

**SIMPLE ENCODER FOR THE BU0836 (Using a Rotary Switch)**

I was looking for a solution to accurately control the position of rotary switches on the Overhead Panel using the BU0836X board and the FSUIPC 737 OHD macro's. Examples being the Wipers, Panel lights and the Pressurisation Auto/Man/Alt controls. Now the wiper controls (like the autobrake) are a fixed no. of positions.



Not only that, but they only have an Increase/Decrease Position level of control and I needed to be able to accurately set the position. For example, from off if I wanted a Low Setting, I needed to move the switch two positions and for

a high setting I needed to move the switch 3 positions. So my first thought was to use a Rotary Encoder on the BU0836 board and let FSUIPC see the button press and apply Wiper + and Wiper - from the available macros. Then it occurred to me that a 2 bit gray encoder is only two sets of contacts having a relationship with a common terminal. The changing relationship telling the control (either software or hardware) which way the encoder had been turned and by how many positions. So use this 'switch state change' to our advantage.

ADDED 24<sup>th</sup> July 2008 The last section deals specifically with making a simulated encoder from a one pole rotary switch. Ideal for the Autobrake. Connect to the BU0836X, let FSUIPC see it as two button presses and use Pete's 'Mouse Macro Technique' to Increase/Decrease the position. This came about by not being able to find 45 degree index two pole switches.

If you understand the operation of an encoder, skip this part and go onto the next bit. But for those who do not understand how 2 bit encoders work, here's a basic rundown.

**TRUTH TABLE 2 BIT BINARY**

DENOTES CONTACT CLOSURE TO COMMON	B	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
	A	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
POS. NO.		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

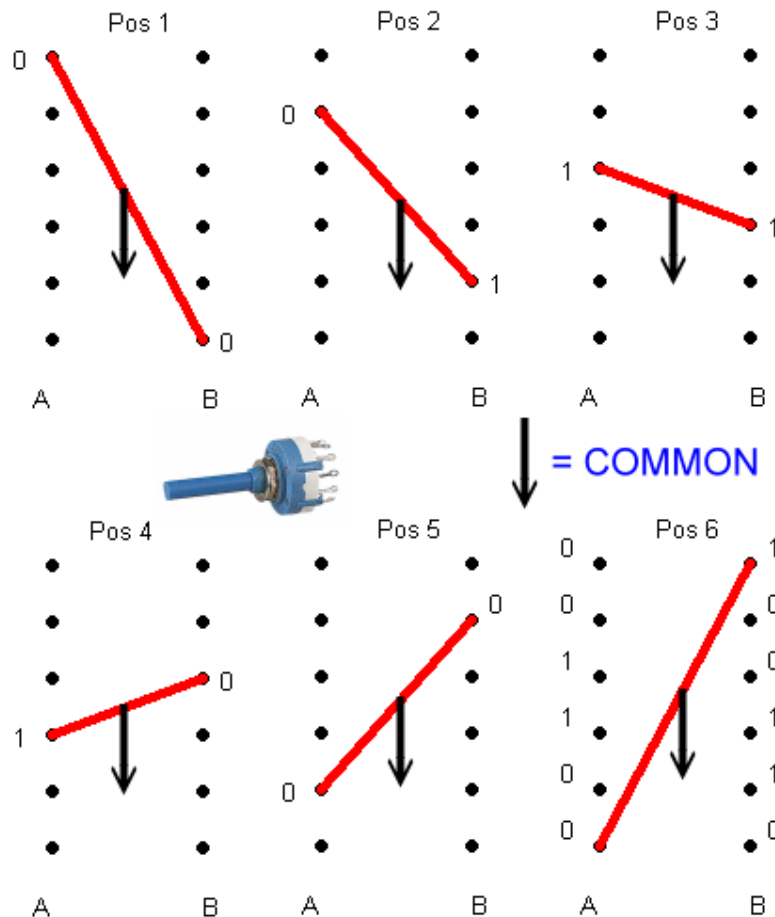
A 2 bit encoder has 3 connections. That is 2 switching contact circuits and 1 common. In the above table you will see that the switching contacts are called A & B. 0 represents an open contact and 1 represents a closed contact. In position 1, neither of the switching contacts are in contact with the common. In position 2 only A is in contact with the common and in position 3 both the switching circuits are in contact with the common and so on. Taking an example, say position 3, both the switches are in contact with the common. Rotating the switch one détente anti clockwise to Pos.2 means only A is in contact. Rotating the switch one détente clockwise to Pos.4 means only B is in contact with the common. So rotating the encoder one détente either way has changed the relationship of the switching circuits to the common. It is the change in this relationship that your control sees and acts upon. So again changing the encoder from Pos.3 to Pos 1 means two switch change states have occurred. The first change tells the control which way you have rotated the encoder and it then starts to count the number of switch change states and do what it is instructed to do for each change. I hope that all makes sense :o))

