

## 第八章 BASCOM-AVR 的应用

### 8.1 基于高级语言 BASCOM-AVR 的单片机开发平台

传统开发单片机微控制器系统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、可移植性和结构性都较差，因此采用汇编语言编写单片机微控制器应用系统程序的周期长，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微控制器系统已成为发展趋势。概括起来说，基于高级语言开发平台开发单片机微控制器系统具有语言简洁，使用方便灵活，可移植性好，表达能力强，可进行结构化程序设计，可进行软件仿真等优点。实践表明，采用高级语言开发平台进行单片机微控制器系统开发的效率比使用汇编语言高几倍甚至几十倍。ATMEL 公司的 AVR AT90 系列单片机微控制器是基于新的精简指令 RISC 结构的，其开发目的就是在能采用高级语言编程，从而能高效地开发出目标产品。目前国际上已有许多公司推出了 C、Basic 等基于高级程序设计语言的 AVR 开发软件和平台。

本章将采用一个以高级程序设计语言 BASIC 为手段的 AVR 单片机微控制器开发平台 --BASCOM-AVR。它的程序设计简洁、方便，专用的面向各种通用接口、且功能强大的语句，实物图形化的仿真平台等特点，配合 AVR 单片机微控制器程序存储器可多次编程、在线下载的优点，使学习和使用 AVR 单片机微控制器变得十分容易。学会使用 BASCOM-AVR 来开发 AVR 单片机微控制器系统，设计人员可以在半个小时就能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。当你在开始学习本章内容，自己动手实践一、二个实验后，你就会体会到本章的精华所在。

★ BASCOM-AVR 语言更详细使用说明见屏幕菜单上的帮助文件<<Help--Index>>有关内容，可用东方快直翻译查看！

#### 8.1.1 本章采用的软件开发平台和硬件系统介绍

本章目的不是讲述如何去设计和开发一个复杂的实际产品，而是以动手实践为主，指导和帮助学习者使用 BASCOM-AVR 快速开发 AVR 单片机微控制器系统的方法。尽管本章的实验都是实现一些基本和简单的功能，但它们往往就是一个实际产品的基本功能模块。当把它们有机地结合起来，就可形成一个实际的电子产品。

在学习本章和动手 DIY 前，先了解和建立一个简单的实践环境。该环境由以下几部分组成：PC 机一台，运行 Windows95/98；

AVR 开发仿真软件平台 BASCOM-AVRDEMO，(<http://www.mcselec.com>，Free)；

AVR 程序下载软件 AVRProg，(<http://www.atmel.com>，Free)；

BASCOM-AVR 开发实验程序由华东师范大学电子科学技术系(ATMEL AVR 实验室)实验通过([ma-chao@online.sh.cn](mailto:ma-chao@online.sh.cn))。

AVR 单片机开发实验硬件基本系统是 SL-AVR 开发下载实验器，能完成实验所需的硬件外围组件和电子元器件。

##### 一. 实验硬件系统介绍

本篇所有实验采用的硬件环境是由 AVR 单片机开发实验基本系统和完成实验所需的硬件外围组件及相应的电子元器件组成。

AVR 单片机开发实验基本系统包括：AVR 单片机开发实验最小系统 SL-AVR 开发下载实验器；RS-232 串口通讯电缆；5V 直流电源和连接引线等。

本篇所有实验均需基于该系统来实现。利用 AVR 单片机开发实验基本系统，可以完成一些基本的实验，学习和掌握 AVR 微控制器的基本特性和使用、AVR 开发仿真软件、程序下载软件、AVR 应用程序设计，以及设计和开发基于 AVR 微控制器的简单电子系统的手段、方法和过程。进一步配合一些硬件外围组件和相应的电子元器件，就可以完成实现一些复杂实验。AVR 开发实验基本系统不仅可用于高校相关专业开设实验，同时也可作为高中、职业学校、青少年科技站开设课程和科技制作应用。对于电子工程师和电子产品设计开发人员，本篇介绍的开发环境和 AVR 开发实验基本系统也是一套高效、简便、实用的产品设计研制开发工具。

★ SL - AVR 开发下载实验器详见<<第三章 开发工具 3.3 AVR 单片机的串行下载>>。

## 二. 软件开发平台介绍

BASCOM-AVR 是 MCS 公司推出的面向 AVR 单片微控制器系列，采用高级程序设计语言 Windows BASIC 的软件开发平台。它的运行环境是 Windows95/98/NT。其主要特点有：

采用可带语句标示符的结构型 BASIC 高级程序设计语言编程结构化的 IF-THEN-ELSE-ENDIF、DO-LOOP、WHILE-WEND、SELECT-CASE 程序设计变量名和语句标示符长达 32 个字符，有位(Bit)、字节(Byte)、整型(Integer)、字(Word)、长型(Long)、字符串(String)多种类型的变量编译产生的运行代码可在所有带内部存储器的 AVR 微控制器微中运行程序语句和 Microsoft VB/QB 高度兼容为标准 LCD 显示器，I2C 芯片和单总线协议芯片等扩充了专用语句内置模拟终端和程序下载功能内置软件仿真平台用于测试优良的程序编辑功能完善的联机帮助功能和大量的例程 DEMO 版本可生成 2KB 程序代码，完全适用于 AT90S2313。

## 8.2 BASCOM-AVR 软件平台的安装与使用

目的: 掌握 AVR 开发平台 BASCOM-AVR 和程序下载软件 AVRProg 的安装与初步应用。

原理: 建立了解和使用 BASCOM-AVR 开发平台是第一步, 本实验将介绍这两个软件工具的安装、基本参数的设置和初步的应用。

器材与器件:

- PC 机一台, 运行 Windows95/98;
- BASCOM-AVR DEMO 版安装软件包;
- AVRProg 安装软件包;

BASCOM-AVR 是 MCS Electronics 公司推出的基于 AVR 系统的软件开发仿真平台。尽管 DEMO 版本仅可生成 2KB 程序代码, 但足可用于实验和学习, 而且完全适合于开发 AT90S2313, 因为 AT90S2313 的最大程序代码容量既为 2KB。

BASCOM-AVR DEMO 版安装软件包由两张 3" 磁盘组成, 用户可到 <http://www.mcselec.com> 免费下载, 或 <<双龙 AVR 电子书光盘>> 得到。

AVRProg 是 ATMEL 公司提供的用于 AVR 系列微控制器程序下载免费软件。我们用它将 BASCOM-AVR 生成的运行代码下载(Download)到 AVR 芯片中。用户可从 ATMEL 公司的网站 <http://www.atmel.com> 或广州天河双龙电子有限公司的网站 <http://WWW.SL.COM.CN> 或 <<双龙 AVR 电子书光盘>> 得到。



### 8.2.1 BASCOM-AVR 和 AVRprog131.exe 的安装

#### 1. 安装 BASCOM-AVR

用 Winzip 将 BASCOM-AVR DEMO 版安装软件包的 `bascom-avr1`, `bascom-avr2` ZIP 文件分别解压到硬盘的临时目录 TEMP 下。

双击运行临时目录下的软件安装程序 SETUP.EXE, 出现安装画面后单击  继续安装过程。阅读软件版权说明后, 单击  继续安装过程, 输入你的名字和公司名称后单击 。以后均单击 , 采用缺省设置, 直到安装结束。

#### 2. 安装 AVRProg

单击运行 Aprogwing.exe 自解压程序, 将下载软件 AVRprog131.exe 解压到硬盘中, 可放在与 BASCOM-AVR 同一个目录下。

### 8.2.2 运行 BASCOM-AVR, 编写 BASIC 源程序

BASCOM-AVR 主窗口

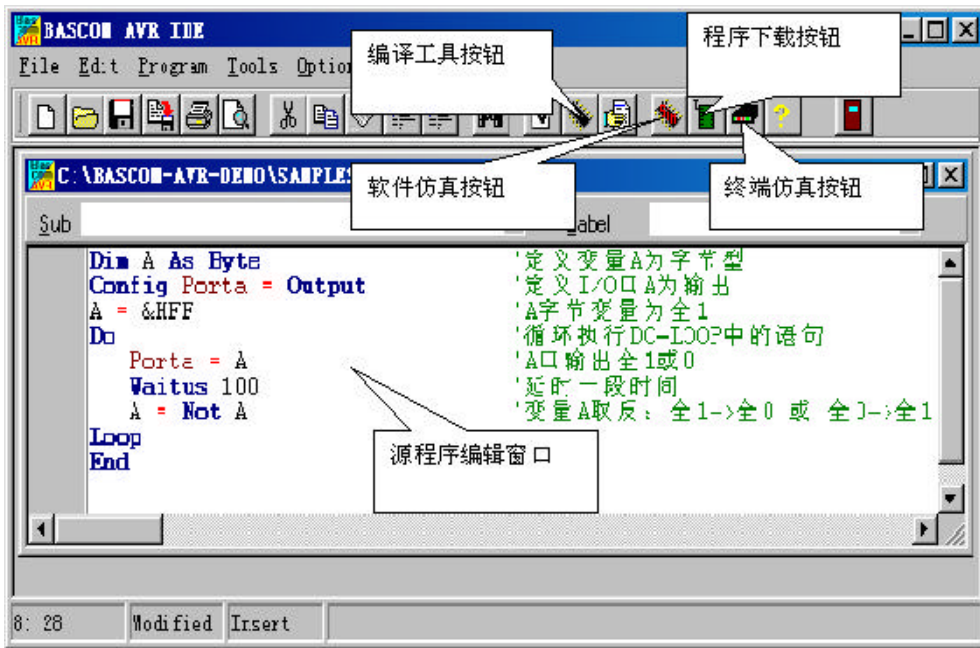


图 8.2.1

运行 BASCOM-AVR，编写一简单的 LED 发光管的 BASIC 控制源程序 exp1.bas。

### 8.2.3 BASCOM-AVR 系统参数设置

选择 **Option**->**Compiler**->**Chip**，进行参数设置(图 8.2.2)。

选择实验所用芯片 90S8515，继续进入 **Output** 设置。

选定输出选项(见图 8.2.3)

