



《无线电》编辑部 编

Arduino

智能硬件开发入门

30 个基础制作实例助你快速上手



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS



Arduino 智能硬件开发入门

30 个基础制作实例助你快速上手

《无线电》编辑部 编



人民邮电出版社

北京

图书在版编目(CIP)数据

Arduino智能硬件开发入门: 30个基础制作实例助你快速上手 / 《无线电》编辑部编. — 北京: 人民邮电出版社, 2016.8

(i创客)

ISBN 978-7-115-42715-1

I. ①A… II. ①无… III. ①单片微型计算机—程序设计 IV. ①TP368.1

中国版本图书馆CIP数据核字(2016)第132225号

内 容 提 要

“i创客”谐音为“爱创客”，也可以解释为“我是创客”。创客的奇思妙想和丰富成果，充分展示了大众创业、万众创新的活力。这种活力和创造，将会成为中国经济未来增长的不熄引擎。本系列图书将为读者介绍创意作品、弘扬创客文化，帮助读者把心中的各种创意转变为现实。

Arduino是如今最流行的开源智能硬件开发平台，也是创客最喜欢的工具之一。它应用广泛，功能强大，降低了学习单片机的门槛，不仅是电子爱好者和电子专业学习人员学习的热门，也受到艺术家、软件开发者的喜爱。借助Arduino，你可以轻松创造出能够进行人机互动的智能硬件和互动艺术作品。

本书是《超炫的35个Arduino制作项目》的升级版，最大变化是由黑白版变成彩色版，然后也调整了部分项目，帮助初学者更好地通过制作实例来学用Arduino。读者可以完整地了解到Arduino应用项目的设计思路和制作方法，从中得到详实的制作资料和思路启发。

本书通过实例全面地介绍了Arduino的基础应用，读者除可掌握扩展版、传感器的使用方法，还能制作出鼠标、键盘、数字示波器、电子秤、签到记录器、数码相机、气体监测平台、宠物互动机器人、智能交互灯、自动浇花系统等具体项目，获得非凡成就感。本书操作步骤清晰、图片简明、可操作性强，内容不仅适合电子爱好者阅读，也适合创客空间、学校参考来开办工作坊和相关课程。

◆ 编 《无线电》编辑部

责任编辑 周 明

责任印制 周昇亮

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京瑞禾彩色印刷有限公司印刷

◆ 开本: 690×970 1/16

印张: 11.25

字数: 235千字

2016年8月第1版

2016年8月北京第1次印刷

定价: 49.00元

读者服务热线: (010)81055339 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广字第8052号

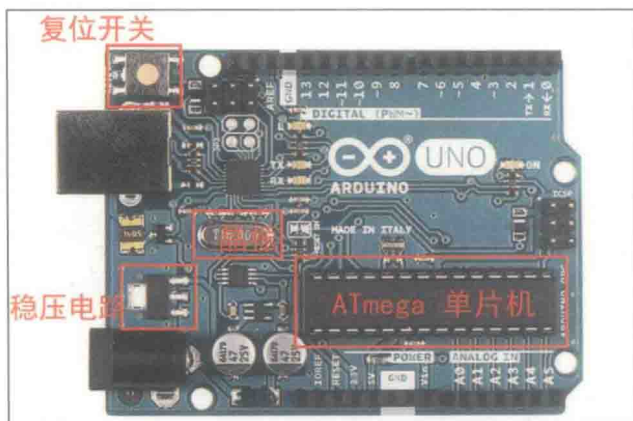
前言

Arduino 为什么这么红?

如果浏览5~10年前的电子制作资料,你一定可以看到五花八门的单片机选型、各种手工焊制的电路板、纷繁复杂的飞线……而近几年的各种基于单片机的电子制作,则统一了许多,各种不同领域、不同功能的电子DIY作品,大都采用了同一种控制模块为核心——Arduino。

那么Arduino到底是什么呢?是一种新的控制芯片,还是一种新的开发软件呢?它又有什么优势,可以在短短几年时间内几乎一统了电子DIY的江湖呢?

