Diseño Basado en Microprocesadores
Práctica 10
Convertidor A/D del LPC4088

Índice

---

1. El convertidor analógico/digital del LPC4088

El microcontrolador LPC4088 incorpora un convertidor analógico/digital (ADC) con las siguientes características:

- Convertidor de aproximaciones sucesivas.
- 12 bits de resolución.
- 8 canales de entrada multiplexados.
- Rango de medida unipolar de VSS a VREFP (VREFP ≤ VDDA). Tasa de
- conversión de hasta 400 kHz.
- Modo ráfaga.
- Inicio de conversión por software, timer o pin externo.
- Modo de bajo consumo.

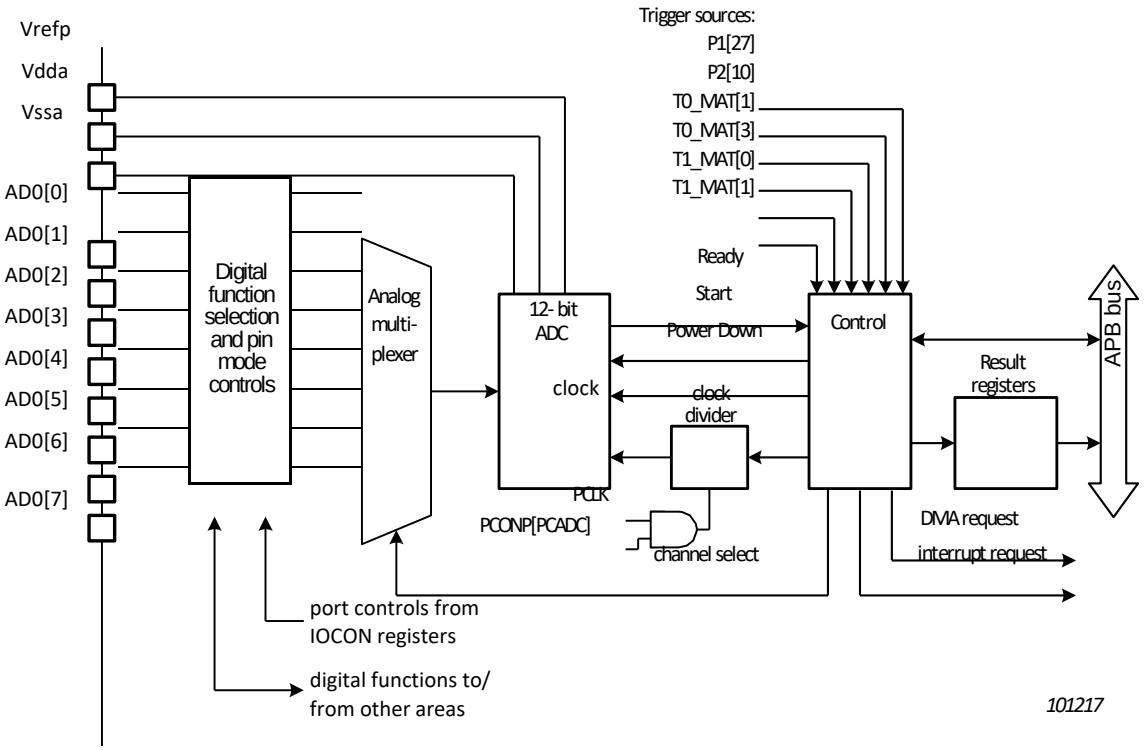


Figura 1: Diagrama de bloques del ADC del LPC4088.

1.1. Registros del ADC

Para interactuar con el ADC, el software usa los registros que se indican en el cuadro 1. En las siguientes secciones se describe la estructura interna de cada uno de ellos.

Cuadro 1: Registros del ADC del LPC4088.

Registro	Descripción
CR	Registro de control.
GDR	Registro de datos global.
DRn	Registros de datos individuales de cada canal (n entre 0 y 7). INTEN Registro de interrupciones.
STAT	Registro de estado.
TRM	Registro de ajuste.

1.1.1. Registro de control CR

El registro de control sirve para configurar los canales a convertir, la temporización, el modo de funcionamiento y el modo de disparo del ADC.

Cuadro 2: Registro de control CR.

