

MEMORIA FINAL

Compromisos y Resultados

Actuaciones Avaladas para la Mejora Docente 2018/2019

Identificación del proyecto	
Código	201800100010293
Título	Puesta a punto e integración del simulador de vuelo SENECA
Responsable	Luis García Barrachina

1. Describa los resultados obtenidos a la luz de los objetivos y compromisos que adquirió en la solicitud de su proyecto¹. Copie en las dos primeras filas de cada tabla el título del objetivo y la descripción que incluyó en el apartado 2 de dicha solicitud e incluya tantas tablas como objetivos contempló.

Objetivo nº 1	Puesta en servicio		
Indicador de seguimiento o evidencias:	<i>0-100% en cada una de las 4 actividades</i>		
Valor numérico máximo que puede alcanzar el indicador (lo estableció en la solicitud del proyecto):	100%	Valor numérico alcanzado por el indicador tras la ejecución del proyecto:	100%
Fecha prevista para la medida del indicador:	<i>Junio 2019</i>	Fecha de medida del indicador:	<i>Abril 2019</i>
Actividades previstas:	<i>1) Testeo de las entradas y salidas de las tarjetas, 2) Comprobación del estado de cada pc, 3) Testeo de las conexiones entre pcs, 4) Arranque y puesta en servicio.</i>		
Actividades realizadas y resultados obtenidos:	<i>Se puso en marcha el simulador. Los 5 ordenadores fueron restaurados. Se sustituyó un hub usb que daba problemas. Se hicieron copias de seguridad de los discos duros.</i>		

¹ La relación incluida en el documento *Actúa* que adjuntó en su solicitud a través de la plataforma de la Oficina Virtual.

Objetivo nº 2	Desarrollo de manuales		
Indicador de seguimiento o evidencias:	0-100% en cada una de las 4 actividades		
Valor numérico máximo que puede alcanzar el indicador (lo estableció en la solicitud del proyecto):	100%	Valor numérico alcanzado por el indicador tras la ejecución del proyecto:	100%
Fecha prevista para la medida del indicador:	Junio 2019	Fecha de medida del indicador:	Septiembre 2019
Actividades previstas:	1) Desarrollo del manual del código 2) Desarrollo del manual de la electrónica 3) Desarrollo del manual del conjunto Rack.		
Actividades realizadas y resultados obtenidos:	Se utilizó el programa doxygen para transcribir el código en c++. Se realizaron manuales operativos de puesta en marcha y técnico para futuras mejoras.		

2. Marque una X bajo las casillas que correspondan en la siguiente tabla. Describa las medidas a las que se comprometió en la solicitud y las que ha llevado a cabo.

Compromiso de compartición / difusión de resultados en el entorno universitario UCA adquirido en la solicitud del proyecto				
1. Sin compromisos	2. Compromiso de impartición de una charla o taller para profesores	3. Adicionalmente fecha y centro donde se impartirá	4. Adicionalmente programa de la presentación	5. Adicionalmente compromiso de retransmisión o grabación para acceso en abierto
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Descripción de las medidas comprometidas				
En la fase de finalización de la actuación avalada, se realizará un seminario abierto, contando con la ayuda de los profesores coordinadores, al resto de profesores de los Grados en Ingeniería Aeroespacial, Grado en Ingeniería Electrónica Industrial e Ingeniería Informática donde se les expondrá:				
<input type="checkbox"/> la situación del simulador, <input type="checkbox"/> sus características <input type="checkbox"/> el trabajo elaborado hasta entonces.				
Descripción de las medidas que se han llevado a cabo				
El día 26 de septiembre se llevó a cabo una sesión de puertas abiertas, la cual fue difundida a los distintos departamentos que trabajan en la Escuela Superior de Ingeniería. Dicha sesión contó con la presencia del becario D. Manuel Tolino Contreras como máximo conocedor de las mejores llevadas a cabo y del responsable del Actúa.				

MEMORIA FINAL

Gestión Económica

Actuaciones Avaladas para la Mejora Docente 2018/2019

Este documento sólo tendrán que completarlo aquellos responsables de proyectos a los que se les concedió financiación económica.

Identificación del proyecto	
Código	201800100010293
Título	Puesta a punto e integración del simulador de vuelo SENECA
Responsable	Luis García Barrachina

Justifique el gasto ejecutado y adjunte las copias de las facturas correspondientes en el campo que se indica en la plataforma de la Oficina Virtual.

Concepto y justificación	Proveedor	Coste con IVA
Material de mantenimiento (bridas, herramientas, almacenamiento, etc.)	GARPLUS DISTRIBUCION Y ASISTENCIA TECNICA SL.	268.20
Hub USB (gasto compartido con orgánica 20DPIM0000)	GARPLUS DISTRIBUCION Y ASISTENCIA TECNICA SL.	13.47
Beca		900
Cuotas seguridad social		145.23
	TOTAL	1326.9



GARPLUS Distribución y Asistencia Técnica S.L.
 Avda. Pta. Del Sur, Edif. Pta. Bahía 1, L-3
 (11408) Jerez de la Frontera (Cádiz)
 C.I.F.- B 11677952
 Tlfn: 956144278 / Whatsapp: 689400072
 WEB: http://www.macromatica.com
 Email comercial@macromatica.com

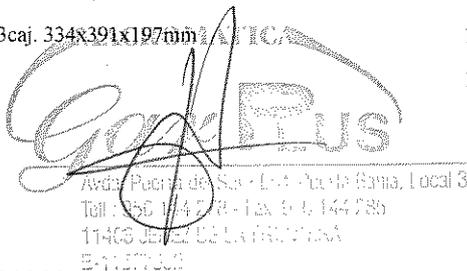
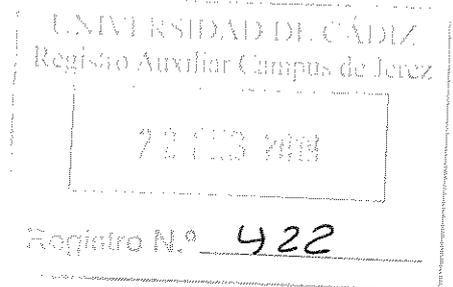
UNIVERSIDAD DE CADIZ
 C/Ancha 16
 11001 Cádiz
 Cádiz

FACTURA

FECHA	FACTURA	CLIENTE	N.I.F.	TELÉFONO	PÁG.
21/02/19	395	43000083	Q1132001G	956015476	1

PRECIOS IVA INCLUIDO

CÓDIGO	DESCRIPCIÓN	UNIDADES	PRECIO	IMPORTE
	ALBARÁN A-21646 FECHA 21/02/19			
	Oficina Contable: (U00500136) Servicios Asuntos Economicos			
	Unidad Tramitadora:(GE0003812)Administración Campus Puerto Real			
	Organo Proponente: CLASIFICACIÓN ORGÁNICA: 20VIRDA049			
	CLASIFICACIÓN ORGÁNICA: 20VIRDA049			
	Organo Gestor: U00500059 DEP. ING. MECÁNICA Y DISEÑO INDUSTRIAL			
	Nº EXPEDIENTE CONTRATO: 2019/0001517			
	-----<			
	Luis Garcia Barrachina <luis.barrachina@uca.es>			
	Dpto Ingenieria Mecánica y Diseño Industrial			
	Escuela Superior de Ingeniería, Campus Puerto Real			
C-U19550000	Aerosol Aire comprimido 400ml 924634	2.00	6.5000	13.00
C-CTK200	Set reparación Startech CTK200 herramientas, 11 piezas c/estuche	1.00	26.5000	26.50
	RS182-127 Kit de herramientas RS Pro, Electrónica	1.00	72.7000	72.70
	RS146-9085 Multímetro digital RS Pro, 10A ac, 600V ac, RS-960	1.00	63.7500	63.75
	RS170-3902 Brida HellaermannTyton, Negro, Nylon 66, 200x4.6mm	1.00	8.6500	8.65
	RS192-5275 Brida HellaermannTyton, Nylon 66, 100x2.3mm	1.00	21.6000	21.60
	RS233-528 Brida RS PRO, Variado, Nylon 66, 100mmx2,5 mm	1.00	8.1500	8.15
	RS838-6545 Cajón almacen NG/TTANSP.PP, 3caj. 334x391x197mm	1.00	53.8500	53.85
C-CAT-UNI	CATALOGO CONSUMIBLES CON PVP	1.00		



LA GARANTIA EN EQUIPOS Y COMPONENTES INFORMATICOS SOLAMENTE INCLUYEN DEFECTOS DE FABRICACION, NUNCA FALLOS DE CONFIGURACION Y/O PROBLEMAS DERIVADOS DE INSTALACIONES, PROGRAMAS, INTERNET,VIRUS... A LOS CUALES SE LES APLICARAN LAS TARIFAS VIGENTES. **TODO SERVICIO DE GARANTIA SERA REALIZADO EN LABORATORIO**
TODO MATERIAL DISPONE DE 1 AÑO DE GARANTIA, SALVO QUE SE ESPECIFIQUE EN EL ARTICULO VARIACION ALGUNA.

Conforme el Cliente	Conforme la Empresa	BASE	%	IVA	TOTAL FRA. (e)
		221.65	21.00	46.55	268.20
		Forma de Pago: TRANSFERENCIA BANCARIA Banco: BANCO BILBAO VIZCAYA ARGENTARIA C/C GARPLUS: BBVA ES14 0182 2108 2101 0151 1963			

GARPLUS DISTRIBUCION Y ASISTENCIA TECNICA SL. Inscrita en el registro mercantil Cádiz Tomo 749 Folio 9 Hoja CA-4129

De acuerdo con lo establecido con la ley Orgánica 15/1999, de 13 de diciembre, de Protección de datos de Crácter Personal (LOPD), le informamos que sus datos estan incorporados en un fichero del que es titular GARPLUS DISTRIBUCION Y ASISTENCIA TECNICA S con la finalidad de prestar a los clientes los servicios profesionales solicitados, realizar gestión administrativa, contable y fiscal, así como envia comunicaciones comerciales sobre nuestros productos y/o servicios. Asimismo, le informamos de la posibilidad de ejercer los derechos de acceso, rectificación, cancelación y oposición de sus datos en el domicilio fiscal de GARPLUS DIST. Y ASIST. TEC. S

Vencimientos: 24/02/19

Entrega a Cuenta:

268.20



GARPLUS DISTRIBUCION Y ASISTENCIA TECNICA SL.

Avda. Pta. Del Sur Edif. Pta. Bahía 1 Local 3

CIF B11677952

UNIVERSIDAD DE CADIZ

C/Ancha 16

11001 Cádiz

Cádiz - ESPAÑA

www.macromatica.com / comercial@macromatica.com / Teléfono 956144278

FECHA	PRESUP.	CLIENTE	N.LF.	TELÉFONO	PÁG.
20/05/19	326	43000083	Q1132001G	956015505	1

PRECIOS IVA INCLUIDO

CÓDIGO	DESCRIPCIÓN	UNIDAD	PRECIO	IMPORTE
--------	-------------	--------	--------	---------

PRESUPUESTO PR326

Oficina Contable: (U00500136) Servicios Asuntos Economicos

Unidad Tramitadora:(GE0003812)Administración Campus Puerto Real

Organo Proponente: CLASIFICACIÓN ORGÁNICA: 20VIRDAÜ49

Organo Gestor: U00500059 DEP. ING. MECÁNICA Y DISEÑO
INDUSTRIA!

Nº EXPEDIENTE CONTRATO:

-----<

Luis Garcia Barrachina <luis.barrachina@uca.es>

Dpto Ingeniería Mecánica y Diseño Industrial

Escuela Superior de Ingeniería

PE/RE0016	L-Link Hub usb 2.0 7 puertos+alimentacion externa LL-UH-707L	1.00	16.3500	16.35
-----------	--	------	---------	-------

BASE	%	IVA	TOTAL (€)
13.51	21.00	2.84	16.35

SIMULADOR DE VUELO PIPER PA34 SENECA III CLASE FNPT II

VOL 1. MANUAL OPERATIVO



UCA

Universidad
de Cádiz



Escuela Superior
de Ingeniería

ÍNDICE

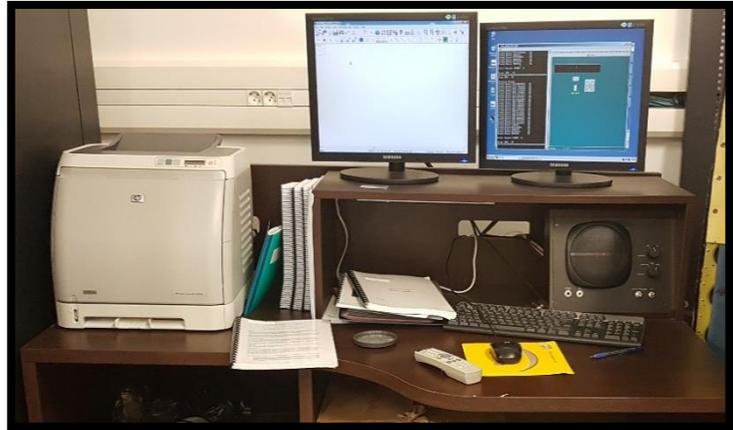
SECCIÓN PRIMERA: DESCRIPCIÓN DEL EQUIPO	3
1. PRESENTACIÓN DEL SIMULADOR	4
2. ARMARIO (RACK) DE ORDENADORES	5
3. MESA DEL INSTRUCTOR	6
4. COCKPIT	7
5. ELECTRÓNICA Y HARDWARE	8
SECCIÓN SEGUNDA: MANUAL DE USUARIO	9
1. ENCENDIDO DE LOS EQUIPOS	10
2. CARGA DE ESCENARIO Y EJECUCIÓN	13
3. PROCEDIMIENTO DE APAGADO DEL EQUIPO	14
SECCIÓN TERCERA: FALLOS DEL SIMULADOR Y SOLUCIONES CONOCIDAS	15
PC Dynamics:.....	16
PC 'IOS'	18
PC 'Hardware'	19
PC 'VISUAL'	21

