INITIALIZATION:

System clk is 90Mhz and PWM clk is also 90MHz.

```
#define PWM_CH  7
SysCtrlRegs.PCLKCR0.bit.HRPWMENCLK = 1;                         // HRPWM

for (j=1;j<PWM_CH;j++)
{
        (*ePWM[j]).TBPRD = 11250;

        if (j == 1)
        {
                (*ePWM[j]).TBPHS.half.TBPHS = 1;
        }
        else
        {
                (*ePWM[j]).TBPHS.half.TBPHS = 0;
        }

        (*ePWM[j]).TBCTL.bit.PHSEN = 1;
        (*ePWM[j]).TBCTL.bit.CTRMODE = 2;
        (*ePWM[j]).CMPCTL.bit.LOADAMODE = 2;
        (*ePWM[j]).AQCTLA.bit.CAD = 10;
        (*ePWM[j]).AQCTLA.bit.CAU = 01;
        (*ePWM[j]).DBCTL.bit.IN_MODE = 0;
        (*ePWM[j]).AQCSFRC.bit.CSFA = 0;
        (*ePWM[j]).AQCSFRC.bit.CSFB = 0;
        (*ePWM[j]).DBCTL.bit.OUT_MODE = 11;
        (*ePWM[j]).DBRED = 0;
        (*ePWM[j]).DBFED = 0;

        // HRPWM
        (*ePWM[j]).HRCNFG.all = 0x0000;
        (*ePWM[j]).HRCNFG.bit.EDGMODE = 3;
        (*ePWM[j]).HRCNFG.bit.HRLOAD = 2;
        (*ePWM[j]).HRPCTL.bit.TBPHSHRLOADE = 1;
        (*ePWM[j]).HRPCTL.bit.HRPE = 1;
        (*ePWM[j]).HRCNFG.bit.AUTOCONV = 1;
        (*ePWM[j]).HRMSTEP = 81;
}

Uint32 CMP;
float Frac_duty_cmpr = 0.0;
Uint16 CMP_val, CMPHR_val;

CMPHR_val = _IQ16(Frac_duty_cmpr);
CMP = ((long)CMP_val) << 16 | CMPHR_val;
EPwm1Regs.CMPA.all = CMP;
```

ISSUE 1:

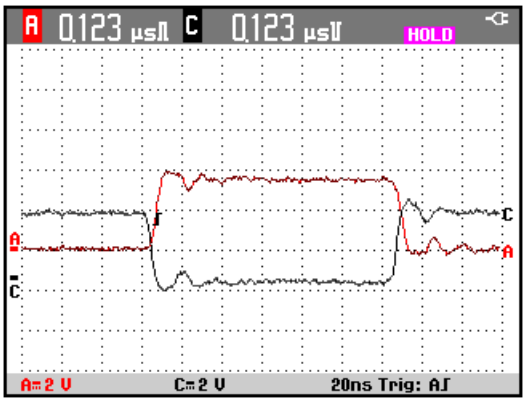
(*ePWM[j]).DBRED = 0;
(*ePWM[j]).DBFED = 0;

Frac_duty_cmpr = 0.0;
CMP_val = 5;

I got the value as below in CMP register.

Epwm1Regs.CMPA.half.CMPAHR = 0
Epwm1Regs.CMPA.half.CMPA = 5

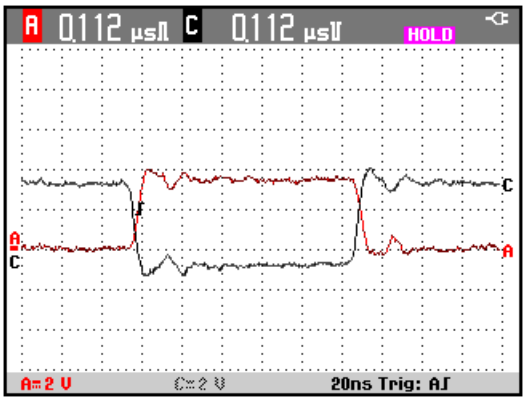
With the above setting, I am getting PWM1A (CH-A) and PWM1B (CH-C) as shown below. I am getting pulse width of **0.123uSec instead of 0.111uSec**.



if I change the setting as below.

(*ePWM[j]).HRCNFG.bit.EDGMODE = 2;
(*ePWM[j]).HRPCTL.bit.TBPHSHRLOADE = 0;
(*ePWM[j]).HRPCTL.bit.HRPE = 0;

I am getting PWM1A (CH-A) and PWM1B (CH-C) as shown below. I am getting pulse width of 0.111uSec. That is correct as per the CMPA value.



I want HRPWM on both edge. But if I enable HRPE then pulse width is not correct. If I disable it than getting proper pulse width. Why this happening??

ISSUE 2:

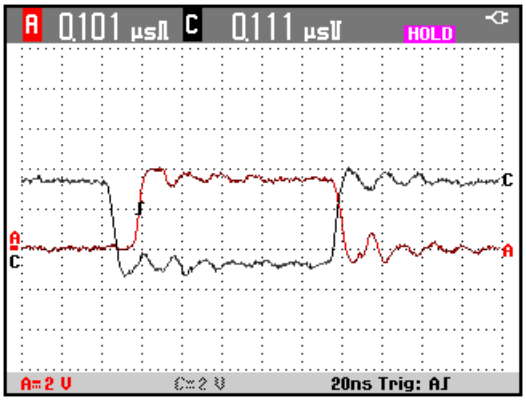
(*ePWM[j]).DBRED = 0;
(*ePWM[j]).DBFED = 0;

CMP_val = 5;

When I applied Frac_duty_cmpr = 0.005, I got the value as below in CMP register.

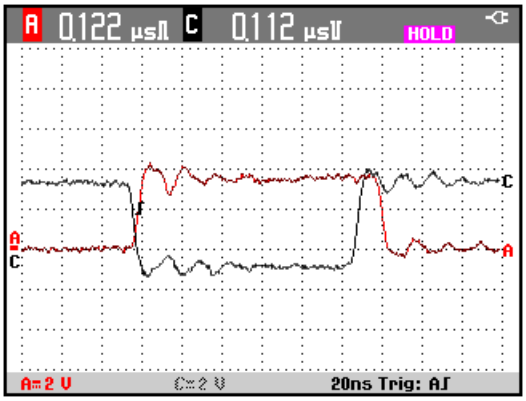
Epwm1Regs.CMPA.half.CMPAHR = 256
Epwm1Regs.CMPA.half.CMPA = 5

I am getting PWM1A (CH-A) and PWM1B (CH-C) as shown below. I am getting pulse width of 0.111uSec for PWM1B and 0.101uSec for PWM1A.



When I applied Frac_duty_cmpr = 0.999, I got the value as below in CMP register.

Epwm1Regs.CMPA.half.CMPAHR = 65280
Epwm1Regs.CMPA.half.CMPA = 5



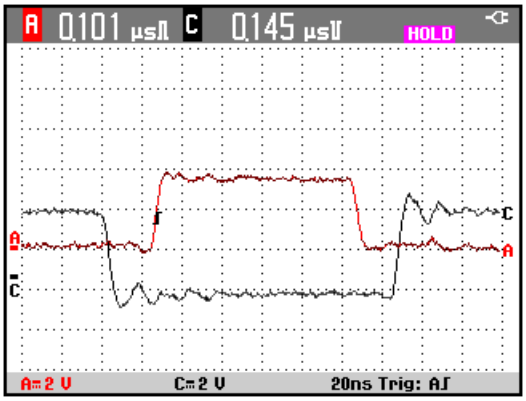
From Above two experiment, I observed that when I applied fraction part of duty cycle in CMPAHR then pulse width of PWM1B is reduced to 0.112uSec and that is remain constant upto 0.999. And PWM1A is reduced to 0.101uSec (22nSec reduced) for 0.005 and then linearly increase upto 0.122uSec as per the fraction part 0.999.
Question is if HRPWM is only applicable to Channel A than why pulse width of Channel B is changed (reduced) with fraction part of Channel A??
And Pulse width of PWM1A is 0.101uSec for 5.005 duty and 0.122uSec for 5.00 duty. Instead of increasing it decreased (0.022uSec).

ISSUE 3:

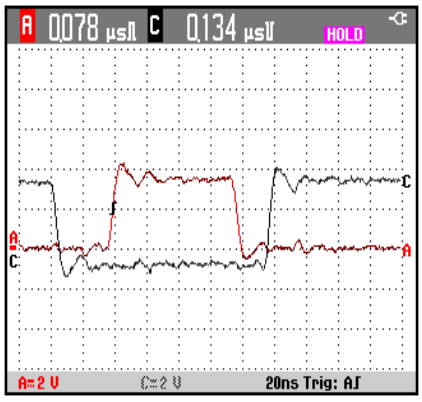
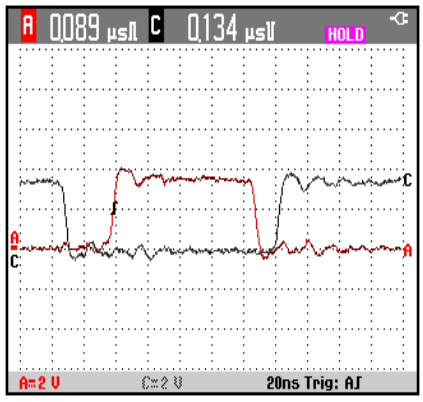
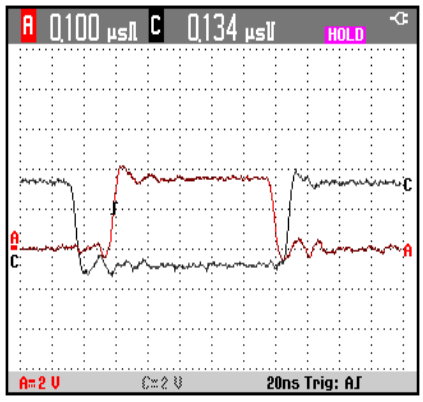
When I applied dead band of 2 clk on both edge,
(*ePWM[j]).DBRED = 2;
(*ePWM[j]).DBFED = 2;

Frac_duty_cmpr = 0.0;
CMP_val = 5;

I am getting pulse width as below.



When I applied fraction part of duty cycle, I got the result as below.

Frac_duty_cmpr = 0.005 CMPAHR = 256 CMPA = 5	Frac_duty_cmpr = 0.5 CMPAHR = 32768 CMPA = 5	Frac_duty_cmpr = 0.999 CMPAHR = 65280 CMPA = 5
		

From Above three experiment, I observed that when I applied fraction part of duty cycle in CMPAHR then pulse width of PWM1B is reduced to 0.134uSec and that is remain constant upto 0.999. And PWM1A is reduced to 0.078uSec (22nSec reduced) for 0.005 and then linearly increase upto 0.100uSec as per the fraction part 0.999.
Same things is happening with dead band also as mention in ISSUE2.