

## 样例程序：RTC 模块使用

### 1. 简介：

本程序实现功能如下：

实现 RTC 基本功能，每 2 秒触发中断，Toggle P1L.1，每 10s 触发中断，Toggle P1L.0。  
涉及到的模块有：RTC，I/O（用于配置 P1L.0，P1L.1 引脚为通用 I/O，且输出）。

### 2. RTC 介绍

RTC 实时时钟，主要用于实现长时间计时。模块包括可选 8:1 分频器，一个定时器 T14，及一个 32 位 RTC 计数器。

特点：

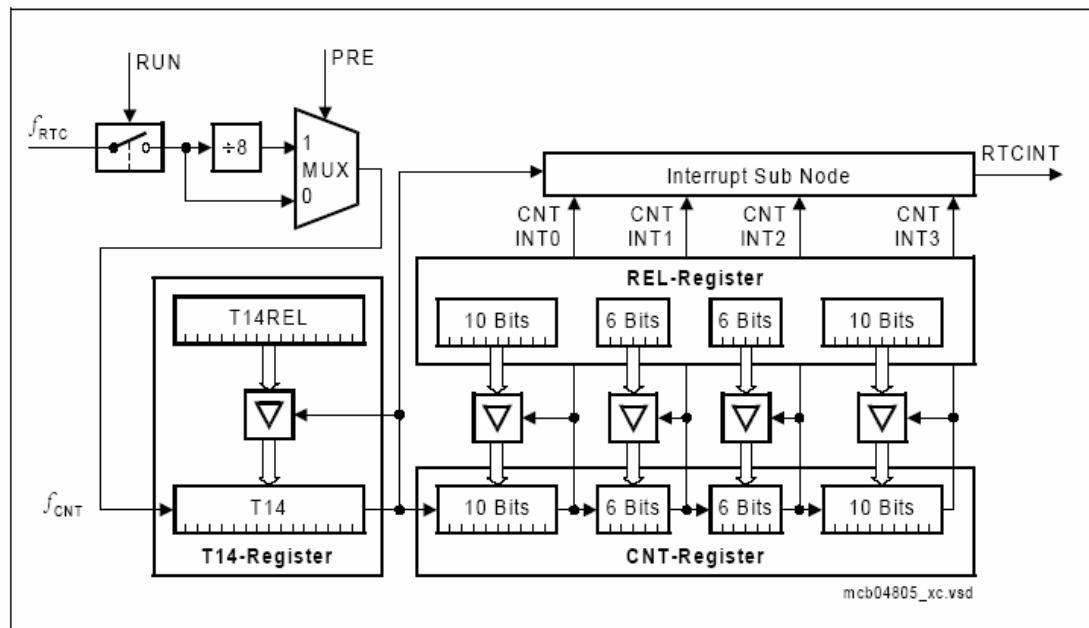
- 48 位定时功能实现长时间定时。
- 内部结构支持时间、日期的计算。
- 可分为 5 个定时器，且有相应的中断。

32 位 RTC 定时器可分为：

- 10-bit 定时器 CNT0。
- 6-bit 定时器 CNT1。
- 6-bit 定时器 CNT2。
- 10-bit 定时器 CNT3。

计数器均向上计数，且有对应的中断。

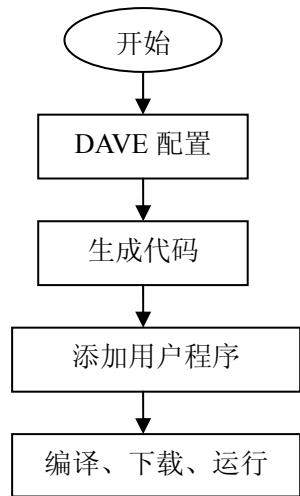
T14、CNT0、CNT1、CNT2、CNT3 的关系如下：



T14 定时中断信号作为 CNT0 的计数时钟，CNT0 的中断信号作为 CNT1 的计数时钟，  
CNT1 的中断信号作为 CNT2 的计数时钟，CNT2 的中断信号作为 CNT3 的计数时钟。

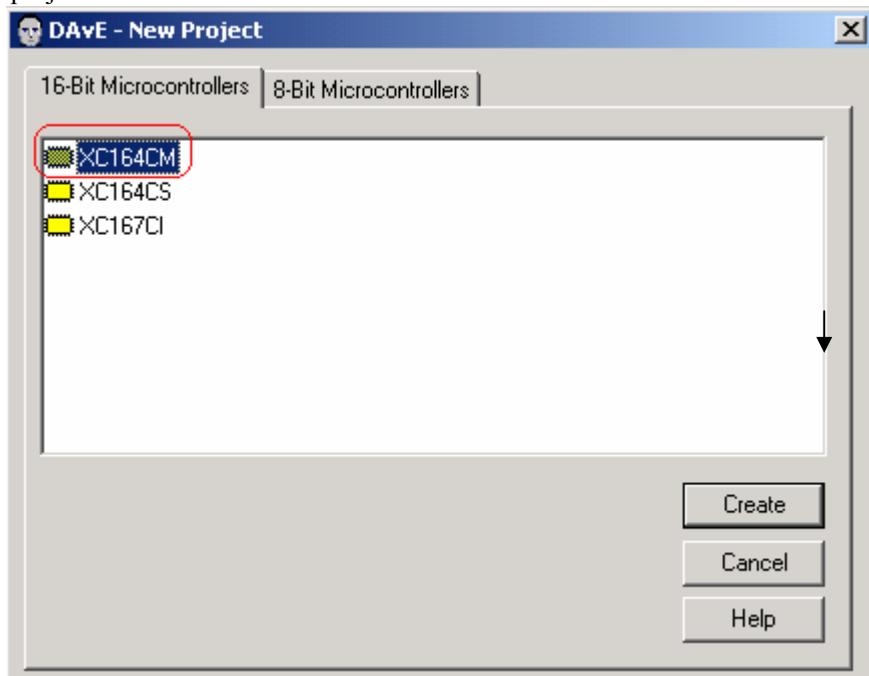
通过设置 CNT0、CNT1、CNT2、CNT3 来实现不同的定时，RTC 支持超过 100 年的定时。