单击 **Programmer**，设定使用的执行代码下载程序。

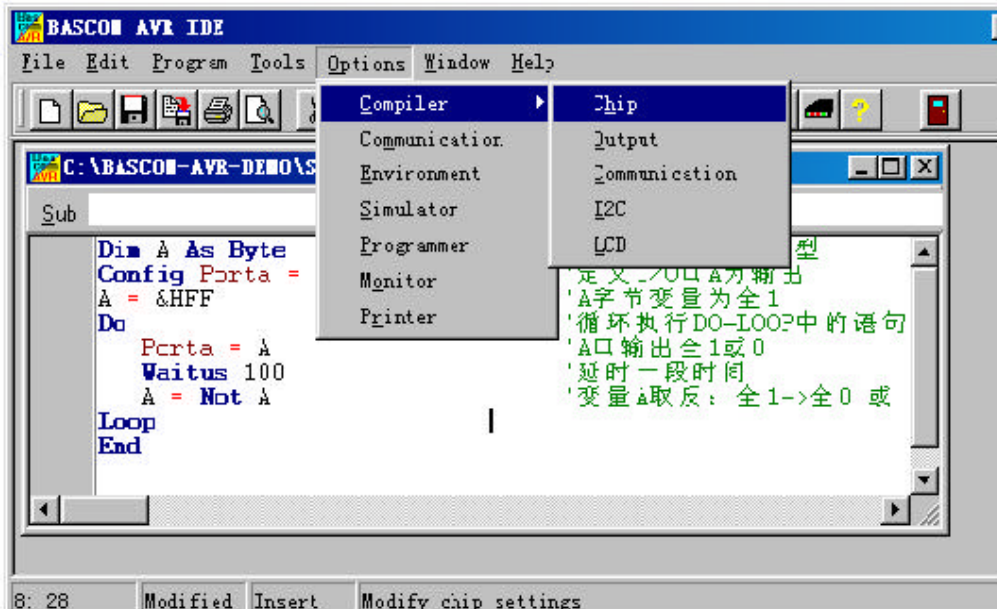


图 8.2.2

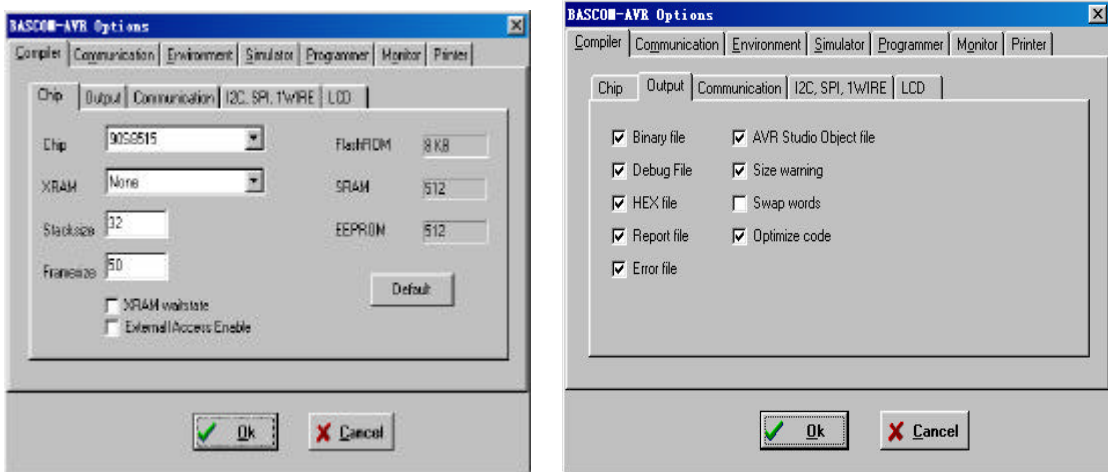


图 8.2.3

采用外部的程序下载器(External programmer)

设定下载程序所在目录和程序名(本例为 c:\BASCOM-AVR-DEMO\AVRprog131.exe)

选择 HEX 格式的下载代码文件类型,单击 **Ok**, 完成参数设置(图 8.2.4)

