Servomotores y su control

Uso y aplicaciones de los microbots

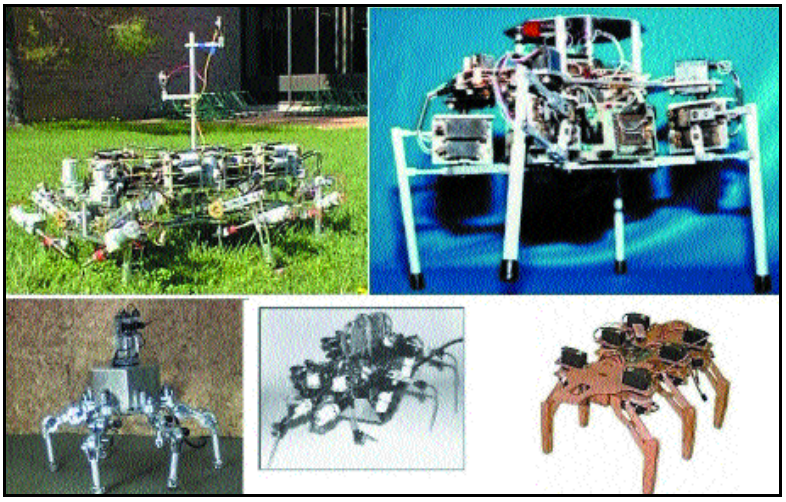
Este mes comenzamos la andadura de la microbótica. Los microbots pueden emplearse desde el simple entretenimiento de aficionados hasta aplicaciones de neurofisiología, cibernética, estudios del comportamiento para evitar obstáculos, vehículos espaciales, etc.

Mucha gente se pregunta si consistirá el próximo hito de la robótica en una serie de máquinas económicas que repten, piensen y se conduzcan como insectos. Así opinan los investigadores del «Laboratorio de insectos» del Instituto de Tecnología de Massachusetts. En este laboratorio se han creado pequeños robots con movimientos y comportamiento de insectos. Sus diseñadores los han bautizado con el nombre de insectoides.

Rodney Brooks, director del laboratorio de insectoides del MIT realiza un nuevo enfoque que él denomina arquitectura de subsunción. Con este enfoque no se añaden al sistema conductas nuevas hasta que las anteriores están perfectamente a punto y en marcha. En la arquitectura de subsunción las conductas complejas son el resultado evolutivo de una variedad de funciones de conducta simples.

Uno de estos robots es Genghis, que entre los sensores de que dispone está dotado de dos barbas de gato y seis sensores de infrarrojos. Los sensores de infrarrojos operan conjuntamente con el módulo de merodeo, dotando a Genghis de una conducta insólita. En esta modalidad Genghis descansa tranquilamente hasta que detecta radiación infrarroja, procedente por ejemplo de un tobillo humano próximo. Cuando esto se produce se activa su módulo de locomoción y en ese momento comienza a avanzar hacia la pierna del desprevenido sujeto. Claro está, la persona tiene tiempo para apartarse, pero mediante el módulo de guiado Genghis puede ser implacable.

Se ha comprobado que sistemas de control muy sencillos pueden dar lugar a comportamientos muy complejos. Se demuestra que la técnica de subsunción sigue funcionando al añadir estratos de comportamiento más y más complejos.



«Foto 1». Diferentes robots con movimientos similares a insectos.

Hexápodos

Una vez nos hemos introducido en el tema, comencemos por el principio. El primer robot que se va ha describir es un hexápodo. Hagamos un poco de historia. El microbot llamado HexWalker (foto 2) fue creado originalmente por los miembros de la sociedad R.S.S.C. (*The Robotics Society of South California*) como un *kit* asequible para iniciarse en la construcción de robots.

Estaba inspirado en un artículo de Gary Malolepsy's aparecido en la revista «The Robot Builder» en el número de febrero-marzo de 1994. El *kit* para la construcción del HexWalker apareció en enero de 1995 fabricado y distribuido por M&T Systems. Después de tres años de febril actividad en la fabrica y de cientos de imitaciones el *kit* dejó de fabricarse en M&T Systems.

Según se lee en su página web, «*las líneas de producción están ahora en silencio, los trabajadores han emigrados a verdes praderas, quedan sólo lejanos ecos de la actividad*».

