

# Encoders: Construcción, Montaje y Programación

Javier de Lope Asiaín  
Departamento de Sistemas Inteligentes Aplicados  
Universidad Politécnica de Madrid

<http://www.sia.eui.upm.es/~jdlope/>  
<mailto:jdlope@eui.upm.es>

15 de noviembre de 2000

## 1 Introducción

En el presente documento se describe cómo construir unos encoders para un robot móvil basado en una placa CT6811 de Microbótica (<http://www.microbotica.es/>). Además de los elementos hardware, también se introduce la programación del driver utilizando interrupciones.

Los usos de este tipo de dispositivos son diversos: hacer que el robot avance una serie de pasos y no un tiempo determinado, controlar la velocidad del robot, detectar posibles situaciones en las que el robot quede detenido ante un obstáculo, etc.

En cualquier caso, siempre se debe tener en cuenta que en robótica móvil se trabaja con una gran incertidumbre introducida por muchos factores (rugosidad de la superficie, peso del robot, características de los sensores, incluso errores en las lecturas) que puede hacer que se pierda la cuenta de los pasos.

Por tanto, este mecanismo no es demasiado apropiado para definir movimientos absolutos del robot en el entorno ya que los estos errores implicarían necesariamente imprecisiones en la estimación de su posición.

## 2 Los sensores

Para la construcción de los encoders vamos a utilizar dos sensores de infrarrojos CNY70 y dos discos con franjas blancas y negras alternas. Colocaremos cada par sensor-disco en cada una de las ruedas motrices del robot. De esta forma dividiremos (con más o menos exactitud dependiendo del número de franjas) una vuelta de rueda en una serie de intervalos o pasos.

Se han seleccionado los sensores de infrarrojos CNY70 por dos razones. En primer lugar, estos sensores son ideales en cuanto a relación calidad-precio para detectar cambios de blanco a negro en una distancia de milímetros. Como para el encoder utilizaremos un disco con franjas de estos colores, este tipo de sensores resultan apropiados.

Por otro lado, la tarjeta CT293+ de Microbótica puede controlar directamente cuatro de estos dispositivos. Como inicialmente los robots vienen equipados con dos sensores para el seguimiento de líneas, quedan otras dos entradas libres para controlar otros tantos sensores que utilizaremos en cada una de las ruedas motrices.

Para el montaje seguiremos las recomendaciones que se incluyen en el manual de la tarjeta CT293+. Básicamente basta con conectar el emisor, el cátodo y, conjuntamente, el ánodo y el colector con las tres patas de uno de los conectores acodados que quede libre en la tarjeta. Se remite al lector al manual de usuario de la tarjeta, página 14, para una información más detallada de las las conexiones a realizar.

Una vez realizadas las conexiones se recomienda usar el programa CT294 para probar las conexiones y verificar el pin al que hemos conectado el nuevo sensor.

### 3 Los discos

Los discos pueden construirse imprimiendo en un papel ligeramente grueso (el de 80 gramos que es el que se utiliza normalmente para fotocopiadoras e impresoras es suficiente) o en una cartulina las plantillas que se muestran en la figura 1. También se recomienda utilizar una impresora láser ya que con las de chorro de tinta o matriciales puede que no cubran bien toda la zona negra o que se tengan problemas en la detección del color negro con los sensores. No obstante, si se presentan estos inconvenientes, se puede repasar las zonas negras que no sean detectadas correctamente con un rotulador negro de punta gruesa.

