

## CONTROL DE MOTORES PASO A PASO

Por G.M. Bosyk, A.R. Martínez y D. Martínez Heimann

Docente: Dr. Cesar Moreno  
Laboratorio de Electrónica  
Facultad de Ciencias Exactas y Naturales  
Universidad de Buenos Aires  
Buenos Aires – Argentina

En el presente informe se describe una experiencia llevada a cabo para controlar motores paso a paso mediante el puerto paralelo de una PC. En primer lugar, se presentan las características básicas de los motores paso a paso y del puerto paralelo de una computadora. Luego se detallan los procedimientos realizados, comenzando por una descripción del entorno de control utilizado. A continuación, se detalla el circuito diseñado y construido para la interfaz de control. Por último, se describen alternativas propuestas para conseguir que el circuito tome decisiones a partir de la lectura de alguna variable externa como puede ser la intensidad de luz sobre una fotorresistencia.

### 1 – INTRODUCCION [1], [2]

El puerto paralelo de una PC, desde el punto de vista del Hardware, es un conector DB25 como se muestra en la figura 1.

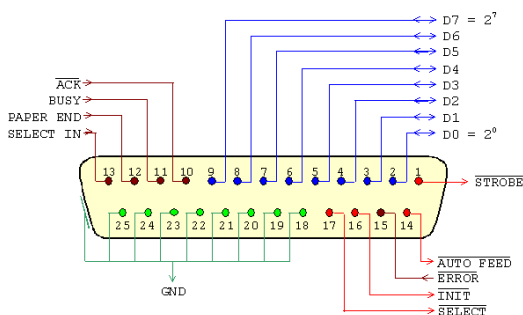


Figura 1: Descripción física del conector DB25 del puerto paralelo.

En este conector:

- 8 líneas (pines) son para salida de datos (registro de DATOS). Sus valores son únicamente modificables a través de software, y van del pin 2 (dato 0, D0) al pin 9 (dato 7, D7).

- 5 líneas son de entrada de datos (registro de ESTADO), únicamente modificables a través del hardware externo. Estos pines son: 11, 10, 12, 13 y 15, del más al menos significativo.
- 4 líneas son de control (registro de CONTROL), numerados del más significativo al menos: 17, 16, 14 y 1. Habitualmente son salidas, aunque se pueden utilizar también como entradas y, por tanto, se pueden modificar tanto por software como por hardware.
- las líneas de la 18 a la 25 son la tierra.

Originalmente fue construido para establecer la comunicación de la PC con la impresora, pero debido a su gran utilidad y facilidad de programación se lo utiliza para comunicar otros periféricos.

Los motores paso a paso, como todo motor, son en esencia un convertidor electromecánico, que transforma la energía eléctrica en mecánica; pero de un modo tan peculiar que constituye en la actualidad una categoría

aparte. En efecto, mientras que un motor convencional gira libremente al aplicar una tensión comprendida dentro de ciertos límites, el motor paso a paso está concebido de tal manera que gira un determinado ángulo proporcional a la "codificación" de tensiones aplicadas a sus entradas.

Existen tres tipos de motores paso a paso:

- de imanes permanentes
- de reluctancia variable
- híbridos.

Los primeros, de *imanes permanentes*, están básicamente constituidos por un rotor sobre el que van aplicados distintos imanes permanentes, y por un cierto número (generalmente cuatro) de bobinas excitadoras bobinadas en su estator. El rotor, que posee una polarización magnética constante, gira para orientar sus polos de acuerdo al campo magnético creado por las fases del estator. Toda la conmutación (excitación de las bobinas) debe ser externamente manejada por un controlador y, habitualmente, los motores y controladores están diseñados para que el motor pueda ser mantenido en una posición o rotar en uno u otro sentido.

En los motores de *reluctancia variable*, el rotor está formado por un material ferromagnético que tiende a orientarse de modo que facilite el camino de las líneas de fuerza del campo magnético generado por las bobinas de estator. No contiene, por tanto, imanes permanentes. Los motores *híbridos* combinan el modo de funcionamiento de los dos anteriores.

