



AVR 单片机 高级语言 BASCOM 程序设计与应用

主编 张茂青

李彦超 张 敏 耿德根 编著

BASCOM

AVR BASCOM

AVR BASCOM

• • • • • • • • • • AVR •

BASCOM

AVR



北京航空航天大学出版社

AVR 单片机高级语言 BASCOM

程序设计与应用

主编 张茂青

李彦超 张 敏 耿德根 编著

北京航空航天大学出版社

内 容 简 介

主要介绍类高级程序设计语言 BASIC 的 AVR 单片机开发平台——BASCOS - AVR。BASCOS - AVR 如同 BASIC 一样,具有程序设计简单易学、用简单的语句可实现较复杂的功能等特点。其开发平台的使用简洁方便,并具有实物图形化的仿真平台,可以直观地仿真程序的运行结果,是 AVR 单片机初学者快速入门的最佳工具。

本书通过大量的典型实例和屏幕截图进行直观的、由浅入深的论述,可作为单片机爱好者的自学读本,也可作高中、高职、技校、大专院校中自动化、计算机、电子等专业的教学参考书,亦可作为电子应用技术的培训教材。

图书在版编目(CIP)数据

AVR 单片机高级语言 BASCOS 程序设计与应用/张茂青主编.

北京:北京航空航天大学出版社,2005.1

ISBN 7-81077-566-9

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

中国版本图书馆 CIP 数据核字(2004)第 122944 号

AVR 单片机高级语言 BASCOS 程序设计与应用

主 编 张茂青

李彦超 张 敏 耿德根 编著

责任编辑 朱伟锋

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(100083) 发行部电话:010-82317024 传真:010-82328026

<http://www.buaapress.com.cn> E-mail:bhpress@263.net

涿州市新华印刷有限公司印装 各地书店经销

*

开本:787×1092 1/16 印张:16 字数:410 千字

2005 年 1 月第 1 版 2005 年 1 月第 1 次印刷 印数:5 000 册

ISBN 7-81077-566-9 定价:22.00 元

前 言

传统单片机系统的开发主要是用汇编语言编写应用程序。由于汇编语言程序的可读性、可移植性和结构性比较差,因此采用汇编语言编写单片机应用程序的时间比较长,调试和排错也比较困难,产品开发周期较长。为了提高编写应用程序的效率,改善程序的可读性和可移植性,缩短产品的开发周期,采用高级语言的开发平台来开发单片机系统已经成为发展趋势。

本书主要介绍类高级程序设计语言 BASIC 的 AVR 单片机开发平台——BASCOM-AVR。BASCOM-AVR 如同 BASIC 一样,具有程序设计简单易学、用简单语句可实现较复杂功能等特点。其开发平台使用简洁方便,并具有实物图形化的仿真平台,可以直观地仿真出运行结果。配合 AVR 单片机程序存储器具有可多次编程和在线下载的优点,使学习和使用 AVR 单片机变得十分容易和方便。使用 BASCOM-AVR 开发 AVR 单片机系统,设计人员可以在半个小时内完成一个功能模块的设计编程和调试;而采用汇编语言,则需要几天甚至几个星期。

本书不是讲述如何设计和开发一个复杂的实际产品,而是指导和帮助初学者通过自己的亲手实践,来熟悉掌握 BASCOM-AVR 的使用。因而本书中将较少涉及 AVR 的硬件设计,以软件仿真为主,硬件为辅。有关硬件设计内容可参考其他 AVR 相关资料。

计算机开发语言经过几十年的发展,涌现出越来越多的开发语言,但目前 BASIC 语言仍然是全球最流行的语言。经市场调查资料显示,使用 BASIC 语言的程序员人数(包括专业和非专业的)要远远多于使用其他语言的程序员人数。

BASCOM 语言是由个人计算机平台移植到单片机平台上的 BASIC 语言。BASCOM 语言继承了 BASIC 语言的编程特点——简单易学且便于使用,特别适合于具有 BASIC 语言基础的人员。

BASCOM 语言的优点如下:

1. 具有良好的可移植性

可移植性的好坏对编程语言来讲是非常重要的。BASCOM 语言具有良好的可移植性。AVR 系列单片机是具有很多型号的一个单片机产品系列,如果使用汇编语言开发,由于汇编语言依赖硬件性很强,所以若要将这个系列中某种型号单片机的代码移植到另一种型号单片机上,代码可能要做很大修改,而使用 BASCOM 语言就不同了。在其他语言中要做的修改,在 BASCOM 中仅仅修改一下头文件即可。下面分别为将用汇编语言和 BASCOM 语言编写的 ATmega8515 芯片的代码移植到 ATmega8535 芯片上所需做的修改。

汇编语言：

```

.include "m8515def.inc"
.org $000
rjmp main
.org $007
rjmp tim0_ovf
main: ldi r16,low(ramend)
      out spl,r16
      ldi r16,high(ramend)
      out sph,r16
ldi r16, $02
      out tmsk,r16
      ldi r16, $05
      out tccr0,r16
      ldi r17,131
      out tcnt0,r17
      ldi r16, $ff
      out ddrc,r16
      sei
here:  rjmp here
tim0_ovf: in r1,sreg
        ldi r17,131
        out tcnt0,r17
        in r18,portc
        com r18
        out portc,r18
        out sreg,r1
        reti

```

```

.include "m8535def.inc"
.org $000
rjmp main
.org $009
rjmp tim0_ovf
main: ldi r16,low(ramend)
      out spl,r16
      ldi r16,high(ramend)
      out sph,r16
ldi r16, $01
      out tmsk,r16
      ldi r16, $05
      out tccr0,r16
      ldi r17,131
      out tcnt0,r17
      ldi r16, $ff
      out ddrc,r16
      sei
here:  rjmp here
tim0_ovf: in r1,sreg
        ldi r17,131
        out tcnt0,r17
        in r18,portc
        com r18
        out portc,r18
        out sreg,r1
        reti

```

BASCOM 语言：

```

$regfile = "m8515.dat"
$crystal = 8000000
Config Timer0 = Timer , Prescale = 1024
Enable Interrupts
Enable Ovf0
On Ovf0 Overflow_isr
Ddrc = &.HFF
Tcnt0 = 131
Do
Loop
End
Overflow_isr:
    Tcnt0 = 131
    Portc = Not Portc
Return

```

```

$regfile = "m8535.dat"
$crystal = 8000000
Config Timer0 = Timer , Prescale = 1024
Enable Interrupts
Enable Ovf0
On Ovf0 Overflow_isr
Ddrc = &.HFF
Tcnt0 = 131
Do
Loop
End
Overflow_isr:
    Tcnt0 = 131
    Portc = Not Portc
Return

```

注：上面程序中加粗的内容为需要修改的内容。

2. 具有良好的可读性

采用 BASCOM 语言编程,不像汇编语言那样,实现某种功能就是对特定的寄存器赋指定的值,而是用很直观的语句来表示。例如,使定时器/计数器 0 上升沿计数的程序如下:

汇编语言:

```
ldi r16,$07  
out tccr0,r16
```

BASCOM 语言:

```
CONFIG TIMER0 = COUNTER,EDGE=RISING
```

通过上面的比较可以很容易地看出,用汇编语言编程,需要了解每一个特殊寄存器的功能;在实现某种功能时,必须了解向某寄存器赋某值。很多情况下,这些寄存器的名称与其实现的功能没有意义上的联系。例如,对于 tccr0,在没有任何资料的情况下,很难联想到它是定时器/计数器 0 的控制寄存器;因此,通过寄存器赋的值来联想它所实现的功能就更难了。就上例而言,若无资料,很难想象对 tccr0 赋 \$07 这个命令字实现什么功能。这些都只是规定,这就要求编程人员对这些名称和数值要熟记,手边还要有单片机资料备查。而 BASCOM 就不同了,它替我们屏蔽掉了很多底层的東西。BASCOM 的语句就是对实现功能的语言描述。就上例而言,若用 BASCOM 语句即为:

```
CONFIG TIMER0 = COUNTER,EDGE=RISING
```