首先我们看看Arduino的实物图,图0.1所示是最常见的一种Arduino实物。图中已经根据功能将各个模块区分出来,相信对单片机开发稍有了解的朋友一定已经发现,这不就是一个以ATmega单片机最小系统为核心的控制板吗?不错,Arduino就是以ATmega单片机为控制核心的单片机控制板,板上除了ATmega328最小系统电路外,还包含了稳压电路、USB转串口电路、一些指示用的LED,以及一些扩展用的电路插座。



■ 图 0.1 最常见的 Arduino 控制板——Arduino UNO

仅仅一个单片机最小系统，为何能风靡全球呢？

下面我们以图表的形式来比较一下单片机裸机、市面上一般的单片机开发板和 Arduino。

表0.1 ATmega328裸机、ATmega328开发板、Arduino的对比

	ATmega328 裸机	ATmega328 开发板	Arduino
运算性能	相同	相同	相同
成本	最低	较高	较低
体积	最小	较大	适中
自由度	最大	很多开发板集成了键盘、数码管、跳线开关等，占用了很多端口	除单片机正常运行所需占用的端口外，其他端口全部留给了用户使用
标准度	完全符合 Atmel 公司公布的芯片资料	大多由各个供应商自主开发，各不相同	完全符合 Arduino 组织公布的标准
工作条件	需要自己搭建最小系统电路	通过 USB 或 COM 端口与 PC 连接即可使用	通过 USB 线连上 PC 即可使用

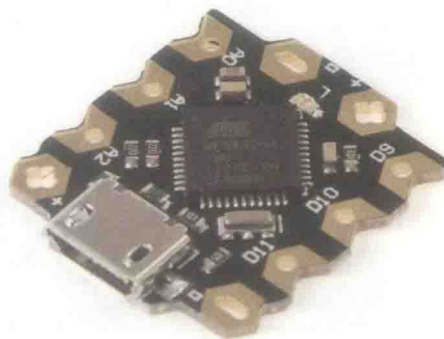
表 0.1 中列出了 3 种选择方案的最直观比较（表中没有比较购买渠道的便利性，随着网络购物的普及，这一点已经不再成为爱好者们需要特别关注的问题），看起来似乎 Arduino 并没有特别的优势，那么为什么 Arduino 会得到如此热度的追捧呢？

要回答这个问题，我们必须首先回顾一下 Arduino 的发展历程。2005 年，意大利北部小镇伊夫雷亚（Ivrea）一家高科技设计学校的老师 Massimo Banzi（国内创客把他亲切地称为“板子大叔”），为了能给学生们提供一种便宜、好用的微控制器平台，与当时在这所学校做访问学者的西班牙籍芯片工程师 David Cuartielles 合作设计了最初的 Arduino 控制板。随后 Arduino 便开始迅速地在欧洲流行起来，并且逐渐将春风吹到了世界各地。

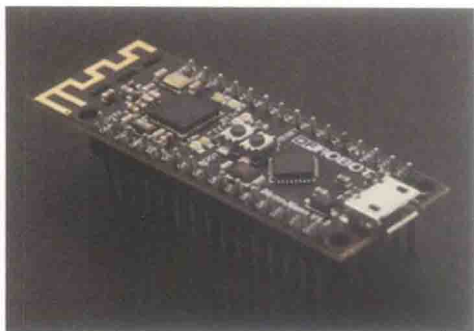
Arduino 独有的优势

Arduino 独有的几种优势表现在下列方面。

开放性：Arduino 是起步比较早的开源硬件项目。各种开源项目目前已经得到广泛的认可和大范围的应用。它的硬件电路和软件开发环境都是完全公开的，在不从事商业用途的情况下，任何人都可以使用、修改和分发它。这样不但可以使用户更好地理解 Arduino 的电路原理，更可以根据自己的需要进行修改，比如由于空间的限制，需要设计异形电路板（见图 0.2），或是将自己的扩展电路与主控制电路设计到一起（见图 0.3）。



■ 图 0.2 国内某厂商开发的异形 Arduino 控制板



■ 图 0.3 国内某厂商开发的集成了蓝牙模块的 Arduino

易用性：对于稍微有心的人，不论基础如何，只要他有兴趣，拿到 Arduino 之后的 1 小时之内，应该就可以成功运行第一个简单的程序了。Arduino 与 PC 的连接采用了当下最主流的 USB 连接，你可以像使用一款智能手机一样，把 Arduino 与电脑直接连起来，而不再需要再额外安装任何驱动程序。而且 Arduino 的开发环境软件也非常简单，一目了然的菜单仅提供了必要的工具栏，除去了一切可能会使初学者眼花缭乱的元素，你甚至可以不用阅读手册便实现例程的编译与下载。

