高中通用技术课程

Arduino 电子控制技术教程（图形版）

学习　探索　设计　实践

北京师范大学第二附属中学

第 1 课 Arduino 入门

1.1 Arduino 套件简介

1 Arduino 介绍

Arduino 是一块简单易学的电子创意设计平台，它让你可以快速做出有趣的东西。Arduino 可以配合一些电子元件使用，例如 LED 灯、蜂鸣器、按键、光敏电阻等等。Arduino 配套软件基于“开放”原则，可以让您免费下载使用，开发出更多令人惊奇的互动作品。

2 设备参数及组成

官方说明：

采用 Atmel 微处理控制器（8 位 16MHz AVR 微处理器，32KB 程序存储空间，1KB 数据闪存，2KB RAM）。

大小尺寸：宽 70mm X 高 54mm。

输入电压：

（1）USB 供电。左侧为 USB 接口和电源接口。我们在使用时，将 USB 线分别接在 Arduino 和电脑 USB 口上。供电电压 5V。

（2） 外部 7V~12V 直流电压输入。

输出电压：5V 直流电压输出和 3.3V 直流电压输出

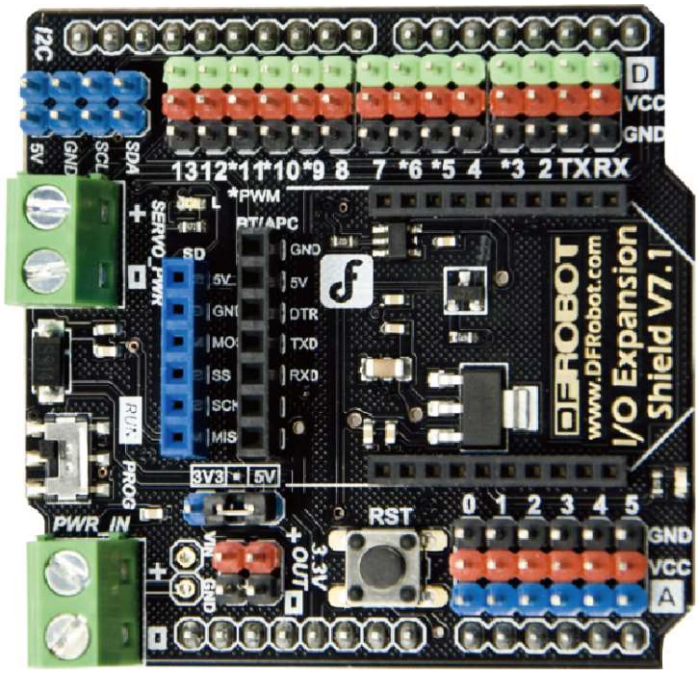
0-13 数字输入/输出接口

红正（5V，VCC），黑负（GND，0V，地）

A 蓝色线或口为模拟输入口，一般接传感器

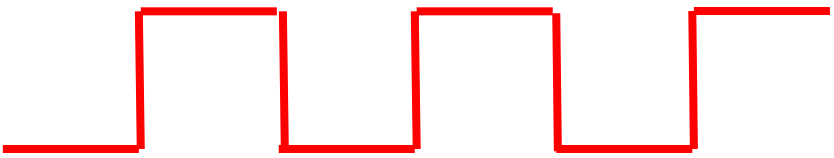
D 绿色线或口为数字输入输出口，一般接被控对象，也可接开关型传感器

严禁接反!!!



A0-A5 模拟输入接口

数字输入/输出端共 0~13，简称 D0-D13。注意 0 和 1（RX TX）用于和电脑通信，平时不要占用。在 Arduino 上，Digital（数字）端口常用于“输出”。它只能输出数字信号，即 0V 或 5V。因此我们可以把它看作是一个特殊的电源，能够根据程序输出 0V 或 5V 电压。

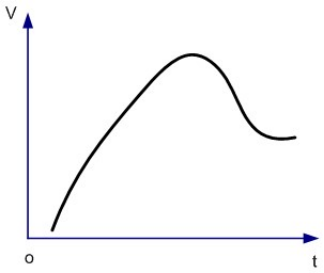


数字信号

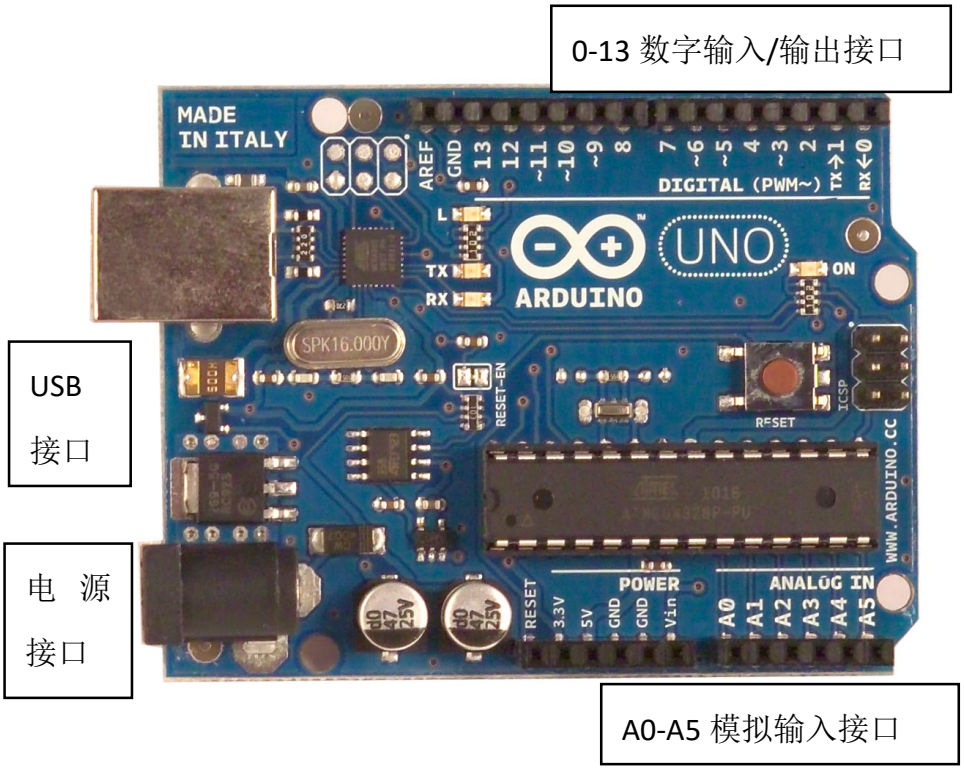
3，5，6，9，10，11 口可输出 8bit (0-255)PWM 信号（通过调节方波高低电平时间比例来调节电压有效值）。

模拟输入端共 0~5，简称 A0-A5。可以检测 0-5V 信号，10bit 量化（0-1023）

Analog In 用于“输入”，也就是检测模拟信号。这里我们可以把它看作是一个量程为 5V 的电压表。读取的电压值被存储进计算机的变量中（0-5V 对应 0-1023）供程序使用。



模拟信号



小知识

“数字”信号只有 2 种状态：高电平（电源正极电压，这里为 5V）低电平（电源负极电压，这里为 0V）。而“模拟”信号是“连续”的，在 Arduino 上可以是 0-5V 中任何数值。

要点总结

Arduino 中，红色管脚为 VCC，电压始终是 5V。黑色管脚为 GND，电压始终是 0V。绿色管脚为 D（数字口），电压可用程序控制，可能是 0V 或 5V 两种状态。蓝色管脚用来读取 0-5V 的电压。

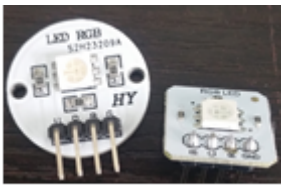
1.2 第一个程序：闪烁 led 灯

任务目标

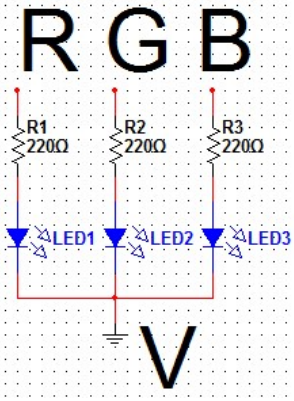
让 LED 灯每秒闪烁 1 次。

元件清单

器材	数量
Arduino UNO+传感器扩展板	1
共阴全彩 LED 模块	1
母母头（两头孔）杜邦线	4



全彩 LED 模块

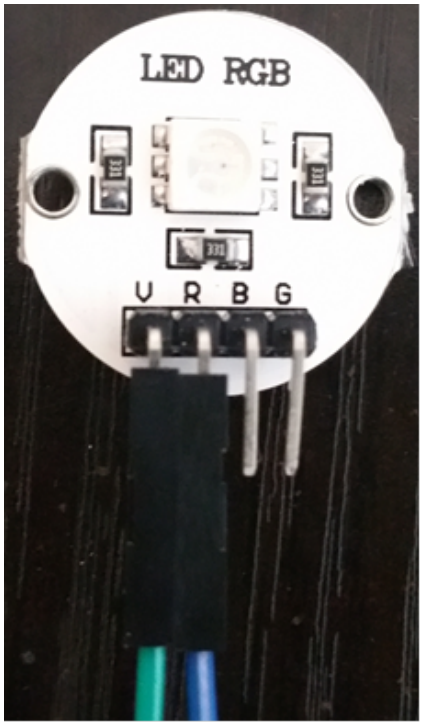


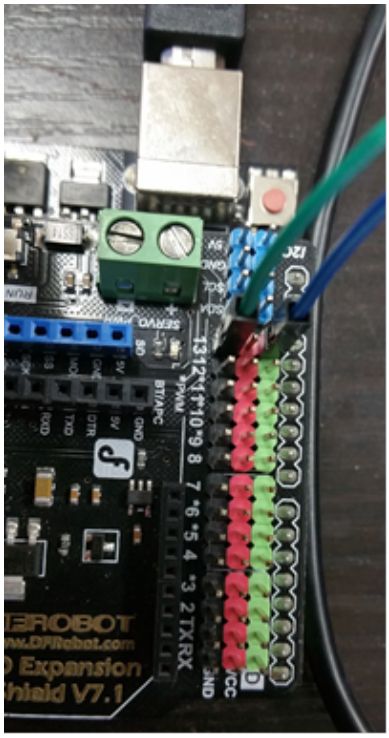
共阴全彩 LED 原理图

电路连接

用两头孔的导线连接全彩 LED 和 Arduino。连线方式如下:

元件	元件端口	Arduino 端口
全彩 LED	V（部分元件上没有 V,用 GND）	GND（黑色）
	R	D13（绿色）





完成后将 Arduino 连到电脑上。

从原理图可知，三个 LED 负极是接在一起的（V 脚），该点接电源负（0V，GND）。R，G，B 脚的电压决定了对应 LED 是否点亮。本项目中只用 1 个灯，我们可以任选一个管脚进行操作。例如用红色 LED 就控制 R 连接的 Digital I/O 第 13 口。