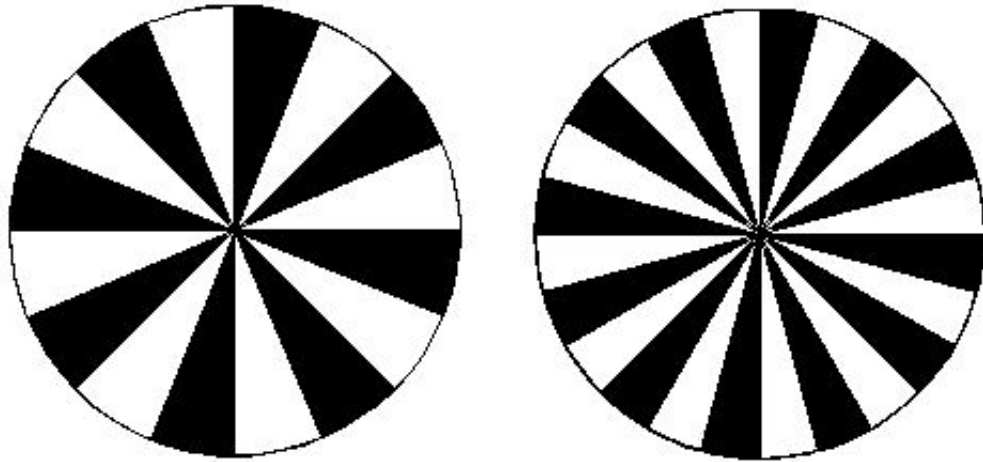


Figura 1: Discos para los encoders de 16 y 24 segmentos

El disco de la izquierda tiene 8 franjas negras y 12 el de la derecha. Normalmente no se habla del número de franjas negras sino de las totales (negras y blancas). Por tanto, el borde o perímetro del disco estará dividido en 16 y 24 segmentos, respectivamente.

Una vez que se han imprimido los discos, comprobado si se detectan bien los cambios de negro a blanco y viceversa y, en su caso, repasado con un rotulador negro, se cortarán con cuidado según el diámetro de las ruedas. El diámetro de los discos de las imágenes es aproximadamente de unos 5.5 centímetros. Si este diámetro es demasiado pequeño para las ruedas con las que se está trabajando, pueden construirse otros discos mayores con cualquier utilidad o programa gráfico.

## 4 Montaje

Los discos, una vez cortados, se deben pegar por la parte interna de las ruedas intentando que sus centros coincidan con los ejes de las ruedas. Para ello se recomienda desmontar la rueda y no cortar los discos.

Los sensores se pegarán en la parte inferior de la estructura del robot, bien con cinta aislante, bien de una forma más robusta. Como este paso es dependiente de la plataforma que se esté utilizando sólo se indica como recomendación que se aproxime lo máximo posible al disco sin que entre en contacto con él para mejorar la detección durante el uso.

Las conexiones de los sensores se podrán hacer en cualquier entrada que quede libre en la tarjeta CT293+ que están indicadas en la placa como **Sensor 1**, **Sensor 2**, **Sensor 3** y **Sensor 4**. La conexión a una entrada en particular hace que se conecte a un bit del puerto A del microcontrolador MC68HC11. Normalmente (verifíquese este punto en cada robot) las entradas de la CT293+ están asociadas con los bits del puerto A que se especifican en la tabla 1:

<i>Sensor i</i>	<i>Bit del Puerto A</i>
1	0
2	1
3	7
4	3

Tabla 1: Entradas sensoriales de la CT293+

Para la conexión, lo único que debemos tener en cuenta es que los bits 0, 1 y 2 del puerto A del microcontrolador MC68HC11 están asociados con tres *capturadores de entrada*, que nos podrán ser de gran utilidad a la hora de capturar las señales que llegan de los sensores. El bit 7 del puerto A también está asociado con otro recurso interno del microcontrolador: el *acumulador de pulsos*.

Este recurso es muy interesante para aplicaciones como la que estamos considerando. Como un acumulador de pulsos tiene asociado un contador que se va incrementando automáticamente con la llegada de nuevos pulsos, la cuenta de los pasos en el giro de las ruedas se haría de forma casi automática. Desafortunadamente, el MC68HC11 tiene un solo acumulador de pulsos por lo que en cualquier caso deberemos hacer uso de al menos un capturador. (*Nota*: Sin confirmar, existen modelos de MC68HC11 que disponen de más de un acumulador de pulsos.)

En nuestro caso, utilizaremos los capturadores de entrada ya que al disponer de sólo un acumulador, la programación de los capturadores hay que hacerla en cualquier caso. La programación del acumulador es muy sencilla (basta con configurar un par de registros internos del microcontrolador) y puede que se añada en un futuro en esta página.

## 5 Programación

Como se ha mencionado anteriormente, utilizaremos los capturadores de entrada para recibir los datos de los sensores. Concretamente, se utilizarán los capturadores de entrada IC3 e IC1, asociados con los bits 0 y 2 del puerto A, respectivamente. Como el código necesario es

similar para uno y otro capturador, sólo comentaremos el primer caso: el capturador IC3 (bit 0 del puerto A) que se corresponderá con el encoder de la rueda derecha según el montaje propuesto.

Un capturador de entrada genera una interrupción cada vez que detecta un flanco en la señal que tiene asociada. El flanco, que es configurable como ahora veremos, puede ser de subida, de bajada o ambos. En nuestro caso, el flanco se producirá siempre que el sensor detecte un cambio de blanco (nivel bajo) a negro (nivel alto).

