MAIN

```c
int main(void)
{
    int i;
    unsigned short val = 0;

    etpwmInit();

    for(;;)
    {
        etpwmStartTBCLK();

        for(i = 0;i<100000;i++);

        etpwmSetCmpB(etpwmREG2, val);
        val++;
        etpwmStopTBCLK();

        if(val == 500)
            val = 0;
    }

    return 0;
}
```

```c
void etpwmInit(void)
{
/* USER CODE BEGIN (1) */
/* USER CODE END */

    /** @b initialize @b ETPWM2 */

    /** - Sets high speed time-base clock prescale bits */
    etpwmREG2->TBCTL = (uint16)0U << 7U;

    /** - Sets time-base clock prescale bits */
    etpwmREG2->TBCTL |= (uint16)((uint16)0U << 10U);

    /** - Sets time period or frequency for ETPWM block both PWMA and
    etpwmREG2->TBPRD = 374U;

    /** - Setup the duty cycle for PWMA */
    etpwmREG2->CMPA = 188U;

    /** - Setup the duty cycle for PWMB */
    etpwmREG2->CMPB = 188U;

    /** - Force EPWMxA output high when counter reaches zero and low
    etpwmREG2->AQCTLA = ((uint16)((uint16)ActionQual_Set   << 0U)
                       | (uint16)((uint16)ActionQual_Clear << 4U));

    /** - Force EPWMxB output high when counter reaches zero and low
    etpwmREG2->AQCTLB = ((uint16)((uint16)ActionQual_Set   << 0U)
                       | (uint16)((uint16)ActionQual_Clear << 8U));

    /** - Mode setting for Dead Band Module
```

1. etpwmInit ()함수를 살펴보자.

1.1 etpwmInit () – TBCTL

etpwmREG2-> TBCTL

35.4.7.1 PWM-Chopper Control Register (PCCTL)

Figure 35-90. PWM-Chopper Control Register (PCCTL) [offset = 3Eh]

15	11	10	8
Reserved		CHPDUTY	
R-0		R/W-0	

7	5	4	1	0
CHPFREQ		OSHTWTH		CHPEN
R/W-0		R/W-0		R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 35-50. PWM-Chopper Control Register (PCCTL) Bit Descriptions

Bits	Name	Value	Description
15-11	Reserved	0	Reserved
10-8	CHPDUTY		Chopping Clock Duty Cycle.
		0	Duty = 1/8 (12.5%)
		1h	Duty = 2/8 (25.0%)
		2h	Duty = 3/8 (37.5%)
		3h	Duty = 4/8 (50.0%)
		4h	Duty = 5/8 (62.5%)
		5h	Duty = 6/8 (75.0%)
		6h	Duty = 7/8 (87.5%)
		7h	Reserved
7-5	CHPFREQ		Chopping Clock Frequency.
		0	Divide by 1 (no prescale, = 12.5 MHz at 100 MHz VCLK3)
		1h	Divide by 2 (6.25 MHz at 100 MHz VCLK3)
		2h	Divide by 3 (4.16 MHz at 100 MHz VCLK3)
		3h	Divide by 4 (3.12 MHz at 100 MHz VCLK3)
		4h	Divide by 5 (2.50 MHz at 100 MHz VCLK3)
		5h	Divide by 6 (2.08 MHz at 100 MHz VCLK3)
		6h	Divide by 7 (1.78 MHz at 100 MHz VCLK3)
		7h	Divide by 8 (1.56 MHz at 100 MHz VCLK3)

1.2 etpwmInit () – TBPRD

etpwmREG2-> TBPRD : 동작주파수를 결정한다. 이때 HCG로 주기를 변경시켜봤을때 값이 바뀌나 확인해본다.

