I've received several questions asking for more concise information detailing how the real B737-300 throttle quadrant is configured to operate within the FSX environment. Before I begin, let me state that everyone's system is different; PC hardware, software, drivers and other variables are rarely similar between different computers. Broadly translating, this means that what have done may not be successful for you. However, the concept will be identical.

This topic is not the easiest to write, as the topic is complicated and convoluted. I am *not* a trained professional writer. But, I hope the information will provide enough detail to guide you in the correct direction.

Let's Connect the TQ & Fly…

Connecting the throttle to the computer is not enough to allow the throttle to operate correctly in FSX. There are many variables that need to be configured. These include the following functions:

- Independent forward throttle movement (engine 1 and engine 2)
- Independent reverse throttle movement (engine 1 and engine 2)
- Auto-throttle buttons
- TOGA buttons
- Speed brake handle movement and deployment
- Flap movement and deployment
- Stab trim switch (pre-wired to a button to stop trim wheels from spinning)
- Parking brake
- Horn cut out (currently not used)
- Trim stabilizer movement (trim wheels)
- Trim control indicators

There are two aspects of the throttle conversion.

The first aspect is the revamping and rewiring of the throttle allowing connection of the various throttle outputs to a device that is readable by flight simulator. ***I did not do this***. Rather, I employed a professional to do this on my behalf. I cannot recommend him enough – his name is Art May and his business is called Northern Simulations based in Florida. Art did all the throttle conversion work and rewiring. Art also installed the main Phidget cards and 12V DC servo motor to allow movement of the trim wheels. To allow communication between the throttle and the computer, Art used a standard Leo Bodnar joystick controller card, two Phidget 1066 cards and a Phidget 1064 card.

The second aspect is the configuration and calibration of the various outputs. I did most of this. There are several methods to calibrate the various functions and I decided at the outset to use Phidget cards, FS2Phidgets software and FSUPIC.

Phidgets are used to control functions that require movement - such as the trim tabs and speed brake arm. FSUPIC is used to register and calibrate various buttons in such a way that the commands can be understood by the hardware (throttle and installed cards) and FSX.

Phidgets

Broadly speaking, Phidgets are a set of "plug and play" building blocks that allow you to control, using USB, various sensing and physical controls from a device to your computer. Phidgets allow you to do a lot of things and their application is almost as varied as your imagination.

I needed Phidgets to control the following:

- Speed brake handle movement and deployment
- Trim control indicator movement
- Trim stabilizer movement (trim wheels spinning) – requires servo motor

What Do I Need To Use Phidgets

To use phidgets you need an actual Phidgets card designed for your application. You also need the FS2Phidgets software which, once downloaded, consists of an MSI self extracting file and a Phidgets Library. The MSI file will install the program onto your computer. Caveat number one! There are several versions of the MSI and library files; I discovered by trail and error that some did what I wanted and others did not work at all. I have no idea why this is the case. I have used the following software and Phidget cards:

- Phidget 2.1.4 MSI (20080924) library
- FS2Phidgets Version 4.3.8
- Phidget Interface kit 004 card (4 relays)
- Phidget 1064 motor servo controller card
- Two Phidget 1066 cards

Some Important Points

- Never connect Phidgets directly to your computer. Always use one or two quality powered USB hub (s). I'm not sure why, but this was advised by several other people
- Always open FSX FIRST, then open Phidgets
- FS2Phidgets must be installed on the same computer that the phidget cards are connected
- Use the default Windows 7 installation folder (X86)
- Install the Phidgets library ***BEFORE*** running the Phidgets MSI
- Ensure you create a desktop shortcut to enabler easy activating of FS2Phiodgets once FSX is open
- Always back-up the FS Variables .ini file. This is the file that contains the information relating to your Phidget set-up

Once the Phidgets library and MSI are installed on your computer, you will be able to open the FS2Phidgets software interface. This is where you begin to program the Phidget cards that you have installed to the various throttle outputs.

Phidgets Set-Up

When the software is open on your desktop you will note a section which has the Phidget serial, FS variable and function name. The serial is the actual Phidget card installed and the FS variable is the

flight simulator variable that you want the card to be assigned to. Looking at the interface, you will note the word INDEX. An INDEX refers to a particular part of a Phidget card – such as the 004 interface card which has 4 relays. Each relay number is referred to as an INDEX.

Let's look at the Phidget serials I have set up for the operation of the throttle quadrant. In the serial table below, you will note the FS variables I have used for the speed brake, elevator trim control and movement of the trim wheels.