**NEXT BIT** It is this Switch Change State that I decided to take advantage of. Now we know newer (after Jan 08) BU0836's (and the BU0836 X board) can read 2 bit encoders, so I thought 'let's take advantage of this ability'. I came up with a 'cunning plan' to wire a simple Rotary Switch to emulate the switching behaviour of a simple 2 bit gray encoder.

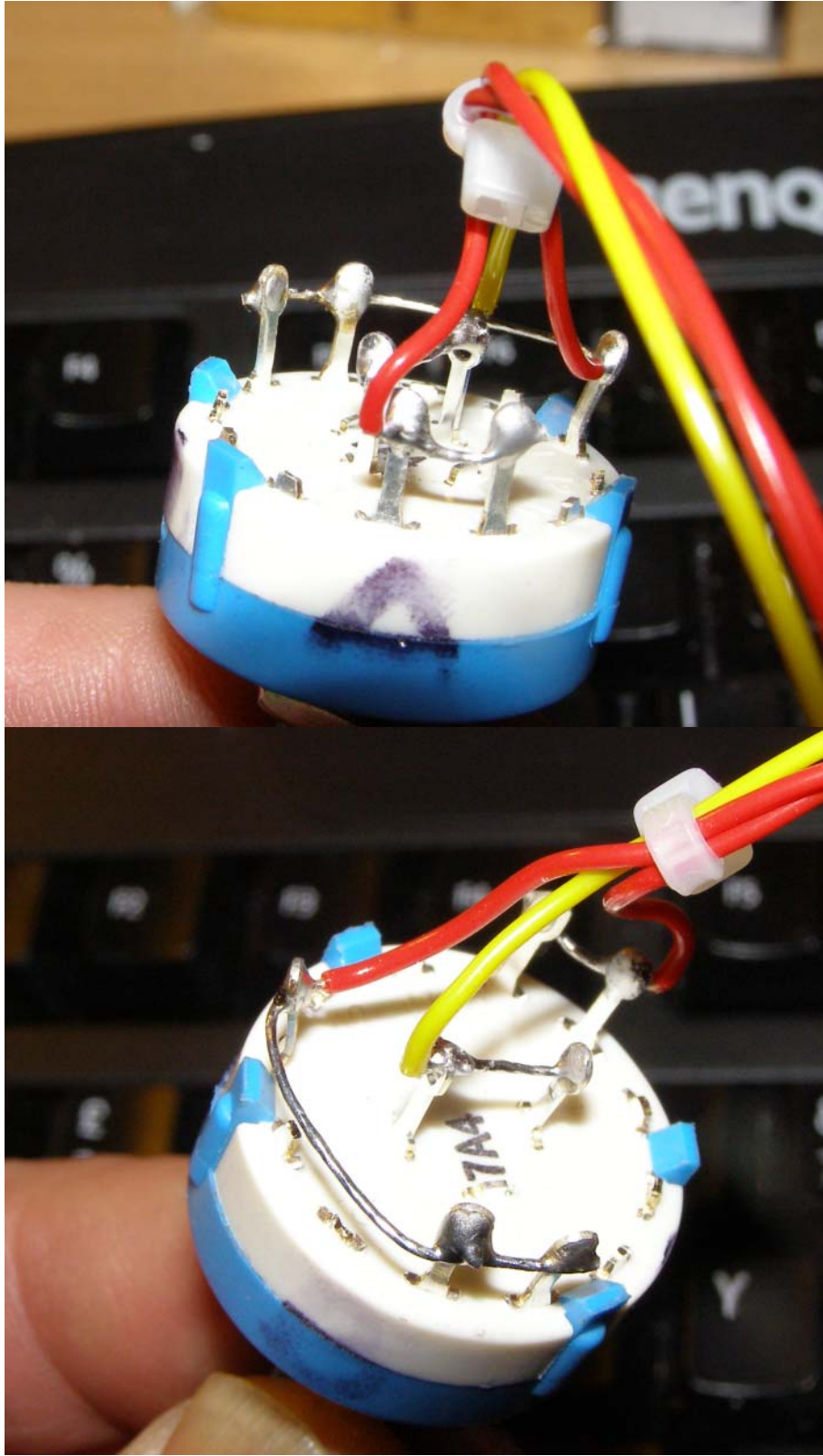


This is a 2 pole – 6 way rotary switch. That is it has 2 centre pins each connecting to 6 outer pins as the switch is rotated. Remembering the Open/Close relationship to the common, it then became possible to emulate a 2 bit encoder by using the two banks of 6 to act as the A & B groups. By joining the two centre pins, we now have the 'common' connection. That leaves us two sets of 6 outer pins for connection as the A & B groups.

Now look at the diagram below. As you rotate the switch different contacts in the A & B groups are in contact with the common. So by connecting certain pins to the A & B leads to the control card, it then becomes possible to simulate the switch change state that a real encoder makes. Except, this time, we have them on a rotary switch with distinct 'Click Positions'.



