

样例程序：CAPCOM6 模块使用

1. 本程序实现的功能如下：

- 配置 CAPCOM6。
 - 使用 CC60,CC61,CC62,COUT60,COUT61,COUT62,COUT63 作为 PWM 输出。
 - 在 T2 定时中断更新各个通道的占空比。
- 使用到的模块：CAPCOM6、GPT1。

2. 模块介绍

2. 1 CAPCOM6

XC164CM 提供功能强大的 PWM 比较捕获单元 CCU6, 内部包含 2 路 PWM 定时器, T12 模块提供 3 对比较/捕捉通道, T13 提供 1 路 PWM 输出, 使 XC164CM 适合各种交流电机或逆变器的控制。内部集成适用于 BLDC 控制的霍尔信号、反电动势检测功能。此外还包括用于多项电机控制的块交换模式。

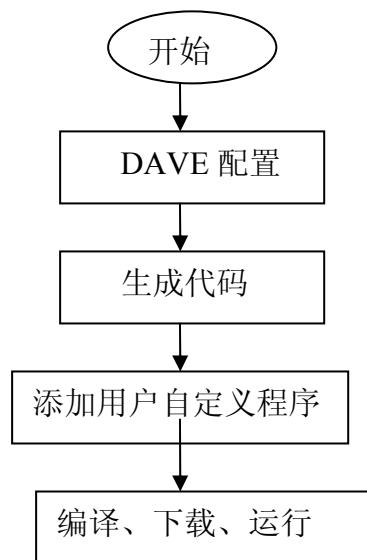
关于 CAPCOM6 模块功能的详细介绍, 请参照 XC164CM 用户手册。

2. 2 GPT1

XC164CM 提供了两个通用定时器模块：GPT1, GPT2。GPT1 内部有 3 个定时器（T2、T3、T4），GPT2 内部有 2 个定时器（T5, T6）。GPT1 内部 3 个定时器可以独立使用, 也可以配合使用, 可工作于定时器, 计数器, QEP 输入（正交编码器输入）。GPT2 内部的两个定时器可以独立使用也可以配合使用。具有灵活的工作模式。

