一个用C语言开发嵌入式软件的实例

汤小戈 黄敏 (上海冶金设计研究院 200070)

摘要:本文论述了用C语言开发无DOS支持的嵌入式应用软件时会碰到的问题及相应的对策,并以Borland C++ 3.1开发环境为例,具体说明为生成可固化运行的程序,应如何修改起动代码和选择编译环境,及如何将生成的EXE文件作必要的修改后进行固化。

关键词:程序设计—C语言 嵌入式软件 固化 磁盘操作系统

C语言既有低级语言的强大功能,又象高级语言易懂易用。在一些曾经是汇编语言一统天下的场合,现在用C语言替代不仅是可能的,而且已成为发展方向。一般讲用C语言开发一个软件比用汇编语言效率要提高2~3倍,相应的软件维护也要容易得多。

与汇编语言不同,一般C语言编译软件,象TC、BC、MSC等生成的可执行程序都需要DOS的支持,这是因为它们提供的起动代码和大部分库函数都要调用DOS,而且这样的程序只能在RAM中运行。如何用C语言开发无DOS支持的嵌入式软件,这是一个比较令人关心的问题,而目前又未见资料详细报导,因此作者把在工作中总结的经验在此介绍给大家,不妨作为抛砖引玉之用。

下面以Borland C++ 3.1开发环境为例说明在开发无DOS支持的嵌入式应用软件会碰到的问题及相应的解决方法。不同的C语言编译软件可能在许多细节上有所不同,但本文给出的方法也适合其他的C语言编译软件。

一、启动代码C0x.OBJ

在一般的应用中,起动代码C0x.OBJ(x可为T、S、C、L和H因编译模式而定)对用户是透明的,连接程序会自动将它同用户的OBJ文件连接。起动代码的作用是在执行用户编写的程序前先完成一些共同的初始化工作,这些初始化工作大部分是同DOS密切相关的,对无DOS支持的嵌入式应用程序,大部分初始化工作是不能执行的,也是不需要的,所以就必须修改启动代码C0x.OBJ,这可通过修改启动代码的源程序C0.ASM来完成:

1.数据定义部分保持不变,即使有些定义的变量不再用到,但保留也不妨

2.去除Get DOS Version代码

3.去除Program Segment(-PSP)和Environment设置代码

4.去除Save Vectors代码和Environment Variables设置代码

5.去除内存需求判断代码和文件柄处理代码

6. 去除 main argument(argc、argv 和取开始执行时间)代码

7. 去除程序结束处理代码,加进自己的结束处理代码

二、浮点问题

浮点涉及 FPINIT. ASM、EMU086. ASM(包括 E087ENTR. ASM 及 E086ENTR. ASM)和 fperr. C 等程序,查看这些源程序不难发现它们要调用 DOS 或 BIOS,一般的调用层次是 abort()→ exit()→--exit()→terminal →int21。--exit()和 terminal 原来在起动代码中,修改后的起动代码已将它们删除,所以原封不动的使用浮点必然会出问题。

对浮点问题可采取的相应对策有两种:

1. 修改与浮点初始化有关的函数 FPINIT. ASM、EMU086. ASM,删除对 DOS 或 BIOS 调用的代码,并用自己的错误处理函数替代原来的浮点错误处理函数(fperr. C),还要保留 INT2、INT34H ~ INT3DH 及 INT75H,用相应的中断函数同这些中断向量相联。

如果调用了数学函数,则被调用的数学函数也要修改。

2. 应用程序不使用浮点,如果需要进行一些精度较高的计算,可自行编制一些简单的多字节加、减、乘、除等必需的子函数。

三、库函数问题

大部分库函数需要 DOS 的支持,就是一些看上去好象同 DOS 无关的函数也可能调用 INT21。还有相当多的库函数需要 PC 机硬件的支持,如时间和日期例程、文本和图形例程。嵌入式软件基本不同文件和屏幕打交道,也不会用到动态内存等功能,而且它所处的硬件环境一般与 PC 机也不同,所以抛弃标准库函数并不会造成多大的不便,用户可以根据自己的特殊需要建立相应的函数,还可方便地使用 Borland C + + 提供的行间汇编功能编写象 inport()、outport()之类同硬件直接打交道的函数。

四、工程制作

为了使修改后得到的 C0MY. ASM 起作用还需要做两件工作:

1. 用 BC 提供的 Build. bat 将 C0MY. ASM 编译成 C0xMY. OBJ(x = T, S, C, M, L, H),当然 Build. bat 也要作适当修改,使得输入文件名为 C0MY. ASM,输出文件名为 C0xMY. OBJ。

2. 必须建立一个 PRJ 文件,在该 PRJ 文件中,除了包含所需的源程序文件外,还必须包括 C0xMY. OBJ 文件,且使 C0xMY. OBJ 作为 PRJ 文件中的第一项。

