

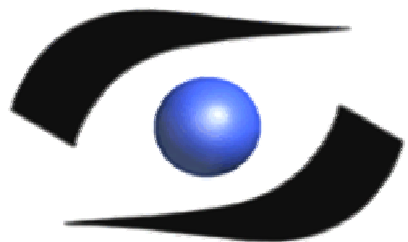
Servomotores

Publicación Interna 9
20-09-2007

Francisco A. Candelas Herías
Juan A. Corrales Ramón



Universitat d'Alacant
Universidad de Alicante



Grupo de Automática, Robótica y Visión Artificial
Departamento de Física, Ingeniería de Sistemas y Teoría de la Señal

1.- Introducción a los Servomotores

1.1.- Definición y Estructura

Un servomotor (o servo) es un motor de corriente continua que tiene la capacidad de ser controlado en posición. Es capaz de ubicarse en cualquier posición dentro de un rango de operación (generalmente de 180°) y mantenerse estable en dicha posición. Los servos se suelen utilizar en robótica, automática y modelismo (vehículos por radio-control, RC) debido a su gran precisión en el posicionamiento.

En general, los servos suelen estar compuestos por 4 elementos fundamentales:

- Motor de corriente continua (DC): Es el elemento que le brinda movilidad al servo. Cuando se aplica un potencial a sus dos terminales, este motor gira en un sentido a su velocidad máxima. Si el voltaje aplicado sus dos terminales es inverso, el sentido de giro también se invierte.
- Engranajes reductores: Tren de engranajes que se encarga de reducir la alta velocidad de giro del motor para acrecentar su capacidad de torque (o par-motor).
- Sensor de desplazamiento: Suele ser un potenciómetro colocado en el eje de salida del servo que se utiliza para conocer la posición angular del motor.
- Circuito de control: Es una placa electrónica que implementa una estrategia de control de la posición por realimentación. Para ello, este circuito compara la señal de entrada de referencia (posición deseada) con la posición actual medida por el potenciómetro. La diferencia entre la posición actual y la deseada es amplificada y utilizada para mover el motor en la dirección necesaria para reducir el error.

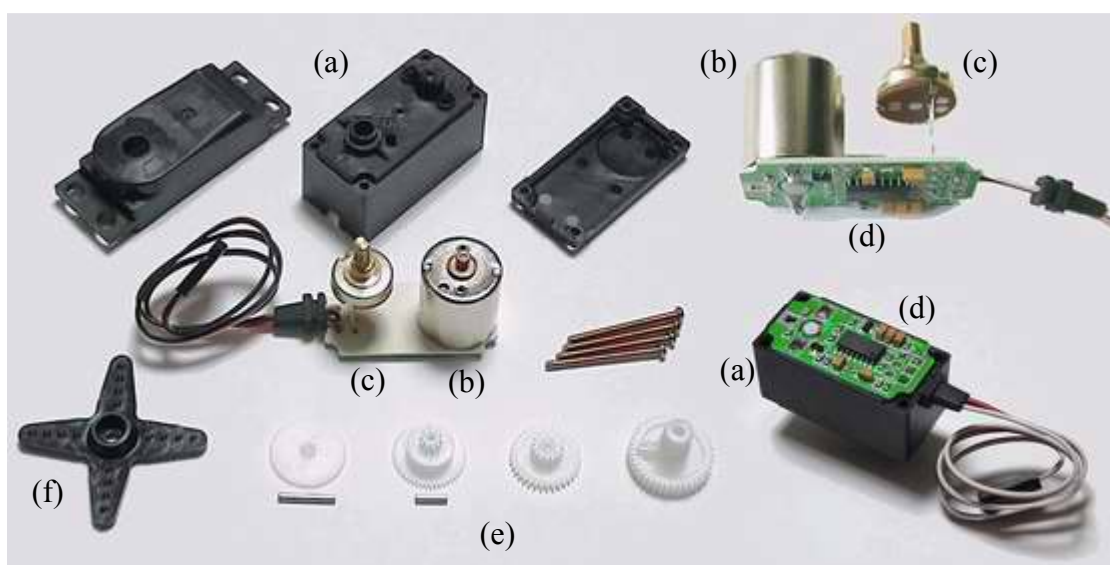


Figura 1. Componentes de un servo: a) carcasa; b) motor DC; c) potenciómetro; d) circuito de control; e) tren reductor; f) brazo (elemento terminal en el eje). Fuente: [1].



