

单片机开发系列

# AVR单片机C语言 开发入门与典型实例 (修订版)

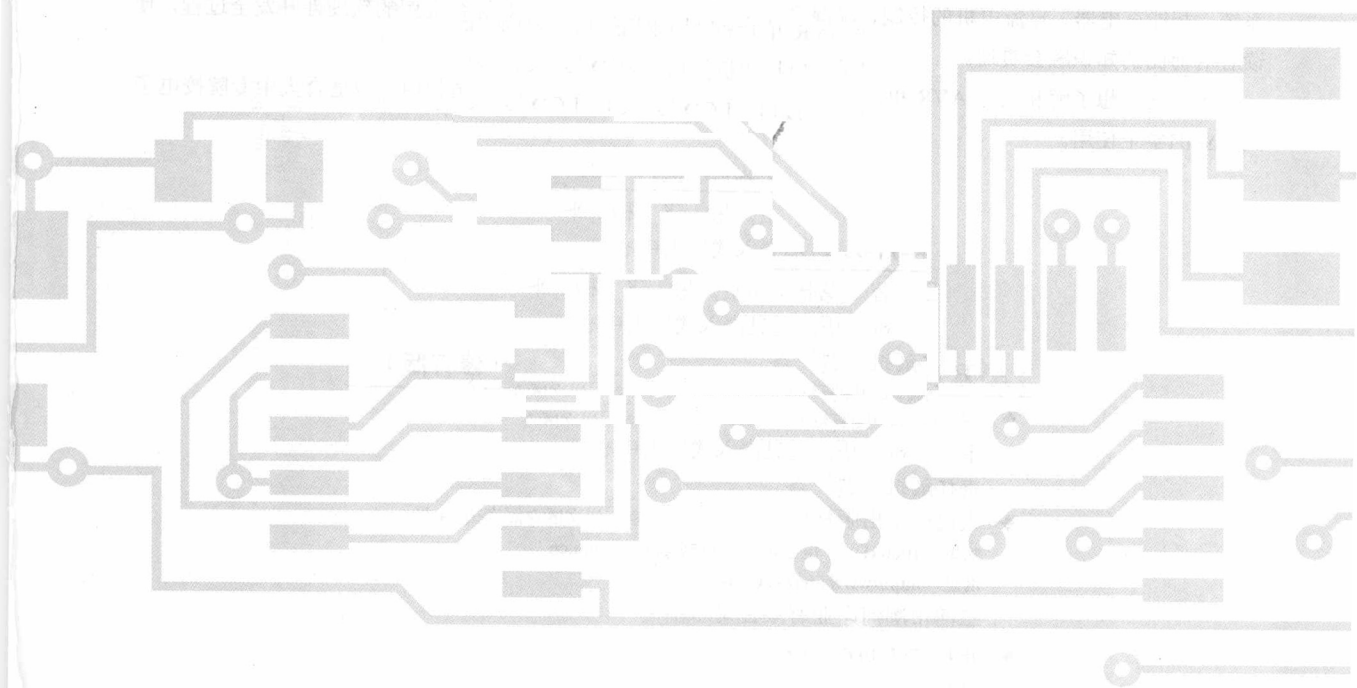
龙脉工作室 朱飞 杨平 编著  
华清远见嵌入式培训中心 审校

从AVR单片机指令和单片机C语言讲起，引导读者快速入门  
9大接口，详细讲解接口电路器件使用、硬件设计、软件设计  
6个综合实例，可直接用于项目开发

单片机开发系列

# AVR单片机C语言 开发入门与典型实例 (修订版)

龙脉工作室 朱飞 杨平 编著  
华清远见嵌入式培训中心 审校



人民邮电出版社  
北京

## 图书在版编目 (C I P) 数据

AVR单片机C语言开发入门与典型实例 / 朱飞, 杨平  
编著. — 2版 (修订本). — 北京: 人民邮电出版社,  
2010.6

ISBN 978-7-115-22738-6

I. ①A… II. ①朱… ②杨… III. ①单片微型计算机  
—C语言—程序设计 IV. ①TP368.1②TP312

中国版本图书馆CIP数据核字 (2010) 第066946号

## 内 容 提 要

本书首先详细讲解 AVR 单片机的基础知识, 包括硬件基础、指令系统、开发环境以及 AVR 单片机 C 语言基础; 然后讲解其 I/O 口打印机接口、定时器/计数器、中断、串行口、程序存储器扩展、数据存储器扩展等模块; 接着讲解其键盘接口、LCD 接口、打印机接口、A/D 接口、D/A 接口、可编程器件接口、I<sup>2</sup>C 总线接口、SPI 总线接口、CAN 总线接口的硬件设计和程序设计; 最后讲解了 MP3 播放器设计、小型打印机系统、智能充电器、直流电机的控制、温湿度传感器、手持无线遥控器等综合实例的软硬件开发全过程, 使读者对前面的知识融会贯通。

