

单片机C语言程序设计 实训100例

— 基于 **Arduino+Proteus** 仿真

◆ 彭伟 编著



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

单片机 C 语言程序设计 实训 100 例

——基于 Arduino+Proteus 仿真

彭 伟 编著



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书基于 Atmel Studio+Arduino IDE 开发平台和 Proteus 硬件仿真平台,精心编写了 100 个 Arduino C 语言程序设计案例并分别提出了难易适中的实训目标。

全书基础设计类案例涵盖 Arduino 最基本的端口 I/O、定时控制、A/D 转换、中断控制、串口通信等程序设计;硬件应用类案例涵盖编/解码器件,串/并与并/串转换器件,LED 显示及驱动器件,字符/图形液晶,TFT 彩屏,实时日历时钟器件,I²C/SPI/1-Wire 总线器件,电动机,温/湿度、压力、雷达传感器,SD 卡等;综合设计类案例包括大量实用型项目设计,如多功能电子万年历、计算器、电子秤、密码锁、大幅面 LED 点阵屏设计、交流电压检测、K 型热电偶及铂电阻温度计、GPS、红外遥控、测距、温室监控、游戏设计、Arduino Web 应用等,因全书大量案例基于库进行设计,故很多复杂的程序均可通过便捷的代码实现。

本书可作为本、专科院校学生学习实践 Arduino C 语言程序设计技术的教材,也可作为工程技术人员或单片机技术爱好者的学习参考书或工具书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

单片机 C 语言程序设计实训 100 例:基于 Arduino+Proteus 仿真 / 彭伟编著. —北京:电子工业出版社,2019.8
ISBN 978-7-121-36788-5

I. ①单… II. ①彭… III. ①单片微型计算机—C 语言—程序设计 IV. ①TP368.1②TP312.8

中国版本图书馆 CIP 数据核字(2019)第 112709 号

策划编辑:曲 昕

责任编辑:康 霞

印 刷:山东华立印务有限公司

装 订:山东华立印务有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×1 092 1/16 印张:27.25 字数:715 千字

版 次:2019 年 8 月第 1 版

印 次:2019 年 8 月第 1 次印刷

定 价:99.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888, 88258888。

质量投诉请发邮件至 zllts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式:(010) 88254468, quxin@phei.com.cn。

前 言

开始准备撰写本书文稿前，首先完成了所规划的大部分案例代码的编写及调试工作。

至开始落笔时，已彻底颠覆了此前我对 Arduino 的偏见！

为什么呢？

首次见到“其貌不扬”的 Arduino UNO，发现其微处理器只不过是一片 AVR 微处理器 (Atmega328)。我曾出版过一本关于 AVR 的书，所使用的微处理器以 Atmega16A 为主，部分使用 Atmega128，7 年过去了再来写一本书，使用的芯片却是其内部资源要弱于前者的 Atmega328。选择这块“其貌不扬”的板子能轻松实现复杂的应用开发吗？能做出“很酷”的东西吗？

对此我首先是怀疑的。然而，由于被 Arduino 应用的洪流所“裹挟”，恰巧又承担了一个创意电子课程开发项目，再加上电子工业出版社曲昕编辑的支持与期待，使我终于加入了 Arduino 的大军。

一年前，到新加坡的一次考察学习中，见到很多同学也在大量使用 Arduino，做出了很多优秀的作品放在展厅里，我开始对 Arduino 有了不一样的认识。

本书中程序的调试是我的 4 本仿真实训系列教材中完成最快的，这当然有前面几个分册作为基础，但我认为最根本的原因是，Arduino 有极其简易的开发环境、类 C 的开发语言、大量优秀的内置库及第三方库，还有极其活跃的网上社区等。使用 Arduino 开发，真的会让你感觉：快、易、稳！

另外，当你遭遇 Arduino UNO 资源紧张时，高端的 Arduino(Genuino) mega2560，甚至 Arduino Due(ARM32)均可作为候选器件。

如同前面讲到的，Arduino 会完全颠覆你最初对它的误解与偏见！

一旦使用 Arduino，一位嵌入式与单片机项目工程师、一位创意电子爱好者的“逆袭”就将从此开始……

值得一提的是，你的起步将完全是“零成本”的，因为本书的所有 Arduino 案例均在仿真环境下实现。

本书提供完整的仿真电路与可执行 HEX 程序，同时提供部分主框架代码，读者可到华信教育资源网 (<http://www.hxedu.com.cn>) 下载。

由于书中图片大多为原图扫描或系统生成后的截图，未做标准化处理，特此说明。

编著者

反侵权盗版声明

电子工业出版社依法对本作品享有专有出版权。任何未经权利人书面许可，复制、销售或通过信息网络传播本作品的行为，歪曲、篡改、剽窃本作品的行为，均违反《中华人民共和国著作权法》，其行为人应承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。

为了维护市场秩序，保护权利人的合法权益，我社将依法查处和打击侵权盗版的单位和个人。欢迎社会各界人士积极举报侵权盗版行为，本社将奖励举报有功人员，并保证举报人的信息不被泄露。