El lector no debe preocuparse por este motivo, ya que se ha localizado un *kit* similar en Inglaterra, el StampBug (foto 3), fabricado por la empresa Milford Instruments (no muy barato que digamos), que es el que se va a detallar.

Pero además, se darán en este y posteriores artículos suficientes explicaciones para que el lector pueda construir su propio hexápodo sin necesidad de adquirir el *kit*, ampliando incluso sus prestaciones y el número de sensores.

Servomotores

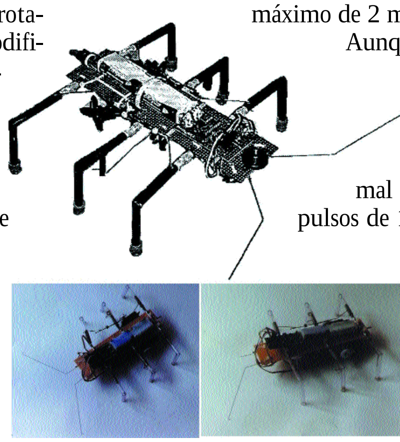
La fuerza que moverá nuestro hexápodo la proporcionarán tres servomotores iguales a los que se emplean en modelismo y radio control para hacer girar la dirección de los coches o mover los timones de los aviones. Además, hemos de dar las gracias al modelismo, que nos permite disponer de este tipo de motores a un precio relativamente asequible.

Un servomotor (foto 4) es un dispositivo en forma de caja negra al que llegan tres cables. Contiene un pequeño motor, una caja de engranajes, un potenciómetro de un

valor aproximado de 5K y un pequeño circuito integrado.

Este motor eléctrico en miniatura ataca a la magnitud que se ha de controlar: el giro y posicionamiento del eje del motor. A su vez, el movimiento de rotación angular del motor modifica la posición del potenciómetro interno, que controla un monoestable también integrado en el servomotor.

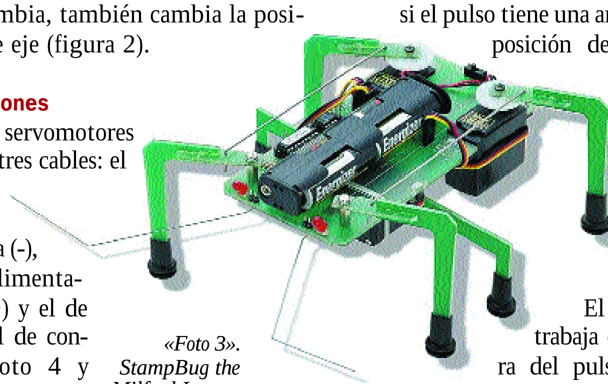
El eje del motor puede ser girado hasta una posición angular específica mediante una señal de control. Mientras se mantenga esta señal de control, el servomotor mantendrá la posición angular del eje. Si la señal de control cambia, también cambia la posición de eje (figura 2).



«Foto 2». El HexWalker orginal.

Conexiones

Los servomotores tienen tres cables: el de masa (-), el de alimentación (+) y el de la señal de control (foto 4 y figura 1). El positivo se conecta a + 5 y el de señal de control a una fuente de pulsos variables entre 1 y 2 milisegundos de duración (figura 3) que se repiten con una frecuencia de unos 12-20 ms. Los cables de los «servos» siguen casi siempre el mismo código de colores; por ejemplo en los Futaba el color rojo (V+), negro (masa) y blanco (señal de control). Los fabricantes JR y Graupner colocan el cable de la señal de control de color naranja, mientras que algunos «servos» Sanwa tienen el cable de masa de color azul.



«Foto 3». StampBug the Milford Instruments.

Funcionamiento

La velocidad del motor, así como la dirección del movimiento de los «servos» se controla mediante servopulsos modulados en amplitud. En la «Figura 2» se muestra la forma de estos servopulsos. El servomotor convierte los servopulsos en un movimiento mecánico. La magnitud del giro del eje del «servo» es proporcional a la anchura del pulso que llega por la línea de control. Este tipo de pulsos está formado por una señal digital que se genera aproximadamente cada 20 milisegundos. La anchura de estos pulsos va de un mínimo de 1 ms a un máximo de 2 ms.