软件程序

找到桌面上 Mixly 图标，双击打开。利用拖拽完成下面的程序。

Mixly编程过程.avi

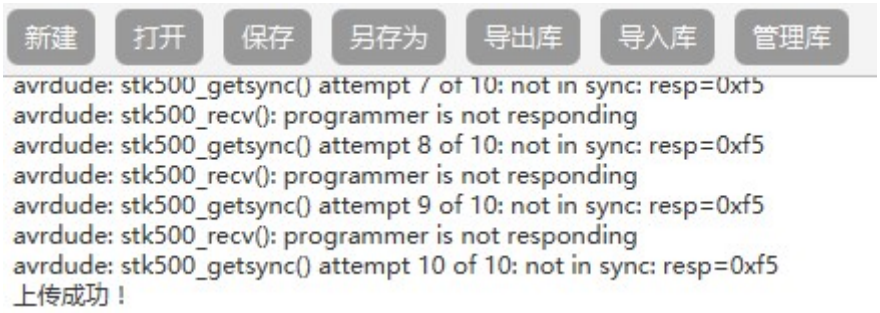
双击图标查看演示视频->

完成后一定记得上传！

错误排查

注意：我们一般不使用 Digital 0，1 口，否则会无法写入程序。

下载程序失败。这样的错误通常提示 stk500



这样的错误可能是由于 COM 口或板子型号出了问题。重新检查设置是否符合下图。

如果还不行，拔掉 Arduino 并关闭软件，再插上 Arduino，打开编程软件即可解决。

特别提示：Mixly 程序保存后再打开，需要在 Mixly 软件内打开，不能双击打开。

特别提示：Arduino 连在电脑上，底层的电源指示灯不亮，意味着可能发生了短路，应立刻从电脑上拔下来检查。

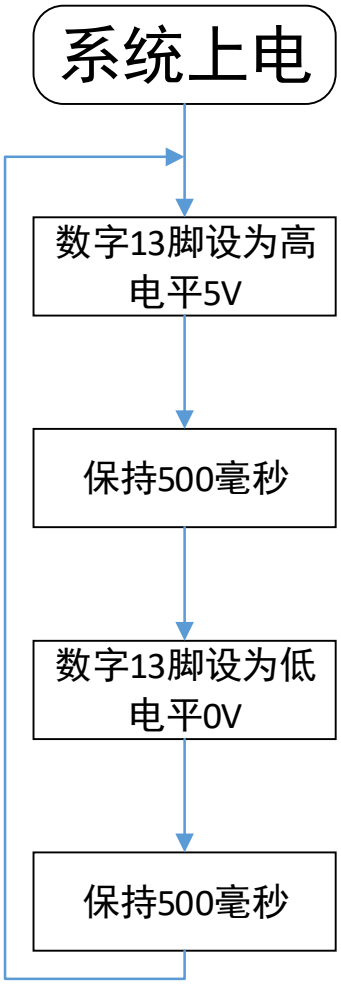
实验结果

LED 灯亮 0.5 秒，灭 0.5 秒不断闪烁。

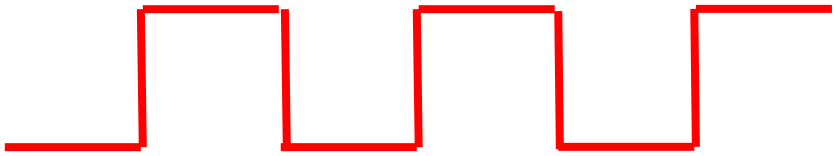
原理分析

（1）程序部分

“主程序”方块里的内容，编写一个周期即可循环执行。



本程序先将数字 13 口定义为高电平（5V）；接着延时 500ms； 再将数字 13 口定义为低电平（0V）；接着再延时 500ms。Arduino 程序默认会不断的循环执行，使 13 口电压形成一个 5V 方波。

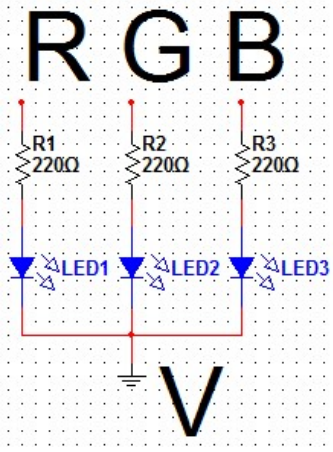


Digital 口可将 bool 型常量、变量转化为 0V 或 5V 的电压值。

（2）电路部分

当 led 灯的正极通过限流电阻与板子上的数字 I/O 口相连，数字口输出高电平 5V 时，led 导通，发光二极管发出亮光，持续 0.5s,；数字口输出低电平 0V 时，led 截止，发光二极管熄灭，持续 0.5s。如下图：

VCC=5V（电源正极），GND=0V（电源负极）



如果想让 led 快速闪烁，可以将延时时间设置的小一些，如果想让 led 慢一点闪烁，可以将延时时间设置的大一些。

自主设计 1　三色循环灯

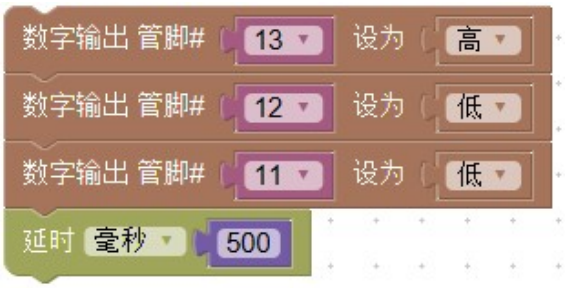
循环点亮红绿蓝三色灯。

提示：

增加连接 2 路 LED，同时增加控制命令。

例如：将 G（绿灯）连接到 D12 脚，B（蓝灯）连接到 D11 脚，

红灯（13 脚）亮，绿灯（12 脚）蓝灯（11 脚）灭，持续 0.5 秒的程序如下所示：



请把剩余部分补全。

自主设计 2：交通灯控制系统

绿灯亮->绿灯闪->黄灯亮->红灯亮，循环执行。

提示：红绿同时亮可以合成黄色。

要点总结

绿色管脚为 D（数字口），电压可用程序控制，可能是 0V 或 5V 两种状态。我们可以利用绿色管脚作为 LED 的电源正极，用黑色管脚 GND 作为 LED 的电源负极，通过控制 D 口电压来控制 LED 的亮灭。

控制多个被控对象，应使用多个数字（D）口。

1.3 夜半鬼火：如何调节 LED 亮度

要求：使 LED 从灭到亮，从亮到灭渐变。

元件清单

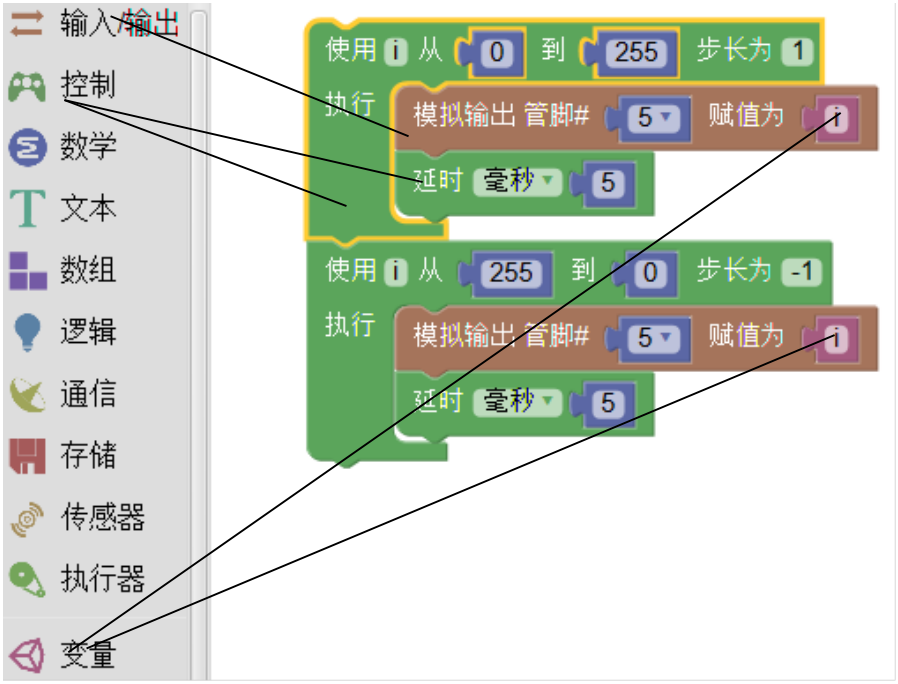
器材	数量
Arduino UNO+传感器扩展板	1
共阴全彩 LED 模块	1
母母头（两头孔）杜邦线	4

电路连接

用两头孔的导线连接全彩 LED 和 Arduino。

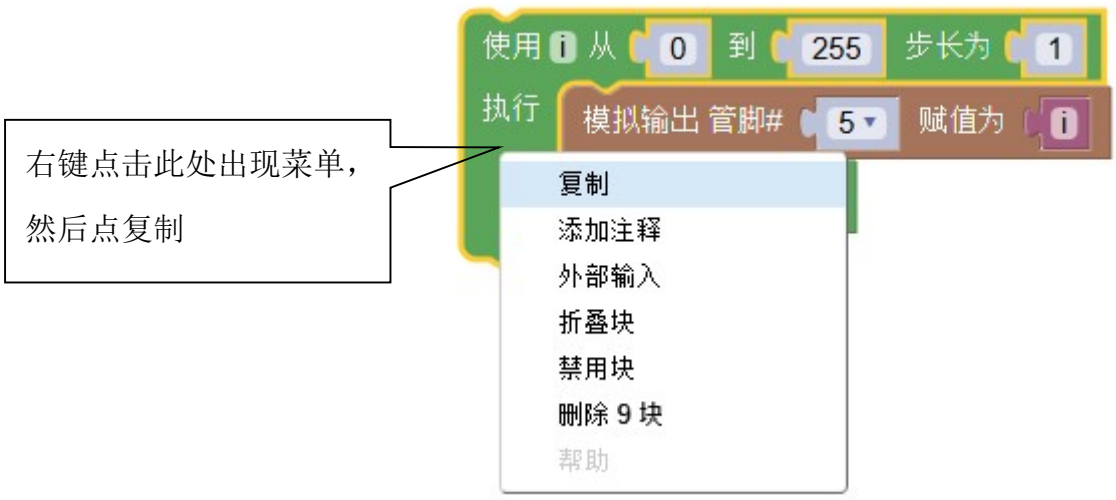
元件	元件端口	Arduino 端口
全彩 LED	V（部分元件上没有 V,用 GND）	GND（黑色）
	R	D5（绿色）

