Virpil Configuration Software Guide

Mazex Edition (unofficial)

Version history

Version	Date	Change Description
0.1	2019-05-13	First draft
0.2	2019-06-06	Added chapter 4 and 5
0.3	2019-06-06	Fixed some grammar
0.4	2019-06-06	Updated 2.4.3 (typo in button numbers)
0.6.1	2019-10-05	Updated for v. 190913 of the software. Added encoders
0.6.2	2020-03-24	Error in 3.1.1 and 3.12 (button 53, not 33 for B1 mode 2).

TABLE OF CONTENTS

1 Introduction

The Virpil control devices are top of the line devices for PC simulator enthusiasts. One of the few problems with the devices is that the configuration software is quite hard to use, and the documentation of it is not complete yet.

The goal of this document it to add some documentation about parts of the configuration that the current manuals don't cover. It is based on the assumption that you have read the official documentation and adds missing chapters or information based on that.

I started writing it for myself to remember how I fixed my setup as I know that it's easy to forget in 6 months and annoying to try to get it right again. Then realized I could tidy it up and let others get some help from it…

The device I do remappings for is the **Moongost T-50 Throttle** (v2) but it should work for other devices in the same way (numbers for buttons may differ though)

The guide is based on the Virpil Configuration Software version 2020-02-04 (*VPC-JOY-200204*).

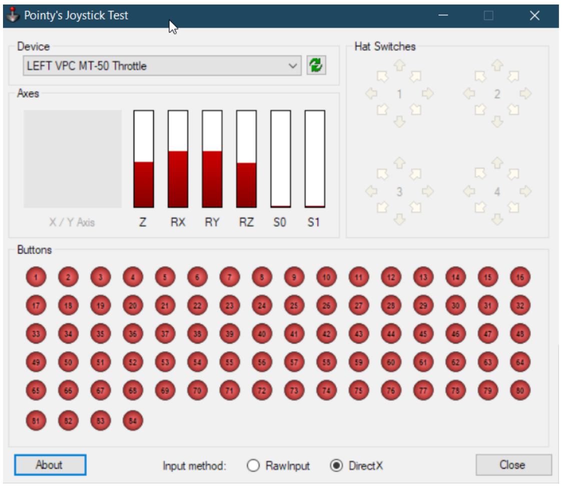
1.1 Intended audience

People that want a long text explaining the way things really work. Or rather, my guess how it works… So this is not for the ones that want a quick "do x, then y and you are done". Look for Youtube videos in that case.

1.2 Tools used

To test the mappings in windows I found this app way back when and it is simple but does what it should – shows the windows button and axis events for an attached device – useful when checking which actual windows button that is sent from the Virpil device when pressing a physical button.

And yes – I use the built in "VPC Joy Tester" that is included in the Configuration software as well, but it is interesting to just verify what an old pre Windows 10 app from 2014 sees the same thing as a verification. Some of our games are in the same situation.



Device: LEFT VPC MT-50 Throttle

Input method: ○ RawInput ● DirectX

http://www.planetpointy.co.uk/joystick-test-application/

1.3 References

Link to the official configuration software and manuals:

https://virpil-controls.eu/downloads.html

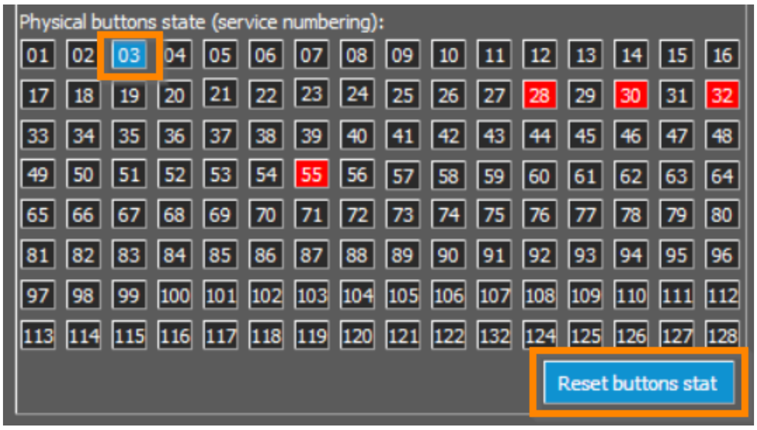
2 Basics

Here we cover some of the basics of how the Virpil configuration software works that are important to understand for the more complex mappings we can do. You are assumed to have read the official documentation and know how to set up basic profiles, load device configurations etc...

2.1 Physical buttons vs logical

When you press a button on your Virpil device – this is known in the Virpil configuration software as a physical button. This is however NOT the actual “button x pressed” event that is sent to Windows.