本书适合想了解和进行 AVR 单片机实际系统开发的读者和工程技术人员使用, 也适合大中专院校电子类专业的学生使用。

## AVR 单片机 C 语言开发入门与典型实例 (修订版)

- ◆ 编 著 龙脉工作室 朱 飞 杨 平
- 审 校 华清远见嵌入式培训中心
- 责任编辑 黄 焱
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号  
邮编 100061 电子函件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
三河市潮河印业有限公司印刷
- ◆ 开本: 787×1092 1/16  
印张: 27  
字数: 655 千字 2010 年 6 月第 2 版  
印数: 4 301 - 8 300 册 2010 年 6 月河北第 1 次印刷

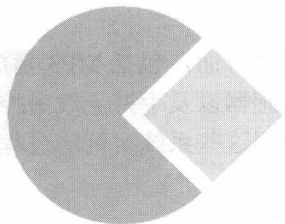
ISBN 978-7-115-22738-6

定价: 49.80 元

读者服务热线: (010) 67132692 印装质量热线: (010) 67129223

反盗版热线: (010) 67171154

# 前 言



## 写作背景

1997年,美国ATMEL公司挪威设计中心的A先生和V先生出于市场需求的考虑,将其先进的Flash技术与8051单片机结合起来,从而推出了8位全新配置的精简指令集(Reduced Instruction Set)的单片机,即AVR单片机。与传统的8051单片机相比,AVR单片机在效率、速度及指令格式上具有明显优势,更适合在嵌入式系统中应用。

近十年来,随着AVR单片机的不断改进和新产品的持续推出,AVR单片机已形成系列产品,其中ATtiny、AT90和ATmega分别对应中高档产品,在国内外已得到广泛应用。

AVR单片机的学习与一般的编程相比困难的地方在于:要完成一个AVR单片机系统,不仅需要全面了解AVR单片机的基本原理,更为重要的是如何针对实际应用的需要设计合适的接口电路。因此,AVR单片机系统开发是一个软硬件结合的综合性工程。

对于初接触AVR单片机的初学者而言,要他去设计一个完整的实际应用电路实在是一件非常困难的事,更谈不上如何对不同方案和器件的选择。基于此,本书以单片机ATmega128(L)为AVR单片机的代表,针对AVR单片机的初学者,在介绍AVR单片机的基本原理和软硬件开发所需的基础知识后,把重点放在典型模块和接口电路的设计上,最后还给出AVR单片机开发的综合实例,从而使读者循序渐进地掌握AVR单片机开发所需的必备知识。

## 本书主要内容

本书共分为4篇。

第1篇为AVR单片机基础篇包括第1章~第6章。

本篇详细介绍了AVR单片机的发展、硬件结构、指令系统以及软硬件开发工具,并重点介绍了AVR单片机的C语言开发。编者建议读者使用ATMEL公司提供的AVR Studio及WinAVR(提供C语言开发所需的GCC编译器),因为这两个软件开发环境是完全免费的,并在不断更新中。本书中介绍的AVR Studio版本为4.13SP2,WinAVR的版本为20071221。

第2篇为AVR单片机典型模块C语言应用实例篇,包括第7章~第12章。

本篇详细讲解了AVR单片机的I/O口、定时器/计数器、中断、串行口及存储器扩展的基本应用实例,并给出相关的经验总结,帮助读者了解AVR单片机开发的基础。

第 3 篇为 AVR 单片机接口典型应用篇，包括第 13 章～第 21 章。

本篇给出 AVR 单片机开发中的高级实例，主要包括键盘接口、LCD 接口、打印机接口、A/D 接口、D/A 接口、可编程器件接口、I<sup>2</sup>C 总线、SPI 总线以及 CAN 总线，同时也给出相关的经验总结以帮助读者开展自己的工作。

第 4 篇的综合系统实例篇，包括第 22 章～第 27 章。

本篇给出一些 AVR 单片机系统开发的实例。这些实例涉及很多方面，包括 MP3 播放器设计、小型打印机系统、智能充电器、直流电机的控制、温湿度传感器以及手持无线遥控器，为读者解析了一些常见的电子产品和工业或控制产品的原理。在介绍这些系统实例的过程中，强调了 AVR 单片机系统开发中软硬件的模块化设计思想。

## 读者对象

本书适合想了解和进行 AVR 单片机实际系统开发的读者和工程技术人员使用，也适合大中专院校电子类专业的学生使用。

## 参与编写的人员

本书由中国科学院自动化研究所的朱飞博士、杨平博士、董翔博士和张正涛博士编写，并由朱飞博士统稿。另外参加编写的还有刘艳伟、严雨、刘燕祎、周晶、周丰、梅乐夫、房明浩、王亮、门店宏、吴洋、石峰、张圣亮、邱文勋、刘鲲、矫津毅、林远长、董前程、岂兴明、汤嘉立、刘变红、周建兴、张高煜、赵红波、邓志宝、刘坤、刘明辉、李鹏、白学明、步十建等，在此一并致以诚挚的谢意。