SECCIÓN PRIMERA: DESCRIPCIÓN DEL EQUIPO

1. PRESENTACIÓN DEL SIMULADOR



RACK DE ORDENADORES



MESA DEL INSTRUCTOR

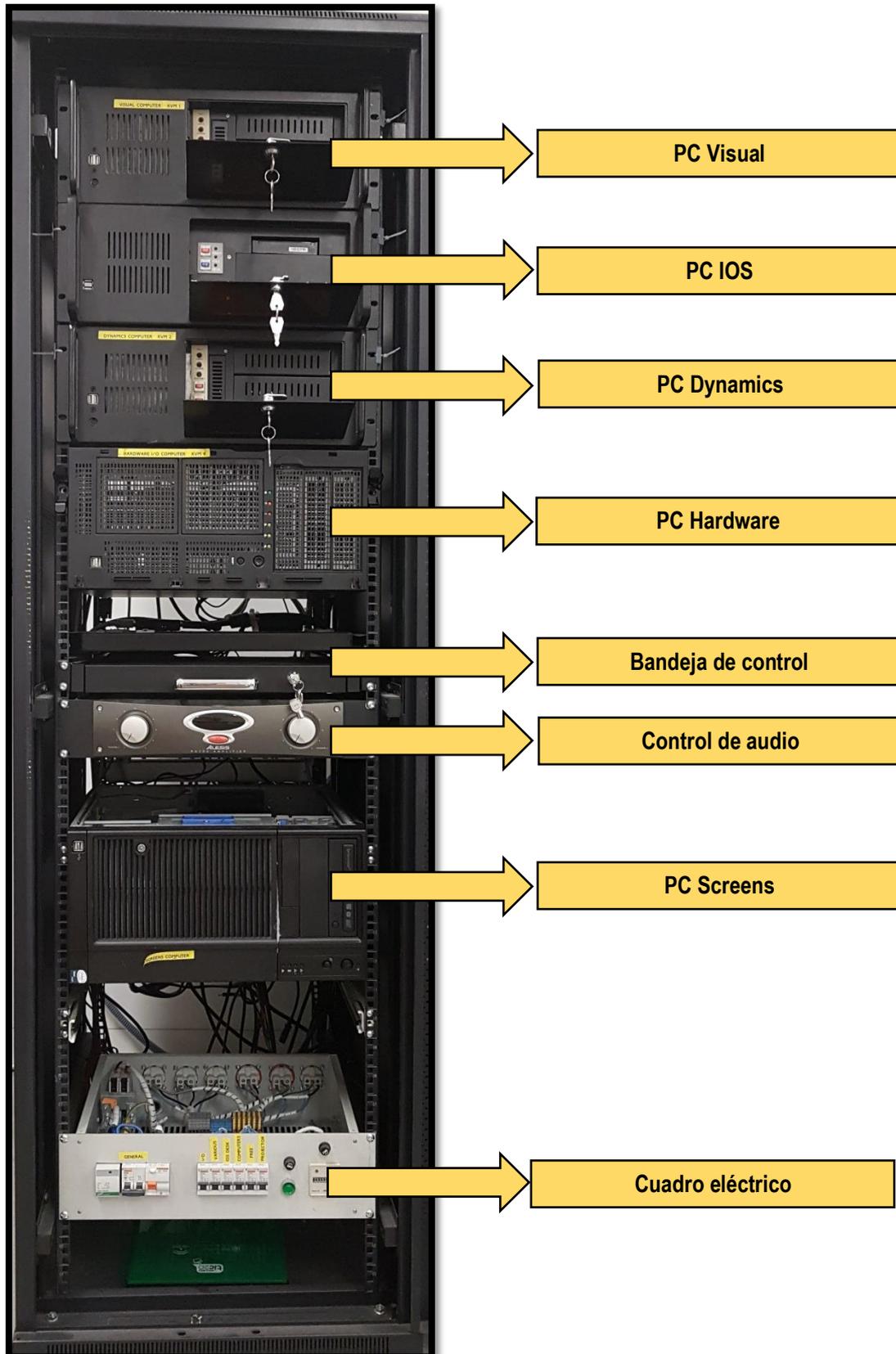


COCKPIT DEL SIMULADOR

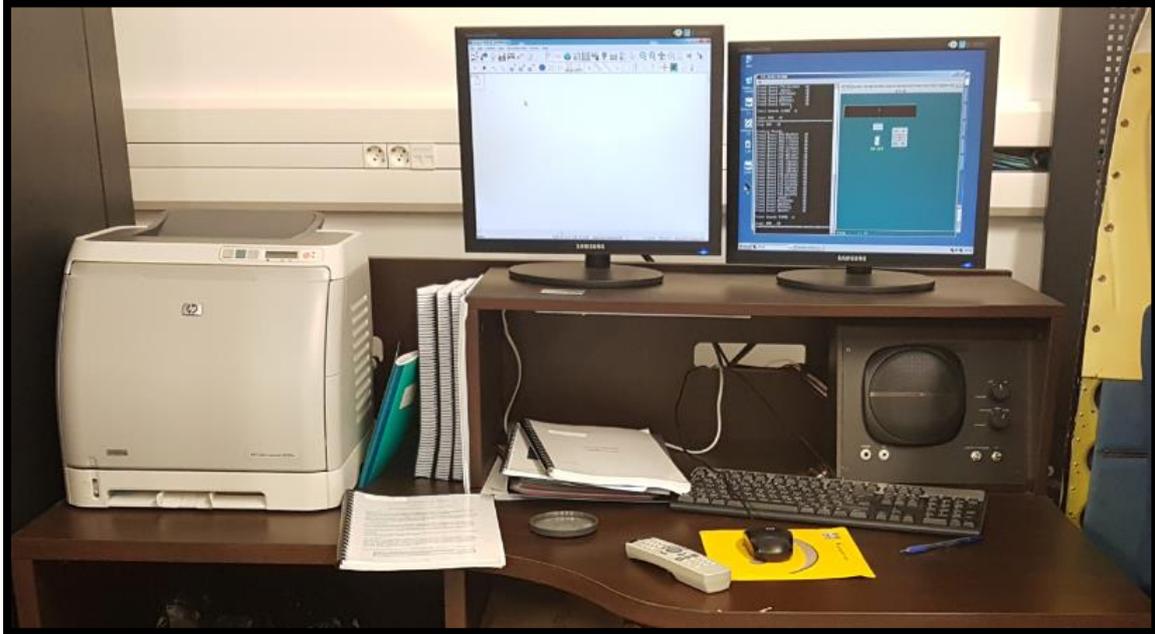


ZONA DE CABLEADO Y ELECTRÓNICA

2. ARMARIO (RACK) DE ORDENADORES

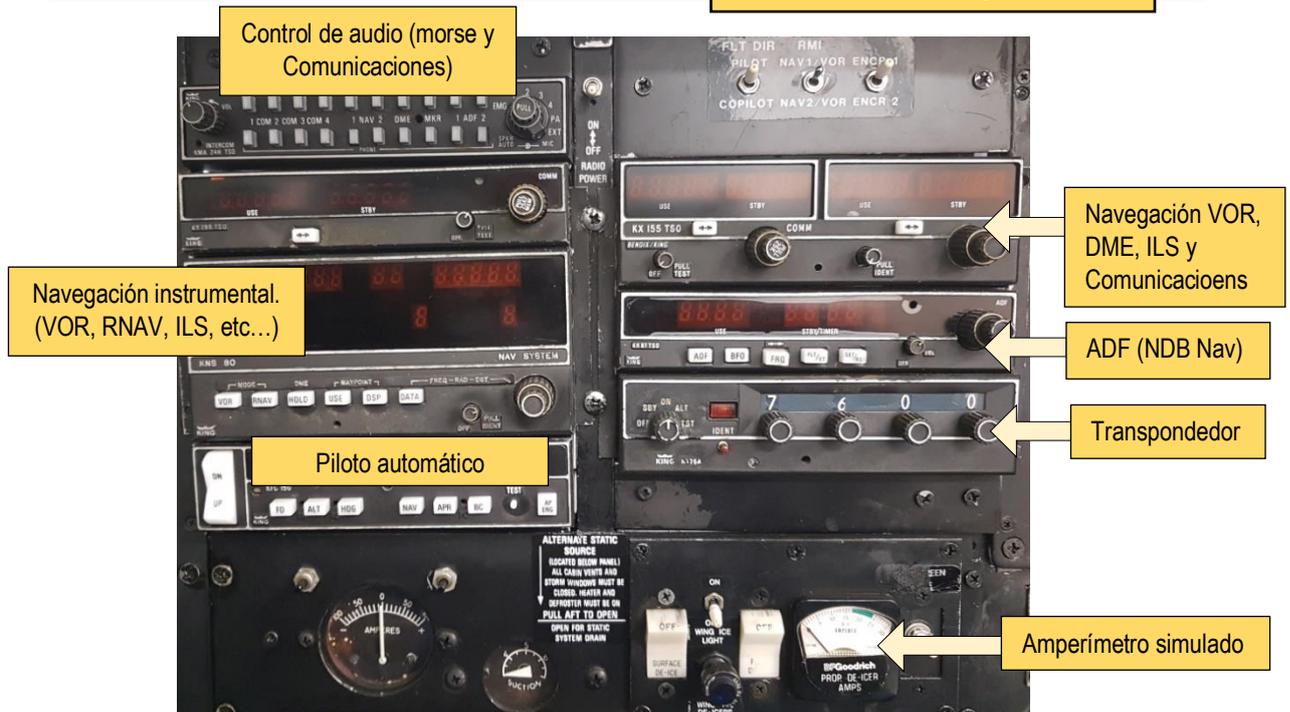


3. MESA DEL INSTRUCTOR

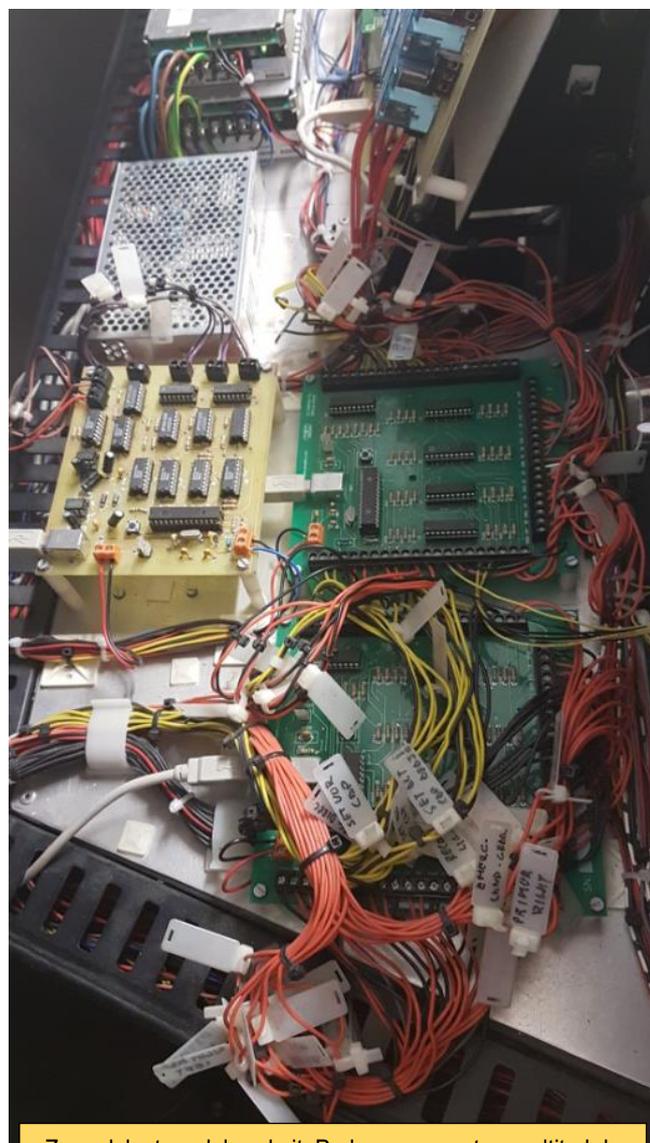
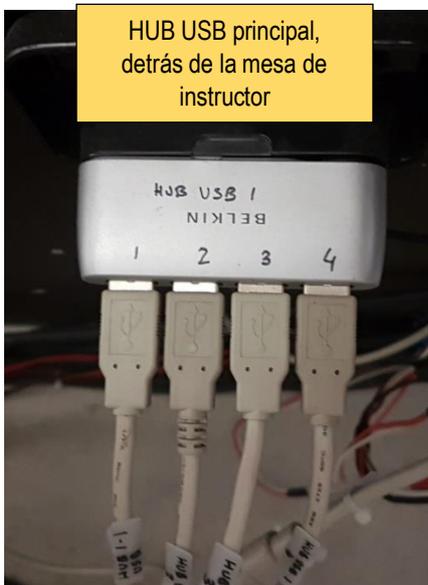


- ✓ En la mesa de instructor se controla el PC IOS. Éste tiene una configuración de pantalla extendida de escritorio. Al iniciarse, el programa IOS ocupará ambas pantallas con dos ventanas independientes.
- ✓ El equipo de audio en esta mesa está en desuso. Su utilidad original es controlar las comunicaciones entre instructor, alumno y supervisor.
- ✓ Se pueden encontrar manuales de utilidad junto a la impresora.
- ✓ El mando del proyector no debe retirarse de esta mesa.

