



La inteligencia de la máquina



Concluimos la descripción básica del robot hexápodo

En futuros artículos intentaremos dotar a nuestro pequeño robot de mayores prestaciones e «inteligencia», aunque antes de que esto ocurra realizaremos una pausa en el apartado de la robótica para tratar de avanzar en otros tan interesantes como la bioingeniería, la telemedicina, las medidas de parámetros medioambientales etc.

El StampBug no es muy inteligente por el momento, sólo sabe caminar por diferentes tipos de superficies y cuando detecta un obstáculo con sus antenas da media vuelta y sigue su camino. Para caminar emplea el método conocido como de los triángulos alternantes. Utilizando esta forma de caminar el robot tiene en cada momento las patas delanteras y traseras de un lado apoyadas en el suelo, así como la pata central del lado opuesto.

Este es el método más simple para que el hexápodo pueda caminar hacia delante y atrás en línea recta y girar a la derecha y a la izquierda. Debe recordarse en este punto que las patas derechas primera y tercera (D1 y D3) están controladas por el «servo» derecho: la D3 directamente y la D1 a través del tendón. Por otro lado, las patas izquierdas primera y tercera (IZ1 e IZ3) están controladas por el «servo» izquierdo: la IZ3 directamente y la IZ1 mediante el tendón izquierdo. Finalmente indicar que las patas centrales, tanto D2 como IZ2, están controladas por el «servo» central.

Para que la operación anterior pueda realizarse, la pata central ha de estar levantada en el momento oportuno. Esto es posible por la diferente disposición del «servo» central, cuyo giro produce la subida o bajada de la pata central de uno de los lados de forma alternativa. Al estar formadas las dos patas centrales por un solo elemento, cuando la de un lado sube, obligatoriamente baja la del opuesto y viceversa.

El robot avanza apagando y encendiendo alternativamente sus ojos, hasta que una de sus antenas detecta un obstáculo. En este punto retrocede y, antes de seguir adelante,

realiza un giro hacia la izquierda o la derecha dependiendo de cuál fue la antena que detectó el obstáculo.

■ El programa

Una vez montado el hexápodo, si hemos elegido adquirir el *kit* sólo debemos seguir las indicaciones que ya se mencionaron el mes pasado. Aquellos más audaces que hayan decidido acometer su propia realización pueden utilizar para el cuerpo y las patas algún

bro es apropiado emplear un Módulo-01SM (ver «Imagen 2»), que incorpora un PIC16F84.

Tal y como prometimos, el programa se ha realizado en lenguaje C, empleando para ello el compilador C2CWES, en castellano, lo que facilita el trabajo a los lectores.

Lo primero que debemos hacer es abrir el C2CWES y escribir el programa «STUMP2.C» que incluimos en el CD-ROM que acompaña a la revista en el apartado «VNU Labs», epígrafe «Microcontroladores». Hay que utilizar el editor que incorpora el propio compilador. A continuación debemos seleccionar la opción *Compilar* para obtener el fichero «STUMP2.ASM» que debemos utilizar a continuación en el grabador MultiPIC (ver «Imagen 4»).

Una vez escrito y compilado el programa habrá que grabarlo en un PIC16F84 empleando el programador MultiPIC. Los pasos a seguir para la realización de las anteriores operaciones ya se describieron en el número de octubre de 1999 en el artículo sobre el empleo del MultiPIC.

■ Descripción del programa

La estructura global del programa se muestra en el organigrama general y en los organigramas de las distintas rutinas. Todos los organigramas se encuentran en el CD-ROM que acompaña a la revista en el apartado «VNU Labs», epígrafe «Microcontroladores».

El organigrama general presenta una visión global de cómo camina el hexápodo. La ejecución del mismo sigue los siguientes pasos. El programa, una vez realizada la configuración de los pines que actuarán como entradas o salidas, procede a comprobar, en primer lugar, la forma de caminar seleccionada dependiendo del terreno por el que se va a mover el hexápodo. Por ejemplo, si el terreno es escarpado, los «servos» centrales harán recorridos más largos, mientras que si el terreno es liso, los recorridos serán más cortos.

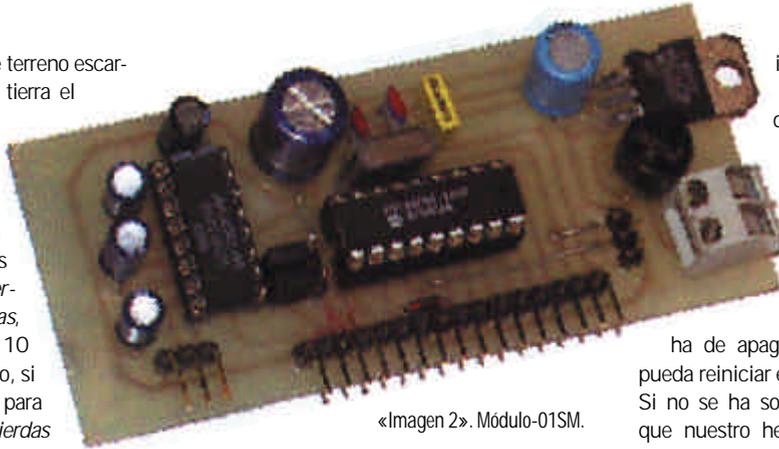


tipo de material plástico, metacrilato o panel de madera. Respecto a la mecanización, los tres servomotores se consiguen en cualquier tienda de aeromodelismo. Los modelos Futaba FP-S3003 tienen un tamaño y unas características apropiadas. Por último, como cere-



Para seleccionar el modo de terreno escarpado habría que «puntear» a tierra el *jumper* al efecto que viene denominado como *mode* y que se corresponde con el pin PBO. Los grados que debe girar el «servo» central vienen determinados por el valor de las constantes *apoya_patas_izquierdas* y *apoya_patas_derechas*, expresadas en unidades de 10 microsegundos. Así, por ejemplo, si seleccionamos un valor de 143 para la constante *apoya_patas_izquierdas* y 157 para *apoya_patas_derechas*, el programa enviará pulsos al «servo» de duración 1,43 ms y 1,57 ms respectivamente, dependiendo de la acción a realizar.

Para saber cuántos grados gira un «servo» con respecto a su posición central, recordaremos que para una duración del ancho de pulso de 1,50 ms el «servo» se mantiene centrado. Si la duración del pulso es menor de 1,50 ms, el «servo» gira en sentido horario, mientras que si es mayor de 1,50 ms lo hace en sentido antihorario. El «servo» gira aproximadamente un grado por cada 0,10 ms. De esta forma, para un ancho de pulso de 1,43 ms el «servo» girará unos 7 grados en sentido horario, mientras que para un ancho de pulso de 1,57 ms girará 7 grados



«Imagen 2». Módulo-01SM.

El programa de control se ha realizado en lenguaje C, empleando el compilador C2CWEs en castellano

en sentido antihorario. Todo se explicó en el artículo «Robótica bioinspirada I. Servomotores y su control» de enero de 2000.

Elegido el modo de caminar, en segundo lugar se comprueba si se ha solicitado la rutina de calibración. Si se solicita este servicio, para lo cual ha de tenerse pulsada la antena

izquierda al encender el robot, esta rutina se encarga de realizar el centrado de los servomotores izquierdo y derecho. Para ello manda pulsos de 1,50 ms de duración a dichos «servos».

Como esta rutina es un bucle sin fin, si se ha realizado esta petición el robot

ha de apagarse y encenderse para que pueda reiniciar el programa (ver «Imagen 6»). Si no se ha solicitado calibración, veremos que nuestro hexápodo por fin comienza a caminar. El programa entra en un bucle continuo, es decir, caminará y realizará las siguientes funciones mientras no lo apaguemos o le duren las pilas.

El comportamiento del hexápodo es el siguiente. Primero comprueba si hay algún sensor (antena) activo; si no lo hay da un paso adelante. La acción de dar un paso adelante es una tarea compleja que requiere de otras acciones más simples para llevarla a buen término y se describirá a continuación. Luego se asegura si la antena izquierda está activa; si no lo está, comprueba el estado de la antena derecha. Si ninguno de los dos sensores estaba activo durante la comprobación, el hexápodo dará otro paso hacia adelante, repitiendo el ciclo.



Detectado un obstáculo se comprueba qué antena ha sido la que se ha activado. Si ha sido la antena izquierda, el robot dará cuatro pasos hacia atrás y girará hacia la derecha siguiendo tranquilamente su camino. Si por el contrario ha sido la antena derecha, su comportamiento será cuatro pasos hacia atrás, giro a la izquierda y andando se hace camino. En este punto se repite el bucle, da un paso adelante y comprueba que no haya ninguna antena contactando obstáculos, tras lo que vuelve a dar un paso hacia delante.

En los organigramas 1 y 2 se ofrece una visión más detallada de cómo funciona el hexápodo. El organigrama 1 es una réplica casi exacta del general, pero en él se muestran las llamadas a las diferentes partes del programa que se encargan de realizar las tareas de dar un paso adelante, dar un paso hacia atrás y girar a un lado u a otro. Éste, junto con el organigrama 2 (excluyendo la subrutina *Walk_Back*), corresponde al cuerpo principal del programa en C.

■ Paso hacia delante

Si no había ningún sensor activo, el hexápodo daba un paso hacia delante. Las tareas

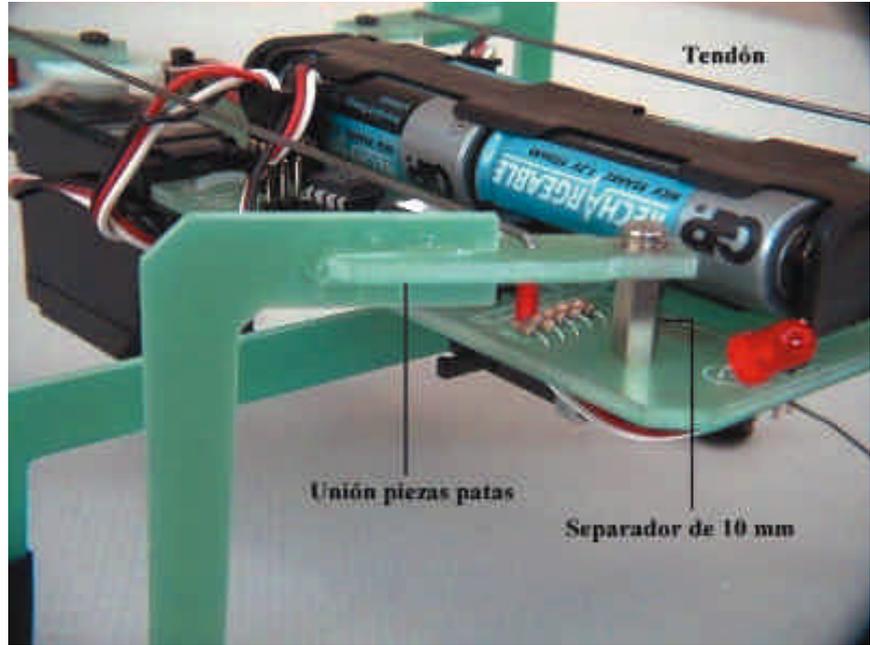


Editor que incorpora el compilador C2CWES. Una vez escrito y compilado el programa habrá que grabarlo en un PIC16F84 empleando el programador MultiPIC.

que se requieren para hacer esto vienen descritas en el organigrama 2, en el punto de entrada A. El proceso a seguir es el descrito a continuación.

El «servo» central gira para apoyar la pata central derecha D2, lo que provoca que se apoyen las patas del lado izquierdo IZ1 e IZ3 y quede levantada la IZ2. El movimiento anterior se indica en el organigrama como «apoya patas izquierdas». A continuación, el «servo» izquierdo gira en sentido antihorario, produciendo como efecto directo el retraso de la pata IZ3 y mediante el tendón el retraso de la IZ1. Este movimiento se indica en el organigrama como «retrasa las patas izquierdas».

Al mismo tiempo que se produce el retraso de las patas izquierdas, se realiza la acción «avanza patas derechas» mediante el giro del «servo» derecho en contra de las agujas del reloj. Para culminar un paso adelante, el «servo» central gira de tal forma que quedan apoyadas las patas IZ2, D1 y D3 mediante la acción «apoya patas derechas». A continuación el «servo» izquierdo gira en sentido



Ensamblado de las patas a la placa principal.

horario provocando que las patas IZ1 e IZ3 avancen (a esta acción la denominaremos «avanza patas izquierdas»). Al mismo tiempo, se retrasan las patas derechas por medio del giro del «servo» derecho, también en sentido horario. Esta acción será descrita como «retrasa patas derechas».

■ Paso hacia atrás

Cada vez que el hexápodo encuentra un obstáculo, su primera acción es alejarse del mismo; para ello repite cuatro veces el proceso de dar un paso hacia atrás (subrutina *Walk_Back* en el organigrama 2). Esta acción se ejecuta independientemente de si se activó el sensor izquierdo o

derecho (puntos de entrada B y C en el organigrama 2).

Para dar un paso hacia atrás el hexápodo «apoya las patas izquierdas». Una vez apoyadas «avanza las patas izquierdas» y «retrasa las patas derechas» al mismo tiempo. Posteriormente, «apoya las patas derechas», «retrasa patas izquierdas» y «avanza patas derechas». Completando así un paso hacia atrás.

■ Giro hacia la derecha

Una vez alejado del obstáculo, y si el sensor que se activó fue el izquierdo, el hexápodo girará hacia la derecha para evitar el objeto que interrumpió su marcha normal (ver organigrama 2 punto de entrada B). Para llevar a buen término el giro hacia la derecha, el hexápodo realiza las siguientes tareas: «apoya patas izquierdas» y una vez apoyadas «retrasa patas izquierdas» y «retrasa patas derechas». A continuación «apoya patas

Programador MultiPIC

El programador MultiPIC es un dispositivo de bajo coste que permite grabar la mayoría de los microcontroladores PIC: desde los de la gama baja de 8 pines como el 12C508 que se encuentra en el sistema anticopia de la PlayStation, hasta los de 40 pines como el 16C74 de la gama media, así como los más modernos flash 16F87X tanto de 28 como de 40 pines. Las características más destacadas son :

- Graba la mayoría de los PIC de gama media: 8 pines , 18 pines , 28 pines, 40 pines.

- Se entrega con dos versiones del

software, una bajo DOS y otra para Windows 95/98.

- La versión Windows 95/98 es un entorno de desarrollo completamente integrado de editor, compilador (MPASM) y grabador.

- Es integrable y compatible con el compilador Basic PBC y con el compilador gratuito Let Basic.

- Dado lo ajustado de su coste, está totalmente garantizado para las series flash PIC16C84, PIC16F84, PIC16F873, PIC16F876, PIC16F874 y PIC16F877, con resultados positivos con otras series como OTP y EPROM. El programador se entrega con un transformador de corriente alterna C.A. de 14-15 voltios.



«Imagen 4». Grabador MultiPIC.

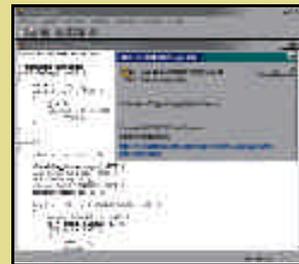
derechas», «avanza patas izquierdas» y «avanza patas derechas». Este proceso se repite un total de cuatro veces.

■ Giro hacia la izquierda

En el caso de que fuese el sensor derecho el activado, el hexápodo realizará esta vez un giro hacia la izquierda (ver organigrama 2 punto de entrada C). Al igual que para el giro hacia la derecha, se repetirá cuatro veces lo siguiente: «apoya patas derechas» y una vez apoyadas «retrasa patas izquierdas» y «retrasa patas derechas». Para a continuación: «apoya

Compilador C2CWEs

Hoy en día el uso del 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 que es compatible con la mayoría de los microcontroladores PIC y Scenix, que soporta variables de 8 y 16 bits al igual que matrices unidimensionales de 8 bits y punteros, que incluye variables predefinidas que hacen referencia a los diferentes registros de los microcontroladores, que permite el empleo de expresiones de 8 y 16 bits, y que cuenta con 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, la inserción de código ensamblador y el empleo y construcción de librerías para uso posterior. El entorno de programación es amigable e intuitivo e incluye un asistente de configuración del microcontrolador que nos permitirá hacer programación multitarea y configurar de forma fácil el modo de operación del microcontrolador.

parámetros: posición del «servo» central, del «servo» izquierdo, del «servo» derecho y un último parámetro que indica a la función si debe comprobar el estado de los sensores. Este último parámetro se incluye porque cuando el hexápodo camina hacia atrás no se deben comprobar los sensores, mientras que en cualquier otro caso sí.

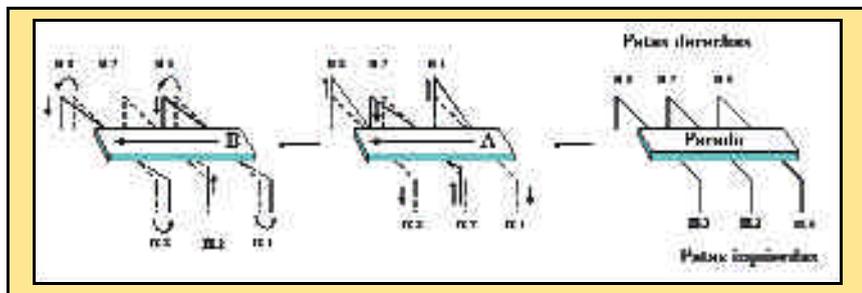
un segundo bucle que se repetirá tantas veces como indique la constante *pulses*. En éste se envían los pulsos correspondientes a los «servos» derecho e izquierdo. También se siguen mandando pulsos al «servo» central para evitar fallos en la sincronización. Entre pulso y pulso se hace una pausa en el programa de 10 ms por las razones indicadas anteriormente y se comprueban los sensores si es necesario. En caso de que algún sensor esté activo cuando se hace la comprobación de los mismos, la rutina finaliza inmediatamente devolviendo como valor de retorno el sensor activado.

Una constante muy importante para el funcionamiento del programa es la constante *pulses*. Jugando con el valor de la misma (por defecto el valor es 20) se puede conseguir diversos efectos. Por ejemplo, si queremos que el hexápodo camine más rápido podemos ponerla a un valor más pequeño. Pero hemos de tener cuidado ya que un valor excesivamente pequeño evitará que los «servos» completen su situación correcta, haciendo el caminar del hexápodo errático.

Si aumentamos el valor de esta constante haremos que el hexápodo camine más lentamente (perfecto para ver cómo camina), pero un valor muy alto haría que permaneciese demasiado tiempo en el primer bucle de la función *Walk*, evitando un correcto refresco de los pulsos en los «servos» derecho e izquierdo.

Lo primero que se tiene que hacer es girar el «servo» central hasta la posición correcta para apoyar las patas que se requieran. Para ello se envía el pulso de duración correspondiente al «servo» central. Esta acción se repite un número de veces determinado por la constante *pulses*, dando tiempo así a que el «servo» central se sitúe. Entre pulso y pulso se hace una pausa de 10 ms para asegurar un correcto funcionamiento del «servo» y se comprueba el estado de los sensores si el cuarto parámetro de la función así lo indica.

A continuación, se realiza



«Imagen 6». Método de los triángulos.

patas izquierdas», «avanza patas izquierdas» y «avanza patas derechas».

■ Control de los «servos»

Hasta ahora se ha descrito cómo se mueve el hexápodo indicando qué es lo que tienen que hacer las diferentes patas del mismo. Observando el organigrama 2 se aprecia cómo aparece constantemente en un cuadro rectangular las acciones a realizar por cada una de las patitas. Cada recuadro simboliza una llamada a la función encargada de controlar directamente los «servos»: la función *Walk*. El funcionamiento de la misma viene descrito en el organigrama 3. A esta función se le pasan cuatro

Más Información

Los lectores interesados en adquirir los diferentes kits, así como el compilador de C en español C2CWEs, encontraréis información detallada en <http://curtis.ugr.es> o podéis solicitarlo directamente al autor del artículo. En la misma dirección tenéis a vuestra completa disposición todos los artículos previamente publicados.

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