关于 GPT1、GPT2 模块功能的详细介绍, 请参照 XC164CM 用户手册。

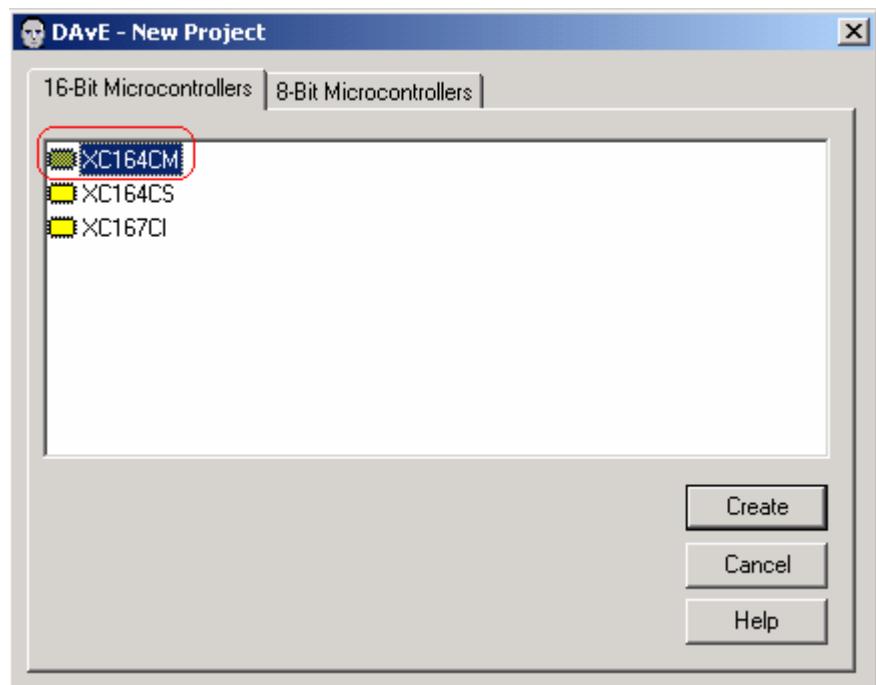
3. 操作流程



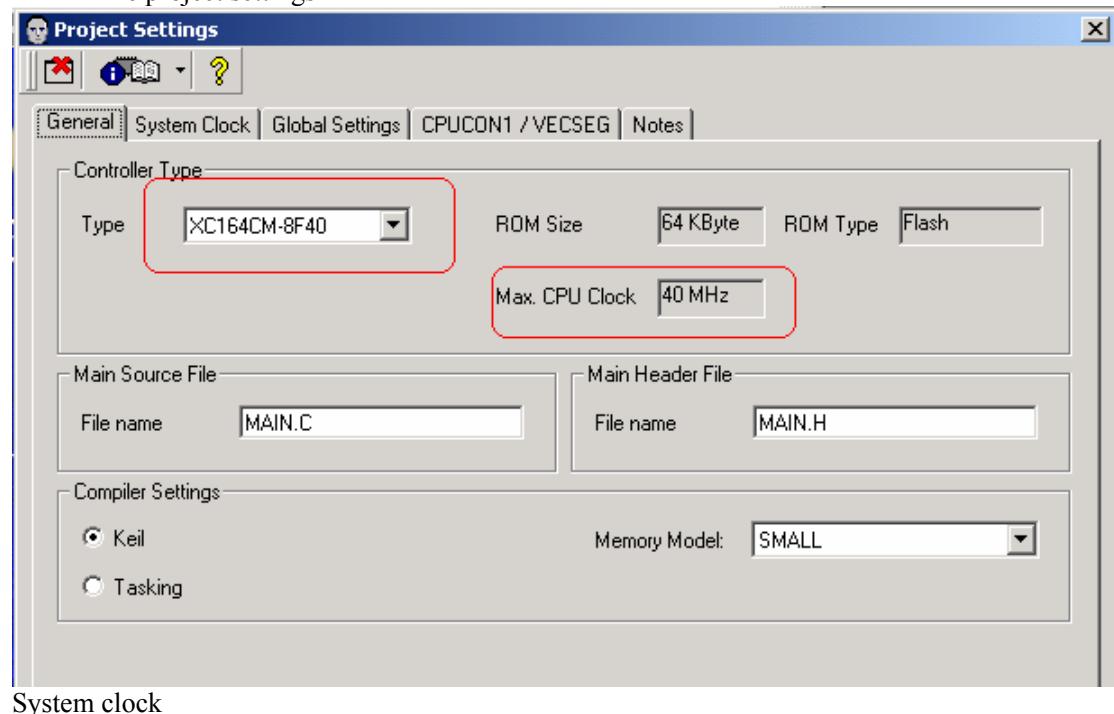
4. DAVE 配置

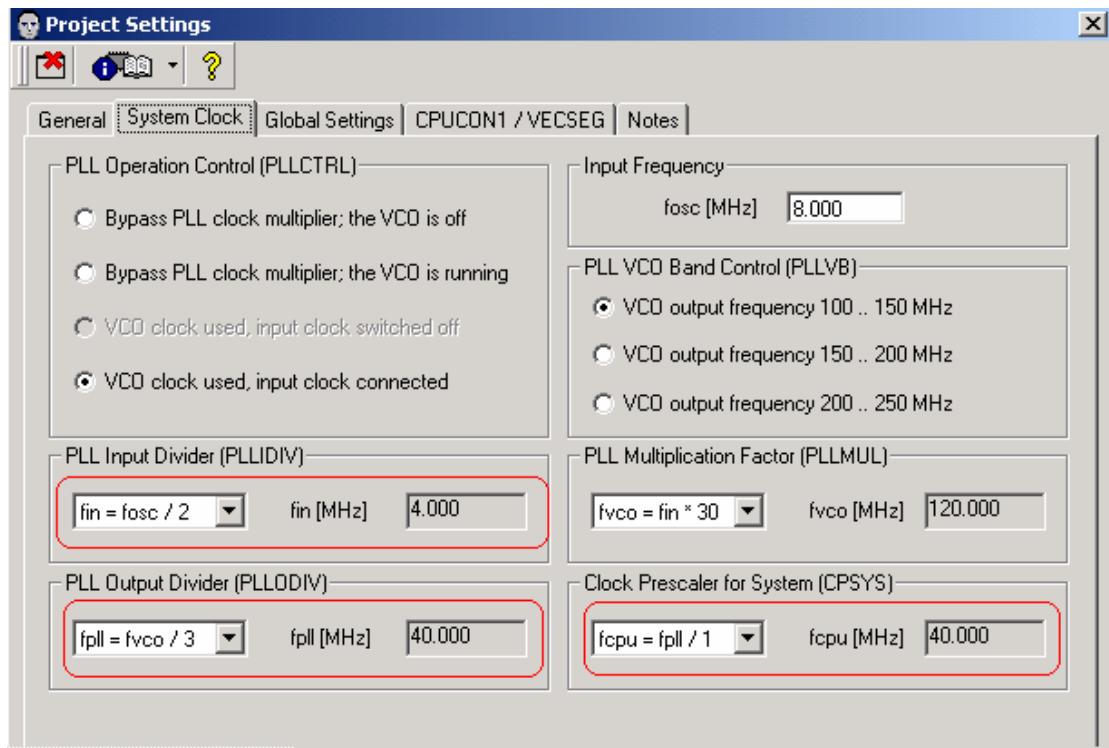
4. 1

New project: select XC164cm,



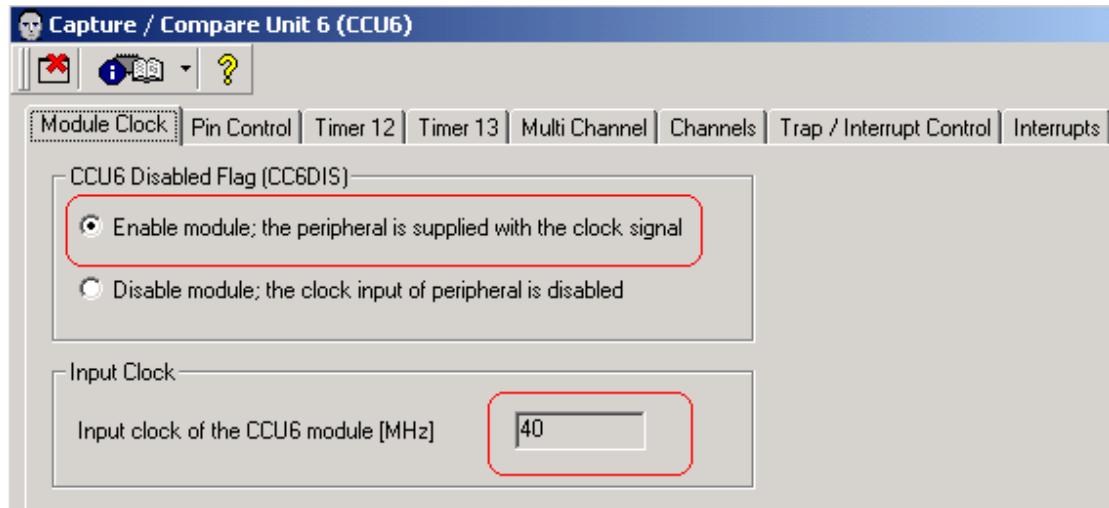
4. 2 The project settings





4. 3 配置 CAPCOM6

使能模块



配置 CC60、CC61、CC62、COUT60、COUT61、COUT62， COUT63 为输出

Capture / Compare Unit 6 (CCU6)

Module Clock | Pin Control | Timer 12 | Timer 13 | Multi Channel | Channels | Trap / Interrupt Control | Interrupts

Control of Pin CC60 (P1L.0)
 Pin CC60 is not used
 Use pin CC60 as input
 Use pin CC60 as output

Control of Pin CC61 (P1L.2)
 Pin CC61 is not used
 Use pin CC61 as input
 Use pin CC61 as output

Control of Pin CC62 (P1L.4)
 Pin CC62 is not used
 Use pin CC62 as input
 Use pin CC62 as output

Control of Pin COUT63 (P1L.6)
 Use pin COUT63 as output

Control of Pin COUT60 (P1L.1)
 Use pin COUT60 as output

Control of Pin COUT61 (P1L.3)
 Use pin COUT61 as output

Control of Pin COUT62 (P1L.5)
 Use pin COUT62 as output

Control of Pin #CC6POS0 (P1H.0)
 Use pin #CC6POS0 as input

Control of Pin #CC6POS1 (P1H.1)
 Use pin #CC6POS1 as input

Control of Pin #CC6POS2 (P1H.2)
 Use pin #CC6POS2 as input

配置 T12:

Capture / Compare Unit 6 (CCU6)

Module Clock | Pin Control | Timer 12 | Timer 13 | Multi Channel | Channels | Trap / Interrupt Control | Interrupts

Input Selection (T12CLK)
Input selection: fcpu / 1 (Resolution: 0.025 纳秒)

Timer 12 Start Control
 Start T12 after initialization (T12RS)

Timer 12 Single Shot Control
 Enable single shot mode (T12SSC)

