样例程序：PEC 模块使用

1．简介：本程序实现功能如下：

使用 GPT1 作为中断源触发 PEC 传输，PEC 将软件中的定义的一个数组的数据传输到另一个数组。传输次数为 10，传输完成后，触发 End of PEC 中断，重置 PEC，继续传输。

2．PEC 介绍

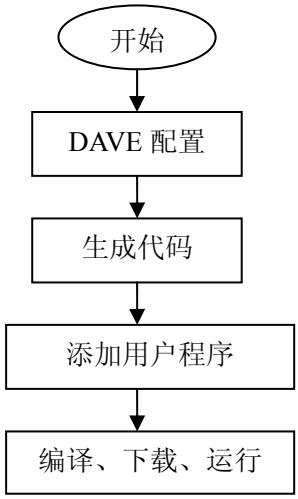
XC164CM 提供 8 路类 DMA 功能的 PEC 通道。可以由任何中断请求触发，在任意两个内存地址之间传递字节数据或字数据。仅仅从当前 CPU 进程中“窃取”一个时钟周期去完成，无需保存/恢复现场。占用时间少，效率高。

每路 PEC 通道拥有独立的计数器，用于指定传输的次数，当传输完指定次数后，可以触发 EOP (End of PEC)中断。

在每次 PEC 传输完成后，可以配置为自动增加源地址或目标地址，灵活性高。

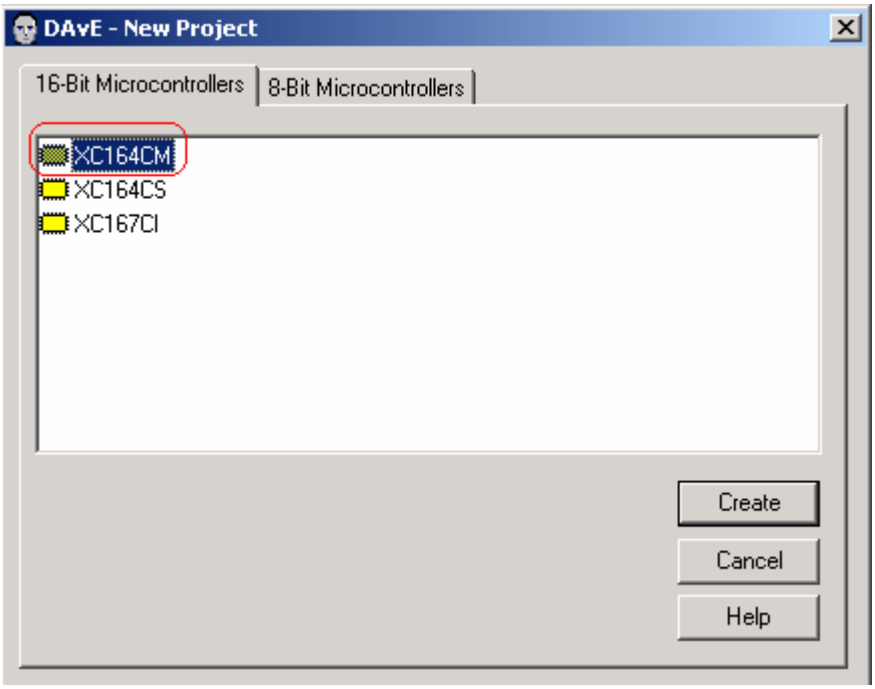
关于 PEC 模块功能的详细介绍，请参照 XC164CM 用户手册。

3．操作流程

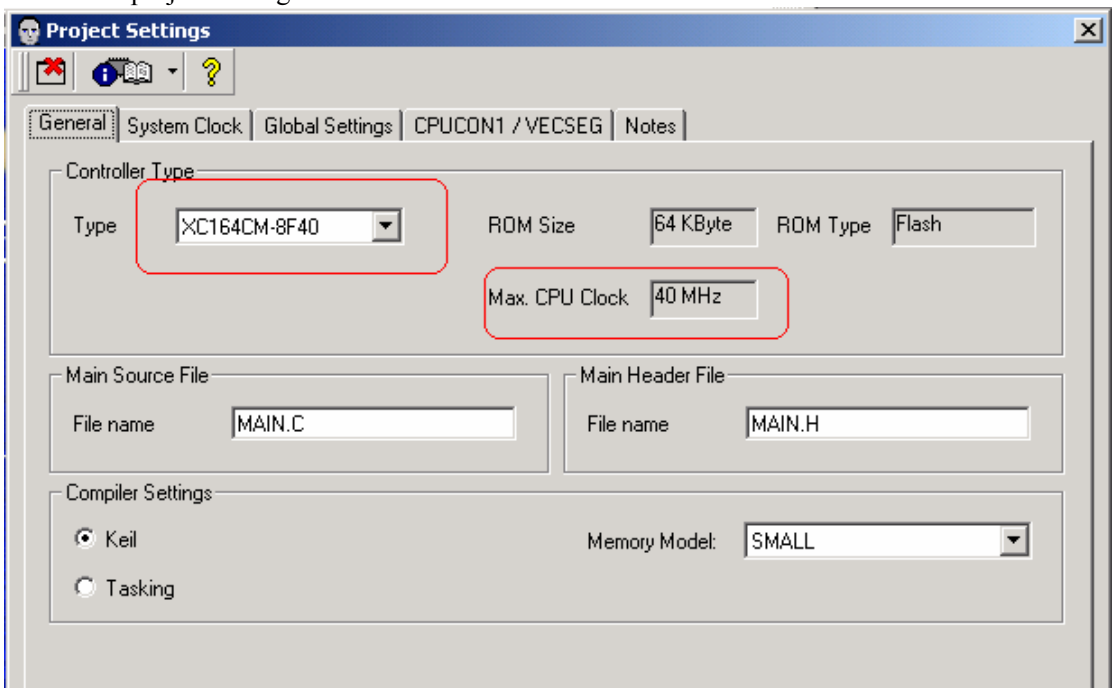