Bit(s)	Símbolo	Descripción	Reset
7:0	SEL	Selecciona cuales de los pines AD0[7:0] son muestreados y convertidos. El bit 0 selecciona el pin AD0[0], el bit 1 el pin AD0[1], etc. En el modo controlado por software, solo uno de los bits debería estar a 1. En modo de barrido por hardware se permite cualquier combinación de bits a 1. Si todos los bits están a 0 equivale a que esté a 1 el bit 0.	1
15:8	CLKDIV	La señal de reloj del bus APB (PCLK) se divide entre el valor de este campo más uno para generar la señal de reloj para el convertidor. El reloj del convertidor debe tener una frecuencia menor o igual a 12.4 MHz.	0
16	BURST	Si BURST = 0 la conversión está controlada por software y requiere 31 ciclos de reloj. Si BURST = 1 el convertidor realiza conversiones continuamente al ritmo seleccionado por el campo CLKDIV barriendo si es necesario los pines seleccionados por los unos en el campo SEL. El proceso de conversión puede ser interrumpido poniendo a 0 este bit pero la conversión en curso se completará. **Importante:** Los bits START deben estar a 000 cuando BURST = 1 o las conversiones no se iniciarán.	0
20:17		Reservados. Valor leído indefinido. Sólo deben escribirse ceros.	NA
21	PDN	Si PDN = 1 el ADC opera normalmente. Si PND = 0 el ADC está en modo de bajo consumo.	0
23:22		Reservados. Valor leído indefinido. Sólo deben escribirse ceros.	NA
26:24	START	Cuando el bit BUSRT es 0, estos bits controlan el momento en el que se inicia la conversión. START = 000 ⇒ Conversiones detenidas. Debe seleccionarse al poner PDN a 0. START = 001 ⇒ Comenzar la conversión inmediatamente. START = 010 ⇒ Flanco en P2[10]/EINT0. START = 011 ⇒ Flanco en P1[27]/CAP0.1. START = 100 ⇒ Flanco en MAT0[1]. START = 101 ⇒ Flanco en MAT0[3]. START = 110 ⇒ Flanco en MAT1[0]. START = 111 ⇒ Flanco en MAT1[1].	0
27	EDGE	Este bit sólo es relevante si START contiene un combinación entre 010 y 111. En estos casos, si EDGE = 0 la conversión comienza cuando se aplica un flanco de subida al pin correspondiente, mientras que si EDGE = 1 la conversión comienza cuando se aplica un flanco de bajada.	
31:28		Reservados. Valor leído indefinido. Sólo deben escribirse ceros.	NA

Los valores 100, 101, 110 y 111 del campo START causan que la conversión comience cuando se produzcan eventos de match en los timer 0 o 1. Para ello, hay que programar

convenientemente el registro EMR (*External Match Register*) del timer elegido. Por ejemplo, podemos programar el timer 0 o 1 para que realice ciclos de una duración determinada (activando el bit de reset correspondiente de su registro MCR) y disparar automáticamente una conversión A/D al final de cada ciclo. Esto generará una frecuencia de muestreo exacta y constante. Para conseguir iniciar las conversiones de esta forma no es necesario configurar la función de match externo en los pines correspondientes.

1.1.2. Registro de datos global GDR

El registro de datos global (GDR) almacena el resultado de la conversión A/D más reciente que se haya completado. También incluye una copia de los flags asociados a esa conversión. El resultado de una conversión puede leerse de este registro o de uno de los registros DRn asociados a cada canal. Es importante usar sólo uno de los dos métodos porque en caso contrario los flags DONE y OVERRUN pueden desincronizarse causando errores en las interrupciones o el DMA.

Cuadro 3: Registro de datos global GDR.

Bit(s)	Símbolo	Descripción	Reset
3:0		Reservados. El valor leído es indefinido. Sólo deben escribirse ceros.	0
15:4	RESULT	Cuando el bit DONE se pone a 1, estos 12 bits proporcionan el valor de la fracción que la tensión aplicada al pin AD0[n] seleccionado por el campo SEL del registro CR representa respecto a la tensión aplicada al pin de referencia VREFP. Si es cero significa que la tensión aplicada es menor, igual o muy próxima o cero. Si es 0xFFF significa que la tensión aplicada es muy cercana, igual o superior a VREFP.	NA
23:16		Reservados. Valor leído indefinido. Sólo deben escribirse ceros.	NA
26:24	CHN	Estos bits indican el canal al que corresponde el resultado en RESULT	NA
29:27		Reservados. Valor leído indefinido. Sólo deben escribirse ceros.	NA
30	OVERRUN	En modo BURST, se pone a 1 para indicar que se han sobrescrito y perdido uno o más resultados de conversión antes de haber sido leídos. Se pone a cero leyendo este registro.	0
31	DONE	Este bit se pone a uno cuando se completa una conversión A/D. Se pone a cero cuando se lee este registro o cuando se escribe en CR. Si se escribe en CR mientras está en curso una conversión, este bit se pone uno y se inicia una nueva conversión.	0