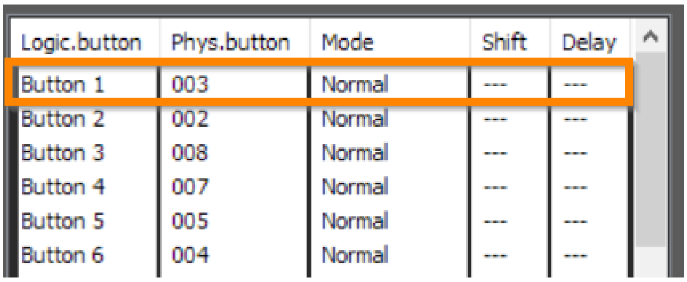
So – if I press the “pinky” button on the Virpil Throttle (the one sticking out on the left side), it is the physical button “03”. On the “Physical buttons state” part in the UI button “03” will be red when I hold it down, and then it will remain blue like below when I release it. If you want to clear the “memory” of buttons pressed previously (blue) you can press the “Reset buttons stat” button in the lower right corner.



Physical buttons state (service numbering):

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96
97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112
113	114	115	116	117	118	119	120	121	122	132	124	125	126	127	128

So – to see the actual button event that is sent to Windows you have to check the mapping table in the right part of the UI. Here we look in the column “Phys.button” for 03 and find “003”. And there is a mapping here to a “Logic.button” which is what Windows will get, “Button 1” in this case.

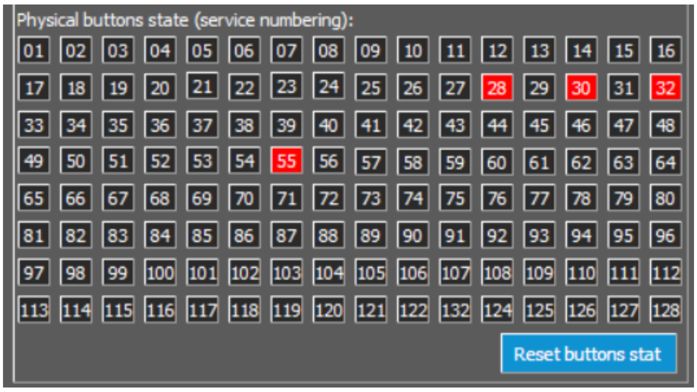


Logic.button	Phys.button	Mode	Shift	Delay
Button 1	003	Normal	---	---
Button 2	002	Normal	---	---
Button 3	008	Normal	---	---
Button 4	007	Normal	---	---
Button 5	005	Normal	---	---
Button 6	004	Normal	---	---

Actually – Windows will get “BUTTON 1 ON” when you depress the button and then the “BUTTON 1 OFF” events when you release it – but DCS sorts that out as “pressed” when it gets the ON event and ignores the OFF event. You can see this by opening the “VPC Joy Tester” app by pressing the blue button in the bottom of the UI. Select the Throttle in the drop down list of the test app first.

2.2 Special button types

OK – so why are then the physical buttons 28, 30, 32 and 55 red all the time? (NOTE: it might be that 29 or 31 are on instead of 28 and 30 below).



Physical buttons state (service numbering):

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96
97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112
113	114	115	116	117	118	119	120	121	122	132	124	125	126	127	128

These are actually the T1, T2 and T3 switches being up or down, and the mode dial set to one position. They are sending the physical button "pressed" all the time for the current state they are in opposed to the other buttons that are not currently "pressed".

Let's go over the types of buttons and switches that are available and how they are configured.

2.3 Normal buttons

There are many buttons on the Virpil devices, and almost all of the hats are just acting as normal buttons when pressed in a direction or pushed.

If we want to change an assignment for a button we map them in the mapping table to the right. We map a physical button to a logical (Windows) button event.

The B1-B9 buttons are a bit special in the default configuration as they are the only ones that have different shift states pre-configured. When you turn the mode dial they will send different logical button events to Windows.

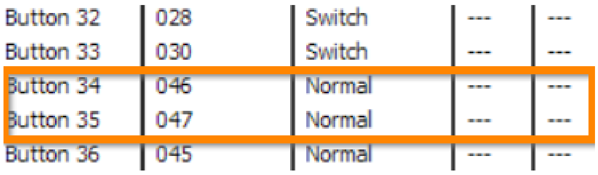
2.4 Switches

The different switches T1 to T6 are mapped a bit differently. As T4-T6 are the most ”normal” and T1 the ”strangest” we go through them in reversed order…

2.4.1 T4, T5 and T6

T4-T6 are really not special in the configuration. They are actually mapped as normal buttons where up is one button and down Is another button. Simple as that.

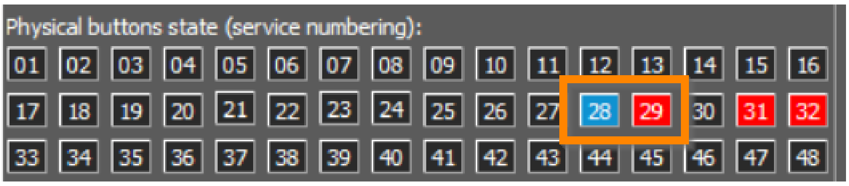
Below is the default configuration for the T4 switch where up (phys 46) is mapped to Windows 34. Down (47) is mapped to Windows 35.



