

# USBKeys user manual

**Author: Manuel Vélez**  
**Translation by: Manuel Hernández-Peña**

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

**ver 1.0**

## INTRO

USBKeys card has been designed to handle matrix type keyboards.

Any matrix keyboard up to 11 x 8 lines (88 keys) can be managed with this card.

This card connects to the USB port in our PC and its controller is controlled via IOCP protocol or via a software keyboard emulator.

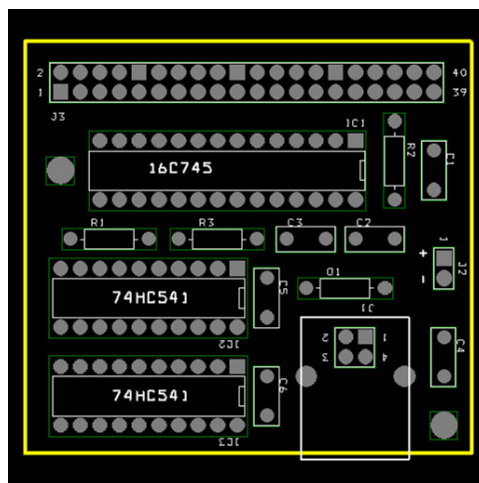
## TECHNICAL ESPECIFICATIONS

- USB port connection
- Manages up to 88 keys arranged in 11 x 8 matrix
- Software for IOCP connection
- Software for an independent connection (keyboard emulator).

## PARTS LIST

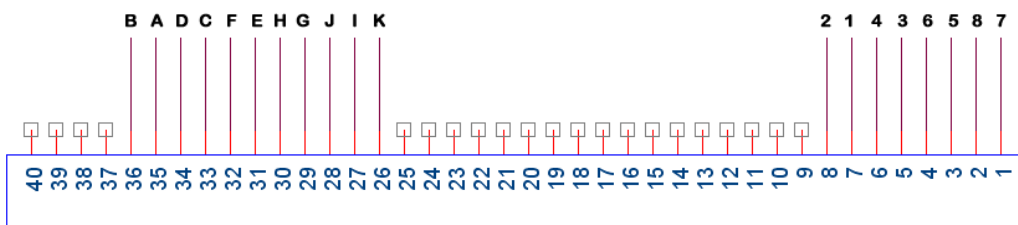
- C1,C4,C5,C6 = Condensers 0,1 mF
- C2,C3 = Condensers 22Pf
- IC1 = Microcontroller 16C745
- IC2,IC3 = Integrated circuit 74HC541
- J1 = USB connector
- J2 = 2 pins connector (power source)
- J3 = 40 pins connector
- Q1 = Quartz oscillator 6MHZ
- R2,R3 = Resistors 10k
- R1 = Resistors 1K5

## CONNECTIONS



- J1 = USB connector
- J2 = Power source connector
- J3 = Input lines connector

## DIAGRAM OF WORKING



J3

The 11 x 8 inputs lines are located at the 40 pins connector as shown in the above picture.

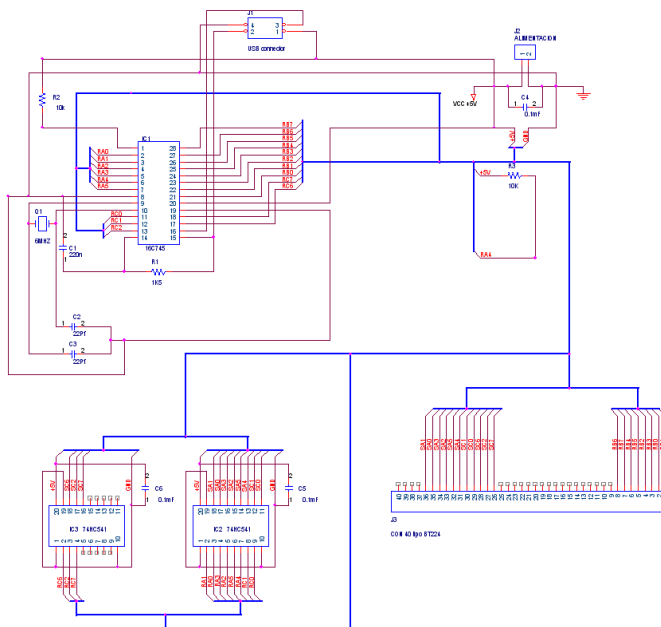
When we press a key, we will shortcircuit one pin in the group-1 (A to K) with one pin in the group-2 (1 to 8). This give us 88 combinations.