### 3. 操作流程：

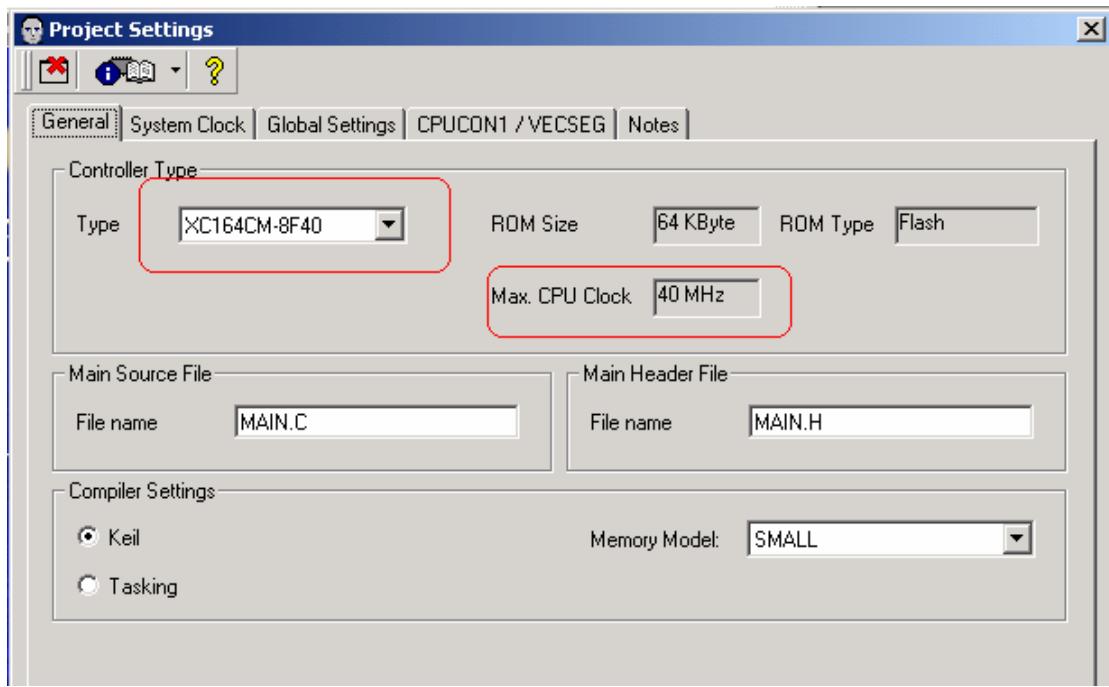


#### 4. DAvE 配置

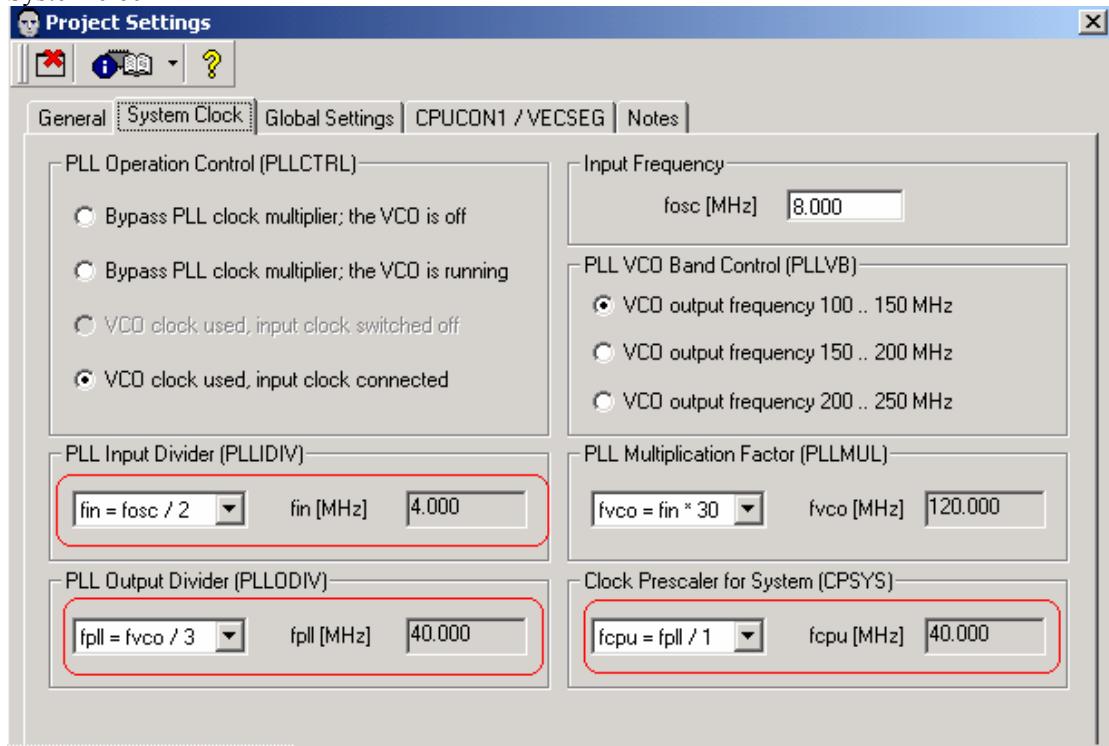
4. 1 New project: select XC164cm



4. 2 The project settings

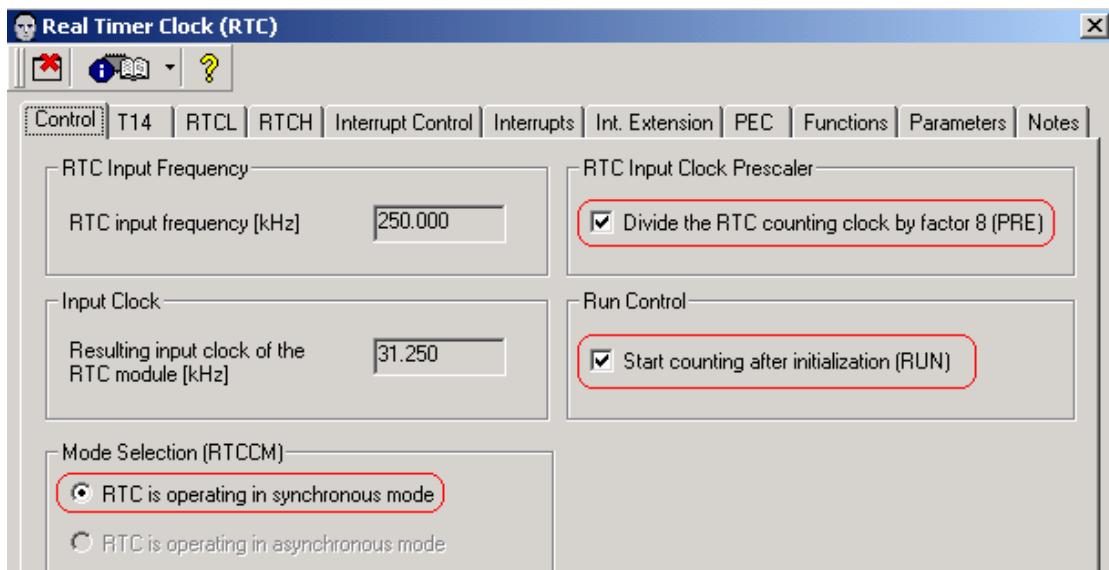


#### System clock

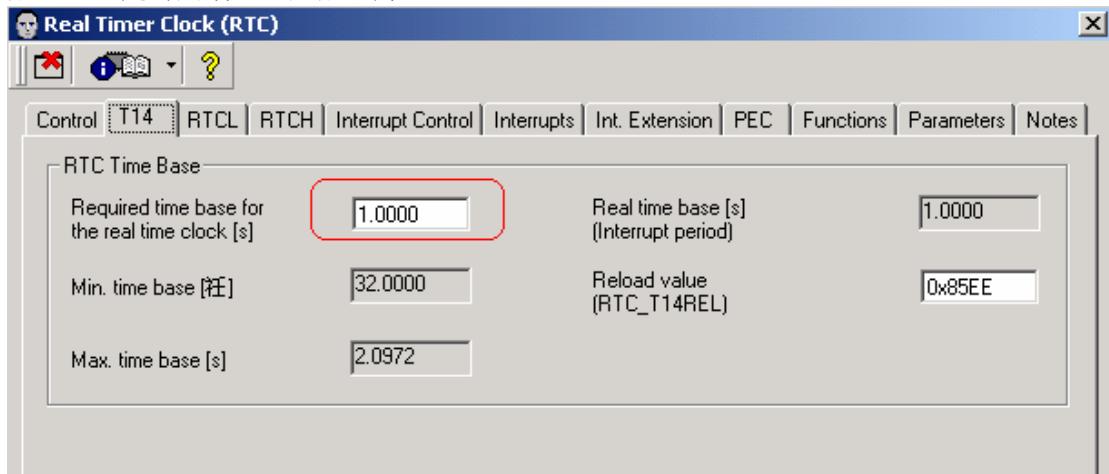


