

OPENCOCKPITS USBKeys Card v1.2

I love this card. It is a cheap and effective way to get key sends into your sim easily and with little hassle once you understand the concepts of the initialization setup in the .ini file. By the way, the image opposite shows v0.1, but I think we are up to v0.3 right now.

Too many people have asked me how I have set up these cards to do the job, that having about my tenth request in my e-mail this morning, I decided to set about trying to explain the setup and configuration simply, as I see it. My opinion is that colleagues who have been in touch seem to be confused by the setting up and configuring of the .ini file to do what they want. The card itself is just a simple USB connection and will self install.

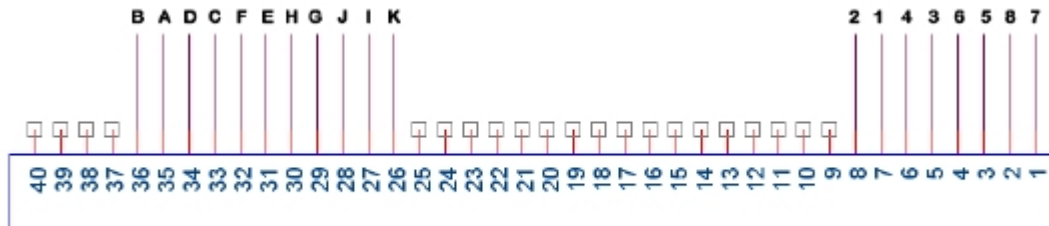


Brief Description.

What you actually get is a small PCB built and tested for 24 euros (that's about 28.50US\$. price as of 20th August 2006). And with the Hagstrom's costing upwards of 100-120US\$, the price has got to be a big factor when choosing an encoder. OK I admit the Hagstrom cards are expandable, but most of us are on a Budget and I personally hate to pay for functions I will never use on an expansion card. You are going to need a 5vdc power source, a USB A to B lead (supplied) and an IDE 40 wire data ribbon cable (you know the type that connects your hard drive to the motherboard).

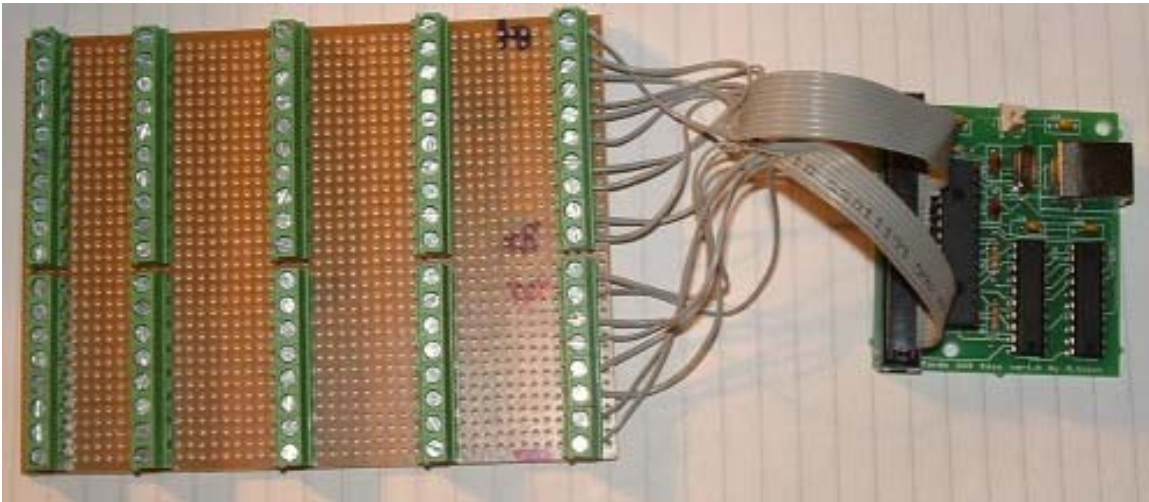


The USBKeys Card simply plugs into one of your USB ports and the system instantly recognizes it and installs it....Good Start. Refer to the USBKeys manual regarding power supply. The USBKeys card requires a 5v supply. The pins are pole sensitive, so please make sure you connect them correct way round otherwise you will have a BAD DAY! So your New Hardware is installed and ready for use. But now you have to connect to it and then tell it what to do. First lets take a look at the connection. Actually where the switches are connected to.



