

Diseño Basado en Microprocesadores

Tema 2. Microcontroladores

- 2.1. Introducción a los microcontroladores
- 2.2. Entradas/Salidas Digitales
- 2.3. Temporizadores
- 2.4. Excepciones
- 2.5. Conversión Analógica/Digital
- 2.6. Comunicación serie RS232C
- 2.7. Teclado, conversión D/A y sonido
- 2.8. Interfaz I2C

2.4. Adquisición analógica

UM10562 Cap. 32: Convertidor A/D

2.5.1. Convertidores analógicos/digitales

2.5.2. Conversión A/D. Características.

2.5.3. Conversión A/D. Configuración

2.5.4. Pines del convertidor A/D

2.5.5. Esquema del convertidor A/D

2.5.6. Registros

2.5.7. Accesos a los registros

2.5.8. Pasos para la programación del convertidor A/D

2.5.9. Biblioteca adc_lpc40xx

2.5.10. Conversión de unidades

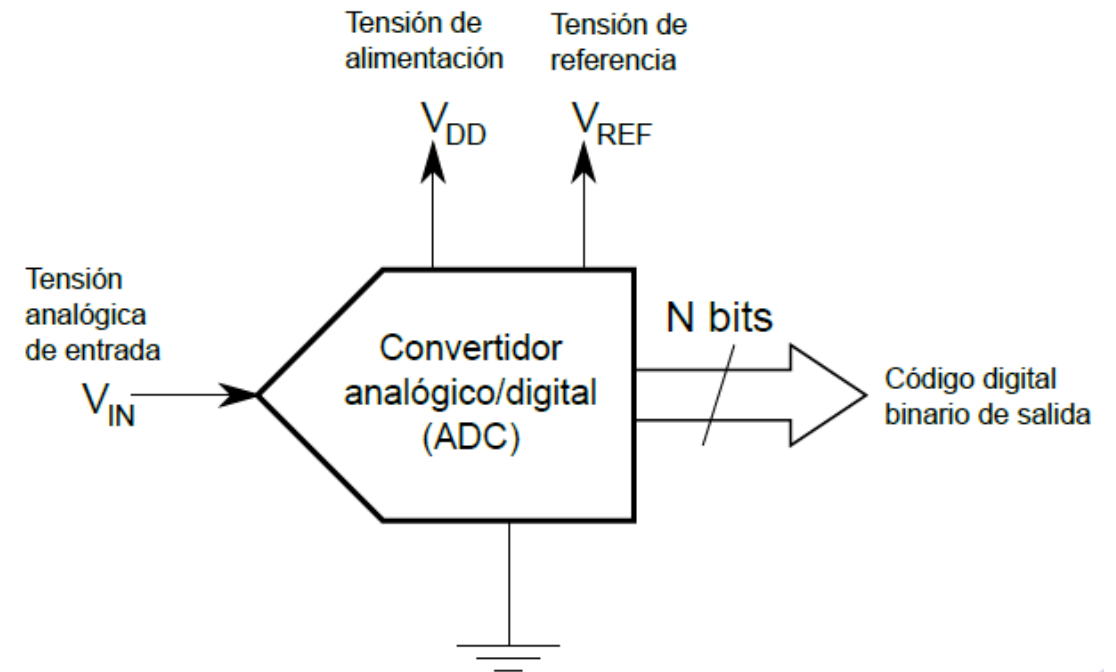
2.5.11. Ejemplo 1

2.5.12. Ejemplo 2

2.5.13. Ejemplo 3

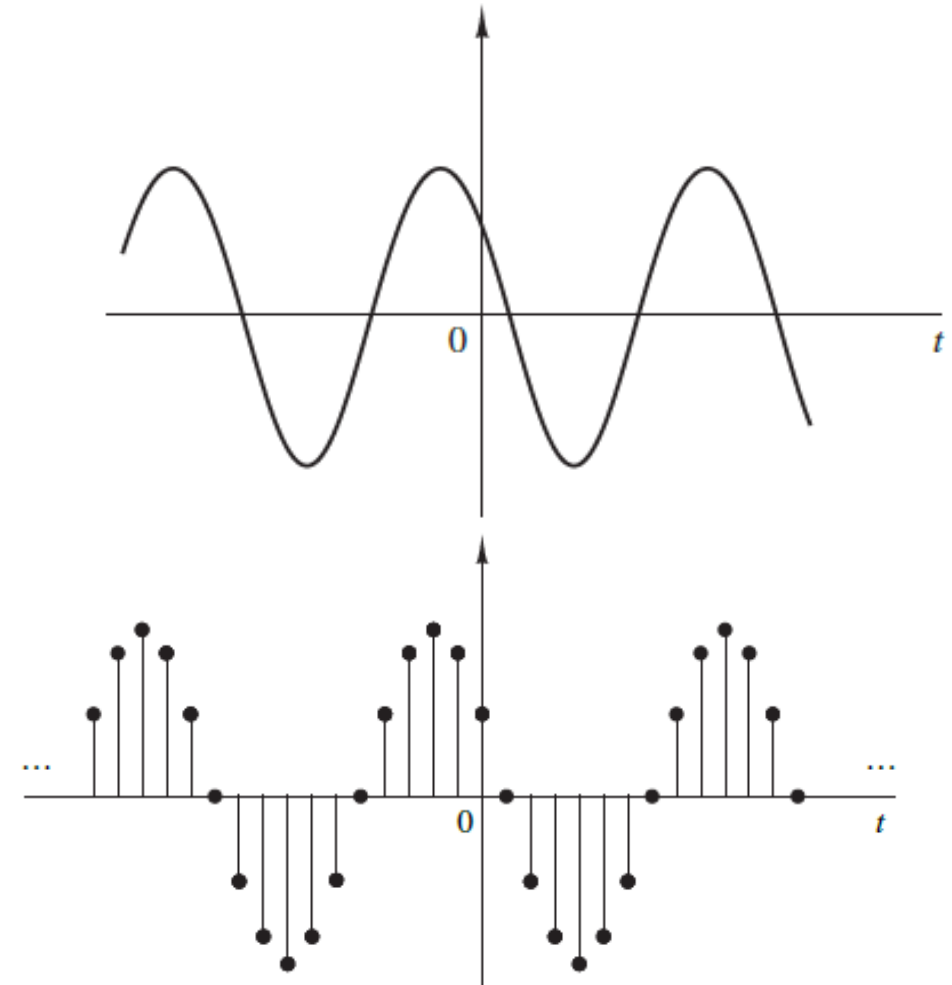
2.5.1. Convertidor analógico/digital (ADC) (1)

- Un convertidor analógico/digital, o **ADC**, permite trasladar una señal de entrada analógica con valores dentro de cierto rango continuo a conjunto discreto de códigos digitales binarios de salida
- Esto permite obtener una representación de la señal de entrada apta para ser procesada por un **sistema digital binario**
- El proceso introduce irremediablemente un error de “**cuantificación**”



