4.5. ADC con TivaWare .

Las funciones de TivaWare para el convertidor ADC facilitan el manejo del módulo. Estas funciones están divididas en funciones para configuración del secuenciador, funciones para inicio de conversión y funciones que tienen que ver con interrupciones.

Las funciones del secuenciador son:

void ADCSequenceConfigure ( uint32\_t ui32Base , uint32\_t ui32SequenceNum , uint32\_t ui32Trigger , uint32\_t ui32Priority ) . Configura que va a iniciar la conversión y la prioridad de la secuencia de muestreo .

ui32Base es el módulo ADC ( puede serADC0\_BASEoADC1\_BASE ) , ui32SequenceNum es el número del secuenciador ( puede ser 0 , 1 , 2 o 3 ) , ui32Triggeres la fuente que inicia la secuencia ( puede serADC\_TRIGGER\_PROCESSOR - se inicia por el procesador con la funciónADCProcessorTrigger ( ) ) .

Hay otras fuentes que pueden hacer que se inicie la conversión, como una comparación analógica, un PWM , un contador , etc . El parámetro ui32Priority sirve para determinar la prioridad de conversión cuando se usan varias secuencias . Este parámetro puede ser 0 , para máxima prioridad hasta 3 , que es la mínima prioridad .

Otras funciones con ADC son las siguientes : ADCSequenceStepConfigure ( ADC0\_BASE , 1 , 0 , ADC\_CTL\_TS ) ;

ADCSequenceEnable ( ADC0\_BASE , 1 ) ;

ADCIntClear ( ADC0\_BASE , 1 ) ;

ADCProcessorTrigger ( ADC0\_BASE , 1 ) ;

ADCIntStatus ( ADC0\_BASE , 1 , false ) ADCSequenceDataGet ( ADC0\_BASE , 1 , ui32ADC0Value ) ;

ADCHardwareOversampleConfigure ( ADC0\_BASE , 64 ) ;

Un ejemplo del uso del ADC con Tivaware es el siguiente :

# include < stdint.h >

# include < stdbool.h >

# include " inc / hw\_memmap.h "

# include " inc / hw\_types.h "

# include " driverlib / debug.h "

# include " driverlib / sysctl.h "

# include " driverlib / adc.h "

int main ( void ) {

uint32\_t ui32ADC0Value [ 4 ] ; / / aqui quedan los valores convertidos .

volatile uint32\_t ui32TempAvg ;

volatile uint32\_t ui32TempValueC ;

/ / reloj a 40Mhz

SysCtlClockSet ( SYSCTL\_SYSDIV\_5 | SYSCTL\_USE\_PLL | SYSCTL\_OSC\_MAIN | SYSCTL\_XTAL\_16MHZ ) ;

/ / habilitación del ADC :

SysCtlPeripheralEnable ( SYSCTL\_PERIPH\_ADC0 ) ;

/ / Usa ADC0 , secuenciador 1 , el procesador inicia la conversion , prioridad maxima ( 0 ) .

ADCSequenceConfigure ( ADC0\_BASE , 1 , ADC\_TRIGGER\_PROCESSOR , 0 ) ;

/ / Se van a tomar cuatro muestras del mismo sensor de temperatura para promediar la medida / / configura al ADC0 con el secuenciador 1 , la secuencia 0 y el sensor de temperatura

ADCSequenceStepConfigure ( ADC0\_BASE , 1 , 0 , ADC\_CTL\_TS ) ;

/ / segunda muestra

ADCSequenceStepConfigure ( ADC0\_BASE , 1 , 1 , ADC\_CTL\_TS ) ;

/ / tercera muestra

ADCSequenceStepConfigure ( ADC0\_BASE , 1 , 2 , ADC\_CTL\_TS ) ;

/ / última muestra. Se habilita interrupción y se indica que es la última muestra

ADCSequenceStepConfigure ( ADC0\_BASE , 1,3 , ADC\_CTL\_TS | ADC\_CTL\_IE | ADC\_CTL\_END ) ;

/ / Habilita ADC0 , secuencia 1 . ADCSequenceEnable ( ADC0\_BASE , 1 ) ;

while ( 1 ) {

/ / Borra bandera de interrupción del ADC . ADCIntClear

( ADC0\_BASE , 1 ) ;

/ / Inicia la conversión.

ADCProcessorTrigger ( ADC0\_BASE , 1 ) ;

/ / Espera hasta que la conversión termine .

while ( ! ADCIntStatus ( ADC0\_BASE , 1 , false ) )

{

}

/ / Lee los valores convertidos de la secuencia .

ADCSequenceDataGet ( ADC0\_BASE , 1 , ui32ADC0Value ) ;

/ / Calcula el valor promedio . Agrega 0.5 para redondear el valor a entero .

ui32TempAvg = ( ui32ADC0Value [ 0 ] + ui32ADC0Value [ 1 ] + ui32ADC0Value [ 2 ] + ui32ADC0Value [ 3 ] + 2 ) / 4 ;

/ / Esta ecuación aparece en las especificaciones del microcontrolador . ui32TempValueC = ( 1475 - ( ( 2475 \* ui32TempAvg ) ) / 4096 ) / 10 ;

}

}