1.1.3. Registros de datos individuales DRn

Estos ocho registros almacenan el resultado de la última conversión de cada canal. También incluyen flags que indican si la conversión ha terminado o si se ha producido un *overrun* (sobrescritura de un resultado con otro antes de que el primero haya sido leído).

Cuadro 4: Registros de datos DRn (n de 0 a 7).

Bit(s)	Símbolo	Descripción	Reset
3:0		Reservados. El valor leído es indefinido. Sólo deben escribirse ceros.	NA
15:6	RESULT	Cuando el bit DONE se pone a 1, estos 12 bits proporcionan el valor de la fracción que la tensión aplicada al correspondiente pin AD0[n] representa respecto a la tensión aplicada al pin de referencia VREFP. Si es cero significa que la tensión aplicada es menor, igual o muy próxima o cero. Si es 0x3FF significa que la tensión aplicada es muy cercana, igual o superior a VREFP.	NA
29:16		Reservados. El valor leído es indefinido. Sólo deben escribirse ceros.	NA
30	OVERRUN	En modo BURST, se pone a 1 para indicar que se han sobrescrito y perdido uno o más resultados de conversión antes de haber sido leídos. Se pone a cero leyendo este registro.	0
31	DONE	Este bit se pone a uno cuando se completa una conversión A/D de este canal. Se pone a cero cuando se lee este registro.	0

1.1.4. Registro de estado STAT

El registro de estado permite comprobar el estado de todos los canales simultáneamente. Los flags DONE y OVERRUN de todos los canales se reflejan en este registro. El flag de interrupción ADINT es el OR de los flags DONE de todos los canales.

Cuadro 5: Registro de estado STAT

Bit(s)	Símbolo	Descripción	Reset
0	DONE0	Refleja el mismo estado que el bit DONE del reg. DR0.	0
1	DONE1	Refleja el mismo estado que el bit DONE del reg. DR1.	0
2	DONE2	Refleja el mismo estado que el bit DONE del reg. DR2.	0
3	DONE3	Refleja el mismo estado que el bit DONE del reg. DR3.	0
4	DONE4	Refleja el mismo estado que el bit DONE del reg. DR4.	0
5	DONE5	Refleja el mismo estado que el bit DONE del reg. DR5.	0
6	DONE6	Refleja el mismo estado que el bit DONE del reg. DR6.	0
7	DONE7	Refleja el mismo estado que el bit DONE del reg. DR7.	0
8	OVERRUN0	Refleja el mismo estado que el bit OVERRUN del registro DR0.	0
9	OVERRUN1	Refleja el mismo estado que el bit OVERRUN del registro DR1.	0
10	OVERRUN2	Refleja el mismo estado que el bit OVERRUN del registro DR2.	0
11	OVERRUN3	Refleja el mismo estado que el bit OVERRUN del registro DR3.	0
12	OVERRUN4	Refleja el mismo estado que el bit OVERRUN del registro DR4.	0
13	OVERRUN5	Refleja el mismo estado que el bit OVERRUN del registro DR5.	0
14	OVERRUN6	Refleja el mismo estado que el bit OVERRUN del registro DR6.	0
15	OVERRUN7	Refleja el mismo estado que el bit OVERRUN del registro DR7.	0
16	ADINT	Bit de petición de interrupción. Se pone a uno cuando cualquiera de los bits DONE de los registros DRn se pone a uno y el correspondiente canal ha sido habilitado en INTEN.	0
31:17		Reservados. El valor leído es indefinido. Sólo deben escribirse ceros.	0

1.1.5. Registro de interrupción INTEN

Este registro permite controlar qué canales generan una interrupción cuando la conversión ha finalizado.