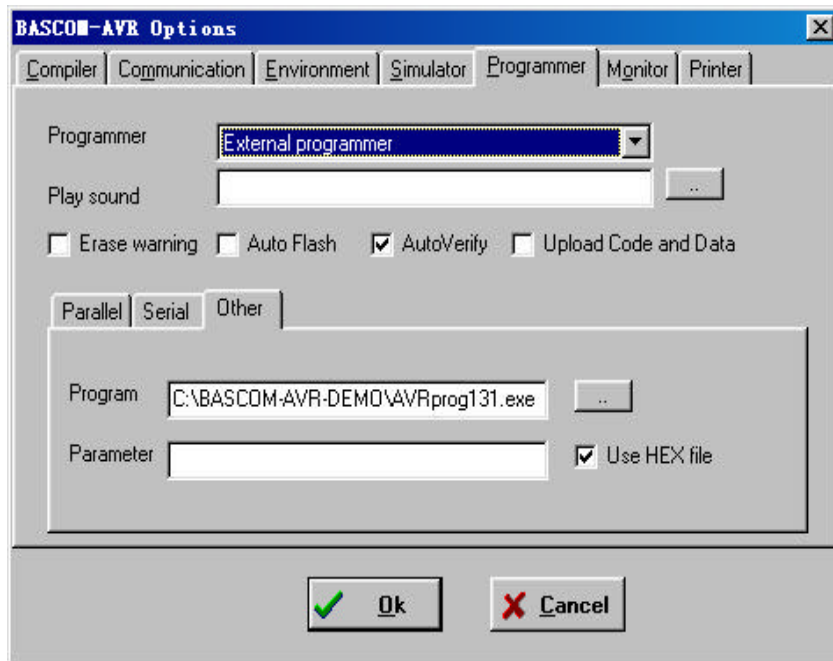


图 8.2.4

#### 8.2.4 编译源程序，生成各类代码文件

单击 BASCOM 主窗口工具条中的编译按钮，将 exp1.bas 编译生成可供仿真、下载的 dbg、obj、hex 等文件。

## 8.2.5 软件仿真

单击 BASCOM 主窗口工具条中的仿真按钮，进入软件仿真窗口。

单击硬件模拟按钮，打开硬件模拟窗口运行程序进行模拟仿真，可看到硬件模拟窗口中 Port a 口的 LED 的闪烁(图 8.2.5)。

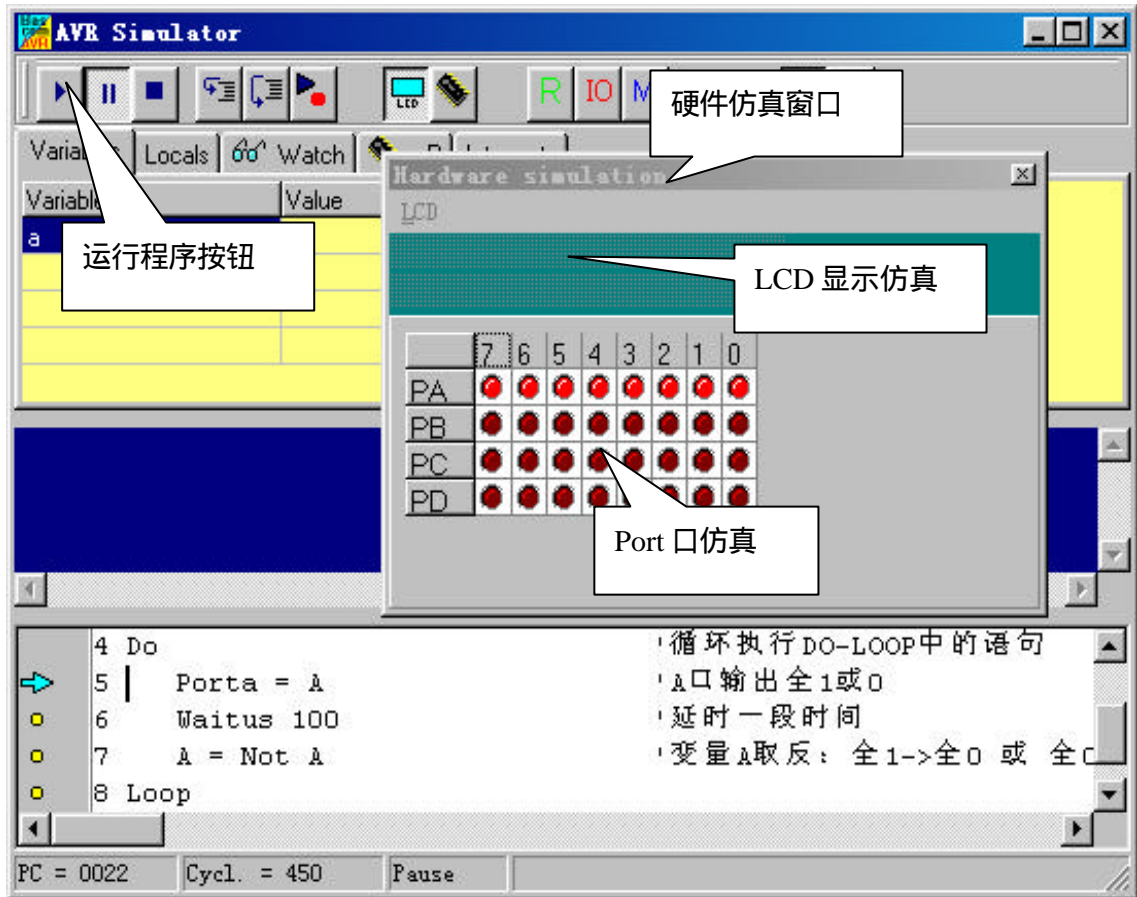


图 8.2.5

### 8.3 AVR 输入输出 I/O 口的应用

AT90S8515 有 Porta、Portb、Portc、Portd 四个口，共 32 根引脚。每个引脚都可以单独定义为输入或输出使用。

#### 8.3.1 LED 发光二极管的控制

**目的：**利用 AVR 的 I/O 口控制 LED 发光二极管。

**原理：**AVR 的 I/O 口输出为低电平“0”，点亮 LED 发光二极管；输出为高电平“1”时，LED 发光二极管熄灭。

**原理图：**图 8.3.1

**功能：**8 个 LED 发光二极管 1 秒钟间歇闪烁。

**程序：**exp1.bas

```

$sim          \此语句用于软件仿真，程序正式下载时清除
Dim A As Byte \定义变量 A 为字节型
Config Porta = Output \定义 Porta 口为输出
A = &HFF     \字节变量 A 为 11111111
Do           \循环执行 DO-LOOP 中的语句
    Porta = A          \Porta 口输出全“1”或“0”
    Wait 1            \延时一秒
    A = Not A         \变量 A 取反：全 1->全 0 或 全 0->全 1
Loop
End

```

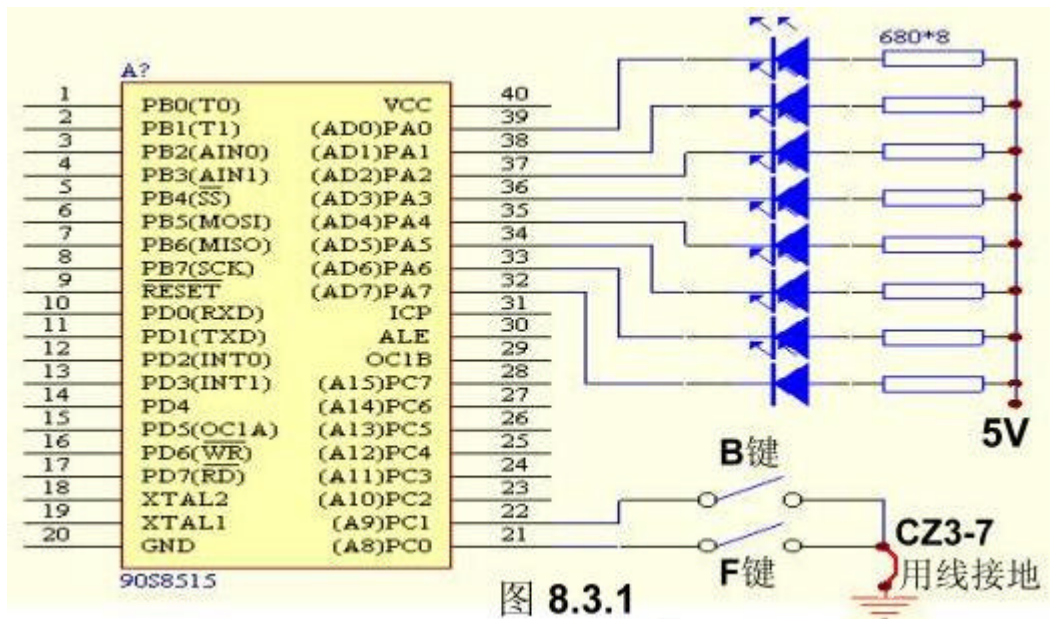


图 8.3.1

**操作过程：**

编写输入 BASIC 源程序。

编译生成各类文件。

利用 BASCOM-AVR 的软件仿真平台仿真调试。

使用 AVRprog 将程序代码下载到 90S8515。

#### 8.3.2 简易手控广告灯

**目的：**利用 AVR 的 I/O 口实现简易手控广告灯。

**原理：**AVR 的 Porta 口输出控制 8 路发光 LED 作为广告灯，Portc.0 和 Portc.1 为输入，控制发光

LED 的闪烁方式。

Portc.0	Portc.1	8 路发光 LED 闪烁方式
0	0	循环左移
0	1	循环右移
1	0	明暗交替
1	1	随机闪烁

原理图：图 6.3.1

程序：exp2.bas

```

Dim A As Byte , B As Byte           `定义变量 A, B 为字节型
Config Porta = Output                `定义 Porta 口为输出, 控制 LED
Config Pinc.0 = Input , Pinc.1 = Input `定义 Portc.0,Portc.1 为输入控制端

Portc.0 = 1 : Portc.1 = 1           `Portc.0,Portc.1 上拉电阻有效, 为"11"
Do
  Select Case B
    Case 0:
      A = &H01
      While B = 0                    `输入控制为"00"
        Porta = A
        Wait 1
        Rotate A , Left , 1          `左移循环
        B = Pinc And &H03           `读 Portc 口的控制信号
      Wend
    Case 1:
      A = &H80
      While B = 1                    `输入控制为"01"
        Porta = A
        Wait 1
        Rotate A , Right , 1         `右移循环
        B = Pinc And &H03
      Wend
    Case 2:
      A = &H00
      While B = 2                    `输入控制为"10"
        Porta = A
        Wait 1
        A = Not A                    `明暗交替
        B = Pinc And &H03
      Wend
    Case 3:
      While B = 3                    `输入控制为"11"
        Porta = Rnd(255)             `产生随机数,随机闪烁
        Wait 1
        B = Pinc And &H03
      Wend
  
```



```

Wend
End Select
Loop
End
    
```

8.3.3 简易电脑音乐放音机

**目的:** 利用 AVR 单片微控制器的 I/O 口输出一定长度和频率的脉冲信号，再经过信号放大，由耳机或喇叭放出乐曲声。

**原理:**

如何产生音乐频率:

利用 BASCOM-AVR 的 SOUND 语句，可以很方便的产生一定长度和频率的脉冲信号。

语句: Sound pin, duration, pulses

说明: pin 指定的 I/O 引脚，如 Portb.0 等。

duration 产生脉冲的时长。

pulses 产生脉冲的频率数。

产生脉冲时长和频率的简单计算表

晶振频率: 4MHZ		一拍: 100							
5 1276	6 1136	7 1012	#1 903	#2 804	4 716	5 638	6 568	7 506	#1 451
#5 1205	#6 1073	1 956	2 865	3 759	#4 676	#5 602	#6 536	1 470	2 426

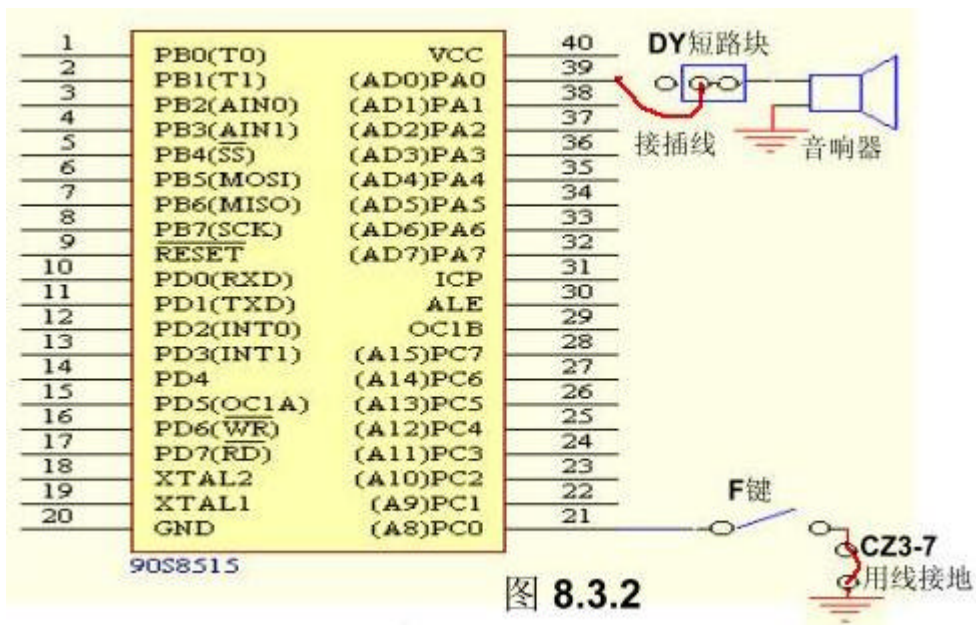


图 8.3.2

原理图:

程序: exp3.bas

Dim S As Integer , F As Integer

Dim A As Byte

```

Config Pinc.0 = Input           `定义 Portc.0 为输入控制端
Portc.0 = 1                     `Portc.0 上拉电阻有效, 为"1"
A = Pinc And &H01              `读选曲控制
Do
  Select Case A                 `第一首歌
    Case 0:
      While A = 0
        Restore Music_1
        Do
          Read S : Read F
          If S = 0 And F = 0 Then Exit Do
          Sound Porta.0 , S , F
          Waitms 50
        Loop
        Wait 2
        A = Pinc And &H01
      Wend
    Case 1:                       `第二首歌
      While A = 1
        Restore Music_2
        Do
          Read S : Read F
          If S = 0 And F = 0 Then Exit Do
          Sound Porta.0 , S , F
          Waitms 50
        Loop
        Wait 2
        A = Pinc And &H01
      Wend
  End Select
Loop
End

```

```

Music_1:                         `两只老虎曲调
Data 100% , 956% , 100% , 865% , 100% , 759% , 100% , 956%
Data 100% , 956% , 100% , 865% , 100% , 759% , 100% , 956%
Data 100% , 759% , 100% , 717% , 200% , 638%
Data 100% , 759% , 100% , 717% , 200% , 638%
Data 50% , 638% , 50% , 568% , 50% , 638% , 50% , 717%
Data 100% , 759% , 100% , 956%
Data 50% , 638% , 50% , 568% , 50% , 638% , 50% , 717%
Data 100% , 759% , 100% , 956%
Data 100% , 865% , 100% , 1276% , 200% , 956%

```

Data 100% , 865% , 100% , 1276% , 200% , 956% , 0% , 0%

Music\_2:

、生日快乐曲调

Data 75% , 1276% , 25% , 1276% , 100% , 1137% , 100% , 1276% , 100% , 956%

Data 200% , 1012% , 75% , 1276% , 25% , 1276% , 100% , 1137%

Data 100% , 1276% , 100% , 865% , 200% , 956%

Data 75% , 1276% , 25% , 1276% , 100% , 638% , 100% , 759% , 100% , 956%

Data 100% , 1012% , 100% , 1136%

Data 75% , 717% , 25% , 717% , 100% , 759% , 100% , 956%

Data 100% , 865% , 200% , 956% , 0% , 0%

## 8.4 LCD 显示器

LCD 液晶显示器在许多电子产品中都有着广泛的应用。BASCOM-AVR 中有专用的语句，可用于开发标准接口的 40 \* 4, 16 \* 1, 16 \* 2, 16 \* 4, 20 \* 2, 20 \* 4, 16 \* 1a 等多种字符液晶显示器。

### 8.4.1 标准 LCD 显示器的应用

**目的:** 字符和自定义字符在标准 LCD 显示器上显示。

**原理:** 利用 BASCOM-AVR 的专用语句和特殊点阵设计工具在 LCD 液晶显示器上显示字符和特殊字符。

标准 LCD 液晶显示器简介:

点阵 LCD 组件由具有高反差、宽视角液晶显示屏和 CMOS 控制驱动器组成。为与 MCU 接口容易，显示器上提供了字符发生器和显示数据 RAM。所有显示功能均由指令控制。

**特点:**

结构紧凑

低功耗

数据线 4 位/8 位可选择

96 个 ASCII 字符代码 + 92 个特殊字母

内装字符发生器和显示数据 RAM

引脚排列(见第三章 开发工具的 3.3 节)

## 2. 引脚功能

脚号	符号	功 能
1	Vss	电源负端，接地（或接 -5V）
2	Vdd	电源正端，接+5V
3	Vo	LCD 反差调整
4	RS	寄存器选择 RS=0，选指令寄存器 RS=1，选数据寄存器
5	R/W	读/写选择 R/W=0，写数据至 LCD R/W=1，从 LCD 读数据
6	E	输入允许
7	DB0	数据总线 4 位数据总线 DB4-DB7 8 位数据总线 DB0-DB7
8	DB1	
9	DB2	
10	DB3	
11	DB4	
12	DB5	
13	DB6	
14	DB7	

原理图：(图 8.4.1)

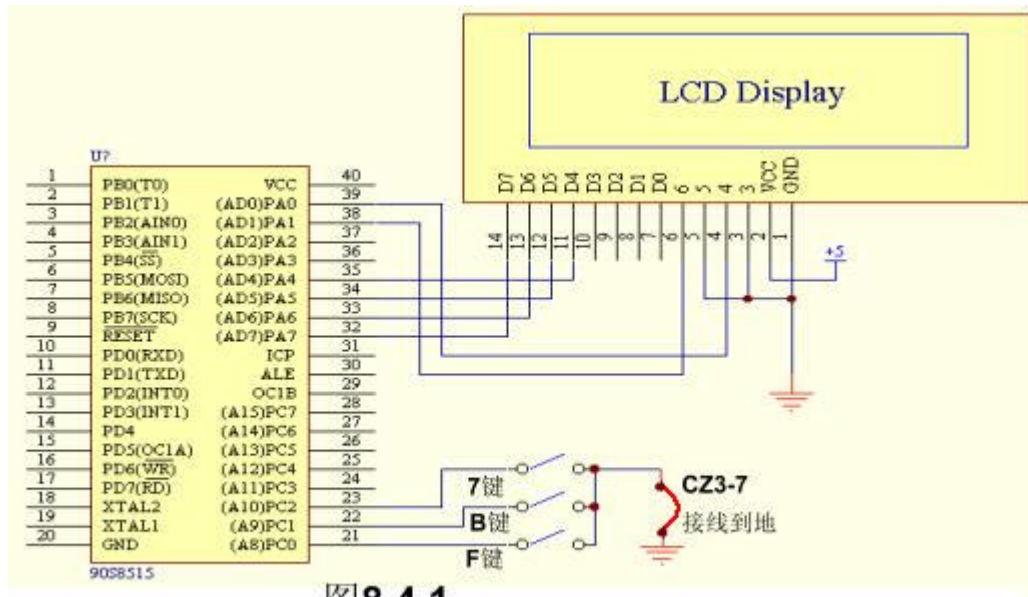


图 8.4.1

程序:exp4.bas

```

$Sim                                     \此语句软件仿真时使用，正式下载时必须取消
'接 LCD 显示器的 I/O 引脚定义
Config Lcdpin = Pin , Db4 = Porta.4 , Db5 = Porta.5 , Db6 = Porta.6 ,
              Db7 = Porta.7 , E = Porta.1 , Rs = Porta.0
Dim A As Byte                             \定义字节变量 A
Config Lcd = 16 * 2                         \定义使用 16*2 行的 LCD 显示器
Deflcdchar 0 , 32 , 31 , 14 , 4 , 4 , 14 , 31 , 32      \定义 0 号特殊字符
Deflcdchar 1 , 31 , 32 , 17 , 27 , 27 , 17 , 32 , 31   \定义 1 号特殊字符

Cls                                         \清屏 LCD
Lcd "Hello world."                         \在第一行显示 "Hello world."
Waitms 250
Lowerline                                  \选择第二行
Lcd Chr(0) ; Chr(1) ; "!@#$$%^&*" ; Chr(1) ; Chr(0)  \在第二行显示自定义字符和特殊
字符
Waitms 250
For A = 1 To 16
  ShiftLcd Right                            \显示内容右移 16 个位子
  Waitms 250
Next
For A = 1 To 28
  ShiftLcd Left                             \显示内容左移 28 个位子
  Waitms 250
Next
For A = 1 To 12
  ShiftLcd Right                            \显示内容右移 12 个位子回原位

```

Waitms 250

Next

End

注：正式下载前可先使用软件仿真(见图 8.2.5)，观察显示效果。

#### 8.4.2 简单游戏机-按钮猜数

目的：设计制作简单的按钮猜数游戏机。

原理：按钮猜数在电脑摇奖,电脑选出幸运号中经常使用。游戏开始连续不停产生 3 组 0-9 之间的随机数，并在 LCD 显示器显示。搬动相应的开关可使对应位停止产生随机数，当 3 组数停止后，3 个数一致或 2 个数一致便给出得奖提示。

原理图：图 6.4.1

程序：Exp5.bas

```
Config Lcdpin = Pin , Db4 = Porta.4 , Db5 = Porta.5 , Db6 = Porta.6 , Db7 = Porta.7 ,
E = Porta.1 , Rs = Porta.0
```

```
Config Lcd = 16 * 2 '定义 LCD 接口
```

```
Config Pinc.0 = Input , Pinc.1 = Input , Pinc.2 = Input ' 定 义
Portc.0,Portc.1,Portc.2 为输入控制端
```

```
Portc.0 = 1 : Portc.1 = 1 : Portc.2 = 1 '上拉电阻有效，为"111"
```

```
Dim A As Byte , B As Byte , C As Byte , I As Byte
```

```
Cursor Off Noblink '消隐 LCD 光标
```

```
Cls '清 LCD 输出显示
```

```
Do
```

```
    I = Pinc And &H07 '读按键值
```

```
    Waitms 50
```

```
    Select Case I
```

```
        Case 7:
```

```
            A = Rnd(90) / 10 : B = Rnd(90) / 10 : C = Rnd(90) / 10
```

```
            Locate 1 , 6 : Lcd A
```

```
'3 个按键均未按下，连续显示
```