交流性：对于初学者来说，交流与展示是非常能激发学习热情的途径。但有些时候，你用 AVR 做了个循迹小车，我用 PIC 做了个小车循迹，对单片机理解还不是特别深刻的初学者，交流上恐怕就会有些困难。而 Arduino 已经划定了一个比较统一的框架，一些底层的初始化采用了统一的方法，对数字信号和模拟信号使用的端口也做了自己的标定，初学者在交流电路或程序时非常方便。

丰富的第三方资源：Arduino 无论硬件还是软件，都是全部开源的，你可以深入了解底层的全部机理，它也预留了非常友好

的第三方库开发接口。秉承了开源社区一贯的开放性和分享性，很多爱好者在成功实现了自己的设计后，会把自己的硬件和软件拿出来与大家分享。对于后来者，你可以在 Arduino 社区轻松找到自己想要使用的一些基本功能模块，比如舵机控制、PID 调速、A/D 转换等。一些功能模块供应商也越来越重视 Arduino 社区，会为自己的产品提供 Arduino 下的库和相关教程。这些都极大地方便了 Arduino 开发者，你可以不必拘泥于基本功能的编写，而把更多的精力放在自己想要做的功能设计中去。

是的，从专业嵌入式开发的技术角度来说，Arduino 并不是第一选择，为了尽可能地照顾初学者甚至是电子开发的门外汉，Arduino 定制了很多底层的设计，自然也损失了很多嵌入式开发的灵活性和效率性，这也是许多经验丰富的嵌入式设计人员对 Arduino 嗤之以鼻的原因之一。那么 Arduino 的定位究竟在何处呢？

Arduino 的定位

Arduino 最初确实是为嵌入式开发的学习而生，但发展到今天，它已经远远超出了嵌入式开发的技术领域。有些人将 Arduino 称为“科技艺术”，很多电子科技领域以外的爱好者，凭借丰富的想象力和创造力，也设计开发出了很多有趣的作品。在国内，Arduino 更多还是作为一种嵌入式学习工具和电子开发原型模块出现，但是它的魅力绝不仅仅如此，它完全可以作为一种新“玩具”，甚至新的艺术载体，来吸引更多领域的人们加入 Arduino 的神奇世界。

Arduino 的硬件

是不是已经迫不及待，想要开始你的 Arduino 之旅了？赶快到购物网站上搜一搜 Arduino……然后是不是被各种不同的 Arduino 控制板搞花眼了？

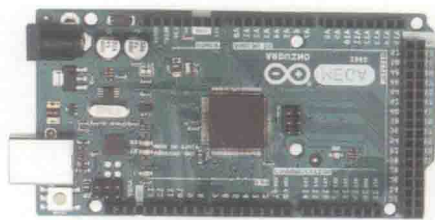
Arduino 并不是一成不变的，每隔一段时间，Arduino 官方组织就会发布最新的设计，在原有基础上进行升级，让爱好者们更方便地使用。同时，为了满足不同层次的开发要求，Arduino 也推出了不同版本的控制板设计，让我们来认识一下使用最多的几种设计。

Arduino UNO（见图 0.4）：UNO 版本应该说是 Arduino 的基础板，也是初学者的第一选择。它提供了基本的数据接口，对初学者来说有足够的存储空间，无论是用来读取传感器，还是驱动电机，或者与电脑通信，都可以轻松地胜任。



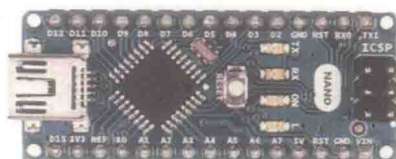
■ 图 0.4 Arduino UNO

Arduino MEGA 2560（见图 0.5）：这一版可以算是 UNO 的升级版，各种接口的数量比 UNO 要多，而且其搭载的单片机也要比 UNO 搭载的更高级一些，运算速度也更快。如果你觉得 UNO 的硬件资源无法满足你的需求，那么你可以尝试一下 MEGA，当然它的价格也会比 UNO 更高一些。



