

存储器区切换功能用户指南

内容

简介

基本操作

存储器切换应用示例

在 Keil 环境下使用存储器区切换功能


简介

为了能够使用华邦 W77E532 微控制器的 2 个 APROM，我们需要用 Keil 或 IAR 软件来编译和连接应用程序。然后把程序烧写到 FLASH ROM 或外部程序存储器中。

存储器区切换功能描述如下。

基本操作

本节介绍如何在 Keil 环境下使用存储器区切换功能。

1. 运行 Keil 程序然后在项目选项处新建或打开一个项目。下面的例子是在 W77E532 的 2 个存储器区中进行切换。该组中有 3 个 C 程序，他们是 root.c,func0.c,func1.c。
2. 在“project”  “Targets,Groups,Files”项中添加 root,bank0 和 bank1 组。见图 1

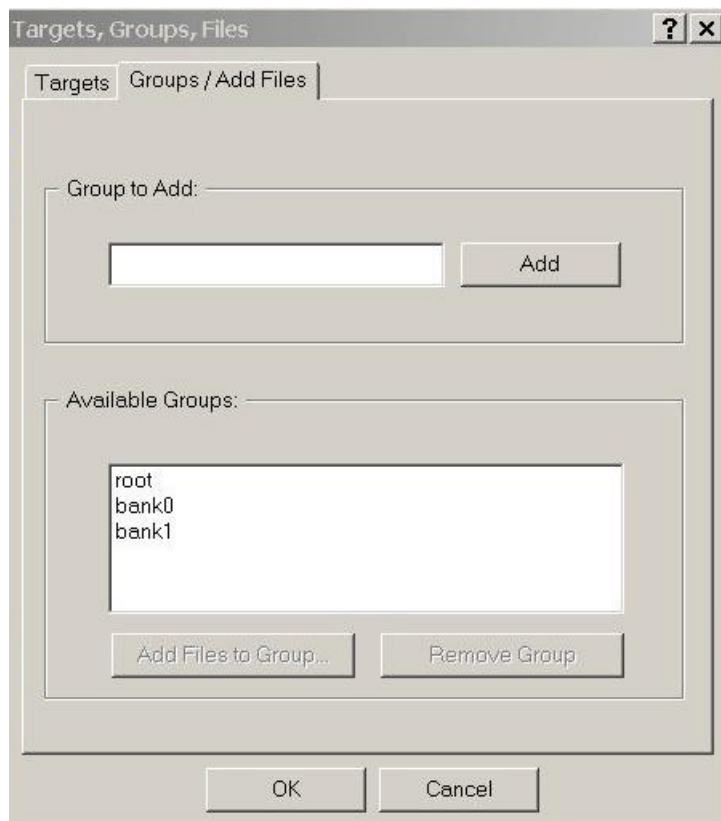



图 1

3. 在“Project”  “Options for Target ‘Target1’”大项中，为“Code Rom Size:”小项选择 Large:64K program，为“Code Banking”选择 Banks:2，将存储器区的起始地址设为 0x8000，结束地址设为 0xFFFF(起始地址可以改变)，见图 2。

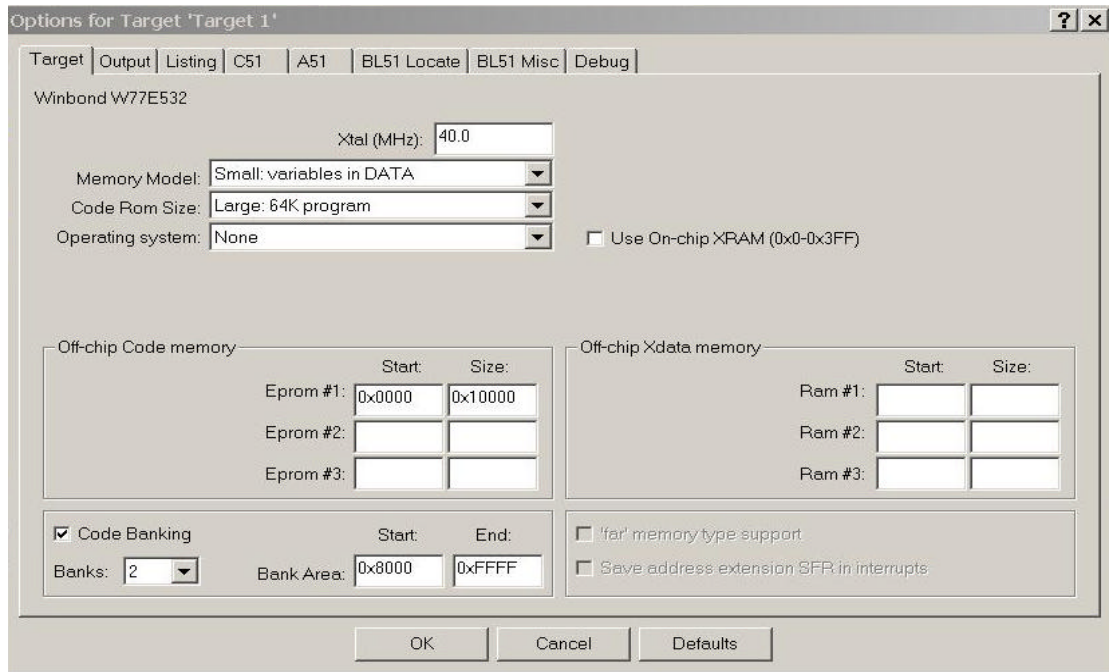


图 2

4. 在“Project”下的“Options for Group ‘root’”大项中来设置 root，然后将“Code Bank:”选为 Common，然后重复第四步来设置 bank 0 和 bank 1 的“Code bank:”为 Bank #0 和 Bank #1。这些设定将选择代码存放的存储器区域。见图 3