3 组随机数

```
            Locate 1 , 8 : Lcd B
```

```
            Locate 1 , 10 : Lcd C
```

```
        Case 6:
```

```
            A = Rnd(90) / 10 : B = Rnd(90) / 10
```

```
            Locate 1 , 6 : Lcd A
```

```
            Locate 1 , 8 : Lcd B
```

```
        Case 5:
```

```
            A = Rnd(90) / 10 : C = Rnd(90) / 10
```

```
            Locate 1 , 6 : Lcd A
```

```
            Locate 1 , 10 : Lcd C
```

```
        Case 4:
```

```
            A = Rnd(90) / 10
```

```
            Locate 1 , 6 : Lcd A
```

```
Case 3:
  C = Rnd(90) / 10 : B = Rnd(90) / 10
  Locate 1 , 10 : Lcd C
  Locate 1 , 8 : Lcd B
Case 2:
  B = Rnd(90) / 10
  Locate 1 , 8 : Lcd B
Case 1:
  C = Rnd(90) / 10
  Locate 1 , 10 : Lcd C
Case 0:
  While I = 0
    Locate 2 , 1
    If A = B And B = C Then
      Lcd "You Win $10000!"
    ElseIf A = B Or B = C Or A = C Then
      Lcd "You Win $100!"
    Else
      Lcd "You Lose $10!"
    End If
    I = Pinc And &H07
  Wend
  Lowerline
  Lcd "          "
End Select
Loop
End
```

'3 个按键均按下  
'判断输赢

8.5 串口通信 UART

大部分的 AVR 芯片都含有一个硬件串行通信接口 UART，利用该接口可以实现控制系统于 PC 的通信，构成 RS-232、RS-485 或 RS-422 的网络。

8.5.1 AVR 系统与 PC 的简易通信

**目的：**实现 AVR 系统与 PC 之间的串口连网和通信。

**原理：**利用 AVR 芯片的 URAT 建立与 PC 机 COM 口的简易 RS-232 连网通信，AVR 系统输出字符在 PC 机的屏幕上显示。

**原理图：**图 8.5.1

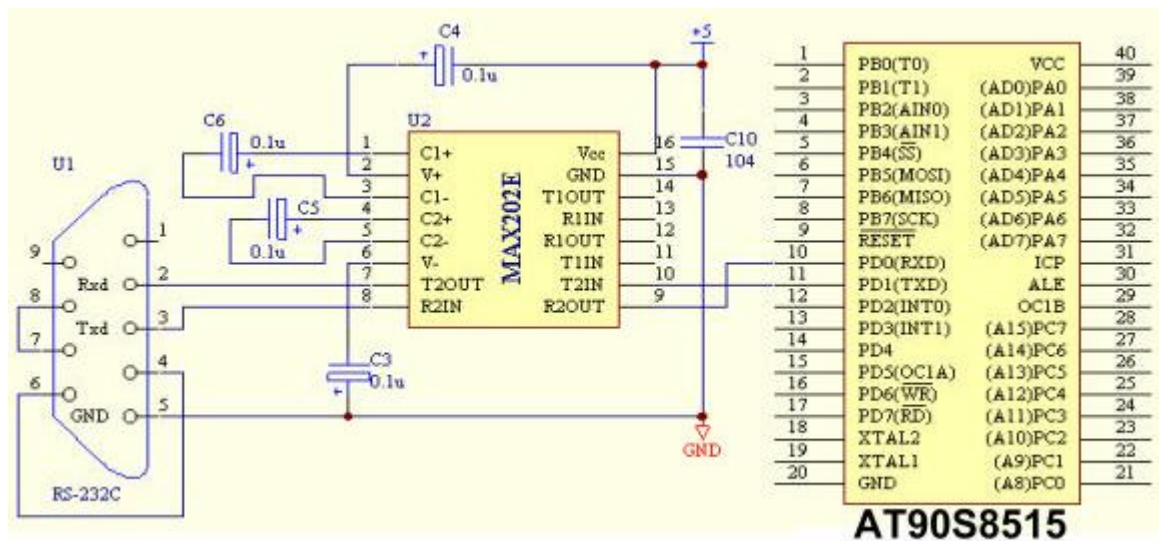


图 8.5.1

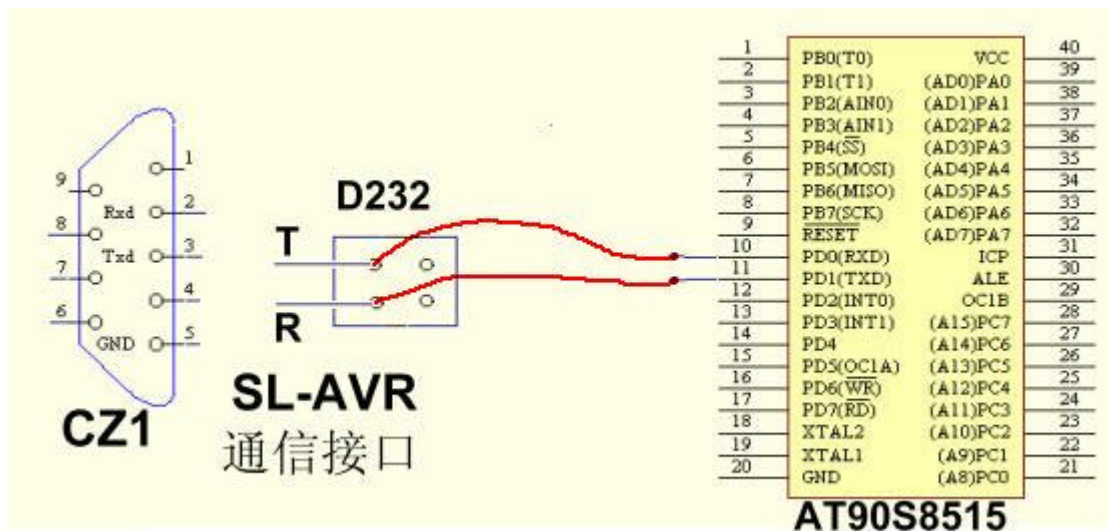


图 8.5.1B

图中 MAX202 为 TTL 电平与 RS-232 电平转换电路。AVR 系统的接口电平为 5v 的 TTL 电平，必须通过一个电平转换电路才能和 PC 机的串行通信 COM 口连接，否则将损坏 PC 机的 COM 口和 AVR



芯片！

SL-AVR 开发实验器上已有 RS232 的电平转换电路，用户可利用系统所提供的 RS-232 连接电缆和 SL-AVR 开发实验器上 D232 短路块，通过 D232 的 R,T 接线端用两根跳线实现图 8.5.1B 的电路。

程序：EXP6.bas

```
$regfile = "8515def.dat"
```

```
$crystal = 4000000           \晶振频率 4M
```

```
$baud = 9600                 \波特率 9600
```

```
\RS-232 缺省设置:8 位数据位, 1 位停止位, 无校验
```

```
Dim A As Byte
```

```
A = 0
```

```
Do
```

```
  Print "This is " ; A ; " Line"   \Print 的内容将在 PC 的屏幕上显示
```

```
  A = A + 1
```

```
  Waitms 250
```

```
Loop
```

```
End
```

#### 实验方法：

实验程序编译下载到 AT90s8515 后，关闭 SL-AVR 开发实验器电源。将 SL-AVR 开发实验器上 D232 短路块断开。使用连线将参照图 8.5.1B 正确连接。

关闭 PC 上运行的 avr 程序下载软件，打开 BASCOM-AVR 的终端仿真功能或 Windows 的超级终端。设置 PC 机的终端仿真口使用的 COM 口号 (COM1 或 COM2) 及相关通信参数：波特率 9600、8 位数据位、1 位停止位、无校验、无数据流控制。(有关 RS-232 的通信原理和方式请参阅其它有关资料)。