Button 32	028	Switch	---	---
Button 33	030	Switch	---	---
Button 34	046	Normal	---	---
Button 35	047	Normal	---	---
Button 36	045	Normal	---	---

2.4.2 T2 and T3 switches

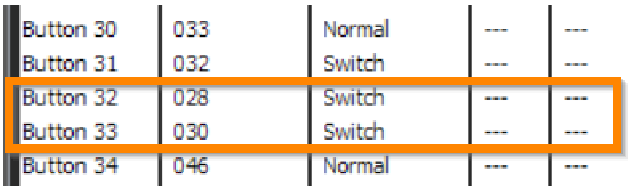
Flip the T2 up and down. You will see that physical button 29 is on all the time when it is in the UP position. When you flip it down it will send button 28 pressed all the time instead. For T3 it is 30 (down) and 31 (up).



Physical buttons state (service numbering):

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48

Scroll down in the mapping list to the right in the UI. You will find that the physical buttons 28 and 30 are mapped a bit down in the list as the button type “Switch” instead of the “normal button” mapping we have for T4,T5 and T6.



Button 30	033	Normal	---	---
Button 31	032	Switch	---	---
Button 32	028	Switch	---	---
Button 33	030	Switch	---	---
Button 34	046	Normal	---	---

IMPORTANT TO UNDERSTAND

So, why is only physical 28 (T2 down) and 30 (T3 down) mapped as “switches”? And why is there no mapping for 29 (T2 up) and 31 (T3 up)?

Because they are set up as switches where you can actually pick either the up or down event as **Virpil wanted them to send the SAME command** when you change the position. It does not matter if it's up or down.

When you flip T3 up it will start sending physical button 31 ON – but it will also send the OFF event for physical button 30 as that gets "released". And as physical button 30 is mapped as a switch (see image above) – that will send the button 33 event to windows – even though it was actually "released" when you moved to the up position. A note is that Windows will get the "button 33 down" event for about 250ms every time you flick it the switch.

In chapter 4 we will look more at T2 and T3, and how to change to send different events to Windows instead for up or down.

2.4.3 The T1 Switch (red cover)

The T1 button (with the red cover) is more "special" than T2 and T3. It is a three way switch like T4,T5 and T6 that is physically pressed to the "down state" by the red cover when that is closed. If you open the lid it will go back to the spring loaded center position that has no button assigned to it. So, when the cover is down, button 32 is pressed all the time. That is why 32 is red in the config screen.

It really has two physical buttons linked – button 32 (down) and 33 (up). These are however mapped differently in the configuration.

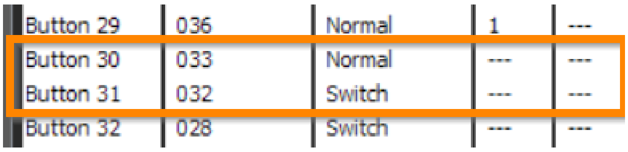
SO – when opening the lid the physical button 32 "OFF" event is sent as the switch moves from being pressed to the center position – and that is mapped to windows as "Button 31 pressed".

AND – when you put the cover down again causing the physical button 32 "ON" event – this is linked so that it ALSO sends the "Button 31 pressed" event to Windows.

BUT – when physical button 32 stays pressed down (lid closed) it will NOT send any "pressed" button to Windows as it is set up as a switch. It will be "red" (on) in the Virpil config screen though...

This is done by using the same switch behavior for the down event as the T2 and T3 use to send the same command when you flick them...

Looking in the mappings configuration we see that physical 33 (up) is a "normal" button sending button 30 to Windows when pressed, while physical 32 (down) is a "switch" type - acting on the "state changed" to send a specific command and not ON or OFF depending if you press or release it. So - if it would have been a "normal" button it would have been sending "ON" all the time when the lid was down. Confused? Starting to get hold of it? As DCS will not react on the "OFF" events we would not have been able to use the fact that is was released when opening the lid.



Button 29	036	Normal	1	---
Button 30	033	Normal	---	---
Button 31	032	Switch	---	---
Button 32	028	Switch	---	---

Example usage for T1

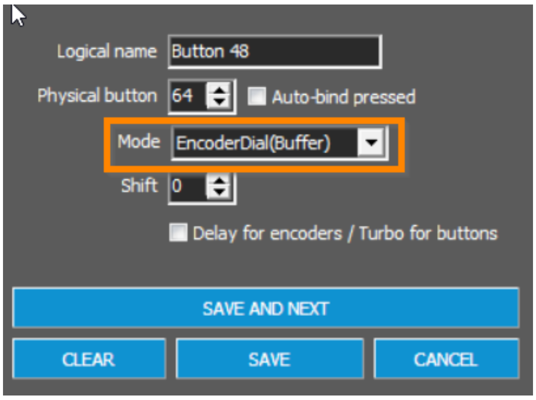
So you can map one event that happens when you open or close it – and then one when you press the switch up. I have set the “open /close lid event” (button 31) to “Arm ejection seat” in DCS F/A 18, and the press up event to eject (3 times). So when I run through the startup I flip the cover up to arm the ejection seat, and then close it (thus double arming my seat as it will send the same arm command both for opening and closing the lid). And IF I forget this during startup – it will do it in the air if I need to eject as I cant get to the button without opening the lid…

2.5 Encoders

Encoders are the rotary encoders E1 to E3, that also has a “normal button” push function.