由于作者水平有限，书中错误和疏漏之处在所难免，恳请广大读者批评指正。本书责任编辑的联系方式是 [huangyan@ptpress.com.cn](mailto:huangyan@ptpress.com.cn)，欢迎大家来信交流。

## 相关学习资料

单片机学习资料：

<http://www.coronabook.com>

嵌入式学习资料：

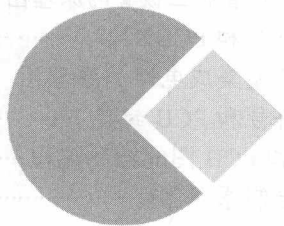
<http://download.farsight.com.cn/download>

Linux 学习次料：

<http://www.linuxdriver.cn>

编者

2010 年 5 月



# 目 录

## 第 1 篇 AVR 单片机基础

### 第 1 章 AVR 单片机概述 ..... 3

#### 1.1 AVR 单片机及其发展 ..... 3

##### 1.1.1 AVR 单片机简介 ..... 3

##### 1.1.2 AVR 单片机的特点 ..... 4

##### 1.1.3 AVR 单片机的分类 ..... 5

#### 1.2 ATmega128 (L) 单片机 ..... 5

##### 1.2.1 ATmega128 (L) 结构和 主要特点 ..... 5

##### 1.2.2 ATmega128 (L) 的主要 性能 ..... 7

##### 1.2.3 ATmega128 (L) 的封装与 引脚 ..... 8

#### 1.3 本章小结 ..... 12

### 第 2 章 ATmega128 (L) 单片机硬件 结构 ..... 13

#### 2.1 ATmega128 (L) 的内核 ..... 13

#### 2.2 ATmega128 (L) 的存储器 ..... 14

##### 2.2.1 ATmega128 (L) 的内部 存储器 ..... 14

##### 2.2.2 ATmega128 (L) 的外部 存储器 ..... 15

#### 2.3 ATmega128 (L) 的系统

#### 时钟及电源管理 ..... 15

##### 2.3.1 系统时钟 ..... 15

##### 2.3.2 电源管理与休眠模式 ..... 18

#### 2.4 ATmega128 (L) 的复位及中断 ..... 19

##### 2.4.1 ATmega128 (L) 的复位 逻辑 ..... 19

##### 2.4.2 ATmega128 (L) 的中断 及中断响应 ..... 20

#### 2.5 ATmega128 (L) 的定时器/ 计数器 ..... 22

##### 2.5.1 8 位定时器/计数器 T/C0 ..... 22

##### 2.5.2 8 位定时器/计数器 T/C2 ..... 26

##### 2.5.3 16 位定时器/计数器 T/C1 和 T/C3 ..... 26

#### 2.6 ATmega128 (L) 的总线接口 ..... 31

##### 2.6.1 同步外设接口 SPI ..... 31

##### 2.6.2 通用同步/异步串行接口 USART0 和 USART1 ..... 32

##### 2.6.3 两线串行 TWI 总线接口 ..... 35

#### 2.7 本章小结 ..... 37

### 第 3 章 ATmega128 (L) 的指令系统 ..... 38

#### 3.1 ATmega128 (L) 的指令系统 概述 ..... 38

3.2	ATmega128 (L) 的指令操作数的寻址方式	43	4.1.2	电阻的原理图符号和 PCB 封装	79
3.3	算术和逻辑运算指令	45	4.1.3	电容的原理图符号和 PCB 封装	80
3.3.1	加法指令	45	4.1.4	二极管和三极管的原理图符号和 PCB 封装	81
3.3.2	减法指令	46	4.1.5	典型集成电路的原理图符号和 PCB 封装	83
3.3.3	取反码和补码指令	48	4.2	原理图和 PCB 图的绘制流程	84
3.3.4	乘法指令	48	4.2.1	绘制原理图	84
3.3.5	逻辑与指令	50	4.2.2	绘制 PCB 图	86
3.3.6	逻辑或指令	51	4.3	地线	90
3.3.7	逻辑异或指令	52	4.3.1	地线的概念及分类	90
3.4	比较和转移指令	53	4.3.2	PCB 设计中地线的处理	91
3.4.1	比较指令	53	4.4	本章小结	91
3.4.2	无条件转移指令	54	<b>第 5 章 AVR 单片机软件开发环境</b>		92
3.4.3	条件转移指令	54	5.1	软件开发环境	92
3.4.4	子程序调用与返回指令	60	5.2	AVR Studio 4.13 SP2 开发环境	94
3.5	数据传输指令	61	5.2.1	AVR Studio 4.13 SP2 开发环境获取与安装	94
3.5.1	直接数据传输指令	62	5.2.2	AVR Studio 4.13 SP2 的菜单	95
3.5.2	间接数据传输指令	63	5.2.3	使用 AVR Studio 4.13 SP2	101
3.5.3	与程序存储器有关的数据传输指令	65	5.3	本章小结	103
3.5.4	I/O 口数据传输指令	67	<b>第 6 章 AVR 单片机 C 语言开发</b>		104
3.5.5	堆栈操作指令	68	6.1	C 语言基础知识	104
3.6	位操作指令	68	6.1.1	标识符和关键字	104
3.6.1	带进位的位操作指令	68	6.1.2	基本数据类型、常量和变量	104
3.6.2	位变量传输指令	70	6.1.3	常见运算符	106
3.6.3	位修改指令	70	6.1.4	程序中的语句种类	107
3.7	MCU 控制指令	74	6.2	C 语言高级编程	110
3.8	ATmega128 (L) 的汇编语言	75	6.2.1	数组	110
3.8.1	汇编语言语句格式及伪指令	75	6.2.2	指针	112
3.8.2	表达式	76	6.2.3	结构、共用体和枚举	115
3.9	本章小结	77	6.2.4	函数	119
<b>第 4 章 AVR 单片机硬件电路设计工具</b>		78			
4.1	Protel 99 SE 中典型元器件的原理图符号和 PCB 封装绘制	78			
4.1.1	元器件封装的概念	78			