- Code 1 = 1 A
- Code 2 = 2 A
- Code 3 = 3 A
- ....
- Code 88 = 8 K

If we shortcircuit several lines at the same time, the card will manage only the first five pressed keys.

**We should never shortcircuit lines in the same group.**

## CARD DIAGRAM



## SOFTWARE USBKEYS

There are two programs to work with USBKeys card: if we want to use IOCP protocol to manage the card we will use one, and if we want to connect the card to a software keyboard emulator, we will use the other.

**We will never start both programs at the same time.**

### SOFTWARE IOCKeys.exe: connecting to IOCP

Card is configured with the file **iockeys.ini**

In this configuration file you can find:

*MUSB=No* , we will set YES in the case we connect more than one card to the same computer. We will write the device number for each card.

*deviceUSB=2048* , will inform controller about the device number to connect with.

If we don't know the device number, we will connect the cards one by one, writing down the number that the controller will show on screen. This number is different for each USB port.

*IOCP\_host=localhost* , IOCP server address.

*IOCP\_port=8092* , IOCP server port.

IOCP server (normally SIOC) can be located at any computer in the network.

*IOCP\_Ini\_Var=0* , First IOCP variable where the pressed keys data will be sent.



When the program starts, will inform us about the connection address, if the IOCP server is connected, if the USB card is connected and working and the card device number.

In addition, the last pressed key is also shown.

To handle with the different keys, we will make a little SIOC program looking for a key stroke into a condition statement and defining the action to be done.

IOCP variable sends an event with the pressed key code, and then sends another event with value 0.

A simple example is:

```
Var 0001 // Input key
{
  IF V0001 = 23 // Key #23
  {
    V0002 = 1
  }
}
```

```
Var 0002, Link IOCARD_OUT, Output 11 // Led
```

### **SOFTWARE encoger keys.exe : connecting to keyboard emulator**

Card is configured with the file **encoger\_keys.ini**

In this configuration file you can find:

*MUSB=No* , we will set YES in the case we connect more than one card to the same computer. We will write the device number for each card.

*deviceUSB=2048* , will inform controller about the device number to connect with.

If we don't know the device number, we will connect the cards one by one, writing down the number that the controller will show on screen. This number is different for each USB port.

window = "a.txt - Notepad" , we will define the window where the keystroke will be sent. Our program must not be running in background.

[ Asignación de teclas – (Keys assignment) ]

```
#1=A
#2=B
#3=C
```

In this example of configuration, when we press the key that corresponds with code 1, the key A is sent to the desired window.

There are special sequences for special keystrokes...

When we are defining keys, we will write \ before any of the following letters to get the corresponding special keystrokes:

```
A = BACKSPACE = #8;
B = TAB = #9;
C = ENTER = #13;
```

D = ESC = #27;  
E = F1 = #228;  
F = F2 = #229;  
G = F3 = #230;  
H = F4 = #231;  
I = F5 = #232;  
J = F6 = #233;  
K = F7 = #234;  
L = F8 = #235;  
M = F9 = #236;  
N = F10 = #237;  
O = F11 = #238;  
P = F12 = #239;  
Q = HOME = #240;  
R = END = #241;  
S = UP = #242;  
T = DOWN = #243;  
U = LEFT = #244;  
V = RIGHT = #245;  
W = PGUP = #246;  
X = PGDN = #247;  
Y = INS = #248;  
Z = DEL = #249;  
1 = SHIFT\_DN = #250;  
2 = SHIFT\_UP = #251;  
3 = CTRL\_DN = #252;  
4 = CTRL\_UP = #253;  
5 = ALT\_DN = #254;  
6 = ALT\_UP = #255;  
\ = \

Example: SHIFT+S would be defined as \1S\2



Once the program is started, it will automatically minimize. To maximize it, we will click in the corresponding icon on the task bar. By right-clicking on this icon, we will be able to close the program.

The program will inform us about the card device number and the last pressed key.