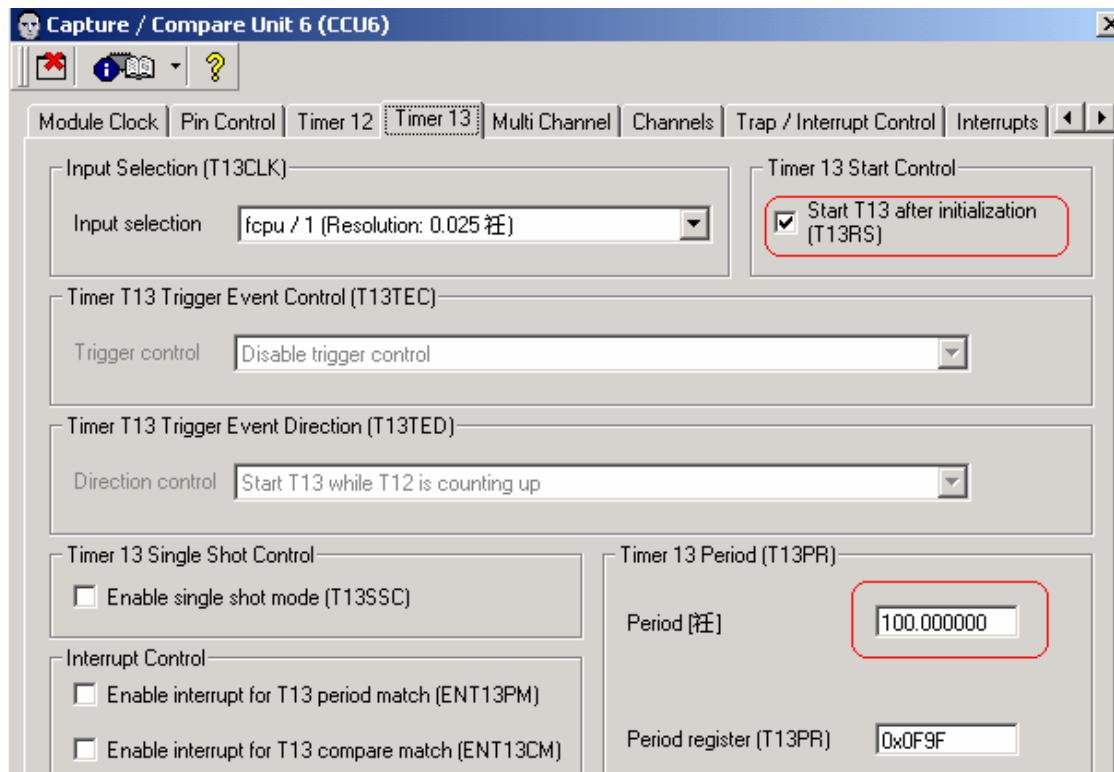
T12 Operating Mode (CTM)
 Edge aligned mode: count up
 Center aligned mode: count up/down

Interrupt Control
 Enable interrupt for T12 period match (ENT12PM)
 Enable interrupt for T12 one match (ENT12OM)

Timer 12 Period (T12PR)
Period [纳秒]: 100.000000
Period register (T12PR): 0x07CF

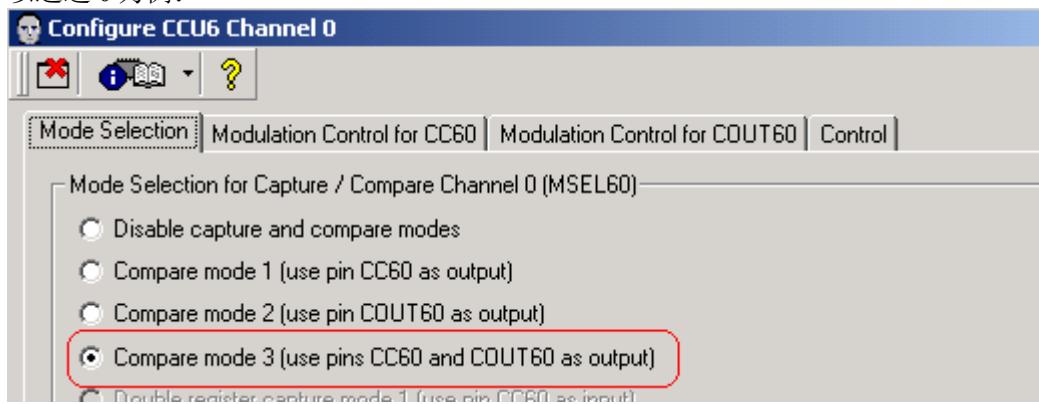
Dead Time Control
Dead time [ns]: 25.000
Dead time register (DTM): 0x01

配置 T13:

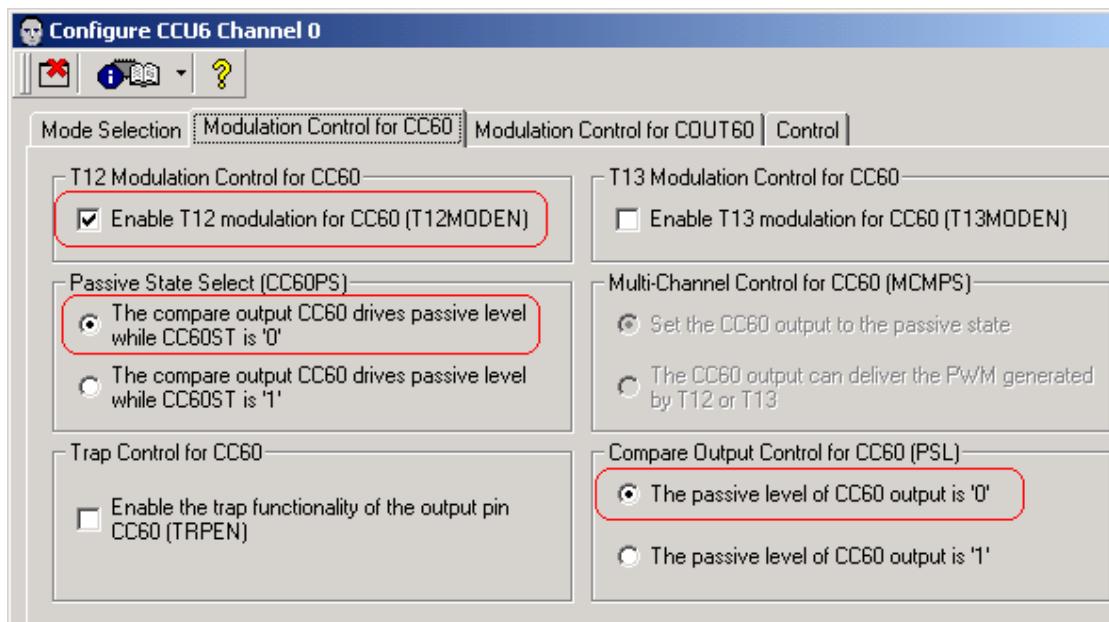


配置各个通道

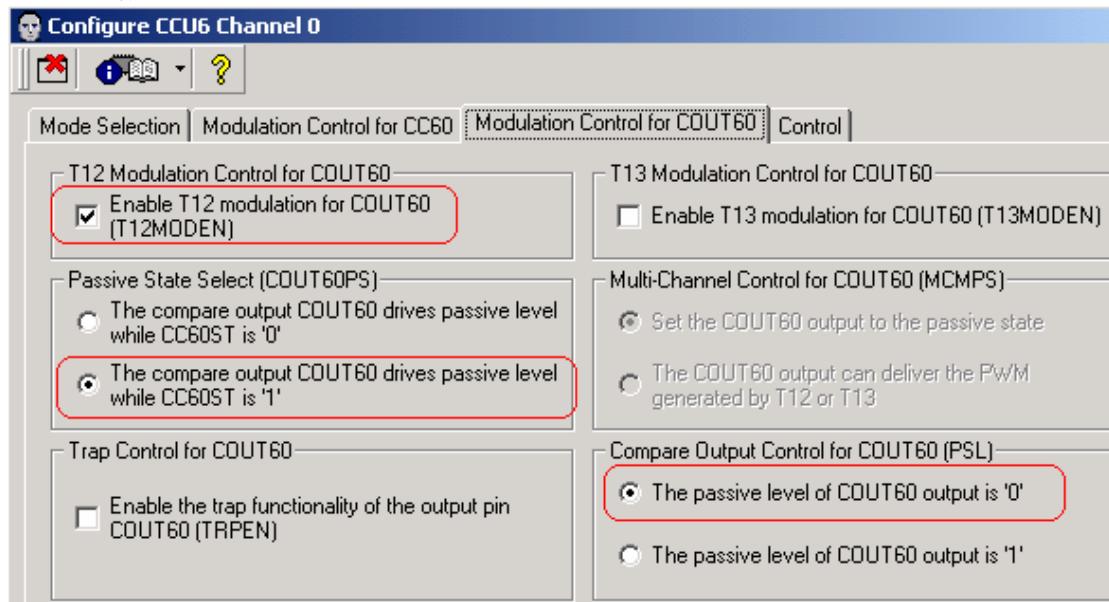
选择 compare mode 3, T12 modulation, deadtime generation, duty cycle=50%,
以通道 0 为例:



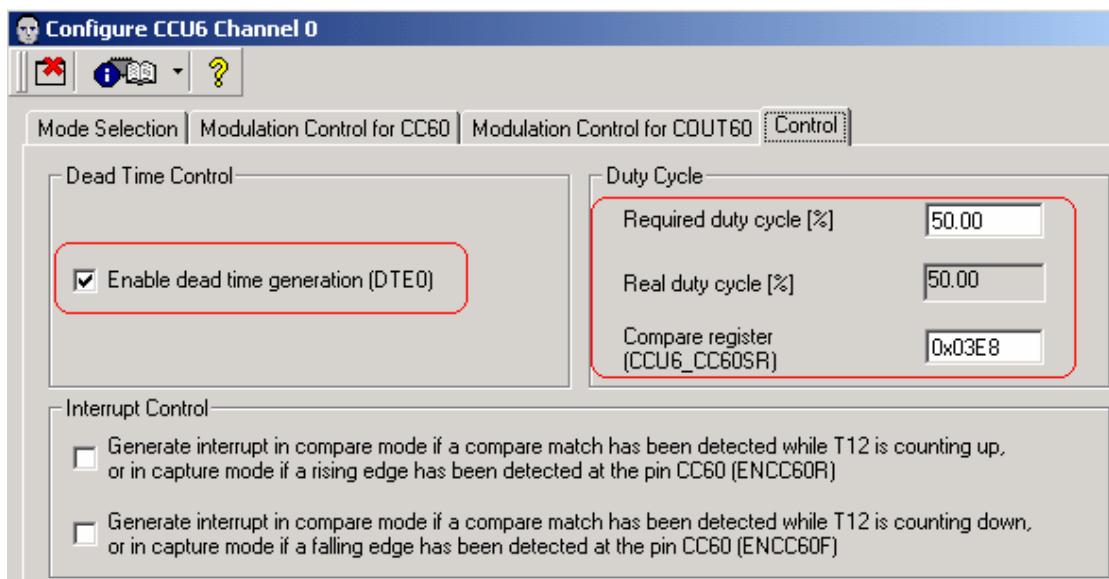
CC60 配置



CC60 配置

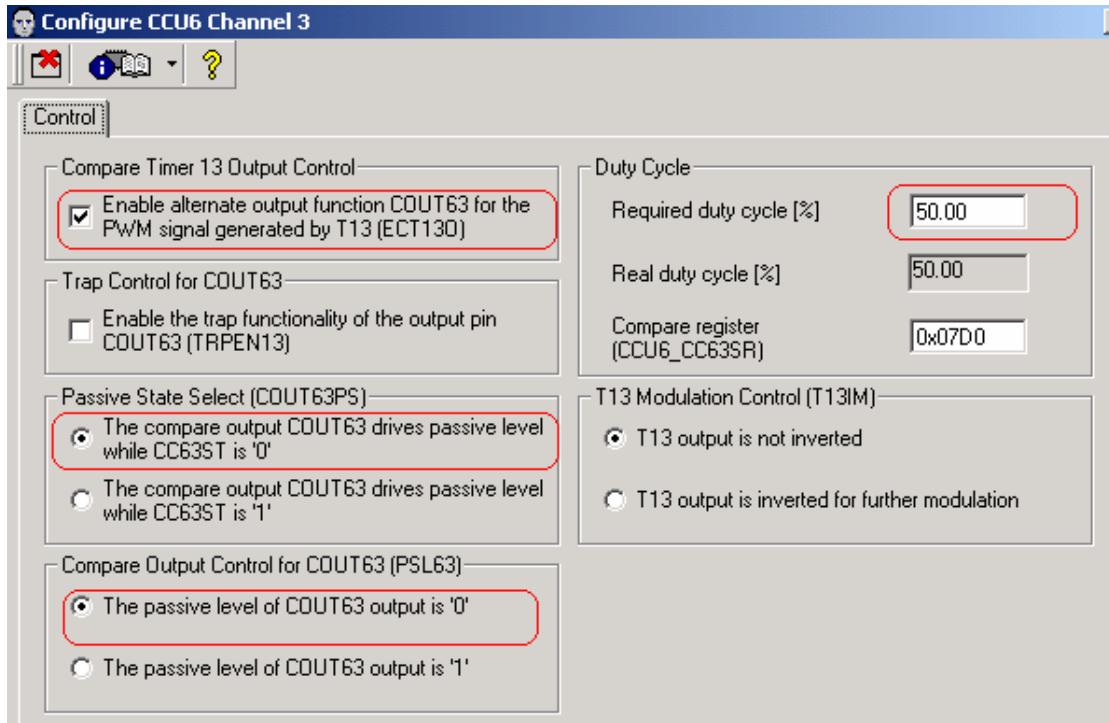


占空比及死区时间配置

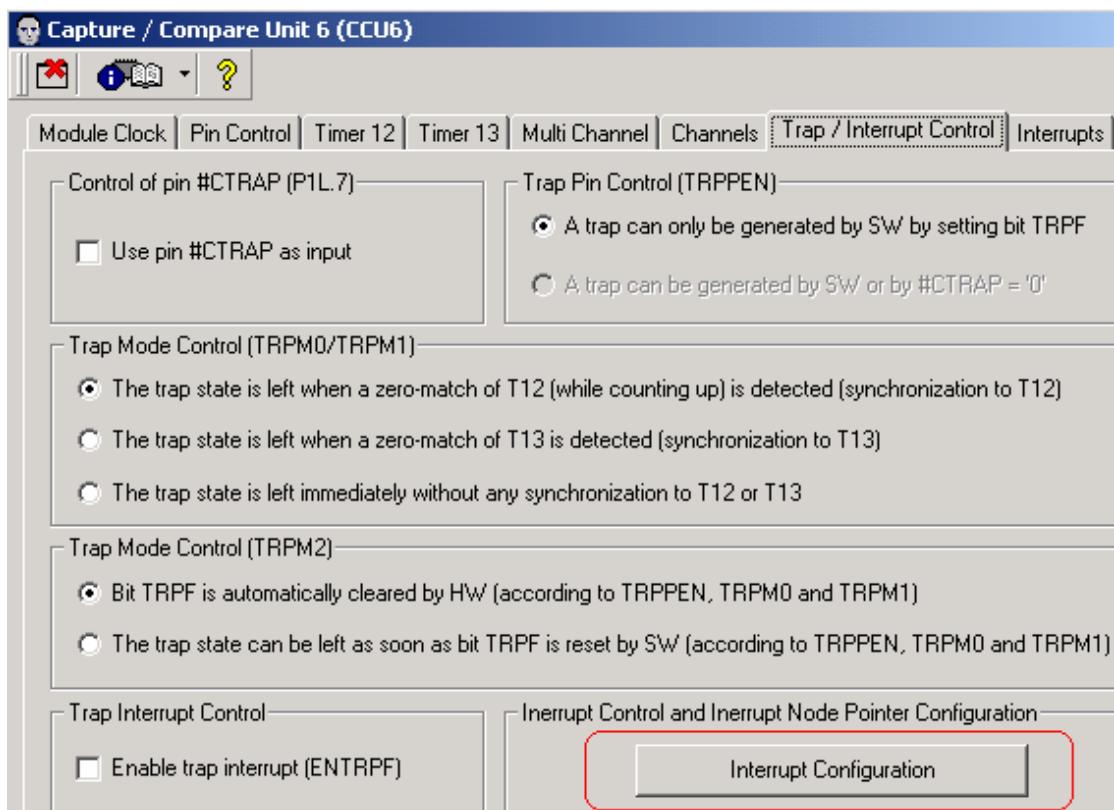


通道 1、2 配置相同。

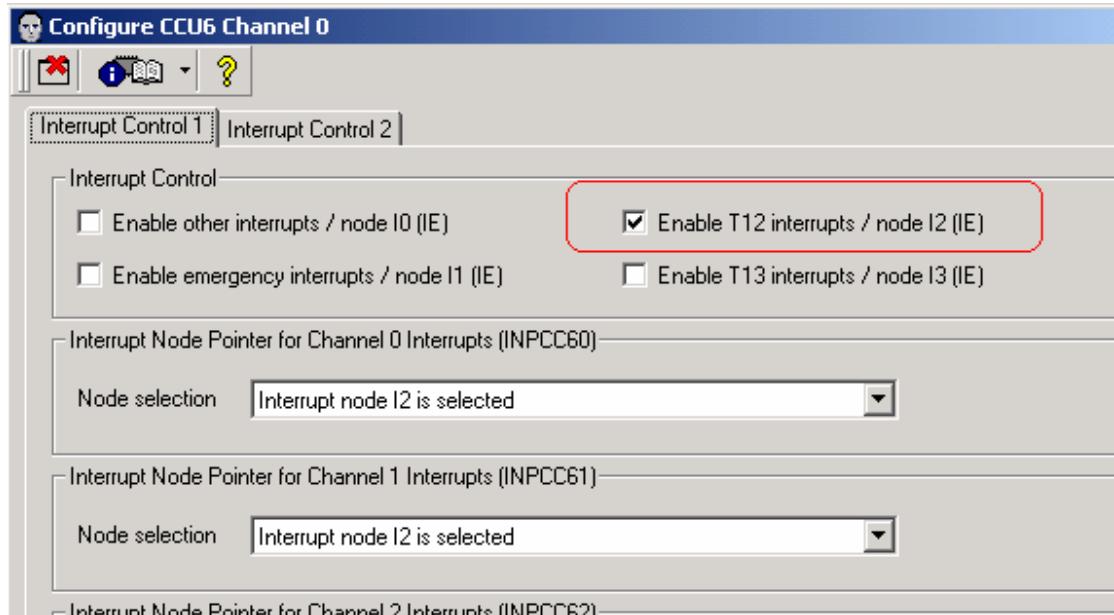
通道 3 设置



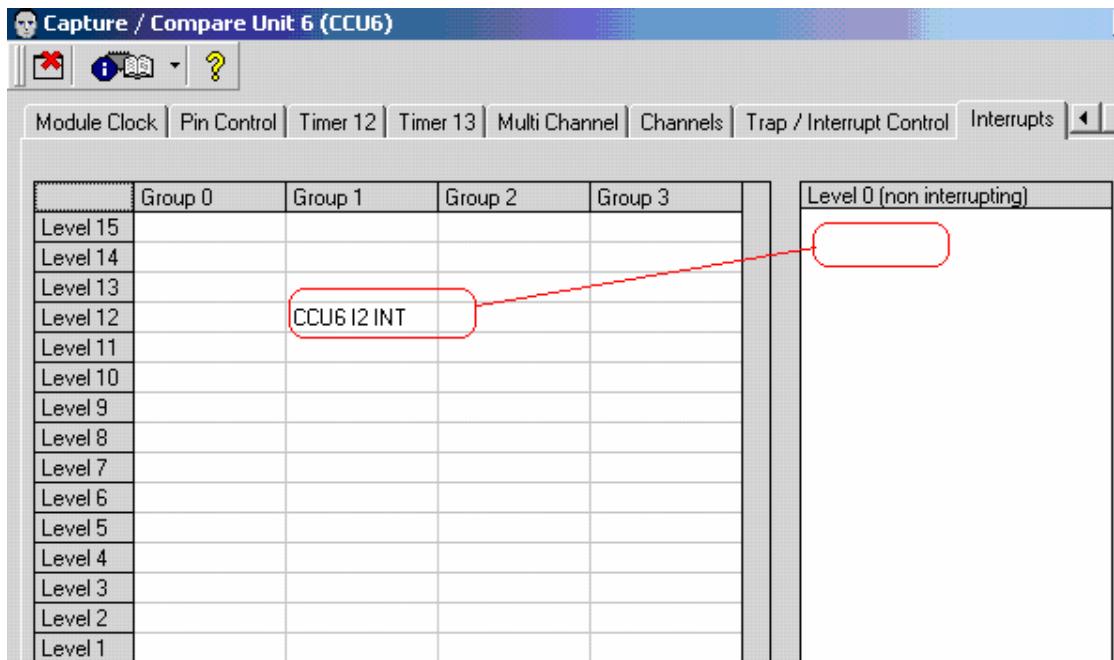
配置中断，在 Trap/Interrupt 页面