软件程序



编写完成后记得上传。上传出错或提示 stk500，按照前面一节所说的方法解决。

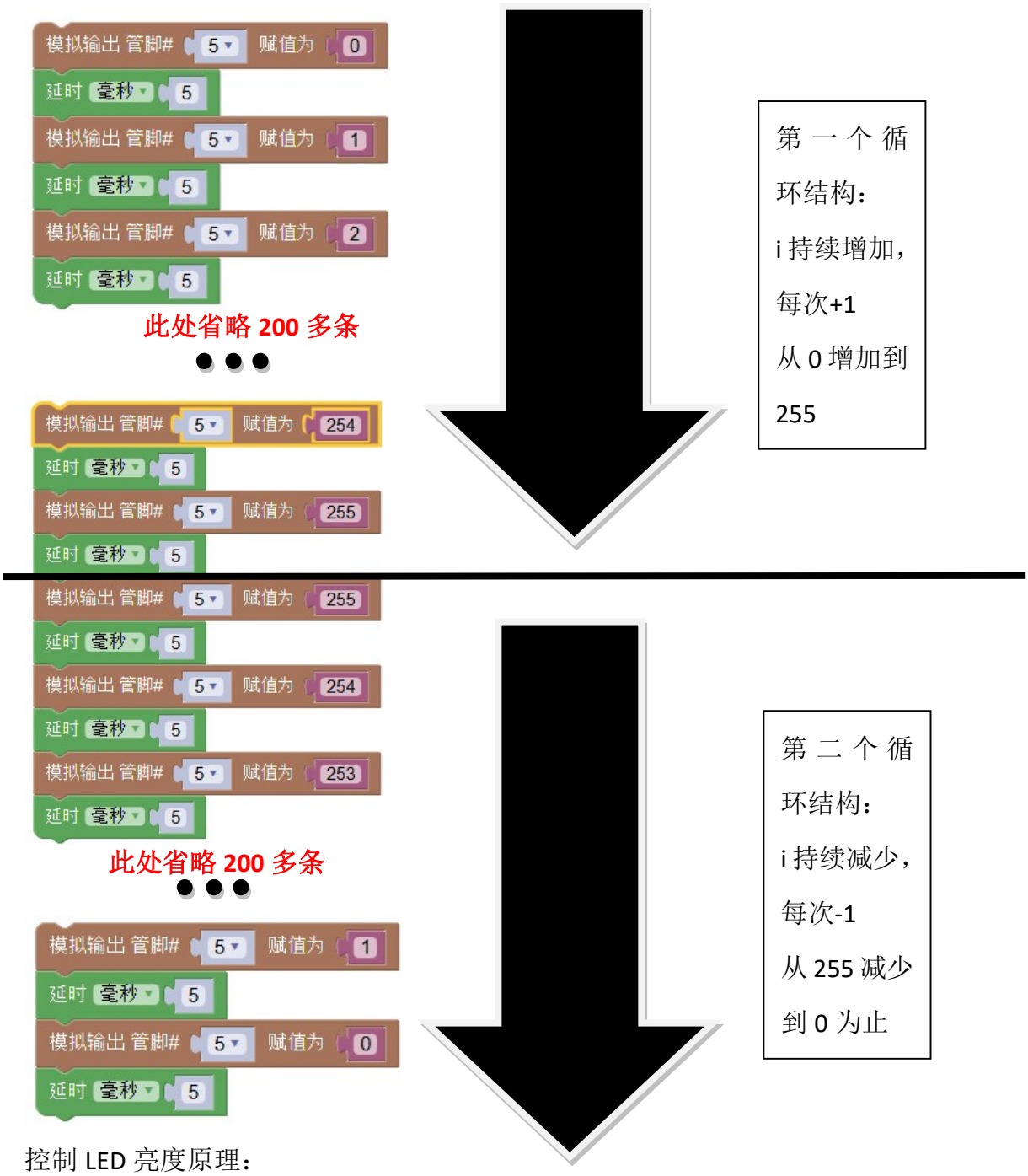
提示：如何复制粘贴(>▽<)：



原理分析

程序由 2 个循环结构组成。首先使 D5 输出从 0（不亮）逐渐增加到 255（最亮），步长为 1（就是每次循环加 1），然后从 255（最亮）逐渐减小到 0（不亮）。程序本身周而复始的运行，LED 就不断的从暗到亮再变暗循环执行。

循环结构的效果等价于以下程序：



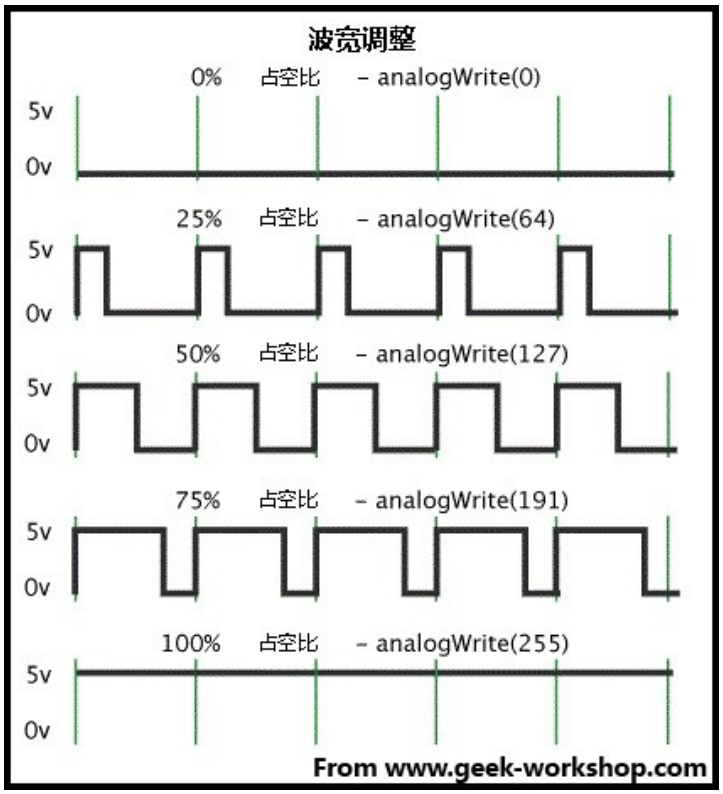
控制 LED 亮度原理：

PWM（Pulse-width modulation，脉宽调制）是使用数字手段来控制模拟输出的一种手段。数字端口的输入电压只有两种：0V 或 5V。使用数字端口输出占空比（5V 持续时间和总时间的比例）不同的方波，可以改变输出信号的有效值，从而改变负载（LED 或电机）的功率。

在我们使用的 UNO 板上，只有 3,5,6,9,10,11 端口能用 PWM 方式调节亮度。

你问老夫怎么知道的？o(￣ε￣*)自己百度：Arduino PWM 或参考单片机芯片 ATmega328P 的 datasheet n(*≧▽≦*)n）

模拟输出可设为 0-255，如下图。



Arduino 中 PWM 频率为 500Hz，超出人眼识别的能力，看起来就是连续变化，实现了模拟输出。实际上看久了会闪瞎

上述程序中，模拟输出的值从 0 上升到 255 再下降到 0，LED 从暗到亮再到暗，模拟鬼火效果。

自主设计

（能抄算什么本事。会改才是王道。）如何实现：LED 初始为红色，红黄绿青蓝紫循环连续变色。

全彩 LED 由 RGB（红绿蓝）三种 LED 组成。用红绿蓝三种光可合成不同颜色的光，如：R=255,G=0,B=0 为红光，R=255,G=255,B=0 为黄光，依次类推。

算法：

	R	G	B	如何改变
初始值：红	255	0	0	初始值
黄	255	255	0	红->黄：绿+
绿	0	255	0	黄->绿：红-
青	0	255	255	绿->青：蓝+
蓝	0	0	255	青->蓝：绿-
紫	255	0	255	蓝->紫：红+
红	255	0	0	紫->红： 蓝-

示例：假设 R 接 D3，G 接 D5，B 接 D6，从红到黄到绿的程序如下：（绿+，红-）

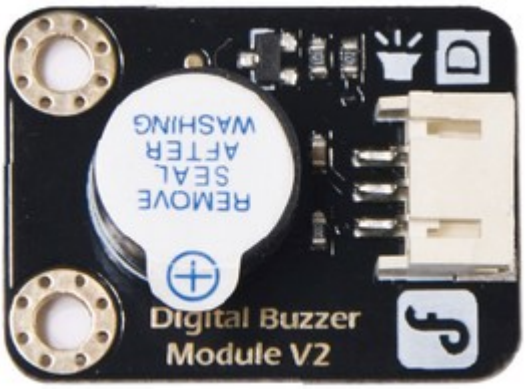
请自行补全剩余程序。

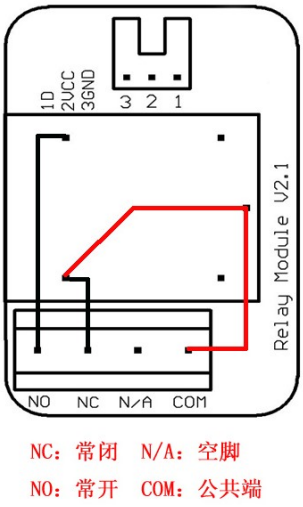
1.5 拓展：继电器与蜂鸣器

将 LED 替换成其他开关型元件（如继电器，蜂鸣器等），就可以实现更多的功能。

蜂鸣器发声原理是电流通过电磁线圈，使电磁线圈产生磁场来驱动振动膜发声的，因

此需要一定的电流才能驱动它，本实验用的蜂鸣器内部带有驱动电路，所以可以直接使用。当与蜂鸣器连接的引脚为高电平时，内部驱动电路导通，蜂鸣器发出声音；当与蜂鸣器连接的引脚为低电平，内部驱动电路截止，蜂鸣器不发出声音。





红正（5V，VCC），黑负（GND，0V，地）

A 蓝色线或口为模拟输入口，一般接传感器

D 绿色线或口为数字输入输出口，一般接被控对象，也可接开关型传感器

严禁接反！！！

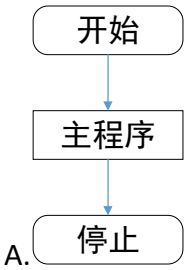
继电器可用来间接控制大功率的电器，我们可将它看作是一个由 arduino 控制的开关。连线时，将 VCC，GND 分别和 arduino 上的 5V，GND 相连；IN 和 arduino 的 Digital 输出端口相连，就可以参照上一节的办法控制继电器的开关了。将大功率负载接在接线柱上，就实现了 arduino 对此负载的控制。

练习

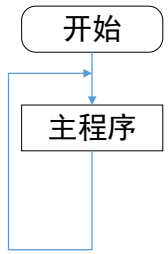
1. Arduino 输出的仅由 0V 和 5V 组成的信号，属于（数字/模拟）信号，管脚颜色为（　　）色。“高”电平指的是（0V/5V）。
2. Arduino 中 VCC 电压为（　）V，GND 电压为（　　）V. VCC 通常用（　）色表示，GND 通常由（　）色表示。VCC 和 GND 直接连接的后果是（　　），现象是

（　　）。

3. Arduino 的工作过程应该是（　　）。上传程序出错应该尝试（　　）。



A.



B.

4. Arduino 模拟输出的信号属于（数字/模拟）信号。数字输出、模拟输出用来（读取信息/控制对象）。输出值的范围是（　　）。

5. 脑补下面两段程序的运行效果。电路如右图所示

第 2 课 模拟传感器和开闭环控制系统

前面我们学习的都是“输出”某个信号，或者说“控制“某个元件。如果制作真正的自动控制系统，还需要让 Arduino 感知外界信息。Arduino 的 Analog In 可以看作是电压表，用来读取 0-5V 的电压值，结果存在内存里供 CPU 处理。如果要感知其他物理量（如照度，加速度，声音等），还需要用到传感器。传感器可以将其他物理量转化为电压信号，供 Arduino 分析处理。

2.1 用传感器感知周围环境

任务目标

灯具的一个重要指标是频闪。虽然灯具的有无频闪在我们看来好像并没有什么区别，用肉眼也无法直观地感觉灯具有无频闪，但频闪会加剧视觉疲劳，危害健康。我们可以用 Arduino 制作简单的科学仪器来观察照明灯具是否有频闪。

元件清单

器材	数量
Arduino UNO+传感器扩展板	1
光线传感器	1

光线传感器

红正（5V，VCC），黑负（GND，0V，地）

A 蓝色线或口为模拟输入口，一般接传感器

D 绿色线或口为数字输入输出口，一般接被控对象，也可接开关型传感器

严禁接反！！！

声音传感器

电路连接

在 analog In 0 口（简称 A0）上接光线传感器。颜色必须对应！

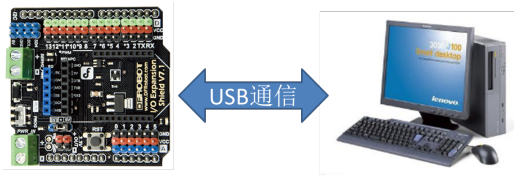
软件程序

0.964 版

0.996 版

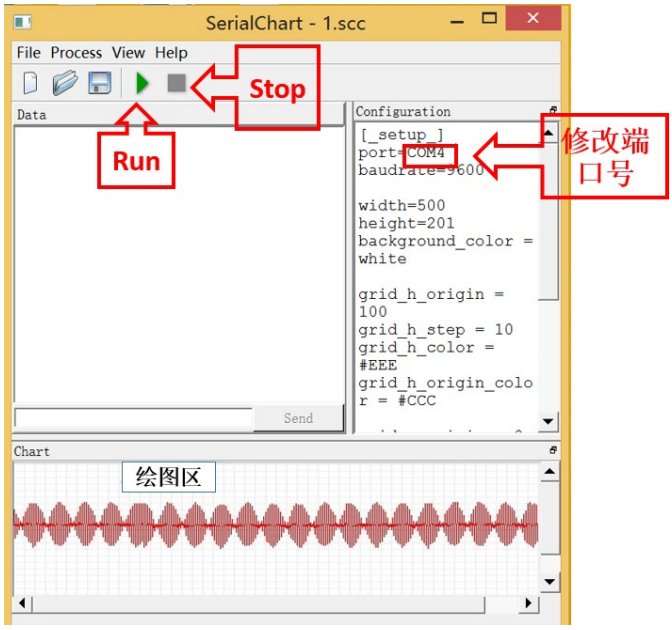
程序功能

重要提示：下载程序时，不能打开”串口监视器“或 SerialChart 软件。任何时间只能有单一程序/功能占用 USB 通信。这个程序可能是：Arduino 传数据、上传程序；串口监视器（它是一个独立程序）；SerialChart 软件。