一看这个语句,就可以把它要实现的功能猜得差不多了,因为它就是一个语言描述,很多语句就是英文单词,可以望词知意。

BASIC 语言在我国计算机知识普及过程中发挥的作用是有目共睹的。在过去的十几年中,BASIC 语言曾经引导许多人从一个“计算机盲”向计算机应用迈出了可贵的第一步。在我国至少有 1 000 多万人学习过 BASIC 语言,其中不少人在入门之后继续提高,成为计算机应用人才,取得了很大的成绩。作者希望通过本书对 BASCOM-AVR 的介绍,能够为单片机技术在我国推广尽自身的绵薄之力。

全书共 9 章,由张茂青担任主编。其中第 2 章和第 9 章的部分内容由耿德根执笔,其余各章均由李彦超、张敏和张茂青共同执笔。全书由张茂青统稿。在本书的编写过程中,得到了双龙电子公司的大力支持;本书开发实验器 SL-AVRAD 由双龙公司提供,书中程序均在该开发实验器上验证通过。

由于作者水平有限,只是为了满足初学者的需要而抛砖引玉,利用工作之余写成本书,可能会有一些缺点或错误,希望得到专家和读者的指正,在适当的时间再作修订补充,以便跟上单片机技术的发展需要。

读者在学习本书的过程中若有疑问,可以在由双龙电子赞助的 21IC BBS (AVR 单片机论坛)或双龙公司论坛(WWW. AVR. COM. CN 的 BASCOM 专栏)提出,本书的编者及论坛中其他热心网友会进行解答。

作者

2004 年 6 月于苏州大学

目 录

第 1 章 AVR 单片机与 BASIC 语言

1.1 BASIC 语言的特点	1
1.2 AVR 单片机的特点	2
1.3 用 BASIC 语言开发单片机的优势	2

第 2 章 BASCOM - AVR 集成开发环境

2.1 基于高级语言 BASCOM - AVR 的单片机开发平台	5
2.1.1 实验硬件系统	5
2.1.2 软件开发平台	8
2.2 BASCOM - AVR 的安装与使用	9
2.2.1 BASCOM - AVR 的安装	9
2.2.2 启动 BASCOM - AVR	14
2.2.3 BASCOM - AVR 使用快速入门	15
2.2.4 BASCOM - AVR 界面简介	19
2.2.5 键盘操作	23
2.2.6 BASCOM - AVR 的参数设置	24
2.2.7 软件仿真	41

第 3 章 BASCOM 程序设计基础

3.1 完整的 BASCOM 例子	49
3.2 BASCOM 基本语法介绍	49
3.2.1 BASCOM 程序的组成	49
3.2.2 字符集	51
3.3 数据类型	52
3.4 常量和变量	53
3.4.1 常 量	53
3.4.2 变 量	54
3.5 运算符和表达式	57
3.5.1 算术运算符	57
3.5.2 关系运算符	58
3.5.3 逻辑运算符	58
3.5.4 类型转换	59
3.6 常用标准函数	60

3.6.1 数值型函数	60
3.6.2 字符串函数	65
3.6.3 数组相关函数	74
3.7 常用语句介绍	75
3.7.1 延时语句	76
3.7.2 加 1 减 1 语句	78
3.7.3 交换两个相同类型变量中的内容	78
3.7.4 位运算	79
3.8 浮点数	82
3.8.1 浮点格式	82
3.8.2 十进制和二进制	83
3.8.3 整型数如何在计算机中表示	83
3.8.4 使用浮点数的运算误差	83
3.8.5 浮点数概念	84
3.8.6 IEEE 细节	84
3.8.7 浮点数其他常见误差	86

第 4 章 赋值与输入/输出语句

4.1 赋值语句	87
4.2 输入/输出语句	88
4.2.1 串口输入语句	89
4.2.2 串口输出语句	90
4.2.3 LCD 输出语句	92
4.2.4 小键盘数据输入	93
4.2.5 读数和置数语句	94
4.3 结束语句	96