4．DAvE 配置

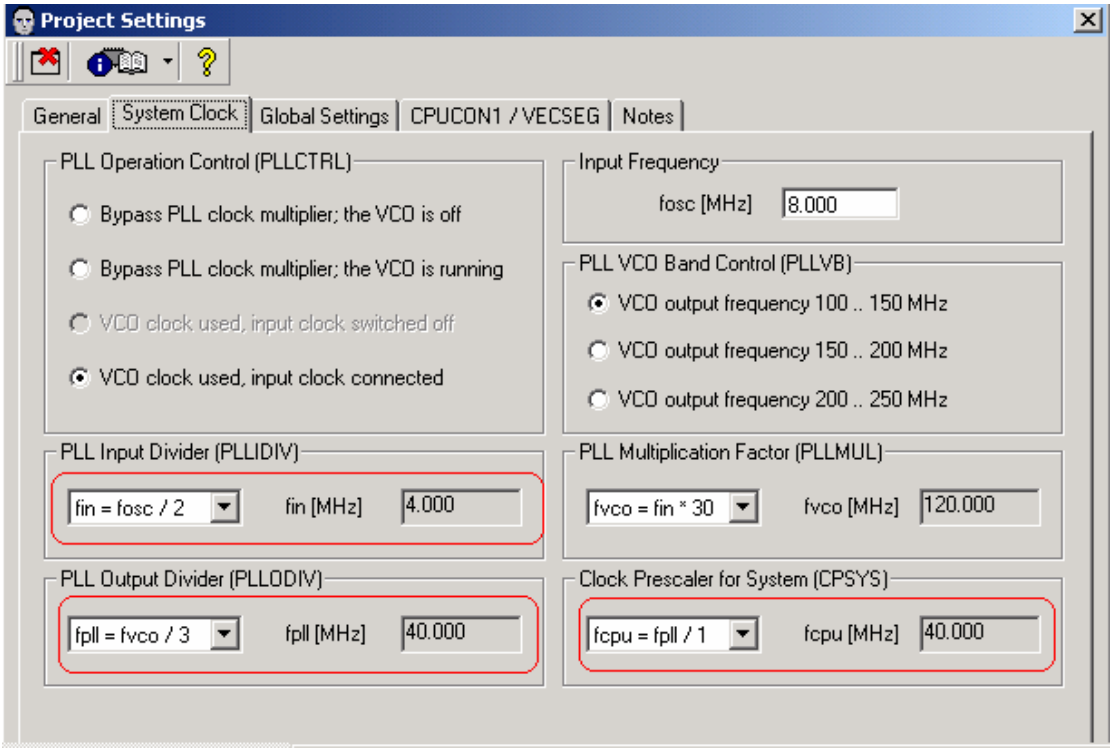
4．1 New project: select XC164cm



4．2 The project settings

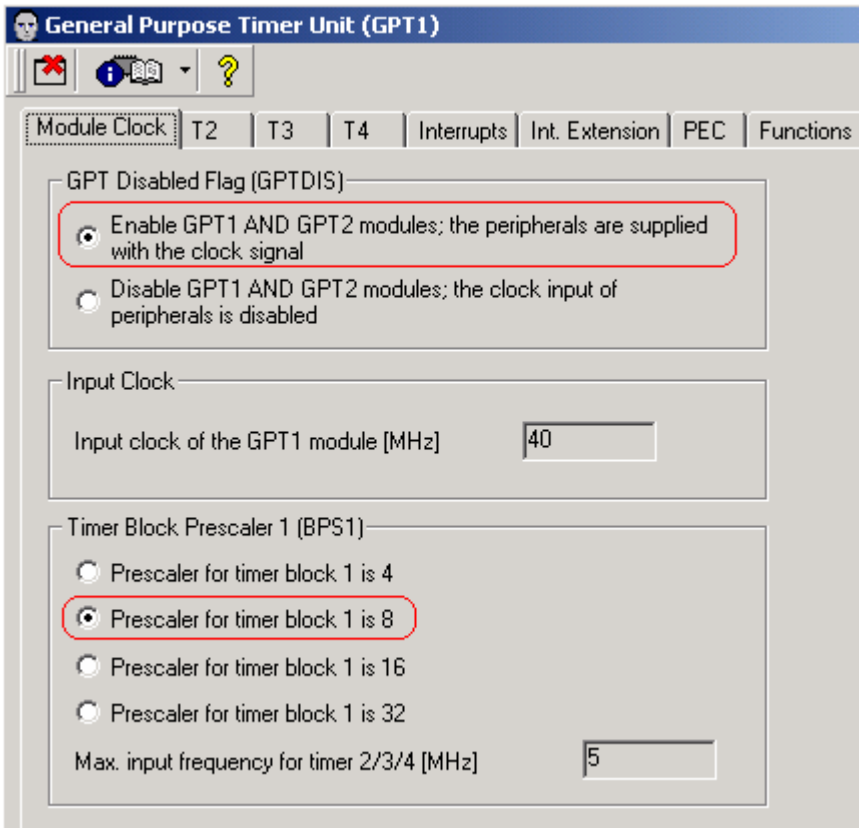


System clock

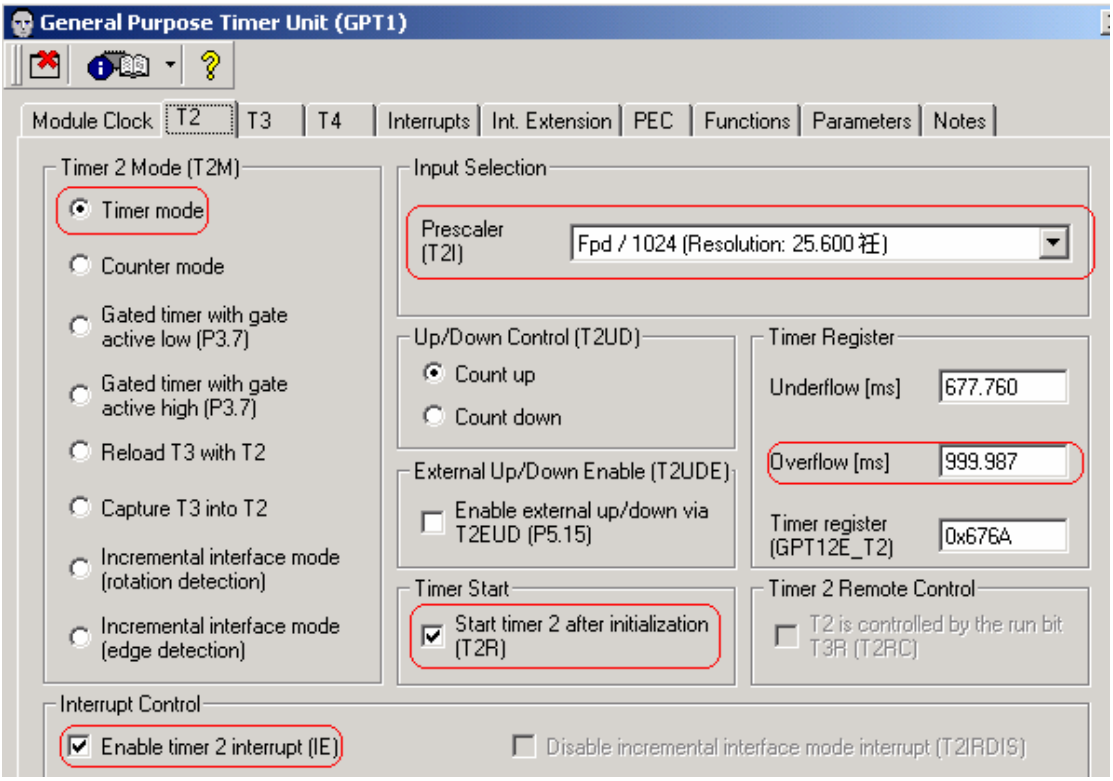


4．3 GPT1

使能模块，选择预分频系数。

