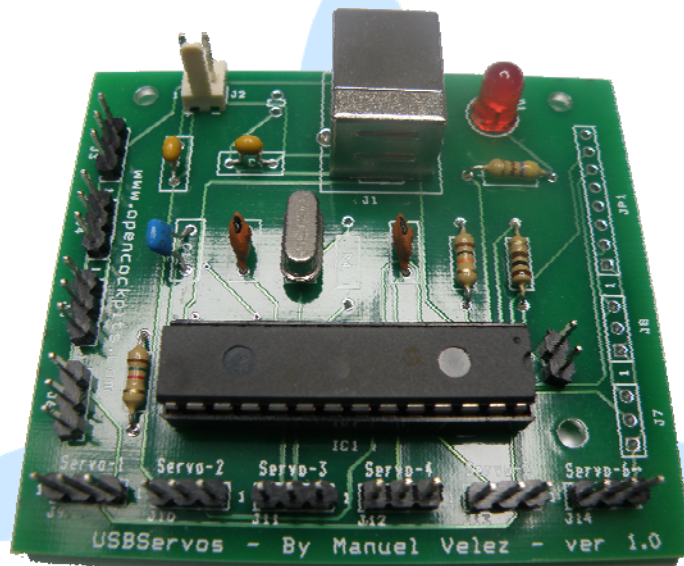




Opencockpits



Manual IOCard USB Servos

Índice:

MANUAL IOCARD USB SERVOS	1
ÍNDICE:.....	2
INTRODUCCIÓN:.....	3
USB SERVOS:	3
ESQUEMA Y COMPONENTES:	3
DESCRIPCIÓN DE LOS CONECTORES:	3
ESQUEMA DE CONEXIÓN:	3
PUESTA EN MARCHA DE LA TARJETA:	5
INSTALACIÓN Y CONFIGURACIÓN:	7
DECLARACIÓN DE SERVOS:	7
DECLARACIÓN DE ENTRADAS ANALÓGICAS:	8
EJEMPLOS DE SCRIPTS:.....	8
<i>Servos:</i>	8
<i>Entradas analógicas:</i>	9
LINKS DE INTERÉS:.....	9

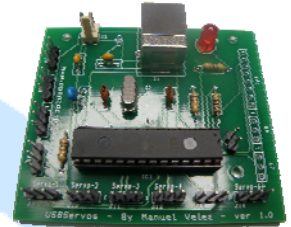
Introducción:

La USB Servos ha sido diseñada para poder manejar hasta 6 servomotores de los usados en radio control y además posee 4 entradas analógicas. La conexión de la tarjeta al ordenador es mediante el puerto USB.

Usb Servos:

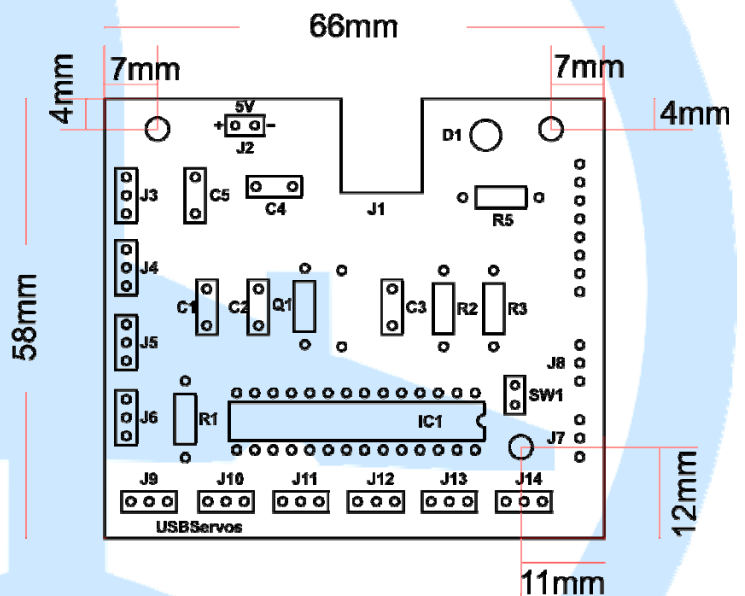
El sistema operativo reconoce y configura la placa como un dispositivo de entrada nada más conectarla. Para manejar la tarjeta haremos uso de SIOC con el protocolo IOCP integrado en él.

Para usar la tarjeta se dará de alta en el archivo sioc.ini, permitiendo gestionar los 6 servos y las 4 entradas analógicas.



Esquema y componentes:

- C1 = CONDENSADOR 220 nf
- C2, C3 = CONDENSADORES 22Pf
- C4, C5 = CONDENSADORES 0,1uF
- D1 = DIODO LED
- IC1 = MICROCONTROLADOR 16C745
- J1 = CONECTOR USB
- J2 = ENTRADA ALIMENTACION SERVOS 5V, 2 PINES (no protegida contra polaridad equivocada)
- J3 ... J14 = CONECTORES 3 PINES
- Q1 = CRISTAL DE CUARZO 6Mhz
- R1 = RESISTENCIA 1K5
- R2 = RESISTENCIA 10K
- R3 = RESISTENCIA 100R
- R5 = RESISTENCIA 470R
- SW1 = RESET (CONECTOR 2 PINES)



Descripción de los conectores:

- J1 = Conector USB, permite la conexión al ordenador directamente POR USB, en el momento de conectarse el ordenador reconocerá la tarjeta como un dispositivo de entrada.
- J2 = Conector de entrada de alimentación a 5V para los servos.
- J3 a J6 = Conectores para entradas analógicas (ver dibujo de conexiones).
- J9 a J14 = Conectores para los servos, los cables de los servos deben adaptarse según tabla adjunta más abajo (ver tabla de conexiones).

Esquema de conexión:

La conexión de la tarjeta es muy sencilla, para los potenciómetros tenemos los conectores de 3 pines J3 a J6, si observamos que las lecturas de las entradas no usadas fluctúan mucho podemos evitarlo poniendo un puente entre los pines 1 y 2 ó 2 y 3 de las mismas.

Para los servos tenemos los conectores J9 a J14, en estas conexiones tendremos que cambiar (si es necesario) el cableado del conector del servo para adaptarlo al conexionado de los pines de la tarjeta, dependiendo de la marca del servo y teniendo siempre en cuenta el orden de conexión de los servos a la tarjeta:

PIN	FUNCIÓN
1	+5v
2	DATA (S)
3	GND

Para una mejor explicación del conexionado, a continuación se adjunta una imagen:

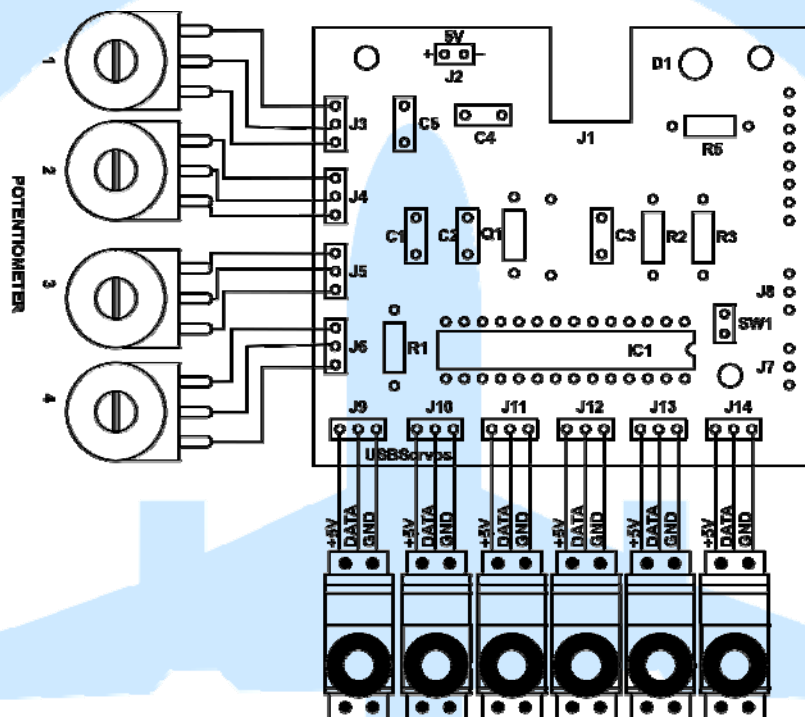
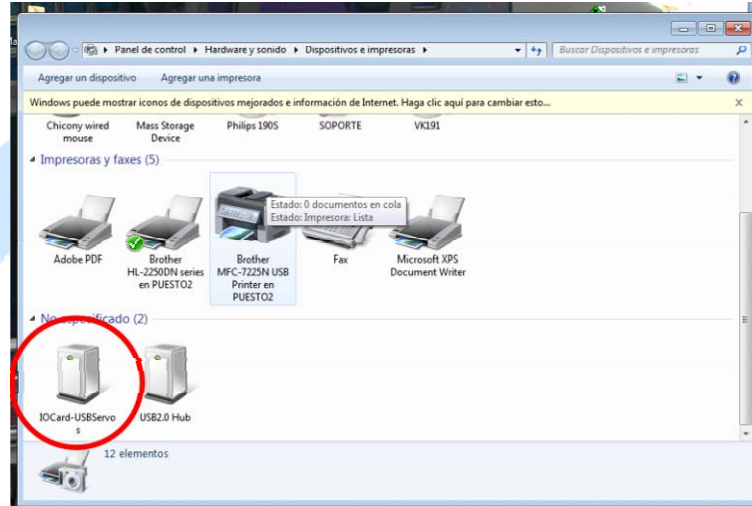


TABLA CONEXIONES POR MARCAS		
FABRICANTE	ORIGINAL	USB SERVOS
FUTABA J		
HITEC		
JR		
AIRTRONICS "Z"		

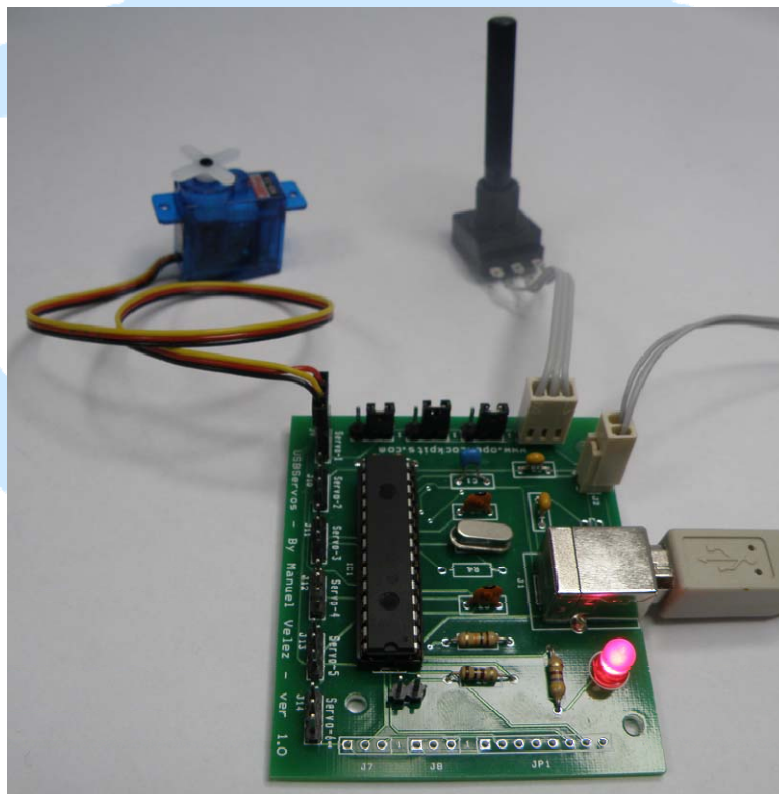
Puesta en marcha de la tarjeta:

Ya conocemos la teoría de conexión de la IOCard USB Servos, ahora pasemos a comprobarla y conectarla para ver los resultados.

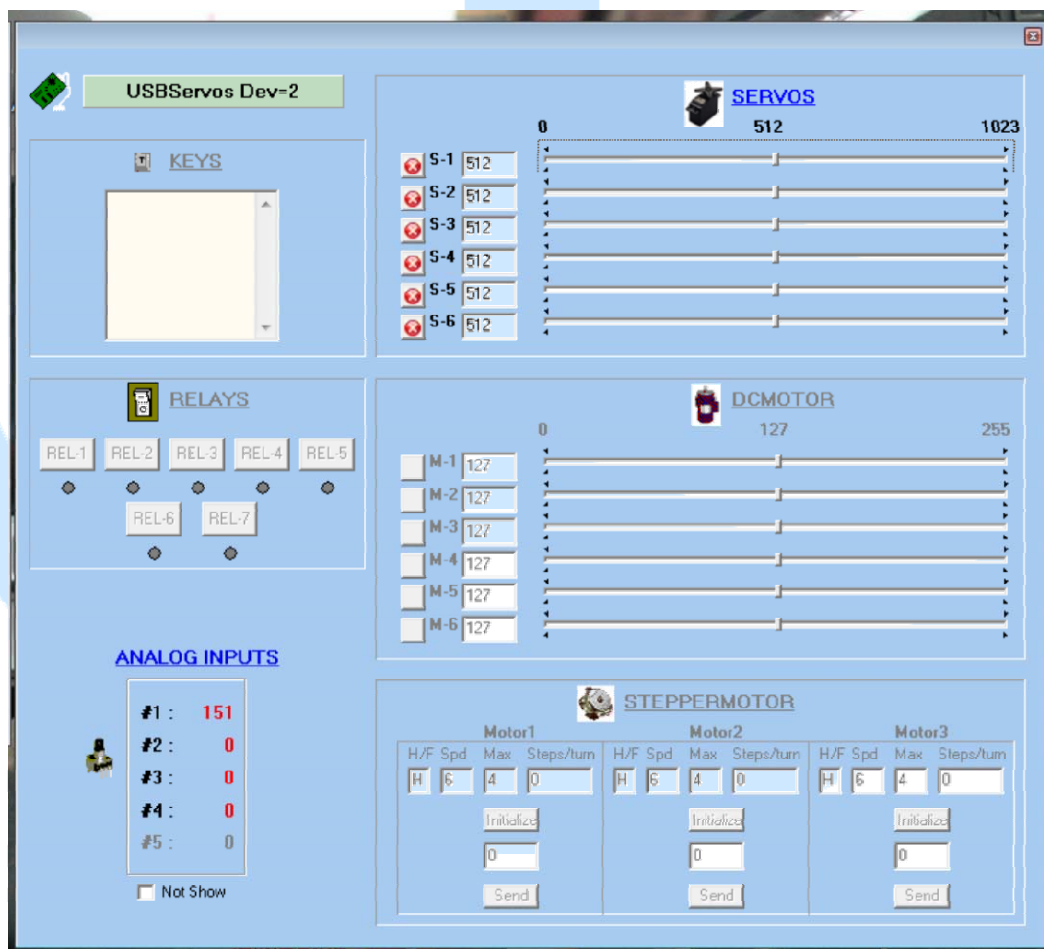
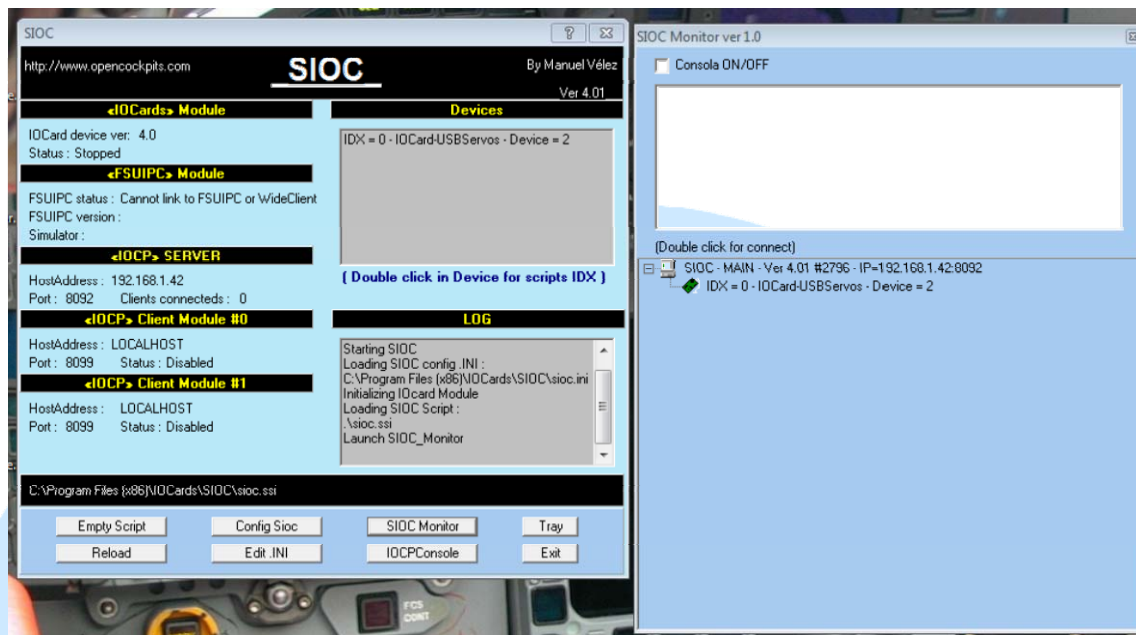
En nuestro ejercicio conectaremos la tarjeta USB Servos a un puerto USB y veremos que nuestro ordenador la reconoce.



Ha sido reconocida, así que conectamos 1 cable de alimentación de 5V a J2 para los servos, 1 potenciómetro en J3 y 1 servo de la marca HITEC (por lo que tendremos que cambiar el orden del cableado de salida del servo a rojo-amarillo-negro, +5V,S,GND) en J9, además para evitar que las lecturas de las entradas analógicas no usadas fluctúen hemos cerrado el circuito de las mismas con unos puentes:



Ahora probaremos con SIOC Monitor si nuestro montaje ha salido bien y todo funciona. Arrancamos SIOC, miramos si nos la ha reconocido y si es así pulsamos el botón Monitor, en IOCard USB Servos y chequeamos la instalación:



Como podemos observar la entrada analógica #1 muestra el valor de nuestro potenciómetro y las demás muestran el valor 0, si giramos el potenciómetro veremos cómo cambia el valor representado de más a menos o al revés (depende de cómo hagamos las conexiones de los pines), si no nos interesa el sentido de subida o bajada de los valores podemos intercambiar los pines exteriores del conector J3 o hacerlo por software mediante un script SIOC.

Probemos ahora el servo, al seleccionar la opción del servo que queremos probar, el servo se moverá automáticamente a la posición indicada en la regla de posiciones del servo (0 a 1023 ticks), es decir, monitor arranca en el centro del rango de movimiento de los servos por lo que el servo se centrará. Si deslizamos el puntero hacia los lados veremos que el servo gira hacia el sentido que le indiquemos.

Advertencia: hemos constatado que algunos servos tienen una zona en sus extremos en la que es muy fácil romper los límites físicos del radio de giro, por ejemplo en los Hitec (200° de giro más o menos desde fábrica) si bajamos del valor 190 en la escala de 0 a 1023 es casi seguro que romperemos el límite físico del servo y este girará loco sin tope alguno (360°).

Deberíamos hacer estas pruebas para todos los conectores J.

Con esta forma de comprobar los servos podemos ver los valores que toman los indicadores de los gauges en una posición determinada y esto nos viene bien para poder programarlos mediante SIOC.

Nota importante sobre la manipulación de los servos:

Mientras los servos estén conectados a la USB Servos y esta tenga alimentación el servo no puede ser manipulado manualmente ya que se romperían los engranajes o quemaríamos el motor, para poder manipular los servos estando conectados a la USB Servos, podemos mandar al servo la orden de ponerse en la posición 0 y así si podremos manipularlo manualmente. La operación manual del servo estando conectado a la USB Servos no es recomendable porque los engranajes (si son de plástico) pueden sufrir daños irreversibles.

Instalación y configuración:

Para poder hacer uso de las características de la USB Servos debemos tener instalado SIOC (última versión a ser posible), el simulador de vuelo FS, FSX, Xplane, etc.

Al final de este documento hay un listado de links para poder descargar el software necesario para poder poner en práctica este manual.

Para gestionar esta tarjeta se usan scripts, para lograr que SIOC controle la IOCard USB Servos deberemos editar la entrada en el fichero sioc.ini, de tal manera que asignemos un índice de dispositivo a cada tarjeta que instalemos, creando una entrada en el fichero sioc.ini por cada tarjeta USB Servos conectada, será en el siguiente formato:

USBServos=XX,YY

XX indica el número de índice (IDX) dentro de nuestro sistema de tarjetas.

YY indica el número de dispositivo del puerto USB al que está conectado (DEVICE).

Por ejemplo, si conectamos dos tarjetas USB Servos con los números de dispositivos 2 y 42 (estos números se pueden averiguar fácilmente con el propio programa SIOC) entonces las declararíamos en el sioc.ini de la siguiente manera:

USBServos=0,2 (como en nuestro ordenador de pruebas).

USBServos=1,42

Declaración de servos:

Para hacer referencia al número de servo exacto, debemos tener en cuenta el número de índice que le hemos asignado a cada una de las tarjetas USB Servos. Debemos definir la salida:

Var VVVV, name NNNN, Link USB_SERVOS, device DD, Output S, posL LLL, posC CCC, posR RRR

VVVV = número variable.

NNNN = nombre variable (opcional).

DD = numero de índice definido en sioc.ini (si la tenemos declarada como 0: no hace falta poner la referencia al número de Device).

S = numero de servo 1-6.

LLL = posición máxima del dispositivo a la izquierda.

CCC = posición central del dispositivo.

RRR = posición máxima del dispositivo a la derecha.

Ejemplo de definición:

Var 0001, name servo_vs, Link USB_SERVOS, Device 1, Output 5

Declaración de entradas analógicas:

Para la lectura de las entradas analógicas se deberá usar el formato siguiente:

Var VVVV, name NNNN, Link USB_ANALOGIC, Device DD, Input EE, posL LLL, posC CCC, posR RRR

VVVV = número variable.

NNNN = nombre variable (opcional).

DD = numero de índice definido en sioc.ini (si la tenemos declarada como 0: no hace falta poner la referencia al número de Device).

EE = numero de entrada analógica 1-5.

LLL = posición máxima del dispositivo a la izquierda.

CCC = posición central del dispositivo.

RRR = posición máxima del dispositivo a la derecha.

Ejemplo de definición de entrada analógica:

Var 1506, name pot_flaps, Link USB_ANALOGIC, Device 01, Input 2, posL 1, posC 128, posR 255

Ejemplos de scripts:

Servos:

Fabricaremos un variometro de 2000 fpm UP y 2000 fpm DOWN, que nos dan un total de 4000 fpm (feet per minute). Conectaremos el servo a la tarjeta USB Servos y la alimentaremos.

Ejecutaremos el Monitor de SIOC y mediante el procedimiento explicado en el capítulo de puesta en marcha de la tarjeta, comprobaremos la posición de los límites del servo (PosL, PosC y PosR, teniendo en cuenta que en nuestro ejemplo usamos un servo Hitec y que si bajamos de 190 del límite izquierdo podemos romperlo).

Para empezar crearemos un archivo de texto llamado "pruebaservo1.txt":

```
// *****
// * Config_SIOC ver 4.01 - By Manolo Vélez - www.opencockpits.com
// *****
// * FileName : pruebaservo1.txt
// * Date : 15/03/2012
```

Var 0001, name servo, Link USB_SERVOS, Device 0, Output 1, PosL 190, PosC 512, PosR 1023

Var 0002, name vs_value, Link FSUIPC_INOUT, Offset \$02C8, Length 4, Type 1

```
{
L0 = &vs_value * 0.7895
L1 = L0 * 0.2555 // 1022 positions / 4000 fpm
L2 = 511 - L1 // CENTER OF GAUGE
```



```

IF L2 > 1022 // UPPER LIMIT
{
  L2 = 1022
}
IF L2 < 1 // LOWER LIMIT
{
  L2 = 1
}
&servo = L2
}
// End of file

```

Escribiremos el script lo guardaremos y lo ejecutaremos con SIOC, arrancaremos el simulador, seleccionaremos un avión ligero (como una Mooney) despegaremos y el servo debe seguir los movimientos del indicador de actitud del avión en FS.

Entradas analógicas:

Ahora usaremos el potenciómetro conectado para mover el servo.

Para empezar crearemos un archivo de texto llamado "pruebaservo2.txt":

```

// *****
// * Config_ SIOC ver 4.01 - By Manolo Vélez - www.opencockpits.com
// *****
// * FileName : pruebaservo2.txt
// * Date : 15/03/2012

Var 0001, name servo, Link USB_SERVOS, Device 0, Output 1, PosL 190, PosC 512, PosR 1023
Var 1506, name trim, Link USB_ANALOGIC, Device 0, Input# 1, posL 1, posC 128, posR 255
{
  &servo = &trim
}
// End of file

```

Escribiremos el script lo guardaremos y lo ejecutaremos con SIOC, no nos hace falta arrancar el simulador, si giramos el potenciómetro el servo se moverá en el rango definido por el potenciómetro.

Con esto damos fin a este manual, os invitamos a leer los manuales de los demás elementos de Opencockpits y del software SIOC y os damos las gracias por confiar en nosotros.

Links de interés:

Zona de soporte para clientes:

<http://www.opencockpits.com/catalog/info/>