2.5.1. Convertidor analógico/digital (ADC) (2)

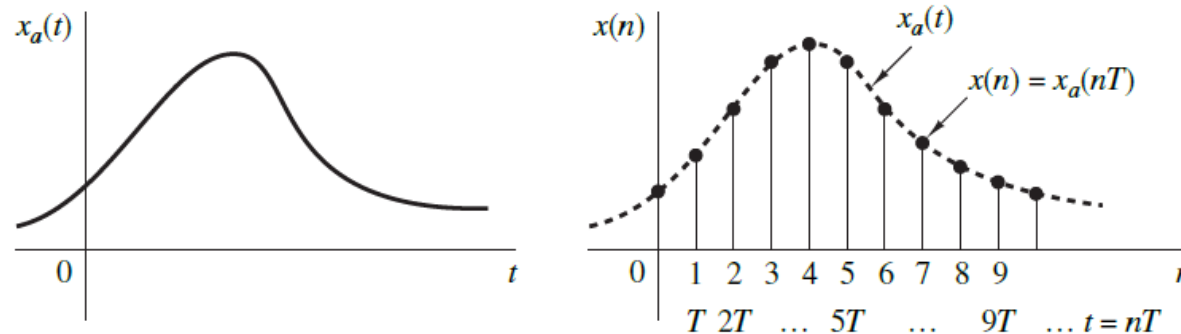
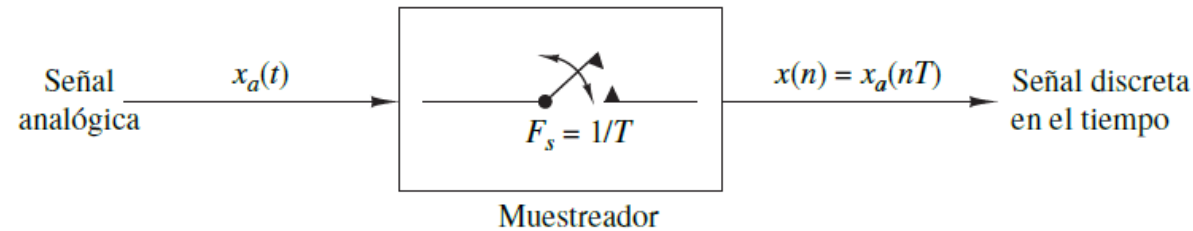
Una **señal en tiempo continuo** es aquella cuyo valor está definido para cualquier instante de tiempo



Una **señal en tiempo discreto** es aquella cuyo valor está definido sólo en determinados instantes de tiempo

2.5.1. Convertidor analógico/digital (ADC) (3)

- Los ADCs **discretizan el valor** de las señales.
- Además, para ser procesadas en un sistema digital, las señales también deben ser **discretizadas en el tiempo** mediante un proceso de muestreo.
- El **proceso de muestreo** consiste transformar una señal en tiempo continuo en una señal en tiempo discreto, preservando el valor de la señal original sólo en determinados instantes de tiempo de muestreo $t_n = nT$
- T es el periodo de muestreo. Su inverso, $F_s = 1/T$, es la frecuencia de muestreo.



2.5.1. Convertidor analógico/digital (ADC) (4)

Teorema de muestreo (de Nyquist-Shannon): si la frecuencia más alta contenida en una señal analógica $x_a(t)$ es $F_{\max} = B$ y la señal se muestrea a una frecuencia $F_s > 2B$, entonces $x_a(t)$ puede recuperarse de forma exacta a partir de los valores de sus muestras mediante:

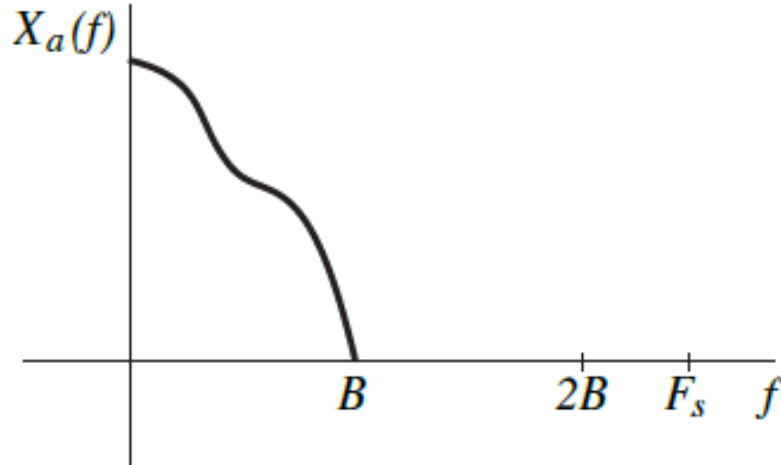
$$x_a(t) = \sum_{n=-\infty}^{\infty} \left(x_a \left(\frac{n}{F_s} \right) \frac{\sin(2\pi Bt)}{2\pi Bt} \right)$$

donde $x_a(n/F_s) = x_a(nT) = x(n)$ son las muestras de $x_a(t)$

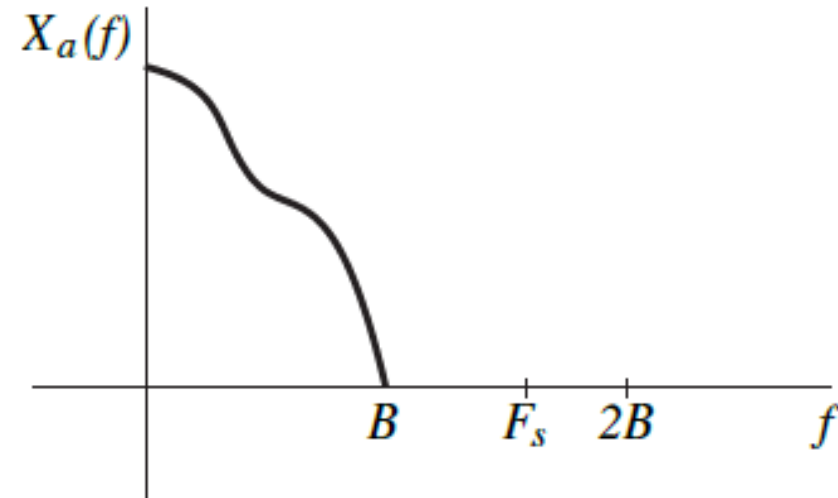
Si no se cumple $F_s > 2B$, se produce un fenómeno de distorsión llamado aliasing que impide la representación fiel de la señal original mediante sus muestras

2.5.1. Convertidor analógico/digital (ADC) (5)

Espectro de una señal con componentes de frecuencia hasta $f = B$:

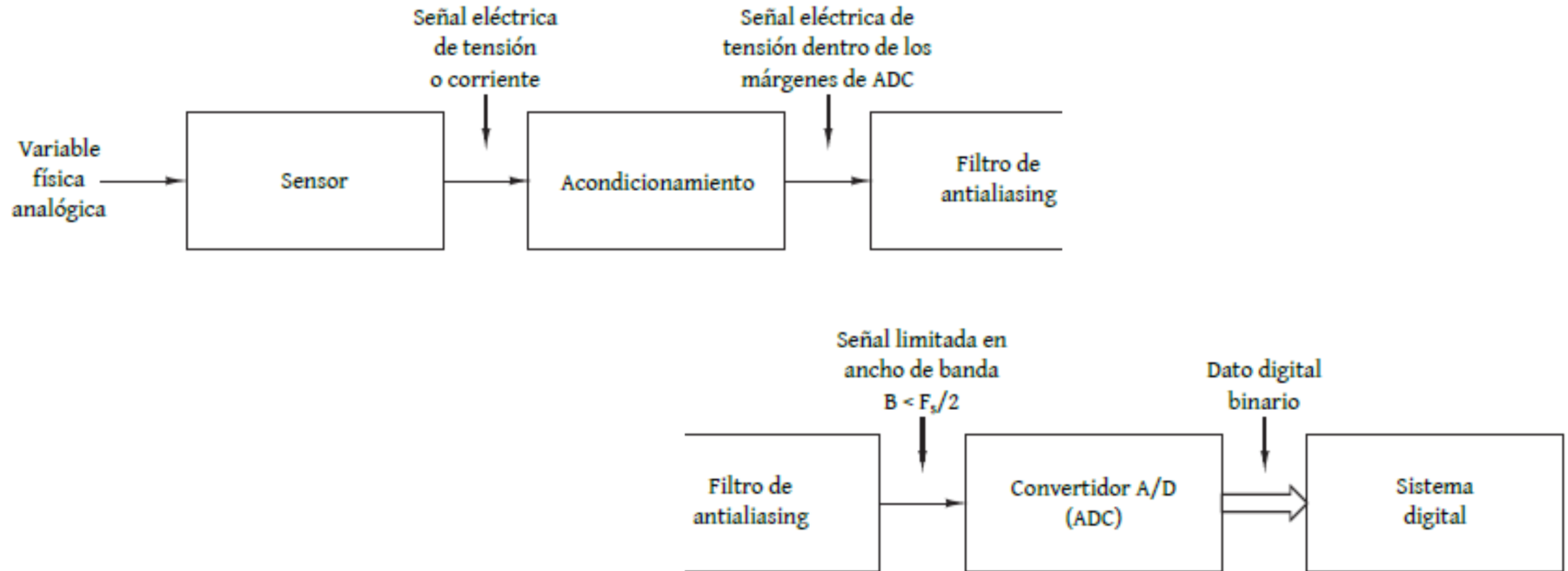


Si $F_s > 2B$ entonces la señal original puede representarse y reconstruirse a partir de sus muestras.



