如何在 C 语言中调用 P89V51RD2 的 IAP 功能

P89V51RD2 具有 IAP（在应用中编程）功能，用户通过在应用程序中调用 IAP 子程序，可实现有选择的对 FLASH 块进行擦除和编程。P89V51RD2 的 bootrom 区为 0000H~1FFFH，为避免和用户的程序代码发生冲突，调用 IAP 功能的代码要从地址 2000H 以后开始存放。以下讲述在 Keil μVision2 下用 C 语言和汇编语言混编的办法实现 IAP 调用的方法。

调用 IAP 需要向指定的寄存器中写入指定的参数，在编写 C51 调用的汇编语言子程序时，指定参数的传入可用传递参数法实现。传递参数法就是将要传递的数据或变量通过形式参数传送，函数的前 3 个参数要使用表 1 规定的寄存器来传递，当寄存器不够用时（因为一次传递最多只能使用 1 组 8 个寄存器）或参数多于 3 个时放在不同存储模式所对应的默认数据段中。

表 1　函数前三个参数使用的寄存器

	Char 或一字节指针	Int 或二字节指针	Long float	三字节通用指针
第一参数	R7	R6, R7	R4~R7	R1, R2, R3
第二参数	R5	R4, R5	R0~R3	R1, R2, R3
第三参数	R3	R2, R3	无	R1, R2, R3

当函数返回值时，一律用寄存器来完成，规则如表 2：

表 2　函数返回值指定用寄存器

返回值类型	寄存器	注
Bit	CY	
Unsigned char	R7	
Unsigned int	R6, R7	R6 放高位，R7 放低位
Unsigned long	R4~R7	R4 放最高位，R7 放最低位
Float	R4~R7	IEEE 标准 R7 放符号位及阶码
指针	R1, R2, R3	R3 放存储空间编码，R2 放偏移地址高位，R1 放偏移地址低位

本文中，主函数及一些子函数用 C 语言编写，IAP 的功能模块用汇编语言编写。在编写汇编程序之前，首先设计包含哑函数的 C 模块 r_wIAP.c，将源程序的读 IAP 函数和写 IAP 用下面的两个空函数来代替

```
unsigned char P89V51RD2_Read_IAP(unsigned int Flash_Address)
{
}
unsigned char P89V51RD2_Write_IAP(unsigned int Flash_Address,unsigned char Value)
{
}
```

选择 r_wIAP.c 文件，单击鼠标右键选择 Options for … ，产生的界面如图 1 所示，在界面的右边，选择 Generate Assembler SRC File 和 Assemble SRC File，用来控制生成.SRC 文件。

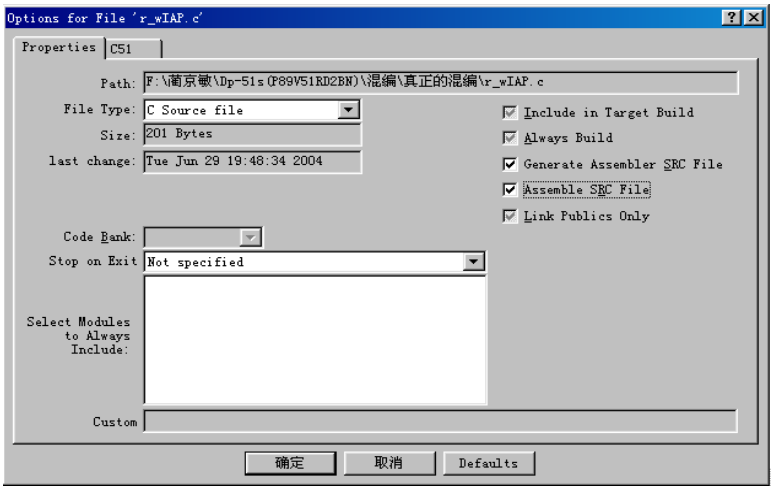

图 1　Properties 对话框页

使用 SRC 控制命令生成.SRC 源程序，在源程序中找到所生成的函数名，重新用汇编语言编写，需要注意的是，汇编子程序的取名要和哑函数生成的函数名一样。具体的转换细节参考《单片机 C 语言 Windows 环境下编程宝典》。

这时将 r_wIAP.c 从 project 移出，将汇编文件 R_WIAP.ASM 添加到 project 中。由于 P89V51RD2 的 bootrom 区为 0000H~1FFFH，要求调用 IAP 功能模块的代码从 2000H 以后的地址存放，在 C 语言中，定位代码存放地址的操作方法如下：

在 Keil μVision2 的 project 菜单中，选择 Options for Taget … ，点击 BL51 Locate，设置 IAP 调用模块的存放地址，如图 2 所示。在编译函数时，还生成对应的.M51 文件，打开当前目录下的.M51 文件，找到要存放在指定地址的函数（注意，在 M51 文件中的函数名和在 C 语言中的函数名有所差别），将它写在“Code”栏中，在括号中注明函数存放地址，若有多个函数需要指定存放，则中间用逗号分开。本例中将 IAP 的读和写函数分别放到 2000H 地址和 2100H 地址。

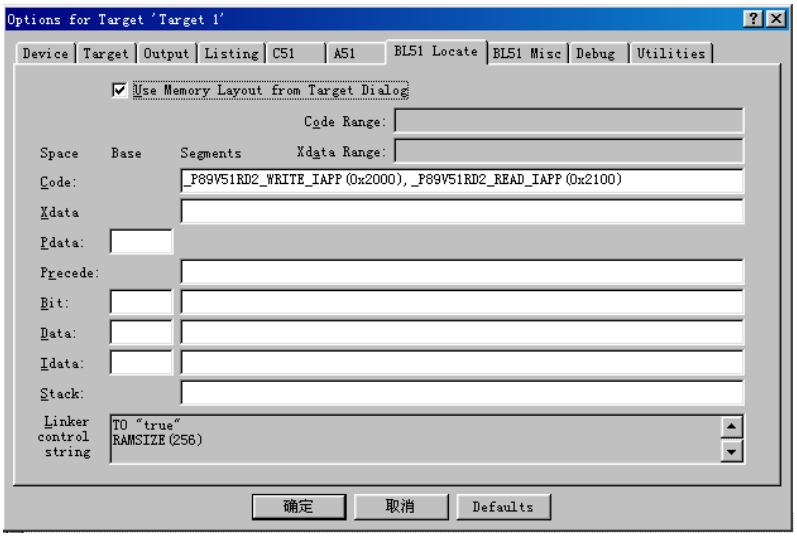

图 2　段定位到 Flash 的 code 存储器

按以上步骤设置后，对文件进行编译链接，编译过程中不再有警告和错误的提示，生成的.HEX 文件即可供用户使用了。

以下给出混编的源程序，仅供参考。

附：

```
#include <reg51.h>
#define   uchar unsigned char
#define   uint   unsigned int

sfr     FCF = 0xB1;
sbit    LED1 = P1^0;
sbit    LED2 = P1^1;
sbit    LED3 = P1^2;
sbit    KEY1 = P3^3;
sbit    KEY2 = P3^4;

uchar P89V51RD2_Write_IAP(unsigned int Flash_Address,unsigned char Value);   //   IAP 字节写函数
uchar P89V51RD2_Read_IAP(unsigned int Flash_Address);                        //   IAP 字节读函数

main()
{
       uchar temp;
       uint    i;

       do{
          if(KEY1= =0){
                 temp = P89V51RD2_Write_IAP(0x8000,0xaa);      //   调用 IAP 写命令
                 if(temp= =0)
                     LED1 = 0;
                 while(KEY1= =0);
                 LED1 = 1;                                      //   写成功,点亮 LED1
                 }
          if(KEY2= =0){
                temp = P89V51RD2_Read_IAP(0x8000);             //    调用 IAP 读命令
                if(temp= =0xAA)
                     LED2 = 0;
                while(KEY2= =0);
                LED2 = 1;                                      //   读成功点亮 LED2
                }
            for(i=0;i<4000;i++);
            LED3 = ~LED3;
        }while(1);
}
```

R_WIAP.ASM 文件

```
PUBLIC                          _P89V51RD2_Write_IAP    ;IAP 字节写子函数
_P89V51RD2_Write_IAPP           SEGMENT         CODE
RSEG                            _P89V51RD2_Write_IAPP
```

```
_P89V51RD2_Write_IAP:    NOP
P89V51RD2_Write_IAP:
                         PUSH    ACC
                         PUSH    DPH
                         PUSH    DPL
                         MOV     R1,#02H       ;调用字节写命令
                         ANL     0B1H,#0FCH    ;清零 BSEL 位
                         MOV     DPH,R6
                         MOV     DPL,R7
                         MOV     A,R5
                         LCALL   1FF0H
                         MOV     R7,A          ;由 R7 返回是否成功写入的消息
                         ORL     0B1H,#01H     ;返回用户程序
                         POP     DPL
                         POP     DPH
                         POP     ACC
                         RET

PUBLIC                   _P89V51RD2_Read_IAP   ;IAP 字节读函数
_P89V51RD2_Read_IAPP     SEGMENT     CODE
RSEG                     _P89V51RD2_Read_IAPP
_P89V51RD2_Read_IAP:     NOP
P89V51RD2_Read_IAP:
                         PUSH    ACC
                         PUSH    DPH
                         PUSH    DPL
                         MOV     R1,#03H       ;调用字节读命令
                         ANL     0B1H,#0FCH    ;清零 BSEL 位
                         MOV     DPH,R6
                         MOV     DPL,R7
                         LCALL   1FF0H
                         MOV     R7,A          ;将读出的数据放到返回值 R7 中
                         ORL     0B1H,#01H     ;返回用户程序
                         POP     DPL
                         POP     DPH
                         POP     ACC
                         RET
                         END
```

参考文献：

1.《P89V51RD2 器件手册》，http://www.zlgmcu.com

2.《单片机 C 语言 Windows 环境下编程宝典》，马忠梅，戚军等，北京航空航天大学出版社。