

---

关于在 KEIL C51 中嵌入汇编以及 C51 与 A51 间的相互调用

作者:ycong\_kuang

有关 c51 调用汇编的方法已经有很多帖子讲到,但是一般只讲要点,很少有对整个过程作详细描述,对于初学者是不够的,这里笔者通过一个简单例子对这个过程进行描述,希望能对初学者有所帮助。几年来,在这个论坛里笔者得到很多热心人指导,因此也希望藉此尽一点绵薄之力。

在这个例子里,阐述了编写 c51 程序调用汇编函数的一种方法,这个外部函数的入口参数是一个字符型变量和一个位变量,返回值是一个整型变量。例中,先用 c51 写出这个函数的主体,然后用 SRC 控制指令编译产生 asm 文件,进一步修改这个 asm 文件就得到我们所需要的汇编函数。该方法让编译器自动完成各种段的安排,提高了汇编程序的编写效率。

step1. 按写普通 c51 程序方法,建立工程,在里面导入 main.c 文件和 CFUNC.c 文件。

相关文件如下:

//main.c 文件

```
#include < reg51.h >
```

```
#define uchar unsigned char
```

```
#define uint unsigned int
```

```
extern uint AFUNC(uchar v_achr,bit v_bflag);
```

```
void main()
```

```
{
```

```
    bit BFLAG;
```

```
    uchar mav_chr;
```

```
    uint    mvintrslt;
```

```
    mav_chr=0xd4; BFLAG=1;
```

```
    mvintrslt=AFUNC(mav_chr,BFLAG);
```

```
}
```

//CFUNC.c 文件

```
#define uchar unsigned char
```

```
#define uint unsigned int
```

```
uint AFUNC(uchar v_achr,bit v_bflag)
```

```
{
```

```
    uchar tmp_vchr;
```

```
    uint  tp_vint;
```

```
tmp_vchr=v_achr;  
tp_vint=(uint)v_bflag;  
return tmp_vchr+(tp_vint<<8);  
}
```

step2. 在 Project 窗口中包含汇编代码的 C 文件上右键, 选择 “Options for ...”, 点击右边的 “Generate Assembler SRC File” 和 “Assemble SRC File”, 使检查框由灰色变成黑色(有效)状态;

step3. 根据选择的编译模式, 把相应的库文件(如 Small 模式时, 是 Keil\C51\Lib\C51S.Lib)加入工程中, 该文件必须作为工程的最后文件;

step4. build 这个工程后将会产生一个 CFUNC.SRC 的文件, 将这个文件改名为 CFUNC.A51 (也可以通过编译选项直接产生 CFUNC.A51 文件), 然后在工程里去掉库文件 (如 C51S.Lib) 和 CFUNC.c, 而将 CFUNC.A51 添加到工程里。

//CFUNC.SRC 文件如下

```
.\CFUNC.SRC generated from: CFUNC.c  
NAME CFUNC
```

```
?PR?_AFUNC?CFUNC      SEGMENT CODE  
?BI?_AFUNC?CFUNC      SEGMENT BIT OVERLAYABLE  
    PUBLIC    ?_AFUNC?BIT  
    PUBLIC    _AFUNC
```

```
    RSEG  ?BI?_AFUNC?CFUNC  
?_AFUNC?BIT:  
    v_bflag?041:  DBIT  1  
; #define uchar unsigned char  
; #define uint unsigned int  
;  
; uint AFUNC(uchar v_achr, bit v_bflag)
```

```
    RSEG  ?PR?_AFUNC?CFUNC  
_AFUNC:  
    USING  0  
        ; SOURCE LINE # 5  
;---- Variable 'v_achr?040' assigned to Register 'R7' ----  
; {  
        ; SOURCE LINE # 6  
;    uchar tmp_vchr;  
;    uint  tp_vint;
```

```
;
;      tmp_vchr=v_achr;
;          ; SOURCE LINE # 10
;---- Variable 'tmp_vchr?042' assigned to Register 'R5' ----
      MOV      R5,AR7
;      tp_vint=(uint)v_bflag;
;          ; SOURCE LINE # 11
      MOV      C,v_bflag?041
      CLR      A
      RLC      A
;---- Variable 'tp_vint?043' assigned to Register 'R6/R7' ----
;      return tmp_vchr+(tp_vint<<8);
;          ; SOURCE LINE # 12
      MOV      R6,A
      MOV      R4,#00H
      CLR      A
      ADD      A,R5
      MOV      R7,A
      MOV      A,R4
      ADDC     A,R6
      MOV      R6,A
;  }
;          ; SOURCE LINE # 13
?C0001:
      RET
; END OF _AFUNC

      END
```

step5. 检查 main.c 的“Generate Assembler SRC File”和“Assemble SRC File”是否有效，若是有效则点击使检查框变成无效状

态；再次 build 这个工程，到此你已经得到汇编函数的主体，修改函数里面的汇编代码就得到你所需的汇编函数了。

#### 参考文献：

1. 徐爱钧，彭秀华。单片机高级语言 C51windows 环境编程与应用，电子工业出版社
2. [www.c51bbs.com](http://www.c51bbs.com), C51 编程:关于在 KEIL C51 中直接嵌入汇编。。。帖子编号： 83838 发表用户:Youth

.....  
.....  
keil 中汇编函数调用 c51 函数 [ycong\_kuang]

在 keil 的写法可参考 89852 帖子，具体如下：

与 89852 帖子相比，第一步在工程里多了一个被汇编调用的 c51 的函数文件 (c51func.c)，至于汇编函数还是先用 c51 编写出主体

(a51func.c), 这样汇编程序接口和段都交给编译器处理, 你只管在编译成汇编代码后按你的要求改写汇编代码就行了。

例程如下:

```
//main.c
```

```
#include < reg51.h >
```

```
#define uchar unsigned char
```

```
#define uint unsigned int
```

```
extern uint AFUNC(uchar v_achr, bit v_bflag);
```

```
void main()
```

```
{
```

```
    bit BFLAG;
```

```
    uchar mav_chr;
```

```
    uint    mvintrslt;
```

```
    mav_chr=0xd4; BFLAG=1;
```

```
    mvintrslt=AFUNC(mav_chr, BFLAG);
```

```
}
```

```
//a51FUNC.c
```

```
#define uchar unsigned char
```

```
#define uint unsigned int
```

```
extern uint CFUNC(uint);
```

```
uint AFUNC(uchar v_achr, bit v_bflag)    //c51 写的汇编函数, 最终要变成汇编代码
```

```
{
```

```
    uchar tmp_vchr;
```

```
    uint  tp_vint;
```

```
    tmp_vchr=v_achr;
```

```
    tp_vint=(uint)v_bflag;
```

```
    return CFUNC(tp_vint);
```

```
    //这里调用一个 c51 函数
```

```
}
```

```
//c51FUNC.c
```

```
#define uchar unsigned char
```

```
#define uint unsigned int
```

```
uint CFUNC(uint v_int)           //被汇编函数调用 c51 函数
{
    return v_int<<2;
}
```

第二步是按 89852 帖子的 step2, 3, 4 把用 c51 写的(汇编)函数变成 a51 文件(今天我试了一下 step3 可以不要)例程编译结果如下:

```
; .\a51func.SRC generated from: a51func.c
NAME      A51FUNC

?PR?_AFUNC?A51FUNC    SEGMENT CODE
?DT?_AFUNC?A51FUNC    SEGMENT DATA OVERLAYABLE
?BI?_AFUNC?A51FUNC    SEGMENT BIT OVERLAYABLE
    EXTRN    CODE (_CFUNC)
    PUBLIC    ?_AFUNC?BIT
    PUBLIC    _AFUNC

    RSEG    ?DT?_AFUNC?A51FUNC
?_AFUNC?BYTE:
    tmp_vchr?042:    DS    1

    RSEG    ?BI?_AFUNC?A51FUNC
?_AFUNC?BIT:
    v_bflag?041:    DBIT    1
; //a51FUNC.c
;
; #define uchar unsigned char
; #define uint unsigned int
;
; extern uint CFUNC(uint);
;
; uint AFUNC(uchar v_achr,bit v_bflag)

    RSEG    ?PR?_AFUNC?A51FUNC
_AFUNC:    ;c51 所写的函数产生的汇编代码从这里开始
    USING    0
            ; SOURCE LINE # 8
;---- Variable 'v_achr?040' assigned to Register 'R7' ----
; {
            ; SOURCE LINE # 9
;    uchar tmp_vchr;
;    uint  tp_vint;
;
;    tmp_vchr=v_achr;
```

```
                ; SOURCE LINE # 13
MOV            tmp_vchr?042, R7
;            tp_vint=(uint)v_bflag;
                ; SOURCE LINE # 14
MOV            C, v_bflag?041
CLR            A
MOV            R6, A
RLC            A
MOV            R7, A
;---- Variable 'tp_vint?043' assigned to Register 'R6/R7' ----
;            这里说明 R6, R7 内容就是 tp_vint
;            return CFUNC(tp_vint);
                ; SOURCE LINE # 16
LCALL          _CFUNC      ;这里调用了用 c51 写的函数
; }
                ; SOURCE LINE # 17
?C0001:
RET
; END OF _AFUNC

END
```

这个文件就是你的汇编函数所在文件,把函数里面的汇编代码修改成你所需的汇编函数就 ok 了。

建议参考 徐爱钧, 彭秀华所写的《单片机高级语言 C51windows 环境编程与应用》或马忠梅所写的

《单片机的 c 语言应用程序设计》有关混合语言编程有关章节

.....  
关于在 KEIL C51 中直接嵌入汇编。。。 [Youth]

有时在 C51 程序中需要嵌入一些汇编代码, 这时当然可以用通常的作法:

按照 C51 与汇编的接口写一个汇编函数, 然后在 C51 程序中调用该函数。(此种方法可在论坛里搜索, 以前有很多帖子讲到, 不再重复)

下面介绍直接嵌入汇编代码的方法:

1、在 C 文件中要嵌入汇编代码片以如下方式加入汇编代码:

```
#pragma ASM
; Assembler Code Here
#pragma ENDASM
```

2、在 Project 窗口中包含汇编代码的 C 文件上右键, 选择 “Options for ...”, 点击右边的 “Generate Assembler SRC File”

和 “Assemble SRC File”, 使检查框由灰色变成黑色(有效)状态;

3、根据选择的编译模式，把相应的库文件(如 Small 模式时，是 Keil\C51\Lib\C51S.Lib)加入工程中，该文件必须作为工程的最  
后文件；

4、编译，即可生成目标代码。