

3.1 Arduino 简介

Arduino 是一款便捷灵活、方便上手的开源电子原型平台,包含硬件(各种型 号的 Arduino 开发板)和软件(Arduino IDE)两部分。

硬件部分指可以用于电路连接的 Arduino 开发板,通过连接各种各样的传感器来感知环境,控制灯光、马达等输出设备来反馈、影响环境,进而实现人机交互。Arduino 开发板型号很多,常用的有 Arduino Uno、Arduino Nano、Arduino Leonardo (如图 3-1)、Arduino Mega 2560 等。以 Arduino Leonardo 为例,常用的引脚有数字引脚(取值范围是 0~13)、模拟引脚(A0~A5)、GND、5V。

在实际使用中通常需要用到扩展板,它是最常用的 Arduino 外围硬件之一,主要用于连接其他传感器,可以节约连接线和其他模块,如图 3-2 所示。在面包板上接插





038



	100.00			
118.	-THENE .	and the second sec	International Property in the local division of the local division	_
10100		100.00	- her makers	
10021044	_	and the second s	and the second se	
1000		and and a second s	and the second s	
10.	100.00	and the second second	C. Desta Barris	
100	19419	The Country of Country		
see 8.	They want to	Million and	Burning Street	
100.0	111-649-0	and the second		
198	114.0	and the second sec		
60.0	1.1	and a second sec		
	111-09	Concernence of the second		
	188-19	11 Belleville		
		Addet for Programs Ridge Taplong Hannel Tanata UAA Santa Contra Santa Contra Santa Contra Santa Kalan Istana Sa Sa Santa		

元件固然方便,但还需要一定的电子知识 来搭建各种电路及转接,而使用扩展板只 需要通过连接线,把各种模块接插到扩展 板上即可,从而快速地搭建出新的项目。

软件部分则是计算机中的程序开发环 境——Arduino IDE,如图 3-3 所示。用 户在 IDE 中编写程序代码,编译成二进 制文件,上传到 Arduino 开发板执行程序。

3.1.1 程序示例

本节以 Arduino Leonardo 开发板的 具体使用过程为例,介绍 Arduino 软硬件 的操作过程。首先,在官方网站(https:// www.arduino.cc/)下载并安装 Arduino 的集成开发环境(Integrated Development Environment, IDE)。

为了测试 Arduino Leonardo 电路板 是否正常工作,加载一个最简单的代码 (例如 Blink),同时,通过简单的操作 熟悉如何将程序上传到开发板中运行。 在 Arduino IDE 中选择"文件"→"示

例"→01.Basics→Blink 命令,如图 3-4 所示,打开 Blink 代码(案例 3-1)。 Arduino Leonardo 电路板上有标有 L 的 LED 灯,这段测试代码就是让这个 LED 灯 闪烁。

```
【案例 3-1】LED 灯闪烁(Blink)
void setup() {
    pinMode(LED_BUILTIN, OUTPUT);
}
```

```
void loop() {
    digitalWrite(LED_BUILTIN, HIGH); // LED 灯亮(HIGH表示高电平)
    delay(1000); // 等待 1000ms
    digitalWrite(LED_BUILTIN, LOW); // LED 灯灭(LOW表示低电平)
    delay(1000); // 等待 1000ms
}
```

Arduino 代码包括两个核心函数 void setup()和 void loop()。void setup() 是程序的初始化函数,只执行一次。void loop()是程序主体,反复执行。 pinMode(pin,mode)函数用于配置引脚为输入或者输出模式,是一个无返回值的函 数。pinMode()函数有两个参数 pin 和 mode, pin 表示要配置的引脚代号,mode 表示设置的模式(INPUT 输入模式和 OUTPUT 输出模式),其中 INPUT 用于读 取信号,OUTPUT 用于输出控制信号。digitalWrite(pin,value)函数用于设置引脚 的输出电压为高电平或低电平,也是一个无返回值的函数。digitalWrite()函数有 两个参数 pin 和 value, pin 表示引脚,value 表示输出的电平:HIGH(高电平)或 LOW(低电平)。delay(ms)是延迟时间函数,参数 ms 是整数,表示时间(以毫秒 为单位)。因此,该程序实现"点亮→等待一秒→熄灭→等待一秒",如此反复, 实现闪烁的效果。

