CPFLIGHT MCPEX1 Relay Outputs

30th May 2009

I'm sure a lot of guys who have one of these

also have one of these.

Yes of course it's a CPFlight MCPEX1 Expansion Board.

Now apart from attaching all those switches to get 'stuff' in to your sim, how about thinking about using it to control something useful, using the output relays ? I'm sat here looking to achieve some switching ideas and I thought why not use the switching relays on the MCPEX1 ? Since I fitted the MIP737 board, I'm only using 3, so 7 spare :o)). This document purely deals with switching the relays, I'll go into detail about my switching and the inputs in another document.

Ok, I know CPFlight give you a few 'pre-programmed' choices, but what do you do if you want to sort out something serious ?   Read the manual :o))   But, with respect to my mate Paolo, the manual assumes we have a Degree in Computer Sciences and does not go into detail nor give any good examples that us 'mere mortals' can understand.   Please read this document in conjunction with the MCPEX1 manual section about digital outputs. We'll discuss digital inputs in another doc once you've got your head around writing the lines for the config files.
From the Manual:

FIELD DESCRIPTION (SEE ABOVE EXAMPLE)
First field: is the text you will see in the selection drop-down list; must be included between " simbols.
Second field: is the FSUIPC address as indicated in "FSUIPC for programmers.doc" document. Address has to be in Hexadecimal notation with "&H" suffix and between " simbols.
Third field: is the number of byte as indicated in "FSUIPC for programmers.doc" document.
Fourth field: is the valueOFF for the control according with the table below
Fifth field: is the valueON for the control according with the table below
Sixth field: is the type of control according with the table below

What's that all about then, well I'll tell you.   Let's look at some examples I used.

Two things you need: First Peter's FSUIPC for Programmers.pdf which you can get at www.schiratti.com/dowson.   In here Peter lists all the 'offsets' (addresses) that you can use with information about that address. And while you're there, get the FSUIPC SDK file as well.   Bundled with that is Pelle Liljendal's excellent FSInterrogate that you can use to check the value change in your chosen offset when you action that item within your sim.

First Example
What this does is switch the relay when an engine is started using the offset 'Combustion Flag', **0894 for Eng 1** and **092C for Eng 2**. I used this to control Eng Gen Power delivery.
Here the Engine is stopped, so the offset value is reporting a 0.

0890	ENG1 Mixture lever	Engines	
0892	ENG1 Starter switch pos	Engines	
▷ 0894	ENG1 Combustion flag	Engines	0
0896	ENG1 N2	Engines	
0898	ENG1 N1	Engines	

And here with the Engine Started, it's reporting a 1 when interrogated.

0890	ENG1 Mixture lever	Engines	
0892	ENG1 Starter switch pos	Engines	
▷ 0894	ENG1 Combustion flag	Engines	1
0896	ENG1 N2	Engines	
0898	ENG1 N1	Engines	

Remember you can get the same result for address 092C for engine 2. Ok, so now we know the value in the offset changes when something happens. Now let's apply that to the output on the CPFlight MCPEX.
First backup your CPFlight DigitalOut.cfg file because you are going to be writing additional lines in this file to do what we want. If things don't go to plan, you can always restore your original file.
This is the line I wrote in the Digital.cfg file that switched the relay when the engine started.
"Engine 1 Running","&H894",2,0,1,0
Now let's look at it a bit closer. From the manual, we know that the control line is made up of 6 fields and all fields are separated by a comma.
First Field. Pretty Self – explanatory, what you see in the Digital Output Drop Down menu in MCP_CONF. Put between " Symbols. Thus, **"Engine 1 Running"**
Second Field. The Offset (Address) that we want to use. In our case 0894. But must be in Hexadecimal Form and contained between " Symbols. Form must have an &H suffix, meaning that you replace any leading 0 with &H. So 0894 becomes &H894. Offset was got from Peter's FSUIPC Doc for Programmers, see the illustration on Page 1 marked '2' in red. For addresses **NOT** starting with 0, simply preceed the address with the &H suffix. E.g address 3590 becomes &H3590 (you'll see later).
Third Field. This is information about the byte that we also get from the Programmers doc, see the illustration on Page 1 marked 3 in red.

Right, now we have the field name "Engine 1 Running" and the offset information. And we know the offset is changing thanks to FSInterrogate, it's time to tell the card what we want to do with the information.
The last three items on the line are values to be used in deciding when to switch the relay (fields 4 & 5) and how to apply these values when deciding what type of control to use (field 6).