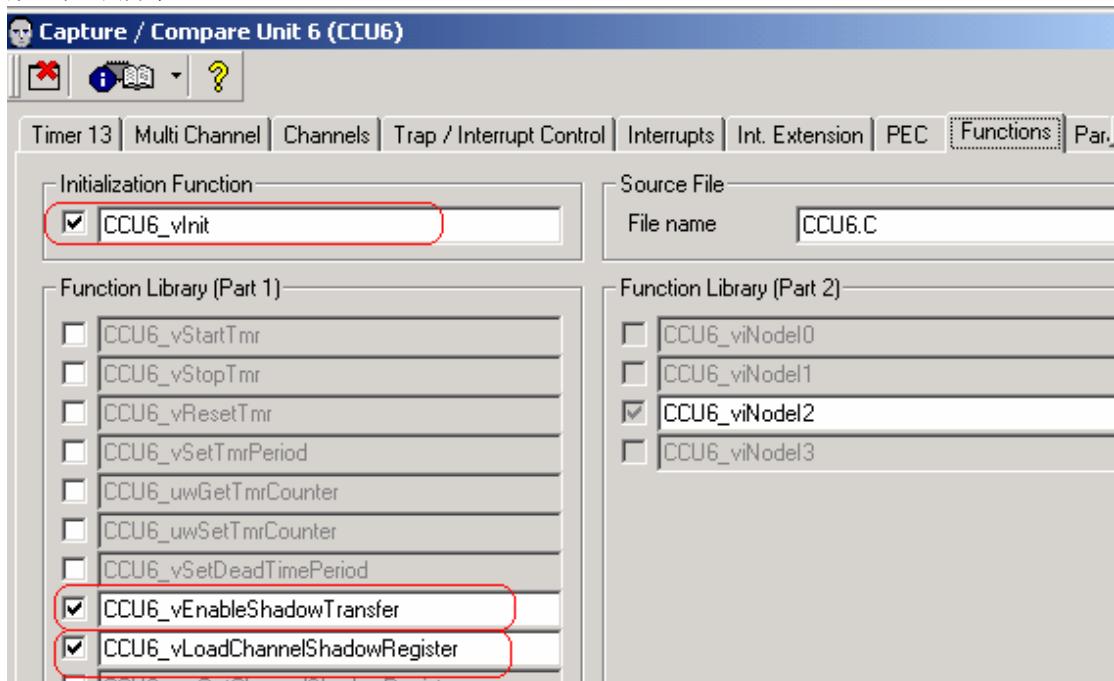
Enable T12 node



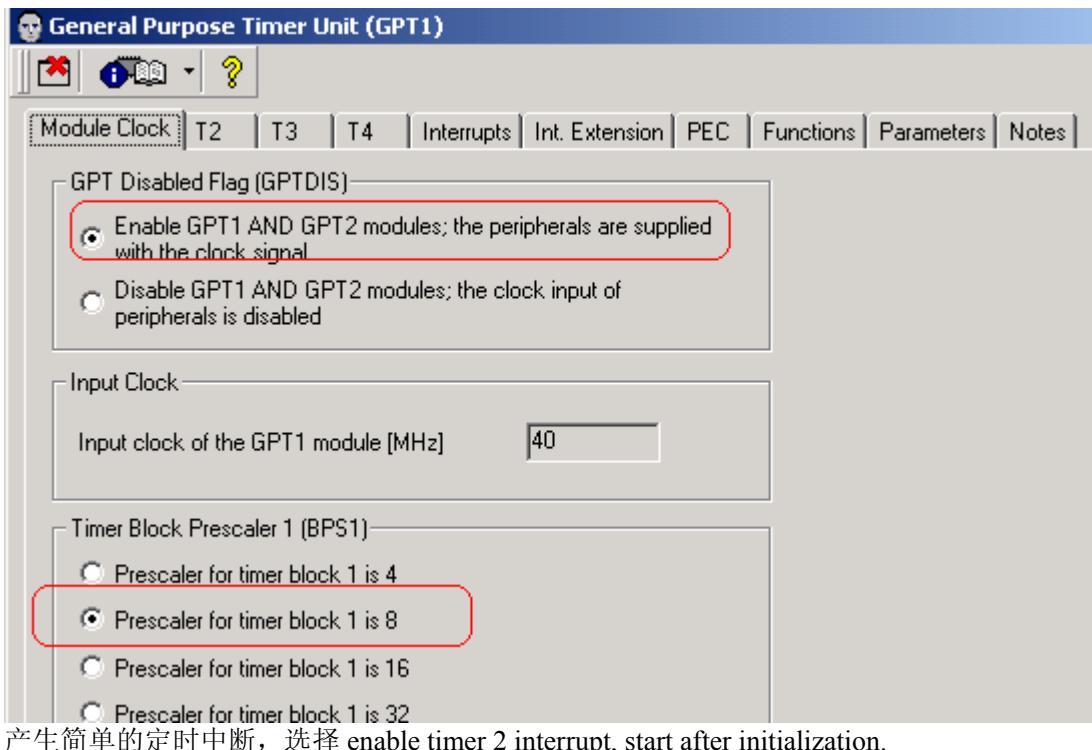
中断优先级配置： 将 CCU6 I2 INT 中断从右边拖到左边表格中。选择优先级和组别。



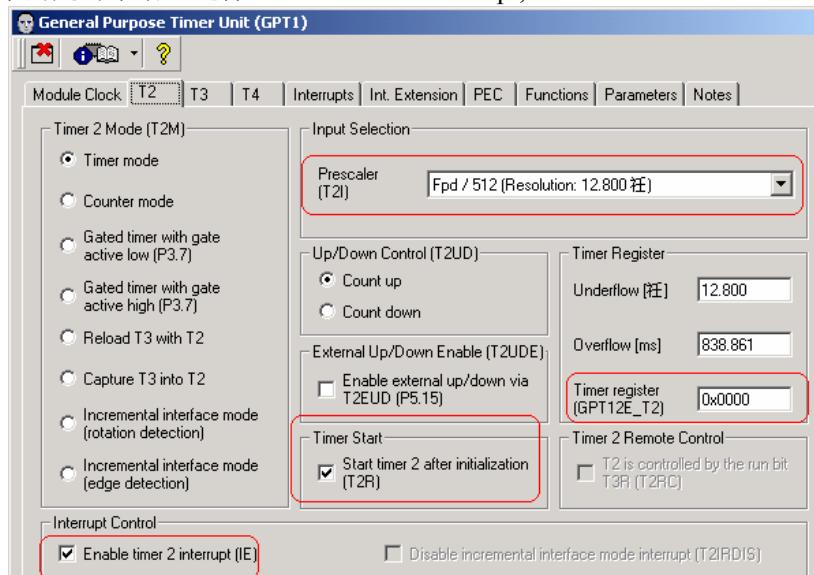
在 functions 页面，选择 CCU6_vInit 以生成 ccu6.c 文件。选择 shadow transfer 相关的两个函数，如下所示：