#### 4. 3 RTC 配置

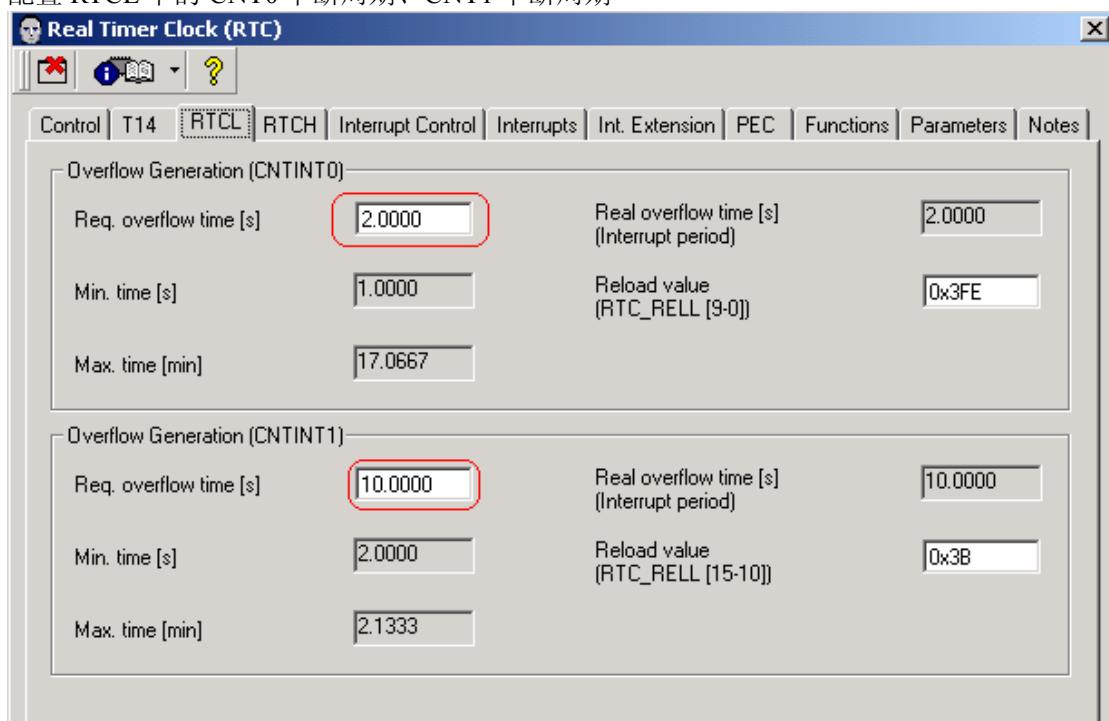
选择分频系数、运行使能。



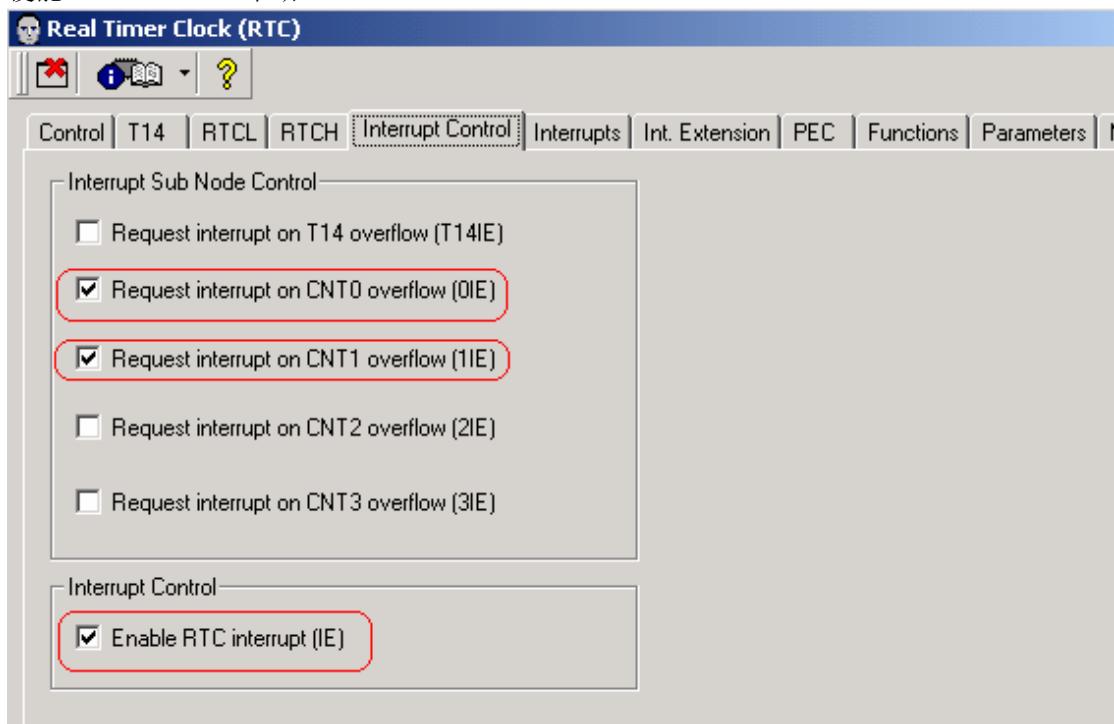
配置 T14 定时周期，此处配置为 1s。



配置 RTCL 中的 CNT0 中断周期、CNT1 中断周期

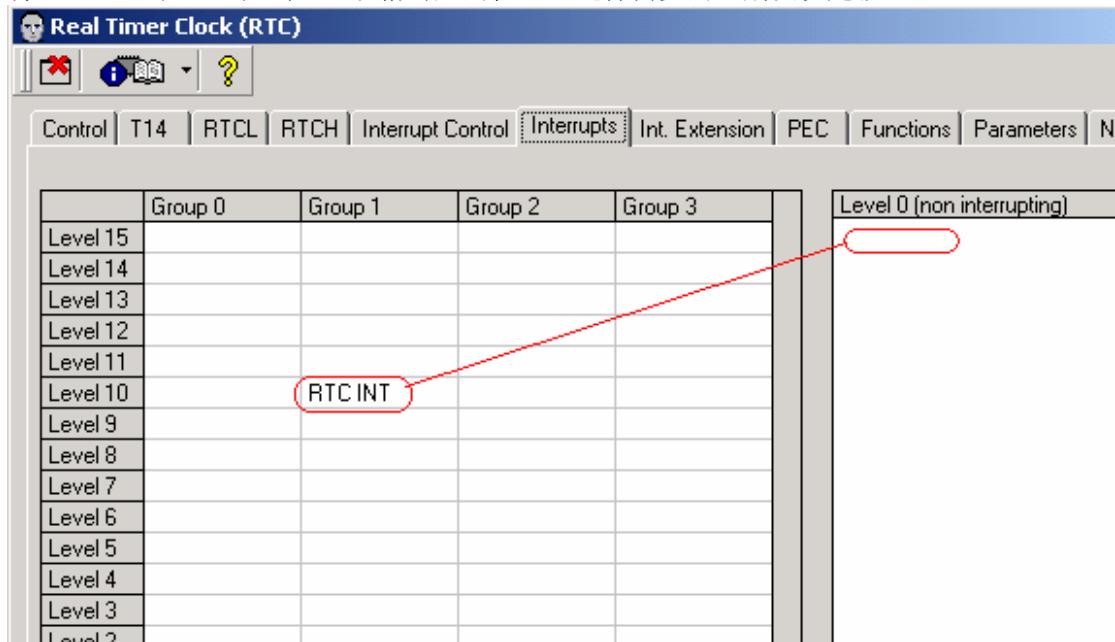


使能 CNT0、CNT1 中断。

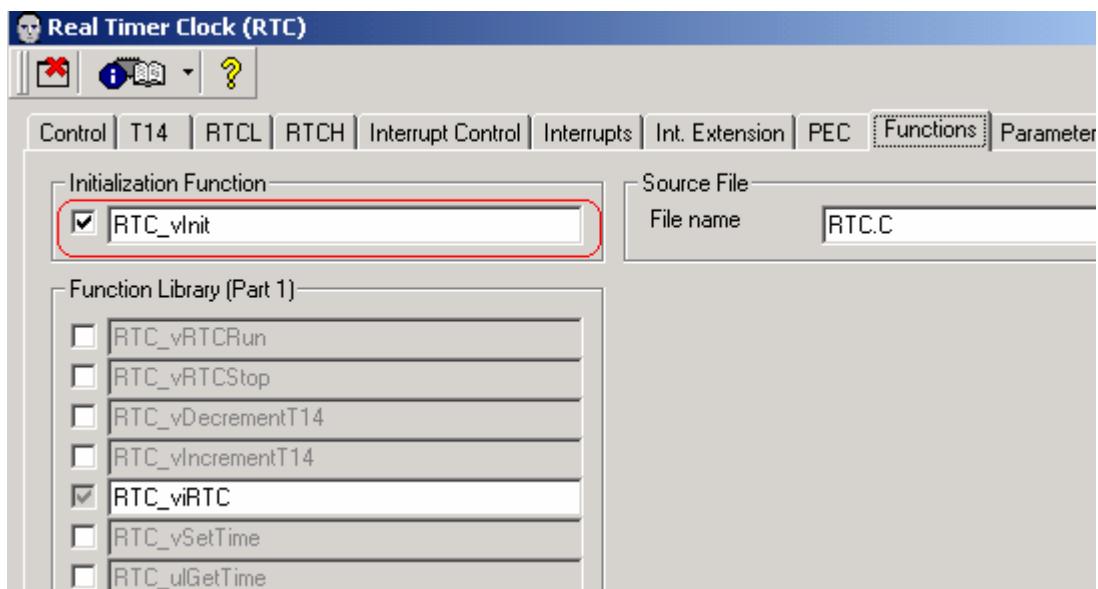