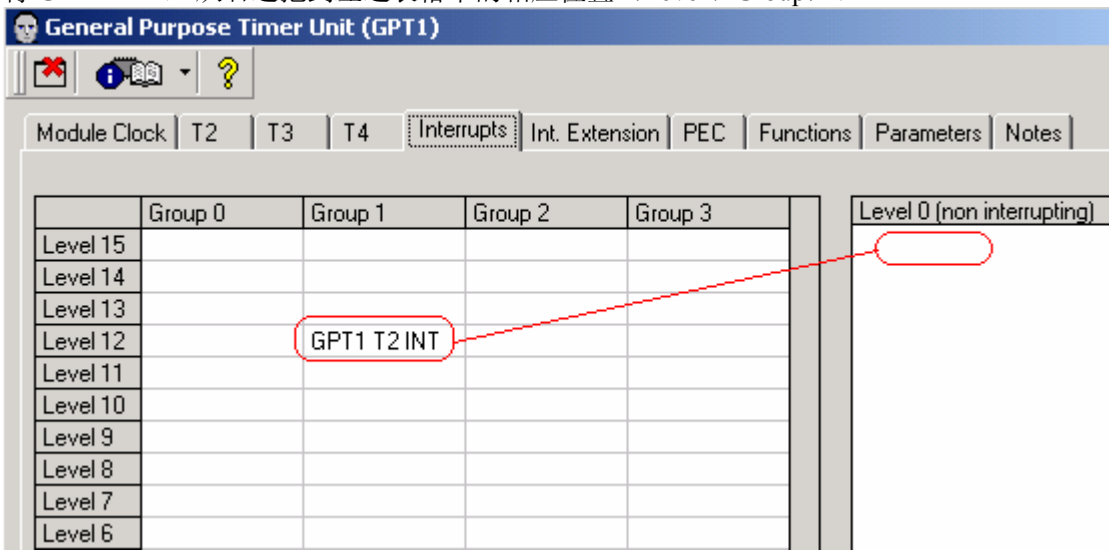


配置 T2：Timer 模式，1s 定时周期，start after initialization, 使能中断。



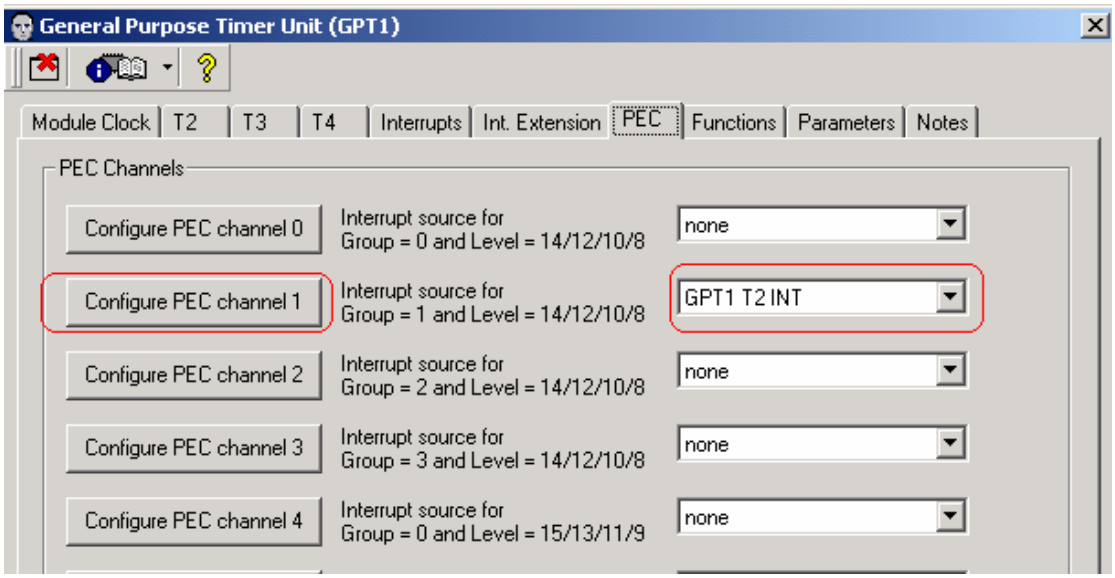
配置 T2 定时中断优先级

将 GPT1 T2INT 从右边拖到左边表格中的相应位置（Level、Group）。

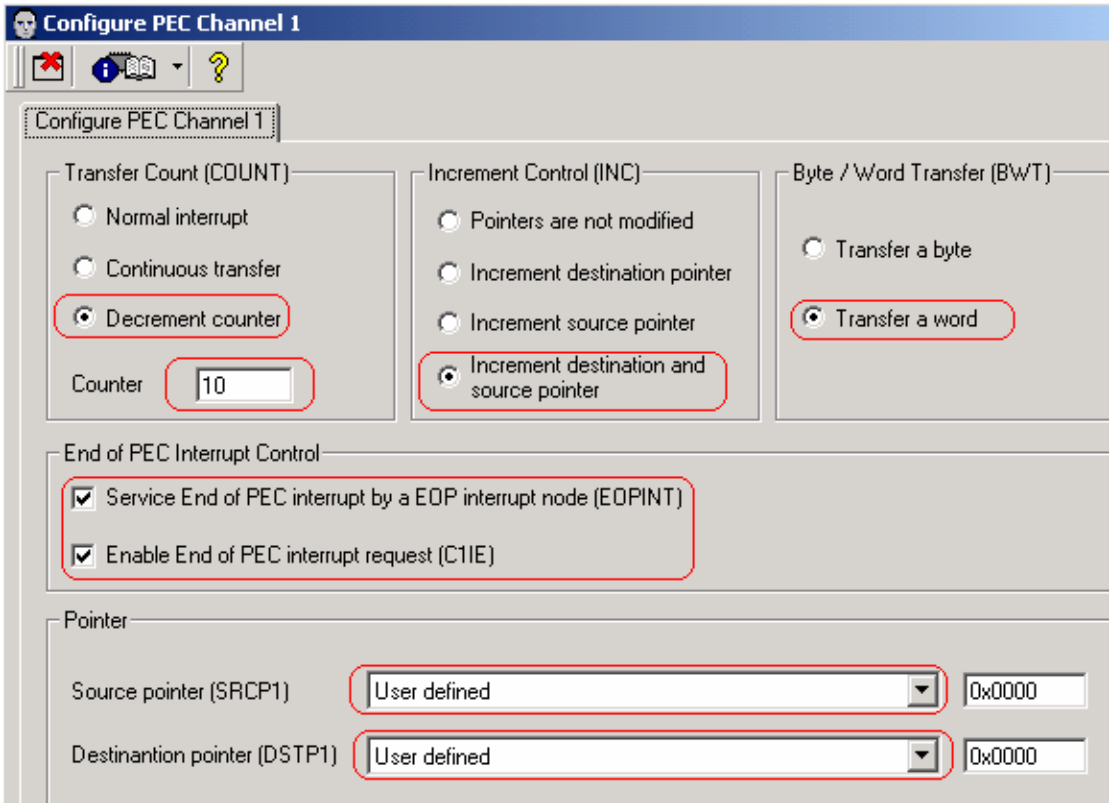


PEC 配置