Aunque la relación anchura del pulso y la posición del eje no está estandarizada, lo normal es que trenes de pulsos de 1,5 ms lleven el eje del servo al centro de su rango, anchura neutra. Si la anchura del pulso es de 1 ms, el servomotor se posiciona en el extremo izquierdo, mientras que si el pulso tiene una anchura de 2 ms la posición del «servo» es el extremo opuesto (figura 2). Esta técnica se conoce como modulación por anchura de pulso, en ingles PWM (*Pulse Width Modulation*).

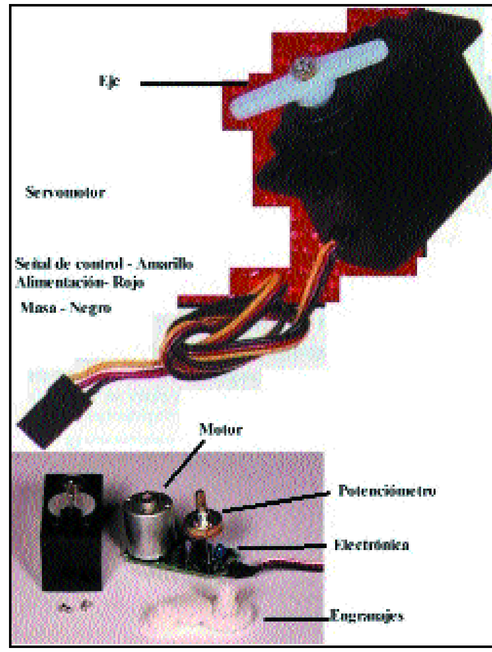
El servomotor (figura 3) trabaja comparando la anchura del pulso de entrada con la anchura del pulso producido por el *timer* interno. A su vez, el período del *timer* interno es controlado por el potenciómetro acoplado al eje del «servo». La diferencia entre la anchura del pulso de entrada y la anchura del pulso interno se utiliza como señal de error. La lógica del «servo» se encarga de determinar la dirección en la que ha de girar el motor para minimizar dicho error. Para ello activa los *drivers* de salida apropiados. El motor girará modificando la posición del potenciómetro de retroalimentación.

Cuando llega el siguiente pulso se vuelve a realizar la comparación, comprobando de forma continua la posición del eje y realizando también constantemente las correcciones necesarias en la posición del mismo.

Como se ha podido apreciar, se trata de un bucle de retroalimentación negativa. Si la posición del potenciómetro no se iguala con la posición deseada del eje, el motor se moverá hacia adelante o hacia atrás, hasta que la posición del potenciómetro sea equivalente a la posición deseada del eje. En este momento la corriente del motor se apaga.

La precisión al posicionarse depende tanto de la precisión del potenciómetro como de la precisión de la anchura de los pulsos que llegan al motor. La mayoría de los modelos de servomotores consiguen una resolución de 0,5 grados.

Cuando se reduce la señal de error a un nivel aceptable, el eje del «servo» se encuentra en la posición correcta. En ese