En los motores paso a paso la señal de control consiste en trenes de pulsos que van actuando rotativamente sobre una serie de electroimanes dispuestos en el estator. Por cada pulso recibido, el rotor del motor gira un determinado número discreto de grados. Para conseguir el giro del rotor en un determinado número de grados, las bobinas del estator deben ser excitadas secuencialmente a una frecuencia que determina la velocidad de giro. Dentro de los motores paso a paso de

imán permanente existen dos tipos, bipolares y unipolares. Estos últimos suelen tener 8, 6 o 5 cables de salida, dependiendo de su conexionado interno. Este tipo se caracteriza por ser más simple de controlar. Un esquema sencillo de este tipo de motores se muestra en la figura 2, para el caso de un motor unipolar de 6 cables.

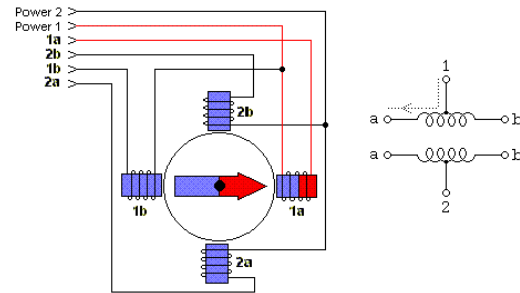


Figura 2 Esquema conceptual de un motor paso a paso unipolar con 6 cables de salida.

Para la realización del proyecto se utilizaron dos motores paso a paso de imán permanente unipolares. La mayoría de estos motores pueden ser manejados a frecuencias de audio permitiendo un giro rápido y, con un controlador apropiado, pueden ser arrancados y parados en posiciones controladas.

Los motores paso a paso son ideales para la construcción de mecanismos en donde se requieren movimientos muy precisos. La principal característica de estos motores es el hecho de poder moverlos un paso por cada pulso que se le aplique. Este paso puede variar desde  $90^\circ$  hasta pequeños movimientos de tan solo  $1.8^\circ$  (e incluso hasta de  $0.72^\circ$ ), es decir, que se para realizar un giro completo ( $360^\circ$ ) se necesitarán 4 pasos en el primer caso y 200 para el segundo.

## 2 – DESARROLLO EXPERIMENTAL

### 2.1 El software de control

El software para el control de los motores paso a paso fue desarrollado en *Smalltalk* en el ambiente Squeak, utilizando el paquete de control de puerto paralelo provisto por Germán Viscuso en [www.smalltalking.net](http://www.smalltalking.net). Este paquete interactúa con la Input/Output dll, compilación de la librería “io.h” de C++, que resuelve la interacción de más bajo nivel con el puerto.

En *Smalltalk* todo concepto del dominio estudiado, en este caso el de los motores paso a paso, debe ser modelado como un objeto. El comportamiento de todo objeto debe estar definido por una clase. Por ello, el paquete utilizado provee una clase de nombre *ParallelPort* que define las responsabilidades del puerto y se encarga de interactuar con la I/O dll. El diagrama de la figura 3 muestra las utilidades del puerto paralelo en este caso. Se observan en la figura los mensajes que un puerto paralelo puede recibir: *setBit:* y *clearBit:*, que dado un número de pin de salida lo colocan en 1 y 0 respectivamente; *setBits:*, que dada una secuencia de ocho bits, configura según corresponda los ocho pines de salida del puerto; *getInput:*, que dado uno de los cinco pines de entrada, retorna su valor binario.

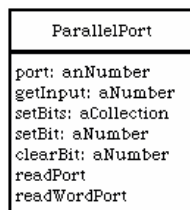


Figura 3: Responsabilidades del puerto paralelo.

Otro objeto del dominio de los motores paso a paso es el que modela una posible secuencia de pasos. El diagrama de la clase que define su comportamiento se observa en la figura 4. Sus elementos son cada uno de

los pasos que la compone y están ordenados. Existe un elemento actual. La secuencia sabe dar un paso hacia el próximo elemento o hacia el anterior (esto permite eventualmente que los motores puedan girar en sentido inverso).