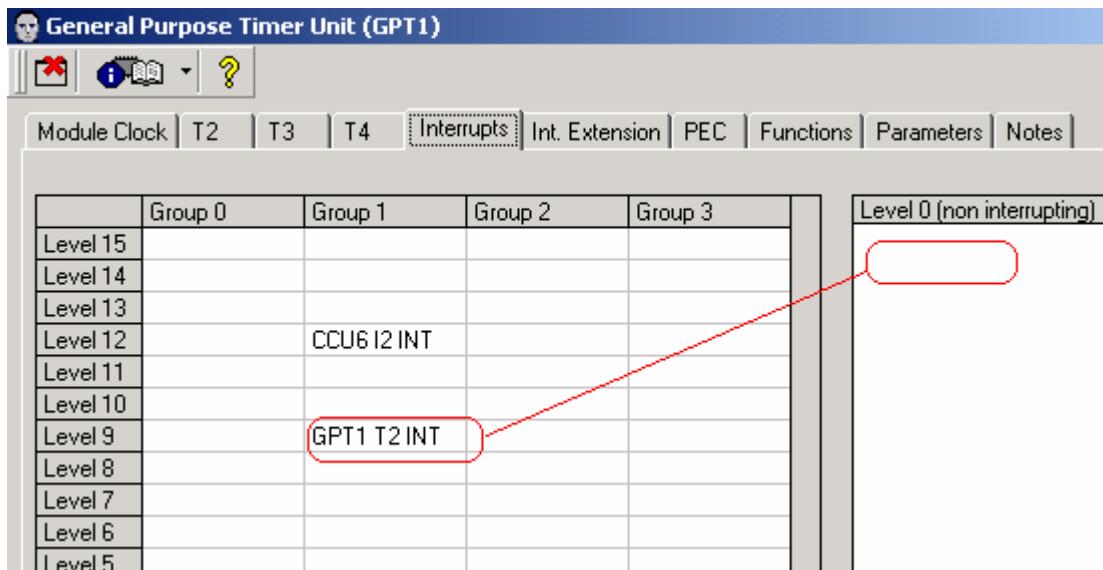
4. 4 配置 T2



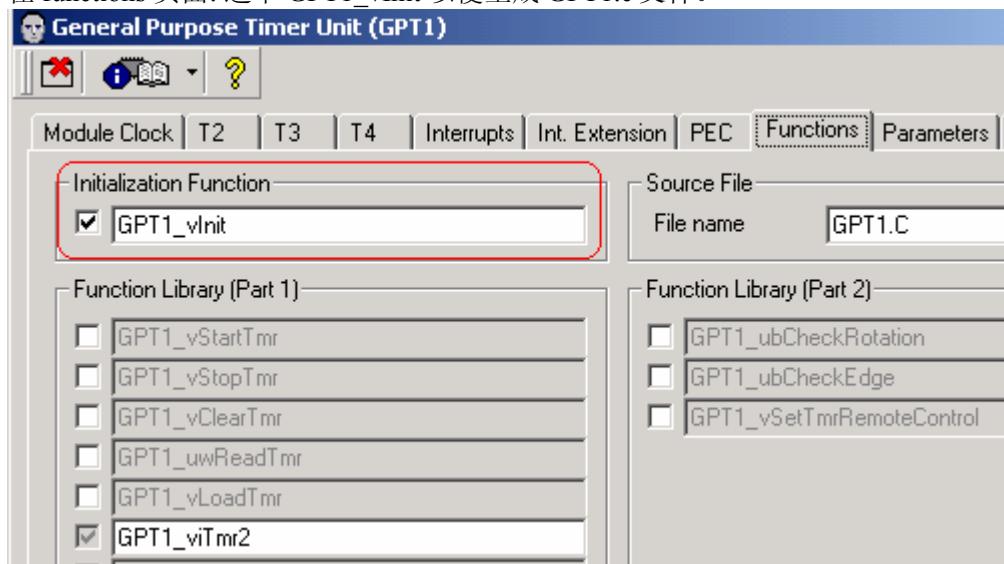
产生简单的定时中断，选择 enable timer 2 interrupt, start after initialization.



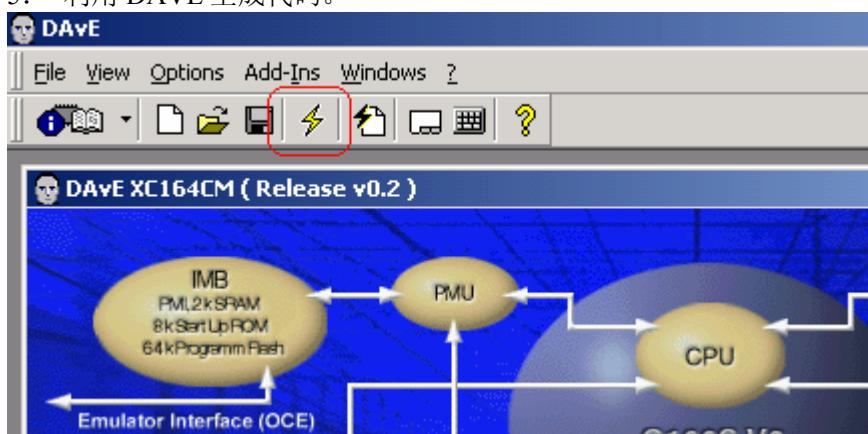
在中断页面, 设置 GPT1 T2INT 的中断优先级和组别。从右边拖到左边表格中相应的位置即可。



在 functions 页面: 选中 GPT1_vInit 以便生成 GPT1.c 文件。



5. 利用 DAVE 生成代码。

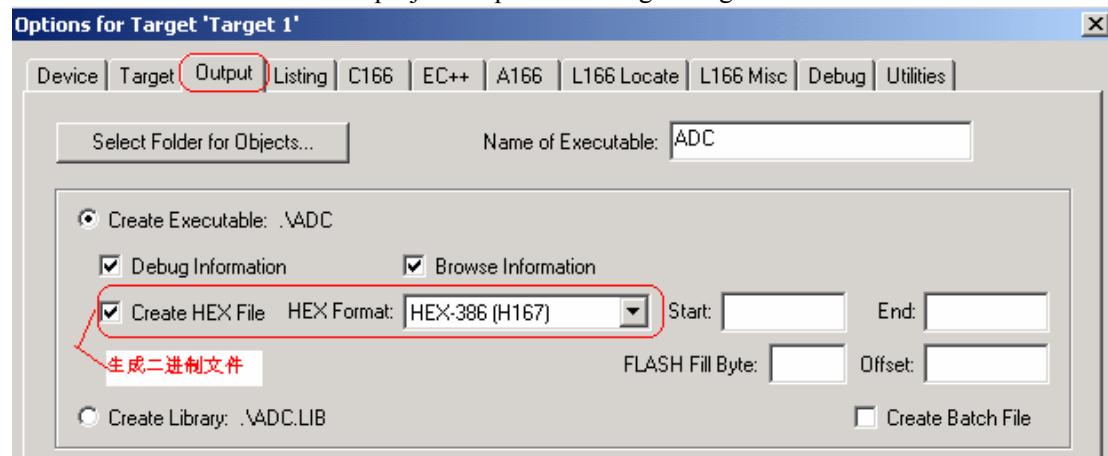




6. 修改用户代码

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 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 GPT1.c
6. 3. 1 声明变量
// USER CODE BEGIN (GPT1_General,7)
unsigned int T12_CompareValue;      // range (0x000 - 0x7cf)
unsigned int T13_CompareValue; // range (0x000 - 0xF9F)
// USER CODE END
6. 3. 2 软件计数，并修改各个通道的占空比
void GPT1_viTmr2(void) interrupt T2INT
{
    // USER CODE BEGIN (Tmr2,2)
```

