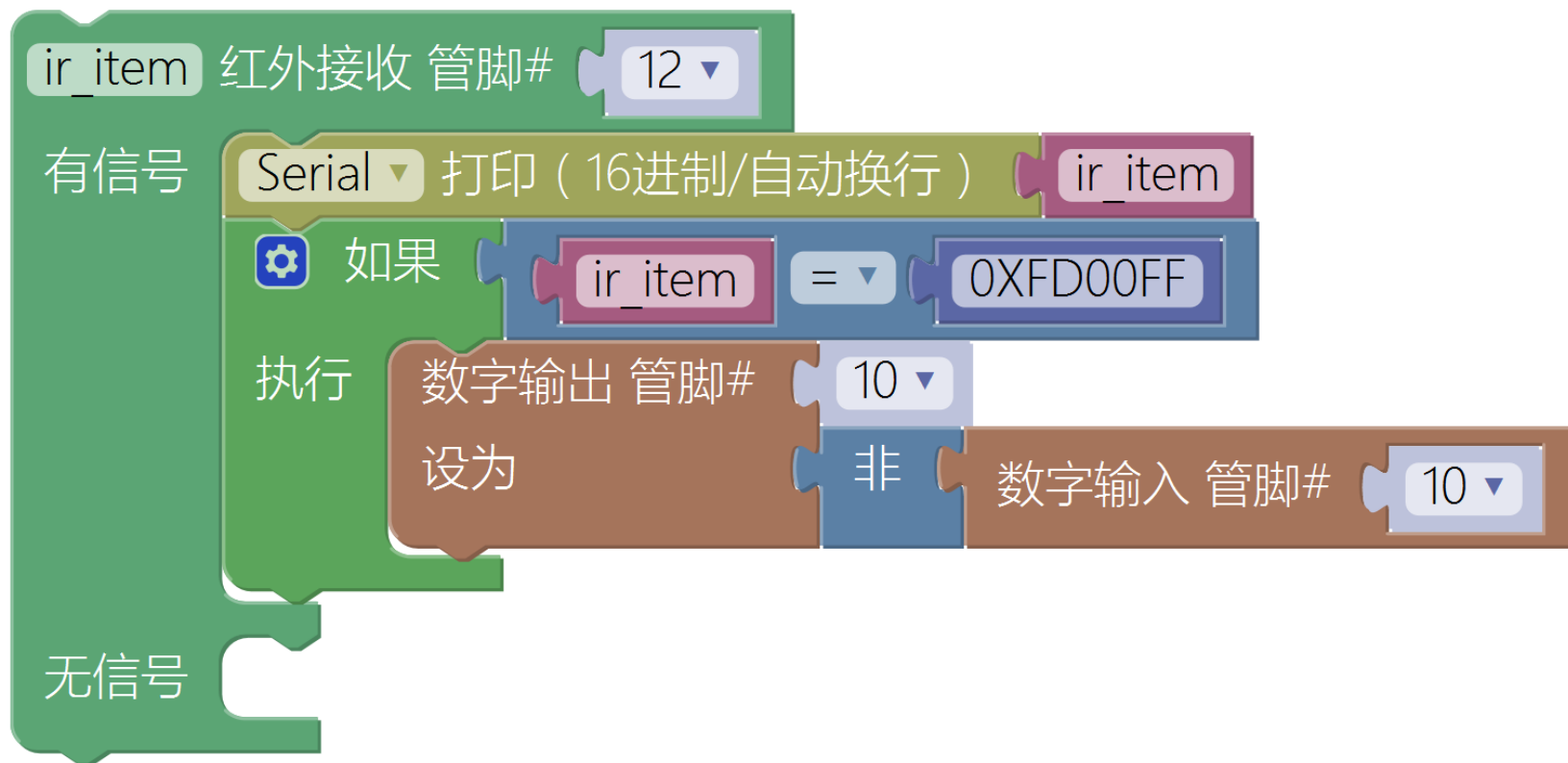
• 用一个符号表示位模式的4位

– 例如，一个16位串只需要_____个符号就可以表示

位模式	十六进制表示	位模式	十六进制表示
0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

软件编写

- 使用“开关”键切换LED灯的亮灭





第7课

Lesson 7

音乐盒

任务发布

- 使用一个开关和一个蜂鸣器制作简易门铃：
- 开关按下后，发出三次“叮咚”的提示。

硬件连接

- 取出一个蜂鸣器模块，用连接线将其与**8号管脚**对应的那组管脚相连（注意插线时颜色的对应）
- 取出一个开关模块，用连接线将其与**2号管脚**对应的那组管脚相连（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