Getting the encoders to work properly in the default configuration for DCS is tricky – as the encoders will send a stream of “button down” events pretty fast, and many gets skipped. So if you map an encoder to a screen brightness dial etc in DCS it will move erratically, and you need to turn it very slowly to make it work in DCS.

My recommendation is to change the encoder dials to the mode ***EncoderDial(buffer)***. This way all the “clicks” will be stored in a buffer and sent in a slower stream to Windows – so it does not matter if you turn the encoder fast. All the clicks will get sent – but it will take a while, so the ingame dial will continue to rotate after your fast turn of the encoder is “done”.



Logical name: Button 48

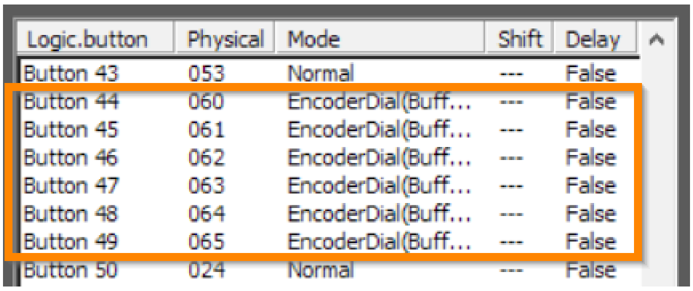
Physical button: 64 ☐ Auto-bind pressed

Mode: EncoderDial(Buffer)

Shift: 0

☐ Delay for encoders / Turbo for buttons

Do this for all six encoders (there is one logical button mapping for each direcion of a dial – so logical button 44 is E1 knob left (ccw), and logical button 45 is E1 right (cw) etc).



Logic.button	Physical	Mode	Shift	Delay
Button 43	053	Normal	---	False
Button 44	060	EncoderDial(Buff…	---	False
Button 45	061	EncoderDial(Buff…	---	False
Button 46	062	EncoderDial(Buff…	---	False
Button 47	063	EncoderDial(Buff…	---	False
Button 48	064	EncoderDial(Buff…	---	False
Button 49	065	EncoderDial(Buff…	---	False
Button 50	024	Normal	---	False

3 Shift Modes

And the button 55 physical button being pressed all the time is actually the mode switch that in my example is set to position 1. Change it and see that is will go to 56,57,58,59 instead. We talk more about the mode dial in chapter 5.

IMPORTANT: The following is based on the precondition you have set up the T-50 throttle with the option "Mode dial as shift state" when creating your initial profile. See the official documentation regarding how to set up profiles, or watch this video:

https://www.youtube.com/watch?v=1ULJGPiFAaM

The default profile for the T-50 Throttle makes it possible to use the Mode Dial to send different “buttons” (commands) for the same physical button to your game. The Mode dial has 5 modes. It is important to note that with the default profile it will only add "shift" button functionality for the B1-B9 buttons on the throttle. So, the default profile has “shift commands” for all 5 modes for the B1-B9, **but not for any other switches or hats**.

Note that there are other options than using the Mode Dial to set the “shift mode”. You can for example configure that holding a specific button will cause you to go into “shift mode 2”.

3.1.1 The default Shift mode setup

As it is important to understand the basics, we start with a more detailed explanation of how the standard buttons in the B1-B9 group are set up with the default “shift states” for the Mode Dial.

Mode 1 behaviour

- Set the mode dial to position “1”
- Press the B1 button on the throttle
- The Virpil configuraton software will show that you pressed the button with ID “40”.
    - This is shown in the Virpil configuration software by button 40 being highlighted red.
    - When you release the button – it will remain blue to show that you have used it
    - The device will for mode 1 send the event “Button 22” to Windows (and your game) – start the joystick test app and verify.

Mode 2 behaviour

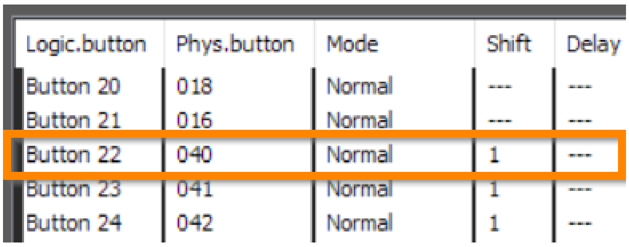
- Set the mode dial to position “2”
- Press the B1 button again
- The Virpil configurator will STILL show that you pressed button “40” as it is the same physical button..
- But to Windows – or your game, it will now send the command “Button 53” as you are in the shift mode 2, as your Mode Dial is in position 2.

Mode 3,4,5 works the same way and are mapped by Virpil in the default config – **for B1-B9**

3.1.2 How are the default shift functions done?

Let’s check the default mappings done for the B1 button we tested in 4.1

- Look in the column “physical button” for “40” (the B1 button)
- Note – as there are 5 different “shift modes” for this physical button there are actually five different rows in the advanced configuration for this button.
- The first row you will find will look like this



Logic.button	Phys.button	Mode	Shift	Delay
Button 20	018	Normal	---	---
Button 21	016	Normal	---	---
Button 22	040	Normal	1	---
Button 23	041	Normal	1	---
Button 24	042	Normal	1	---

- The important thing here is to understand what it means – button 40 will send “button 12” to the game. But - note that it has shift mode 1 set, so it will only send button 22 to windows if the mode dial is set to “mode 1”.
- Scroll further down. You will find another mapping row for physical button 40 for shift mode 2. That will send windows button 53 instead.
- If you continue down you will find mappings for mode 3,4 and 5 as well.

