

样例程序：SPWM + V/F

1. 本程序实现的功能如下：

- 配置 CAPCOM6。
- 使用 CC60,CC61,CC62,COOUT60,COOUT61,COOUT62 作为 PWM 输出，工作于互补输出模式。
- 在 T12 定时中断中，使各个通道的占空比按照 SPWM 规律变化。

使用到的模块：CAPCOM6、GPT1。

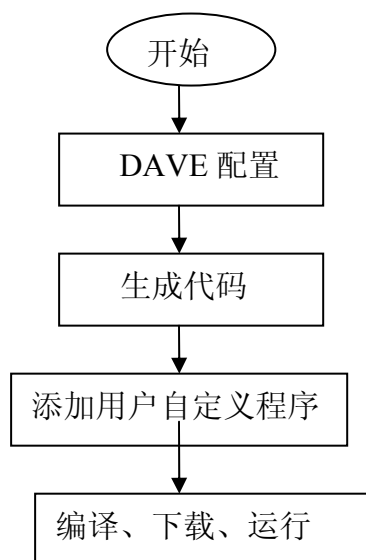
2. 模块介绍

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。

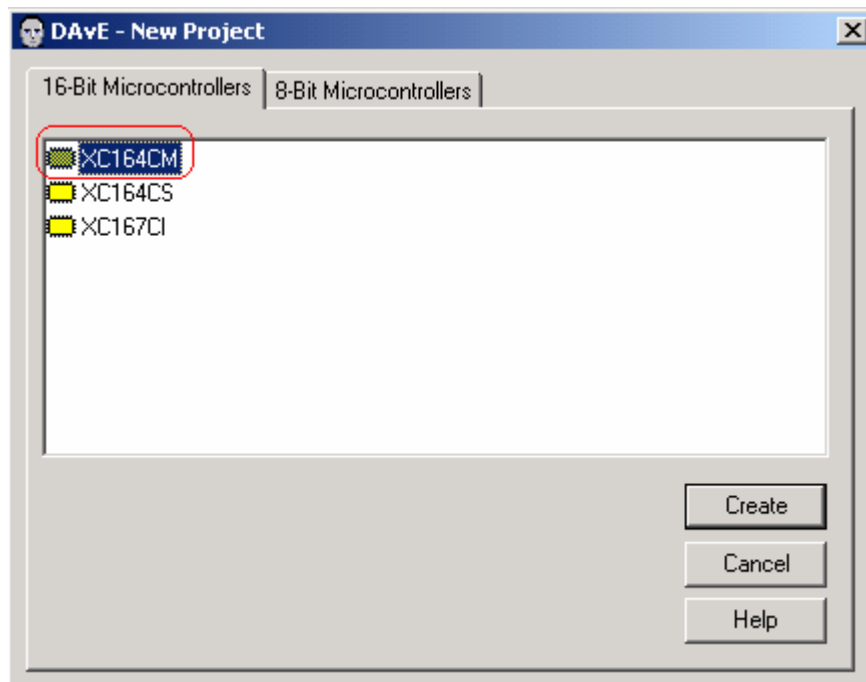
3. 具体实现



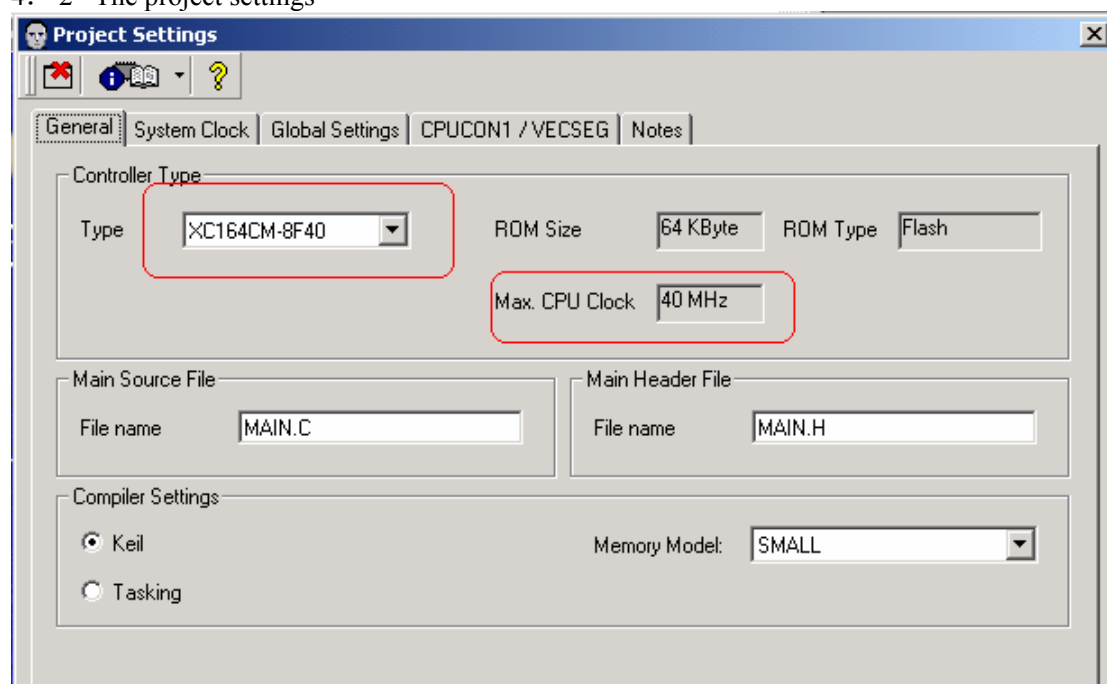
4. DAVE 配置

4.1

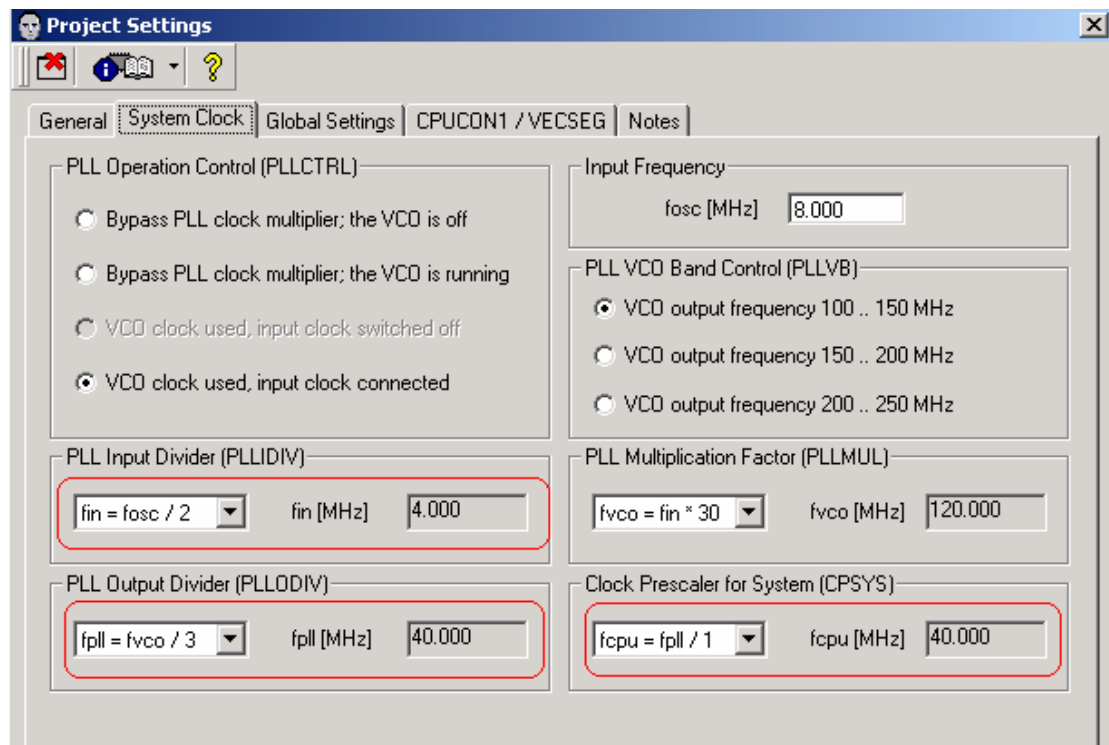
New project: select XC164cm,



4. 2 The project settings

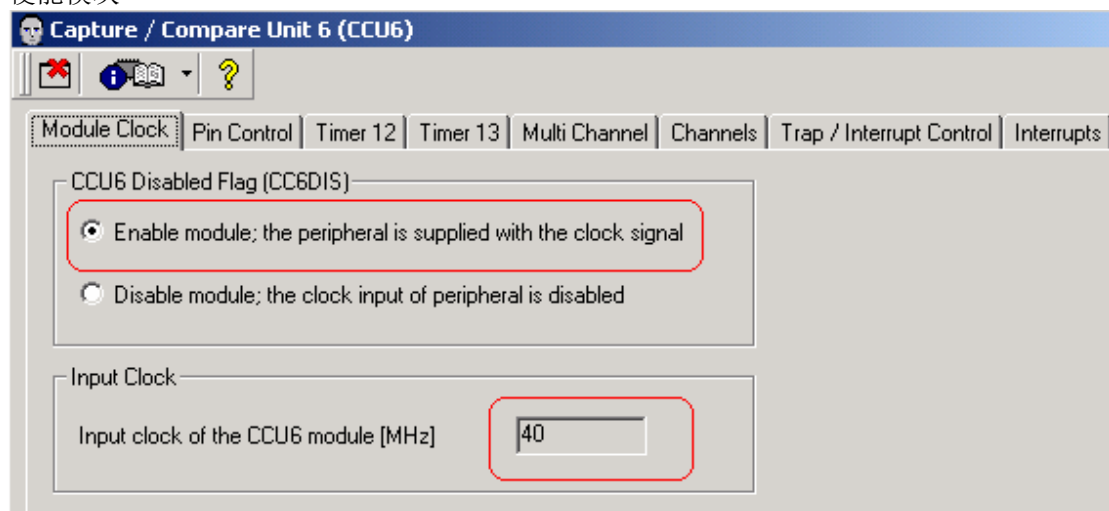


System clock

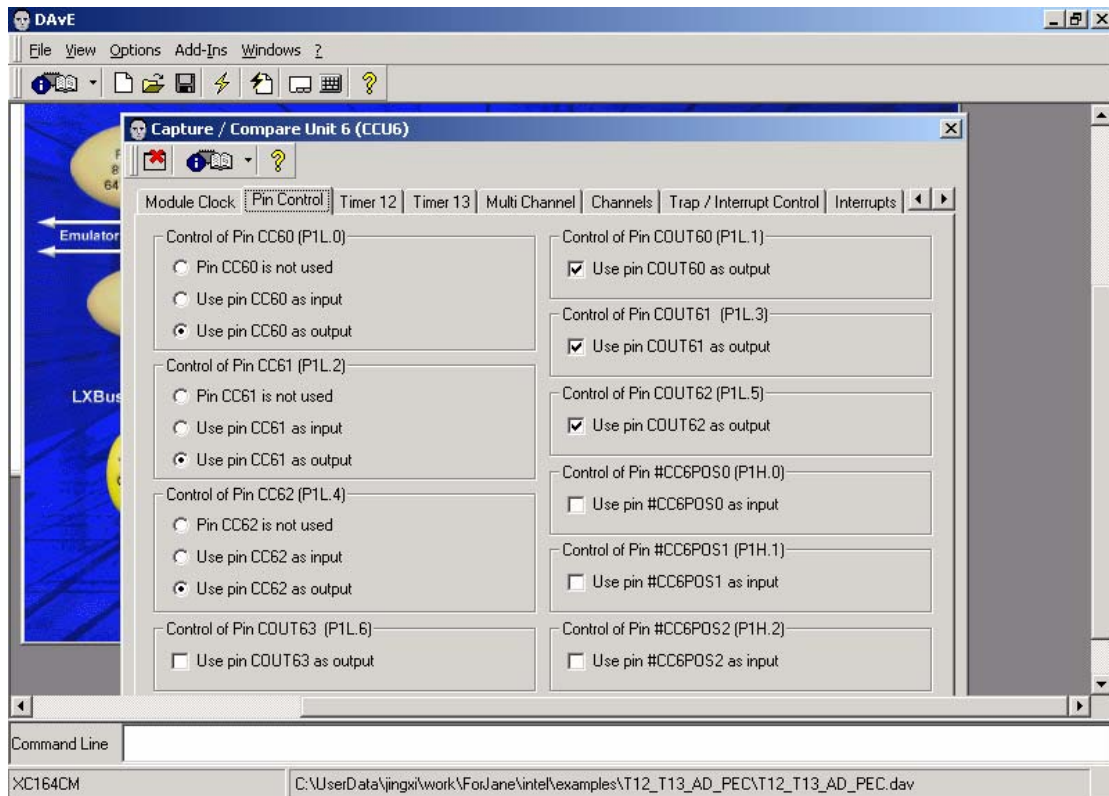


