



The PMDG OVERHEAD PANEL

Hardware Control of the PMDG Overhead Panel, CDU & Other Panels

Ian P.Sissons OSWESTRY England

March 2006

ian@737ng.co.uk



MY GOAL

Apart from initial computers start up, To have a totally mouse and keyboard free cabin from getting into a cold and dark cockpit until shutting down at the final destination. Now that's no mean feat :o)

TIP.

A very complex subject, so please read and understand this document before attempting anything. Take one step at a time, get it working correctly before proceeding to the next step. From experience, as humans we are so eager to get the job finished when we are 'buzzing' that sadly we make mistakes or overlook some minute detail which is going to cause us major headaches.

OVERVIEW

Apart from the built in Keyboard Commands that PMDG allow in the PMDG Menu in your flight sim and the usual commands associated with FS, there is very little that you can do to control this very complex aircraft without reaching for the mouse or keyboard.

So when I started this project, very soon it became clear to me that every user interactive command was communicated to the computer by means of the keyboard, mouse, joystick axis or expensive interface card/hardware/software.

Now we all want as realistic a cockpit as possible, so that's the reason I purchased a manufactured MIP, Throttle Quadrant, MCP, Overhead Panel etc.

Primary Flight Controls are probably the first thing most people cover. Your Joystick/Yoke (with Programmable Buttons, you may even have a throttle as well) and Rudders are the first step to hardware control of your aircraft. Then you may look at Radio's (GoFlight etc) or even an MCP from CPFlight or GoFlight.

All of a sudden you are controlling quite a lot of the aircraft systems by way of hardware.

But a great deal still remains uncontrollable without the dreaded K&M.

Take a look around your cockpit and make notes of what you use the keyboard and mouse for, you will be amazed at how much you still rely on them.

The Keyboard is the easiest candidate for removal because there are several excellent keyboard emulators available at a realistic cost. See the section below about Keyboard Emulators.

KEYBOARD EMULATORS



OK as the name suggests, these handy little gadgets either replace or co-exist alongside your KB and are capable of sending 'pre-defined' keyseeds to your computer by various means. In Europe, the popular one is the OpenCockpits USB Keys Card (on the right) while in the US, Hagstrom (on the left) seems to be the main contender. In fact Hagstrom manufacture quite a range from the basic thru to the sophisticated and support these with various cables, breakout boards and other

accessories, which suggests a strong market for these cards.

I cannot go into too much detail about the Hagstrom Cards as I don't use them, so for this discussion all references made are concerning the Open Cockpits Card.



My Keyboard Emulator has its own .ini file which I can modify with a standard text editor. It plugs into a spare USB slot on my computer which recognises it as a USB keyboard....so far so good.

Communication with the card is via a standard 40 wire IDE lead (same as from your Hard Drive to your Motherboard – you know the type). It has a connection Matrix using 11 Letters and 8 Numbers. So theoretically, we can send 88 single or combined (SH+Key, CTRL+Key, CTRL+SH+key) keystrokes to the computer. Now we are cookin'.

The Matrix has addresses from A1 thru K8 and each address is configured in the .ini file with a particular key or combination. So by connecting a letter and a number (the Matrix Address) the key assigned to that address is sent to FS as a key command.

So it follows that if there is a key command for any particular function in FS, then the keyboard emulator can send that command for you. For example CTRL+. Parking Brake or F7 Extend Flaps one stage.....getting the picture?

The Matrix addresses are wired thru Hardware Switches which make the connection when I turn them on. Making the connection across the matrix address sends the key command to FS and the desired system state changes.

By some planning and re-defining of key commands, I have been able to remove the keyboard completely from the cabin..... so I'm half way there, don't you believe it, the mouse is a force to be reckoned with and really not so easy to dispose of.

Remember the Keyboard Emulator, it will be helping us control the PMDG Overhead with Hardware.

TIP.

There is nothing worse than conflicting keyboard commands. For example, you have assigned a given key command to turn on your landing lights, but in the FS Controls/Assignment Menu, that same key command is assigned to something else. For Example, As you pass thru 10,000' you reach out and switch on your landing lights only to see your Flaps deploy to 40 degrees.....not good. Good management of key commands is a must. I list them in an Excel File which allows me to 1. make sure I don't conflict with any of the PMDG or CPFlight Hardware commands and 2. I can easily refer to or find the next vacant key command.

