

# Documentacion de funciones de librerias de CMSIS utilizadas para EDIII

## ADC: utilizaremos las funciones incluidas en la siguiente libreria:

```
#include "lpc17xx_adc.h"
```

Para configurar el ADC utilizaremos las siguientes funciones:

```
// Configurar el clock del ADC; rate_conversion debe ser <=200KHz
void ADC_Init (LPC_ADC, uint32_t rate_conversion)

// Habilita o deshabilita cualquiera de los 8 canales que posee la LPC,
// por lo que la variable channel puede obtener valores de 0 a 7,
// mientras que state puede ser ENABLE o DISABLE
void ADC_ChannelCmd (LPC_ADC, uint8_t channel, state)

// Habilita o deshabilita las interrupciones en cualquiera de los 8 canales que posee la LPC,
// por lo que la variable channel puede obtener valores de 0 a 7,
// mientras que state puede ser ENABLE o DISABLE
void ADC_IntConfig (LPC_ADC, uint8_t channel, state)

// Nos devuelve SET o RESET dependiendo de si el ADC termino de convertir o no.
// Recordamos que el ADC posee 2 tipos de estados, Burst (0) o Done (1).
// El tipo de estado Done es el que usaremos. En los parametros de la funcion,
// channel corresponde al canal [0-7] mientras que status al tipo de estado por
// el que queremos consultar. Veamos un ejemplo
FlagStatus ADC_ChannelGetStatus (LPC_ADC, uint8_t channel, uint32_t status)

// Aqui estaríamos realizando algo siempre y cuando el canal [0]
// NO haya terminado de convertir (por eso se iguala a RESET).
// Cuando termine de convertir, devolvera un SET y se procedera a realizar otra cosa
while (ADC_ChannelGetStatus (LPC_ADC, 0, 1)==RESET)

// Nos devuelve el resultado de la conversion del registro ADDR0.
// En el parametro channel ponemos el canal del que queremos leer los datos [7-0]
uint32_t ADC_GetData (uint32_t channel)

// Seteamos el modo para operar con el ADC.
// Podemos utilizar esta funcion por ejemplo luego de terminar de convertir,
// para que el ADC comience otra vez a tomar datos. En start_mode podemos elegir entre:
// ADC_START_CONTINUOUS, ADC_START_NOW, ADC_START_ON_EINT0, etc. Generalmente usamos ADC_START.
void ADC_StartCmd (LPC_ADC, uint8_t start_mode)

// Close ADC
void ADC_DeInit (LPC_ADC)
```