举报电话：(010) 88254396; (010) 88258888

传 真：(010) 88254397

E-mail: dbqq@phei.com.cn

通信地址：北京市海淀区万寿路 173 信箱

电子工业出版社总编办公室

邮 编：100036

目 录

第 1 章	Arduino C 语言程序设计概述	1
1.1	Arduino 简介	1
1.2	Arduino 开发环境	2
1.3	Arduino 开发语言	6
1.4	Arduino 程序结构	6
1.5	基本数据类型、有符号数与无符号数的应用及位相关操作	8
1.6	数组、字符串与指针应用	11
1.7	全局变量、静态变量、外部变量及修饰符	13
1.8	运算符的优先级	14
1.9	关于中英文、数字字符的编码处理及应用	15
1.10	Arduino 的常用函数	20
1.11	Arduino 的特殊寄存器	21
1.12	Arduino 相关在线资源	22
第 2 章	Proteus 操作基础	24
2.1	Proteus 操作界面简介	24
2.2	仿真电路原理图设计	25
2.3	选择元件	27
2.4	仿真运行	31
第 3 章	基础程序设计	32
3.1	闪烁的 LED	32
3.2	双向来回的流水灯	35
3.3	花样流水灯	36
3.4	LED 模拟交通灯	38
3.5	分立式数码管循环显示 0~9	40
3.6	集成式数码管动态扫描显示	42
3.7	按键控制 LED 开关	45
3.8	按键消抖处理	46
3.9	按键调节数码管闪烁增/减显示	49
3.10	继电器及双向晶闸管控制照明设备	53
3.11	数码管显示 4×4 键盘矩阵按键	54
3.12	INT0 中断计数	58
3.13	INT0~INT3 中断计数	62
3.14	Timer0 控制 LED 闪烁	65
3.15	使用 TimerOne 库实现 Timer1 控制 LED 闪烁	72
3.16	Timer1 定时器中断控制数码管动态显示	75
3.17	Timer1 定时器中断控制 5×7 LED 点阵屏显示	77

3.18	Timer2 控制门铃声音输出	79
3.19	使用 FrequencyTimer2 库实现 Timer2 控制音阶演奏	83
3.20	使用 MsTimer2 库实现 Timer2 控制交通指示灯	86
3.21	Timer1、Timer2 及 INT0 控制报警器与旋转灯	89
3.22	按键控制定时器选播多段音乐	92
3.23	用模拟与数字输出方法实现 PWM 控制 LED 渐变	94
3.24	模数转换与数码管显示	97
3.25	通过 A/D 转换实现 5 路按键检测	100
3.26	硬件及软件串口应用	103
3.27	双机串口双向通信	106
3.28	PC 与 Arduino 双向串口通信	111
3.29	Arduino 内置 EEPROM 读/写测试	117
3.30	Flash 数据访问	120
第 4 章	硬件应用	124
4.1	74HC138 译码器与反向缓冲器控制数码管显示	124
4.2	用 74HC164 驱动多位数码管显示	127
4.3	串入并出芯片 74HC595 控制数码管显示 4 位数字	129
4.4	并/串转换器 74HC165 的应用	133
4.5	用 74HC148 扩展中断	135
4.6	串行发送数据到 2 片 8×8 点阵屏滚动显示	138
4.7	数码管 BCD 解码驱动器 CD4511 与 DM7447 的应用	140
4.8	拨码开关及 74HC245 的应用	142
4.9	14 段与 16 段数码管演示	144
4.10	16 键解码芯片 74C922 的应用	148
4.11	1602 字符液晶工作于 8 位模式切换显示	150
4.12	1602 字符液晶工作于 4 位模式显示	162
4.13	LGM12864 (KS0108) 液晶屏显示测试	163
4.14	UG-2864 (SSD1306) OLED 屏显示测试	170
4.15	Nokia5110 (PCD8544) 液晶屏显示测试	175
4.16	PG160128A (T6963C) 液晶屏图文演示	178
4.17	EADOGS102 液晶屏显示测试	182
4.18	TFT 彩屏 ILI9341 显示测试	184
4.19	WS2812 串行全彩灯显示测试	190
4.20	DS1302 实时时钟应用	196
4.21	I ² C 接口时钟日历芯片 DS1307 的应用	200
4.22	日历时钟芯片 PCF8583 的应用	206
4.23	I ² C 接口存储器 AT24C04 的读/写与显示 (4 片)	210
4.24	用 I ² C 存储器设计的中文硬件字库的应用	217
4.25	I ² C 接口 DS1621 温度传感器测试	220
4.26	用 I ² C 接口芯片 PCF8574 驱动 LM044L 液晶屏	224

4.27	I ² C 接口数字电位器 AD5242 的应用	226
4.28	用兼容 I ² C 接口的 MAX6953 驱动 4 片 5×7 点阵显示器	228
4.29	用 I ² C 接口控制 MAX6955 驱动 16 段数码管显示	232
4.30	用带 I ² C 接口的 MCP23017 扩展 16 位通用 I/O 端口	235
4.31	使用 I ² C 接口器件 PCA9555 扩展接口	239
4.32	I ² C 接口 18 位 ADC 芯片 MCP3421 的应用	244
4.33	SPI 接口存储器 AT25F1024 的读/写与显示	249
4.34	SPI 接口数字电位器 AD5206 的应用测试	258
4.35	SPI 接口串行共阴驱动器 MAX7219 控制 4+2+2 数码管显示	260
4.36	8 位数码管段位复用串行驱动芯片 MAX6951 的应用	264
4.37	SPI 接口温度传感器 TC72 的应用测试	270
4.38	NTC 热敏电阻应用测试	273
4.39	温度传感器 LM35 全量程应用测试	275
4.40	8 路 12 位模数转换芯片 MCP3208 的应用	278
4.41	压力传感器 MPX4250 的应用	282
4.42	直流电动机正/反转及 PWM 调速控制	284
4.43	ULN2803 驱动单极步进电动机的正/反转	287
4.44	L298N 驱动双极步进电动机运行	292
4.45	伺服电动机控制测试	296
4.46	用 M145026 与 M145027 设计的无线收发系统	299
4.47	温/湿度传感器 DHT22 的应用	302
4.48	SRF04 雷达测距传感器的应用	307
4.49	DS18B20 温度传感器的测试	309
4.50	SD 卡 FAT 文件系统读/写测试	315
第 5 章	综合设计	324
5.1	带日历时钟及温度显示的电子万年历	324
5.2	简易计算器设计	329
5.3	电子秤仿真设计	333
5.4	简易加密电子密码锁	337
5.5	用 T6963C 液晶与 DS1302 设计的可调指针式电子钟	342
5.6	温度控制 L298N 驱动直流电动机调节转速	349
5.7	ULN2003 驱动 5 路继电器及 74HC595 控制电梯数字点阵屏滚动显示	351
5.8	用 74LS595 与 74LS154 及反向驱动器 7406 控制 16×16LED 点阵屏显示	354
5.9	红外遥控收发仿真	358
5.10	交流电压检测与数字显示仿真	363
5.11	3 端可调正稳压器 LM317 应用测试	367
5.12	GP2D12 红外测距传感器的应用	370
5.13	带液晶显示的 K 型热电偶温度计	377
5.14	用 MCP3421 与 RTD-PT100 设计的铂电阻温度计	381
5.15	T6963C 液晶显示的模拟射击训练游戏	388