So – this is what we want to play with to add other buttons, switches or hats to map them to different modes set by the dial.

IMPORTANT NOTE: When you look in the advanced configuration list you will see that most of the physical buttons are **NOT mapped to a shift state**. Look at physical button “2” for example that will send the command “button 4” to you game. As that button DOES NOT HAVE a shift set in the configuration – it will send “button 2” whatever way your mode dial is set.

… One problem is that you can only have 128 logical "buttons" / "commands" that can be sent to the operating system. So you can't have shift states for all the switches, hats etc.

4 Splitting the T2 and T3 switches

As explained in chapter **2.4.2** the T2 and T3 in the default configuration are set up as switches that briefly sends the same "button" to Windows when you change the switch position. This means that if you want to use them for two separate "on" or "off" functions in DCS it will not work. They have to be mapped to "toggle on/off" functions instead with the default setup. And then up might be down depending on the state of the switch when you start the game...

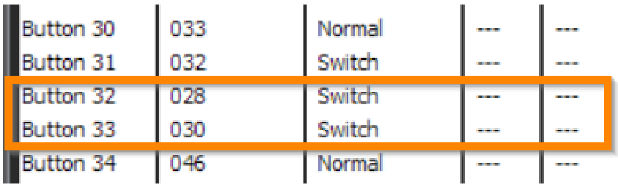
4.1 Our Scenario

For our F18 In DCS, we want to map the T2 "up position" to "Master Arm On" and the T2 "down position" to "Master Arm Off". We want "Exterior lights ON" to T3 up and "Exterior lights OFF" to T3 down.

4.2 Step 1 – modify the original mappings

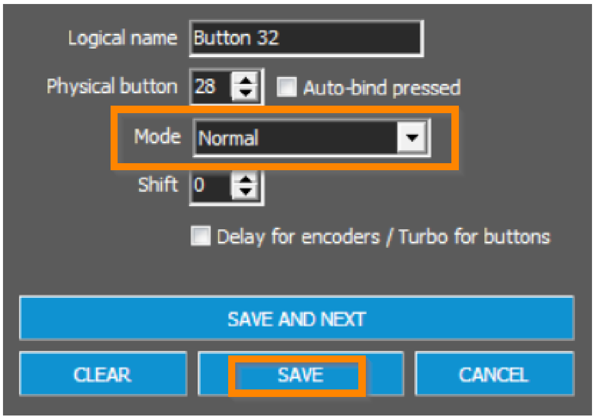
As explained in chapter 2.4.2 the T2 and T3 are a bit oddly mapped in the default configuration.

So we first need to make the physical button 30 (T3 down) and 28 (T2 down) into "real" normal buttons. Double click the first row below (button 028).



Button 30	033	Normal	---	---
Button 31	032	Switch	---	---
Button 32	028	Switch	---	---
Button 33	030	Switch	---	---
Button 34	046	Normal	---	---

Change the mode to "normal". Repeat for physical button 030.



Logical name: Button 32

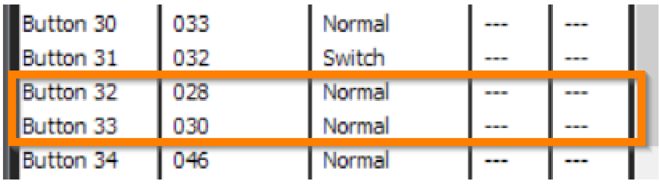
Physical button: 28 □ Auto-bind pressed

Mode: Normal

Shift: 0

□ Delay for encoders / Turbo for buttons

End result:



Button 30	033	Normal	---	---
Button 31	032	Switch	---	---
Button 32	028	Normal	---	---
Button 33	030	Normal	---	---
Button 34	046	Normal	---	---

4.3 Step 2 – add mappings for “missing” buttons

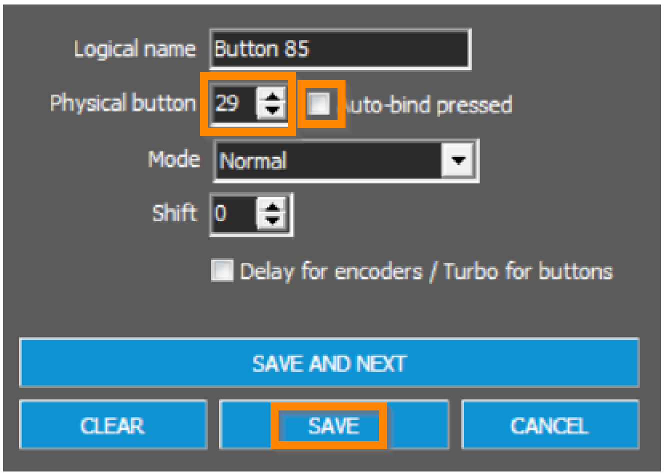
So – physical button 29 (T2 up) and 31 (T3 up) do no have any mappings in the default setup as it relied on 30 and 32 being switches.