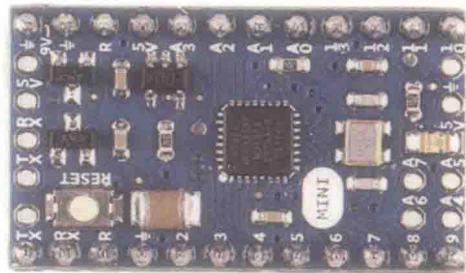
■ 图 0.5 Arduino MEGA 2560

Arduino Nano（见图 0.6）：这个可以说是 UNO 的简化版，优先考虑了体积上的优化，尽量将体积做到最小，可以满足一些手持设备或体型微小的设备使用。当然功能上相比 UNO 也略有缩水。



■ 图 0.6 Arduino Nano

Arduino Mini（见图 0.7）：这是目前体积最小的一款 Arduino 控制板，可用于对体积要求比较高的项目，比如可穿戴设备等。



■ 图 0.7 Arduino Mini

也许有些心急的朋友已经要问了：哪一款可以控制步进电机？哪一款可以实现无线通信？哪一款可以演奏音乐？……

如果我告诉你都不可以，你会不会很失望？

Arduino 作为一种控制芯片，所起的作用类似于我们的大脑，大脑是无法自行运动或是发声的。大脑所起的作用类似于指挥官，Arduino 也是如此，想要完成一些实际任务，离不开外围电路的支持。

很多人又要头疼了，电子电路完全不懂怎么办？其实现在这已经不是什么大问题了。对于电子专业的学生，或是想要学习电子电路的爱好者，当然要一切靠自己，查资料、买元器件、做 PCB、焊电路……这其中当然会有很多辛苦，可从中学到的知识无疑也是装入自己脑袋的。

如果不想在电子电路方面花费太多的精力，也没有问题。现在很多厂商都开发了各种各样的 Arduino 外围功能电路，无论是电机驱动、无线通信、音乐播放，还是读取各种各样物理信号（压力、速度、倾角、方向等）的传感器，都应有尽有，而且使用方便，可以直接通过标准接口（UART、I²C 等）连到 Arduino 控制板上工作。

Arduino 的软件开发方法

说完了硬件，我们接下来谈谈很多人都头疼不已的软件开发吧。

一般的嵌入式代码都是 main 函数中有一个死循环，程序不断地反复执行，单片机在每个循环内读取各个外部端口的数据，然后根据这些数据来做出相应的策略判断，再

把指令通过外部端口传送出去，达到控制外部设备的目的。

Arduino 的程序运行方式与此差不多，但代码结构略有不同。它的代码中没有 main 函数，而是使用了两个不同的必备函数：setup、loop。

相信很多朋友已经猜出来了，Arduino 代码是把初始化的代码放在了 setup 函数中，而把需要不断循环执行的代码放在了 loop 函数中。结构与传统的嵌入式程序虽然不同，但设计思想还是一致的。

Arduino 的代码语法与 C++ 类似，也包含了对类和对象的支持。即便没有任何代码编写的基础，也可以很快上手。关于 Arduino 的代码编写，无论是 Arduino 的官方网站，还是网络上都有相当丰富的资料，这里不再赘述。

很多 Arduino 制作还要涉及另外一种程序的编写，就是上位机程序，一般是 PC 端的程序。你可以采用任何你所熟悉的高级语言进行开发，接收 Arduino 控制板所传递过来的数据（通过串口传输），在屏幕上显示输出或写入文件。VB、VC++、Delphi、Flash Script、Processing 等都能满足你的要求。