J3

Above is the 40 pin IDE socket J3. Plug in the IDE cable carefully. Usually they have a positioning pin to ensure correct alignment. Now the cable will probably have a coloured marker on one wire to denote the #1 lead. You need to pay special attention to this as the letter/numbers are not sequential. But once you get this organized and sorted out, you can sort out how you are going to attach your switches/pushbuttons.



This is how I did it. I got the idea from the CPFlight MCPEX1 Expansion Board. Basically it is a piece of 'Project' or 'Breadboard' that has several copper strips along the underside that can be soldered to. Then I used screw contacts that have two pins which are pushed thru the board and soldered to the underside. Result is that several rows of connectors can be attached to the board allowing you as many Matrix contacts as you need. And of course if you run out of space. A second board can be linked from the other end. Smart eh :o) Very cheap to make. This board cost me about 15 euros.

Software

At the OPENCOCKPITS website www.opencockpits.com you can download the latest encoder_keys software and the manual. Read this document in conjunction with the manual as there are several things in the manual which I have not expanded on here. This .pdf file is simply to help you get a better understanding of this amazing piece of kit. The program that actually runs this little gem is called encoder_keys.exe. When run, it is initialized from the contents of the file encoder_keys.ini which we will look at further down. Once you have run encoder_keys.exe, you will be presented with a small window as above. See the USB running in green, this tells you that the card has been installed and initialized correctly. Then we see the Device No. in Blue, here it's 2048. Note this number because you will need to check in the .ini file that the device no. there is the same so that the .ini file loads the correct keysend configuration for use with the USBKeys card to which it relates.

When you first download the software from OPENCOCKPITS, the .ini file will be empty. This means that you now have to edit the file using a simple text editor like Notepad.

Right is an example of the .ini file when opened.

Note deviceUSB=2048. This is the number that the device returned as it's USB number in the top diagram.

Also note as you look down the file under [assign your keys] that there is a descending column #1 #2 #3 etc. This goes right the way down to 88.

Now we are getting somewhere and it's on to assigning keystrokes into your .ini file.

How It All Works

Before going on to assigning key sends, you have to understand the relationship between the card and the .ini file.

Basically the card is built around a scanning Matrix of 8 numbers (1 to 8) and 11 letters (A to K), much like the Kids Game of Battleship where you used to draw a grid and number each square down with letters and across with numbers. Each square had an alpha/numeric address didn't it? So the USBKeys card Matrix allows 8 number and 11 letter combinations like K4 B2 or G8 etc. 8 x 11, that's 88 different combinations. So it follows that this card can send 88 different keystrokes or key combinations into your simulator, that's one hell of a lot of switches :o)

Remember in the .ini file the column of numbers started at #1 and finished at #88 and that the card can send 88 keystrokes?

Something is starting to happen.....these numbers represent the keystroke that will be sent for each Matrix address.

Understanding the Numbers

In the .ini file, the #(number) sequence represents a particular address on the Matrix. Starting with A1=#1, you work down the column increasing the address as you go in blocks of 8. That means #1 to #8 equals A1 to A8, but when you get to 9 (as there is only 8 numbers) you need to increase the letter by one and start counting again. Accordingly #9 will be address B1, 17 will be C1, 25 will be D1 etc. And each letter group will be numbered 1 thru 8, A1 to A8, B1 to B8, C1 to C8 and so on upto K8.