Lets add them:

Go to the end of the mapping list and double click on the first “empty” row (normally logical Button 85).

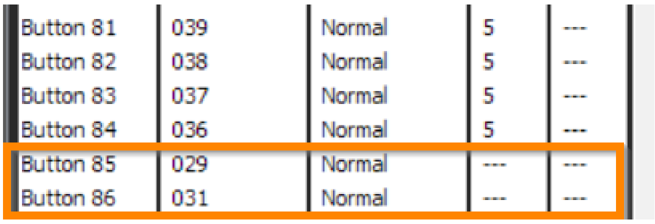
Start by unchecking “Auto-bind pressed button” as you will get the one the mode dial is “pressing all the time otherwise – or some of the T1, T2, T3 switches”

Set physical button 29, mode normal and save. You might click “SAVE AND NEXT” to get directly to the next entry as well which works here as we want to add one more (31). So add the same setup for physical button 31.



Logical name: Button 85
Physical button: 29
□ uto-bind pressed
Mode: Normal
Shift: 0
□ Delay for encoders / Turbo for buttons

End result:



Button 81	039	Normal	5	---
Button 82	038	Normal	5	---
Button 83	037	Normal	5	---
Button 84	036	Normal	5	---
Button 85	029	Normal	---	---
Button 86	031	Normal	---	---

Done – go to DCS or your sim of liking and map separate buttons for up and down position.

5 Adding shift functions to the Mode Dial

5.1 Add shift modes for the T4, T5 and T6 switches

Our Scenario: We want to map new commands for mode 1 – 3 for the T4-T6 switches

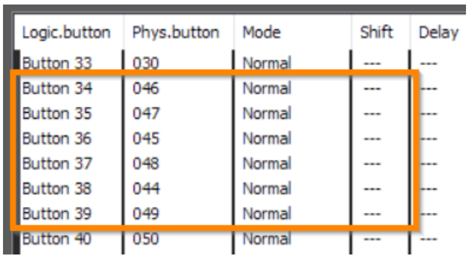
5.1.1 Step 1 – update the original mappings

As the T4-T6 switches do not have any shift state in the default configuration we need to start by adding “shift mode 1” to the original mappings.

The T4-T5 act as two “buttons” each where they send one physical button for up and one for down according to the list below. The mapping is a bit odd where the range starts at 44 and goes to 49, starting from T6 up, then T5 up etc.

Lets find them in the advanced mapping table:

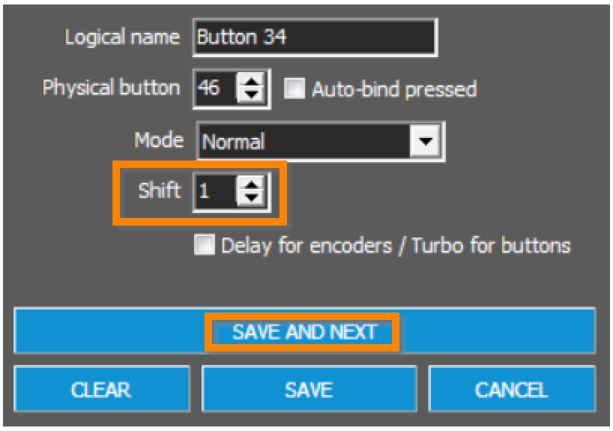
Switch	ID: Pos UP	ID: Pos DOWN
T4	46	47
T5	45	48
T6	44	49



Logic.button	Phys.button	Mode	Shift	Delay
Button 33	030	Normal	---	---
Button 34	046	Normal	---	---
Button 35	047	Normal	---	---
Button 36	045	Normal	---	---
Button 37	048	Normal	---	---
Button 38	044	Normal	---	---
Button 39	049	Normal	---	---
Button 40	050	Normal	---	---

So – lets start with updating the physical button 44 (T6 up). Double click the row for Phys.button 46, change Shift from 0 to 1 and click “SAVE AND NEXT”.

Continue doing this 6 times for physical button **44 to 49** (yes the sorting order is for logical button ID so the order of the Phys buttons is a bit awkward). Just press “CANCEL” when you get to phys button **50** that we are not changing.