4. COCKPIT



5. ELECTRÓNICA Y HARDWARE

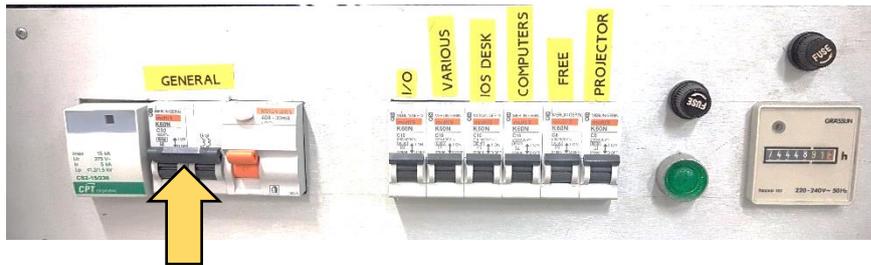


SECCIÓN SEGUNDA: MANUAL DE USUARIO

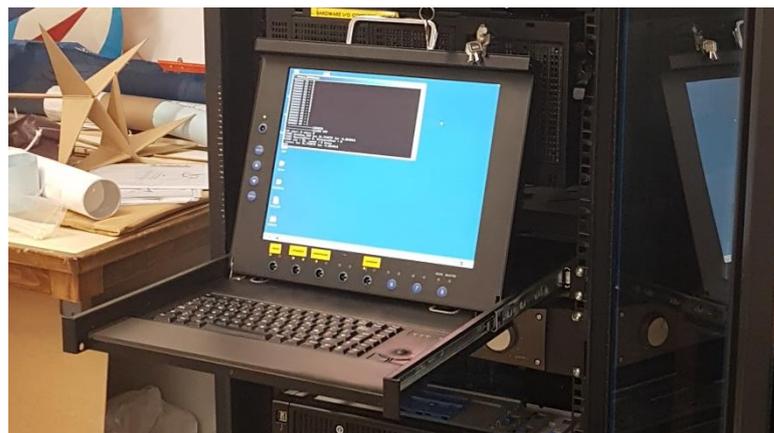
1. ENCENDIDO DE LOS EQUIPOS

FASE 1: PRECARGA

1. Abrir el armario de ordenadores con la llave apropiada.
2. Encender el suministro eléctrico (levantar pestaña del general)

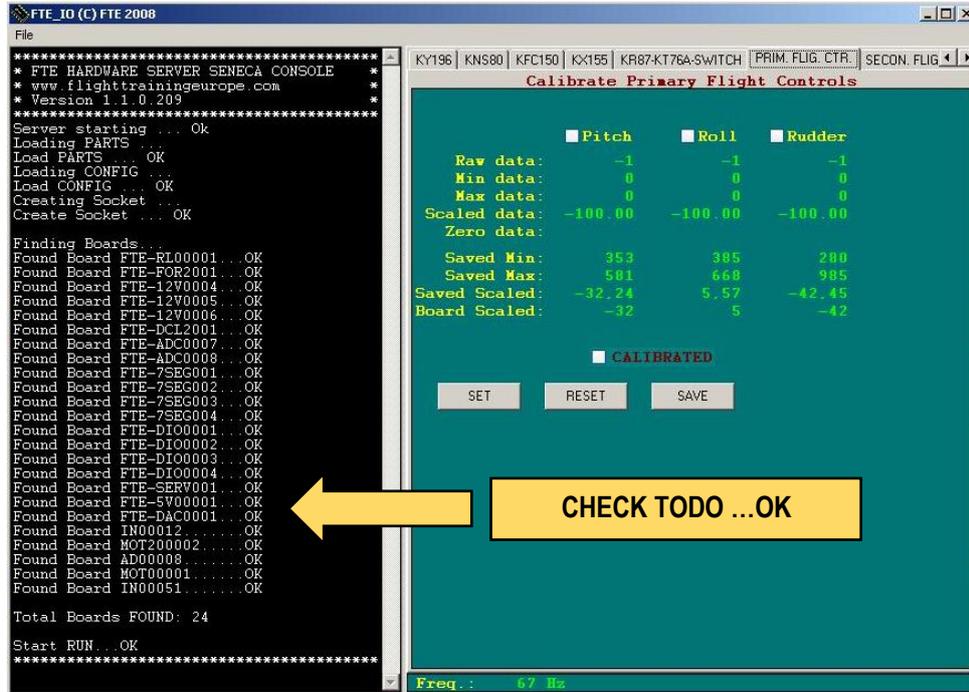


3. Retirar la bandeja de control, tirar del asa y abrir la pantalla. El PC 'Hardware' se inicia por sí solo al encender el suministro eléctrico, para visualizarlo debe estar activada la pantalla 3. **(Realizar inmediatamente el siguiente paso)**

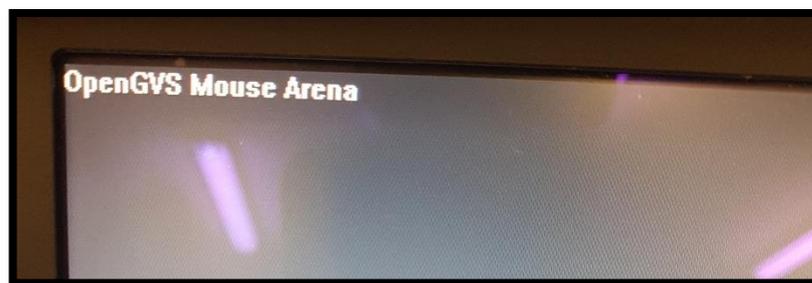


4. Iniciar el PC 'Visual' **(Realizar el siguiente paso al cabo de varios segundos)**
5. Inicial el PC 'Screens'
6. Iniciar el equipo de sonido y ajustar el volumen

- Realizar la comprobación de hardware en 'PC Hardware'. Tras iniciarse el sistema, se abrirán por sí solos los programas FTE_Sound y FTE_IO. En este último, comprobar que todas las placas se han detectado y están conectadas correctamente al sistema (Ver imagen). En caso contrario, NO continuar con la secuencia de encendido y pasar a la **sección tercera** de este manual.

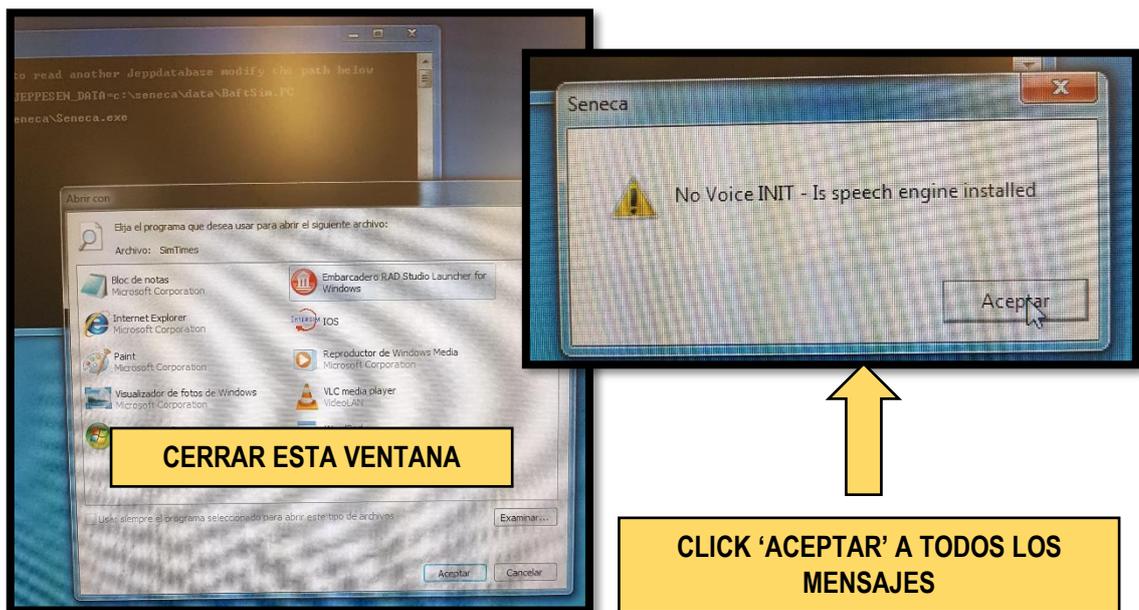


- Activar la pantalla del PC 'Visual' y comprobar que ha encendido correctamente. Debe ejecutarse automáticamente el programa Visual, mostrándose una pantalla gris con el texto 'OpenGVS Mouse Arena' (ver figura)



FASE 2: INICIO DEL PROGRAMA

1. Encender el PC 'Dynamics', activar su pantalla correspondiente y presionar F1 para avanzar el Boot-Up (mientras la placa base del PC no tenga batería esto ocurrirá siempre)
2. Encender el PC 'IOS'. El componente visual de este PC son los monitores de la mesa de instructor. Comprobar que se inicia correctamente, debe cargarse la aplicación Seneca automáticamente.



3. Supervisar el PC 'Dynamics'. No deben aparecer errores en el programa.
4. Retirar la tapadera del proyector.
5. Encender el proyector con el mando a distancia.



2. CARGA DE ESCENARIO Y EJECUCIÓN

Mientras que el programa ejecutándose en el PC 'Dynamics' realiza todos los cálculos y es el 'cerebro' de la simulación, el PC 'IOS' es la interfaz con la que controlaremos la simulación. El programa IOS, permite cargar escenarios, iniciar y parar la simulación, y editar diversos parámetros en tiempo real, entre otras cosas.

1. Cargar un escenario de la carpeta. O bien, utilizar el escenario cargado por defecto.
2. Ajustar los parámetros deseados, más abajo se comentarán las opciones disponibles.
3. Clic en el botón 'Play'. Deben encenderse tanto los instrumentos de cabina, como la visualización. (Es posible que los mandos realicen un movimiento violento provocando un golpe fuerte, por ello se recomienda sujetarlos firmemente.)
4. La simulación está en marcha, para terminarla, clic en el botón stop.
5. Si el avión aparece congelado (inicios desde el aire), apretar el botón situado en la parte trasera del cockpit, detrás de ambos asientos.

Tras una secuencia de encendido correcta y carga de escenario, el simulador debe encontrarse en el siguiente estado:

- Pantallas de instrumentos activas. (Las del copiloto no deben quedarse inmóviles)
- Sonido activado (deben escucharse los sonidos del motor, luces de colisión, y demás sonidos del cockpit)
- Fuerza de retroceso en los mandos. (Los pedales y mandos de control deben estar duros)
- Visualización del escenario en el proyector (pista, aeropuerto, cielo, nubes según configuración... etc.)



3. PROCEDIMIENTO DE APAGADO DEL EQUIPO

1. Cerrar la consola del programa Dynamics
2. Apagar el ordenador Dynamics mediante el menú de Windows.
3. Activar la pantalla del PC Hardware, cerrar ambos programas y apagar el PC mediante el menú de Windows.
4. Activar la pantalla del PC Screens. PRESIONAR: **Doble clic -> ESC -> Alt + F4 -> ENTER.** *(Comprobar después que solo se mantiene encendida la primera luz a la izquierda de la torre)*
5. Activar el PC Visual. Cerrar OpenGVS Arena con **ESC**, y apagar mediante el menú de Windows.
6. Apagar el PC IOS mediante el menú de Windows.
7. Apagar el Proyector (Doble pulsación al botón de encendido) y colocar su tapadera.
8. Apagar el equipo de audio mediante el botón central.
9. Una vez que estén todos los PC's apagados **(por favor, compruébese)**, **apagar el suministro eléctrico mediante el interruptor general en el cuadro eléctrico** (primero a la izquierda, hacia abajo).
10. Cerrar el armario de ordenadores **con llave.**
11. Dejar cerrada la puerta del avión.

SECCIÓN TERCERA: FALLOS DEL SIMULADOR Y SOLUCIONES CONOCIDAS

PC Dynamics:

Programa Dynamics:

1. Mensaje en la consola 'VISUAL COMMS FAIL'

Error:

```
...VISUAL COMMS FAIL
```

Causa:

El programa no consigue establecer conexión con el PC 'Visual'.

Solución:

Comprobar que el PC 'Visual' se ha encendido correctamente, y que se ha iniciado la aplicación tal y como se describe en el apartado de puesta en marcha. Tras ello cerrar la aplicación Dynamics y volverla a iniciar.

Revisar la conexión física entre el PC Visual y el HUB Ethernet, y la conexión entre el PC Dynamics y el HUB Ethernet. Tras ello cerrar la aplicación Dynamics y volverla a iniciar.

Si el problema persiste apagar el simulador y volver a realizar la secuencia de encendido completa.

Si el problema persiste hay un problema en la configuración ethernet. Revisar que la dirección IP de ambos ordenadores está configurada de forma correcta y dichas direcciones coinciden con las establecidas en el archivo HOSTS.

2. Mensaje en la consola 'HARDWARE COMMS FAIL'

Error:

```
...HARDWARE COMMS FAIL
```

Causas:

El programa no consigue establecer conexión con el PC 'Hardware'.

Solución:

Comprobar que el PC 'Hardware' se ha encendido correctamente, y que se ha iniciado la aplicación tal y como se describe en el apartado de puesta en marcha. Tras ello cerrar la aplicación Dynamics y volverla a iniciar.

Revisar la conexión física entre el PC Hardware y el HUB Ethernet, y la conexión entre el PC Dynamics y el HUB Ethernet. Tras ello cerrar la aplicación Dynamics y volverla a iniciar.

Si el problema persiste apagar el simulador y volver a realizar la secuencia de encendido completa.

Si el problema persiste hay un problema en la configuración ethernet. Revisar que la dirección IP de ambos ordenadores está configurada de forma correcta y dichas direcciones coinciden con las establecidas en el archivo HOSTS.

```

c:\ Seneca_DYN404.exe

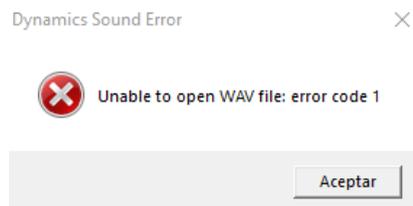
FTE Seneca Dynamics : Version 4.04, Oct 15 2013, 15:54:02
FTE-112 with SCREENS

*****
IOS what: 0 where 0
START heading 0 lat 0.000000 lon 0.000000
START BearingFrom 0, DistanceFrom = 0
START alt = 0, speed = 0 knots
states lat 0.000000 lon 0.000000
INIT
VISUAL
INSTRUCTOR
HARDWARE
EXTERNAL SOUND
SCREENS
HALT
##### QIG Test Mode 0 = Auto, 1 = Manual: <0> #####
=====> HARDWARE COMS FAIL
=====> HARDWARE COMS FAIL
=====> VISUAL COMS FAIL
  
```

3. Error 'Unable to open WAV file'

Error:

'Unable to open WAV file: error code 1'



Causas:

El programa no encuentra los archivos .wav de audio necesarios para su ejecución.

Solución:

Comprobar que en el disco duro local C (o donde esté instalado el sistema operativo) hay una carpeta llamada wav y dentro de la misma hay archivos de audio. Si no es así, consultar el manual técnico para copiar de nuevo dicha carpeta donde corresponde.

PC 'IOS'

1. Errores en el inicio del programa

Error:

Errores varios al inicio del programa.

Causas:

Desconocidas.

Solución:

Ignorarlos.

PC 'Hardware'

1. Fallo de una placa:

Error:

PLACA...OK

PLACA...FAIL

PLACA...OK

PLACA...OK

Causa:

El programa no recibe datos de una de las placas. Windows no la debe reconocer como dispositivo. Se debe tratar de un problema de conexión, puede estar a varios niveles:

- Conexión PC – HUB primario
- Conexión HUB (primario) – HUB de placas
- Conexión HUB de placas – Convertidor (**Interfaz desconocida**)
- Conexión convertidor – placa

Solución:

En primer lugar, realizar un apagado completo y volver a encender, por si se tratase de un fallo aleatorio. Si persiste, clic en el menú, y REFRESH, y comprobar que persiste el mismo error.

Dependiendo de donde esté el fallo de la conexión, se debe desconectar y volver a conectar la placa/placas que no se detectan.

Desconectar la placa que falla, volverla a conectar, y recargar el programa.

ATENCIÓN: Cada tarjeta tiene un rack asignado, NO son intercambiables.

Si no se soluciona el problema, desconectar el cable USB que va desde la placa no detectada al rack USB, sustituir por otro cable y volver a reiniciar. Si persiste, intercambiar el HUB USB afectado por otro y comprobar de nuevo. Si aun así no se detecta la placa, el fallo puede estar o bien a nivel de software (si el programa no se ha modificado, esto no debería ocurrir) o bien a nivel hardware. A nivel hardware la placa no está funcionando correctamente, sería necesario estudiar cómo funciona la placa, que hace que no se detecte y inspeccionar su conexión minuciosamente. Póngase en contacto con un profesional en electrónica y véase los manuales referidos a las placas de hardware para más información.

2. Fallo de múltiples placas:

Error:

PLACA.....FAIL

PLACA.....FAIL

PLACA.....OK

PLACA.....FAIL

Causa:

El programa no recibe datos de varias placas. Windows no las debe reconocer como dispositivos. Se debe tratar de un problema de conexión, puede estar a varios niveles:

- Conexión PC – HUB primario
- Conexión HUB (primario) – HUB de placas
- Conexión HUB de placas – Convertidor (**Interfaz desconocida**)
- Conexión convertidor – placa

Solución:

En primer lugar, realizar un apagado completo y volver a encender, por si se tratase de un fallo aleatorio. Si persiste, clic en el menú, y REFRESH, y comprobar que persiste el mismo error.

Dependiendo de donde esté el fallo de la conexión, se debe desconectar y volver a conectar la placa/placas que no se detectan.

Desconectar las placas que fallan, volverlas a conectar, y recargar el programa.

ATENCIÓN: Cada tarjeta tiene un rack asignado, NO son intercambiables.

Si el problema persiste, en este caso lo más probable es que falle uno de los siguientes componentes:

- HUB de placas
- Cable USB de conexión HUB (Primario) – HUB de placas

Reemplazar cada componente, tras un reinicio completo, individualmente, y comprobar así cual es el que falla. En ambos casos, sustituir el componente por uno nuevo para solucionar el problema. Tenga en cuenta que Windows necesitará un reinicio extra para instalar nuevos controladores para nuevos dispositivos instalados.

3. La pantalla se queda congelada

Error:

La pantalla se queda congelada o parpadeando, imposibilitando el uso.

Causas:

Fallo del adaptador gráfico/tarjeta gráfica.

Solución:

Apagar el PC, apretar la conexión del adaptador gráfico y el cable. Volver a iniciar el PC y mantener una actitud positiva. Si el error persiste, tratar de solucionarlo limpiando la conexión gráfica-placa base con una pistola de aire. También se podría buscar una solución alternativa cambiando el adaptador gráfico y tratar de reemplazar la gráfica.

PC 'VISUAL'

Error:

El PC no termina de arrancar, se queda con un fondo verde y el puntero de ratón únicamente.

Causas:

Desconocidas.

Solución:

Presionar Ctrl + alt + del. Clic en shut down. Esperar a que se apague, y si aparecen carteles para confirmar, aceptarlos. Cuando el ordenador se haya apagado, volverlo a encender (requerirá varias pulsaciones hasta que la luz de encendido se encienda).

SIMULADOR DE VUELO PIPER PA34 SENECA III CLASE FNPT II

VOL 2. MANUAL TÉCNICO



UCA

Universidad
de Cádiz



Escuela Superior
de Ingeniería

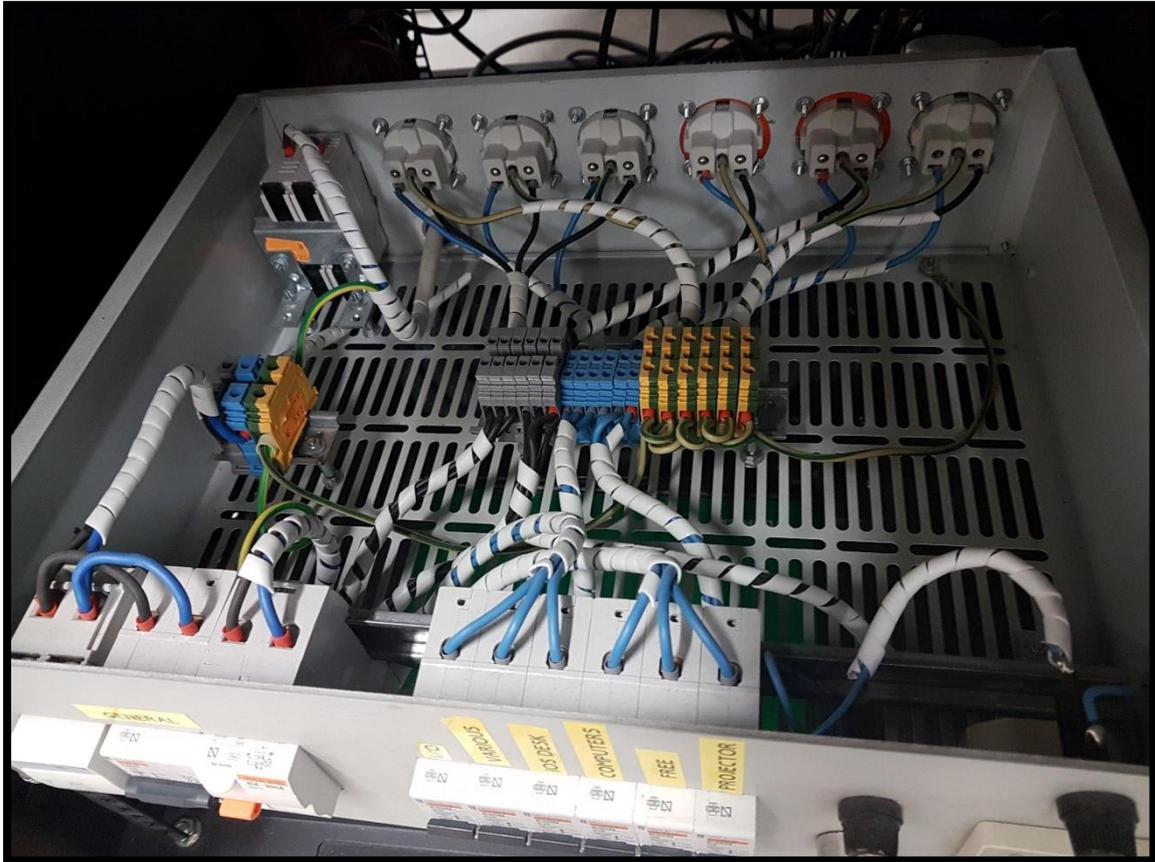
ÍNDICE

SECCIÓN I: COMPONENTES DE HARDWARE.....	3
1. CUADRO ELÉCTRICO (ELEMENTOS BÁSICOS).....	4
1. RACK DE TARJETAS HARDWARE	5
2. COMPONENTES INTERNOS	8
3. MONITORES	8
SECCIÓN II: FUNCIONAMIENTO GENERAL DEL SISTEMA	9
2. ORDENADORES QUE FORMAN LA CONFIGURACIÓN ORIGINAL.....	10
3. COMUNICACIONES ENTRE EQUIPOS	11
4. CONEXIONES DE LAS PLACAS AL PC HARDWARE.....	12
5. ESQUEMA GENERAL DEL SISTEMA.....	13
SECCIÓN III: PC HARDWARE (& SOUND).....	14
1. ESPECIFICACIONES	15
2. SOFTWARE PROPIO	15
3. SISTEMAS OPERATIVOS COMPATIBLES Y EJECUCIÓN EN OTRO PC	15
4. ARCHIVOS Y CARPETAS BACKUP ORIGINALES.....	16
SECCIÓN IV: PC IOS	17
1. ESPECIFICACIONES	18
2. SOFTWARE PROPIO	18
3. SISTEMAS OPERATIVOS COMPATIBLES	19
4. ARCHIVOS Y CARPETAS BACKUP ORIGINALES.....	19
SECCIÓN V: PC DYNAMICS	20
1. ESPECIFICACIONES	21
2. SOFTWARE PROPIO	22
3. SISTEMAS OPERATIVOS COMPATIBLES Y EJECUCIÓN EN OTRO PC	22
4. ARCHIVOS Y CARPETAS BACKUP ORIGINALES.....	23
SECCIÓN VI: PC VISUAL	24
1. ESPECIFICACIONES	25
2. SOFTWARE PROPIO	25
3. SISTEMAS OPERATIVOS COMPATIBLES	26
4. ARCHIVOS Y CARPETAS BACKUP ORIGINALES.....	30
SECCIÓN VII: PC SCREENS.....	31
1. ESPECIFICACIONES	32
2. SOFTWARE PROPIO	32

3.	SISTEMAS OPERATIVOS COMPATIBLES	33
4.	ARCHIVOS Y CARPETAS BACKUP ORIGINALES.....	33
	SECCIÓN VIII: MODIFICACIONES DEL HARDWARE.....	34
1.	ELEMENTOS IRREMPLAZABLES	35
2.	RESTAURACIÓN DE UN DISCO DURO	35
	SECCIÓN XIX: MODIFICACIONES DEL SOFTWARE.....	36
1.	RECOMPILANDO EL PROGRAMA DYNAMICS	37
1.1.	Instalación de Visual Studio 6.0.....	37
1.2.	Versión tradicional Visual Studio 6.0 (VS98)	37
1.3.	Versión compatible con Visual Studio 2008 y en adelante	39
2.	RECOMPILANDO EL PROGRAMA IOS	42
3.	RECOMPILANDO LOS PROGRAMAS INSTRUMENTOS, HARDWARE, SOUND	43
	SECCIÓN XX: ASPECTOS GENERALES DEL CÓDIGO FUENTE	50
1.	CÓDIGO DYNAMICS	51
1.1.	Funciones de transmisión y recepción de datos entre ordenadores	51
1.2.	Datos enviados entre ordenadores:	55
1.3.	Parámetros que definen la aeronave	60
	SECCIÓN XIX: INTRODUCCIÓN DE UN NUEVO PC PARA ESTUDIOS	64
1.	CONEXIÓN DEL NUEVO PC AL EQUIPO	65
	SECCIÓN XX: INTERFAZ DE COMUNICACIÓN CON FLIGHTGEAR.....	71
	Protocolo nativo:.....	72
	Protocolo genérico:.....	72

SECCIÓN I: COMPONENTES DE HARDWARE

1. CUADRO ELÉCTRICO (ELEMENTOS BÁSICOS)



1. RACK DE TARJETAS HARDWARE

Las placas hardware de las que se compone el sistema se listan a continuación. En negrita se muestran las que se sitúan en el rack trasero (el resto se encuentran en la zona delantera del cockpit):

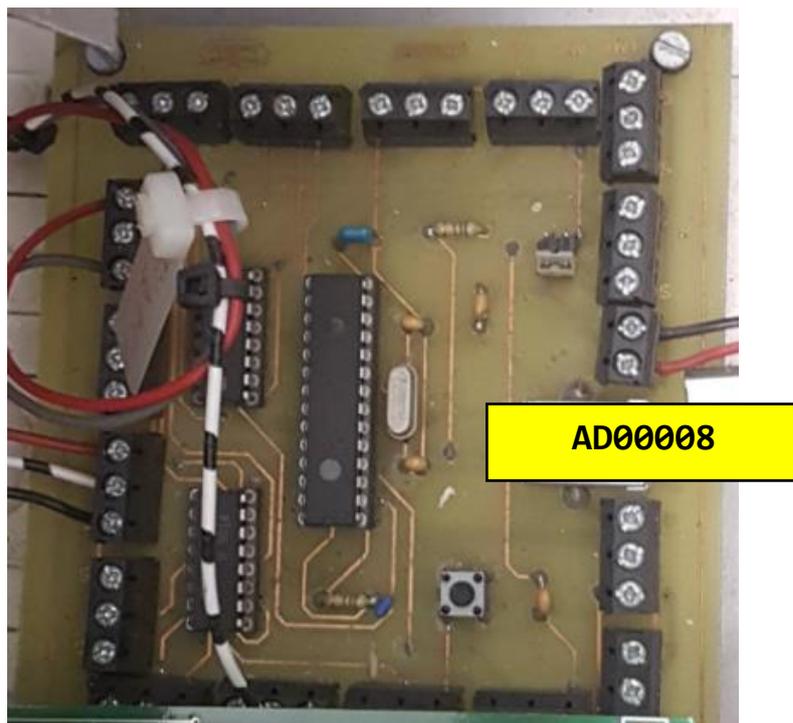
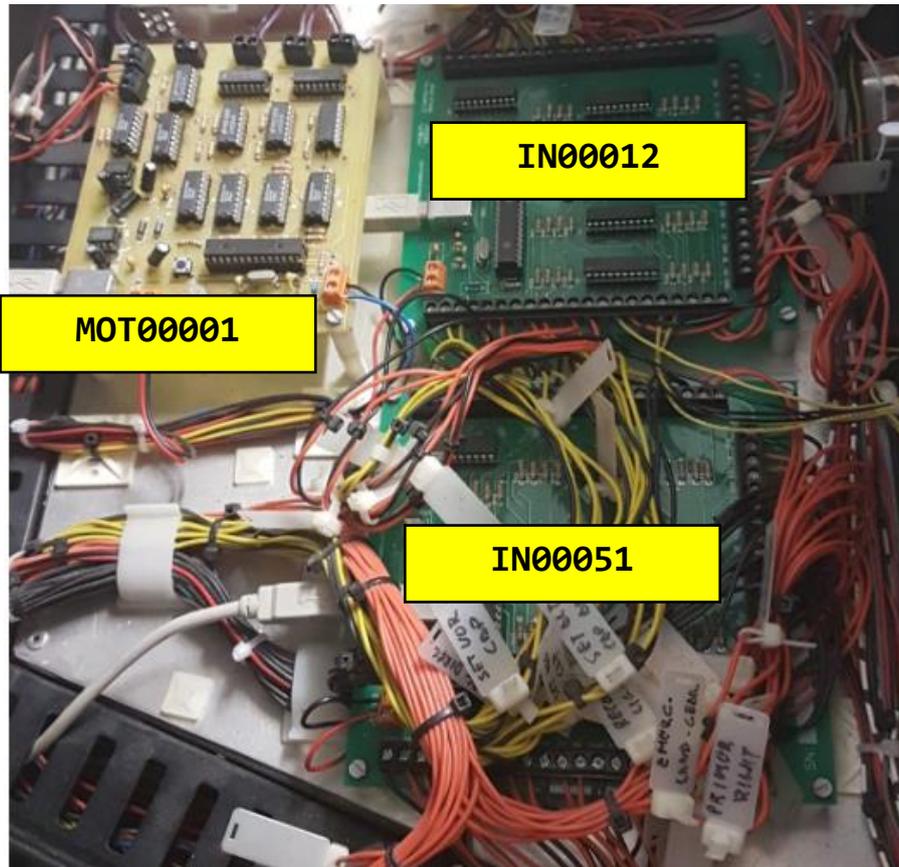
- FTE-RL00001
- **FTE-FOR2001***
- **FTE-12V0004***
- **FTE-12V0005***
- **FTE-12V0006***
- FTE-DCL2001
- **FTE-ADC0007***
- **FTE-ADC0008***
- **FTE-7SEG001***
- **FTE-7SEG002***
- **FTE-7SEG003***
- **FTE-7SEG004***
- **FTE-DIO0001***
- **FTE-DIO0002***
- **FTE-DIO0003***
- FTE-DIO0004
- **FTE-SERV001***
- **FTE-5V00001***
- FTE-DAC0001
- IN00012
- MOT200002
- AD00008
- MOT00001
- IN00051

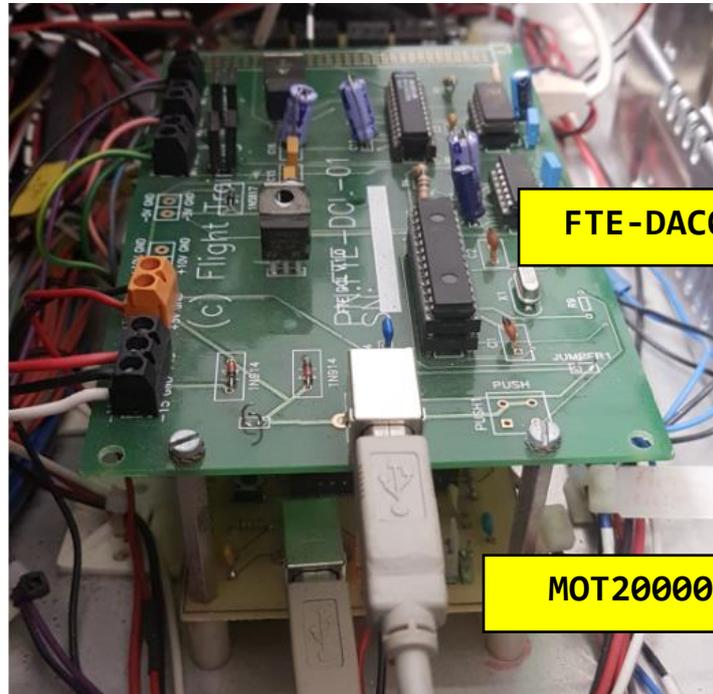


FTE - FOR2001

FTE - 12V0004

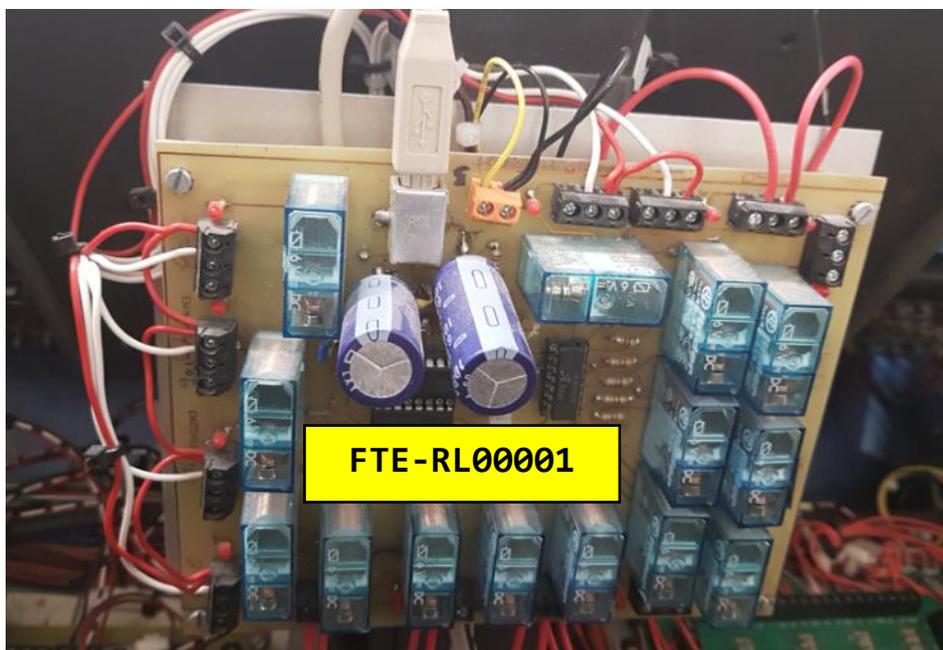
Las dos placas restantes se encuentran en la zona inferior del simulador. (Por comprobar)





FTE-DAC0001

MOT200002



FTE-RL00001

2. COMPONENTES INTERNOS

- Servomotores
- Fuentes de alimentación
- HUB's USB

3. MONITORES

Monitores externos:

Monitores situados sobre la mesa de instructor. Funcionan como pantalla extendida para el PC IOS.

Monitor principal: situado en la bandeja de control con el que se visualizan los PC's dynamics, hardware y Visual.

SECCIÓN II: FUNCIONAMIENTO GENERAL DEL SISTEMA

2. ORDENADORES QUE FORMAN LA CONFIGURACIÓN ORIGINAL

- **PC Dynamics:** Calcula el modelo de vuelo, es el 'cerebro' del simulador. También genera los sonidos básicos durante el vuelo.
- **PC Hardware (& Sound):** Conecta los componentes físicos (Palancas de mando, interruptores, etc) al PC Dynamics. (Es como el sistema nervioso). Además, incluye el programa de sonido que gestiona las comunicaciones con auriculares conectados a la cabina, mesa de instructor, y algunos sonidos de cabina. Permite controlar los equipos de radionavegación desde el programa, ejecutar calibraciones, y diversos tests.
- **PC Visual:** Es el que renderiza el escenario donde se realiza la simulación. También proporciona un feedback a Dynamics del entorno con respecto a la aeronave.
- **PC IOS:** Es la interfaz humana del simulador, desde este PC se controla la simulación (Start, stop, freeze, simulación de fallos, etc), se gestiona el escenario a cargar y se ajustan una serie de parámetros de interés.
- **PC Screens:** Este PC contiene el software que da vida a los indicadores del cockpit. Recibe los datos de los instrumentos calculados por Dynamics, y los renderiza en forma de gráficos animados a representar por las pantallas situadas en el panel de instrumentos. Hay que tener en cuenta, como se explicará en su sección, que este PC solo se puede gestionar de forma remota.

3. COMUNICACIONES ENTRE EQUIPOS

Cada PC está conectado con el resto mediante conexión ethernet. Los cables de conexión conectan cada uno de los ordenadores a un 'hub' ethernet situado en el mismo rack que los PC's (parte trasera).

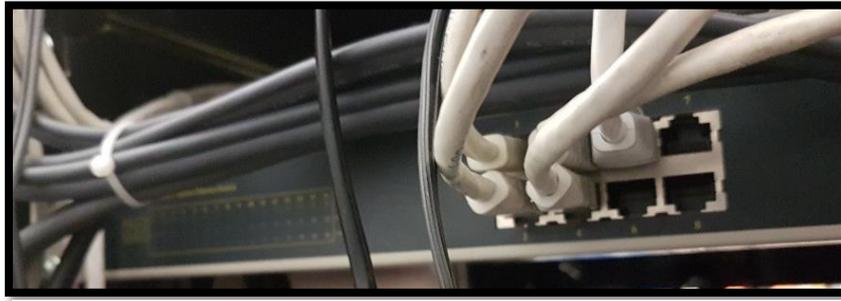


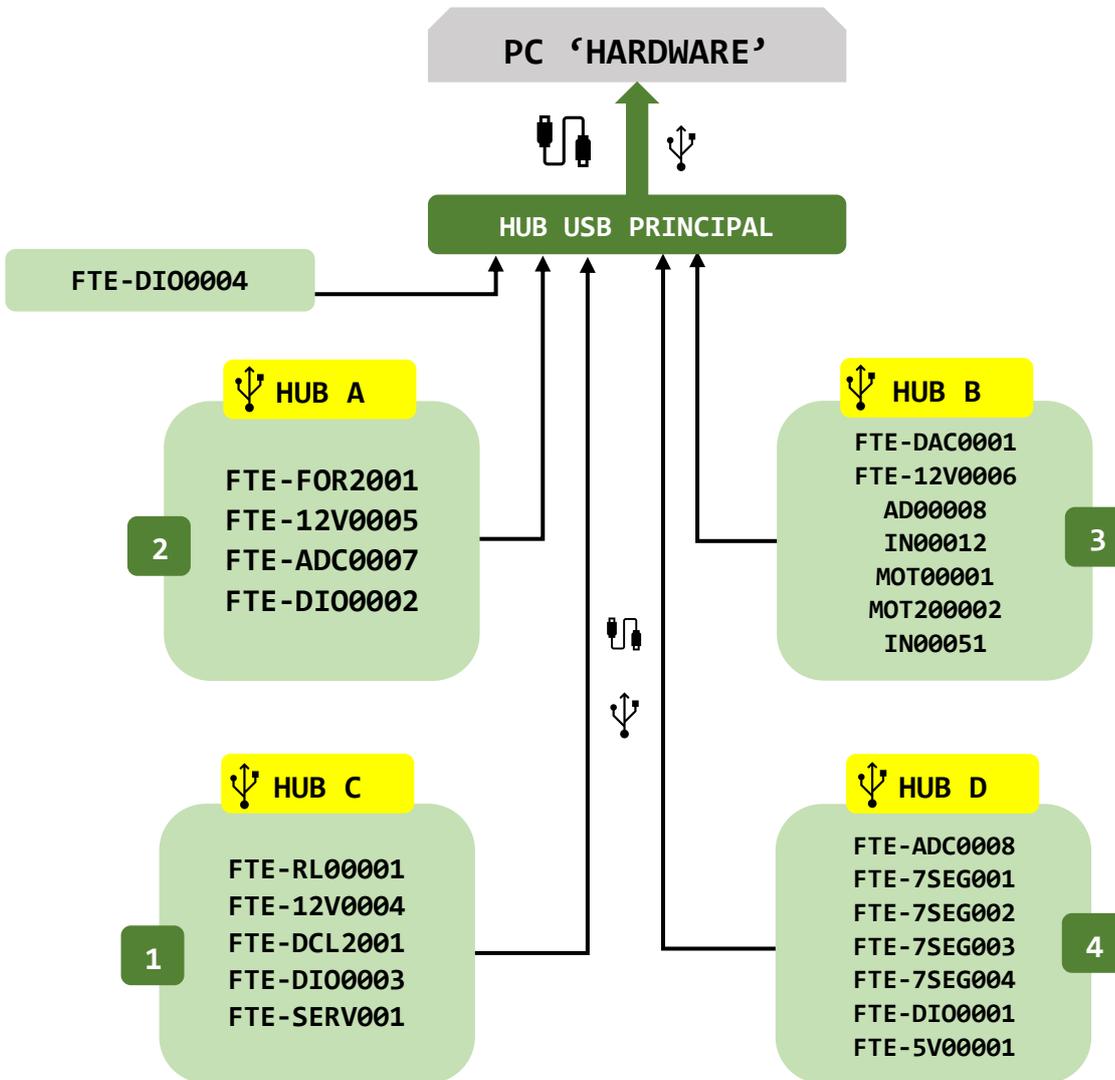
Figura 1. Hub ethernet

Todas las placas hardware se comunican con el PC Hardware mediante conexión USB. Para ello se utilizan hubs USB que agrupan varias de ellas, y éstos hubs a su vez se conectan a un hub principal que se encuentra detrás de la mesa. Este Hub por último se conecta al PC Hardware y de esta forma todas las placas hardware aparecen en ese PC como dispositivos externos individuales.



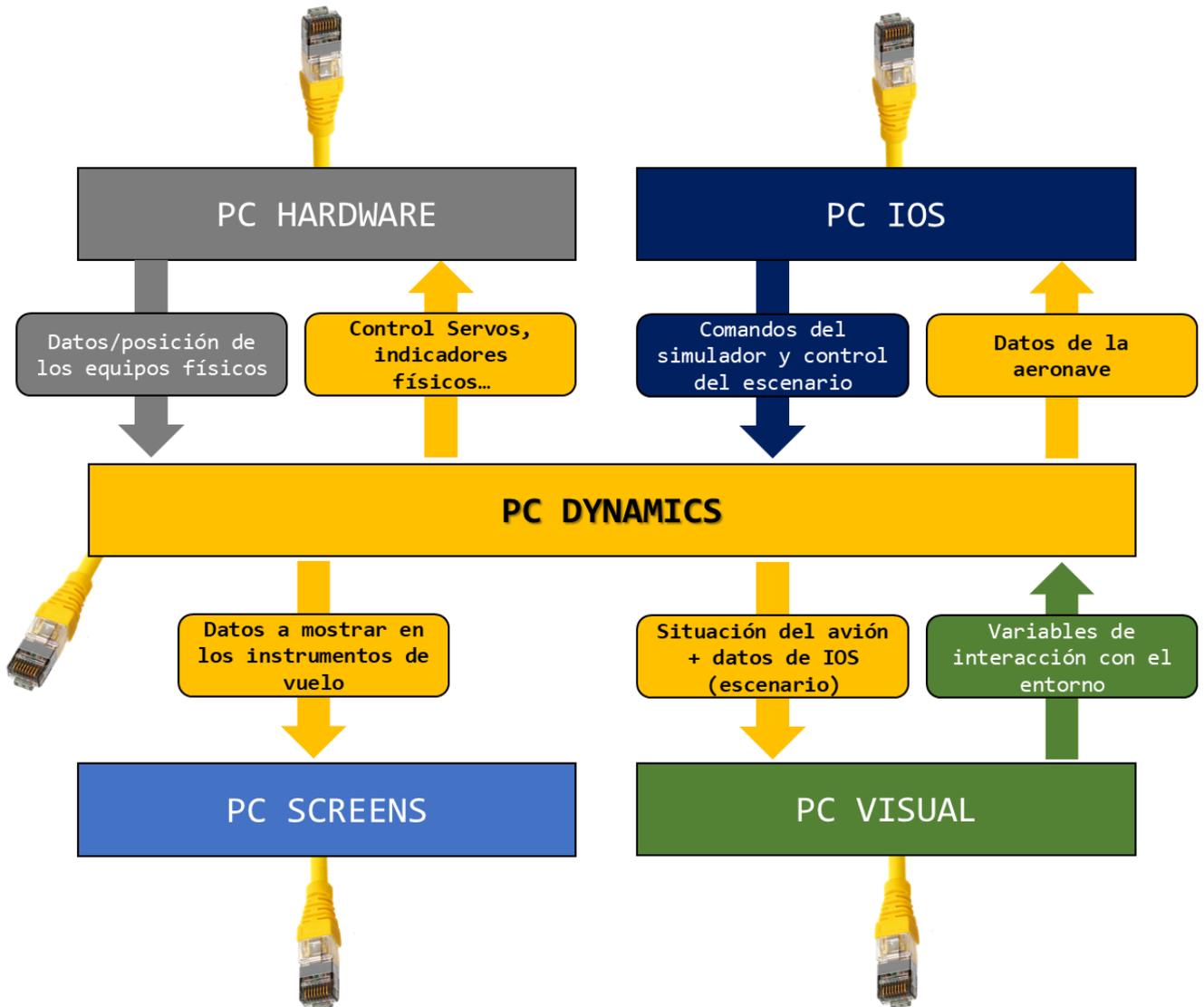
Figura 2. Hub USB principal

4. CONEXIONES DE LAS PLACAS AL PC HARDWARE



Donde a cada HUB llega cada una de las placas nombradas mediante su propio cable USB. Dichos cables están etiquetados con el nombre de la placa a la que corresponde.

5. ESQUEMA GENERAL DEL SISTEMA



Todos los ordenadores se conectan al HUB ethernet. PC Dynamics requiere que esté previamente funcionando el PC Visual y el PC Hardware. Sin embargo, puede funcionar en espera hasta que cargue el PC IOS, ya que al cargarse comienza la simulación congelada, a la espera de las señales de control de IOS. No recibe datos del PC Screens por lo que su funcionamiento no depende de éste.

SECCIÓN III: PC HARDWARE (& SOUND)

1. ESPECIFICACIONES

Sistema operativo:	Windows XP Professional (5.1, Build 2600) Service Pack 3
Nombre del PC:	SENHARD
Procesador:	Intel® Core™ 2 Quad CPU Q6600 2.40GHz (4 CPUs)
Memoria:	2046MB RAM
Disco Duro (Partición 1):	178.5 GBHDD (174.9 GB Libres) NTFS Letra E:
Disco Duro (Partición 2):	60.0 GB HDD (51.9 GB Free) NTFS Letra C:
- Modelo Disco Duro:	ST3250310AS
Audio:	ESI ESP1010e

Adaptador Gráfico 1:

Nombre:	NVIDIA GeForce 8600 GT
Fabricante:	NVIDIA
Memoria:	512.0 MB
Resolución actual:	1024 x 768 (32 bit) (60Hz)

192.128.134.181



2. SOFTWARE PROPIO

En este ordenador se encuentra el software FTE_IO y FTE_Sound. El primer programa se encarga de comunicar todo el hardware con el programa Dynamics. FTE_Sound gestiona la reproducción de los sonidos de cabina.

FTE_IO:

Al ejecutarse, realiza la comprobación de las placas de hardware. Si recibe señal por parte de una placa muestra OK y si no la recibe muestra FAIL, en el listado.

.....

.....

3. SISTEMAS OPERATIVOS COMPATIBLES Y EJECUCIÓN EN OTRO PC

Este programa es, en principio, compatible con cualquier sistema operativo de Windows (XP y en adelante) sin embargo, aún no está resuelta la forma de lograr que opere correctamente en entornos de 64 bits. Esto es debido a que NO existen DRIVERS de dispositivo de las placas hardware del simulador para sistemas de 64 bits. Además, una de las placas da problemas tras sucesivos intentos de conexión mediante un puerto USB 3.0.

Por todo lo anterior, la única forma de reemplazo del PC Hardware es la siguiente:

Se requiere un PC con Windows XP instalado 32 bits. (Comprobado con SP3)

Copiar al PC la carpeta del programa y la carpeta de Drivers (ver el siguiente apartado). Antes de ejecutar el programa, conectar el USB del HUB principal. Windows comenzará a reconocer cada una de las placas como dispositivos desconocidos.

Instalar cada una de las placas como dispositivos de hardware, usando el asistente de Windows según vaya apareciendo. El primer asistente que aparezca se realizará de la siguiente forma:

1. No conectar a internet para buscar controladores (Siguiendo)
2. Buscar los controladores en una ubicación específica (Siguiendo)
3. Seleccionar la carpeta 'Driver' y continuar
4. Comenzará la instalación del controlador. Debe aparecer un cartel de advertencia, clic en Aceptar.

Una vez cerrado el asistente, comenzará otro nuevo asistente para una nueva placa. En esta ocasión y en las restantes:

1. No conectar a internet para buscar controladores (Siguiendo)
2. Buscar automáticamente los controladores en el equipo. (Siguiendo)
3. Clic en aceptar al cartel de advertencia.

El total de asistentes debe ser de 19. Las 5 placas restantes se autoconfiguran como dispositivos HID.

Al finalizar, en el administrador de dispositivos, debe poder visualizarse en la sección de 'Hardware desconocido' en total 19 dispositivos con el mismo nombre '*FTE – BOARDS*'.

Aunque ya se han instalado los controladores de las placas hardware, aún no es posible ejecutar el programa FTE_IO. Para que el programa se comunice con las placas es necesaria una API para Windows llamada libusb. Sin esta librería o protocolo, el programa no detectará las 19 placas que se instalaron como hardware desconocido. Para instalar este último requisito seguir los siguientes pasos:

1. Copiar la carpeta de Libusb en el PC nuevo (contenida en 'Requisitos' dentro del paquete de software de Hardware).
2. Navegar a la carpeta contenida adecuada según el PC, (normalmente será la de x86).
3. Copiar el archivo libusb0.sys a la carpeta [Disco Duro Local]\WINDOWS\System32 \Drivers\
4. Ejecutar el programa **install-filter.exe**. Seguir las instrucciones contenidas en su archivo Léame.
5. Deben crearse 19 filtros, uno para cada placa, sin que aparezcan errores o advertencias.
6. Una vez completado lo anterior, ejecutar FTE_IO, todas las placas deben ser detectadas y en estado OK.

No olvidar configurar la conexión ethernet del PC acorde a lo que requiera el PC Dynamics.

4. ARCHIVOS Y CARPETAS BACKUP ORIGINALES

SECCIÓN IV: PC IOS

1. ESPECIFICACIONES

Sistema operativo:	Windows 7 Professional 64-bit (6.1, Build 7600)
Nombre del PC:	FTE_GEN_RACK
Procesador:	Intel® Core™ i3 CPU 560 @ 3.33GHz (4 CPUs)
Memoria:	4096MB RAM
Disco Duro:	953.8 GB HDD (921.1 GBLibres) NTFS Letra C:
- Modelo Disco Duro:	ST31000524AS ATA Device
Audio 1:	Altavoces (VIA High Definition Audio)
Audio 2:	HD Audio HDMI out (VIA High Definition Audio)
Audio 3:	SPDIF Interface (TX1) (VIA High Definition Audio)

Adaptador Gráfico 1:

Nombre:	NVIDIA GeForce GTX 460 SE
Fabricante:	NVIDIA
Memoria:	2780 MB (Display) 993 MB (Dedicated) 1787 MB (Shared)
Resolución actual:	1280 x 1024 (32 bit) (60Hz)

Adaptador Gráfico 2:

Nombre:	NVIDIA GeForce GTX 460 SE
Fabricante:	NVIDIA
Memoria:	2780 MB (Display) 993 MB (Dedicated) 1787 MB (Shared)
Resolución actual:	1280 x 1024 (32 bit) (60Hz)

192.128.134.183



2. SOFTWARE PROPIO

Este PC ejecuta el software IOS. Esta es la interfaz de inputs y outputs con la que el operador del simulador carga el escenario, modifica parámetros y controla la simulación. En nombre original del ejecutable es Seneca.exe

Además, en este PC se encuentra instalado el entorno de desarrollo (IDE) RAD Embarcadero (Delphi) en una de sus versiones contemporáneas. Este software está licenciado en su instalación en este PC, no se puede trasladar a otro PC y no se dispone de su licencia.

El programa IOS está desarrollado en CPP y desarrollado en VS98. Se dispone del código fuente.

3. SISTEMAS OPERATIVOS COMPATIBLES

El programa IOS es compatible con Windows 7 y Windows 10. No se ha probado en Windows XP pero debería ser compatible teniendo en cuenta que ha sido desarrollado en VS98.

4. ARCHIVOS Y CARPETAS BACKUP ORIGINALES

La carpeta de ejecución del programa se sitúa en el disco duro local, en la carpeta llamada 'SENECA'

La copia de seguridad del disco duro del PC IOS incluye dos particiones.

Se ha tratado de realizar lo siguiente:

Se ha creado una máquina virtual con espacio suficiente para albergar los datos de la partición primaria. A dicha VM se le ha añadido otro disco duro virtual con espacio suficiente para albergar los datos de la partición secundaria. A continuación, se ha creado mediante el software ACRONIS un medio de recuperación (un .iso bootable) que se ha insertado como arranque primario para la máquina virtual. De esta forma, lo que se consigue es revertir la copia de seguridad a los discos duros de la máquina virtual (como si fuese un ordenador físico). Sin embargo, tras completar el proceso y arrancar Windows 7 en la máquina virtual, aparece una pantalla de error crítico y la VM se apaga y reinicia continuamente. Esto puede ser debido a que la instalación de Windows del PC IOS está configurada para funcionar con sus dispositivos (HDD, placa base, etc) y no tolera ejecutarse en un PC distinto. El propósito de este traslado de PC a una máquina virtual no es otro que poder utilizar el software instalado (RAD Embarcadero) pues como hemos dicho el software IOS es compatible con cualquier otro PC.

SECCIÓN V: PC DYNAMICS

1. ESPECIFICACIONES

Sistema operativo:	Windows XP Professional (5.1, Build 2600) Service Pack 3
Nombre del PC:	SENDYNAMIC
Procesador:	Intel® Pentium® 4 CPU 2.80GHz (2 CPUs)
Memoria:	512MB RAM
Disco Duro:	38.2 GB HDD (29.4 GB Libres) NTFS
- Modelo Disco Duro:	ST340014A
Audio:	Realtek AC97 Audio

Adaptador Gráfico 1:

Nombre:	RADEON 9250 (RADEON 9250 AGP)
Fabricante:	ATI Technologies Inc.
Memoria:	256.0 MB
Resolución actual:	1024 x 768 (32 bit) (60Hz)

192.128.134.182

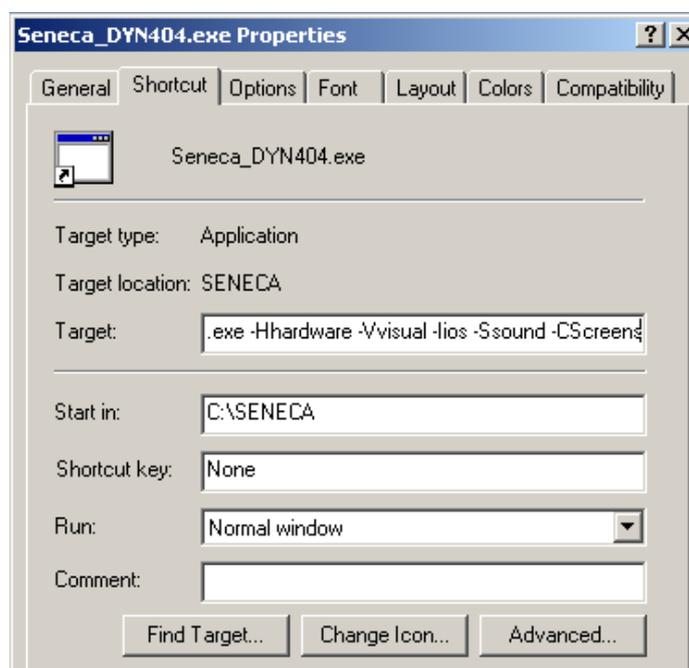


2. SOFTWARE PROPIO

En este PC se ejecuta Dynamics, el software que realiza todos los cálculos. El nombre del ejecutable es seneca_DYN404.exe y para su ejecución es necesario un acceso directo que le pase una serie de comandos según la configuración de ordenadores y del archivo HOSTS.

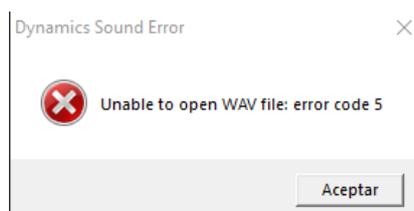
Parámetros necesarios para el programa y configuración originales:

-Hhardware -Vvisual -lios -Ssound -CScreens

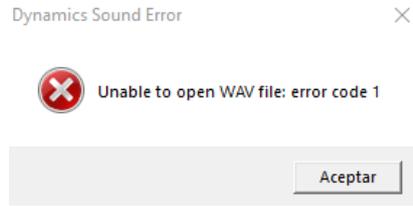


3. SISTEMAS OPERATIVOS COMPATIBLES Y EJECUCIÓN EN OTRO PC

Se ha experimentado el funcionamiento de Dynamics con sistemas operativos más modernos, sin éxito. Al ejecutar el software en Windows 10, con la misma configuración, aparece el error:



Preste atención al número. Si el código de error es 1 el problema está en que la carpeta 'wav' no está copiada a la carpeta raíz del disco duro local donde se encuentra instalado Windows.



Este error es debido a que el programa fue creado para utilizar las herramientas de DirectSound (obsoletas). Estas herramientas/librerías se utilizan para reproducir sonidos en tiempo real a la vez que se ejecuta el programa (esto es sonidos del tren de aterrizaje, de impacto con el terreno, motores, viento, etc). En la versión de Windows XP no aparece este error, mientras que en Windows Vista y versiones posteriores sí, imposibilitando el uso del programa.

Si se modifica el código fuente para eludir los errores derivados de las rutinas de DirectSound, el programa se cerrará con el mensaje 'Este programa dejó de funcionar.'

Ejecución en Windows XP

Se ha comprobado el funcionamiento de Dynamics en Windows XP en una máquina virtual.

Mediante el software de máquinas virtuales Oracle VirtualBox (gratuito), se puede crear un sistema operativo virtual que se ejecute a la vez que el ordenador principal, de forma que podemos tener varias estaciones trabajando al mismo tiempo en un mismo PC y con diferentes versiones de Windows.