第 5 章 BASCOM 语言流程控制

5.1 选择语句	97
5.1.1 两路分支选择结构	97
5.1.2 多行 IF 语句	99
5.1.3 无条件转移语句(GOTO 语句)	102
5.1.4 多路分支选择结构	102
5.1.5 多分支转移语句(ON VALUE 语句)	105
5.1.6 程序举例	106
5.2 循环语句	109
5.2.1 WHILE 循环语句	109
5.2.2 DO 循环语句	110
5.2.3 FOR 循环语句	111

5.2.4 循环嵌套	113
5.2.5 程序举例	114

第6章 BASCOM 语言结构化设计——过程的使用

6.1 过程的声明	117
6.1.1 SUB 过程的声明	117
6.1.2 用户自定义 FUNCTION 过程的声明	118
6.2 过程的定义	118
6.2.1 SUB 过程的定义	118
6.2.2 FUNCTION 过程的定义	119
6.3 过程的调用	120
6.3.1 调用 SUB 过程	120
6.3.2 调用 FUNCTION 过程	121
6.4 参数传送	122
6.4.1 形参与实参	122
6.4.2 引用和传值	122
6.5 变量的作用域	124
6.5.1 局部变量	124
6.5.2 全局变量	126
6.6 过程的嵌套	127
6.6.1 SUB 过程嵌套	128
6.6.2 FUNCTION 过程嵌套	129
6.7 转子与返回	129

第7章 AVR BASCOM 语言应用实例

7.1 一个程序代码完整的编写、调试和下载过程	134
7.2 I/O 端口输出的简单例子	143
7.3 使用 AVRStudio 进行程序代码调试	144
7.4 模仿 CD 或 MP3 播放器液晶屏显示	153
7.5 LCD 中显示自定义字符	157
7.6 贪吃蛇程序	161
7.7 发音程序	164
7.8 控制微型伺服马达	167

第8章 Atmel 单片机硬件结构

8.1 AVR 单片机体系结构概述	171
8.2 存储器	172
8.2.1 Flash 程序存储器	172
8.2.2 数据存储器	172

8.2.3	EEPROM 存储器	175
8.2.4	BASCOM 中存储器操作语句	176
8.2.5	存储器部分在仿真环境中的使用	178
8.3	并行 I/O 端口	182
8.3.1	I/O 端口相关寄存器	182
8.3.2	I/O 端口在仿真环境中的使用方法	185
8.4	中 断	189
8.4.1	中断的概念	189
8.4.2	中断的作用	189
8.4.3	中断优先级	189
8.4.4	中断源	189
8.4.5	中断的执行过程	191
8.4.6	中断仿真	196
8.5	定时器/计数器	199
8.5.1	基本概念	199
8.5.2	定时器/计数器 0(T/C0)	199
8.5.3	定时器/计数器 1	207
8.5.4	定时器/计数器 2	217
8.6	USART 串口通信	218
8.6.1	串行通信的基本概念	218
8.6.2	AVR 系列单片机串行接口	219
8.6.3	BASCOM 中有关 USART 操作	219
8.7	模/数转换	221
8.7.1	A/D 转换在仿真环境中使用	222
8.7.2	A/D 转换应用实例	223
8.8	模拟比较器	227

第 9 章 AVR 单片机存储器的程序下载

9.1	概 述	228
9.2	ISP 下载器接口	228
9.3	ISP 串行下载操作	229
9.3.1	启动 SL ISP 下载软件的方法	229
9.3.2	SL ISP 下载软件介绍	230

附录 A BASCOM 错误代码

附录 B ASCII 码

参 考 文 献

第 1 章 AVR 单片机与 BASIC 语言

1.1 BASIC 语言的特点

BASIC 是 Beginner's All - purpose Symbolic Instruction Code(初学者通用符号指令代码) 的缩写,是国际上广泛应用的一种计算机高级语言。由于 BASIC 语言简单易学,目前仍是计算机入门的主要学习语言之一。

BASIC 是 20 世纪 60 年代中期,由美国达特茅斯学院的 John Kemeney 和 Thomas Kurtz 研发所得。最初它是为教学目的而设计的,到了 20 世纪 70 年代,BASIC 已成为教学上主要的程序语言(Program Language),很多学校都将它列为学习计算机语言的入门课程。