配置 RTC 中断优先级：

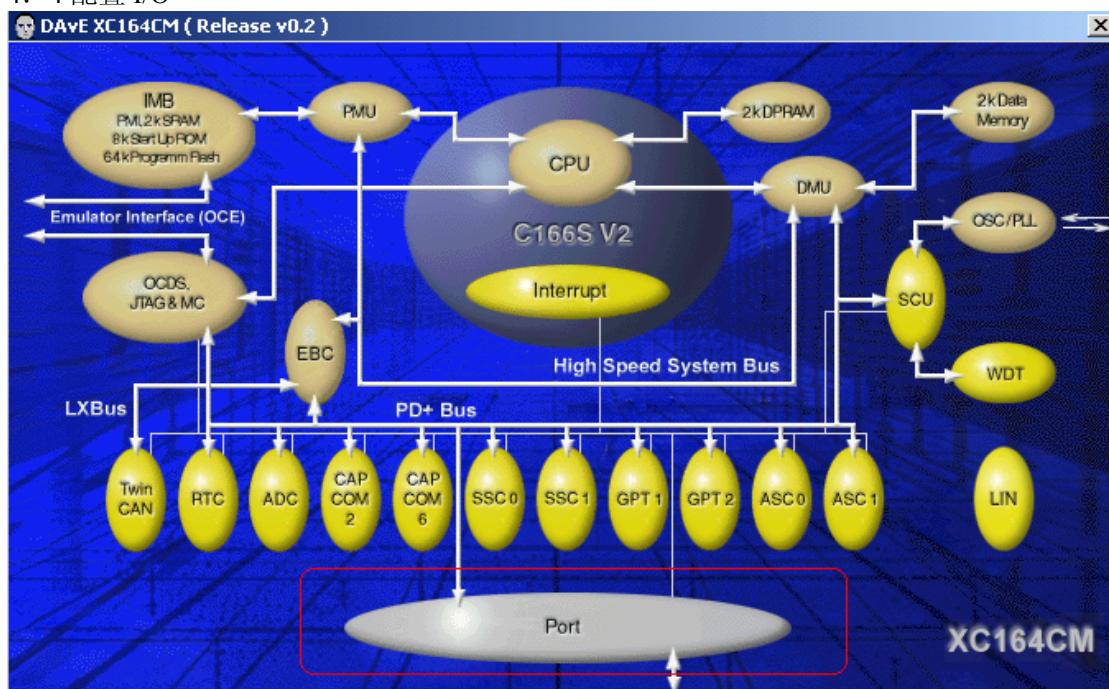
将 RTC INT 从右边拖到左边表格对应的位置。选择需要的组别及优先级。



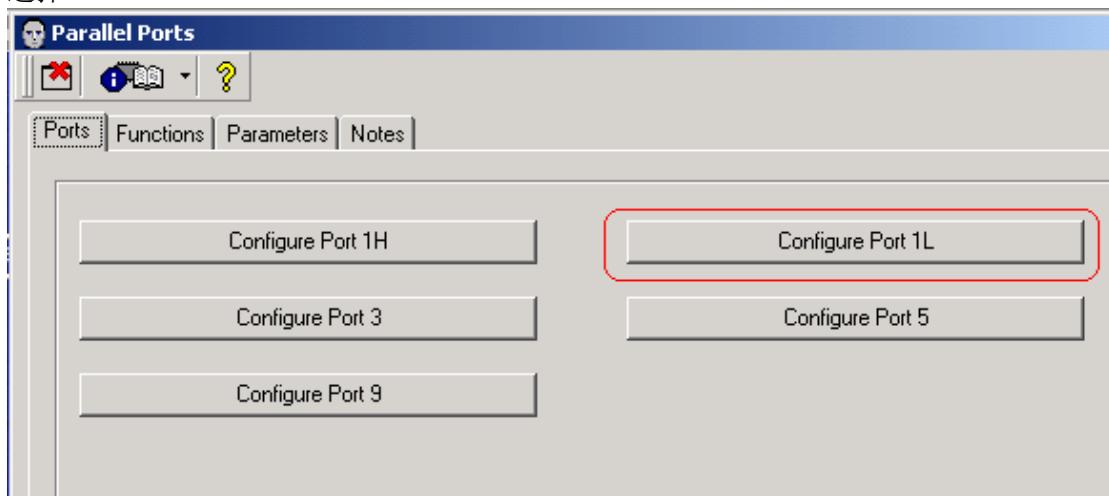
在 Functions 选择 RTC\_vInit 以生成 RTC.c 文件。



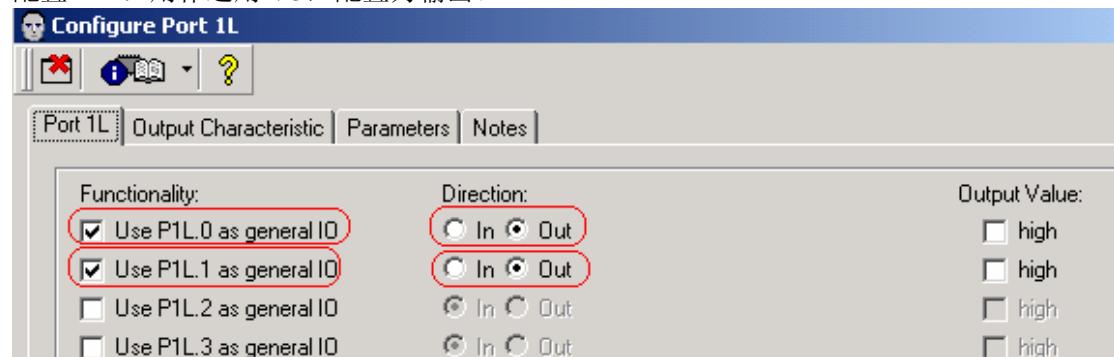
#### 4. 4 配置 I/O



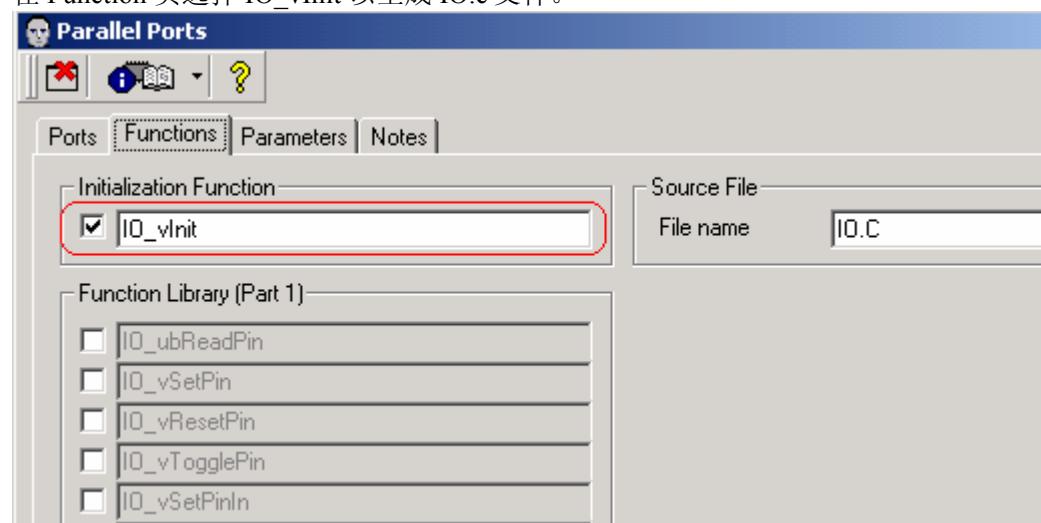