La programación puede dividirse, conceptualmente hablando, en tres partes: Configuración del capturador de entrada, manejo de la interrupción y uso. Lo iremos viendo detenidamente.

### 5.1 Configuración del capturador

La configuración del capturador se realiza modificando tres registros internos del microcontrolador. El primero de ellos es el registro TMSK1 (\$1022). El bit 0 de este registro es el que está asociado con el capturador de entrada IC3. Escribiendo un '1'én este bit permitimos las interrupciones asociadas con eventos en los pines de este capturador de entrada. El resto de los bits corresponden con el resto de capturadores del microcontrolador. Entonces, quedaría según se muestra en la figura 2.

7	6	5	4	3	2	1	0
OC1I	OC2I	OC3I	OC4I	OC5I	IC1I	IC2I	IC3I
X	X	X	X	X	X	X	1

Figura 2: Registro TMSK1

Esto en ensamblador del MC68HC11 se corresponde con la instrucción:

```
LDX #$1000                ; Apunta a los registros de ctrl
BSET TMSK1,X $01          ; Permite interrupciones del IC3
```

El segundo registro a tener en cuenta es el TCTL2 (\$1021). Con los dos bits menos significativos de este registro indicamos qué tipo de flanco queremos utilizar. Los otros bits están asociados con los otros dos capturadores de entrada, quedando otros dos sin uso. En la figura 3 se muestra la configuración utilizada.

7	6	5	4	3	2	1	0
0	0	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A
X	X	X	X	X	X	0	1

Figura 3: Registro TCTL2

Nuevamente en ensamblador del MC68HC11 se corresponde con las instrucciones:

```
LDX #$1000                ; Apunta a los registros de ctrl
```

```

BCLR TCTL2,X $02           ; Flanco de subida del IC3
BSET TCTL2,X $01           ; Flanco de subida del IC3

```

Los bits, tomados dos a dos, de este registro pueden tener los valores recogidos en la tabla 2.

<i>Flanco</i>	<i>EDGxB</i>	<i>EDGxA</i>
Inhibidos	0	0
Subida	0	1
Bajada	1	0
Ambos	1	1

Tabla 2: Configuración del registro TCTL2

Entonces, según estas tablas, estamos configurando el capturador de entrada IC3 en flanco de subida, por lo que, resumiendo, cada vez que el microcontrolador detecte un flanco de subida en su pin PA0 proveniente de un cambio provocado al pasar de un segmento blanco a otro negro y detectado por el sensor, se generará la interrupción asociada con el capturador de entarda IC3.

A continuación veremos cómo capturar esta interrupción y qué hacer cuando se produce.

## 5.2 Manejo de la interrupción

Una de las cosas que tenemos que hacer al estar trabajando con interrupciones es indicar la micro que éstas están permitidas. Esto se hace mediante la instrucción CLI en la parte de inicialización de nuestro programa.

```

CLI                          ; Permite interrupciones

```

Lo que nos queda, ya para el caso que estamos tratando, es reservar espacio para una variable en la que llevaremos la cuenta de los pasos que ha dado el motor, escribir la rutina de tratamiento de la interrupción y asociar dicha rutina con la interrupción.

Definimos la variable `pasos_d` de 16 bits con lo que podemos contar hasta 65535 pasos, si son necesarios más, puede manejarse el desbordamiento y utilizar una segunda variable para ampliar el contador. Esto lo hacemos mediante:

```

pasos_d FDB 0                ; Variable para contar los pasos

```

La rutina de tratamiento de la interrupción en este caso es bastante sencilla. Cada vez que se produzca una interrupción incrementaremos la variable `pasos_d`, borraremos el flag que indica que no se ha tratado la interrupción del IC3 e finalizaremos con una instrucción de retorno de interrupción.

El flag de la interrupción del IC3 está se encuentra en el registro TFLG1 junto con los flags de los otros capturadores. El del IC3 ocupa el bit menos significativo. Estos bits se activan a nivel bajo (se pone a '0') cada vez que se produce una interrupción. Una vez tratada, debemos borrar el flag (escribir un '1') desde la rutina de tratamiento para indicar que la interrupción ha sido tratada, en caso contrario se anidarían las interrupciones.

```

h_enc_d LDD pasos_d           ; Carga el número de pasos
        ADDD #$01            ; Incrementa en 1
        STD pasos_d          ; Almacena el número de pasos
        LDAA #$01            ; Borra el flag de interrupción IC3
        STAA TFLG1,X
        RTI                  ; Retorno de interrupción