Looking at the above diagram, you will see 2 groups of 6 contacts A & B. This is how we determine which pins we are going to connect to so as to achieve our correct 0 / 1 connections in the correct sequence. So, by following the table on page 1, we can determine which contacts need to be in contact with the common at each position to achieve that. (Ok I know my example lettering and the Table lettering differ, but I did the diagram before I stole the image :o) )

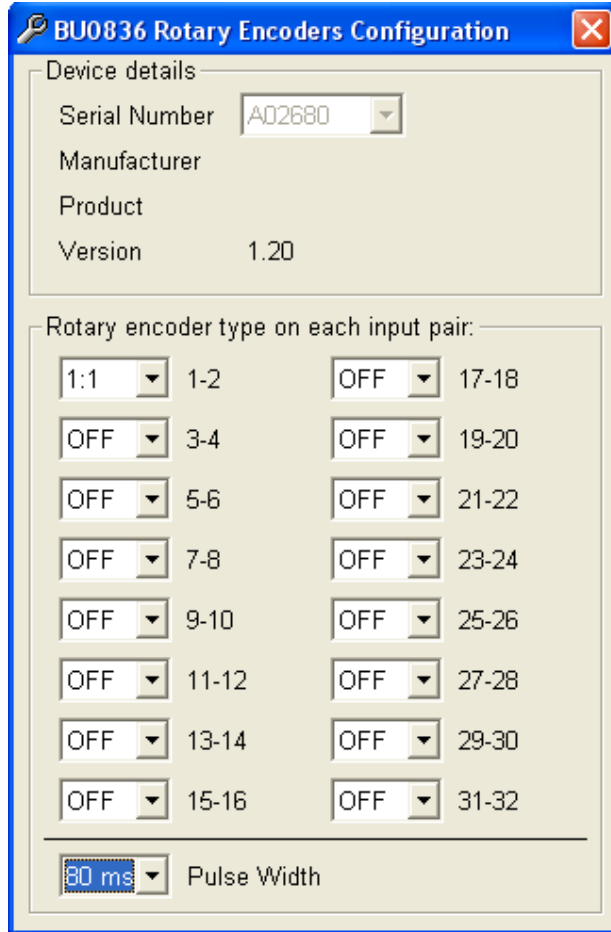


Couple of Shots of the A,B & Common Leads Connected to the switch.

Referring to the Diagram on Page 2, lets see which pins we need to connect in the A & B group. For the A group (starting from the top) it's pin 3 & 4 and for the B group it's pin 1, 4 and 5. I've cut off the excess pins because we don't need them. There you have it, a 6 position 'encoder emulator'. It doesn't have to be attached to Leo's card, could work with anything that handles 2 bit gray input. The BU0836 is the controller of choice here because it is inexpensive, versatile, robust and never gives me problems. Please remember, you can set the number of stops on the rotary switch. Your maximum range is 6 steps. So for the Wipers, I only need 4. Job done.