Logical name: Button 34

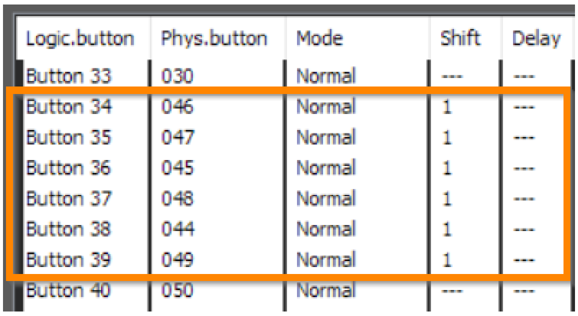
Physical button: 46 □ Auto-bind pressed

Mode: Normal

Shift: 1

□ Delay for encoders / Turbo for buttons

The end result should look like this where all 6 ”buttons” are mapped to shift mode 1:



Logic.button	Phys.button	Mode	Shift	Delay
Button 33	030	Normal	---	---
Button 34	046	Normal	1	---
Button 35	047	Normal	1	---
Button 36	045	Normal	1	---
Button 37	048	Normal	1	---
Button 38	044	Normal	1	---
Button 39	049	Normal	1	---
Button 40	050	Normal	---	---

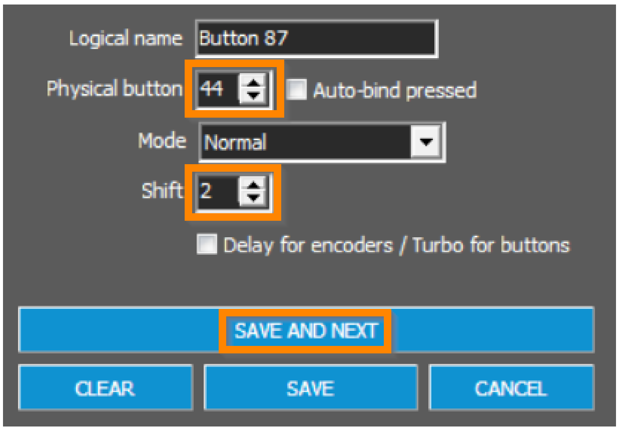
5.1.2 Step 2 – add new virtual buttons

So now we need to add new virtual buttons for shift mode 2 and 3 (or more).

Go to the end of the mapping list and double click on the first empty row. **I always deselect the ”Auto-bind pressed button” as that gets stuck with the mode dial selected button.**

Lets do pysical button 44 first, that is the ”T6 up” event:

- Set Pysical button to 44
- Mode to normal
- Shift to 2
- Click ”SAVE AND NEXT”



Logical name: Button 87

Physical button: 44 □ Auto-bind pressed

Mode: Normal

Shift: 2

□ Delay for encoders / Turbo for buttons

Rinse and repeat for the remaining 5 ”buttons” for shift mode 2!

And when you are done with button 49 for shift mode 2, you start over with button 44 for shift mode 3 etc. The end mapping should look like the one in chapter 6.

5.1.3 Want to add other buttons to a shift mode?

Just follow the same procedure as described above.

Don’t forget to set shift mode 1 to the ”original mapping” though!

6 My final mapping table

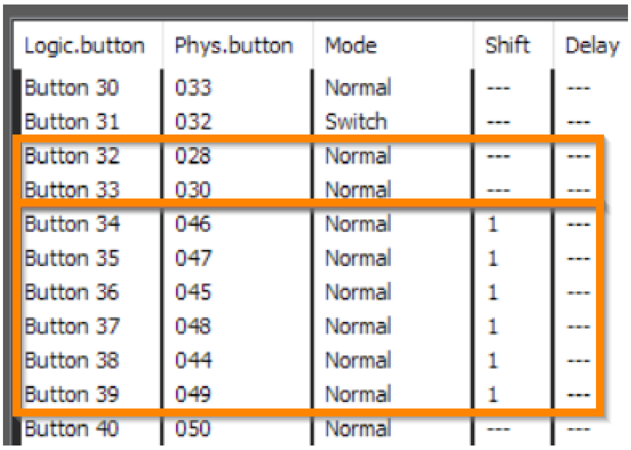
So – in the end, these are my changes for my throttle for adding ”Mode 1-5 shift support” for the T4, T5 and T6 switches, and splitting the T2 and T3.

6.1 Modified buttons

First off, physical button 28 and 30 in the top orange box have been changed to normal instead of switch. This is explained in chapter 4.

Then in the lower orange box – the physical 44 to 49 (the up and down for T4 to T6) have been modified by adding ”shift 1” as we wanted to add other shift states. By default they have no shift state. This is explained in chapter 5.

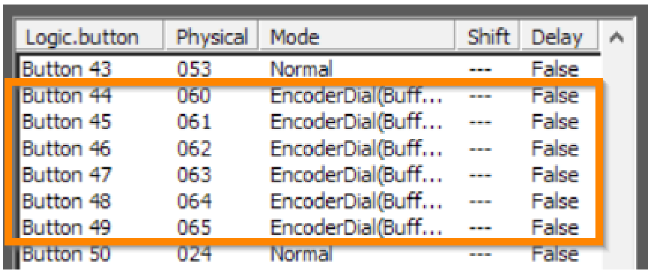
Note that physical button 32 is still on the default setting of a switch, that is the ”T1 down button”.