蜂鸣器相关函数介绍

播放声音 管脚# 频率

–管脚指连接到蜂鸣器的数字管脚，频率是以Hz为单位的频率值。该频率的声音将一直持续，直到结束声音或输出一个不同频率的声音产生。通常与延时配合使用。

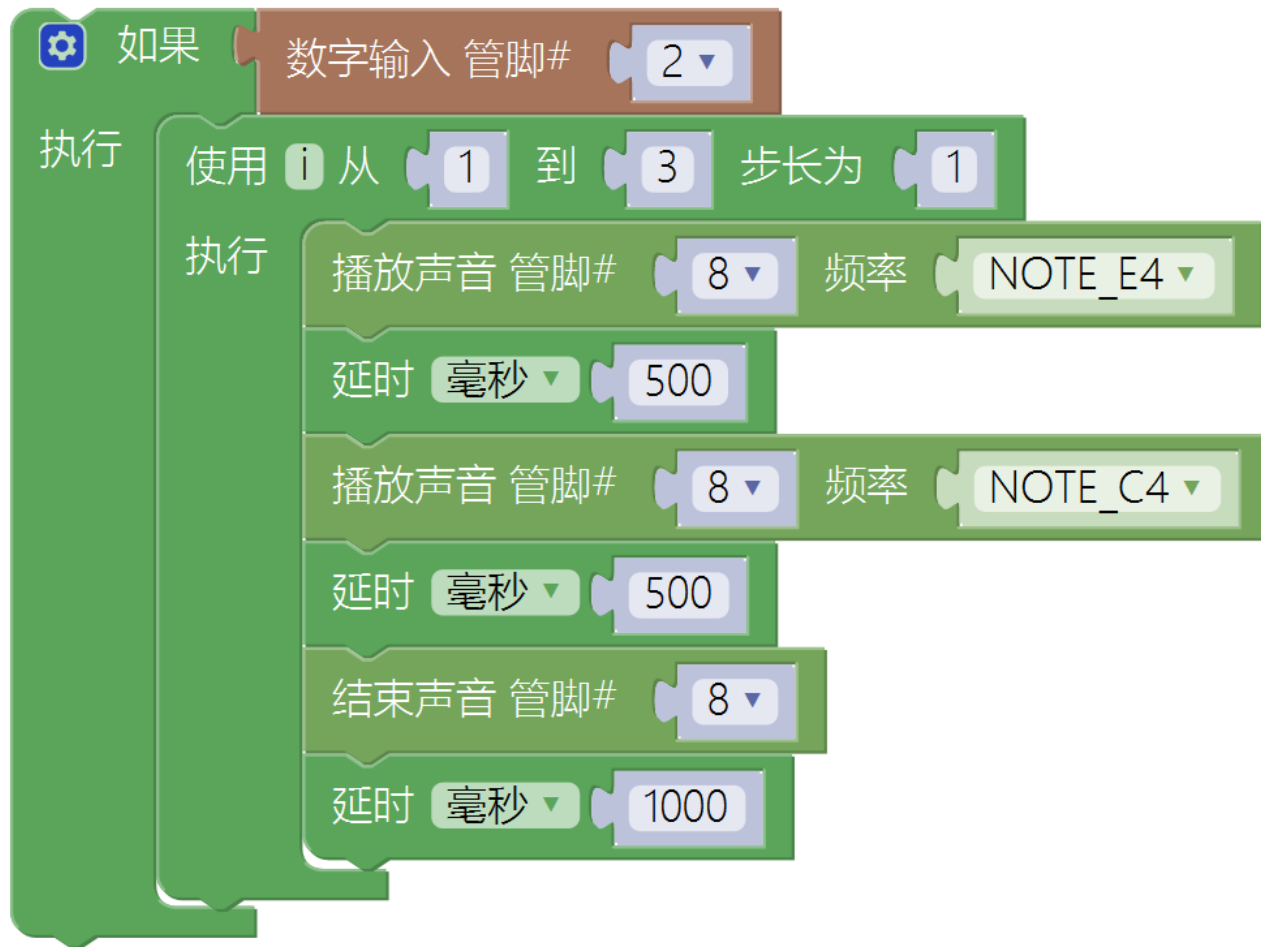
结束声音 管脚#

–结束指定管脚上产生的声音。

NOTE_C3
NOTE_D3
NOTE_E3
NOTE_F3
NOTE_G3
NOTE_A3
NOTE_B3
NOTE_C4
NOTE_D4
NOTE_E4
NOTE_F4
NOTE_G4
✓ NOTE_A4
NOTE_B4
NOTE_C5
NOTE_D5
NOTE_E5
NOTE_F5
NOTE_G5
NOTE_A5
NOTE_B5