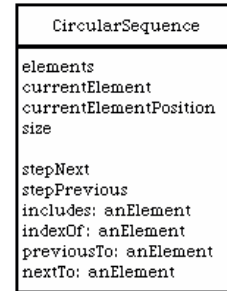


Figura 4: Responsabilidades de una secuencia circular.

Debido a que el motor puede trabajar con tres secuencias posibles, un objeto, que denominaremos su secuenciador, es encargado coordinar el cambio de secuencia. En la figura 5 se observa su diagrama de clase.

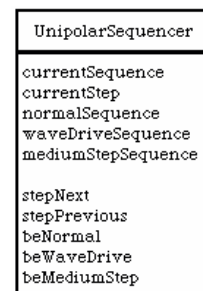


Figura 5: La clase que define el secuenciador unipolar.

Por otra parte, el concepto más importante del recorte de la realidad estudiado tiene también en este programa su clase. Su diagrama está en la figura 6 y lleva el nombre de *StepMotor* (motor de paso). Como se observa, tiene una dirección, un secuenciador y un tiempo de espera entre cada paso (*stepDelay*). Es posible aumentar su velocidad, disminuirla, ir en un sentido y su inverso, modificar la dirección de movimiento y detenerlo.

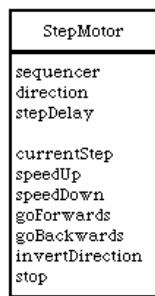


Figura 6: La clase StepMotor.

Finalmente, el encargado de coordinar el trabajo de los objetos enunciados es el controlador del programa. No es un objeto de la realidad estudiada sino del dominio computacional. Su clase se diagrama en la figura 7.

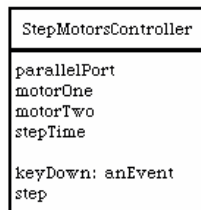
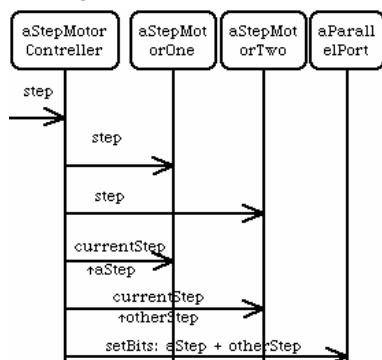


Figura 7: El StepMotorsController.

El controlador recibe el mensaje *step* cada un cierto tiempo, definido en milisegundos en su variable *stepTime*. Como se observa en el diagrama de la figura 8, envía un mensaje homónimo a ambos motores, luego les pide sus pasos actuales (dos secuencias de cuatro bits) y con ellos arma los ocho valores que luego envía al puerto para que todos sus pines de salida tengan sus nuevos valores.

Figura 8: Diagrama de colaboraciones que desencadena la recepción del mensaje *step* por parte del controlador.

En la figura 9 se observa cómo es la relación de conocimiento de los objetos cuyos comportamientos (a excepción de *direction* y *step*) son definidos en las clases anteriores.

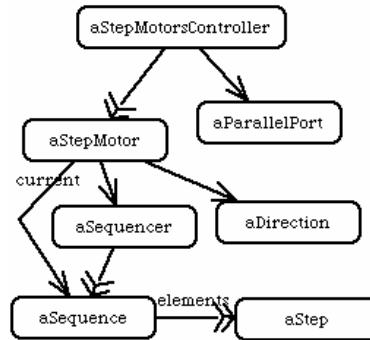


Figura 9: Diagrama de instancias del programa. Se muestra el diagrama para un único motor

En resumen, los elementos de la secuencia son sus pasos. El secuenciador conoce varias secuencias y el motor una sola como su actual. El controlador conoce varios motores (dos precisamente, lo que no se encuentra reflejado en la figura 9) y un puerto paralelo.