Phidget Serial Table

PHIDGET SERIAL	FUNCTION NAME	PHI VALUE	FS VARIABLE ASSIGNMENT	OFFSET	NOTES
119892	Advanced servo	105	Spoiler handle	0BD0	
119461	Trim indicator (trim indicator)	105	Elevator trim indicator	0BC2 / 0BC2	
147268	Motor controller (trim wheels)	105	Elevator trim control (index 0)	0BC0 / 0BC0	INDEX 1 leave unassigned
179566	Interface kit 004 (4 relays)	105	Squat switch	0366	INDEX (relay number 0)
179566	Interface kit 004 (4 relays)	105	Squat switch	0366	INDEX (relay number 1)
179566	Interface kit 004 (4 relays)		Not assigned	Not assigned	INDEX (relay number 2) Nothing connected to relay
179566	Interface kit 004 (4 relays)		Not assigned	Not assigned	INDEX (relay number 3) Nothing connected to relay

Calibrating the Hardware Using Phidgets & FSUPIC

We will now look at some of the throttle functions and learn how to calibrate them correctly.

Trim Wheels

The trim wheels spin when you depress the electric trim tabs on the yoke. The power to move the trim wheels comes from a 12V DC servo motor installed into the quadrant bay beside the wheels.

The motor is connected to a Phidget 1064 motor servo controller card.

Within the FS2Phidgets software interface you will notice, when you highlight serial 147268 (motor controller), a button that allows you calibrate the trim wheels. In this calibration box you can calibrate the speed and acceleration of the trim wheels.

Trim Indicators (trim tab on throttle beside trim wheels)

The trim indicators will most likely also require calibration within the FS2Phidgets software. The trim tab indicators must mimic the correct trim setting for the aircraft. First, open the virtual throttle of the default B737 aircraft in FSX. Select Phidget serial 119461 and click the calibration button. Whilst watching the trim tabs on the actual throttle, alter the various variables until you are satisfied the trim tabs move to the correct locations. The trim tabs must mimic the positions on the virtual throttle. My settings are as follows:

- FS Value 4744
- Target servo position 116
- Slew rate 30
- Range of travel MIN=0 / MAX = 180

Speed Brake

The speed brake is a little more tricky and in-depth. We will look at this in a step process. Some of the information below may not relate to your set-up as it depends on how you have wired your throttle; however, for completeness I have provided the details.

Mimicking the Movement of the Speed Brake

Step 1: You have to ensure that the speed brake movement (arm and detent) is identical to the actual movement of the speed brake in FSX. To check this, open FSX and the default B737. Open the throttle quadrant so you can see the virtual throttle and speed brake. Move the real speed brake lever checking that the movement is identical to that of the virtual speed brake.

If Northern Simulations did your wiring, then the lever movement should be exactly replicated. If the lever position does not mimic the virtual lever, then you will have to open up FSUPIC axis assignments and calibrate the actual position of the speed brake lever.

This is how you do this. Open FSUPIC and select the Axis Assignment tab. Move the speed brake lever checking the position of the lever. When completed, select send to FSUPIC. Now, open the call out box and highlight/select/check spoilers.

Assign FS2Phidgets

Step 2: Open up the FS2Phidgets software and double check that you have Phidgets Serial 119892 (advanced servo) configured to the appropriate FS variable and offset. In this case the FS variable is spoiler handle and offset is OBDO (see serial table above).

At this stage, the speed brake should operate correctly and mimic the virtual speed brake in FSX.

However, there is no motorization yet associated with the speed brake; the lever does not automatically move into detent on landing or close/lock when the reversers are closed. In a real B737, the speed brake operates via the use of an electrical squat switch, which is an on/off button triggered by the pressure of the landing gear on touch down (very basic explanation).

Step 3: To simulate the squat switch, we need to install a Phidget 004 interface card and configure the FS variables of this card to replicate the functionality of a squat switch. This card has four relays attached to it; however, we only need to use two of the relays for the motorization of the speed brake.

Speed Brake Motorisation – Adding Motor & Phidget 004 Interface Card

Step 4: To allow motorisation of the speed brake lever, we need to attach an appropriate motor to the handle of the linkage that moves the speed brake. There are several motors you can use and I have used HiTec HS 815BB servo motor (often used in model sail boats and motor boats). The motor is attached to the quadrant beside the speed brake lever. If you have used Northern Simulations to convert your throttle, the installation and wiring of this motor will already have been done for you.

Step 5: Next, we must wire the speed brake motor and the phidget 004 card to the correct output in the Leo Bodnar joystick controller card.

To determine which output wire on the card is operating the speed brake, run FSX and open the default B737 aircraft in spot mode, so you can visually see the spoilers deploying. Connect and reconnect the wires from the Leo Bodnar card until you discover the correct output for the speed brake.