打开系统电源，在 PC 机的屏幕上将显示 AVR 系统输出的字符。

#### 8.5.2 PC 控制的简易广告灯

目的：实现 PC 机控制的简易广告灯。

原理：由 AVR 系统构成简易广告灯(参见 8.3.2 节)，广告灯的显示方式通过 PC 机控制。

PC 键盘字符	8 路发光 LED 闪烁方式
"0"	循环左移
"1"	循环右移
"2"	明暗交替
其它字符	随机闪烁

原理图：图 8.5.2

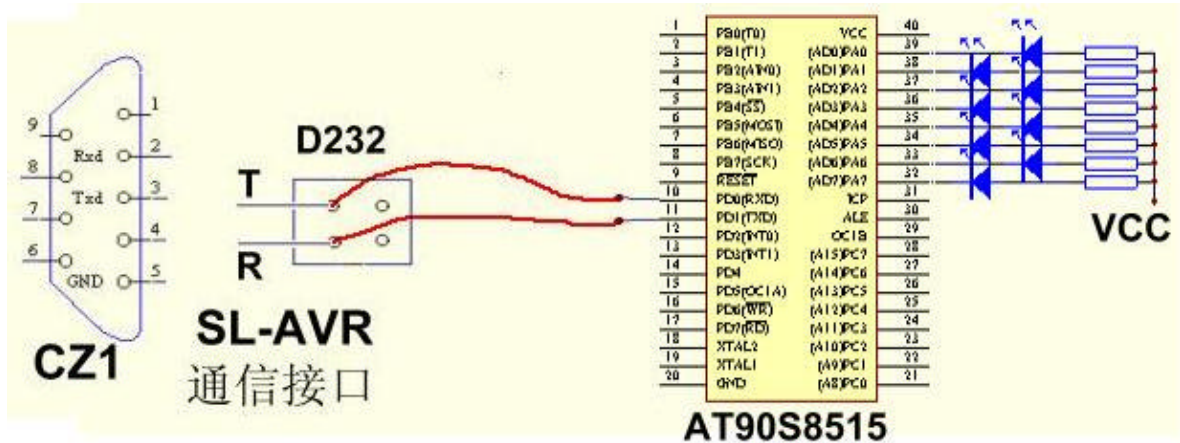


图 8.5.2

```

程序: EXP7.bas
$regfile = "8515def.dat"
$crystal = 4000000           \晶振频率 4M
$baud = 9600                \波特率 9600
                             \RS-232 缺省设置:8 位数据位, 1 位停止位, 无校验
Dim A As Byte , B As Byte   \定义变量 A, B 为字节型
Config Porta = Output       \定义 Porta 口为输出, 控制 LED

B = "0"                      \B 初始值为字符 "0"
Do
  Select Case B
    Case "0":                 \PC 键盘输入字符为: "0"
      A = &H01
      Do
        Porta = A
        Wait 1
        Rotate A , Left , 1   \左移循环
        B = Inkey()           \读 PC 的控制命令
      Loop Until B <> 0       \PC 键盘无输入, 继续左移循环
    Case "1":                 \PC 键盘输入字符为: "1"
      A = &H80
      Do
        Porta = A
        Wait 1
        Rotate A , Right , 1  \右移循环
        B = Inkey()
      Loop Until B <> 0
    Case "2":                 \PC 键盘输入字符为: "2"
      A = &H00
      Do

```

```
    Porta = A
    Wait 1
    A = Not A           `明暗交替
    B = Inkey()
    Loop Until B <> 0
Case Else             `PC 键盘输入其它字符
    Do
        Porta = Rnd(255) `产生随机数,随机闪烁
        Wait 1
        B = Inkey()
        Loop Until B <> 0
    End Select
Loop
End
```

#### 实验方法:

本实验过程同 8.5.1 节。照上节实验方法 1 至 5 步后，按下 PC 机键盘的字符“0”、“1”、“2”及其它字符，按下的字符通过串口传送到 AVR 系统，AVR 系统将根据接收到的字符控制发光二极管的闪烁方式。

更多的实验程序见<<双龙电子书光盘>>

## 8.6 单总线(1-Wire)接口和温度计

“单总线”接口是美国 DALLAS 公司的特有技术(<http://www.dalsemi.com>),它实现了在一条数据线上进行双向数据传输,最大限度的节省了通讯线的数量。使系统布线更方便,成本更低,适合于多点分散系统。同时,DALLAS 公司推出了丰富的单总线产品,如数字温度传感器、电子纽扣(Ibutton)等。这种单总线芯片仅有一根信号线和一根地线。在信号线上综合了双向数字信号线和电源线的功能,每一块芯片都可提供一个独一无二的登记号,使用户可以灵活的构成不同功能的系统。而且,单总线产品基本为单芯片,成本低,可靠性高,例如:DS-1820,集传感器、温度转换、联网通讯于一体。

由于单总线仅使用一根信号线进行双向数据传输,因此单总线的通信协议就比其它的串行通信协议要复杂的多,通信时序和时间定时要求非常严格。在一般的电子产品开发中往往要花费大量的精力和时间编写和调试单总线接口的底层及通信程序。BASCOM-AVR 提供了简便的 1Wire 语句,可以用于方便和快速地开发单总线产品。

**目的:** 用 DS-1820 和 AVR 系统构成 0.1 度精度的温度计。

**原理:** 美国 Dallas 公司生产的单总线数字温度传感器 DS1820 可把温度信号直接转换成串行数字信号供微机处理。由于每片 DS1820 含有唯一的硅串行数,所以在一条总线上可挂接任意多个 DS1820。从 DS1820 读出的信息或写入 DS1820 的信息,仅需要一根数据线(单总线接口)。读写及温度变换功率来源于数据总线,总线本身也可以向所挂接的 DS1820 供电,而无需额外电源。DS1820 提供九位温度读数,构成多点温度检测系统而无需任何外围硬件。本实验采用一片 DS1820 与 AVR 单片微控制器和 LCD 显示器组成一个温度检测系统。

### 1. DS1820 的特性

单线接口: 仅需一根数据线与 MCU 连接无需外围元件可由总线提供电源;

测温范围为  $-55^{\circ}\text{C} \sim 75^{\circ}\text{C}$ , 精度为达  $0.1^{\circ}\text{C}$ ;

内置 A/D 变换, 转换时间为 200ms; 九位温度读数,

用户自设定温度报警上下限, 其值是非易失性的;

### 2. DS1820 引脚及功能

DS1820 引脚见图 8.6.1(PR35 封装)。

GND: 地

DQ: 数据输入/输出脚(单总线接口, 可作寄生供电)

VDD: 电源电压。

### 3. DS1820 操作指令及时序

参见 Dallas 公司 DS1820 数据手册。

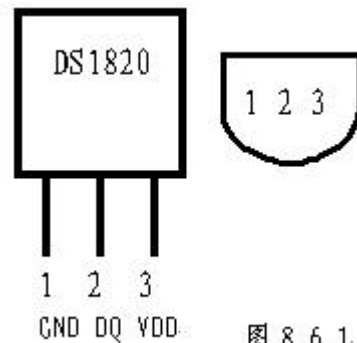


图 8.6.1

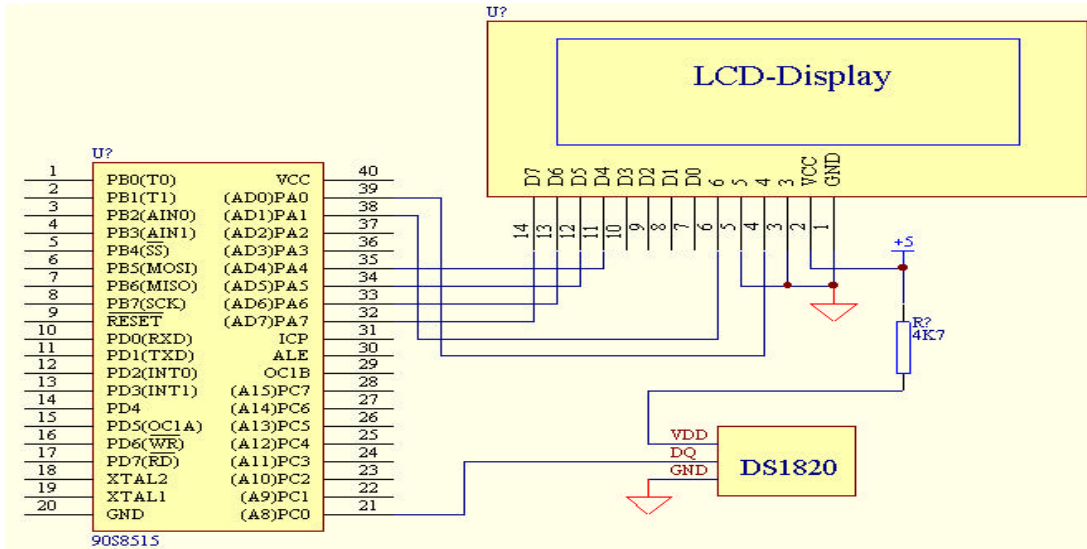


图 8.6.2

原理图：图 8.6.2

程序：Exp8.bas

\$regfile = "8515def.dat"

\$crystal = 4000000

Deflcdchar 0 , 14 , 10 , 14 , 32 , 32 , 32 , 32 , 32      ‘定义度’ 的显示字符

Deflcdchar 1 , 4 , 10 , 17 , 4 , 31 , 2 , 4 , 8      ‘定义汉字’ 今’ 的显示字符

Deflcdchar 2 , 31 , 4 , 31 , 4 , 12 , 10 , 18 , 17      ‘定义汉字’ 天’ 的显示字符

Dim I As Byte , Tmp As Byte , Crc As Byte

Dim T As Integer , T1 As Integer

Dim Data1(9) As Byte

Config 1wire = Portc.0      ‘Portc.0 接 DS1820 的数据线

Config

Lcdpin=Pin,Db4=Porta.4,Db5=Porta.5,Db6=Porta.6,Db7=Porta.7,E=Porta.0,Rs=Porta.1

Config Lcd = 16 \* 2      ‘定义 LCD 接口

Cursor Off Noblink      ‘清 LCD 显示，无光标

Cls

Locate 1 , 1 : Lcd "Demo for DS1820"      ‘第一行显示提示字符串

Locate 2 , 1 : Lcd Chr(1) ; Chr(2)      ‘第二行 1, 2 列显示汉字’ 今天’

Locate 2 , 8 : Lcd Chr(0) ; "C"      ‘第二行 8 列显示字符’ ° C’