6.2.5 预处理	122	6.3.2 AVR 单片机存储器中数据的访问	126
6.3 AVR Studio 4.13 SP2 对 ANSI C 的扩展	125	6.3.3 AVR 单片机的中断服务程序	130
6.3.1 I/O 寄存器和端口操作	125	6.4 本章小结	132

## 第 2 篇 AVR 单片机典型模块 C 语言应用实例

<b>第 7 章 I/O 的应用</b>	135	8.3 利用定时器/计数器实现按键的复用	148
7.1 I/O 端口概述	135	8.3.1 实例功能	148
7.2 I/O 口控制发光二极管	136	8.3.2 硬件设计	148
7.2.1 实例功能	136	8.3.3 程序设计及详解	149
7.2.2 典型器件介绍	136	8.3.4 经验总结	150
7.2.3 硬件设计	136	8.4 本章小结	150
7.2.4 程序设计及详解	137	<b>第 9 章 中断的应用</b>	151
7.2.5 经验总结	138	9.1 中断概述	151
7.3 I/O 口控制八段数码管	138	9.2 利用定时器中断报警	151
7.3.1 实例功能	138	9.2.1 实例功能	151
7.3.2 典型器件介绍	138	9.2.2 典型器件介绍	151
7.3.3 硬件设计	138	9.2.3 硬件设计	152
7.3.4 程序设计及详解	139	9.2.4 程序设计及详解	152
7.3.5 经验总结	140	9.2.5 经验总结	153
7.4 I/O 模拟 1-Wire 总线控制		9.3 利用中断实现对按键的响应	153
DS18B20	140	9.3.1 实例功能	153
7.4.1 实例功能	140	9.3.2 典型器件介绍	153
7.4.2 典型器件介绍	140	9.3.3 硬件设计	154
7.4.3 硬件设计	141	9.3.4 程序设计及详解	154
7.4.4 程序设计及详解	142	9.3.5 经验总结	155
7.4.5 经验总结	145	9.4 本章小结	155
7.5 本章小结	145	<b>第 10 章 串行口的应用</b>	156
<b>第 8 章 定时器/计数器应用</b>	146	10.1 串行通信概述	156
8.1 定时器/计数器概述	146	10.2 RS-232 总线接口	157
8.2 利用定时器实现方波输出	146	10.2.1 实例功能	157
8.2.1 实例功能	146	10.2.2 典型器件介绍	157
8.2.2 硬件设计	146	10.2.3 硬件设计	158
8.2.3 程序设计及其详解	147	10.2.4 程序设计及详解	159
8.2.4 经验总结	148		



10.2.5 经验总结 .....	160	11.3.3 硬件设计 .....	175
10.3 RS-485 总线接口 .....	160	11.3.4 程序设计及详解 .....	175
10.3.1 实例功能 .....	160	11.3.5 经验总结 .....	177
10.3.2 典型器件介绍 .....	160	11.4 扩展 Flash 存储器 .....	177
10.3.3 硬件设计 .....	161	11.4.1 实例功能 .....	177
10.3.4 程序设计及详解 .....	162	11.4.2 典型器件介绍 .....	177
10.3.5 经验总结 .....	164	11.4.3 硬件设计 .....	179
10.4 单片机之间的通信 .....	164	11.4.4 程序设计及详解 .....	179
10.4.1 实例功能 .....	164	11.4.5 经验总结 .....	181
10.4.2 典型器件介绍 .....	164	11.5 本章小结 .....	181
10.4.3 硬件设计 .....	164	<b>第 12 章 数据存储器扩展</b> .....	182
10.4.4 程序设计及详解 .....	164	12.1 数据存储器概述 .....	182
10.4.5 经验总结 .....	167	12.2 扩展 SRAM .....	182
10.5 本章小结 .....	167	12.2.1 实例功能 .....	182
<b>第 11 章 程序存储器扩展</b> .....	168	12.2.2 典型器件介绍 .....	183
11.1 程序存储器概述 .....	168	12.2.3 硬件设计 .....	184
11.2 扩展 EPROM .....	169	12.2.4 程序设计及详解 .....	185
11.2.1 实例功能 .....	169	12.2.5 经验总结 .....	186
11.2.2 典型器件介绍 .....	170	12.3 扩展 NVSRAM .....	186
11.2.3 硬件设计 .....	171	12.3.1 实例功能 .....	186
11.2.4 程序设计及详解 .....	171	12.3.2 典型器件介绍 .....	187
11.2.5 经验总结 .....	172	12.3.3 硬件设计 .....	188
11.3 扩展 E <sup>2</sup> PROM .....	173	12.3.4 程序设计及详解 .....	189
11.3.1 实例功能 .....	173	12.3.5 经验总结 .....	190
11.3.2 典型器件介绍 .....	173	12.4 本章小结 .....	190