Cuadro 6: Registro de habilitación de interrupciones INTEN

Bit(s)	Símbolo	Descripción	Reset
0	ADINTEN0	Habilitación de interrupciones de fin de conversión del canal 0.	0
1	ADINTEN1	Habilitación de interrupciones de fin de conversión del canal 1.	0
2	ADINTEN2	Habilitación de interrupciones de fin de conversión del canal 2.	0
3	ADINTEN3	Habilitación de interrupciones de fin de conversión del canal 3.	0
4	ADINTEN4	Habilitación de interrupciones de fin de conversión del canal 4.	0
5	ADINTEN5	Habilitación de interrupciones de fin de conversión del canal 5.	0
6	ADINTEN6	Habilitación de interrupciones de fin de conversión del canal 6.	0
7	ADINTEN7	Habilitación de interrupciones de fin de conversión del canal 7.	0
8	ADGINTEN	Habilitación de interrupción cuando el bit DONE de GDR se pone a uno.	1
31:9		Reservados. Valor leído indefinido. Sólo deben escribirse ceros.	NA

1.1.6. Registro de ajuste TRIM

Este registro lo ajusta el código de inicialización del microcontrolador. Contiene valores de ajuste para el ADC y el DAC. El valor de ajuste de offset puede ser sobrescrito. El fabricante del microcontrolador no da información sobre el significado de los valores en este registro, así que carece de utilidad para nosotros.

Cuadro 7: Registros de ajuste TRIM

Bit(s)	Símbolo	Descripción	Reset
3:0		Reservados. El valor leído es indefinido. Sólo deben escribirse ceros.	NA
7:4	ADCOFFS	Ajuste de offset. Lo inicializa el código de arranque. Puede ser sobrescrito por el usuario.	0
11:8	TRIM	Escrito por el código de arranque. No puede ser sobrescrito por el usuario.	0
31:12		Reservados. El valor leído es indefinido. Sólo deben escribirse ceros.	NA

2. Funciones de manejo del ADC

Para facilitar el manejo del ADC desde nuestros programas, se ha desarrollado el fichero fuente adc_lpc40xx.c y su correspondiente fichero de cabecera adc_lpc40xx.h con las siguientes funciones:

- `void adc_inicializar(uint32_t frecuencia_adc, uint32_t canales)`

  Inicializa el ADC sacándolo de modo de bajo consumo, ajusta la frecuencia de reloj de funcionamiento interna y selecciona la función analógica en los canales deseados. El significado de los parámetros es:

  frecuencia_adc indica la frecuencia de reloj a la que se quiere hacer funcionar el ADC.

  canales indica qué canales analógicos se van a usar. La función selecciona la función analógica en los pines correspondientes. Los bits 0 a 7 del argumento canales indica si debe activarse el canal analógico correspondiente. Por ejemplo, si canales vale 0x61 (01100001 en binario) la función activará la función analógica en los pines que tienen como función alternativa los canales 0, 5 y 6, que son los bits que están a 1 en dicho valor.

- `uint32_t adc_convertir(uint32_t canal)`

  Selecciona el canal indicado por el parámetro canal y dispara una conversión. Espera hasta que la conversión finalice y retorna el resultado de 12 bits.

- `float32_t adc_traducir_a_tensión(uint32_t resultado_adc)`

  Recibe el resultado de una conversión A/D y devuelve la tensión de entrada al ADC correspondiente.

3. Ejercicios

3.1. Ejercicio 1

En la esquina superior izquierda de tarjeta LPC4088 Developer's Kit (ver figura 2) hay un potenciómetro cuyo cursor está conectado al canal 2 del ADC del LPC4088. Los extremos del potenciómetro están conectados a VSSA (0 voltios) y V3A (3.3 voltios) (ver la figura 3).