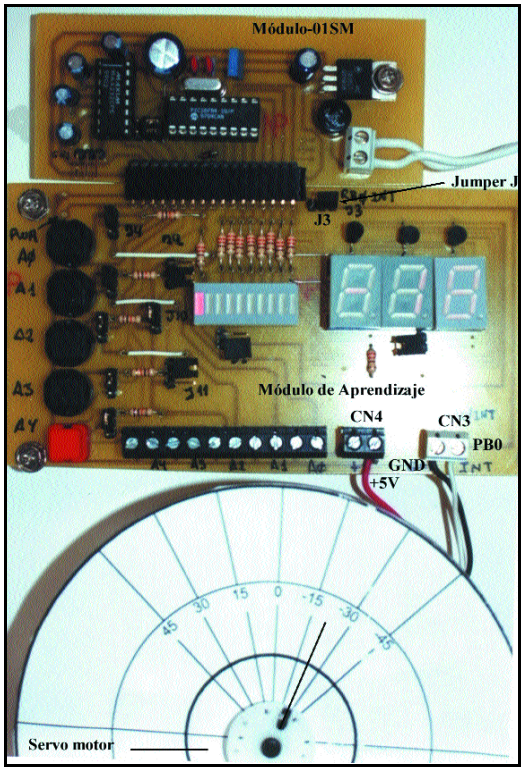


. *«Foto 4». Servomotor y sus mecanismos internos.*

PRIMERA PARTE

```
'*********************************************************************
' Programa: [illegible]         v1.0        [illegible] 1999
' [illegible]
' [illegible]
'*********************************************************************
'
'     [illegible]
'
'     Programa que controla un servo a través de la patita PORTD.0
'     Pulsando el botón A0 lo desplaza hacia la izquierda.
'     Pulsando el botón A4 lo desplaza hacia la derecha.
'     Cada pulsación desplazará el servo 1 grado aproximadamente.
'
'
'     [illegible]
'
'     ENTRADAS:   [illegible]
'                 [illegible]
'     SALIDAS:    PORTD.0         control del servo
'
'*********************************************************************

'*********************************************************************
'CONSTANTES
'*********************************************************************
Symbol SERVO = 0          'pin 0 del puerto para el control del servo
Symbol [illegible]
Symbol [illegible]
Symbol [illegible]
Symbol [illegible]
Symbol [illegible]
Symbol [illegible]
Symbol [illegible]
Symbol [illegible]        'botón para desplazamiento izquierda
symbol      signo = b4    'variable que contiene el signo del display
symbol      MSD = b5      'MSD del display
Symbol      LSD = b6      'LSD del display
Symbol      [illegible]
Symbol      [illegible]
Symbol      [illegible]
Symbol      [illegible]   'salida al puerto D para mostrar 3
symbol      cuatro = [illegible]  'salida al puerto D para mostrar 4
symbol      cinco = [illegible]   'salida al puerto D para mostrar 5
Symbol      seis = [illegible]    'salida al puerto D para mostrar 6
Symbol      [illegible]
Symbol      [illegible]
Symbol      [illegible]
Symbol [illegible]
10
                          'microsegundos del pulso que controla el servo.
Symbol posicion = b2      'contiene el valor en grados de la posición del
servo
                          '[illegible]
                          '
                          '     [illegible]
                          '     [illegible]
                          '     +45 grados = [illegible]
```



«Foto 5». Módulo 0-1 y el de aprendizaje.

momento la señal de error suele ser de unos 5µs, diferencia entre el ancho del pulso de la señal de entrada y el ancho del pulso de la señal interna. Esto se corresponde con una fracción de grado del recorrido del servomotor. Al ser el cero demasiado crítico, cuando el error está en este rango, conocido como zona muerta o *guard band*, el «servo» apaga los *drivers* del motor.

Si la señal de error no está por debajo de estos 5µs, la electrónica interna continuará intentando cancelar el minúsculo error, haciendo girar el motor atrás o adelante en un movimiento conocido como *hunting*. La electrónica interna tiene como misión mantener la anchura de los pulsos del monostable interno igual a la anchura de los pulsos de entrada.

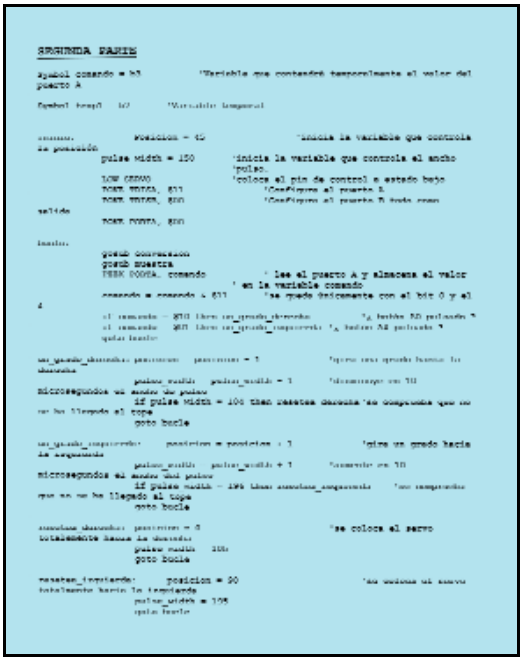
Debido a que hay una relación fija entre el ángulo de rotación del potenciómetro y la anchura del pulso interno, la magnitud de rotación del «servo» se puede controlar directamente con la anchura de los pulsos aplicados (figura 3). En conclusión, el circuito electrónico integrado en el motor convierte la anchura del pulso de entrada en una posición determinada del eje de salida.