Known Limitations (You need to Know This Before You Assign Keys)

I use two of these cards, they have expanded my level of control in the cockpit environment to such a degree that I never need to use the Mouse or KB after initial loading of the Flight/Aircraft. In fact this is all done from the External Instructor Station. No mouse or KB comes in the 'pit' with us.

- 1.** The USBKeys card does not repeat key sends while the button or switch is held in the ON position. So there are certain controls which you cannot assign to the USBKeys card. Once a button or switch has been pressed, the USBKeys Card sends the key assigned to that particular Matrix Address where the switch or pushbutton is connected to. It only sends the key once. It does not repeat, nor can it be made to repeat thru FSUIPC. So bearing this in mind, do not assign keys to controls which need to be repeated like Trim, Radio Controls or OBS Knobs.
- 2.** The USBKeys card cannot work properly with permanently ON switches. From experience I have found that with permanently ON switches, when that switch is turned ON, correctly the card sends the key assigned to it. However the switch is still bridging the address and erroneous switching will occur with other switches in the same Letter group. For example, a switch is bridging A & 4. You turn it to the ON position and it correctly sends the key assigned to that address. However when you switch another switch connected to the A group (e.g. A1 or A7), because the switch connected across A4 is still connected, that or another switch in the A group may operate instead of the switch you intend to action. So use only pushbuttons, momentary ON toggle switches which are spring loaded back to

centre OFF or any other switch which returns to it's OFF position when released and breaks the connection across it's Matrix Address.

These are not major drawbacks, just things to bear in mind when deciding how to setup your switching using this card.

Rob Mountford has mailed me asking about switches lighting LED's for annunciators etc. I'll deal with switching annunciators in another document.

What Do You Want To Do With A Pushbutton or Switch?

First thing is to remove any key commands from the FlightSim menu that you don't use. There are loads of them (I actually took them all out and started again, putting back only about 3 or 4 like ALT, ALT + ENTER, P end Esc). Make a note of what you have taken out because you can re-assign them to the USBKeys card or any other method you choose to deliver the command.

Actually what I have done is start an Excel file having all the Keys and Key Combinations listed by What it Controls, how it is sent and to what part of the simulator (the sim itself, the Aircraft or a particular module) it is sent to and what it does. This makes it really easy to find the next free key or combination and also prevents the chance of having rouge controls where two things are assigned the same keystroke. You know, you are over the Mediterranean at 33,000ft. You push the HDG selector button and both engines stop because the HDG button and the fuel valves are both assigned the same keystroke. You're going to have your afternoon spoilt :o) Finding rogue controls can be a b*****d and you can spend days looking and re-testing, I know I've been there. When it's this complex-keep it tidy.

Now armed with a list of free keys and key combinations you can set about deciding whether you can deliver them thru the USBKeys card based on the known limitations and assigning them into the USBKeys.ini file.

Nobody's set up is the same, so you have to be artistic here. Make a list of what you want to do with a switch and then decide whether it can be accomplished with a single keystroke. Keep it simple. Start with single numbers like 1,2,3,4,5,6 etc and then single letters A,B,C,D,E etc before going onto Key Combinations like ALT + something, CTRL + something and SHIFT + something.

Directing Keysends

There are several ways to accomplish what you want to do. You may want to send this key to FS2004, to your installed Aircraft or through a third party utility or Module like FSUIPC,FSClient or Key2Mouse (more on directing keys to FSUIPC in a minute).