5.16	简易飞船游戏.....	392
5.17	可接收串口信息的带中英文硬字库的 80×16LED 点阵屏.....	399
5.18	GPS 导航系统仿真	405
5.19	温室综合监控系统仿真设计.....	409
5.20	Arduino Web 服务应用.....	415

第 1 章 Arduino C 语言程序设计概述

Arduino 是全球最流行的开源硬件，有着优秀的硬件开发平台，引领了硬件开发的趋势。其简单的开发方式使开发者能够更关注创意与实现，更快完成自己的项目开发，大大节约了学习成本，缩短了开发周期。

正是因为 Arduino 的种种优势，越来越多的硬件开发者已经或准备开始使用 Arduino 来开发他们的项目、产品；越来越多的软件开发者使用 Arduino 进入硬件、物联网等开发领域；大学里计算机与电子信息、自动化，甚至艺术类专业，也纷纷开设 Arduino 相关课程。

本书将引导读者轻松学习 Arduino 内置资源、扩展硬件开发技术及综合类项目设计技术，另外给出 100 个在 Atmel Studio+Arduino 开发平台编写的 Arduino C 程序，同时提供所有的 Proteus 仿真电路，可到华信教育资源网下载学习。阅读本书要求已初步掌握 Arduino C 程序设计基本技术，本章仅介绍使用 C 语言设计 Arduino 应用系统必须参考和重点掌握的内容，这些内容会给读者阅读、调试、研究全书案例及进行设计实训提供帮助。

1.1 Arduino 简介

Arduino 由一个欧洲开发团队于 2005 年冬季开发，其成员包括 Massimo Banzi、David Cuartielles、Tom Igoe、Gianluca Martino、David Mellis 和 Nicholas Zambetti 等。

Massimo Banzi 之前是意大利伊夫雷亚 (Ivrea) 一家高科技设计学校的老师，他的学生们常抱怨找不到便宜好用的微控制器。2005 年冬天，Massimo Banzi 与 David Cuartielles 讨论了这个问题。David Cuartielles 是一名西班牙籍芯片工程师，当时在这所学校做访问学者。两人决定设计自己的电路板。由于 Massimo Banzi 喜欢去一家名叫 di Re Arduino 的酒吧，该酒吧以 1000 年前意大利国王 Arduino 的名字命名。为了纪念这个地方，他将这块电路板命名为 Arduino。

Arduino 的硬件原理图、电路图、IDE 软件及核心库文件都是开源的，在开源协议范围内可以任意修改原始设计及相应代码。Arduino 的硬件是用来连接电路的 Arduino 板。表 1-1 列出了几种常见的 Arduino 板。

表 1-1 几种常见的 Arduino 板

名称	Duemilanove	UNO R3	Nano	Mini	Leonardo	Mega2560 R3	Due
MCU	Atmega168/328	Atmega328	Atmega168/328	Atmega168/328	Atmega32u4	Atmega2560	AT91SAM3X8E
工作电压	5V	5V	5V	5V	5V	5V	3.3V
数字 I/O	14	14	4	14	20	54	54
PWM	6	6	6	6	7	15	12
模拟 I/O	6	6	8	8	12	16	12
时钟	16MHz	16MHz	16MHz	16MHz	16MHz	16MHz	84MHz
Flash	16KB/32KB	32KB	16KB/32KB	16KB/32KB	32KB	256KB	512KB
SRAM	1KB/2KB	2KB	1KB/2KB	1KB/2KB	2.5KB	8KB	96KB
EEPROM	512B/1KB	1KB	512B/1KB	512B/1KB	1KB	4KB	-
USB 芯片	FTDI FT232RL	Atmega16u2	FTDI FT232RL	-	-	Atmega16u2	-