字母	音名
C	Do
D	Re
E	Mi
F	Fa
G	Sol
A	La
B	Si

软件编写



任务发布

- 如何使用蜂鸣器播放一段旋律？

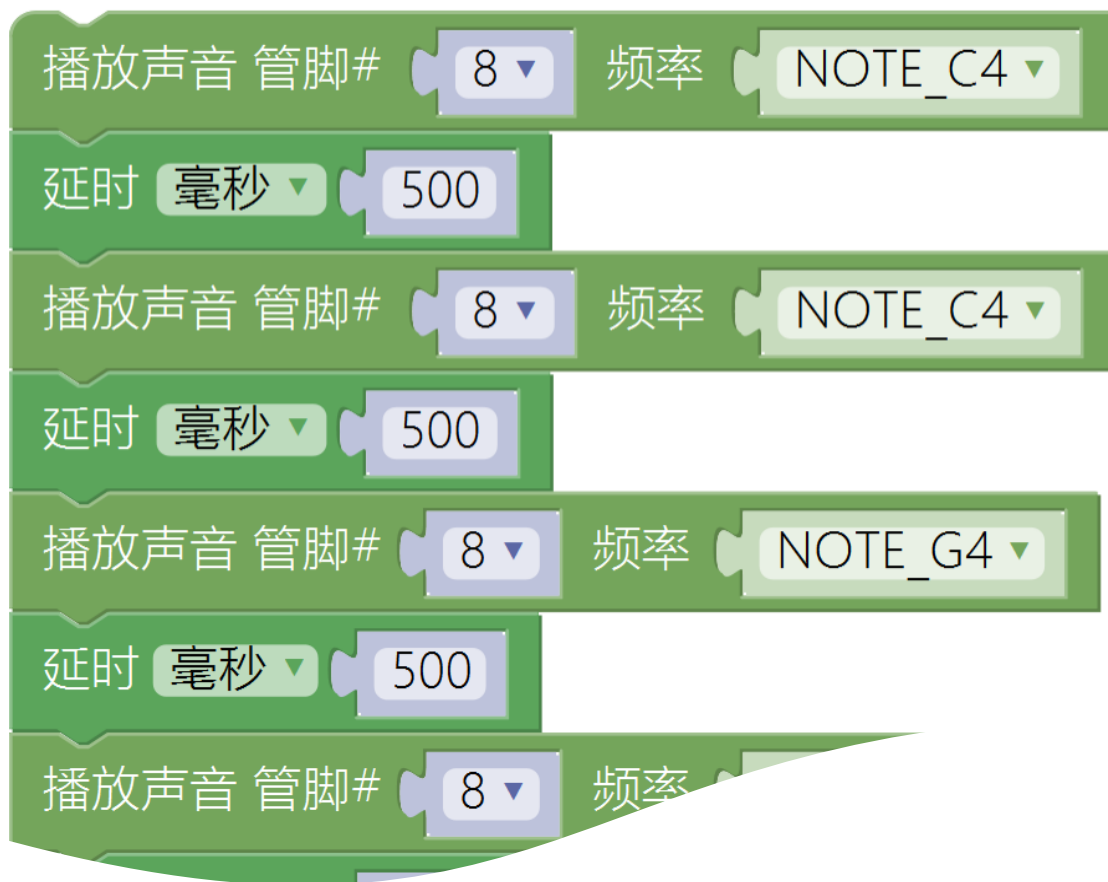
频率与音高的关系

- 标准音A的频率为440Hz
- 每个八度频率相差一倍，如： $f_{C6} = 2f_{C5}$
- 钢琴键上相邻两个音（如C~#C）的频率关系：

$$f_{n+1} = f_n \cdot 2^{\frac{1}{12}}$$



软件编写



有更好的实现方式吗？

构建数组

- 基准频率 (do、re、.....、si)
- 频率变化
 - 跨八度 (简谱中的“高音点儿”、“低音点儿”)
 - 临时变化音 (#、b号)
- 每个音的音名 (音高)
- 每个音的时值 (长短)

软件编写

初始化

```
小数 ▼ tonelist [ ] 从字符串 “ 1046.5,1174.7,1318.5,1396.9,1568,1760,1975.5 ” 创建数组  
整数 ▼ musiclist [ ] 从字符串 “ 1,2,3,1,1,2,3,1,3,4,5,3,4,5,5,6,5,4,3,1,5,6,5,4,... ” 创建数组  
整数 ▼ highlist [ ] 从字符串 “ 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,... ” 创建数组  
整数 ▼ updownlist [ ] 从字符串 “ 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,... ” 创建数组  
小数 ▼ rhythmlist [ ] 从字符串 “ 1,1,1,1,1,1,1,1,1,2,1,1,2,0.5,0.5,0.5,0.5,1,1,... ” 创建数组  
声明 speed 为 小数 ▼ 并赋值 120.0  
声明 updown 为 整数 ▼ 并赋值 0
```


软件编写

```
playmusic
  执行
    使用 i 从 1 到 32 步长为 1
      执行
        播放声音 管脚# 8 频率 tonelist 的第 m...
        延时 毫秒 1000 × 60 ÷ speed × r...
        结束声音 管脚# 8
        延时 毫秒 10
    结束
  结束

如果 数字输入 管脚# 2
  执行
    执行 playmusic
```

The image shows two Scratch code blocks. The first block is a purple 'playmusic' block with a settings icon. It contains a green '执行' (do) block with a loop '使用 i 从 1 到 32 步长为 1'. Inside the loop are four green blocks: '播放声音 管脚# 8 频率 tonelist 的第 m...', '延时 毫秒 1000 × 60 ÷ speed × r...', '结束声音 管脚# 8', and '延时 毫秒 10'. The second block is a brown '如果' (if) block with a settings icon, containing the condition '数字输入 管脚# 2' and a green '执行' block with '执行 playmusic'.