This part only applies if you are connecting your 'pseudo-encoder' to the BU0836 or X cards.

First get the encoder config utility from Leo Bodnar's website. Plug in your controller and



run the utility. You will see a screen like this. Select your controller (they all have a serial no.) if you have more than one attached.

Select a button pair you want to connect your rotary switch (sorry I meant encoder) too and select 1:1. If OFF is selected, the inputs act like standard joystick buttons. In this example Button Pair 1 –2 have been selected to accept an encoder input, so this is the pair we will attach the rotary A & B leads to. For a BU0836 card these will be Column 1 & 2 and for the BU0836X card it will be B1 and B2. Re: the common lead, on the standard card this is connected to Row 1 and on the X board it is connected to the GND of either B1 or B2.

Point to note: if you are using a standard card, it is advisable to use diodes in the A & B leads to prevent Loopback. Please refer to Leo's website for correct installation of diodes.

Now this example shows a Pulse Width of 80ms. I generally use 48ms as a starting point which usually gives me good results.

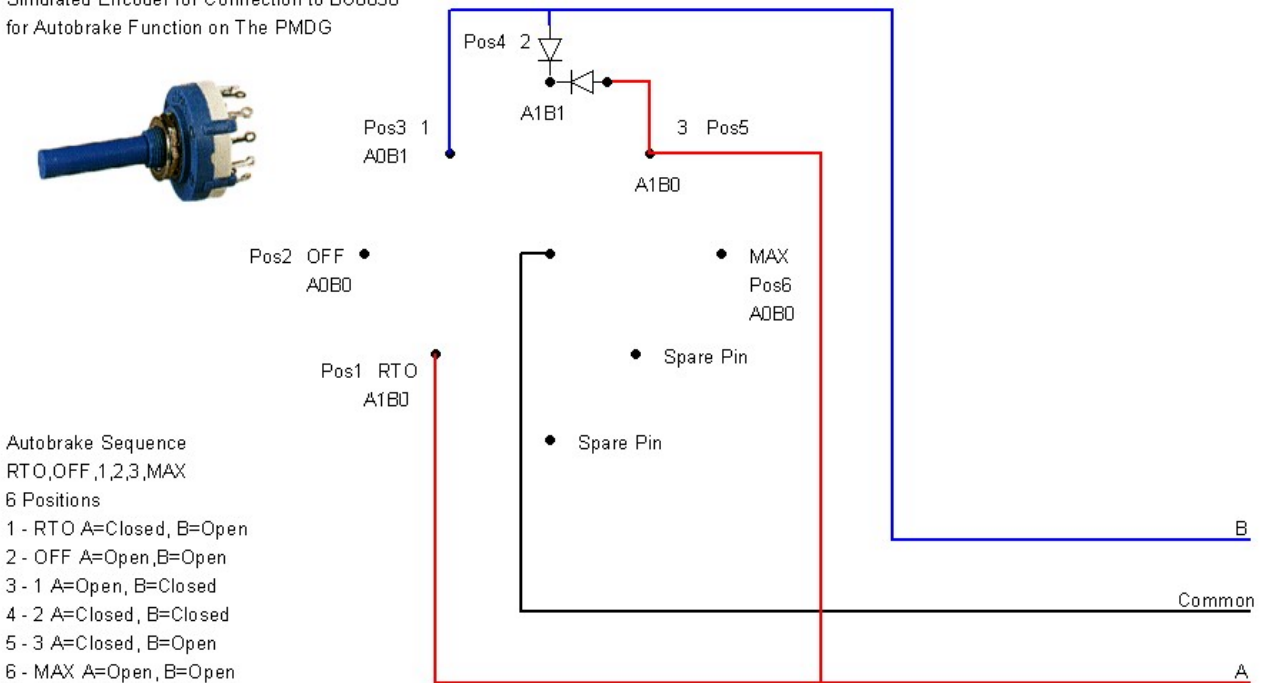
There you have it. It works and it works well. Rotate the switch one way and FSUIPC sees it as Stick# Button 0 and the other way FSUIPC sees it as Stick# Button 1 to which you can apply your 737 OHD macro's from the Dropdown Menu. Job sorted .... :o))

I would suggest that you use good quality rotary switches. I found in my 'early' experiments that cheap rotary's gave me the odd 'bounce' issue.

