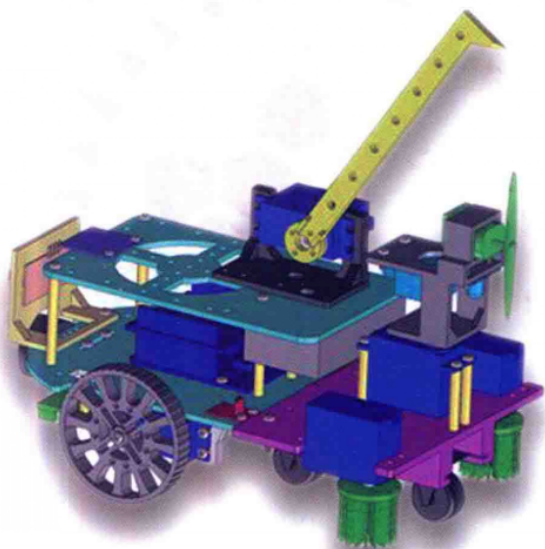


智能机器人系列创新实践教材
新工科机器人设计制作应用型人才培养精品教材

Arduino

机器人制作、编程与创新应用

陈勇志 © 编 著



西南交通大学出版社

图书在版编目 (C I P) 数据

Arduino 机器人制作、编程与创新应用 / 陈勇志编著
· 一成都: 西南交通大学出版社, 2020.11

智能机器人系列创新实践教材 新工科机器人设计制
作应用型人才培养精品教材

ISBN 978-7-5643-7763-2

I. ①A… II. ①陈… III. ①智能机器人 - 教材
IV. ①TP242.6

中国版本图书馆 CIP 数据核字 (2020) 第 204458 号

智能机器人系列创新实践教材
新工科机器人设计制作应用型人才培养精品教材

Arduino Jiqiren Zhizuo、Biancheng yu Chuangxin Yingyong
Arduino 机器人制作、编程与创新应用

陈勇志 编著

责任编辑 李华宇

封面设计 何东琳设计工作室

出版发行 西南交通大学出版社
(四川省成都市金牛区二环路北一段 111 号
西南交通大学创新大厦 21 楼)

邮政编码 610031

发行部电话 028-87600564 028-87600533

网址 <http://www.xnjdcbs.com>

印刷 四川森林印务有限责任公司

成品尺寸 170 mm × 230 mm

印张 9.25

字数 138 千

版次 2020 年 11 月第 1 版

印次 2020 年 11 月第 1 次

书号 ISBN 978-7-5643-7763-2

定价 30.00 元

课件咨询电话: 028-81435775

图书如有印装质量问题 本社负责退换

版权所有 盗版必究 举报电话: 028-87600562

前 言

Arduino 主要是以 AVR 单片机为核心控制器的单片机应用开发板，开发人员为了使没有单片机基础的人员也能制作自己想做的东西，便为 Arduino 开发了许多常用的函数和应用库，这样就可以不用去操作寄存器，使操作简单起来。例如比较 5 个数据的最小值，在 Arduino 中直接使用 min 函数即可，而不需要多次使用条件语句进行比较，简化了大部分代码。因此可以说，学习 Arduino 其实是快速开发机器人的一种方法。

机器人是一个典型的光机电算一体化系统，它融合了光学、机械、电子、传感器、通信、计算机软硬件和人工智能等众多先进技术，是目前世界各国高校进行创新思维训练、创新技能训练、工程实践训练最理想的平台之一。本书以 Arduino UNO 开发板控制的机器人为例来介绍 Arduino 的开发与应用。

本书以文娱机器人产品的设计和开发为主线，应用系统工程的设计理念，将各个专业的技术和模块与机器人系统集成，循序渐进地开发和制作文娱机器人产品，最终开发和制作出一个具有中等复杂程度的机器人产品。利用本教材给的案例和作业，可以学习 Arduino 的编程和程序的调试。从一开始，了解机器人的传感器设置与机器人设计，到后面自行设计程序，需要大家多做多试，相互之间多加讨论，共同学