BASIC 语言自其问世以来共经历了以下 4 个阶段:

第 1 阶段:(1964 年~70 年代初) BASIC 语言问世;

第 2 阶段:(1975 年~80 年代中)微机上固化的 BASIC 语言;

第 3 阶段:(80 年代中~90 年代初)结构化 BASIC 语言;

第 4 阶段:(1991 年以来) Visual BASIC。

BASIC 是一种易学易用的高级语言,非常适合初学者学习应用。常用的编译软件有 True BASIC、TurboBASIC、QuickBASIC、Visual BASIC、CAREALIZER、GFA BASIC 和 POWER BASIC 等。在众多的 BASIC 版本中,由 Microsoft 公司推出的 Visual BASIC 使用最为广泛。

1991 年,Microsoft 公司推出了 Visual BASIC1.0 版。这在当时引起了很大轰动。许多专家把 VB 的出现当作是软件开发史上一个具有划时代意义的事件。其实,以我们现在的目光来看,VB1.0 的功能实在是太弱了。但在当时,它是第一个“可视”的编程软件。这使得程序员欣喜之极,都尝试在 VB 平台上进行软件创作。Microsoft 公司也不失时机地在 4 年内接连推出 VB2.0、VB3.0 和 VB4.0 3 个版本。并且从 VB3.0 开始,Microsoft 公司将 ACCESS 的数据库驱动集成到了 VB 中,这使得 VB 的数据库编程能力大大提高。从 VB4.0 开始,引入了面向对象的程序设计思想;近而 VB 还引入了“控件”的概念,使得大量已经编好的 VB 程序可以被我们直接拿来使用。如今,VB 已经有了 VB.NET(VB7.0)版。

通过几年的发展,VB 已成为一种真正专业化的开发语言。很多用户认为可用 Visual BASIC 快速创建 Windows 程序,现在还可以编写企业水平的客户/服务器程序以及强大的数据库应用程序。Visual BASIC 新版中还将会有更多的可用功能。

从 BASIC 语言近 40 年的发展历史可以看出,由于它具有易学易用的特点,将来会有更快的发展。我们相信,随着 BASIC 语言功能的不断强大,其使用者的数量将不断增长。

1.2 AVR 单片机的特点

ATMEL 公司的 AVR 系列单片机是增强 RISC 内载 Flash 的单片机。AVR 系列单片机基于新的精简指令 RISC 结构。这种结构是在 20 世纪 90 年代开发出来的,它综合了半导体集成技术和软件性能的新结构。这种结构使得 AVR 单片机在 8 位微处理器市场上具有最高 1 MIPS/MHz 的能力。

为了加快进入市场的时间和简化维护支持,对于单片机来说,用高级语言编程成为一种标准编程方法。AVR 单片机的开发目的就在于能采用高级语言编程,从而高效地开发出目标产品。为了对目标代码的大小、性能及功耗进行优化,AVR 单片机采用了大型快速存取寄存器文件和快速单周期指令。

快速存取 RISC 寄存器文件由 32 个通用工作寄存器组成。传统的、基于累加器的大量程序代码,用以实现在累加器与存储器之间的数据传送。在 AVR 单片机中,用 32 个通用工作寄存器代替累加器,从而可以避免传统累加器与存储器之间数据传送所造成的瓶颈现象。

在 AVR 单片机中,当前一条指令执行时,即可取出现行指令,然后用 1 个周期执行指令。在其他的 CISC 以及类似的 RISC 结构中,外部振荡器的时钟被分频降低到传统的内部执行周期,这种分频最大达 12 倍。AVR 单片机是用 1 个时钟周期执行 1 条指令的,它是 8 位单片机中第一个真正的 RISC 单片机。

AVR 单片机有良好的性能价格比,该系列中有引脚少的器件,也有含较大容量存储器引脚较多的器件。由于 AVR 单片机采用的是 Harvard 结构,故它们的程序存储器和数据存储器是分开的。可直接访问 8 MB 程序存储器和 8 MB 数据存储器。寄存器文件被双向映射,并可被访问,如同片内允许快速上下转换的那部分 SRAM 存储。

