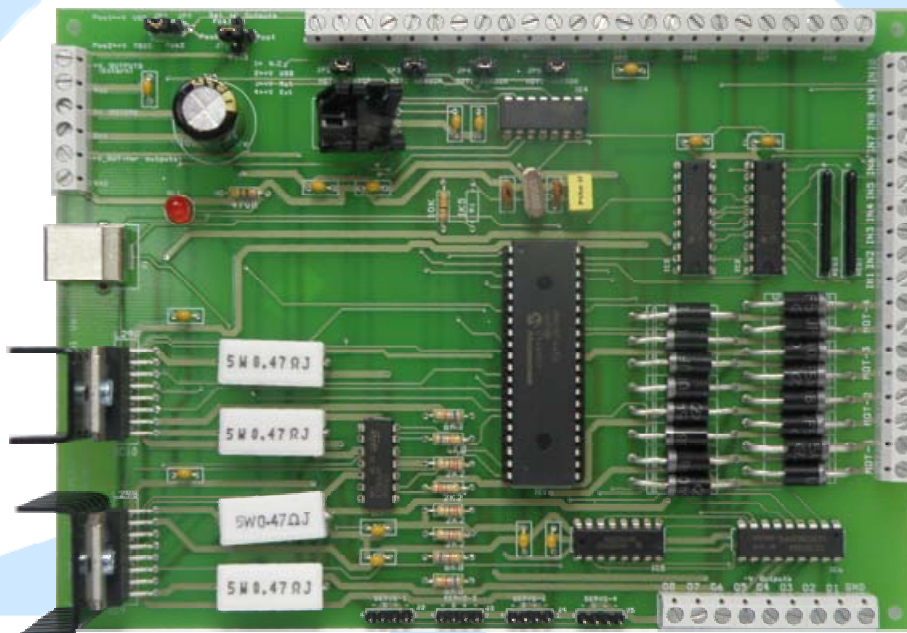




Opencockpits



Manual locard USB DCMotors Plus

Índice:

MANUAL IOCARD USB DCMOTORS PLUS.....	1
ÍNDICE:.....	2
INTRODUCCIÓN:.....	3
USB DCMOTORS PLUS:	3
ESPECIFICACIONES:	3
ESQUEMA Y MEDIDAS:.....	4
NOTAS SOBRE LA ELECTRÓNICA :	4
DESCRIPCIÓN DE LOS CONECTORES:.....	5
CONFIGURACIÓN DE JUMPERS:.....	6
CONEXIONES:.....	7
PUESTA EN MARCHA DE LA TARJETA:	8
DEFINICIÓN DE LOS ELEMENTOS DE LA TARJETA PARA SU USO CON SIOC.	8
<i>Notas de programación :</i>	9
MONTAJE DE PRUEBAS:.....	10
<i>Definición en Sioc.ini.</i>	11
<i>Definición del script de pruebas.</i>	12
CARGA DEL SCRIPT Y PRUEBAS:.....	12
PRUEBA DE LA USB DCMOTORS PLUS:.....	13
LINKS DE INTERÉS:.....	14

Introducción:

La USB DCMotors Plus ha sido diseñada para poder interactuar con distintos motores, salidas y entradas. Puede ser usada para la motorización de mandos de gases, sistemas de fuerza controlada (force feedback) y en distintos tipos de automatismos. Esta tarjeta se conecta a través del puerto USB del ordenador y se gestiona través del protocolo IOCP usando como mínimo la versión 4.2 de Sioc.

USB DCMotors Plus:

A diferencia de su hermana menor, la USB DCMotors, esta tarjeta permite el uso de motores de mucha más potencia, incluye la posibilidad de usar adicionalmente servo-motores, así como el uso de salidas y entradas tanto analógicas como digitales.

Además añade en el firmware un sistema de autoposicionado con el fin de gestionar sistemas de autoposicionamiento automáticos sin necesidad de programación alguna.

La tarjeta además incluye auto-sense de los motores. Es decir, a través de entradas analógicas la tarjeta puede medir en todo momento la intensidad que pasa por las bobinas de los motores con lo que el usuario puede saber qué esfuerzo están realizando los motores. Esto es muy útil para detección de final de carrera en los movimientos y también nos permite actuar dependiendo de la fuerza que se genera en los motores.

En un futuro, sin necesidad de cambiar de electrónica, sólo con un firmware diferente, se podrán gestionar también motores paso a paso de gran potencia.

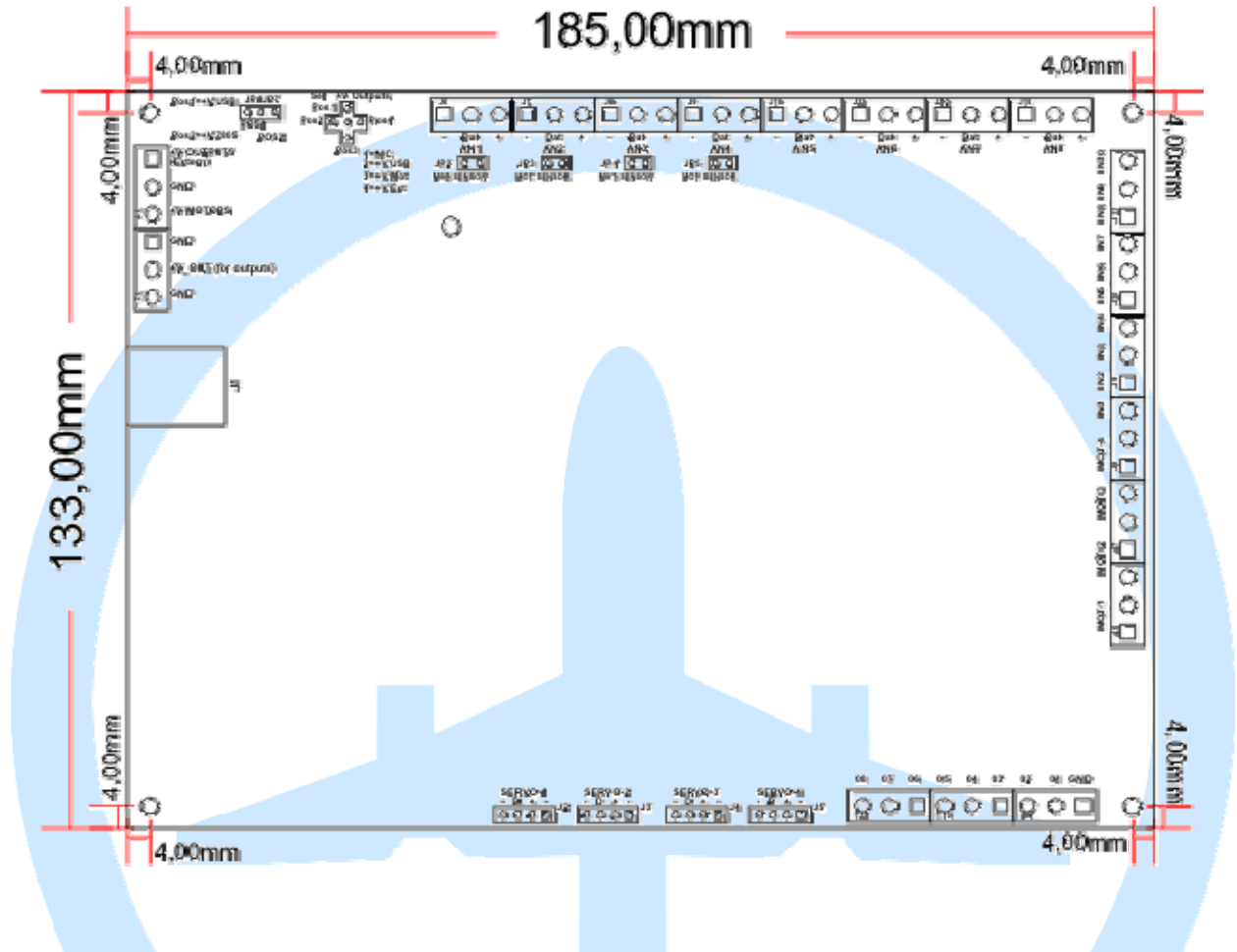


Especificaciones:

- Corriente de entrada para motores hasta 48 Voltios.
- Hasta 4 motores DC. 2 amperios constantes, 2.5 Amp no constantes, 3 amperios de picos.
- Control de velocidad PWM de los motores (8 bits).
- Control del sentido de los motores.
- Protección de sobrettemperatura.
- Hasta 4 servo-motores tipo RC con control de 10bits de resolución.
- 8 Salidas digitales de hasta 50 V/500mA
- Fuente de alimentación interna de 5 voltios seleccionable para ser usada con servo motores y/o salidas digitales.
- Segunda fuente de alimentación opcional para salidas digitales.
- 8 Entradas analógicas de 10bits internas / 8 bits externas.
- Posibilidad de auto-sensado de corriente de cada motor mediante entradas analógicas.
- Sistema de auto-posicionamiento de motores (max. 3 motores) mediante el uso de entradas analógicas (usando potenciómetros, sensores magnéticos, etc.).
- Hasta 10 entradas digitales.

Esquema y medidas:

- J1 = CONECTOR USB.
- JP1, JP6, JP7 = CONECTORES 3 PINES.
- JP2...JP5 = CONECTORES 2 PINES.
- J2 (Servo1)...J5 (Servo4) = CONECTORES 4 PINES.
- J6 (An-1)...J13 (An8), T1...T11 = CONECTORES DE TORNILLO.



Notas sobre la electrónica :

Dependiendo de los motores y el uso que se les dé, puede que la tarjeta sufra calentamientos, por eso es necesaria una buena ventilación de la misma e incluso para usos prolongados en potencia, sería aconsejable un ventilador.

Algunos motores DC pueden generar bastante ruido eléctrico. Para evitarlo se aconseja el uso de condensadores "antiparasitarios" e incluso si se detecta gran cantidad de fluctuaciones electrónicas en las entradas, es aconsejable el uso de condensadores electrolíticos en los conectores de las entradas analógicas (entre GND y positivo, sobre 10mF, 63V o mayor) para así evitar problemas en la lectura de las entradas analógicas e incluso si los picos de ruido eléctrico son grandes, evitar la desconexión automática de la tarjeta del puerto USB del ordenador.





La tarjeta no dispone de dispositivo de protección de corto-circuitos y sobrecarga, sólo dispone de protección contra sobre-calentamiento, con lo que sería aconsejable usar algún tipo de protección (fusible o protector de sobre-tensión).

Descripción de los conectores:

- J1 = Conector USB, permite la conexión al ordenador directamente, en el momento de conectarse el ordenador reconocerá la tarjeta e instalará el dispositivo.
- J2...J5 = Conectores para los servo-motores. El conector es de 4, para conectar 3 a la izquierda o 3 a la derecha.
Pines 2, 3 y 4 se usan para los servos y gauges de Opencockpits y compatibles (2 Rojo/+, 3 Amarillo/Dato y 4/Negro -).
Pines 1, 2 y 3 son para servos marca FUTABA, HITEC, JR y compatibles. De todas maneras es muy recomendable revisar los cables:

COLOR	VALOR
Negro/ marrón	GND
Blanco/ amarillo/naranja	DATO
Rojo o azul	+

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

- J6...J13 = Conectores de tornillo para las entradas analógicas AN1...AN8:

GND.

Datos.

+5V.

Las entradas AN5, AN6 y AN7 pueden ser usadas para el auto-posicionamiento de los motores MOT-1, MOT-2 y MOT-3.

- T1 y T2 = Conector de tornillo para:

+V outputs (pin 1 y pin 5), positivo alimentación externa para salidas. Conectar el negativo en un GND y el positivo en uno de los pin. El otro se usará como común a las salidas, pues en este caso el común de las salidas es el positivo y no el negativo.

GND (pines 2, 4 y 6) = Negativo para las fuentes de alimentación.

+V motors (pin 3) = Positivo de la alimentación para motores. El negativo puede conectarse a cualquier GND.

- T3, T4 y T5 = Conectores de tornillo para:

MOT-1...MOT-4, conexión de los polos de los motores 1 al 4.

- T5...T8 = Conectores de tornillo para:

IN1...IN10, entradas analógicas. Si están abiertas (sin conexión) se devolverá un 0 en la variable de software. Si se conectan a GND se devolverá el valor 1 a la variable correspondiente.

- T9, T10 y T11 = Conectores de tornillo para:

GND, punto de conexión extra a GND para las entradas.

O1...O8, salidas digitales. En estos terminales se conectan los negativos de cada salida. Importante tener en cuenta que el común para las salidas es el positivo (+V_OUT). Si en la variable software correspondiente se pone un 1, entre el positivo de las salidas y cada terminal de salida pasará la corriente.

Configuración de jumpers:

JP1-JP7:

Posición 1, la tarjeta toma la corriente de 5 voltios del propio puerto USB. Esto es útil para realizar pruebas o cuando no se vayan a usar servo-motores y las salidas sean de un consumo pequeño.

Posición 2, la corriente de 5 voltios es tomada de la alimentación interna, que a su vez se genera a partir de la alimentación de los motores.

Jumper Selector +V para Salidas :

Pos 1, posición no usada.

Pos 2, la salida usará la alimentación de 5 voltios (que a su vez es seleccionada por los jumpers JP1-JP7).

Pos 3, la salida usará la alimentación de los motores.

Pos 4, la salida usará una alimentación externa diferente.

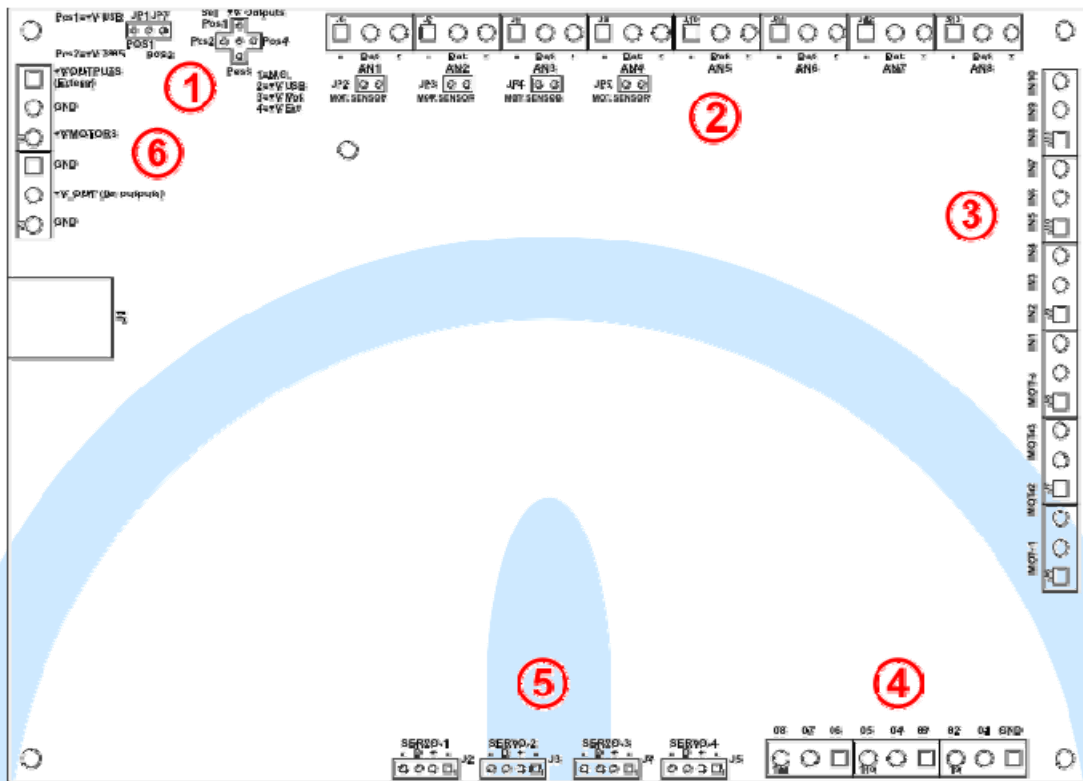
Conectores JP2 a JP5 :

Abierto, las entradas AN1 a AN4 funcionan como entradas analógicas normales.

Cerrado, las entradas AN1 a AN4 están conectadas al sense de los motores MOT-1, MOT-2, MOT-3 y MOT-4 respectivamente.

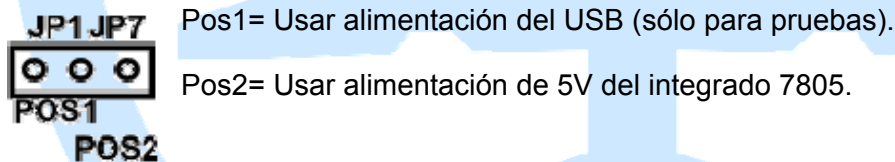
Conexiones:

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

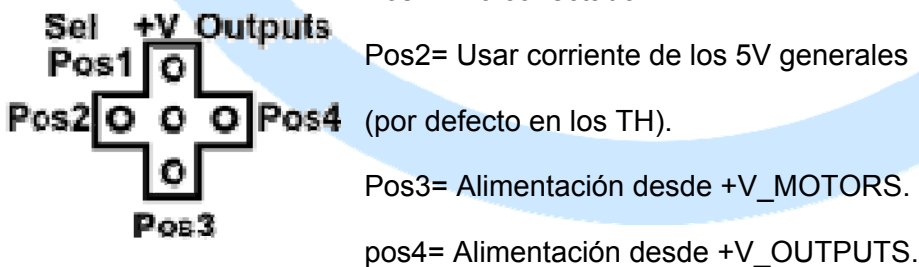


1.- Alimentación de la placa y componentes:

Selector de alimentación para los 5V generales:



Selector de alimentación para las salidas:



2.- 8 entradas analógicas.

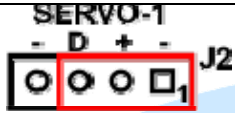

Las primeras 4 con jumper cerrado miden la corriente de los motores 1 al 4 (sense).

Las siguientes 3 con el software activado se usan para autoposicionar los motores 1 al 3.

3.- Entradas digitales 1 a 10, se conectan cortocircuitando con cualquier GND.

4.- Salidas digitales. Se conecta el negativo a estos conectores y el positivo al conector general V_OUT (for outputs) en la regleta de alimentación.

5.- Conexiones de Servomotores. Conector doble preparado para los dos tipos habituales, poner atención cuando se conecten los servos.

PINES (1, 2 y 3): Futaba, Hitec, JR, etc.	PINES (2, 3 y 4): Opencockpits servos and gauges.
	

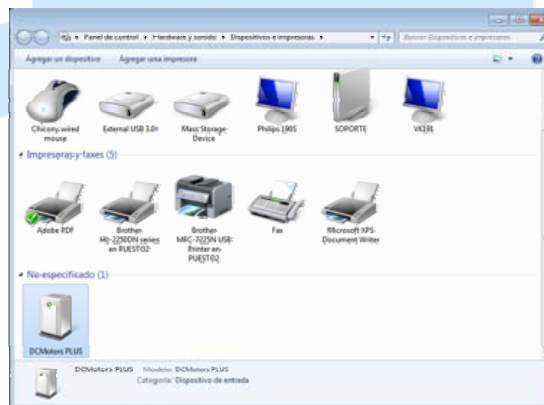
6.- +V_OUTPUTS: Conectar si queremos alimentación específica para las salidas (No usar en el Throttle).

+V_MOTORS: Alimentación general de motores y fuente interna (7805).

+V_OUT: (salida con la misma corriente que +V_OUTPUTS), común positivo alimentación de salidas.

Puesta en marcha de la tarjeta:

Ya conocemos la placa IOCard USB DCMotors Plus, pasemos ahora a conectarla al ordenador, comprobarla y probar su funcionamiento. Al conectarse al ordenador será inmediatamente reconocida como dispositivo USB.



Nota importante: nunca se deben de conectar o desconectar los motores estando la tarjeta conectada al puerto USB y a la alimentación.

Definición de los elementos de la tarjeta para su uso con Sioc.

La programación y configuración de la tarjeta se realiza mediante el programa SIOC versión 4.2 o superior y se prueba mediante IOCPConsole.

La definición de todos los elementos de la tarjeta para su uso con Sioc es :


```
// *****
// * Definición de USB DCMotors Plus - By Manuel Velez - www.opencockpits.com
// *****

Var 0001, name Servo1, Link IOCARD_SERVO, device X, Output 1, PosL 0, PosC 512, PosR 1023
...
Var 0004, name Servo4, Link IOCARD_SERVO, device X, Output 4, PosL 0, PosC 512, PosR 1023
Var 0005, name Motor1, Link IOCARD_MOTOR, device X, Output 0, Aceleration 0
...
Var 0008, name Motor4, Link IOCARD_MOTOR, device X, Output 3, Aceleration 0
Var 0009, name output1, Link IOCARD_OUT, device X, Output 0
...
Var 0016, name output8, Link IOCARD_OUT, device X, Output 7
Var 0017, name autopos1A, Link IOCARD_CONFIG, device X, Output 1
Var 0018, name autopos1B, Link IOCARD_CONFIG, device X, Output 2
Var 0019, name autopos2A, Link IOCARD_CONFIG, device X, Output 3
Var 0020, name autopos2B, Link IOCARD_CONFIG, device X, Output 4
Var 0021, name autopos3A, Link IOCARD_CONFIG, device X, Output 5
Var 0022, name autopos3B, Link IOCARD_CONFIG, device X, Output 6
Var 0023, name input1, Link IOCARD_SW, device X, Input 0
...
Var 0032, name input10, Link IOCARD_SW, device X, Input 10
Var 0033, name analog1, Link IOCARD_ANALOGIC, device X, Input 1, PosL 0, PosC 128, PosR 255
...
Var 0040, name analog8, Link IOCARD_ANALOGIC, device X, Input 8, PosL 0, PosC 128, PosR 255
```

Notas de programación :

Las entradas se activan con 1 y desactivan con 0.

Las salidas se activan con 1 y desactivan con 0.

Las entradas analógicas van de 0 a 255.

Los servo-motores van de 0 a 1023 (en la posición 0 el servo-motor queda sin alimentación y tiene movimiento libre: el servo puede moverse manualmente o con un motor pero con cuidado para no dañar los engranajes).

Los motores tienen dos formas de funcionamiento :

1. Por defecto o si se ponen a 0 las variables de configuración A y B de cada motor, el motor se controla por software mediante la variable de motor declarada en el script, de tal manera que con un valor de 0 a 127 (0 parado, 127 máxima velocidad) irá en un sentido y con un valor de 128 a 255 (128 parado, 255 máxima velocidad) irá en el otro sentido.

Rango de Valores	Velocidad
0	Parado
1→127	mín→max (un sentido)
128	Parado
129→255	mín→max (sentido contrario)

2. Si las variables de configuración A o B tienen un dato, el motor correspondiente se pone en modo auto-posicionamiento. Así los motores son controlados por la tarjeta, intentando acelerar y girar para hacer llevar el valor de la entrada analógica respectiva, al valor objetivo seleccionado en la variable del motor. El motor cuando llegue al punto se parará automáticamente. La variable de configuración A define la aceleración con que el motor arrancará y frenará, la variable B es la sensibilidad de ajuste a la zona de llegada. A veces tendremos que jugar con estos valores hasta encontrar el funcionamiento deseado.

Montaje de pruebas:

Usaremos un potenciómetro, un motor de corriente continua (DC, que podremos coger de cualquier electrodoméstico como una impresora o un juguete), un led, un interruptor y un servo.

Conectaremos nuestros elementos de la siguiente manera:

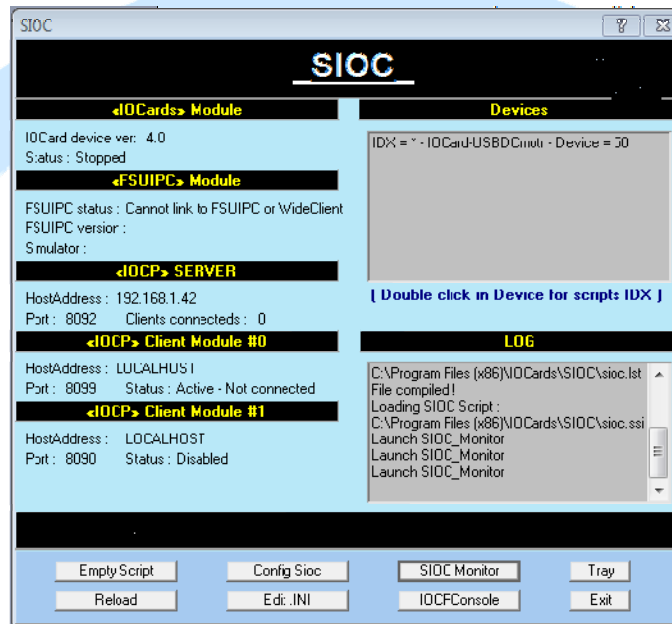


Nuestro motor de pruebas y el servo funcionan a 5V, por lo que alimentaremos la tarjeta USB-DCMotors Plus a 5V desde un alimentador externo para las pruebas, es decir, JP1 cerrado en POS2 y JP6 en Pos3. El motor lo conectamos a la salida MOT-1 pines 1 y 2 (el sentido de giro puede cambiarse bien vía SIOC, bien vía hardware cambiando la conexión de los cables del

motor). El potenciómetro lo hemos conectado a la entrada AN1 (J6 con JP2 abierto), el led a la salida O1 y al positivo común, el interruptor a IN1 y el servo a Servo-1 (en la posición 2-3-4 porque es un servo configurado con cableado Opencockpits).

Definición en Sioc.ini.

Arrancamos el SIOC (versión 4.2 o superior), observamos la información que da la ventana superior derecha y vemos si está bien configurada nuestra tarjeta. Si lo está, aparecerá con un número de IDX reconocido (si no está bien declarada aparecerá un símbolo *), con su nombre y el puerto USB donde está conectada (Device = XXX).



Si no está bien configurada, vamos a ello:

Pulsamos el botón EDIT .INI, se abrirá el editor de textos con nuestro archivo de configuración sioc.ini y buscamos la sección [IOCard Master] y en ella veremos como declarar una USB DCMotors Plus:

```
[ MASTER=(Indice device),(Tipo),(Número de tarjetas),(Número device) ]
[ tipo = 14 : Tarjeta USBDCmotorPLUS ]
```

Es decir:

```
MASTER=YY,14,1,XX
[USB DCMotors Plus]
```

YY = número con el que identificaremos la tarjeta en nuestros scripts (IDX en Sioc = DEVICE en script).

14 = tipo de tarjeta, en este caso una USB DCMotors Plus.

1 = número de tarjetas USB DCMotors Plus. Este valor es fijo y no se puede cambiar.

XX = número del puerto USB donde se ha conectado la tarjeta.

Recomendamos usar el mismo número de IDX que el usado para declarar el tipo de tarjeta que es, siempre que sólo tengamos una sola tarjeta del mismo tipo conectada, en el caso de que sean más, da igual el número de IDX que se ponga pero no puede estar repetido. En nuestro caso declaramos la USB DCMotors Plus como:

```
MASTER=14,14,1,271
[USB DCMotors Plus]
```

271 = puerto USB de nuestro ordenador de pruebas, en su caso el número XX de su pantalla de Sioc.

Guardamos y salimos, le damos al botón RELOAD y nuestra tarjeta ya estará correctamente configurada para poder gestionarla con un script.

Es importante fijarse bien en el IDX asignado, que en nuestro ejemplo es el 14, para poder configurar y usar la tarjeta, los potenciómetros y los motores acoplados a ella. Ahora pasemos a crear el script de pruebas.

Definición del script de pruebas.

Para este capítulo nos basaremos en la definición de los elementos de la tarjeta. Usaremos de momento sólo las definiciones de los elementos que vamos a probar realmente para no hacer este manual excesivamente pesado, los demás elementos se usan igual.

Con un editor de textos sencillo como notepad crearemos un script llamado Prueba_USB_DCMotors_Plus.txt:

```
// *****
// * Definición de USB DCMotors Plus - By Manuel Velez - www.opencockpits.com
// *****
// * FileName : Prueba_USB_DCMotors_Plus.txt
// * Date : 12/07/2012
Var 0001, name Servo1, Link IOCARD_SERVO, device 14, Output 1, PosL 0, PosC 512, PosR 1023
// esta salida es la SERVO-1
Var 0002, name Motor1, Link IOCARD_MOTOR, device 14, Output 0, Aceleration 0
// esta salida es la MOT-1
Var 0003, name output1, Link IOCARD_OUT, device 14, Output 0
// esta salida es la O1
Var 0004, name input1, Link IOCARD_SW, device 14, Input 0
// esta entrada es la IN-1
Var 0005, name analog1, Link IOCARD_ANALOGIC, device 14, Input 1, PosL 0, PosC 128, PosR 255
// esta entrada analógica es la AN-1
// END OF FILE
```

Lo guardamos y pasamos a cargarlo en Sioc, con este sencillo script podemos probar las características de la USB DCMotors Plus.

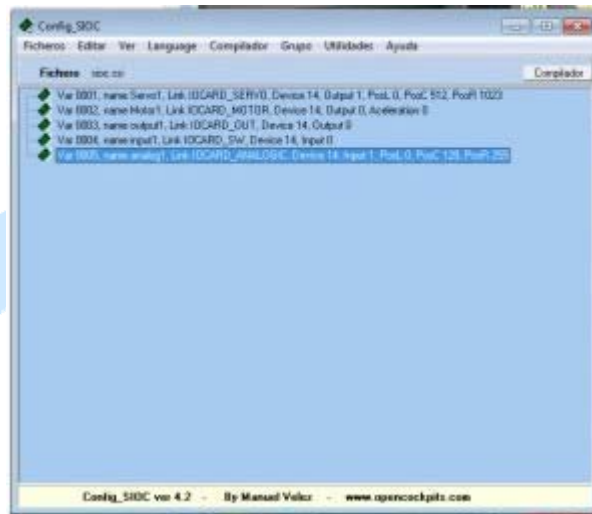
Carga del script y pruebas:

En este apartado vamos a cargar el script de pruebas que acabamos de realizar en Sioc, para ello arrancamos Sioc, pulsamos el botón Edit .INI, aparecerá el sioc.ini en el editor de textos, buscamos la sección "[***** SIOC *****]" y dentro de ella la subsección "[Fichero de configuracion]/[Configuration File]" y sustituimos la línea existente por la siguiente:

```
CONFIG_FILE=.\Prueba_USB_DCMotors_Plus.txt
```

Guardamos los cambios y cerramos el editor de textos. Nos aseguramos de que el script que hicimos en el capítulo anterior y hemos referenciado en sioc.ini está en el mismo directorio que Sioc.exe.

Como aún tenemos Sioc ejecutándose, primero pulsamos el botón Reload para actualizar los cambios hechos en sioc.ini y después pulsamos el botón Config Sioc, que abrirá otra ventana y esta ventana contendrá el script que hemos declarado en sioc.ini, si no contiene errores.

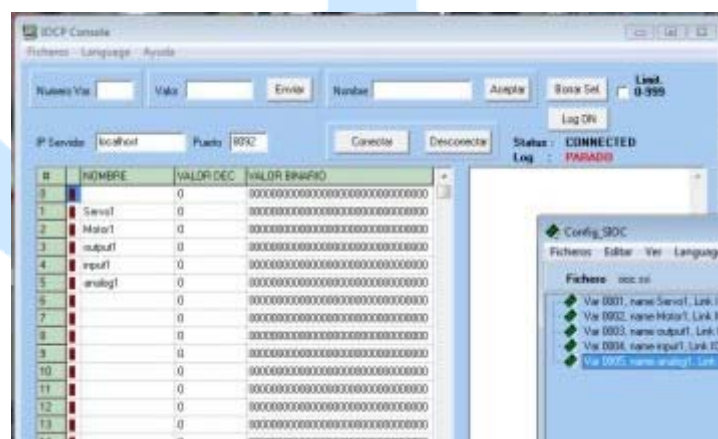


Si no hay errores cerramos el Config Sioc y continuaremos con la prueba de la USB DCMotors Plus, si hay errores, nos indicará los que ha encontrado y deberemos reparar el script para solucionarlos.

Si todo ha ido bien, conectaremos los elementos a la tarjeta y la alimentaremos de forma externa (aunque la carga que se le va a meter es pequeña pero no conviene recalentar los componentes de la placa) y después conectaremos la tarjeta al ordenador que junto con Sioc la reconocerán y estaremos listos para empezar a hacer pruebas.

Prueba de la USB DCMotors Plus:

Arrancamos Sioc, pulsamos el botón IOCPConsole y aparecerá la siguiente ventana:



Pulsamos Log ON y Conectar, las variables presentarán los valores reales en ese momento:

Servo1: Aquí podemos poner la posición que queramos que coja el servo, si tecleamos en valor la posición 200, el servo se irá a esa posición, si introducimos la posición 800 irá a esa posición, etc. Si introducimos el valor 0, el servo quedará destrabado y podremos moverlo libremente con las manos (con sumo cuidado de no romper los límites físicos de los engranajes).

Motor1: mostrará el valor 0 (parado), si le enviamos un valor entre 1 y 127 girará en un sentido, si mandamos el valor 128 ó 0 se parará de nuevo y si enviamos un valor entre 129 y 255 girará en el sentido contrario. La velocidad del motor incrementará al igual que incrementa el valor que se le envía (recordar la tabla de la página 10).

Output1: mostrará el valor 0 inicialmente y si le enviamos el valor 1, el led conectado en O1 se encenderá y con un 0 el led se apagará. (Recordar que los leds necesitan una resistencia para no quemarse y las salidas O tienen el positivo común).

Input1: mostrará el estado actual del interruptor que está conectado a IN1, 0 si el interruptor está abierto y 1 si está cerrado.

Analog1: mostrará la lectura del potenciómetro conectado en AN1, si giramos el eje del potenciómetro los valores mostrados variarán al unísono.

La aplicación de la tarjeta a su uso con los simuladores es exactamente igual que con otras IOCards, por ello les invitamos a leer los manuales de los demás elementos de Opencockpits y del software SIOC y les damos las gracias por confiar en nosotros.

Con esto damos fin a este manual.

Links de interés:

Zona de soporte para clientes:

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