写入程序，打开最右侧的按钮“串口监视器”，可以看到读数。Serial 打印的意思是：将数据发送到电脑上显示。

关闭串口监视器（Serial Monitor）。打开桌面上 SerialChart 软件，按照 Arduino 的 COM 口修改配置文件。



（图为学校原来用的日光灯管频闪波形）

然后点击 Run 按钮，用手触摸 A0 口，可以看到变化的波形。看看自然光和日光灯发出的光线有何不同。完成后记得关闭软件和串口监视器。

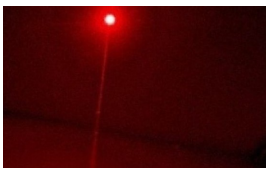
换上声音传感器，对着它吹口气，看看传感器输出的波形。（告诫：注意您的口水！）

小知识

光线传感器利用电阻和光敏管组成串联分压电路，照度改变时，光敏管两端电压也随之变化。声音传感器可将振动转化为微弱的电压值，然后通过放大器放大得到输出电压。Analog In 口可以将电压转化为可被程序处理的变量。

2.2 光控蜂鸣器

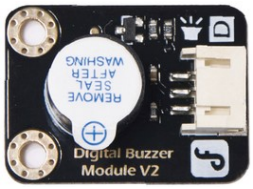
任务目标



有一种激光防御装置：光源发射一道激光照射在光线传感器上。如果激光被入侵者遮挡，就会报警。我们尝试搭建一个报警器模型：有光时蜂鸣器不响，没光时报警。

元件清单

器材	数量
Arduino UNO+传感器扩展板	1
蜂鸣器模块	1
光线传感器	1
共阴全彩 LED 模块	1
母母头杜邦线	2



蜂鸣器模块

光线传感器

注意区分蜂鸣器模块和声音传感器！

电路连接

元件	Arduino 端口
蜂鸣器模块	D3
光线传感器	A0

首先参考前一节程序。用手遮住光线传感器，在串口监视器中读取数值；在不遮光的条件下再读一次。据此找出区分“有光”“无光”状态的阈值。

参考程序

看颜色找模块。记得根据实际情况修改阈值和管脚号。

重要提示：分支结构模块使用方法

1.将该模块拖出

2.单击齿轮图标，出现下图

3.把“否则”拖拽到下图位置

4.再次单击齿轮图标

原理分析

前面已经学习了顺序结构和循环结构。这里采用的是分支结构。程序有 2 个分支:首先判断模拟输入是否大于阈值（是否有光）。如果条件满足，则执行“将 D3 设为低电平”这个分支，蜂鸣器不响;如果条件不满足，则执行“否则”分支中的内容(声音忽高忽低，模仿警笛)。2 个分支必然会执行其中的一个，执行完毕后运行整个分支结构下面的一句：延时 100 毫秒。然后程序又会回到第一句。

自主设计

1 设计一个吹蜡烛的小装置。要求：平时灯亮，吹气后，灯灭 2 秒然后恢复点亮。

提示：使用声音传感器和 LED 灯盘。颜色自选。

2 脉搏波可以反映很多生理信息，如心率，血压，血氧等。利用光控灯装置，自主设计脉搏波测量装置。提示：用大功率 LED 照亮手指，把指肚压在光传感器上，调节 SerialChart 中的某些参数，测量出脉搏波形。

2.3 光控灯

2.3.1 光控路灯模型

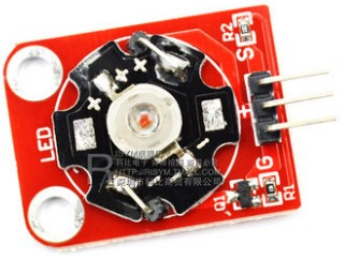
任务目标

太阳能光控路灯在白天光照强时不亮，晚上自动点亮。我们可以用 Arduino 模拟这个控制过程。

元件清单

器材	数量
Arduino UNO+传感器扩展板	1
光线传感器	1
大功率 LED 模块	1
母母头杜邦线	3

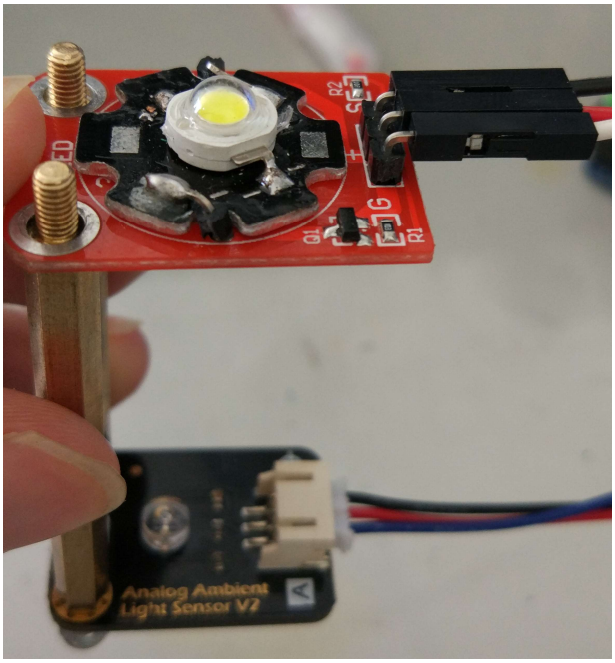
光线传感器

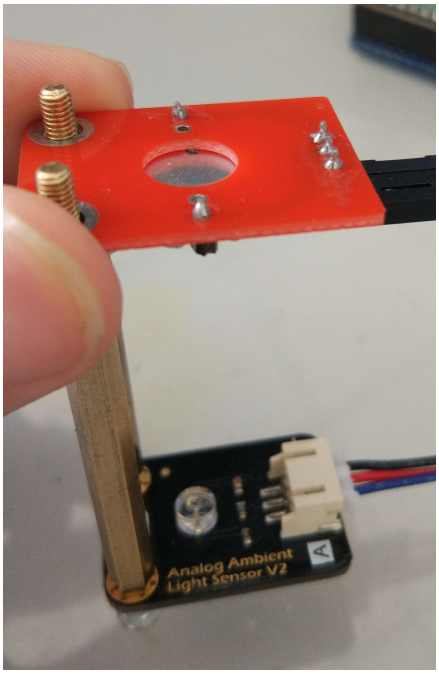


大功率 LED 模块

电路连接

元件	元件端口	Arduino 端口
大功率 LED	G	GND（黑色）
	S	D3（绿色）
	+	VCC（红色）
光线传感器		A0（颜色对应）





考虑应该采用上图中哪种方式组装？如图进行装配，然后用螺母固定 LED 模块。

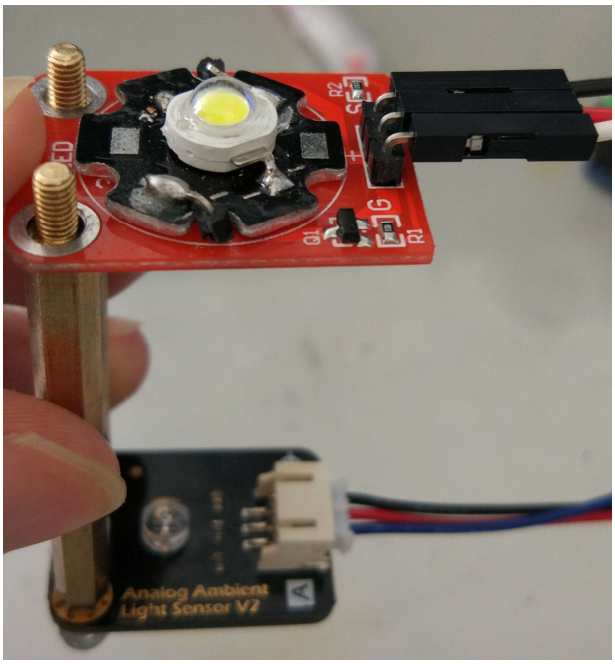
用手遮住光线传感器，在串口监视器中读取数值；在不遮光的条件下再读一次。据此找出区分“有光”“无光”状态的阈值。

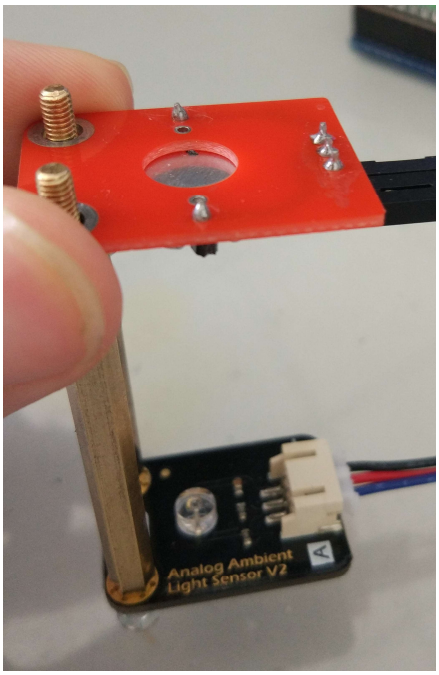
参考程序

看颜

2.3.2 闭环照明控制系统

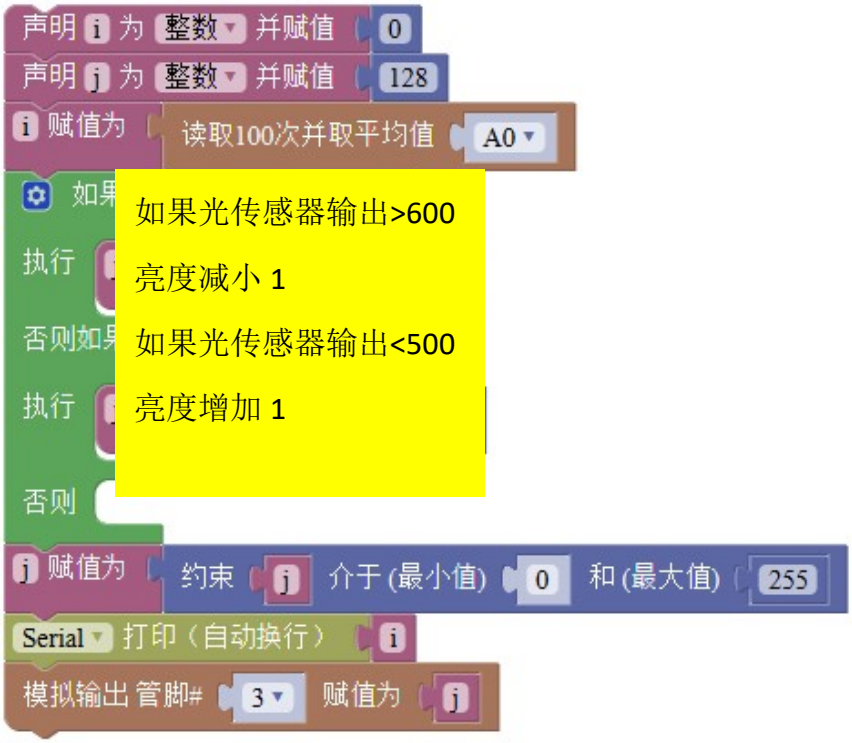
米家飞利浦智睿台灯宣传品上写：“内置高精度环境光传感器，时刻监测环境光的变化……台灯亮度可以随环境光的变化自适应调整，持续给您真正健康、舒适的光亮。”请用 Arduino DIY 一个可以使光照保持恒定的智能台灯原型。