PROGRAMMEABLE JOYSTICK BUTTONS



Now we all know what this is don't we. CH must have sold millions of these. Very soon in the evolution of Flight Simming, CH Products developed their Yokes. For most simmers a very realistic and inexpensive replacement for their Joystick (which it actually is in a re-modelled way). It is bristling with add on programmable buttons which can ease the workload and add to the realism factor substantially. In fact I still have 2 in my pit until I decide whether to buy the PFC Jetliner Yoke or the Symulatory verion.

Take a look at your favourite aircraft and assign the buttons on your yoke or joystick to model the way the aircraft manufacturer gives control to the real pilot. I have these buttons all assigned to a function either in FS or thru FSUIPC to imitate the operation of the Boeing Yoke including the PTT switch for when I'm online with Vatsim and talking to Air Traffic Control.

Make full use of programmable switches, anything that eases the workload is a step in the right direction.

MY PERSONAL GRIPE

We as Flight Simmers and Cockpit Builders are presented with a vast array of Software, Hardware and other associated add-ons for the Microsoft FS. Each offers us different qualities and experiences. But it appears to me that each Software developer and Hardware company has gone down its own road and developed a product that basically interacts with FS core, but that's all. So each Aircraft add-on or piece of hardware talks to FS partly in a common dialogue and goes off and does its own thing for the rest of the time using its own language and programming.

There is no standard existing between software/hardware company's in the FS market which means that Company A's program will not work with Company B's Hardware, unless of course there is some kind of 'back scratching' going on.

We the end user suffer for this by having to find 'workarounds' or worse still we have to simulate the simulation. Ok lets pretend that when we push this button that happens. How bad is that?

FSUIPC by Pete Dowson partly addresses this by allowing us to gain access to some extra internal FS offsets/commands/whatever you want to call them. God Bless Pete and FSUIPC because without it we would not be in the evolutionary state we are now with this hobby. Please support Pete by registering your copy of FSUIPC. It gives him the incentive to further develop and you will be able to get at a lot more FS functions with a registered copy.

So because no standard exists, we face two problems. One is that we have to become locked into someone's system, which developers like because that means you have no choice but to buy the expensive software and interfaces that goes with their setup and it keeps prices artificially high. The second problem is that with no standards, there is no exact way to overcome any particular problem or develop the next stage of your project.

Despite my efforts to try to make life easy with the PMDG Overhead, I have been unable to get any information from anywhere regarding what goes on inside the program. So the only solution is to find a 'workaround' until such time as PMDG can find it in themselves to make the information freely available.

I find it hard to comprehend that in this age of technology where data can be transmitted around the world in nanoseconds and just about every other thing in our lives is governed by a standard (even the water you drink) that we have to be subject to this kind of non-cooperation between add on suppliers. There is something seriously wrong and I intend to go looking once my project is finished....i do so enjoy a challenge.

Come On Guys, let's get a standard going here.....with FS-X due out, now's the time to do it or we'll have to wait another couple or three years for it to happen.

OK I've put my Soap Box away so lets get on with the job in hand.

The PMDG OVERHEAD PANEL

Fair Play it is a good representation of the 737NG Overhead Panel. But there is no way to control it without the Bloody Mouse (when I grow up I'm going to build a factory making mouse traps :o)



So remember we spoke about no standards, mouse operation only, workarounds? Well then in the absence of any firm information, a solution had to be found. I will do the final assembly of the pit during the summer and there was no way I could do that without integrating my hardware overhead panel to control my PMDG.

Then the answer had to be a workaround. Now the PMDG Overhead is operated by mouse clicks only, so that's where I started. I went looking for utilities that would simulate a mouse click. Lo and behold somebody had one. At Luciano Napolitana's WideView site (www.wideview.it) I found his Key2Mouse utility. This is a .dll file that works within FS, so loads automatically when you start the program. Basically Key2Mouse can map any given point on your screen and simulate a right click, left click or wheel motion to that screen address which you assign a key command to.