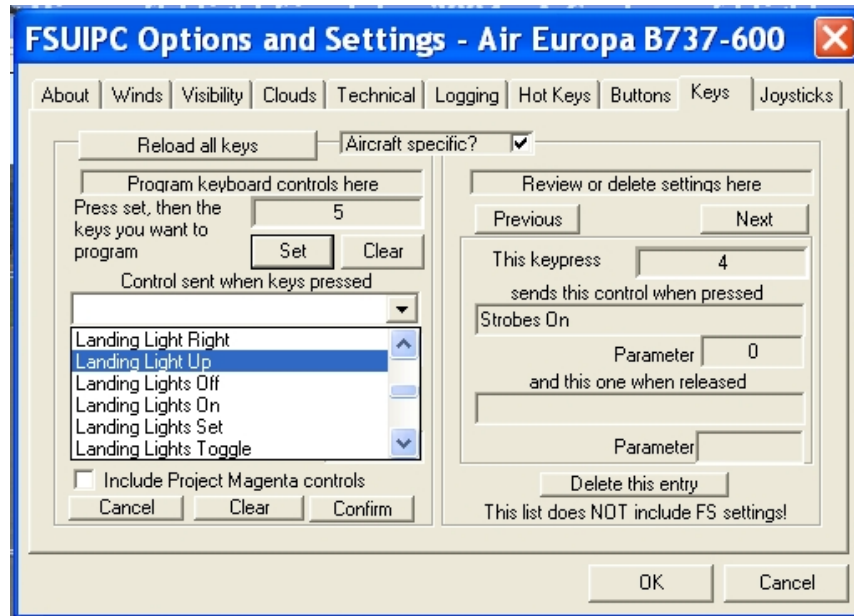
Now is the time to finish your list and assign a keystroke (or combination to that connection). Here's an example of the first few lines in my Excel file

Keystroke	Sent To	Action	Address	Via
1	FSUIPC	Autostart Eng.1	A1	USBKeys
2	FSUIPC	Autostart Eng.2	A2	USBKeys
3	FSUIPC	Strobes Off	A3	MCPEX1
4	FSUIPC	Strobes On	A4	MCPEX1
5	FSUIPC	Landing Lights On	A5	MCPEX1
6	FSUIPC	Landing Lights Off	A6	MCPEX1
7	Key2Mouse	Eng. 1 Gen OFF	A7	MCPEX1
8	Key2Mouse	Eng. 1 Gen ON	A8	MCPEX1
9	Key2Mouse	Eng. 2 Gen OFF	B1	MCPEX1
0	Key2Mouse	Eng. 2 Gen ON	B2	MCPEX1

Directing Keystrokes thru FSUIPC (or any other 3rd Party Module)

Peter (God Love Him, Where Would We Be Without Him) DOWSON has given us much more control over our simulators with FSUIPC than Microsoft could ever have dreamed would be required.

If you have registered copy of FSUIPC (you did register it didn't you?) then you can expand your experience most noticeably by directing key sends from the USBKeys card to FSUIPC which in turn will execute commands within the sim for you.



For example, FS2004 gives you the option to only toggle certain things. That is to say if a light is on, you toggle it to off or if it is off you toggle it to on. The Landing Lights and the Strobes are good examples. But we want more accurate and realistic control than that.

Look in my example list above and you will see that no.1 and 2 send keys 1 & 2 thru the USBKeys card to FSUIPC to autostart Eng.1 and 2 (as long as the fuel levers are correctly set). The Ground Start Switch position is wired thru the USBKeys card because the switch is returned to off once the start sequence has been completed. In the Keys menu of FSUIPC, Pete gives us the possibility to use keystrokes to accurately control an extensive extra list of simulator commands accurately. And on top of that by using the Aircraft specific checkbox, you can have a whole new list for all aircraft in your hangar.

Think about it, the list is endless and only as good as your imagination.

Finally, Getting to Grips With Editing the encoder keys.ini File

Here's an example encoder_keys.ini file that will help you understand it. Disregard anything contained within square brackets. Notes to help you understand it are in red and again do not form part of the file.

[fichero de configuracion para USBKeys]
[Configuration file for USBKeys]

[Múltiples USBs]
[Non unique USB]
MUSB=No

(if you have more than one card on your system, put Yes here)

[Número de periférico para el USB]
[USB Device number]
deviceUSB=2048

(when you first connect your card, this is the USB device no which your computer reports. this means that this file initialises card no. 2048 (or the no. your computer reports)).

