

# 51C 程序设计中应注意的问题

张景元 淄博大学计算机科学与工程系 (255000)

## Abstract

This paper studied C language compilation implement, it indicated several critical problem other than commonly C language, and must advert go to memory area, and accessing method of special, and define of parallel interface, and define bit variable. It besides programmed to coalescent example of expansion keyboard /display with the 8279 chip.

Keywords: monolithic processor, 51C compilation implement, 8279 chip, keyboard /display

## 摘 要

本文对单片机的 C 语言编译器进行了研究分析, 指出了与一般 C 语言的区别及其在软件开发过程中须注意的存储区的定位及访问、专用寄存器的访问方法、并行接口的定义、位变量的定义等几个关键性的问题。并且结合 8279 扩展键盘 / 显示器实例进行了编程。

关键词: 单片机, 51C 编译器, 8279 芯片, 键盘 / 显示器

## 1 引言

随着微电子技术、计算技术、通信技术的不断发展, 计算机在工业控制中的应用已越来越广泛。单片机以体积小、重量轻、灵活性强、价格低等优点, 被广泛应用于智能仪表、电气设备、家用电器等的研制开发, 其中又以 8 位机更为普遍。在 80 年代中期, Intel 公司将 8051 内核使用权以专利互换或出售形式转给世界许多著名的 IC 制造厂商 (如: PHILIPS、西门子、AMD、OKI、NEC、Atmel 等), 使得 8051 成为具有众多厂商支持的、发展出上百个品种的大家族, Intel 推出的 80C251 也是与 8051 在机器代码级兼容的, 保证了 51 系列用户在 21 世纪的技术领先性。另外,

很多厂商为 8051 开发了大量的配套接口芯片, 极大地方便了用户, 因此, 尽管 16 位机开始流行, 但 51 系列的 8 位机在将来很长一段时期内还是研发控制设备的主流机。随着硬件的发展, 8051 软件工具已有 C 编译器及实时多任务操作系统 RTOS (Real-Time Operating System) 在 RTOS 支持下, 程序设计调试更容易、更可靠、实时性更强, 并且缩短了研发周期。“51C”虽然具有常规 C 工具的共性, 但毕竟不是 ANSIC、Turbo C, 它有其自身的特点, 用户必须十分清楚它的一些特殊规定, 才能在研发过程中避免错误。本文将根据开发中的体会, 就须注意的有关问题进行介绍。

这样的数据结构简单明了, 便于编程和存贮。

为了提高运行速度, 部分程序采用宏汇编语言编写, 包括: 发送 / 接收数据、在屏幕上画点及产生光条等。整个应用软件界面采用下拉式菜单, 汉字显示, 并有求助功能, 使用方便。一台规划级计算机可同时规划多台控制级计算机, 便于集中管理。

## 5 工程应用

本文所述的双坐标运动轨迹控制系统已成功应用于某生产厂的绣花机产品中。它可以单机运行, 也可以双机运行或多机运行。可以用固定的花案

(EPROM) 控制绣花, 也可由规划级计算机设计图案后下送给控制级计算机来控制绣花。通过多机通讯, 不同绣花机可绣不同的花案, 使用方便。实际运行表明, 该系统功能强, 运行可靠, 颇受用户欢迎。

## 参考文献

1. 袁南儿, 计算机新型控制策略及其应用, 清华大学出版社, 1998
2. 袁南儿, SCM-STD 分布式计算机控制系统, 电子与自动化, 1992, 5

## 2 51C 程序结构及其编译器

C 语言是源于编写 UNIX 操作系统的一种语言,是一种结构化程序设计语言,产生的代码紧凑,可以深入到机器内部编程,具备若干汇编语言所具有的特点,又优于汇编语言。C 程序本身并不依赖于机器的硬件系统,基本上不作修改就可以根据单片机的不同较快地移植过来,因此,早在 85 年就开始出现 8051 单片机的 C 语言编译器了,将 C 语言代码编译成 51 单片机的机器代码。常见的产品有 American Automation、Archimedes、Avocet、Bso Tasking、Franklin、Intermetrics、MCC、Dunfields 等,它们各具特点,但用的比较多的还是 Archimedes 和 Franklin。Franklin 产生的代码紧凑,使用也方便;Archimedes 的性能完善,资料完善。51C 程序结构与一般 C 语言程序结构并没有本质上的差别,也是只有一个主函数和若干函数组成。在 51C 的头文件中有其库函数的说明,可以通过 include 预处理指令将头文件包含在自己的文件中即可调用。每个函数的定义及调用规则基本上与一般的 C 一样,实际上 51C 程序的设计过程也就是一系列函数的定义过程,熟悉 C 语言程序设计的人很容易掌握。

### 3 编程过程中需注意的问题

尽管 51C 与一般 C 程序设计方法基本相同,但毕竟是针对单片机及其硬件系统所进行的程序设计,与常规意义上的 C 编程有所区别。在进行程序设计时,必需注意以下几个问题。

#### 2.1 存储区的定位及访问

首先必须了解编译器的数据类型以及与 51 单片机存储器结构的对应关系,编译器定义的任何数据类型必须以一定的存储类型方式定位在单片机的某一存储区中,否则没有任何实际意义。比如, Franklin51C 编译器所支持的存储类型中, data 类型可以直接寻址片内数据存储区,访问速度也比较快; Bdata 类型能够对片内数据存储区进行位寻址,并且允许位与字节混合访问; idata 类型就只能间接寻址片内数据存储区,它可以访问片内全部 RAM 地址空间;而在使用 Code 存储类型时,编译器就会自动将其定义在代码空间(ROM 或 EPROM)中。