Si $F_s < 2B$ el muestreo produce distorsión de *aliasing* y la señal original no puede representarse ni reconstruirse correctamente a partir de sus muestras.

2.5.1. Convertidor analógico/digital (ADC) (6)



2.5.2. Conversión A/D. Características

- 8 canales
- Convertidor de aproximaciones sucesivas de 12 bits.
- Entrada multiplexada
- Modo de bajo consumo (powerdown)
- Rango de medida de 0 a 3V (VSS a VREFP) (no exceder VDDA)
- Tiempo de conversión de 12 bits $\geq 2,5\mu\text{s}$ (31 clocks)
- Modo conversión Burst (ráfagas) para simple o múltiple entrada
- Conversión opcional según transición en pin externo o señal de match del timer

2.5.3. Conversión A/D. Configuración

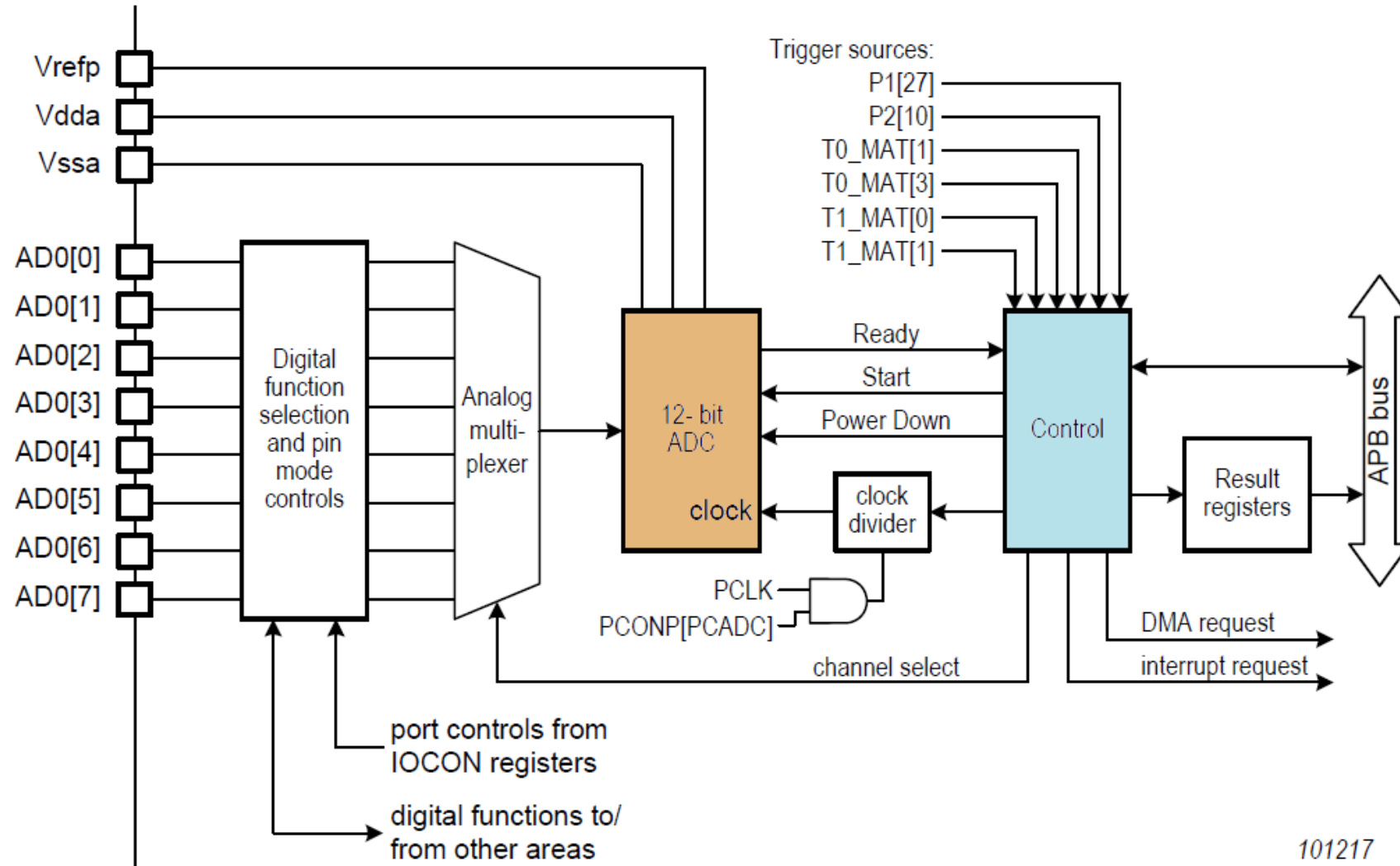
- En el reset el ADC es deshabilitado. Para habilitarlo 1º poner el bit 12 del registro PCONP del SC a uno y 2º habilitar el bit PDN del registro AD0CR. Para deshabilitar a la inversa
- Reloj: se usa el PCLK que es $CCLK/2$. Para escalar el reloj para el ADC, usar los bits CLKDIV
- Seleccionar la función del pin en el registro IOCON
- Se puede habilitar interrupciones en el ADC

2.5.4. Pines del convertidor A/D

Table 676. ADC pin description

Pin	Type	Description
AD0[7] to AD0[0]	Input	<p>Analog Inputs. The ADC cell can measure the voltage on any of these input signals. Digital signals are disconnected from the ADC input pins when the ADC function is selected on that pin in the Pin Select register.</p> <p>Warning: if the ADC is used, signal levels on analog input pins must not be above the level of V_{DDA} at any time. Otherwise, A/D converter readings will be invalid. If the A/D converter is not used in an application then the pins associated with A/D inputs can be used as 5 V tolerant digital IO pins.</p>
V_{REFP}	Reference	<p>Voltage Reference. This pin provides a voltage reference level for the ADC and DAC. Note: V_{REFP} should be tied to VDD(3V3) if the ADC and DAC are not used.</p>
V_{DDA} , V_{SSA}	Power	<p>Analog Power and Ground. These should typically be the same voltages as V_{DD} and V_{SS}, but should be isolated to minimize noise and error. Note: V_{DDA} should be tied to VDD(3V3) and V_{SSA} should be tied to VSS if the ADC and DAC are not used.</p>

2.5.5. Esquema del convertidor



101217

2.5.6. Registros

Nombre	Descripción	Tipo	Reset	Offset
CR	Registro de control AD. El registro CR debe ser escrito para seleccionar el modo de operación antes de que una conversión AD ocurra.	R/W	0x01	0x000
GDR	Registro de dato global AD. Contiene el resultado de la última conversión AD.	R/W	NA	0x004
INTEN	Registro de habilitación de interrupción AD. Contiene los bits de habilitación que permite que la bandera DONE de cada canal AD sea incluida o excluida para contribuir a la generación de una interrupción AD.	R/W	0x0100	0x00C
DRn	Registro de dato del canal ADn (0:7). Este registro contiene el resultado de la última conversión realizada en el canal n.	RO	NA	0x010 a 0x02C
STAT	Registro de estado AD. Contiene las banderas de DONE y OVERRUN para todos los canales. También contiene la bandera de interrupción.	RO	0x0	0x030
TRM	Registro trim ADC	R/W	0x0	0x034

