**1. Configuración del ADC**

**1. Introducción:**

El ADC es un periférico que convierte una señal analógica de entrada en una digital de n bits.

**El LPC1768 dispone de un conversor A/D de 12 bits:**

- Conversor de aproximaciones sucesivas

- 8 entradas multiplexadas.

- Frecuencia máxima de conversión de 200KHz

- Tiempo mínimo de conversión 5us

65 ciclos del reloj de entrada que debe ser menor o igual a 13MHz

- SPAN de VREFN a VREFP (no debe superar VDDA)

- Posibilidad de conversión en modo ráfaga (“Burst”)

- Múltiples posibles señales de inicio de conversión.

- Para configurar el ADC debemos alimentarlo con como cualquier otro periférico y además activarlo en modo normal con

- La frecuencia con que alimentemos el ADC condiciona su tiempo de conversión y la frecuencia máxima a la que puede convertir .

Elegimos el PCLK con PCLKSEL y un preescaler de 8 bits que posee el ADC llamado **.**

- El ADC tiene 8 canales de conversión multiplexados (multiplexor interno)

Para seleccionar el canal a convertir (solo uno a la vez) utilizamos los 8 bits del ADCR

· En modo Normal, solo podemos seleccionar uno de los canales a la vez para ser convertidos.

· En modo Burst, seleccionamos todos los canales que queramos sean convertidos en ráfaga, los canales seleccionados se muestrearan continuamente a máxima frecuencia sin necesidad de señal de conversión. (O eso dice). Para activar este modo usamos el bit del ADCR

- Podemos configurar GPIOs para que sean entrada de estos canales AD0.

· Los pines de entrada deben ser configurados sin pull-up ni pull-down

- La señal de Inicio de Conversión es la que indica al ADC cuando realizar la conversión, podemos seleccionar señales tanto internas como externas con los bits del ADCR

También se lo podemos indicar por software. Si la IC está conectada a una señal, se puede configurar el inicio con un flanco de subida o de bajada con del ADCR.

- Podemos configurar que se produzca una interrupción cuando se termine de convertir un canal o cualquiera de ellos con el registro

- El SPAN es el rango de conversión del ADC. El rango de señal que el ADC convierte en 12 bits viene dado por los pines dedicados de la placa (no GPIO)

VREFN 🡪 0x000 VREFP 🡪 0xFFF

- Al finalizar la conversión, el resultado e información de la conversión se guarda en 2 sitios:

· : Contiene el resultado, canal convertido y OVERRUN del último canal convertido.

· Hay 1 por canal y contiene el resultado y el OVERRUN del canal

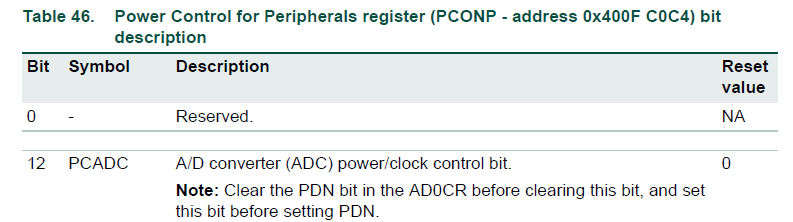
- Para saber cuál fue el último canal convertido y los OVERRUN de los canales usamos el registro que suele ser usado en la función de atención a la interrupción para este propósito

- Para limpiar al interrupción, limpiamos el bit DONE del canal correspondiente o leemos el registro o

**2. Procedimiento de configuración:**

**1) Encendido y activación del ADC:**

Primero de todo debemos alimentar el ADC mediante el bit correspondiente del PCONP



Si el bit de control de un periférico está a 1, este está activado y si está a 0, está desactivado

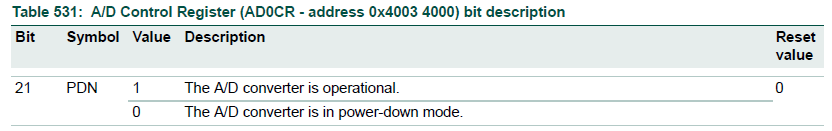
**LPC\_SC->PCONP |= ( 1 << 12);**

Una vez alimentado el ADC, este puede estar funcionando en:

- Modo normal 🡪 Realiza la conversión

- Modo bajo consumo 🡪 No realiza la conversión.