Do

    1wwrite &HCC : 1wwrite &H44      ‘写入 DS1820 命令，启动温度转换

    Waitms 255

    Waitms 255      ‘等待转换结束

    1wreset      ‘DS1820 初始化

    1wwrite &HCC      ‘写入 DS1820 命令，跳过 ROM 操作

    1wwrite &HBE      ‘写入 DS1820 命令，读温度值

    Data1(1) = 1wread(9)      ‘读 DS1820 温度，共 9 个字节

    1wreset      ‘DS1820 初始化

```

If Err = 1 Then
  Locate 2 , 3 : Lcd "-----"          ‘无 DS1820 显示” -----”
Else
  Crc = 0
  For I = 1 To 9                          ‘进行 CRC 校验
    Tmp = Crc Xor Data1(i)
    Crc = Lookup(tmp , Crc8)
  Next I
  If Crc = 0 Then                          ‘CRC 校验正确, 进行温度换算
    Tmp = Data1(1) And 1
    If Tmp = 1 Then Decr Data1(1)
    T = Makeint(data1(1) , Data1(2))
    T = T * 50 : T = T - 25
    T1 = Data1(8) - Data1(7) : T1 = T1 * 100
    T1 = T1 / Data1(8) : T = T + T1 : T = T / 10
  End If
  If Crc = 0 Then                          ‘CRC 校验正确, 温度显示
    If T < 0 Then
      T = Abs(t)
      Locate 2 , 3 : Lcd "-"            ‘负温度显示” -” 号
    Else
      Locate 2 , 3 : Lcd " "          ‘正温度显示” “空格
    End If
    T1 = T Mod 10 : T = T / 10          ‘以下显示温度, 精度 0.1 度
    If T < 10 Then
      Locate 2 , 4 : Lcd " " ; T ; "." ; T1
    Else
      If T > 99 Then
        Locate 2 , 3 : Lcd T ; "." ; T1
      Else
        Locate 2 , 4 : Lcd T ; "." ; T1
      End If
    End If
  End If
  Else
    Locate 2 , 3 : Lcd "****.*"        ‘CRC 校验错, 显示错误标志” ****.*”
  End If
End If
Loop
End

Crc8:                                     ‘CRC 计算用的表格
Data 0 , 94 , 188 , 226 , 97 , 63 , 221 , 131 , 194 , 156
Data 126 , 32 , 163 , 253 , 31 , 65 , 157 , 195 , 33 , 127
Data 252 , 162 , 64 , 30 , 95 , 1 , 227 , 189 , 62 , 96
Data 130 , 220 , 35 , 125 , 159 , 193 , 66 , 28 , 254 , 160

```

Data 225 , 191 , 93 , 3 , 128 , 222 , 60 , 98 , 190 , 224  
Data 2 , 92 , 223 , 129 , 99 , 61 , 124 , 34 , 192 , 158  
Data 29 , 67 , 161 , 255 , 70 , 24 , 250 , 164 , 39 , 121  
Data 155 , 197 , 132 , 218 , 56 , 102 , 229 , 187 , 89 , 7  
Data 219 , 133 , 103 , 57 , 186 , 228 , 6 , 88 , 25 , 71  
Data 165 , 251 , 120 , 38 , 196 , 154 , 101 , 59 , 217 , 135  
Data 4 , 90 , 184 , 230 , 167 , 249 , 27 , 69 , 198 , 152  
Data 122 , 36 , 248 , 166 , 68 , 26 , 153 , 199 , 37 , 123  
Data 58 , 100 , 134 , 216 , 91 , 5 , 231 , 185 , 140 , 210  
Data 48 , 110 , 237 , 179 , 81 , 15 , 78 , 16 , 242 , 172  
Data 47 , 113 , 147 , 205 , 17 , 79 , 173 , 243 , 112 , 46  
Data 204 , 146 , 211 , 141 , 111 , 49 , 178 , 236 , 14 , 80  
Data 175 , 241 , 19 , 77 , 206 , 144 , 114 , 44 , 109 , 51  
Data 209 , 143 , 12 , 82 , 176 , 238 , 50 , 108 , 142 , 208  
Data 83 , 13 , 239 , 177 , 240 , 174 , 76 , 18 , 145 , 207  
Data 45 , 115 , 202 , 148 , 118 , 40 , 171 , 245 , 23 , 73  
Data 8 , 86 , 180 , 234 , 105 , 55 , 213 , 139 , 87 , 9  
Data 235 , 181 , 54 , 104 , 138 , 212 , 149 , 203 , 41 , 119  
Data 244 , 170 , 72 , 22 , 233 , 183 , 85 , 11 , 136 , 214  
Data 52 , 106 , 43 , 117 , 151 , 201 , 74 , 20 , 246 , 168  
Data 116 , 42 , 200 , 150 , 21 , 75 , 169 , 247 , 182 , 232  
Data 10 , 84 , 215 , 137 , 107 , 53

## 8.7 I2C 总线接口和简易 IC 卡读写器

I2C 总线是 PHILIPS 公司推出的串行总线，是各种总线中使用信号线较少，并且具有自动寻址，多主机时钟同步和仲裁等功能很强的总线。因此，许多接口芯片如：RAM，LCD 驱动，时钟，A/D，D/A 都采用 I2C 接口。IC 卡在当今社会的各个行业有着广泛的应用，而且大多数的 IC 卡的接口都采用 I2C 总线。

**目的：**用 AVR 系统构成简易 IC 卡读写器。

**原理：**

I2C 串行总线使用两根信号线：一根双向的数据线 SDA，另一根是时钟线 SCL。所有连接到 I2C 总线上的设备的串行数据都接到总线的 SDA 线，各设备的时钟线 SCL 接到总线的 SCL。I2C 数据总线传输时序如图 8.7.1，关于 I2C 的详细内容请参考有关的书籍和资料。

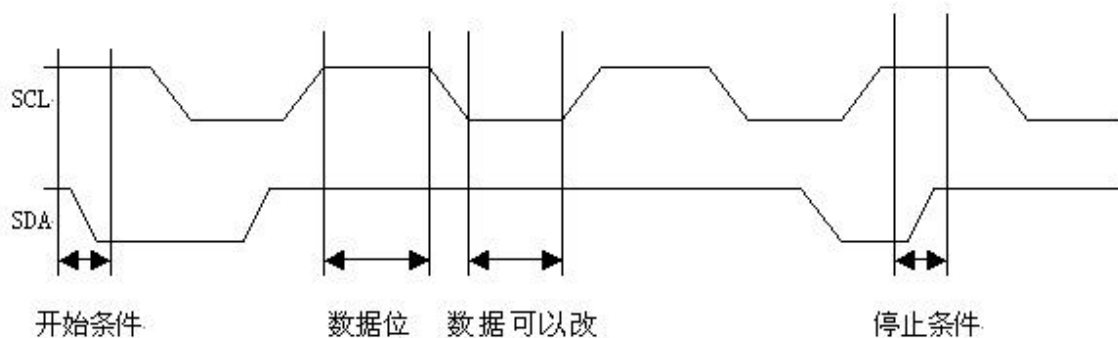


图 8.7.1

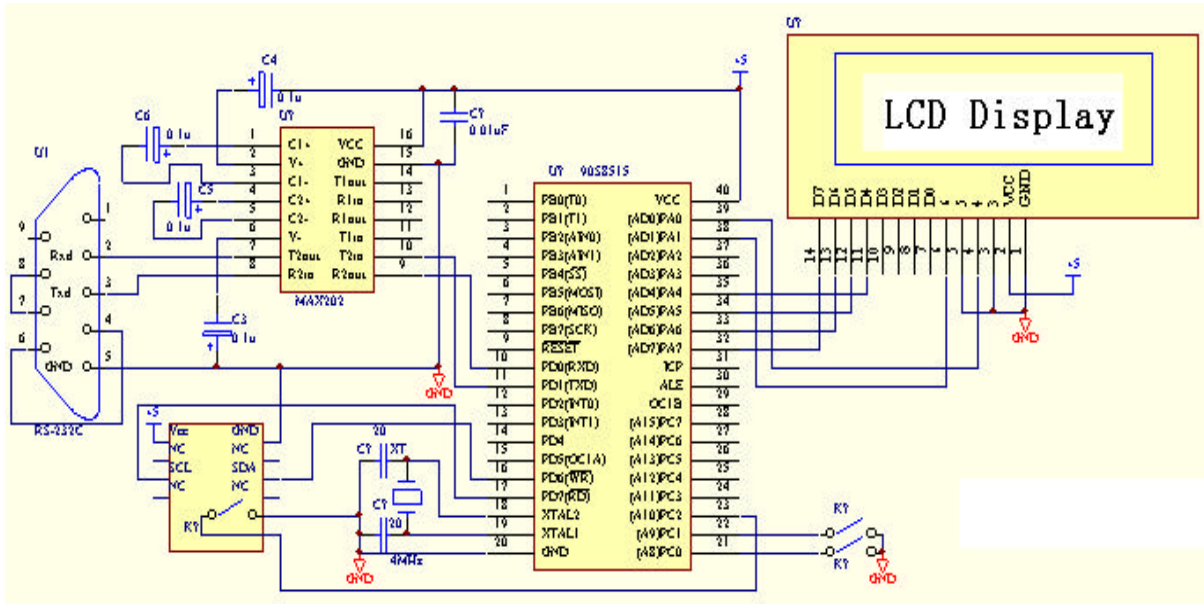
AVR 系列的单片微控制器没有专用的 I2C 总线硬件接口，因此需要用两根 I/O 线来模拟实现 I2C 总线的功能。这就要编写相应的 I2C 底层程序模拟 I2C 数据总线的传输时序，实现 I2C 总线启动、停止、读、写及应答的功能。BASCOM-AVR 提供了专用的 I2C 语句，利用它们就能非常方便地设计和实现 I2C 总线的读写。