Logic.button	Phys.button	Mode	Shift	Delay
Button 30	033	Normal	---	---
Button 31	032	Switch	---	---
Button 32	028	Normal	---	---
Button 33	030	Normal	---	---
Button 34	046	Normal	1	---
Button 35	047	Normal	1	---
Button 36	045	Normal	1	---
Button 37	048	Normal	1	---
Button 38	044	Normal	1	---
Button 39	049	Normal	1	---
Button 40	050	Normal	---	---

6.2 Encoder tweaks

As described in chapter 2.5 – I have set the E1 to E3 encoders to buffered:



Logic.button	Physical	Mode	Shift	Delay
Button 43	053	Normal	---	False
Button 44	060	EncoderDial(Buff...	---	False
Button 45	061	EncoderDial(Buff...	---	False
Button 46	062	EncoderDial(Buff...	---	False
Button 47	063	EncoderDial(Buff...	---	False
Button 48	064	EncoderDial(Buff...	---	False
Button 49	065	EncoderDial(Buff...	---	False
Button 50	024	Normal	---	False

6.3 Added buttons (end of list)

Button 84 in the top of the image is the last of the ”default buttons” that Virpil has mapped (it is the B6 button in shift 5 mode).

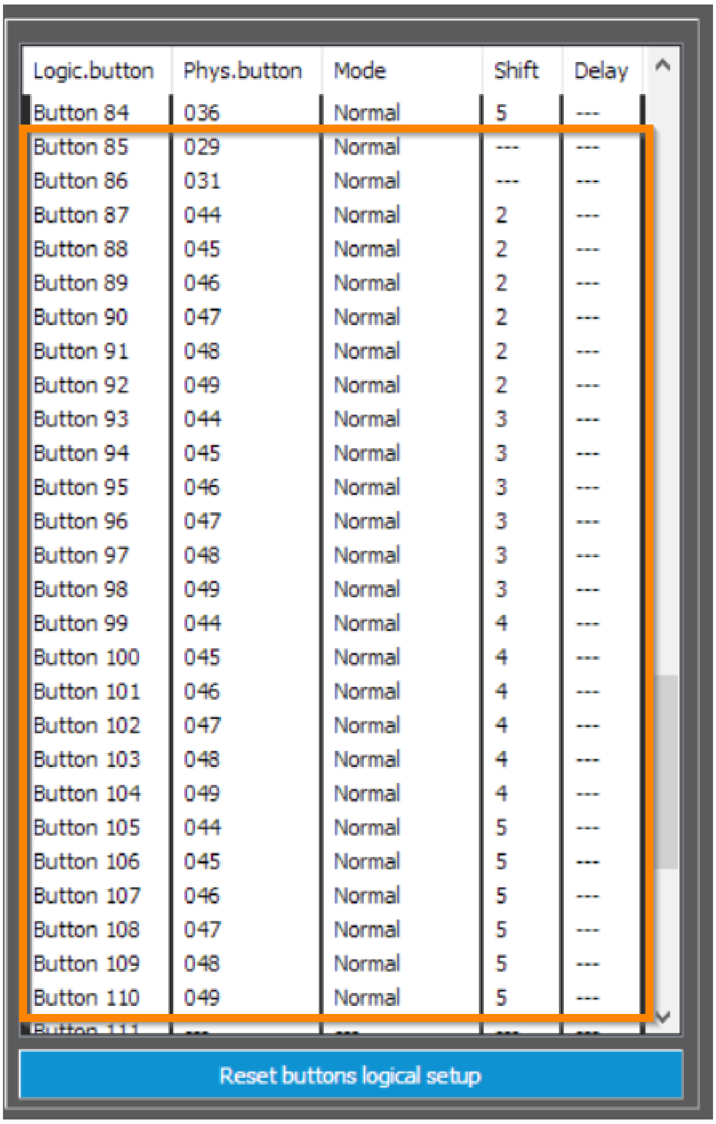
So we start off with the mapping the phys button 29 (T2 up) as a normal button that will send 85 to Windows.

Then phys button 31 (T3 up) as a normal sending 86.

Note – phys 29 and 30 did not have any mappings in the default config as explained in chapter 2.4.2.

Then the rest is all the shift modes for the T4,T5,T6 switches starting with physical 44 for shift mode 2 – that is the T6 up ”button” that now will send button 87 in mode 2 instead of the default 38 for mode 1.

Etc..



Logic.button	Phys.button	Mode	Shift	Delay
Button 84	036	Normal	5	---
Button 85	029	Normal	---	---
Button 86	031	Normal	---	---
Button 87	044	Normal	2	---
Button 88	045	Normal	2	---
Button 89	046	Normal	2	---
Button 90	047	Normal	2	---
Button 91	048	Normal	2	---
Button 92	049	Normal	2	---
Button 93	044	Normal	3	---
Button 94	045	Normal	3	---
Button 95	046	Normal	3	---
Button 96	047	Normal	3	---
Button 97	048	Normal	3	---
Button 98	049	Normal	3	---
Button 99	044	Normal	4	---
Button 100	045	Normal	4	---
Button 101	046	Normal	4	---
Button 102	047	Normal	4	---
Button 103	048	Normal	4	---
Button 104	049	Normal	4	---
Button 105	044	Normal	5	---
Button 106	045	Normal	5	---
Button 107	046	Normal	5	---
Button 108	047	Normal	5	---
Button 109	048	Normal	5	---
Button 110	049	Normal	5	---
Button 111	---	---	---	---