习，共同进步。

机电控制在现代工业中占有非常重要的地位，小到无人机，大到工厂里的大型机床，都是属于机电控制这方面的。学习 Arduino，熟练掌握 Arduino，对大家后期学习单片机或者学习其他编程语言（如 STM32、C#、Python 等）都有很大的帮助，除了基础知识的掌握，另外尤其重要的是锻炼逻辑严密的编程思维。在现在创客文化比较流行的环境下，Arduino 的简单易用，使得追求个性的人有了更多的发挥空间，人们可以用它来实现自己的创意，满足自己的个性追求。

本书第 1~3 章介绍的是 Arduino 相关基础知识，如果读者已经有一定的 Arduino 开发经验或者学过 Arduino 机器人制作和编程入门教程，可以直接从第 4 章开始学。

本书不仅适合大学生学习，同时也适合有激情、有较高领悟力的中学生学习，还适合 Arduino 机器人制作爱好者自主学习。本书可作为高等院校机器人制作课或者工程实践课的教材，也可作为实验室新进成员培训与学习的参考书。

陈勇志

2020 年 8 月



第 1 章	数字电子技术基础	001
1.1	数字量与模拟量	001
1.2	数 制	001
1.3	逻辑代数基础与门电路	002
第 2 章	C 语言基础	004
2.1	数据类型及相关语法	004
2.2	关键字	006
2.3	运算符	006
2.4	控制结构	008
第 3 章	初识 Arduino	012
3.1	Arduino 概述	012
3.1.1	Arduino 的硬件接口	014
3.1.2	Arduino 程序开发软件	020

3.1.3	Arduino 库文件加载	022
3.1.4	Arduino 程序上传	024
3.1.5	Arduino 的应用	025
3.2	Arduino 的电气特性	026
3.2.1	AVR 内核结构	027
3.2.2	内部存储器	029
3.2.3	AVR 外围功能	029
3.2.4	封 装	033
3.3	Arduino 基础示例	035
3.3.1	Arduino 控制 LED 灯闪烁	035
3.3.2	Arduino 呼吸灯	038
3.3.3	Arduino 串口通信	041
第 4 章	传感器技术基础	047
4.1	传感器初识	047
4.1.1	传感器的定义与应用	047
4.1.2	传感器的分类	048
4.1.3	传感器的选用原则	051
4.2	常用传感器及其应用	052
4.2.1	双路巡线传感器	052
4.2.2	舵 机	056
4.2.3	超声波传感器	060
4.2.4	蜂鸣器	063
4.2.5	红外线接收器	067
4.2.6	OLED 显示屏	072

4.2.7	蓝牙模块	075
4.2.8	QTI 传感器	082
第 5 章	Arduino 机器人设计与组装	086
5.1	Arduino 机器人组装	087
5.1.1	Arduino 机器人的零件安装	087
5.1.2	机器人各零件的螺纹连接	089
5.2	Arduino 机器人的电路	091
第 6 章	工程训练	094
6.1	概 述	094
6.2	基于超声波的模块化训练	094
6.2.1	移动、避障与警报	094
6.2.2	避障距离可视化	099
6.2.3	超声波跟随	103
6.3	基于巡线传感器的模块化训练	106
6.3.1	自动巡线（直线曲线篇）	106
6.3.2	自动巡线与智能绕障	108
6.3.3	自动巡线（直角锐角拐弯篇）	113
6.4	基于红外遥控的模块化训练	116
6.4.1	红外遥控+自动避障	116
6.4.2	状态显示+红外遥控+自动避障	119
6.5	基于蓝牙的模块化训练	124
6.6	灭火机器人	128
6.6.1	灭火机器人设计	128

6.6.2	火焰传感器与继电器模块	129
6.6.3	电路设计	131
6.6.4	程序设计	131
附录 1	Arduino 机器人接口配置	135
附录 2	Arduino 常用函数清单	136
参考文献		139

数字电子技术基础

1.1 数字量与模拟量

数字量的特点是其变化在时间上和数量上都是离散的、不连续的。例如，使用一个普通的按钮开关，它的状态只有开与闭，它的变化总是发生在一系列离散的瞬间。常把数字量的信号称为数字信号，并且把产生或可读取数字信号的电路称为数字电路。

模拟量的特点是其变化在时间上或数值上是连续的。例如，呼吸灯现象中，通过 LED 灯的电流值是随时间连续变化的。常把模拟量的信号称为模拟信号，并且把产生或可读取模拟信号的电路称为模拟电路。

