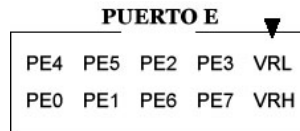


**Guía práctica para el uso de
sensores de distancia con el
microcontrolador
MC68HC11 sobre una tarjeta
CT6811**

Gustavo García Pascual

Guía rápida para el uso de sensores de distancia con el microcontrolador MC68HC11 sobre la tarjeta CT6811

El microcontrolador MC68HC11 permite el uso de un conversor A/D con 8 canales situados en su puerto E. Este puerto se corresponde también con el puerto E de la placa CT6811, cuya estructura es la siguiente:



El primer paso es configurar el microcontrolador para que el **puerto E** funcione en modo analógico, de modo que se active el conversor A/D que tiene integrado.

Para tener la mayor precisión posible en la medida es conveniente poner los valores *VRH* y *VRL* a los valores máximo y mínimo de tensión que vamos a medir. En el caso de que estemos midiendo señales procedentes de sensores de distancia SHARP GP2D12 estos valores serán de 3 V y masa. En cualquier caso, los valores *VRH* y *VRL* no deben exceder de los 6 V por arriba y los 0 V por abajo. Hay que tener cuidado con las tensiones que se introducen en este puerto ya que, aunque se permite superar la tensión *VCC*, **una tensión inferior a *GND* podría dañar irreversiblemente el microcontrolador, ya sea en *VRL* como en cualquiera de los 8 canales.**

Para activar el modo de conversión A/D del microcontrolador hay que definir un bit del registro ***OPTION***, cuya dirección es \$1039. Dicho bit es el *ADPU*. Hay que ponerlo a nivel alto, y entonces el microcontrolador entenderá que se va a usar el conversor. Como la conversión se realiza mediante unos condensadores, habrá que esperar un tiempo después de activar este bit para poder empezar a hacer medidas.

Para situarnos con más claridad mostramos aquí la sintaxis del registro ***OPTION*** (\$1039).

ADPU	CSEL	IRQE	DLY	CME	Sin Uso	CR1	CR0
7	6	5	4	3	2	1	0

El bit *CSEL* es conveniente ponerlo a nivel bajo, ya que así el microcontrolador trabaja a la misma frecuencia del bus, consiguiendo una mayor velocidad en las conversiones y minimizando el nivel de ruido.

Otro registro importante en la conversión A/D es el ***ADCTL*** (\$1030). Nos permite seleccionar los canales que vamos a usar y nos indica cuándo se ha completado la conversión. Su estructura es la siguiente:

CCF	SIN Uso	SCAN	MULT	CD	CC	CB	CA
7	6	5	4	3	2	1	0

El bit *CCF* se pone a 1 cuando la conversión ha concluido y se han guardado los resultados en unos registros que explicaremos posteriormente (***ADR1-ADR4***). Incluyendo una instrucción de espera que compruebe el estado de este bit podremos saber en qué instante se ha completado la conversión (esto se podrá ver con más detalle más adelante en un ejemplo).

El bit *SCAN* activa el modo de conversión continua. Si se pone a 1, se estarán realizando mediciones continuamente, sin que se le indique. Si se pone a 0, se realizarán las conversiones sólo cuando nosotros lo indiquemos. En ese caso, el microcontrolador entenderá que deseamos hacer una nueva conversión cuando escribamos en el registro *ADCTL*.

El bit *MULT* establece el modo de múltiple canal o de único canal. El modo de múltiple canal, que es el que usaremos, realiza la conversión de los 4 canales y guarda cada resultado en un registro distinto. En ese caso, los bits *CD* y *CC* sirven para elegir cuáles serán los cuatro canales que se usarán. Para activar este modo habrá que poner este bit a 1. En el modo de único canal se realizan cuatro conversiones consecutivas sobre el mismo canal, especificado en los bits *CD-CA*, que explicamos a continuación, y se guardan los resultados en los registros *ADR1-ADR4*. Para activar este modo habrá que poner este bit a 0.