Para el control del mismo usamos el Bit PDN del registro AD0CR:

****

**LPC\_ADC->ADCR |= (1 << 21);**

Cuando hay un Reset, el ADC está desactivado por lo que tendremos que:

- Encenderlo

- Ponerlo en modo operativo.

Para apagarlo tenemos que primero ponerlo en modo bajo consumo.

**LPC\_ADC->ADCR |= (1 << 21);**

**LPC\_SC->PCONP |= ( 1 << 12);**

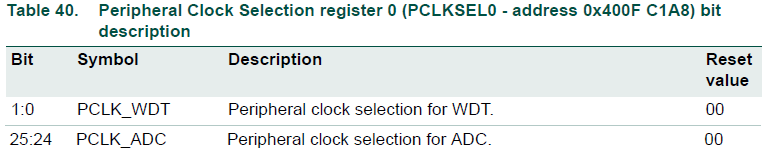
Normalmente al configurar el ADC, primero lo alimentamos con PCON, después configuramos sus registros y por último lo activamos con PDN.

**2) Establecimiento de la frecuencia del Reloj del ADC:**

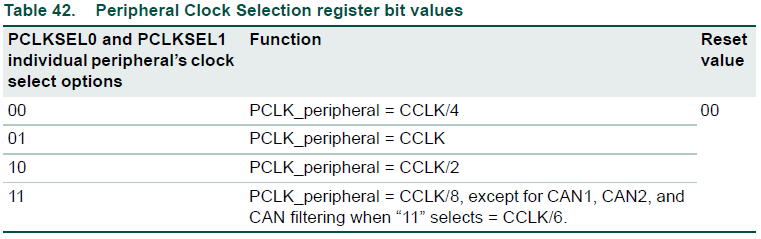
La frecuencia de entrada al converso debe ser menor que 13 MHz:

Esta frecuencia afecta al tiempo de conversión y a la máxima frecuencia de funcionamiento:

- Los registros que controlan la frecuencia del PCLK del ADC son:



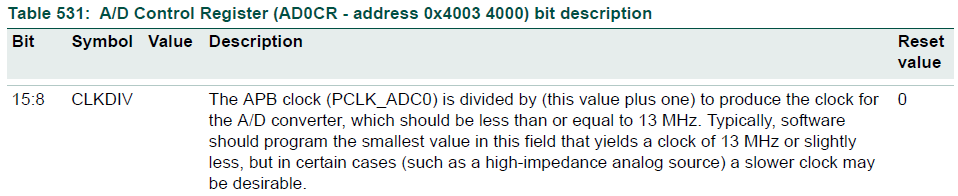
- El significado de los bits [25:24] es:



- Para coger la frecuencia de 25 MHz 🡪 CCLK / 4

**LPC\_SC->PCLKSEL0 &= ~ ( (1 << 24) |(1 << 25) ) ;**

- El ADC posee un Prescaler con el cual disminuir la frecuencia llamado CLKDIV:



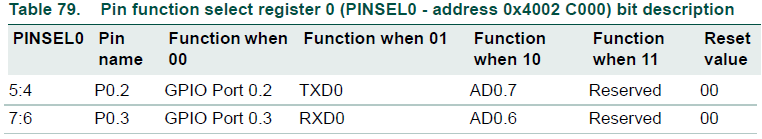
Para dividir la frecuencia por 5 y conseguir 5 MHz escribiríamos:

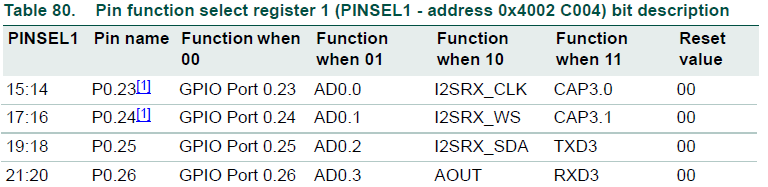
**LPC\_ADC->ADCR = ( 4 << 8 );**

**3) Selección de los pines de entrada al ADC:**

Tenemos hasta 8 entradas (canales) del ADC posibles que son multiplexadas, ADC.0 – 7.