胡泊

2016 年 4 月 28 日

CONTENTS

目录

第1章 初识 Arduino

01	树莓派、Arduino、传统单片机开发板，用谁好	2	3.2	仿真项目：电压表	16
1.1	分类	2	3.3	结束语	19
1.2	理念的区别	3	04	一起用面包板自制 Arduino 吧	20
1.3	性能的差距	5	4.1	硬件连接	20
1.4	总结	6	4.2	通过下载器下载程序	21
02	图形化编程软件——ArduBlock	7	4.3	通过串口下载程序	22
2.1	ArduBlock 的今世前缘	7	05	用洞洞板走进 Arduino 之门	24
2.2	ArduBlock 软件的下载与安装	8	5.1	电路	25
2.3	按钮与 LED 的互动实例	9	5.2	元器件和制作	25
2.4	ArduBlock 与 Arduino 联机运行	10	5.3	安装编程平台	26
2.5	电位计与舵机互动实例	11	5.4	烧入引导程序	26
2.6	ArduBlock 使用心得	12	5.5	安装 USB 转串口线的驱动程序	27
03	零硬件成本玩 Arduino	13	5.6	试跑 Arduino IDE	27
3.1	仿真项目：直流电机控制	14	5.7	试试字符液晶	29
			5.8	再补一句	29
			06	Arduino+Tick Tock 扩展板编程入门	30
			6.1	Arduino 扩展板简介	30

6.2	扩展板的焊接过程	31
6.3	程序下载方法	36
6.4	Tick Tock 扩展板编程	37
6.5	图形化编程	41
6.6	启用 Tick Tock 上的传感器	45

第 2 章 基础应用

07	超简易冷热指示装置	50
7.1	设计	50
7.2	温度测量	50
7.3	map 函数和 LED 控制	51
08	二进制温度计	53
8.1	温度计算	53
8.2	温度计的电路设计	53
8.3	Arduino 程序设计	55
09	锂电池电量测量装置	57
9.1	电路原理	57
9.2	程序设计	58
10	播放音乐超简单	60
10.1	制作简易的 SD 读卡器	61
10.2	音频格式的转换	61
10.3	连接硬件、下载程序	61
11	驱动触摸屏	63
11.1	连接方法	64
11.2	触摸屏的使用与编程	64

12 Step by Step 学红外遥控 67

12.1	红外接收	67
12.2	红外发射	70
12.3	让电视遥控器控制机顶盒	72
12.4	用电视遥控器控制电脑实现 无线键盘	74

13 轻松与 Flash 进行网络通信 .. 76

13.1	Arduino 与 Flash 的通信 过程	77
13.2	调试	78

14 基于 LabVIEW 的温度监测系统 80

14.1	Arduino 下位机部分	80
14.2	LabVIEW 上位机部分	82
14.3	实验与演示	84

15 具有温度补偿的超声波测距系统 85

15.1	超声波传感器 SR-04	85
15.2	Arduino 部分设计	86
15.3	LabVIEW 程序设计	88
15.4	小结	90

16 让 Arduino 成为一个 Web 服务器 91

16.1	电路原理	92
16.2	制作	93
16.3	编程	94

第3章 创意制作

17	自制鼠标	98	23	自己打造签到记录器	129
17.1	制作步骤.....	99	23.1	方案设计.....	129
17.2	程序设计.....	100	23.2	具体功能.....	130
17.3	未来的功能扩展.....	101	23.3	硬件连接.....	132
18	开源键盘	103	23.4	软件设计.....	133
19	用最常见的材料制作闪电云	107	24	自制气体监测平台	134
19.1	连接电路.....	108	24.1	硬件连接.....	134
19.2	外形制作.....	109	24.2	软件编写与烧录.....	136
19.3	闪电云程序.....	111	24.3	总结.....	137
20	超声波感应式简易电子琴	112	25	扩展网络摄像头的 I/O 端口	138
20.1	硬件准备.....	112	25.1	方案设计.....	138
20.2	制作过程.....	113	25.2	硬件连接.....	139
21	自制数字示波器	116	25.3	PC 端软件.....	140
21.1	硬件工作原理.....	116	26	Arduino+TTL 摄像头自制拉风数码相机	143
21.2	搭建试验电路.....	117	26.1	硬件连接.....	144
21.3	程序设计.....	118	26.2	程序与使用.....	146
21.4	调试与使用.....	120	27	逗猫机器人	148
21.5	制作过程.....	121	27.1	立即开始.....	148
22	自制电子秤	124	27.2	需要准备的工具和元器件.....	148
22.1	工作原理.....	124	27.3	在面包板上搭建这个项目.....	149
22.2	硬件连接.....	125	28	姿态控制智能交互灯	
22.3	程序编写.....	126		LightBox	151
22.4	结束语.....	128	28.1	电路连接.....	152