En cuanto a los bits *CD-CA*, como se deja intuir, su función es la de la selección de los canales que se usarán en la conversión. Si usamos el modo de múltiple canal (que en la práctica será el que usemos, ya que emplearemos al menos 2 sensores de distancia), sólo nos interesa el bit *CC*, dejando los demás para el olvido. Si usamos el modo de único canal, elegiremos qué canal activamos mediante los bits *CC-CA*. El bit *CD* debe ponerse siempre a 0, ya que al ponerlo a 1 el microcontrolador entra en un modo especial que no nos interesa. En la siguiente tabla se muestra qué configuración de los bits *CC-CA* debemos usar según la funcionalidad que necesitamos:

Modo de único canal (<i>MULT=0</i>)			
CC	CB	CA	Canal medido
0	0	0	PE0
0	0	1	PE1
0	1	0	PE2
0	1	1	PE3
1	0	0	PE4
1	0	1	PE5
1	1	0	PE6
1	1	1	PE7

Modo de múltiple canal (<i>MULT=1</i>)		
CC	Canales medidos	Resultados guardados en
0	PE0-PE3	ADR1 a ADR4 respectivamente
1	PE4-PE7	ADR1 a ADR4 respectivamente

Como indicamos anteriormente, los resultados de las conversiones se guardan en cuatro registros de 8 bits, mapeados en las siguientes direcciones de memoria:

ADR4 \$1034
ADR3 \$1033
ADR2 \$1032
ADR1 \$1031

El bit más significativo de cada registro coincide con el bit más significativo de la conversión. Como sólo hay cuatro registros no se pueden muestrear 8 canales a la vez, aunque existan los 8 canales.

Veamos como ejemplo este programa que realiza la conversión de la señal proveniente de un potenciómetro que cambia sus valores entre *Gnd* y *Vcc*. Este ejemplo está extraído del manual de usuario de la tarjeta *CT293+* de la empresa Microbótica S.L.

```

;+-----+
;| Programa ejemplo para ser ejecutado en la tarjeta CT6811.
;| Este programa se debe cargar en la RAM interna del 6811.
;|
;| Ejemplo de utilización del canal 0 del conversor A/D. Cuando la
;| tensión supera los 2.5 voltios se enciende el led de la placa.
;+-----+
; Usaremos como entrada analógica el canal 1, es decir la
; entrada PE0. Los niveles de referencia (VRH, VRL) serán (Vcc,GND)
; respectivamente. Es decir los jumpers JP1 y JP2 de la CT6811 deben
; estar conectados.

          ORG $0000
OPTION EQU $39
ADCTL EQU $30
PORTA EQU $00
ADR1 EQU $31

INICIO
  LDX #$1000
  LDAA #$80
  STAA OPTION,X          ; encender el conversor
  LDAA #$20              ; configuración conversor:
  STAA ADCTL,X          ; SCAN -> activo
                          ; MULT -> inactivo
                          ; ADR1 -> seleccionar primer canal
sigue BRCLR ADCTL,X $80 sigue ; espera a que termine conversión
  LDAA ADR1,X           ; leer el resultado de la conversión
  CMPA #$7F             ; comparar con la mitad (127 en decimal)
                          ; que corresponde a 2.5v de entrada.
                          ; si es menor apagar el led
  BLO menor            ; si es menor apagar el led
  LDAA #$40             ; No--> encender el led
  
```

```

        STAA PORTA,X
        BRA sigue           ; Realizar la siguiente conversión
menor  CLRA
        STAA PORTA,X       ; Apagar el led.
        BRA sigue           ; Realizar la siguiente conversión

        END