Registro CR

Bits	Símbolo	Valor	Descripción	Reset
7:0	SEL		Selecciona el pin o canal que va a ser muestreado y convertido. En modo controlado por software solo debe haber un bit a 1.	0x01
15:8	CLKDIV		El PCLK es dividido por este valor para obtener la frecuencia del convertidor AD	0
16	BURST	0	Conversión controlada por software (31 pulsos)	0
		1	Conversión por ráfaga (múltiple) START debe ser 0	
20:17			Reservado, no se deben usar.	NA
21	PDN	1	Convertidor operativo	0
		0	Convertidor en modo bajo consumo	
23:22			Reservado, no se deben usar.	NA
26:24	START	0 a 7	Cuando BURST es 0 indica el modo de arrancar el convertidor: 0 no arranca, 1 arranca software, 2..7 hard	0
27	EDGE	1	Comienza conversión con flanco de bajada	0
		0	Comienza conversión con flanco de subida (Solo para START de 2 a 7) señales CAP/MAT	

Registro GDR

Table 679: A/D Global Data Register (GDR - address 0x4003 4004) bit description

Bit	Symbol	Description	Reset value
3:0	-	Reserved. Read value is undefined, only zero should be written.	NA
15:4	RESULT	When DONE is 1, this field contains a binary fraction representing the voltage on the AD0[n] pin selected by the SEL field, as it falls within the range of V_{REFP} to V_{SS} . Zero in the field indicates that the voltage on the input pin was less than, equal to, or close to that on V_{SS} , while 0xFFF indicates that the voltage on the input was close to, equal to, or greater than that on V_{REFP} .	NA
23:16	-	Reserved. Read value is undefined, only zero should be written.	NA
26:24	CHN	These bits contain the channel from which the RESULT bits were converted (e.g. 000 identifies channel 0, 001 channel 1...).	NA
29:27	-	Reserved. Read value is undefined, only zero should be written.	NA
30	OVERRUN	This bit is 1 in burst mode if the results of one or more conversions was (were) lost and overwritten before the conversion that produced the result in the RESULT bits. This bit is cleared by reading this register.	0
31	DONE	This bit is set to 1 when an A/D conversion completes. It is cleared when this register is read and when the ADCR is written. If the ADCR is written while a conversion is still in progress, this bit is set and a new conversion is started.	0

Registro INTEN

Table 680: A/D Interrupt Enable register (INTEN - address 0x4003 400C) bit description

Bit	Symbol	Value	Description
0	ADINTEN0		Interrupt enable
		0	Completion of a conversion on ADC channel 0 will not generate an interrupt.
		1	Completion of a conversion on ADC channel 0 will generate an interrupt.
1	ADINTEN1		Interrupt enable
		0	Completion of a conversion on ADC channel 1 will not generate an interrupt.
		1	Completion of a conversion on ADC channel 1 will generate an interrupt.
2	ADINTEN2		Interrupt enable
		0	Completion of a conversion on ADC channel 2 will not generate an interrupt.
		1	Completion of a conversion on ADC channel 2 will generate an interrupt.
3	ADINTEN3		Interrupt enable
		0	Completion of a conversion on ADC channel 3 will not generate an interrupt.
		1	Completion of a conversion on ADC channel 3 will generate an interrupt.
4	ADINTEN4		Interrupt enable
		0	Completion of a conversion on ADC channel 4 will not generate an interrupt.
		1	Completion of a conversion on ADC channel 4 will generate an interrupt.
5	ADINTEN5		Interrupt enable
		0	Completion of a conversion on ADC channel 5 will not generate an interrupt.
		1	Completion of a conversion on ADC channel 5 will generate an interrupt.
6	ADINTEN6		Interrupt enable
		0	Completion of a conversion on ADC channel 6 will not generate an interrupt.
		1	Completion of a conversion on ADC channel 6 will generate an interrupt.
7	ADINTEN7		Interrupt enable
		0	Completion of a conversion on ADC channel 7 will not generate an interrupt.
		1	Completion of a conversion on ADC channel 7 will generate an interrupt.
8	ADGINTEN		Interrupt enable
		0	Only the individual ADC channels enabled by ADINTEN7:0 will generate interrupts.
		1	The global DONE flag in ADDR is enabled to generate an interrupt in addition to individual ADC channels that are enabled to generate interrupts.
31:9	-		Reserved. Read value is undefined, only zero should be written.

Registro DRn

Bits	Símbolo	Descripción	Reset
3:0		Reservados.	NA
15:4	RESULT	Cuando DONE es 1, contiene la fracción binaria que representa el voltaje en el pin de entrada VSS a VREFP Fuera de limite: (0/0xFFF)	NA
29:16		Reservados.	NA
30	OVERRUN	Cuando BURST es 1, indica si ha habido sobreescritura de conversiones.	NA
31	DONE	Se pone a 1 cuando una conversión se ha completado	NA

Registro STAT

Table 682: A/D Status register (STAT - address 0x4003 4030) bit description

Bit	Symbol	Description	Reset value
0	DONE0	This bit mirrors the DONE status flag from the result register for A/D channel 0.	0
1	DONE1	This bit mirrors the DONE status flag from the result register for A/D channel 1.	0
2	DONE2	This bit mirrors the DONE status flag from the result register for A/D channel 2.	0
3	DONE3	This bit mirrors the DONE status flag from the result register for A/D channel 3.	0
4	DONE4	This bit mirrors the DONE status flag from the result register for A/D channel 4.	0
5	DONE5	This bit mirrors the DONE status flag from the result register for A/D channel 5.	0
6	DONE6	This bit mirrors the DONE status flag from the result register for A/D channel 6.	0
7	DONE7	This bit mirrors the DONE status flag from the result register for A/D channel 7.	0
8	OVERRUN0	This bit mirrors the OVERRRUN status flag from the result register for A/D channel 0.	0
9	OVERRUN1	This bit mirrors the OVERRRUN status flag from the result register for A/D channel 1.	0
10	OVERRUN2	This bit mirrors the OVERRRUN status flag from the result register for A/D channel 2.	0
11	OVERRUN3	This bit mirrors the OVERRRUN status flag from the result register for A/D channel 3.	0
12	OVERRUN4	This bit mirrors the OVERRRUN status flag from the result register for A/D channel 4.	0
13	OVERRUN5	This bit mirrors the OVERRRUN status flag from the result register for A/D channel 5.	0
14	OVERRUN6	This bit mirrors the OVERRRUN status flag from the result register for A/D channel 6.	0
15	OVERRUN7	This bit mirrors the OVERRRUN status flag from the result register for A/D channel 7.	0
16	ADINT	This bit is the A/D interrupt flag. It is one when any of the individual A/D channel Done flags is asserted and enabled to contribute to the A/D interrupt via the ADINTEN register.	0
31:17	-	Reserved. Read value is undefined, only zero should be written.	NA

Registro TRM

Table 683: A/D Trim register (TRM - address 0x4003 4034) bit description

Bit	Symbol	Description	Reset value
3:0	-	Reserved. Read value is undefined, only zero should be written.	NA
7:4	ADCOFFS	Offset trim bits for ADC operation. Initialized by the boot code. Can be overwritten by the user.	0
11:8	TRIM	written-to by boot code. Can not be overwritten by the user. These bits are locked after boot code write.	0
31:12	-	Reserved. Read value is undefined, only zero should be written.	NA

2.5.7. Acceso a los registros (1)

```
/*----- Analog-to-Digital Converter (ADC) -----*/  
typedef struct  
{  
    __IO uint32_t CR;      /*!< Offset: 0x000    A/D Control Register (R/W)*/  
    __IO uint32_t GDR;     /*!< Offset: 0x004    A/D Global Data Register (R/W)*/  
    uint32_t RESERVED0;  
    __IO uint32_t INTEN; /*!< Offset: 0x00C    A/D Interrupt Enable Register  
(R/W)*/  
    __IO uint32_t DR[8]; /*!< Offset: 0x010-0x02C A/D Channel 0..7 Data  
Register (R/W)*/  
    __IO uint32_t STAT; /*!< Offset: 0x030    A/D Status Register (R/ )*/  
    __IO uint32_t ADTRM;  
} LPC_ADC_TypeDef;
```

2.5.7. Acceso a los registros (2)

```
#define LPC_APB0_BASE    (0x40000000UL)
#define LPC_ADC_BASE    (LPC_APB0_BASE + 0x34000)
#define LPC_ADC          ((LPC_ADC_TypeDef *) LPC_ADC_BASE)
```