Fourth Field. Val Off, well we know that's 0, so 0 it is. * This is only to fill the field, see field 6.
Fifth Field. Val On. With the engine running, the value was 1, so 1 goes here.
Sixth Field. The type of control to apply. From the manual control type 0 means that the relay will be switched when the value in our offset is equal to Val On, 1 in our case (field 5).
- Not to confuse you, but this type of control does not look at Val OFF, it only switches on when the Value at the address is equal to Val On. It will switch off when the value is anything else but equal to Val On. We have included a 0 in field 4 purely to make the line useable by FS_COM.
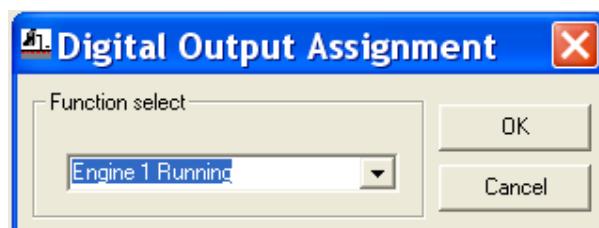
Second Example

What this does is switch the relay when the Idle/Cut Off Lever is set to Cutoff.   I am forever grateful to Peter Dowson for pointing this out to me and to Paolo at CPFlight for teaching me how to get the lines written for the DigitalOut.cfg file.
Using the relay to route 5vdc to the Spar & Engine Valve Annunciators, when the Fuel Lever is in the Cutoff Position, the corresponding LED's are Lit.

OK, so how do we do it. From the FSUIPC Programmers doc, we know that Engine 1 Fuel Valve address is 3590 and Engine 2 is 3594.   So a quick check in FSInterrogate confirms this.

▷ 3590	Engine 1 Fuel Valve [FS2002+]	Fuel			0

Engine 1 Fuel Lever in Cutoff, value of parameter in offset = 0

▷ 3590	Engine 1 Fuel Valve [FS2002+]	Fuel			1

Engine 1 Fuel Lever in Idle, value of parameter in offset = 1

So, the first three parts of the line can be written.
"Eng1 Fuel Valve Open","&H3590",4

Now we need to add the values and the switching action to the line in the DigitalOut .cfg file.
Again I use a 0 type of switching action here, so again the fourth value will be 0 just to complete the field values.
Referring to the parameter values above, I want the relay on to light the LED's when the fuel lever has cut the fuel (off). That means the ValOn is going to be 0.   If you want to switch the other way, just change the ValOn to 1.
There you have it, the complete line:
"Eng1 Fuel Valve Open","&H3590",4,0,0,0

		FS2004]
3550	56	*Reserved for FSUIPC*
3590	4	Engine 1 Fuel Valve, [FS2002+]
3594	4	Engine 2 Fuel Valve, [FS2002+]
3598	4	Engine 3 Fuel Valve, [FS2002+]
359C	4	Engine 4 Fuel Valve, [FS2002+]

For Engine 2, simply change the label from Eng1 to Eng2 and the offset address from &H3590 to &H3594.
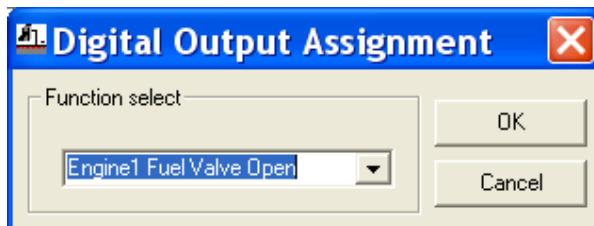
Third Example
Right, the preceeding examples have been pretty straightforward, so now let's take a look at some slightly more difficult applications (there's a reason). This example will look at the Parking Brake which uses slightly different values in the switching part of the line

▷ 0BC8	Brake parking	Controls			0

▷ 0BC8	Brake parking	Controls			32767

		brake=16383). You c
2	**3**	use the byte at 0C00 t [Note: In FS2002 rea written value.]
0BC8	2	Parking brake: 0=off,
0BCA	2	Braking indicator: br

Parking Brake Offset Address is 0BC8 from the programmers doc. Byte info is 2.   Label is Parking Brake.   So first part of line is:

"Parking Brake","&HBC8",2

Now lets take a look at the switching values.

Look at the two interrogation lines on the last page. Parking Brake OFF value is 0. Parking Brake ON value is 32767.
Now look at the MCPEX manual for switching types for digital out actions.