第8课

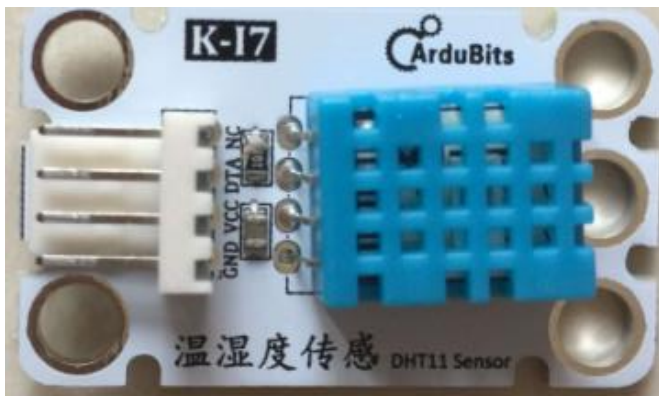
Lesson 8

温湿度检测仪

第8课 温湿度检测仪

硬件连接

- 取出一个温湿度传感器模块，用连接线将其与**9号管脚**对应的那组管脚相连（注意插线时颜色的对应）



温湿度传感器
能够获取周围环境的温度和空气湿度

第8课

温湿度检测仪

软件编写



想一想

· 如何使设备脱离计算机工作？

An abstract graphic on the left side of the slide. It features a large blue inverted triangle overlapping a larger orange inverted triangle. A red circle is positioned at the intersection of the two triangles. A large green circle overlaps the bottom right of the orange triangle. A cyan circle overlaps the bottom right of the green circle. A red line starts from the bottom left and points towards the orange triangle. A cyan line starts from the top and points towards the green circle.

第9课

Lesson 9

噪音计

硬件连接

- 取出舵机模块，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）
- 取出一个模拟声音传感器模块，用连接线将其与A2**号管脚**对应的那组管脚相连（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。
- 套件中舵机的最大旋转角度：180度

第9课 噪音计

软件编写



软件编写

The image shows a sequence of Scratch code blocks for a noise meter project. The blocks are as follows:

- Serial 打印 (自动换行)**: A green block with a dropdown menu set to "Serial" and the text "打印 (自动换行)".
- 模拟输入 管脚#**: A brown block with a dropdown menu set to "A2".
- 舵机 管脚#**: A green block with a dropdown menu set to "8".
- 角度 (0~180)**: A green block with a dropdown menu set to "角度 (0~180)".
- 映射**: A blue block with a dropdown menu set to "映射".
- 模拟输入 管脚#**: A brown block with a dropdown menu set to "A2".
- 从 [0 , 1024] 到 [0 , 180]**: A blue block with two input fields: the first contains "0" and "1024", and the second contains "0" and "180".
- 延时(毫秒)**: A green block with an input field containing "100".