## 第 3 篇 AVR 单片机接口典型应用

<b>第 13 章 键盘接口</b> .....	193	13.3.2 硬件设计 .....	196
13.1 键盘概述 .....	193	13.3.3 程序设计及详解 .....	196
13.2 外接 4 个键盘 .....	193	13.3.4 经验总结 .....	202
13.2.1 实例功能 .....	193	13.4 本章小结 .....	202
13.2.2 硬件设计 .....	193	<b>第 14 章 LCD 接口</b> .....	203
13.2.3 程序设计及详解 .....	194	14.1 LCD 概述 .....	203
13.2.4 经验总结 .....	195	14.2 外接字符点阵 LCD .....	205
13.3 与 4 × 4 键盘的接口设计 .....	195	14.2.1 实例功能 .....	205
13.3.1 实例功能 .....	196	14.2.2 典型器件介绍 .....	205

14.2.3	硬件设计	207	16.3.4	程序设计及详解	239
14.2.4	程序设计及详解	208	16.3.5	经验总结	242
14.2.5	经验总结	211	16.4	外接 $\Sigma$ - $\Delta$ 型 A/D	242
14.3	外接图像点阵 LCD	211	16.4.1	实例功能	242
14.3.1	实例功能	211	16.4.2	典型器件介绍	242
14.3.2	典型器件介绍	211	16.4.3	硬件设计	246
14.3.3	硬件设计	216	16.4.4	程序设计及详解	246
14.3.4	程序设计及详解	216	16.4.5	经验总结	249
14.3.5	经验总结	219	16.5	本章小结	249
14.4	本章小结	219	<b>第 17 章 D/A 接口实例</b>	<b>250</b>	
<b>第 15 章 打印机接口</b>	<b>220</b>		17.1	D/A 转换器概述	250
15.1	微型打印机概述	220	17.2	外接倒 T 型 D/A	251
15.2	外接串行打印机	221	17.2.1	实例功能	251
15.2.1	实例功能	221	17.2.2	典型器件介绍	251
15.2.2	典型器件介绍	221	17.2.3	硬件设计	254
15.2.3	硬件设计	223	17.2.4	程序设计及详解	255
15.2.4	程序设计及详解	224	17.2.5	经验总结	256
15.2.5	经验总结	225	17.3	外接权电流型 D/A	256
15.3	外接并行打印机	225	17.3.1	实例功能	256
15.3.1	实例功能	225	17.3.2	典型器件介绍	257
15.3.2	典型器件介绍	225	17.3.3	硬件设计	259
15.3.3	硬件设计	226	17.3.4	程序设计及详解	259
15.3.4	程序设计及详解	227	17.3.5	经验总结	261
15.3.5	经验总结	228	17.4	本章小结	262
15.4	本章小结	228	<b>第 18 章 可编程器件接口</b>	<b>263</b>	
<b>第 16 章 A/D 接口</b>	<b>229</b>		18.1	可编程器件概述	263
16.1	A/D 转换器概述	229	18.2	可编程并行接口芯片接口	263
16.2	外接逐次比较型 A/D	231	18.2.1	实例功能	263
16.2.1	实例功能	231	18.2.2	典型器件介绍	264
16.2.2	典型器件介绍	231	18.2.3	硬件设计	266
16.2.3	硬件设计	233	18.2.4	程序设计及详解	267
16.2.4	程序设计及详解	233	18.2.5	经验总结	269
16.2.5	经验总结	235	18.3	可编程中断控制器接口	270
16.3	外接半闪烁型高速 A/D	235	18.3.1	实例功能	270
16.3.1	实例功能	235	18.3.2	典型器件介绍	270
16.3.2	典型器件介绍	236	18.3.3	硬件设计	273
16.3.3	硬件设计	239	18.3.4	程序设计及详解	273