## 2.2 Interfaz de control

Debido a que el puerto paralelo trabaja como máximo a 5V y 20mA, no es útil ni recomendable conectar directamente un motor paso a paso debido a la baja potencia de salida. Para solucionar este inconveniente se construyó una interfaz de control, encargada de aumentar la potencia del puerto paralelo. La misma está basada en el circuito integrado 74LS244, un buffer de 8-bit, 500mA y entrada TTL. El circuito de la interfaz se observa en la figura 10.

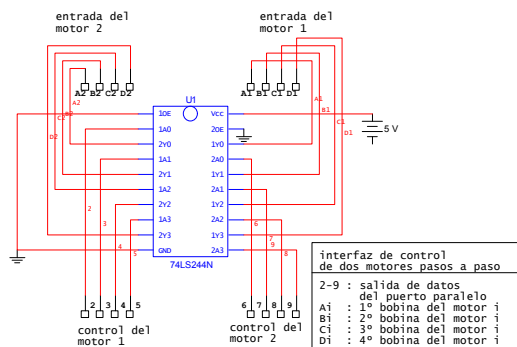


Figura 10: Descripción física del conector.

A través de los cuatro bits menos significativos de la línea de datos del puerto paralelo se controla el motor 1 y por medio de los cuatro bits más significativos se controla el motor 2. Se programaron en lenguaje *Smalltalk* tres tipos de secuencias para controlar a los motores, las cuales se describen en la tabla 1.

PASO	SECUENCIA 1 (NORMAL)	SECUENCIA 2 (WAVE DRIVE)	SECUENCIA 3 (MEDIO PASO)
	A B C D	A B C D	A B C D
1	1 1 0 0	1 0 0 0	1 0 0 0
2	0 1 1 0	0 1 0 0	1 1 0 0
3	0 0 1 1	0 0 1 0	0 1 0 0
4	1 0 0 1	0 0 0 1	0 1 1 0
5			0 0 1 0
6			0 0 1 1
7			0 0 0 1
8			1 0 0 1

Tabla 1: Secuencias de bits para el control de un motor P-P unipolar (A, B, C y D bobinas).

Las secuencias 1 y 2 son de paso doble. La primera es la denominada secuencia normal y es la recomendada por el fabricante. Debido a que por cada paso hay un par de bobinas activas se obtiene un alto par de paso y retención. La secuencia 2 (*wave drive*) generalmente brinda un funcionamiento más suave. La contrapartida es que al estar una sola bobina activa el par de paso y retención es menor. Combinando las secuencias 1 y 2, se obtiene una de medio paso, que consta de ocho tiempos en vez de cuatro.

Mediante el uso del programa desarrollado se logró manejar satisfactoriamente los motores paso a paso en lo respectivo a su velocidad y dirección de giro, utilizando la secuencia normal y controlando las salidas del puerto paralelo mediante el teclado de la computadora.

## 2.3 Lectura de una variable externa

Una vez logrado el manejo de los motores mediante la salida del puerto paralelo, se intentó utilizar las entradas del mismo, con el objetivo de conseguir un circuito que tomase decisiones de comportamiento en base a estímulos externos.

En primer lugar se buscó desarrollar un sensor tal que al recibir una intensidad de luz determinada enviase una señal analógica de 5V (1 lógico) para luego ser adquirida por la línea de estado del puerto paralelo y así decidir cambios en el comportamiento del motor. Para ello, se pensó en utilizar una fotorresistencia conectada a un comparador Schmitt-Trigger como se muestra en la figura 11.

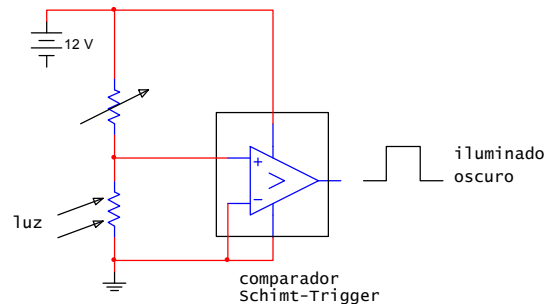


Figura 11: Esquema del sensor de luz.