Key2Mouse

Not only that, but it allows you to assign either a single click for basic switches or a toggle action where a switch has two positions. For accurate synchronisation between my hardware panel and the software panel, I chose to assign 2 different switch positions to some switches so that I got an accurate On/Off synchronisation. For example, the Engine No.1 Generator has an On Bus Position (Blue Lights Out) and an Off Bus Position (Blue Lights On), so it followed that if I threw the switch down to bring the generator On Bus, I would want the PMDG to do the same and vice versa.



Luciano lets you use the full version for 10 minutes for trial purposes after which you have to re-load your FS to get another 10 minutes to experiment. The licenced version has no restrictions on it obviously.

So you pick your switch, decide how you want to action it, create a macro in Key2Mouse and assign a key command to that action.

Continue across the Overhead Panel in this manner until you are satisfied with the Overall Operation and Performance of your configuration.

Read the manual that comes with Key2Mouse, it explains everything quite well.

What I did was pick simple switches first to experiment with like the cross feed valve which is a single motion switch. Each toggle moves the dial 90 degrees, so using a rotary switch wired at 90 intervals, data on one side and 5v for the LEDs on the other I got the cross feed valve switch working and the

annunciators lighting up at the correct time in synch with my hardware overhead.

Now remember that Key2Mouse allows you to assign a key command to each switch action. So it would follow that you would need to use the keyboard to send that command forward wouldn't it? But not if you have a keyboard emulator (he he he). So by using Hardware Switches on the Keyboard Emulator to generate key commands, these are sent into FS and Key2Mouse acts upon them by executing the macro associated with that keysend.

Wow, it's heavy stuff this, I'm sweating thinking about it :o)

COMPLETION.

OK if you look at the diagram of the overhead above, you will notice a series of coloured arrows pointing to each switch. This is correct at the time of writing. Each coloured arrow represents a state of completion.

RED...This switch has been mapped in Key2Mouse, has had a keysend assigned to it which has also been assigned an address on the Keyboard Emulator matrix and functions in synchronisation with the corresponding hardware switch on the SYMULATORY Overhead Panel.

YELLOW.....Next To be configured/assigned as above, but not yet done.

PURPLE...These switches as far as I can see are 'Animated Only' and will only be hooked up when all other operational switches have been connected.

GREEN.....For Future Use (I have a Plan :o))

LET's MAKE IT EASY FOR YOU.

Keyboard Emulator replaces your keyboard. Obviously you will have to know how to define key commands in the initialisation file. Get to know this like second nature. Learn to use the Emulator by connecting up some simple switches to command basic functions in FS. Practise makes perfect. When and only when you have mastered the keyboard emulator and understand how it works go on to the next step.

TIP. Remember we are aiming to remove as many key and mouse command as possible from FS, so go thru your Key Commands in the FS Menu and delete all the commands you never use. You will be surprised at how many there are. Remember to keep a record of what commands are used by what so you do not have one command trying to do two things. FS itself will usually act on the command first giving undesired effects.

Then move on to the Mouse emulator utility. There are others besides Key2Mouse but I use this because it is a dedicated FS Module and is relatively inexpensive to licence. Key2Mouse also lets you edit and change Macros if you decide add, delete or change a key or switch function.

A lot of work involved, but the satisfaction of seeing it all work is amazing. Flight Simmers are a pretty resilient bunch and we tend to be on a 'mission' with something like this.

Finally to re-cap on what we have discussed. This is a method of controlling the PMDG Overhead Panel by means of Hardware Switches. There is no other simple way.

We have mapped the Hotspot of each switch we wish to operate in Key2Mouse. This mapped address has been assigned a keystroke and an action. So by sending a keystroke to FS and subsequently Key2Mouse, the action which has been assigned will be executed. Basically that switch will turn on or off.

You are free to choose whether you use a Keyboard or an emulator to send the Keystroke, I choose the emulator because of my requirement to control the PMDG from hardware switches. This also allows me to have my hardware Overhead Panel synchronised with the PMDG for reasons of accuracy and correct annunciation.