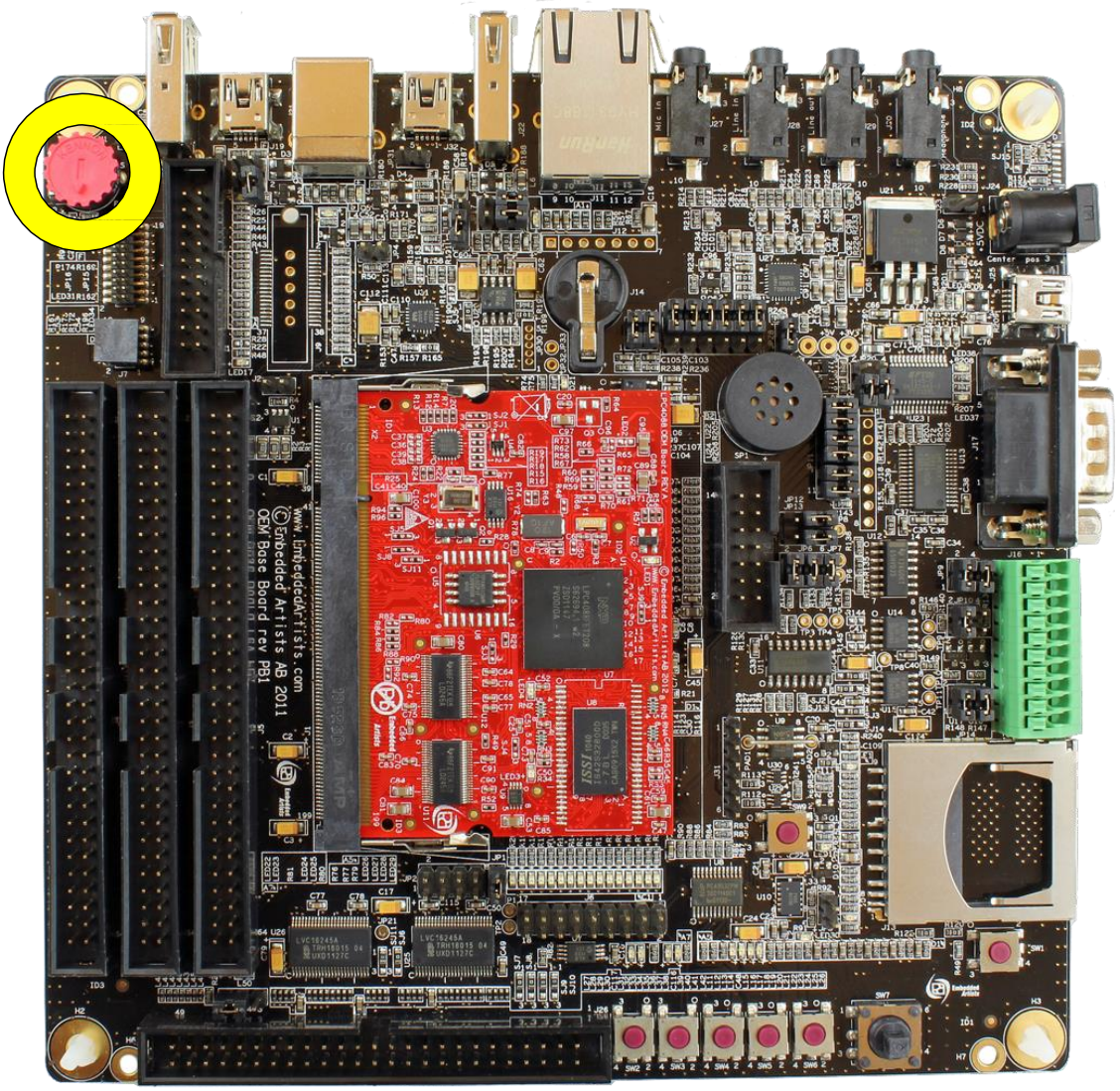


Figura 2: Potenciómetro en la tarjeta LPC4088 DEveloper's Kit.

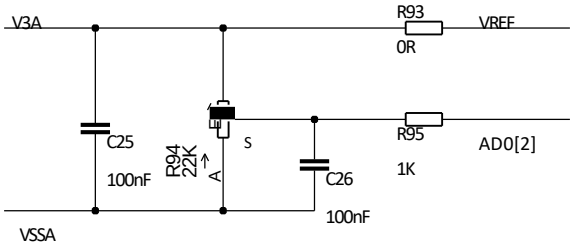
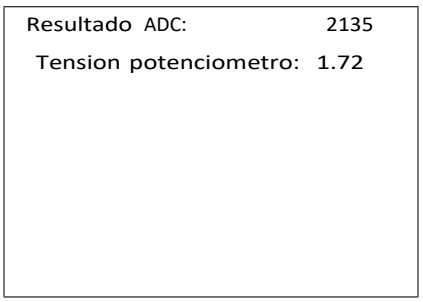


Figura 3: Esquema de conexiones del potenciómetro.

El canal 2 del convertidor A/D está accesible en el pin P0[25]/**AD0[2]**/I2S_RX_SDA/U3_TXD del microcontrolador.

Escribe un programa que muestre en la pantalla LCD la tensión en el cursor del poten- ciómetro. El programa también debe mostrar en la pantalla el dato obtenido directamente del ADC. Por ejemplo



La medida debe actualizarse una vez por segundo. Emplea un timer para generar el tiempo entre medidas (usa las funciones timer_inicializar, timer_iniciar_ciclos_ms, timer_esperar_fin_ciclo).

