



Comentario Técnico: CTC-075

Título: Utilización de ADC en HT32F125x

Autor: Sergio R. Caprile, Senior Engineer

Revisiones	Fecha	Comentarios
0	29/02/12	

En este comentario técnico describimos la forma de utilización del área conversor AD en el HT32F125x de Holtek

Al momento de publicación de este documento, la literatura de Holtek no es del todo clara en el uso del conversor AD, particularmente en modo polled. Este Comentario Técnico intenta aclarar y proveer un ejemplo de utilización.

### Detección de fin de conversión y lectura

Para esto debemos observar los bits *ADIRAW* en el registro *ADCIRAW*, según el tipo de conversión. Este bit se setea al finalizar la conversión, y debe ser reseteado manualmente.

El dato convertido se almacena en un registro interno y requiere de algunos ciclos de clock del ADC para pasar al registro de datos. Por este motivo, si se lee inmediatamente el registro de datos, se obtiene un dato anterior. Esto puede comprobarse observando el bit *ADVLD* de dicho registro, que se setea al escribirse el nuevo valor. Esto no ocurre si se lee este registro en el handler de interrupciones del ADC, dado que la interrupción se produce luego de almacenado el nuevo valor.

### Ejemplos de uso, modo polled

#### Inicio de conversión

```
// reset ADIRAWC y set ADSC para iniciar conversión
void ADC_start(void)
{
    HT_ADC->ICLR=(1<<2);
    HT_ADC->TSR |= (1<<0);
}
```

#### Detección de fin de conversión

```
// lee ADIRAWC
#define ADC_ready()    (HT_ADC->IRAW&(1<<2))
```

#### Lectura del dato convertido

```
// lee DR[ch]
#define ADC_read(ch)  ((uint16_t)HT_ADC->DR[ch])
```

#### Chequeo de validez del dato convertido

```
// test DR[ch]
#define ADC_valid(ch) (HT_ADC->DR[ch]&(1U<<31))
```

#### Espera por validez del dato convertido

```
uint16_t ADC_waitvalid(unsigned int ch)
{
    uint32_t val;

    while(!((val=HT_ADC->DR[ch])&(1U<<31)));
    return((uint16_t)val);
}
```

#### Utilización

```

main()
{
    int i;

    SysTick_Setup();
    init_ADC();
    ADC_start(); // inicia conversión
    while(1){
        while(!ADC_ready());
        // podemos esperar un tiempo razonable (algunos ciclos del clock del ADC)
        Delay(2);
        i=ADC_read(0); // lee resultado
        // o leer el flag hasta que esté seteado
        i=ADC_waitvalid(0);
        Delay(1000);
        ADC_start(); // inicia conversión
    }
}

```

### Ejemplos de uso, por interrupciones

#### Inicio de conversión

```

// set ADSC para iniciar conversión
void __INLINE ADC_start(void)
{
    HT_ADC->TSR |= (1<<0);
}

```

#### Handler

```

void ADC_IRQHandler(void)
{
    static uint16_t d;

    HT_ADC->ICLR = (1<<2); // reset ADIRAWC
    d=(uint16_t)HT_ADC->DR[0]; // obtiene resultado
}

```

#### Utilización

```

main()
{
    SysTick_Setup();
    init_ADC();
    ADC_start(); // inicia conversión
    while(1){
        Delay(1000);
        ADC_start(); // inicia conversión
    }
}

```