

## **IOCard Manual**

# **USBDCmotors**

**By: Manuel Vélez**

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

**ver 1.0**

## **INTRODUCTION**

The USBDCmotors card has been designed to management up to 6 Direct Current (DC) motors and 5 analog inputs.

As interesting questions, this card incorporates a connection to the port USB of the computer and the controller is negotiated by means of the protocol IOCP.

The card is able to control the sense of the motor so much, as the speed of the same one by means of PWM.

It is based on the USBStepper card, making modifications in the original outline and with another code for the microcontroller.

## **TECHNICAL SPECIFICATIONS**

- Connection to port USB (Except for the variant without this device).
- Capacity to connect 6 motors DC (up 1 Amper and 36 Volts).
- Capacity to connect 5 analog converters of 8 bits
- Software for connection to IOCP.

## **COMPONENTS LIST**

C1, C4,C5,C6,C7 = Condensers 0,1 mF

C2,C3 = Condensers 22Pf

D1 to D24 = Diodes 1N4007

IC1 = 16C745 Microcontroller

IC2,IC3,IC4 = L293E ICs

J1 = USB Connector

J2 = Power source Connector of 2 pines

J3,J4,J5,J9,J10,J11 = 3 pins Connector

J6,J7,J8 = 5 pins Connectors

Q1 = Quartz Cristals 6MHZ

R1 = Resistance 100R

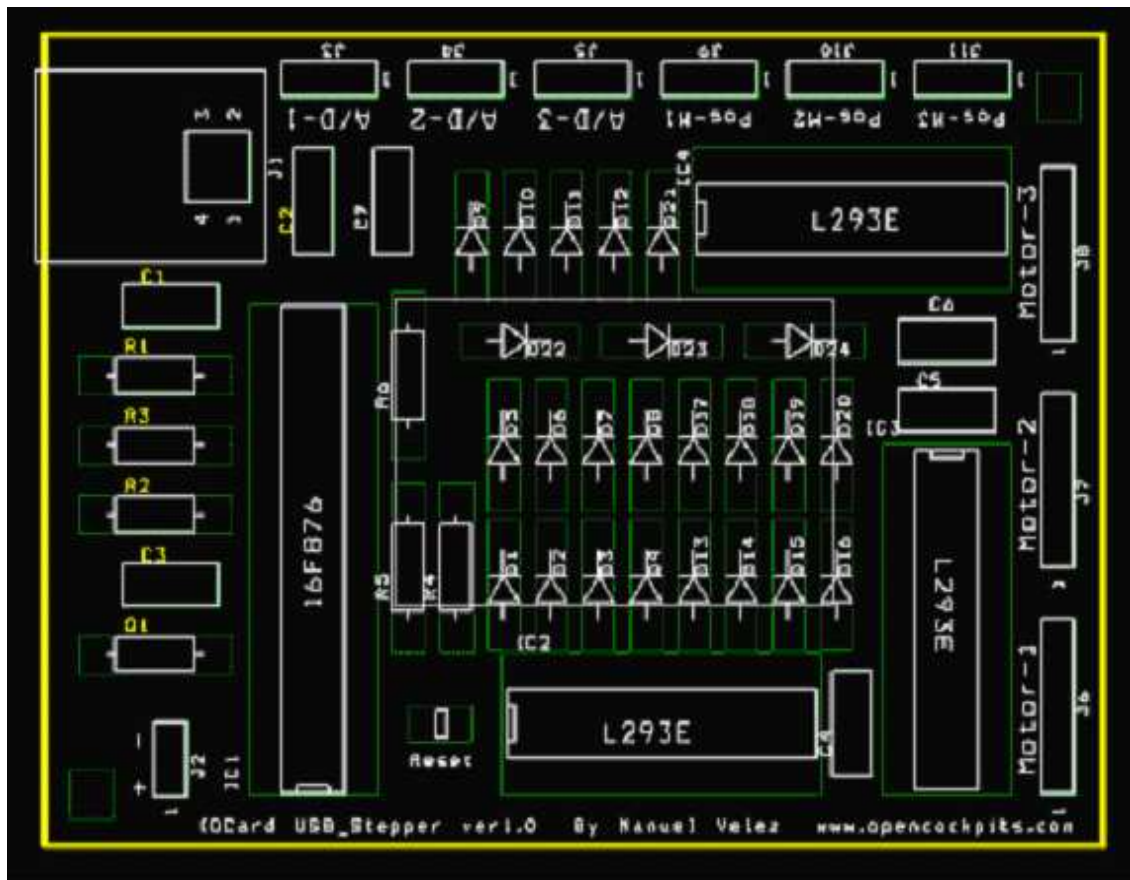
R2 = Resistances 10k

R3 = Resistance 1K5

SW1 = Connector of Reset of 2 pines

R4,R5,R6 = they are not used in this circuit

## CONNECTIONS



J1 = USB Connector.

J2 = Card Reset.

J3, J4, J5, J9 and J10 = Connectors for the Analog inputs. (Potentiometer).

J6, J7, J8 = DC Motors Connectors.

J11 = it is not used.

### **J3, J4, J5, J9 and J10 – Analog inputs.**

The axes can be connected to the connectors J3, J4, J5, J9 and J10 respectively.

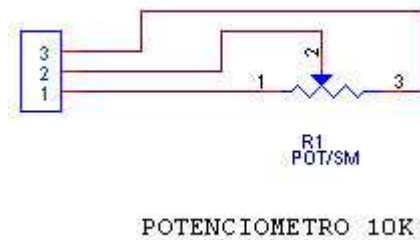
The use of 10Ks potentiometer, is recommended although they can be used of 100Ks and others.

The pines 1 and 3 are respectively at the GND and +5V, being the pin 2 the one that corresponds to the value of potentiometer input.

Pin 1 = GND

Pin 2 = Data

Pin 3 = +5V



### **J6 to J8 – DC Motors connections (2 motors for connector)**

Pin 1 = Motor #1, positive

Pin 2 = Motor #1, negative

Pin 3 = Motor #2, positive

Pin 4 = Motor #2, negative

Pin 5 = Not used



## **SOFTWARE**

### **IOCDCmotors PROGRAMS: connection to IOCP**

The Card is configured by the IOCDCmotors.ini file.

This file contains the following configurations:

MUSB=No, (We will Put to YES in case We have more than a card connected in the same computer. We will need to tell him in that device is connected the card).

deviceUSB=2048, (We will tell to the controller that device is the one that should connect.

If We don't know that device is, We will connect one by one each card and We will go the number that the controller indicates us scoring. This number is different for each port USB of the computer.)

IOCP\_host=localhost, (We will Indicate the address of the IOCP server where it should be connected (SIOC)).

IOCP\_port=8092, (This is IOCP port).

To highlight that the IOCP server,(the normal thing is that SIOC is used), It will be able to be in any computer of the net, in such a way that the card can put them in anyone of the computers.

USB\_AD=5 (We Indicate the number of A/Ds that We will use. (0-5))

IOCP\_timeout=4000 (maximum Time of answer for the packages IOCP.)

IOCP\_Var0=0 (Variable IOCP to control the #1 Motor )

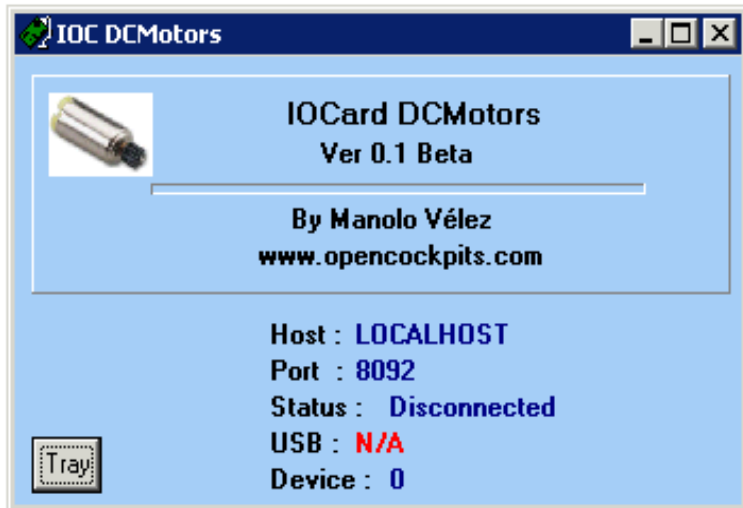
.....

IOCP\_Var5=5 (Variable IOCP to control the #6 Motor )

IOCP\_Var6=6 (Variable IOCP where to read the #1 A/D converter)

.....

IOCP\_Var10=10 (Variable IOCP where to read the #5 A/D converter)



When the program is started it indicates the connection address, if it has been connected to the IOCP server and if there is a USB card is found and working besides in that Device is this card.

To negotiate the motors, We will only have to create a small code in SIOC in which we indicate it the speed to the variable of control of the motor with which We want to move the motor (0=Stopped, 127=Maximun speed) and the turn direction for that which We will add it 128 to the value of the speed if We want that it moves in opposite direction.

The program also has a function TRAY so that it disappears of the screen and be hidden in the taskbar, or if It wants himself, initially it can be hidden putting YES the option of having minimized in the :INI file.

## PROGRAM EXAMPLE

As program example, this it is an self-position system. It is the one that has been used in the example videotape that you can discharge from [www.opencockpits.com](http://www.opencockpits.com)

```
Var 0000, Value 0 // Initializing
{
  &control = TIMER 999 ,0 ,2
}

Var 0001, name motor // Motor control (0-127) 0=left +128=Right

Var 0006, name ad // Analogic Input connecter to potentiometr

Var 0010, name control, Link SUBROUTINE // Subrutine for control (each 20ms)
{
  L0 = &obj - &ad
  L1 = 0
  IF L0 < 0
  {
    L1 = 128
  }
  L0 = ABS L0
  L2 = &velr + L1
  IF L0 <= &aprox
  {
    L2 = &vell + L1
  }
  IF L0 <= &margen
  {
    L2 = 0
  }
  &motor = L2
}

Var 0011, name obj // objective position (fixed in example)

Var 0012, name margen, Value 5 // % error

Var 0013, name aprox, Value 20 // points for approach break

Var 0014, name velr, Value 100 // Speed for follow objective

Var 0015, name vell, Value 30 // Speed for approach
```