

根据“实验十数据采集实验”的范例程序，打开程序LAB11_main.c及相关头文件，阅读程序段落。

与ePWM模块设置相关的程序语句

```
void InitEPwm1Parameters(void)
{
    // InitEPwm1Gpio();
    // Disable TBCLK within the ePWM
    EALLOW;
    SysCtrlRegs.PCLKCR0.bit.TBCLKSYNC = 0; //停止epwm模块内部的时间基准时钟
    EDIS;
    // High Speed Time-base Clock Prescale Bits, These bits determine part of the time-
    base clock prescale
    // TBCLK = SYSCLKOUT / (HSPCLKDIV*CLKDIV)=150/(6*1)=25
    EPwm1Regs.TBCTL.bit.HSPCLKDIV = 0x03; //高速时间基准时钟预分频位 两倍
    EPwm1Regs.TBCTL.bit.CLKDIV = 0x00; //时间基准时钟预分频位 等于0即1分频
    // Set Period for EPWM1
    EPwm1Regs.TBPRD = 208; //设定时间基准器计数器的周期 208-fs 20kHz, 139-fs
    30kHz 149--27.9kHz T(PWM1)=TBCLK/(TBPRD*2*3)=25/(208*3*2) = 0.02MHz , 20KHz
    EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; //增减计数模式
    // Setup Compare A = 2 TBCLK counts
    EPwm1Regs.CMPA.half.CMPA = 2; //计数比较寄存器A CMPA 当前工作的CMPA的值不断和时间基准计
    数器TBCTR比较
    // Phase is 0 for Synchronization Event
    EPwm1Regs.TBPHS.half.TBPHS = 0x0000; //TBCTR不装载相位寄存器TBPHS的值
    // Clear TB counter
    EPwm1Regs.TBCTR = 0x0000; //事件基准计数寄存器TBCTR 读取写到其中的TBCTR的值 清除
    // Phase loading disabled
    EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE; //禁止TBCTR对TBPHS的装载
    // Enable the TBCTL Shadow
    EPwm1Regs.TBCTL.bit.PRDL = TB_SHADOW; //TBCTR装载其映射寄存器的值
    // Disable EPWMxSYNCO signal
    EPwm1Regs.TBCTL.bit.SYNCOSEL = TB_SYNC_DISABLE; //禁用EPWMxSYNCO signal
    // CMPA Register operating mode, 0 means operates as a double buffer, all writes via
    the CUP access the shadow register
    EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW; //映射模式，双缓冲模式，所有CPU写操作访问映射
    寄存器
    // Active CMPA Load From Shadow Select Mode when CTR=0
    EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR = Zero
    // Set actions
    // Force EPWMA output high when the counter equals the active CMPA register and the
    counter is incrementing
    EPwm1Regs.AQCTLA.bit.CAU = AQ_SET; //计数递增 强制ePWMxA输出高
    // Force EPWMA output low Action when the counter equals the active CMPA register
    and the counter is decrementing
    EPwm1Regs.AQCTLA.bit.CAD = AQ_CLEAR; //计数递减 强制ePWMxA输出低
    // Dead-Band Generator Rising Edge Delay Count Register=0
    // EPwm1Regs.DBRED=0;
    // Dead-Band Generator Falling Edge Delay Count Register=0
    // EPwm1Regs.DBFED=0;
    // Enable ADC Start of SOCA Pulse
```

```

EPwm1Regs.ETSEL.bit.SOCAEN = 1;      //使能ePWMxSOCA脉冲
// Select SOC from CPMA on upcount
EPwm1Regs.ETSEL.bit.SOCASEL = 2;      //TBCTR=TBPRD时产生ePWMxSOCA
// Select how many selected ETSEL events need to occur before an EPWMxSOCA pulse is
generated;//在第三个事件产生ePWMxSOCA脉冲
EPwm1Regs.ETPS.bit.SOCAPRD = 3;
// Enable event time-base counter equal to period (TBCTR = TBPRD)
EPwm1Regs.ETSEL.bit.INTSEL = ET_CTR_PRD; // TBCTR=TBPRD时产生ePWMxSOCA
// Enable EPWMx_INT generation
EPwm1Regs.ETSEL.bit.INTEN = 0;        //禁止ePWMx_INT产生
// These bits determine how many selected ETSEL[INTSEL] events need to occur before
an interrupt is generated.
EPwm1Regs.ETPS.bit.INTPRD = ET_3RD;   //在第三个事件产生中断
// Enable TBCLK within the ePWM
EALLOW;
SysCtrlRegs.PCLKCR0.bit.TBCLKSYNC = 1;
EDIS;
}

```

寄存器TBCTL与TBPRD各字段的数值及其含义

TBCTL时基控制寄存器

- CTRMODE: 计数器模式, 用于选择计数器的工作模式。
在此例程中配置语句为 `EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN;` //增减计数模式
- PHSEN: 相位使能, 当该位为1时, 定时器产生相位差。
在此例程中配置语句为 `EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE;` //禁止TBCTR对TBPHS的装载
- PRDLD: 周期寄存器重载
在此例程中配置语句为 `EPwm1Regs.TBCTL.bit.PRDLD = TB_SHADOW;` //TBCTR装载其映射寄存器的值
- SYNCOSSEL: 同步输出选择, 用于选择何时产生同步事件。
在此例程中配置语句为 `EPwm1Regs.TBCTL.bit.SYNCOSSEL = TB_SYNC_DISABLE;` //禁用EPWMxSYNCO signal
- HSPCLKDIV: 高速时钟分频因子, 用于将输入时钟分频。
$$TBCLK = SYSCLKOUT / (HSPCLKDIV \times CLKDIV)$$

在此例程中配置语句为 `EPwm1Regs.TBCTL.bit.HSPCLKDIV = 0x03;` //高速时间基准时钟预分频位 是/6
- CLKDIV: 时钟分频因子, 用于将输入时钟分频。
在此例程中配置语句为 `EPwm1Regs.TBCTL.bit.CLKDIV = 0x00;` //时间基准时钟预分频位 等于0即1分频

TBPRD

PWM事件的频率由时基周期 (TBPRD) 寄存器和时基计数器的模式控制。

每一步的时间增量由时基时钟 (TBCLK) 定义。

例如: 在上下计数模式下, 时基计数器从零开始递增, 直到达到周期 (TBPRD) 值。当达到周期值时, 时基计数器递减, 直到它达到零。此时, 计数器重复该模式并开始递增。

对于增或减模式: $T_{PWM} = (TBPRD + 1) \times T_{TBCLK}$, $F_{PWM} = 1/(T_{PWM})$

对于增减模式: $T_{PWM} = 2 \times TBPRD \times T_{TBCLK}$, $F_{PWM} = 1/(T_{PWM})$

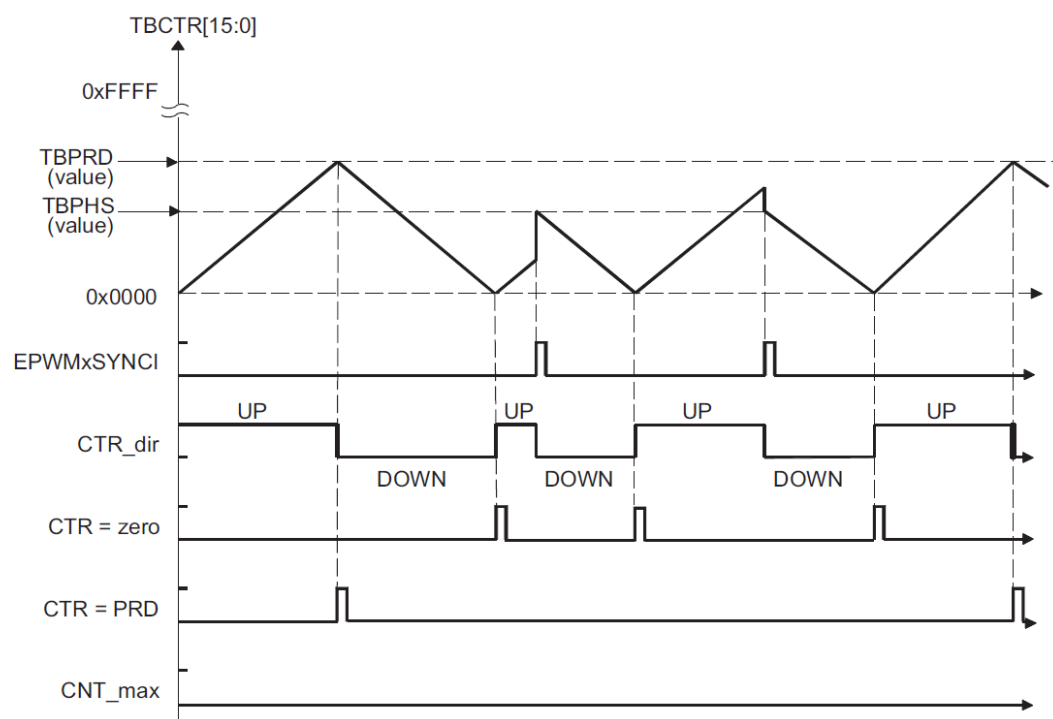
在此例程中配置语句为

```
EPwm1Regs.TBPRD = 208;      //设定时间基准器计数器的周期
//TBCLK = SYSCLKOUT/(HSPCLKDIV*CLKDIV) = 150 MHz/(6*1) = 25MHz
//T(TB_PWM)=TBCLK/(TBPRD*2)=25/(208*2) = 0.06MHz = 60KHz
```

时间基准模块TB产生事件的频率

以下时序图显示了在增减模式下何时生成事件以及时基如何响应EPWMxSYNCl信号。

Figure 12. Time-Base Up-Down-Count Waveforms, TBCTL[PHSDIR = 0] Count Down On Synchronization Event



不考虑EPWMxSYNCl信号时，产生CTR=zero事件的频率等于CTR=PRD的频率，为

$$TBCLK/(TBPRD * 2) = 25/(208 * 2) = 0.06MHz$$

寄存器ETSEL和ETPS各字段的数值及其含义

ETSEL:

- ETSEL.bit.SOCAEN: 使能ePWMxSOCA脉冲，值为1时使能。
- ETSEL.bit.SOCASEL: 选择SOC来源，值为2时，TBCTR=TBPRD时产生ePWMxSOCA。
- ETSEL.bit.INTSEL: 选择产生中断的事件，值为2时选择计数器等于PRD的事件。
- ETSEL.bit.INTEN: 使能ePWMx_INT产生，值为0时禁用。

ETPS:

- ETPS.bit.SOCAPRD: 产生ePWMxSOCA脉冲需要的事件数，值为3时需要3个事件。

- ETPS.bit.INTPRD：产生ePWMx_INT中断需要的事件数，值为3时需要3个事件。

ADCSOC信号的产生频率

根据这行代码：

```
// EPWM SOCA enable bit for SEQ1  
AdcRegs.ADCTRL2.bit.EPWM_SOCA_SEQ1=1; //允许epwm_SOCA触发SEQ1
```

可以了解到EPWM SOCA信号被配置为可以触发ADC的SEQ1采样序列的启动。

EPWM SOCA信号是通过ePWM的事件时间基准(ET)寄存器产生的。在上面的代码中，已经将ePWM的时间基准(TB)计数器的周期设为208，即TBPRD=208。因此，EPWM SOCA信号的频率可以根据以下公式计算：

$$f = TBCLK / (TBPRD * 2 * ETPS.SOCAPRD)$$

其中，TBCLK是时间基准时钟频率，由以下公式计算：

$$TBCLK = SYSCLKOUT / (HSPCLKDIV * CLKDIV)$$

在代码中，HSPCLKDIV被设为6，CLKDIV被设为1，因此：

$$TBCLK = SYSCLKOUT / (HSPCLKDIV * CLKDIV) = 150MHz / (6 * 1) = 25MHz$$

将这些值代入前面的公式，可得：

$$f = 25MHz / (208 * 2 * 3) = 20kHz$$

因此，EPWM SOCA信号的产生频率是20kHz。

此程序运行后所产生的效果

该程序使用DSP2833x芯片实现了EPWM触发AD采集，并通过无缓存DA输出采集的数据。程序中，通过配置EPWM1的上升下降模式和事件产生间隔来实现AD的采样频率控制。通过初始化AD采样和EPWM1的参数，并通过PIE控制器和中断向量表，实现了EPWM1定时触发AD采样和数据传输的功能，控制了采样频率为20kHz。该程序还初始化了SPI通讯和AD9747数字模拟转换器，用于输出采集到的数据。最终整体实现了以1kHz频率、1V幅度和500mV偏移量的信号源的AD采集和DA输出。