重新考虑应该采用上图中哪种方式组装。连线同上一节。如图进行装配，然后用螺母固定 LED 模块。

参考程序 1：



第 3 课 Arduino 与数字式传感器

3.1 数字式传感器

前面我们学习了模拟输入及模拟式传感器。模拟式传感器可将非电物理量转化为连续变化的电压，由 Arduino 读取并处理。还有一类传感器可以把被测参量转换成离散的数字量输出的传感器。例如按钮只有按下/弹起两个状态；磁敏干簧管只有导通/断开两个状态；它们做成传感器模块后，输出电压只有 0V/5V 两个状态，它们都属于简单的数字式传感器。另外，红外遥控等数字化通信手段使用的信号都是数字量，红外接收头（将红外遥控信号转化为电压）也属于数字式传感器。

3.2 简单数字式传感器应用：收费抬杆

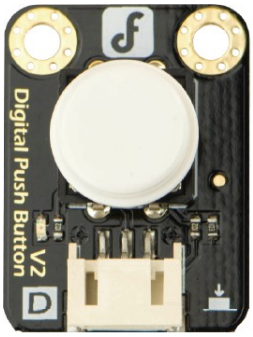
任务目标

制作简单模拟收费抬杆装置：按下按钮，抬杆直立；其他情况下抬杆水平。

元件清单

器材	数量
Arduino UNO+传感器扩展板	1
舵机	1
按钮模块	1
光线传感器	1
共阴全彩 LED 模块	1
母母头杜邦线	4

电路连接



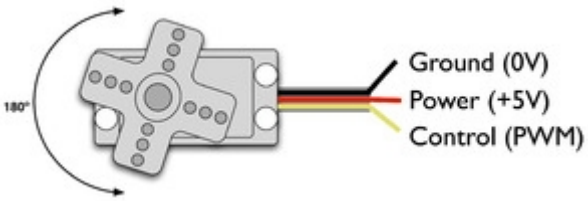
红正（5V，VCC），黑负（GND，0V，地）

A 蓝色线或口为模拟输入口，一般接传感器

D 绿色线或口为数字输入输出口，一般接被控对象，也可接开关型传感器

严禁接反！！！

按钮模块接 D4，按下时输出高电平，否则输出低电平。

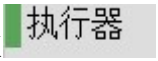


舵机有一个三线的接口。**黑色（或棕色）的线接 GND**，红线接+5V 电压（VCC），黄线（或是白色或橙色）接 D 口。将控制线连接到单片机任意数字（Digital I/O）端口（除 0，1 外）。本例中连接到第 9 脚上。

特别提示：VCC GND 切勿接反！！

元件	Arduino 端口
舵机	D9
按钮模块	D4
光线传感器	A0

软件程序

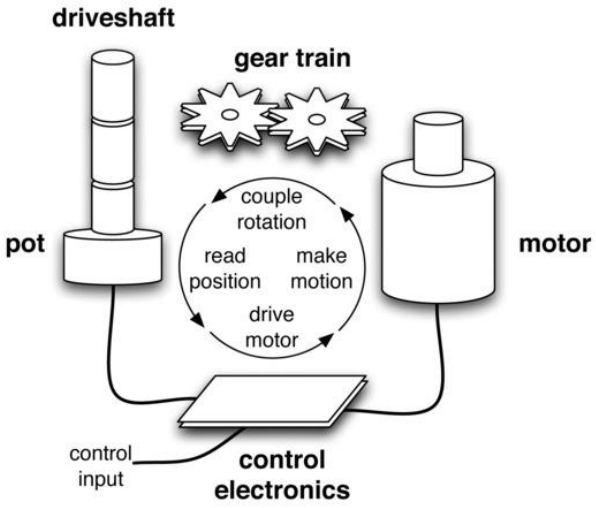
舵机模块在 

中。

原理分析

按钮模块**按下时输出高电平，否则输出低电平。这种只有 2 种状态的传感器，属于数字传感器，可以用数字（D）口读取它的状态。**

舵机常用在机器人技术，控制玩具汽车和飞机等。舵机的旋转不像普通电机那样只是古板的转圈圈，它可以根据指令旋转到 **0 至 180 度之间**的任意角度然后精准的停下来。舵机由直流电机、减速齿轮组、传感器和控制电路组成。舵机转动的位置一变，位置检测器的电阻值就会跟着变，通过控制电路读取该电阻值的大小，就能根据阻值调整电机的速度和方向，使电机向指定角度旋转。下图显示的是一个标准舵机的部件分解图和舵机闭环反馈控制的工作过程。



舵机模块有 3 个参数。**管脚表示舵机连接的数字口序号；角度表示舵机旋转到的位置。延时与控制中的延时作用相同。**

自主设计

制作模拟收费抬杆装置：有车辆挡住光传感器并且按下按钮，舵机的舵盘竖起并点亮绿灯；其他情况下不动并亮红灯。

收费抬杆.mp4

提示：可能用到逻辑运算“且”（AND，“与”）。同时满足“无光”“按下按钮”这两个条件才会执行舵机旋转的操作。**“且”模块在逻辑大类中。**如果只满足其中一个就执行，应该把“且”改为“或”。另有“非”模块，也在逻辑大类中。

LED 和光传感器的使用方法可以参考前面的教程。

3.3 包含编码的数字式传感器应用：红外遥控风扇

红外遥控系统由下列几部分组成：

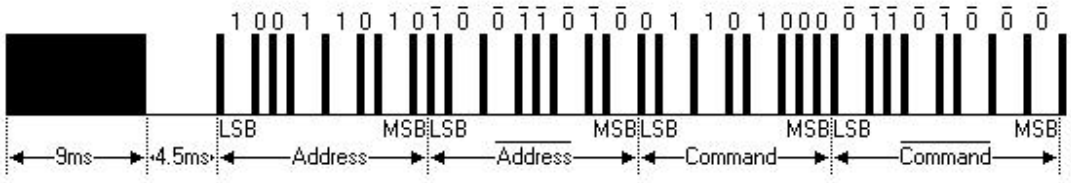
（1） 发射部分（红外遥控器）

发射电路采用红外发光二极管来发出经过调制的红外光波，一般由按键、编码调制电路、驱动和发射电路组成。当按下按键时，编码调制电路产生所需的指令编码信号，再由驱动电路进行功率放大后，由发射电路向外发射信号。

（2）接收部分（接收头和控制器）

红外接收电路通常被厂家集成在一个元件中，成为一体化红外接收头。 内部电路包括红外检测二极管，放大器，限幅器，带通滤波器，积分电路，比较器等。红外接收头将红外信号接收下来，并还原为指令代码，传送给控制器，由控制器指挥各种执行电路来实现控制。

接收得到的指令码通常用十六进制数表示。十六进制数前面要加上 0x，电脑才知道它是十六进制数而不是普通字符。



提示：在使用红外遥控时，请不要使用 3，11 脚做模拟输出。否则有可能出现问题。

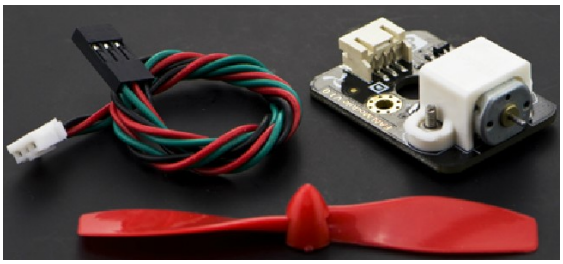
3.3.1 红外遥控调速风扇

任务目标

制作红外遥控风扇，按 1 停止转动，按 2 低速转动，按 3 高速转动。

元件清单

器材	数量
Arduino UNO+传感器扩展板	1
红外遥控器	1
红外接收头	1
风扇模块	1
共阴全彩 LED 模块	1
母母头杜邦线	4



红外遥控器　　红外接收头　　风扇模块

电路连接

元件	Arduino 端口
红外接收头	D8
风扇模块	D5

红正（5V，VCC），黑负（GND，0V，地）

A　蓝色线或口为模拟输入口，一般接传感器

D 绿色线或口为数字输入输出口，一般接被控对象，也可接开关型传感器

严禁接反！！！

颜色必须对应！

直流电机（direct current machine）是指能将直流电能转换成机械能的旋转电机因电流较大，用 Arduino 控制需借助晶体管或继电器做执行器来间接控制。风扇模块已经集成了这些元件，可直接连在数字口上用模拟输出（前面的鬼火控制亮度的方式）调速。

软件程序

首先从 通信 拖出以下内容

上传到 Arduino 中。上传完成后，打开串口监视器。按下遥控器上的按钮就能看到对应的控制代码。

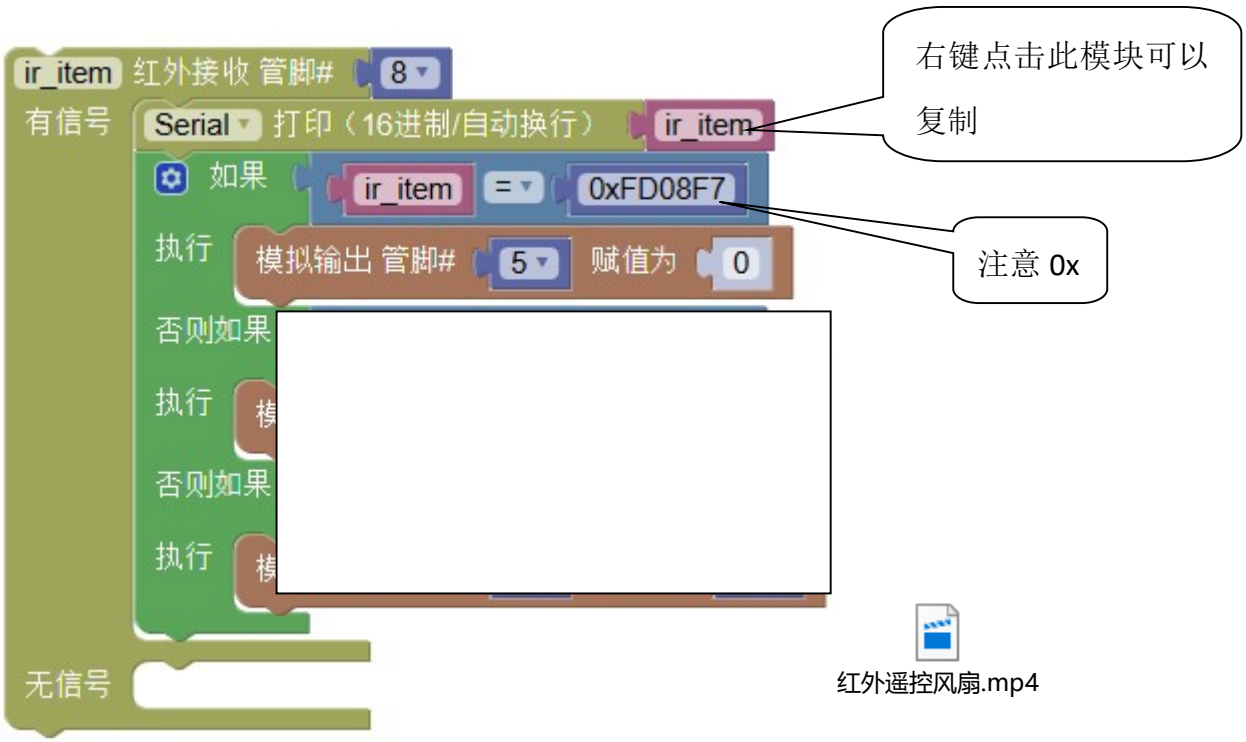
例如，按下 1 时，可能会有如下的反应：

这里面 NEC 代表红外遥控器控制协议的类型。FD08F7 是代表 1 键的控制代码。如果持续按住任何键，就会出现 FFFFFFFF，意思是重复按键。如果受到干扰或按下的方式不对，IR TYPE 会是 UNKNOWN，这代表无意义的乱码。

实验时用手捂住红外遥控器和接收头，可避免干扰

按下 2 键显示 FD8877……注意前面加上 0x 表示十六进制数。

探测出每个键的代码后，继续编写以下程序：



风扇用模拟输出（PWM）控制。输出 0 时不动，128 慢速旋转，255 以最快速度旋转。

自主设计：红外遥控智能家居

在前面遥控风扇基础上，增加遥控全彩 LED 灯的功能。要求按另外几个键可以控制灯的颜色。也可以考虑控制其他被控对象（如蜂鸣器）。

提示： 先完成遥控风扇，打开串口监视器，分别按下别的按钮，得到对应的按键代码，再编写控制部分。

记得上传程序时关闭串口监视器！

不要用 3，11 口做模拟输出。

避免在红外接收程序内部使用延时模块。这会导致程序暂时失去响应。

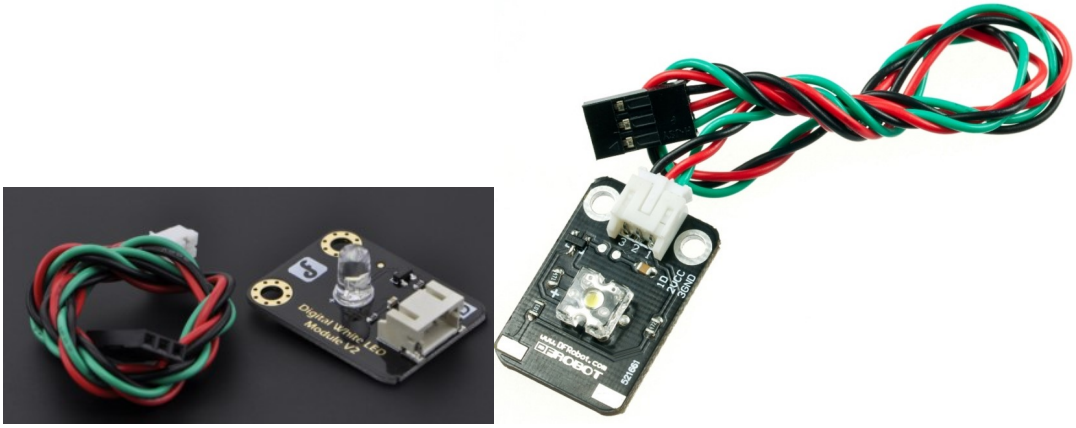
3.3.2 红外遥控与接收实验（选学内容）

任务目标

完成一个完整的红外通信转发系统。Arduino 1 号收到遥控器发送的指令后，闪一下 LED 灯，然后转发一个指令给 Arduino 2 号，让它也闪一下灯。

Arduino 1 号元件清单

器材	数量
Arduino UNO+传感器扩展板	1
红外遥控器	1
红外接收头	1
单色 LED 模块	1
红外发射头	1



单色 LED 模块

红正（5V，VCC），黑负（GND，0V，地）

A 蓝色线或口为模拟输入口，一般接传感器

D 绿色线或口为数字输入输出口，一般接被控对象，也可接开关型传感器

严禁接反！！！

红外发射头

Arduino 1 号电路连接

元件	Arduino 端口
红外遥控头	D7
LED 灯	D13
红外发射头	D3

Arduino 1 号程序代码

Arduino 2 号元件清单

器材	数量
Arduino UNO+传感器扩展板	1
红外接收头	1
单色 LED 模块	1

Arduino 2 号电路连接

元件	Arduino 端口
红外接收头	D7
单色 LED 模块	D5

Arduino 2 号程序代码

原理分析

完成上传程序后，将 Arduino 1 号的发射头和 2 号的接收头相对，用手挡住周围的光线。

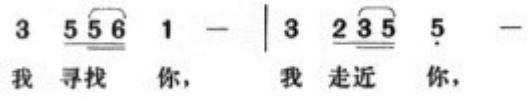
对 Arduino 1 的接收头按动红外遥控器，Arduino 1 的 LED 会闪烁一次，然后发送 ABCDEF 给 Arduino 2。Arduino 2 收到以后，1 也会让 LED 闪烁一次。

照此原理，可以做出更多互动效果，实现不同设备之间的自动化通信、协作。

3.4 用扬声器播放音乐

任务目标

让扬声器播放下面的音乐片段



元件清单

器材	数量
Arduino UNO+传感器扩展板	1
扬声器	1
公母头（一头针一头孔）杜邦线	2

扬声器

电路连接（注意这里不是按颜色接的！）

元件	元件端口	Arduino
扬声器	红线	D6
	黑线	GND

软件程序

1. 在 Mixly 下的 arduino-1.7.8\libraries 文件夹内新建 Mixlymusic 文件夹。
2. 在上述文件夹内新建文本文件，文件名为 Mixlymusic.h。内容如下：

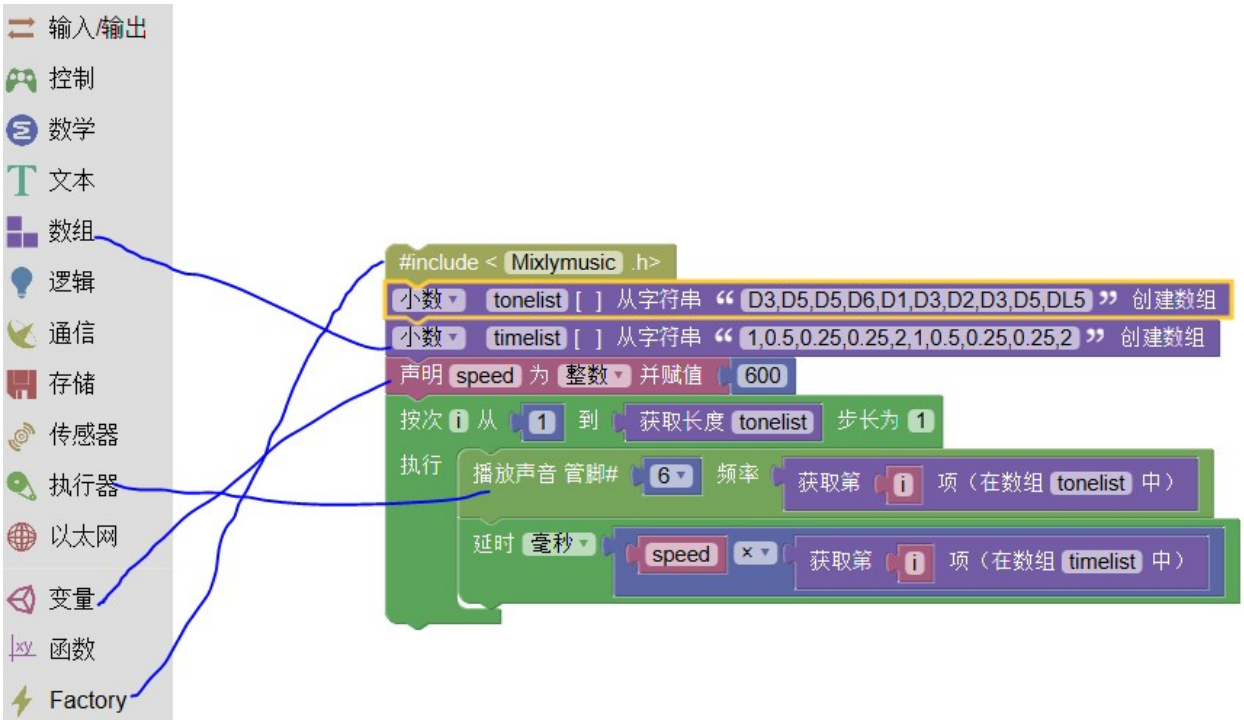
```
//每种声音对应的频率
#define D0 -1
#define D1 294
#define D2 330
#define D3 370
#define D4 393
```

```
#define D5 441
#define D6 495
#define D7 556

#define DL1 147
#define DL2 165
#define DL3 190
#define DL4 196
#define DL5 221
#define DL6 248
#define DL7 278

#define DH1 589
#define DH2 661
#define DH3 700
#define DH4 786
#define DH5 882
#define DH6 990
#define DH7 1120
```

3.在 Mixly 中，编写下面的程序



原理分析

首先讲下简单的乐理知识，知道音乐是怎么演奏出来的自然就可以通过代码来进行编排了。

1.演奏单音符 一首乐曲有若干音符组成，一个音符对应一个频率。知道对应的频率，让 arduino 输出到蜂鸣器，蜂鸣器就会放出相应的声音。这里有个表供参考：

音调 音符	1 .	2 .	3 .	4 .	5 .	6 .	7 .
A	221	248	278	294	330	371	416
B	248	278	294	330	371	416	467
C	131	147	165	175	196	221	248
D	**147**	**165**	**175**	**196**	**221**	**248**	**278**
E	165	175	196	221	248	278	312
F	175	196	221	234	262	294	330
G	196	221	234	262	294	330	371

音调 音符	1	2	3	4	5	6	7
A	441	495	556	589	661	742	833
B	495	556	624	661	742	833	935
C	262	294	330	350	393	441	495
D	**294**	**330**	**350**	**393**	**441**	**495**	**556**
E	330	350	393	441	495	556	624
F	350	393	441	495	556	624	661
G	393	441	495	556	624	661	742

音调 音符	. 1	. 2	. 3	. 4	. 5	. 6	. 7
A	882	990	1112	1178	1322	1484	1665
B	990	1112	1178	1322	1484	1665	1869
C	525	589	661	700	786	882	990
D	**589**	**661**	**700**	**786**	**882**	**990**	**1112**
E	661	700	786	882	990	1112	1248
F	700	786	882	935	1049	1178	1322
G	786	882	990	1049	1178	1322	1484

我们将上述频率写入 Mixlymusic.h。

#define D1 294的意思是：我们在Mixly中可以用D1来代替294这个频率。这样编程就容易多了。同理，DH2代表高音2，DL3代表低音3。D0代表没有声音。

注意：表中个别频率有误，写入 Mixlymusic 库时经过调整。我们的库用的是 D 调，如有别的需求可以参照修改。

在程序中如要使用.h 库文件，必须先有 

。

2.音符的演奏时间 每个音符都会播放一定的时间，这样才能构成一首优美的曲子，而不是生硬的一个调的把所有的音符一股脑的都播放出来。如何确定每个音符演奏的单位时间呢？我们知道，音符节奏分为一拍、半拍、1/4 拍、1/8 拍，我们规定一拍音符的时间为 1；半拍为 0.5；1/4 拍为 0.25；1/8 拍为 0.125……，所以我们可以为每个音符赋予这样的拍子播放出来，音乐就成了。

另外，参数 speed 可以控制整体的播放速度。

上述音符和节拍数据，都放在“数组”中。**所谓数组，就是相同数据类型的元素按一定顺序排列的集合，就是把有限个类型相同的变量用一个名字命名，然后用编号区分他们的变量的集合，这个名字称为数组名，编号称为下标。**