28.2 亚克力外壳和 RGB LED 部分	153
---------------------------------	-----

29 花园管家——自动浇花 系统	157
-----------------------------------	------------

29.1 浇灌单盆花的制作步骤	157
-----------------------	-----

29.2 浇灌多盆花的制作步骤	159
-----------------------	-----

30 解放双手，呵护植物——自动浇 花系统的实作与改进	162
--	------------

30.1 初体验	162
----------------	-----

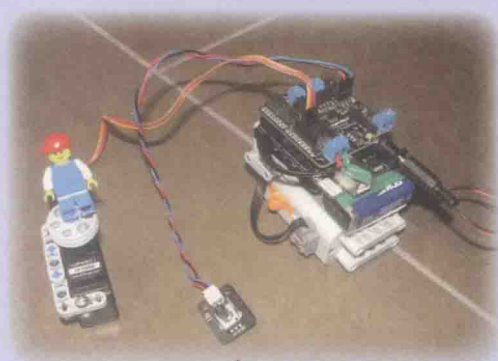
30.2 再创造	163
----------------	-----

30.3 终制作	164
----------------	-----

30.4 做后感	166
----------------	-----

第 1 章

初识 Arduino



- 01 树莓派、Arduino、传统单片机开发板，用谁好
- 02 图形化编程软件——ArduBlock
- 03 零硬件成本玩 Arduino
- 04 一起用面包板自制 Arduino 吧
- 05 用洞洞板走进 Arduino 之门
- 06 Arduino+TickTock 扩展板编程入门



树莓派、Arduino、传统单片机开发板，用谁好

◇望雪

几十年前的电子爱好者，最喜欢的就是电烙铁、面包板和收音机；十几年前，出现了单片机，于是玩具就成了电烙铁、面包板和单片机；到了 2015 年，贴片技术的不断普及，让面包板不再那么有用武之地，经济的发展也让现成的板卡价格降到了一个合理的范围，购买现成的电路板，甚至自己打样

电路板来使用，已经成了一个趋势。现在，我们面前的选择就空前丰富，一方面，是以 Arduino 和树莓派为首的开源硬件阵营，另一方面，则是以 STM32、51 和 S3C2440 为首的传统单片机开发板阵营。笔者根据自己的使用体验，与读者简单谈一谈它们在实际应用上的区别。



1.1 分类

首先要明确两个问题。

第一个，我在这里把电子爱好者的硬件对象简单分成开源硬件阵营和传统开发板阵营，并不是指传统开发板不开源，事实上它们也是全开源的，之所以这么叫，是因为“开源硬件”是一个从国外兴起、在国内大热的概念，Arduino 和树莓派都是从国外引进来的，而那些传统开发板很多是我们本土设计开发的。它们虽然都是开源的，但是在使用和开发的理念上有所不同，所以才有必要区分开来。需要注意的是，并不是说国外来的就是开源硬件，国内的就是传统开发板，重

点在于理念的不同。目前，国产的开源硬件也正如雨后春笋一样冒出来了。

第二个，无论属于哪个阵营，它们用的 CPU 都得分成 MCU（微控制器，或者称为单片机）和 MPU（微处理器）两类，二者的本质区别在于 MMU（内存管理单元），也就是对于虚拟内存空间的支持。树莓派和 S3C2440 就属于 MPU 类的，而 Arduino 和 STM32 就属于 MCU 类的。它们在运算能力上有着巨大的差距。

基于上面的分类，如果把它放到一个天梯图内，我们就能得到这样一张开源硬件的比较图（见图 1.1）。



■ 图 1.1 开源硬件比较图

需要注意的是，这张图只比较了它们的纯运算性能，越靠上的，运算性能越强，但是这张图并不是按照比例画的，实际上，树莓派的运算性能可以达到 Arduino 的 100 倍。接下来，我就根据上面的两种分类方法来讲讲它们的区别。

1.2 理念的区别

开源硬件和传统开发板的区别，最重要的其实也就是理念的差别。开源硬件注重的是快速开发，而传统的开发板则注重精通其中的技术细节。举个例子，比如，要用 DS1302 和 12864 液晶屏来做一个简单的液晶屏万年历，我们来对比一下 51 平台和 Arduino 平台的不同。