35.4.1.4 Time-Base Period Register (TBPRD)

Figure 35-66. Time-Base Period Register (TBPRD) [offset = 08h]

15	0
TBPRD	
R/W-0	

LEGEND: R/W = Read/Write; -n = value after reset

Table 35-26. Time-Base Period Register (TBPRD) Field Descriptions

Bits	Name	Description
15-0	TBPRD	These bits determine the period of the time-base counter. This sets the PWM frequency. Shadowing of this register is enabled and disabled by the TBCTL[PRDLD] bit. By default this register is shadowed. then the shadow is enabled and any write or read will automatically go to the case, the active register will be loaded from the shadow register when the time- then the shadow is disabled and any write or read will go directly to the active register, that is the register actively controlling the hardware. • The active and shadow registers share the same memory map address.

이 비트는 시간축 카운터의주기를 결정합니다. 이것은 PWM 주파수를 설정합니다.

이 레지스터의 음영 처리는 TBCTL [PRDLD] 비트에 의해 활성화 및 비활성화됩니다. 기본적으로 이 레지스터는 음영 처리됩니다.

PWM 주파수는 Time-base period(TBPRD) 레지와 Time-base Counter 모드에 의해 제어 된다.

1.3 etpwmInit () - CMPA

etpwmREG2-> CMPA :

35.4.2.2 Counter-Compare A Register (CMPA)

Figure 35-69. Counter-Compare A Register (CMPA) [offset = 10h]

15	0
CMPA	
R/W-0	

LEGEND: R/W = Read/Write; -n = value after reset

Table 35-29. Counter-Compare A Register (CMPA) Field Descriptions

Bits	Name	Description
15-0	CMPA	The value in the active CMPA register is continuously compared to the time-base counter (TBCTR). When the values are equal, the counter-compare module generates a "time-base counter equal to counter compare A" event. This event is sent to the action-qualifier where it is qualified and converted it into one or more actions. These actions can be applied to either the EPWMxA or the EPWMxB output depending on the configuration of the AQCTLA and AQCTLB registers. The actions that can be defined in the AQCTLA and AQCTLB registers include: • Do nothing; the event is ignored. • Clear: Pull the EPWMxA and/or EPWMxB signal low. • Set: Pull the EPWMxA and/or EPWMxB signal high. • Toggle the EPWMxA and/or EPWMxB signal. Shadowing of this register is enabled and disabled by the CMPCTL[SHDWAMODE] bit. By default this register is shadowed. • If CMPCTL[SHDWAMODE] = 0, then the shadow is enabled and any write or read will automatically go to the shadow register. In this case, the CMPCTL[LOADAMODE] bit field determines which event will load the active register from the shadow register. • Before a write, the CMPCTL[SHDWAFULL] bit can be read to determine if the shadow register is currently full. • If CMPCTL[SHDWAMODE] = 1, then the shadow register is disabled and any write or read will go directly to the active register, that is the register actively controlling the hardware. • In either mode, the active and shadow registers share the same memory map address.

1.4 etpwmInit () – AQCTLA - 45분

```
etpwmREG2->AQCTLA = ((uint16)((uint16)ActionQual_Set   << 0U)
                     | (uint16)((uint16)ActionQual_Clear << 4U));


etpwmREG2->AQCTLB = ((uint16)((uint16)ActionQual_Set   << 0U)
                     | (uint16)((uint16)ActionQual_Clear << 8U));
```

1.5 etpwmInit () – DBCTL 46분20초

```
etpwmREG2->DBCTL = ((uint16)((uint16)0U << 5U)    /* Source for Falling edge delay(0-PWMA, 1-PWMB) */
                  | (uint16)((uint16)0U << 4U) /* Source for Rising edge delay(0-PWMA, 1-PWMB)  */
                  | (uint16)((uint16)0U << 3U)  /* Enable/Disable EPWMxB invert       */
                  | (uint16)((uint16)0U << 2U)  /* Enable/Disable EPWMxA invert       */
                  | (uint16)((uint16)0U << 1U) /* Enable/Disable Rising Edge Delay   */
                  | (uint16)((uint16)0U << 0U));   /* Enable/Disable Falling Edge Delay  */
```