18.3.5	经验总结	276	20.2.2	典型器件介绍	299
18.4	可编程计数器/定时器接口	276	20.2.3	硬件设计	301
18.4.1	实例功能	276	20.2.4	程序设计及详解	301
18.4.2	典型器件介绍	276	20.2.5	经验总结	307
18.4.3	硬件设计	278	20.3	利用 SPI 总线实现单片机 之间的通信	307
18.4.4	程序设计及详解	279	20.3.1	实例功能	307
18.4.5	经验总结	281	20.3.2	典型器件介绍	307
18.5	本章小结	281	20.3.3	硬件设计	307
<b>第 19 章</b>	<b>I<sup>2</sup>C 总线接口</b>	<b>282</b>	20.3.4	程序设计及详解	308
19.1	I <sup>2</sup> C 总线接口概述	282	20.3.5	经验总结	309
19.2	与 AT24C64 接口	283	20.4	本章小结	309
19.2.1	实例功能	283	<b>第 21 章</b>	<b>CAN 总线接口</b>	<b>310</b>
19.2.2	典型器件介绍	283	21.1	CAN 总线接口概述	310
19.2.3	硬件设计	286	21.2	利用 SIA1000 实现数据 传输	311
19.2.4	程序设计及详解	286	21.2.1	实例功能	311
19.2.5	经验总结	288	21.2.2	典型器件介绍	311
19.3	与温湿度传感器 SHT75 接口	289	21.2.3	硬件设计	315
19.3.1	实例功能	289	21.2.4	程序设计及详解	316
19.3.2	典型器件介绍	289	21.2.5	经验总结	320
19.3.3	硬件设计	292	21.3	利用 CAN 总线实现单片 机间的通信	320
19.3.4	程序设计及详解	292	21.3.1	实例功能	320
19.3.5	经验总结	297	21.3.2	典型器件介绍	320
19.4	本章小结	297	21.3.3	硬件设计	320
<b>第 20 章</b>	<b>SPI 总线接口</b>	<b>298</b>	21.3.4	程序设计及详解	320
20.1	SPI 总线接口概述	298	21.3.5	经验总结	325
20.2	SPI 接口的 Flash 的访问	299	21.4	本章小结	325
20.2.1	实例功能	299			

## 第 4 篇 综合系统实例

<b>第 22 章</b>	<b>MP3 播放器设计</b>	<b>329</b>	22.1.4	硬件电路	333
22.1	硬件电路设计	329	22.2	软件设计	337
22.1.1	实例背景	329	22.2.1	软件设计流程	337
22.1.2	硬件设计流程	330	22.2.2	软件模块分析	337
22.1.3	典型器件选型及介绍	330	22.2.3	程序设计及详解	337
			22.3	经验总结	341

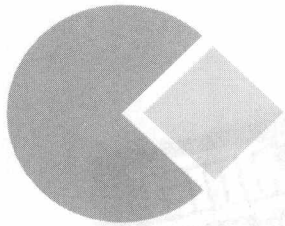
22.4 本章小结 .....	342	25.1.4 硬件电路 .....	373
<b>第 23 章 小型打印机系统</b> .....	<b>343</b>	25.2 软件设计 .....	376
23.1 硬件电路设计 .....	343	25.2.1 软件设计流程 .....	376
23.1.1 实例背景 .....	343	25.2.2 软件模块分析 .....	377
23.1.2 硬件设计流程 .....	344	25.2.3 程序设计及详解 .....	377
23.1.3 典型器件选型及介绍 .....	347	25.3 经验总结 .....	383
23.1.4 硬件电路 .....	347	25.4 本章小结 .....	383
23.2 软件设计 .....	350	<b>第 26 章 温湿度传感器</b> .....	<b>384</b>
23.2.1 软件设计流程 .....	350	26.1 硬件电路设计 .....	384
23.2.2 软件模块分析 .....	351	26.1.1 实例背景 .....	384
23.2.3 程序设计及详解 .....	351	26.1.2 硬件设计流程 .....	385
23.3 经验总结 .....	354	26.1.3 典型器件选型及介绍 .....	385
23.4 本章小结 .....	355	26.1.4 硬件电路 .....	387
<b>第 24 章 智能充电器</b> .....	<b>356</b>	26.2 软件设计 .....	390
24.1 硬件电路设计 .....	356	26.2.1 软件设计流程 .....	390
24.1.1 实例背景 .....	356	26.2.2 软件模块分析 .....	390
24.1.2 硬件设计流程 .....	357	26.2.3 程序设计及详解 .....	390
24.1.3 典型器件选型及介绍 .....	357	26.3 经验总结 .....	399
24.1.4 硬件电路 .....	360	26.4 本章小结 .....	399
24.2 软件设计 .....	362	<b>第 27 章 手持无线遥控器</b> .....	<b>400</b>
24.2.1 软件设计流程 .....	362	27.1 硬件电路设计 .....	400
24.2.2 软件模块分析 .....	363	27.1.1 实例背景 .....	400
24.2.3 程序设计及详解 .....	363	27.1.2 硬件设计流程 .....	401
24.3 经验总结 .....	368	27.1.3 典型器件选型及介绍 .....	402
24.4 本章小结 .....	368	27.1.4 硬件电路 .....	405
<b>第 25 章 直流电机的控制</b> .....	<b>369</b>	27.2 软件设计 .....	408
25.1 硬件电路设计 .....	369	27.2.1 软件设计流程 .....	408
25.1.1 实例背景 .....	369	27.2.2 软件模块分析 .....	408
25.1.2 硬件设计流程 .....	369	27.2.3 程序设计及详解 .....	408
25.1.3 典型器件选型及介绍 .....	370	27.3 经验总结 .....	418
		27.4 本章小结 .....	418