```

A continuación podemos ver otro programa que sirve como ejemplo para el uso de los sensores de distancia SHARP GP2D12 en modo de único canal, haciendo que el microbot avance hacia delante cuando el sensor detecta un objeto cercano:

```

;+-----+
;| Programa ejemplo para ser ejecutado en la RAM interna
;| de la tarjeta CT6811.
;|
;| Utiliza la entrada PE4 del conversor A/D en modo de
;| único canal. Cuando se detecta un objeto cercano avanza
;| el microbot.
;+-----+
;
; La entrada analógica que usaremos será la PE4.
; Para simplificar el conexionado de prueba, los niveles
; de referencia que usaremos (VRH, VRL) serán (Vcc, GND)
; respectivamente. Es decir, los jumpers JP1 y JP2 de la
; CT6811 deben estar conectados.

        ORG $0000           ; El programa se cargará en la RAM interna
OPTION EQU $39             ; $1039 = Dirección del registro OPTION
ADCTL EQU $30             ; $1030 = Dirección del registro ADCTL
PORTA EQU $00             ; $1000 = Dirección del puerto A
ADR1 EQU $31              ; $1031 = Dirección del registro ADR1

inicio
        LDX #$1000         ; Registro índice X := 1000
        LDAA #$80
        STAA OPTION,X     ; Activamos el conversor A/D
        LDAA #$24         ; Configuramos el conversor:
        STAA ADCTL,X      ; SCAN -> activo (mediciones continuadas)
                          ; MULT -> 0 (canal único)
espera BRCLR ADCTL,X $80 espera ; Esperamos a que termine la conversión
        LDAA ADR1,X       ; Leemos el resultado de la conversión
        CMPA #$30         ; Comparamos el resultado con el umbral
        BLO para          ; Si no hay objeto cerca, nos paramos
        LDAA #$18         ; Si lo encuentra, avanza hacia delante
        STAA PORTA,X
        BRA espera        ; Realiza la siguiente conversión

para   LDAA #$00

        STAA PORTA,X
        BRA espera

        END

```

El siguiente ejemplo hace exactamente lo mismo que el anterior pero usando el modo de múltiple canal:

```

;+-----+
;| Programa ejemplo para ser ejecutado en la RAM interna |
;| de la tarjeta CT6811.                                |
;|                                                       |
;| Utiliza la entrada PE4 del conversor A/D en modo de  |
;| múltiple canal. Cuando se detecta un objeto cercano |
;| avanza el microbot.                                  |
;+-----+
;
; La entrada analógica que usaremos será la PE4.
; Para simplificar el conexionado de prueba, los niveles
; de referencia que usaremos (VRH, VRL) serán (Vcc, GND)
; respectivamente. Es decir, los jumpers JP1 y JP2 de la
; CT6811 deben estar conectados.

        ORG $0000                ; El programa se cargará en la RAM interna
OPTION EQU $39                    ; $1039 = Dirección del registro OPTION
ADCTL EQU $30                     ; $1030 = Dirección del registro ADCTL
PORTA EQU $00                    ; $1000 = Dirección del puerto A
ADR1 EQU $31                     ; $1031 = Dirección del registro ADR1

inicio
        LDX #$1000              ; Registro índice X := 1000
        LDAA #$80
        STAA OPTION,X          ; Activamos el conversor A/D
        LDAA #$34              ; Configuramos el conversor:
        STAA ADCTL,X           ; SCAN -> activo (mediciones continuadas)
                                ; MULT -> activo (múltiple canal, PE4-PE7)
espera BRCLR ADCTL,X $80 espera ; Esperamos a que termine la conversión
        LDAA ADR1,X            ; Leemos el resultado de la conversión
        CMPA #$30              ; Comparamos el resultado con el umbral
        BLO para               ; Si no hay objeto cerca, nos paramos
        LDAA #$18              ; Si lo encuentra, avanza hacia delante
        STAA PORTA,X
        BRA espera             ; Realiza la siguiente conversión

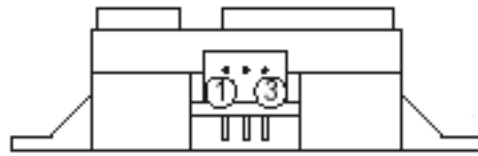
para   LDAA #$00

        STAA PORTA,X
        BRA espera

        END

```

Anexo: SHARP GP2D12



- ① V_O
- ② GND
- ③ V_{CC}

Tabla orientativa de conversión de valor digital a distancia

Valor	Distancia (mm)	Valor	Distancia (mm)	Valor	Distancia (mm)
0x85	80	0x4E	160	0x2D	300
0x7D	90	0x48	170	0x26	400
0x75	100	0x46	180	0x20	500
0x6F	110	0x43	190	0x1B	600
0x64	120	0x40	200	0x19	700
0x5C	130	0x3A	210	0x17	800
0x55	140	0x36	220	0x14	900
0x54	150	0x33	230	0x0C	1200
		0x32	240		
		0x2C	250		