AVR 单片机采用低功率、非挥发的 CMOS 工艺制造,通过 SPI 口和一般的编程器,可对 AVR 单片机的 Flash 存储器进行编程。

由于 AT90 系列单片机存在一些设计上的缺陷,目前已停产(但是市场上仍可以购买到该系列单片机),取而代之的型号为 ATmega8515 和 ATmega8535。

1.3 用 BASIC 语言开发单片机的优势

传统的单片机系统开发主要是用汇编语言编写系统程序。由于汇编语言程序的可读性、可移植性和结构性比较差,因此采用汇编语言编写单片机应用系统程序的周期较长,调试和排错也比较困难,产品开发周期长。为了提高编写系统和应用程序的效率,改善程序的可读性和可移植性,缩短产品的开发周期,采用高级语言的开发平台来开发单片机系统已经成为发展趋势。

在众多的高级语言中,BASIC 语言以其简单易学的特点,被广大初学者所选用。以高级程序设计语言 BASIC 为手段的 AVR 单片机开发平台——BASCOM - AVR,完全继承了 BASIC 简单易学的特点,并且它又针对单片机的特点,扩展了专门面向各种通用接口且功能强大的语句,使设计人员可以在半个小时内完成一个功能模块的设计编程和调试;而采用汇编语言,则需要几天甚至几个星期。BASCOM - AVR 集成开发环境中包含的实物图形化的仿

真平台,使用户可以在程序下载到单片机之前,先仿真运行程序,仿真正确后,再将程序下载到单片机中。这样可以减少向单片机下载程序的次数,提高编程效率。

由于 BASCOM 诸多的优点,使得它在国内的单片机爱好者中非常流行。网上关于 BASCOM 的资料也非常多,有条件的读者,可以上网搜索。目前 BASCOM 语言在国内的发展还处于起步阶段,但我们相信,凭借 BASIC 语言在国内的广泛应用基础和 BASCOM 本身简单易学的特点,它将很快在国内单片机编程爱好者中流行起来。

为了使大家能够更好地了解 BASCOM 语言,下面分别给出使用汇编语言、C 语言和 BASCOM 语言来实现的单片机 ATmega16 定时器/计数器 0 产生对称方波的例子。

◆ 汇编语言

```
.include "m16def.inc"
.org $000
rjmp main
.org $009
rjmp tim0_ovf                ;转定时器 0 溢出处理
main:      ldi r16,low(ramend)    ;栈指针置初值
           out spl,r16
           ldi r16,high(ramend)
           out sph,r16
           ldi r16,$01           ;TIMER0 溢出中断使能
           out tmsk,r16
           ldi r16,$05           ;1 024 分频
           out tccr0,r16
           ldi r17,131          ;T/CO 置初值 131
           out tcnt0,r17
           ldi r16,$ff          ;PA 口作输出
           out ddra,r16
           sei                  ;全局中断使能
here:      rjmp here            ;死循环,等待中断
tim0_ovf:  in r1,sreg            ;保存 sreg
           ldi r17,131
           out tcnt0,r17
           in r18,porta         ;读 A 口数据寄存器
           com r18              ;数据取反
           out porta,r18       ;写 A 口数据寄存器
           out sreg,r1         ;恢复 sreg
           reti
```

◆ C 语言

```
#include<Mega16.h>
interrupt[TIM0_OVF]void timer0_ovf_isr(void)
{
```

```

TCNT0=131;                // T/C0 重新置初值 131
PORTA=~PORTA;             // A 口输出数据取反
}

void main()
{
DDRA=0xff;                // A 口定义为输出
TCCR0=0x05;               //分频系数为 1 024
TCNT0=131;                // T/C0 置初值 131
TIMSK=0x01;               // TIMER0 溢出中断使能
#asm("sei")               //全局中断使能
while(1)                  //死循环,等待中断
;
}