Al probar el programa observarás que la indicación de tensión fluctúa bastante. Esto se debe al ruido en la tensión de alimentación de 3.3 V a la que está conectado uno de los extremos del potenciómetro. Por otra parte, en el diseño de la tarjeta no se puesto cuidado en la tensión de referencia del convertidor sino que está conectada directamente a la tensión de alimentación de 3.3 V así que el ruido presente en la alimentación afecta mucho a las conversiones. Para evitar este problema el fabricante de la tarjeta debería haber incluido un circuito de referencia de tensión de precisión y conectar su salida al pin de referencia de tensión, VREFP, del microcontrolador.

3.2. Ejercicio 2

En este ejercicio conectaremos un módulo sensor que usa un termistor NTC para medir temperatura. Como se muestra en la figura 4, el módulo está compuesto por un divisor de tensión formado por una resistencia fija de 10 kΩ y una resistencia de coeficiente negativo de temperatura NTC o termistor NTC.

El módulo se suministrará con cables de color rojo (alimentación), negro (GND) y blanco (salida de señal). Conectaremos el módulo al conector J5 de la tarjeta LPC4088 Developer's Kit como se muestra en la figura 5. Esto establecerá las conexiones que se indican en el cuadro 8.

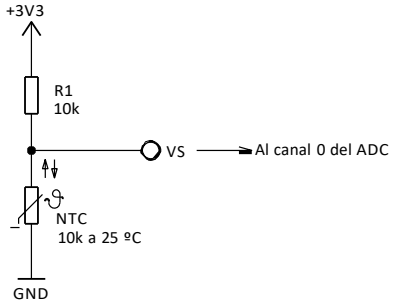


Figura 4: Esquema del módulo sensor con NTC.