扩展任务

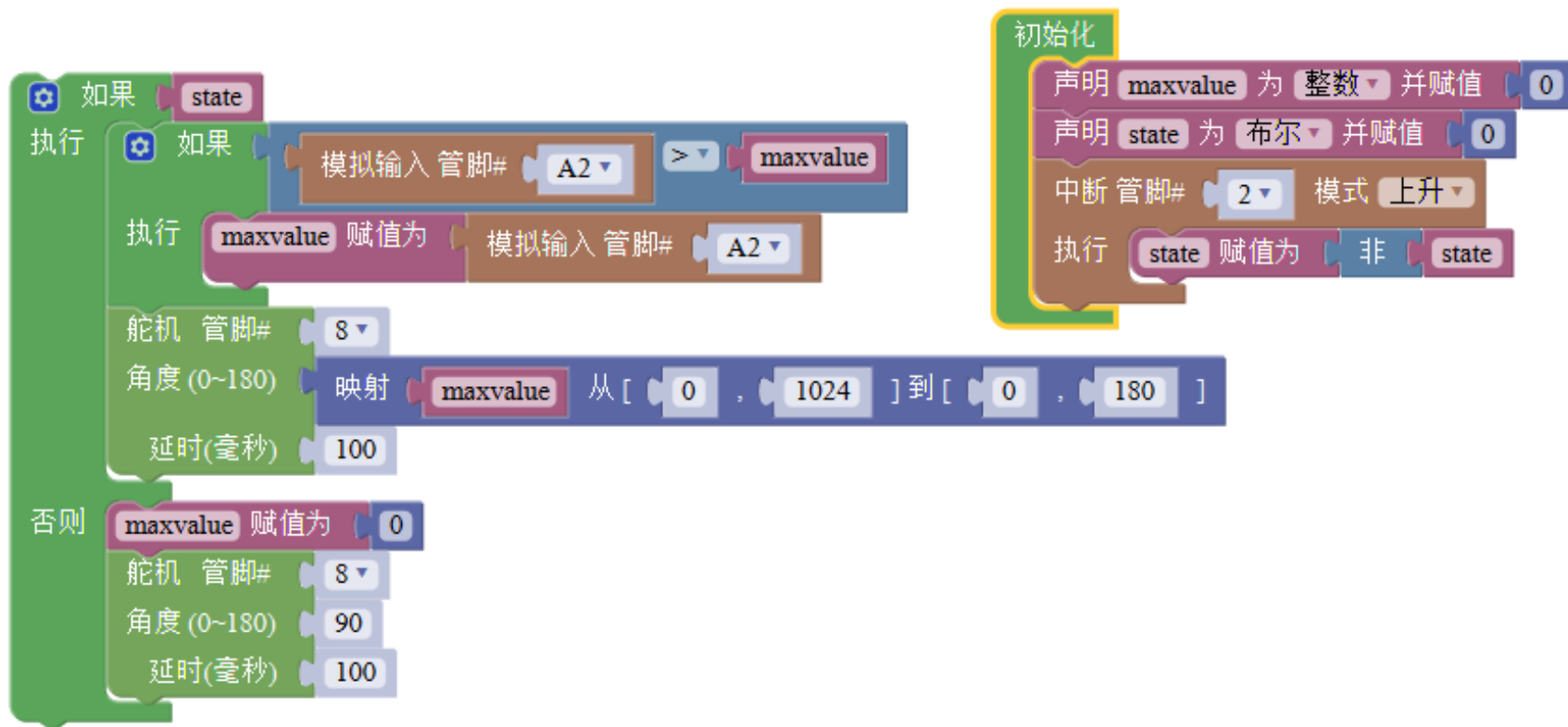
任务要求：

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。

扩展任务

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。





第10课

Lesson 16

超声波测距仪

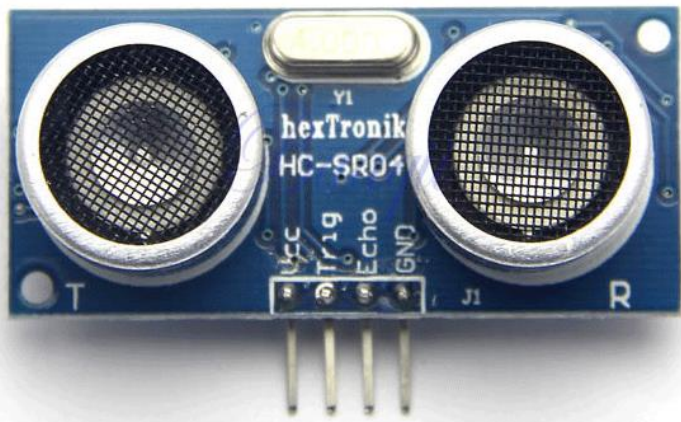
硬件连接

- 取出一个超声波测距模块，用连接线将其与**4号、5号管脚**对应的那组管脚相连（注意插线时颜色的对应）

知识讲解

• 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 $S = 340 \times \frac{t_2 - t_1}{2}$ 。