Ventajas

Entre las ventajas que aporta el empleo de un «servo» están las siguientes: poco peso, alta potencia (par de fuerza), fiabilidad, fortaleza (los «servos» y su electrónica normalmente sobreviven a choques y funcionan en ambientes de alta temperatura, suciedad, humedad y vibraciones), simplicidad, versatilidad y bajo coste.

En las tiendas de modelismo pueden encontrarse muchos tipos de servomotores de las casas Futaba, FMA, Multiplex, Sanwa, etc. con par de fuerza que va desde unos 3 Kg/cm y una velocidad de giro de unos 0,22 segundos como el Futaba FPS3003, hasta el FPS9402 con un par de 8 Kg, una velocidad de giro de 0,1 segundos y engranajes metálicos. Aunque para el aprendizaje es suficiente un «servo» de los más baratos, los mejores tienen engranajes metálicos y mayor velocidad de giro, que junto al par de fuerzas determinan su precio.

Dando un repaso a las revistas de radiocontrol, se puede constatar que en los comercios españoles dedicados a estos menesteres los «servos» más económicos tienen un precio que ronda las 2.500-3.000 pesetas, como los servos Futaba FPS-3003 o los FMA300, con un par de unos 3,5 Kg y una velocidad de giro de unos 0,23 segundos. Podéis echar



Las bondades de C

Hoy en día el uso del lenguaje C se ha extendido en la programación de sistemas hardware debido a que, a pesar de ser un lenguaje de alto nivel, está relativamente cerca del lenguaje ensamblador. Aunque al principio puede ser algo complicado, una vez dominada la sintaxis y el uso de punteros (básico en este lenguaje), el C se vuelve una herramienta de programación muy potente y eficaz.

En los artículos publicados anteriormente se prometió que también se trataría el lenguaje C para los microcontroladores PIC. En estos meses se ha estado preparando la versión en castellano del compilador C2C desarrollado por Pavel Baranov. Entre las características más destacadas del compilador C2C encontramos las siguientes:

—Compatible con la mayoría de los microcontroladores PIC y Scenix.

—Soporta variables de 8 y 16 bits, al igual que matrices unidimensionales de 8 bits y punteros.

—Incluye variables predefinidas que hacen referencia a los diferentes registros de los microcontroladores.

—Permite el empleo de expresiones de 8 y 16 bits.

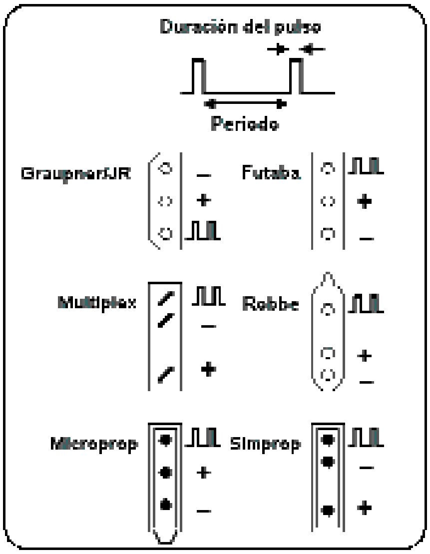
—Gran variedad de funciones integradas en el compilador como conversión a BCD, transmisión serie, etc. También permite la inclusión de rutinas de tratamiento de interrupciones.

—Inserción de código ensamblador.

—Empleo y construcción de librerías para uso posterior.

—Entorno de programación amigable e intuitivo que incluye un asistente de configuración del microcontrolador que nos permitirá, entre otras cosas, la programación multitarea y configurar de forma fácil el modo de operación del microcontrolador.

—Características básicas del lenguaje C como definición de funciones y procedimientos que permiten el paso/devolución de parámetros.



«Figura 1». Conexiones de diferentes marcas de servomotores.

un vistazo a algunas páginas web de estos comercios: www.ciberia.es/Relcomhobbyes o www.rcompany, entre otras.

El voltaje de alimentación nominal, suele ser el que pueden proporcionar cuatro baterías de NiCd 4 x 1.2 V = 4.8 V. En la práctica este valor puede variar. Algunas compañías de radiocontrol fabrican paquetes de cinco celdas de NiCd que proporcionan un valor nominal de 6 V.

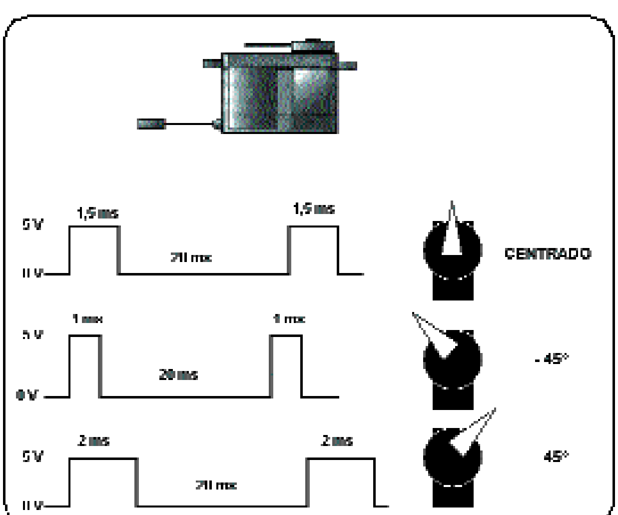
Si el voltaje es demasiado bajo, la respuesta del «servo» se hace más lenta. En el hexápodo se emplearán cuatro pilas alcalinas de 1,5 V. Aunque puede parecer que el consumo será alto, el consumo de corriente depende en su mayor parte del momento (par de fuerza) que se aplique al «servo» y puede llegar en algunos casos a ser mayor de 1 A si el «servo» se para sin poder llegar a la posición especificada.

Esquema de control

Hay dos formas de contemplar este tipo de esquemas de control. Desde el punto de vista del controlador, es un sistema de bucle abierto. No existe retroalimentación entre el servomotor y el sistema que genera los pulsos. Desde el punto de vista del nivel local (interior del «servo») es un sistema de bucle cerrado. La electrónica del servomotor está constantemente tratando de eliminar la diferencia entre los comandos y la posición actual.

Esta doble personalidad es una característica muy importante, ya que el «servo» necesita una atención mínima por parte del controlador, pero a su vez de forma constante resiste activamente corrigiendo las influencias externas que pueden llevar el eje lejos de la posición ordenada.

Aunque los «servos» son los posicionadores casi ideales, son también fáciles de modificar para aplicaciones especiales. Por ejemplo, se puede alterar el circuito de retroalimentación para modificar el rango de giro. La mayoría de los servomotores se han diseñado para un viaje de unos 90º, pero en muchos casos esta limitación puede superarse.