2 Type and actions for Digital output

Type	ValOff	ValOn	Action
0	not used	Any within the range	Output activate when: Actual value = ValOn
1	OFF threshold	ON threshold	Output activate when: Actual value > ValOn Output activate when: Actual value < ValOff
2	not used	not used	Output activate when: Actual value <> 0
3	low value	high value	Output activate when: Actual value included between ValOff and ValOn
4	not used	Logic AND mask	Output activate when: (Actual value AND ValOn) <> 0

Now we know that the parameter values for the PB are 0 for off and 32767 for on. It then follows that we can use switching action 2, because that will switch on when the value at the address is anything other than 0. Getting the idea :o))
As ValOff and ValOn are 'not used', all that needs to be in these fields is 0 just to make the line 6 fields long.
There you have it, Parking Brake line is **"Parking Brake","&HBC8",2,0,0,2**
I use it to light a Red LED on my TQ when my Parking Brake is set.

Fourth Example
This one reads the parameter value in offset address 0366. This indicates whether the aircraft is in the air or on the ground.
So again with this information it's easy to write the first 3 values in the control line.
"Aircraft on Ground","&H366",2
Now have a look at FSInterrogate to see it does indeed work.

		50KHz)
0?3C	**3**	ADF frequency settable
0366	2	Aircraft on ground flag

▷	0366	Plane on ground	Pos./Attitude	1

▷	0366	Plane on ground	Pos./Attitude	0

Now decide what switching action you want and how you want to switch it.   Do you want the relay switched when the plane is on the ground or switched when it is in the air ?   For my example, I'm going to switch it when on the ground. Now I know that the value at 0366 is 1 when on the ground, so that's my ValOn. So, it appears that switching option 0 is my best bet. That means ValOff is not used so we can just put 0 in there. ValOn will be 1 (plane on ground) and the switching type will be 0.   That means the command line will be: **"Aircraft on Ground","&H366",2,0,1,0**

Digital Output Assignment

Function select

Aircraft on Ground

OK

Cancel
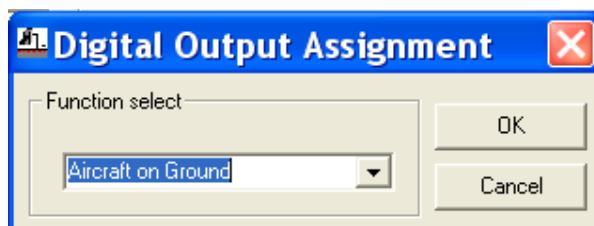
Multiple Condition Outputs.

There may be occasions where you need to apply a Multiple Condition Switching scenario. For Example, I automatically make GND PWR available when I am at the Gate/Stand on the APU. Parking Brake Set with both engines stopped.   This is part of the reason I used some of the above examples.

It then follows that I need to fulfil 4 conditions to have that happen.

Have a look at this diagram and then I'll explain what I mean.
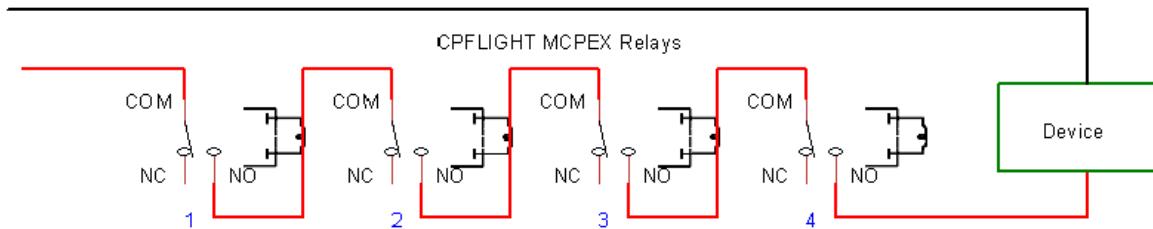
Four relays linked. 1. Parking Brake Set. 2. Aircraft On Ground. 3. Engine 1 Combustion Flag (Stopped) and 4. Engine 2 Combustion Flag (again Stopped). As each condition is met, the corresponding relay will switch and feed the common on the next relay.

When all 4 conditions are met, there is continuity between the input at relay 1 and the output at relay 4 thus switching the device (in my case a switching circuit) delivering ground power to the aircraft. If for example, an engine is started, that control line in FS_COM will turn the relay off cutting the continuity and the control circuit will turn off, automatically disengaging GND PWR.

I hope this has been helpful to somebody. The reason it came about is twofold. A couple of people asked me about doing this and secondly, Mark Deponeo has got me involved in a project that will carry the idea onto as much of the overhead as possible.   Even automating the circuit systems.

As always thanks for your time and if you know a better way to do something, tell me, don't keep it to yourself.

Ian

Oswestry, UK 4th June 2009.

ian@737ng.co.uk

www.737ng.co.uk