# 第 1 篇 AVR 单片机基础

- 第 1 章 AVR 单片机概述
- 第 2 章 ATmega128 (L) 单片机硬件结构
- 第 3 章 ATmega128 (L) 的指令系统
- 第 4 章 AVR 单片机硬件电路设计工具
- 第 5 章 AVR 单片机软件开发环境
- 第 6 章 AVR 单片机 C 语言开发





# 第 1 章 AVR 单片机概述

本章主要介绍 AVR 单片机的发展历史及其主要应用，并重点介绍了 ATmega128 (L) 单片机，分析其结构、主要特点、性能封装和引脚定义。本章主要内容包括以下两个方面。

- AVR 单片机及其发展；
- ATmega128 (L) 单片机简介。

## 1.1 AVR 单片机及其发展

1983 年 Intel 公司推出 8051 单片机。随着技术的发展和进步，以及市场对产品功能和性能要求的不断提高，早先的 8051 单片机越来越不能适应中高档产品的应用开发要求。于是许多公司对早先的 8051 单片机进行了改进，不断推出新的产品。

### 1.1.1 AVR 单片机简介

1997 年，美国 ATMEL 公司将其先进的 Flash 技术与 8051 单片机结合起来，从而推出了 8 位 AVR 单片机（本书中简称为 AVR 单片机），如图 1.1 所示。与传统的采用复杂指令系统（CISC，Complex Instruction Set Computer）的 8051 单片机体系不同的是，AVR 单片机采用更适合中高档电子产品和嵌入式系统应用需要的精简指令系统 RISC（Reduced Instruction Set Computer）。

精简指令系统（RISC）是 20 世纪 90 年代开发出来的一种综合了半导体集成技术并可提高软件性能的新指令结构，是为了提高 CPU 运行速度而设计的指令系统。它的关键技术在于采用流水线（pipelining）和等长指令体系结构，从而实现在一个时钟周期里完成一条或多条指令。同时，RISC 指令系统还采用了通用快速寄存器组的结构，大量使用寄存器之间的操作，减轻了 CPU 中的处理器、控制器和其他功能单元的负担。

AVR 单片机就是通过使用 RISC 指令系统来简化单片机 CPU 的指令功能，使指令的平均执行时间缩短，从而提高 CPU 的性能和速度。在使用相同的晶片技术和时钟的情况下，使用 RISC 指令系统的 CPU 的运行速度是使用 CISC 指令系统 CPU 的 2~4 倍。正因为如此，RISC 指令系统在高端系统中得到了广泛应用，例如，ATMEL 公司的 ARM 产品就是采用 RISC 指令系统，如图 1.2 所示。

此外，AVR 单片机采用了低功率和非挥发的 CMOS 制造工艺，内部分别集成了 Flash、



E<sup>2</sup>PROM (Electrically-Erasable Programmable Read-Only Memory) 和 SRAM (Static Random Access Memory) 3 种不同性能和用途的存储器。除了可以通过 SPI 接口和通用编程器对 AVR 单片机的 Flash 程序存储器和 E<sup>2</sup>PROM 数据存储寄存器进行编程外, 大多数的 AVR 单片机还支持 ISP (In-System Programming) 在线编程和 IAP (In-Application Programming) 在应用编程。

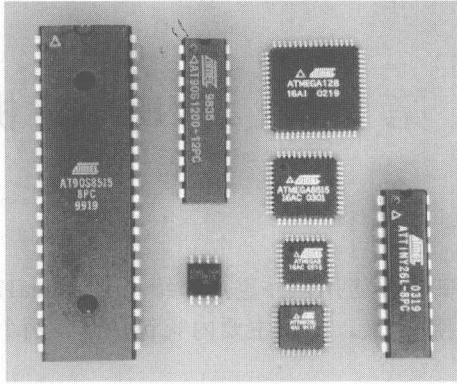


图 1.1 ATMEL 公司 8 位 AVR 单片机

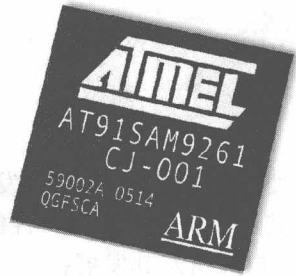


图 1.2 ATMEL 公司的 ARM 系列产品

### 1.1.2 AVR 单片机的特点

自从 ATMEL 公司推出 AVR 单片机以来, 产品受到越来越多单片机开发者的关注, 主要是因为 AVR 单片机具有以下特点。

- 采用 RISC 指令系统 CPU, 32 个通用工作寄存器, 从而避免了传统的 8051 单片机 CISC 指令系统存在的指令长度不等长、指令数多、CPU 利用率低和执行速度慢等缺点。