Algunas de ellas se pueden activar en los registros:



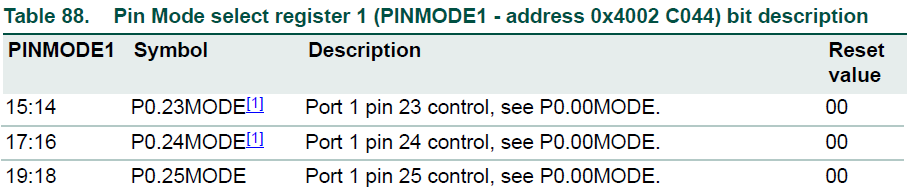


- Si por ejemplo quisiésemos activar las entradas AD0.0 (en P.023) haríamos:

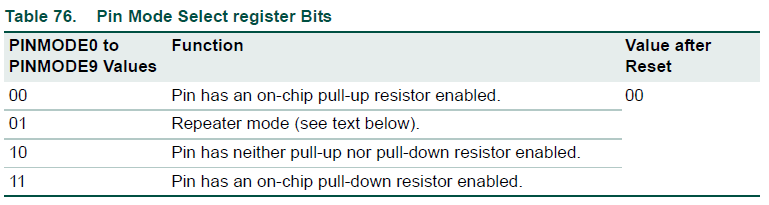
**LPC\_PINCON->PINSEL1 &= ~ ( 3 << 14);**

**LPC\_PINCON->PINSEL1 |= ( 1 << 14);**

- Además el fabricante nos recomienda que la entrada no tenga ni pull up ni pull down por lo que modificamos los registros PINMODE:



- El significado de los 2 bits de cada GPIO son:



- En nuestro caso, queremos desactivar el pull up/down por lo que escribimos un 10

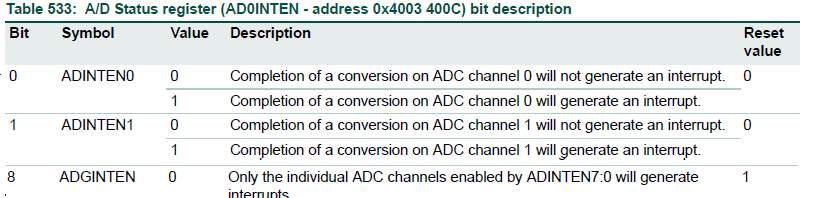
**LPC\_PINCON->PINMODE1 &= ~( 3 << 14);**

**LPC\_PINCON->PINMODE1 |= ( 2 << 14);**

**5) Activamos las interrupciones de fin de conversión del ADC**

Cada vez que se produce el fin de una conversión en cualquiera de los 8 canales, se puede generar la interrupción pertinente.

Mediante el registro ADINTEN podemos seleccionar que canales producirán una interrupción cuando sean muestreados.

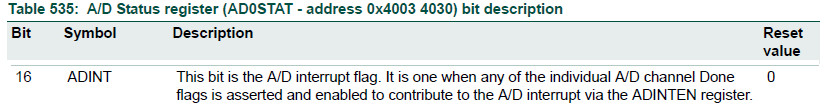


- Si queremos que haya una interrupción cuando haya una conversión del canal 0 hacemos:

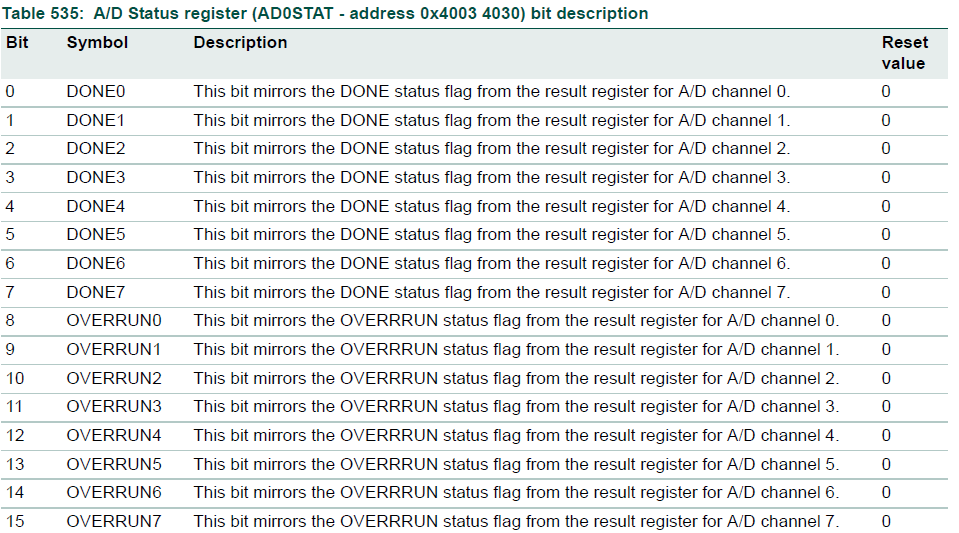
**LPC\_ADC->ADINTEN |= ( 1 << 0);**

- El flag de interrupción del AFDC es el bit ADINT registro ADC0STAT.

Cuando se produzca una interrupción tendremos que limpiar este flag aunque como veremos, leer el registro del dato convertido lo limpia solo.



- Para saber cual de los 8 canales ocasionó la interrupción y si machacaron o no un dato leído anteriormente usamos el mismo registro ADC0STAT.



- Para dar prioridad y activar la función de atención a la interrupción del ADC hacemos:

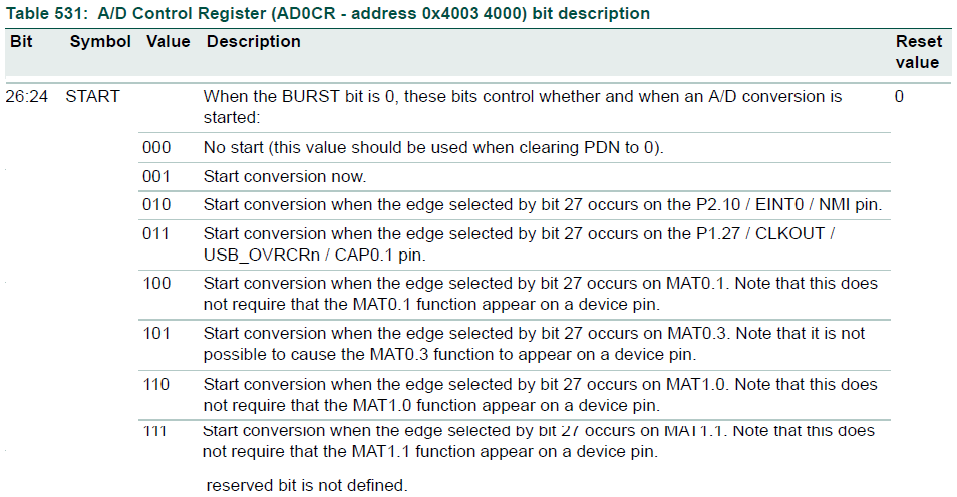
**NVIC\_SetPriority(ADC\_IRQn,0x01);**

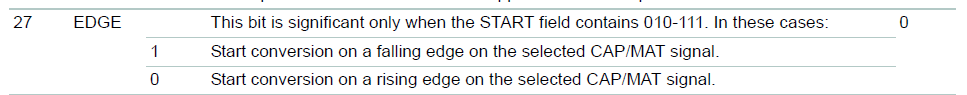
**NVIC\_EnableIRQ(ADC\_IRQn);**

**6) Configuración de la señal de inicio de conversión del ADC:**

Podemos indicar el inicio de conversión al ADC vía software o dándole una señal interna o externa que se lo indique en flanco se subida o bajada.