#### 2.2 专用寄存器的访问方法

其次是片内特殊功能寄存器 SFR 的访问,编译器都提供了专门的方法,这些方法与一般 C 语言是不兼容的,只适用于单片机 C 编程。比如, Franklin51C 编译器是通过引入关键字“sfr”来实现

的,其语法结构为: sfr SFR\_name ' = ' int constant ;',例如: sfr SCON = 0x80,就定义了串口控制寄存器的地址为 80H。

#### 2.3 并行接口的定义

51 单片机有四各 8 位并行口,共 32 条 I/O 线,这是大家熟知的,这四个口原则上都可以作为 I/O 口使用,只是在需要外部总线扩展时,才将 P0 用作数据 低地址总线, P2 用作高地址总线。除此之外,还可以在片外扩展硬件 I/O 口。51 单片机没有专用的 I/O 指令,所有口地址与数据存储地址时统一编址的。在使用 51C 对单片机进行编程时,对这些口的访问也是通过关键字“sfr”定义的,可以在头文件中定义,也可以在程序的开始部分进行定义。例如: sfr P0 = 0x90,就定义了 P0 口的地址为 90H。

#### 2.4 位变量的定义

51C 除了支持一般 C 语言所具有的数据类型外,还支持“位”数据类型。它是通过关键字“Bit”来实现的。比如: Bit disp\_in,就定义了变量 disp\_in 为位变量。但是,不能将位变量定义成指针,也不能定义位数组。

## 4 应用举例

笔者在研制开发螺旋钢箍机微机控制系统时,采用 Intel 8279 芯片扩展了键盘 显示器。键盘采用 44 物理键阵, 6 位 LED 显示管, 时钟 6MH, 命令字为 2AH, 保证 100KH 的内部工作频率, 数据口地址为 8000H, 状态口地址为 8001H。其 51C 程序如下:

```
#include <eg51.h >
#include <absacc.h >
#define COM XBYTE [001] /* 数据口 * /
#define DAT XBYTE [000] /* 状态口 * /
#define uchar unsigned char
uchar code table [] { 0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d,
0x7d, 0x07, 0x7f, 0x6f, 0x77, 0x7c, 0x39, 0x5e, 0x79,
0x71 };
uchar idata disp_buf [8] = {0, 1, 2, 3, 4, 5, 6, 7 };
sBit cflag = ACC^7;
uchar Key_in ();
uchar deky ();
void disp (uchar idata * d );
void main (void )
{
    uchar i;
    COM = 0xd1;
    Do {ACC = COM;}
    While (cflag = = 1 );
```

```

COM = 0x00; COM = 0x2a;
While ( )
  {or (=0; i &#x28; i++ )
  {isp (isp_buf );
  disp_buf [ ]= Key_in ();
  }
}
}
void disp (t ) /* 显示函数 * /
uchar idata * d;
{ uchar i;
COM = 0x90;
For (=0; i &#x28; i++ )
{COM = i+0x80;
DAT =table [d ];
d ++;
}
}
uchar Key_in (void ) /* 取键值函数 * /
{
uchar i
while (key ()= = 0 );
COM = 0x40;
i = DAT; i = &#x3f;
Return ( );
}
uchar deky (void ) /* FIFO 检查函数 * /
{

```

```

uchar k;
k = COM;
return (k&#x0f );
}

```

### 5 结束语

目前支持 51 系列单片机的语言除了汇编、51C 以外,还有 BASIC 和 PL M。BASIC 是一种初学编程的语言,现在已很少有人去用了。PL M 是 Intel 从 8080 微处理器开始为其系列产品开发的编程语言,它很象 Pascal,也是一种结构化程序设计语言,使用关键字定义结构,用其开发的程序通过 PL M 编译器编译后也能生成紧凑的代码。但是,它的运算能力较差,没有丰富的库函数支持,学习 PL M 无异于学习一种新语言。因此,比较而言用 51C 对单片机编程是最方便、快捷的方法,如果熟悉 ANSIC、Turbo C,只要在此基础上遵从它的一些特殊规定就能开发出较优化的程序,开发周期则比用汇编语言短的多。

### 参考文献

1. 陈宝江等, MCS 单片机应用系统实用指南,机械工业出版社,1997
2. 何力民, 单片机应用系统设计,北京航空航天大学出版社,1995
3. Microtek 公司, EasyPack E 仿真器用户手册,台湾,1996
4. 夏华龙等, AEDK 系列仿真开发机用户手册,1995

图 4 控制界面功能结构图

(上接第 52 页)

} 共有功能  
下不罗列

### 5 结论

控制系统软件的调试分三步进行:第一步对接口板管脚变量的连接进行了检查,各变量输入正常;第二步对自动/手动两种调节方式下各控件功能进行了调试;第三步对常规 PID + 自适应 Smith 预估器的应用分别进行了调试。现场调试表明,整个系统运行良好,满足控制系统的功能要求。

### 参考文献

1. 曹朴芳,对中国造纸工业发展的思考,中国造纸,1998,4
2. Genie 用户手册
3. Jong Hae Kin, Enn Tae Jenng and Hong Bae Park. Robust Control for Parameter Uncertain Delay Systems in State and Control Input. Automatic, 1996, 9
4. 江青茵,无辨识自适应控制预估算法及应用,自动化学报,1997, 2
5. 王连铮,张建华,孟庆贺,增益预估自适应纯滞后补偿控制器,自动化学报,1997, 2
6. 东箭工作室, Visual Basic 5.0 中文版程序设计,清华大学出版社,1997. 11