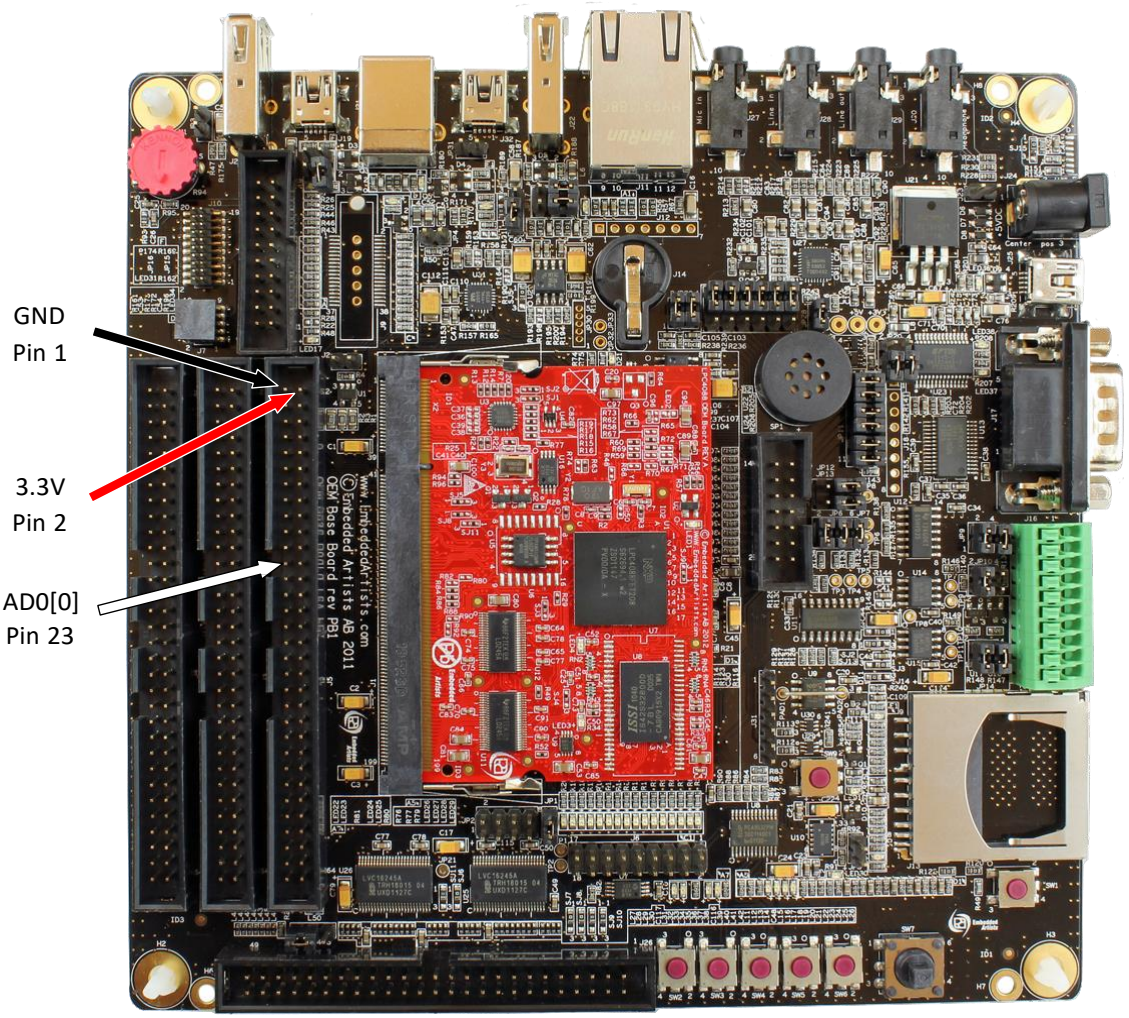


Figura 5: Conexión del módulo NTC al conector J5 de la tarjeta.

Cuadro 8: Conexiones del módulo sensor al conector J5.

Terminal del sensor	**Señal del sensor**	**Pin de J5**
Negro	GND	1
Rojo	+3.3V	2
Blanco	Salida (V_S)	23

La salida del módulo sensor (V_S) quedará conectada al pin P0[23]/**AD0[0]**/I2S_RX_SCK/T3_CAP0 del microcontrolador.

La tensión de salida del módulo es la caída de tensión en la NTC.

$$V_S = \frac{R_{NTC}}{R_1 + R_{NTC}} V_{CC} \quad (V)$$

Resolviendo para R_{NTC}

$$R_{NTC} = \frac{R_1 V_S}{V_{CC} - V_S} \quad (\Omega)$$

La relación entre la resistencia de la NTC y la temperatura a la que está sometida está dada por la ecuación de Steinhart-Hart

$$T = \frac{1}{a + b\ln(R_{NTC}) + c(\ln(R_{NTC}))^3} - 273.16 \quad (^\circ C)$$

Los coeficientes a, b y c dependen del tipo de NTC. Tomaremos los siguientes valores

a = 0.5089218645E-3f; b = 0.2484818972E-3f; c = 0.01313142875E-6f; (termo retráctil negro)

o $a = 1.129 \times 10^{-3}$ $b = 2.341 \times 10^{-4}$ $c = 8.767 \times 10^{-8}$ (cinta Kapton)

Conecta los terminales del módulo sensor a los pines del conector J5 de la tarjeta LPC4088 Developer's Kit que se indican en el cuadro 8. Guíate por la figura 5. **La conexión debe hacerse con la tarjeta apagada y desconectada del PC**.

Amplía el programa del ejercicio anterior para que, además de la tensión del potenciómetro, también se muestre en el LCD la temperatura ambiente. La temperatura debe mostrarse con una cifra decimal.

Para traducir la tensión de salida del módulo sensor en la temperatura correspondiente usa una función con el siguiente prototipo

```
float32_t ntc_traducir_tension_a_temperatura(float32_t V_s)
```

donde el argumento `V_s` es la tensión que entrega el módulo sensor de temperatura basado en NTC de la figura 4. Esta función debe estar en el fichero fuente `termistor_ntc.c`. Será de utilidad la función `log` de `math.h` que permite calcular el logaritmo natural.

3.3. Ejercicio 3

Al probar el programa observarás que la indicación de temperatura fluctúa mucho. Esto no se debe a que la temperatura esté variando rápidamente. La causa es que, como se comentó antes, en el diseño de la tarjeta no se han cuidado adecuadamente las tensiones de alimentación y de referencia del convertidor. Estas tensiones se toman de la tensión de alimentación de +3.3 V general que alimenta al microcontrolador, por lo que está afectada de gran cantidad de ruido. Para paliar el problema podemos promediar un número grande de medidas individuales.

Modifica el programa para que la temperatura mostrada en la pantalla sea el promedio de 100 medidas de temperatura tomadas con un intervalo de 10 ms entre medidas.