Joystick Controller Interface to Aerowinx PSX

In Aerowinx PSX the systems of the real Boeing 747-400 aircraft are modeled. Hence, the toe brakes are considered linked as are the flight controls and tillers. In simulators, unless real components are used, the toe brakes are separate as are the tillers. And in some cases the elevator, aileron and rudder controls are also separate. What is needed is an interface program that will read the inputs from a joystick controller (controllers), combine the inputs from the Captain and First Officer side and deliver an integrated input to PSX for use by the flight model. If the simulator has the controls linked the interface program needs to null out the separate set of inputs and provide the single, linked input value to PSX.

DONE Variable name: Tiller Qh426
Category: Qh (32 bit integer, human controlled mechanical component)
Range: -999 ... 999
Read and write

Suggest an internal variable of TillerCp and TillerFo to read the values from the captain and first officer side, then combine the values and adjust for a minimum value of -999 and a maximum value of 999. In cases where an axis(s) is reversed invert the values according so PSX correctly interprets left and right direction from the minus and plus value ranges provided to PSX.

Flight controls

DONE Variable name: FltControls Qs120
Category: Qs
Min length: 5 bytes
Max length: 14 bytes
Read and write

The string is separated by two semicolons. There must be no semicolon at the end. The values are yoke and pedal positions set by the pilot. Each value is an integer ranging from -999 to 999. The sequence is elevator, aileron, and rudder. The values can be (provisionally) used to animate the control surfaces in the outside view. As they represent only the pilot action on the yoke and pedals, these data outputs do not include system failures, control ratio, aileron lockout etc.

As for the tiller above, suggest an internal variable for elevatorCp, elevatorFo, aileronCp, aileronFo, RudderCp and rudderFo. Then combine and adjust to the min and max values provided above.

Toe brake pedals

Variable name: Brakes Qs357
Category: Qs

Min length: 3 bytes
Max length: 9 bytes
Read and write

There's a single semicolon in the middle, separating the left and right toe brake pedal angles. The values are scaled so that the full range is 0 to 1000.

Again, assign an internal variable for toeBrakeLeftCp, toeBrakeRightCp, toeBrakeLeftFo, and ToeBrakeRightFo. Then combine the values and scale the integrated value between 0 and 1000 to the Brakes variable and write to PSX.

In addition to the analog axis of the joystick controllers there are buttons that need to be programmed as well. They are:

"Qh398=""StabTrimCp""; Mode=ECON; Min=-1; Max=1;"
"Qh399=""StabTrimFo""; Mode=ECON; Min=-1; Max=1;"
"Qh400=""ApDisc""; Mode=DELTA; Min=0; Max=1;"
"Qh82=""LcpPttCp""; Mode=DELTA; Min=0; Max=1;"
"Qh93=""LcpPttFo""; Mode=DELTA; Min=0; Max=1;"

For Qs variables (strings), the min-max values refer to the allowed string length. If min and max is equally 13, for example, the string must be exactly 13 characters long.

WXR - weather radar panel pushbuttons

Code: Qs104
Name: WxRdrPanelPush
Mode: ECON
Length: 13 .. 13
Add-ons can: Read & write
Example:
FWTmGareFWTmG
Description:

Each character in the string from left to right indicates the "pushed"-status of a certain button. Lower case means "pushed". The order from left to right:

1 - TFR captain
2 - WX captain
3 - WX+T captain
4 - MAP captain
5 - GC captain
6 - AUTO
7 - L/R
8 - TEST
9 - TFR F/O

10 - WX F/O
11 - WX+T F/O
12 - MAP F/O
13 - GC F/O

WXR - weather radar panel rotary selectors

Code: Qs105
Name: WxRdrPanelTurn
Mode: ECON
Length: 7 .. 15
Add-ons can: Read & write
Example:
4713;3;2197;0
Description:

The string stores the position values of the two gain rotary selectors and the two tilt potentiometers. All values are separated by semicolons. There must be no semicolon at the end. All values are integer values. The string sequence from left to right:

1 - Tilt captain (range 0 .. 4713)
2 - Gain captain (range -6 .. 3)
3 - Tilt F/O (range 0 .. 4713)
4 - Gain F/O (range -6 .. 3)