五、编译、连接环境

在生成所需的 PRJ 文件后,就可以开始编译、连接。编译、连接也要遵循一定的规则:

1. PRJ 文件中的 C0xMY. OBJ 要同编译模式相一致,即如果打算用 TINY 模式编译,则 PRJ 文件中就要用 C0TMY. OBJ。一般嵌入式软件最终都要固化运行,而且这样的软件一般不会超过 64K,所以建议用 TINY 模式编译,这样生成的 EXE 文件才可用 EXE2BIN. EXE 变成 BIN 文件。

2. 使用 C0xMY. OBJ 后,连接时不能选浮点和标准库函数。

上述两条规则反映在 BC 的集成环境下即:

(1) Options/Compiler/Code generation/Model 选 TINY。

(2) Options/Compiler/Advanced code generation/Floating Point 选 None。

(3) Option/Linker/Libraries/Standard Run-time Libraries 选 None

六、固化问题

按上述步骤生成的 EXE 文件适合作调试用,为了生成可固化运行的程序还需要解决许多问题:

1. CPU 初始化

在固化运行时,由于没有 DOS 或监控程序支持,所以应用程序必须自己完成 CPU 的初始化。CPU 的初始化应在系统上电后首先进行,且初始化代码基本是固定的,因此这部分代码最适合放在起动代码中,把它变成对用户透明的。

以 80C186CPU 为例,这样修改后的起动代码不妨称为 C0Embed. ASM,除了应完成 C0MY. ASM 的工作外它

还需完成：

⑴ 激活所有存储区选择线

⑵ 激活所有 I/O 选择线

⑶ 给有关寄存器赋初值

2.生成二进制文件

用 C0Embed.ASM 编译成的 C0Tbed.OBJ 替代 C0xMY.OBJ，再按照上述方法生成 EXE 文件。该 EXE 文件必须经重定位后才能用 DOS 中的 EXE2BIN 转换成 BIN 文件，这种 BIN 文件是标准的 intel 二进制文件，绝大多数的 EPROM 固化器都能固化，但是从运行的角度看还需作进一步的修改。

3.修改码段写

在用 TC 或 BC 生成的 EXE 文件中，除 huge 模式外，DGROUP@就是数据段的段地址，即 DS 的值取自 DGROUP@。DGROUP@是定义在代码段中的变量，它的值是在程序运行时确定的，在起动代码中赋值。当程序固化运行时，代码段是不能有写操作的。为避免码段写，修改后的 C0Embed.OBJ 中给 DGROUP@赋值的语句已被删除，但是这样 DGROUP@的值就不是程序运行所要求的，为解这个问题，一种变通的方法就是在 BIN 文件中给 DGROUP@赋值。与重定位项不同的是 DGROUP@在 EXE(BIN) 文件中的位置没在文件头中给出，但可以从 MAP 文件中查到，而且对同一个启动代码，DGROUP@的位置是固定的。在嵌入式应用中，RAM 地址是已知的，故将其地址赋给 DGROUP@就确定了数据段地址，而且使得数据段在 RAM 而不是 E-PROM 中。

4.调整数据段的偏移位置

在 80X86 系列嵌入式系统中，数据段地址一般为 0000H，但是这样数据区的位置就固定了，在有些应用中如监控程序为了给应用程序留下连续的空间，就需要调整其本身的数据区的位置。

随便调动数据区的位置是危险的，这样很可能会使程序找不到正确的数据。通过分析 C 语言生成的 EXE 文件的内存分配图发现 far 类型的数据处在数据段的开头，只要在源文件中定义一个 far 类型的数组而不用它，就相当于将整个数据区往后移了，当然这个 far 类型的数组应定义在其他 far 类型数据的前面。

-TEXT class 'CODE'
code
-DATA class 'DATA'
far data
-DATA class 'DATA'
initialized data
-BSS class 'BSS'
uninitialized data
HEAP
FREE SPACE
STACK

5.固化

经过修改码段写后的 BIN 文件就是可以固化运行的程序了，固化的段地址在重定位时已确定了，而偏移地址固定为 0100H，这是由 DOS 和 BC 决定的。如果硬件是 16 位数据总线，EPROM 就分成奇、偶两片，固化时 EPROM 的起始地址就应是 0080H，因为两片 EPROM 并头各留 80H 个字节，其和就是 0100H 字节。

结束语

本文提出的用 C 语言开发无 DOS 支持的嵌入式软件的方法是可行的，并且具有较普遍的意义，作者应用这种方法成功地开发出了某型数字控制器上的系统支持软件。

参考资料

[1] 《Borland C++ 2.0 入门与使用指南》. 北京：海洋出版社，1991

[2] 《Borland C++ 2.0 程序设计手册与资源开发工具》. 北京：海洋出版社，1991

[3] 萧黎等.《Turbo C 2.0 运行库函数源程序与参考大全》. 北京：中国科学院希望高级电脑公司。

（来稿时间：1997 年 9 月）