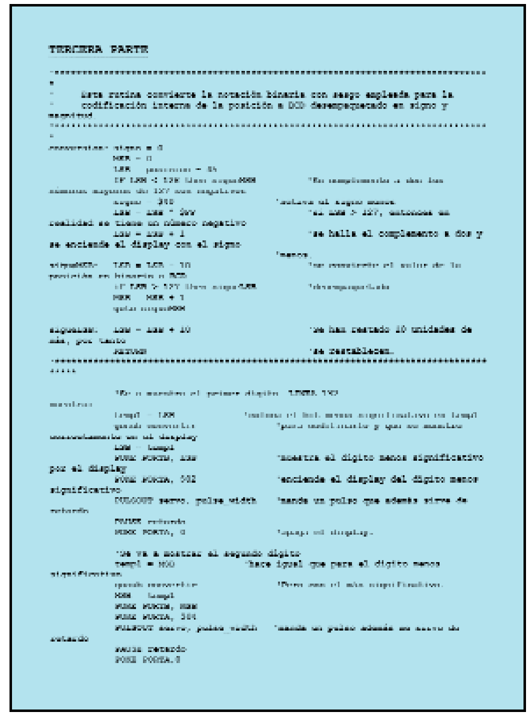


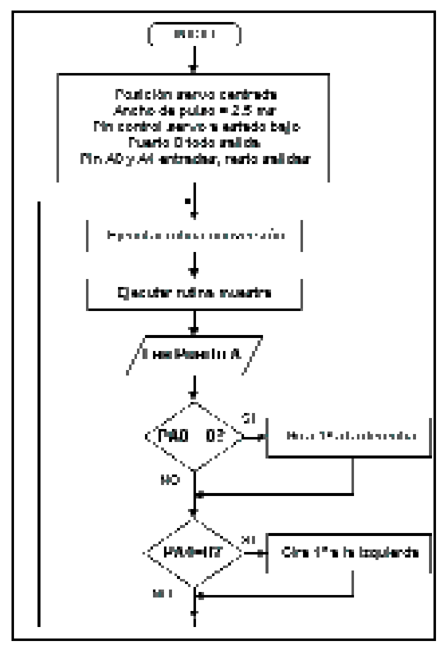
«Figura 2». Posiciones del eje de un servomotor según la anchura del pulso de entrada. Un tren de pulsos, con repetición comprendida entre 40 y 100Hz, controla la posición del «servo». Típicamente los «servos» tienen un rango de 90º, pero puede ampliarse a 180º o más. Muchos «servos» poseen una resolución de la posición mejor de medio grado.

Cuando se necesite mayor cantidad de giro de la que el fabricante ha dotado al «servo», la mejor solución es actuar modificando el potenciómetro del circuito de retroalimentación. Para ello se añade una resistencia de un valor comprendido entre 1K5 a 2K2 en serie con cada extremo del potenciómetro y luego se vuelve a montar. De esta forma los pulsos de la señal de control aumentarán el rango de giro.

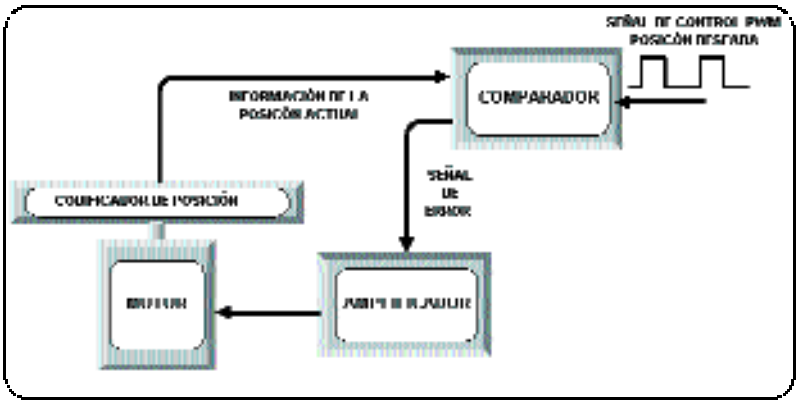
También se puede modificar el «servo» para que se comporte como un pequeño motor controlador mediante pulsos. Si se quita el potenciómetro interno y se sustituye por dos resistencias de 2K5, el circuito interno creerá que el eje del motor se encuentra siempre en posición centrada, así pues, si se envía señal de control para que se posicione a la derecha, el «servo» tratará de corregir continuamente la posición y girará en ese sentido. Se tiene de este modo un motor con engranajes cuya dirección de rotación pueden ser controlada por un tren de pulsos mediante la técnica PWM.

Por otro lado, si se aplican señales analógicas al circuito de retroalimentación se





pueden mezclar los efectos del control digital remoto con el ajuste analógico local. Estas modificaciones se detallarán extensamente en posteriores artículos, ya que para los hexápodos los «servos» se emplean tal y como los diseñó el fabricante.

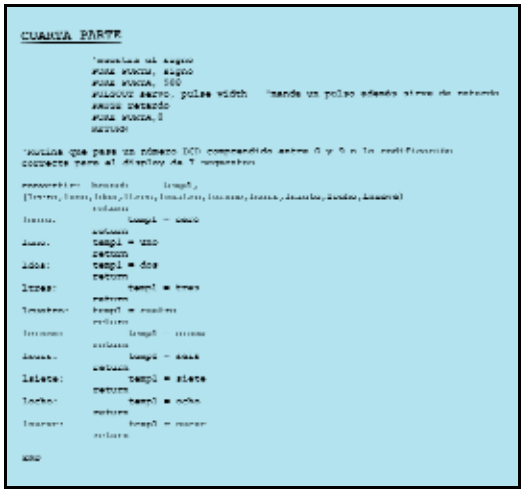


«Figura 3». Esquema de bloques de funcionamiento de un servomotor

Comprobador

La aplicación de este mes consistirá en realizar un comprobador de servomotores, empleando para ello el Módulo-01 y el de aprendizaje (foto 5).