35.4.4.1 Dead-Band Generator Control Register (DBCTL)

Figure 35-75. Dead-Band Generator Control Register (DBCTL) [offset = 1Ch]

15	14	8
HALFCYCLE	Reserved	
R/W-0	R-0	

7	6	5	4	3	2	1	0
Reserved		IN_MODE		POLSEL		OUT_MODE	
R-0		R/W-0		R/W-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 35-35. Dead-Band Generator Control Register (DBCTL) Field Descriptions

Bits	Name	Value	Description
15	HALFCYCLE		Half Cycle Clocking Enable Bit.
		0	Full cycle clocking enabled. The dead-band counters are clocked at the TBCLK rate.
		1	Half cycle clocking enabled. The dead-band counters are clocked at TBCLK × 2.
14-6	Reserved	0	Reserved
5-4	IN_MODE		Dead Band Input Mode Control. Bit 5 controls the S5 switch and bit 4 controls the S4 switch shown in Figure 35-28. This allows you to select the input source to the falling-edge and rising-edge delay. To produce classical dead-band waveforms the default is EPWMxA In is the source for both falling and rising-edge delays.
		0	EPWMxA In (from the action-qualifier) is the source for both falling-edge and rising-edge delay.
		1h	EPWMxB In (from the action-qualifier) is the source for rising-edge delayed signal. EPWMxA In (from the action-qualifier) is the source for falling-edge delayed signal.
		2h	EPWMxA In (from the action-qualifier) is the source for rising-edge delayed signal. EPWMxB In (from the action-qualifier) is the source for falling-edge delayed signal.
		3h	EPWMxB In (from the action-qualifier) is the source for both rising-edge delay and falling-edge delayed signal.
3-2	POLSEL		Polarity Select Control. Bit 3 controls the S3 switch and bit 2 controls the S2 switch shown in Figure 35-28. This allows you to selectively invert one of the delayed signals before it is sent out of the dead-band submodule. The following descriptions correspond to classical upper/lower switch control as found in one leg of a digital motor control inverter. These assume that DBCTL[OUT_MODE] = 1,1 and DBCTL[IN_MODE] = 0,0. Other enhanced modes are also possible, but not regarded as typical usage modes.
		0	Active high (AH) mode. Neither EPWMxA nor EPWMxB is inverted (default).
		1h	Active low complementary (ALC) mode. EPWMxA is inverted.
		2h	Active high complementary (AHC). EPWMxB is inverted.
		3h	Active low (AL) mode. Both EPWMxA and EPWMxB are inverted.

1.6 etpwmInit () - DBRED 50분

```
/** - Set the rising edge delay  */
    etpwmREG2->DBRED = 110U;
```

➔ 전압이 팍 튈 때 데드라인을 delay주어 사인파를 사각파로 만든다 *

1.7 etpwmInit () - PCCTL

```
/** - Enable the chopper module for ETPWMx
    *     -Sets the One shot pulse width in a chopper modulated wave
    *     -Sets the dutycycle for the subsequent pulse train
    *     -Sets the period for the subsequent pulse train
    */
    etpwmREG2->PCCTL = ((uint16)((uint16)0U << 0U) /* Enable/Disable chopper module */
                     | (uint16)((uint16)0U << 1U)    /* One-shot Pulse Width */
                     | (uint16)((uint16)3U << 8U)  /* Chopping Clock Duty Cycle */
                     | (uint16)((uint16)0U << 5U));  /* Chopping Clock Frequency */
```

35.4.7 PWM-Chopper Submodule Register

35.4.7.1 PWM-Chopper Control Register (PCCTL)

Figure 35-90. PWM-Chopper Control Register (PCCTL) [offset = 3Eh]

15	11	10	8
Reserved		CHPDUTY	
R-0		R/W-0	

7	5	4	1	0
CHPFREQ		OSHTWTH		CHPEN
R/W-0		R/W-0		R/W-0

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 35-50. PWM-Chopper Control Register (PCCTL) Bit Descriptions

Bits	Name	Value	Description
15-11	Reserved	0	Reserved
10-8	CHPDUTY		Chopping Clock Duty Cycle.
		0	Duty = 1/8 (12.5%)
		1h	Duty = 2/8 (25.0%)
		2h	Duty = 3/8 (37.5%)
		3h	Duty = 4/8 (50.0%)
		4h	Duty = 5/8 (62.5%)
		5h	Duty = 6/8 (75.0%)
		6h	Duty = 7/8 (87.5%)
		7h	Reserved
7-5	CHPFREQ		Chopping Clock Frequency.
		0	Divide by 1 (no prescale, = 12.5 MHz at 100 MHz VCLK3)
		1h	Divide by 2 (6.25 MHz at 100 MHz VCLK3)
		2h	Divide by 3 (4.16 MHz at 100 MHz VCLK3)
		3h	Divide by 4 (3.12 MHz at 100 MHz VCLK3)
		4h	Divide by 5 (2.50 MHz at 100 MHz VCLK3)
		5h	Divide by 6 (2.08 MHz at 100 MHz VCLK3)
		6h	Divide by 7 (1.78 MHz at 100 MHz VCLK3)
		7h	Divide by 8 (1.56 MHz at 100 MHz VCLK3)

1.8 etpwmInit () - TZSEL

35.4.5.2 Trip-Zone Select Register (TZSEL)

Figure 35-79. Trip-Zone Select Register (TZSEL) [offset = 26h

15	14	13	12	11	10	
DCBEVT1	DCAEVT1	OSHT6	OSHT5	OSHT4	OSHT3	OS
R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/
7	6	5	4	3	2	
DCBEVT2	DCAEVT2	CBC6	CBC5	CBC4	CBC3	CE
R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 35-39. Trip-Zone Submodule Select Register (TZSEL) Field Des

Bits	Name	Value	Description
One-Shot (OSHT) Trip-zone enable/disable. When any of the enabled pins go low, a one-shot trip event oc module. When the event occurs, the action defined in the TZCTL register is taken on the EPWMxA and EP one-shot trip condition remains latched until the user clears the condition via the TZCLR register.			
15	DCBEVT1		Digital Compare Output B Event 1 Select.
		0	Disable DCBEVT1 as one-shot-trip source for this ePWM module.
		1	Enable DCBEVT1 as one-shot-trip source for this ePWM module.
14	DCAEVT1		Digital Compare Output A Event 1 Select.
		0	Disable DCAEVT1 as one-shot-trip source for this ePWM module.
		1	Enable DCAEVT1 as one-shot-trip source for this ePWM module.
13	OSHT6		Trip-zone 6 (TZ6) Select.
		0	Disable TZ6 as a one-shot trip source for this ePWM module.
		1	Enable TZ6 as a one-shot trip source for this ePWM module.
12	OSHT5		Trip-zone 5 (TZ5) Select.
		0	Disable TZ5 as a one-shot trip source for this ePWM module.
		1	Enable TZ5 as a one-shot trip source for this ePWM module.
11	OSHT4		Trip-zone 4 (TZ4) Select.
		0	Disable TZ4 as a one-shot trip source for this ePWM module.
		1	Enable TZ4 as a one-shot trip source for this ePWM module.
10	OSHT3		Trip-zone 3 (TZ3) Select.
		0	Disable TZ3 as a one-shot trip source for this ePWM module.
		1	Enable TZ3 as a one-shot trip source for this ePWM module.
9	OSHT2		Trip-zone 2 (TZ2) Select.
		0	Disable TZ2 as a one-shot trip source for this ePWM module.
		1	Enable TZ2 as a one-shot trip source for this ePWM module.
8	OSHT1		Trip-zone 1 (TZ1) Select.
		0	Disable TZ1 as a one-shot trip source for this ePWM module.
		1	Enable TZ1 as a one-shot trip source for this ePWM module.
Cycle-by-Cycle (CBC) Trip-zone enable/disable. When any of the enabled pins go low, a cycle-by-cycle trip ePWM module. When the event occurs, the action defined in the TZCTL register is taken on the EPWMxA a cycle-by-cycle trip condition is automatically cleared when the time-base counter reaches zero.			
7	DCBEVT2		Digital Compare Output B Event 2 Select.
		0	Disable DCBEVT2 as a CBC trip source for this ePWM module.
		1	Enable DCBEVT2 as a CBC trip source for this ePWM module.
6	DCAEVT2		Digital Compare Output A Event 2 Select.
		0	Disable DCAEVT2 as a CBC trip source for this ePWM module.
		1	Enable DCAEVT2 as a CBC trip source for this ePWM module.

The main signals used by the ePWM module are:

- **PWM output signals (EPWMxA and EPWMxB).**

  The PWM output signals are made available external to the device through the I/O Multiplexing Module (IOMM) as described in the IOMM chapter.

- **Trip-zone signals (TZ1 to TZ6).**

  These input signals alert the ePWM module of fault conditions external to the ePWM module. Each ePWM module can be configured to either use or ignore any of the trip-zone signals. The TZ1 to TZ3 trip-zone signals can be configured as asynchronous inputs, or double-synchronized using VCLK3, or double-synchronized and filtered through a 6-VCLK3-cycle counter before connecting to the ePWM modules. This selection is done by configuring registers in the IOMM. TZ4 is connected to an inverted eQEP1 error signal (EQEP1ERR), or to an inverted eQEP2 error signal (EQEP2ERR), or an OR-combination of EQEP1ERR and EQEP2ERR. This selection is also done via the IOMM registers. TZ5 is connected to the system clock fail status. This is asserted whenever an oscillator failure is detected, or a PLL slip is detected. TZ6 is connected to the debug mode entry indicator output from the CPU. This allows you to configure a trip action when the CPU halts.

- **Time-base synchronization input (EPWMxSYNCI) and output (EPWMxSYNCO) signals.**

  The synchronization signals daisy chain the ePWM modules together. Each module can be configured to either use or ignore its synchronization input. The clock synchronization input and output signal are brought out to pins only for ePWM1 (ePWM module #1). The synchronization output for ePWM1 (EPWM1SYNCO) is also connected to the SYNCI of the first enhanced capture module (eCAP1).

- **ADC start-of-conversion signals (EPWMxSOCA and EPWMxSOCB).**

  Each ePWM module has two ADC start of conversion signals. Any ePWM module can trigger a start of conversion. Which event triggers the start of conversion is configured in the Event-Trigger submodule of the ePWM.

- **Peripheral Bus**

  The peripheral bus is 32-bits wide and allows both 16-bit and 32-bit writes to the ePWM register file.

1. 처음에 프로그램 실행시 20ms 가나오지 않아 서보모터가 제대로 동작 하지 않는 것을 확인했다.
2. 모터는 원래 20ms 에서 동작해야 하기 때문에 설정에 문제가 있다.
3. TMS570LC4357 탭에서 PLL1→ HCLK 1 → VCLK3 14 를 사용한다.
4. 바꿔야 할 부분은 ETPWM 탭에서 HSPCLKDIV 값을 10 으로 설정해준다.
5. TB CLOCK 이 1M 가 나오게 해주어야 한다. (이유: TBPRD count 값이 2^16bit 까지만 표현 한다.)
   그래서 HSPCLKDIV 값으로 한번더 분주를 해주어 주파수를 낮추어 count 값이 작아지게 하는 것이다.
6. etpwmREG2->TBPRD = 19999U; 가 되어 있는지 확이 해야 한다. 약 20000 이라는 숫자가 나온것.
7. 이유: 1M clock 에서 몇번의 카운트가 되어야 20m/s 가 나오는가를 계산한다.
   T = 1/f 인데 여기서 주기는 1M 에서 x 번의 카운트 한 만큼이 되기때문.
   (20m/s = x/1M) 가 된다. (x = 20000) 이 나온다.