1.2.- Principios de Funcionamiento

Los servos disponen de tres cables (Figura 2): dos cables de alimentación (positivo y negativo/masa) que suministran un voltaje 4.8-6V y un cable de control que indica la posición deseada al circuito de control mediante señales PWM (“Pulse Width Modulation”).

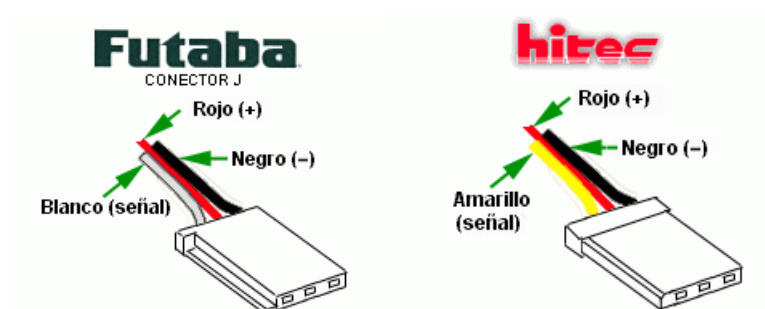


Figura 2. Colores de los cables de los principales fabricantes de servos. Fuente: [1].

Las señales PWM utilizadas para controlar los servos están formadas por pulsos positivos cuya duración es proporcional a la posición deseada del servo y que se repiten cada 20ms (50Hz). Todos los servos pueden funcionar correctamente en un rango de movimiento de 90°, que se corresponde con pulsos PWM comprendidos entre 0.9 y 2.1ms. Sin embargo, también existen servos que se pueden mover en un rango extendido de 180° y sus pulsos de control varían entre 0.5 y 2.5ms (Figura 3). Antes de utilizar un servo habrá que comprobar experimentalmente su rango de movimiento para no dañarlo. Para mantener fijo un servo en una posición habrá que enviar periódicamente el pulso correspondiente; ya que si no recibe señales, el eje del servo quedará libre y se podrá mover ejerciendo una leve presión.

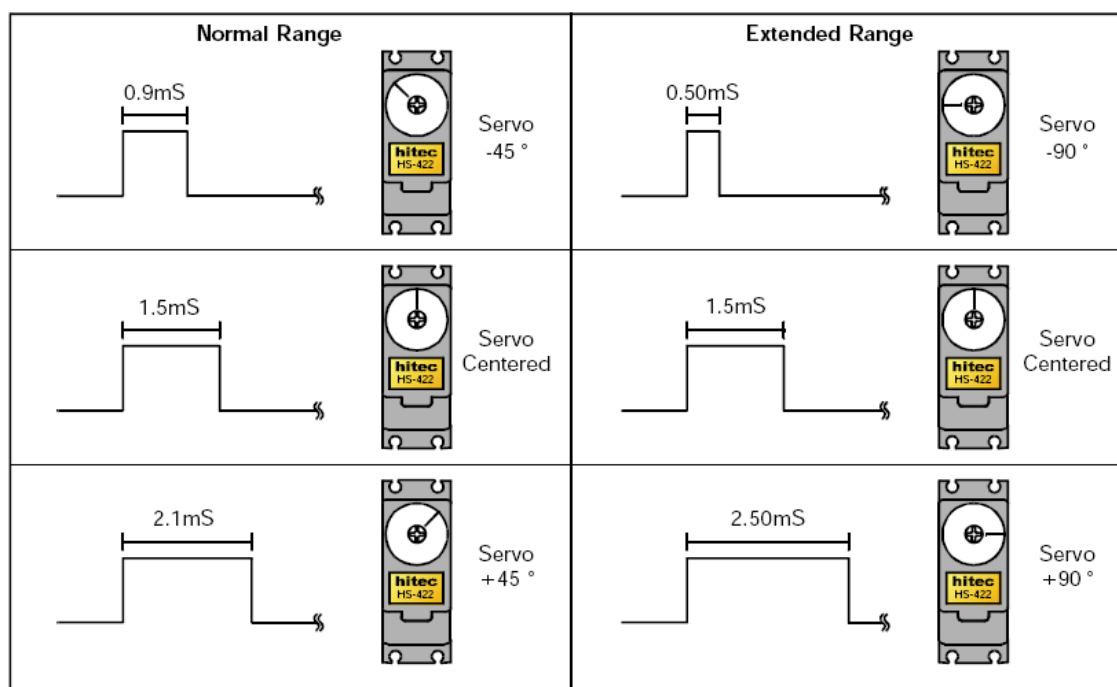


Figura 3. Pulsos PWM para controlar servos. Fuente: [2].

1.3.- Tipologías

Existen dos tipos de servos: *analógicos* y *digitales*. Ambos tipos de servos son iguales a nivel de usuario: tienen la misma estructura (motor DC, engranajes reductores, potenciómetro y placa de control) y se controlan con las mismas señales PWM. La principal diferencia entre ellos radica en la adición de un microprocesador en el circuito de control de los servos digitales [3]. Este microprocesador se encarga de procesar la señal PWM de entrada y de controlar el motor mediante pulsos con una frecuencia 10 veces superior a los servos analógicos.

El aumento en la frecuencia de excitación del motor en los servos digitales permite disminuir su tiempo de respuesta (menor deadband), aumentar su resolución de movimiento y suavizar su aceleración/deceleración. El uso de un microprocesador permite también a los servos digitales programar distintos parámetros de configuración que son fijos en los analógicos: sentido de giro, posición central inicial, topes en el recorrido del servo, velocidad de respuesta del servo y resolución. Para establecer estos parámetros se deben utilizar aparatos específicos de cada marca. El principal inconveniente de los servos digitales es que consumen más energía que los analógicos al tener que generar más pulsos de control para el motor.

2.- Control de Servos

Para controlar los servos se les deben enviar pulsos PWM a través del cable de control, tal como se describió en el apartado 1.2. En los sistemas de modelismo, se utilizan dos componentes para controlar los servos: un receptor y una emisora. El receptor es el componente que se encarga de recibir los comandos inalámbricamente de la emisora y transformarlos en los pulsos PWM correspondientes que son enviados a los servos. La emisora es un mando que transmite las órdenes al receptor a través de señales inalámbricas con modulación AM, FM o PCM.

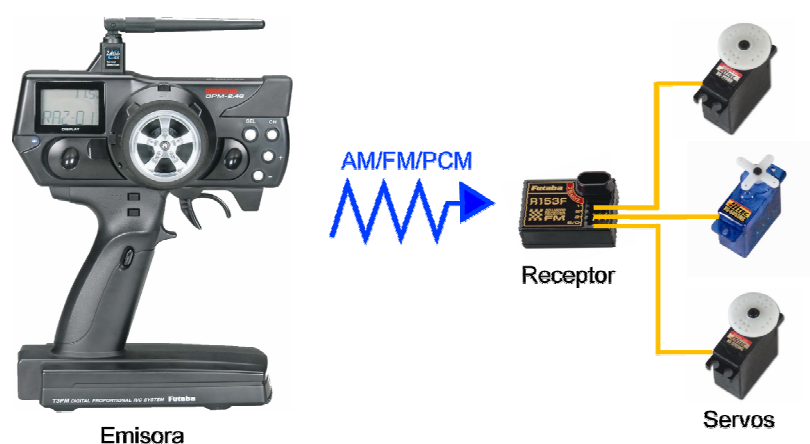


Figura 4. Pulsos PWM para controlar servos. Fuente: [4, 5].

Sin embargo, estos sistemas no se pueden conectar a un PC y la única forma de controlar los servos es con el mando. Existen otros circuitos controladores especialmente diseñados para controlar servos a través del PC. Suelen utilizar la interfaz serie (RS-232) para recibir los comandos de control. Estos comandos suelen ser diferentes dependiendo del modelo.



3.- Comparativa de Controladores Serie de Servos

Un SSC (*Serial Servo Controller*, Controlador Serie de Servos) es un dispositivo utilizado para controlar servos desde un PC a través del puerto serie. Los SSC aceptan comandos con un determinado formato desde el puerto serie del PC y los transforman en pulsos PWM que son enviados a los servos que se desea controlar.

3.1.- Mini SSC II

Mini SSC II (Figura 5) es un controlador de la empresa *Scott Edwards Electronics Inc.* que puede controlar hasta un máximo de 8 servos. Se conecta al PC a través de un cable serie (con conector RJ11 en el extremo del controlador y conector DB9 en el extremo del PC) y puede comunicarse con el puerto serie a dos velocidades: 2400bps o 9600bps. Este controlador requiere dos fuentes de alimentación: una de 9-12V para la circuitería y 5V para los servos.

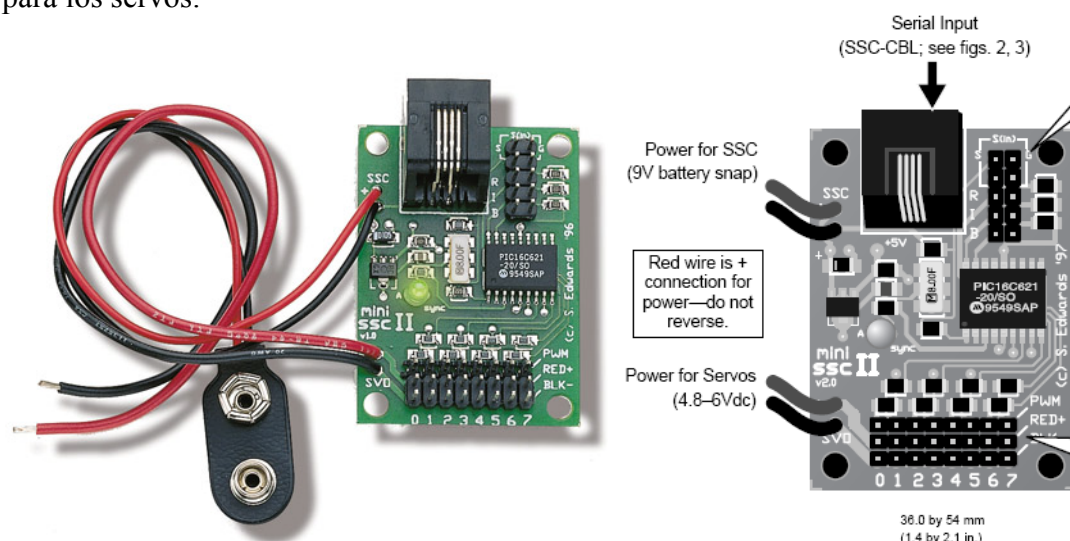


Figura 5. Controlador Mini SSC II. Fuente: [6].

El Mini SSC necesita tres bytes para controlar la posición de un servo. En primer lugar, se debe mandar un byte SYNC (255, 0xFF) para indicar que comienza un nuevo comando. A continuación, se manda un byte SERVO que identifica el servo a controlar. Finalmente, se manda el byte POSITION que indica la posición donde debe moverse el servo.

Comando Mini SSC: [SYNC (255)] [SERVO (0-254)] [POSITION (0-254)]

Las principales ventajas de este controlador son su reducido tamaño y la gran cantidad de software disponible debido a su amplia utilización en estos últimos años. Sus principales inconvenientes son la discontinuidad en su desarrollo (ya no se desarrollan nuevos modelos) y la necesidad de utilizar dos fuentes de alimentación distintas.

Este dispositivo se puede hacer inalámbrico conectando un adaptador bluetooth-serie a la salida serie del controlador. El adaptador utilizado deberá poder funcionar entre 2400 y 9600bps, 8/N/1 y no deberá disponer de control de flujo. El adaptador *BlueRS+E* de *Stollman GmbH* ha sido probado con éxito en aplicaciones de radio-control [7].

3.2.- SSC-32

SSC-32 (Figura 2) es un controlador de la empresa *Lynxmotion* que permite controlar hasta un máximo de 32 servos. Se trata de un controlador más completo que el Mini SSC ya que dispone de un conjunto de funcionalidades adicionales: control de servos por tiempo/velocidad/posición, movimiento síncrono de varios servos, consulta de posición de los servos y utilización de los pines de control de los servos como salidas digitales TTL. Además, dispone de 4 entradas (A, B, C y D) que pueden ser leídas de manera digital (bits) o de manera analógica (voltajes).

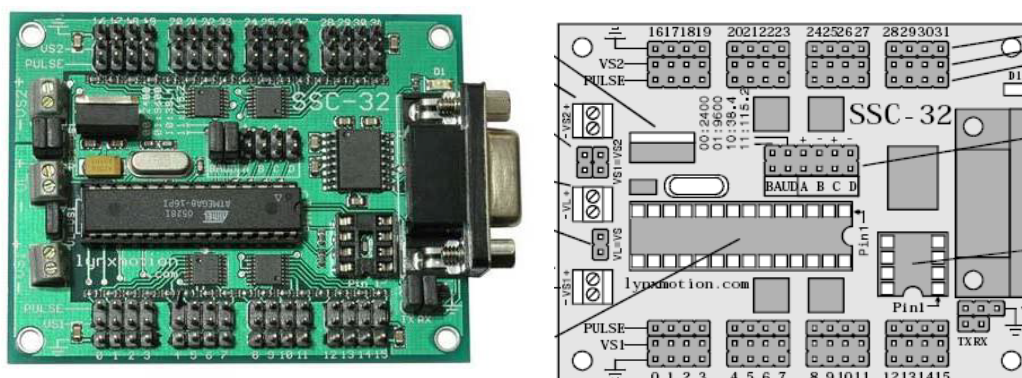


Figura 6. Controlador SSC-32. Fuente: [2].

Este controlador se puede transformar en inalámbrico con un adaptador bluetooth-serie, al igual que el Mini SSC. Pero, también se puede conectar a una tarjeta modem bluetooth [8] (*BlueSMiRF WRL-00582 o WRL-00158*), creando un módulo más compacto sin necesidad de alimentación adicional.

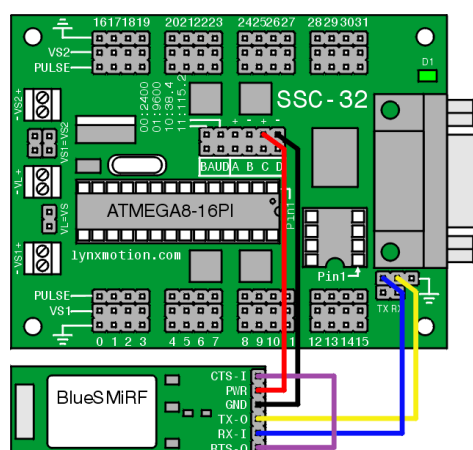


Figura 7. Conexiones entre SSC-32 y tarjeta modem bluetooth. Fuente: [8].

Este controlador presenta un gran número de ventajas respecto al Mini SSC: control más complejo de los servos, disponibilidad de 4 entradas, diversas configuraciones de alimentación (fuente única para placa y servos o fuentes separadas), mayor rango de velocidades del puerto serie (2400bps, 9600bps, 38.4 Kbps y 115.2Kbps), posibilidad de conectar el SSC-32 a un micro-controlador a través de comunicación serie TTL... Su único inconveniente es su mayor tamaño al disponer de un conector DB9 estándar.



3.3.- Arduino

Arduino es una plataforma open-hardware (diseño de libre distribución) basada en el diseño de una placa sencilla de entradas y salidas (analógicas y digitales) y un entorno de desarrollo que implementa el lenguaje Processing/Wiring. Existen varias versiones de Arduino según la interfaz de conexión al PC: serie, USB, Bluetooth y ZigBee. Esta placa presenta una gran comunidad de desarrollo y permite la conexión de sensores y actuadores variados.

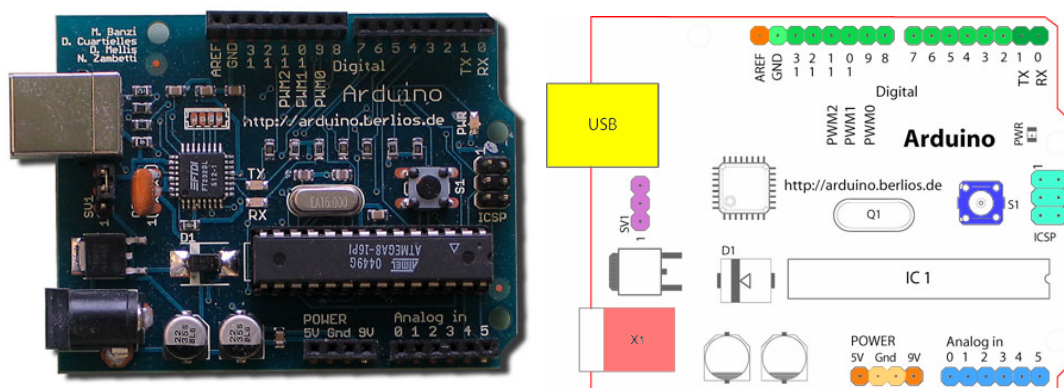


Figura 8. Controlador Arduino. Fuente: [9].

Este dispositivo no es un controlador específico para servos, como los dos anteriores, sino que es una placa de I/O que permite el control de servos. Además de controlar servos, también permite la lectura de varios tipos de sensores (acelerómetros, sensores de distancia, sensores de temperatura, sensores de humedad, sensores de luz, potenciómetros...) y el control de otros actuadores. Esta plataforma está en continua expansión y actualmente existe un distribuidor español, *Libelium* [10] (spin-off de la Universidad de Zaragoza), que desarrolla el sistema *SquidBee*. Este sistema pretende crear redes inalámbricas de sensores interconectados a través de módulos (denominados *motes*) basados en Arduino y ZigBee. Cada *SquidBee mote* está compuesto por una tarjeta Arduino con un módulo ZigBee (*Maxstream XBee OEM RF*) y tres sensores (humedad, temperatura y luminosidad). A este módulo se pueden añadir los sensores/actuadores que se deseen utilizando las entradas/salidas (12 digitales y 6 analógicas) de la tarjeta Arduino. Para poder leer los datos de los sensores y controlar los actuadores mediante un PC se utilizará un módulo *SquidBee gateway*. Este módulo internamente tiene los mismos componentes que el mote (tarjeta Arduino y módulo ZigBee) pero se encuentra configurado para recibir los datos de los sensores a través de ZigBee y comunicarlos al PC a través de USB. La comunicación mote-gateway es bidireccional.

No obstante, esta placa presenta ciertos inconvenientes para el control de servos al no ser un controlador específico. No permite opciones de control tan complejas sobre los servos como el SSC-32. Además, el cableado de los servos es más aparatoso ya que hay que conectar el cable de control a una salida digital y los cables de alimentación a otros pines que se encuentran en el lado opuesto de la placa. Por ello, no se pueden utilizar directamente los conectores estándares de tres pines de los servos en Arduino.

4.- Referencias

1. Carletti, E.J. *Servos: Características Básicas*. 2007 [cited 2007; Available from: http://robots-argentina.com.ar/MotorServo_basico.htm.
2. Lynxmotion, *User Manual SSC-32 Ver 2.0*. 2005: Pekin. p. 4.
3. Fubata, *Digital FET Servos*. 2007.
4. Futaba. *Futaba Radio Control*. 2007 [cited 2007; Available from: <http://www.futaba-rc.com/>.
5. Hitec. *Hitec RCD USA*. 2007 [cited 2007; Available from: <http://www.hitecrcd.com/>.
6. Scott Edwards Electronics, *Mini SSC II Serial Servo Controller, User's Manual*. 2000.
7. Mobile Review. *How to bluetooth-enable your remote controlled car?* 2003 [cited 2007; Available from: <http://www.mobile-review.com/articles/2003/bt-cars-en.shtml>.
8. Frye, J. *BlueSMiRF and SSC-32 Configuration Tutorial*. 2007 [cited 2007; Available from: <http://www.lynxmotion.com/images/html/build117.htm>.
9. SmartProjects. *Arduino*. 2007 [cited 2007; Available from: <http://www.arduino.cc>.
10. Libelium. *Libelium Comunicaciones Inalámbricas Distribuidas*. 2007 [cited 2007; Available from: <http://www.libelium.com/>.