第一步是连接硬件，我们假设硬件的连接已经完成了，因为在这方面，两边没有什么区别。

第二步一般都是上网先搜索相应的硬件驱动程序，这个环节上 51 单片机系统和 Arduino 的差别就出来了。Arduino 有一个库的概念，一般都是把库直接解压到 Arduino 安装目录的 Libraries 文件夹下

就可以了，通常不需要关心库里面的代码是如何写的，只要使用就可以了。比如，DS1302 的库，使用很简单，在开始的地方声明一下连接，然后创建一个对象。

```
/* 接口定义 */
uint8_t CE_PIN   = 5;
uint8_t IO_PIN   = 6;
uint8_t SCLK_PIN = 7;
/* 创建 DS1302 对象 */
DS1302 rtc(CE_PIN, IO_PIN, SCLK_PIN);
```

但是 51 单片机系统就不一样，51 单片机系统没有很明确的封装，网上找到的，也不是库，而是驱动代码。甚至很多时候，还是集成在一个工程文件里的，如果需要使用，还得自己把它从里面剥离出来。比如，这是网上找到的 51 驱动程序的一部分。

```
//DS1302 写入子程序
void DS1302_Write(uchar temp)
{
    uchar i;
    CLK=0; // 将 DS1302 时钟脉冲拉低
    _nop_(); // 延时一指令周期
    RST=1; // RST 置高电平
    _nop_(); // 延时一指令周期
    for(i=0; i<8; i++) // 循环 8 次
    {
        DAT=temp&0x01; // 向 DS1302 写入一字节数据
        _nop_(); // 延时一指令周期
        CLK=1; // 拉高时钟脉冲
        temp>>=1; // 右移一位
        CLK=0; // 拉低时钟脉冲
    }
}
```

值得注意的是，里面出现了很多的“_nop_();”，也就是空指令，用于延迟一个时钟。很多时候，如果使用的是增强型的 51 单片机，其运行速度可能会比外部设备

的运行速度快很多，这样就必须在驱动程序中人工加入延时。而且不同的单片机，运行速度也不一样，延时还得单独进行调整。DS1302 还好，延时不怕高，只怕低，也就是调试的时候尽可能加大延时就不会出问题，但是 DS18B20 就很“矫情”了，多了不行，少了也不行，这样程序写起来就很麻烦。这其实也就是 51 单片机系统写驱动时的一个特点，别的地方拿来的东西还不一定能用，得自己修改才行。假设你用同样的方法搞定了 12864 的驱动，然后是主代码的实现，也就是读取时间并且显示到液晶屏上。

用 Arduino 的朋友，事情就很简单了，首先从 DS1302 获取时间：Time t = rtc.time();，然后格式化并输出：

```
sprintf(buf, "%04d-%02d-%02d\n%02d:%02d:%02d", t.yr, t.mon, t.date, t.hr, t.min, t.sec);
LCDA.DisplayString(0,0, buf, sizeof(buf));
```

这样就完成了，但是如果是 51，噩梦就又要开始了。首先，读取时间没什么问题，用驱动程序提供的函数把它写入 t 变量里面：DS1302_GetTime(&t);，然后就麻烦了，如果用的是无字库的液晶屏，驱动程序是不提供显示字符的函数的。怎么办？自己实现咯。首先得有字形，用取模软件得到了字模，也就是一堆数据，如图 1.2 所示。

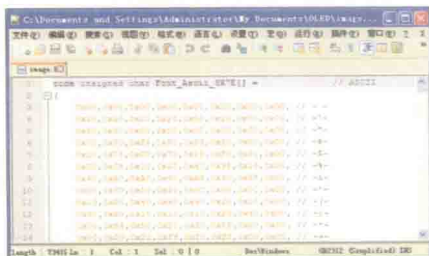


图 1.2

其次是显示的程序。根据液晶控制器的数据手册分析，需要两个函数，一个是打点函数，另一个是显示字符的函数。假设找到的驱动程序里已经包括了打点，自己只要写显示字符的函数就可以了，结合之前的取模数据，不难写出这样的函数：

```
void LCD_Display_Chrc(u8 left,u8 top,u8 chr)
{
    u8 x,y;
    unsigned int ptr;
    ptr=(chr-0x20)*8;
    for (y=0;y<8;y++)
    {
        for (x=0;x<5;x++)
        {
            if (((Curr_Font[ptr] <<x)&0x80)==0x80)
                OLED_Point(left+x, top+y,color);
            ptr++;
        }
    }
}
```