图 1-1 给出了常见 Arduino UNO 实物图及核心控制器 Atmega328 的引脚图。

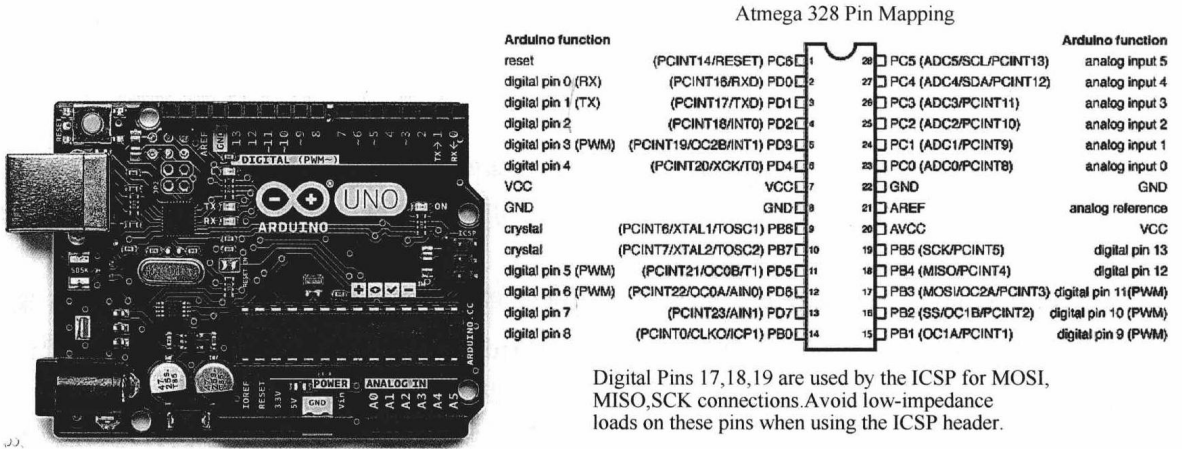


图 1-1 常见 Arduino UNO 实物图及核心控制器 Atmega328 的引脚图

以下将对 Arduino UNO 的相关资源做简要介绍。

Arduino/Genuino UNO 的基本概述如下：

- 核心微控制器为 Atmega328。
- 工作电压为 5V、输入电压为（推荐）7~12V、输入电压为（范围）6~20V。
- 14 路数字 I/O 引脚（6 路可作为 PWM 输出）。
- 6 路模拟输入通道（A0~A5 或称为 AD0~AD5）。
- I/O 引脚直流电流 40mA、3.3V 引脚直流电流 50mA。
- Flash Memory 32KB（Atmega328，其中 0.5KB 用于 bootloader）。
- SRAM 2 KB（Atmega328）、EEPROM 1 KB（Atmega328）。
- 工作时钟为 16MHz。

Arduino UNO 的 3 种供电方式：

- 外部直流电源通过电源插座供电。
- 电池连接电源连接器的 GND 和 VIN 引脚。
- USB 接口直接供电。

特殊功能引脚（或通道）如下：

- 外部中断（2、3）：INT0，INT1 中断触发引脚，可设成上升沿、下降沿或同时触发。
- 脉宽调制 PWM（3、5、6、9、10、11）：提供 6 路 8 位 PWM 输出。
- 串口 RX/TX（0、1）：提供 TTL 电平串口通信。
- SPI 接口：SPI 接口引脚为 10~13，对应 SS、MOSI、MISO、SCK。
- TWI 接口（兼容 I²C）：引脚 SDA、SCL 对应 A4、A5。
- 6 路模拟输入通道（A0~A5）：每一路均有 10 位分辨率，默认输入信号范围为 0~5V，可通过 AREF 调整输入上限。
- LED（13）：用于测试 LED 的保留接口，输出高电平时点亮，反之熄灭。

1.2 Arduino 开发环境

Arduino 提供了 IDE 开发环境，可在 Windows、Macintosh OS X、Linux 上运行，用于编写程序代码并上传（烧写）到 Arduino 板，烧写后的 Arduino 板能通过各类传感器感知环境，并通过控制 LED、电动机及其他装置来反馈、影响外部环境。下面介绍 Arduino 的三种开发环境：

1) Arduino IDE 开发环境

在 Arduino 官网 <http://www.arduino.cc> 导航栏 SOFTWARE 菜单下, 或者直接进入 <https://www.arduino.cc/en/main/software>, 均可找到当前版本的 Arduino 安装程序(当前版本为 arduino-1.8.3-windows.exe), 下载后执行该 exe 文件即可完成安装操作。所安装的 Arduino IDE 启动后的界面如图 1-2 所示。在该界面下即可完成代码编辑、生成及上传(烧写)操作。