So for example, I have assigned CTR+SH+F9 to Engine 1 Generator Switch Up (Off Bus) and CTR+SH+F0 to Engine 1 Generator Switch Down (On Bus). So when I push the switch down it connects the two lines for the Matrix address and the Keyboard Emulator sends CTR+SH+F0 to FS/Key2Mouse and the switch moves down and goes on bus with the corresponding Annunciator action. When I push the switch up, the emulator sends CTR+SH+F9, the switch moves up and the annunciators come on. The same principle will apply to every switch you have mapped/configured. I even amazed myself.....

KNOWN LIMITATIONS.

Because Key2Mouse maps your screen position in pixels, once set, you cannot move the panel around the screen. This is not a problem with the Overhead as it opens full screen anyway.

The Panel you are directing the keysend to has to be open. Again no big deal, because when I get in the cockpit, I synchronise the software/hardware panel which takes seconds and then close the overhead (on a button of course).

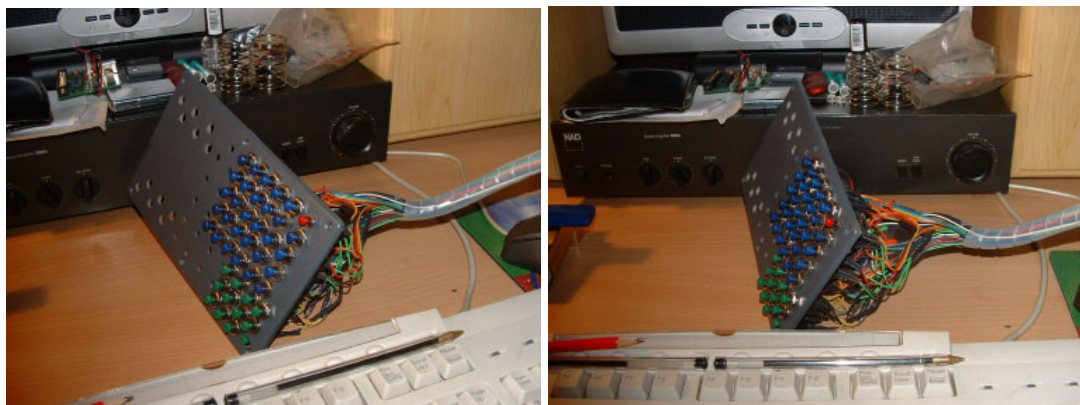
For your information the Lighting/Starter Panel is setup thru FSUIPC, so I do not need to have the software panel open again until shutdown (unless there is an in-flight problem of course) as my hardware panel is synchronised to my software one and becomes my main reference of system state throughout the flight.

I know it's a workaround, but it works. It works well, is 100% reliable and because Key2Mouse is a dedicated FS Module, it works seamlessly within FS.

AND THERE'S A BONUS.

Not only can this technique be applied to the PMDG Overhead Panel, it can also be applied to any other panel within the aircraft. So that got me thinking. The only thing I need the mouse for now is the CDU.....not any more I don't :o)))

So I went away, locked myself in my workshop and came out with the basis of an idea. Using a second emulator and finding enough free key combinations has allowed me to apply this idea to the CDU as well. The display is not a problem, I worked that one out ages ago. I have two beautiful dummy CDU's in my cockpit and before I pull them apart, I have to prove the concept. So here's the working prototype which I want to complete and test before I start building a key matrix into the dummy CDU's.



I will have it finished this week with all 70 buttons functioning, trust me I know where every wire goes and what it does. Where would we be without rainbow cable :o)

The prototype shows the Alpha/Numeric Keys in place. I already have the Page and LSK Keys working, just need to get them onto the faceplate.



This is an image I took for explanation purposes. The colour coding is the same as the Overhead Panel with the exception that I now have the Alpha/Numeric keys operational.

So there you have it. No more mouse or keyboard, have I died and gone to heaven?

As always, this is written from personal experience. I like to think that I'm reasonably intelligent, but if I've missed something or you have a better way to do it, please do let me know. With no standards, we all rush off in different directions to achieve the same result. Accordingly any comments, feedback or criticism along as it's constructive is always welcome.

I hope you enjoyed reading this document and that it made you think a little. If it did, then I have achieved what I set out to do.....Thank you for your time and your interest

Ian P.Sissons OSWESTRY Shropshire UK