之后，还需要一个显示字符串的函数。如果了解 C 语言里的指针和字符串，还是比较简单的。下面是一个很简单的显示字符串的函数：

```
void LCD_String(u8 left,u8 top,u8* s)
{
    u8 x=0;
    while(*s)
    {
        LCD_Display_Chrc(left+x,top,*s++);
        x+=6;
    }
}
```

别看这么点代码量，如果对 GRAM 结构、指针没有比较透彻的了解，一天都不

一定搞得定。实现了这些之后，就可以和 Arduino 一样写出最终的显示代码了。

从上面这个例子，我们可以看出，两边的区别很明显。Arduino 很适合快速开发，实现自己想要的效果，而不需要掌握太多的专业知识，这也是 Arduino 快速流行起来的原因，不同行业的人都可以加入进来，参与基于 Arduino 产品的开发。但是，这并不是说单片机不好，学单片机的可以自豪地说我的基础知识掌握得很扎实，因为单片机的程序都得脚踏实地来写，在写的过程中，就对那些代码和相关的知识有了深入的理解。有了这些知识，如果给你换一套硬件，比如把 12864 LCD 换成 320240 的 TFT，只要有数据手册，照样可以写程序，但是 Arduino 如果找不到库，那就“玩脱”了，是吧？

笔者个人认为，如果是以学习为目的的话，传统的单片机更值得推荐；如果是玩玩，为了完成作品，Arduino 不失为一个好选择，毕竟更加方便，能够更加专注于实现自己的创意，这也是很多创客的理念。

1.3 性能的差距

讲完两大阵营的理念区别，再来讲一讲另一个爱好者中存在的疑问，那就是树莓派和单片机或者树莓派和 Arduino 的选择问题，其实也就是 MPU 和 MCU 的选择问题。

这里又得分两种情况来讨论，一类是没有学过任何东西，想直接拿树莓派入门的，另一类是在玩过 Arduino 或者 51 单片机之后想要玩玩更高端平台。

先讲讲第一类。对于第一类玩家，如果已经有了编程基础，可以直接试着玩树莓派，如果只是把树莓派当成一个小型 PC 或者低

功耗服务器，完全没有问题。对于编程水平好的，我更推荐 CubieBoard CC80 或者 Jetson TK1，因它们的性能会强大很多，玩起来会更加舒服。如果没有编程基础或者想要玩软硬结合的，还是建议从 PC 编程或者 Arduino、单片机开始学习。

对于第二类，选择就不那么简单了。首先要知道，在过去（其实抛开开源硬件，现在也是一样），MPU 是一种很复杂的东西，一般得熟练掌握单片机之后才能去接触 MPU。而且 MPU 通常涉及 Linux 内核的修改、Linux 驱动的开发、Linux 软件的编写这类事情，需要的知识储备很多。对于一般爱好者来说，玩 MPU 完全就是天方夜谭。但是 MPU 又是那么吸引人，数百兆的主频，数十兆的存储空间，是 MCU 根本无法想象的。MPU 就像是一台真正的 PC，而 MCU 只能算个玩具。国外的开源硬件大佬自然也看见了这一点，于是就有了树莓派这样的产品。树莓派就是融合了开源硬件易用性和 MPU 强大功能的产品。现在，对于想要玩高端平台的爱好者，最主要的选择就是树莓派和 STM32（虽然它们根本不是一个量级的东西，但是由于 2440 类的东西开发难度太大，业余爱好者一般较少涉足）。那么我们就来对比一下，树莓派和 STM32 分别能做什么，都能做的东西开发起来有什么区别见表 1.1。

再来看看对于都可以做的东西，两者开发上有什么区别。以网络视频监控为例吧。STM32 的开发流程是这样的：硬件选型→设计 PCB→焊接→调试硬件→编写 DCMI 和 RMII 驱动→移植 TCP/IP 协议栈→调整摄像头驱动→编写网页服务器程序→完成。