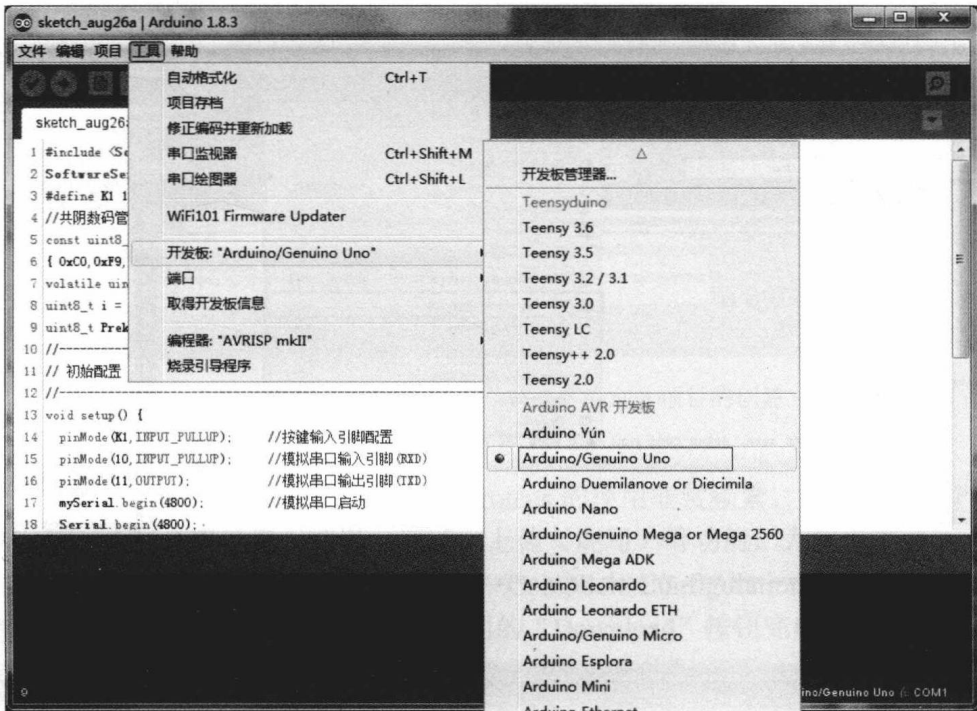


图 1-2 Arduino IDE 启动后的界面

Arduino IDE 安装便捷, 使用简单, 但与专业级开发工具相比还有较大差距, 如在项目管理、代码缩放、代码辅助输入等方面, Arduino IDE 均不能提供很好的支持。

2) Atmel Studio+Arduino 开发环境

为提供更强大、更便捷的开发体验, 本书所有案例均在 Atmel Studio+Arduino 环境下开发。搭建 Atmel Studio+Arduino 开发环境时, 除要求先安装前面提到的 Arduino 安装包 arduino-1.8.3-windows.exe 以外, 还需要完成下述两项安装:

(1) 下载安装 Atmel Studio (当前版本为 7.0)。由于 Atmel 已并入 Microchip, 其下载地址为 <https://www.microchip.com/avr-support/atmel-studio-7> (当前版本文件为 as-installer-7.0.1645-full.exe, 887MB)。安装 Atmel Studio 7 时, 第 1 步可能会出现 Windows Update 出错, 对此要根据提示补全相关补丁。图 1-3 是完成安装并启动 Atmel Studio 7 后显示的主界面。

(2) 扩展更新 (Extensions and Updates)。完成 Atmel Studio 7 的安装后, 单击 Atmel Studio 7 的工具菜单, 找到扩展更新 (Extensions and Updates...) 菜单, 打开如图 1-4 所示 Atmel Studio 7 的扩展更新 (Extensions and Updates) 界面, 找到“Arduino IDE for Atmel Studio 7”, 然后单击“Install”按钮。因为相关包需在线下载, 安装进程可能比较缓慢, 还可能提示进入相关网站完成注册等。

完成安装后, 启动 Atmel Studio 7, 单击“File→New Project”, 在图 1-5 所示窗口中选择“Arduino Sketch”创建项目, 创建成功后, 项目文件夹下会出现默认的*.ino 文件, 在该源程序文件中可看到默认的 setup/loop 函数框架, 在该框架中完成代码编写后即可编译运行。

在项目文件夹中可看到的文件或文件夹有项目解决方案文件 (*.atsln)、项目文件 (*.cppproj)、Arduino 主程序文件 (*.ino)、在 debug 模式下生成的 HEX 文件 (debug 文件夹下的 *.hex)、在 release 模式下生成的 HEX 文件 (release 文件夹下的 *.hex)。debug/release 两种模式的主要差别是前者默认带串口调试支持,在虚拟终端将输出大量提示信息,调试时明显缓慢。作为发布版的 release 模式,则不会默认带有这些支持,本书各案例捆绑的 HEX 文件均在 release 模式下生成。