```

Por último, nos queda indicar dónde se encuentra el manejador de la interrupción. El vector de dirección es dependiente del modo en el que esté operando el microcontrolador. Para los modos en los que tengamos los vectores de interrupción en la RAM interna (sin ampliaciones de memoria), esto puede indicarse de la siguiente forma:

```

        ORG $00E2            ; Instala el manejador
        JMP h_enc_d

```

### 5.3 Ejemplo de utilización

Como ejemplo de utilización se incluye un programa que hace que el robot avance mientras consulta la variable `pasos_d` dentro de un bucle. Cuando dicha variable alcanza un determinado valor (8 en este caso, es decir, una vuelta completa de la rueda), sale del bucle y detiene al robot. El programa se muestra completo, con la configuración del captador (subrutina `init`) y rutina de tratamiento (subrutina `h_enc_d` como vimos anteriormente).

```

* -----
* encoder.asm
* -----

PORTA   EQU $0              ; Puerto A
TCTL2   EQU $21             ; Activaciones de flancos ICn
TMSK1   EQU $22             ; Interrupciones de captadores
TFLG1   EQU $23             ; Flags de captadores

        ORG $0000           ; Programa en RAM interna

        BRA inicio

pasos_d FDB 0                ; Variable para contar los pasos

inicio  LDX #$1000           ; Apunta a los registros de ctrl
        BSR init            ; Inicializa interrupciones

avanza  LDAA #$18            ; Avanza
        STAA PORTA,X

bucle   LDD pasos_d          ; Carga los pasos de la rueda dcha
        CPD #8              ; Lo compara con 8
        BNE bucle           ; Si no se ha lledado, continua
        CLRA                ; Detiene el robot

```

```

                STAA PORTA,X

stop    BRA stop                ; Y se para

h_enc_d LDD pasos_d            ; Carga el número de pasos
        ADDD #$01              ; Incrementa en 1
        STD pasos_d            ; Almacena el número de pasos
        LDAA #$01              ; Borra el flag de interrupción IC3
        STAA TFLG1,X
        RTI                    ; Retorno de interrupción

init    BCLR TCTL2,X $02       ; Flanco de subida del IC3
        BSET TCTL2,X $01       ; Flanco de subida del IC3
        BSET TMSK1,X $01       ; Permite interrupciones del IC3
        CLI                    ; Permite interrupciones
        RTS

                ORG $00E2       ; Instala el manejador
        JMP h_enc_d

                END

```

## 6 Comentarios adicionales

Se han utilizado los flancos de subida con lo que estamos detectando cambios de blanco a negro en el sensor. También podríamos configurar el capturador con flancos de bajada con lo que las interrupciones se producirían con cambios de negro a blanco. Estas dos configuraciones son equivalentes.

También podemos configurar el capturador tanto para flancos de subida como de bajada con lo que recibiremos señales tanto en los cambios de blanco a negro como en los de negro a blanco. De esta forma conseguimos duplicar la resolución del encoder (dividimos cada vuelta de rueda en 16 ó 24 pasos utilizando los discos disponibles).

También se pueden utilizar discos con un mayor número de segmentos pero eso depende del campo de visión de los sensores utilizados. En el caso de los CNY70, éste es muy amplio y dado el diámetro de los discos no se detectaban bien los cambios. El de 24 segmentos es el mayor que se puede utilizar en estas condiciones sin que se produzcan pérdidas en las detecciones.

La instalación definitiva se ha realizado con los discos de 16 segmentos. Se ha preferido perder algo de resolución para mantener la cuenta en base 2. Con 16 segmentos podemos utilizar 3 ó 4 bits (con uno o ambos flancos, respectivamente) para llevar la cuenta de los pasos intermedios y utilizar el bit 4 y 5, respectivamente, para llevar la cuenta de vueltas de rueda completas. Nótese que con 24 segmentos esto no hubiese sido posible.

## Referencias

- [1] C. Doblado, A. Prieto-Moreno, J.J. San Martín, and J. González. *Microcontrolador MC68HC11: Fundamentos, recursos y programación*. Microbótica, Madrid, 1999.
- [2] J. Jones and A. Flynn. *Mobile Robots. Inspiration to implementation*. AK Peters, Wellesley, Massachusetts, 1993.
- [3] Microbótica, Madrid. *Tarjeta CT293+: Manual de usuario*, 1999.
- [4] Microbótica, Madrid. *Tarjeta de desarrollo CT6811: Manual de usuario*, 1999.