**Functions for calling ADC**

DrvADC\_Open(ADC\_SINGLE\_END,ADC\_SINGLE\_OP , 0x80,INTERNAL\_HCLK , 1);

**//choose channel, mode . clock ,division**

DrvADC\_StartConvert(); // start A/D conversion

while(DrvADC\_IsConversionDone()==FALSE);

value = ADC->ADDR[7].RSLT & 0xFFF;

SYSCLK->PWRCON.XTL12M\_EN = 1; **//Enable 12Mhz and set HCLK->12Mhz**

SYSCLK->CLKSEL0.HCLK\_S = 0;

or

**void InitADC(void)**

{

/\* Step 1. GPIO initial \*/

GPIOA->OFFD|=0x00800000; **//Disable digital input path**

SYS->GPAMFP.ADC7\_SS21\_AD6=1; **//Set ADC function**

/\* Step 2. Enable and Select ADC clock source, and then enable ADC module \*/

SYSCLK->CLKSEL1.ADC\_S = 2; **//Select 22Mhz for ADC**

SYSCLK->CLKDIV.ADC\_N = 1; **//ADC clock source = 22Mhz/2 =11Mhz;**

SYSCLK->APBCLK.ADC\_EN = 1; **//Enable clock source**

ADC->ADCR.ADEN = 1; **//Enable ADC module**

**/\* Step 3. Select Operation mode \*/**

ADC->ADCR.DIFFEN = 0; **//single end input**

ADC->ADCR.ADMD = 0; **//single mode**

**/\* Step 4. Select ADC channel \*/**

ADC->ADCHER.CHEN = 0x80;

**/\* Step 5. Enable ADC interrupt \*/**

ADC->ADSR.ADF =1; **//clear the A/D interrupt flags for safe**

ADC->ADCR.ADIE = 1;

NVIC\_EnableIRQ(ADC\_IRQn);

**/\* Step 6. Enable WDT module \*/**

ADC->ADCR.ADST=1;

}

**Setting up clock for ADC**

UNLOCKREG();

SYSCLK->PWRCON.XTL12M\_EN = 1; //Enable 12Mhz and set HCLK->12Mhz

SYSCLK->CLKSEL0.HCLK\_S = 0;

LOCKREG();

**Checking flag and collecting result**

while(ADC->ADSR.ADF==0); // ADC Flag, wait till 1 (A/DC conversion done)

ADC->ADSR.ADF=1; // write 1 to ADF is to clear the flag

adc\_value=ADC->ADDR[7].RSLT; // input 12-bit ADC value

**print using lcd**

sprintf(TEXT,"Value: %d",value); // convert ADC0 value into text

print\_lcd(1, TEXT); // output TEXT to LCD

**PWM0**

**void InitPWM(void)**

{

**/\* Step 1. GPIO initial \*/**

SYS->GPAMFP.PWM0\_AD13=1;

**/\* Step 2. Enable and Select PWM clock source\*/**

SYSCLK->APBCLK.PWM01\_EN = 1;//Enable PWM clock

SYSCLK->CLKSEL1.PWM01\_S = 3;//Select 22.1184Mhz for PWM clock source

PWMA->PPR.CP01=1; **//Prescaler 0~255, Setting 0 to stop** output clock

PWMA->CSR.CSR0=0; // PWM clock = clock source/(Prescaler + 1)/divider

/\* Step 3. Select PWM Operation mode \*/

//PWM0

PWMA->PCR.CH0MOD=1; **//0:One-shot mode, 1:Auto-load mode**

//CNR and CMR will be auto-cleared after setting CH0MOD form 0 to 1.

PWMA->CNR0=0xFFFF;

PWMA->CMR0=0xFFFF;

PWMA->PCR.CH0INV=0; //Inverter->0:off, 1:on

PWMA->PCR.CH0EN=1; //PWM function->0:Disable, 1:Enable

PWMA->POE.PWM0=1; //Output to pin->0:Diasble, 1:Enable

}

**Setting up clock for pwm**

UNLOCKREG();

SYSCLK->PWRCON.XTL12M\_EN = 1;

SYSCLK->CLKSEL0.HCLK\_S = 0;

LOCKREG();

**Input from ADC to PWM 0 when convertion is over for**

while(ADC->ADSR.ADF==0);

ADC->ADSR.ADF=1;

PWMA->CMR0=ADC->ADDR[7].RSLT<<4;