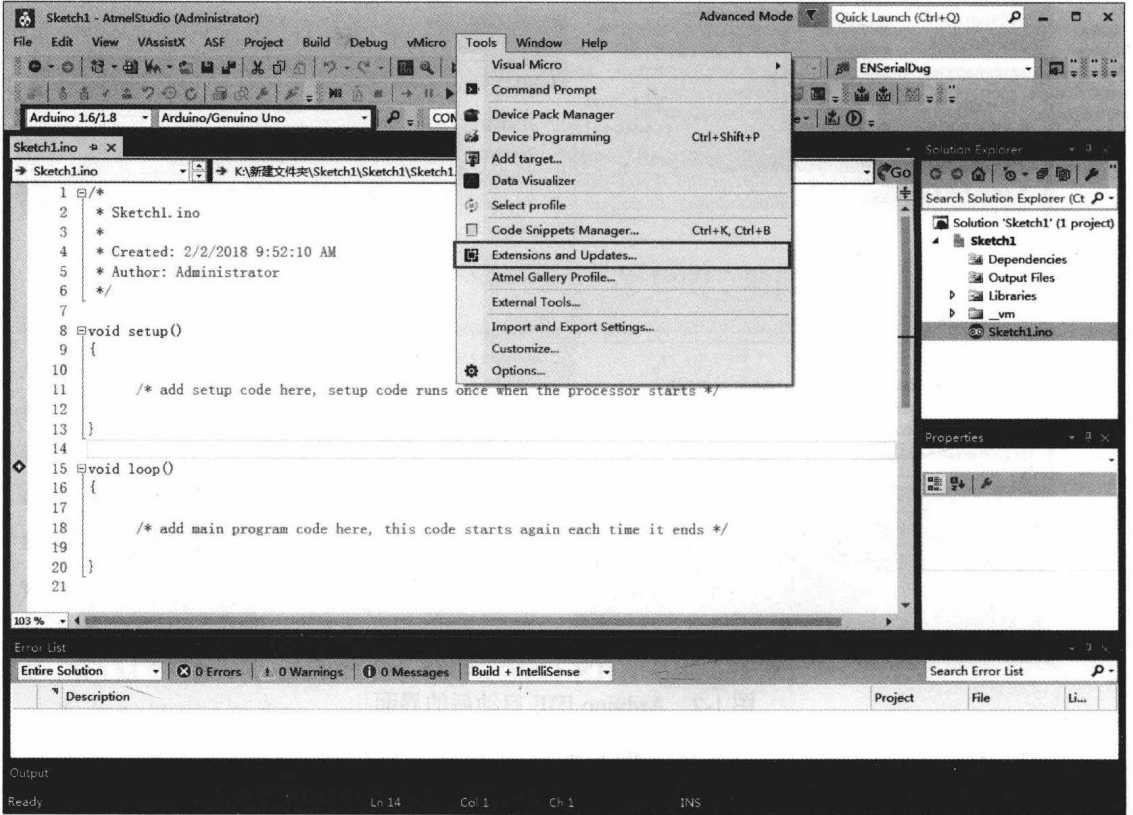


图 1-3 完成安装并启动 Atmel Studio 7 后显示的主界面

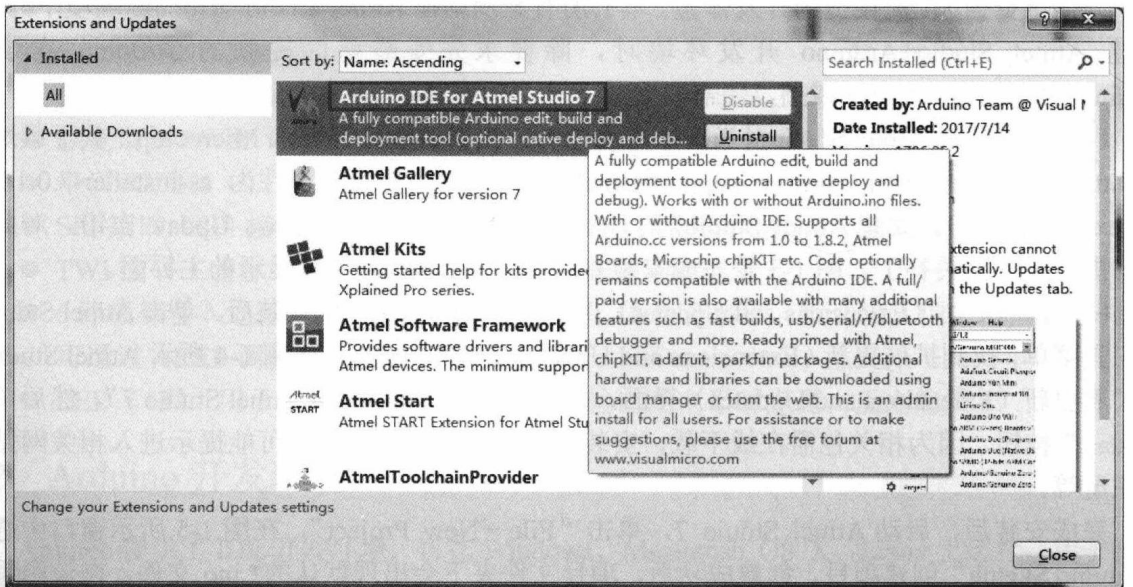


图 1-4 Atmel Studio 7 的扩展更新 (Extensions and Updates) 界面

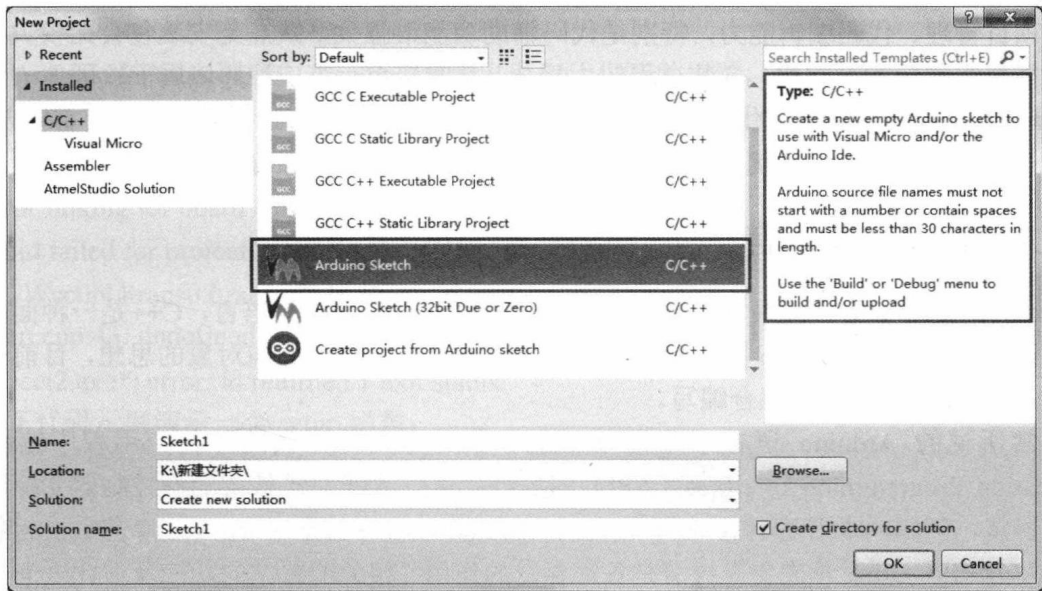


图 1-5 Atmel Studio 7 下 Arduino Sketch 项目的创建

3) Proteus 8 自带的 IDE

本书所有案例均在仿真环境下运行，对 Proteus 的使用非常频繁。当前版本的 Proteus 8 开始具备了项目管理、代码编辑、编译、调试、上传（烧写）等功能，为实现本书 Arduino 项目的编译，开发者还需要单击 Proteus 的 System→Compilers Configuration 菜单，打开如图 1-6 所示对话框，单击对话框中“Arduino AVR”后面的“Download”按钮完成编译器的下载并安装，随后即可在 Proteus 8 自带的项目开发环境下完成 Arduino 的程序开发（包括仿真电路设计、运行与调试）。