写完一段代码后需要验证一下代码有无错误。单击"验证" [√],如图 3-5 所示, 界面下方显示"编译完成",标识代码无误。以后写代码的过程中,输入完代码, 都需要校验一下,然后再上传到 Arduino 开发板中。

在将程序上传到 Arduino 开发板 之前,还要查验 Arduino 开发板型号 以及相应的串口。选择"工具"→"开 发板"→ Arduino Leonardo 命令,如 图 3-6 所示,设置开发板型号。

然后设置串口,选择"工具"→"端 口"→COM3(Arduino Leonardo) 命令,如图3-7所示。注意:每台计 算机设备管理器显示的串口不同。



039

第3章 Arduino

交互艺术装置

实现技术

040

1.89.10	236,900		
20 E	(UMB)(A	1011	
	ALCONOMIC		
	2.01	Inches.	
the part	ACRES .	DAVIDARIAN TO	The second
-	AC168	Car-948-0	
-	without class makes from any line	anter .	
	State 'Amountaine'	12	
	WI TOW MAN INVENT		animal
and property	SHITTER		2040 putney put
	BRIE WITH HAT	1	ellis de millio
100 A			
14 T C 14	ALC: N. A. S. COMPANY, MICH.	ALC: NAME OF COMPANY	
			And the second s

最后,单击"上传"按钮,如图3-8所示, 界面下方显示"上传成功",表示程序已 经上传到 Arduino 开发板中。

程序上传完毕后就可以看到程序运行的效果: Arduino Leonardo 板上标有L的 LED 灯在闪烁,如图 3-9 所示。

3.1.2 串口监视器

串口监视器是开发环境 IDE 中的重要 功能,可以监控串行接口(简称串口)的 通信状况,是能看到程序具体执行情况的 输出窗口。Arduino 开发板与计算机的通信 方式就是串行通信,在 Windows中,串口 称为 COM,并以 COM1、COM2 等编号 标识不同的串口。如果想让 Arduino 开发 板回传数据,则需要在编写程序的时候添 加相关的代码指令,这就是串口监视器的 指令。


```
第
Arduino
```

041

在 Arduino IDE 中,选择"工具"→"串口监视器"命令,打开串口监视器, 如图 3-10 所示。

N 100 100 3	(N was				5)				
80 B	norda agrice	104-4							3.2
_	The state	(Deliveration)	-						
	#110.5m	114-140-14	and the month in						
in sectors	002188	(0++0+0++)							
ALCONOM DA	WEIGH / BETTER COMMENSION								
-	199 'Ashini Lenach' BD 'OMI (Ashini Lenach' BD 'DMI)		a In 12 for 14 for						
-	BUB 'HONY and' ROWING		⁰ mittig (pr. m. 11.0)						
			1						
		_							
CHARGE STREET		CONTRACTOR OF							
				11 Juni					
				 (2) 0 + 0 (3) 0 + 0 (4) 0 + 0 (5) 0 + 0 (6) 0 + 0 (7) 0 + 0 (8) 0 + 0	II Dor Harring	MU	1 - 900 2318	- 4	123
			Annual I						

在案例 3-1 代码的基础上,修改代码让串口控制灯亮(案例 3-2),在串口 监视器中输入字符 1 点亮 LED 灯,输入字符 0 熄灭 LED 灯。setup()函数中增加串 口初始化代码 Serial.begin(9600),同时在串口监视器的右下角要选择波特率 9600b/s (串口监视器中默认选项为波特率 9600b/s)。在串口监视器的输入框中输入字符 1,单击"发送"按钮,可以将输入框内输入的数据送到 Arduino 的串口。Arduino 的串口通过 Serial.read()函数读取数据,读到字符 1 点亮 LED 灯,读到字符 0 熄灭 LED 灯。