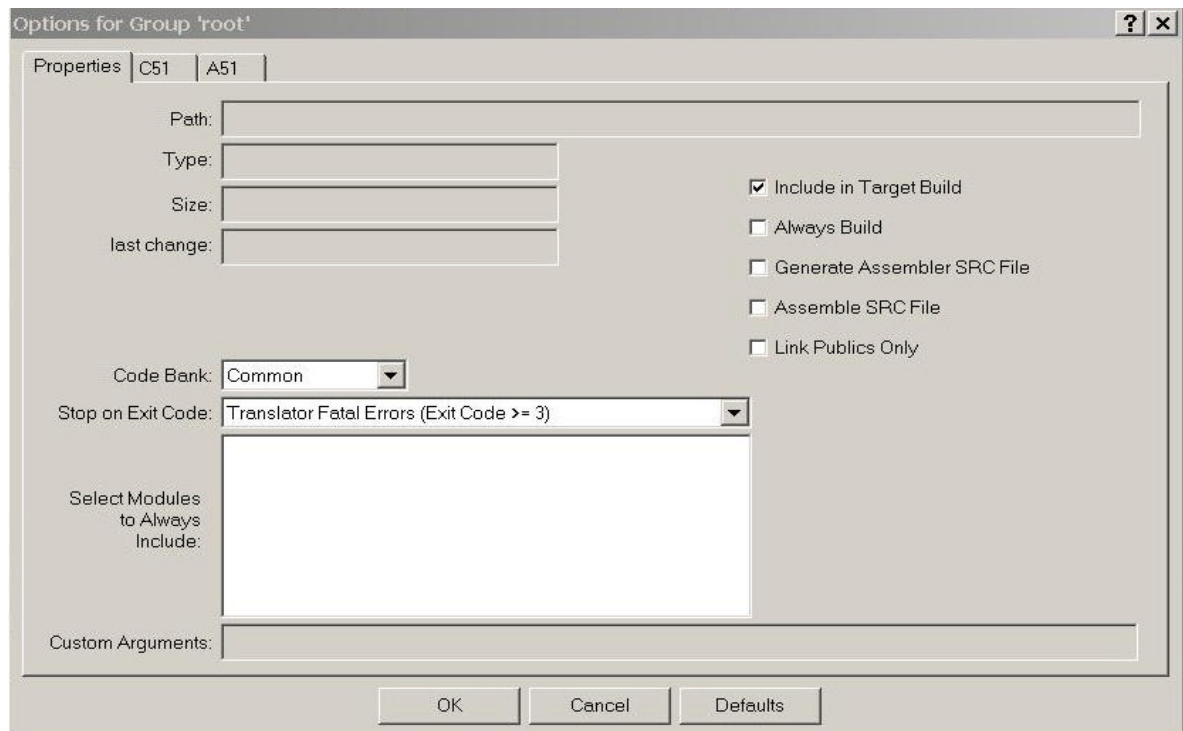


图 3

5. 用户需要将 L51_BANK.A51 文件拷贝至用户目录下，并为 root 组添加文件。如果文件的设置与用户电路图不一致，那么用户需要对该文件进行修改。要修改的部分如下：

```

*****配置部分*****
?B_NBANKS SET 2 ;定义最大存储器区数目 *
; ;?B_BANKS 最大值为 32 *
;注意:?B_NBANKS 的有效值为 2, 4, 8, 16, 或 32. 如果用户应用程序要使用 3 个存储器区则
; 数目应当定义为 4 *
; *
; *
?B_MODE EQU 0 ;0 通过 8051 口进行存储器切换 *
; ;1 通过 XDATA 口进行存储器切换 *
; *
?B_RTX EQU 0 ;0 针对没有 RTX-51 FULL 的应用程序 *
; ;1 针对有 RTX-51 FULL 的应用程序 *
; *
IF ?B_MODE = 0; *
;-----*
;如果 ?BANK?MODE 为 0 则定义如下数值 *
;通过 8051 口进行存储器切换定义 口 地址/位 *
?B_PORT EQU P1 ;default is P1 *
?B_FIRSTBIT EQU 7 ;default is Bit 7 *
;-----*

```

6. 在 c_root.c 中定义变量 ROMCON 或依例子设置用户的包含文件“sfr ROMCON = 0xAB”。
7. 完成所有步骤后，用户可以在 build 项上来 build 整个项目。之后系统将自动编译，连接，转换存储器区目标文件，并会在项目目录里创建 2 个 16 进制文件。用户可以对它直接编程或用 2 进制文件来对它编程。
8. 验证用户函数是否正确。

存储器区切换应用示例

存储器区切换功能示意代码：

```

/*****/
/* 8051 C 语言下存储器区切换示意程序*/
/* 文件名为 "c_root.c",并定义在 Common 区 */
/* 文件包含 main() 程序和 delaytime() 子程序 */
/* */
/*****/
#include "REG54_1.h"

sfr ROMCON = 0xAB; // 定义 SFR ROMCON 寄存器

extern void func0(void);
extern void func1(void);
main()
{
    P1_7=0; // 初始化主程序至 bank0
    ROMCON=0x0f; // 用 P1.7 ( A16 ) 来使能 2 个存储器区
    do
    {
        P1_7=0; // 设 A16 为 "0" 选择 bank0
        func0(); // 调用位于 bank0 区的 func0 子程序
        P1_7=1; // 设 A16 为 "1" 来选择 bank1
        func1(); // 调用位于 bank1 区的 func1 子程序
    }
}

```

```

    }while(1);
}
/*****/
/* delay subroutine */
/*****/
void delaytime(unsigned int count)
{
    unsigned int i,j,k;

    for(i=0;i<count;i++)
        for(j=0;j<100;j++)
            for(k=0;k<120;k++)
                ;
}

/=====/

/*****/
/* 文件名:func0.c */
/* 位于 bank0 区 */
/*****/
#include "REG54_1.h"

extern void delaytime(unsigned int);

void func0(void)
{
    P3=0xd6; // 端口 3 数据输出
    delaytime(50);
}

/=====/

/*****/
/* 文件:func1.c */
/* 位于 bank1 区 */
/*****/
#include "REG54_1.h"

extern void delaytime(unsigned int);

void func1(void)
{
    P3=0x7a; //端口 3 数据输出
    delaytime(50);
}

```