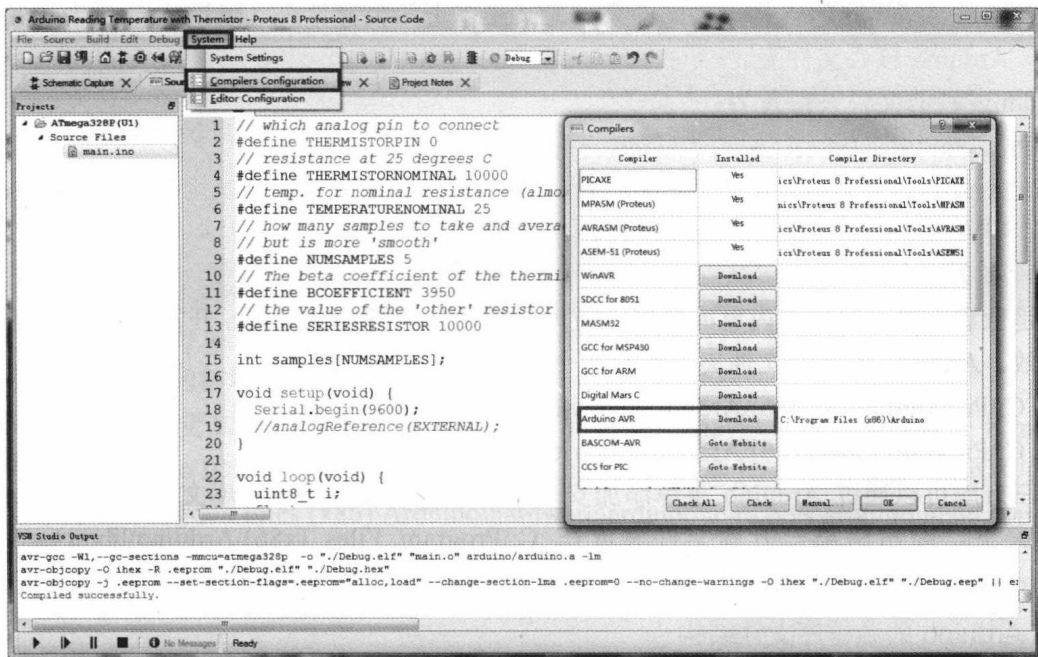


图 1-6 Proteus 8 环境下的 Arduino 项目编译器（配置）

综合来看，上述三种 Arduino 开发环境各有特点：第一种安装简便、快速，但相关辅助功能比较弱；第二种环境下的 Atmel Studio+Arduino 开发平台属于专业级开发平台，其界面非常

友好，项目管理、代码编辑能力，特别是代码辅助功能很强大，缺点是安装包会比较大，部分安装还需在在线状态下完成，安装过程较为复杂，耗时较多；第三种环境属于软硬件一体化集成开发环境，其仿真能力非常优秀，但软件开发环境支持能力与 Atmel Studio 相比还有较大差距。

鉴于第二种开发环境的优势，本书所有案例均选择在 Atmel Studio+Arduino 下开发。

1.3 Arduino 开发语言

Arduino 使用 C/C++编写程序，C 语言是一种面向过程的编程语言，C++是一种面向对象的编程语言。早期的 Arduino 核心库使用 C 语言编写，后来引入面向对象的思想，目前最新的 Arduino 核心库采用 C/C++混合编写。

通常所说的 Arduino 语言，是指 Arduino 核心库文件提供的各种应用程序编程接口（Application Programming Interface, API）的集合。这些 API 对更低层的微控制器支持库进行了二次封装，例如，本书基于 AVR 单片机的 Arduino 核心库就是对 avr-gcc 库的二次封装。在传统开发方式下，开发者需要先搞清楚各寄存器的作用、相关配置位的功能及彼此之间的关系等，然后通过读/写相关寄存器来实现开发目标。在 Arduino 环境下，这些操作将被大大简化，开发者在没有太多微控制器（单片机）硬件基础及程序设计基础的情况下，也能通过简单学习快速进入开发状态，但必要的 Arduino 基本程序结构及 C 语言基础知识、Arduino 相关库应用仍不可缺少。

1.4 Arduino 程序结构

1. Arduino 的安装程序文件结构

全书案例全部在 Atmel Studio+Arduino 开发环境下完成设计，为从更深层次认识 Arduino，开发者要熟悉 Arduino 的安装程序文件夹，尤其是以下两类文件夹（以默认 C 盘为例）：

（1）Arduino 核心文件夹。

Arduino 核心文件夹位置为：

C:\Program Files (x86)\Arduino\hardware\arduino\avr\cores\arduino

该文件夹中有后面将要讨论的 main.cpp，还有串口驱动（大量调试均要使用到的 Serial）、TWI/I²C/SPI/USB 驱动、数据流处理、打印、音频输出等。

（2）库文件夹。

正是依靠大量 Arduino 内置库及第三方库的应用，才使得 Arduino 开发变得非常便捷、稳定、高效。Arduino 内置库文件夹主要有两个，分别在以下位置：

① C:\Program Files (x86)\Arduino\hardware\arduino\avr\libraries。

② C:\Program Files (x86)\Arduino\libraries。

第三方库可以通过 Arduino 菜单添加安装，也可以直接将第三方库文件夹复制到 Arduino 库文件夹下，通常将第三方库文件夹放在 C:\Program Files (x86)\Arduino\libraries 下，如 Keyboard、TimerOne、DS1307RTC、TFT、U8g2、arduino_uip-master 等。