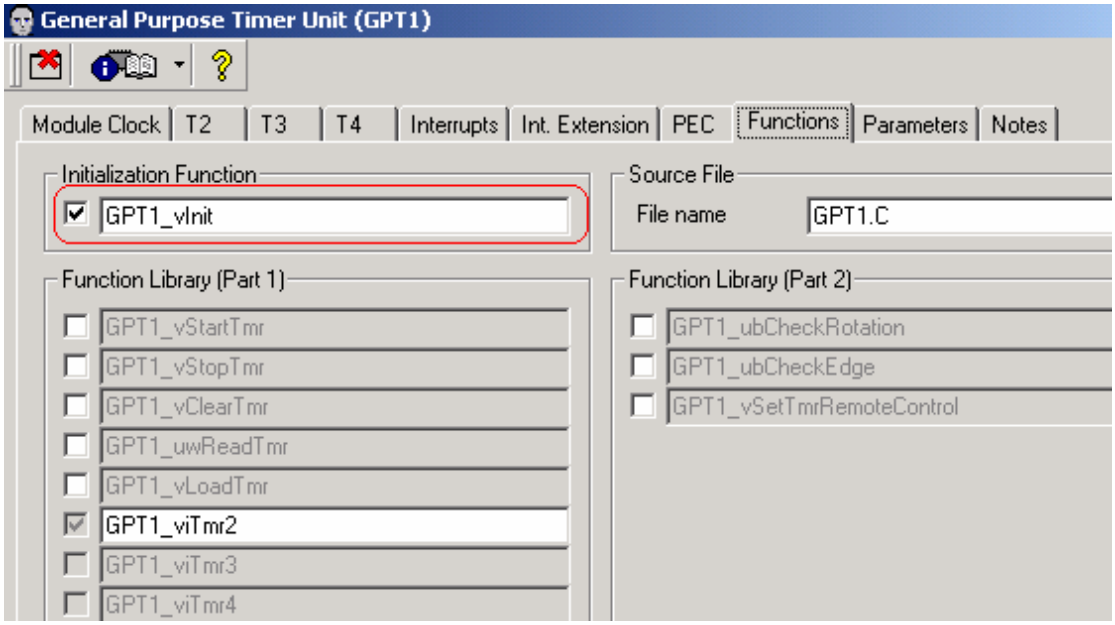
根据 T2 定时中断的优先级，选择通道 1 的中断源为 GPT1 T2INT。



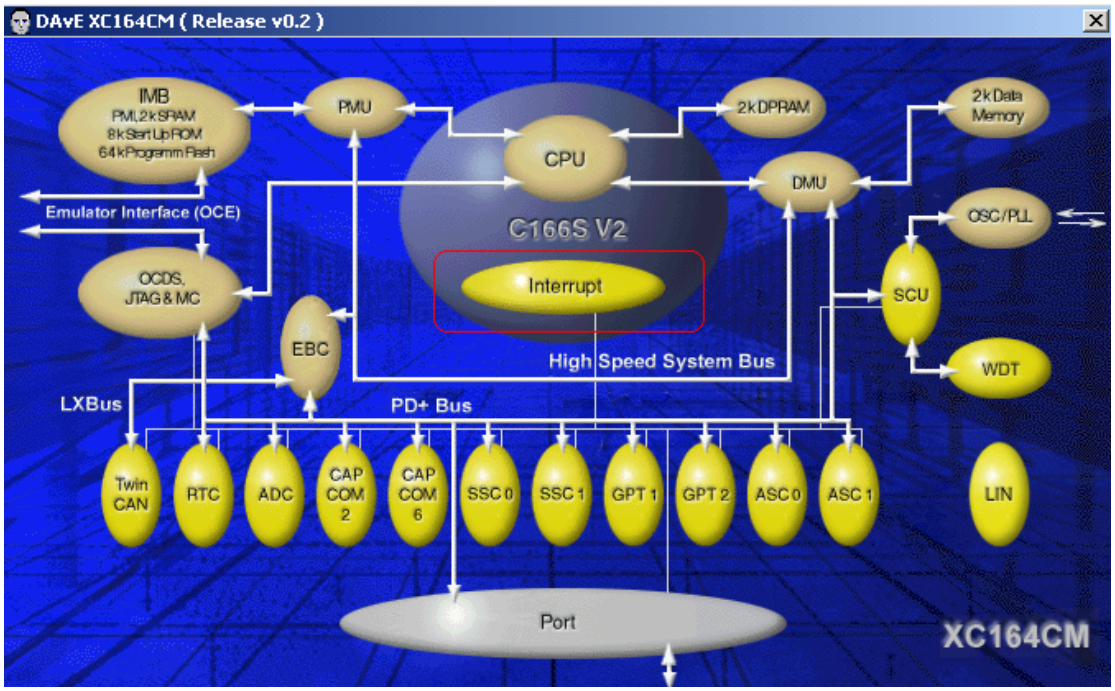
配置 PEC 通道 1
Decrement Counter = 10, increment destination and source point. Transfer a word
使能 PEC 中断， destination、source 地址由用户指定。