选择 P1L



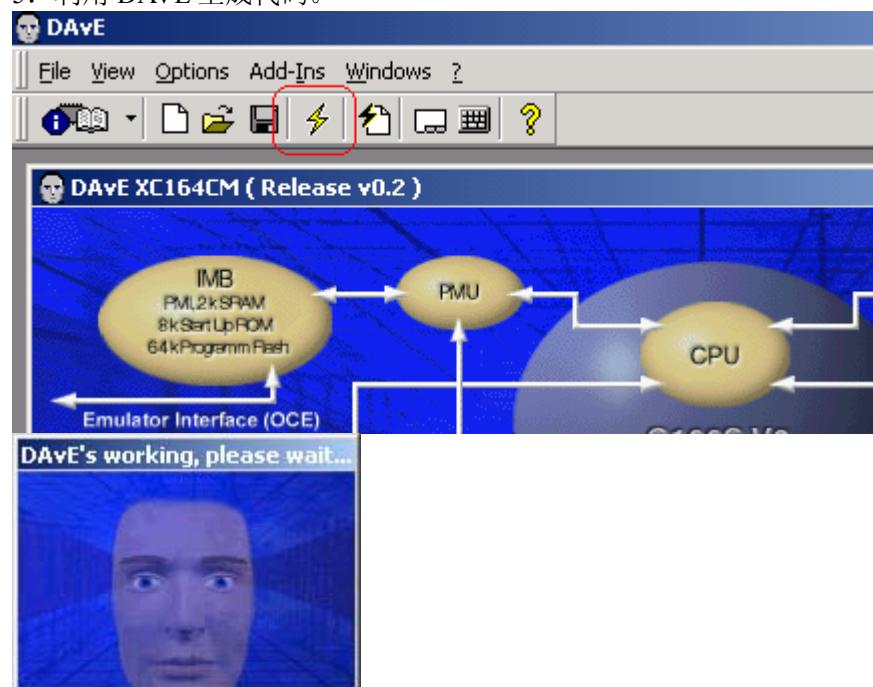
配置 P1L: 用作通用 I/O, 配置为输出。



在 Function 页选择 IO\_vInit 以生成 IO.c 文件。



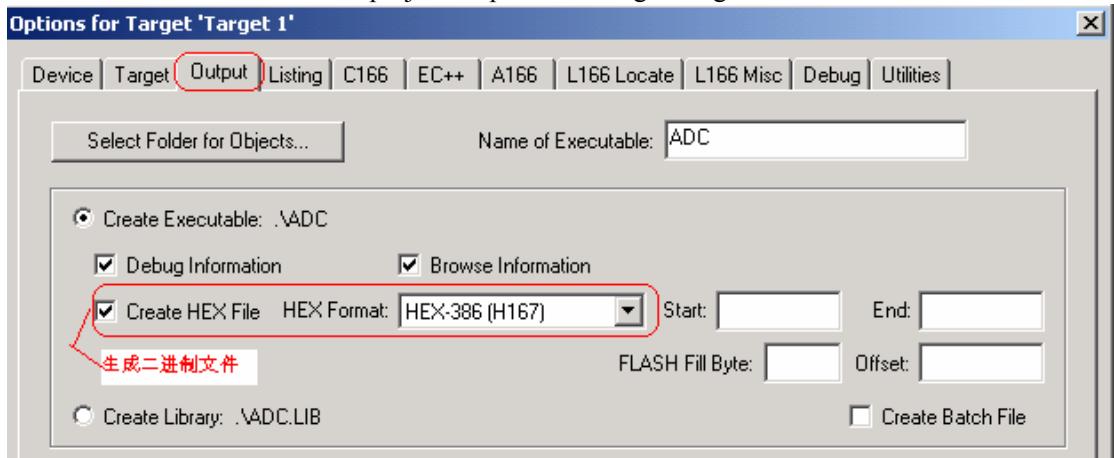
5. 利用 DAVE 生成代码。



6. 修改用户代码

6. 1 生成 uVision 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

在下列位置添加 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 RTC.c

在 void RTC\_v RTC(void) interrupt RTCINT 中修改。

添加 CNT0 中断时 Toggle P1L\_1，中断周期 2s。

```
if(RTC_ISNC_CNT0IR)      // if counter CNT0 overflow
{
    // USER CODE BEGIN (RTC,4)
    IO_P1L_1 = !IO_P1L_1; // Toggle P1L_1;
    // USER CODE END

    RTC_ISNC_CNT0IR = 0;
}
```

添加 CNT1 中断时 Toggle P1L\_0，中断周期 10s。

```
if(RTC_ISNC_CNT1IR)      // if counter CNT1 overflow
{
    // USER CODE BEGIN (RTC,5)
    IO_P1L_0 = !IO_P1L_0;
    // USER CODE END
```

```
RTC_ISNC_CNT1IR = 0;  
}
```