4. 3 配置 CAPCOM6

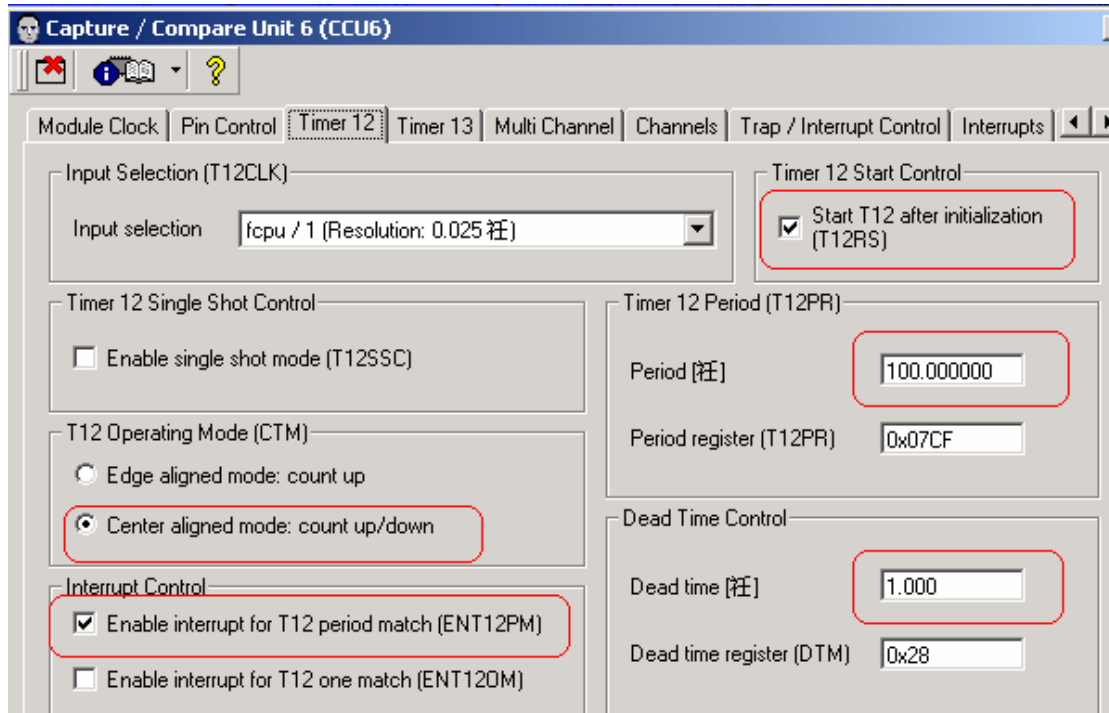
使能模块



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



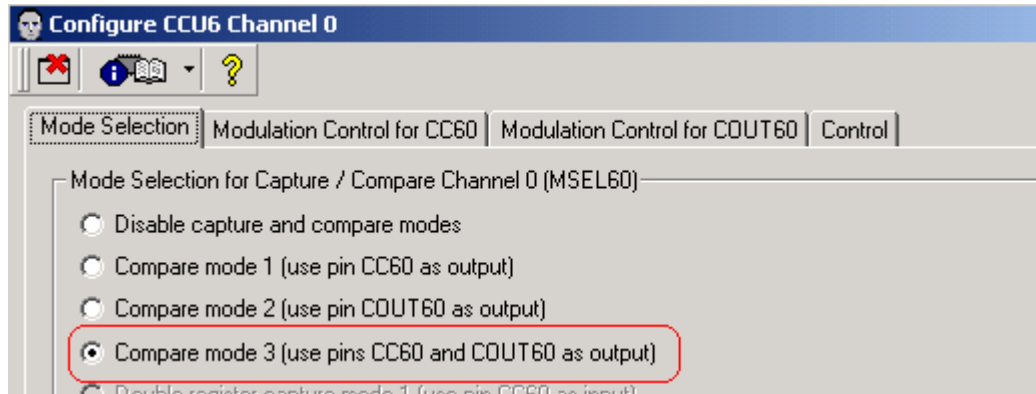
配置 T12



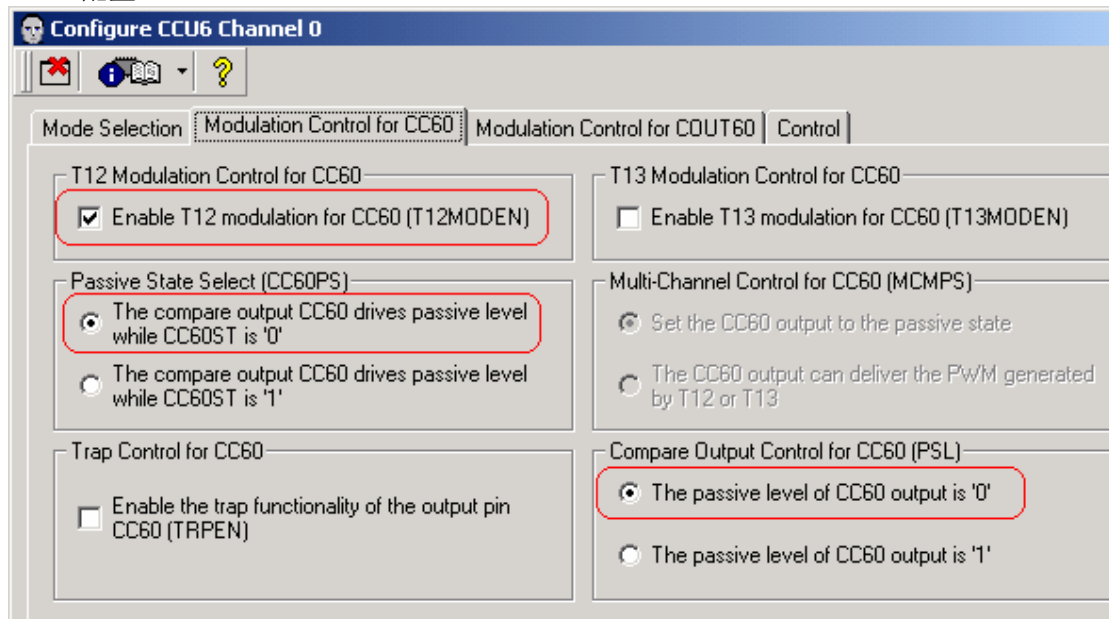
配置各个通道

选择 compare mode 3, T12 modulation, deadtime generation, duty cycle=50%,

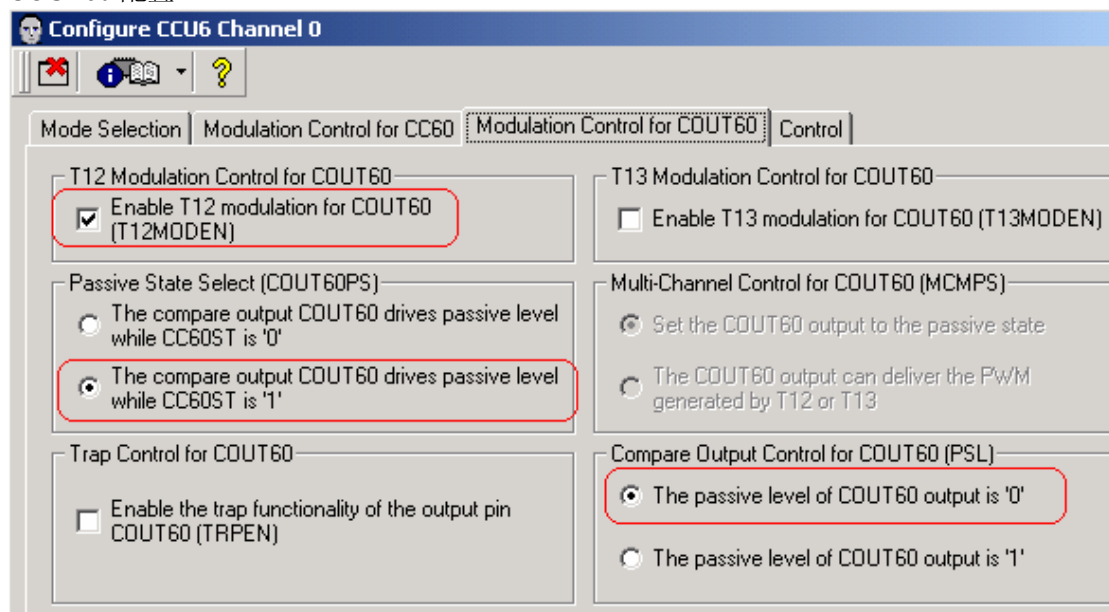
以通道 0 为例：



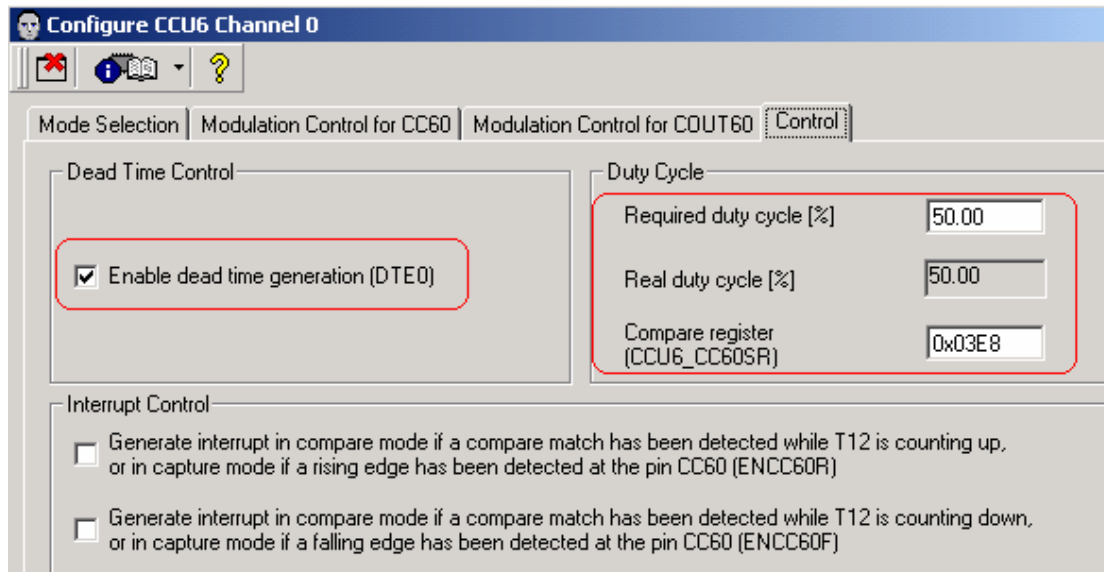
CC60 配置



COUT60 配置

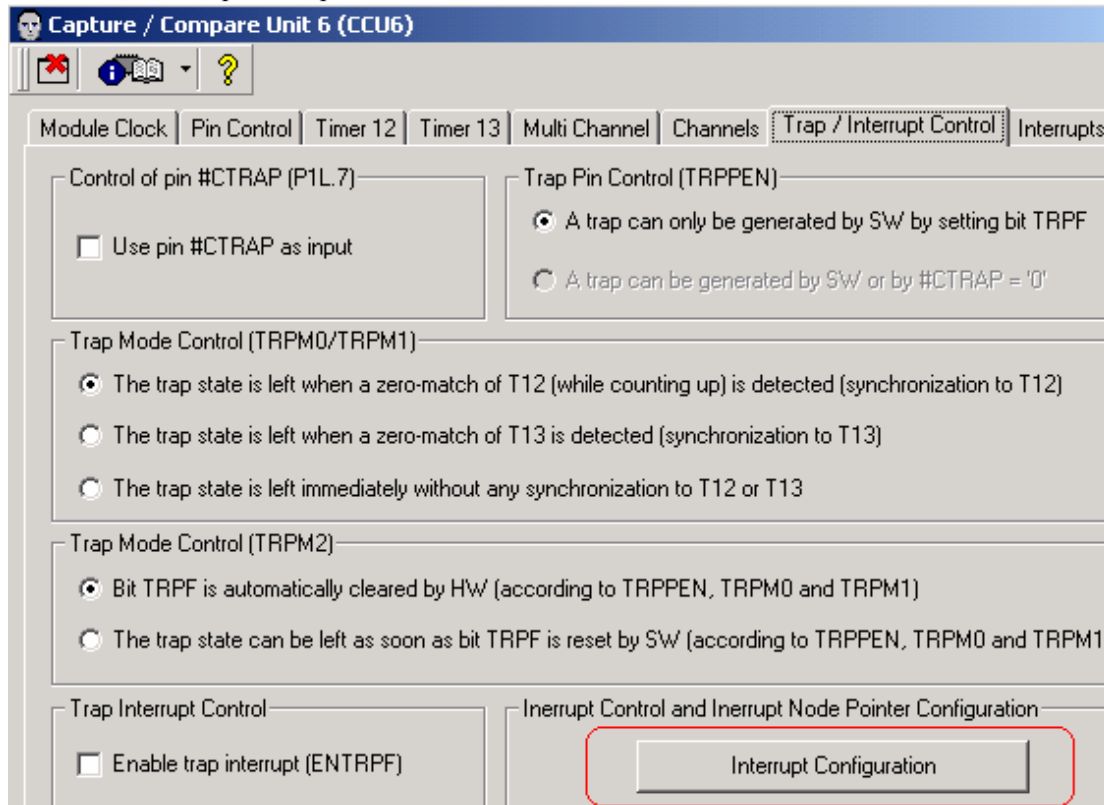


占空比及死区时间配置

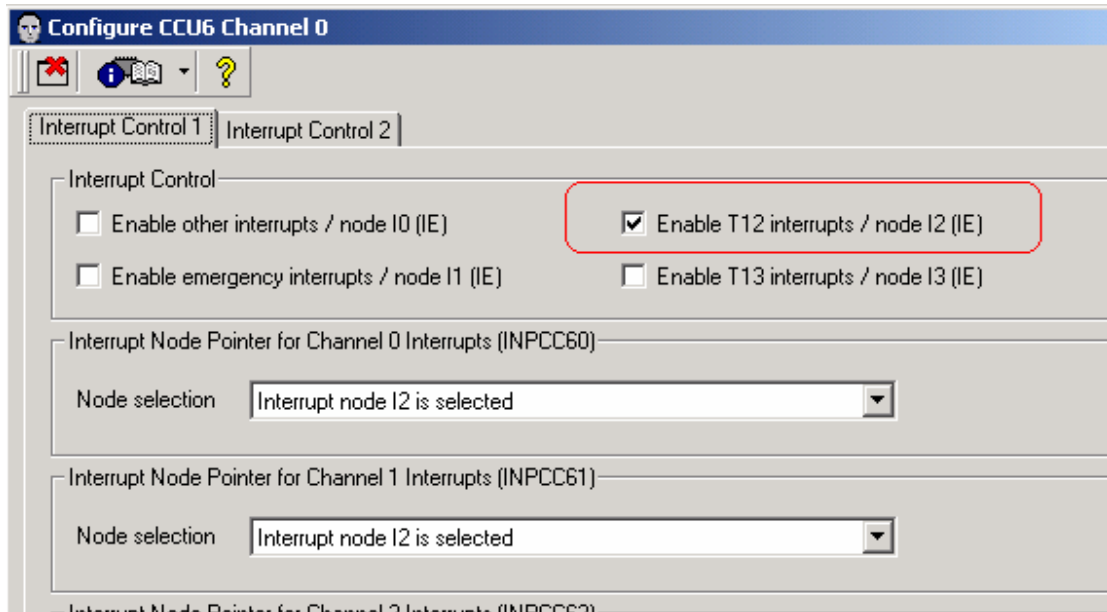


通道 1、2 配置相同。

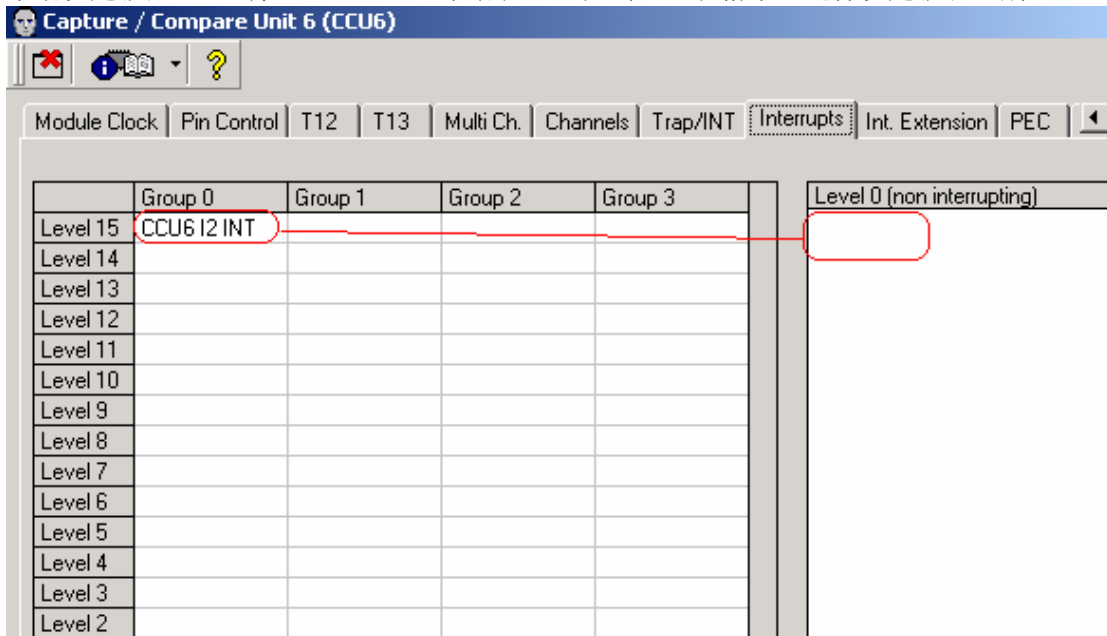
配置中断，在 Trap/Interrupt 页面



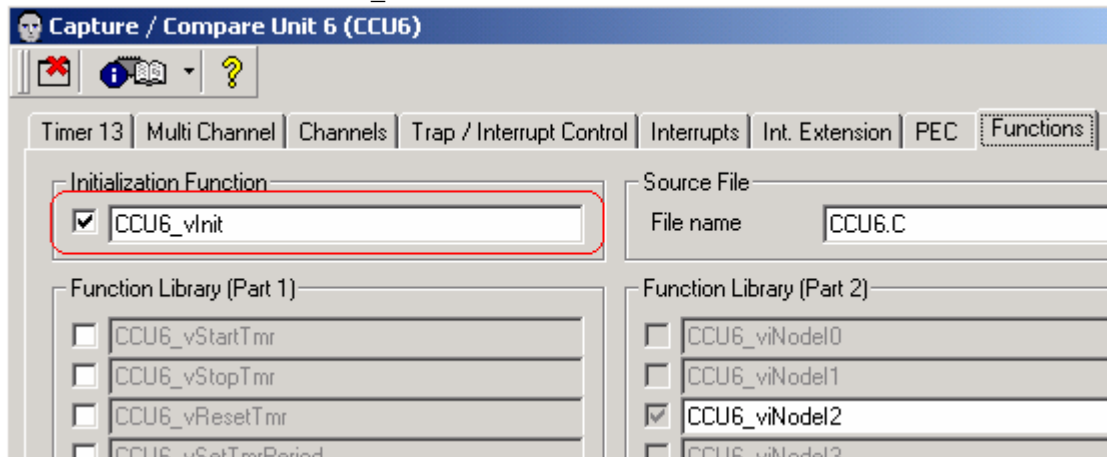
Enable T12 node



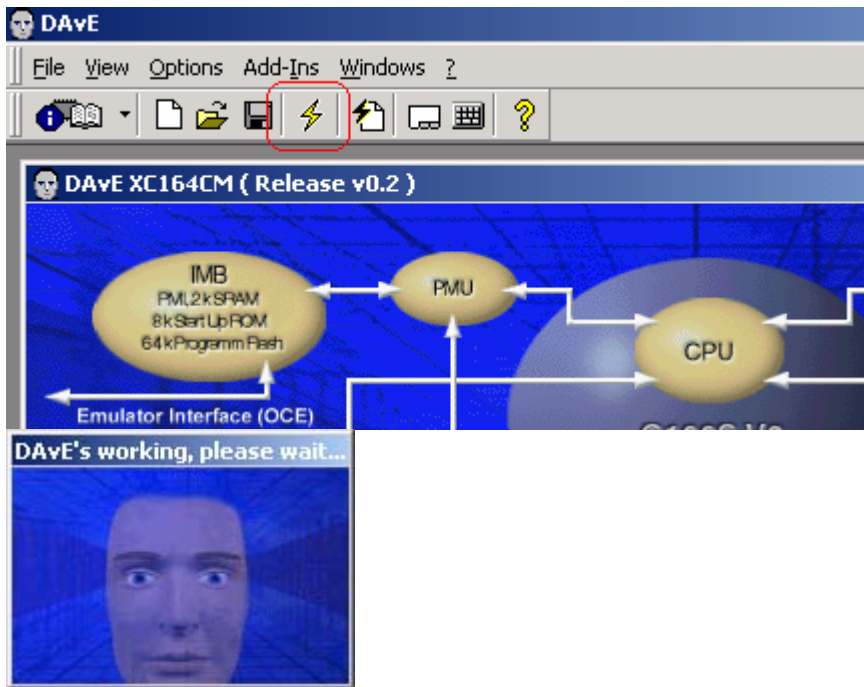
中断优先级配置：将 CCU6 I2 INT 中断从右边拖到左边表格中。选择优先级和组别。



在 functions 页面，选择 CCU6_vInit，生成 ccu6.c 文件。



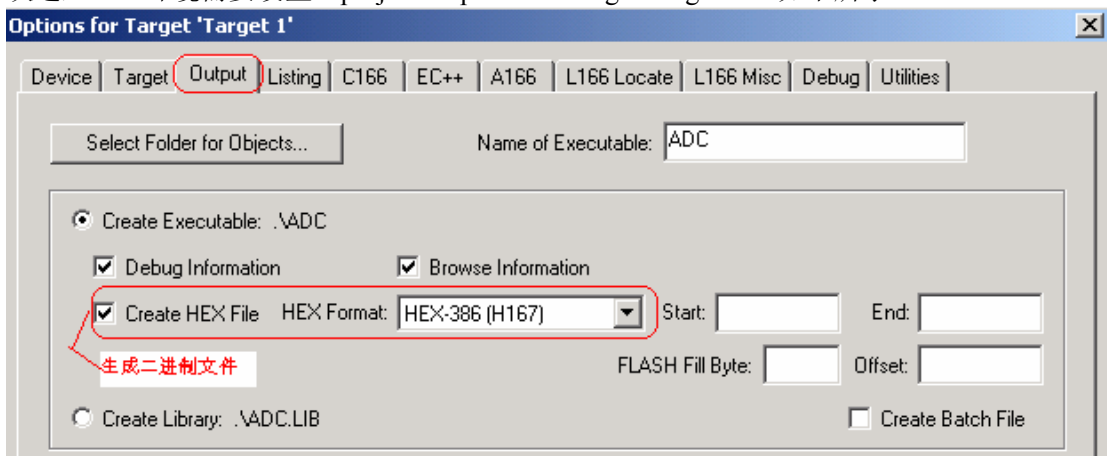
5. 利用 DAVE 生成代码。



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 Main.c

添加 while(1);

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

// USER CODE END

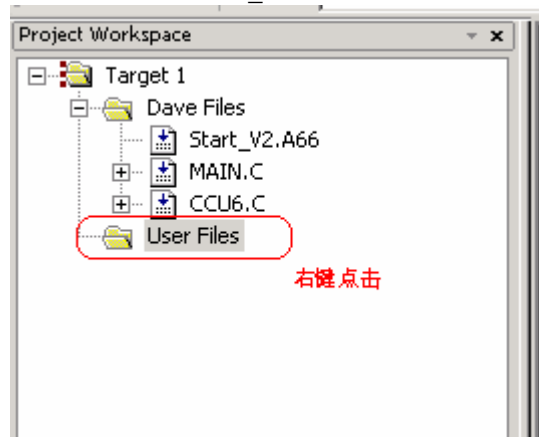
MAIN_vInit();

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

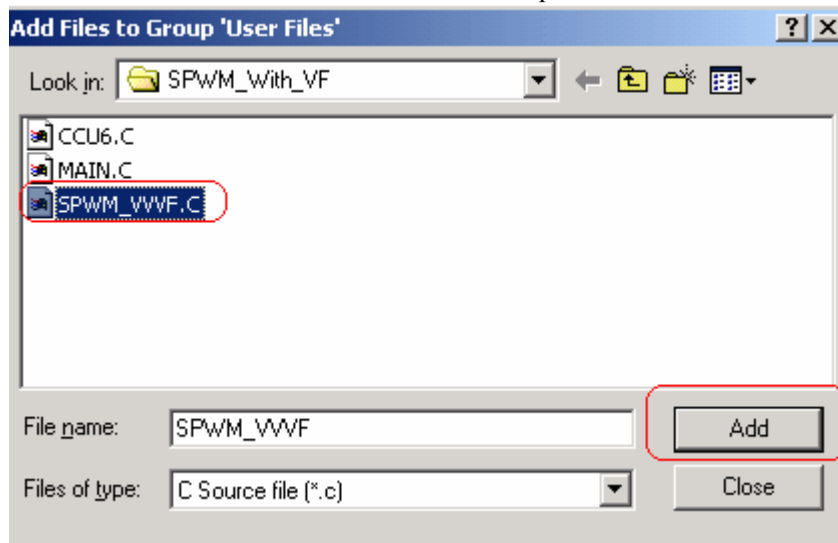
6.3 将例程中的 SPWM_VVVF.h, SPWM_VVVF.c 文件拷贝到项目所在文件夹。

关于 spwm 生成相关的说明请参照 SPWM_VVVF.C 实现。

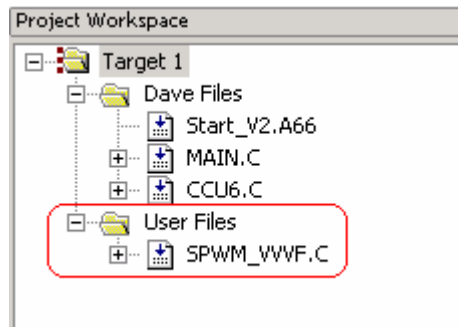
6. 4 向项目中添加 SPWM_VVVF.C。



出现的菜单中选择 Add Files to Group "User Files"



添加完后如下所示



6. 3 Main.h

添加对文件 SPWM_VVVF.h 的调用。
// USER CODE BEGIN (MAIN_Header,10)
#include "SPWM_VVVF.h"
// USER CODE END

6. 4 CCU6.c

在 T12 定时中断中添加产生 VVVF 相关的程序
void CCU6_viNodeI2(void) interrupt CCU6_NodeI2_INT
{
 // USER CODE BEGIN (NodeI2,2)

 // USER CODE END

```


if(CCU6_ISR & 0x0080) // if CCU6_ISR_T12PM
{
    // timer T12 period match detection

    // USER CODE BEGIN (NodeI2,19)
    SPWM_VVVF();
    // USER CODE END
    CCU6_ISR = 0x0080; // clear flag CCU6_ISR_T12PM
}

} // End of function CCU6_viNodeI2

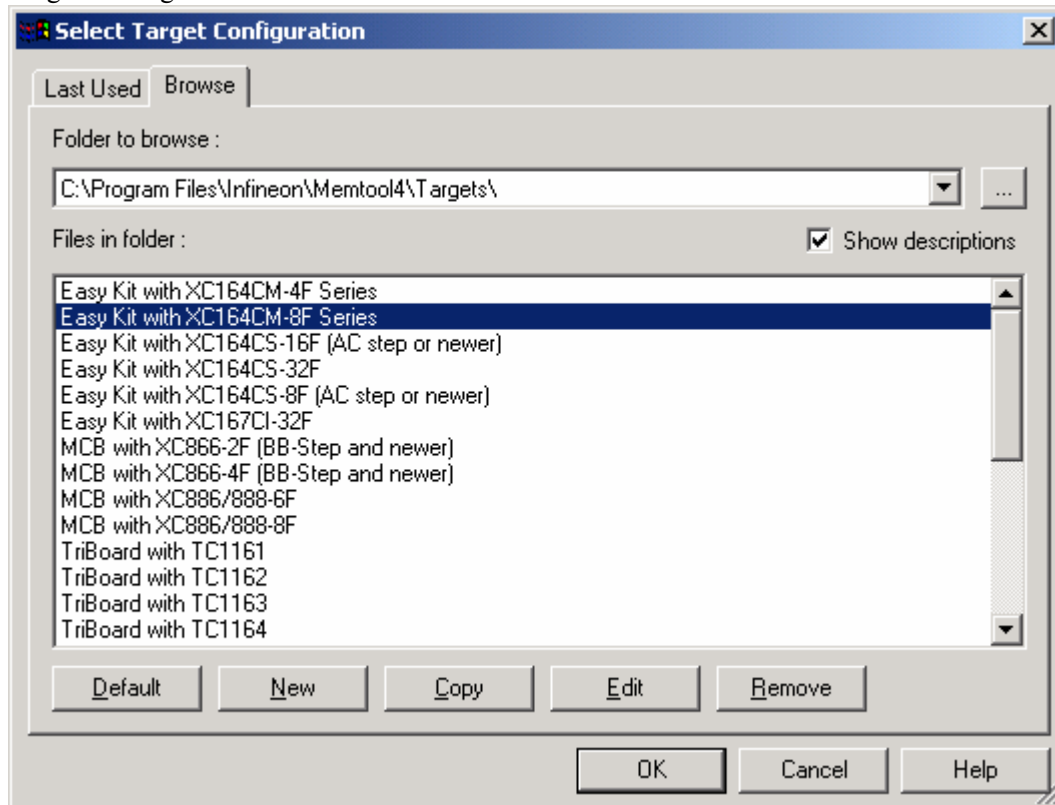
```

7. 编译

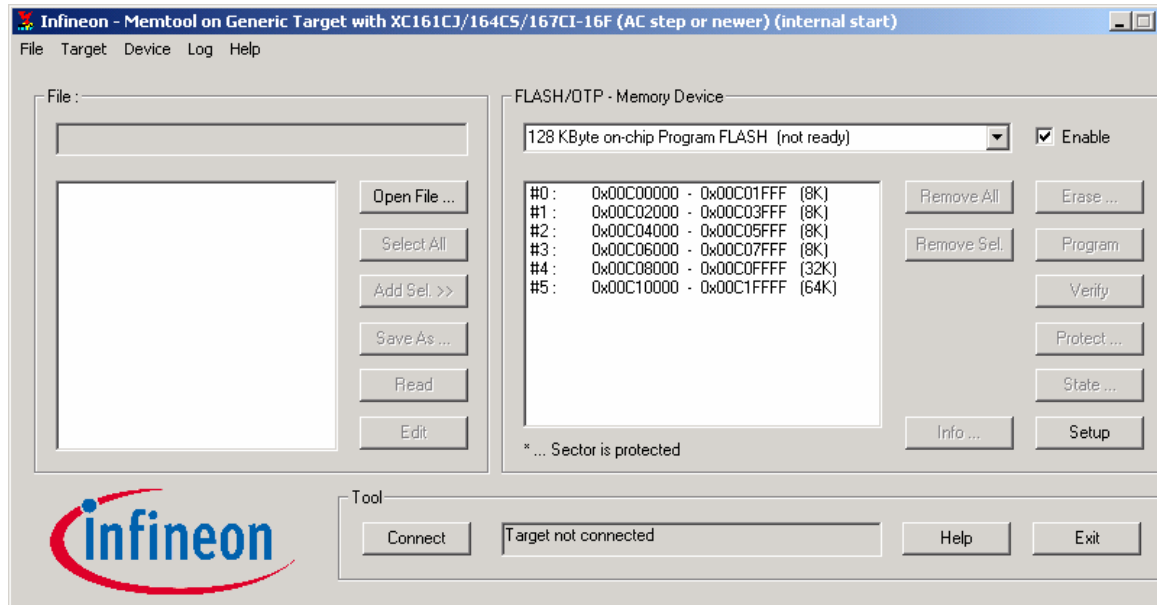
点击  图标进行编译连接。如有错误进行更改，直到出现‘0 Errors found.’。

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：

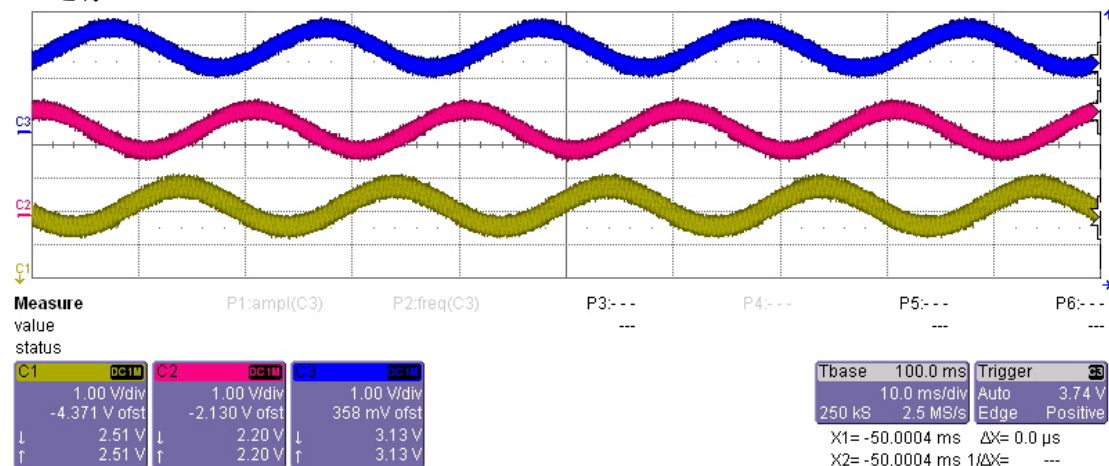


点击 OK 出现如下对话框。



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



附：SPWM_VVVF.C 实现

SPWM_VVVF.c

➤ SIN 表

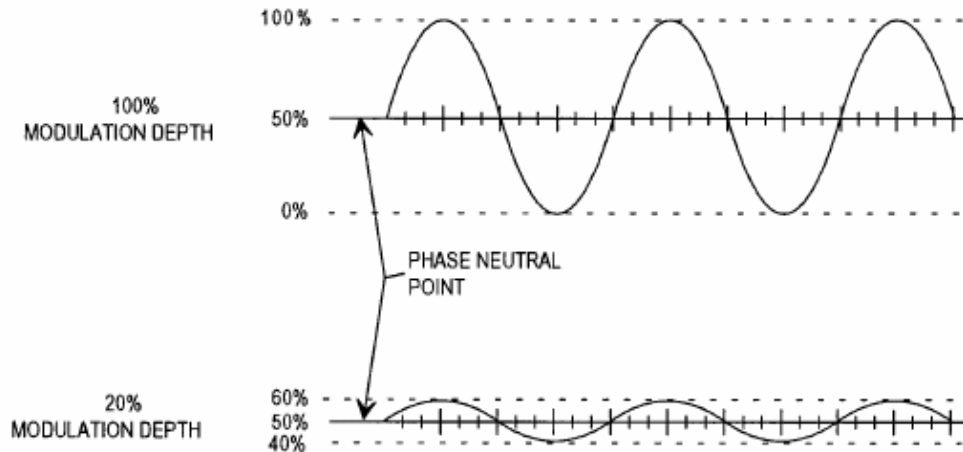
计算 0-90 度的 SIN 表，对应于 0 - 0x7fff。由于 SIN 函数的对称性，其他象限从第一个象限的数据生成。

```
const unsigned int sin_TABLE[250]; // SIN 表
```

```
unsigned char Get_F_Command(); // 获取频率信息。例程中仅通过软件指定。
```

```
unsigned int UVW_Cal(unsigned int index); // 根据象限不同，利用 0-90 度对应的 SIN 数据计算其他象限的 SIN 值，并根据 V/F 曲线得到对应的数据 PWM 比较值。
```

PWM 计算值：包含 V/F 控制部分。



$$U_{\text{Phase}} = \text{Neutral} + A * \sin[0^\circ - 360^\circ]$$

其中 A 为根据 V/F 控制得到的调制深度而决定的正弦波的幅值。Neutral 为中线。

```
comp_temp = (long)(((long)sin_TABLE[index] * (long)VF_Mul_F >> 16) + 0x7fff) *
(long)Neutral >> 16;
```

其中

- Neutral 为 1/2 T12 周期值。
- VF_Mul_F 为根据 V/F 曲线及当前输出频率得到的 PWM 调制深度。

➤ SPWM_VVVF(void)

定义变量

```
static float u_index = 0, v_index = 333, w_index = 667;
```

//各通道的指针

```
static unsigned int Tu, Tv, Tw;
```

//个通道的比较值

```
static float step;
```

//步长

```
unsigned int F_OUT = 200; //输出频率值
```

计算步长

//载波频率 10k, 在一个正弦波中包含 1000 点

//设 $n = \{ (1/F_{\text{out}}) / (1/F_{\text{carry}}) \}$; $\rightarrow n = F_{\text{carry}} / F_{\text{OUT}}$;

// $\text{step} = 1000 / n = F_{\text{OUT}} / 10$;

```
step = (float) F_OUT / 10;
```

更新 PWM 占空比。

```
CCU6_CC60SR = Tu;
```

```
CCU6_CC61SR = Tv;
```

```
CCU6_CC62SR = Tw;
```

```
CCU6_vEnableShadowTransfer_CCU6_TIMER_12();
```