```

◆ BASCOM 语言

```

$regfile = "m16def.dat"
$crystal = 8000000
Config Timer0 = Timer , Prescale = 1024 ' T/C0 作定时器使用,分频系数为 1 024
Enable Interrupts                      '全局中断使能
Enable Ovf0                            ' TIMER0 溢出中断使能
On Ovf0 Tim0_ovf_isr                   '定义 TIMER0 溢出中断服务程序入口地址
Ddra = &.HFF                          ' A 口定义为输出
Tcnt0 = 131                            ' T/C0 置初值 131
Do                                     '死循环,等待中断产生
Loop
End
Tim0_ovf_isr:                          ' TIMER0 溢出中断服务程序
Tcnt0 = 131                            ' T/C0 重新置初值 131
Porta = Not Porta                     ' A 口输出数据取反
Return

```

第 2 章 BASCOM - AVR 集成开发环境

本章主要介绍 BASCOM - AVR 的集成开发环境(Integrated Development Environment, IDE)。该集成开发环境包含程序代码开发、调试和仿真的平台。本章将详细地介绍开发环境的安装以及开发环境使用的快速入门,并且用大量的屏幕截图,直观、通俗地介绍该集成开发环境的使用。

2.1 基于高级语言 BASCOM - AVR 的单片机开发平台

BASCOM - AVR 的单片机开发平台由实验硬件系统和软件开发平台两部分组成。

2.1.1 实验硬件系统

SL - AVRAD 单片机开发实验系统是广州天河双龙电子有限公司开发的新产品,是一种学习、开发、下载和实验四合一的单片机开发实验系统(等于 AVR 编程器+模拟仿真器+实验器+科研样机)。该开发实验器的硬件采用模块化设计,便于用户灵活组成科研项目所需的硬件结构。

