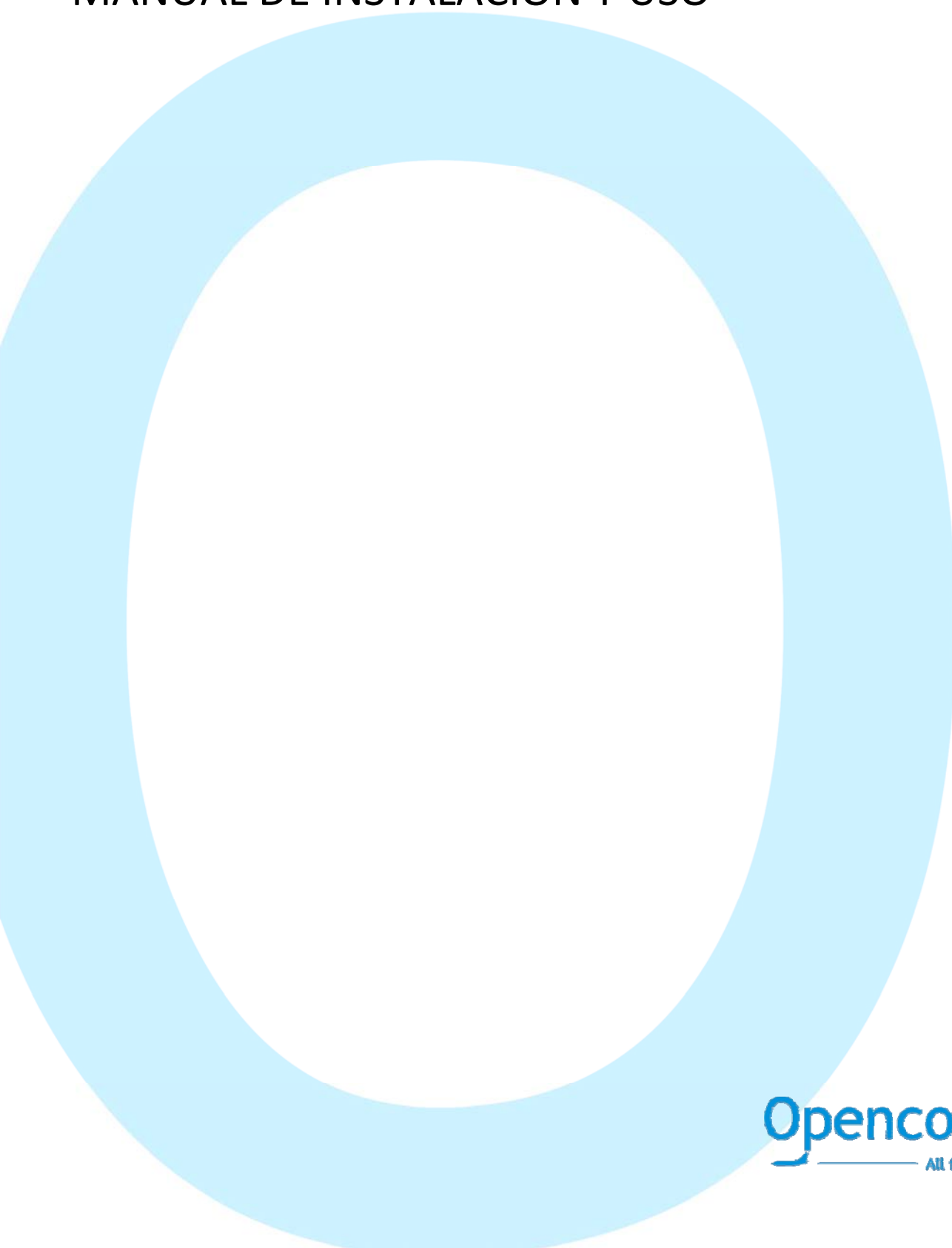


OPENCOCKPITS IOCards USBDCMotors

MANUAL DE INSTALACION Y USO



## INTRODUCCION

La USBDCMotors puede gestionar hasta 6 motores de corriente continua de hasta 1 Amperio (incluso con picos de 2 Amperios, pero si los motores van a consumir 1 Amperio durante tiempo, es mejor montarle un disipador de calor) y 5 entradas analógicas.

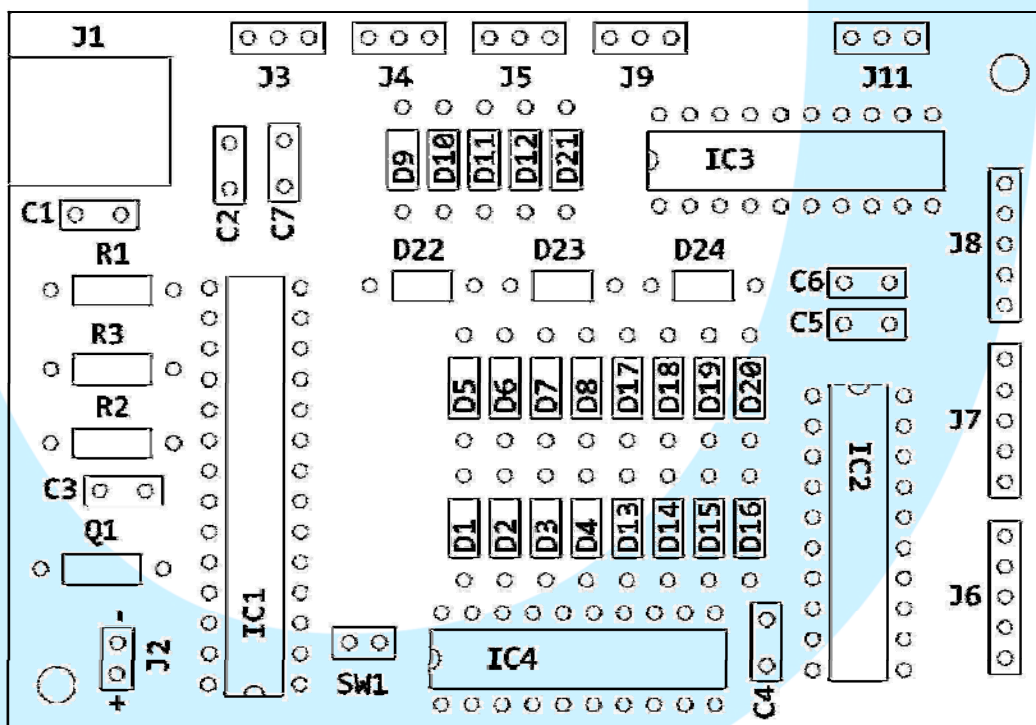
Esta tarjeta se conecta y configura a través del puerto USB del ordenador y se gestiona el controlador a través del protocolo IOCP.

Tanto el sentido del motor como la velocidad del mismo se pueden controlar mediante PWM (Pulse Width Modulation).

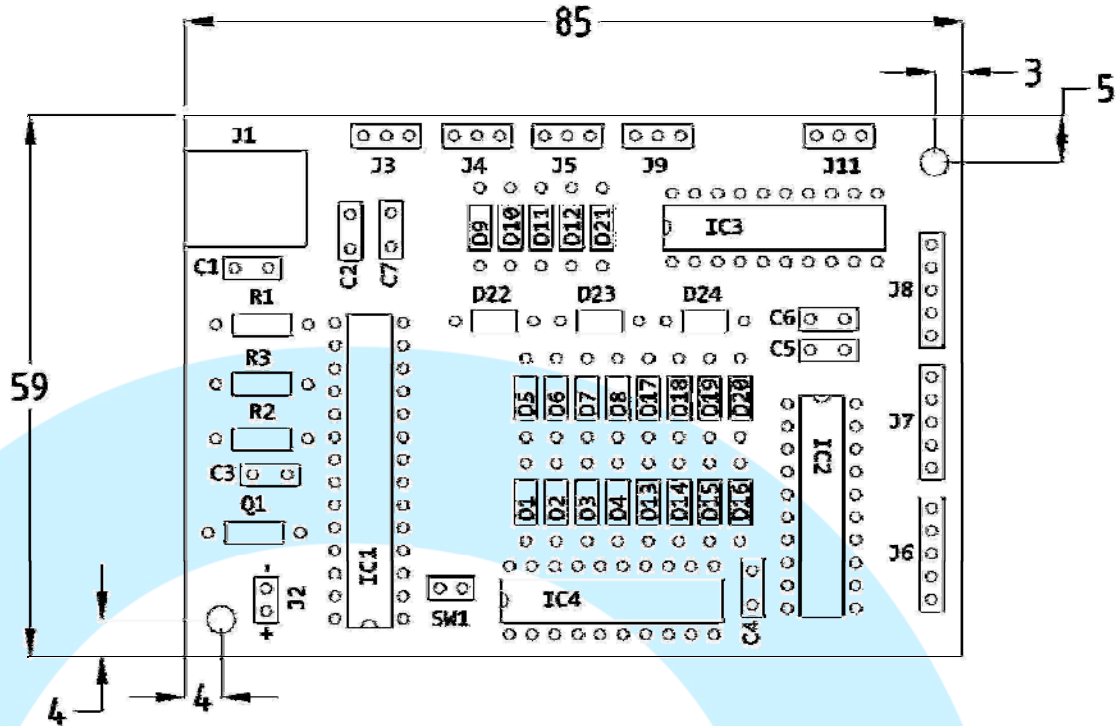
El diseño del circuito esta basado en la USBStepper a la que le hemos cambiado algunos componentes y el código del microcontrolador.

## LISTADO DE COMPONENTES

- C1, C4, C5, C6, C7 = CONDENSADOR 0.1mF
- C2, C3 = CONDENSADORES 22Pf
- D1...D24 = DIODOS 1N4007
- IC1 = MICROCONTROLADOR 16C745
- IC2, IC3, IC4 = INTEGRADO L293E
- J1 = CONECTOR USB
- J2 = CONECTOR ALIMENTACION 2 PINES
- J3, J4, J5, J9, J10, J11 = CONECTORES 3 PINES
- J6, J7, J8 = CONCTORES 5 PINES
- Q1 = CRISTAL DE CUARZO 6MHZ
- R1 = RESISTENCIA 100R
- R2 = RESISTENCIA 10K
- R3 = RESISTENCIA 1K5
- SW1 = RESET (CONECTOR 2 PINES)



## DIMENSIONES PRINCIPALES

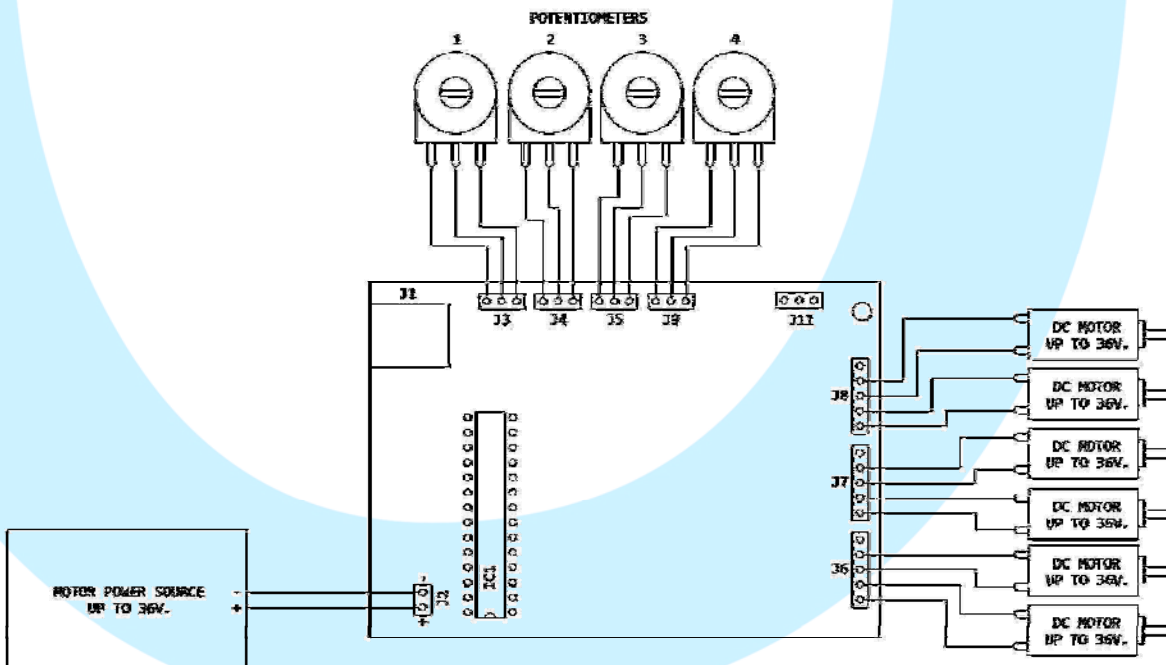


## DESCRIPCION DE LOS CONECTORES:

- J1 = Conector USB, permite la conexión al ordenador directamente, en el momento de conectarse el ordenador reconocerá la tarjeta e instalará el dispositivo.
- J2 = Conector de alimentación, para alimentar los motores, se le debe suministrar el voltaje que precisemos enviar a los motores. El voltaje es único para los seis motores.
- J3, J4, J5 y J9 = Conectores para entradas analógicas (ver dibujo de conexiones)
- J6, J7 y J8 = Conectores para los motores, usaremos 2 pines consecutivos para cada motor, dejando el pin5 libre (ver dibujo de conexiones).

## CONEXIONES:

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



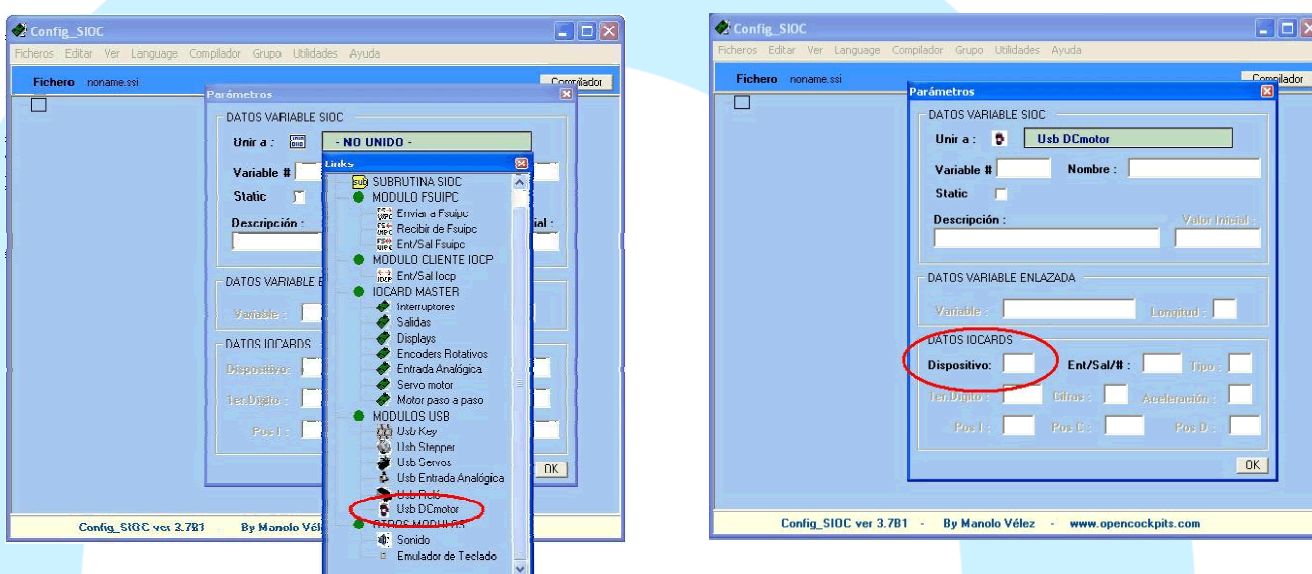
## USO DE LA TARJETA:

Para el manejo de la tarjeta, usaremos básicamente el lenguaje propio de programación de las IOCards, el SIOC.

El primer paso sería editar el fichero sioc.ini, para indicarle a SIOC que tarjeta esta activa, para ello editaríamos la siguiente línea:

USBDCmotor=XX,YY

Donde XX representa el número de orden de la tarjeta dentro de nuestro sistema, se numera para poder acceder a ella con el parámetro Device (ver imagenes siguientes) en la declaración de variables en SIOC. En el caso de que solo usemos esta tarjeta, podemos dejar el parámetro en 0, ya que esto indica a sioc, que solo tenemos conectada dicha tarjeta. Si usamos el 0 SIOC lo interpreta como la primera tarjeta conectada y no tenemos que definirlo en la declaración de datos, para ello dejaríamos el campo Dispositivo en blanco.



YY, a su vez, representa el identificador del puerto USB en donde esta conectada la tarjeta. Esta información la podemos averiguar ejecutando la utilidad USBCheck o el propio sioc.exe (en la ventana superior derecha), y nos indicara dicho número. Este identificador es adecuado ponerlo.

Ejemplos:

- 1 sola tarjeta en el puerto 43

USBDCmotor=0,43

- 2 tarjetas, una en el puerto 27 y otra en el 32

USBDCmotor=0,27

USBDCmotor=1,32

Si queremos hacer uso de las entradas analógicas en esta tarjeta, deberemos configurar asimismo en el sioc.ini la siguiente línea:

USBAnalogic=XX,YY

Usando los mismos números usados para configurar la propia tarjeta, por ejemplo, para la tarjeta conectada en el puerto 43:

USBAnalogic=0,43

O las dos conectadas en los puertos 27 y 32:

USBAnalogic=0,27

USBAnalogic=1,32

Fijese que estamos usando los mismos números que en la definición de la propia tarjeta, esto nos asegura que tendremos acceso a las entradas analógicas en cada una de las tarjetas.

Una vez configurado el sioc.ini, ya podemos tener acceso a las tarjetas a través de SIOC, para ello deberemos tener un script adecuado al uso que queramos hacer de la tarjeta. Vea el ejemplo siguiente que nos proporciona un posicionador para las palancas de gases en función a la posición de las palancas en el Flight Simulator (autothrottle), adaptado para Project Magenta:

```

Var 0000, name INICIO, Value 0
{
  &CONTROL = TIMER 999 ,0 ,2
}

Var 0011, name OBJ_N1_E1

Var 0012, name MARGEN_ERROR, Value 5

Var 0013, name APROX, Value 20

Var 0014, name VEL_NORMAL, Value 100

Var 0015, name VEL_APROX, Value 30

Var 0100, name N1_E1, Link FSUIPC_IN, Offset $0898, Length 2
{
  &OBJ_N1_E1 = &N1_E1 / 64
}

Var 0101, name EJE_MOTOR_1, Link FSUIPC_INOUT, Offset $088C, Length 2

Var 0102, name TOGA_PM, Link FSUIPC_INOUT, Offset $050A, Length 2

Var 0103, name AT_DISC_PM, Link FSUIPC_INOUT, Offset $050A, Length 2

Var 0200, name SW_TOGA, Link IOCARD_SW, Input 10, Type P
{
  IF &SW_TOGA = 1
  {
    &TOGA_PM = SETBIT 0
  }
  ELSE
  {
    &TOGA_PM = CLEARBIT 0
  }
}

Var 0201, name SW_AT_DISC, Link IOCARD_SW, Input 11, Type P
{
  IF &SW_AT_DISC = 1
  {
    &SW_AT_DISC = SETBIT 2
  }
  ELSE
  {
    &SW_AT_DISC = CLEARBIT 2
  }
}

```

Var 0001, name MOTOR\_1, Link USB\_DCMOTOR, Device 0, Output 1

Var 0006, name ANALOG\_E1, Link IOCARD\_ANALOGIC, Device 0, Input 1, PosL 0, PosC 127, PosR 255

Var 0999, name CONTROL, Link SUBROUTINE

```

{
  C1 = TESTBIT &TOGA_PM ,0
  C2 = TESTBIT &AT_DISC_PM ,2
  C0 = C1 OR C2
  IF C0
  {
    L0 = &OBJ_N1_E1 - &ANALOG_E1
    L1 = 0
    IF L0 < 0
    {
      L1 = 128
    }
    L0 = ABS L0
    L2 = &VEL_NORMAL + L1
    IF L0 <= &APROX
    {
      L2 = &VEL_APROX + L1
    }
    IF L0 <= &MARGEN_ERROR
    {
      L2 = 0
    }
    &MOTOR_1 = L2
  }
  ELSE
  {
    &EJE_MOTOR_1 = &ANALOG_E1 / 64
  }
}

```

Basicamente lo que hace el script <sup>1</sup>, es comparar la posición de la palanca del FS (offset \$0880) con la posición del eje analógico (var 0006) y esa relación (mayor o menor) hace que el motor gire en un sentido u otro para igualar esos valores, además incluye dos velocidades (aproximación y normal), para que cuando se este acercando a su objetivo disminuya la velocidad.

Podemos arrancar el Flight Simulator, el sioc.exe y si todo esta bien conectado y configurado, las palancas de gases (mecánicamente construidas) deberían seguir los movimientos de las palancas (virtuales) del Flight Simulator.

**Nota:**

Los programas de software, circuitos y contenidos publicados en este documento y en nuestra web, son propiedad de sus desarrolladores, quienes NO dan su consentimiento para su uso con fines lucrativos o comerciales salvo autorización expresa y por escrito.

El software y el contenido publicado, así como cualquier código desarrollado puede ser distribuido cuantas veces se quiera y por los medios que se desee, sin necesidad de obtener autorización por escrito, siempre y cuando en la publicación se cite al autor y la fuente de donde proviene

[www.opencockpits.com](http://www.opencockpits.com)

<sup>1</sup> Nota del autor: El script se ha escrito específicamente para el manual, no esta testado. Para notificar si existe algún error, por favor indíquelo a support@opencockpits.com