[Nombre exacto de la ventana donde se enviarán las teclas]
[Name of window for key send]
[window = "FS2004"]

window = "FS2004"
(where the keysend is sent)

[Asignación de teclas]
[assign your keys]

(This is where you assign what keys are sent from a particular Matrix Address. Remember that there are 8 no.s and 11 letters so you start at A1 which is 1, A2 which is 2 (the A's are easy), B1 which is 9, C1 which is 17, D1 which is 25 etc until you reach K8 which is 88. getting the picture? the left most no following the hash is the configuration as explained above. the second alpha/numeric character group following the equals sign is the key character that is sent to the sim. for example #1=1 (matrix address A1) sends the keystroke 1 to the sim, #10=0 (Matrix Address B2) sends the keystroke 0 to the sim) (now we come to the slightly more complicated keysend combinations. using special combinations, the USBKeys card can send combined keystrokes like CTRL + something or ALT + something or even a whole range of special keys. see the Opencockpits USBKeys manual for the complete choice which is listed at the end of the document. this is accomplished by adding numeric characters and the forward slash key. for example #21=\34\4. ok #21 is Matrix Address C3. it is configured to send CTRL + 3. in the example, \3 means CTRL Down, 3 is the character to be sent with the CTRL key and finally \4 means CTRL key UP. a second example is #17=\10\2. right #17 dictates the Matrix address which is C1, \1 means SHIFT Down, 0 is the character sent with the SHIFT key and finally \4 means SHIFT Up. so the keysend assigned to C1 (#17) = SHIFT + 0 :o) it's not rocket science, you just need to learn how it all comes together. For a full list of what extended keys and key combinations you can use, see the USBKeys manual at www.opencockpits.com)

#1=1	A1
#2=2	
#3=3	
#4=4	
#5=5	
#6=6	
#7=7	
#8=8	
#9=9	B1
#10=0	
#11=.	
#12=-	
#13=\c	
#14=	
#15=	
#16=	
#17=\10\2	C1
#18=\31\4	
#19=\32\4	
#20=\33\4	
#21=\34\4	
#22=\35\4	
#23=\36\4	
#24=\37\4	
#25=\38\4	D1
#26=\39\4	
#27=\30\4	
#28=\3\16\2\4	
#29=\3\19\2\4	
#30=\3\10\2\4	
#31=\1\j\2	
#32=\1\k\2	
#33=\1\l\2	E1
#34=\1\m\2	
#35=\1\p\2	
#36=\3\k\4	
#37=\3\m\4	
#38=\3\1\e\2\4	
#39=\3\1\f\2\4	
#40=\3\1\i\2\4	
#41=\3\1\j\2\4	F1
#42=\3\1\k\2\4	
#43	
#44=\Z	
#45=/	
#46=\A	
#47=	
#48=	
#49=\E	G1
#50=\F	
#51=\G	
#52=\H	
#53=\I	

```
#54=\J
#55=\K
#56=\L
#57=\1\F\2      H1
#58=\1\I\2
#59=\M
#60=\P
#61=\1\E\2
#62=\O
#63=
#64=            I1
#65=\1D\2
#66=\1E\2
#67=\1G\2
#68=\1H\2
#69=\1J\2
#70=\1K\2
#71=
#72=
#73=            J1
#74=
#75=\1X\2
#76=\1W\2
#77=\1U\2
#78=\1V\2
#79=\1L\2
#80=\1Q\2
#81=            K1
#82=
#83=
#84=
#85=
#86=
#87=
#88=
```

[End of File]

Well that's should take care of things as far as USBKeys is concerned. Thanks for your interest and hopefully we have managed to clear up some confusion. Take it easy and remember 'If In Doubt - Get More Beer'.

ian@737nq.co.uk

USBKeys.pdf v1.2

22nd August 2006.