SL - AVRAD 单片机开发实验系统实物图如图 2.1 所示,硬件接口电路布局图如图 2.2 所示。

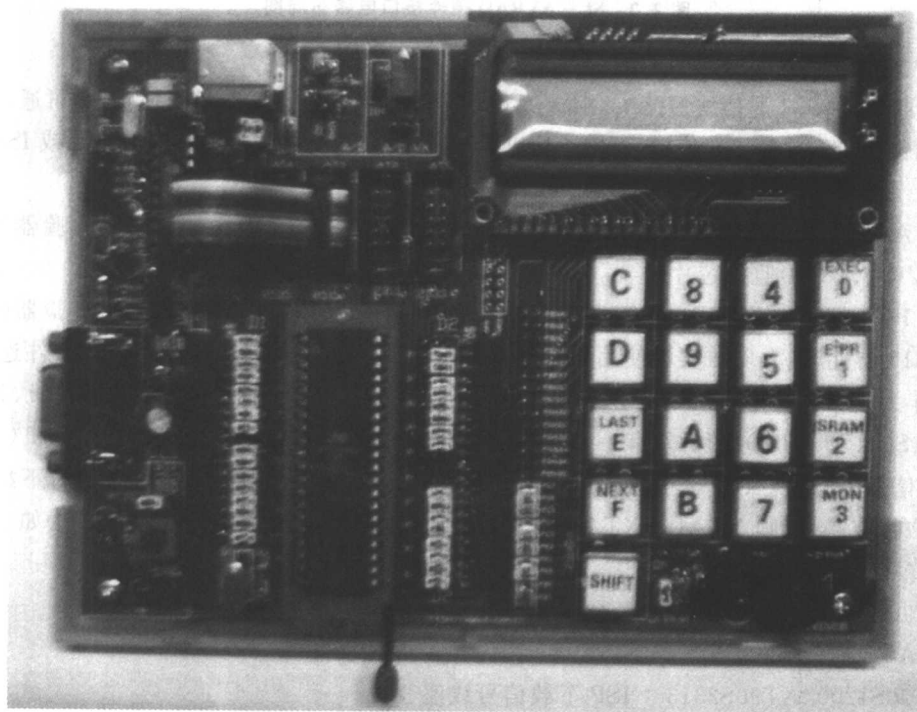


图 2.1 SL - AVRAD 单片机开发实验系统实物图

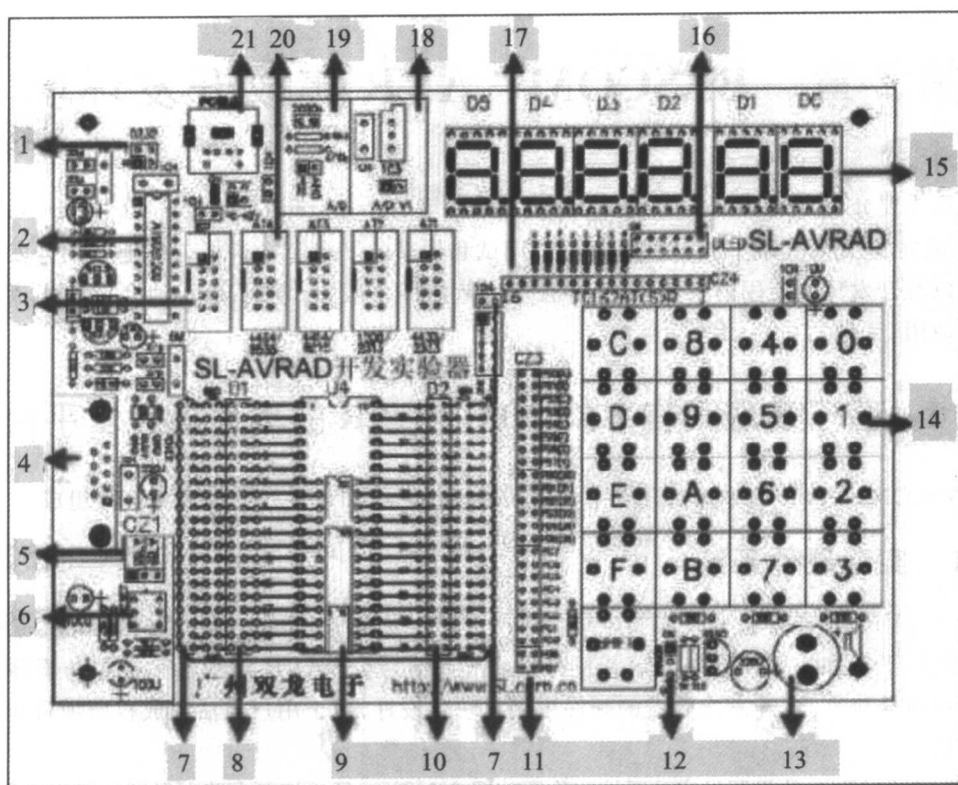


图 2.2 SL-AVRAD 硬件接口电路布局图

硬件接口电路各部分说明如下：

(1) D232: D232 通信线短路块。插上短路块,即接到 AT90S2313;拔出短路块,可通过 T 和 R 端用插针线接到单片机通信口,可做单片机异步通信 UART 或主从同步通信 SPI 或 ISP 下载通信。

(2) AT90S2313: AVR 开发实验器监控芯片。AT90S2313 为增强型开发实验器通用单片机(AVR/AT89S)监控芯片。

(3) ISP: DC10 的(ISP)插座,即 AVR 单片机的下载信号插座。本开发实验器配一片 ATmega16 器件,绝大多数实验使用该器件,硬件(用短路块)连接出厂时也按该器件连接,其他器件作为选配件。ISP 下载插座引脚功能分别为 VCC、GND、XTAL2、XTAL1、MOSI、 $\overline{\text{RESET}}$ 、SCK 和 MISO,随机附有一条 10 线信号线,由用户接插到对应 AVR 单片机(AT1~AT4)的信号引脚上。ISP 也可接到用户板,用于 AVR 单片机(或用转接线)的串行下载编程。如果用户板有晶振,则 XTAL1/XTAL2 两信号线无需接出。AT1~AT4 插座接法如下:

● AT1 插针座为:

AT90S4433/AT90S2333 ISP 下载信号线座

● AT2 插针座为:

AT90S1200/AT90S2313 ISP 下载信号线座

● AT3 插针座为: