



Opencockpits



Manual Conexiones Pedestal B737.

Índice:

MANUAL CONEXIONES PEDESTAL B737.....	1
ÍNDICE:.....	2
INTRODUCCIÓN:.....	3
DESCRIPCIÓN DEL PEDESTAL:.....	3
VERSIÓN COMPLETA.....	3
VERSIÓN LIGHT.....	4
VERSIÓN BASE.....	4
CONEXIÓN DEL PEDESTAL AL ORDENADOR:.....	4
CONFIGURACIÓN DE LAS IOCARDS:.....	5
MANUAL:.....	5
AUTOMÁTICO:.....	5
COMPROBACIÓN DE ENTRADAS Y SALIDAS:.....	7
CONEXIÓN DE LA USB DIMMCONTROL (OPCIONAL):.....	7
LINKS DE INTERÉS:.....	9
ANEXO 1. DEFINICIÓN DE VARIABLES:.....	10
ANEXO 2. SCRIPT ESPECÍFICO PARA FSX Y PROJECT MAGENTA:.....	17

Introducción:

El pedestal B737 es un módulo diseñado para poder manejar los múltiples parámetros de radio, extinción de incendios, control de audio, trimado de timón y radar. Para ello incluye toda la electrónica necesaria. Todo ello conectado al ordenador por 2 cables USB solamente (opcionalmente 3 cables USB).

Además está diseñado para que el usuario sólo tenga que enchufarlo a corriente y al ordenador.

La electrónica incluida es la siguiente:

- Fuente de alimentación con salidas a 3,3V, 5V y 12V.
- 1 USB Expansion.
- 2 Master.
- 6 Displays.
- 1 USB Servos.
- 1 USB DimmControl.
- 2 PCB Pedestal.

En este manual vamos a explicar la puesta en marcha del pedestal y su configuración a nivel de hardware y software para el FSX sin entrar en los múltiples add-ons existentes. Al final del manual está disponible el listado de entradas y salidas disponibles del pedestal para que pueda usarse con cualquier add-on compatible con los sistemas Opencockpits y un script específico para FSX.

Descripción del pedestal:

El pedestal está construido sobre una sólida caja de aluminio que contiene toda la electrónica. La electrónica está integrada y ordenada en una estructura de material plástico sólidamente atornillada a la caja de aluminio y está organizada por capas para una fácil localización de posibles averías y más rápida reparación.

El pedestal se encuentra disponible en tres versiones, completamente montado, Light y Base.

Versión Completa.

La versión completa tiene todos los paneles y funciones operativas:

- 2 COMM.
- 2 NAV.
- 2 ADF.
- 2 AUDIO.
- 1 FIRE ENGINES.
- 1 FIRE CARGO.
- 1 RADAR
- 1 ATC.
- 1 RUDDER TRIM.
- 1 PANEL ILUMINACIÓN.
- 1 PANEL STAB TRIM.

Todos los paneles tienen sus funciones operativas (botones, encoders, rotativos, leds, indicadores, dígitos, servo del rudder, retroiluminación...). Algunas entradas de algunos rotativos (botones de los filtros de los Audios y botón del modo del Rádar) tienen sus posiciones reales pero no tienen conexión con la electrónica pero pueden programarse por software.

Versión Light.

La versión Light tiene los siguientes paneles y funciones operativas:

- 1 COMM.
- 2 NAV.
- 1 ADF.
- 1 AUDIO.
- 1 FIRE CARGO.
- 1 RADAR
- 1 ATC.
- 1 RUDDER TRIM.
- 1 PANEL ILUMINACIÓN.
- 1 PANEL STAB TRIM.
- 5 PANELES DE CIERRE.

Este módulo puede completarse fácilmente con los módulos terminados IDC disponibles.

Versión Base.

La versión Base consta de la caja de aluminio y de toda la electrónica necesaria para manejar todos los paneles y funciones pero no incluye ningún panel, módulo, tapadera o cable IDC.

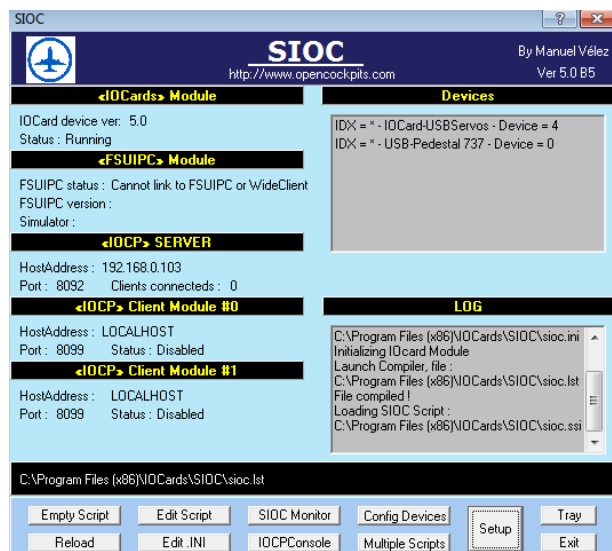
Esta versión es ideal para los usuarios que ya tienen los paneles IDC o los módulos USB, que se pueden convertir fácilmente en IDC retirándoles la PCB trasera (excepto el ATC USB que no se puede convertir a IDC).

Conexión del pedestal al ordenador:

El pedestal se envía totalmente montado listo para ser alimentado y conectado al ordenador. El paquete incluye dos cables USB enchufados en sendas tarjetas USB y un cable de alimentación conectado a la fuente.

Las dos tarjetas que se conectan por defecto al ordenador son una USB Expansion y una USB Servos. Existe una tercera tarjeta USB, la DimmControl que maneja los canales de retroiluminación de los paneles, que no está conectada al pedestal porque viene programada de fábrica y no es necesario su reprogramación aunque que podría conectarse y ser reprogramada según criterio del usuario. La fuente de alimentación tiene un rango de 120V-240V y 50-60Hz con una potencia de 500W (según modelos puede tener o no interruptor de encendido).

Una vez alimentado y conectado el pedestal al ordenador lanzamos el software SIOC y debe aparecer lo siguiente:



Configuración de las IOCARDS:

La tarjeta USB Expansion que se utiliza para el pedestal es exactamente igual a las demás pero tiene un nombre diferente en el firmware, USB-Pedestal 737, para ayudar a diferenciarla de otras conectadas al mismo ordenador y para un futuro software de autoconfiguración. También tiene una declaración en sioc.ini diferente:

MASTER= XX,20,YY,ZZ

Donde:

XX, es el número de idx de la tarjeta que se asocia al número de DEVICE dentro de un script.

20, es la forma de decirle a Sioc que la tarjeta es una USB-Pedestal 737.

YY, es el número que indica la cantidad de Masters que tiene conectada a sus puertos (de 1 hasta 4 y en el caso del pedestal 2).

ZZ, es el número del puerto USB donde está conectada.

Ejemplo de declaración de un pedestal:

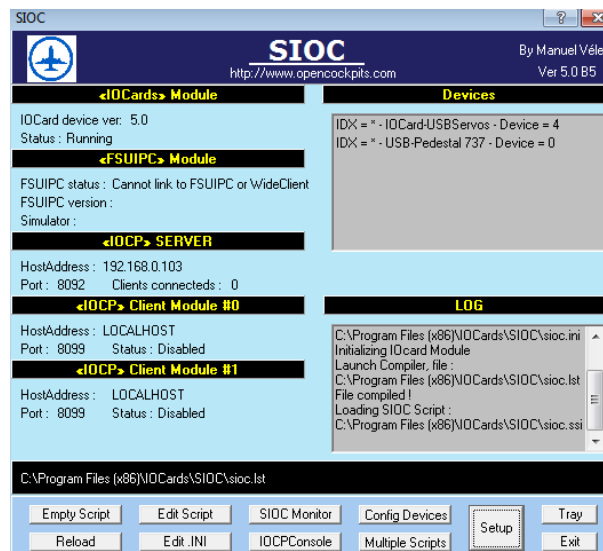
MASTER= 40,20,2,0 (como en nuestro ordenador de pruebas).

Para la configuración de las tarjetas tenemos dos modos:

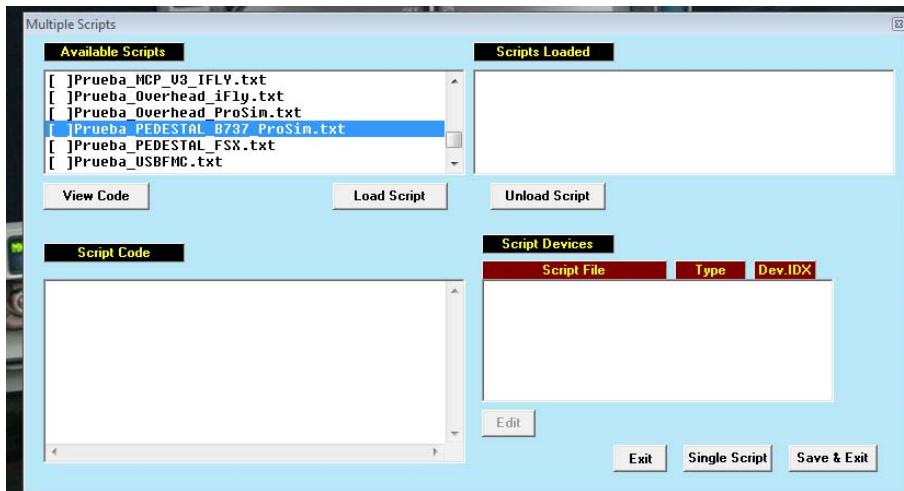
Manual: editando el archivo sioc.ini, que es un poco laborioso y precisa de un conocimiento medio-avanzado de Sioc. Este modo puede verse en los manuales de las tarjetas USB Expansion+Master y otros, disponibles en la guía de ayuda de construcción de cabinas de la web de Opencockpits.

Automático: Este modo se puede utilizar desde la versión 5.0b3 en adelante y consiste en seleccionar los scripts y las IOCards con el ratón en un menú de Sioc y unirlos de manera automática. Veamos pues el modo automatico.

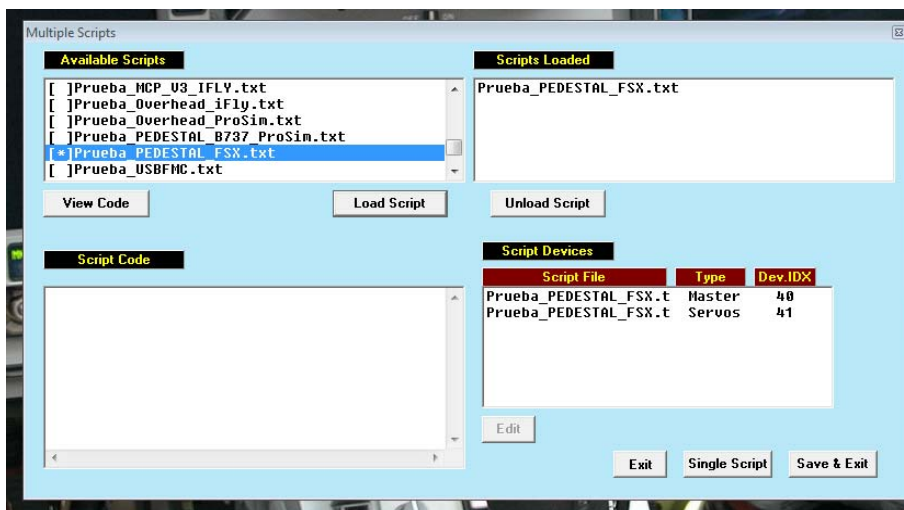
1. Descargar e instalar la última versión disponible de Sioc desde la página de ayuda de la guía de construcción de la web de Opencockpits.
2. Descargar el script del módulo pedestal para nuestro simulador y add-on (si se tiene, ver la disponibilidad de los scripts en la pestaña de descargas en la página del producto). En este caso vamos a utilizar FSX sin add-ons pero con FSUIPC instalado, el script está al final de este manual como anexo y disponible en la web de Opencockpits.
3. Lanzar Sioc y seleccionar el botón "Abrir la carpeta de Scripts". Copiar el archivo descargado (.ssi o .txt) en dicha carpeta y cerrar la carpeta.
4. Conectar las tarjetas del pedestal al ordenador. Sólo es necesario conectar la USB Pedestal B737 y la USB Servos ya que la USB Dimmcontrol viene programada de fábrica aunque podría conectarse también. Las IOCards aparecerán reconocidas pero no configuradas:



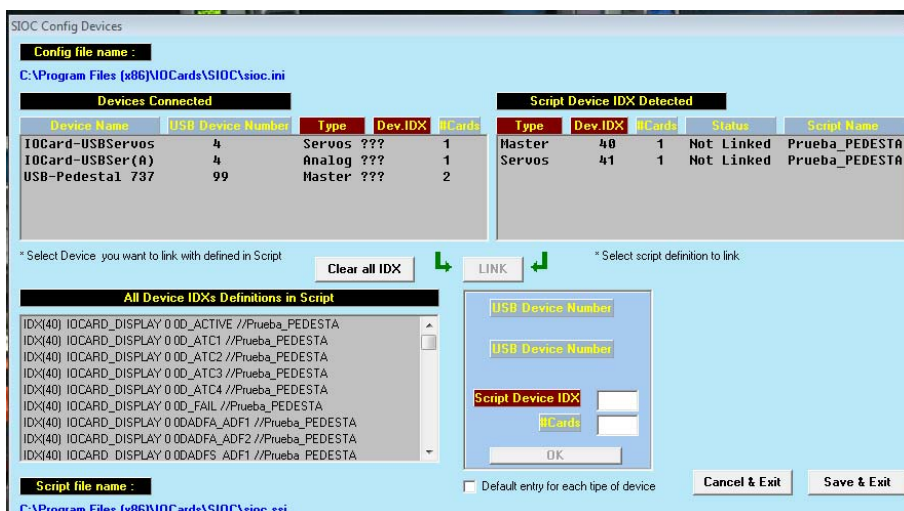
- Pulsar el botón "Multiple Scripts" y buscar el script prueba_pedestal_fsx.txt en la ventana de la izquierda.



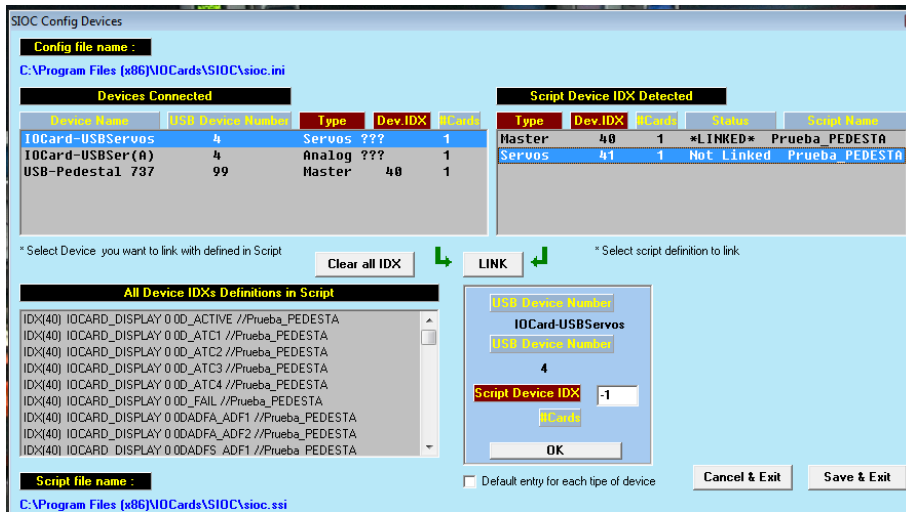
- Pulsar el botón "Load Script" y automáticamente aparecerá en la ventana de la derecha como cargado y en la ventana inferior derecha aparecerán descritos los devices que controla dicho script, la USB Pedestal B737 aparece como Master con device-IDX 40 y la USB Servos aparece como Servos con device-IDX 41.



- Salvamos y salimos. Sólo nos falta asignar los dispositivos al script. Pulsando en "Config Devices" se abre la siguiente ventana:



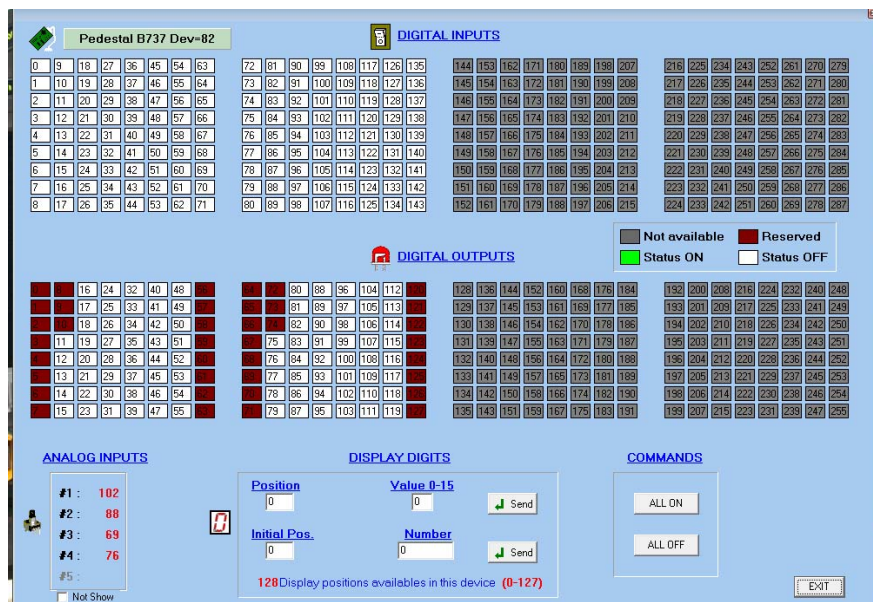
- En la ventana izquierda aparecen los dispositivos conectados al ordenador y en la derecha aparecen los dispositivos que necesita el script.
- Seleccionamos en la ventana izquierda el dispositivo USB-Pedestal B737 y en la derecha el dispositivo Master que le corresponde y los unimos con el botón "Link". Con la USBServos procedemos de la misma manera:



- Salvamos y salimos y ya tenemos configurado nuestro pedestal y Sioic para su uso con FSX.

Comprobación de entradas y salidas:

Una vez que ya tenemos las tarjetas configuradas podemos comprobar todos los elementos del pedestal como interruptores, encoders, indicadores y dígitos. Para ello usaremos el Sioic Monitor.



Conexión de la USB DimmControl (opcional):

La retroiluminación del pedestal se divide en 3 canales diferentes debido a que existen 2 tipos de retroiluminación y consumos diferentes en los paneles: retroiluminación clásica por leds estándar y retroiluminación BKI.

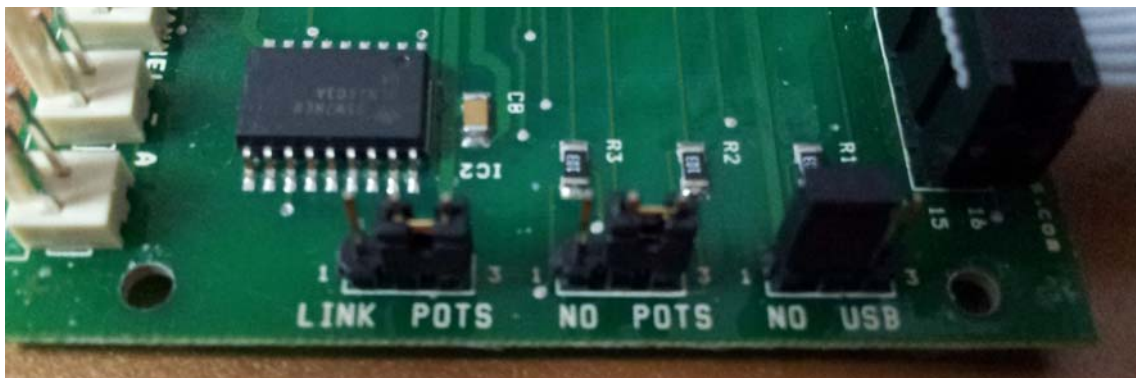
La tecnología BKI es similar a la original empleada en los paneles originales de Boeing, se trata de incrustar la retroiluminación dentro de los paneles, con un aumento de la calidad de la retroiluminación y una bajada importante del consumo de energía así como para evitar contaminación lumínica alrededor de los paneles.

La tarjeta USB DimmControl gestiona 3 canales independientes:

- Paneles leds estándar canal 1 (A).
- Módulo de control fuego motores canal 2 (B).
- Paneles BKI canal 3 (C).

La tarjeta USB DimmControl viene programada de fábrica para igualar la intensidad lumínica de todos los paneles en modo NO USB (sin conectar por USB) mediante un potenciómetro conectado a ella y al panel de iluminación, pero para poder cambiar los límites de cada canal debe conectarse por USB al ordenador cambiando temporalmente el modo del puente NO USB de activado (1) a desactivado (3):

Estado original: modo NO USB.

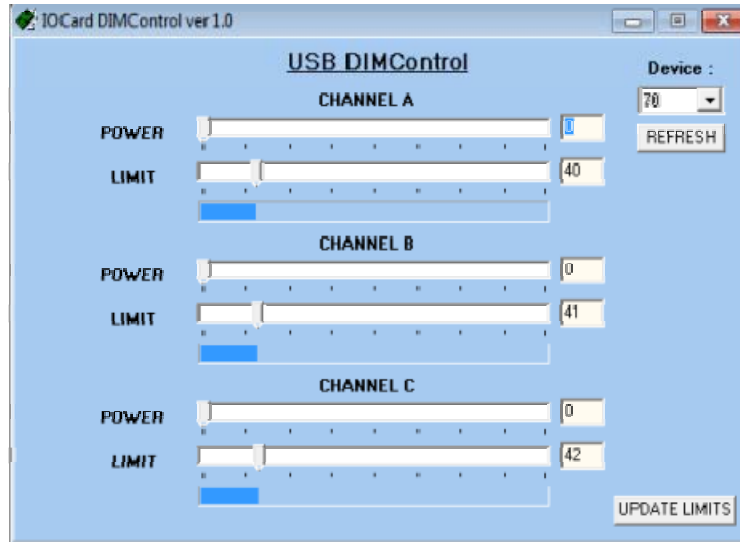


Estado para programar: modo USB.



En este modo se conecta el cable USB a la tarjeta y al ordenador y se puede utilizar el programa de configuración de la USB DimmControl descargable desde la web de Opencockpits o directamente con un script de Sioc.

Cuando la tarjeta se conecta al ordenador es reconocida inmediatamente por el software de control iocard_dimcontrol.exe que lee los valores por defecto grabados en la eeprom:



Una vez cargados los valores por defecto, podemos cambiar el valor de salida de cada canal de forma independiente usando las barras deslizantes. También podemos cambiar los límites máximos de entrega de potencia de cada canal y memorizarlos en la eprom si pulsamos el botón UPDATE LIMITS.

El rango es de 256 pasos. Dichos pasos se tienen de cero a la potencia máxima de la tarjeta, es decir: si la potencia máxima del canal A está en 40, el valor de la iluminación no pasará nunca de 40 aunque nos queden más pasos en el potenciómetro o en la barra del software iocar_dimcontrol.exe.

El establecimiento de los límites es muy importante y sólo puede hacerse desde el modo USB ya sea independiente o con Sioc (versión 4.7B1 o superior) y una vez terminada la configuración debe dejarse el puente NO USB en la posición activa (1).

Los límites de fábrica para el pedestal son:

CANAL	LÍMITE POTENCIA
A (1)	127
B (2)	210
C (3)	66

En la web de Opencockpits, en el apartado de descargas del Pedestal se irán actualizando los scripts para los add-ons disponibles.

Con esto damos fin a este manual, os invitamos a leer los manuales de los demás elementos de Opencockpits y del software SIOC y os damos las gracias por confiar en nosotros.

Links de interés:

Zona de soporte para clientes:

<http://www.opencockpits.com/catalog/info/>

Anexo 1. Definición de variables:

Aquí listamos las variables de las entradas y salidas disponibles del pedestal para poder programar un script en cualquier add-on compatible con los productos Opencockpits mediante el uso del software Sioc.

```
// DIGITS
//COMMS
Var 100, name CM1ACT, Link IOCARD_DISPLAY, DEVICE 40, Digit 0, Numbers 6
Var 102, name CM1STBY, Link IOCARD_DISPLAY, DEVICE 40, Digit 6, Numbers 6
Var 104, name CM2ACT, Link IOCARD_DISPLAY, DEVICE 40, Digit 64, Numbers 6
Var 106, name CM2STBY, Link IOCARD_DISPLAY, DEVICE 40, Digit 70, Numbers 6
// NAVS
Var 108, name NV1ACT, Link IOCARD_DISPLAY, DEVICE 40, Digit 16, Numbers 5
Var 110, name NV1STBY, Link IOCARD_DISPLAY, DEVICE 40, Digit 21, Numbers 5
Var 112, name NV2ACT, Link IOCARD_DISPLAY, DEVICE 40, Digit 80, Numbers 5
Var 114, name NV2STBY, Link IOCARD_DISPLAY, DEVICE 40, Digit 85, Numbers 5
// ADFS
Var 116, name ADF1ACT, Link IOCARD_DISPLAY, DEVICE 40, Digit 32, Numbers 5
Var 118, name ADF1STBY, Link IOCARD_DISPLAY, DEVICE 40, Digit 37, Numbers 5
Var 120, name ANT1ACT, Link IOCARD_DISPLAY, DEVICE 40, Digit 43, Numbers 1 // 0 =
ON, 1 = OFF
Var 122, name ANT1STBY, Link IOCARD_DISPLAY, DEVICE 40, Digit 44, Numbers 1 // 0 =
ON, 1 = OFF
Var 124, name ADF1A, Link IOCARD_DISPLAY, DEVICE 40, Digit 42, Numbers 1 // 0 = ON, 1
= OFF
Var 126, name ADF1S, Link IOCARD_DISPLAY, DEVICE 40, Digit 45, Numbers 1 // 0 = ON, 1
= OFF
Var 128, name ADF2ACT, Link IOCARD_DISPLAY, DEVICE 40, Digit 96, Numbers 5
Var 130, name ADF2STBY, Link IOCARD_DISPLAY, DEVICE 40, Digit 101, Numbers 5
Var 132, name ANT2ACT, Link IOCARD_DISPLAY, DEVICE 40, Digit 107, Numbers 1 // 0 =
ON, 1 = OFF
Var 134, name ANT2STBY, Link IOCARD_DISPLAY, DEVICE 40, Digit 108, Numbers 1 // 0 =
ON, 1 = OFF
Var 136, name ADF2A, Link IOCARD_DISPLAY, DEVICE 40, Digit 106, Numbers 1 // 0 = ON, 1
= OFF
Var 138, name ADF2S, Link IOCARD_DISPLAY, DEVICE 40, Digit 109, Numbers 1 // 0 = ON, 1
= OFF
// ATC
Var 140, name ATC, Link IOCARD_DISPLAY, DEVICE 40, Digit 90, Numbers 5

// OUTPUTS
// COMMS
Var 200, name COM1DOT, Link IOCARD_OUT, DEVICE 40, Output 11 // COMM 1 DECIMAL
DOT
Var 202, name COM2DOT, Link IOCARD_OUT, DEVICE 40, Output 75 // COMM 2 DECIMAL
DOT
// NAVS
Var 204, name NAV1DOT, Link IOCARD_OUT, DEVICE 40, Output 12 // NAV 1 DECIMAL DOT
Var 206, name NAV2DOT, Link IOCARD_OUT, DEVICE 40, Output 76 // NAV 2 DECIMAL DOT
// ADFS
Var 208, name ADF1DOT, Link IOCARD_OUT, DEVICE 40, Output 13 // ADF 1 DECIMAL DOT
Var 210, name ADF2DOT, Link IOCARD_OUT, DEVICE 40, Output 77 // ADF 2 DECIMAL DOT
// AUDIOS
Var 212, name AU1VH1MI, Link IOCARD_OUT, DEVICE 40, Output 14 // AUDIO1 VHF1 MIC
LED
Var 214, name AU1VH2MI, Link IOCARD_OUT, DEVICE 40, Output 15 // AUDIO1 VHF2 MIC
LED
```

Var 216, name AU1VH1VO, Link IOCARD_OUT, DEVICE 40, Output 16 // AUDIO1 VHF1 VOL LED
 Var 218, name AU1VH2VO, Link IOCARD_OUT, DEVICE 40, Output 17 // AUDIO1 VHF2 VOL LED
 Var 220, name AU1NA1VO, Link IOCARD_OUT, DEVICE 40, Output 18 // AUDIO1 NAV1 VOL LED
 Var 222, name AU1NA2VO, Link IOCARD_OUT, DEVICE 40, Output 19 // AUDIO1 NAV2 VOL LED
 Var 224, name AU1AD1VO, Link IOCARD_OUT, DEVICE 40, Output 20 // AUDIO1 ADF1 VOL LED
 Var 226, name AU1AD2VO, Link IOCARD_OUT, DEVICE 40, Output 21 // AUDIO1 ADF2 VOL LED
 Var 228, name AU1MKRVO, Link IOCARD_OUT, DEVICE 40, Output 22 // AUDIO1 MARKER VOL LED
 Var 230, name AU2VH1MI, Link IOCARD_OUT, DEVICE 40, Output 78 // AUDIO2 VHF1 MIC LED
 Var 232, name AU2VH2MI, Link IOCARD_OUT, DEVICE 40, Output 79 // AUDIO2 VHF2 MIC LED
 Var 234, name AU2VH1VO, Link IOCARD_OUT, DEVICE 40, Output 80 // AUDIO2 VHF1 VOL LED
 Var 236, name AU2VH2VO, Link IOCARD_OUT, DEVICE 40, Output 81 // AUDIO2 VHF2 VOL LED
 Var 238, name AU2NA1VO, Link IOCARD_OUT, DEVICE 40, Output 82 // AUDIO2 NAV1 VOL LED
 Var 240, name AU2NA2VO, Link IOCARD_OUT, DEVICE 40, Output 83 // AUDIO2 NAV2 VOL LED
 Var 242, name AU2AD1VO, Link IOCARD_OUT, DEVICE 40, Output 84 // AUDIO2 ADF1 VOL LED
 Var 244, name AU2AD2VO, Link IOCARD_OUT, DEVICE 40, Output 85 // AUDIO2 ADF2 VOL LED
 Var 246, name AU2MKRVO, Link IOCARD_OUT, DEVICE 40, Output 86 // AUDIO2 MARKER VOL LED
 // FIRE ENGINES
 Var 248, name ENG1_OVHL, Link IOCARD_OUT, DEVICE 40, Output 23 // ENGINE 1 OVERHEAT
 Var 250, name APUBOTDISL, Link IOCARD_OUT, DEVICE 40, Output 24 // APU BOTTLE DISCHARGE
 Var 252, name APUDETINOPL, Link IOCARD_OUT, DEVICE 40, Output 25 // APU DET INOPERATIVE
 Var 254, name FIREFAULTL, Link IOCARD_OUT, DEVICE 40, Output 26 // FIRE ENGINES FAULT
 Var 256, name FIREWHELLL, Link IOCARD_OUT, DEVICE 40, Output 27 // FIRE WHELL WELL
 Var 258, name ENG2_OVHL, Link IOCARD_OUT, DEVICE 40, Output 28 // ENGINE 2 OVERHEAT
 Var 260, name LBOTDISL, Link IOCARD_OUT, DEVICE 40, Output 29 // ENGINE 1 BOTTLE DISCHARGE
 Var 262, name RBOTDISL, Link IOCARD_OUT, DEVICE 40, Output 30 // ENGINE 2 BOTTLE DISCHARGE
 Var 264, name FIRE1L, Link IOCARD_OUT, DEVICE 40, Output 31 // ENGINE 1 FIRE HANDLE LIGHT
 Var 266, name FIREAL, Link IOCARD_OUT, DEVICE 40, Output 32 // APU FIRE HANDLE LIGHT
 Var 268, name FIRE2L, Link IOCARD_OUT, DEVICE 40, Output 33 // ENGINE 2 FIRE HANDLE LIGHT
 Var 270, name FIREG1L, Link IOCARD_OUT, DEVICE 40, Output 34 // FIRE ENGINE 1 GREEN LED
 Var 272, name FIREGAL, Link IOCARD_OUT, DEVICE 40, Output 35 // FIRE APU GREEN LED

Var 274, name FIREG2L, Link IOCARD_OUT, DEVICE 40, Output 36 // FIRE ENGINE 2
 GREEN LED
 // FIRE CARGO
 Var 276, name FCDISCL, Link IOCARD_OUT, DEVICE 40, Output 87 // FIRE CARGO
 DISCHARGE KORRY
 Var 278, name FCDISCBARL, Link IOCARD_OUT, DEVICE 40, Output 88 // FIRE CARGO
 DISCHARGE BAR KORRY
 Var 280, name FCAFTL, Link IOCARD_OUT, DEVICE 40, Output 89 // FIRE CARGO AFT
 KORRY
 Var 282, name FCAFTARML, Link IOCARD_OUT, DEVICE 40, Output 90 // FIRE CARGO AFT
 ARMED KORRY
 Var 284, name FCFWDL, Link IOCARD_OUT, DEVICE 40, Output 91 // FIRE CARGO FWD
 KORRY
 Var 286, name FCFWDARML, Link IOCARD_OUT, DEVICE 40, Output 92 // FIRE CARGO
 FWD ARMED KORRY
 Var 288, name FCDETFAULT, Link IOCARD_OUT, DEVICE 40, Output 93 // FIRE CARGO
 DETECTOR FAULT KORRY
 Var 290, name FCEXTFWDL, Link IOCARD_OUT, DEVICE 40, Output 94 // FIRE CARGO EXT
 FWD KORRY
 Var 292, name FCEXTAFTL, Link IOCARD_OUT, DEVICE 40, Output 95 // FIRE CARGO EXT
 AFT KORRY
 // STAB TRIM
 Var 294, name STBLCKL, Link IOCARD_OUT, DEVICE 40, Output 96 // STAB TRIM LOCK
 FAIL INDICATOR
 Var 296, name STBAUTOL, Link IOCARD_OUT, DEVICE 40, Output 97 // STAB TRIM AUTO-
 UNLOCK INDICATOR
 // ATC
 Var 300, name ATCFAULTL, Link IOCARD_OUT, DEVICE 40, Output 98 // ATC XP-FAULT
 INDICATOR
 Var 302, name ATCATCL, Link IOCARD_OUT, DEVICE 40, Output 99 // ATC DISPLAY "ATC"
 INDICATOR
 Var 304, name ATCATC1L, Link IOCARD_OUT, DEVICE 40, Output 100 // ATC DISPLAY "1"
 INDICATOR
 Var 306, name ATCATC2L, Link IOCARD_OUT, DEVICE 40, Output 101 // ATC DISPLAY "2"
 INDICATOR

// ROTARY ENCODERS

// COMMS

Var 350, name E_CM1DEC, Link IOCARD_ENCODER, DEVICE 40, Input 0, Aceleration 1,
 Type 2 // COMM 1 ENCODER DECIMAL
 Var 352, name E_CM1ENT, Link IOCARD_ENCODER, DEVICE 40, Input 2, Aceleration 1,
 Type 2 // COMM 1 ENCODER ENTEROS/INTEGER
 Var 354, name E_CM2DEC, Link IOCARD_ENCODER, DEVICE 40, Input 72, Aceleration 1,
 Type 2 // COMM 2 ENCODER DECIMAL
 Var 356, name E_CM2ENT, Link IOCARD_ENCODER, DEVICE 40, Input 74, Aceleration 1,
 Type 2 // COMM 2 ENCODER ENTEROS/INTEGER
 // NAVS
 Var 358, name E_NAV1DEC, Link IOCARD_ENCODER, DEVICE 40, Input 9, Aceleration 1,
 Type 2 // NAV 1 ENCODER DECIMAL
 Var 360, name E_NAV1ENT, Link IOCARD_ENCODER, DEVICE 40, Input 11, Aceleration 1,
 Type 2 // NAV 1 ENCODER ENTEROS/INTEGER
 Var 362, name E_NAV2DEC, Link IOCARD_ENCODER, DEVICE 40, Input 81, Aceleration 1,
 Type 2 // NAV 2 ENCODER DECIMAL
 Var 364, name E_NAV2ENT, Link IOCARD_ENCODER, DEVICE 40, Input 83, Aceleration 1,
 Type 2 // NAV 2 ENCODER ENTEROS/INTEGER
 // ADFS

```

Var 366, name E_ADF1DEC, Link IOCARD_ENCODER, DEVICE 40, Input 18, Aceleration 1,
Type 2 // ADF 1 ENCODER DECIMAL
Var 368, name E_ADF1ENT, Link IOCARD_ENCODER, DEVICE 40, Input 20, Aceleration 1,
Type 2 // ADF 1 ENCODER ENTEROS/INTEGER
Var 370, name E_ADF2DEC, Link IOCARD_ENCODER, DEVICE 40, Input 90, Aceleration 1,
Type 2 // ADF 1 ENCODER DECIMAL
Var 372, name E_ADF2ENT, Link IOCARD_ENCODER, DEVICE 40, Input 92, Aceleration 1,
Type 2 // ADF 1 ENCODER ENTEROS/INTEGER
// ATC
Var 374, name E_ATCLEFT, Link IOCARD_ENCODER, DEVICE 40, Input 113, Aceleration 1,
Type 2 // DIGIT SELECTION ENCODER
Var 376, name E_ATCRIGHT, Link IOCARD_ENCODER, DEVICE 40, Input 115, Aceleration 1,
Type 2 // DIGIT INCR/DECR ENCODER
// RUDDER TRIM
Var 378, name E_RUDDER, Link IOCARD_ENCODER, DEVICE 40, Input 67, Aceleration 1,
Type 2 // RUDDER TRIM ENCODER

// SWITCHES
// COMMS
Var 400, name S_CM1TST, Link IOCARD_SW, DEVICE 40, Input 4
Var 402, name S_CM2TST, Link IOCARD_SW, DEVICE 40, Input 76
Var 404, name S_CM1SWP, Link IOCARD_SW, DEVICE 40, Input 5
Var 406, name S_CM2SWP, Link IOCARD_SW, DEVICE 40, Input 77
// NAVS
Var 408, name S_NAV1TST, Link IOCARD_SW, DEVICE 40, Input 13
Var 410, name S_NAV2TST, Link IOCARD_SW, DEVICE 40, Input 85
Var 412, name S_NAV1SWP, Link IOCARD_SW, DEVICE 40, Input 14
Var 414, name S_NAV2SWP, Link IOCARD_SW, DEVICE 40, Input 86
// ADFS
Var 416, name S_ADF1SWP, Link IOCARD_SW, DEVICE 40, Input 23
Var 418, name S_ADF2SWP, Link IOCARD_SW, DEVICE 40, Input 95
Var 420, name S_ADF1ANT, Link IOCARD_SW, DEVICE 40, Input 22
Var 422, name S_ADF2ANT, Link IOCARD_SW, DEVICE 40, Input 94
Var 424, name S_ADF1TONE, Link IOCARD_SW, DEVICE 40, Input 24
Var 426, name S_ADF2TONE, Link IOCARD_SW, DEVICE 40, Input 96
// ATC
Var 428, name S_ATCIDENT, Link IOCARD_SW, DEVICE 40, Input 78
Var 430, name S_ATCXPN2, Link IOCARD_SW, DEVICE 40, Input 79
Var 432, name S_ATCALTS2, Link IOCARD_SW, DEVICE 40, Input 80
Var 434, name R_ATCSTB, Link IOCARD_SW, DEVICE 40, Input 139 // ROTARY SWITCH
Var 436, name R_ATCALT, Link IOCARD_SW, DEVICE 40, Input 140 // ROTARY SWITCH
Var 438, name R_ATCXP, Link IOCARD_SW, DEVICE 40, Input 141 // ROTARY SWITCH
Var 440, name R_ATCTA, Link IOCARD_SW, DEVICE 40, Input 142 // ROTARY SWITCH
Var 442, name R_ATCTARA, Link IOCARD_SW, DEVICE 40, Input 143 // ROTARY SWITCH
// AUDIOS
Var 444, name S_AU1VH1MI, Link IOCARD_SW, DEVICE 40, Input 27 // AUDIO1 VHF1 MIC
SWITCH
Var 446, name S_AU1VH2MI, Link IOCARD_SW, DEVICE 40, Input 28 // AUDIO1 VHF2 MIC
SWITCH
Var 448, name S_AU1VH1VO, Link IOCARD_SW, DEVICE 40, Input 29 // AUDIO1 VHF1 VOL
SWITCH
Var 450, name S_AU1VH2VO, Link IOCARD_SW, DEVICE 40, Input 30 // AUDIO1 VHF2 VOL
SWITCH
Var 452, name S_AU1NA1VO, Link IOCARD_SW, DEVICE 40, Input 31 // AUDIO1 NAV1 VOL
SWITCH
Var 454, name S_AU1NA2VO, Link IOCARD_SW, DEVICE 40, Input 32 // AUDIO1 NAV2 VOL
SWITCH

```

Var 456, name S_AU1AD1VO, Link IOCARD_SW, DEVICE 40, Input 33 // AUDIO1 ADF1 VOL SWITCH

Var 458, name S_AU1AD2VO, Link IOCARD_SW, DEVICE 40, Input 34 // AUDIO1 ADF2 VOL SWITCH

Var 460, name S_AU1MKRVO, Link IOCARD_SW, DEVICE 40, Input 35 // AUDIO1 MARKER VOL SWITCH

Var 462, name S_AU1RT, Link IOCARD_SW, DEVICE 40, Input 38 // AUDIO1 R-T SWITCH

Var 464, name S_AU1IC, Link IOCARD_SW, DEVICE 40, Input 39 // AUDIO1 I-C SWITCH

Var 466, name R_AU1FILTV, Link IOCARD_SW, DEVICE 40, Input 37 // AUDIO1 FILTER V ROTARY SWITCH

Var 468, name R_AU1FILTR, Link IOCARD_SW, DEVICE 40, Input 36 // AUDIO1 FILTER R ROTARY SWITCH

La variable del filtro B del audio no está implementada a nivel de hardware pero puede implementarse por software.

Var 470, name S_AU1ALT, Link IOCARD_SW, DEVICE 40, Input 40 // AUDIO1 ALT-NORM SWITCH

Var 472, name S_AU2VH1MI, Link IOCARD_SW, DEVICE 40, Input 99 // AUDIO2 VHF1 MIC SWITCH

Var 474, name S_AU2VH2MI, Link IOCARD_SW, DEVICE 40, Input 100 // AUDIO2 VHF2 MIC SWITCH

Var 476, name S_AU2VH1VO, Link IOCARD_SW, DEVICE 40, Input 101 // AUDIO2 VHF1 VOL SWITCH

Var 478, name S_AU2VH2VO, Link IOCARD_SW, DEVICE 40, Input 102 // AUDIO2 VHF2 VOL SWITCH

Var 480, name S_AU2NA1VO, Link IOCARD_SW, DEVICE 40, Input 103 // AUDIO2 NAV1 VOL SWITCH

Var 482, name S_AU2NA2VO, Link IOCARD_SW, DEVICE 40, Input 104 // AUDIO2 NAV2 VOL SWITCH

Var 484, name S_AU2AD1VO, Link IOCARD_SW, DEVICE 40, Input 105 // AUDIO2 ADF1 VOL SWITCH

Var 486, name S_AU2AD2VO, Link IOCARD_SW, DEVICE 40, Input 106 // AUDIO2 ADF2 VOL SWITCH

Var 488, name S_AU2MKRVO, Link IOCARD_SW, DEVICE 40, Input 107 // AUDIO2 MARKER VOL SWITCH

Var 490, name S_AU2RT, Link IOCARD_SW, DEVICE 40, Input 110 // AUDIO2 R-T SWITCH

Var 492, name S_AU2IC, Link IOCARD_SW, DEVICE 40, Input 111 // AUDIO2 I-C SWITCH

Var 494, name R_AU2FILTV, Link IOCARD_SW, DEVICE 40, Input 109 // AUDIO2 FILTER V ROTARY SWITCH

Var 496, name R_AU2FILTR, Link IOCARD_SW, DEVICE 40, Input 108 // AUDIO2 FILTER R ROTARY SWITCH

Var 498, name S_AU2ALT, Link IOCARD_SW, DEVICE 40, Input 112 // AUDIO2 ALT-NORM SWITCH

La variable del filtro B del audio no está implementada a nivel de hardware pero puede implementarse por software.

// FIRE ENGINES

Var 500, name S_OVHTDET1A, Link IOCARD_SW, DEVICE 40, Input 46 // FIRE ENGINE 1 OVERHEAT A DETECTOR SWITCH

Var 502, name S_OVHTDET1B, Link IOCARD_SW, DEVICE 40, Input 45 // FIRE ENGINE 1 OVERHEAT B DETECTOR SWITCH

Var 504, name S_FETSTOVH, Link IOCARD_SW, DEVICE 40, Input 47 // FIRE ENGINES TEST OVH-FIRE SWITCH

Var 506, name S_FETSTFAULT, Link IOCARD_SW, DEVICE 40, Input 48 // FIRE ENGINES TEST FAULT-INOP SWITCH

Var 508, name S_BELLCOUT, Link IOCARD_SW, DEVICE 40, Input 49 // FIRE ENGINES BELL CUT OUT SWITCH

Var 510, name S_OVHTDET2A, Link IOCARD_SW, DEVICE 40, Input 51 // FIRE ENGINE 2 OVERHEAT A DETECTOR SWITCH

Var 512, name S_OVHTDET2B, Link IOCARD_SW, DEVICE 40, Input 50 // FIRE ENGINE 2
 OVERHEAT B DETECTOR SWITCH
 Var 514, name S_EXT1TEST, Link IOCARD_SW, DEVICE 40, Input 53 // FIRE ENGINE
 EXTINGUISHER 1 TEST SWITCH
 Var 516, name S_EXT2TEST, Link IOCARD_SW, DEVICE 40, Input 52 // FIRE ENGINE
 EXTINGUISHER 2 TEST SWITCH
 Var 518, name S_HND1DW, Link IOCARD_SW, DEVICE 40, Input 55 // FIRE ENGINE
 HANDLE 1 DW SWITCH
 Var 520, name S_HND1L, Link IOCARD_SW, DEVICE 40, Input 54 // FIRE ENGINE HANDLE
 1 LEFT SWITCH
 Var 522, name S_HND1R, Link IOCARD_SW, DEVICE 40, Input 56 // FIRE ENGINE HANDLE
 1 RIGHT SWITCH
 Var 524, name S_HNDADW, Link IOCARD_SW, DEVICE 40, Input 58 // FIRE ENGINE
 HANDLE APU DW SWITCH
 Var 526, name S_HNDAL, Link IOCARD_SW, DEVICE 40, Input 57 // FIRE ENGINE HANDLE
 APU LEFT SWITCH
 Var 528, name S_HNDAR, Link IOCARD_SW, DEVICE 40, Input 59 // FIRE ENGINE HANDLE
 APU RIGHT SWITCH
 Var 530, name S_HND2DW, Link IOCARD_SW, DEVICE 40, Input 61 // FIRE ENGINE
 HANDLE 2 DW SWITCH
 Var 532, name S_HND2L, Link IOCARD_SW, DEVICE 40, Input 60 // FIRE ENGINE HANDLE
 2 LEFT SWITCH
 Var 534, name S_HND2R, Link IOCARD_SW, DEVICE 40, Input 62 // FIRE ENGINE HANDLE
 2 RIGHT SWITCH
 // FIRE CARGO
 Var 536, name R_FDETFA, Link IOCARD_SW, DEVICE 40, Input 123 // FIRE CARGO DET
 SELECT FWD A ROTARY SWITCH
 Var 538, name R_FDETFNORM, Link IOCARD_SW, DEVICE 40, Input 122 // FIRE CARGO
 DET SELECT FWD NORM ROTARY SWITCH
 Var 540, name R_FDETFB, Link IOCARD_SW, DEVICE 40, Input 121 // FIRE CARGO DET
 SELECT FWD B ROTARY SWITCH
 Var 542, name R_FDETA, Link IOCARD_SW, DEVICE 40, Input 120 // FIRE CARGO DET
 SELECT AFT A ROTARY SWITCH
 Var 544, name R_FDETANORM, Link IOCARD_SW, DEVICE 40, Input 119 // FIRE CARGO
 DET SELECT AFT NORM ROTARY SWITCH
 Var 546, name R_FDETAB, Link IOCARD_SW, DEVICE 40, Input 118 // FIRE CARGO DET
 SELECT AFT B ROTARY SWITCH
 Var 548, name S_FARMF, Link IOCARD_SW, DEVICE 40, Input 128 // FIRE CARGO FWD
 ARMED KORRY SWITCH
 Var 550, name S_FARMA, Link IOCARD_SW, DEVICE 40, Input 127 // FIRE CARGO AFT
 ARMED KORRY SWITCH
 Var 552, name S_FTEST, Link IOCARD_SW, DEVICE 40, Input 117 // FIRE CARGO TEST
 SWITCH
 Var 554, name S_FDISCH, Link IOCARD_SW, DEVICE 40, Input 126 // FIRE CARGO
 DISCHARGE KORRY SWITCH
 // STAB TRIM
 Var 556, name S_STBOVRD, Link IOCARD_SW, DEVICE 40, Input 135 // STAB TRIM
 OVERRIDE-NORM SWITCH
 Var 558, name R_STBDUNLK, Link IOCARD_SW, DEVICE 40, Input 136 // STAB TRIM DOOR
 UNLOCK ROTARY SWITCH
 Var 560, name R_STBDAUTO, Link IOCARD_SW, DEVICE 40, Input 137 // STAB TRIM DOOR
 AUTO ROTARY SWITCH
 Var 562, name R_STBDDENY, Link IOCARD_SW, DEVICE 40, Input 138 // STAB TRIM DOOR
 DENY ROTARY SWITCH
 //RUDDER TRIM
 Var 564, name S_RDAILUL, Link IOCARD_SW, DEVICE 40, Input 63 // RUDDER TRIM
 AILERON UP LEFT SWITCH
 Var 566, name S_RDAILUR, Link IOCARD_SW, DEVICE 40, Input 64 // RUDDER TRIM
 AILERON UP RIGHT SWITCH

Var 568, name S_RDAILDL, Link IOCARD_SW, DEVICE 40, Input 65 // RUDDER TRIM
AILERON DOWN LEFT SWITCH

Var 570, name S_RDAILDR, Link IOCARD_SW, DEVICE 40, Input 66 // RUDDER TRIM
AILERON DOWN RIGHT SWITCH

// RADAR

Var 572, name R_RADTST, Link IOCARD_SW, DEVICE 40, Input 69 // RADAR TEST
ROTARY SWITCH

Var 574, name R_RADWX, Link IOCARD_SW, DEVICE 40, Input 70 // RADAR WX ROTARY
SWITCH

Var 575, name R_RADTURB, Link IOCARD_SW, DEVICE 40, Input 71 // RADAR WX-TURB
ROTARY SWITCH

La variable del radar MAP no está implementada a nivel de hardware pero puede implementarse por software.

// ANALOG

Var 576, name A_LFLOOD, Link IOCARD_ANALOGIC, DEVICE 40, Input 1, PosL 0, PosC
127, PosR 255 // ILLUMINATION FLOOD POTENTIOMETER

Var 578, name A_RADGAIN, Link IOCARD_ANALOGIC, DEVICE 40, Input 2, PosL 0, PosC
127, PosR 255 // RADAR GAIN POTENTIOMETER

Var 580, name A_RADTILT, Link IOCARD_ANALOGIC, DEVICE 40, Input 3, PosL 0, PosC
127, PosR 255 // RADAR TILT POTENTIOMETER

// SERVO

Var 582, name SERVO_RUDDER, Link USB_SERVOS, Device 41, Output 1, PosL 150, PosC
512, PosR 1023, Type 1

Anexo 2. Script específico para FSX y Project Magenta:

Basándonos en estas definiciones de entradas y salidas podemos generar un script específico para cada add-on o avión, como por ejemplo el B737 que viene por defecto en FSX:

```
// *****
// * Config_SIOC ver 5.0 - By Manolo Vélez - www.opencockpits.com
// *****
// * FileName : Script pedestal FSX.txt Ver.1.0
// * PMagenta & FSX Pedestal script
// * Date : 22/06/2014

var 0,value 0
{
  &ent_com1 = 118
  &dec_com1 = 0
  &act_com1 = 118000
  &enc_com1 = 1
  &ent_com2 = 118
  &dec_com2 = 0
  &act_com2 = 118000
  &enc_com2 = 1
  &ent_nav1 = 108
  &dec_nav1 = 0
  &act_nav1 = 10800
  &enc_nav1 = 1
  &ent_nav2 = 108
  &dec_nav2 = 0
  &act_nav2 = 10800
  &enc_nav2 = 1
  &ent_adf1 = 10
  &dec_adf1 = 0
  &act_adf1 = 1000
  &enc_adf1 = 1
  &danta_adf1 = 1
  &dants_adf1 = 1
  &dadfa_adf1 = 0
  &dads_adf1 = 0
  &ent_adf2 = 10
  &dec_adf2 = 0
  &act_adf2 = 1000
  &enc_adf2 = 1
  &danta_adf2 = 1
  &dants_adf2 = 1
  &dadfa_adf2 = 0
  &dads_adf2 = 0
// ATC
  &timerup = 0
  &D_ACTIVE = 1200
  &lastst = 0
  &DIG_POS_STATUS = 5
  &DIS_ATC = 1
  &DIS_1 = 1
  &DIS_2 = 0
  &timeratc = 0
  &timeratc = TIMER 1,0,35
}

// *****
// ***** COM1 *****
// *****
Var 2, name enc_com1
{
  if &enc_com1 = 1
```

```

{
  &io_pt_com1 = 1
  &io_pt2_com1 = 1
  call &sub_d_com1
  &id2_com1 = &act_com1
}
else
{
  &io_pt_com1 = 0
  &io_pt2_com1 = 0
  &id_com1 = -999999
  &id2_com1 = -999999
}
}
var 4, name ent_com1
var 6, name dec_com1
var 8, name ie_dec_com1, link IOCARD_ENCODER, input 0, type 2, aceleration 1, device 40
{
  if &enc_com1 = 1
  {
    I0 = &ie_dec_com1 * 25
    I1 = &dec_com1 + I0
    if I1 < 0
    {
      I1 = 975
    }
    if I1 > 975
    {
      I1 = 0
    }
    &dec_com1 = I1
    call &sub_d_com1
  }
}
var 10, name ie_ent_com1, link IOCARD_ENCODER, input 2, type 2, aceleration 1, device 40
{
  if &enc_com1 = 1
  {
    I0 = &ie_ent_com1 * -1
    &ent_com1 = rotate 118,135,I0
    call &sub_d_com1
  }
}
var 12, name id_com1,link IOCARD_DISPLAY, digit 6, numbers 6, device 40
var 14, name sub_d_com1, link SUBROUTINE
{
  I0 = &ent_com1 * 1000
  &id_com1 = I0 + &dec_com1
}
var 16, name io_pt_com1, link IOCARD_OUT, output 11, device 40
var 18, name ii_sw_com1, link IOCARD_SW, input 5, device 40
{
  if &ii_sw_com1 = 1
  {
    if &enc_com1 = 1
    {
      I0 = &ent_com1 * 1000
      I0 = I0 + &dec_com1
      I1 = div &act_com1 1000
      I2 = mod &act_com1 1000
      &act_com1 = I0
      &ent_com1 = I1
      &dec_com1 = I2
      call &sub_d_com1
      &id2_com1 = &act_com1
    }
  }
}

```

```

}
}
}
var 20, name act_com1
var 22, name id2_com1, link IOCARD_DISPLAY, digit 0, numbers 6, device 40
{
  if &id2_com1 > 0
  {
    &xp_com1 = div &id2_com1 10
  }
}
var 24, name io_pt2_com1, link IOCARD_OUT, output 11, device 40
var 26, name xp_com1, value 11800
{
  l0 = &xp_com1 - 10000
  &fcom1 = tobcd l0
}
// *****
// ***** COM2 *****
// *****
var 102, name enc_com2
{
  if &enc_com2 = 1
  {
    &io_pt_com2 = 1
    &io_pt2_com2 = 1
    call &sub_d_com2
    &id2_com2 = &act_com2
  }
  else
  {
    &io_pt_com2 = 0
    &io_pt2_com2 = 0
    &id_com2 = -999999
    &id2_com2 = -999999
  }
}
var 104, name ent_com2
var 106, name dec_com2
var 108, name ie_dec_com2, link IOCARD_ENCODER, input 72, type 2, aceleration 1, device 40
{
  if &enc_com2 = 1
  {
    l0 = &ie_dec_com2 * 25
    l1 = &dec_com2 + l0
    if l1 < 0
    {
      l1 = 975
    }
    if l1 > 975
    {
      l1 = 0
    }
    &dec_com2 = l1
    call &sub_d_com2
  }
}
var 110, name ie_ent_com2, link IOCARD_ENCODER, input 74, type 2, aceleration 1, device 40
{
  if &enc_com2 = 1
  {
    l0 = &ie_ent_com2 * -1
    &ent_com2 = rotate 118,135,l0
    call &sub_d_com2
  }
}

```

```

}
var 112, name id_com2, link IOCARD_DISPLAY, digit 70, numbers 6, device 40
var 114, name sub_d_com2, link SUBROUTINE
{
  I0 = &ent_com2 * 1000
  &id_com2 = I0 + &dec_com2
}
var 116, name io_pt_com2, link IOCARD_OUT, output 75, device 40
var 118, name ii_sw_com2, link IOCARD_SW, input 77, device 40
{
  if &ii_sw_com2 = 1
  {
    if &enc_com2 = 1
    {
      I0 = &ent_com2 * 1000
      I0 = I0 + &dec_com2
      I1 = div &act_com2 1000
      I2 = mod &act_com2 1000
      &act_com2 = I0
      &ent_com2 = I1
      &dec_com2 = I2
      call &sub_d_com2
      &id2_com2 = &act_com2
    }
  }
}
var 120, name act_com2
var 122, name id2_com2, link IOCARD_DISPLAY, digit 64, numbers 6, device 40
{
  if &id2_com2 > 0
  {
    &xp_com2 = div &id2_com2 10
  }
}
var 124, name io_pt2_com2, link IOCARD_OUT, output 75, device 40
var 126, name xp_com2, value 11800
{
  I0 = &xp_com2 - 10000
  &fcom2 = tobcd I0
}
// *****
// ***** NAV1 *****
// *****
var 54, name enc_nav1
{
  if &enc_nav1 = 1
  {
    &io_pt_nav1 = 1
    &io_pt2_nav1 = 1
    call &sub_d_nav1
    &id2_nav1 = &act_nav1
  }
  else
  {
    &io_pt_nav1 = 0
    &io_pt2_nav1 = 0
    &id_nav1 = -999999
    &id2_nav1 = -999999
  }
}
var 56, name ent_nav1
var 58, name dec_nav1
var 60, name ie_dec_nav1, link IOCARD_ENCODER, input 9, type 2, aceleration 1, device 40
{
  if &enc_nav1 = 1

```

```

{
  I0 = &ie_dec_nav1 * 25
  I1 = &dec_nav1 + I0
  if I1 < 0
  {
    I1 = 975
  }
  if I1 > 975
  {
    I1 = 0
  }
  &dec_nav1 = I1
  call &sub_d_nav1
}
}
var 62, name ie_ent_nav1, link IOCARD_ENCODER, input 11, type 2, aceleration 1, device 40
{
  if &enc_nav1 = 1
  {
    I0 = &ie_ent_nav1 * -1
    &ent_nav1 = rotate 108,117,I0
    call &sub_d_nav1
  }
}
var 64, name id_nav1,link IOCARD_DISPLAY, digit 21, numbers 5, device 40
var 66, name sub_d_nav1, link SUBROUTINE
{
  I0 = &ent_nav1 * 100
  I1 = div &dec_nav1 10
  &id_nav1 = I0 + I1
}
var 68, name io_pt_nav1, link IOCARD_OUT, output 12, device 40
var 70, name ii_sw_nav1, link IOCARD_SW, input 14, device 40
{
  if &ii_sw_nav1 = 1
  {
    if &enc_nav1 = 1
    {
      I0 = &ent_nav1 * 100
      I1 = div &dec_nav1 10
      I0 = I0 + I1
      I1 = div &act_nav1 100
      I2 = mod &act_nav1 100
      &act_nav1 = I0
      &ent_nav1 = I1
      I0 = mod I2 10
      I2 = I2 * 10
      if I0 > 0
      {
        I2 = I2 + 5
      }
      &dec_nav1 = I2
      call &sub_d_nav1
      &id2_nav1 = &act_nav1
    }
  }
}
var 72, name act_nav1
var 74, name id2_nav1,link IOCARD_DISPLAY, digit 16, numbers 5, device 40
{
  if &id2_nav1 > 0
  {
    &xp_nav1 = &id2_nav1
  }
}
}

```

```

var 76, name io_pt2_nav1, link IOCARD_OUT, output 12, device 40
var 78, name xp_nav1, value 10800
{
  I0 = &xp_nav1 - 10000
  &fnav1 = tobcd I0
}
// *****
// ***** NAV2 *****
// *****
var 154, name enc_nav2
{
  if &enc_nav2 = 1
  {
    &io_pt_nav2 = 1
    &io_pt2_nav2 = 1
    call &sub_d_nav2
    &id2_nav2 = &act_nav2
  }
  else
  {
    &io_pt_nav2 = 0
    &io_pt2_nav2 = 0
    &id_nav2 = -999999
    &id2_nav2 = -999999
  }
}
var 156, name ent_nav2
var 158, name dec_nav2
var 160, name ie_dec_nav2, link IOCARD_ENCODER, input 81, type 2, aceleration 1, device 40
{
  if &enc_nav2 = 1
  {
    I0 = &ie_dec_nav2 * 25
    I1 = &dec_nav2 + I0
    if I1 < 0
    {
      I1 = 975
    }
    if I1 > 975
    {
      I1 = 0
    }
    &dec_nav2 = I1
    call &sub_d_nav2
  }
}
var 162, name ie_ent_nav2, link IOCARD_ENCODER, input 83, type 2, aceleration 1, device 40
{
  if &enc_nav2 = 1
  {
    I0 = &ie_ent_nav2 * -1
    &ent_nav2 = rotate 108,117,I0
    call &sub_d_nav2
  }
}
var 164, name id_nav2,link IOCARD_DISPLAY, digit 85, numbers 5, device 40
var 166, name sub_d_nav2, link SUBROUTINE
{
  I0 = &ent_nav2 * 100
  I1 = div &dec_nav2 10
  &id_nav2 = I0 + I1
}
var 168, name io_pt_nav2, link IOCARD_OUT, output 76, device 40
var 170, name ii_sw_nav2, link IOCARD_SW, input 86, device 40
{

```

```

if &ii_sw_nav2 = 1
{
  if &enc_nav2 = 1
  {
    I0 = &ent_nav2 * 100
    I1 = div &dec_nav2 10
    I0 = I0 + I1
    I1 = div &act_nav2 100
    I2 = mod &act_nav2 100
    &act_nav2 = I0
    &ent_nav2 = I1
    I0 = mod I2 10
    I2 = I2 * 10
    if I0 > 0
    {
      I2 = I2 + 5
    }
    &dec_nav2 = I2
    call &sub_d_nav2
    &id2_nav2 = &act_nav2
  }
}
}
var 172, name act_nav2
var 174, name id2_nav2, link IOCARD_DISPLAY, digit 80, numbers 5, device 40
{
  if &id2_nav2 > 0
  {
    &xp_nav2 = &id2_nav2
  }
}
var 176, name io_pt2_nav2, link IOCARD_OUT, output 76, device 40
var 178, name xp_nav2, value 10800
{
  I0 = &xp_nav2 - 10000
  &fnav2 = tobcd I0
}
// *****
// ***** ADF1 *****
// *****
var 218, name enc_adf1
{
  if &enc_adf1 = 1
  {
    &io_pt_adf1 = 1
    call &sub_d_adf1
    &id2_adf1 = &act_adf1
    &o_adf1 = 1
  }
  else
  {
    &io_pt_adf1 = 0
    &id_adf1 = -999999
    &id2_adf1 = -999999
    &o_adf1 = 0
  }
}
}
var 220, name ent_adf1
var 222, name dec_adf1
var 224, name ie_dec_adf1, link IOCARD_ENCODER, input 18, type 2, aceleration 5, device 40
{
  if &enc_adf1 = 1
  {
    I0 = &ie_dec_adf1 * 1
    I1 = &dec_adf1 + I0
  }
}

```

```

if l1 < 0
{
  l1 = 99
}
if l1 > 99
{
  l1 = 0
}
&dec_adf1 = l1
call &sub_d_adf1
}
}
var 226, name ie_ent_adf1, link IOCARD_ENCODER, input 20, type 2, aceleration 4, device 40
{
  if &enc_adf1 = 1
  {
    l0 = &ie_ent_adf1 * -1
    l1 = &ent_adf1 + l0
    if l1 < 10
    {
      l1 = 179
    }
    if l1 > 179
    {
      l1 = 10
    }
    &ent_adf1 = l1
    call &sub_d_adf1
  }
}
var 228, name id_adf1,link IOCARD_DISPLAY, digit 37, numbers 5, device 40
var 230, name sub_d_adf1, link SUBROUTINE
{
  l0 = &ent_adf1 * 100
  l1 = &dec_adf1
  &id_adf1 = l0 + l1
}
var 234, name ii_sw_adf1, link IOCARD_SW, input 23,device 40
{
  if &ii_sw_adf1 = 1
  {
    if &enc_adf1 = 1
    {
      l0 = &ent_adf1 * 100
      l1 = &dec_adf1
      l0 = l0 + l1
      l1 = div &act_adf1 100
      l2 = mod &act_adf1 100
      &act_adf1 = l0
      &ent_adf1 = l1
      &dec_adf1 = l2
      call &sub_d_adf1
      &id2_adf1 = &act_adf1
    }
  }
}
var 236, name act_adf1
var 238, name id2_adf1,link IOCARD_DISPLAY, digit 32, numbers 5, device 40
{
  if &id2_adf1 > 0
  {
    &xp_adf1 = &id2_adf1
  }
}
}

```



```

var 240, name io_pt_adf1, link IOCARD_OUT, output 13, device 40
Var 242, name o_adf1
var 254, name xp_adf1
{
  I0 = div &xp_adf1 10000
  I1 = I0 * 10000
  I2 = &xp_adf1 - I1
  I2 = div I2 10
  &fadf1A = tobcd I2
  I0 = I0 * 100
  I1 = div &xp_adf1 10
  I1 = I1 * 10
  I2 = &xp_adf1 - I1
  I2 = I2 + I0
  &fadf1B = tobcd I2
}
var 256, name ant_adf1, link IOCARD_SW, input 22, device 40
{
  if &ant_adf1 = 1
  {
    &danta_adf1 = 0
    &dants_adf1 = 0
    &dadfa_adf1 = 1
    &dadfs_adf1 = 1
  }
  else
  {
    &danta_adf1 = 1
    &dants_adf1 = 1
    &dadfa_adf1 = 0
    &dadfs_adf1 = 0
  }
}
var 258, name danta_adf1, link IOCARD_DISPLAY, digit 43, numbers 1, device 40
var 260, name dants_adf1, link IOCARD_DISPLAY, digit 44, numbers 1, device 40
var 262, name dadfa_adf1, link IOCARD_DISPLAY, digit 42, numbers 1, device 40
var 264, name dadfs_adf1, link IOCARD_DISPLAY, digit 45, numbers 1, device 40
Var 266, name S_ADF1TONE, Link IOCARD_SW, Input 24, device 40
{
  &toneadf1 = CHANGEBITN 0 &S_ADF1TONE
}
// *****
// ***** ADF2 *****
// *****
var 318, name enc_adf2
{
  if &enc_adf2 = 1
  {
    &io_pt_adf2 = 1
    call &sub_d_adf2
    &id2_adf2 = &act_adf2
    &o_adf2 = 1
  }
  else
  {
    &io_pt_adf2 = 0
    &id_adf2 = -999999
    &id2_adf2 = -999999
    &o_adf2 = 0
  }
}
var 320, name ent_adf2
var 322, name dec_adf2
var 324, name ie_dec_adf2, link IOCARD_ENCODER, input 90, type 2, aceleration 5, device 40
{

```

```

if &enc_adf2 = 1
{
  I0 = &ie_dec_adf2 * 1
  I1 = &dec_adf2 + I0
  if I1 < 0
  {
    I1 = 99
  }
  if I1 > 99
  {
    I1 = 0
  }
  &dec_adf2 = I1
  call &sub_d_adf2
}
}
var 326, name ie_ent_adf2, link IOCARD_ENCODER, input 92, type 2, aceleration 4, device 40
{
  if &enc_adf2 = 1
  {
    I0 = &ie_ent_adf2 * -1
    I1 = &ent_adf2 + I0
    if I1 < 10
    {
      I1 = 179
    }
    if I1 > 179
    {
      I1 = 10
    }
    &ent_adf2 = I1
    call &sub_d_adf2
  }
}
var 328, name id_adf2,link IOCARD_DISPLAY, digit 101, numbers 5, device 40
var 330, name sub_d_adf2, link SUBROUTINE
{
  I0 = &ent_adf2 * 100
  I1 = &dec_adf2
  &id_adf2 = I0 + I1
}
var 334, name ii_sw_adf2, link IOCARD_SW, input 95,device 40
{
  if &ii_sw_adf2 = 1
  {
    if &enc_adf2 = 1
    {
      I0 = &ent_adf2 * 100
      I1 = &dec_adf2
      I0 = I0 + I1
      I1 = div &act_adf2 100
      I2 = mod &act_adf2 100
      &act_adf2 = I0
      &ent_adf2 = I1
      &dec_adf2 = I2
      call &sub_d_adf2
      &id2_adf2 = &act_adf2
    }
  }
}
var 336, name act_adf2
var 338, name id2_adf2,link IOCARD_DISPLAY, digit 96, numbers 5, device 40
{
  if &id2_adf2 > 0
  {

```

```

    &xp_adf2 = &id2_adf2
}
}
var 340, name io_pt_adf2, link IOCARD_OUT, output 77, device 40
Var 342, name o_adf2
var 354, name xp_adf2
{
  I0 = div &xp_adf2 10000
  I1 = I0 * 10000
  I2 = &xp_adf2 - I1
  I2 = div I2 10
  &fadf2A = tobcd I2
  I0 = I0 * 100
  I1 = div &xp_adf2 10
  I1 = I1 * 10
  I2 = &xp_adf2 - I1
  I2 = I2 + I0
  &fadf2B = tobcd I2
}
var 356, name ant_adf2, link IOCARD_SW, input 94, device 40
{
  if &ant_adf2 = 1
  {
    &danta_adf2 = 0
    &dants_adf2 = 0
    &dadfa_adf2 = 1
    &dadfs_adf2 = 1
  }
  else
  {
    &danta_adf2 = 1
    &dants_adf2 = 1
    &dadfa_adf2 = 0
    &dadfs_adf2 = 0
  }
}
var 358, name danta_adf2, link IOCARD_DISPLAY, digit 107, numbers 1, device 40
var 360, name dants_adf2, link IOCARD_DISPLAY, digit 108, numbers 1, device 40
var 362, name dadfa_adf2, link IOCARD_DISPLAY, digit 106, numbers 1, device 40
var 364, name dadfs_adf2, link IOCARD_DISPLAY, digit 109, numbers 1, device 40
Var 366, name S_ADF2TONE, Link IOCARD_SW, Input 96, device 40
{
  &toneadf2 = CHANGEBITN 0 &S_ADF2TONE
}
// *****
// ***** ATC *****
// *****
Var 400, name DIG_POS_STATUS
{
  &finatc = 0
  IF &DIG_POS_STATUS = 5
  {
    &timerup = 0
    CALL &DIGITS2_SUB
  }
  else
  {
    &timerup = 1
    CALL &DIGITS2_SUB
  }
}
Var 402, name DIG_1_SUB, Value 0
{
  CALL &DIGITS_SUB
}

```

```

Var 404, name DIG_2_SUB, Value 0
{
  CALL &DIGITS_SUB
}
Var 406, name DIG_3_SUB, Value 2
{
  CALL &DIGITS_SUB
}
Var 408, name DIG_4_SUB, Value 1
{
  CALL &DIGITS_SUB
}
Var 410, name D_ACTIVE, Link IOCARD_DISPLAY, Device 40, Digit 90, Numbers 4
Var 412, name D_FAIL, Link IOCARD_DISPLAY, Device 40, Digit 94, Numbers 1
Var 416, name DIS_ATC, Link IOCARD_OUT, Device 40, Output 99
Var 418, name DIS_1, Link IOCARD_OUT, Device 40, Output 100
Var 420, name DIS_2, Link IOCARD_OUT, Device 40, Output 101
Var 422, name LED_FAIL, Link IOCARD_OUT, Device 40, Output 98
Var 424, name E_LEFT, Link IOCARD_ENCODER, Device 40, Input 113, Aceleration 0, Type 2
{
  L0 = &E_LEFT * 1
  &DIG_POS_STATUS = ROTATE 1 ,5 ,L0
}
Var 426, name E_RIGHT, Link IOCARD_ENCODER, Device 40, Input 115, Aceleration 1, Type 2
{
  IF &DIG_POS_STATUS = 1
  {
    L0 = &E_RIGHT * -1
    &DIG_1_SUB = ROTATE 0 ,7 ,L0
  }
  IF &DIG_POS_STATUS = 2
  {
    L0 = &E_RIGHT * -1
    &DIG_2_SUB = ROTATE 0 ,7 ,L0
  }
  IF &DIG_POS_STATUS = 3
  {
    L0 = &E_RIGHT * -1
    &DIG_3_SUB = ROTATE 0 ,7 ,L0
  }
  IF &DIG_POS_STATUS = 4
  {
    L0 = &E_RIGHT * -1
    &DIG_4_SUB = ROTATE 0 ,7 ,L0
  }
}
Var 428, name I_IDENT, Link IOCARD_SW, Device 40, Input 78
{
  IF &I_IDENT = 1
  {
    &IVAP_SB_IDENT = 1
  }
}
Var 430, name I_XPNDR, Link IOCARD_SW, Device 40, Input 79
{
  IF &STANDBY = 0
  {
    CALL &check_xp
  }
}
var 431, name check_xp, link subrutine
{
  IF &I_XPNDR = 1
  {
    &DIS_1 = 0
  }
}

```

```

    &DIS_2 = 1
  }
  ELSE
  {
    &DIS_1 = 1
    &DIS_2 = 0
  }
}
Var 432, name STANDBY, Link IOCARD_SW, Device 40, Input 139
{
  IF &STANDBY = 1
  {
    &IVAP_SB_MODE = 1
    &DIS_1 = 0
    &DIS_2 = 0
  }
  ELSE
  {
    CALL &check_xp
  }
}
Var 434, name I_R_ALT, Link IOCARD_SW, Device 40, Input 140
Var 436, name I_R_XPN, Link IOCARD_SW, Device 40, Input 141
{
  IF &I_R_XPN = 1
  {
    &IVAP_SB_MODE = 0
  }
}
Var 438, name I_R_TAO, Link IOCARD_SW, Device 40, Input 142
{
  IF &I_R_TAO = 1
  {
    &TCAS_MAGENTA = 51
  }
}
Var 440, name I_R_TAR, Link IOCARD_SW, Device 40, Input 143
{
  IF &I_R_TAR = 1
  {
    &TCAS_MAGENTA = 51
  }
}
Var 442, name DIGITS_SUB, Link SUBROUTINE
{
  L0 = &DIG_4_SUB * 1000
  L1 = &DIG_3_SUB * 100
  L0 = L0 + L1
  L1 = &DIG_2_SUB * 10
  L0 = L0 + L1
  L0 = L0 + &DIG_1_SUB
  &D_ACTIVE = L0
  &FS_XPDER_ID = TOBCD &D_ACTIVE
}
// ***** TIMER *****
// *****
var 444, name timeratc, link subroutine
{
  if &timerup = 0
  {
    &lastst = 0
  }
  else
  {
    IF &DIG_POS_STATUS = 1

```

```

{
  IF &lastst = 0
  {
    &D_ATC1 = -999999
  }
  ELSE
  {
    &D_ATC1 = &DIG_1_SUB
  }
}
IF &DIG_POS_STATUS = 2
{
  IF &lastst = 0
  {
    &D_ATC2 = -999999
  }
  ELSE
  {
    &D_ATC2 = &DIG_2_SUB
  }
}
IF &DIG_POS_STATUS = 3
{
  IF &lastst = 0
  {
    &D_ATC3 = -999999
  }
  ELSE
  {
    &D_ATC3 = &DIG_3_SUB
  }
}
IF &DIG_POS_STATUS = 4
{
  IF &lastst = 0
  {
    &D_ATC4 = -999999
  }
  ELSE
  {
    &D_ATC4 = &DIG_4_SUB
  }
}
IF &lastst = 0
{
  &lastst = 1
}
ELSE
{
  &lastst = 0
}
IF &finatc = 50
{
  &DIG_POS_STATUS = 5
}
ELSE
{
  &finatc = &finatc + 1
}
}
// DOOR
IF &lockcount > 0
{
  &lockcount = &lockcount - 1
}

```

```

    &STBAUTOL = CHANGEBITN 0 &STBAUTOL
}
ELSE
{
    &STBAUTOL = 0
}
// CARGO
IF &contacargo > 0
{
    &contacargo = &contacargo + 1
    IF &contacargo > 5
    {
        &contacargo = 0
        &FCAFTL = 1
        &FCFWDL = 1
    }
}
// AILTRIM
IF &S_RDAILUL = 1
{
    IF &S_RDAILDL = 1
    {
        &ailt_o = LIMIT -16000, 16000, -500
    }
}
IF &S_RDAILUR = 1
{
    IF &S_RDAILDR = 1
    {
        &ailt_o = LIMIT -16000, 16000, 500
    }
}
}
var 446, name lastst
var 448, name timerup
Var 450, name D_ATC1, Link IOCARD_DISPLAY, Device 40, Digit 90, Numbers 1
Var 452, name D_ATC2, Link IOCARD_DISPLAY, Device 40, Digit 91, Numbers 1
Var 454, name D_ATC3, Link IOCARD_DISPLAY, Device 40, Digit 92, Numbers 1
Var 456, name D_ATC4, Link IOCARD_DISPLAY, Device 40, Digit 93, Numbers 1
var 458, name DIGITS2_SUB, link subroutine
{
    &D_ATC1 = &DIG_1_SUB
    &D_ATC2 = &DIG_2_SUB
    &D_ATC3 = &DIG_3_SUB
    &D_ATC4 = &DIG_4_SUB
}
var 460, name finatc, value 0
// *****
// ***** AUDIO 1 *****
// *****
Var 500, name S_AU1VH1MI, Link IOCARD_SW, DEVICE 40, Input 27 // AUDIO1 VHF1 MIC SWITCH
{
    if &S_AU1VH1MI = 1
    {

        &audio_o = SETBIT 7
        &audio_o = CLEARBIT 6

    }
}
Var 502, name S_AU1VH2MI, Link IOCARD_SW, DEVICE 40, Input 28 // AUDIO1 VHF2 MIC SWITCH
{
    if &S_AU1VH2MI = 1
    {
        &audio_o = CLEARBIT 7
    }
}

```

```

    &audio_o = SETBIT 6
  }
}
Var 504, name AU1VH1MI, Link IOCARD_OUT, DEVICE 40, Output 14 // AUDIO1 VHF1 MIC LED
Var 506, name AU1VH2MI, Link IOCARD_OUT, DEVICE 40, Output 15 // AUDIO1 VHF2 MIC LED
Var 508, name S_AU1VH1VO, Link IOCARD_SW, DEVICE 40, Input 29 // AUDIO1 VHF1 VOL SWITCH
{
  if &S_AU1VH1VO = 1
  {
    I0 = TESTBIT &audio_i,5
    &audio_o = CHANGEBITN 5,I0
  }
}
Var 510, name S_AU1VH2VO, Link IOCARD_SW, DEVICE 40, Input 30 // AUDIO1 VHF2 VOL SWITCH
{
  if &S_AU1VH2VO = 1
  {
    I0 = TESTBIT &audio_i,5
    &audio_o = CHANGEBITN 5,I0
  }
}
Var 512, name AU1VH1VO, Link IOCARD_OUT, DEVICE 40, Output 16 // AUDIO1 VHF1 VOL LED
Var 514, name AU1VH2VO, Link IOCARD_OUT, DEVICE 40, Output 17 // AUDIO1 VHF2 VOL LED
Var 516, name S_AU1NA1VO, Link IOCARD_SW, DEVICE 40, Input 31 // AUDIO1 NAV1 VOL SWITCH
{
  if &S_AU1NA1VO = 1
  {
    I0 = TESTBIT &audio_i,4
    &audio_o = CHANGEBITN 4,I0
  }
}
Var 518, name S_AU1NA2VO, Link IOCARD_SW, DEVICE 40, Input 32 // AUDIO1 NAV2 VOL SWITCH
{
  if &S_AU1NA2VO = 1
  {
    I0 = TESTBIT &audio_i,3
    &audio_o = CHANGEBITN 3,I0
  }
}
Var 520, name S_AU1AD1VO, Link IOCARD_SW, DEVICE 40, Input 33 // AUDIO1 ADF1 VOL SWITCH
{
  if &S_AU1AD1VO = 1
  {
    I0 = TESTBIT &audio_i,0
    &audio_o = CHANGEBITN 0,I0
  }
}
Var 522, name S_AU1AD2VO, Link IOCARD_SW, DEVICE 40, Input 34 // AUDIO1 ADF2 VOL SWITCH
Var 524, name S_AU1MKRVO, Link IOCARD_SW, DEVICE 40, Input 35 // AUDIO1 MARKER VOL SWITCH
{
  if &S_AU1MKRVO = 1
  {
    I0 = TESTBIT &audio_i,2
    &audio_o = CHANGEBITN 2,I0
  }
}
Var 526, name AU1NA1VO, Link IOCARD_OUT, DEVICE 40, Output 18 // AUDIO1 NAV1 VOL LED
Var 528, name AU1NA2VO, Link IOCARD_OUT, DEVICE 40, Output 19 // AUDIO1 NAV2 VOL LED
Var 530, name AU1AD1VO, Link IOCARD_OUT, DEVICE 40, Output 20 // AUDIO1 ADF1 VOL LED
Var 532, name AU1AD2VO, Link IOCARD_OUT, DEVICE 40, Output 21 // AUDIO1 ADF2 VOL LED
Var 534, name AU1MKRVO, Link IOCARD_OUT, DEVICE 40, Output 22 // AUDIO1 MARKER VOL LED

// *****
// ***** AUDIO 2 *****

```



```
// *****
Var 600, name S_AU2VH1MI, Link IOCARD_SW, DEVICE 40, Input 99 // AUDIO1 VHF1 MIC SWITCH
{
  if &S_AU2VH1MI = 1
  {
    &audio_o = SETBIT 7
    &audio_o = CLEARBIT 6
  }
}
Var 602, name S_AU2VH2MI, Link IOCARD_SW, DEVICE 40, Input 100 // AUDIO1 VHF2 MIC SWITCH
{
  if &S_AU2VH2MI = 1
  {
    &audio_o = CLEARBIT 7
    &audio_o = SETBIT 6
  }
}
Var 604, name AU2VH1MI, Link IOCARD_OUT, DEVICE 40, Output 78 // AUDIO1 VHF1 MIC LED
Var 606, name AU2VH2MI, Link IOCARD_OUT, DEVICE 40, Output 79 // AUDIO1 VHF2 MIC LED
Var 608, name S_AU2VH1VO, Link IOCARD_SW, DEVICE 40, Input 101 // AUDIO1 VHF1 VOL SWITCH
{
  if &S_AU2VH1VO = 1
  {
    I0 = TESTBIT &audio_i,5
    &audio_o = CHANGEBITN 5,I0
  }
}
Var 610, name S_AU2VH2VO, Link IOCARD_SW, DEVICE 40, Input 102 // AUDIO1 VHF2 VOL SWITCH
{
  if &S_AU2VH2VO = 1
  {
    I0 = TESTBIT &audio_i,5
    &audio_o = CHANGEBITN 5,I0
  }
}
Var 612, name AU2VH1VO, Link IOCARD_OUT, DEVICE 40, Output 80 // AUDIO1 VHF1 VOL LED
Var 614, name AU2VH2VO, Link IOCARD_OUT, DEVICE 40, Output 81 // AUDIO1 VHF2 VOL LED
Var 616, name S_AU2NA1VO, Link IOCARD_SW, DEVICE 40, Input 103 // AUDIO1 NAV1 VOL SWITCH
{
  if &S_AU2NA1VO = 1
  {
    I0 = TESTBIT &audio_i,4
    &audio_o = CHANGEBITN 4,I0
  }
}
Var 618, name S_AU2NA2VO, Link IOCARD_SW, DEVICE 40, Input 104 // AUDIO1 NAV2 VOL SWITCH
{
  if &S_AU2NA2VO = 1
  {
    I0 = TESTBIT &audio_i,3
    &audio_o = CHANGEBITN 3,I0
  }
}
Var 620, name S_AU2AD1VO, Link IOCARD_SW, DEVICE 40, Input 105 // AUDIO1 ADF1 VOL SWITCH
{
  if &S_AU2AD1VO = 1
  {
    I0 = TESTBIT &audio_i,0
    &audio_o = CHANGEBITN 0,I0
  }
}
Var 622, name S_AU2AD2VO, Link IOCARD_SW, DEVICE 40, Input 106 // AUDIO1 ADF2 VOL SWITCH
Var 624, name S_AU2MKRVO, Link IOCARD_SW, DEVICE 40, Input 107 // AUDIO1 MARKER VOL SWITCH
{
```

```

if &S_AU2MKRVO = 1
{
  I0 = TESTBIT &audio_i,2
  &audio_o = CHANGEBITN 2,I0
}
}
Var 626, name AU2NA1VO, Link IOCARD_OUT, DEVICE 40, Output 82 // AUDIO1 NAV1 VOL LED
Var 628, name AU2NA2VO, Link IOCARD_OUT, DEVICE 40, Output 83 // AUDIO1 NAV2 VOL LED
Var 630, name AU2AD1VO, Link IOCARD_OUT, DEVICE 40, Output 84 // AUDIO1 ADF1 VOL LED
Var 632, name AU2AD2VO, Link IOCARD_OUT, DEVICE 40, Output 85 // AUDIO1 ADF2 VOL LED
Var 634, name AU2MKRVO, Link IOCARD_OUT, DEVICE 40, Output 86 // AUDIO1 MARKER VOL LED
// *****
// ***** STAB TRIM *****
// *****
Var 650, name S_STBOVRD, Link IOCARD_SW, DEVICE 40, Input 135 // STAB TRIM OVERRIDE-
NORM SWITCH
{
  IF &S_STBOVRD = 1
  {
    &stabtrim_o = SETBIT 13
  }
  else
  {
    &stabtrim_o = CLEARBIT 13
  }
}
Var 652, name R_STBDAUTO, Link IOCARD_SW, DEVICE 40, Input 137 // STAB TRIM DOOR AUTO
ROTARY SWITCH
{
  IF &R_STBDAUTO = 1
  {
    &lockcount = 15
  }
}
var 654, name lockcount, value 0
Var 656, name STBAUTOL, Link IOCARD_OUT, DEVICE 40, Output 97 // STAB TRIM AUTO-UNLOCK
INDICATOR
Var 658, name R_STBDUNLK, Link IOCARD_SW, DEVICE 40, Input 136 // STAB TRIM DOOR UNLOCK
ROTARY SWITCH
{
  IF &R_STBDUNLK = 1
  {
    &lockcount = 0
  }
}
Var 660, name R_STBDDENY, Link IOCARD_SW, DEVICE 40, Input 138 // STAB TRIM DOOR DENY
ROTARY SWITCH
{
  IF &R_STBDDENY = 1
  {
    &lockcount = 0
  }
}
// *****
// ***** CARGO *****
// *****
Var 700, name S_FTEST, Link IOCARD_SW, DEVICE 40, Input 117 // FIRE CARGO TEST SWITCH
{
  IF &S_FTEST = 1
  {
    IF &testrun = 0
    {
      &testrun = 1
      &FCDISCL = 1
      &FCDISCBARL = 1
    }
  }
}

```

```

    &FCEXTFWDL = 1
    &FCEXTAFTL = 1
    &contacargo = 1
}
else
{
    &contacargo = 0
    &testrun = 0
    &FCDISCL = 0
    &FCDISCBARL = 0
    &FCEXTFWDL = 0
    &FCEXTAFTL = 0
    &FCAFTL = 0
    &FCFWDL = 0
}
}
}
var 702, name testrun, value 0
var 704, name contacargo, value 0
Var 776, name FCDISCL, Link IOCARD_OUT, DEVICE 40, Output 87 // FIRE CARGO DISCHARGE
KORRY
Var 778, name FCDISCBARL, Link IOCARD_OUT, DEVICE 40, Output 88 // FIRE CARGO DISCHARGE
BAR KORRY
Var 780, name FCAFTL, Link IOCARD_OUT, DEVICE 40, Output 89 // FIRE CARGO AFT KORRY
Var 784, name FCFWDL, Link IOCARD_OUT, DEVICE 40, Output 91 // FIRE CARGO FWD KORRY
Var 790, name FCEXTFWDL, Link IOCARD_OUT, DEVICE 40, Output 94 // FIRE CARGO EXT FWD
Var 792, name FCEXTAFTL, Link IOCARD_OUT, DEVICE 40, Output 95 // FIRE CARGO EXT AFT
// *****
// ***** FIRE ENGINES *****
// *****
Var 804, name S_FETSTOVH, Link IOCARD_SW, DEVICE 40, Input 47 // FIRE ENGINES TEST OVH-
FIRE SWITCH
{
    IF &S_FETSTOVH = 1
    {
        &ENG1_OVHL = 1
        &ENG2_OVHL = 1
        &FIREWHELLL = 1
        &FIRE1L = 1
        &FIREAL = 1
        &FIRE2L = 1
    }
    else
    {
        &ENG1_OVHL = 0
        &ENG2_OVHL = 0
        &FIREWHELLL = 0
        &FIRE1L = 0
        &FIREAL = 0
        &FIRE2L = 0
    }
}
}
Var 806, name S_FETSTFAULT, Link IOCARD_SW, DEVICE 40, Input 48 // FIRE ENGINES TEST
FAULT-INOP SWITCH
{
    IF &S_FETSTFAULT = 1
    {
        &APUDETINOPL = 1
        &FIREFAULTL = 1
    }
    else
    {
        &APUDETINOPL = 0
        &FIREFAULTL = 0
    }
}

```

```

}
Var 814, name S_EXT1TEST, Link IOCARD_SW, DEVICE 40, Input 53 // FIRE ENGINE
EXTINGUISHER 1 TEST SWITCH
{
  IF &S_EXT1TEST = 1
  {
    &FIREG1L = 1
    &FIREGAL = 1
    &FIREG2L = 1
  }
  ELSE
  {
    IF &S_EXT2TEST = 0
    {
      &FIREG1L = 0
      &FIREGAL = 0
      &FIREG2L = 0
    }
  }
}
}
Var 816, name S_EXT2TEST, Link IOCARD_SW, DEVICE 40, Input 52 // FIRE ENGINE
EXTINGUISHER 2 TEST SWITCH
{
  IF &S_EXT2TEST = 1
  {
    &FIREG1L = 1
    &FIREGAL = 1
    &FIREG2L = 1
  }
  ELSE
  {
    IF &S_EXT1TEST = 0
    {
      &FIREG1L = 0
      &FIREGAL = 0
      &FIREG2L = 0
    }
  }
}
}
Var 848, name ENG1_OVHL, Link IOCARD_OUT, DEVICE 40, Output 23 // ENGINE 1 OVERHEAT
Var 852, name APUDETINOPL, Link IOCARD_OUT, DEVICE 40, Output 25 // APU DET INOPERATIVE
Var 854, name FIREFAULTL, Link IOCARD_OUT, DEVICE 40, Output 26 // FIRE ENGINES FAULT
Var 856, name FIREWHELLL, Link IOCARD_OUT, DEVICE 40, Output 27 // FIRE WHELL WELL
Var 858, name ENG2_OVHL, Link IOCARD_OUT, DEVICE 40, Output 28 // ENGINE 2 OVERHEAT
Var 862, name RBOTDISL, Link IOCARD_OUT, DEVICE 40, Output 30 // ENGINE 2 BOTTLE
DISCHARGE
Var 864, name FIRE1L, Link IOCARD_OUT, DEVICE 40, Output 31 // ENGINE 1 FIRE HANDLE LIGHT
Var 866, name FIREAL, Link IOCARD_OUT, DEVICE 40, Output 32 // APU FIRE HANDLE LIGHT
Var 868, name FIRE2L, Link IOCARD_OUT, DEVICE 40, Output 33 // ENGINE 2 FIRE HANDLE LIGHT
Var 870, name FIREG1L, Link IOCARD_OUT, DEVICE 40, Output 34 // FIRE ENGINE 1 GREEN LED
Var 872, name FIREGAL, Link IOCARD_OUT, DEVICE 40, Output 35 // FIRE APU GREEN LED
Var 874, name FIREG2L, Link IOCARD_OUT, DEVICE 40, Output 36 // FIRE ENGINE 2 GREEN LED
// *****
// ***** RADAR *****
// *****
Var 900, name R_RADTST, Link IOCARD_SW, DEVICE 40, Input 69 // RADAR WX-TEST ROTARY
SWITCH
{
  IF &R_RADTST = 1
  {
    &radar_pm = 381
  }
}
}
Var 902, name R_RADMAP, Link IOCARD_SW, DEVICE 40, Input 71 // RADAR WX-TURB-MAP
ROTARY SWITCH

```

```

{
  IF &R_RADMAP = 1
  {
    &radar_pm = 380
  }
}
// *****
// ***** RUDDER TRIM *****
// *****
Var 964, name S_RDAILUL, Link IOCARD_SW, DEVICE 40, Input 63 // RUDDER TRIM AILERON UP
LEFT SWITCH
Var 966, name S_RDAILUR, Link IOCARD_SW, DEVICE 40, Input 64 // RUDDER TRIM AILERON UP
RIGHT SWITCH
Var 968, name S_RDAILDL, Link IOCARD_SW, DEVICE 40, Input 65 // RUDDER TRIM AILERON
DOWN LEFT SWITCH
Var 970, name S_RDAILDR, Link IOCARD_SW, DEVICE 40, Input 66 // RUDDER TRIM AILERON
DOWN RIGHT SWITCH
Var 978, name E_RUDDER, Link IOCARD_ENCODER, DEVICE 40, Input 67, Aceleration 1, Type 2 //
RUDDER TRIM ENCODER
{
  I0 = &E_RUDDER * -500
  &ruddert_o = LIMIT -16000, 16000, I0
}
Var 980, name grudtrim, link USB_SERVOS, DEVICE 41, Output 1, PosL 0, PosC 511, PosR 1023 //
RUDDER TRIM INDICATOR
Var 982, name leftservo, value 623 // ***** Change de value for your left extreme value for 15 in the
indicator ****
var 984, name centerservo, value 510 // ***** change de value for your center value in the indicator ****
Var 986, name rightservo, value 387 // ***** Change de value for your right extreme value for 15 in the
indicator ****
var 988, name runindicator, link subrutine
{
  if &ruddert_o = 0
  {
    I0 = &centerservo
  }
  else
  {
    if &ruddert_o < 0 // LEFT
    {
      I0 = &centerservo - &leftservo
    }
    else // RIGHT
    {
      I0 = &rightservo - &centerservo
    }
  }
  I0 = I0 * &ruddert_o
  I0 = I0 / 16384
  I0 = I0 + &centerservo
}
&grudtrim = I0
}
// ***** OFFSETS *****
Var 5250, Link FSUIPC_OUT, name fcom1, Offset $034E, Length 2 // COM1 Active
Var 5252, Link FSUIPC_OUT, name fcom2, Offset $3118, Length 2 // COM2 Active
Var 5254, Link FSUIPC_OUT, name fnav1, Offset $0350, Length 2 // NAV1 Active
Var 5256, Link FSUIPC_OUT, name fnav2, Offset $0352, Length 2 // NAV2 Active
Var 5258, Link FSUIPC_OUT, name fadf1A, Offset $034c, Length 2 // adf1 Active part1
Var 5260, Link FSUIPC_OUT, name fadf1B, Offset $0356, Length 2 // ADF1 Active part2
Var 5262, Link FSUIPC_OUT, name fadf2A, Offset $02D4, Length 2 // ADF2 Active part1
Var 5264, Link FSUIPC_OUT, name fadf2B, Offset $02D6, Length 2 // ADF2 Active part2
Var 5266, Link FSUIPC_OUT, name toneadf1, Offset $3107, Length 2 // ADF1 TONE
Var 5268, Link FSUIPC_OUT, name toneadf2, Offset $02FB, Length 2 // ADF2 TONE
Var 5270, name FS_XPDER_ID, Link FSUIPC_INOUT, Offset $0354, Length 2
Var 5272, name IVAP_SB_MODE, Link FSUIPC_INOUT, Offset $7B91, Length 1

```

```

Var 5274, name IVAP_SB_IDENT, Link FSUIPC_INOUT, Offset $7B93, Length 1
Var 5276, name XPDER_FAIL, Link FSUIPC_INOUT, Offset $0B6F, Length 1
{
  IF &XPDER_FAIL = 1
  {
    &D_FAIL = -999997
    &LED_FAIL = 1
  }
  ELSE
  {
    &D_FAIL = -999999
    &LED_FAIL = 0
  }
}
Var 5278, name TCAS_MAGENTA, Link FSUIPC_INOUT, Offset $04F4, Length 2
Var 5280, name audio_o, Link FSUIPC_INOUT, Offset $3122, Length 1 // Radio Audio Bits
Var 5282, name audio_i, Link FSUIPC_IN, Offset $3122, Length 1 // Radio Audio Bits
{
  &AU1VH1MI = TESTBIT &audio_i,7
  &AU1VH2MI = TESTBIT &audio_i,6
  &AU1VH1VO = TESTBIT &audio_i,5
  &AU1VH2VO = TESTBIT &audio_i,5
  &AU1NA1VO = TESTBIT &audio_i,4
  &AU1NA2VO = TESTBIT &audio_i,3
  &AU1AD1VO = TESTBIT &audio_i,0
  &AU1MKRVO = TESTBIT &audio_i,2

  &AU2VH1MI = TESTBIT &audio_i,7
  &AU2VH2MI = TESTBIT &audio_i,6
  &AU2VH1VO = TESTBIT &audio_i,5
  &AU2VH2VO = TESTBIT &audio_i,5
  &AU2NA1VO = TESTBIT &audio_i,4
  &AU2NA2VO = TESTBIT &audio_i,3
  &AU2AD1VO = TESTBIT &audio_i,0
  &AU2MKRVO = TESTBIT &audio_i,2
}
Var 5284, name stabtrim_o, Link FSUIPC_INOUT, Offset $050a, Length 2 // stab trim override
Var 5286, name radar_pm, link FSUIPC_INOUT, offset $04f4, length 2 // PMagenta Commands
Var 5288, name ruddert_o, link FSUIPC_INOUT, offset $0c04, length 2, type 1 // Rudder trim
{
  CALL &runindicator
}
Var 5290, name ailt_o, link FSUIPC_INOUT, offset $0c02, length 2, type 1 // Ailerons trim

```