Para ejecutar Dynamics en una máquina virtual se requiere:

- Máquina virtual con Windows XP SP3 funcionando.
- Instalación de las carpetas tal y como se describe en el siguiente apartado.
- Configuración de la conexión de red de la máquina virtual tipo NAT,

No hace falta en principio configurar la dirección IP para esta máquina virtual (ni en el caso de PC físico). Pero si surgen problemas desconocidos, tratar de aplicarle dicha configuración.

4. ARCHIVOS Y CARPETAS BACKUP ORIGINALES

Para el funcionamiento y trabajo con el código de forma adecuada, la configuración de carpetas original es la siguiente:

Disco local C:

- Carpeta bae-sims
- Carpeta wav
- Carpeta SENECA

Escritorio:

Acceso directo al ejecutable seneca_DYN404.exe situado en la carpeta SENECA.

SECCIÓN VI: PC VISUAL

1. ESPECIFICACIONES

Sistema operativo:	Microsoft Windows 98 (4.10, Build 1998)
Nombre del PC:	simvi2
Procesador:	Intel® Celeron® CPU 2.60 GHz
Memoria:	240MB RAM
Disco Duro:	9.4 GB HDD (8.7 GB Libres)

Adaptador Gráfico 1:

Nombre:	Intel® 82865G Graphics Controller
Fabricante:	Intel Corporation
Memoria:	14. MB
Resolución actual:	1024 x 768 (32 bit)

Adaptador Gráfico 2:

Nombre:	Quantum3D Obsidian2 3D Graphics Card [Voodoo ² AGP]
Fabricante:	3dfx Interactive
Memoria:	14. MB
Resolución actual:	1024 x 768 (32 bit)

192.128.134.184



2. SOFTWARE PROPIO

El PC visual contiene el programa de visualización con el que cuenta el simulador originalmente. Al arrancar el PC, el software se ejecuta solo. Si se requiere iniciar manualmente por cualquier motivo, ejecutar el acceso directo situado en el escritorio con el icono de MS-DOS.

El programa está basado en una antigua plataforma basada, presumiblemente, en OpenGL, llamada OpenGVS. Esta plataforma/sistema de visualización, era propiedad de los fabricantes de la tarjeta gráfica conectada al proyector para la renderización de la simulación (3dfx Interactive, compañía disuelta en el año 2002) (Adaptador gráfico 2).

Dicho programa, cuyo nombre no se encuentra especificado, es un programa comercial, de cuyo código fuente no se dispone y es sólo compatible con Windows 98, concretamente con el PC en el que se encuentra instalado (PC Visual), debido a las siguientes particularidades:

- El programa está diseñado para funcionar exclusivamente con la tarjeta gráfica instalada (Adaptador Gráfico 2, véase los detalles en el apartado anterior). Si se instala otra gráfica en su lugar, la renderización aparecería con el texto 'DEMO' por todas partes, imposibilitando el uso del simulador.

- Dicho adaptador está obsoleto, no se encuentra en el mercado y resultaría casi imposible encontrar un reemplazo, por lo que cuando el adaptador instalado alcance el final de su vida el PC Visual dejará de resultar útil y deberá ser retirado.

3. SISTEMAS OPERATIVOS COMPATIBLES

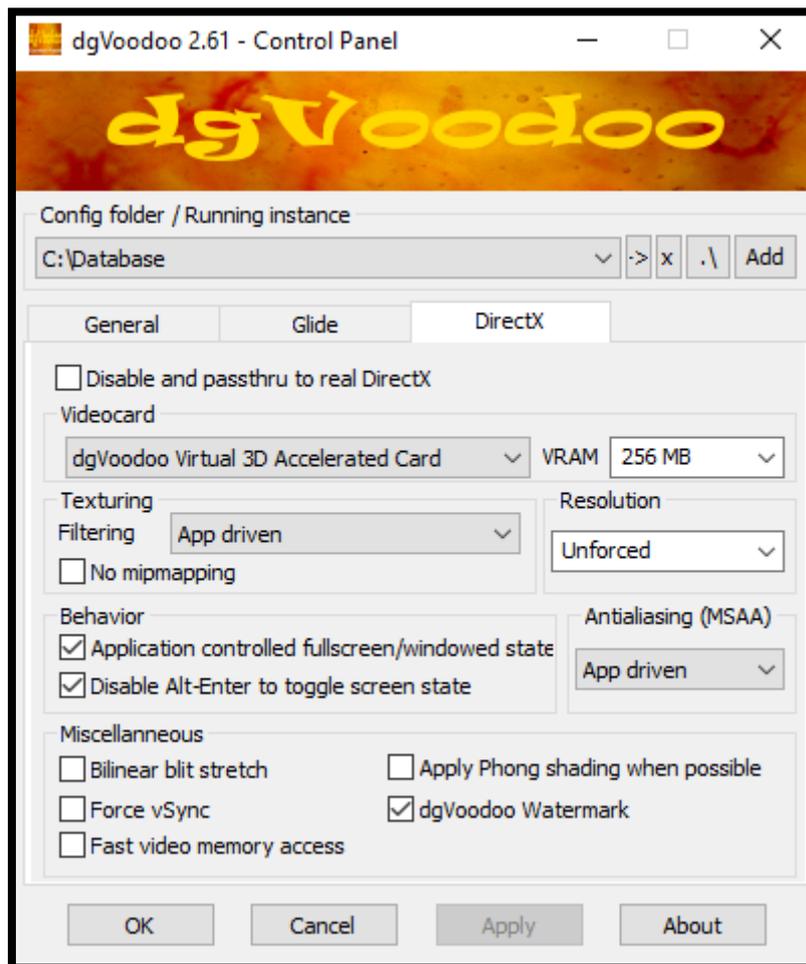
A continuación, se va a explicar como realizar un ‘hack’ del programa Visual para ejecutarlo en cualquier sistema operativo con cualquier tarjeta gráfica.

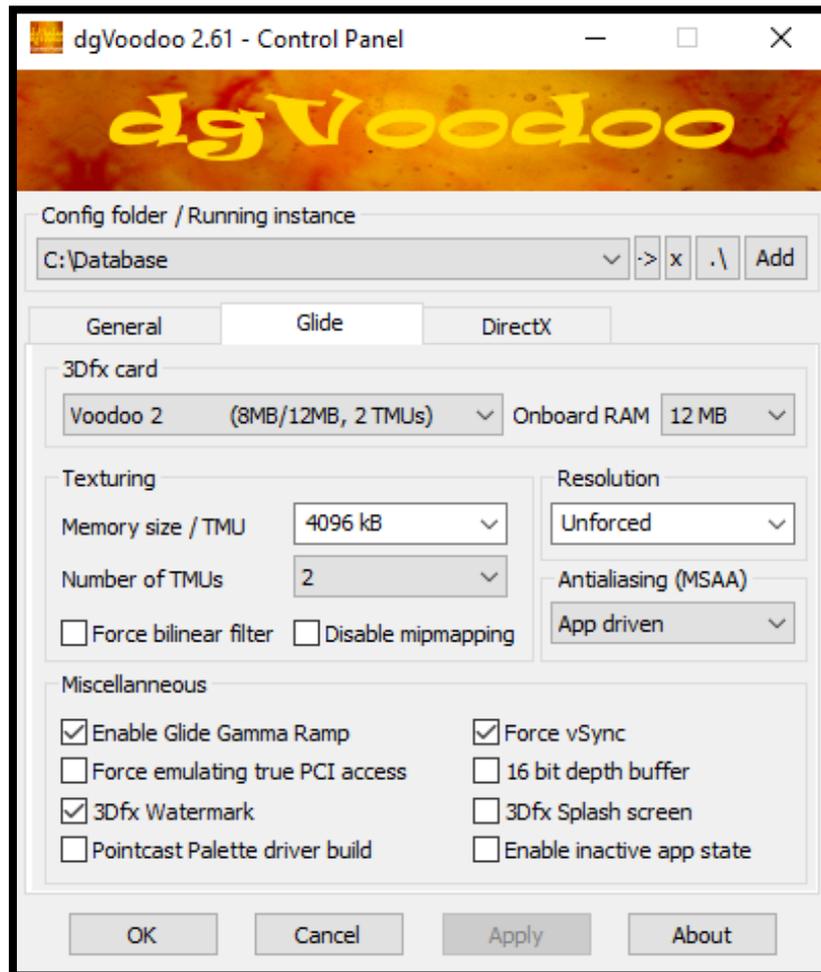
Copiar la carpeta del programa tal y como se describe en el siguiente apartado.

Instalar dgVoodoo2_61, usando la versión x64 (QEmu).

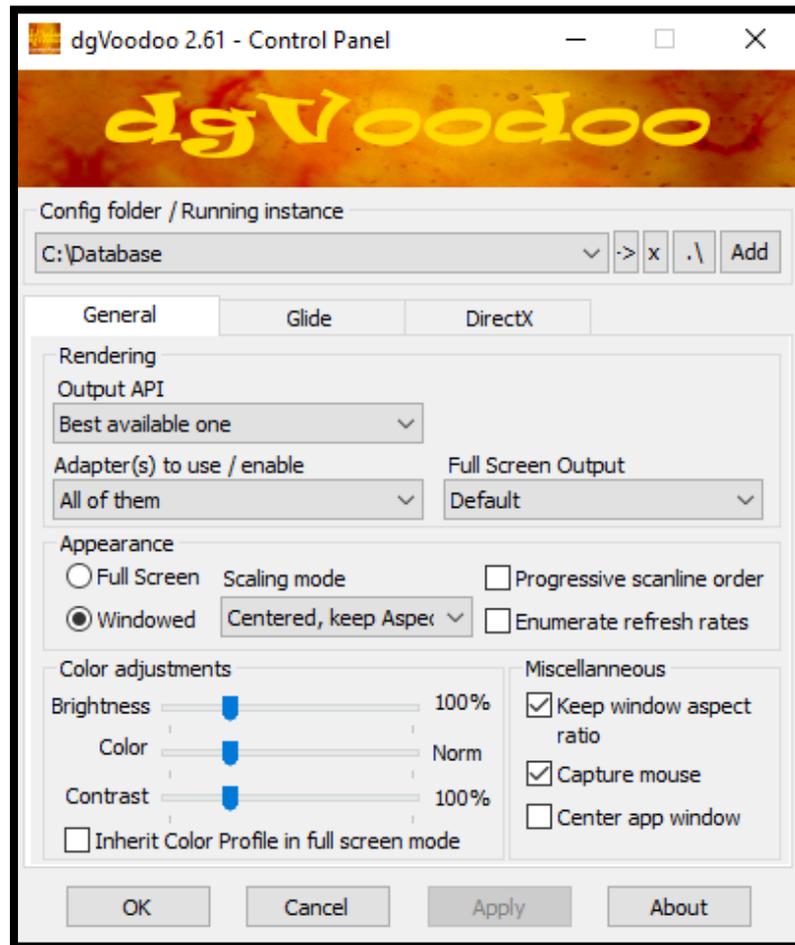
En el pack de software adjunto a este manual se incluirá una versión preparada.

La configuración probada del programa es la que se muestra en las siguientes capturas:





Probar distintas configuraciones si fuese necesario



A continuación, se muestra el archivo de configuración donde se pueden editar parámetros de interés:

Archivo **chanset.cfg**

En este archivo se pueden observar parámetros de configuración tales como posición de la imagen, resolución, iluminación, y FOV (campo de visión). Editando el FOV por ejemplo a 120 grados permitiría investigar la proyección con varios proyectores, o cualquier otra forma para obtener una proyección panorámica.

```
#Config file for openGVS application
#Setup # channels

FRAME_BUFFERS 1

CHANNEL_NO 0
FRAME_BUFFER 0
ANGLE_OFFSET_Y 0 #initially about y axis extend to x and z
POS_OFFSET 0,0,0
LIGHT_SOURCET 0.0,10000.0,10000.0
VIEWPORT -1.0, 1.0, -1.0, 1.0
#RESOLUTION 800,600
RESOLUTION 1920,1080
#RESOLUTION 640,480
FOV 80.0 #default
NEAR_CLIP 0.5
#rem far clip is also unlimited fog distance
FAR_CLIP 50000.0
XRES 1920
#XRES 800
CFG_CHANNEL_END
```

4. ARCHIVOS Y CARPETAS BACKUP ORIGINALES

El Programa OpenGVS se encuentra en:

Mi PC > Disco Duro local C > Database

En el escritorio se pueden encontrar algunos Drivers propios y el acceso directo al programa Visual, el nombre debe contener 'BAE'.

SECCIÓN VII: PC SCREENS

1. ESPECIFICACIONES



ATENCIÓN:

En ningún momento desconectar ningún cable que esté conectado a alguna pantalla, o a los adaptadores Matrox TripleHeadToGo. Esto resultaría en la pérdida de la configuración de pantallas de instrumentos, requiriendo una nueva configuración que se desconoce. Esto supondría la pérdida con difícil o laboriosa recuperación de los instrumentos de cabina. Tampoco conectar un monitor extra.

Este PC se opera remotamente desde el PC Hardware, o desde el PC Dynamics mediante el software de escritorio remoto UltraVNC. Para acceder al mismo, abrir el programa UltraVNC viewer en alguno de los PC's anteriores, pulsar 'Connect' y escribir como contraseña '1' sin comillas. Aparecerá el escritorio remoto.

192.128.134.182



2. SOFTWARE PROPIO

Programa de representación de los instrumentos de cabina.

FTE_Instruments:

Recibe datos de Dynamics, y utiliza las variables en tiempo real para renderizar todos los elementos de los instrumentos simulados. Para ello traza vectores, colorea, y representa símbolos mediante geometrías sencillas, hasta formar la simulación visual del instrumento real correspondiente.

3. SISTEMAS OPERATIVOS COMPATIBLES

- Windows Vista
- Windows 7
- Windows 10 (Supuestamente, no probado)

4. ARCHIVOS Y CARPETAS BACKUP ORIGINALES

Se adjunta todo el archivo de copias de seguridad del simulador.

SECCIÓN VIII: MODIFICACIONES DEL HARDWARE

1. ELEMENTOS IRREMPLAZABLES

No hay nada irremplazable como tal, pero sí que requiera de una laboriosa reprogramación de varios códigos, y multitud de configuraciones:

- Hardware interno (placas, monitores de las screens)
- Hardware específico (componentes reales y simulados de la aeronave)
- Bandeja de control

Por ejemplo, en el caso de que se rompa una placa hardware, se tendría que estudiar su funcionamiento, entradas y salidas, diseñar otra que haga lo mismo y conectarla al PC hardware mediante USB respetando los valores inputs y outputs tal y como se describe en la documentación adjunta a dicha placa.