从所用数学工具的角度来看数字电路与模拟电路，数字电路的特征是使用布尔代数（true 与 false）；模拟电路的特征是使用微分方程、拉斯变换等。

1.2 数 制

数制就是我们常说的十进制、二进制、八进制、十六进制等。

1. 十进制

基数为十：0~9，位权为： 10^i 。

例如： $256.7=2\times 10^2+5\times 10^1+6\times 10^0+7\times 10^{-1}$ ，其中 2、5、6、7 这些数字就是基数。

2. 二进制

基数为二：0~1，位权为： 2^i 。

例如： $1001=1\times 2^3+0\times 2^2+0\times 2^1+1\times 2^0=9$ （十进制）。

3. 八进制

基数为八：0~7，位权为： 8^i 。

例如： $0312=3\times 8^2+1\times 8^1+2\times 8^0=202$ （十进制）。

注：八进制，Octal，缩写为 OCT 或 O，常以数字 0 开始表明该数字是八进制。

4. 十六进制

基数为十六：0~F，位权为： 16^i 。

例如： $0X73=7\times 16^1+3\times 16^0=115$ （十六进制 73 转十进制就是 115）。

注：常以 0X 开头表示十六进制。

1.3 逻辑代数基础与门电路

布尔代数（逻辑代数）：描述客观事物逻辑关系的数学方法，其变量取值只有两种，true（1）或 false（0），称为二值逻辑。

二值逻辑中，每个逻辑变量的取值只有“0”和“1”两种可能；此时 0、1 不表示大小，只代表两种不同的逻辑状态。在 Arduino 的二值逻辑中，用 1 表示高电平，用 0 表示低电平。

逻辑代数的基本运算有三种：与、或、非。

实现基本逻辑运算和复合运算的单元电路称为门电路。常用的门电路有与门、或门、非门、与非门、或非门、异或门、与或非门等。图 1-3-1 所示为与、或、非三种逻辑电路。只有决定结果的全部条件同时成立时，结果才发生，这种因果关系叫作逻辑与。决定结果的各个条件中只要有一个满足，结果就会发生，这种因果关系叫作逻辑或。条件具备时，结果不发生；条件不具备时，结果一定发生，这种因果关系叫作逻辑非。

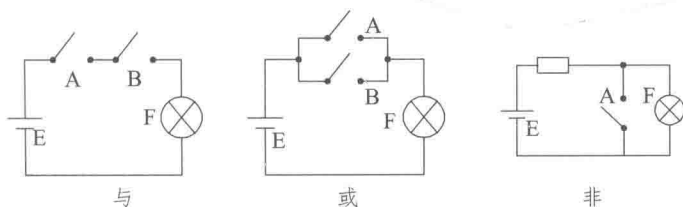


图 1-3-1 逻辑电路

在 Arduino 编程中，逻辑与用“&&”表示，逻辑或用“||”表示，逻辑非用“!”表示，常用 if 语句与以上三种逻辑运算（与、或、非）进行条件的判断。例如：

```
if((i > 3) && (i < 8)){ 动作一 }
```

如果 i 值大于 3 且小于 8，那么就执行动作一；

```
if((i < 3) || (i > 8)){ 动作二 }
```

如果 i 值小于 3 或者大于 8，那么就执行动作二；

```
if(i != 0){ 动作三 }
```

如果 i 值不等于 0，那么就执行动作三。

C 语言基础

2.1 数据类型及相关语法

1. int (整型)

int 一般用于定义整数的变量，例如：

```
int ledPin = 10;
```

语法是

```
int var = val;
```

var 表示变量名

val 表示赋给变量的值，整数的范围为 -32768 到 32767 ($-2^{15} \sim 2^{15}-1$)。

int 还会用于定义整型的数组，例如：

一维数组：

```
int a[5] = {1, 2, 3, 4, 5};
```

二维数组：

```
int a[5][3]={{80,75,92}, {61,65,71}, {59,63,70}, {85,87,90}, {76,77,85}};
```

2. float (单精度浮点型)

float 一般用于定义有小数点的变量，例如：

```
float c=3.14;
```

语法是

```
float var = val;
```

var 表示变量名称;

val 表示赋给变量的值, 浮点数的取值范围在 $-3.4028235 \times 10^{38} \sim 3.4028235 \times 10^{38}$ 。