Step 6: Next, you must connect the wire from this output (on the Leo Bodnar card) to one of the relays on the Phidget 004 card. Each relay has two variables – open common (OC) and normally open (NO). It is very important, to ensure correct operation, that you select the correct variable when inserting the wire into the relay.

Failure to do so may cause power to continually be directed towards the servo motor causing overheating and failure. The servo motor should only receive power when the squat switch is activated on landing, opening the circuit, to move the speed brake handle into the fully deflected position (handle towards you).

The phidget 004 card needs to be connected in-line between the servo motor and the Leo Bodnar card. The 004 card has 4 relays; however, two are only needed to replicate the squat switch (on/off). Each relay has two variables – open common (OC) and normally open (NO). It is very important, to ensure correct operation, that you select the correct variable when inserting the wire into the relay.

Table showing Phidget relay codes & wire placement for speed brake motorization

CIRCUIT RELAY NUMBER	PHIDGET RELAY CODE	NOTES
RELAY ZERO	OC (open common)	Wire from phidget high current motor card (controls speed brake)
RELAY ZERO	NO (normally open)	Wire from phidget high current motor card (controls speed brake)
RELAY ONE	1C (open common)	Wire from speed brake terminal of Leo Bodnar card
RELAY ONE	NC (normally open)	Wire from speed brake terminal of Leo Bodnar card
RELAY TWO	2C (open common)	Wire from speed brake terminal of Leo Bodnar card (duplicate relay one settings)
RELAY TWO	NC (normally open)	Wire from speed brake terminal of Leo Bodnar card (duplicate relay one settings)

Using FSUPIC

All other throttle functions have been programmed and calibrated using a registered version of FSUPIC. This is relatively straightforward; however, some FSUPIC offset values may not operate as intended. There are several reasons for this, one which maybe the actual version of FSUPIC you are using. It's a matter of starting with the recommended offsets, which should work, and moving forward from this point.

Below is a table outlining the utilised function, offset and parameters. The FSUPIC button allocation may change depending upon your set-up.

Table showing FSUPIC commands and offsets

FUNCTION	FS COMMAND OR FSUPIC KEYSTROKE	FSUPIC	FSUPIC PARAMETER / OFFSET	NOTES
Reversers engine 1	Throttle 1 set	Button 0	-16384 / 0	✓ control to repeat
Reversers engine 2	Throttle 2 set	Button 1	-16384 / 0	✓ control to repeat
Fuel levers 1	Offset byte set	Button 7	X3590 / X01 X3590 / X00	✓ control to repeat
Fuel levers 2	Offset byte set	Button 8	X3594 / X01 X3594 / X00	✓ control to repeat
Parking brakes	Parking brakes	Button 5	FSX	☒ control to repeat
TOGA	Offset word set	Button 2	Offset = X53DB Value = X0237	✓ control to repeat Fill in top section only
A/T disconnect	Offset word set	Button 3	Offset = X53DB Value = X0233	☒ control to repeat Fill in top section only
Auto pilot yoke button		Button ?	Offset = X53DB Value = X0228	☒ control to repeat Fill in top section only
Stab Trim switch				Hard-wired to stop servo engine (trim wheels)

✓ Always select for FS control within FSUPIC.

You will note that some functions, namely: engine 1/2 throttle advancement, the stab trim switch and flaps, are not recorded in the table. This is because Northern Simulation wired these functions during the revamp of my throttle quadrant in such a way that FSX automatically sees them. In the case of the stab trim switch, it has been configured via a button to switch off the spinning trim wheels, but not the trim indicator tabs. The flaps, I have calibrated separately in Sim Avionics.

FSUPIC Profile

One aspect of the set up I have not mentioned is FSUPIC profiles. A profile is an excellent way to keep all your button and key assignments associated with the one aircraft type. You create a profile within FSUPIC, name it, and then select the profile whenever you are calibrating or programming. This keeps the FSUPIC file neat and tidy.

Further Information

FS2Phidgets website - http://www.phidgets.com/

FS2Phidgets forum - http://www.phidgets.com/phorum/

Conclusion

There you have it.

I'm well aware that this is a very convoluted and complicated subject. Equally, it's rather difficult to write in such a way that the information is easy to understand.

One aspect which is important to realize is that, although the above has worked for me, it maybe different in your set-up. Each computer is different and there are several methods to convert throttles to FS use. Please use the information as a guide only.

If there is only word of advice I can give you it is: " try to keep your set-up simple"

The image below shows the connection of the Interface 004 card. Replace the yellow LED with the Leo Bodnard card and the battery with the throttle unit