## Manufacturing a Simulated Encoder From a One Pole Rotary Switch

Then the next step was I had a couple of mates who wanted to use 45 degree index switches for their MIP Functions, most notably the Autobrake. Now you need 6 positions for this being RTO,OFF,1,2,3 and MAX. But single pole 45 degree index switches give you 8 positions, so it followed that a 2 pole switch is only going to give you four. Not enough. So off I goes to the workshop and looked at the problem. This is the solution:-

45 Degree Index 1 Pole 8 Way Rotary Switch  
Simulated Encoder for Connection to BU0836  
for Autobrake Function on The PMDG



Let's go thru it. If you look at the contact table on page 2, remember the sequence of Open/Closed relationship with the common? What happens is that as you rotate the switch, either the A or B or both are in contact with the common to give us the rotational direction and position of the encoder (or in our case the 'Simulated Encoder'). Then I started to draw out the 00 10 11 01 00 10 11 etc sequence. Looking at this allowed me to work out where A & B needed to be connected to make the rotary appear to be an encoder.

This was relatively easy until I came to the 11 connection. It is essential that both Letter Groups are kept isolated from each other at all times except when we intentionally want them both connected to the common. Even then, they have to be connected independently and not to each other.

This where the diodes came into play. So remembering that for the Autobrake 6 positions were needed so the switch was set to only give that many positions using the stop collar. I decided to start at the OFF Position (where the Autobrake is most of the time Except for T/O and Landing) and give it the 00 portion in the sequence. From there I could work up for the 1,2,3 and MAX positions and down for the RTO. This meant that there was only one position that had both the A & B lines connected to the common.

Here's the sequence RTO=10,OFF=00,1=01,2=11 (both closed),3=10 and MAX=00.

Looking at Position 4 (the 2 setting on the Autobrake), both the A & B lines are connected to this pin thru 1N4001 diodes to prevent the two lines connecting to each other. Result a 6 position 45 degree index rotary switch that acts like a 2 bit gray encoder. Can be connected to any interface board that handles 2 bit gray (1:1) encoder input. **RESULT !**

### **And There's More !**

The above example is aimed at the Autobrake. Distinct click positions based on a 45 degree index. But, you may want to use the whole 8 positions for selecting items in your cockpit. In that case, every time you come across the 'both lines need to be connected to the common' scenario, you will have to fit diodes on these inputs to the pins.

If you look at my diagram above, I've used 6 positions, there are two spare pins. These are Position 7 and Position 8. Following my connection regime, Position 7 would have A0B1 as it's connection and Position 8 would have A1B1 as it's turn in the sequence, which of course means these connections would have to be thru diodes.

Look at the table on page 2, you will see that every fourth connection sequence is where A & B both need to be closed with the common to retain proper operation. So, it then becomes possible to connect any switch for any number of positions.

For example:

1 Pole 12 Way (30 degree Index) = Any number of switch positions upto 12. Simply set the stop ring to limit the number of positions that the switch can operate in.

1 Pole 8 Way (45 degree index) = Any number of switch positions upto 8. Again just set the stop ring for the number of Positions you want.

### **And There's Even More !**

Here's a new twist on the idea. If you split the rotary switch (do it in a plastic bag so the ball bearings don't go flying off), when the shaft is pushed out of the housing, you will see a stop lug on the boss. Carefully removing this stop lug with a craft knife enables the switch to rotate as many stops in any direction as you need. You are no longer limited by the number of positions on the switch.



But remember your switch has to have a number of positions that you can divide by 4 to give correct sequencing of the A/B/Common relationship.

There you go, home made encoders from Rotary Switches connected to the BU0836X board and delivered into FS by FSUIPC.....the world has changed forever Your switch positions now match the lines on your MIP/Overhead or Whatever :o))

So another problem sorted, I'm a happy Laddie. Think I deserve a Beer.

As always, thanks for your time and any comments or improvements, please feel free to let me know..... Ian

24<sup>th</sup> July 2008

[www.737ng.co.uk](http://www.737ng.co.uk)

[ian@737ng.co.uk](mailto:ian@737ng.co.uk)