- Para seleccionar la señal de inicio de conversión usamos el registro START:



- Si queremos que se active con una señal externa como el P0.10 o una señal interna como MAT.0.1, podemos indicar si el inicio de la conversión se realiza cuando se produzca el flanco de subida o el de bajada

En este modo seleccionaremos un único canal y una señal de inicio de conversión.

- Ejemplo inicio de conversión con el MAT0.1 en flanco de subida

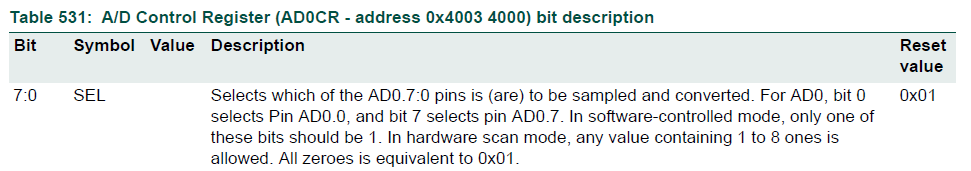
**LPC\_ADC->ADCR |= ( 4 << 24 );**

**LPC\_ADC->ADCR &= ~ ( 1 << 27 );**

**🡪 Selección del canal de entrada:**

Dado que tenemos 8 posibles canales de entrada conectados a un multiplexor y un solo conversor, tendremos que indicar mediante un registro, cual de ellos va ser convertido.

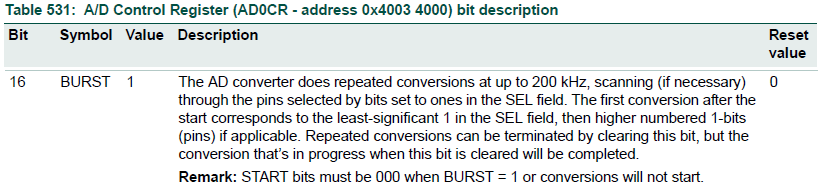
Para ello usamos el registro SEL:



El valor que debemos darles a estos 8 bits es diferente según el modo de funcionamiento

**🡪 Modos de funcionamiento:**

El ADC posee 2 modos de funcionamiento que son elegidos mediante el bit BURST



Modo normal 🡪 BURST = '0'

Se realiza la conversión del canal seleccionado por el registro SEL

Para la selección del canal ponemos un '1' en el correspondiente bit de SEL.

🡪 Sólo puede haber uno de los canales seleccionados.

- Ejemplo de muestreo del canal 2 en modo normal:

**LPC\_ADC->ADCR &= ~ ( 1 << 16 ); // Modo normal**

**LPC\_ADC->ADCR = ( 1 << 2 ); //Selección del canal 2**

Modo ráfaga 🡪 BURST = '1'

En este modo le indicamos al ADC los canales que queremos muestrear escribiendo un '1' en los SEL correspondientes y a cada conversión va conmutando de uno a otro.

🡪 Debe haber tantos canales seleccionados como queremos muestrear.

- Ejemplo de muestreo de los canales 1, 3.

**LPC\_ADC->ADCR |= ( 1 << 16 ); // Modo Burst**

**LPC\_ADC->ADCR |= ( 1 << 1 )| ( 1 << 3); //Selección de los canales 1,3**

El campo START del ADCR debe estar a 0, este modo no necesita señal de inicio de conversión, el ADC empezará a muestrear continuamente los canales.