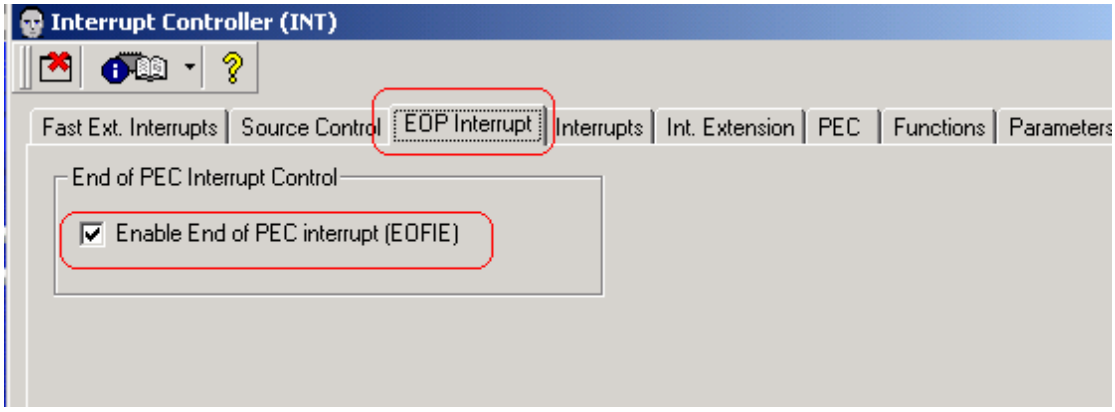
Functions
选中 initialization function 已生成 GPT1.c 文件