Lectura: dato = LPC_ADC -> DR[0]

Escritura: LPC_ADC-> CR = valor

2.5.8. Pasos para la programación del convertidor A/D

- 1) Seleccionar la función de entrada al convertidor en los pines (**IOCON**)
 - Por ejemplo, el pin P0_25 que tiene de funciones P0[25]/ADC0[2]/I2S_RX_SDA/U3_TXD
 - LPC_IOCON->P0_25 = 1;
- 2) Activar la alimentación del convertidor (*Tabla 16 – pág. 30*)
 - bit PCADC del registro **PCONP** [12] a 1
 - LPC_SC->PCONP (Activar este bit antes que el bit PDN)
- 3) Ajustar el registro de control CR
 - a) Activar el bit PDN para sacar el A/D del modo powerdown **CR** [21]
 - b) Programar el campo **CLKDIV** con el valor adecuado para conseguir un reloj del convertidor con $\text{frec ADC} < 12,4\text{MHz}$. (*PCLK 60 MHz*)
- 4) Adquisición
 - a) Seleccionar la entrada del canal n en el campo SEL de **CR** [7:0]
 - b) Disparar la conversión en el campo START de **CR** [26:24]
 - c) Esperar a que finalice la conversión comprobando el bit DONE del registro **GDR**
- 5) Convertir el valor del registro **DRn** a las unidades físicas correspondientes

2.5.9. Conversión de unidades

$$R = \frac{10 \text{ k}\Omega}{270^\circ} \text{ posición } (^\circ)$$

$$x = \frac{3,3\text{V}}{10\text{k}\Omega} R$$

$$n = \frac{2^{12}}{3,3\text{V}} x$$

Variable continua
posición

Potenciómetro

- Rango
[0,270°] - [0,10kΩ]
- $R = f(\text{ángulo})$

Temperatura,
posición,
intensidad,
voltaje,
resistencia, etc.

Sensor analógico

Tensión V

- Rango: [0,3.3]V
- $x = g(R)$

ADC

Muestra

- Rango: [0,2¹²]
- $n = h(x)$

Conversión
software

Conversión a las
unidades reales
deshaciendo las
conversiones

2.5.10. Biblioteca adc_lpc40xx.h

void adc_inicializar (uint32_t frecuencia_adc, uint32_t canales)

param_1: peripheralClock/256 < frecuencia_adc < 12.400.000 (12,4 MHz)

Param_2:	canales	0	1	2*	3	4	5	6	7
	puerto	0	0	0	0	1	1	0	0
	pin	23	24	25	26	30	31	12	13
	función	1	1	1	1	3	3	3	3

(*potenciómetro)

uint32_t adc_convertir (uint32_t canal)

param_1: canal elegido

param_2: resultado conversión valor entre 0 y 4096 (2^{12})

float adc_traducir_a_tensión (uint32_t resultado_adc)

param_1: resultado conversión valor entre 0 y 4096

param_2: valor en voltios

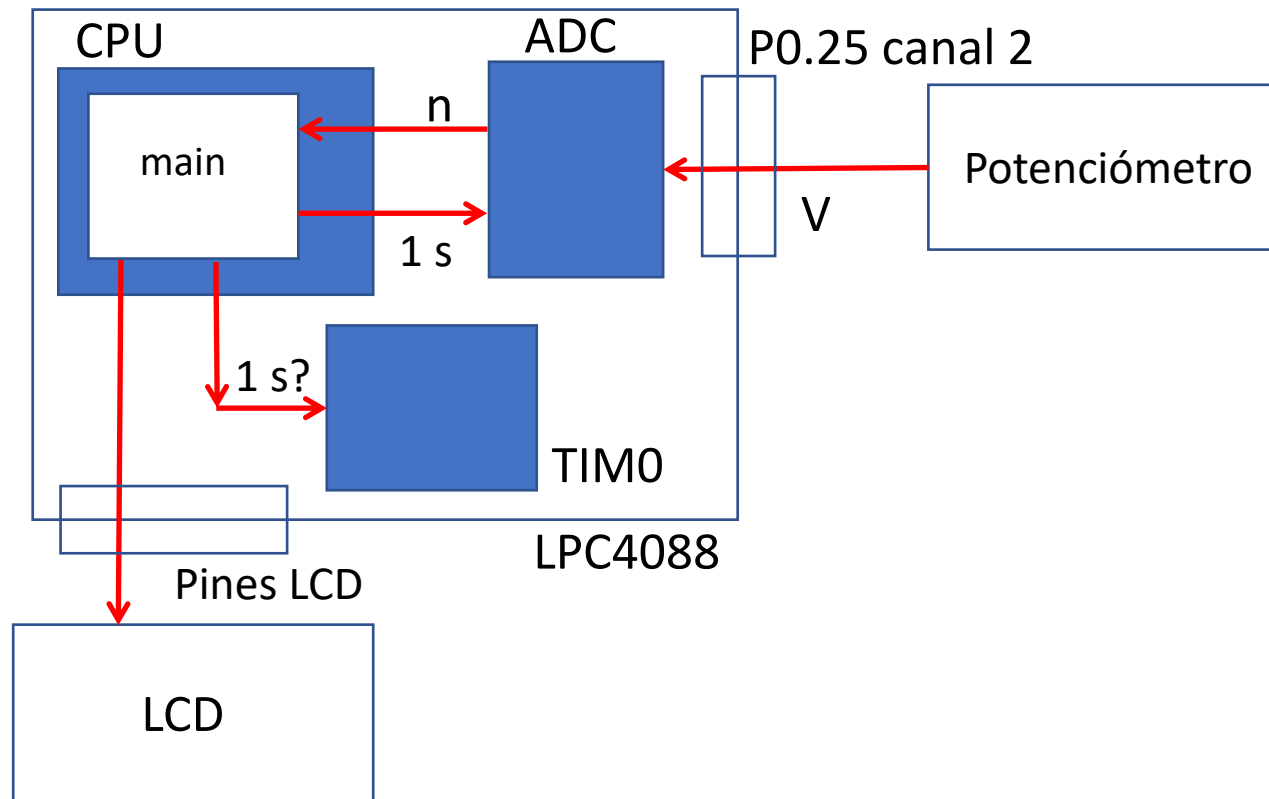
float ntc_traducir_tensión_a_temperatura (float V_s)

param_1: valor en voltios

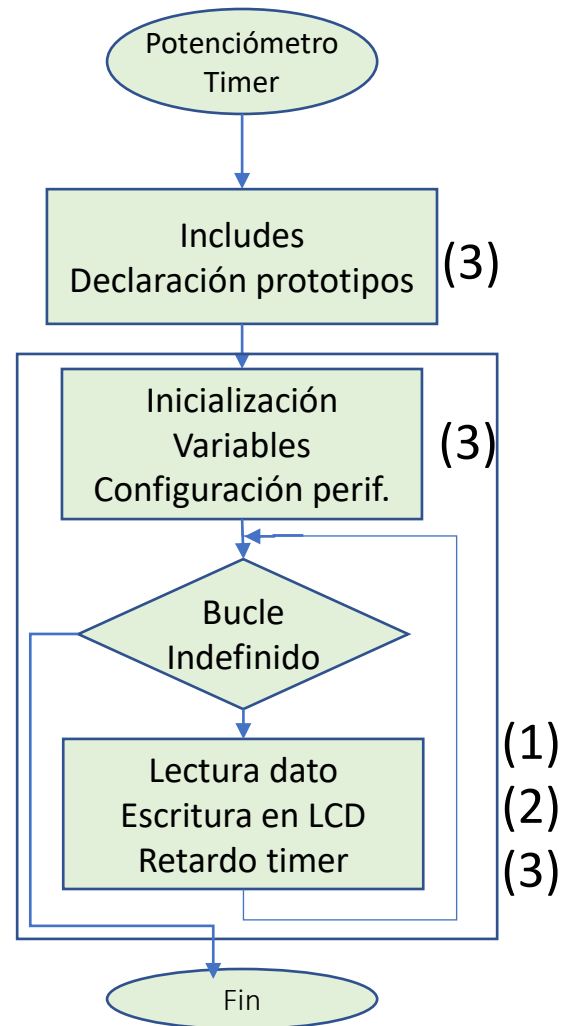
param_2: temperatura en °C del KY-013 basado en NTC