实验中采用的 IC 卡为 ATMEL 公司的 AT24C01A/2/4/8/16，它是一种通用的读写存储卡。该类 IC 卡上的芯片就是采用 I2C 总线接口的串行 CMOS EEPROM。关于 AT24CXX 系列芯片的详细资料可访问 <http://www.atmel.com> 网站。

AVR 系列的单片微控制器内部还提供了一定容量的 EEPROM，可以用于保存系统的初始数据，设定值或密码口令字等，这一性能为开发产品提供了好多的方便，它不仅可使系统设计节省硬件（EEPROM 芯片）和连线，而且还减小了空间，提高了系统的可靠性和保密性。本设计中，使用了 AVR 片内的 EEPROM 来保存密码，并同 IC 卡上的密码核对，判别 IC 卡的非法性。

本设计将 IC 卡的读、写、判别结合在一起，用户使用 PC 的键盘输入 8 位自定密码，通过 RS-232 送到 IC 卡读写器，将密码写入用户的 IC 卡中（也可同时写入 AVR 的 EEPROM 中作为系统密码）。IC 卡读写器还可读取 IC 卡上的密码，并同系统密码核对，用于判别 IC 的非法性。IC 卡读写器采用 LCD 液晶显示器，用于显示系统的状态和 IC 卡上的密码等。





原理图：见图 8.7.2

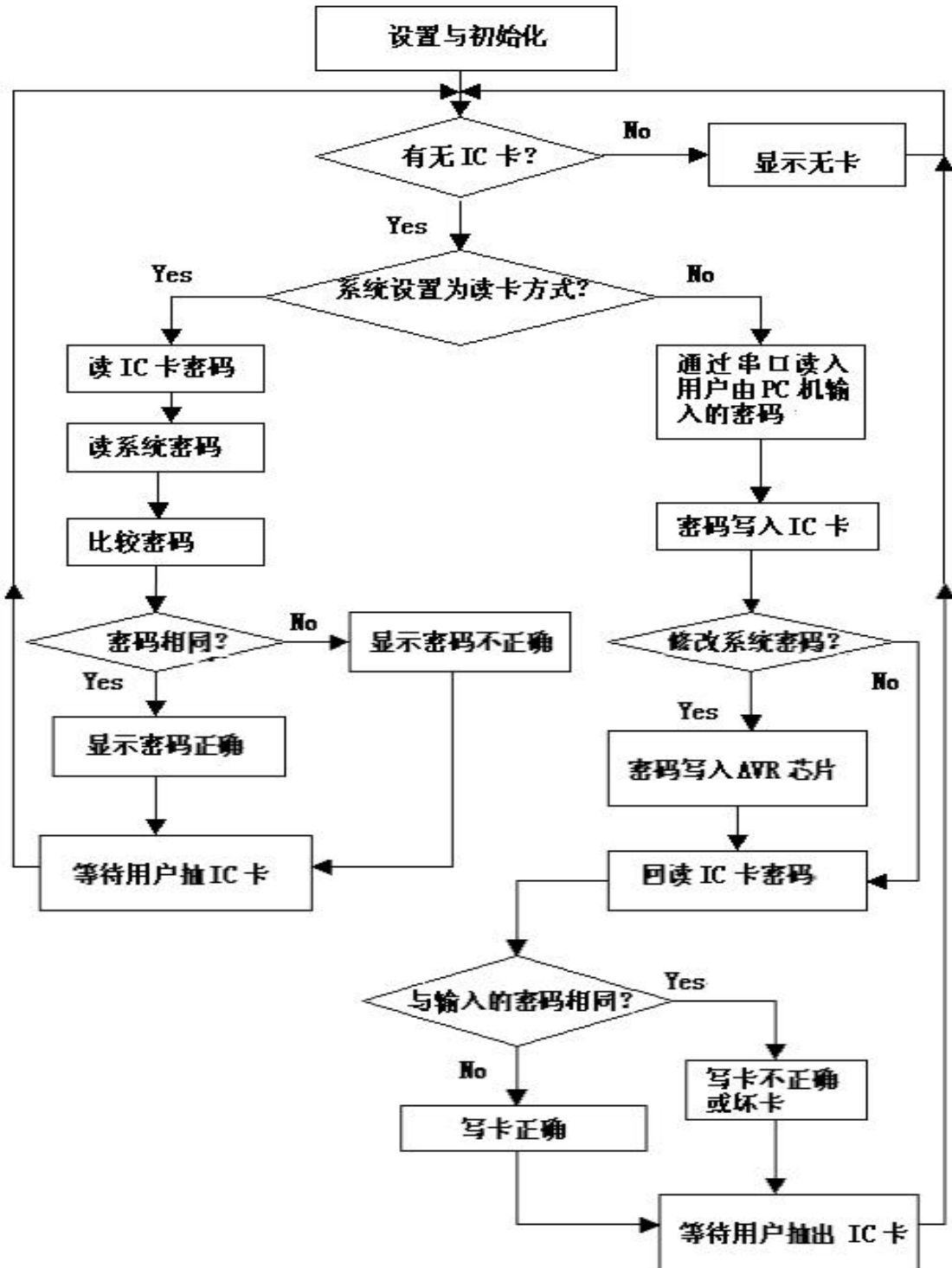


图 8.7.3 流程图

程序: Exp9.bas

```
$regfile = "8515def.dat"
```

```
$crystal = 4000000
```

```
$baud = 9600
```

```
Dim I As Byte , Temp As Byte
```

```

Dim Data1(8) As Byte , Data2(8) As Byte
Dim Card_ok As Bit

Config
Lcdpin=Pin,Db4=Porta.4,Db5=Porta.5,Db6=Porta.6,Db7=Porta.7,E=Porta.0,Rs=Porta.1
Config Lcd = 16 * 2          '定义 LCD 显示屏接口

Config Scl = Portd.7        '定义 Portd.7 为 I2C 总线的 Scl
Config Sda = Portd.6        '定义 Portd.6 为 I2C 总线的 Sda
Const Adresw = &HA0         '定义 IC 卡的写地址指令字
Const Adresr = &HA1         '定义 IC 卡的读地址指令字

Config Pinc.0 = Input , Pinc.1 = Input , Pinc.2 = Input
Portc.0 = 1 : Portc.1 = 1 : Portc.2 = 1      '定义 3 个 I/O 口为输入开关接口

Cursor Off Noblink        '清 LCD 显示

Do
  Cls
  Locate 1 , 1 : Lcd "Demo for IC_Card"      '显示提示字符
  If Pinc.2 = 1 Then                          '检测有无 IC 卡插入
    Locate 2 , 1 : Lcd "No IC_Card"         " '无 IC 卡插入
    Wait 1
  Else                                         '有 IC 卡插入
    If Pinc.0 = 0 Then                          '检测系统设置为写卡方式?
      Cls
      Locate 1 , 1 : Lcd "Enter Password: "   '写卡方式, 要求输入密码字
      I = 1
      Locate 2 , 1
      Do
        Temp = Inkey()                        '由 RS-232 口接收 PC 输入
        If Temp <> 0 Then                      '的密码字符,
          Data1(i) = Temp : I = I + 1        '长度为 8 个
          Lcd Chr(temp)                       '同时在 LCD 上显示密码
          Print Chr(temp);
        End If
      Loop Until I > 8
      Print
      For I = 1 To 8                            '将 8 个密码字写入 IC 卡中
        I2cstart                               '写入地址为 1-8
        I2cwbyte Adresw
        I2cwbyte I
        I2cwbyte Data1(i)
        I2cstop
    
```

```

        Waitms 10
    Next

    If Pinc.1 = 0 Then
        For I = 1 To 8
            Writeeprom Data1(i) , I
        Next
    End If

    For I = 1 To 8
        I2cstart
        I2cwbyte Adresw
        I2cwbyte I
        I2cstart
        I2cwbyte Adresr
        I2crbyte Data2(i) , Nack
        I2cstop
    Next

    Card_ok = 0
    For I = 1 To 8
        If Data1(i) <> Data2(i) Then
            Card_ok = 1
            Exit For
        End If
    Next

    Locate 2 , 1
    If Card_ok = 0 Then
        Lcd "Write Card ok! "
    Else
        Lcd "No or Bad Card!?"
    End If
    Do
        Loop Until Pinc.2 = 1
    Else
        CIs : Locate 1 , 1
    For I = 1 To 8
        I2cstart
        I2cwbyte Adresw
        I2cwbyte I
        I2cstart
        I2cwbyte Adresr
        I2crbyte Data1(i) , Nack

```

```
        I2cstop                                ‘读 IC 卡上的密码字 8 个
        Lcd Chr(data1(i))                      ‘显示在 LCD 上
    Next
    For I = 1 To 8
        Readeeprom Data2(i) , I                ‘读系统密码字 8 个
    Next

    Card_ok = 0
    For I = 1 To 8                              ‘将系统密码和 IC 卡上的密码
        If Data1(i) <> Data2(i) Then           ‘比较
            Card_ok = 1
            Exit For
        End If
    Next
    Locate 2 , 1
    If Card_ok = 1 Then                          ‘密码不符
        Lcd "Password not ok!"
    Else
        Lcd "Password Ok!      "              ‘密码相符
    End If
    Do
    Loop Until Pinc.2 = 1                        ‘等待用户将 IC 抽出
End If
End If
Loop
End                                           ‘返回循环
```