



普通高中教科书

XINXI  
JISHU

# 信息技术

选择性必修 6

开源硬件项目设计



教育科学出版社

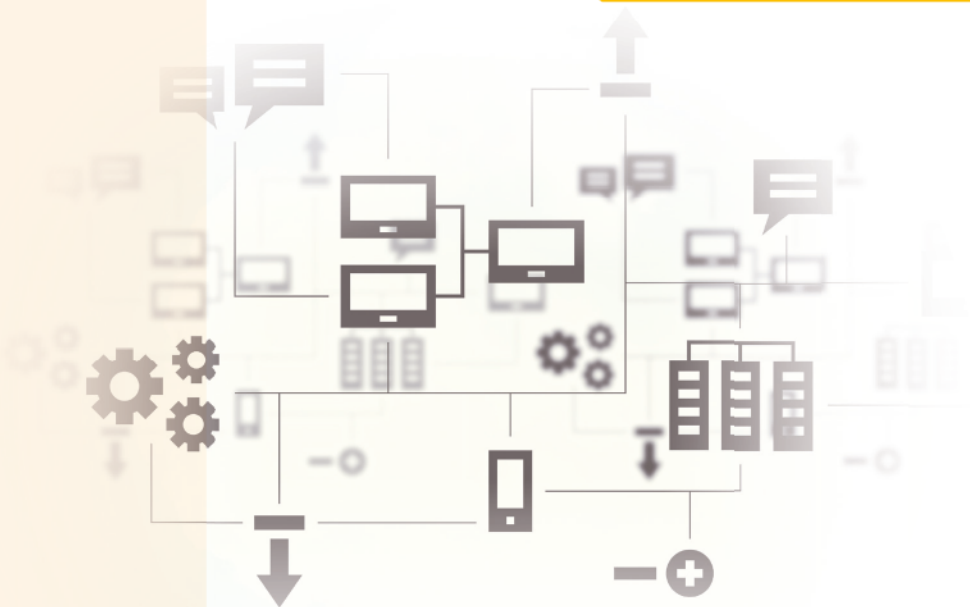
普通高中教科书

XINXI  
JISHU

# 信息技术

选择性必修 6

开源硬件项目设计



教育科学出版社  
· 北京 ·

总主编 李 艺 董玉琦  
本册主编 张义兵  
本册副主编 钟柏昌  
主要编者 张义兵 钟柏昌 张 禄 韩 蕾 闫 妮  
李婷婷 刘正云

出版人 李 东  
责任编辑 贾立杰  
版式设计 国美嘉誉文化 王 辉  
责任校对 张晓雯  
责任印制 叶小峰

## 普通高中教科书

### 信息技术 选择性必修 6 开源硬件项目设计

教育科学出版社出版发行  
(北京·朝阳区安慧北里安园甲 9 号)  
邮编: 100101

总编室电话: 010-64981290 编辑部电话: 010-64989637

出版部电话: 010-64989487 市场部电话: 010-64989009

传真: 010-64891796

网址: <http://www.esph.com.cn>

各地新华书店经销

黄冈市新华印刷股份有限公司印装

开本: 890 毫米 × 1240 毫米 1/16 印张: 10

2020 年 7 月第 1 版 2021 年 12 月第 5 次印刷

---

ISBN 978-7-5191-2252-2

定价: 16.80 元(含光盘)

图书出现印装质量问题, 本社负责调换。

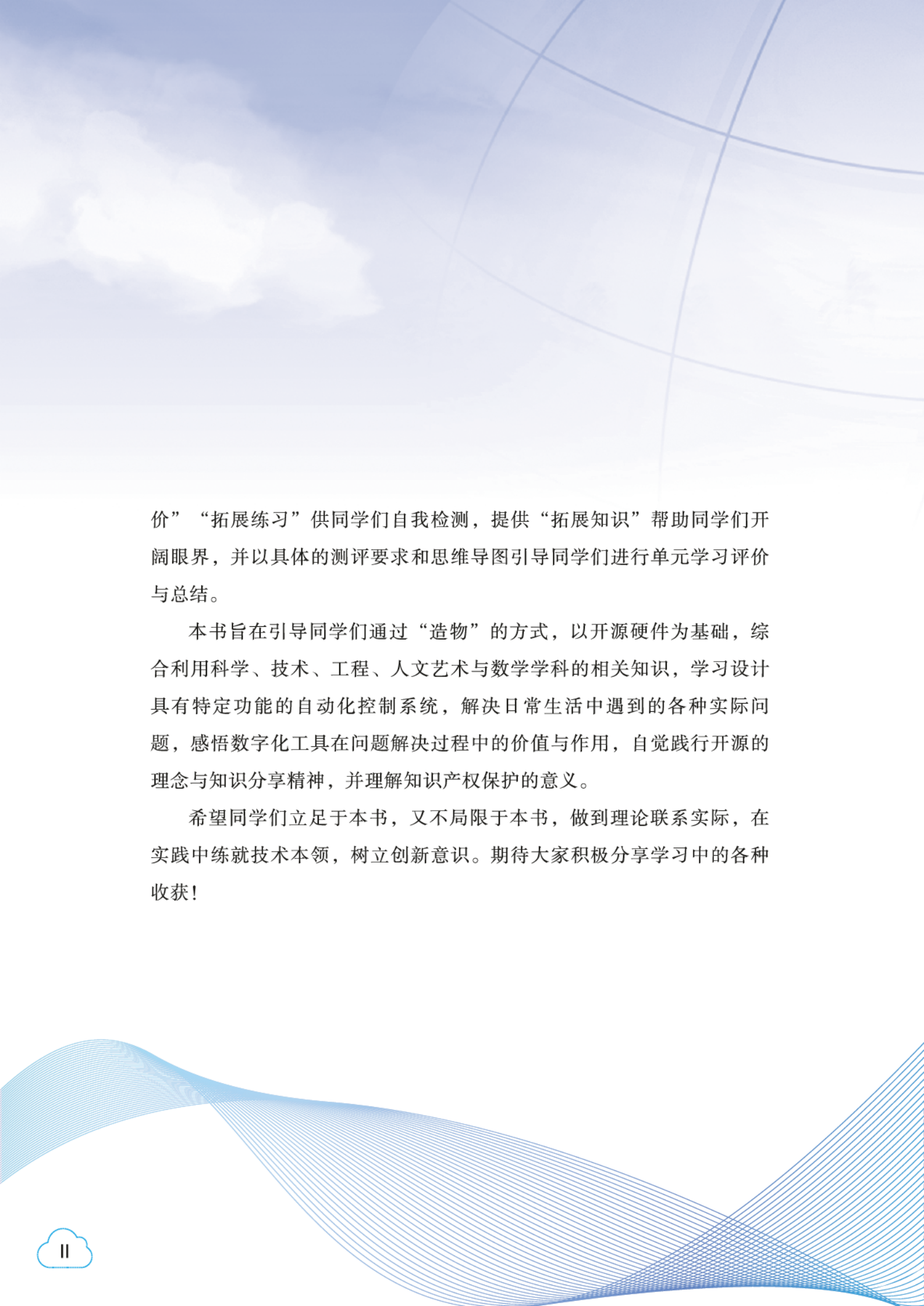
图片来源: 高品(北京)图像有限公司

# 前 言

“开物成务。”

一个有智慧的人，懂得如何利用事物发展的内在规律分析和解决问题；同样，一个有“智慧”的自动化控制系统，饱含了造物者对该系统要素的理解，以及对支撑该系统正常工作所需运行规则的设计。这些要素包括控制器、传感器、驱动器、执行器等；而运行规则则包含如何自动接收和过滤外界信息，如何将接收到的信息处理加工成有意义的数据并形成决策，以及如何将决策外化为执行器的特定行为。因此，如何设计和制作这种具有一定“智慧”的人造系统，有助于我们理解不同智能系统的运作过程，也有利于培养我们的实践创新能力。

本书根据《普通高中信息技术课程标准（2017年版）》编写，由初识开源硬件、实验模拟型项目设计、科学探究型项目设计、趣味交互型项目设计四个单元组成，旨在从认识基础硬件出发，带领同学们学习开源硬件的本体知识，然后利用开源硬件开展科学探究，以及设计趣味性联机交互系统，以解决学习和生活中的一些实际问题。本书内容主要以项目学习的方式组织，在给出明确的“学习目标”的基础上，设计生动、有趣的“任务”引领“活动”，设置必要的“作品发布与评



价”“拓展练习”供同学们自我检测，提供“拓展知识”帮助同学们开阔眼界，并以具体的测评要求和思维导图引导同学们进行单元学习评价与总结。

本书旨在引导同学们通过“造物”的方式，以开源硬件为基础，综合利用科学、技术、工程、人文艺术与数学学科的相关知识，学习设计具有特定功能的自动化控制系统，解决日常生活中遇到的各种实际问题，感悟数字化工具在问题解决过程中的价值与作用，自觉践行开源的理念与知识分享精神，并理解知识产权保护的意义。

希望同学们立足于本书，又不局限于本书，做到理论联系实际，在实践中练就技术本领，树立创新意识。期待大家积极分享学习中的各种收获！

# 目 录

<b>第 1 单元</b>	<b>初识开源硬件</b>	<b>1</b>
1.1	开源运动与开源硬件	2
1.2	Arduino 操作基础与开发流程	8
	单元学习评价	14
	单元学习总结	15
<b>第 2 单元</b>	<b>实验模拟型项目设计</b>	<b>16</b>
2.1	点亮 LED	17
2.2	控制蜂鸣器	26
2.3	按钮开关	33
2.4	声光检测	40
2.5	数码管显示	49
2.6	红外遥控	57
2.7	超声波测距	65
2.8	双电机控制	72
2.9	循线	81
	单元学习评价	87
	单元学习总结	88

<b>第 3 单元 科学探究型项目设计</b>	<b>89</b>
3.1 水温监测	90
3.2 单摆监测	97
3.3 电压检测	104
单元学习评价	111
单元学习总结	112
<b>第 4 单元 趣味交互型项目设计</b>	<b>113</b>
4.1 造型切换	114
4.2 角色交互	121
4.3 坐标定位	129
4.4 侦测	137
单元学习评价	147
单元学习总结	148
<b>附    录</b>	<b>149</b>
附录1 项目设计方案	149
附录2 音符、音调与频率的对应表	150
附录3 Arduino 编程手册	151
<b>后    记</b>	<b>154</b>

# 第 1 单元 初识开源硬件

随着科技的发展与社会的进步，人们已经步入智能生活时代。智能家居、智能穿戴设备、智能交通等越来越多的科技产品出现在我们的身边，生活也因此变得更加方便、快捷、舒适。那么，同学们有没有想过自己动手设计并开发一些智能设备？开源硬件功能强大且易于上手，将帮助大家实现“创造”的梦想。

本单元将初步介绍开源硬件。通过本单元的学习，同学们将了解与开源硬件相关的概念、背景，以及开源硬件的特征、类型和应用范围；了解到一种非常典型和被广泛应用的开源硬件——Arduino，并初步掌握 Arduino 的使用方法，体会使用 Arduino 进行项目开发的乐趣。





## 1.1 开源运动与开源硬件

随着现代社会的开放式发展，在信息产业领域出现了一种新的潮流——开源运动。开源运动打破了信息技术由信息技术企业或信息技术专业人员所专有的传统，普通大众也可以借助开源软、硬件开展“创造”活动，从单纯的产品消费者转变为兼具产品创造者的身份。因此，作为新时代的公民，在成为信息技术产品创造者之前，首先应该了解开源运动及相关的开源硬件。



### 学习目标

- ★ 知道开源运动的背景与发展历史。
- ★ 了解开源硬件的概念、本质与特征。
- ★ 掌握开源硬件的常见分类与应用范围。

软件的开源运动由来已久，影响甚广，作为一种开放性思想支持下的活动，广受社会关注和欢迎，且成为引领技术与市场发展的重要力量。在开放思想的影响下，由部分企业推动，也相应地形成了一场关于硬件的开源运动。当然，硬件的开源与软件的开源既有相似之处，又有所不同。本节我们将围绕“开源运动与开源硬件调研”项目展开学习，目的是帮助同学们了解开源运动与开源硬件的相关历史和发展过程，了解开源、共享理念，为开启智能生活打下基础。本项目主要包含“开源运动调研”和“开源硬件调研”两个任务。



### 任务一 开源运动调研

#### ※ 活动 了解自由软件运动

开源运动与自由软件运动有着密切的联系，可以说开源运动是

自由软件运动的后续效应。20世纪60年代，美国麻省理工学院计算机专业的学生以及IBM公司等开发的软件都是自由分发的，那个时候的商业模式都是以硬件为中心，没有人会想到软件的权利，也不会将其应用于商业行为。他们提供软件的同时也提供源代码，软件可以被修改并因此得到改进，这便是开源软件的早期雏形。请同学们根据表1.1.1的提示，分组上网查找自由软件运动期间发生的重大事件，并完善表1.1.1。

表1.1.1 自由软件运动发展历史调查表

时间	事件
20世纪80年代	
20世纪90年代	
.....	



自由软件和开源软件之间的区别可以理解成社会运动（自由软件）和软件开发方法（开源软件）之间的区别。

开源软件意味着在开放发行版本和授权许可的同时也开放源代码；自由软件则意味着使用该软件的用户决定如何处理软件的代码、发行版本和授权。

## ● 开源运动

1998年，乔恩“疯狗”霍尔（Jon “maddog” Hall）、拉里·奥古斯丁（Larry Augustin）、埃里克·雷蒙德（Eric S. Raymond）和布鲁斯·佩伦斯（Bruce Perens）等开始倡导开放源码运动，即开源运动。他们以各自卓越的专业基础极大地推动了这项运动的发展。

开源运动和20世纪90年代末互联网的兴起一起造就了Linux操作系统的流行，随后出现了很多支持开源运动的公司，如Corel、Sun Microsystems、IBM。在21世纪初，互联网泡沫破碎，开源软件作为某些昂贵的专有软件的替代品，被推到重要的位置上。大量易用的应用程序的出现使它的发展势头进一步增强。显著的价格优势和更强大的功能使Linux成为商业和家庭用户的可行选择。就这样，开源的理念改变了这个充满专利和授权的行业。

莱纳斯·托瓦尔兹（Linus Torvalds）曾说，Linux成功的奥秘并不在于源代码本身，而是在于其开发方式，即允许所有程序员参与开发源码，与他人共享自己的开发成果。在此理念的基础上，开源社区不断壮大，创新浪潮持续高涨。因此，开源运动的本质是对自由、开放、透明、共享、协同、整合等信念的主张与施行，这种信念不仅影响着软件行业，也影响着整个社会。



Linux 开放源代码：可以被任何人获取和使用；可以根据个人的要求和所使用的平台定制；可以自由地以当前或修改的形式重新发布。



1997年，开源运动的发起人之一布鲁斯·佩伦斯发起了开源硬件认证计划并比较详细地介绍了开源硬件。他是一位无线电发烧友，他提出开源硬件认证计划是为了让硬件生产商可以自行认证他们的开源硬件产品。



## 任务二 开源硬件调研

### ※ 活动1 调查开源硬件的由来及发展

开源硬件属于开源文化的一部分，是受开源软件启发出现的概念。事实上，硬件开放可以追溯到很久之前。随着集成电路的发展，硬件的开放也经历了三个时期的发展，请同学们根据表1.1.2的提示，分组上网查找这三个时期中发生的重要事件，并完善表1.1.2。

表1.1.2 开源硬件发展历史调查表

时间	事件
20世纪60至70年代	
20世纪80年代	
20世纪90年代末至21世纪	

### ● 开源硬件

与开源软件相对应，开源硬件是指将硬件的详细信息，如设计图、电路图、材料清单等都公开，与他人共享。开源硬件鼓励合作、分享与创新等精神，一般具有以下几个方面的特征。

一是开源分享。作为开源文化的一部分，开源硬件最大的特点莫过于开源分享。一方面不受单一厂商的约束，可以避免产品、技术和价格上的垄断；另一方面有丰富的开源资源可以利用，有大量的作品创意和代码可供模仿和借鉴。

二是扩展性好。开源硬件一般都留有一些扩展接口，给用户极大的自主性。因此，很多开源硬件不仅在硬件方面可以相互兼容，还可以与众多软件的编程环境相结合。开源硬件也因其良好的扩展性促进了不同团队之间的合作。

三是成本优势。开源硬件从硬件设计到驱动、软件开发工具包（SDK）都是开放授权并可免费获得的，另外开源硬件也有不同的价位，可以满足不同的预算需求。

## ● 常见的开源硬件产品

无论是3D打印机，还是各类可穿戴设备，大多数开源项目都是基于开源硬件开发的。当前市场上的开源硬件有上百种，以下几种比较有代表性。

### 1. Arduino

Arduino 诞生于意大利一所高科技设计学校。由于学生们经常抱怨找不到便宜、好用的控制板，马西莫·班齐（Massimo Banzi）与戴维·库阿塔尔斯·鲁伊斯（David Cuartielles Ruiz）讨论之后决定自己组队设计控制板，并为控制板设计编程语言。这块控制板被命名为 Arduino，从设计编程语言到设计电路，仅用了5天时间。随后，Arduino 的开发团队决定将其源代码开放，允许任何人自由使用该源代码生产和销售控制板的复制品，或通过修改源代码重新设计控制板。Arduino 控制板具有体积小、质量轻、价格便宜、易上手等特点，是最常见的开源硬件之一。如图 1.1.1 所示的是发烧友以 Arduino MEGA2560 为主控制器设计制作的 3D 打印机。

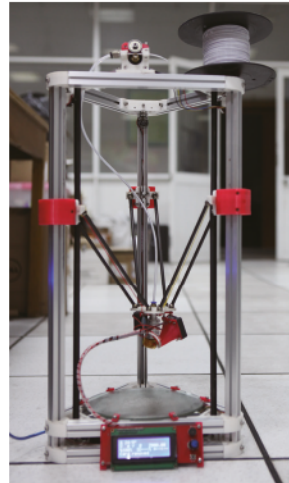


图 1.1.1 用 Arduino 制作的 3D 打印机

### 2. Raspberry Pi

Raspberry Pi 可翻译为树莓派，是一款基于 ARM 的微型计算机主板，以 SD/MicroSD 卡为存储设备，可连接键盘、鼠标和网线，同时拥有视频模拟信号的电视输出接口和 HDMI 高清视频输出接口，



图 1.1.2 智能冰柜

以上部件全部整合在一张仅比信用卡稍大的主板上，具备所有个人计算机的基本功能，只需接通电视机和键盘，就能实现诸如电子表格处理、文字处理、玩游戏、播放高清视频等诸多功能。如图 1.1.2 所示的智能冰柜，是科技爱好者杰米·贝利（Jamie Bailey）使用树莓派 2 打造的，它能够监测剩余啤酒数量、温度以及冰柜门是否正常关闭等。

### 3. BeagleBoard

BeagleBoard 是全球较早出现的开源硬件平台。BeagleBoard 的设计初衷是为全球的开发者和爱好者提供一个低价格、高性能、使用方便的嵌入式开发板。其功能强大，包含 2D/3D 图形硬件加速模块，并且完全开源。作为开源硬件的典范，BeagleBoard 在全球拥有众多开发者，他们共享软件资源，相互交流开发经验。目前很多嵌入式开发板的设计，都以 BeagleBoard 作为参考标准。如图 1.1.3 所示的 GamingCape 掌上游戏机，是基于 BeagleBoard 的一款名为 BeagleBone Black 的控制板开发的，它将所有硬件都安装在一个定制的 3D 打印外壳内。

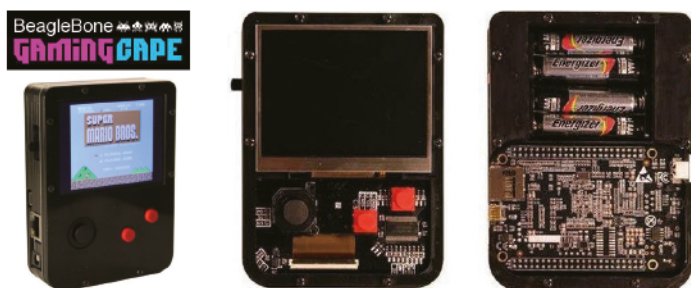


图 1.1.3 GamingCape 掌上游戏机

### 4. Intel Edison

Intel Edison 由英特尔中国研究院设计开发，并在 2014 年 9 月的英特尔信息技术峰会上正式发布。它是一款专门针对小型智能设备设计的双核芯片，可以通过 Wi-Fi 和蓝牙对外通信。在处理性能上，Intel Edison 相当于一台奔腾计算机，在体积上，它只有 SD 卡大小。如图 1.1.4 所示的 Mimo Baby 智能婴儿连体衫，有一个装有 Intel Edison 的乌龟造型的盒子，可以感测婴儿的睡眠状况、体温、呼吸情况以及分析婴儿的睡姿等，经过数据分析传到智能手机上，让父母随时随地了解婴儿的身体状况。

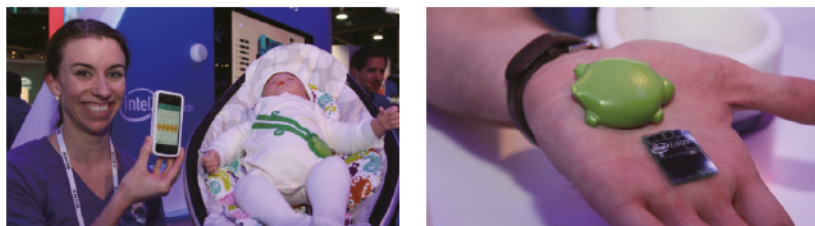


图 1.1.4 Mimo Baby 智能婴儿连体衫

### 5. micro:bit

micro:bit 是一款由英国广播电视公司（BBC）设计的 ARM 架构的

单片机，它在控制板中内嵌了蓝牙、重力传感器（加速度传感器）、磁力传感器（电子罗盘）、温度传感器、三个按钮、 $5 \times 5$  LED 点阵等，既可以进行图形化编程（如 MakeCode 编辑器），也可以用代码编程（如 Python 编辑器）。使用 micro:bit 可以设计出很多酷炫的小发明。图 1.1.5 中左边的图所示的项目，是一位电子爱好者用它设计制作的一款电子吉他。此外，micro:bit 还可以连接至其他的计算设备，包括 Arduino 和 Raspberry Pi 等，实现更多复杂的任务。图 1.1.5 中右边的图所示的项目，就是用 Raspberry Pi 控制数个 micro:bit 的 LED 点阵，实现滚动显示文字的功能，图中所示文字为“HI WORLD!”。

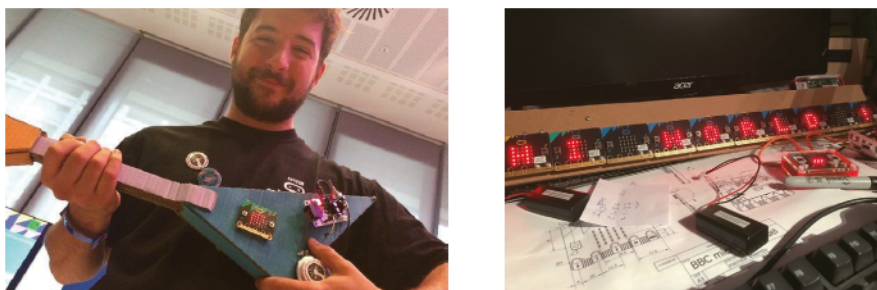


图 1.1.5 利用 micro:bit 设计的作品

## ※ 活动 2 调查国内的开源硬件产品

请同学们进行小组合作，利用互联网查找开源硬件产品，填写表 1.1.3，最后每个小组选择一款比较有代表性的产品进行全班汇报。

表 1.1.3 开源硬件产品调查表

产品名称	
开源硬件类型	
开发者信息	
产品简介	
影响力	
应用实例	

## 1.2 Arduino 操作基础与开发流程



由于使用 Arduino 开发的项目均具有一定的智能，因此也可以称为开源机器人或 Arduino 机器人。在本书中，不严格区分开源硬件项目、开源机器人、Arduino 机器人。

目前市面上的开源硬件有上百种，它们各具特色。其中，Arduino 使用简单、价格低廉，有不同类型的控制板和兼容的电子元器件，非常适合机器人等创意电子作品的开发；另外，Arduino 的编程环境也很丰富，有适合初学者的图形化编程，也有适合高级用户的代码编程。本书以 Arduino 为例介绍开源硬件。



### 学习目标

- ★ 认识 Arduino 控制板与 IDE 编程环境。
- ★ 能够正确地安装 Arduino 控制板的驱动。
- ★ 能够给 Arduino 控制板上传程序，了解开源项目的开发流程。

本节通过“认识 Arduino”项目展开学习，同学们可以了解 Arduino 的操作基础，打开学习 Arduino 的大门，从此进入开源硬件的世界。本项目主要包含“使用 Arduino 的前期准备”和“Arduino 使用初体验”两个任务。




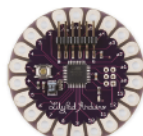



### 任务一 使用 Arduino 的前期准备

#### ※ 活动1 认识 Arduino 控制板

Arduino 控制板的类型有很多，如 UNO、LEONARDO、MEGA、Lilypad 等。请同学们利用网络资源查找 Arduino 控制板的类型及各自的特点和使用实例，完成表 1.2.1。

表 1.2.1 Arduino 控制板调查表

Arduino 控制板的类型	特点	使用实例
 UNO		
 LEONARDO		
 MEGA		
 Lilypad		

 单片机又称单片微控制器，是一种集成电路芯片，它采用超大规模集成电路技术把具有数据处理能力的中央处理器（CPU）、随机存储器（RAM）、只存储器（ROM）、多种输入/输出接口和中断系统、定时器/计数器等功能集成在一起，相当于一台微型的计算机。和计算机相比，单片机只缺少了输入/输出设备。

## ● Arduino 控制板

Arduino 控制板是采用 8 位 AVR 单片机开发的一种开源硬件。本书使用的是基于 Arduino UNO 开发的一款控制板，如图 1.2.1 所示。与常见的标准 UNO 板相比，该款控制板增加了扩展板（也称盾板）、电机驱动和超声波接口等，以支持电机、超声波以及更多的外接接口。



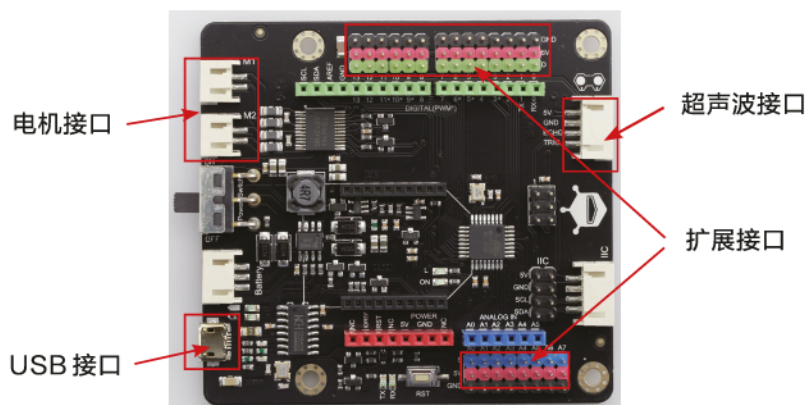


图 1.2.1 Arduino 控制板

## ● Arduino 的系统构成


如同计算机系统一样，一个完整的 Arduino 系统包括存储器、运算器、控制器、输入设备、输出设备等。其中，存储器、运算器、控制器集成在 Arduino 控制板中，而输入设备、输出设备统称外部设备。Arduino 控制板与外部设备之间的通信称为输入/输出（Input/Output, I/O）。输入过程是从信息处理系统外部将待处理的信息经必要的转换，以系统能够接受的形式送入系统；输出过程则是把经过信息处理系统处理过的信息以人或其他系统能够接受的形式输出。

### ※ 活动2 安装 Arduino 控制板的驱动

在使用 Arduino 控制板时，需要为计算机安装驱动。驱动程序在 Arduino IDE 安装目录的“drivers”文件夹中。下面以 Windows 7 的 64 位操作系统为例，介绍安装驱动的过程。

首先在“drivers”文件夹中找到“dpinst-amd64.exe”文件后双击运行，在弹出的“设备驱动程序安装向导”对话框中单击“下一步”按钮，如图 1.2.2 所示。

在安装过程中，会先后三次弹出安装提示对话框，如图 1.2.3 所示，均单击“安装”按钮即可。

 Windows 10 自带 Arduino 驱动程序，无须安装即可识别 Arduino 控制板。

如果是 32 位的系统，则需要运行“dpinst-x86.exe”文件。

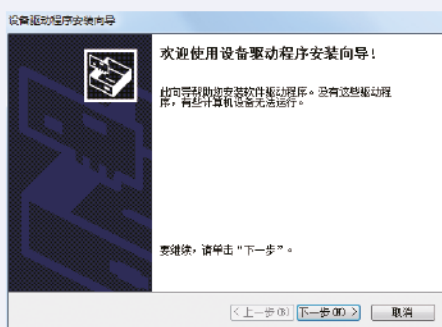


图 1.2.2 “设备驱动程序安装向导”对话框

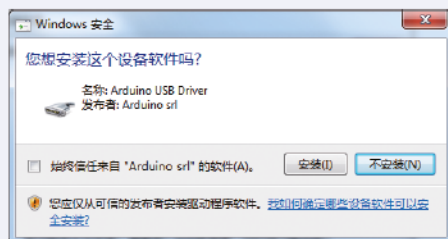


图 1.2.3 安装提示对话框

安装成功后会显示安装完成对话框，如图 1.2.4 所示。

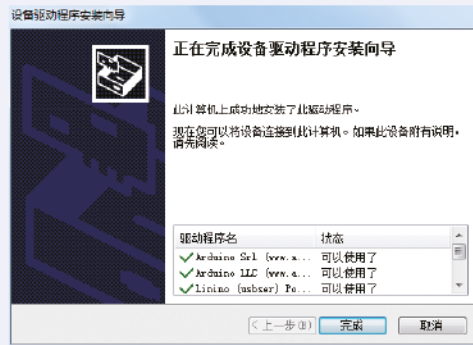



图 1.2.4 安装完成对话框

## ● IDE 编程环境

在桌面上找到图标，打开后便是 Arduino 的 IDE 编程环境，如图 1.2.5 所示。在程序编辑界面中自动呈现了两个函数，即 setup() 与 loop() 函数。在 Arduino 控制板上电或复位后，setup() 函数将只运行一次，用来初始化变量、设置引脚模式等，而 loop() 函数将被重复执行，直到切断控制板的电源。

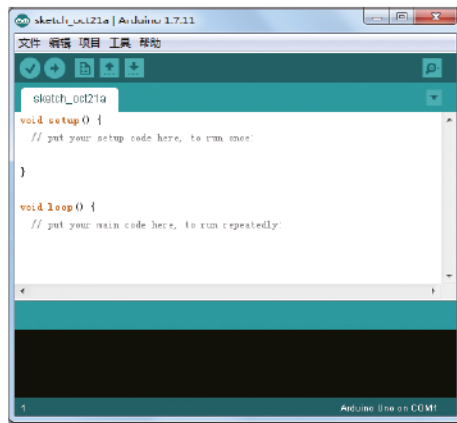


图1.2.5 IDE 编程环境



## 任务二 Arduino 使用初体验

### ※ 活动1 给 Arduino 控制板上传示例程序

要赋予 Arduino 机器人特定的行为，必须给控制板输入相应的程序。在 IDE 的菜单栏中依次选择“文件→示例”命令，将会看到许多 Arduino 程序示例，在“01.Basics”类别中找到 Blink，如图 1.2.6 所示，单击打开该示例程序。

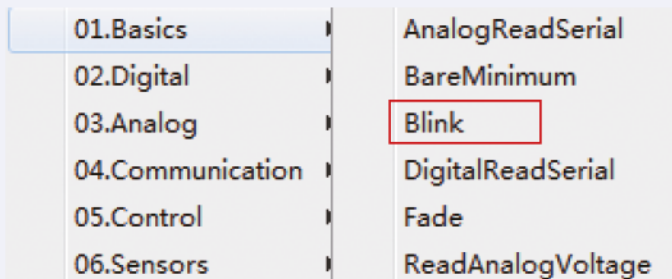






图 1.2.6 Blink 示例程序

 在Arduino中，编译是验证程序是否有语法错误；上传是在确定程序没有语法错误后将程序从计算机中传到控制器中。

 端口号可以在计算机的设备管理器中查看；板的类型选择UNO即可。

用数据线将Arduino控制板连接到计算机上，在“工具”菜单下选择合适的端口号和板的类型，如图1.2.7所示。IDE中的图标和分别用于程序的编译和上传，当选择合适的端口号和板的类型之后，就可以将程序上传到Arduino控制板中了。

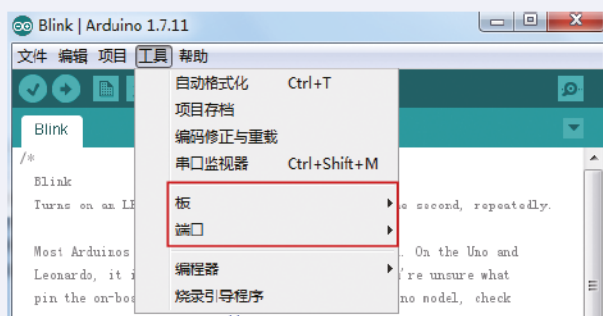


图1.2.7 选择端口号和板的类型

程序上传成功之后，IDE中会有上传成功提示，如图1.2.8所示。仔细观察Arduino控制板，有什么变化吗？

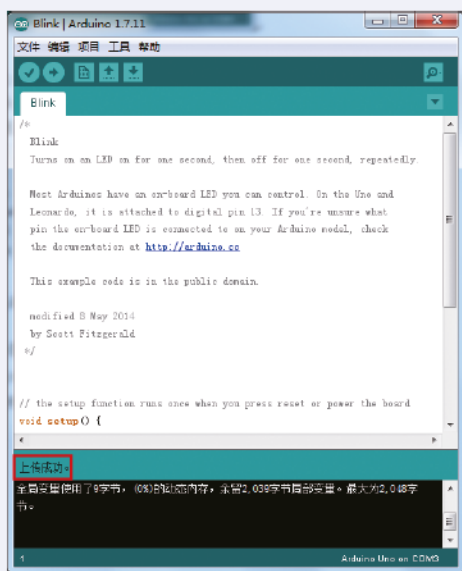


图1.2.8 上传成功提示

## ● 开源硬件项目开发的一般流程

事实上，无论是用Arduino，还是用其他开源硬件，开发一个项目的流程都是相似的，一般包括以下几个步骤。

一是项目规划。从实际需求出发，通过调研明确项目需求；围绕项目需求，设计初步的可选方案；经过进一步的分析论证，确定一个最终的可行方案；对选择的方案做进一步的细化设计。

二是项目实现。开源硬件项目的实现包括两个方面：一是通过硬件搭建和外形包装形成实物造型，二是通过编写程序并上传控制板以实现

预期的功能。

三是测试与优化。即通过测试与调试确保项目功能的完整性，通过优化使项目更符合实际需要。

### ※ 活动2 Arduino 开源项目调研

本着开源共享的精神，网络上有很多开源社区提供Arduino开源项目的介绍和源代码，请同学们查找开源社区中感兴趣的Arduino项目，填写表1.2.2。

表1.2.2 Arduino 项目调查表

序号	开源社区	项目名称	项目功能简介
1			
2			
3			
4			



### 拓展练习

#### 一、单项选择题

1. 本书所采用的Arduino控制板的型号属于（ ）  
A. MEGA      B. LEONARDO      C. UNO      D. Zero
2. 下面不属于开源硬件的是（ ）  
A. Arduino      B. Android      C. Intel Edison      D. Raspberry Pi

#### 二、填空题

1. 一个完整的Arduino系统包括输入设备、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_等。

2. Arduino IDE编程环境中自带了setup()与loop()两个函数，当Arduino控制板每次上电或复位时，\_\_\_\_\_只运行一次；\_\_\_\_\_是循环执行的函数。

## 单元学习评价

通过本单元的学习，我们了解了开源硬件的相关背景内容，体验了Arduino开发项目的基本过程。你是如何看待开源运动与开源硬件的？你能否对一个开源硬件项目进行简要的评价呢？请参与小组交流、讨论并反思，开展小组评价或自我评价。

1. 开源硬件的一般特征包括\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_。
2. 常见的开源硬件类型有\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_。
3. 请上网查找一个开源硬件设计的项目，对其进行简要的评价。

项目名称：\_\_\_\_\_

项目解决的主要问题：\_\_\_\_\_

涉及的开源硬件类型：\_\_\_\_\_

涉及的输入设备：\_\_\_\_\_

涉及的输出设备：\_\_\_\_\_

项目的优点：\_\_\_\_\_

项目的缺陷：\_\_\_\_\_

你认为该项目的开发可能经历了哪些过程？

---

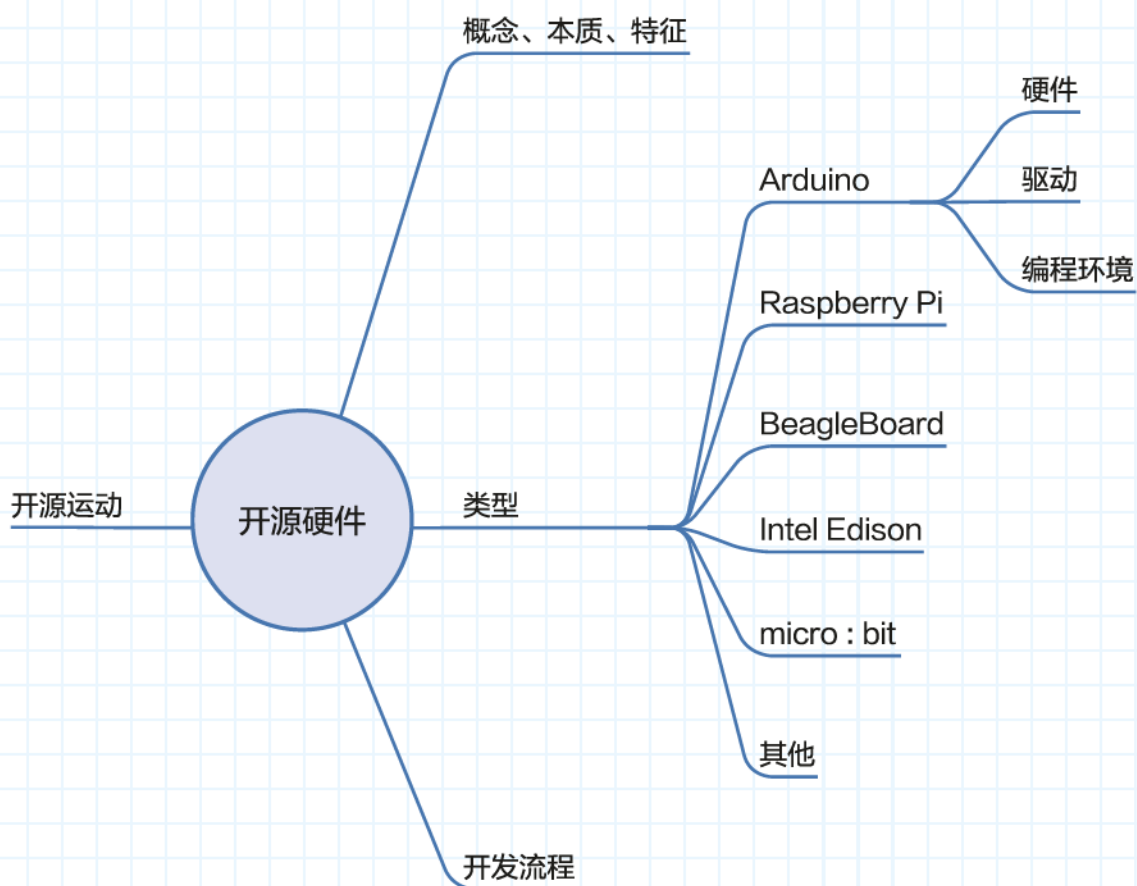
---

---

---

---

# 单元学习总结



## 第 2 单元 实验模拟型项目设计

在了解开源硬件的基础上，本单元将带领同学们掌握开发开源机器人的硬件与软件的基础，通过硬件搭建和程序编写，学习制作具有一定智能的创意作品。为了将知识学习与能力培养结合起来，每一节的内容都围绕一个项目展开，每个项目都侧重对某类传感器的使用，也就是说，我们将在完成项目的过程中获取新知识。

本单元共包含9节，每节包含一个常规项目，并在“拓展练习”环节提供若干项目选题。这些常规项目都属于实验模拟型项目，主要以模拟实验或模拟再现生产生活中既有科技产品或事物为目标。每个项目中有多个任务，先是项目规划，确认项目的可行性，然后进行项目实施，最后测试，并对项目进行优化。在完成项目的过程中，适时讲解同学们所需要的知识。所有的作品制作完成后，组织同学们开展作品的自评、互评等。在“拓展练习”环节，希望同学们从生活出发，发挥想象，设计和制作出更具特色的作品。

本单元的内容是围绕输入/输出设备展开的，通过前4节我们可以建立初步的数字输入/输出和模拟输入/输出的概念；后面5节涉及更为复杂的输入/输出以及多个输入/输出设备的配合使用。



## 2.1 点亮LED

在生活中，大多数电器设备都有指示灯，用于提示设备的工作状态，这些指示灯通常为LED（Light Emitting Diode，发光二极管）。LED具有能耗低、亮度高、使用寿命长等特点，被广泛应用在各个领域中。



### 学习目标

- ★ 熟悉Arduino编程环境，理解数字输出与延时函数的功能。
- ★ 能够编程控制LED按一定规律亮和灭。
- ★ 通过连接LED和Arduino控制板，感悟标准化构件和接口的意义。

交通信号灯亮的时间长短不同，我们能不能自己制作一个交通信号灯，并控制灯的亮灭规律呢？如何才能实现交通信号灯的亮灭效果呢？

要制作一个交通信号灯，需要考虑以下问题：如何自动“变换”灯的颜色？如何控制灯亮的时间？使用何种颜色的灯？需要多少个灯？灯的控制策略如何？要解决这些问题，需要完成以下三个任务：交通信号灯项目规划、人行横道交通信号灯的实现、人行横道交通信号灯的测试与优化。



### 任务一 交通信号灯项目规划

#### ※ 活动1 交通信号灯项目调研

常见的交通信号灯有机动车交通信号灯和人行横道交通信号灯。请同学们回忆一下自己看到的交通信号灯，填写表2.1.1。



表 2.1.1 交通信号灯调查表

序号	位置	类型（行人用/机动车用）	红绿灯交替过程
1			
2			
3			
4			

有些交通信号灯亮的时间设置不是很合理。例如，有些马路很宽，人行横道绿灯亮时的持续时间却很短，行人来不及走过去，绿灯就熄灭了；夜间车流量和人流量较小，人行横道交通信号灯亮灭的时间设置仍和白天一样，导致经常会有车在空无一人的路口等上数十秒。因此，我们可以制作一个人行横道交通信号灯，使它可以根据人们的实际需求亮灭，以方便通行。

从生活中人行横道交通信号灯的应用来看，要实现不同颜色交通信号灯的交替亮灭，主要任务是控制交通信号灯亮灭的时长，同时考虑是否需要辅助的倒计时显示。

### ※ 活动2 人行横道交通信号灯的可选方案分析与设计

经过上面的分析，从灯的数量和是否显示倒计时两个角度出发，我们可以初步设计几种方案，如表2.1.2所示。

表 2.1.2 人行横道交通信号灯的可选方案

类别	方案	方案描述
灯的数量	方案一	用1个LED交替显示红、绿两种颜色来模拟人行横道交通信号灯
	方案二	用2个LED分别显示红、绿两种颜色来模拟人行横道交通信号灯
倒计时显示	方案三	用LED模拟人行横道交通信号灯，并显示倒计时
	方案四	用LED模拟人行横道交通信号灯，不显示倒计时

在实施方案之前，我们可以从两个方面分析每个方案的利弊：从控制LED亮灭的难度看，单色LED的亮灭比较容易实现，而控制全彩LED（RGB LED）的实现方式较为复杂；从实际生活应用的角度看，并非所有人行横道交通信号灯都显示倒计时，而且实现倒计时效果需要使用数码管，实现方法较为复杂。在综合考虑学习难度和成本等因素后，可以选择方案二和方案四作为最终实施方案，即用两个LED分别显示红、绿两种颜色来模拟人行横道交通信号灯，并且不加倒计时显示功能。

要实现以上方案，关键在于如何使LED按一定规律亮灭。这涉及两个问题：一是点亮和熄灭LED，二是LED亮灭状态的保持。具体方案如表2.1.3所示。

表 2.1.3 人行横道交通信号灯设计方案


作品名称	人行横道交通信号灯
作品功能	红灯与绿灯交替亮灭，实现对人行横道交通信号灯的模拟
器材清单	Arduino控制板、USB数据线、红色LED、绿色LED
运行流程	<pre> graph LR     A[红灯亮, 保持5秒] --&gt; B[熄灭红灯、绿灯亮, 保持3秒]     B --&gt; C[熄灭绿灯, 保持0.5秒]     C --&gt; D[绿灯亮, 保持0.5秒]     D --&gt; E[熄灭绿灯, 保持0.5秒]     E --&gt; F[绿灯亮, 保持0.5秒]     F --&gt; G[熄灭绿灯]           </pre>
实现过程	① 将LED与Arduino控制板连接起来，并记录LED连接的引脚 ② 编写程序，控制LED的亮灭 ③ 上传程序，测试效果



LED按发光颜色来区分有单色LED和全彩LED（又称变色LED），常见的单色LED有红、绿、黄、蓝、白五种，全彩LED则可以动态发出不同颜色的光。

## ● LED的功能与结构

LED是一种固态半导体器件，可以实现电信号到光信号的转换。LED具有单向导电性，一般来说，长脚为正极，短脚为负极，在使用时不能接反。LED可用于交通信号灯、电梯楼层显示屏、各种仪表盘和LED灯带中，如图2.1.1所示。

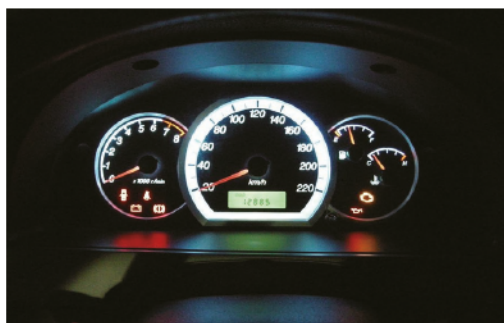
 LED允许通过的电流较小，使用时要在电路中连接保护电阻，避免因电流过大而烧毁。有些产品把LED和保护电阻集成在一起，使用较方便。以图2.1.2为例，左图和中图的LED在使用时需要连接保护电阻，而右图的食人鱼LED为集成模块，不需要另加保护电阻，本节采用的LED即为此类。食人鱼LED模块将四脚LED集成在电路板上，而该模块与控制板连接时用的是3P线。



交通信号灯



电梯楼层显示屏




汽车仪表盘



LED灯带

图2.1.1 LED的几种用途

LED可以分为不同的类别，按引脚数分，可以分为两脚LED、三脚LED、四脚LED等。一般来说，两脚LED主要是单色LED，只能发出某一种颜色的光；三脚LED主要是双色LED；四脚LED主要是三色LED或食人鱼LED。图2.1.2中是几种常见的LED，本节采用的是最右面的食人鱼LED模块。

 在电子电路中，高电平是电压高的状态，一般记为1；低电平是电压低的状态，一般记为0。

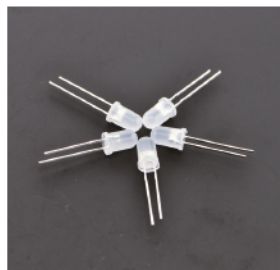



图2.1.2 常见的LED

 Arduino 控制板上的模拟引脚也可以作为数字引脚使用，编号为14（对应模拟引脚A0）到19（对应模拟引脚A5），但A6、A7仅可当作模拟输入引脚使用。

## ● LED与数字输出

在1.2节中，我们已经了解了输入过程与输出过程。输入或输出的信息可能是0（低电平）或1（高电平）的数字量，也可能是在一定范围内连续变化的模拟量（如温度、压力等）。数字输出就是控制板给外部数字设备输出低电平或高电平，相对控制板而言，这种外部数字设备称为数字输出设备。

LED能够将电信号转换为光信号，是一种输出设备。LED既可作为数字输出设备使用，也可作为模拟输出设备使用。LED作为数字

输出设备使用时，向其输出 1（高电平）表示点亮 LED，向其输出 0（低电平）表示熄灭 LED；当 LED 作为模拟输出设备使用时，可以输出不同的亮度值。

## ● Arduino 数字引脚与 LED 的连接

本书所用 Arduino 控制板包含扩展板，其数字引脚共有 14 组，编号为 0~13，图 2.1.3 中红框内的两排绿色引脚均为数字引脚。需要注意的是，上面三排引脚是为了方便与各类元器件的连接，在下排数字引脚（UNO 标准板引脚）的基础上增设的，其中黑色引脚为 GND，红色引脚为 5V 电压。食人鱼 LED 模块有三根引脚，正好与 Arduino 控制板增加的三排引脚一一对应，只需使用 3P 线（P 是 Pin 的简称）按照对应的颜色把 LED 模块与 Arduino 控制板连接起来。



一般来说，我们将集成电路中用于引出与外围电路接线的接口称为引脚，引脚可划分为脚跟（bottom）、脚趾（toe）、脚侧（side）等部分。引脚也有公母之分，针状的为公头（针脚），而另一种孔状的则为母头。另外，与公母头对应的连接为公母线。图 2.1.3 中的所有接口均可称为引脚，上排形状与针类似的插头为公头（针脚），下排孔状的插头为母头。

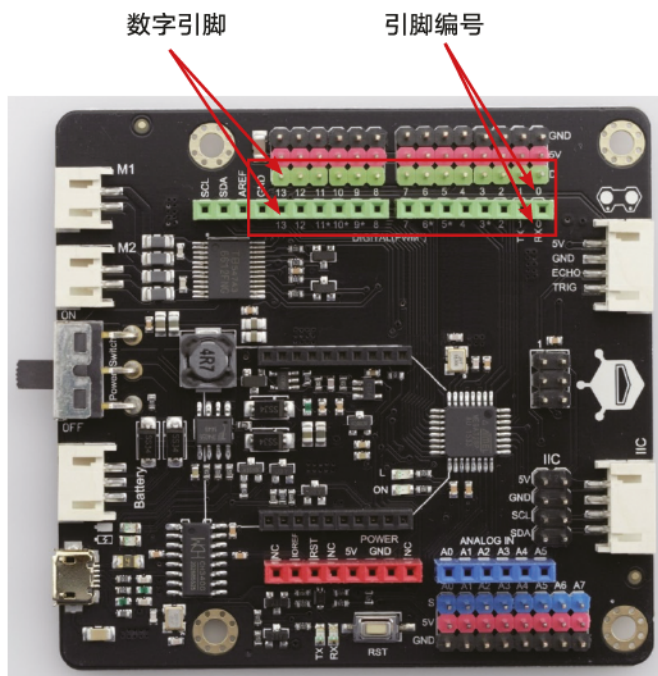


图 2.1.3 Arduino 控制板数字引脚



## 任务二 人行横道交通信号灯的实现

### ※ 活动 1 人行横道交通信号灯的硬件搭建

准备好器材后，请同学们参考图 2.1.4 将红色 LED 连接在 Arduino 控制板的数字引脚 5 上，将绿色 LED 连接在数字引脚 6 上。

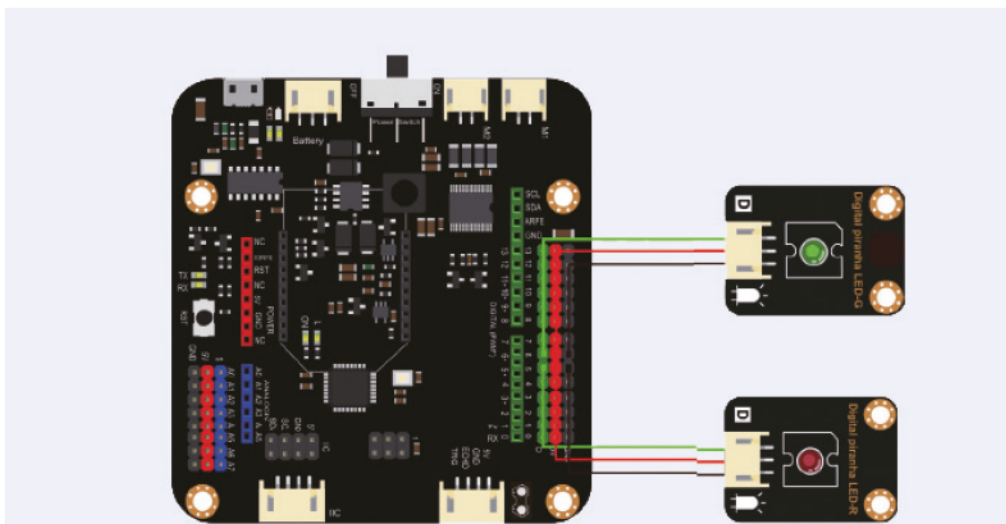


图 2.1.4 人行横道交通信号灯硬件连接图

### ※ 活动2 人行横道交通信号灯的程序编写

将人行横道交通信号灯的硬件连接好以后，接下来需要编写程序使LED按一定规律亮和灭。那么，在Arduino程序中如何控制LED的亮和灭呢？

在Arduino中，要控制LED的亮和灭，需要使用pinMode()函数和digitalWrite()函数。pinMode()函数用来设置引脚的模式（输入或输出），digitalWrite()函数用来设置数字引脚的电平状态（高电平或低电平）。

#### ● pinMode()函数

形式：pinMode(pin, mode)

参数：pin 表示所要设置的引脚；mode 表示设置的模式——INPUT（输入）或 OUTPUT（输出）。

含义：将一个引脚设置成输入或输出模式。

#### ● digitalWrite()函数

形式：digitalWrite(pin, value)

参数：pin 表示所要设置的引脚；value 表示输出的电压，HIGH表示高电平，LOW表示低电平。

含义：将一个引脚输出的电平设置成高电平或低电平，类似开关的作用。

以下为参考程序，请同学们按照提示完善程序并上传到控制板中，验证程序是否达到了预期的效果。

```
void setup()
{
  pinMode(5,OUTPUT);           //设置引脚5为输出模式
  pinMode(6,OUTPUT);           //设置引脚6为输出模式
}
void loop()
{
  digitalWrite(5, HIGH);       //点亮红灯
  digitalWrite(5, LOW);        //熄灭红灯
  //点亮绿灯
  //熄灭绿灯
}
```



编写程序时，需注意代码书写的规范性。例如：在函数等代码后加上分号，注意代码的大小写，在重要代码的后面用“//”注释代码的功能，除注释内容外所有代码和符号均为英文字符。



### 任务三 人行横道交通信号灯的测试与优化

#### ※ 活动1 使用延时函数

程序上传后，我们会发现两个LED一直保持亮的状态，没有实现预期的有规律亮灭的效果。怎样才能解决这个问题呢？此时需要使用延时函数来控制灯的亮灭时长。

#### ● 延时函数

LED亮灭的时间控制，比如亮5秒、灭1秒，实际上是让LED保持亮的状态一段时间，再保持灭的状态一段时间。上述示例是在两种状态之间快速切换，未设置亮或灭状态的保持时间，导致人眼无法直接观察到亮和灭的变化过程，因此需要采用延时函数来改进。

**形式：**delay(ms)

**参数：**表示延时时长，单位是毫秒（ms）。

**含义：**当程序执行到延时模块时保持当前状态相应的时间，随后继续运行该模块后面的程序。

以延长红灯亮的时间为例，其loop()函数如下所示。

```
void loop()
```



Arduino 中的另一个延时函数是delayMicroseconds( $\mu$ s)，其功能与delay(ms)函数一样，只是单位为微秒( $\mu$ s)。

```

{
    digitalWrite(5, HIGH);
    delay(5000);                //红灯亮5秒
    digitalWrite(5, LOW);
    delay(2000);                //红灯灭2秒
}

```



重复的代码可以用复制、粘贴的方式快速编写，这也是软件复用思想的一种体现。

请同学们修改程序，使交通信号灯能够交替亮灭。

### ※ 活动2 进一步测试与修改人行横道交通信号灯

根据自己的理解和需求，添加或修改既有人行横道交通信号灯的设计方案，并设计制作合适的外形，对原型做一定的包装，使作品更加美观、实用。



## 作品发布与评价

请参考表2.1.4开展作品的自评、互评等，与同学分享自己的制作经验。

表2.1.4 作品评价表

评价指标	得分
1. 交通信号灯能够实现红绿灯的切换（3分）	
2. 交通信号灯的红绿灯变化时间有规律，运行稳定（5分）	
3. 交通信号灯的包装美观实用（2分）	
总分（10分）	



## 拓展练习

请同学们仔细观察身边的环境，看看有哪些装置是可以利用本章节所学的LED数字输出和延时函数的知识，结合相关硬件来模拟的。可以参考以下装置。

1. 机动车交通信号灯：使用LED模拟机动车交通信号灯的效果，或用多个LED组成“箭头”形状，模拟方向指示灯。
2. 流水灯：使用多个LED组成特定造型，通过延时函数控制每个灯亮灭的时间，实现流水灯的效果。
3. 利用LED发射求救信号：国际上有通用的SOS求救信号，可以用LED亮灭的效果模拟SOS信号。

当然，同学们也可以选择设计其他项目主题。请分小组讨论并确定主题，形成具体设计方案后填写附录1所示的项目设计方案，再通过小组合作尝试实现。



SOS是国际莫尔斯电码求救信号。它的莫尔斯电码为：...---...（三个圆点加三个破折号，再加三个圆点）。在电报中很容易发出和识别，因此被确定为国际通用海难求救信号。用LED模拟SOS信号，可以从灯亮的时间长短来考虑，例如红色LED短时间内闪烁三次，绿色LED长时间内闪烁三次，再是红色LED短时间内闪烁三次。



## 2.2 控制蜂鸣器

常见的智能电子装置不仅需要灯光效果，有时还需要借助声音来实现一些特殊功能。蜂鸣器就是一种常见的声音播放设备，可用作打印机、复印机、报警器、电子玩具、电话机、定时器等电子产品的发声器件。

### 学习目标

- ★ 掌握蜂鸣器的使用方法，理解模拟输出的含义与作用。
- ★ 能够利用tone() 函数实现声音信号的模拟输出。
- ★ 能够正确使用变量和for 语句模拟声音的渐变效果，从而理解程序的循环执行思想。
- ★ 在实现声音渐变效果的过程中，体会迭代、测试与调试思想。

警车上的警灯不断闪烁，同时还伴有响亮的警报声，以便提醒路人避让。我们也可以利用上节课的知识做出警灯闪烁的效果。可是，警报声音效果该怎么实现呢？

要模拟警报，需要考虑使用什么硬件来发出声音，以及如何发出并改变声音。要解决这些问题，需要完成以下三个任务：警报项目规划、警报的实现、警报的测试与优化。

### 任务一 警报项目规划

#### ※ 活动1 警报项目调研

在日常生活中使用警报的场合和设备有很多，例如警车、救护车、防盗报警器等。那么，不同警报的声音效果有何不同呢？请同

学们利用网络查找常见的警报，了解它们各自的发声方式，并完成表2.2.1。

表2.2.1 常见的警报调查表

序号	常见的警报	功能	发声方式
1			
2			
3			
4			

通过调研不难发现，有些警报的声音始终为一个或某几个频率，效果有些单调。因此，我们可以制作一个警报装置，实现声音由弱到强、再由强到弱的渐变效果，甚至做出更有创意的警报声音。

警报的声音效果主要与实现警报的硬件设备和发声方式有关，其中模拟警报的发声方式是关键。

### ※ 活动2 警报的可选方案分析与设计

结合活动1的调研，我们可以从设备选择、发声方式两个角度来模拟警报，初步形成几种方案，如表2.2.2所示。

表2.2.2 警报的可选方案

类别	方案	方案描述
设备选择	方案一	使用蜂鸣器模拟警报效果
	方案二	使用带功放的蜂鸣器模拟警报效果
发声方式	方案三	使声音在某几个频率下播放
	方案四	使声音在某段频率内由弱到强变化，然后以某个频率播放一段时间，最后在某段频率内由强到弱变化



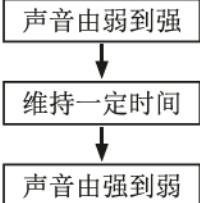
发声方式指警报声具有的规律，如只有一个频率、两个频率交替等。

上述方案各有利弊：从音效的角度看，带功放的蜂鸣器比蜂鸣器的效果更逼真，可以选择方案二；从引人注意的角度看，方案四比方案三的效果好。因此，基于统筹与折中的工程设计思想，在综

合考虑音效、实际效果等因素后，我们可以以方案二和方案四为基础设计最终方案：使用带功放的蜂鸣器模拟音调渐变的警报效果，并以模拟110警报声为目标。

要实现以上方案，关键在于控制声音由强到弱及由弱到强的渐变，如表2.2.3所示的设计方案。

表2.2.3 警报的设计方案

作品名称	警报
作品功能	模拟现实中警报的声音效果
器材清单	Arduino控制板、USB数据线、带功放的蜂鸣器
运行流程	 <pre> graph TD     A[声音由弱到强] --&gt; B[维持一定时间]     B --&gt; C[声音由强到弱]           </pre>
实现过程	<ol style="list-style-type: none"> <li>① 连接带功放的蜂鸣器模块与Arduino控制板</li> <li>② 编写程序，实现声音渐变效果</li> <li>③ 上传程序，测试效果</li> <li>④ 设计警报装置外形</li> </ol>



**方波**：占空比为50%的矩形波称为方波，即高/低电平持续的时间相同，改变方波的频率即改变高/低电平交换的速率，从而改变蜂鸣器发声的频率。

## ● 蜂鸣器

蜂鸣器分为有源蜂鸣器和无源蜂鸣器，这里的“源”指振荡源。有源蜂鸣器中含有振荡源，通电即可发声，但只能发出一种固定频率的声音；无源蜂鸣器不含振荡源，需要使用方波来驱动，可以发出不同频率的声音。要实现声音的变化效果，可以使用基于无源蜂鸣器开发的带功放（功率放大，即扩音）功能的蜂鸣器模块（也称“功放喇叭”）。该模块有三个引脚，除VCC（电源）和GND（接地）外，Signal连接数字引脚，如图2.2.1所示。

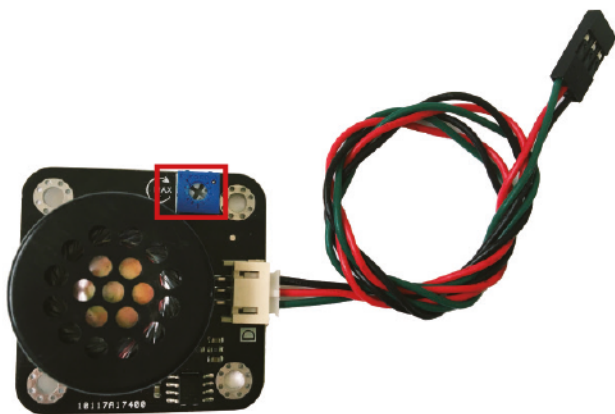



图 2.2.1 带功放的蜂鸣器模块

 带功放的蜂鸣器模块中的电位器（如图 2.2.1 中的红色方框所示）可以调节音量大小，实际使用时可能出现音量调到了最小导致无法听到声音而误认为模块损坏的情况。



## 任务二 警报的实现

### ※ 活动 1 警报的硬件搭建

准备好器材后，请同学们参考图 2.2.2 将带功放的蜂鸣器连接在 Arduino 控制板的数字引脚 3 上。

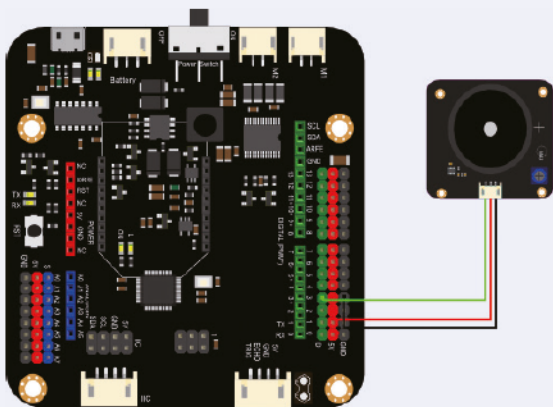


图 2.2.2 硬件连接图

### ※ 活动 2 警报的程序编写


连接好硬件之后，怎样才能让带功放的蜂鸣器发出我们需要的声音效果呢？此时需要使用模拟输出方式和 `tone()` 函数。

## ● 模拟量与模拟输出

警报的声音是在一定范围内变化的，即不只有 0 和 1 两种状态。我们把这种在一定范围内连续变化的物理量称为模拟量，如温度、压力等。把从控制板向外部设备输出模拟量的输出方式称为模拟输出。

## ● `tone()` 函数

声音频率的改变需要利用 `tone()` 函数来实现。

 以控制板为参照，能够向控制板输入模拟量的外部设备称为模拟输入设备；能够接收控制板输出的模拟量的外部设备称为模拟输出设备。

**形式:** tone(pin, frequency)

**参数:** pin 指要产生声音的引脚, 该引脚可以连接带功放的蜂鸣器来播放声音; frequency 指声音的频率, 单位 Hz。

**含义:** 在一个引脚上产生一个特定频率的方波 (占空比为 50%)。

在同一时刻, 一个引脚只能产生一个声音, 如果该引脚已经在播放声音, 调用 tone() 函数将不会有任何效果, 因此要使声音在两个不同的频率之间变化, 需要设置延时。

请同学们根据上述知识, 编写声音频率由低到高的控制程序。



### 任务三 警报的测试与优化

#### ※ 活动1 实现声音渐变效果

经过测试可以发现, 根据上述步骤完成的程序难以达到预期的声音渐变效果, 原因是相邻两个频率间隔较大。同时, 使用顺序结构需要重复编写多条语句, 不利于读写和修改, 需要采用变量和循环结构来优化。

#### ● 变量

使用变量之前要先声明, 其语法为:

数据类型 变量名 = 变量的初始值

变量名可以由字母、数字和下划线组成, 中间不能包含空格, 且必须以字母或下划线开始。数据类型指明了数据的长度和处理方式, 本项目中用到的数据类型为 int, 即整型变量, 用来存放整数。

变量可以先定义, 再初始化, 例如:

```
int a;
```

```
a=999;
```

也可以在定义的同时进行初始化, 例如:

```
int a=999;
```

#### ● for 语句

要实现声音的渐变效果, 声音频率应逐步增加 (如每次增加 1Hz), 可用循环结构实现。当循环次数确定时, 可以利用 for 语句来实现。

**形式:**

```
for (循环变量赋初值;条件;计数)
```

```
{
```

```
    循环体;
```

```
}
```



**注意:** 变量名不能为 Arduino 中的保留字, 如 HIGH、LOW、TRUE、FALSE 等。

除了整型之外, Arduino 语言还支持无符号的整型、布尔型、字符型、无符号的字符型、字节型、长整型、无符号的长整型、浮点型、双精度型、字符串型和数组型。

**参数：**“循环变量赋初值”用来设置循环变量的初值，只在循环开始时执行一次；


“条件”指循环的执行条件，决定什么时候退出循环；

“计数”控制变量每循环一次后按什么方式变化。

如果条件为真，则执行循环体和“计数”；当条件为假时，结束循环。

依据上述变量和 for 语句的使用方法，可以编写使声音频率由低到高改变的程序，如下所示。请同学们把程序补充完整并测试效果。

```
int i;                //声明变量 i 的数据类型为 int
void loop()
{
  for(i=650;i<=1450;i++) //用循环的方式将频率从 650Hz 增加到 1450Hz
  {
    tone(3,i);
    delay(0.3);
  }
  //补充将频率从 1450Hz 降低到 650Hz 的程序
}
```

 i++ 即 i=i+1；  
i-- 即 i=i-1。

### ※ 活动2 进一步测试与修改警报

根据自己的理解和需求，添加或修改既有的警报设计方案，形成有一定个人特色的作品。



## 作品发布与评价

参考表 2.2.4 开展作品质量的自评、互评等，与同学分享自己的制作经验。

表 2.2.4 作品评价表

评价指标	得分
1. 能够控制蜂鸣器实现声音频率的改变（2分）	
2. 警报有流畅的渐变效果（4分）	
3. 能够真实模拟 110 警报效果（4分）	
总分（10分）	



LED 的亮度调节需要使用 `analogWrite (pin, mode)` 函数, 其中引脚可使用 3、5、6、9、10、11, 值为 0~255。



全彩 LED 模块是由红、绿、蓝三个单色 LED 组合而成的, 该模块的三种颜色信号的输出为 0~255, 可以通过模拟输出控制三种颜色信号的变化得到多种灯光色彩, 也可以通过高低电平控制三个单色 LED 的亮灭。



## 拓展练习

蜂鸣器在生活中常用作报警器, 为了醒目, 还会结合灯的使用。请同学们尝试结合有关蜂鸣器和前面所学的 LED 灯的知识, 设计一个具有创意的作品。以下项目主题供参考。

1. 防空警报: 使用蜂鸣器实现防空警报效果。
2. 呼吸灯: 使用 LED 实现亮度由灭到亮, 再由亮到灭的呼吸效果。
3. 交通信号灯升级版: 利用全彩 LED 实现不同颜色信号之间的切换。

当然, 同学们也可以选择其他的项目主题。请进行小组合作, 讨论确定主题, 形成具体设计方案后填写附录 1 所示的项目设计方案, 并通过小组合作尝试实现。

## 2.3 按钮开关

机器人除了需要 LED、蜂鸣器等输出设备，还需要能够检测外界环境信息的输入设备。按钮就是一种常见的数字传感器（数字输入设备），它可以用来接收开关信号。



### 学习目标

- ★ 掌握按钮的使用，理解数字输入的含义与作用，掌握随机函数的使用。
- ★ 能够利用按钮控制蜂鸣器。
- ★ 在利用选择结构控制系统执行的过程中，体会程序设计中的分类思想。

阿丢放学后回家，来到家门前按下门铃，妈妈笑着给他开了门。进入家门后，阿丢的妈妈说道：“每次听到门铃声我都会吓了一跳，如果能让我自己设计门铃的声音就好了！”受到妈妈的启发，阿丢决定制作一款智能门铃，你能帮他实现吗？

要制作门铃系统，需要考虑使用哪种触发设备和声音播放元件，以及如何设置铃声并控制其播放。要解决这些问题，需要完成以下三个任务：门铃系统项目规划、门铃系统的实现、门铃系统的测试与优化。



### 任务一 门铃系统项目规划

#### ※ 活动1 门铃系统项目调研

门铃有很多类型，为进一步了解其特点，请同学们开展调研，填写表 2.3.1。



在古代，拜访他人时可以通过直接敲击对方家门来告知其有客来访，而现在许多住宅都安装了门铃系统，访客只需按下按钮，室内就会响起铃声。



表2.3.1 常见的门铃调查表

序号	门铃的类型	工作机制	适用范围
1			
2			
3			
4			

我们要做的家用门铃其实就是一种结构简单的有线门铃系统，不同的是，普通家用门铃的铃声都是厂家预先设定好的，而这里需要我们来设计铃声，解决阿丢妈妈的困扰。

门铃系统的实现主要与门铃的硬件设备和铃声播放方式有关，其中铃声播放方式是关键。

### ※ 活动2 门铃系统的可选方案分析与设计

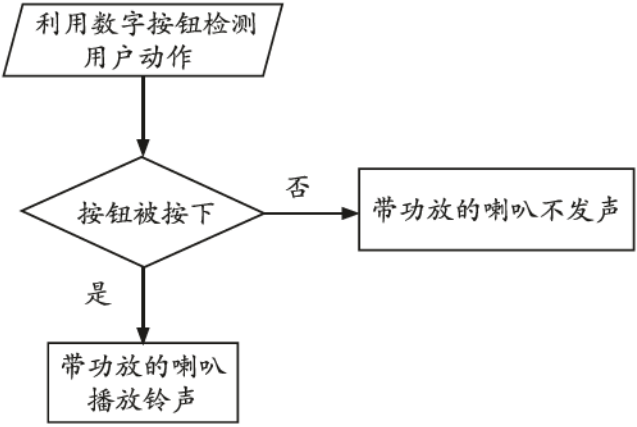
根据上述思路，我们可以从触发方式、声音播放元件和铃声播放方式三个角度初步设计出几种门铃方案，如表2.3.2所示。

表2.3.2 门铃系统的可选方案

类别	方案	方案描述
触发方式	方案一	使用按钮作为门铃的触发装置
	方案二	使用数字触摸开关作为门铃的触发装置
声音播放元件	方案三	使用普通蜂鸣器播放铃声
	方案四	使用带功放的蜂鸣器播放铃声
铃声播放方式	方案五	为声音播放装置设置固定的铃声
	方案六	为声音播放装置设置不同的铃声，触发门铃后随机播放铃声

每个方案都各有利弊：从实际应用的角度看，数字触摸开关在安装时需要离固定面有一定高度，否则可能产生信号误差；从音效的角度看，带功放的蜂鸣器比普通蜂鸣器声音大，能够较好地起到提醒作用；从趣味性的角度看，方案六比方案五更有趣。在综合考虑实际应用、音效和趣味性后，我们可以以方案一、方案四和方案六为基础形成最终实施方案：使用按钮触发门铃，使用带功放的蜂鸣器随机地播放铃声，如表2.3.3所示。

表2.3.3 门铃系统的设计方案

作品名称	创意门铃
作品功能	按钮被按下后，门铃播放提示铃声
器材清单	Arduino 控制板、USB 数据线、按钮、蜂鸣器
运行流程	 <pre> graph TD     A[/利用数字按钮检测 用户动作/] --&gt; B{按钮被按下}     B -- 否 --&gt; C[带功放的喇叭不发声]     B -- 是 --&gt; D[带功放的喇叭 播放铃声]   </pre>
实现过程	<ol style="list-style-type: none"> <li>① 按钮和蜂鸣器与 Arduino 控制板连接</li> <li>② 编写程序，实现按钮对蜂鸣器的控制</li> <li>③ 上传程序，测试效果</li> <li>④ 包装门铃外形</li> </ol>


## ● 按钮与数字输入

不同于输出设备 LED，按钮能够将信号传递到控制板中，属于数字传感器。数字传感器是能够向控制板输入数字量的设备。控制板读取数字传感器接收到的数字量称为数字输入，在 Arduino 系统中就是读取数字传感器的电平状态（低电平或高电平）。

当按钮处于静止状态时，其内部电路断开，输入低电平；当按钮被按下时，形成通路，输入高电平。如图 2.3.1 所示，当按钮被按下时，模块上的 LED 提示灯会点亮。



图 2.3.1 按钮模块

 有些按钮与教科书中用到的相反，即当按钮被按下时，其内部电路断开，输入低电平；当按钮处于静止状态时，形成通路，输入高电平。

按钮模块上有大写字母 D (Digital 的缩写) 则表示其为一款数字传感器。与之对应，大写字母 A (Analog 的缩写) 则表示该传感器为模拟传感器。



## 任务二 门铃系统的实现

### ※ 活动1 门铃系统的硬件搭建

准备好器材后，请同学们参考图 2.3.2 将按钮连接在 Arduino 控制板的数字引脚 2 上，将蜂鸣器连接在数字引脚 8 上。

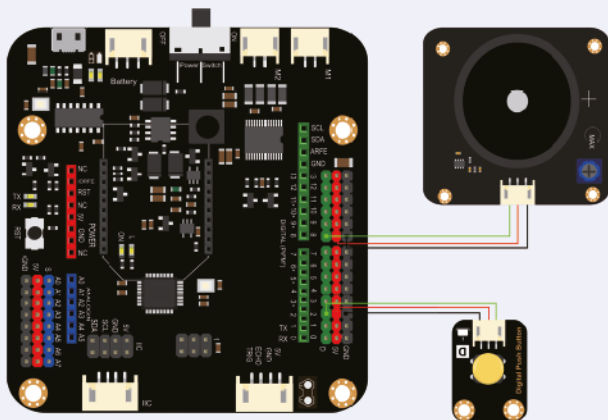


图 2.3.2 硬件连接图

### ※ 活动2 门铃系统的程序编写

连接好硬件后，如何才能知道按钮有没有被按下呢？这与点亮 LED 的方法有什么区别吗？通过前面对按钮的介绍，我们知道需采用数字输入模式读取按钮的值，并通过判断按钮的输入值来决定是否触发门铃发声。

#### ● 按钮值的读取与判断

如何才能获取按钮的状态？可以使用 `digitalRead()` 函数读取按钮的值（状态），其中参数 `pin` 为按钮连接的数字引脚。读取按钮的值实际上就是读取连接按钮的数字引脚的值。

#### ● `digitalRead()` 函数

形式：`digitalRead(pin)`

参数：`pin` 表示连接数字传感器的引脚。

含义：读取某一数字引脚的输入值。

由于按钮只有高电平、低电平两种状态，因此可以使用选择结构判断按钮的值，以执行相应的操作。选择结构即通过判断给定条件是否成立来控制程序的执行流程，由条件语句和执行语句两部分组成。在 Arduino 中，其语法如下。

形式:

```
if (条件语句1)
{
    执行语句1;
}
else if (条件语句2)
{
    执行语句2;
}
else if (条件语句3)
{
    执行语句3;
}
...
else
{
    执行语句n;
}
```

**参数:** 条件语句即给定的条件; 执行语句即当条件成立时所执行的语句。

**含义:** 通过判断给定条件来控制程序的执行流程。

请同学们根据上述知识, 编写门铃系统的控制程序, 实现以下控制方式: 当按下按钮后, 蜂鸣器持续播放声音一段时间。其实, 除了这种控制方式, 还可以采用即时响铃的方式, 即按下按钮时蜂鸣器播放声音, 松开按钮后停止播放声音。同学们能尝试实现这种响铃方式吗?



### 任务三 门铃系统的测试与优化

#### ※ 活动1 使用随机函数实现铃声的随机播放

完成上述任务后, 按钮已经能够控制蜂鸣器播放声音, 但是, 如何使门铃更有趣呢? 根据之前的设想, 可以设置多种铃声, 每次有人按门铃时随机选择其中一种铃声播放, 这样门铃是不是更有趣了? 那么如何才能做到“随机”呢?



只需更改选择结构中的执行语句便可实现该种控制方式, 即将延时时间更改为 100ms。由于时间缩短了, 因此需要一直按住按钮来播放铃声。



**random()函数**生成的数列并非真正意义上的随机数。实际上，它们是通过固定且可重复的算法生成的。该数列有很长的周期性，但其具有类似于随机数的统计特性，我们称之为伪随机数。未连接任何电子元件的模拟引脚输入值是真随机数列，randomSeed(seed)以其作为参数初始化随机函数，可使随机函数生成真随机数列。

## ● 随机函数

**形式1: random(max)**

**参数:** max 为生成随机数的最大值，但不包含在随机数中。

**含义:** 随机生成从0到(max-1)的数值。

**形式2: random(min, max)**

**参数:** min 为生成随机数的最小值，包含在随机数中；max 为生成随机数的最大值，但不包含在随机数中。

**含义:** 随机生成从min到(max-1)的数值。

**设置随机数种子: randomSeed(seed)**

**参数:** seed 为随机数“种子”；通常情况下，我们将seed 设置为一个变化的量，比如使用未连接任何电子元件的模拟引脚数值。

**含义:** 用于初始化随机函数，使其生成随机序列。



**使用纸盒模拟房门**，在正面和反面各挖一个方形口。把按钮固定在正面，喇叭固定在反面，通过方形口将线接到纸盒内的控制板上。示例只是简单的包装，同学们可以按照自己的想法，将门铃包装得更加美观。

请同学们根据上述随机函数的使用方法，编写随机播放铃声的程序。

### ※ 活动2 包装铃声系统

在应用该门铃系统前，还需要对其做必要的外形包装。可用胶带或尼龙扎带将按钮固定在纸板上，并显示门铃提示信息，将蜂鸣器安装在门内，如图2.3.3所示。请同学们自己动手包装门铃系统，使其既实用又美观。



图2.3.3 外形包装

### ※ 活动3 进一步测试与修改门铃系统

根据自己的理解和需求，添加或修改门铃设计功能，完成具有个人特色的作品。例如，铃声可能过大或过小，同学们可以通过修改声音播放频率与延时时长来调整铃声播放效果。



## 作品发布与评价

参考表2.3.4开展作品质量的自评、互评等，与同学分享自己的制作经验。

表2.3.4 作品评价表

评价指标	得分
1. 能够使用按钮触发门铃（3分）	
2. 门铃能够随机播放铃声（3分）	
3. 门铃铃声设计优美（2分）	
4. 门铃的包装实用、美观（2分）	
总分（10分）	



## 拓展练习

请同学们仔细想想，按钮还有哪些用处？结合前面学过的有关蜂鸣器和LED灯的知识，可以设计哪些具有创意的作品呢？可以参考以下项目主题。

1. 模拟弹奏乐器：使用多个按钮，每个按钮对应一个特定音符，并使用蜂鸣器播放相应的声音（音符、音调与频率的对应表见附录2），以模仿弹奏乐器的效果。

2. 随机抽奖彩灯：准备一个底盘并将其分区，在每个区域内安装一个LED并标明该区域的奖项。程序启动后，LED均未被点亮，当按下按钮后，随机点亮某一区域的LED，表示该区域被选中。

请进行小组合作，讨论确定主题。除已经给出的方案之外，还可以选择其他设计主题，形成具体设计方案后填写附录1所示的项目设计方案，并尝试实现。

## 2.4 声光检测

生活中有很多电器设备是利用光线或声音进行控制的，例如光控灯、声控风扇等。常见的光线检测和声音检测设备有光线传感器和声音传感器，它们能将获取的光线或音频信息转换为 Arduino 控制板可以处理的数字信号。



### 学习目标

- ★ 掌握光线传感器和声音传感器的使用方法，理解模拟输入的含义与作用。
- ★ 能够编程控制光线传感器和声音传感器实现声光检测功能。
- ★ 能够正确使用嵌套选择结构。
- ★ 能够利用串口输出方法测试与调试程序。

很多楼道灯都可以用声音或光线控制亮灭。同学们可以想一想，能不能利用所学的知识自己制作一个智能楼道灯呢？

要制作一个这样的楼道灯，有以下问题需要思考：如何自动“得知”何时需要亮灯？亮灯时间为多久？使用何种传感器？控制策略如何？要解决这些问题，需要完成以下三个任务：楼道灯项目规划、楼道灯的实现、楼道灯的测试与优化。



## 任务一 楼道灯项目规划

### ※ 活动1 楼道灯项目调研

其实，生活中有不同类型的楼道灯。请同学们观察生活中的楼道灯或利用网络资源，进一步了解楼道灯的控制方式，发现它们的优势与不足，并完成表2.4.1。

表2.4.1 楼道灯调查表

序号	楼道地点	楼道灯亮灭的控制方式	优势与不足
1			
2			
3			
4			

通过调研我们可能发现，有一些楼道灯的设计不是很完善。例如，一些楼道灯在夜晚保持常亮，无人经过也不熄灭；一些楼道灯在白天只要有声音发出也会亮，造成了能源的浪费。因此，我们可以制作一个智能楼道灯的原型产品，使它能够根据外界环境的变化合理地自动亮灭。

从控制楼道灯亮和灭的实际需求来看，主要需要解决环境声音、亮度和人的检测问题。

### ※ 活动2 楼道灯的可选方案分析与设计

根据上面的分析，可以从单一检测和组合检测两种方式出发设计楼道灯，初步形成几种方案，如表2.4.2所示。

表2.4.2 楼道灯的可选方案

类别	方案	方案描述
单一检测	方案一	利用声音传感器检测环境声音，在某个音量范围内，楼道灯亮，否则熄灭
	方案二	利用光线传感器检测光线亮度，在比较昏暗的光线范围内，楼道灯亮，否则熄灭
	方案三	利用热释电红外传感器检测是否有人经过，当有人时，楼道灯亮，否则熄灭



热释电红外传感器能够过滤自然界中的白光信号，只有特定波长如人体中辐射出的红外信号才能照射进传感器内部的敏感元件上，从而激发出电子，形成电流。



续表

类别	方案	方案描述
组合检测	方案四	为避免白天因环境噪声而误亮灯的问题,利用光线传感器检测光线亮度,在比较昏暗的光线范围内,若同时利用声音传感器检测到人的声音也在一定范围内,楼道灯亮,否则熄灭
	方案五	为避免白天因环境噪声而误亮灯的问题,利用光线传感器检测光线亮度,在比较昏暗的光线范围内,若同时利用热释电红外传感器也能检测到人的声音也在一定范围内,楼道灯亮,否则熄灭
	方案六	为避免白天因环境噪声而误亮灯的问题,利用光线传感器检测光线亮度,在比较昏暗的光线范围内,若同时利用声音传感器检测到人的声音也在一定范围内,并且利用热释电红外传感器也能检测到人的声音也在一定范围内,楼道灯亮,否则熄灭

每个方案各有利弊:从学习的难度来看,多个传感器的使用涉及多重条件判断,难度较大,用一个传感器实现的效果较单一,因此方案一、二、三难度适中;从实用性的角度来看,两个传感器的结合使用能减少不必要的亮灯;从成本角度来看,光线传感器的成本低于另外两种传感器。因此,在综合考虑学习难度、实用性和成本等因素后,可以选择方案四作为最终实施方案。

要实现方案四,关键在于环境光线亮度和声音信息的获取与组合控制。这涉及两个问题,一是环境光线亮度和声音的自动检测,二是环境光线亮度和声音大小与楼道灯亮灭的对应关系。请同学们根据上面的描述,将表2.4.3中的运行流程补充完整。

表2.4.3 楼道灯的设计方案

作品名称	楼道灯
作品功能	根据环境光线亮度和声音大小变化自动控制楼道灯的亮灭
器材清单	Arduino 控制板、USB 数据线、光线传感器、声音传感器、黄色LED
运行流程	
实现过程	<ol style="list-style-type: none"> <li>① 光线传感器、声音传感器、黄色LED与Arduino控制板连接</li> <li>② 编写程序,实现楼道灯自动亮和灭</li> <li>③ 上传程序,测试效果</li> </ol>

## ● 光线传感器

光线传感器利用光敏元件（光敏电阻或光敏二极管）对环境光线的强度进行检测，将光照强度（光照度）转换为电信号（输出电压）后输入 Arduino 控制板，经过模数转换后将模拟量的输出电压以二进制形式编码，供后继使用。

如图 2.4.1 所示，光线传感器有三个引脚，蓝色引脚用于输入信号，红色引脚为 VCC，黑色引脚为 GND。其主要参数如表 2.4.4 所示。

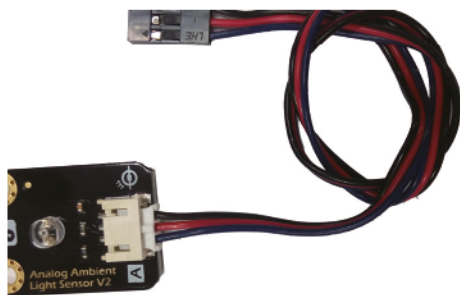


图 2.4.1 光线传感器

表 2.4.4 光线传感器的主要参数

使用电压	感应的流明范围	模拟输出电压	反应时间
3V~5V	1 lux~6000 lux	0V~5 V	15 $\mu$ s

## ● 声音传感器

声音传感器是另外一种常用的模拟传感器，能够将感知的外部声音转化为模拟电信号，即采用反馈的电压值表示声音的大小。如图 2.4.2 所示，声音传感器有三个引脚，蓝色

引脚用于输入信号，红色引脚为 VCC，黑色引脚为 GND。声音传感器带有灵敏度调节电位器，朝 Max 方向调节，则同样大小的声音输入到控制板的电信号会更大，相反则变小。其主要参数如表 2.4.5 所示。

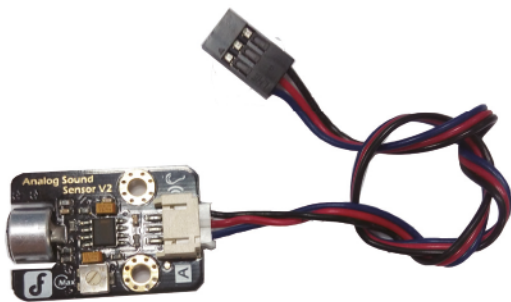


图 2.4.2 声音传感器

表 2.4.5 声音传感器的主要参数

工作电压	工作电流	模拟输出电压	最长响应时间
5V	<10mA	0V~5V	220ms



### 模拟传感器

能够将自然界的物理信号，例如声、光、温度等转化为电信号，并以模拟电压的方式输出的设备，叫模拟传感器。光线传感器、声音传感器和温度传感器等都是模拟传感器。如图 2.4.1 所示，模拟传感器上都有字母标识 A。



Arduino 控制板需要使用模数转换器将模拟量转化为它能处理的数据。Arduino 控制板中的模数转换器的精度为 10bit，也就是说能读取 2 的 10 次方（1024）个状态，因此输入值的范围为 0~1023。读取的数值对应的都是一个电压值，比如 512 对应 2.5V。

## ● Arduino 控制板上的模拟引脚

与数字引脚类似，模拟引脚也有两排，下面三排引脚是在UNO标准板上增加的，方便与各类元器件的连接。其中黑色引脚为GND，红色引脚为5V电压，蓝色引脚与前排引脚一样，也是模拟引脚。

Arduino 控制板上的模拟引脚共有 8 组，如图 2.4.3 所示，编号为 A0~A7，输入值的范围为 0~1023。

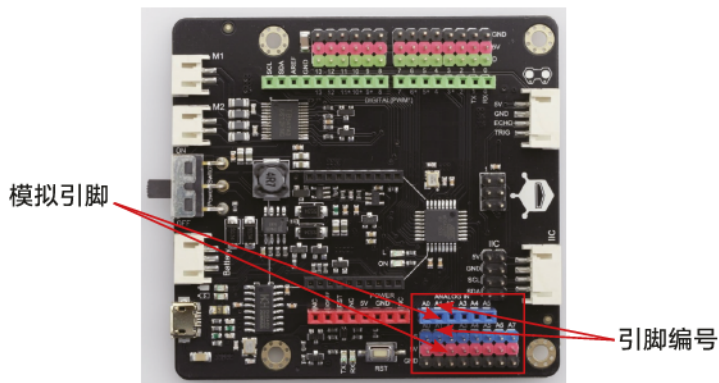


图 2.4.3 Arduino 控制板上的模拟引脚



## 任务二 楼道灯的实现

### ※ 活动 1 楼道灯的硬件搭建

准备好器材后，请同学们参考图 2.4.4 将光线传感器接在Arduino控制板的模拟引脚 A1 上，声音传感器接在模拟引脚 A2 上，黄色LED接在数字引脚 7 上。

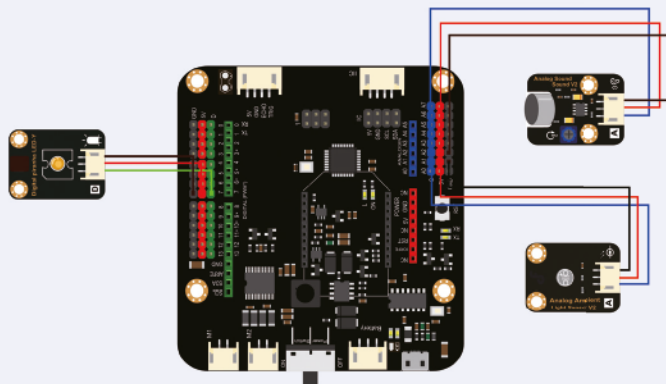


图 2.4.4 楼道灯硬件连接图

### ※ 活动 2 楼道灯的程序编写

楼道灯需要根据外部环境中的光线与声音变化来控制LED的亮和灭，那么究竟多大的光线亮度和音量可以作为触发LED亮灭的条件呢？为此，首先应该测试一下环境中的光线和声音对应的模拟量数值再做选择。那么，Arduino是怎样读取和显示传感器获取的模拟量数值的呢？

## ● 模拟传感器值的读取

获取模拟传感器输入的信息，需要使用 `analogRead()` 函数读取引脚的模拟量电压值。

形式：`analogRead(pin)`

参数：`pin` 为要获取模拟量电压值的引脚。

含义：读取模拟引脚的模拟电压值。

## ● 串口通信

要查看从模拟引脚读取的模拟值，就涉及串口通信的问题。串口通信（Serial Communication）是指传感器等外部设备和主控器之间进行数据传输的一种通信方式。这种通信方式使用的数据线少，可以节约通信成本，但其传输效率要比并行传输（同时传输多个二进制数据）低。

利用串口监视器可以观察传感器输入的模拟量数据。为此，首先需要初始化串口通信方式，然后用串口监视器显示串口通信数据。

### 串口通信的初始化

形式：`Serial.begin(speed)`

参数：`speed` 为串口通信速度，一般取值 300、1200、2400、4800、9600、14400、19200、28800、38400、57600 或 115200，Arduino 与计算机通信时，一般设置波特率为 9600。

含义：设置通信速度，打开串口。

### 在串口监视器中输出获取的通信数据


形式：`Serial.print(value)`


参数：`value` 为输出的值，可以是任意数据类型。


含义：不换行输出串口数据。

以光线传感器为例，用串口监视器查看其检测到的光线数据的程序如下所示。

```
void setup()
{
  Serial.begin(9600);           //打开串口
}
void loop()
{
  Serial.print(analogRead(1)); //输出光线传感器检测到的数据
```

 在 Arduino 编程环境中，串口监视器可在工具菜单中找到。

 波特率是指数据通信的速度，如果控制板与计算机之间的通信速度不一致，那么收到的信息就是乱码或收不到信息。

 **Serial.println 函数**  
形式：`Serial.println (value)`  
参数：`value` 为输出的值，可以是任意数据类型。  
含义：输出串口数据并换行。

```

Serial.println();           //换行
delay(200);                 //延迟0.2秒，方便观察显示的数据
}

```

参考上述串口输出方法，请同学们分组测试，了解光线传感器和声音传感器检测值的变化范围；然后在实际环境中多次测试，分别确定控制LED亮灭的光线亮度阈值和音量大小阈值。

接下来需要解决的问题是：如何组合两个传感器的阈值作为控制LED亮灭的最终条件？根据方案四，楼道灯需要先判断光线传感器的值，当光线值在一定范围内时（如在比较昏暗的条件下），再判断声音传感器的值，当两个条件的判断都为真，执行LED亮的语句。显然用单个if选择结构不能达到这样的目的，需要使用嵌套选择结构。

### ● 嵌套选择结构

嵌套选择结构是在一个选择结构内部再嵌入一个或多个选择结构，以解决更复杂的条件判断问题。

**形式：**

```

if ( 条件语句1)
{
    if ( 条件语句2)
    {
        执行语句1;
    }
    else
    {
        执行语句2;
    }
}
else
{
    执行语句3;
}

```

**参数：**条件语句即给定的条件；执行语句即判断条件后所执行的语句。

**含义：**通过判断给定的多个条件来控制程序的执行流程。

请同学们参考上述知识，编写当光线和声音同时满足一定条件时楼道灯亮的程序。



### 任务三 楼道灯的测试与优化

#### ※ 活动1 提高楼道灯的实用性

在上面的程序中，我们设置的是在环境光线亮度和音量大小达到一定阈值时LED亮，否则LED熄灭。在实际生活中，楼道灯亮起来时会保持一定时间再熄灭，以便于人们行走。请同学们以身边的楼道为例，测量通过楼道所需的时间，作为程序中控制楼道灯亮的延时参数。

#### ※ 活动2 进一步测试与修改楼道灯

根据自己的理解和需求，添加或修改既有楼道灯的设计方案，形成有一定个人特色的作品。例如，不同环境中，触发灯亮的光线亮度阈值和音量大小数值可能不同，同学们可以根据实际需求进行调整，测试能否达到预期效果。



当引脚作为输入时，可以不采用pinMode做显性定义。



### 作品发布与评价

参考表2.4.6开展作品质量的自评或互评，与同学分享自己的制作经验。

表2.4.6 作品评价表

评价指标	得分
1. 当光线或声音数值都满足一定条件时，触发楼道灯亮，否则熄灭（5分）	
2. 楼道灯保持亮的时间合适（3分）	
3. 优化后的楼道灯具有实际应用价值（2分）	
总分（10分）	



## 拓展练习

请同学们综合运用光线传感器、声音传感器和前面学过的知识，设计一个有趣的智能装置来装扮你的房间。可以参考以下项目主题。

1. 声控灯光秀：使用声音传感器和全彩 LED，使不同声音大小对应显示 LED 的不同颜色。

2. 吹奏乐器：参考本书附录 2，使用声音传感器和带功放的蜂鸣器，使不同声音大小对应相应的音符，然后用带功放的蜂鸣器播放声音，模拟吹奏乐器。

当然，同学们也可以选择其他的项目主题。请进行小组合作，讨论确定主题，形成具体设计方案后填写附录 1 所示的项目设计方案，并通过小组合作尝试实现。

## 2.5 数码管显示

机器人往往都具有显示设备，以便与使用者更好地进行交互。数码管是一种常见的显示设备，它具有结构简单、操作灵活等特点，可用于显示时间、日期、温度等数字信息。



### 学习目标

- ★ 了解数码管的显像原理并掌握其使用方法。
- ★ 能够利用振动传感器控制数码管。
- ★ 能够正确使用while循环结构，体会程序设计中的条件循环思想。
- ★ 能够正确定义与调用子程序，理解子程序的软件复用和结构化程序设计思想。

微信表情中的色子非常有趣，我们能否利用开源硬件也制作一个电子色子呢？要制作智能电子色子，要考虑两个问题：使用哪种显像设备和触发装置？如何呈现变化中的数字？要解决这些问题，需要完成以下三个任务：电子色子项目规划、电子色子的实现、电子色子的测试与优化。



### 任务一 电子色子项目规划

#### ※ 活动1 随机事件调研

电子色子本质上就是一个随机事件生成器。在我们的生活中会遇到各种各样的随机事件，在这些事件发生前，无法预知其结果，比如购买饮料获得“再来一瓶”的奖励、使用社交软件抢到的红包数额等。在我们身边还有哪些随机事件呢？请同学们思考并填写表2.5.1。



表2.5.1 随机事件调查表

序号	随机事件	能否测得概率	概率计算方法 (概率不可测时不填)
1			
2			
3			
4			

当然，最常见的随机事件生成器非“色子”莫属。色子一般由六面组成，每面分别刻有相应的点数，通过人工摇动来获得点数。事实上，任何一种随机事件生成器及其随机事件生成过程，都包含了“随机”“概率均等”“机会均等”理念，可以理解为建立在“机会均等”立场上的一种选择机制，也是一种合理的社会管理机制。

我们要做的电子色子，也打算模拟六面色子：当有人摇动电子色子时，显示设备上随机显示从1到6之间的数字。

电子色子的实现主要与触发装置、显像元件和数字显示方式有关。其中，如何控制显像元件显示数字是关键。

### ※ 活动2 电子色子的可选方案分析与设计

根据上述思路，可以从触发方式、显像元件和呈现方式三个角度设计几种制作电子色子的方案，如表2.5.2所示。

表2.5.2 电子色子的可选方案

类别	方案	方案描述
触发方式	方案一	使用振动传感器作为电子色子的触发装置
	方案二	使用数字钢球倾角传感器作为电子色子的触发装置
显像元件	方案三	使用数码管显示数字信息
	方案四	使用LCD显示器显示数字信息
呈现方式	方案五	启动程序后，显像元件不显示信息，电子色子被触发后直接显示数字信息
	方案六	启动程序后显像元件随机地显示一到六之间的数字，触发电子色子后停止随机显示并只显示最后一个数字信息



数字钢球倾角传感器是一种比较简单的倾角传感器，它利用钢球的特性，通过重力作用使钢球向低处滚动，从而使开关闭合或断开。

每个方案各有利弊：从实际操作的角度看，数字钢球倾角传感器过于敏感，稍微晃动就会改变状态；从显示效果的角度看，数码管显示数字的效果比 LCD 显示器清晰；从趣味性的角度看，在用户操作前动态地随机显示数字，能够将用户带入掷色子的氛围中。因此，在综合考虑实际操作、显示效果和趣味性等因素后，我们以方案一、方案三和方案六为基础形成最终实施方案：程序启动后，电子色子随机显示数字，晃动振动传感器后只显示最后一个数字。

要实现以上方案，关键在于当振动传感器检测到振动后如何中断数码管信息的随机显示。请同学们根据上面的描述，将表 2.5.3 中的运行流程补充完整。

表 2.5.3 电子色子的设计方案

作品名称	电子色子
作品功能	晃动振动传感器后，数码管停止随机显示并将最后的数字作为掷色子的结果
器材清单	Arduino 控制板、USB 数据线、振动传感器、数码管
运行流程	
实现过程	① 振动传感器和数码管与 Arduino 控制板连接 ② 编写程序，实现振动传感器对数码管的控制 ③ 上传程序，测试效果

## ● 数码管

数码管也称 LED 数码管，是由多个发光二极管封装在一起的“8”字形发光元件，常见的数码管如图 2.5.1 所示。

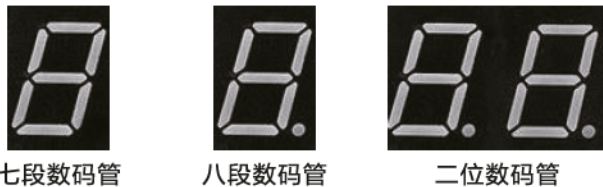



图 2.5.1 常见的数码管



按段数（LED 的个数）分类，数码管可分为七段数码管和八段数码管，八段数码管比七段数码管多一个小数点；按显示的位数分类，可分为一位、二位、三位等数码管；按 LED 的组织连接方式分类，可分为共阴极数码管和共阳极数码管。

 图2.5.2中的数码管是集成模块，模块右侧共有9个引脚，并分别注明了与控制板对应的引脚号。从上到下依次为：GND引脚，数字引脚13、12、11、10、9、8、7、6。

本节使用的是由8个LED组成的一位数码管，如图2.5.2所示。模块右侧标识有a、b、c、d、e、f、g、dp的引脚分别对应数码管上的a、b、c、d、e、f、g、dp段LED。

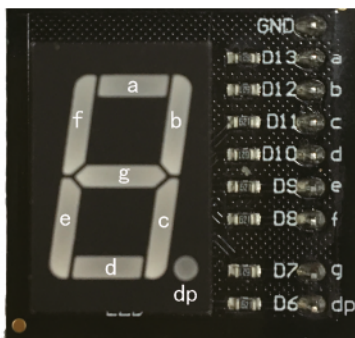


图2.5.2 数码管模块

LED的引脚有阴极和阳极之分，将多个LED封装为数码管时，需要将所有LED的阴极或阳极连接在一起形成公共极。共阴极数码管即将所有LED的阴极接在一起形成公共阴极，在使用时将公共阴极连接到引脚GND上。给LED的阳极输出高电平时，它就被点亮；输出低电平时则熄灭。共阳极数码管的工作原理与共阴极数码管的相反。

## ● 振动传感器

振动传感器是一种能够感知振动的数字传感器，如图2.5.3所示。它可通过振动来通断电路，当其振动时，内部电路会瞬间变为通路。振动传感器处于静止状态时给控制板输入高电平，感知到振动时给控制板输入低电平。

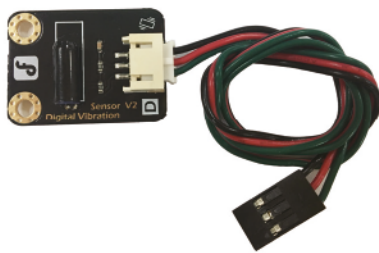


图2.5.3 数字振动传感器



## 任务二 电子色子的实现

### ※ 活动1 电子色子的硬件搭建

准备好器材后，请同学们参考图2.5.4将数码管连接在数字引脚6~13和GND上，将振动传感器连接在数字引脚3上。

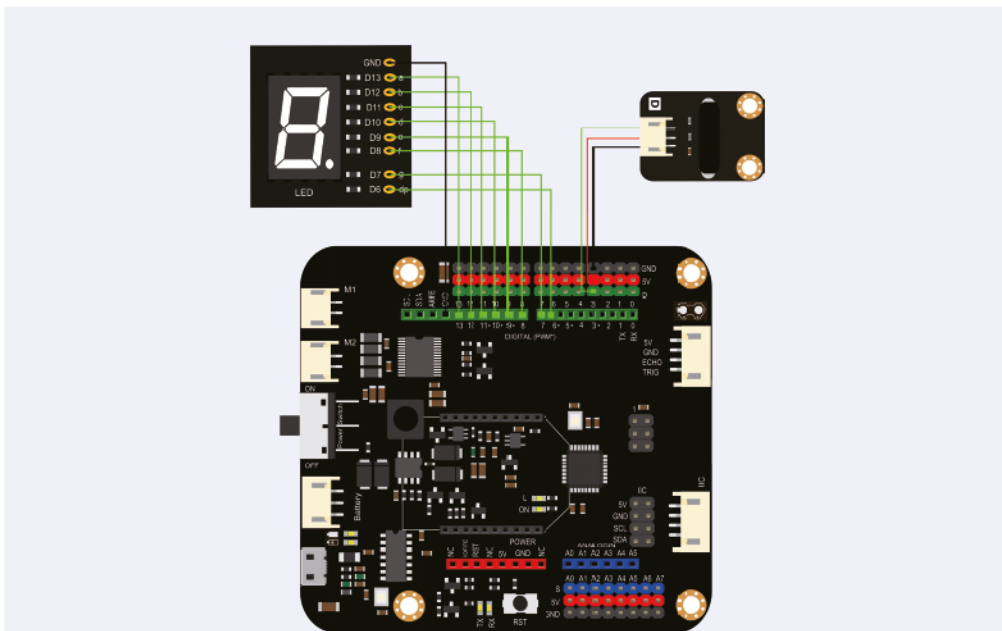


图 2.5.4 硬件连接图

## ※ 活动 2 电子色子的程序编写

根据之前学过的 LED 控制方法，需要利用循环结构来控制数码管显示随机数字。

### ● 控制数码管的显示

如何控制数码管使其显示数字呢？实际上，控制数码管也就是控制各段位 LED。以点亮 a 段 LED 为例，将数码管与 Arduino 控制板连接后，只需为数字引脚 13 输出高电平即可将 a 段 LED 点亮，输出低电平即可将其熄灭。同理，可通过控制各段位 LED 使数码管显示相应的数字。例如，使数码管显示数字 1，即点亮 b、c 段的 LED（为数字引脚 12、11 输出高电平），熄灭其他段的 LED（为数字引脚 13、10、9、8、7、6 输出低电平）。

### ● while 循环结构

当振动传感器感知到振动后，可使用 while 循环结构来中断数码管的随机显示。while 循环结构由条件语句和执行语句两部分组成，若条件成立就重复执行循环体内的语句，否则跳出循环体。在 Arduino 中，while 循环结构的语法如下。

形式：

```
while (条件语句)
{执行语句;}
```

**参数：**条件语句即给定的条件；执行语句即条件成立时所执行的语句。

**含义：**通过判断给定条件来控制程序的重复执行。

请同学们根据上述知识，结合随机数编写利用电子色子控制数码管显示数字的程序。



### 任务三 电子色子的测试与优化

#### ※ 活动1 编写子程序

在主程序 loop() 中直接写入数字显示程序会使程序显得冗长。为了使程序更具可读性与规范性，实现代码的可重复使用，提高编程效率，可以使用子程序来优化程序结构。

#### ● 子程序

子程序的使用包括两部分，即子程序的定义与调用。在主程序外部定义子程序，编程过程中可直接调用子程序来完成特定任务。子程序的定义一般由返回值类型、子程序名称、子程序参数和执行语句组成。

**形式1：**返回值类型 名称(数据类型 参数1, 数据类型 参数2)  
 {  
   指令序列;  
 }

**参数：**返回值类型，即子程序返回值的数据类型；名称，即子程序的名字；数据类型，即参数的数据类型；参数，即在指令序列中涉及的参数，也可不设置参数；指令序列，即子程序的执行语句。

**含义：**定义子程序并设置子程序的功能。

**形式2：**名称(参数1, 参数2);

**参数：**名称即所调用子程序的名称；参数即子程序中参数的值，若子程序中没有定义参数，此处可为空。

**含义：**在主程序中调用子程序。

请同学们根据子程序的使用方法，在程序起始部分定义子程序 number1、number2、number3、number4、number5、number6，分别用来显示数字 1~6，并在主程序中直接调用。



因子程序 number1、number2、number3、number4、number5、number6 并不需要返回值，故可以省略定义这些子程序的返回值类型 int。

### ※ 活动2 进一步测试与修改电子色子

根据自己的理解和需求，修改或添加电子色子的功能，解决可能存在的问题，得到更加合理的作品。例如，若电子色子对一个随机数字的显示时间太长，则当电子色子显示了用户想要的数字时，用户可以马上晃动振动传感器来暂停电子色子的显示并得到该数字，这样就破坏了结果的“随机”性质。同学们可以采用缩短随机数字显示时长的方法来避免这种情况的发生。



## 作品发布与评价

参考表 2.5.4 开展作品质量的自评、互评等，与同学分享自己的制作经验。

表 2.5.4 作品评价表

评价指标	得分
1. 能够使数码管显示不同的数字 (3分)	
2. 能够使用振动传感器控制数码管 (3分)	
3. 数码管能随机显示数字 (2分)	
4. 程序设计规范、简洁 (2分)	
总分 (10分)	



## 拓展训练

请同学们结合所学知识，尝试将数码管、振动传感器、LED灯、蜂鸣器等元件进行多样组合，联系生活实际，设计一个实用与创意并存的作品。可以选择以下项目主题。

1. 车牌摇号生成器：使用振动传感器和串口输出，设计一个简易的车牌摇号装置。
2. 带倒计时功能的交通灯：使用LED制作交通灯，并使用数码管为其增加倒计时功能，从而帮助行人和车辆更好地观察通行或禁行的时长。
3. 自行车防盗报警装置：在自行车上安装振动传感器和蜂鸣器，当感受到振动时，蜂鸣器播放报警声。

请进行小组合作，讨论确定主题，除了已经给出的方案，也可以选择其他具有创意的设计主题，形成具体设计方案后填写附录1所示的项目设计方案，并尝试实现。

## 2.6 红外遥控

通过前面的学习，我们知道可以利用声光检测来触发楼道灯，实现非接触式的控制效果。在智能系统的开发中，还有一种非接触式的控制方法——遥控。常用的遥控技术有无线电遥控、红外遥控、超声波遥控等，其中，红外遥控技术具有抗干扰能力强、信息传输可靠、功耗低、成本低等显著优点，广泛应用于各类家用电器。利用该技术可以实现对一些设备的较远距离控制，如风扇、电机、LED、舵机等。



### 学习目标

- ★ 了解红外遥控原理。
- ★ 掌握字符串的使用方法和键值的解码方法。
- ★ 掌握电机的接线方法，能够利用PWM方式驱动电机。
- ★ 能够利用红外遥控套件实现远程控制。

风扇通常是通过按键来控制风速的，有没有更方便的方法来控制风扇呢？

要制作一个不用接触即可换挡的风扇，需要考虑采用遥控的方法替换原来的换挡方式，并使不同的遥控信号与风扇的换挡状态建立对应关系。要解决这些问题，需要完成以下三个任务：遥控风扇项目规划、遥控风扇的实现、遥控风扇的测试与优化。



### 任务一 遥控风扇项目规划

#### ※ 活动1 遥控风扇项目调研

为了设计新的遥控风扇，有必要先调研一下现实生活中常用



电器的遥控方法和原理。请同学们利用网络查找常用电器的遥控方式，了解它们各自的特点，填写表2.6.1。

表2.6.1 常用电器的遥控方式调查表

序号	电器	遥控方式	优势与不足
1			
2			
3			
4			

通过调研，我们可以发现，常用电器的遥控方式多使用遥控器来实现，也有个别的采用其他非接触式的控制方式，且有效遥控距离各不相同。也就是说，遥控换挡的实现主要与遥控设备及其可控距离有关，其中控制方式涉及可控距离，是需要重点考虑的因素。

### ※ 活动2 遥控风扇的可选方案分析与设计

结合活动1的调研，可以从遥控设备类型和控制方式两个角度出发设计遥控风扇，初步形成几种方案，如表2.6.2所示。

表2.6.2 遥控风扇的可选方案

方案	方案描述
方案一 (红外测距遥控)	使用红外距离传感器测距，不同距离代表不同挡位
方案二 (红外遥控器遥控)	使用红外遥控套件，不同按键代表不同挡位
方案三 (声音传感器遥控)	使用声音传感器套件，不同音量代表不同挡位
方案四 (语音识别遥控)	使用语音识别传感器套件，用语音控制挡位

每个方案各有利弊：从控制的有效性看，方案一和方案二控制较为稳定，抗干扰能力强，而方案三和方案四容易受周围事物的影响出现误操作；从使用的方便性看，方案二可以远距离调节挡位，用户体验较佳，而其他方案需要用户在离风扇较近处进行控制，不够方便。因此，基于统筹与折中的工程设计思想，在综合考虑控制的有效性和使用的方便性等因素后，选择方案二作为最终实施方案。

要实现方案二，关键在于如何利用按键调节挡位。请根据上面的描述，补充完整如表 2.6.3 所示的设计方案。

表 2.6.3 遥控风扇的设计方案

作品名称	遥控风扇
作品功能	利用红外遥控器的按键控制风速
器材清单	Arduino 控制板、USB 数据线、红外遥控套件、直流电机、风扇
运行流程	
实现过程	<ol style="list-style-type: none"> <li>① 直流电机、红外接收头与 Arduino 控制板连接</li> <li>② 遥控器解码、编写遥控风扇程序</li> <li>③ 上传程序，测试效果</li> </ol>

## ● 直流电机

直流电机是应用电磁感应原理实现直流电能和机械能互相转换的装置。当它作为电动机运行时是直流电动机，将电能转换为机械能；作为发电机运行时是直流发电机，将机械能转换为电能。

由于数字引脚直接输出的电流非常小，而电机转动需要较大电流，因此需要连接专门的电机输出接口才能正常工作。Arduino 控制板提供了 M1 和 M2 两个电机接口，如图 2.6.1 所示。

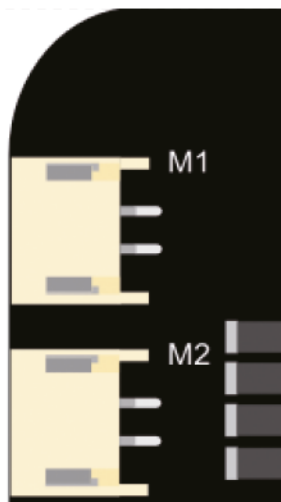



图 2.6.1 M1 和 M2 接口

## ● 红外遥控原理

红外遥控是利用波长为  $0.76\mu\text{m} \sim 1.5\mu\text{m}$  的近红外线传送控制信号的。常见的红外遥控系统一般分为发射端和接收端两个部分。在发射端按下某一按键后，会发出一连串经过调制后的信号，经过接收端接收后向控制板输入解调后的数字脉冲。每个按键对应不同的脉冲，故识别出不同的脉冲就能识别出不同的按键。

 发射部分的主要元件为红外发光二极管，它实际上是一只特殊的发光二极管，在其两端施加一定电压时，它便发出红外线而不是可见光。接收部分的红外接收管是一种光敏二极管。

红外遥控的优势是不影响周边环境、不干扰其他电气设备；而且电路调试简单，只要给定电路连接无误，一般无须任何调试即可投入工作；编解码简单，可进行多路遥控。多路控制的红外发射部分一般有多个按键，代表不同的控制功能。

本节中使用的红外遥控套件主要包括红外遥控器（红外发射端）和红外接收头两部分，如图 2.6.2 所示。该红外遥控器有 21 个按键，每个按键都有对应的键值（数字脉冲）。红外接收头有三个引脚，标有 D、VCC 和 GND 的引脚分别对应连接 Arduino 控制板的数字引脚、5V 和 GND 引脚。



图 2.6.2 红外遥控套件

## 任务二 遥控风扇的实现

**💡** 由于本节只需要使用一个电机，故可以将电机连接到 M1 或 M2 接口。由于 M1 和 M2 占用了数字口 4、5、6、7，而 0 和 1 用于控制板与串口通信，故红外接收头可以接到除 0、1、4、5、6、7 以外的数字口。

### ※ 活动 1 遥控风扇的硬件搭建

准备好器材后，请同学们参考图 2.6.3 将红外接收头连接在 Arduino 控制板的数字引脚 11 上，将电机接在 M1 上。

### ※ 活动 2 遥控风扇的程序编写

完成硬件搭建任务后，首先需要设置风扇挡位。由于风扇是由电机驱动的，如果使用数字输出模式来控制电机，将无法改变电机的转速。此时需要采用 PWM 模拟输出模式来控制电机。

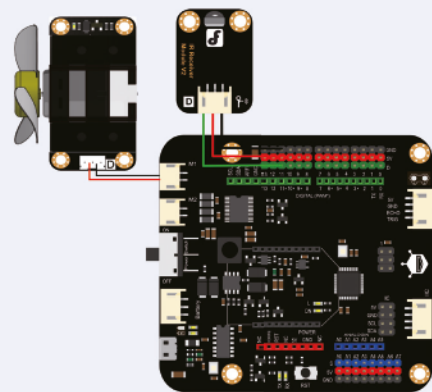


图 2.6.3 硬件连接图

**💡** 本节的 PWM 模拟输出是硬件 PWM；2.2 节的为软件 PWM。硬件 PWM 是通过硬件上的相关电路实现脉冲频率或占空比的控制。软件 PWM 只是调整脉冲频率，其优点是不会被限制于某个引脚，并且频率可自由调节；缺点是会占用系统定时器，容易与其他使用了同一个定时器的库产生冲突。

## ● PWM 控制

驱动电机需要采用 PWM（Pulse Width Modulation，脉冲宽度调制或脉宽调制）模拟输出模式。简单地说，就是通过控制高/低电平信号交替频率（脉冲频率）或高电平信号的持续时间（占空比），从而控制电机转速。脉冲频率或占空比越高，电机转速越快。

Arduino 控制板有 6 个支持 PWM 的数字引脚，这些引脚标有“\*”标记，分别是 3、5、6、9、10、11 等引脚。PWM 的输出值为 0~255，通过改变输出值可以调节直流电机的转速。在 Arduino 中，利用 analogWrite() 函数实现 PWM 模拟输出。

**形式：**analogWrite(pin,value)

**参数：**pin 指进行 PWM 输出的引脚；value 指占空比，数值为 0~255，0 表示完全关闭，255 表示完全打开。

**含义：**从一个引脚输出 PWM 模拟值。

Arduino 控制板利用 PWM 模式控制电机的引脚分配如表 2.6.4 所示。

表 2.6.4 PWM 电机引脚分配

引脚编号	功能
4	电机 M1 的方向控制
5	电机 M1 的 PWM 控制
6	电机 M2 的 PWM 控制
7	电机 M2 的方向控制

给引脚 4 和 7 设置高电平或低电平即可控制电机的正转或反转，一般高电平控制电机正转，低电平控制电机反转，但也可能相反，这取决于电机的接线顺序；给引脚 5 和 6 输出 PWM 值可以控制电机的转动速度，PWM 的取值越大，转动速度越大。当 PWM 取值较小时，可能会因为电流太小而无法使电机转动，但可以听到电机嗡嗡的声响。

假设驱动 1 挡、2 挡、3 挡风速的 PWM 值分别为 80、150、250。驱动 1 挡风速的程序如下所示。请同学们编写电机转速由 1 挡变为 2 挡再变为 3 挡的程序，将程序上传到控制板，观察风速变化是否达到了预期。

```
void setup()
{
  pinMode(4,OUTPUT);    //电机连接在 M1，设置引脚 4 控制电机的方向
  pinMode(5,OUTPUT);    //定义引脚 5 为控制电机 PWM 输出
}
void loop()
{
  digitalWrite(4,HIGH); //设置电机为正转
  analogWrite(5,80);    //设置电机速度为 80
}
```



### 任务三 遥控风扇的测试与优化

#### ※ 活动 1 遥控换挡

完成以上任务后，还不能用遥控器实现换挡功能。主要原因是风扇还不能识别其按键，无法根据按键信息来调整风速，因此，接下来需要解决按键解码的问题。



本书所用控制板在 UNO 标准板基础上集成了 2 路直流电动机驱动器，并通过改变两个数字 I/O 引脚的高低电平和两个 PWM 引脚的 PWM 值控制直流电机。若要在 UNO 标准板上驱动电机，需要安装电机驱动扩展板。



库就像日常所用的工具箱一样，里面有许多函数（工具），当我们需要相关函数时，不用自己编写，而是直接调用即可，从而提高程序的编写效率。在Arduino中，库是一个包含若干文件的文件夹，且至少包含两个文件：头文件（扩展名为“.h”）和源程序文件（扩展名为“.cpp”）。头文件包含库的声明，即库的功能说明；源程序文件包含库的实际实现。



遥控器按键与键值对应表

按键名称	键值
红色按钮	FD00FF
VOL+	FD807F
FUNC/ STOP	FD40BF
左侧两个 三角	FD20DF
暂停键	FDA05F
右侧两个 三角	FD609F
下三角	FD10EF
VOL-	FD906F
上三角	FD50AF
0	FD30CF
EQ	FDB04F
ST/REPT	FD708F
1	FD08F7
2	FD8877
3	FD48B7
4	FD28D7
5	FDA857
6	FD6897
7	FD18E7
8	FD9867
9	FD58A7

## ● 红外解码

在Arduino中获取键值的方法叫红外解码，即获取红外遥控器按键的键值并将其值赋给字符串变量。通过下述程序可以在串口监视器中观察每个按键的键值。

```
#include <IRremote.h>           //引用具有红外解码功能的IRremote函数库
int irReceiverPin = 11;        //定义引脚变量，指定接收红外信号的引脚
IRrecv irrecv(irReceiverPin);
decode_results results;

void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn();
}

void loop()
{
  if (irrecv.decode(&results))
  {
    Serial.println(results.value, HEX); //以16进制换行输出接收编码
    Serial.println();                 //为了便于观看输出结果增加一个空行
    irrecv.resume();                  //接收下一个值
  }
}
```

在红外遥控库中，编码的数据类型为 decode\_results，因此使用语句 decode\_results results 定义存储解码的变量。该数据类型是一种结构体，results.value 表示结构体中实际的编码值。

IRrecv() 函数用来定义接收头的引脚，语法为 IRrecv irrecv(recvpin)，recvpin 可以是除了 0 和 1 外的任意数字引脚。

enableIRIn() 函数的语法为 irrecv.enableIRIn()，用来初始化红外接收器，使传感器能进行红外信号接收。

decode() 函数的语法为 irrecv.decode(&results)，用来判断是否已收到编码，如果接收成功则返回 True，否则返回 False，并将结果保存到 results 结构中。

resume() 函数的语法为 irrecv.resume()，一旦编码被解码必须调用其对接收器进行复位和初始化，才能接收下一组编码。

请利用以上解码程序测试一下遥控器上常用按键的键值，并与上一页边栏提示的键值做比较。

要实现利用按键控制风速，需要对红外接收头接收到的键值进行判断，只有键值匹配时才能改变风速。然而，红外接收头接收到的值为数值型，而键值为字符型，因此需要进行数据类型转换。ltoa函数能实现数据类型转换功能。

**形式：**char \*ltoa(long value,char \*string,int radix)

**参数：**value 指要转换的数值；string 指转换后指向字符串的指针；radix 指转换后的进制，如HEX表示十六进制。

**含义：**把value的值转换为radix进制的字符串，并把结果保存在string中。

由于字符串键值为6位，因此可以在使用该函数前先定义一个长度为6的字符串变量：char a[6]。但是，使用ltoa(results.value, a, HEX)转化后的十六进制键值为小写字母，还需要利用子程序将其转化为大写字母，转化程序如下所示。

```
void charsToUpper(char *str)          /*str指要转换的字符串
{
    for(int p=0;str[p] != 0; p++)
    {
        //toupper()函数可以将小写字母转换为大写字母
        str[p] = toupper(str[p]);
    }
}
```

请同学们将上述子程序添加到setup()之前，在函数ltoa(results.value, a, HEX)后调用charsToUpper(a)将字符串a转换为大写形式，然后在其后增加一个选择结构用来判断键值和执行风速调节语句。应注意选择结构的判断条件需要使用字符串比较函数strcmp()。例如：strcmp(a,"FD08F7")==0，表示当按下“1”键时条件成立。

### ※ 活动2 进一步测试与修改遥控风扇

根据自己的理解和需求，添加或修改既有遥控换挡的设计方案，并完成该作品的外形包装，形成有一定个人特色的作品。



## 作品发布与评价

参考表2.6.5开展作品的自评、互评等，与同学分享自己的制作经验。

表2.6.5 作品评价表

评价指标	得分
1. 能够利用模拟输出方式控制风扇的速度（2分）	
2. 能够获取遥控器按键的键值（2分）	
3. 能够实现遥控换挡功能（4分）	
4. 挡位设计合理（2分）	
总分（10分）	



风扇的转向需要使用舵机来实现，感兴趣的同学可以查阅舵机的使用方法。



## 拓展练习

自从有了遥控装置，我们的生活更加便捷了。同学们可以利用红外遥控套件和前面所学的知识来设计一个自己的遥控装置，以帮助你解决生活中的问题。可以参考以下项目主题。

1. 遥控风扇方向：使用红外遥控套件、直流电机、舵机和风扇，利用红外解码原理实现风扇方向的切换，如当按下1时风扇向左转，按下2时风扇向右转。

2. 遥控道闸：使用红外遥控套件和直流电机，制作一个遥控道闸，如“下三角”按键控制道闸下降、“上三角”按键控制道闸上升。

当然，同学们也可以选择其他的项目主题。请进行小组合作，讨论确定主题，形成具体设计方案后填写附录1所示的项目设计方案，并通过小组合作尝试实现。

## 2.7 超声波测距

2.6节的风扇虽然实现了遥控换挡，但却需要人为控制，不够“智能”。要使风扇能够根据外界环境调节风速，还需要安装其他非接触式传感器。超声波传感器是一种常用的非接触式传感器，利用它可以精确地检测机器人与周边障碍物的距离，从而实现根据距离的变化自动控制机器人的行为。



### 学习目标

- ★ 掌握超声波传感器的使用方法，了解其测距原理。
- ★ 能够利用超声波传感器实现对智能装置的自动控制功能。
- ★ 能够正确使用映射函数，并理解不同数据集之间的等比例转换思想。
- ★ 在利用子程序优化程序结构的过程中，体会软件复用和结构化程序设计思想。

制作完遥控风扇后，我们还可以从智能家居的角度进一步改进风扇，使它具备自动变速功能。

根据前面几节课的学习，要实现自动变速就要考虑两个问题：使用哪种传感器？如何控制风速？要解决这些问题，需要完成以下三个任务：自动变速风扇项目规划、自动变速风扇的实现、自动变速风扇的测试与优化。



### 任务一 自动变速风扇项目规划

#### ※ 活动1 自动变速风扇项目调研

为了获得更多灵感，有必要对智能家居的控制方式进行专题调



研。请同学们利用网络查找目前常见的智能家居，了解它们各自的控制特点，填写表2.7.1。

表2.7.1 智能家居调查表

序号	智能家居	控制方式	优势与不足
1			
2			
3			
4			

通过调研可以发现，常见的智能家居大多是通过感应光线、声音、温度等属性实现自动控制的。因此，我们可以制作一个能够根据外界环境的变化自动改变转速的风扇，以获得更愉快的用户体验。

影响风扇风速变化的因素主要与控制距离和环境温度有关，其中有关距离或温度的检测是关键。

### ※ 活动2 自动变速风扇的可选方案分析与设计

根据活动1的调研，可以从距离控制和温度控制两个角度出发设计自动变速风扇，初步形成几种方案，如表2.7.2所示。

表2.7.2 自动变速风扇的可选方案

方案	方案描述
方案一 (距离控制)	利用超声波传感器测距，在某个范围内，与人的距离越大，风扇转动速度越大，反之转速越小
方案二 (距离控制)	利用热释电红外传感器检测人，当有人时再利用超声波传感器测距，在某个范围内，与人的距离越大，风扇转动速度越大，反之转速越小
方案三 (距离控制)	利用红外距离传感器测距，在某个范围内，与人的距离越大，风扇转动速度越大，反之转速越小
方案四 (温度控制)	利用温度传感器控制风扇，温度高于一定值时，风扇开启，温度越高，风扇转动速度越大，反之转速越小

每个方案各有利弊：从学习内容的难度来看，方案二涉及热释电红外传感器，并且还有选择结构的嵌套，难度较大，其他三种方案比较适中；从成本角度看，使用红外距离传感器的成本要远高于

使用超声波传感器。因此，在综合考虑学习难度、课堂时间和成本等因素后，可以选择方案一作为最终实施方案。

要实现方案一，关键在于如何控制风速随着人与风扇的距离而自动变化，这又涉及两个问题，一是距离信息的自动检测，二是距离与风扇速度的对应关系。请根据上面的描述，将表2.7.3所示的设计方案补充完整。

表2.7.3 自动变速风扇的设计方案

作品名称	自动变速风扇
作品功能	根据距离自动控制风扇的转速
器材清单	Arduino 控制板、USB 数据线、HC-SR04 超声波传感器、直流电机、风扇
运行流程	
实现过程	<ol style="list-style-type: none"> <li>① 直流电机、超声波传感器与 Arduino 控制板连接</li> <li>② 编写程序，实现风扇自动变速</li> <li>③ 上传程序，测试效果</li> </ol>

## ● 超声波传感器

声音是由物体振动产生的，人耳能够听到的声音频率为 20Hz~20kHz，超过 20kHz 的称为超声波，低于 20Hz 的称为次声波。利用超声波可以设计各种传感器，并在通信、医疗、家电等方面得到广泛应用。图 2.7.1 所

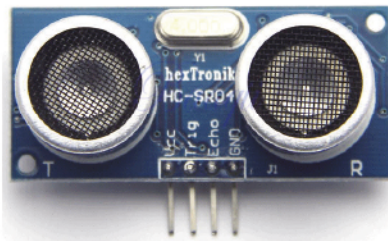


图 2.7.1 HC-SR04 超声波传感器

示的 HC-SR04 超声波传感器具有测距功能，其主要参数如表 2.7.4 所示。HC-SR04 超声波传感器有四个引脚，除了 VCC 和 GND 之外，Trig（超声波发射端）和 Echo（超声波接收端）均连接数字引脚。

表 2.7.4 HC-SR04 超声波传感器的主要参数

使用电压	静态电流	电平输出	感应角度	探测距离	精度
直流 5V	小于 2mA	高/低	不大于 15°	2cm~450cm	0.2cm



一般来说，在使用 HC-SR04 超声波传感器时，应先将其在电路板上插好后再通电，避免产生高电平的误动作，如果产生了，重新通电方可解决。

可以使用橡皮筋或者胶带将该超声波传感器固定在合适位置。

由于超声波指向性强，能量消耗缓慢，在介质中传播的距离较远，因而超声波经常用于距离的测量，如测距仪和物位测量仪等。虽然目前在测距量程上能达到百米，但测量的精度往往只能达到厘米数量级。

本书所用Arduino控制板中自带的超声波接口默认 Trig 连接数字引脚8，Echo 连接数字引脚12。当然，也可以直接将超声波传感器连接到其他数字引脚上。

因超声波传感器的传输时间以 $\mu\text{s}$ （微秒）为单位、检测的距离单位为 $\text{cm}$ ，而 $1\text{s}=1000000\mu\text{s}$ ，所以声波的传播速度为 $340\text{m/s}$ ，即 $0.034\text{cm}/\mu\text{s}$ 。根据测距公式：距离=（声速 $\times$ 高电平时间）/2，可以推导出距离 $s=(0.034\times\text{高电平时间})/2=0.017\times\text{高电平时间}$ 。

## ● 超声波传感器测距的原理

超声波发射器向某一方向发射超声波，与此同时开始计时，超声波在空气中传播，途中碰到障碍物后立即返回，超声波接收器收到反射波便立即停止计时，然后就可以根据发射和接收的时间差计算出发射点到障碍物的实际距离。测距的公式表示为： $L=C\times T$ 。其中， $L$ 为测量距离的长度， $C$ 为超声波在空气中的传播速度，即 $340\text{m/s}$ ， $T$ 为从发射到接收所需时间的一半。



## 任务二 自动变速风扇的实现

### ※ 活动1 自动变速风扇的硬件搭建

准备好器材后，请同学们参考图2.7.2将电机接在Arduino控制板的电机接线柱M1上，超声波传感器的Trig连接在数字引脚8上，Echo连接在数字引脚12上。

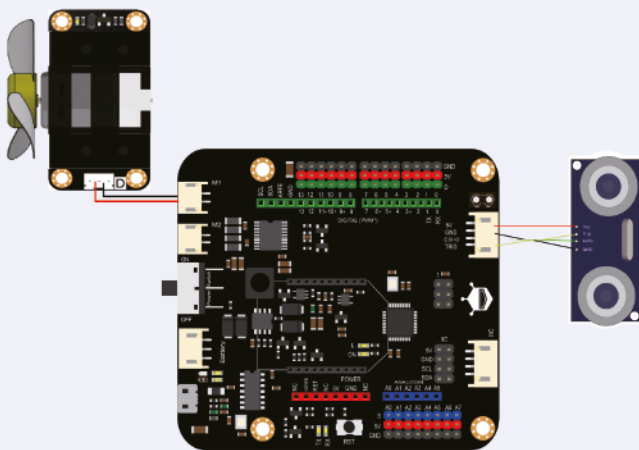


图2.7.2 硬件连接图

### ※ 活动2 自动变速风扇的程序编写

测距原理很好理解，但关键问题在于如何获取超声波从发送到接收所需的时间。

## ● 超声波传感器测距的实现

HC-SR04 超声波传感器测距原理较复杂，首先通过至少 $10\mu\text{s}$ 的高电平信号触发测距；然后，超声波传感器自动发送八个 $40\text{kHz}$ 的方波，并自动检测是否有信号返回；最后，如果检测到有信号返回，则输出高电平。那么，距离=（声速 $\times$ 高电平时间）/2。超声波测距并通过串口监视器显示距离的程序如下所示。

```

void setup()
{
  Serial.begin(9600);
  pinMode(5,OUTPUT); //分配电机连接的引脚
  pinMode(8,OUTPUT); //定义Trig（超声波发射端）
  pinMode(12,INPUT); //定义Echo（超声波接收端）
}
void loop()
{
  int s; //定义变量s
  digitalWrite(8,LOW);
  delayMicroseconds(2);
  digitalWrite(8,HIGH);
  delayMicroseconds(10); //通过输入Trig端10μs的高电平信号触发测距
  digitalWrite(8,LOW);
  //检测超声波接收端高电平持续的时间（μs），赋予变量s
  s=pulseIn(12,HIGH);
  s=0.017*s; //计算得出超声波测得的距离值（cm），赋予变量s
  Serial.println(s); //通过串口输出距离值
  delay(100);
  analogWrite(5,s); //设置风扇的转速
}

```

根据上述知识，请同学们编写控制风扇自动变速的程序，将程序上传到控制板，用手由远到近挡住超声波传感器，观察风速变化是否达到了预期效果。



### 任务三 自动变速风扇的测试与优化

#### ※ 活动1 使用映射函数与编写子程序

经过测试，上述程序不能达到预期效果，因为HC-SR04超声波传感器的探测距离是2cm~450cm，与电机的PWM输出范围不匹配。

#### ● 映射函数

为了解决电机的PWM输出范围与超声波测距范围不匹配的问题，需要使用映射函数。

**形式:**  $\text{map}(\text{value}, \text{fromStart}, \text{fromEnd}, \text{toStart}, \text{toEnd})$

**参数:**  $\text{value}$  为需要被映射的变量;

$\text{fromStart}$  为变量当前范围的开始 (映射起点值的下限);

$\text{fromEnd}$  为变量当前范围的结束 (映射起点值的上限);

$\text{toStart}$  为映射后变量范围的开始 (映射目标值的下限);

$\text{toEnd}$  为映射后变量范围的结束 (映射目标值的上限)。



如果映射后的开始值  $\text{toStart}$  大于映射后的结束值  $\text{toEnd}$ , 输出范围会是相反的。例如  $\text{map}(s, 2, 450, 255, 60)$ , 如果  $s$  为 2, 映射后会变为 255。

**含义:** 将一个在某区间变化的变量, 按照比例重新映射到另外一个区间中, 它在映射前后的函数关系为

$$\text{return} = (\text{value} - \text{fromStart}) + \text{toStart}$$

以上述超声波传感器为例, 其有效探测距离为 2cm~450cm, 而电机的转动数值为 0~255, 但要驱动电机带动风扇转起来, 需要输入较大的值, 如 60~255。那么, 映射函数可以写成:  $\text{map}(s, 2, 450, 60, 255)$ , 如果  $s$  为 2, 映射后会变为 60; 如果  $s$  为 450, 映射后会变为 255。映射函数的映射机制如图 2.7.3 所示。从图中可以看出, 映射函数将被映射的数值等比例放大或缩小了。也就是说, 映射函数实现了不同数据集合之间的自动转换。

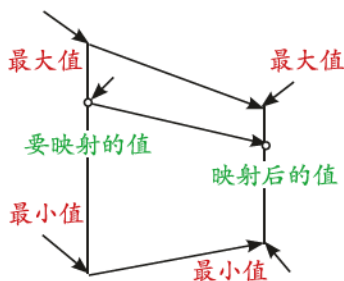


图 2.7.3 映射函数的映射机制图示

请同学们使用映射函数和子程序优化程序结构, 并进行测试。

### ※ 活动 2 进一步测试与修改自动变速风扇

编译并上传程序, 可能会发现风扇转速变化不明显。原因在于映射的值范围过大, 因此可以使用选择结构 ( $\text{if}\cdots\text{else}\cdots$ ) 对测距数值进行判断。例如, 若距离小于 100cm, 则自动变速, 否则风扇按某个固定风速转动。请同学们用选择结构优化程序, 测试能否达到预期效果。还可以根据自己的理解 and 需求, 添加或修改既有自动风扇的设计方案, 形成具有一定个人特色的作品。



## 作品发布与评价

参考表2.7.5开展作品质量的自评、互评等，与同学分享自己的制作经验。

表2.7.5 作品评价表

评价指标	得分
1. 能够利用测距原理实现超声波测距（2分）	
2. 风扇能够随着距离变化实现自动变速（5分）	
3. 风扇转速变化满足实际需求（3分）	
总分（10分）	



## 拓展练习

超声波传感器在日常智能装置中非常常见，请综合运用超声波传感器和前面学过的知识设计一个具有创意的智能装置。可参考以下项目主题。

1. 倒车雷达：使用蜂鸣器和超声波传感器，使车在后退过程中能够根据与障碍物的距离自动报警，离障碍物越近报警声音越大或频率越快。
2. 创意LED：使用全彩LED和超声波传感器，制作一个自动调色灯，使灯光颜色随距离值变化而有序变化。

当然，同学们也可以选择其他的项目主题。请进行小组合作，讨论并确定主题，形成具体设计方案后填写附录1所示的项目设计方案，再尝试实现。

## 2.8 双电机控制

无论是遥控风扇还是自动变速风扇，都只用到了一个电机，用于控制风扇的转动。但在更复杂的任务当中，通常需要两个甚至更多电机的配合。例如，利用双电机控制小车的两个轮子，不仅可以控制小车的前进，还可以通过调节各个电机的转向与转速使小车完成后退、转弯等更为复杂的动作。



### 学习目标

- ★ 了解双轮驱动方式，理解双轮差速原理，掌握防跌落传感器的使用。
- ★ 能够使用防跌落传感器实现小车的防跌落功能，感受从简易小车到智能小车的迭代过程。
- ★ 能够使用逻辑运算符表示条件语句的逻辑关系，以优化程序结构。

本节课要制作一台智能小车，使其不仅能够稳定行驶，还能够遇到塌方、悬崖等路况时防止跌落。

要制作具备防跌落功能的小车，需要考虑使用几个电机驱动小车以及使用何种传感器探测路况。要解决这些问题，需要完成以下三个任务：防跌落小车项目规划、防跌落小车的实现、防跌落小车的测试与优化。



### 任务一 防跌落小车项目规划

#### ※ 活动1 防跌落小车项目调研

网上有很多相关的案例，但是实现方法各有不同。请同学们开展调查，了解常见的防跌落小车的构成与制作方法，填写表2.8.1。

表2.8.1 防跌落小车调查表

序号	防跌落小车的类型	构成	制作方法
1			
2			
3			
4			

通过调研可以发现，防跌落小车主要由驱动装置和距离检测装置组成。其中，如何判断小车处在“悬崖”边缘是关键。

### ※ 活动2 防跌落小车的可选方案分析与设计

根据上述思路，可以从驱动方式和距离检测元件两个角度出发设计防跌落小车，初步形成几种方案，如表2.8.2所示。

表2.8.2 防跌落小车的可选方案

类别	方案	方案描述
驱动方式	方案一	使用双电机驱动小车
	方案二	使用四个电机驱动小车
距离检测元件	方案三	使用超声波传感器检测小车前方是否有较大高度差
	方案四	使用数字防跌落传感器检测小车前方是否有较大高度差

经过调研，每个方案各有利弊：从学习内容的角度看，控制四个电机驱动小车的操作较复杂；从实际应用的角度看，防跌落传感器体积较小，容易安装。因此，综合考虑后，我们以方案一和方案四为基础设计最终的实施方案：使用双电机驱动小车，并结合防跌落传感器实现小车的防跌落功能。



使用四个电机驱动的小车称为四驱车。对于普通的两驱车，当其中的一只驱动车轮打滑时，另一只车轮也会失去动力，此时小车便无法行驶。当四驱车遇到这种情况时，另外两只车轮仍具有牵引力，因此四驱车能够适应复杂些的路况。




要实现以上方案，关键在于当防跌落传感器检测到前方有较大高度差时如何改变小车的行驶状态。请同学们根据上面的描述，将表2.8.3中的运行流程补充完整。

表2.8.3 防跌落小车的设计方案

作品名称	防跌落小车
作品功能	小车向前行驶，当检测到周围有较大高度差时，改变行驶方向
器材清单	Arduino控制板、小车、外接电池、USB数据线、防跌落传感器
运行流程	
实现过程	<ol style="list-style-type: none"> <li>① 搭建小车，将防跌落传感器、电机与Arduino控制板连接</li> <li>② 编写程序，驱动小车并实现小车的防跌落功能</li> <li>③ 上传程序，测试效果</li> </ol>

## ● 防跌落传感器

 合理安装防跌落传感器可以准确地检测出台阶等具有高度落差的路况，从而让机器人考虑是否绕道行驶。

当防跌落传感器与地面的距离过近时，会影响其检测结果，因此安装防跌落传感器时，应注意使其与地面的距离不小于1cm。

防跌落传感器是一款能够在检测范围内探测周围环境是否存在较大距离差的数字传感器，如图2.8.1所示。该传感器采用红外测距模块，当检测到的距离值小于或等于10cm时输入低电平，当检测到的距离值大于10cm时输入高电平。

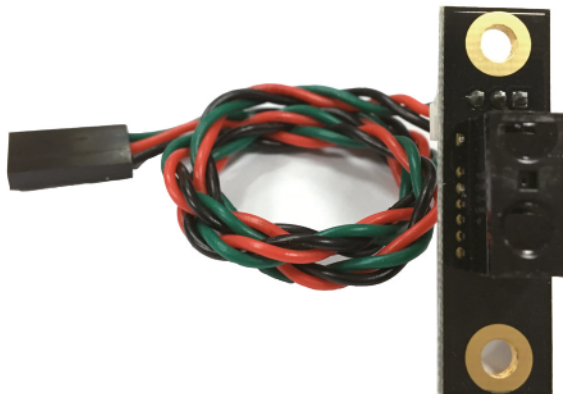


图2.8.1 防跌落传感器



## 任务二 防跌落小车的实现

### ※ 活动1 防跌落小车的硬件搭建

准备好器材后，请同学们参考以下步骤完成组装工作。

(1) 组装车体：组装顺序参考图 2.8.2。

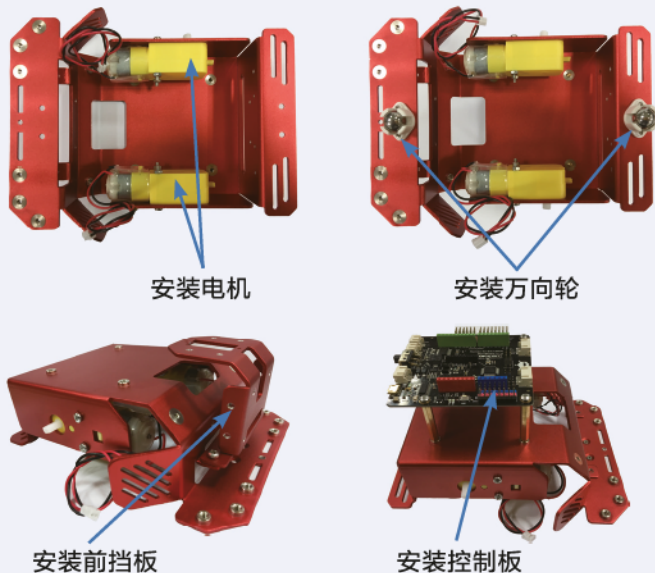


图 2.8.2 小车搭建顺序

(2) 安装配件：将车轮与电机相连，四个防跌落传感器分别安装在小车的左前方、右前方、左后方、右后方，效果图可参考图 2.8.3。

(3) 接线：将左侧电机连接到 Arduino 控制板的 M1 电机接口，将右侧电机连接到 M2 电机接口，左前方、右前方、左后方、右后方四个传感器分别接在数字引脚 9、10、11、12 上，如图 2.8.4 所示。

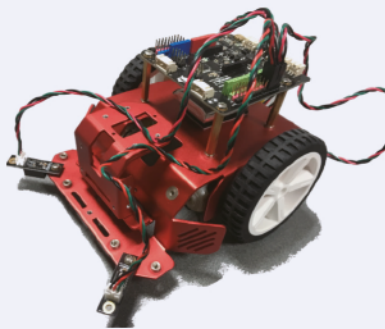


图 2.8.3 小车搭建效果图

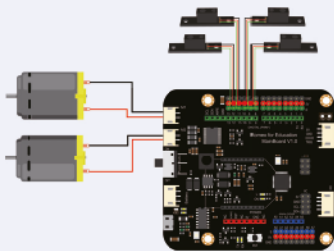


图 2.8.4 硬件连接图

### ※ 活动2 防跌落小车的程序编写

两个电机需要相互配合才能使小车向不同方向行驶，但是这其中有什么规律呢？小车什么情况下会转弯，什么情况下会直行呢？



当控制板连接过多元件时，看上去接线会比较混乱，影响美观，此时可使用扎带来整理连接线。

## ● 双轮差速原理

控制双驱小车的运动实际上就是控制两个电机的转向与转速，通过两个电机的配合来实现小车的前进、后退、转弯等动作。

当小车左右两个电机的转速与转向均相同时，小车能够前进或后退，如图 2.8.5 所示；当两个电机的转速相同而转向相反时，小车会原地转圈，如图 2.8.6 所示；当两个电机的转速和转向均不同时，小车会往向后转的电机一侧转弯，如图 2.8.7 所示；当两个电机转向相同而转速有差异时，小车会向转速低的电机一侧转弯，如图 2.8.8 所示。



图 2.8.5 小车前进与后退



图 2.8.6 小车原地转圈



图 2.8.7 小车左转与右转



图 2.8.8 小车左转与右转

小车前进、原地转圈、左转和停止的子程序如下所示，使用时可直接调用。后退和右转的程序分别参考前进和左转的程序。

```
void setup()
{
    pinMode(4,OUTPUT);    //定义引脚4为控制电机M1的方向
    pinMode(5,OUTPUT);    //定义引脚5为控制电机M1的PWM输出
    pinMode(6,OUTPUT);    //定义引脚6为控制电机M2的PWM输出
    pinMode(7,OUTPUT);    //定义引脚7为控制电机M2的方向
}
//定义小车行驶状态子程序，方便需要时调用
void straight()    //小车直行
{
    digitalWrite(4,HIGH);
    digitalWrite(7,HIGH);
    analogWrite(5,100);
    analogWrite(6,100);
}
void circle()    //小车原地转圈
{
    digitalWrite(4,HIGH);
    digitalWrite(7,LOW);
    analogWrite(5,100);
    analogWrite(6,100);
}
void left()    //小车左转
{
    digitalWrite(4,HIGH);
    digitalWrite(7,HIGH);
    analogWrite(5,50);
    analogWrite(6,100);
}
void stop()    //小车停止
{
    analogWrite(5,0);
    analogWrite(6,0);
}
void loop()
{
    straight();
```

```

delay(1000);
circle();
delay(1000);
left();
delay(1000);
stop();
delay(1000);
}

```

请同学们根据上述知识编写防跌落小车的程序，测试防跌落小车的运动情况。



### 任务三 防跌落小车的测试与优化

#### ※ 活动1 编辑条件语句的逻辑关系

在编写程序时，多个选择结构可能会使用相同的执行语句。能否把这些选择结构的条件语句合并在一起呢？条件语句之间的关系又应该如何表达呢？

#### ● 条件语句的逻辑关系

在选择结构中，当执行语句受多个条件限制时，可使用逻辑运算符（布尔运算符）表示条件语句的逻辑关系，以优化程序结构。

**形式1：**条件语句1 &&条件语句2

**含义：**当条件语句1和条件语句2同时成立时。

**形式2：**条件语句1 || 条件语句2

**含义：**当条件语句1和条件语句2中的任意一个条件成立时。

**形式3：**!条件语句

**含义：**当条件语句不成立时。

根据上述知识，使用逻辑运算符“||”来表达条件之间的逻辑关系，即当四个防跌落传感器中的任意一个检测到有较大高度差时，小车便停止运动。loop()程序如下所示，请同学们根据提示将程序补充完整。

```

void loop()
{
    //判断四个防跌落传感器的值
    if(digitalRead(9)==true || digitalRead(10)==true ||

```

```
digitalRead(11)==true || digitalRead(12)==true )
{
    //请补充小车停止程序
}
else
{
    //请补充小车直行程序
}
}
```

### ※ 活动2 进一步测试与修改防跌落小车

根据自己的理解 and 需求，添加或修改防跌落小车的功能，形成具有一定个人特色的作品。例如，为了使小车更加智能，当防跌落传感器检测到有较大高度差时，小车在避免跌落的同时应该继续行驶。即当检测到前方有较大高度差时，小车先后退然后转弯，转弯一定角度后再直行；当检测到后面有较大高度差时，小车向前行驶。请尝试编写程序提高防跌落小车的智能行驶水平。



## 作品发布与评价

参考表2.8.4开展作品质量的自评、互评等，与同学分享自己的制作经验。

表2.8.4 作品评价表

指标	得分
1. 小车结构搭建合理、防跌落传感器安装正确（2分）	
2. 小车能够实现直行、转弯等动作（3分）	
3. 小车能够实现防跌落功能（3分）	
4. 能够正确使用逻辑运算符表达条件语句间的关系（2分）	
总分（10分）	



## 拓展练习

本节制作的防跌落小车功能单一，请综合利用超声波传感器、红外遥控器、蜂鸣器等元件，设计一个具有创意的多功能小车。可参考以下项目主题。

1. 遥控小车：为机器人小车安装红外接收头，使用红外遥控器控制小车的运动状态。

2. 避障小车：为机器人小车安装超声波传感器，制作自动避障小车。当超声波检测到前方有障碍物时停止运动或改变运动方向。

请进行小组合作，讨论并确定主题，除了已有方案，也可以选择其他具有创意的设计主题，形成具体设计方案后填写附录1所示的项目设计方案，再尝试实现。

## 2.9 循线

通过双轮差速原理驱动小车行驶时会发现，即使左右两个电机的转速一致，小车仍很难在一个方向上长时间持续直行。如要使小车按一定的路线前进，就需要给小车安装“眼睛”——循线传感器。该传感器是数字传感器，可用来检测道路上的标线，一旦走偏，小车可自主调整，回到路线上。通过采取适当的循线策略，可以使小车完成更复杂的循线挑战赛。



### 学习目标

- ★ 掌握循线传感器的使用，了解其工作原理。
- ★ 能够根据比赛场地需求，选择合适的循线策略。
- ★ 能够利用循线传感器完成循线挑战赛，通过测试与调试不断优化循线效果。

学会驱动小车后，同学们可以进行小车竞速比赛了。例如，可以规划出一条路线，约定能沿着路线最先到达终点的小车即获胜。为赢得比赛，就要找到有效的循线方法。

要完成循线挑战赛，首先要考虑如何自动“得知”路线的变化以及如何根据路线的变化改变行驶方向。要解决这些问题，需要完成以下三个任务：循线挑战赛项目规划、循线挑战赛的实现、循线挑战赛的测试与优化。



### 任务一 循线挑战赛项目规划

#### ※ 活动1 循线挑战赛项目调研

实际道路的路线有S形、T形等不同类型。请同学们利用网络查找常见的路线类型，并分析它们的路线特点，完成表2.9.1。



机器人循线赛在青少年机器人比赛中占有很大比重，如国际奥林匹克机器人大赛中的常规赛、国际青少年机器人世界杯比赛中的搜救、中国科协的综合技能比赛、RoboRAVE国际机器人大赛中的循线挑战赛等。



表2.9.1 常见的路线调查表

序号	路线类型	示意图	路线特点
1			
2			
3			
4			

可见，每一种路线都各有特点。现在，假定比赛路线如图2.9.1所示，若小车从左侧直道出发，沿黑色轨迹线行至弯道终点即挑战成功。



图2.9.1 循线挑战路线

要完成该循线任务，主要需考虑路径识别传感器的类型和数量。其中，使用传感器的数量是关键。

### ※ 活动2 循线挑战赛的可选方案分析与设计

结合活动1的调研，可以从传感器类型和数量两个角度出发设计循线小车，初步形成几种方案，如表2.9.2所示。

表2.9.2 循线挑战赛的可选方案

类别	方案	方案描述
传感器类型	方案一	使用数字循线传感器检测地面及其上面的黑色轨迹线，然后根据检测结果决定是否调整行驶路径
	方案二	使用模拟灰度传感器检测地面及其上面的黑色轨迹线，然后根据检测结果决定是否调整行驶路径
传感器数量	方案三	使用一个传感器进行路线检测
	方案四	使用两个传感器进行路线检测
	方案五	使用三个或三个以上的传感器进行路线检测

每个方案各有利弊：从路线的复杂度看，本节的循线任务路线较为简单，选用方案三和方案四等较为简单的路径检测策略即可，而方案五更适用于复杂路线；从检测的精确度看，方案四和方案五效果较好，而方案三则不够精确；从路线的可辨别性看，本任务预设的轨迹线和地面颜色有明显差别，方案一的检测方法即可实现，而方案二需要分别检测轨迹线和地面的灰度值。因此，在综合考虑学习难度、课堂时间和精确性等因素后，可以选择方案一和方案四

作为最终实施方案。

要实现以上方案，关键在于如何根据检测结果调整行驶方向。这又涉及两个问题，一是路线信息的自动检测，二是检测结果与行驶方向的对应关系。请同学们根据上面的描述，补充完整表 2.9.3 所示的设计方案。

表 2.9.3 循线挑战赛的设计方案

作品名称	循线挑战赛
作品功能	利用循线传感器使小车完成循线任务
器材清单	Arduino 控制板、USB 数据线、循线传感器、直流电机、小车
运行流程	
实现过程	<ol style="list-style-type: none"> <li>① 将两个循线传感器分别安装在小车车头的两侧</li> <li>② 编写程序，完成循线任务</li> <li>③ 上传程序，测试效果</li> </ol>

## ● 循线传感器

红外线在不同颜色的物体表面具有不同的反射强度，循线传感器就是利用红外线的反射特性进行工作的：在小车行驶过程中循线传感器的发射管不断向地面发射红外光；当红外光遇到白色时发生漫反射，反射光被循线传感器的接收管接收；如果遇到黑色则红外光被吸收，接收管就接收不到红外光，表示小车偏离了路线，据此就可以调整小车的行进方向。

图 2.9.2 所示为将要用到的循线传感器，它是数字输入传感器，其主要参数如表 2.9.4 所示。当检测到黑色时，其返回值为 0；当检测到



图 2.9.2 循线传感器

白色时，其返回值为 1。循线传感器有三个引脚，除了 VCC 和 GND 之外，Signal 口可连接数字引脚 2~13。

表2.9.4 循线传感器的主要参数

使用电压	静态电流	电平输出	探测距离
3V~5V	小于10mA	高/低	1cm~2cm



## 任务二 循线挑战赛的实现

### ※ 活动1 循线挑战赛的硬件搭建

准备好器材后，请同学们参考图2.9.3将左侧电机接在M1上，右侧电机接在M2上；左右两个循线传感器分别连接在数字引脚2、3上。为了保证小车在黑线正上方，左右两个循线传感器分别安装在车头的两侧，且两者间距离应大于黑线的宽度。

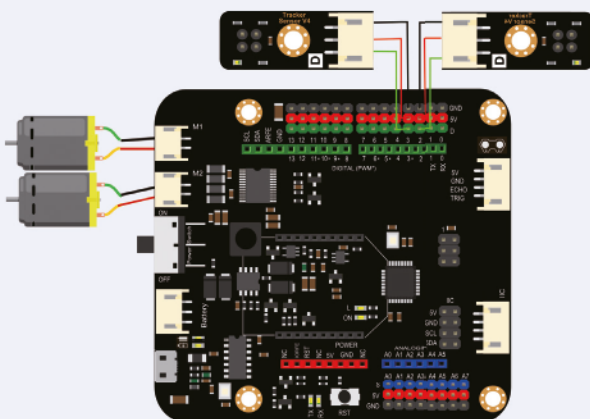


图2.9.3 硬件连接图

### ※ 活动2 循线挑战赛的程序编写

如果把小车放在赛道上进行观察，会发现若小车沿着路线行走，两个循线传感器正好一左一右位于路线两侧，而偏离路线时两个循线传感器相对于路线的位置则会发生变化。因此，可以通过分析两个循线传感器在路线上的位置分布来进行路径检测与调整。

### ● 路径检测策略

小车在循线的过程中，左右两个传感器会出现以下检测情况（图中红色圆点代表左右两个循线传感器，小车从左往右循线前进）。

（1）左右两个循线传感器都检测到白色，说明小车在黑线的上方，可以直走，如图2.9.4所示。



图2.9.4 小车需直走

(2) 左侧循线传感器检测到了黑色, 右侧循线传感器检测到了白色, 说明小车在前进的过程中, 向右偏转过度, 需要向左转弯以做调整, 如图 2.9.5 所示。



图 2.9.5 小车需左转

(3) 右侧循线传感器检测到了黑色, 左侧循线传感器检测到了白色, 说明小车在前进的过程中, 向左偏转过度, 需要向右转弯以做调整, 如图 2.9.6 所示。



图 2.9.6 小车需右转

请同学们根据上述知识, 编写小车向左转、向右转和直行的程序。把小车放在左侧直道上 (车头要刚刚越过起始位置), 观察其能否沿黑色轨迹线行驶。



### 任务三 循线挑战赛的测试与优化

#### ※ 活动 1 弯道循线策略

通过观察小车的循线效果, 会发现其弯道循线能力较差, 会出现跑出轨迹线的情况。要使小车沿直道和弯道均能行驶, 需要根据实际路线调整小车的双轮差速。请同学们根据实际路况进行调试, 确保小车能完成循线任务。

此外, 还会发现小车到达终点后不能自动停止, 需要添加使小车停下来的程序, 由于小车到达终点时两侧的传感器均检测到黑色, 因此可以用 `(!digitalRead(2))&&!digitalRead(3)` 进行条件判断。请同学们根据上述思路优化程序。

#### ※ 活动 2 进一步测试与修改循线程序

根据自己的理解和需求, 添加或修改既有循线的设计方案, 使循线任务完成时间更短、循线更准确。



## 作品发布与评价

参考表2.9.5开展作品质量的自评、互评等，与同学分享自己的制作经验。

表2.9.5 作品评价表

评价指标	得分
1. 能够实现小车直道循线（2分）	
2. 能够实现小车循线转弯（2分）	
3. 能够实现小车循线通过复杂路口，如T形路口或十字路口（3分）	
4. 能够快速完成循线任务（3分）	
总分（10分）	



## 拓展练习

请利用循线传感器和前面学过的知识完成以下循线任务。同学们在搭建小车时，可以增加其他输入输出模块，设计更有创意的小车。

1. S形路线，小车从  $a$  点行驶到  $b$  点，如图 2.9.7 所示。
2. 十字形路线，小车的循线轨迹为  $a \rightarrow b \rightarrow c \rightarrow d \rightarrow a$ ，如图 2.9.8 所示。

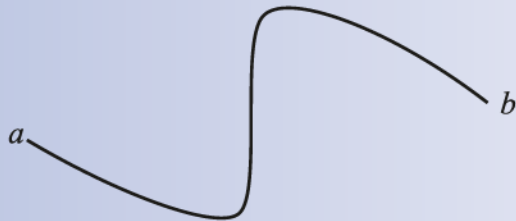


图 2.9.7 S形路线

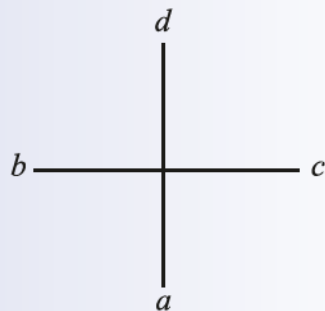
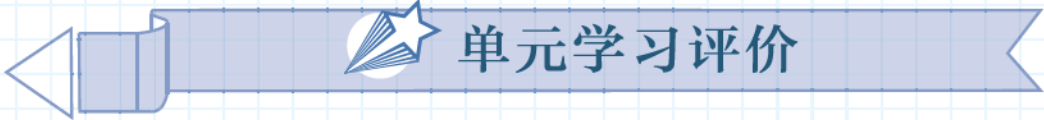


图 2.9.8 十字形路线

当然，同学们也可以绘制其他的循线路径。请进行小组合作，讨论确定主题，形成具体设计方案后填写附录 1 所示的项目设计方案，并尝试实现。



## 单元学习评价

通过本单元的学习，我们体验了实验模拟型项目的设计与制作，认识了数字、模拟、输入、输出等相关概念。你是否已经都掌握了呢？请参与小组交流、讨论并反思，开展小组评价或自我评价。

1. 请上网查找一项智能家居的应用案例，对其进行简要的分析。

简要描述一下该智能家居的应用案例：\_\_\_\_\_

该应用案例中的输出设备有\_\_\_\_\_，属于\_\_\_\_\_（数字/模拟）型设备；该应用案例中的输入设备有\_\_\_\_\_，属于\_\_\_\_\_（数字/模拟）型设备。

2. 请结合智能家居的应用案例，查看自己家里的物品能否改成自动化控制，根据所学知识提出设计或改进方案。

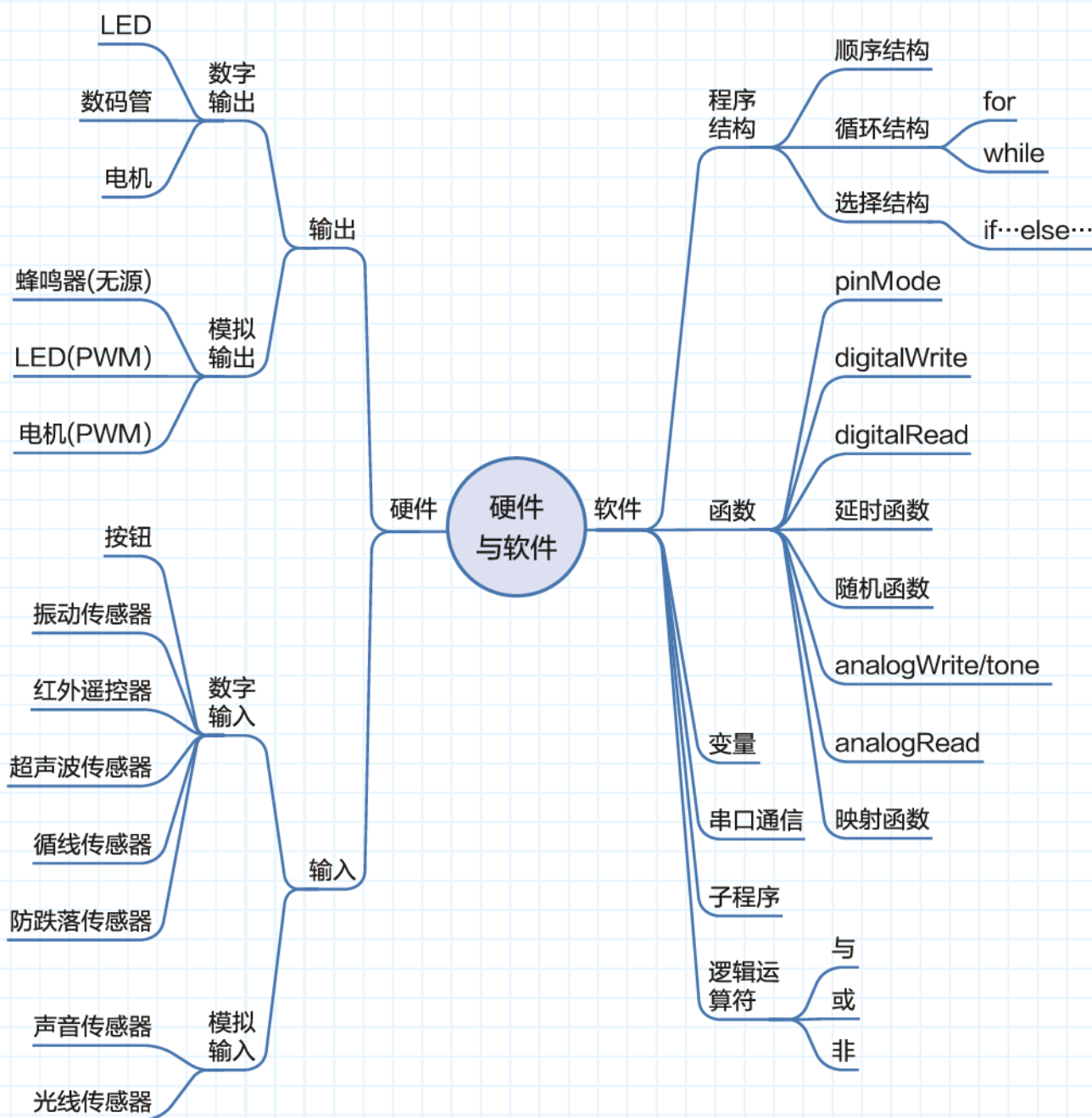
方案设计表

作品名称	
作品功能	
器材清单	
运行流程	
实现过程	

硬件搭建（通过连接图或者文字进行描述）：

程序编写：

# 单元学习总结



## 第 3 单元 科学探究型项目设计

在掌握开源硬件的基础知识后，本单元同学们将开始利用开源硬件进行科学探究。

本单元共包含 3 节，每节一个探究项目，同学们将利用传感器探究物理、化学等科学知识。每节的“拓展练习”环节还提供了若干选题，以启发同学们扩大探究范围。这些探究项目从生活中的科学问题出发，利用开源硬件收集数据，利用计算机程序或其他数据统计软件分析数据，帮助同学们在获得开源硬件知识的同时提高科学探究能力，并巩固科学知识。因此，本单元的核心价值主要表现为培养大家利用开源硬件开展科学探究的意识并掌握一些探究方法。

本单元内容是按照探究项目展开的，每一节均涉及科学探究的一般过程，首先提出问题，接着设计实验、收集数据，最后分析数据、得出结论。每个项目分别侧重对某种传感器的使用。





## 3.1 水温监测

Arduino 不仅可以用来设计和制作独立的实体机器人，还可以用来开发联机系统以开展科学探究实验。例如，利用防水温度传感器可以实时监测水温变化，也可以用来探究不同材质、厚度和口径的容器的保温效果等。



### 学习目标

- ★ 掌握防水温度传感器的使用方法。
- ★ 能够搭建物理装置监控水温变化，并运用于科学探究。
- ★ 了解科学探究的一般过程和方法。

很多同学都会带水杯来上课，谁的杯子保温效果更好呢？可以通过实验探究不同杯子的保温效果。

要探究不同杯子的保温效果，首先需要思考以下问题：如何才能“得知”不同杯子中的水温变化情况？探究过程如何设计？如何收集和分析数据？要解决这些问题，需要完成以下三个任务：形成实验假设与设计实验、进行实验探究与收集数据、分析实验数据与得出结论。



### 任务一 形成实验假设与设计实验

#### ※ 活动1 不同杯子的保温效果调研及实验假设

生活中杯子种类繁多，其材质、容量和形状各异。为了进一步了解杯子的保温效果，请同学们调查一下本组同学常用的杯子，看看这些杯子有哪些不同的类别，主观感受的保温效果如何，并完成表3.1.1。

表3.1.1 不同杯子的保温效果调研

序号	杯子的所有人	杯子的种类	材质	主观感受的保温效果
1	阿丢	陶瓷杯	陶瓷	中
2				
3				
4				

根据常识，将热水倒入杯中后，相比塑料杯，陶瓷杯很快就会变得烫手。根据学过的物理知识推理，塑料的导热性可能比陶瓷差。因此，可以假设“与陶瓷杯相比，塑料杯的保温效果更好”。这个假设是否成立？请同学们通过科学实验来探究。

### ※ 活动2 不同杯子的保温效果探究实验设计

根据前面的描述与实验假设，水温数据的获取和温度的显示是关键，前者可以用防水温度传感器实现，后者可以通过串口监视器或数码管显示。获取的温度数据还需要记录下来以便于分析，这又涉及温度读取的周期。因此，可以从温度的显示和温度获取的周期两个角度设计实验装置，形成多种方案，如表3.1.2所示。

表3.1.2 探究不同杯子的保温效果的可选方案

类别	方案	方案描述
温度显示方式	方案一	使用串口监视器显示温度
	方案二	使用四位数码管显示温度
温度读取周期	方案三	实时监测温度，不断读取温度数据
	方案四	每隔几分钟读取一次温度数据

每个方案各有利弊：从学习内容的难度看，用串口监视器显示温度数据很容易实现，记录数据时，只需从串口监视器复制温度数据到表格中即可；使用四位数码管显示温度可便于查看，但显示温度的程序较复杂，且需要手工记录温度数据。对于温度读取的周期，因水温变化速度较慢，对本实验而言，不断读取实时温度得到的数据量大且无必要。从成本的角度看，使用四位数码管会增加成本。因此，在串口监视器中显示防水温度传感器获取的温度数据，并每隔几分钟读取一次温度是较为可行的实验方案。

为了实现上述方案，可以从以下几个方面设计实验方案，如表3.1.3所示，请同学们检查一下具体实验方案的合理性。



温度的显示直接来源于防水温度传感器的电信号，因此本实验建构的测量系统误差主要由防水温度传感器自身的误差决定，不需要另做标定。



出于安全考虑，热水的温度不要超过70℃，且水杯尽量不要倒满，以免热水溢出烫伤。



水温数据记录的次数和间隔时间可根据实际情况进行调整，原则上是能得到较明显的水温下降数据即可。

表3.1.3 探究不同杯子的保温效果的实验方案

需解决的问题	实验设计内容
水温测量	使用防水温度传感器，因此要掌握防水温度传感器的使用方法
控制变量和操作性定义	自变量为塑料杯和陶瓷杯，因变量为保温效果。控制无关变量，比如，初始水温应相同，在陶瓷杯和塑料杯中应倒入等量的水，以及两个杯子的杯形、容量应大体一致等
实验过程	将70℃左右的水分别倒入两个相同纸杯至相同高度，以获取等量的水
	将两个纸杯中的水分别倒入陶瓷杯和塑料杯中，并记录初始水温
	使用防水温度传感器每隔5分钟测量水杯中的水温，并记录数据，每个杯子测量5次后停止
数据收集与结论	通过利用Excel绘制图表的方式，对采集的数据进行处理，比较陶瓷杯和塑料杯的保温效果

要实施上述实验方案，首先需要设计一个水温监测装置。根据前面的分析，形成的水温监测方案如表3.1.4所示，请同学们将程序的运行流程补充完整。

表3.1.4 水温监测方案

作品名称	水温监测
作品功能	获取和显示水温
器材清单	Arduino控制板、USB数据线、2个防水温度传感器、2个转接器、1个陶瓷杯、1个塑料杯、2个纸杯（转移热水用）、70℃的水、秒表、记录表
运行流程	
实现过程	<ol style="list-style-type: none"> <li>① 防水温度传感器与Arduino控制板连接</li> <li>② 编写程序，获取和显示水温数据</li> <li>③ 上传程序，测试效果</li> </ol>

## ● 防水温度传感器

防水温度传感器可以用于直接浸入水中测量水温。本节使用的是 DS18B20 防水温度传感器，为数字传感器，如图 3.1.1 所示，其参数如表 3.1.5 所示。从图中可以看出，DS18B20 防水温度传感器是三线接口，分别为红线（VCC）、灰线（GND）、黄线（DATA）。



图 3.1.1 防水温度传感器


 防水温度传感器内部使用多个二进制位存储温度数据，其存储的温度数据可以转换为模拟量，而不是 0 或 1。

表 3.1.5 防水温度传感器参数表

使用电压	温度测量范围	温度数据读取	反应时间
3V~5.5V	-10°C~85°C (误差 ±0.5°C)	9~12 位 (可配置)	小于 750ms

DS18B20 在使用时还需连接一个转接器，本书采用了如图 3.1.2 所示的转接器，它可以接数字输入模块和模拟输入模块。通过该转接器，能够让非标准元件简单地连接到 Arduino 控制板，实现即插即用的效果。与其他传感器连接时，需注意 A 为信号口、B 为 VCC、C 为 GND。

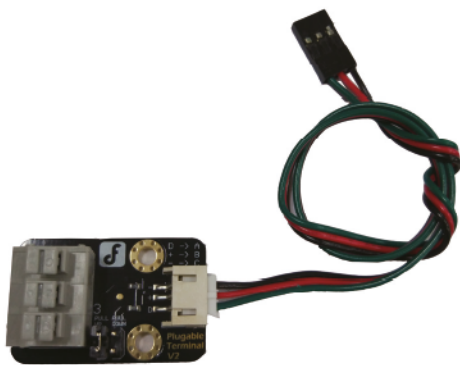



图 3.1.2 防水温度传感器转接器

 这里用到的三线接口为 A 型接口，还有一种接口为 B 型接口，B 型接口分别为红线（VCC）、黄线（GND）、绿线（DATA）。

## 任务二 进行实验探究与收集数据

### ※ 活动 1 探究不同杯子的保温效果的硬件搭建

准备好器材后，请同学们参考图 3.1.3 将防水温度传感器连接在转接器上，转接器接在数字引脚 2 上。

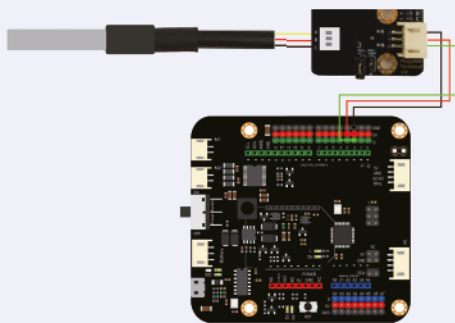


图 3.1.3 水温监测硬件连接图

### ※ 活动2 探究不同杯子的保温效果的程序编写

将水温监测装置的硬件连接好以后，需要编写程序获取和显示水温。水温监测样例程序如下，请同学们测试该程序，看能否正确显示水温。

```
#include <OneWire.h>    //引用单总线OneWire库函数
#include <DS18B20.h>    //引用能获取DS18B20传感器检测温度的库函数
//定义DS18B20数据口接在2号引脚上，并以单总线的方式传输数据
OneWire DS18B20_pin_2(2);
void setup()
{
  Serial.begin(9600);    //打开串口
}
void loop()
{
  Serial.println(getTemperature(DS18B20_pin_2));    //串口输出温度
  delay(1000);
}
```

### ● OneWire 库

单总线(One-Wire)是一种数据传输的时序，它采用单根信号线实现数据的双向传输。单总线数据传输线路简单，能够节约输入输出(I/O)端口资源，便于扩展和维护，适用于单主机(单控制板)系统。

OneWire库是针对单总线设计的扩展库，DS18B20温度传感器与控制板连接时，就是使用一条线实现两者的双向通信，因此在Arduino中编写程序时，需调用OneWire库函数。

### ● DS18B20 库

在DS18B20库中，getTemperature()函数用来读取传感器获取的温度数据，参数为传感器连接的引脚。

### ※ 活动3 探究不同杯子的保温效果的数据收集

请同学们利用自己制作的实验装置，收集实验数据，完成表3.1.6。

表 3.1.6 不同杯子保温效果的数据记录表

杯子的种类	初始温度	5分钟	10分钟	15分钟	20分钟



### 任务三 分析实验数据与得出结论

#### ※ 活动 不同杯子保温效果的数据分析与结论

经过实验得到的两组样例数据如表 3.1.7 所示。为了进一步分析这些数据，可以将数据输入 Excel 中，生成塑料杯和陶瓷杯的保温效果比较图，如图 3.1.4 所示。

表 3.1.7 不同杯子保温效果的实验数据

杯子的种类	初始温度	5分钟	10分钟	15分钟	20分钟
塑料杯	71.9	67.1	62.8	59.4	56.6
陶瓷杯	71.4	64.8	59.8	55.7	51.6



图 3.1.4 塑料杯和陶瓷杯的保温效果比较

从图 3.1.4 可以看出，本实验中陶瓷杯的水温下降比塑料杯快，因此，可以得出结论：塑料杯比陶瓷杯的保温效果好。这一结论支持了之前的假设。同学们得出的结论又是怎样的呢？

需要注意的是，在本实验中虽然对杯子的形状及容量进行了控制，但没有考虑塑料杯和陶瓷杯的厚度，如果使用不同厚度的塑料杯或陶瓷杯进行实验，得出的结论可能会不一样。



使用防水温度传感器测量水温时，为防止探头浸入水中的深度不同可能导致测量结果不同，要将金属探头全部浸入水中。



利用机器人开展实验探究，其意义在于通过收集数据和分析数据得出结论，从而验证假设。因此，如果实验得出的结论与假设不一致甚至相反，都是可能的。重要的是要对假设与结论相异的原因进行分析。



## 作品发布与评价

参考表3.1.8开展作品质量的自评、互评等，与同学分享自己的制作经验。

表3.1.8 作品评价表

评价指标	得分
1. 实验装置能准确监测水温的变化（4分）	
2. 能用串口监视器查看水温（3分）	
3. 实验操作符合规范（2分）	
4. 实验装置设计实用、美观（1分）	
总分（10分）	



## 拓展练习

同学们还有哪些与温度有关的疑惑吗？可以选择一个感兴趣的问题，综合运用防水温度传感器和之前所学的知识进行实验设计、问题探究。可参考以下项目主题。

例如：根据本节内容，不同塑料杯和陶瓷杯的保温效果可能不同，甚至同一材料的杯子，如果壁厚、杯形、容量等条件不同，杯子的保温效果也不同。因此，同学们可以设计实验进一步探究这些问题。

当然，同学们也可以选择其他的项目主题。请进行小组合作，讨论并确定主题，形成具体设计方案后填写附录1所示的项目设计方案，再尝试实现。

## 3.2 单摆监测

Arduino 不仅能够用来探究杯子的保温效果，还可以用来测量单摆振动的周期。不同于一般的物理实验中人工计时计数的方式，用 Arduino 探究单摆的周期，可以使用红外数字避障传感器监测单摆的振动数据，从而得到单摆的振动周期。



### 学习目标

- ★ 掌握红外避障传感器的使用方法。
- ★ 能够自主设计验证单摆周期的实验方案，并搭建实验装置。
- ★ 能够编程收集实验数据，完成探究实验，体验科学探究的一般过程。

不同长度的钟摆来回摆一次的时间相同吗？如果它们的周期相同，有没有其他因素会影响钟摆的摆动周期呢？我们可以设计实验来解答上述问题。

钟摆属于单摆的一种，上述问题实际上就是要探究单摆的振动周期。那么，单摆的振动周期可能与哪些因素有关？如何自动获取单摆振动次数及时间？探究过程如何设计？如何收集和分析数据？要解决这些问题，需要完成如下三个任务：形成实验假设与设计实验、进行实验探究与收集数据、分析实验数据与得出结论。



### 任务一 形成实验假设与设计实验

#### ※ 活动1 单摆周期实验的假设

单摆的周期可能与单摆的摆长、振幅、摆球的质量等因素有关。因此，我们可以形成若干实验假设，请同学们将表3.2.1补充完整。



表3.2.1 探究单摆周期的实验假设

探究内容	假设
单摆周期与摆长的关系	当小球质量与振幅为定值时，单摆周期与摆长无关
	当小球质量与振幅为定值时，摆长越长单摆周期越长
	当小球质量与振幅为定值时，摆长越短单摆周期越长
单摆周期与振幅的关系	
单摆周期与摆球质量的关系	

### ※ 活动2 探究单摆周期的实验设计

根据引言描述与实验假设，要探究单摆周期，单摆的振动次数和振动时间的数据收集是关键。因此，从单摆数据收集的角度设计实验可以形成两种方案，如表3.2.2所示。

表3.2.2 记录单摆周期的可选方案

方案	方案描述
方案一	人工记录单摆的振动次数和振动时间，并计算单摆周期
方案二	使用红外数字避障传感器获取单摆的振动次数和振动时间，并利用程序自动计算单摆周期

两种方案各有利弊：人工计时计数容易造成较大的误差，利用传感器获取的振动次数和振动时间更准确，也更便于计算周期。因此，可以利用红外数字避障传感器监测单摆的振动并计算周期。

可以从以下几个方面设计实验方案，如表3.2.3所示。请同学们将实验方案补充完整。

表3.2.3 单摆振动周期探究实验方案

需解决的问题	实验设计内容
振动次数及时间的记录	使用红外数字避障传感器，因此要掌握红外数字避障传感器的使用方法
控制变量和操作性定义	自变量为单摆的摆长、振幅、摆球质量，因变量为单摆的振动周期。控制无关变量，例如，摆线应固定好且不能有弹力，避免单摆振动时线滑动或摆长改变；小球振动时要在同一个铅直平面内等



若小球只限于在铅直平面内振动，则为平面单摆；若小球的振动不限于铅直平面，则为球面单摆。本节实验中所涉及的单摆默认为平面单摆。

续表

需解决的问题	实验设计内容	
实验过程	探究单摆周期与摆长的关系	首先将摆长设置为 40 cm，将单摆支架与摆线、单摆小球连接好；将单摆拉起，利用量角器将摆角设定为 $5^\circ$ ，将小球无初速度释放，记录单摆振动 40 次（20 个振动周期）所需的时间；将上一步的操作重复三次，求周期的平均值作为单摆周期。之后分别将摆长设定为 30 cm 和 20 cm，不改变小球的质量和摆角，测量单摆振动的周期
	探究单摆周期与振幅的关系	
	探究单摆周期与摆球质量的关系	
数据收集与结论	利用 Arduino 编程控制红外数字避障传感器获取单摆振动次数和振动时间，计算单摆周期，验证或证伪假设	



为了减小实验数据的误差，重复测量三次单摆周期，取其平均值作为最终的数据结果。

要实现上述实验方案，首先需要制作一个单摆监测装置。根据前面的分析，单摆监测方案如表 3.2.4 所示，请同学们将程序的运行流程补充完整。

表 3.2.4 单摆监测装置设计方案

作品名称	单摆监测装置
作品功能	记录单摆的振动次数和时间，计算单摆的振动周期
器材清单	Arduino 控制板、USB 数据线、红外数字避障传感器、单摆支架、长为 50 cm 的摆线、三个质量不同的小球、量角器、量尺
运行流程	
实现过程	<ol style="list-style-type: none"> <li>① 将红外数字避障传感器与 Arduino 控制板连接，将红外数字避障传感器对准小球振动时最低点的位置</li> <li>② 编写程序，监测单摆的振动并计算振动周期</li> <li>③ 上传程序，测试效果</li> </ol>

## ● 红外数字避障传感器

红外数字避障传感器也称红外接近开关，是一种集发射与接收于一体的数字光电传感器，当检测到障碍物时输出低电平，无障碍

物时输出高电平。红外数字避障传感器自带3P接线，分别为红线（5V）、绿线（GND）、黄线（DATA），该传感器背面有一个电位器，可以调节检测距离，如图3.2.1所示，其参数如表3.2.5所示。

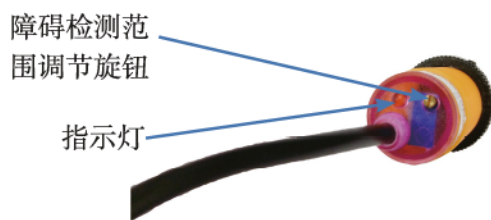


图3.2.1 红外数字避障传感器

表3.2.5 红外数字避障传感器参数表

使用电压	障碍检测范围（可调）	标准检测物体	响应时间
5V	指向角： $\leq 15^\circ$ ； 有效距离：3cm~80cm	太阳光10000lx以下； 白炽灯3000lx以下	$< 2\text{ms}$



## 任务二 进行实验探究与收集数据

### ※ 活动1 探究单摆周期的硬件搭建

准备好器材后，将传感器连接在数字引脚2上。

### ※ 活动2 探究单摆周期的程序编写

下面是获取单摆数据的程序，请同学们测试程序，看能否正确获取单摆周期。

```
int n;
double T1;
double T2;
void setup()
{
  Serial.begin(9600);
  pinMode (2,INPUT);
}
void loop()
{
  //定义单摆振动次数为n
  //定义单摆振动开始的时间为T1
  //定义单摆振动结束的时间为T2
  //打开串口
  //初始化2号引脚为输入模式
```

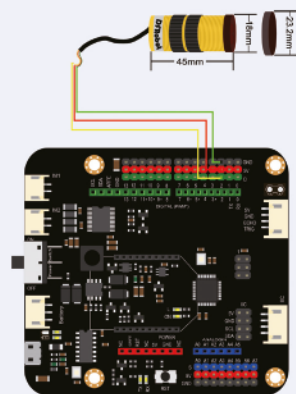


图3.2.2 单摆监测装置的硬件连接图

```

if (!digitalRead(2))          //如果2号引脚上的传感器检测到小球
{
    //检测小球直到它完全通过后才进行次数增加，避免传感器对小球重复检测
    while (!digitalRead(2))
    {
    }
    n=n+1;                    //单摆振动次数加1
}
if(n==5)                     //如果单摆振动次数为5
{
    T1=millis();             //记录振动开始时间T1
}
if(n==45)                    //如果单摆振动次数为45
{
    T2=millis();             //记录振动结束时间T2
    Serial.print("T1=");
    Serial.print(T1);        //串口输出T1
    Serial.println();
    Serial.print("T2=");
    Serial.print(T2);        //串口输出T2
    Serial.println();
    Serial.print("T=(T2-T1)/20=");
    Serial.print((T2-T1)/20); //串口输出单摆周期T=(T2-T1)/20
    Serial.println();
    while(n==45)            //避免传感器对小球重复检测
    {
    }
}
}

```



单摆刚开始振动时稳定性较差，因此需要忽略前几次的实验数据，比如从单摆第5次振动时开始计时和计数，若振动次数为45次，则实际记录为40次。



每次传感器检测到小球时，是经过了半个周期，因此计算单摆周期需要将次数除以2，即  $(45-5) \div 2 = 20$  个周期。

## ● millis()函数

调用 millis() 函数可以获取 Arduino 通电后或复位后的运行时间，单位是 ms。记录的最长时间为 9 小时 22 分，如果时间超出将会归零。需要注意的是，millis() 函数的返回值是 unsigned long 型（无符号长整型）数据，因此定义相关变量时不能使用 int 型（整型）数据。

### ※ 活动3 探究单摆周期的数据收集

请同学们利用自己制作的实验装置，收集实验数据，完成表 3.2.6。



实验操作过程中应注意：松开小球使其开始运动时，要保持小球在同一水平面内运动，且小球不能旋转；红外数字避障传感器对准小球振动时的最低点位置；实验装置要固定好。

表 3.2.6 探究单摆周期数据记录表

探究内容	摆角 ( $^{\circ}$ )	小球质量 (g)	摆长 (cm)	周期 (s)
单摆周期与摆长的关系				
单摆周期与振幅的关系				
单摆周期与摆球质量的关系				



### 任务三 分析实验数据与得出结论

#### ※ 活动 探究单摆周期的数据分析与结论

请同学们分析、整理实验数据，检查前面的假设哪些成立、哪些不成立，并总结影响单摆周期的因素。



### 作品发布与评价

参考表 3.2.7 开展作品质量的自评、互评等，与同学分享自己的制作经验。

表 3.2.7 作品评价表

评价指标	得分
1. 实验装置能准确监测单摆的周期 (5分)	
2. 实验操作符合规范 (3分)	
3. 实验装置设计实用、美观 (2分)	
总分 (10分)	




## 拓展练习

红外数字避障传感器可以测距离，同学们仔细回想一下，有哪些物理探究实验涉及距离？从中选择一个感兴趣的问题，综合运用之前所学的知识进行实验设计、问题探究。可参考以下项目主题。

例如：单摆在摆角小于  $5^\circ$  时，可近似认为是简谐运动，此时单摆运动的周期公式为  $T=2\pi\sqrt{\frac{l}{g}}$ ，其中  $l$  指摆长， $g$  是当地重力加速度。那么，我们能否利用单摆实验测出当地的重力加速度呢？同学们可以设计实验进一步探究这个问题。

当然，同学们也可以选择其他的项目主题。请进行小组合作，讨论并确定主题，形成具体设计方案后填写附录 1 所示的探究项目设计方案，再尝试实现。

## 3.3 电压检测

 物体传导电流的能力叫作导电性。水果电池属于电解质导电。电解质导电是因为电解质在溶解或熔融时产生能自由移动的离子从而具有导电性。

不同的水果具有不同的导电性，水果电池就是利用水果的导电性产生电能的装置。为进一步分析不同水果的导电性及其影响因素，需要利用实验展开探究。

### 学习目标

- ★ 掌握电压检测模块的使用方法，能够用串口输出电压值。
- ★ 了解水果电池的基本组成和原理。
- ★ 通过探究活动，体验多学科知识的综合应用和科学探究方法。

化学课上，老师曾讲过水果电池。那么，水果的导电性与什么因素有关呢？我们可以利用开源硬件进行探究，从而得出相关结论。

要制作水果电池，需要考虑以下问题：如何使水果产生电能？使用何种方式测得其导电性？如何控制干扰变量以科学比较不同水果的导电性？要解决这些问题，需要完成以下三个任务：形成实验假设与设计实验、进行实验探究与收集数据、分析实验数据与得出结论。

### 任务一 形成实验假设与设计实验

#### ※ 活动1 水果导电性影响因素调研

为了设计水果电池，有必要先调研一下水果导电性与什么因素有关，以便指导实验方案的设计。请同学们利用网络查找可能的影响因素，将调查结果填写在表3.3.1中。

表3.3.1 影响水果导电性因素的调查表

序号	影响因素	导电效果
1		
2		
3		
4		

通过调研可以发现，水果导电性的影响因素有很多，如水果种类、pH值、成熟程度等。考虑到实验时间、器材等限制，我们可以先对水果种类进行探究，同时将电极种类、电极插入深度、电极间距等当作干扰变量予以控制。根据初步调研，可以认为越酸的水果导电性越好。这一假设是否成立？请同学们通过科学实验来探究。

### ※ 活动2 水果电池的可选方案分析与设计

水果导电性的探究主要与检测设备有关，因此可从检测设备出发设计实验装置，初步形成几种方案，如表3.3.2所示。

表3.3.2 水果电池的可选方案

方案	方案描述
方案一	使用LED检测，亮度强弱代表导电性的强弱
方案二	使用电压检测模块，数值大小代表导电性的强弱
方案三	使用电流传感器模块，数值大小代表导电性的强弱

每个方案各有利弊：从测量效果的易观测性看，肉眼很难区分出LED亮度的不同，因此方案一难以得出准确结论，而方案二和方案三可直接读取实际数值，由此得出的结论更加准确；从可测量性看，由于水果电池产生的电压较低，电阻较大，因此产生的电流很小，电流变化不如电压明显；从成本看，方案一、方案二所需成本较低，而方案三所需成本较高。因此，在综合考虑以上因素后，可以选择方案二作为最终实施方案，即将检测到的电压值作为评价水果导电性的一个重要指标。

为了实现方案二，可以从以下几个方面设计实验方案，如表3.3.3所示。请同学们将表3.3.3补充完整。



**干扰变量**是指除了实验因素（自变量）以外的所有影响实验结果的变量，这些变量不是本实验所要研究的变量，所以又称无关变量、无关因子。在本实验中，如果只计划探究不同水果的导电性，那么水果就是一个实验因素（自变量），电压值是因变量，其他影响导电性的因素，如电极种类、电极插入深度、电极间距等属于干扰变量，需要控制在相同水平。



表3.3.3 探究不同水果导电性实验方案

需解决的问题	实验设计内容
电压数据的获取	使用电压检测模块检测水果电池的电压
控制变量和操作性定义	自变量为水果种类，因变量为水果电池的电压。控制无关变量，如使电极与水果的接触面积、电极间距相等
实验过程	
数据收集与结论	利用Arduino编程获取水果电池的电压值，通过分析结果来验证或证伪假设

要实现上述实验方案，首先需要搭建水果电池电路，可以选择铜和锌作为电极材料，将其插入不同水果制成电池；然后将电压检测模块与电路进行连接以测取不同水果的电压值。请根据上面的描述，将表3.3.4所示的设计方案补充完整。

表3.3.4 水果电池电压检测设计方案

作品名称	水果电池
作品功能	利用电压检测模块探究不同水果的导电性
器材清单	Arduino控制板、USB数据线、电压检测模块、铜锌电极、带鳄鱼夹导线、水果
运行流程	
实现过程	<ol style="list-style-type: none"> <li>① 将电压检测模块与Arduino控制板、水果电路连接</li> <li>② 编写程序，读取电压值</li> <li>③ 上传程序，测试效果</li> </ol>

## ● 电压检测模块

本实验中所用的电压检测模块如图 3.3.1 所示，它是一种能够感知电压大小的模拟输入传感器，此模块能使输入的电压缩小 5 倍。由于 Arduino 控制板模拟输入电压最大为 5V，故电压检测模块的输入电压不能大于  $5V \times 5 = 25V$ 。该模块下方的白色接口与 Arduino 控制板连接，其中蓝色接线与模拟口连接，正极接 VCC，负极接 GND。上方的绿色接线柱与水果电路连接，其中 GND 接电路负极，Input 接电路正极。

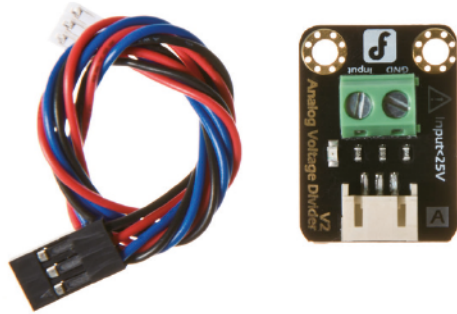


图 3.3.1 电压检测模块



## 任务二 进行实验探究与收集数据

### ※ 活动 1 水果电池电压检测的硬件搭建

准备好器材后，请同学们参考图 3.3.2 将电压检测模块连接在模拟引脚 A0 上。

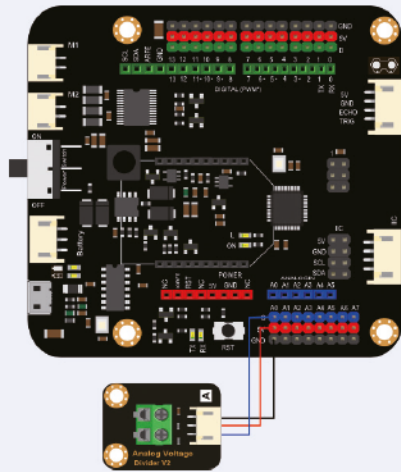


图 3.3.2 硬件连接图

### ※ 活动 2 水果电池电路设计

完成硬件搭建任务后，需要对水果电池进行设计和组装。为了获得较大的电压，还需要准备多个或多块水果，并用铜和锌电极进行串联，请同学们参考图 3.3.3 将所需水果电路连接起来。

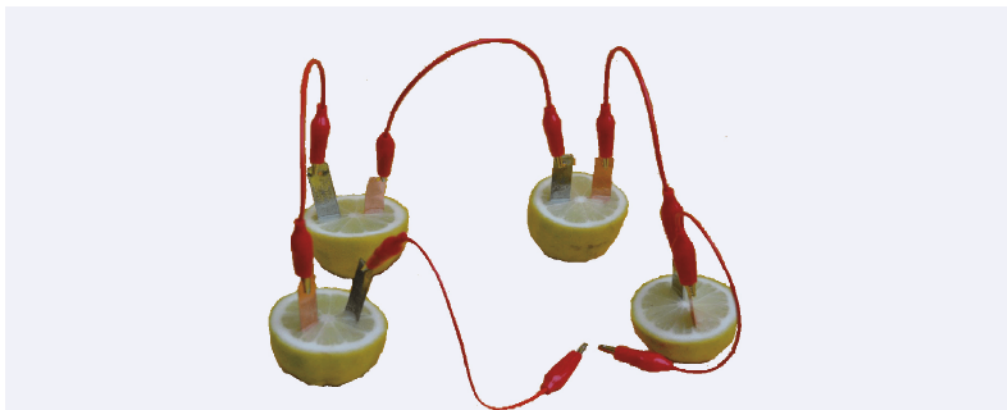


图 3.3.3 水果电路连接示例

### ※ 活动3 编写程序获取电压值

完成上述工作后，再将正极（铜片）、负极（锌片）分别与电压检测模块上绿色接线柱的 Input 和 GND 接口连接。可是如何用电压检测模块获取电路的电压值呢？

### ● 读取电压值



也可以采用 2.7 节中提到的映射函数进行转换。

Arduino 控制板中的模数转换器的精度为 10bit，能读取 2 的 10 次方（1024）个模拟状态，其能够读取的模拟量变化范围为 0~1023，而 Arduino 控制板上模拟输入引脚允许的电压变化范围是 0V~5V，因此模拟输入引脚能够感知到的最小电压变化是  $5/1024$ 。若模拟引脚输入值为  $x$ ，则可被表示为  $5x/1024$ ；此外，根据前面的描述，由于电压检测模块将输入的电压缩小了 5 倍，因此测得电路的电压值应扩大 5 倍，即  $25x/1024=x/40.96$  才是实际的电压值。

请同学们根据上述知识，编写获取电压值的程序。

### ※ 活动4 收集数据

完成以上任务后，可以选择苹果、橘子、柠檬、橙子、梨等水果展开探究。请同学们根据实际条件，选择合适的水果开展实验，填写表 3.3.5。

需要注意的是，为减小测量误差，连接电路前要将铜片和锌片擦拭干净；同时，要待串口监视器上显示的数值基本稳定后再记录数据。若发现无法读取数据，请检查正负极是否接错，或者铜和锌是否有直接接触；此外，使用不同水果做实验时，电极间距、插入深度应保持一致。

表3.3.5 数据记录表

实验序号	水果种类	电极种类	电极间距 (cm)	插入深度 (cm)	电压(V)
1	苹果	铜、锌			
2	橘子	铜、锌			
3	柠檬	铜、锌			
4	橙子	铜、锌			
5	梨	铜、锌			



水果的酸味来源于其所含的有机酸——苹果酸、柠檬酸、酒石酸等。

酸是指电离时产生的阳离子全部是氢离子的化合物。当酸的浓度越高时，电离产生的氢离子越多。



### 任务三 分析实验数据与得出结论

#### ※ 活动 分析数据并得出结论

请同学们根据实际数据进行分析，验证之前的假设。在实验结束后请同学们总结出现的问题，并在小组中分享开展实验时需要注意的事项。



### 作品发布与评价

参考表3.3.6开展探究活动的自评、互评等，与同学分享自己的制作经验。

表3.3.6 作品评价表

评价指标	得分
1. 能够正确连接电路(2分)	
2. 能够获取电压值(2分)	
3. 能够通过实验收集数据(4分)	
4. 能够分析数据并得出正确结论(2分)	
总分(10分)	



## 拓展知识

水果电池的发电原理是，两种电极材料的电化学活性不同，其中稍活泼的金属电极更容易失去电子，作为负极；另一种金属电极的活性较弱，不易失去电子，作为正极。两极之间溶液中离子的定向移动和外部导线中电子的定向移动构成了闭合回路，使两个电极反应不断进行，发生有序的电子转移过程，产生电流，实现化学能向电能的转化。

金属的活动性从高到低依次为：钾、钙、钠、镁、铝、锌、铁、锡、铅、氢、铜、汞、银、铂、金。



## 拓展练习

影响电压的因素很多，有电极材料和电极间距，以及电解液的温度、浓度、pH值等。从中选择一个感兴趣的、可以利用电压检测模块检测的因素，综合运用前面学过的知识设计实验，进行问题探究。可参考以下项目主题。

1. 探究不同电极对电压的影响：在保持水果种类、插入深度和电极间距不变的情况下，改变电极种类，探究电极与电压的关系。

2. 探究电极插入深度对电压的影响：在保持水果种类、电极种类和电极间距不变的情况下，改变插入深度，探究电极插入深度与电压的关系。

当然，同学们也可以选择其他的项目主题。请进行小组合作，讨论确定主题，形成具体设计方案后填写附录1所示的项目设计方案，并通过小组合作尝试实现。



## 单元学习评价

通过本单元的学习，我们掌握了科学探究的一般方法，体验了科学探究的一般过程，形成了利用开源硬件开展科学探究以解决问题的意识。现在，你是否具备开展或评价科学探究活动的能力？请参与小组交流并反思，开展自评或互评。

1. 本单元采用的科学探究过程包括\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_。

2. 利用开源硬件开展科学探究有何优缺点？

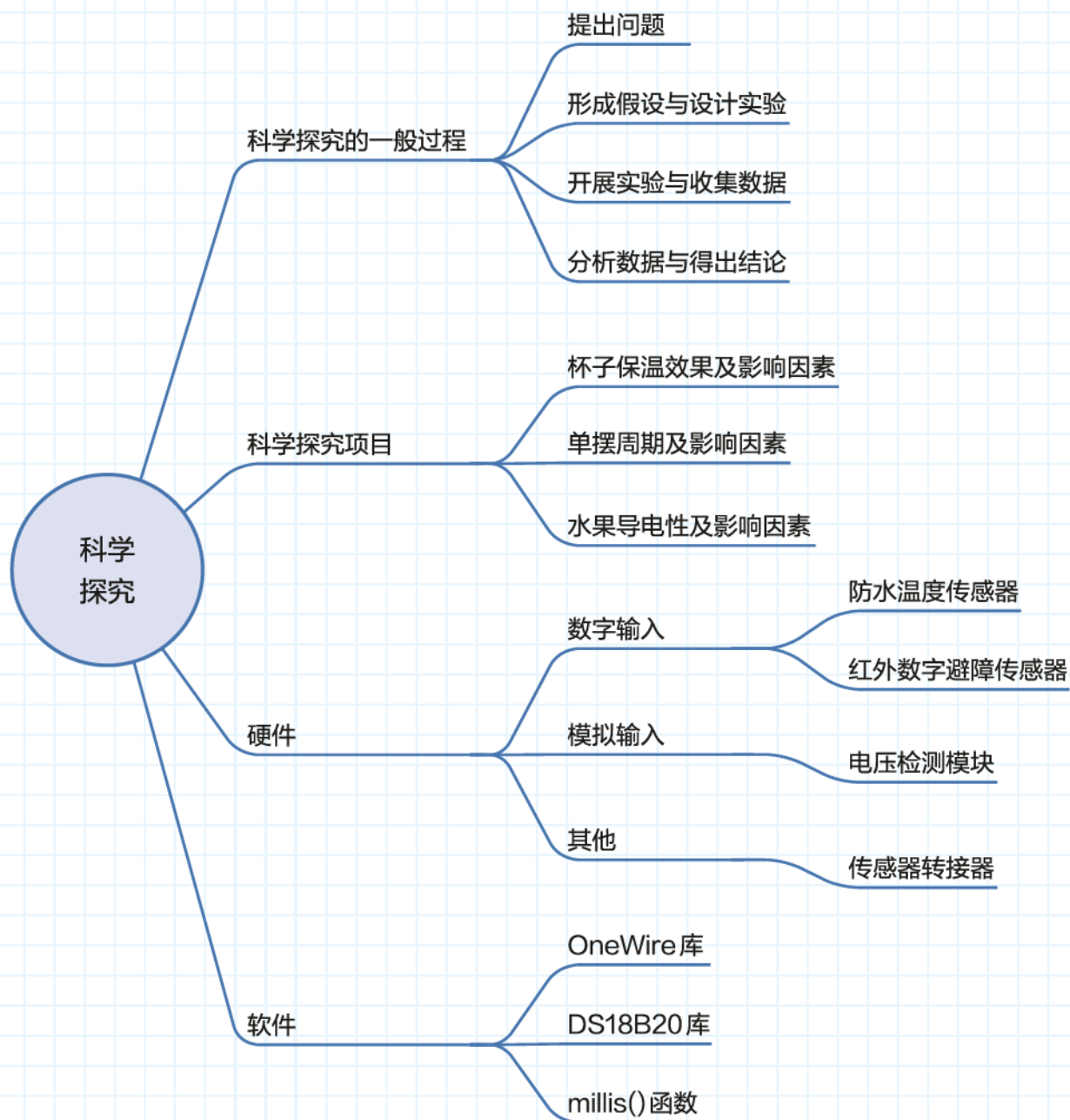
3. 上网搜索一个利用开源硬件开展科学探究的案例，对案例进行评价，重点评价方案的可行性以及探究过程的科学性，指出案例中存在的缺陷并提出改进措施。

4. 结合本单元的学习内容，利用开源硬件设计并制作一套探究小车匀变速直线运动的实验装置，实现无须手动测量及计算，便可通过计算机得到所需数据及小车运动的 $v-t$ 图像。形成具体设计方案后填写附录1所示的探究项目设计方案，再尝试实现。探究完成后参考评价要点开展自评或互评。

评价要点：

- (1) 能否形成小车匀变速直线运动实验的假设并设计实验方案。
- (2) 能否根据实验方案搭建实验装置，并编写程序实现实验数据的测量和记录。
- (3) 能否利用实验装置进行实验探究，实时获取小车在斜面运动过程中的数据。
- (4) 能否导出实验数据，利用数据统计软件直观地分析小车速度随时间变化的规律，得出结论。

# 单元学习总结



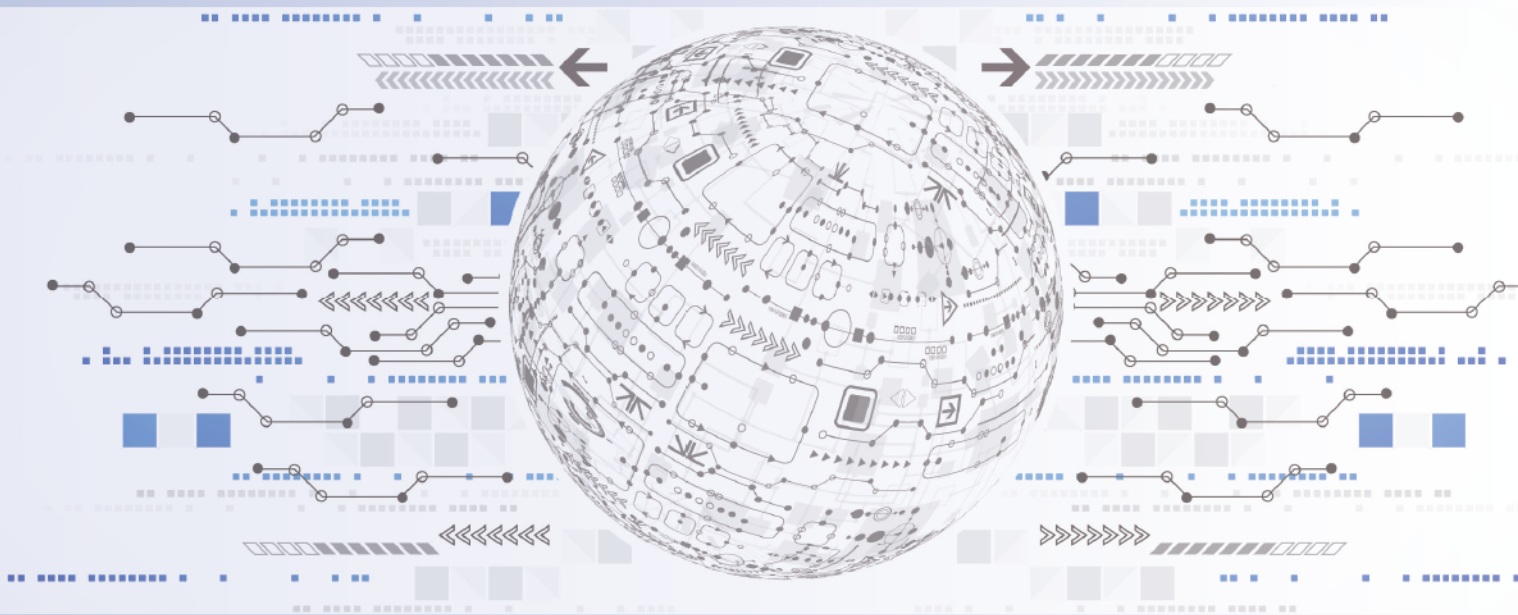
## 第 4 单元 趣味交互型项目设计

通过前面的学习，我们掌握了开源硬件的基础知识和基本技能，并初步尝试了如何利用开源硬件开展科学探究，积累了较为丰富的实践经验。本单元将带领同学们利用开源硬件开发一类新的作品——趣味交互型联机系统。

所谓趣味交互型联机系统，是指在计算机中编写和运行程序，将开源硬件作为控制端，通过开源硬件与计算机程序进行交互，而非采用鼠标与键盘交互。本单元所采用的计算机程序端使用了一种图形化的编程软件——S4A，它是一款专门针对 Arduino 的 Scratch 编程环境。

本单元共包含四节，每节包含一个趣味交互项目，分别呈现了模拟仿真、数字故事和交互游戏的开发过程；每节的“拓展练习”环节提供了若干拓展项目选题，以进一步激发同学们的创意。

本单元内容按照作品的交互复杂程度展开。希望同学们能通过设计和制作有趣、好玩的联机交互作品，进一步感受学习的快乐，深深爱上开源硬件项目设计。





## 4.1 造型切换



Scratch是由美国麻省理工学院设计开发的一款面向青少年的简易编程工具，它将编程指令以图形化的方式封装起来，通过拼搭积木形状的程序模块来完成程序的编写。

S4A是“Scratch for Arduino”的简称，是一款基于Scratch衍生的二次开发软件，具有Scratch图形化程序设计的特点和功能，并且提供了对Arduino的支持。使用S4A能够非常方便地设计、制作出有趣、好玩的机器人交互系统。其中，造型切换是S4A最常用、最基本的功能。



### 学习目标

- ★ 了解S4A的界面结构及其功能，掌握S4A与Arduino控制板通信的方法，以及查看、读取传感器输入值的方法。
- ★ 掌握造型切换和按钮消抖的方法，感受图形化编程环境的便捷性。
- ★ 能够结合按钮模拟电路的控制效果，感受联机作品的交互特性。

物理课上，老师组装实物电路，向同学们演示电路中各电子元件的关系。学习了前面的课程后，我们也可以利用按钮和LED灯模拟电路中的开关效果。

要模拟电路的开关效果，需要思考在S4A界面中如何表现电路的断开与闭合、如何获取控制信息并切换电路的状态。要解决这些问题，需要完成以下三个任务：控制电路项目规划、控制电路的实现、控制电路的测试与优化。



### 任务一 控制电路项目规划

#### ※ 活动1 控制电路项目调研

电路有不同的类型，且其复杂程度各不相同，因此首先需要确定模拟的电路类型。请同学们根据物理课中学到的知识开展讨论，填写表4.1.1。

表 4.1.1 电路类型调查表

序号	电路类型	基本电路图
1		
2		
3		
4		

为降低操作难度，可以先做一个最简单的串联电路，由电池、开关和灯泡组成，其基本功能为：当 Arduino 端的触发装置被触发后，S4A 端的电路实现闭合与断开。

制作该作品的基本思路为：从作品功能看，其主要由 Arduino 端的触发装置和 S4A 端的界面组成，当电路被触发后，电路进行状态切换；从触发方式看，可使用按钮或碰撞传感器来实现用户动作的响应。

## ● S4A 界面

S4A 界面由菜单栏、工具栏、控制按钮、舞台区、角色区、模块区和脚本区等几部分组成，如图 4.1.1 所示。

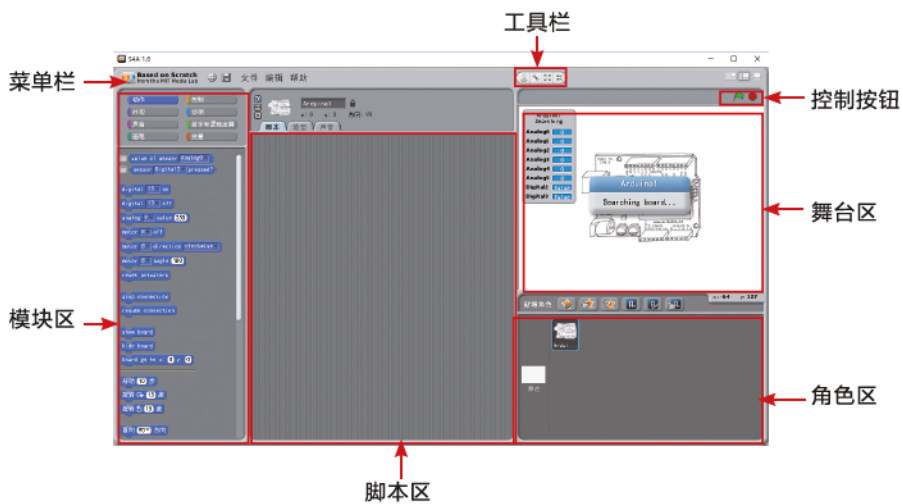


图 4.1.1 S4A 界面

工具栏：复制、剪切、放大和缩小舞台中的角色。


控制按钮：启动与停止程序。

舞台区：显示案例中的角色及传感器信息显示面板。

角色区：编辑案例中的角色对象。

模块区：显示各类型的程序模块。

脚本区：程序模块编辑区域。

 传感器信息显示面板显示模拟引脚0~5和数字引脚2和3的输入值。当不需要其显示在舞台中时，可右击面板，选择“隐藏”选项。同时，可右击控制板角色，选择“show sensor board”命令使其显示。

## ● S4A 与 Arduino 之间的通信

打开S4A后，舞台中央显示如图4.1.2所示的“Searching board...”提示框，此时S4A与Arduino间无法通信。为了实现两者间的通信，可为Arduino控制板上传固件“S4AFirmware.ino”，再次打开S4A后提示框消失，舞台中的传感器信息显示面板显示相应引脚的输入值。

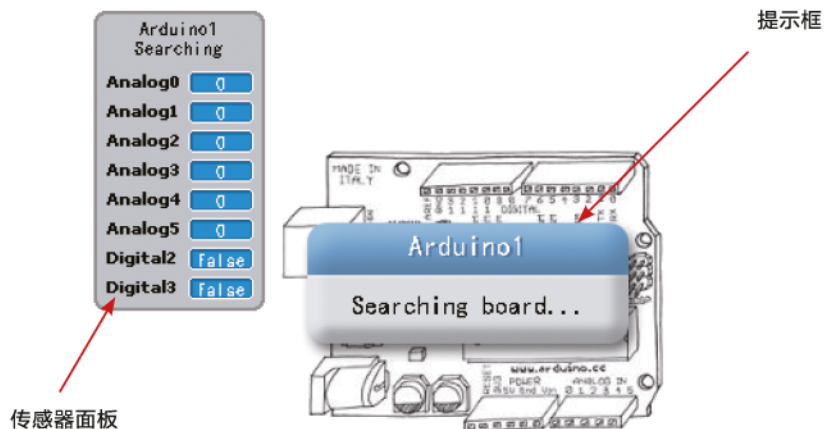


图 4.1.2 舞台区的信息显示

### ※ 活动2 控制电路的可选方案分析与设计

根据制作思路，可以从触发方式和角色设置两个角度模拟电路控制，初步形成几种方案，如表4.1.2所示。

表 4.1.2 控制电路的可选方案

类别	方案	方案描述
触发方式	方案一	使用按钮作为系统的触发装置
	方案二	使用碰撞传感器作为系统的触发装置
角色设置	方案三	设置“电路”角色，并为其编辑开路与断路两种造型，当电路被触发后切换角色的造型
	方案四	设置“开关”角色，当电路被触发后开关角色旋转一定的角度，实现电路的断开与闭合效果

每个方案各有利弊：从实际应用的角度看，使用按钮更接近现实中的电路控制手段，能够更生动地模拟现实；从素材处理的角度看，方案三只需设计两种不同效果的电路图，而方案四在此基础上还需设计开关的造型。在综合考虑实际应用和素材处理后，可以方案一和方案三为基础形成最终实施方案：设计“电路”角色并使用按钮控制电路。

要实现以上方案，关键在于如何实现使用按钮来控制电路的断开与闭合。具体设计方案如表 4.1.3 所示，请同学们根据上面的描述将运行流程补充完整。

表 4.1.3 控制电路的设计方案

作品名称	控制电路
作品功能	按下按钮后，电路进行状态切换
器材清单	Arduino 控制板、USB 数据线、按钮
角色	电路
运行流程	
实现过程	<ol style="list-style-type: none"> <li>① 将按钮与 Arduino 控制板连接</li> <li>② 设置角色及造型</li> <li>③ 编写程序，实现利用按钮对电路进行控制</li> <li>④ 测试效果，优化作品</li> </ol>



## 任务二 控制电路的实现

### ※ 活动 1 控制电路的硬件搭建

准备好器材后，请同学们参考图 4.1.3 将按钮连接在数字引脚 2 上。

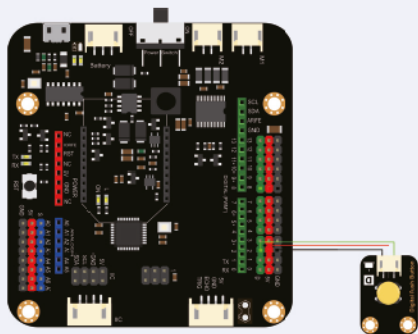


图 4.1.3 硬件连接图

### ※ 活动 2 设置角色及造型

设置“电路”角色并为其导入“电路断开”与“电路闭合”造型，造型素材如图 4.1.4 所示。同学们也可以自己编辑、制作“电路”角色的造型来完善作品。

**💡** 本案例选用含有小灯泡的电路作为素材并通过灯泡的亮、灭来表现电路的闭合与断开效果，因此，当电路闭合时应该提高界面的亮度，当电路断开时应该降低界面的亮度。

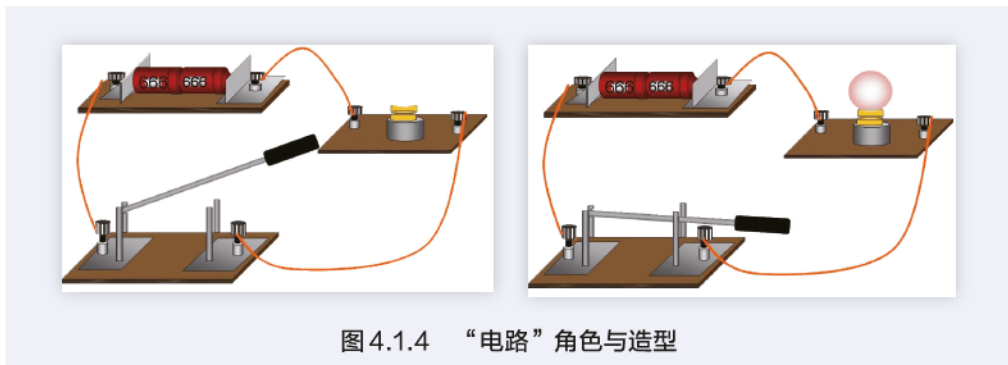


图 4.1.4 “电路”角色与造型

### ● 造型设置与切换

选中某一角色对象后，选择“造型”选项对该角色的造型进行编辑。可自行绘制造型图案，也可导入外部文件或使用S4A软件提供的造型素材，如图4.1.5所示。

角色造型的切换需要使用程序模块来实现。在“外观”模块分类中选择 **切换到造型** 模块可将角色切换到某一指定的造型，选择 **下一个造型** 模块将角色切换到下一个造型。



图 4.1.5 编辑角色的造型

**💡** 单击控制按钮中的绿旗图标运行程序，在编辑程序时首先要选择“控制”分类里的 **当被单击** 模块，然后按顺序拼接其他程序模块。

### ※ 活动3 控制电路的程序编写

如何获取按钮的值并实现“电路”造型的切换呢？使用传感器数值获取模块和造型切换模块可以实现以上功能。

### ● 引脚数值的输入与输出

S4A将Arduino控制板上的引脚按照数字输出、数字输入、模拟输出和模拟输入进行了分类，每个引脚只对应一种输出或输入方式，如表4.1.4所示。

表 4.1.4 引脚分类

类型	引脚号
数字输出	数字引脚10~13
数字输入	数字引脚2、3
模拟输出	数字引脚5、6、9
模拟输入	模拟引脚0~5

**💡** S4A将程序模块按功能分为八类(如下图所示)，需要某一种功能的模块时在相应类型中选择即可。



在“动作”模块分类中可选择 `digital 13 on` 或 `digital 13 off` 输出高、低电平；选择 `analog 9 value 255` 模块为元件输出模拟值；选择 `sensor Digital2 pressed?` 模块获取数字传感器的值；选择 `value of sensor Analog0` 模块获取模拟传感器的值。

通过学习，我们可以选择 `下一个造型` 模块切换电路的状态，当按钮被按下时电路切换状态；选择“控制”分类中的 `如果` 模块进行条件判断。在程序执行过程中需不停地判断按钮的状态，因此使用“控制”分类中的 `重复执行` 模块控制程序循环执行，参考程序如图 4.1.6 所示。

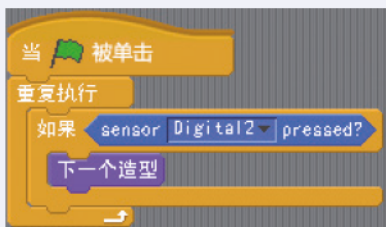



图 4.1.6 控制电路程序

 当角色处于最后一个造型时，使用 `下一个造型` 模块会切换到第一个造型。



### 任务三 控制电路的测试与优化

#### ※ 活动 1 按钮消抖

在测试时可能发现，有时只按了一下按钮，但是“电路”角色的造型会切换多次。其原因与按钮的抖动有关。

#### ● 按钮消抖

按钮通常使用机械弹性开关，由于机械触点的弹性作用，当机械触点断开或闭合时会发生抖动，因此一次按键动作有时会被误读为多次。为了避免按钮抖动影响程序的运行效果，可以使用延时模块进行按钮消抖，即获取一次按键动作后延长一段时间再读取按钮的值。

在“控制”中有 `等待 1 秒` 模块，请同学们尝试将该模块放在程序中的合适位置，延时之后再判断一次按钮状态，以解决按钮抖动问题。

#### ※ 活动 2 进一步测试与修改控制电路项目

根据自己的理解和需求，添加或修改控制电路作品的功能，形成具有个人特色的作品。也可在 Arduino 端连接 LED 模块，同步响应电路中灯泡的亮灭。



## 作品发布与评价

参考表4.1.5开展作品质量的自评、互评等，与同学分享自己的制作经验。

表4.1.5 作品评价表

指标	得分
1. 能够设置角色并编辑角色的造型（2分）	
2. 能够使用按钮控制电路（3分）	
3. 能够实现电路角色的造型切换（3分）	
4. 运行效果不受按钮抖动的影响（2分）	
总分（10分）	



## 拓展练习

请综合运用按钮、切换造型和其他传感器，设计一个更加智能的、具有创意的作品。可以参考以下项目主题。

1. 声控电路：使用声音传感器作为触发开关，根据声音传感器的值来控制电路的断开与闭合。

2. 光控电路：使用光线传感器作为触发开关，根据光线传感器的值来控制电路的断开与闭合。

请进行小组合作，讨论并确定主题，除了已经给出的方案也可选择其他具有创意的设计主题，形成具体设计方案后填写附录1所示的项目设计方案，再尝试实现。

## 4.2 角色交互

S4A 作品中的各个角色之间还可以进行交互，使用广播模块可以做到这一点。方法是通过建立角色间的联系，使角色可以通过广播模块将自身的状态“广播”出去，同时，接收到广播的角色可以根据广播内容做出“响应”。



### 学习目标

- ★ 了解角色显示与隐藏的方法，了解特效模块和声音模块的使用方法。
- ★ 掌握火焰传感器和模拟角度传感器的使用方法，并能够利用它们控制舞台中的角色。
- ★ 掌握广播模块的使用方法，体会“广播与接收”（事件监听机制）这一面向对象编程的思想方法。

家里刚刚换了新的燃气灶，妈妈一边看说明书一边操作新燃气灶。那么，能不能利用 S4A 模拟燃气灶的控制过程呢？这样就能在不接触实物的情况下安全地教授其他人如何使用燃气灶了。

要模拟燃气灶，需要考虑选择何种传感器“触发”燃气灶的点火、使用何种传感器调节燃气灶火力的大小，以及如何进行 S4A 端的角色设置。要解决这些问题，需要完成以下三个任务：模拟燃气灶项目规划、模拟燃气灶的实现、模拟燃气灶的测试与优化。





## 任务一 模拟燃气灶项目规划

### ※ 活动1 模拟燃气灶项目调研

燃气灶有许多类型。为进一步了解燃气灶的特点，请同学们借助网络或实地开展调研并填写表4.2.1。

表4.2.1 常见的燃气灶调查表

序号	燃气灶类型	火力控制方法	适用范围
1			
2			
3			
4			

本节课拟模拟的燃气灶是家用的台式灶，使用时首先要打火，打火后通过旋转控制旋钮调节火力的大小。为此，我们可以在Arduino端使用硬件模拟打火和火力控制过程，在S4A端呈现燃气灶的状态变化。

为了完成该作品，要解决的主要问题是燃气灶的点火装置、火力调节装置和S4A端火力大小的表现方式，其中燃气灶控制过程和S4A端的表现形式是关键。

### ※ 活动2 模拟燃气灶的可选方案分析与设计

根据上述思路，可以从点火装置、火力调节装置和角色设置三个角度初步设计几种模拟燃气灶的方案，如表4.2.2所示。

表4.2.2 模拟燃气灶的可选方案

类别	方案	方案描述
点火装置	方案一	使用火焰传感器作为点火装置
	方案二	使用光线传感器作为点火装置
火力调节装置	方案三	使用按钮调节火力大小
	方案四	使用模拟角度传感器（电位器）调节火力大小

续表


类别	方案	方案描述
角色设置	方案五	将未点火的燃气灶设为背景，并设置“火焰”角色，通过切换“火焰”造型（如关火、一级火力、二级火力、三级火力和四级火力五个造型）表现火力状态
	方案六	设置“燃气灶”角色，通过切换不同火力状态的“燃气灶”造型（如关火、一级火力、二级火力、三级火力和四级火力五个造型）表现火力状态

每个方案各有利弊：从硬件特性的角度看，与光线传感器相比，火焰传感器对火焰更加敏感；从实际应用的角度看，使用模拟角度传感器调节火力大小比按钮更符合生活经验；从显示效果的角度看，切换一体化的“燃气灶”角色比单独切换“火焰”角色的效果要更加逼真。在综合考虑硬件特性、实际应用和显示效果后，可以方案一、方案四和方案六为基础形成最终实施方案：使用火焰传感器检测是否打火，检测到火源后界面中出现“火柴”角色模拟点火过程，并使用模拟角度传感器控制火力的大小。

要实现以上方案，关键在于解决“火柴”角色如何获取点火的信息这一问题。请同学们根据上面的描述，将表4.2.3中的运行流程补充完整。

表4.2.3 模拟燃气灶的设计方案

作品名称	模拟燃气灶
作品功能	能够模拟燃气灶的打火与火力调节过程
器材清单	Arduino 控制板、USB 数据线、火焰传感器、模拟角度传感器
角色	燃气灶、火柴
运行流程	
实现过程	<ol style="list-style-type: none"> <li>① 将火焰传感器和模拟角度传感器与 Arduino 控制板连接</li> <li>② 编写程序，实现打火与火力调节的模拟过程</li> <li>③ 上传程序，测试效果</li> </ol>

 请同学们在老师指导下使用火源。火焰传感器虽能检测到火焰，但其本身并不防火，使用时请勿直接接触火源。

## ● 火焰传感器

火焰传感器可以用来探测波长在760纳米~1100纳米的光源，如图4.2.1所示。

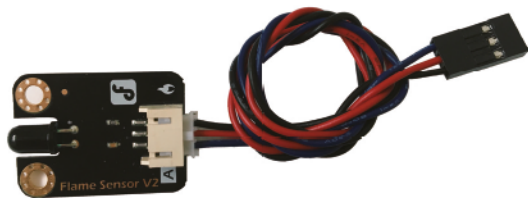


图4.2.1 火焰传感器

## ● 模拟角度传感器

模拟角度传感器是基于电位器制作的一款模拟传感器，旋转角度范围为0到300度，输入值的范围为0~1023，如图4.2.2所示。

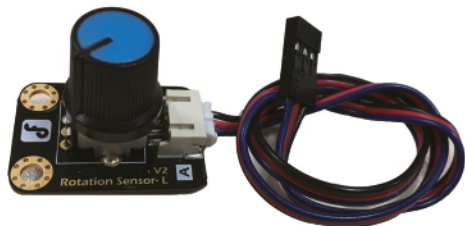



图4.2.2 模拟角度传感器



## 任务二 模拟燃气灶的实现

### ※ 活动1 模拟燃气灶的硬件搭建

准备好器材后，请同学们参考图4.2.3将火焰传感器连接在模拟引脚A3上，模拟角度传感器连接在模拟引脚A0上。

 电位器通常由电阻体和可移动的电刷组成，是一种阻值可调节的电阻元件。当电刷沿电阻体移动时，电阻与电刷的位移量呈一定变化关系。

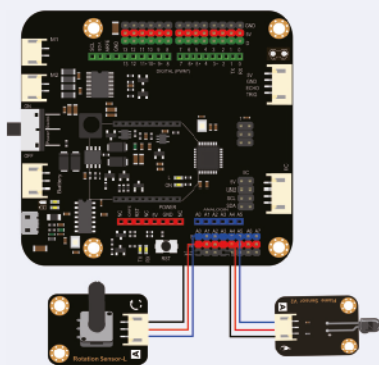


图4.2.3 硬件连接图

### ※ 活动2 设置角色及造型

设置“燃气灶”角色并为其导入关火、一级火力、二级火力、三级火力和四级火力五个造型，造型素材如图4.2.4所示；同时还要设置“火柴”角色及其造型，造型素材如图4.2.5所示。同学们也可以自己编辑、制作“燃气灶”和“火柴”角色的造型来完善作品。



图4.2.4 “燃气灶”角色的造型



图 4.2.5 “火柴”角色的造型

### ※ 活动 3 模拟燃气灶的程序编写

观察传感器信息显示面板中 A3 引脚的值，假定当火焰传感器的值大于 500 时，可以认为有人在进行打火动作。程序开始执行后，首先将“燃气灶”角色切换到“关火”造型，然后判断传感器是否检测到了打火动作，程序如图 4.2.6 所示。

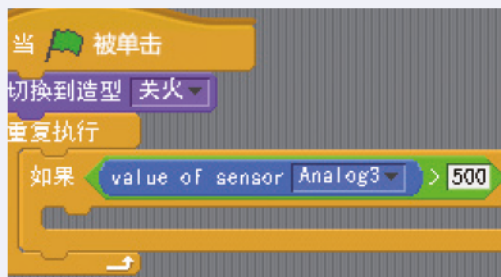


图 4.2.6 造型切换与火焰检测

同时，可以将模拟角度传感器的输入值分为五个不同的区间，如图 4.2.7 所示，即当输入值小于 20 时，燃气灶关火；当输入值大于等于 20 但小于 300 时，燃气灶为一级火力；当输入值大于等于 300 但小于 600 时，燃气灶为二级火力；当输入值大于等于 600 但小于 900 时，燃气灶为三级火力；当输入值大于等于 900 时，燃气灶为四级火力。



图 4.2.7 “燃气灶”角色切换造型的脚本

当检测到火焰后，显示“火柴”角色，并在一秒后隐藏。在编写程序的过程中，会发现选中“火柴”角色后，动作模块分类中没有针对控制板信息输入与输出的编程模块。那么，“火柴”角色如何才能知道打火的消息？如何实现“火柴”角色的显示与隐藏？此时使用广播模块、显示模块和隐藏模块可以解决这些问题。


## ● 控制板角色与非控制板角色

能够与控制板进行数据的输入与输出的角色称为控制板角色，不能实现这种功能的角色称为非控制板角色。S4A 可以连接多个 Arduino 控制板，控制板之间只能一对一地进行信息的输入与输出。在新增角色选项中可新建控制板角色和非控制板角色，如图 4.2.8 所示。





图 4.2.8 新增角色





控制板角色可以利用  控件发出广播事件，所有角色甚至是舞台背景都可以根据接收到的广播信息改变自己的状态，如切换造型。通过这种方式，S4A 实现了不同角色之间的互动联系。这就是面向对象程序设计思想中的事件监听机制。

## ● 广播模块

如果只使用一个控制板，并试图使多个角色与其进行信息的输入与输出，则需要采用事件监听机制——广播与接收：使用“控制”模块分类中的  模块将某一事件以消息的形式传递出去；使用  模块接收消息，并通过程序编写设置相应角色的行为。

## ● 显示/隐藏模块

使用“外观”模块分类中的  模块使角色显示在舞台中；使用  将角色隐藏，此时角色相当于“隐身”，虽然不显示外形但该角色仍处于舞台之中。



## 任务三 模拟燃气灶的测试与优化

### ※ 活动1 添加打火声音

现实中燃烧的火焰是跳动的，而模拟燃气灶作品中的火焰是静

止的。那么如何使火焰产生动态的效果呢？可以使用“外观”模块分类中的 **将 旋转 特效增加 3** 模块实现该效果。同时，还可以为作品添加音效，当“火柴”角色隐藏时播放打火的声音。

## ● 特效模块

选择“外观”模块分类中的 **将 旋转 特效增加 3** 模块可使角色产生扭曲效果。当参数为正值时角色顺时针方向扭曲，为负值时角色逆时针方向扭曲。通过设置角色旋转角度，使其不停地朝相反方向扭曲，可实现角色造型的动态变化效果。同时，还可以根据实际需要，选择颜色、超广角模块、像素滤镜、马赛克、亮度和虚像等特效。

## ● 声音模块

“声音”中的 **弹奏鼓声 39 0.2 拍** 模块可播放不同效果的鼓声，**设定乐器为 1** 模块可设置不同类型的乐器，**弹奏音符 60 0.5 拍** 模块可设置乐器播放的音符与节拍，**播放声音** 模块可播放录音或文件中的声音。

使用 **弹奏鼓声 39 0.2 拍** 模块模拟打火声音，并根据特效模块的使用方法编写模拟火焰抖动的程序，相关设置如图 4.2.9 所示。请同学们将编程模块放在程序中的合适位置，并测试作品的运行效果。



图 4.2.9 模拟火焰抖动效果的设置

### ※ 活动 2 进一步测试与修改模拟燃气灶作品

请同学们根据自己的理解和需求，添加或修改模拟燃气灶的功能，制作具有个人特色的作品。比如，测试作品时会发现在“火柴”角色消失之前就会播放打火的声音。此时可以利用等待模块，使“火柴”角色消失时再播放打火的声音。



## 作品发布与评价

参考表4.2.4开展作品质量的自评、互评等，与同学分享自己的制作经验。

表4.2.4 作品评价表

指标	得分
1. 能够使用火焰传感器模拟打火动作（3分）	
2. 能够使用模拟角度传感器模拟燃气灶的火力调节（3分）	
3. 能够实现“燃气灶”角色和“火柴”角色的交互（2分）	
4. 能够播放打火声音并实现火焰的动态效果（2分）	
总分（10分）	



## 拓展练习

请同学们综合运用火焰传感器、模拟角度传感器和广播模块等知识，在S4A端显示模拟动画，设计一个具有创意的作品。可以参考以下设计主题。

1. 模拟风扇的风力调节：S4A端显示“风扇”角色，使用模拟角度传感器调节风扇的转速。

2. 我是鼓手：S4A端显示一面鼓和鼓槌，使用按钮或振动传感器等“打”鼓。

请进行小组合作，讨论并确定设计主题，除了已经给出的方案，也可以选择其他具有创意的设计主题，形成具体设计方案后填写附录1所示的项目设计方案，再尝试实现。

## 4.3 坐标定位

为了更加准确地获取角色在舞台中的位置，S4A为舞台设置了坐标系。S4A选取舞台中心位置为坐标原点，x坐标的数值为-240到240，y坐标的数值为-180到180。用户可以使用坐标模块将角色移动到某一特定的坐标位置。



### 学习目标

- ★ 掌握S4A中停止脚本运行的方法，掌握文本框显示的方法。
- ★ 掌握温度传感器的使用方法，并能够利用它控制舞台中的角色。
- ★ 掌握坐标定位的方法，能够利用坐标控制角色的移动。

今天天气突然升温，妈妈打开了家里的风扇，正在看书的阿丢抬头一看，原本趴在沙发上睡觉的小狗走到风扇前又趴下睡着了。他觉得小狗的这一举动非常可爱，便突发奇想：能否用S4A呈现这个生活趣事呢？

要呈现这个故事，需要考虑以下几个问题：选择何种传感器检测环境温度？如何获取小狗与风扇的相对位置并实现小狗向风扇的移动？要解决这些问题，需要完成以下三个任务：“怕热的小狗”项目规划、“怕热的小狗”的实现、“怕热的小狗”的测试与优化。



### 任务一 “怕热的小狗”项目规划

#### ※ 活动1 “怕热的小狗”项目调研

“怕热的小狗”只是阿丢生活中发现的趣事之一，其实我们身边不乏有趣、感人的小故事，请同学们回忆和分享生活中发生过的趣事，填写表4.3.1。



表4.3.1 生活趣事

序号	主角	发生时间	内容
1			
2			
3			
4			

为了增强故事的趣味性，我们可以把故事情节修改为：随着温度的升高，小狗感到天气变热，于是从沙发处走到自动变速风扇前，风扇像是知道小狗的意图一样，自动打开为小狗扇风。可以在Arduino端使用传感器获取环境中的温度，在S4A端呈现小狗和风扇的动作。

本项目的实现主要与角色位置设定和风扇的触发有关，其中角色位置的初始化和位置移动是关键。

### ※ 活动2 “怕热的小狗”的可选方案分析与设计

根据上述思路，可以从角色坐标定位和风扇触发方式两个角度初步设计几种项目方案，如表4.3.2所示。

表4.3.2 “怕热的小狗”的可选方案

类别	方案	方案描述
角色坐标定位	方案一	手动拖动角色进行坐标定位
	方案二	使用移动模块进行角色的坐标定位
风扇触发方式	方案三	小狗到达风扇处时发送广播“告知”风扇转动
	方案四	风扇侦测其与“小狗”角色的距离，并根据距离值判断是否转动

每个方案各有利弊：从坐标定位精确性的角度看，使用移动模块进行角色坐标定位比手动拖动角色的方法更精确；从学习内容的角度看，方案三使用广播模块，方案四需要使用侦测模块，此处使用已学过的广播模块来实现。在综合考虑坐标定位精确性和学习内

容后，我们以方案二和方案三为基础形成最终实施方案：使用移动模块设定角色的初始坐标，当温度达到一定值时，“小狗”角色移向“风扇”角色，此时风扇转动。

要实现以上方案，关键在于实现“小狗”角色的定向移动和风扇的触发。请同学们根据上面的描述，将表 4.3.3 中的运行流程补充完整。

表 4.3.3 “怕热的小狗”的设计方案

作品名称	“怕热的小狗”
作品功能	模拟小狗因为温度升高主动打开风扇纳凉的故事情节
器材清单	Arduino 控制板、USB 数据线、温度传感器
角色	小狗、风扇
运行流程	
实现过程	<ol style="list-style-type: none"> <li>① 将温度传感器与 Arduino 控制板连接</li> <li>② 编写程序，模拟“怕热的小狗”的故事情节</li> <li>③ 上传程序，测试效果</li> </ol>

## ● 温度传感器

温度传感器能够检测环境温度，其测温范围是 $-40^{\circ}\text{C}\sim 150^{\circ}\text{C}$ ，输出电压与温度成正比，如图 4.3.1 所示。

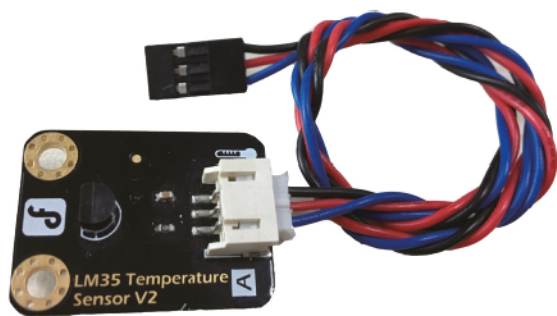


图 4.3.1 温度传感器



## 任务二 “怕热的小狗”的实现

### ※ 活动1 “怕热的小狗”的硬件搭建

准备好器材后，请同学们参考图4.3.2将温度传感器连接在模拟引脚A0上。

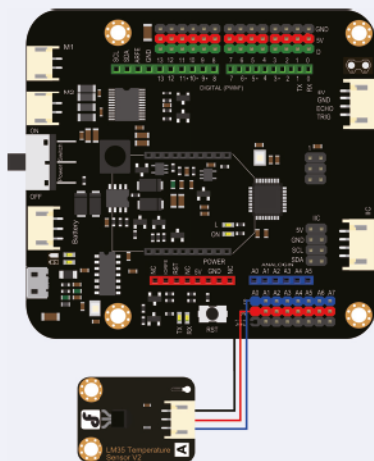


图4.3.2 硬件连接图

### ※ 活动2 设置角色及造型

设置“小狗”角色和“风扇”角色，并为“小狗”角色设置睡觉、有疑问、走动1和走动2四个造型，造型素材如图4.3.3所示；同时，为“风扇”角色设置扇叶转动不同角度的造型，造型素材如图4.3.4所示。



图4.3.3 “小狗”角色的造型



图4.3.4 “风扇”角色的造型

由于故事发生在家中，因此将活动场所设定为客厅，选择舞台角色并为其导入客厅造型，造型素材如图4.3.5所示。



图 4.3.5 舞台造型

### ※ 活动 3 给“怕热的小狗”编写程序

根据故事情节，在程序运行时，首先应将小狗切换到“睡觉”造型，并为角色设置初始位置。如何使角色准确地移动到舞台中的某一特定位置呢？使用“动作”模块分类中的“移到”模块可以实现该功能。



S4A不能自动设置角色的初始位置，运行程序时，角色仍处于上一次程序运行结束时所在的位置。因此要为角色设置初始坐标来规定其初始位置。

## ● 坐标定位

“动作”模块分类中的 **移到 x: -155 y: -82** 模块能将角色移动到某一特定坐标处，**在 1 秒内，平滑移动到 x: -155 y: -82** 模块能使角色在限定时间内移动到某一特定坐标处，**移到** 模块能将角色移动到另一角色处，**x坐标** 和 **y坐标** 模块能获取角色当前的x坐标值和y坐标值。

如何知道当前坐标呢？在舞台下方显示了当前鼠标指针所处的坐标值，如图 4.3.6 所示；在角色状态栏的下方显示了当前角色的坐标值，如图 4.3.7 所示。



图 4.3.6 鼠标指针坐标值显示



当鼠标指针离开舞台区域时，舞台下方仍会以舞台中心为坐标原点显示鼠标指针当前的坐标值。

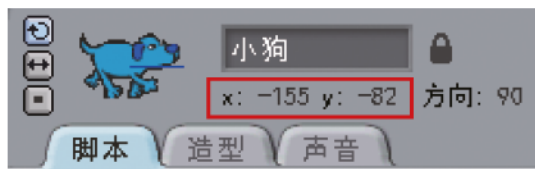


图 4.3.7 角色坐标值显示

若使用 **移到 x: -160 y: -85** 模块将小狗移到了沙发处，请同学们为“风扇”角色设置合适的初始位置。

在观察传感器信息显示面板中A0引脚的值后，假定当温度传感器的值大于55时，小狗感觉到气温升高要去打开风扇，“小狗”角色和“风扇”角色的行为可以安排如下，请同学们根据描述完善脚本。

(1) “小狗”角色切换到疑问造型并从沙发上移动到地板上。

(2) 使用 **在 3 秒内，平滑移动到 x: 143 y: -114** 模块，使“小狗”角色在3秒内移动到风扇前方。同时通过不断切换小狗走动1造型和走动2造型实现小狗的走动效果，因为“小狗”角色的移动和造型切换是同时进行的，需使用广播模块实现该功能。即当“小狗”角色向风扇移动前发送广播“走动”，“小狗”角色接收到该广播后重复切换小狗走动1造型和走动2造型直到其移动到风扇前方。接收到广播“走动”的脚本如图4.3.8所示。

(3) 当“小狗”角色走到风扇前时，发送广播“打开风扇”，“小狗”角色切换到睡觉造型。“风扇”角色接收到广播“打开风扇”后，通过重复切换下一个造型来实现转动效果。



图4.3.8 “小狗”角色走动效果参考脚本

### 任务三 “怕热的小狗”的测试与优化

#### ※ 活动1 停止执行脚本

由于程序重复检测温度传感器的值，当温度升高后条件一直成立，因此会重复执行选择结构内的脚本。那么，如何使选择结构内的脚本只运行一次呢？使用停止执行脚本模块可以实现该功能。

#### ● 停止执行脚本模块

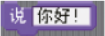
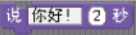

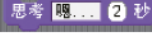
“控制”中的 **停止执行这个脚本** 模块能够停止运行其所在的程序段，而 **全部停止** 模块能够停止所有脚本的运行。


请同学们使用 **停止执行这个脚本** 模块完善脚本。

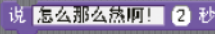
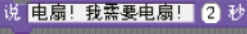
## ※ 活动2 表现小狗的内心活动

为了更好地表现“小狗”角色的内心活动，可以使用文本框表现小狗的内心活动。

### ● 添加文本框

“外观”中的  模块可在舞台中显示相关文字的对话型文本框， 模块可以规定对话型文本框在舞台中显示的时间， 模块可在舞台中显示相关文字的气泡型文本框， 模块可以规定气泡型文本框在舞台中显示的时间。

 文本框模块只能够在舞台中显示相关文字，不能播放声音。

当小狗切换到疑问造型并移动到地板上时，使用  模块和  模块显示其内心活动。请同学们将这两个模块放在程序中合适的位置。

## ※ 活动3 进一步测试与修改“怕热的小狗”作品

请同学们根据自己的理解和需求，添加或修改“怕热的小狗”的功能，制作具有个人特色的作品，使故事更具趣味性。



## 作品发布与评价

参考表4.3.4开展作品质量的自评、互评等，与同学分享自己的制作经验。

表4.3.4 作品评价表

指标	得分
1. 能够初始化角色的坐标（3分）	
2. 能够实现温度升高后小狗向风扇移动的动作（3分）	
3. 当小狗移动到风扇前时，风扇能够转动（2分）	
4. 能够使用对话框表现小狗的内心活动（2分）	
总分（10分）	



## 拓展练习

移动模块是一个简单的指令，但可以结合温度传感器、声音传感器、光线传感器等传感器，实现很多功能。请同学们综合运用之前所学的知识，设计一个趣味故事。可以参考以下设计主题。

1. 小猫吹气球：小猫看到气球后，非常好奇，想要尝试一下吹气球，当气球大到一定程度时爆炸。可以在 Arduino 端连接声音传感器，通过用户吹气模拟小猫吹气球的动作。

2. 植物的生长：在 Arduino 端连接光线传感器，当光线传感器的值达到一定值时，舞台中的植物逐渐长大。

请进行小组合作，讨论并确定主题，除了已经给出的方案，也可以选择其他具有创意的设计主题，形成具体设计方案后填写附录 1 所示的项目设计方案，再尝试实现。

## 4.4 侦测

通过前面的学习，我们知道 S4A 中的角色可以利用传感器检测外界环境中的信息，那么角色能否检测舞台环境中的信息呢？此时可以使用“侦测”模块来实现。“侦测”模块具有距离检测、颜色检测等多种检测功能。根据检测结果，可以控制程序的运行、增强角色之间的交互效果。



### 学习目标

- ★ 掌握变量和随机数模块的使用方法，能够用程序控制角色的朝向。
- ★ 掌握摇杆的使用方法，并能够利用它控制舞台中的角色。
- ★ 掌握侦测模块的使用方法，体会侦测背后的事件监听机制。

通过本学期的学习，我们不仅能够使用开源硬件模拟生活中的科技产品、进行探究实验等，还能制作趣味性的产品说明、数字故事等。为了检验自己的学习成果，本节课将要制作一款趣味性强的交互游戏：利用 S4A 模拟战争场景来开发一款“飞机坦克大战”游戏。

要模拟飞机坦克大战，需要考虑如何控制飞机和坦克的移动、如何进行 S4A 端的角色设置、如何“触发”飞机和坦克发射炮弹、如何判断飞机或坦克被击中，以及如何获取被击中的飞机或坦克数量。要解决这些问题，需要完成以下三个任务：飞机坦克大战项目规划、飞机坦克大战的实现、飞机坦克大战的测试与优化。



### 任务一 飞机坦克大战项目规划

#### ※ 活动 1 飞机坦克大战项目调研

要模拟飞机坦克大战，首先需要确定游戏规则。请同学们讨论



飞机坦克大战的基本规则，填写表4.4.1。

表4.4.1 飞机坦克大战的游戏规则

序号	规则
1	
2	
3	
4	

假定我们要模拟的飞机坦克大战以坦克战胜飞机为目标，即操作者使用传感器控制坦克的运动方向和炮弹发射，飞机不断从舞台上空飞过，并随机向下投放炸弹；坦克的任务是尽可能多地向飞机发射炮弹并将其击中，如果飞机发射的炸弹击中坦克，那么游戏结束。

飞机坦克大战的实现，主要与Arduino端的硬件选择、角色之间的相互侦测和分值设置有关，其中角色间的相互侦测是关键。

### ※ 活动2 飞机坦克大战的可选方案分析与设计

根据上述思路，可以从硬件选择、角色间的侦测方式和得分方式三个方面设计几种飞机坦克大战的方案，如表4.4.2所示。

表4.4.2 飞机坦克大战的可选方案

类别	方案	方案描述
硬件选择	方案一	使用倾斜传感器控制坦克的运动方向，用按钮触发炮弹的发射
	方案二	使用摇杆控制坦克的运动方向，并用摇杆上的按钮触发炮弹的发射
角色间的侦测方式	方案三	使用距离侦测检测角色之间的距离，当距离小于某一值时判定两个角色碰到
	方案四	使用颜色侦测检测颜色块是否碰到，若两个角色的颜色相碰即判定两者碰到
得分方式	方案五	得分=坦克击中飞机的数量
	方案六	得分=坦克击中飞机的数量+坦克成功躲避的飞机炸弹数量



侦测模块中有直接判定两个角色是否碰到的模块，但是无法实现角色间的双向侦测。比如，角色A检测是否碰到角色B，若碰到则执行语句1，这是可以实现的。若想让角色B同时检测是否碰到角色A，若碰到则执行语句2是无法实现的，因为无法同时运行执行语句1和执行语句2。

每个方案各有利弊：从实际应用的角度看，使用摇杆能够同时控制坦克的运动和炮弹的发射，操作更方便；从造型素材处理的角度看，使用方案四进行角色间的侦测要保证每个角色均由一种颜色组成，增加了素材处理的难度；从程序编写的角度看，方案五更易实现。在综合考虑实际应用、素材处理和程序编写后，我们以方案二、方案三和方案五为基础形成最终实施方案：使用摇杆控制坦克的运动和炮弹的发射，使用距离侦测检测角色间是否碰到，同时游戏分值设为坦克击中飞机的个数；若坦克被击中，则游戏结束。

要实现以上方案，关键在于获取角色间的距离与角色的精确定位。请同学们根据上面的描述将表 4.4.3 中的运行流程补充完整。

表 4.4.3 飞机坦克大战的设计方案

作品名称	飞机坦克大战
作品功能	模拟飞机坦克间的火力对抗
器材清单	Arduino 控制板、USB 数据线、摇杆
角色	坦克、坦克炮弹左、坦克炮弹右、飞机左、飞机右、飞机炸弹左、飞机炸弹右
运行流程	
实现过程	<ol style="list-style-type: none"> <li>① 将摇杆与 Arduino 控制板连接</li> <li>② 编写程序，实现飞机坦克大战间的火力对抗</li> <li>③ 上传程序，测试效果</li> </ol>

## ● 摇杆

本节使用的摇杆可以输出 x 轴和 y 轴两路模拟信号和 z 轴一路按钮数字信号，如图 4.4.1 所示。其由三组引脚组成，其中引脚 x 获取摇杆的横向控制，值

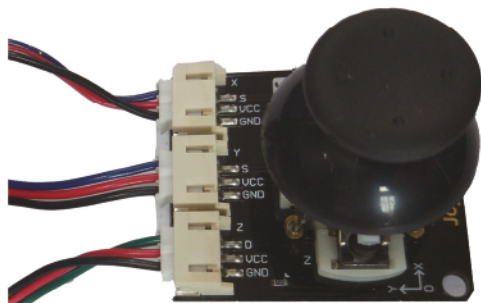


图 4.4.1 摇杆



当摇杆静止时，x 引脚与 y 引脚的输入值均为 507。但实际使用时会有一些偏差，因此使用前应观察不同状态下摇杆的输入值。

为0~1023；引脚y获取摇杆的纵向控制，值为0~1023；引脚z获取摇杆按钮的值，当按钮按下时为False，反之为True。



## 任务二 飞机坦克大战的实现

### ※ 活动1 飞机坦克大战的硬件搭建

本游戏设计中坦克只在水平方向运动，因此只需要连接摇杆的x轴和z轴。准备好器材后，请同学们参考图4.4.2将摇杆的引脚x连接在模拟引脚A3上，将摇杆的引脚z连接在数字引脚2上。

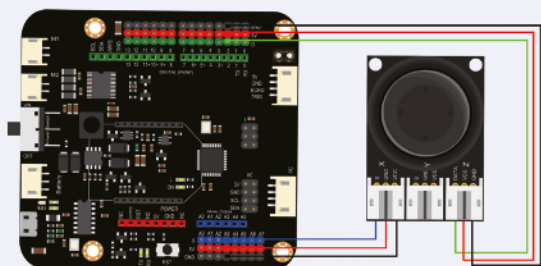


图4.4.2 硬件连接图

### ※ 活动2 设置角色及造型

我们需要设置以下七个角色。

(1) “坦克”角色，包含坦克和爆炸两个造型，如图4.4.3所示。

(2) “坦克炮弹左”角色（坦克向左运动时发射的炮弹），其造型如图4.4.4所示。



图4.4.3 “坦克”角色的造型



图4.4.4 “坦克炮弹左”角色的造型

(3) “坦克炮弹右”角色（坦克向右运动时发射的炮弹），其造型在图4.4.4基础上修改即可。

(4) “飞机左”角色（程序启动时向左飞行的飞机），包含飞机和爆炸两个造型，如图4.4.5所示。

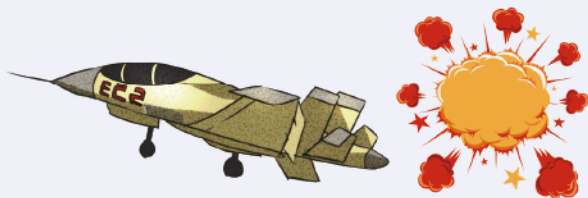


图4.4.5 “飞机左”角色的造型

(5) “飞机右”角色（程序启动时向右飞行的飞机），其造型在图 4.4.5 基础上修改即可。

(6) “飞机炸弹左”角色（飞机左角色发射的炸弹），包含飞机炸弹和爆炸两个造型，如图 4.4.6 所示。

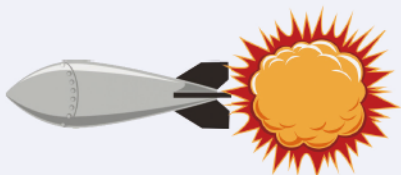


图 4.4.6 “飞机炸弹左”角色的造型

(7) “飞机炸弹右”角色（飞机右角色发射的炸弹），其造型在图 4.4.6 基础上修改即可。

同时，还要为舞台角色设置战争背景和游戏结束背景，如图 4.4.7 所示。同学们也可以自己选择、编辑角色的造型素材完善作品。



图 4.4.7 舞台角色的造型

角色虽有左右之分，但是其造型设置与脚本编写基本相同。上面已完成了左向的角色设置，请同学们继续完成右向的角色设置。

### ※ 活动 3 飞机坦克大战的程序编写

(1) 角色的初始化。因为飞机坦克大战项目中的角色较多且角色间存在较复杂的交互，所以要先设置各角色的初始位置，以免影响程序的执行效果。例如，分别用 `移到 x: 0 y: -120` 模块、`移到 x: 243 y: 78` 模块、`移到 x: 218 y: -159` 模块和 `移到 x: 205 y: 168` 模块初始化“坦克”“坦克炮弹左”“飞机左”和“飞机炸弹左”等角色的坐标；同时还要设置各个角色的初始造型，并设置“飞机左”角色的初始状态为显示，“坦克炮弹左”和“飞机炸弹左”角色的初始状态为隐藏。请同学们参考上述操作将右向角色初始化。

(2) 识别摇杆的动作。通过观察传感器信息显示面板分析摇杆左右移动动作与输入值的关系，据此设定当 x 引脚输入值小



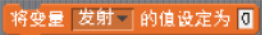


摇杆的控制方向与如何手持有关，使用时可根据实际使用状况规定摇杆的控制方向。

于500时触发坦克向右运动，当输入值大于510时触发坦克向左运动。同时，当摇杆上的按钮被按下，即z引脚的输入值为False时触发炮弹的发射。为了优化程序设计，可以通过设置变量来获取摇杆的输入值。



选中变量名前的小方框也可以控制变量在舞台中的显示。




## ● 变量

“变量”中的  用于新建变量。例如，新建一个名称为“发射”的变量后，将会出现  模块、 模块、 模块和  模块，分别用以改变变量的值或控制变量在舞台中的显示。

可以设置三个变量：“坦克运动方向”用来存放模拟引脚A3（摇杆的x引脚）的值；“发射”用来存放数字引脚2（摇杆的z引脚）的值；“得分”用来存放用户玩游戏的得分，并使其显示在舞台中。

(3) 改变“坦克”角色的运动。可以使用面向模块、移动步数模块和碰到边缘就反弹模块改变“坦克”角色的运动。

## ● 面向模块

可以通过改变“动作”中的  模块的参数值改变角色在上、下、左、右四个方向的朝向， 模块能够使角色朝另一角色所在的方向移动， 模块可以获取角色的当前朝向。

## ● 移动步数模块

利用“动作”中的  模块能够实现角色在舞台中的运动。

## ● 碰到边缘就反弹模块

利用“动作”中的  模块可使角色碰到舞台的边缘后水平翻转并自动反向运动。

根据上述知识，可以编写“坦克”角色向左运动时的控制程序，如图4.4.8所示。请同学们参考该程序编写“坦克”角色向右运动时的控制程序。

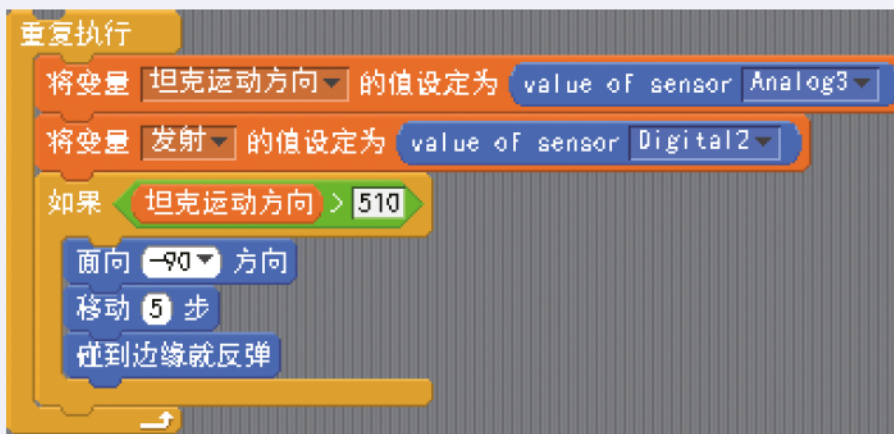


图 4.4.8 “坦克”角色运动控制程序

(4) 控制坦克发射炮弹。由于“坦克炮弹”角色有左右之分，因此需要根据“坦克”角色的朝向确定向哪个方向发射炮弹，这需要使用嵌套选择结构，如图 4.4.9 所示，请同学们将该程序段放到程序中的合适位置。

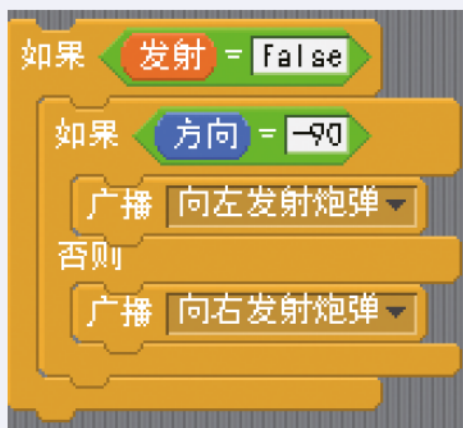


图 4.4.9 “坦克”角色发射炮弹程序

(5) 控制坦克炮弹角色的运动。以“坦克炮弹左”角色为例，当“坦克炮弹左”角色接收到广播“向左发射炮弹”后，移动到“坦克”角色炮筒的位置，可先使其移动到“坦克”角色车身的位置，再调整其 x 与 y 坐标的值。可以使用 `移到 坦克` 模块、`将 x 坐标增加 -50` 模块和 `将 y 坐标增加 50` 模块实现“坦克炮弹左”角色向“坦克”角色炮筒位置的移动，同时使用显示模块使其显示在舞台中。

“坦克炮弹右”角色的运动与之相似。请同学们根据上述思路编写程序。

(6) 判断坦克炮弹是否击中飞机。以“坦克炮弹左”角色击中“飞机”角色为例，当“坦克”角色发出的“坦克炮弹左”角色朝舞台左上方飞行时，若碰到“飞机”角色，则“飞机”角色切换到“爆炸”造型并且游戏分值加 1。那么，如何得知是否击中了飞机呢？可使用侦测模块实现。

## ● 侦测模块

“侦测”中的 **碰到** 模块能够检测角色是否碰到其他角色、鼠标指针或舞台边缘，**碰到颜色** 模块可以检测角色是否碰到某一特定的颜色，**颜色 碰到了 颜色?** 模块可以检测两种颜色是否相碰，**到 的距离** 模块能够获得与其他角色或鼠标指针的距离。

可以使用 **到 的距离** 模块检测是否击中了飞机，即当“坦克炮弹左”角色与“飞机”角色的距离小于50时，判定其击中了飞机，同时广播“炸到飞机左”，并将得分加1，之后隐藏造型并移动到初始位置。图4.4.10为“坦克炮弹左”角色击中“飞机左”角色的程序。请同学们补充“坦克炮弹左”角色击中“飞机右”角色、“坦克炮弹右”角色击中“飞机左”角色、“坦克炮弹右”角色击中“飞机右”角色的程序。

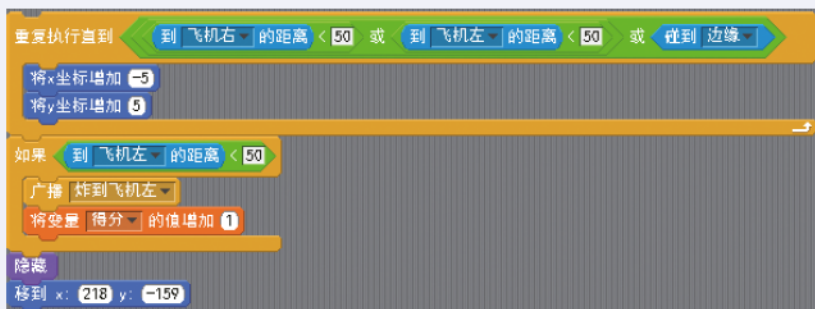


图4.4.10 “坦克炮弹左”角色的程序

(7) 控制飞机的运动。以“飞机左”角色为例，当“飞机左”角色与坦克炮弹的距离大于50时，表示飞机处在安全状态，它将在舞台上重复飞行。请同学们根据该思路编写“飞机左”角色和“飞机右”角色重复飞行的程序。

(8) 显示飞机在被坦克炮弹击中后的状态。以“飞机左”角色被击中为例，当“飞机左”角色接收到广播“炸到飞机左”后，切换到“爆炸”造型并等待0.2秒后隐藏，接着移动到初始位置，等待1秒后切换到“飞机”造型并显示。然后，“飞机左”角色继续在舞台上飞行，碰到炸弹停止飞行，并发送广播“炸到飞机左”。请同学们根据上述思路编写该段程序，并编写“飞机右”角色被击中的程序。

(9) 飞机炸弹的发射。程序启动后，飞机在飞行过程中一直向下发射炸弹。以“飞机炸弹左”角色为例，“飞机炸弹左”角色每隔一段时间就移动到“飞机左”角色的下方并显示，然后降落直到碰到舞台边缘或坦克；如果碰到坦克就广播“炸到坦克了”，然后隐藏；如果碰到舞台边缘就切换到“爆炸”造型。“飞机炸弹左”角色的程序如图 4.4.11 所示，请同学们补充“飞机炸弹右”角色的程序。



图 4.4.11 “飞机炸弹左”角色的程序

(10) 游戏结束。当“坦克”角色接收到广播“炸到坦克了”时，切换到“爆炸”造型；当舞台角色接收到广播“炸到坦克了”时，切换到游戏结束造型，并使用 **全部停止** 模块停止所有程序的执行。请同学们完成该程序的编写。

### 任务三 飞机坦克大战的测试与优化

#### ※ 活动 1 随机发射飞机炸弹

要让游戏更具趣味性，还可以随机地向下发射飞机炸弹，而不是隔固定时间发射。那么如何获取随机数呢？可以使用随机数模块实现。

#### ● 随机数模块

“数字和逻辑运算”中的 **在 1 到 10 间随机选一个数** 模块可以随机地获取设定区间内的数值。

使用 **在 3 到 6 间随机选一个数** 模块并将其嵌套在 **等待 1 秒** 模块中，使“飞机炸弹左”角色在 3 到 6 秒内随机地出现。请同学们将该模块放到“飞机炸弹左”角色脚本中的合适位置。

#### ※ 活动 2 进一步测试与修改飞机坦克大战作品

请同学们根据自己的理解和需求，添加或修改飞机坦克大战的游戏设计，制作具有个人特色的作品。同学们可以通过修改“坦克”“飞机”和“炮弹”等角色的移动速度来设置游戏的难度；可以给“坦克”“飞机”和“炮弹”等角色设置声音，使游戏更具表现力。





## 作品发布与评价

参考表4.4.4开展作品质量的自评、互评等，与同学分享自己的制作经验。

表4.4.4 作品评价表

指标	得分
1. 能够使用摇杆控制坦克运动并发射炮弹（3分）	
2. 能够使用移动步数模块与面向模块实现各角色的运动（2分）	
3. 能够使用侦测模块实现各角色间的交互（3分）	
4. 能够通过设置变量来记录游戏的得分（2分）	
总分（10分）	



## 拓展练习

摇杆和侦测模块在游戏中应用较多，请同学们综合运用之前所学的知识，设计一款属于自己的趣味游戏。可以参考以下设计主题。

1. 大鱼吃小鱼：用摇杆控制鱼的运动，并吃掉比自己小的鱼，随着吃掉小鱼数量的增加，自己的体积逐渐增大，得分也增多。

2. 打地鼠：界面中随机出现地鼠，用户使用摇杆控制界面中打地鼠工具的方向，打到的地鼠数量越多得分越高。

请进行小组合作，讨论并确定主题，除了已经给出的方案，也可以选择其他具有创意的设计主题，形成具体设计方案后填写附录1所示的项目设计方案，再尝试实现。



## 单元学习评价

通过本单元的学习，我们了解了趣味交互型联机系统的工作原理及其特点，体验了开发趣味交互型联机系统的一般过程。你能根据需求独立设计、开发不同类型的趣味交互型联机系统吗？请结合本单元的学习内容，完成以下评价。

1. S4A将Arduino控制板上的引脚分为\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_四类，每个引脚只对应一种输出或输入模式。

2. 第1小组正在使用S4A设计他们的联机交互系统作品，此时需要实现两个角色A、B之间的交互，即当角色A在舞台中的横坐标位置为0时，角色B开始移动。有组员认为可以设置变量，变量值随角色A横坐标位置的变化而变化，当变量达到某一定值时，角色B开始移动；也有组员认为可以使用广播模块，当角色A的横坐标位置为0时发出广播，角色B接收广播并开始移动。

你能帮助第1小组判断哪位同学的方案更佳吗？理由是什么？

3. 元旦联欢会上张老师想要设计一款魔术游戏，当请同学们分别走到屏幕前的固定位置时，屏幕会根据不同的人显示不同的卡通形象，卡通人物的名字即为本次联欢活动中同学们的代号。请使用S4A帮助张老师实现这款游戏，并填写以下任务单。

任务一：魔术游戏项目规划

◆ 魔术游戏项目调研

◆ 魔术游戏的可选方案分析与设计

任务二：魔术游戏的实现

◆ 魔术游戏的硬件搭建

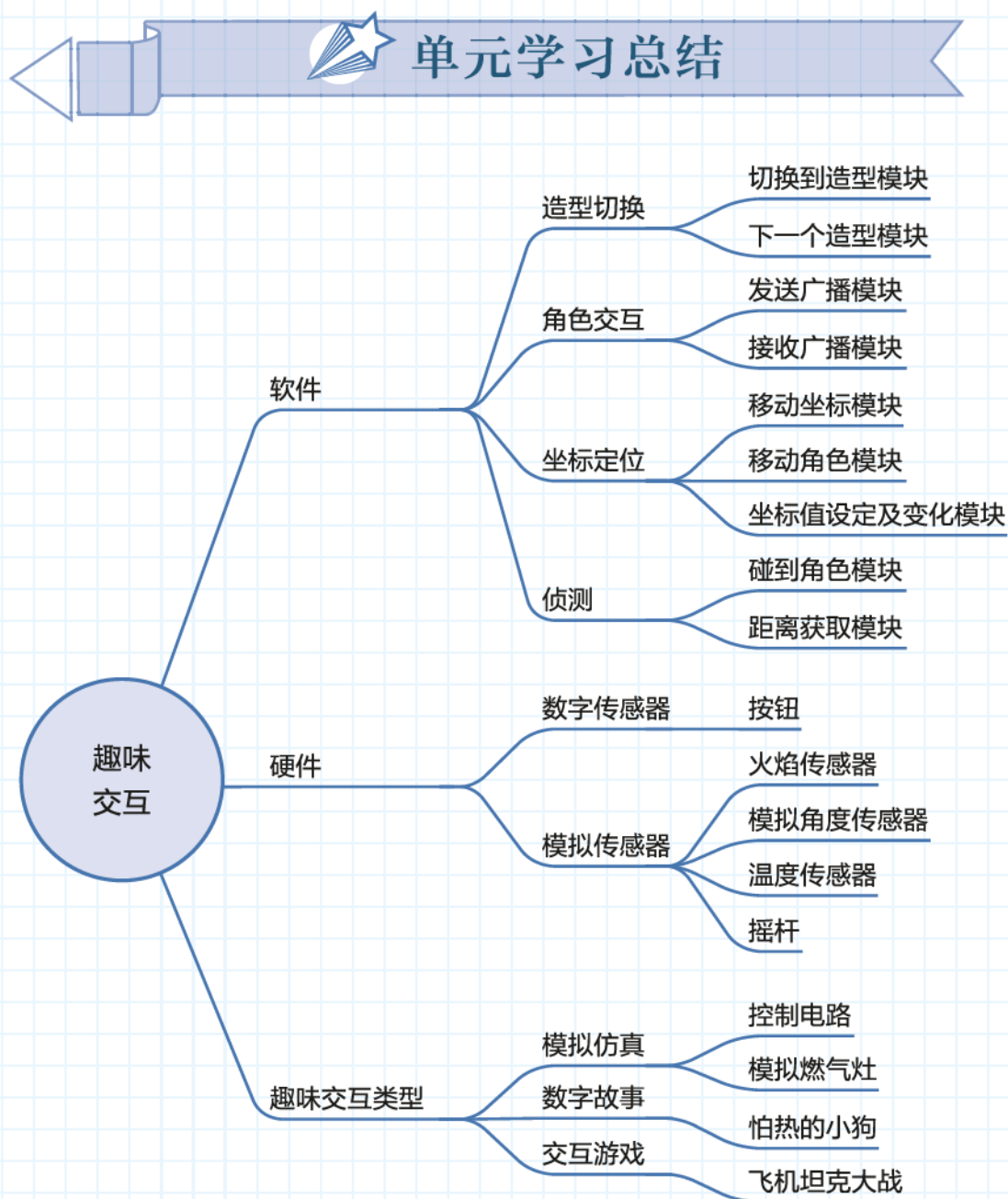
◆ 设置角色及造型

◆ 魔术游戏的程序编写

任务三：魔术游戏的测试与优化

◆ 测试中出现的问题及原因分析

◆ 优化方案



# 附录

## 附录1 项目设计方案

作品名称	
作品功能	
可选方案与选择	
基本思路与关键问题	
器材清单	
运行流程	
实现过程	

附录2 音符、音调与频率的对应表

低音	音符 音调	1 <sup>#</sup>	2 <sup>#</sup>	3 <sup>#</sup>	4 <sup>#</sup>	5 <sup>#</sup>	6 <sup>#</sup>	7 <sup>#</sup>
	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
中音	音符 音调	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
高音	音符 音调	1 <sup>#</sup>	2 <sup>#</sup>	3 <sup>#</sup>	4 <sup>#</sup>	5 <sup>#</sup>	6 <sup>#</sup>	7 <sup>#</sup>
	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

附录3 Arduino 编程手册（加粗的为教科书中用到的内容）

程序结构	基本结构	<code>loop()</code>
		<code>setup()</code>
	控制结构	<code>if, if...else...</code>
		<code>for</code>
		<code>switch case</code>
		<code>while, do...while</code>
		<code>break</code>
		<code>continue</code>
		<code>return</code>
		<code>goto</code>
		相关语法
	<code>{</code>	
	<code>//, /**/</code>	
	<code>#define</code>	
	<code>#include</code>	
运算符	算术运算符	<code>=</code>
		<code>+, -</code>
		<code>*, /</code>
		<code>%</code>
	比较运算符	<code>==, !=</code>
		<code>&lt;, &gt;, &lt;=, &gt;=</code>
	布尔运算符	<code>&amp;&amp;,   , !</code>
	指针运算符	<code>*, &amp;</code>
	位运算符	<code>&amp;</code>
		<code> </code>
		<code>^</code>
		<code>~</code>
		<code>&lt;&lt;, &gt;&gt;</code>
	复合运算符	<code>++, --</code>
		<code>+=, -=, *=, /=, &amp;=,  =</code>

续表

变量	范围	HIGH, LOW
		INPUT, OUTPUT
		true, false
		整型常量 (int), 浮点型常量 (float)
	数据类型	void
		boolean
		char, unsigned char
		byte
		int, unsigned int
		word
		long, unsigned long
		short
		float, double
		string, String
		array
		数据类型转换
	byte()	
	int()	
	word()	
	long()	
	float()	
变量作用域	static (静态变量)	
	volatile (易变变量)	
	const (不可改变变量)	
辅助工具	sizeof()	
	PROGMEM	
基本函数	数字 I/O	pinMode()
		digitalWrite()
		digitalRead()
	模拟 I/O	analogReference()
		analogRead()

续表

基本函数	模拟 I/O	<b>analogWrite()</b>
	高级 I/O	<b>tone()</b> , <b>noTone()</b>
		<b>shiftOut()</b> , <b>shiftIn()</b>
		<b>pulseIn()</b>
	时间	<b>millis()</b> , <b>micros()</b>
		<b>delay()</b> , <b>delayMicroseconds()</b>
	数学	<b>min()</b> , <b>max()</b>
		<b>abs()</b>
		<b>constrain()</b>
		<b>map()</b>
		<b>pow()</b>
		<b>sqrt()</b>
	随机数	<b>sin()</b> , <b>cos()</b> , <b>tan()</b>
		<b>randomSeed()</b> , <b>random()</b>
	位运算	<b>lowByte()</b> , <b>highByte()</b>
		<b>bitRead()</b> , <b>bitWrite()</b>
		<b>bitSet()</b>
		<b>bitClear()</b>
		<b>bit()</b>
	中断	<b>attachInterrupt()</b> , <b>detachInterrupt()</b>
		<b>interrupts()</b> , <b>noInterrupts()</b>
	串口通信	<b>Serial</b>
		<b>begin()</b>
		<b>available()</b>
		<b>read()</b> , <b>write()</b>
		<b>flush</b>
		<b>print()</b> , <b>println()</b>
<b>peak()</b>		
<b>serialEvent()</b>		



## 后 记

为全面落实立德树人根本任务，着力发展学生的核心素养，根据《普通高中课程方案（2017年版）》的精神，我们按照《普通高中信息技术课程标准（2017年版）》的要求对高中信息技术教科书进行了修订。

本书的修订由张义兵、钟柏昌、张禄、韩蕾、闫妮、李婷婷、刘正云等直接参与。在修订过程中，得到了许多专家、学者和老师们的指导与帮助。李艺、董玉琦等为本书的编写提供了理论上的指导。叶琛、曹恒来、徐劲梅、傅骞等审阅了本书修订稿的部分章节，并提出了宝贵意见。在此，我们对所有关心、支持本书编写与修订的专家、学者和老师们表示衷心的感谢。

本书选用了一些图片和文字资料，对相关作者和出版社，我们一并表示诚挚的谢意。

编者

2019年6月