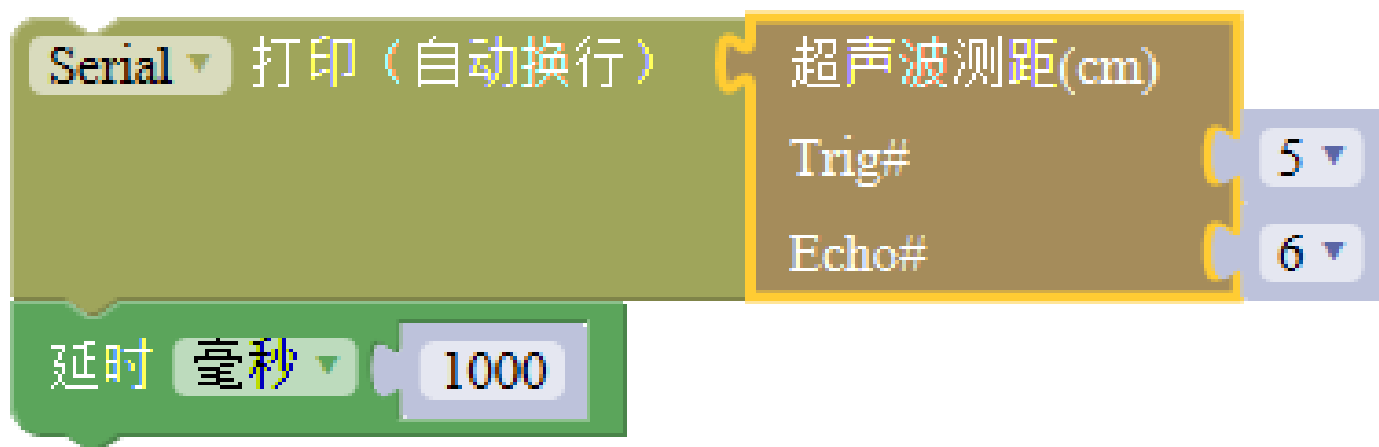
针脚	接线
VCC	5V
Trig	数字口#4 (发射端)
Echo	数字口#5 (接收端)
GND	GND

任务发布

- 倒车雷达的工作参数：

距离	反馈
>100cm	/
60~100cm	缓和的“滴滴”声
40~60cm	急促的“滴滴”声
<40cm	滴声长鸣

软件编写



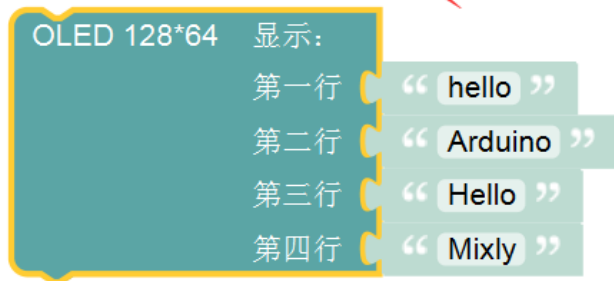
LCD液晶屏的使用

• 硬件连接



引脚	接线
VCC	5V
GND	GND
SDA	A4
SCL	A5

• 测试



软件编写



The image shows a Scratch script for an OLED 128*64 display. The script is contained within a teal block labeled "OLED 128*64 显示:". It consists of four lines of code:

- 第一行: "distance:"
- 第二行: 超声波测距(cm) Trig# 5 Echo# 6. This block is highlighted with a yellow border.
- 第三行: "Hello"
- 第四行: "Mixly"



综合任务1

Task 1

制作公园人数计数器

任务1 制作公园人数计数器

任务发布

使用一个液晶屏模块和一个红外接近开关模块，完成公园进入人数的计数器



两人一组，使用一个液晶屏模块和二个红外接近开关模块，完成公园当前实际人数的计数器

软件编写





综合任务2

Task 2

智能交通信号灯

任务2 智能交通信号灯

任务发布

两组红绿灯

- 完成对一组汽车灯，一组行人灯的控制

可控交通灯

- 使用开关控制，只有有人按下按钮时行人灯才变绿

无障碍交通灯

- 使用蜂鸣器提示行人红灯停下（缓慢）/绿灯通行（急促）



综合任务3

Task 3

庖丁解“车”

任务2 庖丁解“车”

任务发布

双闪灯

- 开关控制双闪灯，按下按钮双闪工作，再次按下停止

智能车钥匙

- 设置密码，遥控输入正确方可打开车锁（使用灯代替）

倒车雷达

- 不同距离发出不同速度的提示音



谢谢聆听

北师大教育学部创客教育实验室