2.5.11. Ejemplo ADC con timer (1)

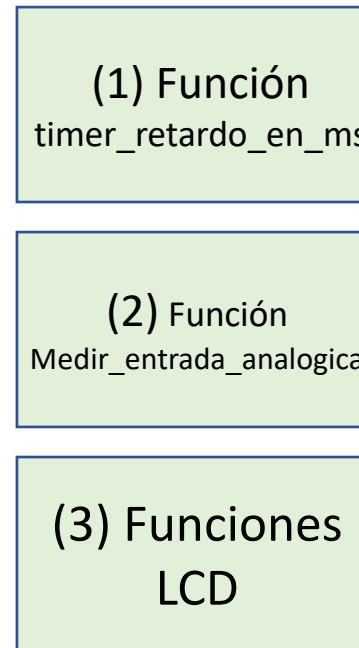
Obtener el valor digital de una conversión A/D de un potenciómetro conectado al canal 2 del LPC4088. Usar el timer 0 para generar un retardo de un segundo entre muestra y muestra. No se usan interrupciones.



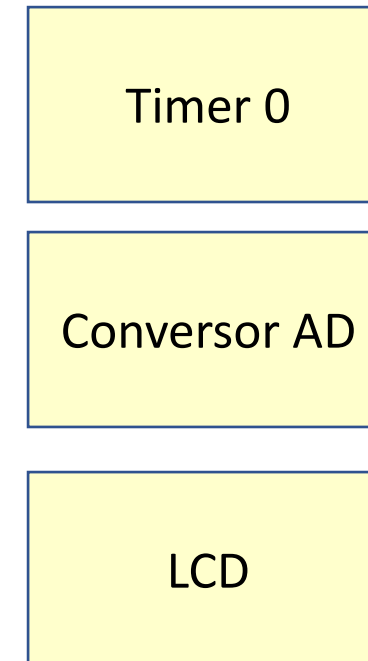
Ejemplo ADC con timer (2)



Funciones



Periféricos



Ejemplo ADC con timer (3)

```
/*=====
Fichero: entrada_analogica.c Potenciómetro conectado al canal analógico P0[25]/ADC0[2] del
LPC4088. Se usa el timer 0 para generar un retardo de un segundo entre muestra y muestra. No
se usan interrupciones.
=====*/

#include <LPC407x_8x_177x_8x.h>
#include <stdio.h>
#include "glcd.h"
#include "timer_lpc40xx.h"
uint32_t medir_entrada_analogica(void);

int main(void)
{uint32_t medida;
  glcd_inicializar ();
  glcd_borrar(NEGRO);
  timer_inicializar(TIMER0);
```

Ejemplo ADC con timer (4)

```
/* Seleccionar la función de entrada al convertidor en el pin P0[25]/ADC0[2] */
LPC_IOCON->P0_25 |= 1;
/* Activar la alimentación del convertidor.*/
LPC_SC->PCONP |= 1 << 12;
/* Ajustar el registro de control AD0CR del convertidor A/D:
    - Activar el bit PDN para sacar el A/D de modo powerdown.
    - Programar el campo CLKDIV para una frecuencia ADC de 1 MHz*/
LPC_ADC -> CR = (1<<21) | ((PeripheralClock/1000000-1)<<8);
/*Bucle principal de la aplicación.*/
while (TRUE)
{
    medida = medir_entrada_analogica();
    glcd_printf (0,0,WHITE, BLACK, FONT16X32, "Valor: %4u", medida);
    timer_retardo_ms (TIMER0, 1000);
}
}
```

Ejemplo ADC con timer (2)

```
/*=====
```

Función: medir_entrada_analogica

Utilidad: Convierte a digital la señal de salida de un potenciómetro aplicada al canal 2 del convertidor analógico/digital del LPC4088 y devuelve el valor de la conversión

Entrada: Ninguna.

Retorna: medida digital de la señal analógica

```
=====*/
```

```
#include <LPC407x_8x_177x_8x.h>
```

```
uint32_t medir_entrada_analogica(void)
```

```
{ /* Seleccionar sólo la entrada 2 en el campo SEL de CR.*/
```

```
    LPC_ADC->CR = (LPC_ADC->CR & ~0xFF) | (1<<2);
```

```
    /* Disparar una conversión. */
```

```
    LPC_ADC->CR |= 1<<24;
```

```
    /* Esperar a que finalice comprobando el bit DONE del registro GDR.*/
```

```
    while (!(LPC_ADC->GDR & (1<<31)));
```

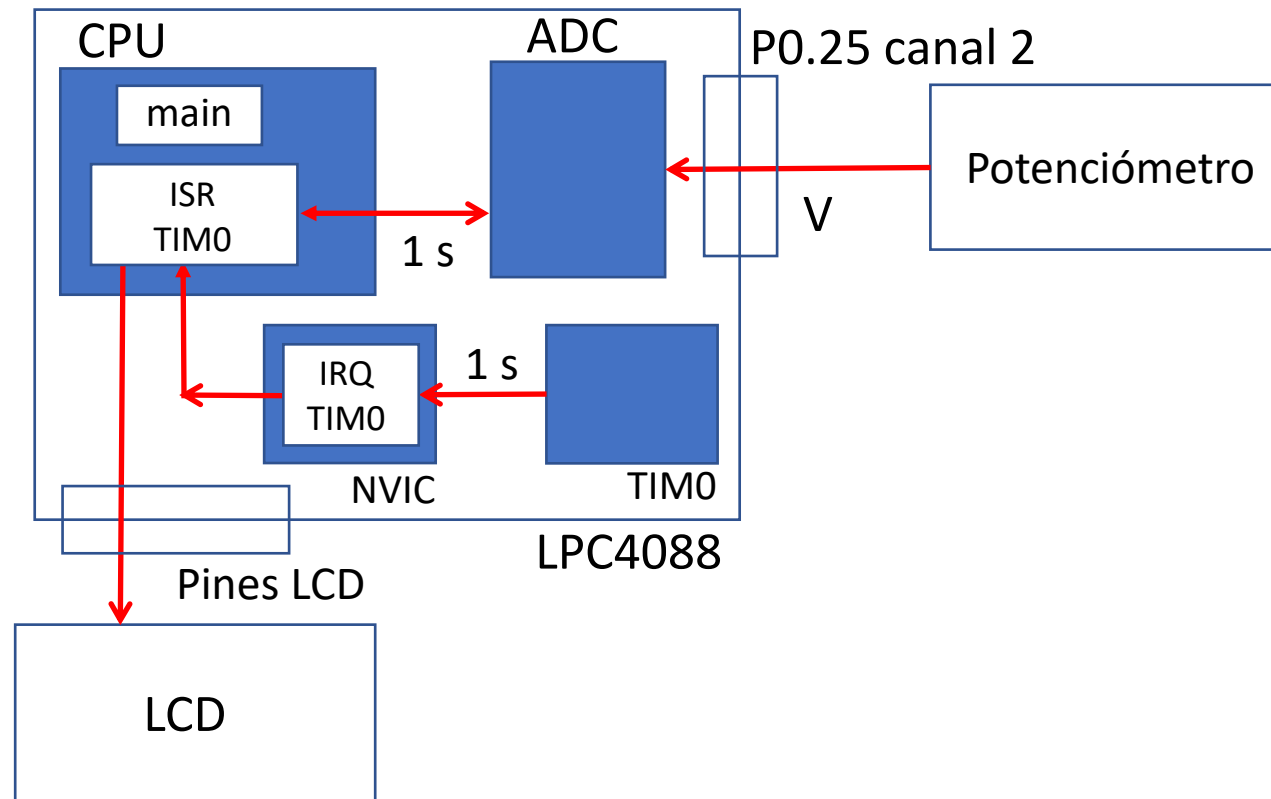
```
    /* Obtener el valor en DR[2] */
```

```
    return ((LPC_ADC->DR[2] >> 4) & 0xFFF);
```