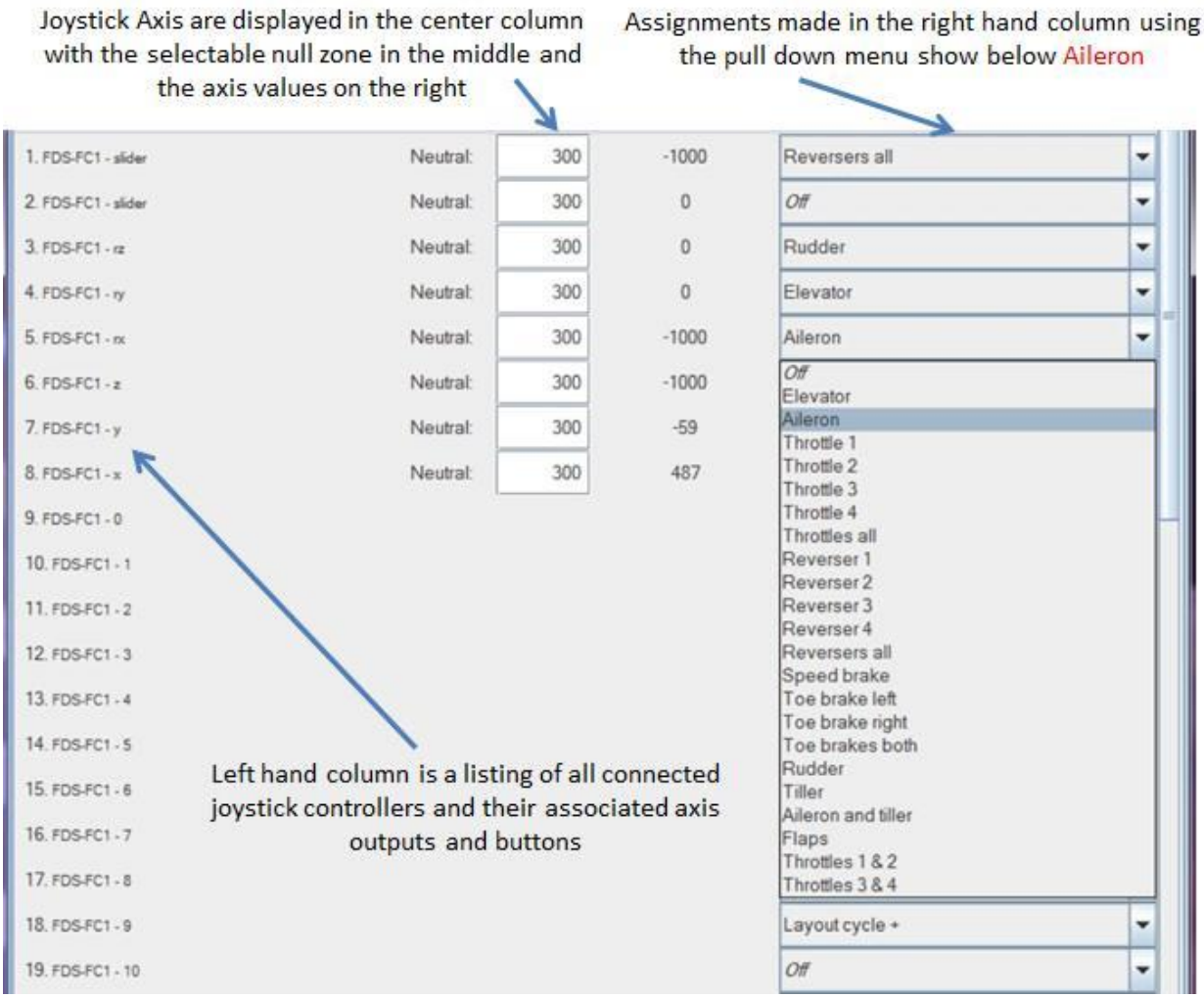
For our configuration, the rotary selector is also a potentiometer. So we need to take the full range of the axis and divide into zones, then assign a value for each zone for the gain (-6 to 3). Therefore the read from the controller will be two potentiometers but the write to PSX will be the one axis and the one discrete value selector switch per the above Qs105 input.

Finally, we have some ancillary analog inputs for display intensity control, fuel dumping and pressure settings for the airplane.

Qh273 JettRemain Mode=ECON Min=0 Max=62830
Qh297 LdgAltTurn Mode=ECON Min=0 Max=62830
Qh87="LcpOutbdCp"; Mode=ECON; Min=0; Max=4713;
Qh89="LcpNdCp"; Mode=ECON; Min=0; Max=4713;
Qh98="LcpOutbdFo"; Mode=ECON; Min=0; Max=4713;
Qh100="LcpNdFo"; Mode=ECON; Min=0; Max=4713;
Qh101="LcpMapFo"; Mode=ECON; Min=0; Max=4713;
Qh139="EicasBrtUpr"; Mode=ECON; Min=0; Max=4713;
Qh140="EicasBrtLwrInner"; Mode=ECON; Min=0; Max=4713;
Qh141="EicasBrtLwrOuter"; Mode=ECON; Min=0; Max=4713;

Provisioning Input Screen

A simple web interface should be developed to facilitate assignment of the various joystick axis and buttons that are assigned. The following is a sample copied from the main Aerowinx program:



Most of the assignments will be axis oriented. However, there are several buttons on the control yokes for trim, AP disconnect and push-to-talk functions. A simple Engish word or words should be used in the pull down menu and a translation table built into the program to convert to the variable names used in the program. Also, a right justified "X" should be displayed for the joystick buttons when a button is pushed for the center column.

Finally, since this program will be shared as freeware with sim users of PSX who are not computer literate the program should be launched using an icon that can be placed in the start menu of the Windows PC.