```

// USER CODE END

// USER CODE BEGIN (Tmr2,5)
// 软件计数
T12_CompareValue += 0x10;
if(T12_CompareValue > 0x7cf)
    T12_CompareValue = 0;

T13_CompareValue += 0x10;
if(T13_CompareValue > 0xF9F)
    T13_CompareValue = 0;
//更新各个通道的比较值。
CCU6_vLoadChannelShadowRegister_CCU6_CHANNEL_0(T12_CompareValue);
CCU6_vLoadChannelShadowRegister_CCU6_CHANNEL_1(T12_CompareValue);
CCU6_vLoadChannelShadowRegister_CCU6_CHANNEL_2(T12_CompareValue);
CCU6_vLoadChannelShadowRegister_CCU6_CHANNEL_3(T13_CompareValue);

// Shadow transfer
CCU6_vEnableShadowTransfer_CCU6_TIMER_12();

CCU6_vEnableShadowTransfer_CCU6_TIMER_13();

// USER CODE END

} // End of function GPT1_viTmr2

```

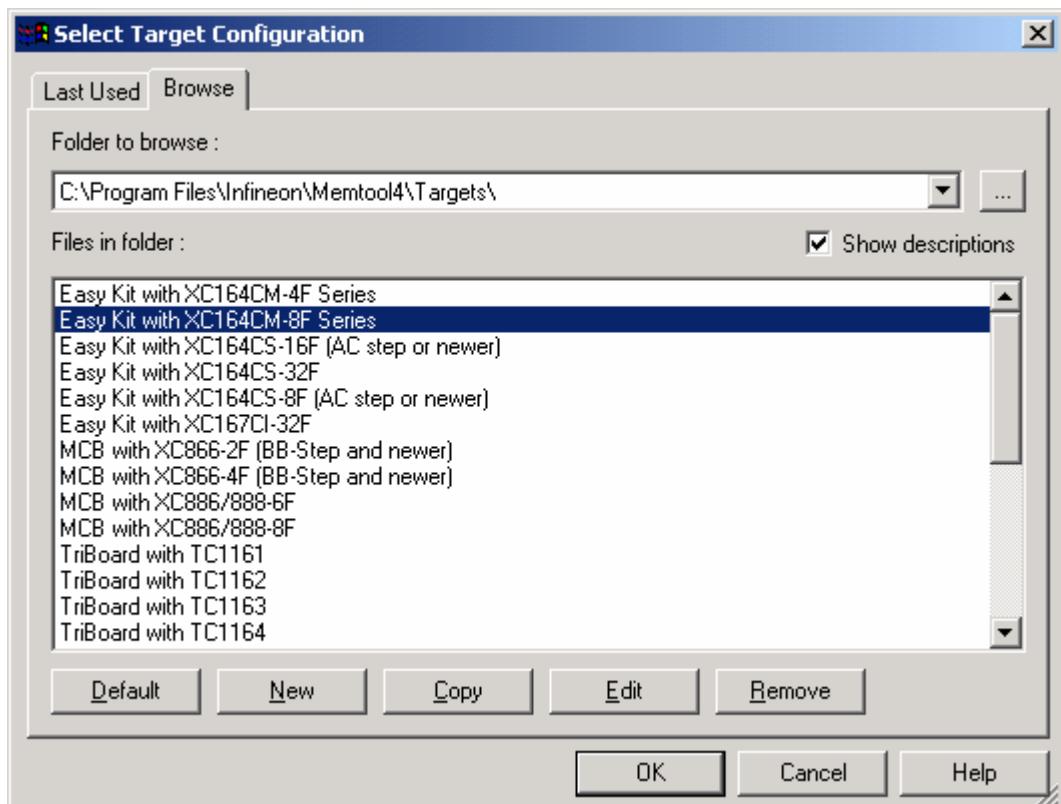
7. 编译



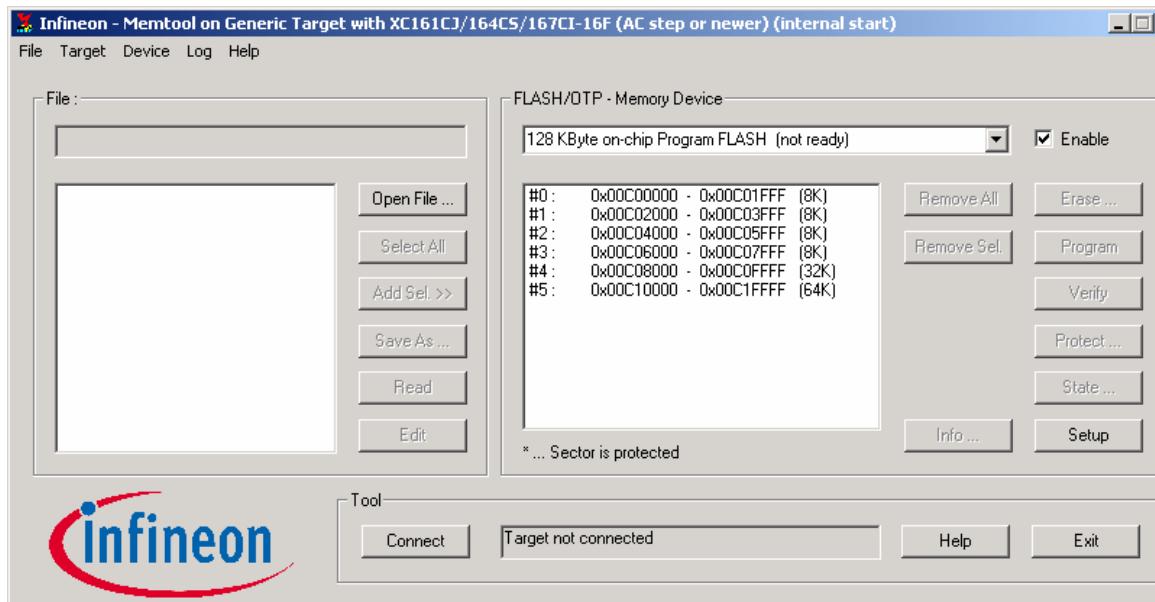
点击图标进行编译连接。如有错误进行更改，直到出现‘0 Errors found.’。

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

可观察到各个通道的占空比在逐渐循环变化。波形如下所示：