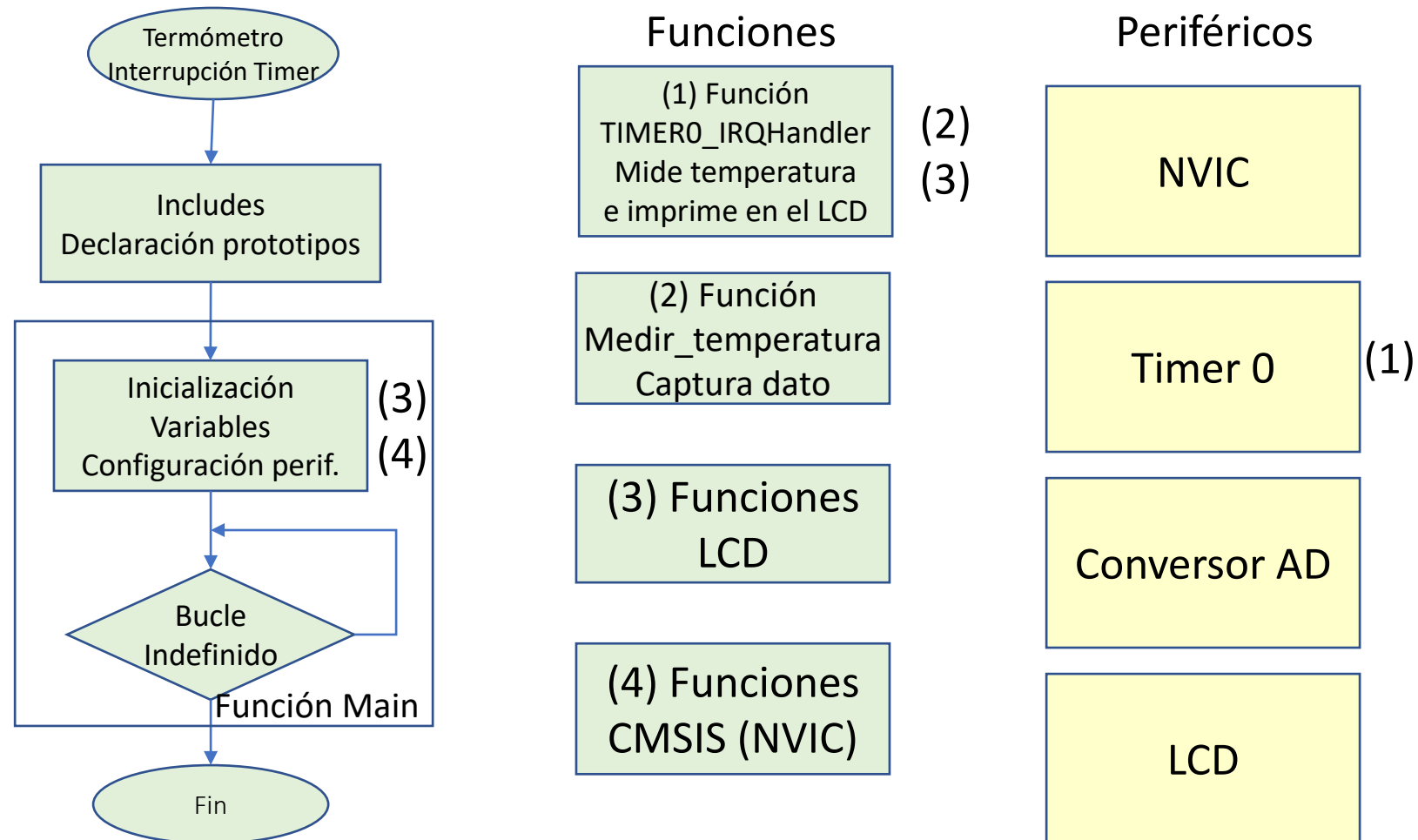
```
}
```

2.5.12. Ejemplo ADC con timer e interrupción (1)

Obtener el valor digital de una conversión A/D de un potenciómetro conectado al canal 2 del LPC4088. Usar el timer 0 para generar un retardo de un segundo entre muestra y muestra usando su interrupción para realizar la captura.



Ejemplo de ADC con timer e interrupción (2)



Ejemplo de ADC con timer e interrupción (3)

```
/*=====
Fichero: termometro_con_timer0_e_interrupcion.c
Termómetro digital con un LM35 conectado al canal analógico AD0.5 del LPC4088. Se usa el
timer 0 para generar una interrupción cada segundo. La medida de la temperatura y la
impresión en pantalla del valor se realiza dentro de la subrutina de interrupción. El LM35
tiene una precisión de 10 mV/°C.
=====*/
```

```
#include <LPC407x_8x_177x_8x.h>
```

```
#include "glcd.h"
```

```
#include "timer_lpc40xx.h"
```

```
float medir_temperatura(void);
```

```
void TIMER0_IRQHandler(void);
```

```
int main(void)
```

```
{ glcd_inicializar();
```

```
  glcd_borrar();
```

```
  /* Configurar el convertidor A/D */
```

```
  /* Seleccionar la función de entrada al convertidor en el pin P1[31]/USB_OVRCCR2/
```

```
  SSP1_SCK/AD0[5]/I2C0_SCL */
```

```
  LPC_IOCON->P1_31 |= 3;
```


Ejemplo de ADC con timer e interrupción (4)

```
/* Activar la alimentación del convertidor.*/  
LPC_SC->PCONP |= 1 << 12;  
/* Ajustar el registro de control AD0CR del convertidor A/D:  
- Activar el bit PDN para sacar el A/D de modo powerdown.  
- Programar el campo CLKDIV para una frecuencia ADC de 1 MHz*/  
LPC_ADC -> CR = (1<<21) | ((PeripheralClock/1000000-1)<<8);  
/*Configurar y habilitar la interrupción del Timer 0. */  
    NVIC_ClearPendingIRQ(TIMER0_IRQn);  
    NVIC_EnableIRQ(TIMER0_IRQn);  
    NVIC_SetPriority(TIMER0_IRQn, 0);  
    __enable_irq();  
/* Programar el Timer 0 para que provoque un evento "match" cada sg.*/  
    timer_inicializar(TIMER0);  
    timer_iniciar_ciclos_ms(TIMER0);  
    while (1);  
}
```

Ejemplo de ADC con timer e interrupción (5)

```
/*=====
Función:  TIMER0_IRQHandler
Utilidad: Función de la interrupción del timer 0.
Mide la temperatura con el LM35 y la muestra en el LCD.
Entrada:  Ninguna.
Retorna:  Nada.
=====*/

void TIMER0_IRQHandler(void)
{
    float temperatura = 0;

    NVIC_ClearPendingIRQ (TIMER0_IRQn); /* Borrar el flag de petición. */
    LPC_TIM0 -> IR = 1;
    temperatura = medir_temperatura();
    glcd_printf (0,0,WHITE, BLACK, FONT16X32, "Valor: %1.2f °C",
                temperatura);
}
```

Ejemplo de ADC con timer e interrupción (6)