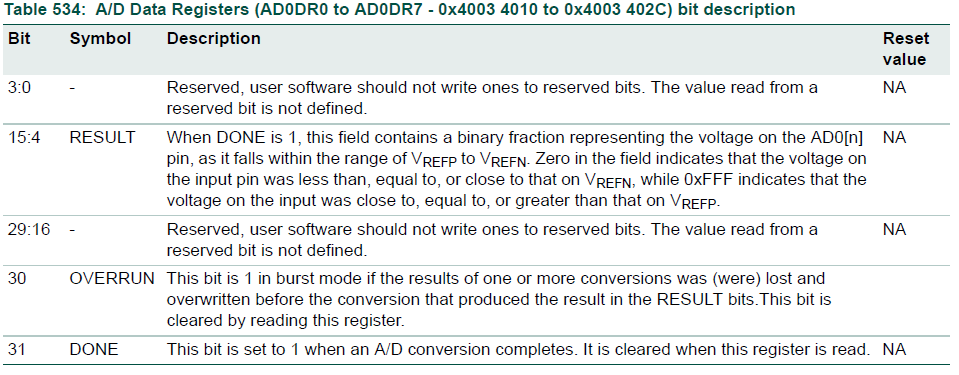
**7) Lectura de los datos convertidos:**

Para la lectura de los datos convertidos podemos usar diferentes registros.

No es aconsejable usar ambos, hay que usar sólo uno todo el tiempo.

**🡪 A/D Data Registers (AD0DR0 to AD0DR7)**

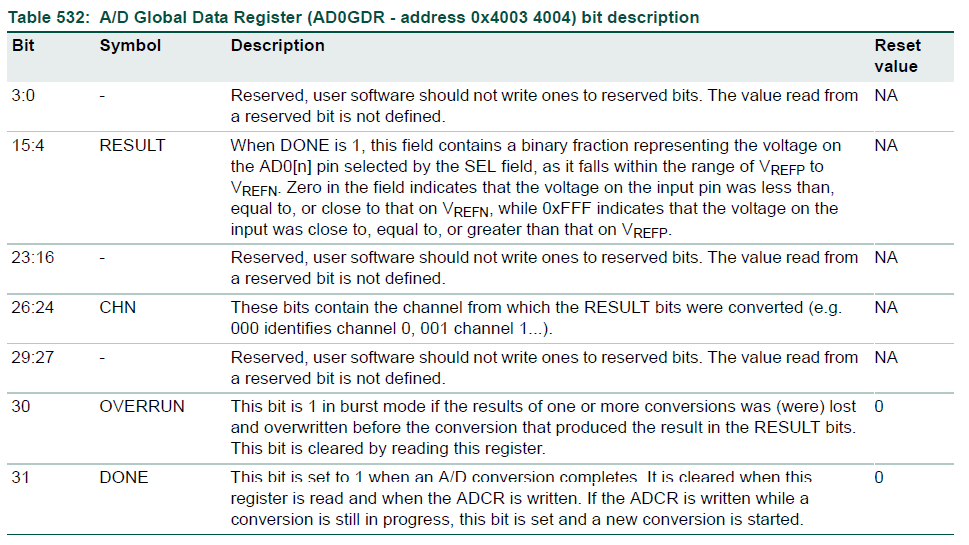
Existe un registro de estos por cada canal y nos dice de cada uno su RESULTADO, DONE y OVERRUN



- El bit de atención a interrupción DONE es limpiado cuando leemos este registro.

**🡪 A/D Global Data Register:**

Tiene el resultado de la última conversión del ADC por lo que podemos usarlo para obtener el dato convertido de cualquier canal:



- RESULT (ADGDR [15:4]): Contiene los 12 bits de la última conversión.

- OVERRUN (ADGR[30]): Si está a '1' significa que esta conversión ha machado a la última antes de que fuese leída, perdiendo ese dato.

- El CHM (ADGDR [26:24]) contiene cual de los 8 canales corresponde al resultado de RESULT

- DONE (ADGR[31]): Si está a '1' nos indica que se ha realizado la conversión.

Cuando se produce una interrupción tenemos que limpiar el bit DONE, pero también es automáticamente limpiado cuando: 1) Leemos el ADGDR

2) Modificamos el ADCR