El objetivo de este dispositivo era obtener una señal analógica alta cuando sobre la fotorresistencia incidiera una intensidad de luz predeterminada. La resistencia variable se incluyó para calibrar el disparo del comparador. No se tuvo éxito al implementar este circuito ya que la calibración resultaba muy sensible y el Schmitt-Trigger quedaba siempre disparado en un nivel alto. Luego, se pensó en otra alternativa más fácil de calibrar

y que permitiera tener mayor sensibilidad que dos niveles (oscuro/iluminado del caso anterior). Esta alternativa consiste en construir un transductor de luz y adquirir la señal del mismo de la siguiente forma: a una fuente de corriente de 1 mA se conecta una fotorresistencia la cual, dependiendo de la intensidad de luz sobre la superficie de la fotorresistencia, entrega una tensión que varía de  $\sim 0.1V$  (máxima oscuridad) a  $\sim 10V$  (máxima luz); luego se transforma ésta señal analógica en digital por medio de un conversor ADC804 de 8 bits; por último, se adquiere la señal digitalizada por las líneas de estado del puerto paralelo. Esto no se pudo implementar por falta de tiempo, pero en la figura 12 se muestra un esquema de lo propuesto:

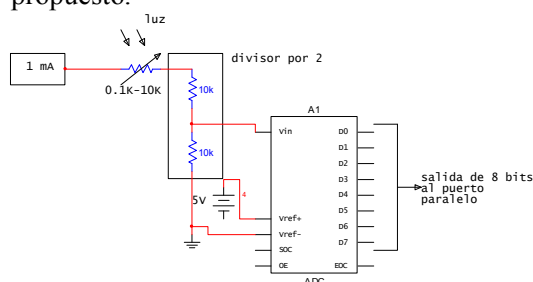


Figura 12: Trasductor de luz.

En la práctica, a fin de verificar la posibilidad de controlar el comportamiento de los motores mediante las entradas del puerto paralelo, se conectaron dos líneas de estado a dos llaves de 5 V regulados. Se consiguió que a partir de éstas lecturas los motores invirtieran su sentido si se leía un 1 (nivel alto) sobre la línea de estado correspondiente.

### 3 – CONCLUSIONES

Fundamentalmente, se logró cumplir con éxito la parte principal del proyecto consistente en el control de motores paso a paso mediante el puerto paralelo de una PC.

En particular, se construyó y diseñó una interfaz de control con su respectivo software capaz de controlar de forma independiente dos motores paso a paso unipolares de imán permanente. Además, se logró que los motores invirtieran sus sentidos a partir de la lectura de una variable externa, en este caso, el encendido/apagado de una fuente de 5 V regulada.

Se intentó implementar un mecanismo de respuesta a estímulos externos (sensibilidad a luz) mediante el uso de fotorresistencias, el cual no logró ser completado debido a la alta sensibilidad de los componentes disponibles. Finalmente, se propuso una alternativa a este respecto basada en el uso de un conversor analógico-digital que permita digitalizar la respuesta a fin de ser analizada mediante el software de control.

La elección de este proyecto se basó en principio en el alto interés que presenta el uso de motores paso a paso en muchas áreas relacionadas con la tecnología (disqueteras, impresoras, relojes, etc.). Por otra parte, dentro del grupo de trabajo existe un interés particular por las posibilidades que presentan estos dispositivos en el área de diseño y construcción de robots. En particular, se pensó como proyecto a futuro, el diseño y construcción de robots que puedan decidir su movimiento en base a señales luminosas que reciban mediante sensores ubicados en posiciones apropiadas en sus armazones.

### 4 – REFERENCIAS

- [1] “Control automatizado y robótica para el área de tecnología”. Disponible en [http://www.cpr2valladolid.com/tecno/cyr\\_01](http://www.cpr2valladolid.com/tecno/cyr_01).
- [2] “Unipolar stepper motors and control”. Disponible en <http://www.stepperworld.com>.