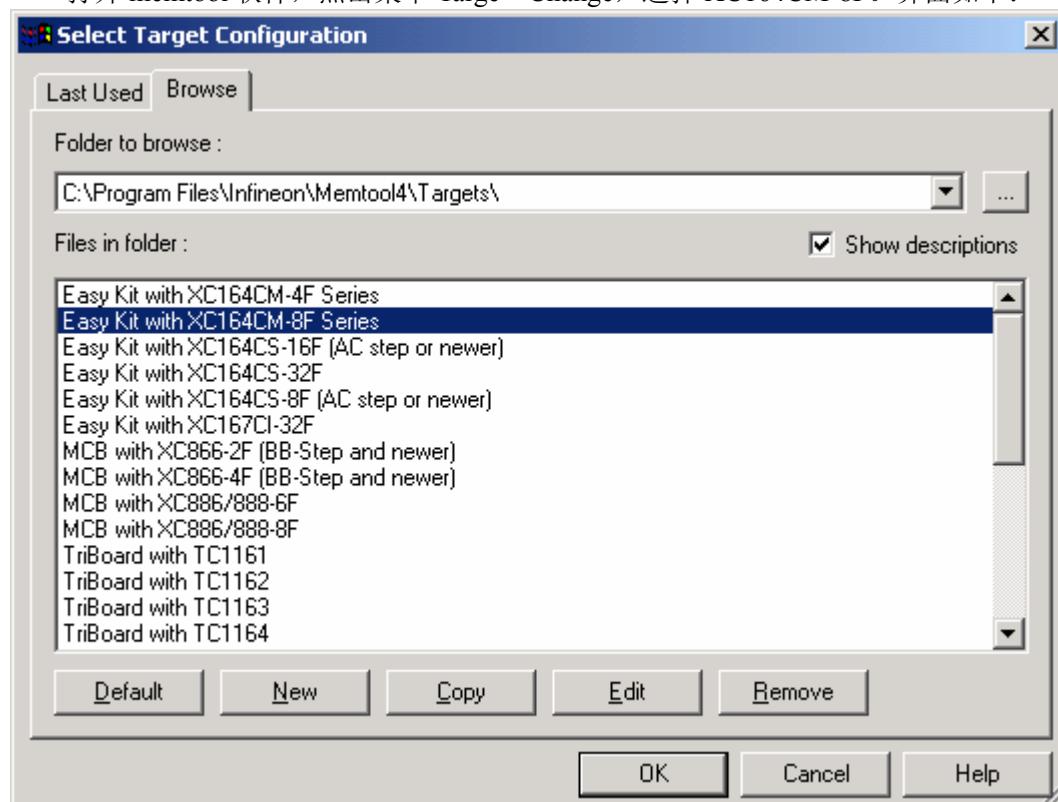
## 7. 编译

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

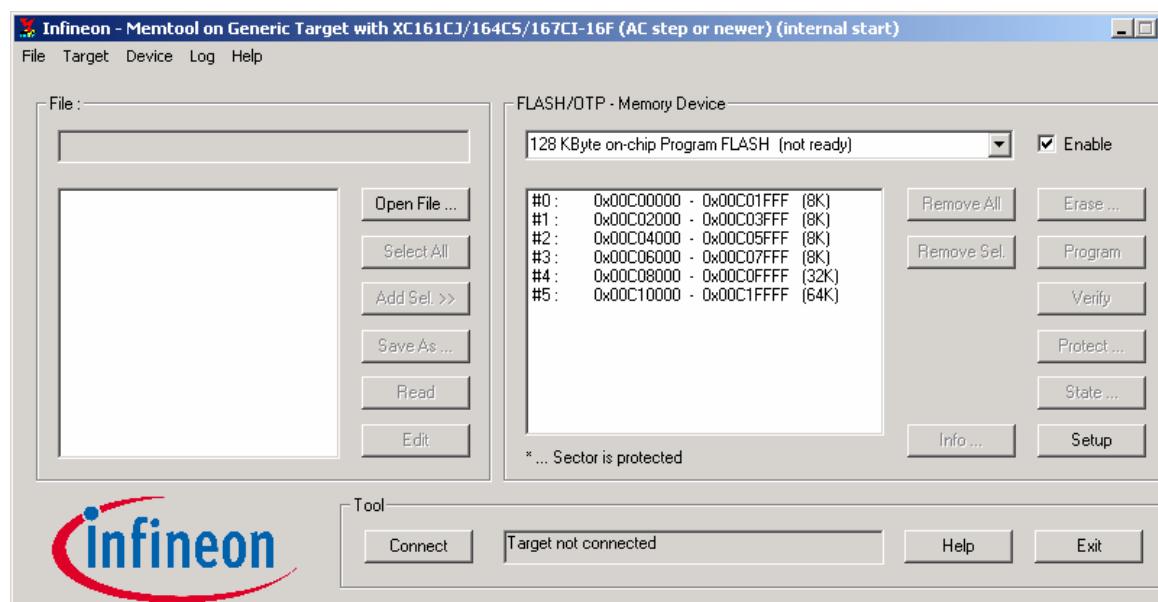
利用 memtool 软件将上面生成的 h86 文件下载到单片机。

打开 memtool 软件，点击菜单 Target—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

P1L\_0 Toggle 周期 10 s,P1L\_1 Toggle 周期 2s。波形如下所示：