这一句代表我们把这些音符放在名字是 tonelist 的数组中。

这句代表从中取出第 i 个值用于播放

这里要求 timelist 和 tonelist 中的元素数量一致，音高和时长一一对应。

最后这部分，作用是按照乐谱逐个播放。

3.5 Arduino 控制电脑播放音乐

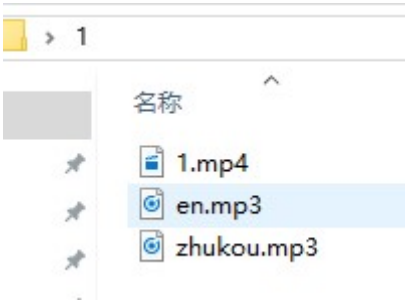
这里需要用到老师提供的软件，可到北师大二附中 moodle 平台上下载。地址：

http://m.bsdefz.edu.cn/maker

该软件可实现控制电脑播放 MP3 和 MP4 文件。

Step 1

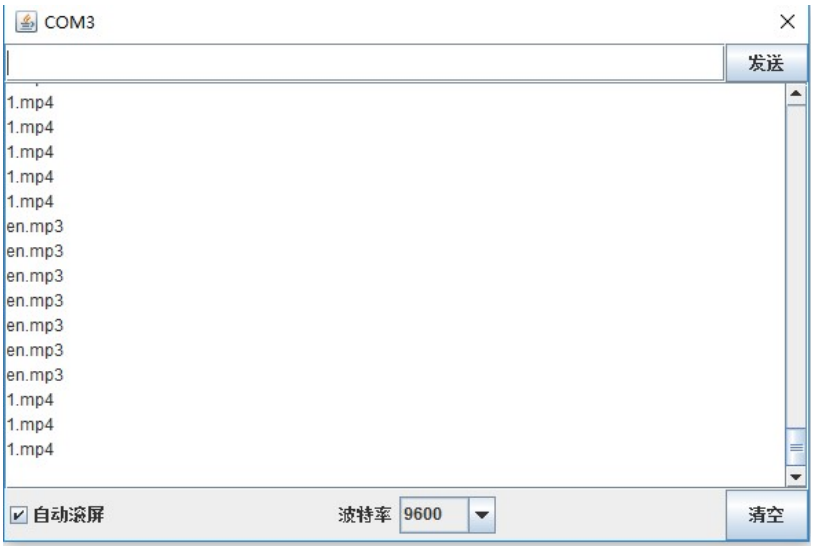
在 Arduino 程序中，用 Serial 打印（自动换行）模块，发送你想播放的媒体文件的文件名。例如，我们打算用一个接在 4 脚上的按钮控制播放音乐，可以这样写：



在后面一定要有延时（100ms 以上）。持续发送数据可能造成软件崩溃。

程序如果连续接到多个指令,只执行一次播放任务。如果需要再次播放相同文件，需要先发送一个别的指令，才能再次发送文件名。

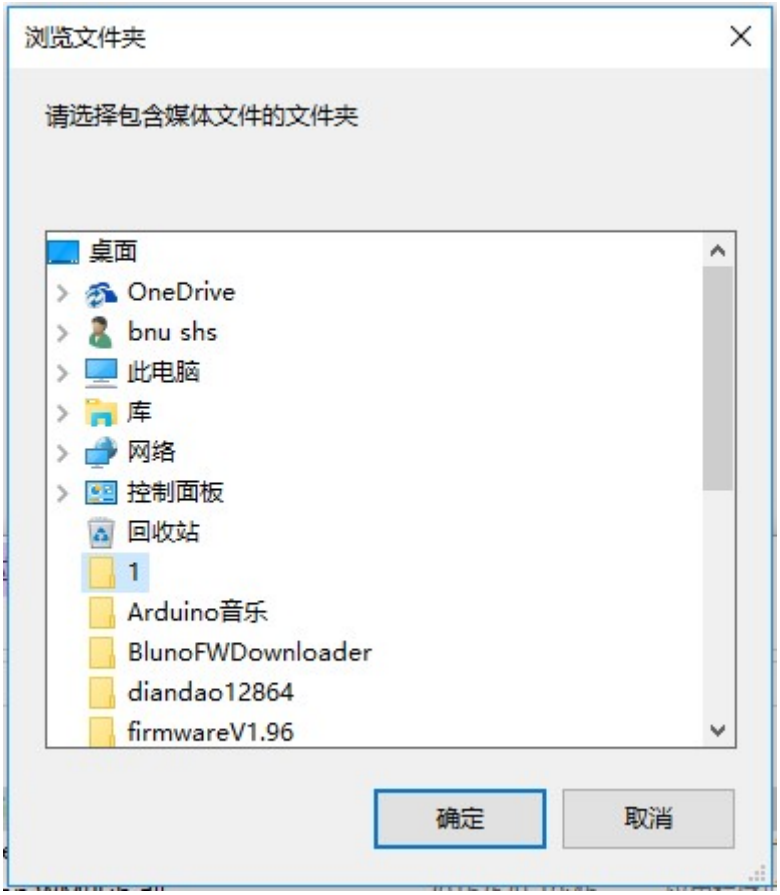
在串口监视器验证发送成功后，关闭整个 Mixly 再操作电脑上的软件。



Step 2

刚打开软件时或单击“选择 BGM 文件夹”按钮，就可以选择包含媒体文件的文件夹。

注意：所有你要播放的媒体文件要放在同一个文件夹中。



Step 3

选择文件夹后，在下拉列表中选择 Arduino 对应的 COM 口。

注意：上传程序需要暂时关闭本软件，否则会占用 COM 口导致上传失败。

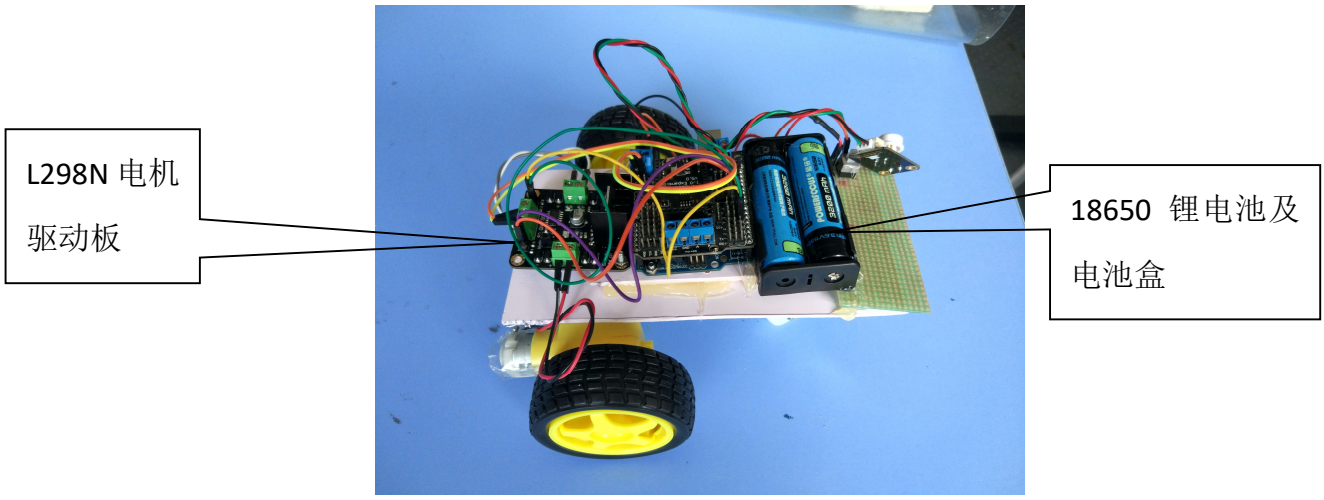
一般不是 COM1

Step 4

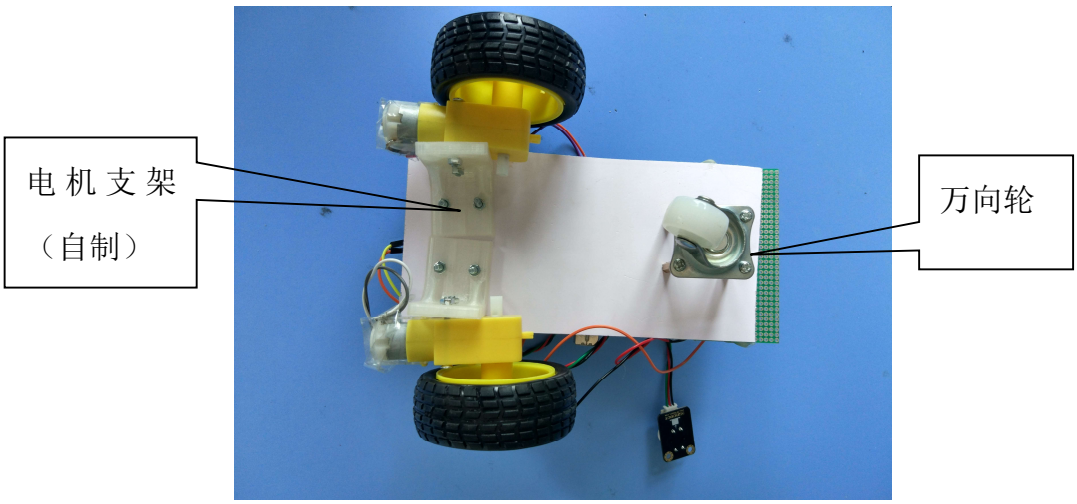
然后就可以测试效果了。当软件收到指令，就会播放相应的媒体文件。下一次如果收到不同的指令，就会自动切换播放。

第四章 智能小车设计

我们可以用 Arduino 控制 2 个电机，结合结构件、传感器等组成一个自动或手动控制的小车。



正面图



背面图

实验器材

Arduino 控制器，小车套装（含电机、车轮 2 个，万向轮 1 个，螺丝螺母等五金件，PVC 板材（制作底盘及主体结构），各种传感器，电机驱动板 L298N,18650 锂电池及电池盒。

制作部分要点介绍

1.电机需要焊接导线。焊接导线时需要注意不要损坏电极铜片，避免短路、断路。焊好的电线需要额外固定，避免意外拉坏铜片。**焊接时特别注意不要烫到烙铁电线绝缘皮，以免短路。**

2.万向轮的作用是给小车提供支点，使小车保持稳定。注意合理选择万向轮、电机的位置使小车稳定。

3. 18650 电池电压 3.7-4.2V，容量 3200mAh。**如果发生短路，容易燃烧或爆炸。时刻警惕电池盒的线头。不用时取出电池。安全千金难买。**

电机驱动板电路连接

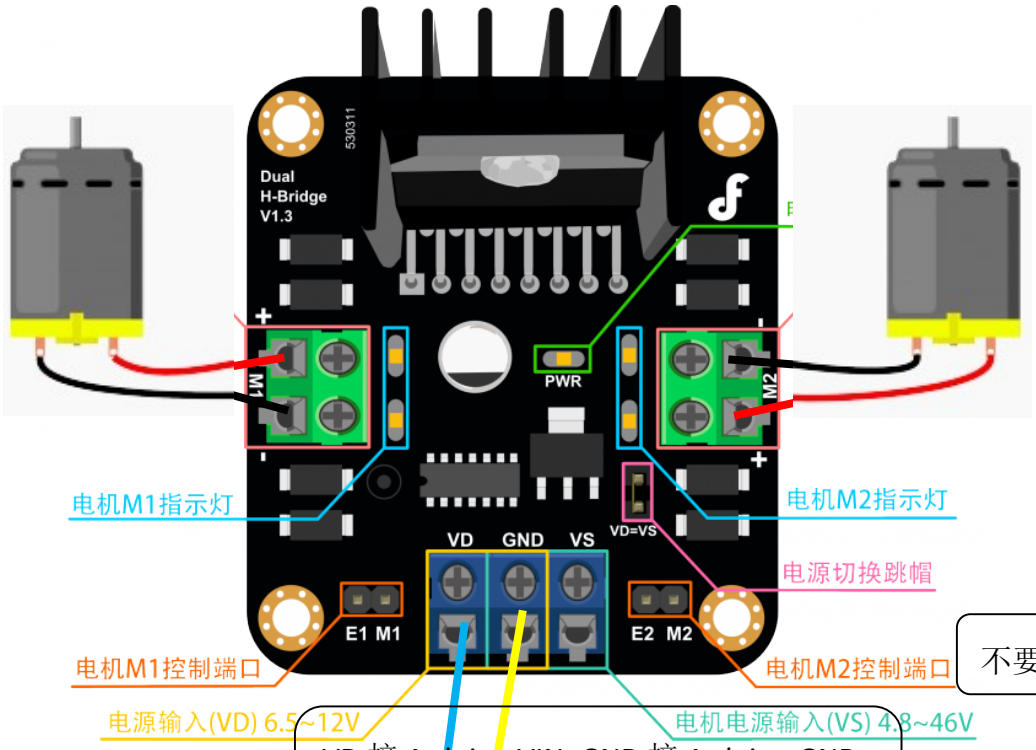
Arduino 单个管脚的输出电流不能超过 40mA，不能满足电机需要。因此要借助电机驱动板 L298N 放大电流，实现对电机的间接控制。