- 采用哈佛 (Harvard) 结构的流水线技术。在执行一条指令的同时, CPU 就已经取得下一条指令, 绝大多数指令为单周期指令, 而且指令执行速度可以达到 20MHz。

- 除了支持通过串行外设接口 SPI 和常见编程器对单片机的 Flash 程序存储器和数据存储器 E<sup>2</sup>PROM 进行编程外, 绝大部分 AVR 单片机还支持程序的 ISP 在线编程, ATmega 系列单片机还支持 IAP 在应用编程。

- 片内集成了可擦写 1000 次的 Flash 程序存储器, 同时还有大容量的可以擦写 100000 次的 E<sup>2</sup>PROM。这一点给使用者的开发、生产和维护带来方便。

- 集成了丰富的外设, 包括 I<sup>2</sup>C、SPI、E<sup>2</sup>PROM、RTC、看门狗定时器、ADC、PWM 和片内振荡器等, 大大方便了产品开发。

- 具有较宽的工作电压范围, 工作电压在 1.8~6V 之间, 电源的抗干扰能力强。

- 具有 6 种休眠功能, 功耗低, 能够从低功耗模式迅速唤醒, 并且采用可编程的频率选择和 CMOS 集成工艺。

- 不再需要把芯片从系统上拆下来再使用专门的编程器来烧写程序, 而是通过在线编程将编译好的目标程序直接烧写入 Flash 程序存储器。这极大地方便了程序的调试, 方便系统的软件升级, 尤其对于表面贴 SMD 封装, 从而更利于系统的微型化。

- C 语言源程序的效率高, 具有 32 个通用工作寄存器, 可以线性寻址。

- I/O 口驱动能力强, 可以直接驱动 LED。

- 内置看门狗定时器, 可以防止程序跑飞, 提高产品的可靠性。

- 同 PIC 单片机一样, AVR 单片机可以重新设置启动复位。内置电源上电复位 (POR) 和电源掉电检测 (BOD) 大大提高了单片机的可靠性, 不用外加复位延时电路。

- 具有 34 个中断源及相应的中断向量, 不同的中断向量的入口地址不同, 可以快速响应中断请求。

- 与传统的 8051 单片机的 USART 占用片内定时器不同, AVR 的 USART 不占用定时器, 而是使用独特的波特率发生器, 有 SPI 传输功能, 从而数据可以高速传输。因此, 晶振就可以工作在一般标准整数频率, 而 USART 的波特率可以达到 576Kbit/s。

- 具有 8 位和 16 位定时器/计数器, 可以用作比较器、计数器、外部中断和 PWM。

- Flash 存储器具有保密死锁功能, 保密性强, 并且 Flash 程序存储器在芯片内部, 可以通过自编程 (Self Programming) 方式远程下载已加密的更新代码。

- 具有多种封装形式, 可以满足不同系统的开发需要。

- ATMEL 公司提供完全免费的开发环境 AVR Studio, 支持汇编和高级语言源程序调试的模拟和仿真环境。

### 1.1.3 AVR 单片机的分类

AVR 单片机可以广泛应用于计算机外部设备、工业实时控制、仪器仪表通信设备和家用电器等各个领域。从 AVR 单片机的应用场合来看, AVR 单片机可以分为以下 9 大类:

- Automotive AVR;
- AVR Z-Link;
- CAN AVR;
- LCD AVR;
- Lighting AVR;
- megaAVR;
- Smart Battery AVR;
- tinyAVR;
- USB AVR。

## 1.2 ATmega128 (L) 单片机

ATmega128 (L) 单片机是内置 128KB Flash 程序存储器的在线可编程 8 位微控制器, 是目前 AVR 单片机中功能最强的单片机。本书的所有内容也都是围绕 ATmega128 (L) 单片机来展开, 读者只要掌握了 ATmega128 (L) 单片机的开发应用, 就很容易掌握其他 AVR 系列的单片机了。

### 1.2.1 ATmega128 (L) 结构和主要特点

ATmega128 (L) 是基于 AVR RISC 结构的 8 位低功耗 CMOS 微处理器。由于其先进的指令集以及单周期指令执行时间, ATmega128 (L) 的数据吞吐率高达 1MIPS/MHz, 从而可以缓减系统在功耗和处理速度之间的矛盾, 其结构框图如图 1.3 所示, 其主要特点如下。

- CPU 内核具有丰富的指令集和 32 个通用工作寄存器。所有的寄存器都直接与算术逻辑单元 (ALU) 相连接, 使得一条指令可以在一个时钟周期内同时访问两个独立的寄存器。