2. Arduino 的源程序文件结构

1) 两个主要函数：setup、loop

无论是直接在 Arduino IDE 下开发 Arduino 程序，还是在 Atmel Studio+Arduino 环境下开发 Arduino 程序，所创建的 Arduino 项目均称为 Sketch（草图、梗概），其核心源程序文件后缀均为*.ino，其中最重要的两个部分为 setup 与 loop。当然，在核心文件*.ino 的外围，还有自编

的其他 C/C++ 程序文件或第三方 C/C++ 库文件。

在 Arduino 的 *.ino 程序中，setup() 通常出现在代码开始位置，主要用于初始化相关变量，完成相关端口引脚配置等。setup() 函数仅在上电后，或者在初始化 Arduino 板之后运行一次。

如果核心程序文件 *.ino 中缺少 setup 函数，将导致编译出现以下错误：

Error linking for board Arduino/Genuino Uno

Build failed for project 'XXXXXXXXXXXX'

ccGwyc0N.ltrans0.ltrans.o*: In function main

main.cpp:43: undefined reference to setup (提示 main.cpp 中出现了未定义 setup 的引用)

collect2.exe*: error: ld returned 1 exit status

以下代码示例演示一个 setup 函数：

```
int buttonPin = 3; // 按键连接 Arduino IO3 号引脚
void setup() { // 初始配置函数（仅执行一次）
    Serial.begin(9600); // 串口波特率配置为 9600
    pinMode(buttonPin, INPUT); // 按键连接的引脚配置为输入
}
void loop() { // 主循环（暂为空）
    // ...
}
```

在 Arduino 程序中，loop() 函数通常在 setup() 之后使用，loop 处于持续循环运行状态，实际上 loop 自身并不会实现循环，其循环的实现显然是依靠后面将要解析的主框架文件 main.cpp 中的 for 循环，是它使 loop 得以被反复调用，从而实现循环。

以下代码中包括了一个 loop 函数示例：

```
int buttonPin = 3; // 按键连接 Arduino IO3 号引脚
void setup() { // 初始配置函数（仅执行一次）
    Serial.begin(9600); // 串口波特率配置为 9600
    pinMode(buttonPin, INPUT); // 按键连接的引脚配置为输入
}
void loop() { // 主循环（无限循环检查按键并控制串口输出）
    if (digitalRead(buttonPin) == HIGH) // 检测到按键输入高电平
        Serial.write('H'); // 串口输出 H
    else
        Serial.write('L'); // 否则串口输出 L
    delay(1000);
}
```

Arduino 程序中的 loop 函数将在 Arduino 板上电后持续运行，直至系统关机。

2) 对 main.cpp 的解析

为使开发者能够更加深入地认识 setup 与 loop 的支持机理，以下再来深入看一下 main.cpp 程序文件。在 Arduino IDE 安装路径下可搜索到 main.cpp，以本书所用的 Win7 调试环境为例，其文件全称为 C:\Program Files (x86)\Arduino\hardware\arduino\avr\cores\arduino\main.cpp，它是 Arduino 框架的主程序。打开该文件可看到如下主要内容：

```
/*main.cpp - Main loop for Arduino sketches
Copyright (c) 2005-2013 Arduino Team. All right reserved.
.....
*/
#include <Arduino.h>
// Declared weak in Arduino.h to allow user redefinitions.
int atexit(void (* /*func*/ )()) { return 0; }
// Weak empty variant initialization function.
```

```

// May be redefined by variant files.
void initVariant() __attribute__((weak));
void initVariant() { }
void setupUSB() __attribute__((weak));
void setupUSB() { }
int main(void) { //main.cpp 的主程序 main
    init();
    initVariant();
    #if defined(USBCON)
        USBDevice.attach();
    #endif
    setup(); //初始配置 (*.ino 的核心函数 setup)
    for (;;) {
        loop(); //主循环 (*.ino 的核心函数 loop)
        if (serialEventRun) serialEventRun(); //串口处理
    }
    return 0;
}

```

参阅 Arduino 主框架文件 main.cpp 内的上述代码，可以很清晰地看到，编译 Arduino 程序时，真正的主文件是 main.cpp，对*.ino 中的 setup 及 loop 的调用均由 main.cpp 中的 main 函数发起，显然 loop 自身并不能实现循环，其循环的实现是因为 main 内部的 for(;;){ loop();..... }。

1.5 基本数据类型、有符号数与无符号数的应用及位相关操作

开始 Arduino C 程序设计，需要熟悉 Arduino 环境下的 C 语言基本数据类型，并熟悉有符号数、无符号数的基本应用，大量位相关的基本操作，这些操作会在程序设计中大量、频繁的使用。

1. Arduino C 语言基本数据类型

Arduino C 语言基本数据类型如表 1-2 所示。

表 1-2 Arduino C 语言基本数据类型

类 型	等价定义	长度 (位)	数据范围	备 注
void		0		只用作函数声明，表示没有返回值
boolean	bool	1	true,false	每个布尔变量占一字节内存
char		8	-128~127	
unsigned char	uint8_t	8	0~255	
byte	uint8_t	8	0~255	
int	int16_t	16	-32 768~32 767	在 Arduino Due 上是 4 字节
unsigned int	uint16_t	16	0~65 535	
word	uint16_t	16	0~65 535	
long		32	-2 147 483 648~2 147 483 647	
unsigned long	uint32_t	32	0~2 ³² -1	
short		16	-32 768~32 767	
float		32	-3.4028235E+38~3.4028235E+38	只有 6~7 位小数精度
double		32		Arduino Due: 8 字节，其余等同于 float