【案例 3-2】串口控制灯亮

```
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
  Serial.begin(9600);
}
void loop() {
  if (Serial.available()>0)
  {
    char ch=Serial.read();
    if (ch=='1')
        digitalWrite(LED_BUILTIN,HIGH);
```

```
else if (ch=='0')
    digitalWrite(LED_BUILTIN,LOW);
}
```

COM3	19	(D) (X)
		1.8.8
and the second se		
Wills would.		
COURT Date theating (AU	e - seas gate	- 218
	业生里	
	лшыт	

交互艺术装置 实现技术

}

在案例 3-2 代码的基础上,修改代 码让串口具备程序调试功能(案例 3-3)。 通过 Serial.println()函数从串行端口输 出数据,跟随一个回车。在串口监视器中 输入字符1点亮 LED 灯,同时串口监视 器的中心显示区域输出 LED 灯的状态; 输入字符0熄灭 LED 灯,同时串口监视 器的显示区域输出"Hello world!",如 图 3-11 所示。

```
【案例 3-3】串口调试功能
```

```
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
  Serial.begin(9600);
}
void loop() {
if (Serial.available()>0)
  {
    char ch=Serial.read();
    if (ch=='1')
    {
      digitalWrite(13,HIGH);
      Serial.println(digitalRead(LED BUILTIN));
    }
    else if (ch=='0')
    {
      digitalWrite(LED_BUILTIN,LOW);
      Serial.println("Hello world!");
    }
```

}

}

3.1.3 面包板

面包板是一种搭建电路的电子元件。因为早期制作电路时,人们会用到家中做 面包的板子,并在板子上连接导线等,面包板因此得名。面包板是制作临时电路和 测试原型的常见元件,其上有很多小插孔,各种电子元器件可根据需要随意插拔, 节省电路的组装时间,而且元件可以重复使用,非常适合电子电路的组装、调试和 训练,能同时容纳最简单和最复杂的电路,因而被广泛使用。面包板使用热固性酚 醛树脂制造,板底有金属条,使用前应确定哪些元件的引脚需要连在一起,再将要 连接在一起的引脚插入同一组的5个小孔中。

面包板有各种规格,如170孔(35mm×47mm),400孔(85mm×55mm), 800孔(165mm×55mm)等。每块面包板的四边都有凸起和凹槽,同一大小规格 的面包板可以进行拼接使可用空间变大。如图 3-12 所示,面包板的上下两侧分别 有两列插孔,一般是作为电源引入的通路。上方第一行标有"+"的一列有5组插孔, 每组5个(内部5个孔连通),均为正极。上方第二行标有"-"的一列有5组插孔, 每组5个(内部5个孔连通),均为接地。面包板下方第一行与第二行结构同上。 如需用到整个面包板,通常将"+"与"+"用导线连接起来,"-"与"-"用导 线连接起来。

连接孔分为上下两部分,用来插接原件和跳线,是主工作区。同一列中的5个 插孔(即a-b-c-d-e,f-g-h-i-j)是互相连通的;列和列(即1~30)之间以及 凹槽上下部分(即e和f)是不连通的。在面包板的中间有一个长长的凹槽,说明

上下两部分是断开的,中间的距离可以 插入标准窄体的芯片,同时方便用镊子 伸到集成电路下面取出集成电路。

使用面包板时有以下注意事项:

(1)必须修理整齐多次使用过的集成电路的引脚;采用边安装边调试的方法。

(2) 连线要求紧贴在面包板上,以

免因碰撞弹出面包板,造成接触不良。

(3)在布线过程中,要求把各元器件放置在面包板上的相应位置以及将所用 的引脚号标在电路图上,保证调试和查找故障的顺利进行。

(4)避免集成电路的粗暴插拔,以免引脚折在插孔内;不要将金属线折断在 插孔内。

使用 LED 灯、电阻、电源和导线,在面包板上制作一个简易小型串联电路,将3个 LED 灯和3个电阻与电源串联起来,如图 3-13 所示。

使用 LED 灯、电阻、电源和导线,在面包板上制作一个简易小型并联电路,将1个电阻与电源串联,然后与3个 LED 灯串联起来,如图 3-14 所示。

将 Arduino 与面包板结合可以拓展 Arduino 的连接能力,快速搭建更复杂的 电路结构,以便复杂程序的实现,案例 3-4 通过在面包板上搭建并联电路并与 Arduino 的 PIN 口连接,进行了多个 LED 灯闪烁控制的实验,连线如图 3-15 所示。

【案例 3-4】Arduino 与面包板控制多个 LED 灯顺序点亮并倒序熄灭

```
void setup()
{
                            // 设置 8~10 号引脚为输出状态
  for(int i=8;i<11;i++)</pre>
   pinMode(i,OUTPUT);
}
void loop() {
                            // 依次计每个引脚的 LED 灯点亮并持续 1s
  for(int i=8;i<11;i++){</pre>
   digitalWrite(i,HIGH);
   delay(1000);
  }
    for(int i=10;i>7;i--) { // 倒序熄灭每个引脚的 LED 灯
   digitalWrite(i,LOW);
    delay(1000);
   }
}
```


图 3-14 使用面包板搭建的简易并联电路连接图

图 3-15 多个 LED 灯顺序点亮并倒序熄灭的硬件连接图

045

第3章

3.2 传感器

3.2.1 概述

Arduino 可以连接丰富的传感器,用来感知环境和接收用户的交互行为,实现 "人 – 机 – 环境"的交互。将常用的感知环境的传感器汇总,如图 3–16 所示,有 感知环境光的传感器,如光敏传感器、光线传感器、灰度传感器、颜色传感器、 IIC 颜色传感器;感知环境声音的传感器,如声音传感器;感知环境温湿度的传感器, 如温湿度传感器、线性温度传感器、土壤湿度传感器;感知水情况的传感器,如水 位传感器、水流量传感器、水温传感器、雨水传感器、水蒸气传感器;感知特殊环 境的传感器,如火焰传感器、酒精传感器、气体烟雾传感器、PM2.5 传感器。传感 器的输出类型和感应范围如表 3–1 所示。

	@1,0 Ma	
(1)光敏传感器	(2)光线传感器	(3) 灰度传感器
(4)颜色传感器	(5) IIC 颜色传感器	(6)声音传感器
	1.10	
(7)温湿度传感器	(8)线性温度传感器	(9)土壤湿度传感器
图 3-16 环境传感器示例		

047

Arduino

(10)水位传感器 (11)水流量传感器 (12)水温传感器	ŗ
 (13)雨水传感器 (14)水蒸气传感器 (15)火焰传感器 	-
(16)酒精传感器 (17)气体烟雾传感器 (18)PM2.5传感;	器

表 3-1 环境传感器详情

序号	名 称	说 明	输出类型	测量范围
1	光敏传感器	可见光感应	模拟量	1 ~ 5000
2	光线传感器	环境光感应	模拟量	0 ~ 1023
3	灰度传感器	色彩感应	模拟量	0 ~ 511
4	颜色传感器	色彩感应	模拟量	红、绿、蓝数值
5	IIC 颜色传感器	色彩感应	模拟量	色温、亮度、红、绿、蓝 及白光数值
6	声音传感器	声音感应	模拟量	0 ~ 1023
7	温湿度传感器	温湿度感应	模拟量	0~50℃,误差2℃,湿度 范围随温度变化
8	线性温度传感器	温度感应	模拟量	- 40℃~ +110℃, 误差 0.5℃