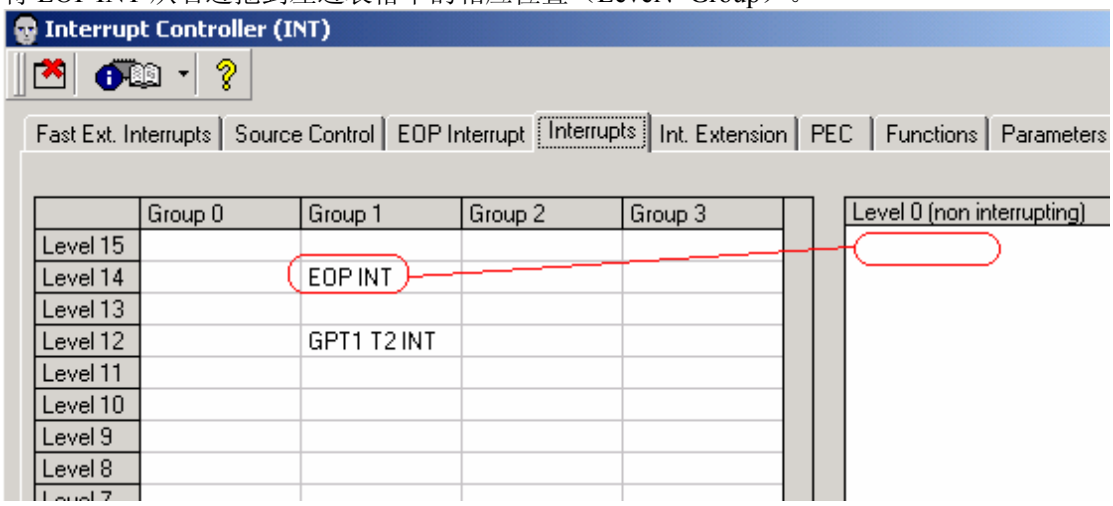
4．4　INT 配置，生成 End of PEC 中断。



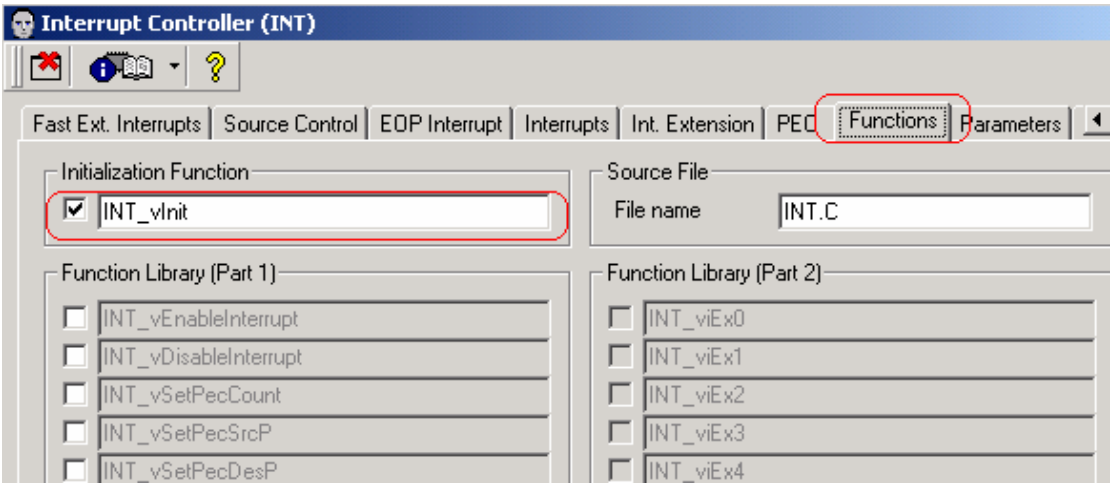
配置 EOP 中断



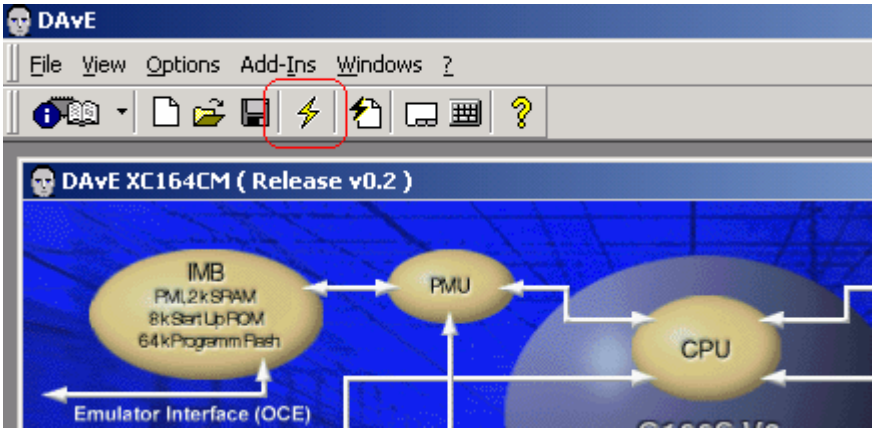
配置 EOP 中断的优先级
将 EOP INT 从右边拖到左边表格中的相应位置（Level、Group）。



Functions
选中 initialization function 以生成 Int.c 文件



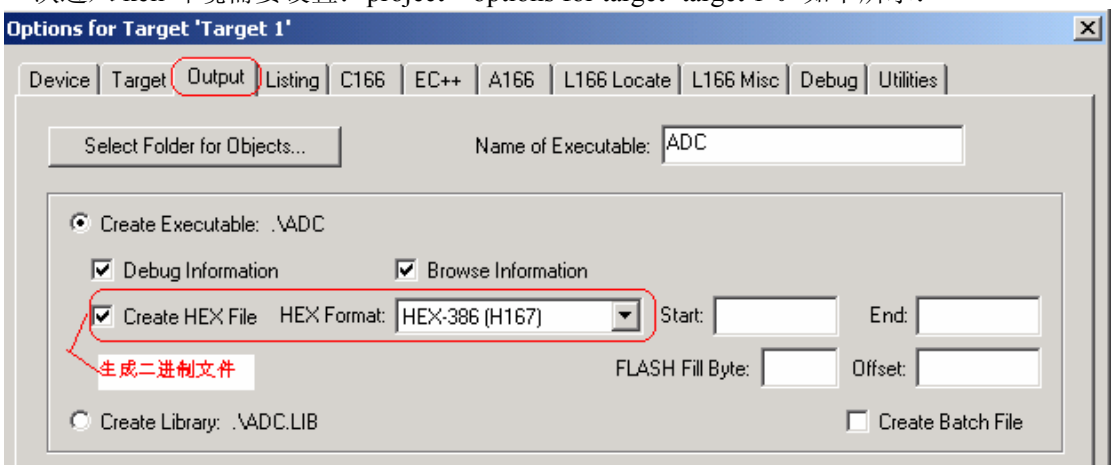
5　利用 DAVE 生成代码。



6．修改用户代码

6．1 生成 uVsion 工程文件。

做完以上步骤之后工程文件夹中会出现 keil 图标的 dpt 文件，双击进入 keil 环境。第一次进入 keil 环境需要设置：project－options for target ‘target 1’。如下所示：



6．2　修改 main.c

添加 while(1);

```
void main(void)
{
  // USER CODE BEGIN (Main,2)

  // USER CODE END

  MAIN_vInit();

  // USER CODE BEGIN (Main,4)
  while(1);  // 添加 while(1)。
  // USER CODE END。
```

6．3　修改 GPT1.c

添加定义变量

```
unsigned int PEC_Source[10] = {0,1,2,3,4,5,6,7,8,9};  // 源数组
unsigned int PEC_Des[10];  // 目标数组
```

在 GPT1_vInit() 中添加 PEC 的源地址、目标地址

```
  // USER CODE BEGIN (GPT1_Function,3)
  // manual define the source and destination address
  SRCP1  =  _sof_(PEC_Source);      //set source pointer
  DSTP1  =  _sof_(PEC_Des);      //set destination pointer
  // USER CODE END
```

6．4　修改 INT.c

添加引用外部变量声明：

```
extern unsigned int PEC_Des[10];
extern unsigned int PEC_Source[10];
```

修改 void INT_vInit(void)