3. byte (字节型)

一个字节存储 8 位无符号数, 从 0 到 255。

例如:

```
byte b = B10010; // "B" 是二进制格式 (B10010 等于十进制 18)
```

“//” 表示单行注释, “/**/” 表示多行注释, 注释是不会被编译器识别的。

4. unsigned char (无符号字符型)

unsigned char 一般用于定义十六进制的数组, 例如:

```
unsigned char data1[3]= {0x55,0x06,0x01};
```

在 Arduino 串口通信应用章节中, 对于发送十六进制数组的程序设计, 需要开发者懂得如何定义。

5. String (定义字符串)

例如:

```
String text1 = "This string";
```

6. void (函数声明)

void 只用在函数声明中。它表示该函数将不会被返回任何数据到它被调用的函数中。例如:

```
void LED_work(void)
```

```
{
```

```
int ledPin = 10;
```

```
...
```

```
}
```

2.2 关键字

1. HIGH 与 LOW

在 Arduino 开发中，HIGH 代表高电平，LOW 代表低电平。HIGH 与 LOW 只发生在 Arduino 的数字引脚定义中。HIGH 可以用 1 来表示，LOW 可以用 0 来表示。

2. INPUT 与 OUTPUT

在 Arduino 开发中，INPUT 代表输入，OUTPUT 代表输出，也是发生在 Arduino 的数字引脚定义中。

2.3 运算符

运算符见表 2-3-1。

表 2-3-1 运算符

算术运算符	比较运算符	复合运算符	布尔运算符	指针运算符
= (赋值)	== (等于)	++自加	&& (逻辑与)	* 指针运算符
+ (加)	!= (不等于)	--自减	(逻辑或)	&地址运算符
- (减)	< (小于)	+=复合加	!(非)	
* (乘)	> (大于)	-=复合减		
/ (除)	<= (小于等于)	*=复合乘		

算术运算符	比较运算符	复合运算符	布尔运算符	指针运算符
% (求余)	>=(大于等于)	/=复合除		
		&=复合与		
		=复合或		

1. 算术运算符与比较运算符

“`i = 1`”与“`i == 1`”是不同含义的。前者是代表将 1 赋值给 `i`；而后者代表 `i` 等于 1，一般用于作为判断语句中的条件，也就是说，“`i == 1`”是常带括号出现的。

例如：

```
if(i == 1)
{动作一}
```

或者

```
while(i == 1)
{动作一}
```

2. 复合运算

(1) 自加“++”。

例如：

```
i++; //相当于 i = i + 1
```

(2) 自减“--”。

例如：

```
i--; //相当于 i = i - 1
```

(3) 复合加“+=”。

例如：

```
i+=5; //相当于 i = i + 5
```

(4) 复合减“-=”。

例如：

```
i-=5; //相当于 i = i- 5
```

3. 布尔运算

(1) && (逻辑与)。

例如：

```
((i > 10) && (i < 20)); //代表 i 大于 10 且小于 20
```

(2) || (逻辑或)。

例如：

```
((x > 0) || (y > 0)); //代表 x 大于 0 或者 y 大于 0
```

2.4 控制结构

1. #include 包含

#include 用于包含头文件，例如：

```
#include <Servo.h>
```

2. #define 宏定义

例如：

```
#define ledPin 10 // 编译器在编译时会将任何提及 ledPin 的地方替换成数值 10
```

3. if 语句

示例 1：

```
if(条件一)  
{执行动作一}
```

示例 2：

```
if(i < 2) {i*=5;} //如果 i 小于 2，那么 i=i*5
```

4. if...else... 语句

示例 1:

```
if(条件一)
    {执行动作一 }
else
    {执行动作二}
```

示例 2:

```
if(条件一)
    {执行动作一 }
else if
    {执行动作二}
else
    {执行动作三}
```

5. for 循环

语句结构为:

```
for (初始化部分;条件判断部分;数据递增部分)
{
    // 动作一
}
```

示例:

```
for (int i=0; i<= 255; i++)
{
    analogWrite(PWMPin, i); //模拟输出的动作
    delay(10);
}
```

6. switch case 判断语句

一般要配合 break 使用。

示例: