```
//==================================================
=========================================

// 선행처리 지시

//--------------------------------------------------
-----------------------------------------

#include "DSP28x_Project.h"                    // Device
Headerfile and Examples Include File



#define    SYSTEM_CLOCK        150E6    /* 150MHz */

#define    TBCLK               150E6    /* 150MHz */

//==================================================
=========================================







//==================================================
=========================================

//    함수 선언

//--------------------------------------------------
-----------------------------------------

void InitEPwm5Module(void);

interrupt void EPwm5Isr(void);



//isr함수 선언

interrupt void Xint3_isr(void);

interrupt void Xint4_isr(void);

interrupt void Xint5_isr(void);

interrupt void Xint6_isr(void);

//==================================================
=========================================
```

```
//==================================================
=========================================

// 시스템에서 사용할 전역 변수 선언

//--------------------------------------------------
-----------------------------------------


Uint16 Loop_cnt;

Uint16 SW1_cnt, SW2_cnt, SW3_cnt, SW4_cnt, SW1;

Uint16 ADC_value01;

Uint16    BackTicker;

Uint16    EPwm5IsrTicker;



float32 PWM_DUTY_RATIO_A = 0.2;

float32 PWM_CARRIER = 20E3;

float32    PwmDutyRatioA;

float32    FallingEdgeDelay;

float32    RisingEdgeDelay;



//==================================================
=========================================





//==================================================
=========================================

//    메인 함수 - 시작

//==================================================
=========================================

void main(void)

{


//==================================================
========================================

   // Step 1. Disable Global Interrupt

//------------------------------------------------------------------------------------------

   DINT;

   IER = 0x0000;

   IFR = 0x0000;

//========================================================================================




//========================================================================================

   // Step 2. 시스템 컨트롤 초기화:

//------------------------------------------------------------------------------------------

   InitSysCtrl();


   EALLOW;

   SysCtrlRegs.HISPCP.bit.HSPCLK  =  1;                  // HSPCLK = SYSCLKOUT/(HISPCP*2)

   GpioCtrlRegs.GPAPUD.bit.GPIO8  =  0;        /*  Enable pull-up on GPIO6 (EPWM5A) */

   GpioCtrlRegs.GPAPUD.bit.GPIO9  =  0;        /*  Enable pull-up on GPIO7 (EPWM5B) */

   GpioCtrlRegs.GPAMUX1.bit.GPIO8 = 1;     /* Configure GPIO6 as EPWM5A */

   GpioCtrlRegs.GPAMUX1.bit.GPIO9 = 1;     /* Configure GPIO7 as EPWM5B */

   EDIS;                                   // HSPCLK = 150MHz/(1*2) = 75MHz



//========================================================================================




//========================================================================================

   // Step 3. 인터럽트 초기화:

//------------------------------------------------------------------------------------------

   InitPieCtrl();

   IER = 0x0000;

   IFR = 0x0000;

/*--------------------------------------------------------------------------------

   Step 4

   4.1 Pie Vector Table Re-allocation

-------------------------------------------------------------------------------*/

   InitPieVectTable();




/*--------------------------------------------------------------------------------

   Step 5
```

```
  5.1 Interrupt Service routine re-mapping and Interrupt vector enable

------------------------------------------------------------------------------*/

  /* Interrupt Service Routine Re-mapping */

  EALLOW;

  PieVectTable.EPWM5_INT = &EPwm5Isr;

  EDIS;



  // Vector Remapping

  EALLOW;

  PieVectTable.XINT3 = &Xint3_isr;

  PieVectTable.XINT4 = &Xint4_isr;

  PieVectTable.XINT5 = &Xint5_isr;

  PieVectTable.XINT6 = &Xint6_isr;

  EDIS;

//===========================================================================================



  /* Enable PIE group 3 interrupt 4 for EPWM4_INT */

  PieCtrlRegs.PIEIER3.bit.INTx5 = 1;



  /* Enable CPU INT3 for EPWM4_INT */

  IER |= M_INT3;



/*------------------------------------------------------------------------------

  Step 6

  6.1 Initialize Periphrals for User Application


------------------------------------------------------------------------------*/

  /* Initialize EPWM4 Module */

  InitEPwm5Module();



//===========================================================================================

  // Step 3. ADC 초기화

//-------------------------------------------------------------------------------------------

  InitAdc();



  // ADC 설정

  AdcRegs.ADCTRL3.bit.ADCCLKPS = 3;                // ADCCLK = HSPCLK/(ADCCLKPS*2)/(CPS+1)

  AdcRegs.ADCTRL1.bit.CPS = 1;             // ADCCLK = 75MHz/(3*2)/(1+1) = 6.25MHz

  AdcRegs.ADCTRL1.bit.ACQ_PS = 3;              // 샘플/홀드 사이클 = ACQ_PS + 1 = 4 (ADCCLK기준)

  AdcRegs.ADCTRL1.bit.SEQ_CASC = 1;           // 시퀀스 모드 설정: 직렬 시퀀스 모드 (0:병렬 모드, 1:직렬 모드)

  AdcRegs.ADCMAXCONV.bit.MAX_CONV1 = 1;         // ADC 채널수 설정: 2개(=MAX_CONV+1)채널을 ADC

  AdcRegs.ADCCHSELSEQ1.bit.CONV00 = 0;          // ADC 순서 설정: 1번째로 ADCINA0 채널을 ADC

  AdcRegs.ADCCHSELSEQ1.bit.CONV01 = 8;          // ADC 순서 설정: 2번째로 ADCINB0 채널을 ADC


//===========================================================================================
//===================================================
=========================================

   // GPIO 초기화


//---------------------------------------------------
-----------------------------------------

   EALLOW;

   GpioCtrlRegs.GPBMUX1.bit.GPIO44 = 0;       // 핀 기능
선택: GPIO44

   GpioCtrlRegs.GPBMUX1.bit.GPIO45 = 0;       // 핀 기능
선택: GPIO45

   GpioCtrlRegs.GPBMUX1.bit.GPIO46 = 0;       // 핀 기능
선택: GPIO46

   GpioCtrlRegs.GPBMUX1.bit.GPIO47 = 0;       // 핀 기능
선택: GPIO47

   GpioCtrlRegs.GPBDIR.bit.GPIO44  =  0;              //
GPIO44 입출력 선택: Input

   GpioCtrlRegs.GPBDIR.bit.GPIO45  =  0;              //
GPIO45 입출력 선택: Input

   GpioCtrlRegs.GPBDIR.bit.GPIO46  =  0;              //
GPIO46 입출력 선택: Input

   GpioCtrlRegs.GPBDIR.bit.GPIO47  =  0;              //
GPIO47 입출력 선택: Input

   EDIS;


//===================================================
=========================================






//===================================================
=========================================

   // Step 5. Qualification 초기화


//---------------------------------------------------
-----------------------------------------

   EALLOW;

   GpioCtrlRegs.GPBCTRL.bit.QUALPRD1  =  0xFF;       //
(GPIO40~GPIO47) Qual period 설정

   GpioCtrlRegs.GPBQSEL1.bit.GPIO44  =  2;           //
Qualification using 6 samples

   GpioCtrlRegs.GPBQSEL1.bit.GPIO45  =  2;           //
Qualification using 6 samples

   GpioCtrlRegs.GPBQSEL1.bit.GPIO46  =  2;           //
Qualification using 6 samples

   GpioCtrlRegs.GPBQSEL1.bit.GPIO47  =  2;           //
Qualification using 6 samples

   EDIS;


//===================================================
=========================================






//===================================================
=========================================

   // Step 6. XINT 초기화


//---------------------------------------------------
-----------------------------------------

   EALLOW;

   GpioIntRegs.GPIOXINT3SEL.bit.GPIOSEL  =  47;    // 외
부 인터럽트 XINT3로 사용할 핀 선택: GPIO47

   GpioIntRegs.GPIOXINT4SEL.bit.GPIOSEL  =  46;    // 외
부 인터럽트 XINT4로 사용할 핀 선택: GPIO46

   GpioIntRegs.GPIOXINT5SEL.bit.GPIOSEL  =  45;    // 외
부 인터럽트 XINT5로 사용할 핀 선택: GPIO45

   GpioIntRegs.GPIOXINT6SEL.bit.GPIOSEL  =  44;    // 외
```

```
부 인터럽트 XINT6로 사용할 핀 선택: GPIO44

   EDIS;



   XIntruptRegs.XINT3CR.bit.POLARITY = 0;          // XINT3 인터럽트 발생 조건 설정: 입력 신호의 하강 엣지

   XIntruptRegs.XINT4CR.bit.POLARITY = 0;          // XINT4 인터럽트 발생 조건 설정: 입력 신호의 상승 엣지

   XIntruptRegs.XINT5CR.bit.POLARITY = 0;          // XINT5 인터럽트 발생 조건 설정: 입력 신호의 하강 엣지

   XIntruptRegs.XINT6CR.bit.POLARITY = 0;          // XINT6 인터럽트 발생 조건 설정: 입력 신호의 하강 & 상승 엣지



   XIntruptRegs.XINT3CR.bit.ENABLE = 1;            // XINT3 인터럽트 : Enable

   XIntruptRegs.XINT4CR.bit.ENABLE = 1;            // XINT4 인터럽트 : Enable

   XIntruptRegs.XINT5CR.bit.ENABLE = 1;            // XINT5 인터럽트 : Enable

   XIntruptRegs.XINT6CR.bit.ENABLE = 1;            // XINT6 인터럽트 : Enable



                                                   // 외부 인터터트 포합된 백터 활성화

   PieCtrlRegs.PIEIER12.bit.INTx1 = 1;             // PIE 인터럽트(XINT3) : Enable

   PieCtrlRegs.PIEIER12.bit.INTx2 = 1;             // PIE 인터럽트(XINT4) : Enable

   PieCtrlRegs.PIEIER12.bit.INTx3 = 1;             // PIE 인터럽트(XINT5) : Enable

   PieCtrlRegs.PIEIER12.bit.INTx4 = 1;             // PIE 인터럽트(XINT6) : Enable

   IER |= M_INT12;                                 // CPU 인터럽트(INT12) : Enable

//=====================================================================================================

// Step 7. Initialize Application Variables

//-----------------------------------------------------------------------------------------------------

   SW1_cnt = 0;

   SW2_cnt = 0;

   SW3_cnt = 0;

   SW4_cnt = 0;

   SW1 = 0;

   Loop_cnt = 0;

   ADC_value01 = 0;

   BackTicker = 0;

   EPwm5IsrTicker = 0;



   PwmDutyRatioA = PWM_DUTY_RATIO_A;

   FallingEdgeDelay = (1.0 / TBCLK) * EPwm5Regs.DBFED;

   RisingEdgeDelay = (1.0 / TBCLK) * EPwm5Regs.DBRED;



//=====================================================================================================




//=====================================================================================================

   // Enable global Interrupts and higher priority real-time debug events:


//-----------------------------------------------------------------------------------------------------
```

```
   EINT;   // Enable Global interrupt INTM

   ERTM;   // Enable Global realtime interrupt DBGM


//=================================================
========================================




//=================================================
========================================

   // IDLE loop. Just sit and loop forever :


//-------------------------------------------------
----------------------------------------

   for (;;)

   {

      AdcRegs.ADCTRL2.bit.SOC_SEQ1 = 1;   // ADC 시퀀스 시작

      DELAY_US(0.64L);   // ADC 시퀀스 변환시간(약0.64usec)만큼지연

      ADC_value01 = AdcRegs.ADCRESULT0;   // ADC 결과 저장


      if (SW1 == 1 && SW3_cnt % 2 == 0 && SW4_cnt % 2 == 0)

      {


         EPwm5Regs.DBFED = 0;   /* 1usec, Falling Edge Delay */

         EPwm5Regs.DBRED = 0;   /* 1usec, Rising Edge Delay */

         FallingEdgeDelay = (1.0 / TBCLK) * EPwm5Regs.DBFED;

         RisingEdgeDelay = (1.0 / TBCLK) * EPwm5Regs.DBRED;


         PWM_CARRIER = 20E3;


         PwmDutyRatioA = ((float32)ADC_value01 / (float32)65535 )*(float32)0.475;

         EPwm5Regs.TBPRD = (TBCLK / PWM_CARRIER) / 2;

         EPwm5Regs.CMPA.half.CMPA = (EPwm5Regs.TBPRD + 1) * PwmDutyRatioA;

         EPwm5Regs.CMPB = (EPwm5Regs.TBPRD + 1) * (1 - PwmDutyRatioA);



         Loop_cnt++;


         BackTicker++;

         SW1_cnt = 0;

      }

      else if (SW1 == 1 && SW3_cnt % 2 == 1 && SW4_cnt % 2 == 0)

      {


         EPwm5Regs.DBFED = 0;

         EPwm5Regs.DBRED = 0;

         FallingEdgeDelay = (1.0 / TBCLK) * EPwm5Regs.DBFED;

         RisingEdgeDelay = (1.0 / TBCLK) * EPwm5Regs.DBRED;

         PWM_CARRIER = 20E3;

         PwmDutyRatioA = ((float32)ADC_value01 /
(float32)65535) * (float32)0.475;
         EPwm5Regs.TBPRD = (TBCLK / PWM_CARRIER) / 2;
         EPwm5Regs.CMPB = (EPwm5Regs.TBPRD + 1) * (1 - PwmDutyRatioA);



         Loop_cnt++;


         BackTicker++;
         SW1_cnt = 0;
      }
      else if (SW1 == 1 && SW3_cnt % 2 == 0 && SW4_cnt % 2 == 1)
      {
         EPwm5Regs.DBFED = 0;                /* 1usec, Falling Edge Delay */
         EPwm5Regs.DBRED = 0;                /* 1usec, Rising Edge Delay */
         FallingEdgeDelay = (1.0 / TBCLK) * EPwm5Regs.DBFED;
         RisingEdgeDelay = (1.0 / TBCLK) * EPwm5Regs.DBRED;
         PWM_CARRIER = 20E3;
         PwmDutyRatioA = ((float32)ADC_value01 / (float32)65535 )* (float32)0.475;
         EPwm5Regs.TBPRD = (TBCLK / PWM_CARRIER) / 2;
         EPwm5Regs.CMPA.half.CMPA = (EPwm5Regs.TBPRD + 1) * PwmDutyRatioA;


         Loop_cnt++;

         BackTicker++;
         SW1_cnt = 0;
      }


      else if (SW3_cnt % 2 == 1 && SW4_cnt % 2 == 1)
      {
         EPwm5Regs.CMPA.half.CMPA = 0;
         EPwm5Regs.CMPB = 5000;
         Loop_cnt++;
         BackTicker++;
      }
      //-------------------프로젝트2
      else if (SW1 == 2)
      {
         PWM_CARRIER = 100E3 + 200E3 * (float32)ADC_value01 / (float32)65535;
         EPwm5Regs.TBPRD = (TBCLK / PWM_CARRIER) / 4;
         EPwm5Regs.CMPA.half.CMPA = (EPwm5Regs.TBPRD + 1) * PwmDutyRatioA;
         EPwm5Regs.CMPB = (EPwm5Regs.TBPRD + 1) * (1 - PwmDutyRatioA);
         Loop_cnt++;
         BackTicker++;
         SW1_cnt = 0;
      }
      // -----------프로젝트3
      else if (SW1 >= 3)
      {

         EPwm5Regs.DBFED = 75 * (float32)ADC_value01 / (float32)65535;

         EPwm5Regs.DBRED = 75 * (float32)ADC_value01 / (float32)65535;

         FallingEdgeDelay = (1.0 / TBCLK) * EPwm5Regs.DBFED;

         RisingEdgeDelay = (1.0 / TBCLK) * EPwm5Regs.DBRED;

         EPwm5Regs.CMPA.half.CMPA = (EPwm5Regs.TBPRD + 1) * 0.5;

         EPwm5Regs.CMPB = (EPwm5Regs.TBPRD + 1) * (1 - 0.5);

         Loop_cnt++;

         BackTicker++;

         SW1_cnt = 0;

      }

      if (SW3_cnt % 2 == 1)

      {

         EPwm5Regs.CMPA.half.CMPA = 0;

      }



      if (SW4_cnt % 2 == 1)

      {

         EPwm5Regs.CMPB = 5000;

      }

  }

}

//   메인 함수 - 끝

//   ISR 함수 정의
```

```
//-----------------------------------------------------------------------------------------

interrupt void Xint3_isr(void)

{

    DINT;
    DELAY_US(500000);
    SW1_cnt++;
    SW1 = SW1_cnt;
    EINT;
    PieCtrlRegs.PIEACK.all = PIEACK_GROUP12;
}

interrupt void Xint4_isr(void)

{

    SW2_cnt++;

    PieCtrlRegs.PIEACK.all = PIEACK_GROUP12;

}

interrupt void Xint5_isr(void)

{

    SW3_cnt++;

    PieCtrlRegs.PIEACK.all = PIEACK_GROUP12;

}

interrupt void Xint6_isr(void)

{
    SW4_cnt++;

    PieCtrlRegs.PIEACK.all = PIEACK_GROUP12;

}
interrupt void EPwm5Isr(void)
{
    EPwm5IsrTicker++;

    /* Clear INT flag for this timer */

    EPwm5Regs.ETCLR.bit.INT = 1;

    /* Acknowledge this interrupt to receive more interrupts from group 3 */
```

```
   PieCtrlRegs.PIEACK.bit.ACK3 = 1;

}

void InitEPwm5Module(void)

{
   /* Setup Counter Mode and Clock */

   EPwm5Regs.TBCTL.bit.CTRMODE = 2;        /* Count
Up (Asymmetric) */

   EPwm5Regs.TBCTL.bit.HSPCLKDIV = 0;       /* TBCLK
= SYSCLKOUT / (HSPCLKDIV * CLKDIV) = 150MHz */

   EPwm5Regs.TBCTL.bit.CLKDIV = 0;

   /* Setup Phase */

   EPwm5Regs.TBPHS.half.TBPHS = 0;          /* Phase
is 0 */

   EPwm5Regs.TBCTL.bit.PHSEN = 0;          /* Disable
phase loading */

                              /*     Setup     Period
(Carrier Frequency) */

   EPwm5Regs.TBPRD = (TBCLK / PWM_CARRIER) / 4;
/* Set Timer Period, (150MHz/20KHz)/4 = 1,875 () */

   EPwm5Regs.TBCTR = 0;                      /* Clear
Counter */

                              /*  Set  Compare  Value
*/

   EPwm5Regs.CMPA.half.CMPA                          =
(Uint16)((EPwm5Regs.TBPRD        +        1)        *
PWM_DUTY_RATIO_A);    /* Set Compare A Value to
50% */

   EPwm5Regs.CMPB = (Uint16)((EPwm5Regs.TBPRD + 1)
*(1 - PWM_DUTY_RATIO_A));



   /* Set actions */

   EPwm5Regs.AQCTLA.bit.CAU  =  1;           /*  Set
EPWM5A on CNTR=Zero */

   EPwm5Regs.AQCTLA.bit.CAD  =  2;          /*  Clear
EPWM5A on CNTR=CMPA, Up-Count */

   EPwm5Regs.AQCTLB.bit.CBU = 1;      /* Set EPWM5A
on CNTR=Zero */

   EPwm5Regs.AQCTLB.bit.CBD  =  2;          /*  Clear
EPWM5A on CNTR=CMPA, Up-Count */

/* Set Dead-time */

   EPwm5Regs.DBCTL.bit.IN_MODE = 2;      /* EPWMxA
is the source for both falling-edge & rising-edge delay
*/

   EPwm5Regs.DBCTL.bit.OUT_MODE  =  3;           /*
Dead-band is fully enabled for both rising-edge delay
on EPWMxA and falling-edge delay on EPWMxB */

   EPwm5Regs.DBCTL.bit.POLSEL = 2;          /* Active
High Complementary (AHC). EPWMxB is inverted */

   EPwm5Regs.DBFED = 450;                  /* 3usec,
Falling Edge Delay */

   EPwm5Regs.DBRED = 450;                  /* 3usec,
Rising Edge Delay */

                             /* Set Interrupts */

   EPwm5Regs.ETSEL.bit.INTSEL = 1;      /* Select INT
on CNTR=Zero */

   EPwm5Regs.ETPS.bit.INTPRD  =  1;        /* Generate
INT on 1st event */

   EPwm5Regs.ETSEL.bit.INTEN = 1;       /* Enable INT
*/

}
```