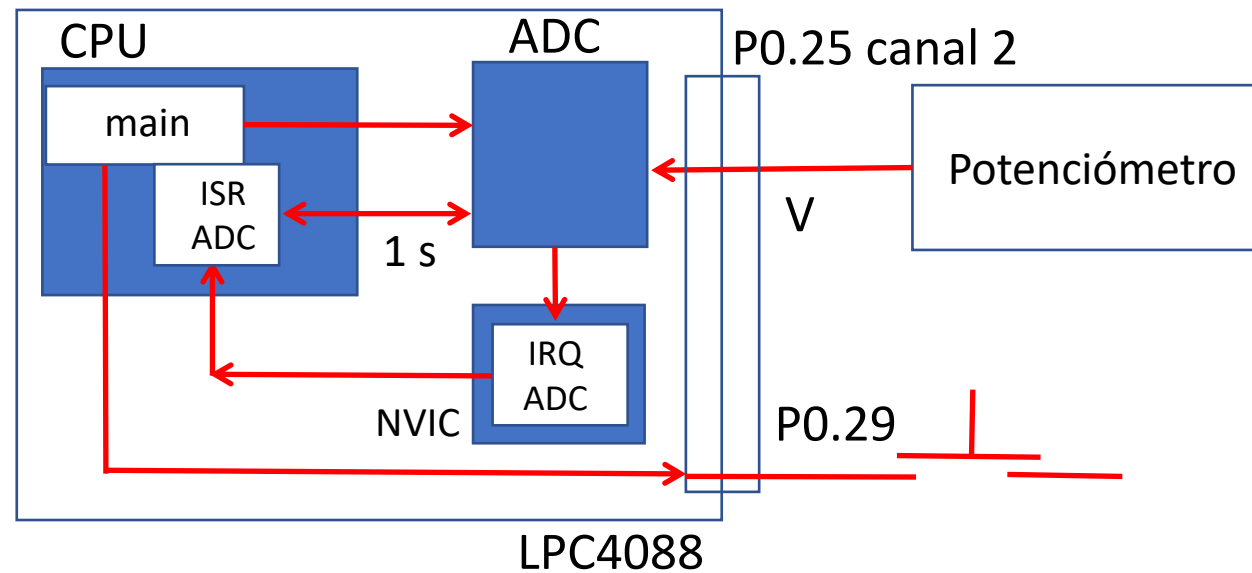
```
/*=====
Función:  medir_temperatura
Utilidad: Convierte a °C la señal de salida de un sensor de temperatura LM35 con una
precisión de 10 mV/°C aplicada al canal 5 del convertidor A/D del LPC4088
Entrada: Ninguna.
Retorna: °C de la señal analógica
=====*/

float medir_temperatura(void)
{
    /* Seleccionar sólo la entrada 5 en el campo SEL de CR.*/
    LPC_ADC->CR = (LPC_ADC->CR & ~0xFF) | (1<<5);
    /* Disparar una conversión. */
    LPC_ADC->CR |= 1<<24;
    /* Esperar a que finalice comprobando el bit DONE del registro GDR.*/
    while (!(LPC_ADC->GDR & (1<<31)));
    /* Obtener el valor en DR[5] y convertirlo en °C*/
    return (((LPC_ADC->DR[5] >> 4) & 0xFFF) * 330 ) / 2^12 );
}
```

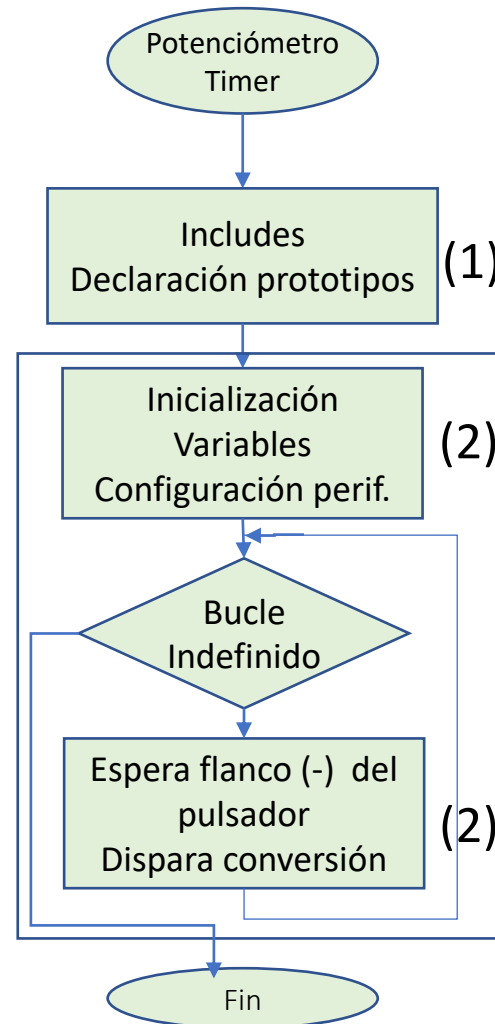
2.5.13. Ejemplo de interrupción por el ADC (1)

```
/*=====
* Se ha conectado un sensor de presión barométrica al canal 5 del ADC.
* El sensor produce una tensión de salida
*   Vout = 0.25 V/bar = 0.00025 V/mbar (voltios/milibar)
*   Vout = 0.00025P voltios, donde P es la presión en milibares.
* La relación entre el código de salida y la tensión de entrada del ADC es
*   VinADC = 3.3*C/1024
* Al conectar el sensor al ADC Vout = VinADC, luego:
*   0.00025P = 3.3*C/1024 => P = 3.3*C/(0.00025*1024) = 330000*C/(25*1024)
* Se mantienen las ctes para evaluar la expresión en el micro.
* Cada vez que se pulsa el botón BUT1 conectado al pin P0.29 se inicia por
* software una conversión. Al finalizar la conversión se dispara la interrupción
* del ADC. En la rutina de servicio de interrupción se recoge el resultado de la
* conversión, se calcula la presión correspondiente (en milibares) y se
* almacena en un buffer circular de 10 muestras.
*=====*/
```

Ejemplo de interrupción por el ADC (2)



Ejemplo de interrupción por el ADC (3)



Funciones

(1) Función
ADC_IRQHandler
Captura y
almacena dato (2)

(2) Funciones
CMSIS (NVIC)

Periféricos

NVIC (2)

Conversor AD (1)

Ejemplo de interrupción por el ADC (4)

```
#include "LPC407x_8x_177x_8x.h"

void ADC_IRQHandler(void);

int main(void)
{ LPC_IOCON->P1[31] |= 3; /*Función de entrada al convertidor P1[31]/USB_OVRCR2/
                           SSP1_SCK/ AD0[5]/I2C0_SCL*/
  LPC_SC -> PCONP |= (1 << 12); /* Activar la alimentación del convertidor.*/
  /*Activar el bit PDN y CLKDIV para conversor a 1 MHz*/
  LPC_ADC -> CR = (1<<21) | ((PeripheralClock/1000000-1)<<8);
  /*Habilitar interrup. cuando bit DONE del AD0DR5 se ponga a 1.*/
  LPC_ADC -> INTEN = (1<<5);
  NVIC_ClearPendingIRQ (ADC_IRQn);
  NVIC_EnableIRQ (ADC_IRQn);
  NVIC_SetPriority (ADC_IRQn, 0);
  __enable_irq();
```

Ejemplo de interrupción por el ADC (5)

```
while (1)
{
    /* Esperar flanco negativo en P0.29 producido por el botón BUT1.*/

    while(LPC_GPIO0->PIN & (1<<29));
    while(!(LPC_GPIO0->PIN & (1<<29)));

    /* Disparar una nueva conversión del canal 5. Cuando la conversión
     * finalice se ejecutará la rutina de interrupción ADC_IRQHandler*/

    LPC_ADC -> CR = (LPC_ADC -> CR & 0xFFFFFFF00) | (1<<5) | (1<<24);
}
} /* Fin del main*/
```


Ejemplo de interrupción por el ADC (6)

```
/*=====
 * Función:  ADC_IRQHandler
 * Rutina de servicio de interrupción del convertidor A/D.
 *=====*/
void ADC_IRQHandler(void)
{ static uint32_t buffer_circular_presiones[10];
  static uint32_t indice_buffer = 0;
  uint32_t resultado_adc;
  uint32_t presion;
  NVIC_ClearPendingIRQ (ADC_IRQn); /* Borrar el flag de petición. */
/*Al leer el reg. AD0DR5 se borra el flag de petición de interrup. (ADINT en AD0STAT)*/
  resultado_adc = (LPC_ADC -> DR[5] >> 4) & 0xFFF;
  presion = 330000*resultado_adc/(25*2^12);
  buffer_circular_presiones[indice_buffer] = presion;
  indice_buffer++;
  if (indice_buffer > 9) indice_buffer = 0;
}
```