添加 EOP 中断时的相关处理，此时 PEC 传输 10 次完毕，所有源数组的值被传到目标数据的指定位置。

在中断处理中清除目标数组中的值，重置 PEC 控制寄存器。

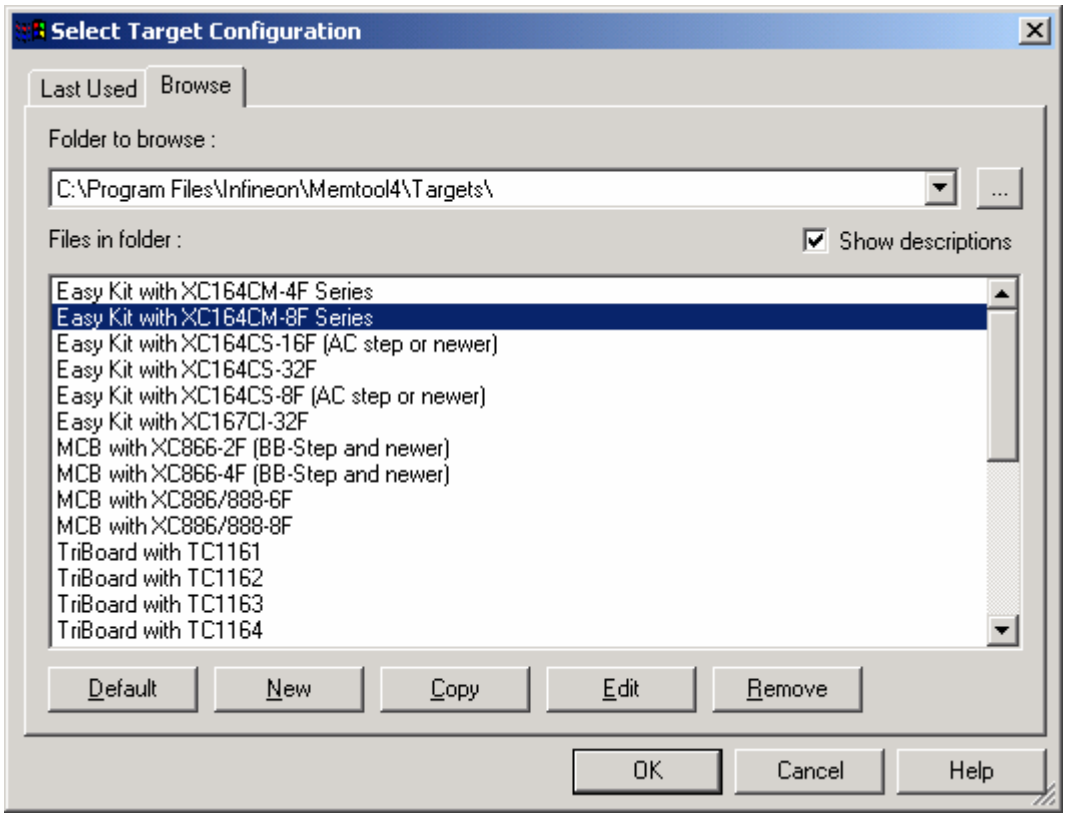
```
if(PECISNC_C1IR)   // end of PEC channel 1
 {
  PECISNC_C1IR = 0;

  // USER CODE BEGIN (EOP,4)
        //清除目标数组的值，
        for(i=0;i<10;i++)
                PEC_Des[i] = 0;

        //  重值 PEC 通道的源地址、目标地址
        SRCP1  =  _sof_(PEC_Source);     //set source pointer
        DSTP1  =  _sof_(PEC_Des);     //set destination pointer

        //重置 PEC 控制器的值，重新使能 PEC
        // re-enable the PEC
        PECC1        =  0x560A;
  // USER CODE END

 }
```

7．编译

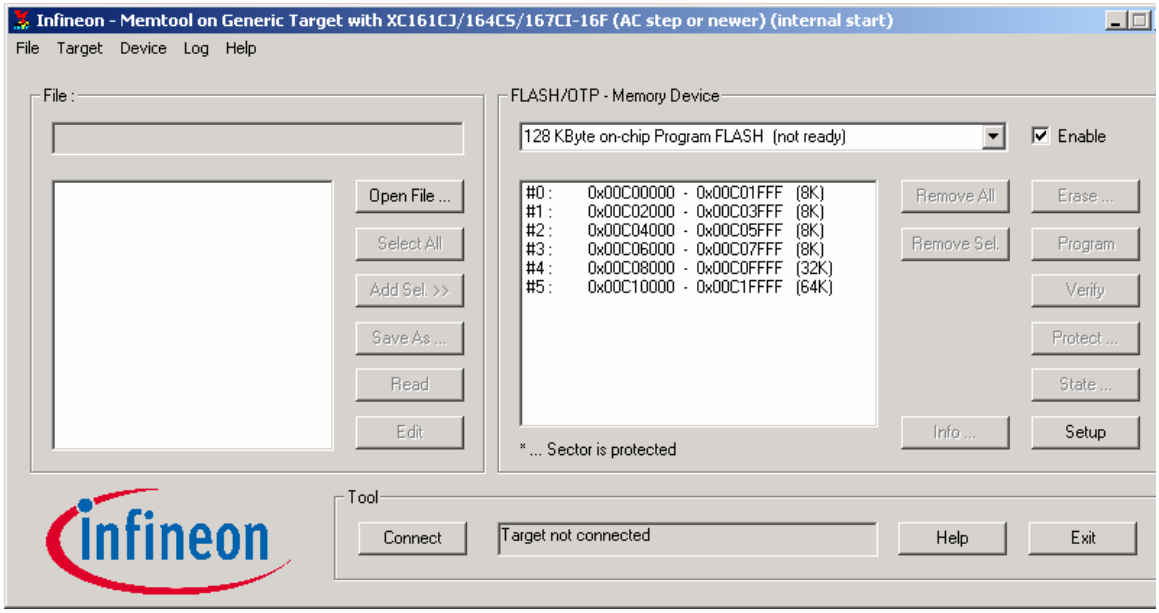
点击  图标进行编译连接。如有错误进行更改，直到出现‘0 Errors found.’。

8．下载

利用 memtool 软件将上面生成的 h86 文件下载到单片机。打开 memtool 软件，点击菜单 Targe－Change，选择 XC164CM-8F。界面如下：



点击 OK 出现如下对话框。



点击‘connect’进行通讯连接。通讯成功之后，按照顺序 open file…－select all－add sel.>>将 h86 文件添加到右边框中，然后选择’Erase…’和’Program’进行擦除、编程。如有必要可点击’Verify’进行校验。

9．运行