Se utilizan dos pulsadores del Módulo de aprendizaje: el A0 que al ser pulsado hará girar el motor en pasos de 1º hacia la derecha y el A4 que girará el motor en pasos de 1º hacia la izquierda. La señal de control PWM se obtiene del pin PB0, disponible en una de las bornas el conector de salida CN3 del Módulo de aprendizaje. También la alimentación del «servo» se ha tomado del Módulo de aprendizaje, ya que se dispone de +5 en las bornas de CN4, mientras que la masa se encuentra junto con PBO en la borna CN3.



Los *displays* de siete segmentos del Módulo de aprendizaje indicarán la posición del eje del «servo» en número de grados girado. Para que funcionen los *displays* de siete segmentos, deberemos cambiar la posición de los *jumpers* J5, J6, J7, J8, J9 y J11, de tal forma que ahora en lugar de controlar los botones A1, A2 y A3, ataquen los transistores que controlan los *displays*.

En la posición central del motor marcarán 00, mientras que si se avanza hacia la derecha irán apareciendo los grados del giro, es decir, si giramos 15 grados a la derecha en los *displays* aparecerá 15, mientras que si giramos 20 grados a la izquierda en los *displays* aparecerá el número –20.

Programa

En el listado del programa (dividido en cuatro partes a lo largo del artículo) se dan todas las explicaciones para su perfecto entendimiento. En él sólo aparecen dos instrucciones nuevas: PULSOUT y GOSUB-RETURN. Además del programa que se presenta para los lectores que siguen el curso de Pbasic, el mismo también se ha escrito completamente en ensamblador.

PULSOUT Pin, Duración

Pin: Valores de 0 a 7.

Duración: Hasta 655,350.

Esta instrucción genera un pulso en Pin de un ancho especificado en Duración en unidades de 10µs. Como la variable Duración es de 16 bits, pueden generarse pulsos de hasta 655.350µs. El pulso es generado invirtiendo la polaridad del Pin dos veces, por lo que el estado inicial del mismo determina su polaridad. El Pin es programado automáticamente como salida y puede estar comprendido entre 0 y 7.

Ejemplo:

Pulsout 5, 100 'Envía un pulso de 1 ms de ancho al Pin 5

GOSUB Etiqueta

Esta instrucción salta a la línea que comienza por Etiqueta y ejecuta el código que le sigue. Cuando encuentra la instrucción RETURN regresa y continúa la ejecución del programa por la línea siguiente a la instrucción GOSUB. El código que se encuentra entre Etiqueta y la instrucción RETURN se conoce como subrutina.

Las subrutinas se pueden anidar, es decir, es posible que una subrutina llame a otra subrutina, pero el anidamiento está restringido a un máximo de cuatro niveles.

RETURN

Retorno de una subrutina. Como se ha indicado, cuando se encuentra una instrucción RETURN se continúa la ejecución del código que sigue a la instrucción GOSUB que generó el salto a la subrutina.

Bibliografía

Microcontroladores PIC. La solución en un chip. Martín Cuenca, E., Angulo J.M y Angulo, I. (1998). 2ª Edición. Paraninfo-ITP.

Diseño y Realización de Aplicaciones Industriales con Microcontroladores PIC Martín Cuenca, E. y Moreno Balboa, J.M. (en preparación).

Fundamentos de Electrónica Moderna. Teoría y Diseño de Circuitos. Martín Cuenca, E. y Moreno Balboa, J.M. (1998).

A new model describing coordination pattern of the legs of a Walking Stick Insect. Cruse, H. (1997). Biological Cybernetics 107-113.

A Model of Leg coordination in the stick insect, Carasius mososus: I. A geometrical consideration of contralateral and ipsilateral coordination mechanims between two adjacent legs. Dean, J. (1991). Biological Cybernetics 393-402.

A Design concept for Legger robots derived from walking stick insect Weidemann, H.J. et al. (1993). Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. 545-552.

Mathematical recreations. Insectoids invade a field of Robots Dewdney, A.K. (1991). Scientific American. 93-95.

Más información

Los lectores interesados en adquirir el compilador de Basic PBC o el compilador de C en castellano C2CWes pueden solicitarlo directamente al autor del artículo.

Dr. Eugenio Martín Cuenca
(emartin@goliat.ugr.es)
Ing. Luis Javier Asensio Morcillo