该元件接错电路容易烧毁，玩坏了需要赔偿。

参照下图完成电路连接。**一点都不要接错！有问题问老师。自以为是会烧坏元器件**

接线柱使用方法.mp4

接线柱使用方法

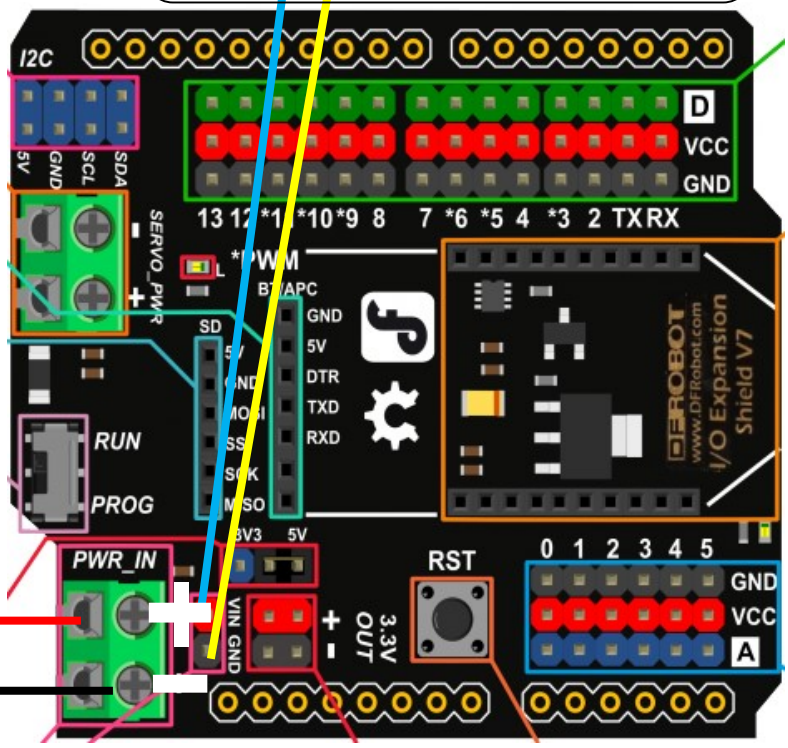


电机线接反不会烧，只会反转。

不要拔掉!!

VD 接 Arduino VIN，GND 接 Arduino GND。

用公母线连接。严禁反接



池盒红线
正极）接+
，黑线（负
）接 - 孔
严禁反接

接完以后，再用母母头线把 L298N 的 M1 接 Arduino D4，E1 接 D5，M2 接 D7，E2 接 D6。如果要更改端口，E1,E2 必须接有 PWM 模拟输出功能的管脚：3,5,6,9,10,11。

程序设计：如何通过电机驱动板控制电机

基本原理：

L298N 的 E1,E2: 分别为两个电机控制的调速端口(可使用 PWM 模拟输出调速)。

特别说明：电机输入电压应为 6V，而我们所用的电源是 2 节 18650 电池，电压约为 7.4-8.4V。因此为避免烧坏电机，模拟输出值一般可在 100-200 之间。过低容易卡转烧坏电机，过高则功率过大也容易烧坏。

M1,M2: 正反转控制信号输入端。设为高电平则对应电机正转，低电平时对应电机反转。

举个例子：

E1 设为模拟输出 255，M1 设为低电平，效果是 1 号电机全速正转。

E2 设为模拟输出 128，M2 设为高电平，效果是 2 号电机半速反转。

E1 设为模拟输出 0，无论 M1 设为什么，1 号电机都不转。

电机导线如反接，旋转方向也会反过来。

将 2 个电机组成小车后，小车的前进方向可以通过控制两个电机转速来调节。例如：

左电机转速=右电机转速，两者同向旋转：直线前进或后退

左电机转，右电机不转：右转弯

左电机转速<右电机转速，两者同向旋转：左转弯

编程示例

提示：向 Arduino 上传程序时，必须断开电池才能连接 USB 线。测试电机时，必须断开 USB 线连上电池。

下面的示例让电机 1 全速反转，电机 2 半速正转。结果是小车原地打转。

其他的可自行探索。

提示：即使两个电机的模拟输出给成一样的数，由于物理特性有轻微差异，转速也可以显著不同，导致小车不能走直线。需要自己适当调节模拟输出值来校正。

提示：严禁让电机卡转！

传感器介绍

红外避障传感器，特别注意它的连线颜色特殊！！！

红线（VCC）接红色口

绿线负（GND）接黑色口

黄线为传感器输出，接绿色口（D）

严禁接反!!!

有障碍时输出高电平，无障碍时低电平。

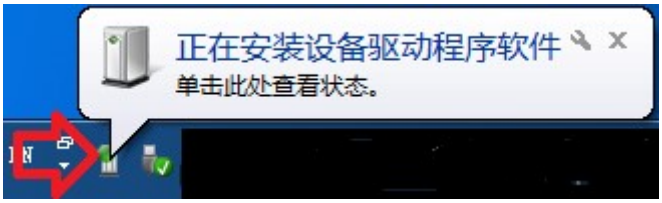
传感器背面有一个电位器可以调节障碍的检测距离。有效距离 3-80CM 可调。

超声波测距模块

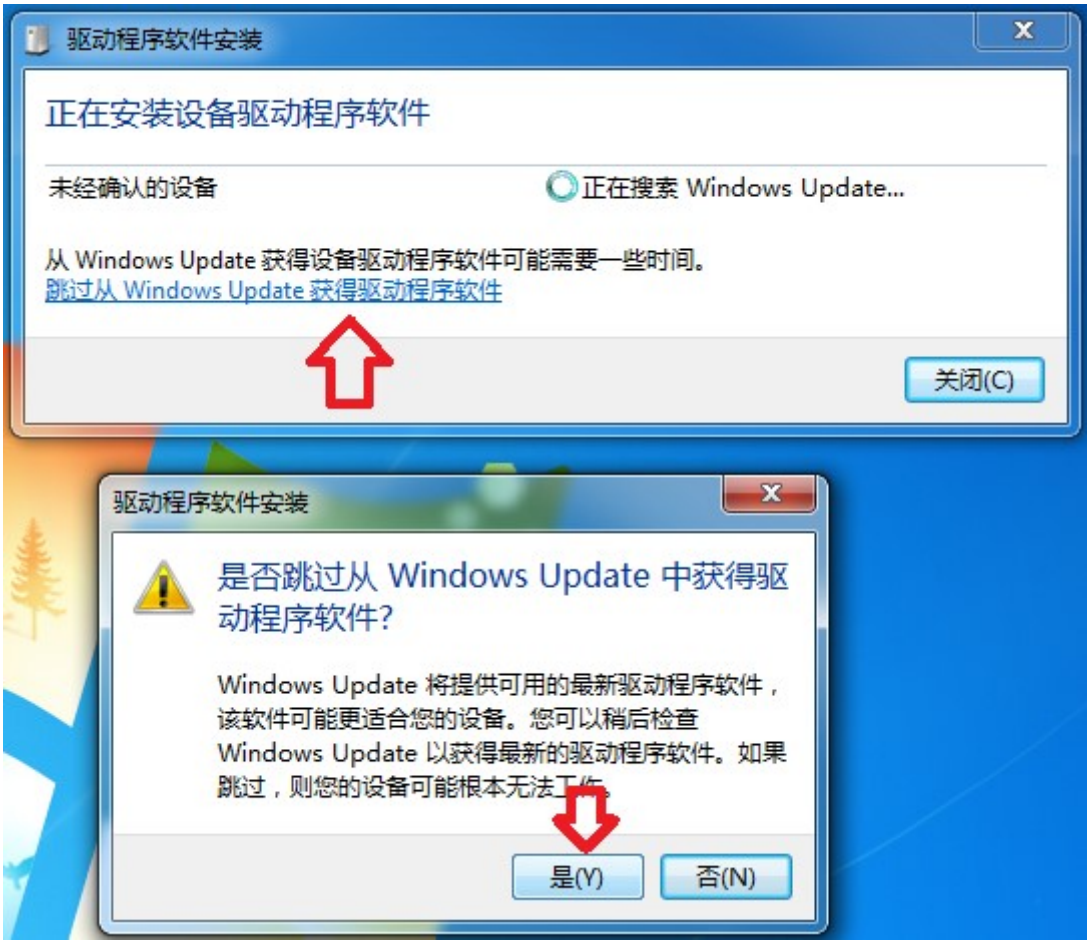
Trig 和 ECHO 接 D 口。用传感器中的超声波模块编程。

附录 准备工作：Arduino 初始设置（第一次连接）

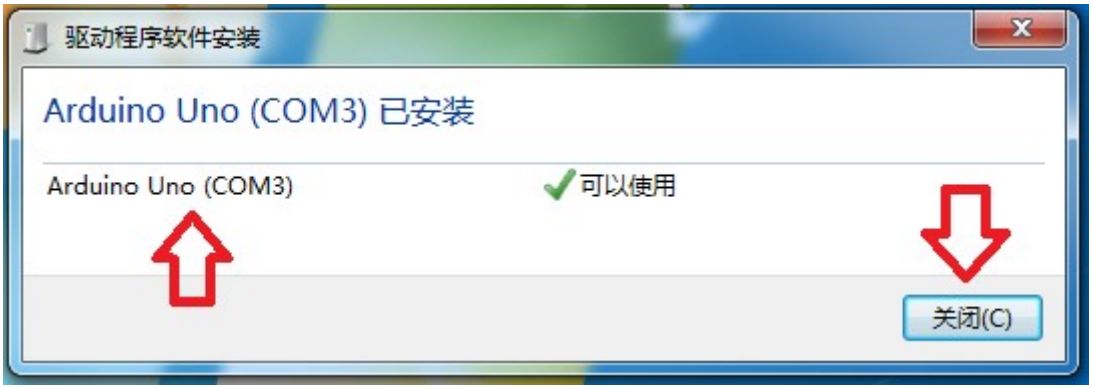
插入 USB 线，Arduino 控制板上的电源指示灯会被点亮，电脑右下角会出现:



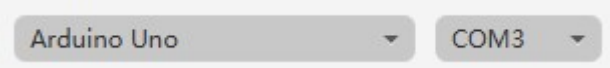
点击这个图标：出现下面的对话框，**然后点击“跳过从 Windows Update 获得驱动程序软件”**，然后点“是”



然后会从本机安装，完成后显示下面对话框，**记住它安装到 COM 几**，点关闭。


驱动程序软件安装
Arduino Uno (COM3) 已安装
Arduino Uno (COM3)
可以使用
关闭(C)

打开软件核对右下角


Arduino Uno
COM3