2. RESTAURACIÓN DE UN DISCO DURO

En el pack de software adjunto al manual se incluyen las copias de seguridad de los discos duros de los PC's Visual, Hardware, Dynamics e IOS. No se incluye la copia de seguridad del PC Screens ya que éste PC funciona con una interfaz distinta a SATA y no existe forma de clonar el disco duro. Haría falta un PC similar para lograr tal operación.

Para restaurar un disco duro o bien montar virtualmente las imágenes de los discos duros en formato .tib es necesario el software Acronis True Image. Una vez instalado el proceso para restaurar las copias de seguridad es bastante intuitivo (consultar la web del programa o la documentación correspondiente para más detalle) y a rasgos generales consiste en:

1. Conectar un disco duro sin formato o en blanco al PC.
2. Arrancar el PC y mediante Acronis, restaurar una imagen de disco al disco duro insertado.
3. Reemplazar el disco duro del PC concreto con el nuevo, el PC no debe notar diferencia alguna.

El proceso ha sido comprobado con un portátil personal y Windows XP.

SECCIÓN XIX: MODIFICACIONES DEL SOFTWARE

1. RECOMPILANDO EL PROGRAMA DYNAMICS

El programa central, dynamics, está escrito en lenguaje C++, desarrollado en su mayor parte durante el año 2000. Para compilarlo y crear el archivo ejecutable que se emplea en la operación del simulador, se utilizó el software comercial Microsoft Visual Studio 6.0 (VS98). Éste es un entorno de desarrollo de software de Microsoft® que aún continúa en el mercado en la actualidad. Sin embargo, como se explicará más adelante, la única versión capaz de recompilar el programa es la versión mencionada. La versión que se ejecuta en el PC Dynamics al momento de la entrega del simulador a la ESI, es una versión compilada con VS98 y modificada por última vez en el año 2013-2014, con el código situado en la carpeta 'bae-sims'

1.1. Instalación de Visual Studio 6.0

En primer lugar, es necesario tener la carpeta de instalación de dicho programa. Dicha carpeta de instalación se puede encontrar en la carpeta 'Documentos' del pc IOS en un archivo comprimido. También puede obtenerse por otros medios, se recomienda buscar en internet en caso de no tener acceso al simulador.

Descomprimir el archivo y copiar la carpeta 'Microsoft Visual Studio' a la ruta:

[Disco Duro Local]:\Archivos de programa\

Importante remarcar que el disco duro local debe ser el que tenga Windows instalado y que la carpeta de archivos de programa NO es Archivos de programa (x86).

A continuación, navegar a la carpeta:

\Microsoft Visual Studio\COMMON\MSDEV98\BIN

Crear un acceso directo al ejecutable MSDEV.EXE situado en dicha carpeta, en el escritorio.

 MSDEV.EXE	18/06/1998 5:17	Aplicación	273 KB
---	-----------------	------------	--------

Renombrar el acceso directo como Visual Studio 6, y ejecutarlo para comprobar que funciona.

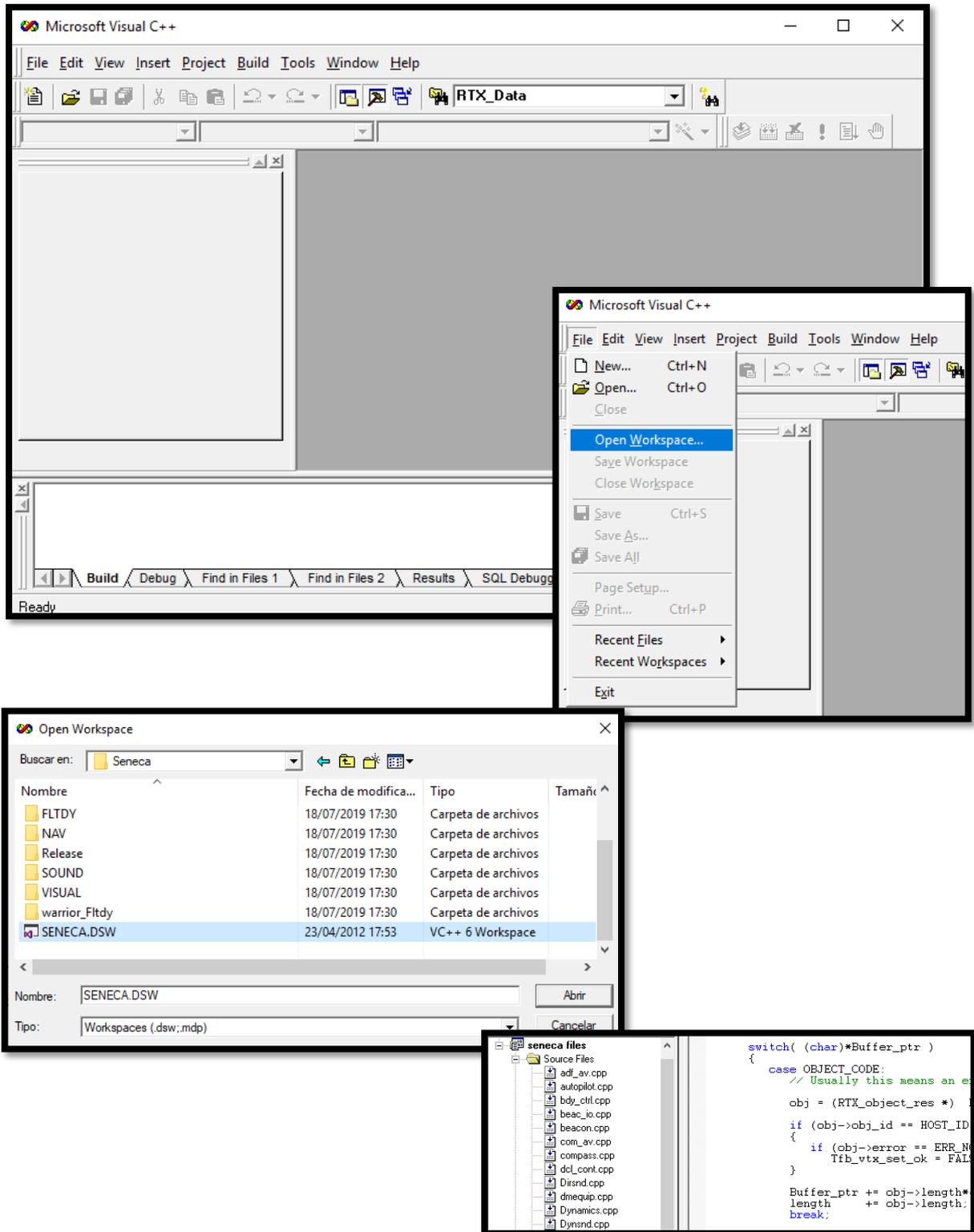
1.2. Versión tradicional Visual Studio 6.0 (VS98)

Esta versión es la última versión utilizada en el simulador durante su vida útil. Emplea el sistema de visualización por defecto (PC Visual), y el software y compilador se encuentran instalados en el PC Dynamics. El código fuente se encuentra en la carpeta 'bae-sims' y el programa en la carpeta SENECA. Es importante mencionar que el programa no funcionará si la carpeta 'wav' de archivos de audio no se encuentra en la ruta preestablecida en el código.

Para la compilación de la versión del código aquí tratada (VS98) son necesarias las librerías Wsock32.lib, Winmm.lib y Dsound.lib. Están incluidas en el entorno de desarrollo Visual Studio 98. Sin embargo, es importante tenerlas en cuenta si se tratase de compilar en versiones más recientes tras realizar las modificaciones pertinentes.

En primer lugar, copiar las carpetas 'bae-sims' y 'SENECA' a la carpeta raíz del disco duro local (en el que está Windows instalado).

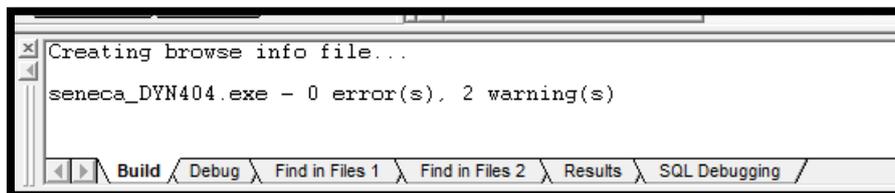
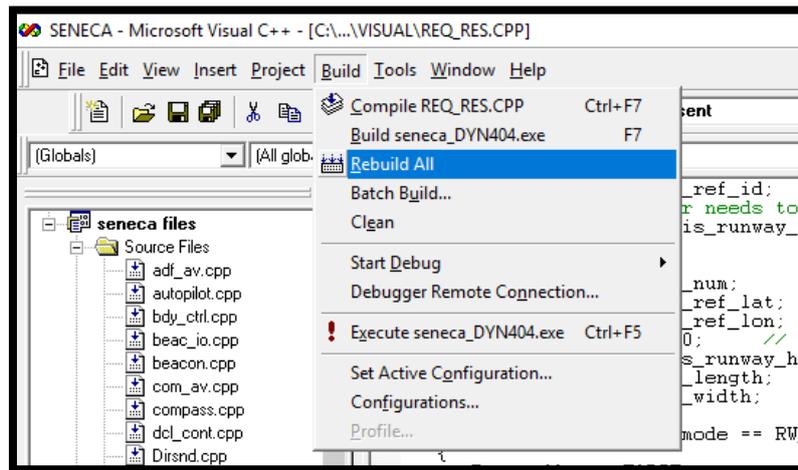
A continuación, abrir Visual Studio 6.0.



Abrir el workspace SENECA.DSW, situado en la carpeta:
\\bae-sims\Seneca

Debe cargarse sin errores y mostrarse el árbol de archivos de código fuente.

Para recompilar el programa: Build -> Rebuild all

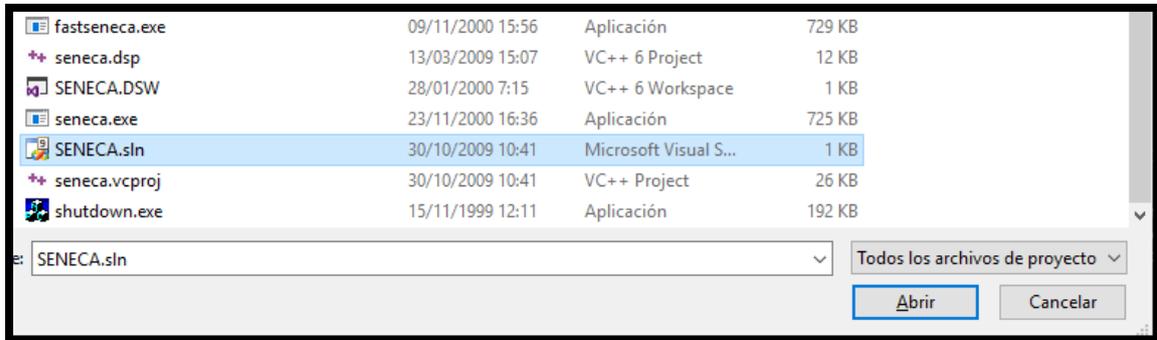


El ejecutable resultante aparecerá en la carpeta C:/SENECA/ como **seneca_DYN404.exe**

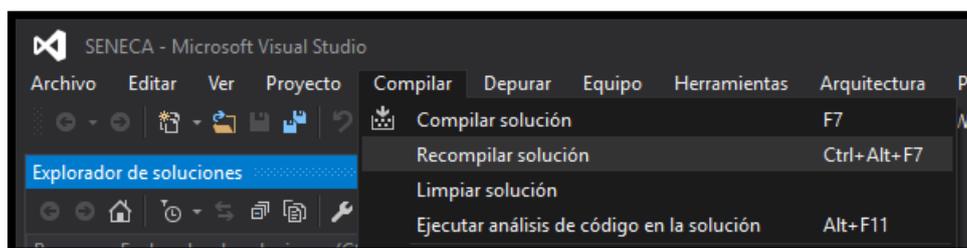
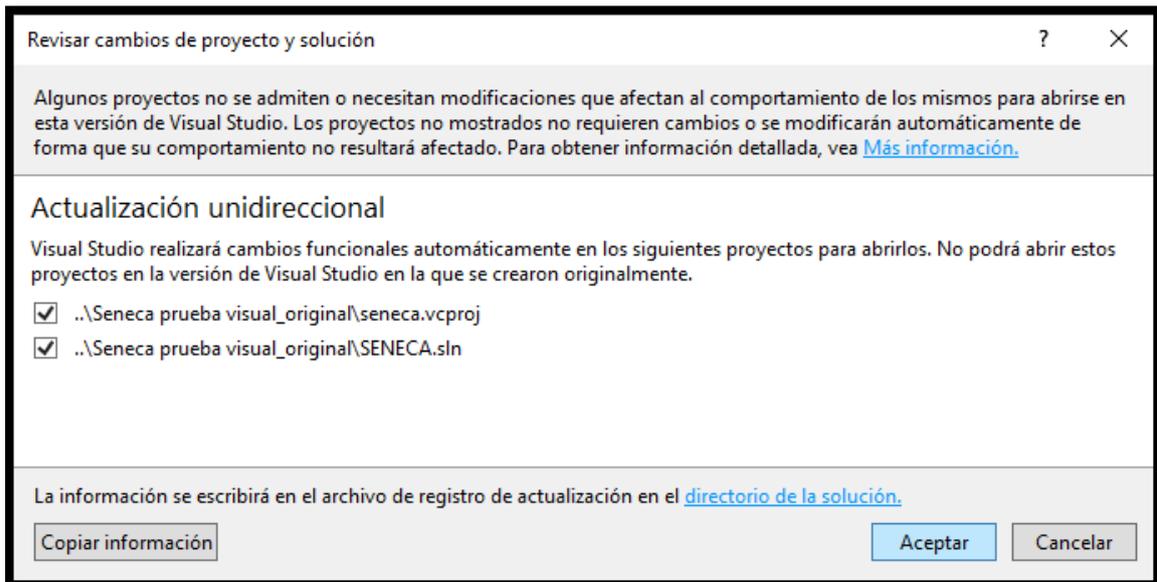
1.3. Versión compatible con Visual Studio 2008 y en adelante

Debido a lo anteriormente explicado en el apartado PC VISUAL, se desarrolló un nuevo código en la versión .NET de Visual Studio. En esta versión lo que se pretende es el desarrollo de un nuevo sistema de visualización, basado en el software libre 'FlightGear'. Para ello se modifica el archivo de transmisión de datos (etherhost.cpp) de forma que ahora los paquetes de datos no salen hacia el PC Visual sino a un nuevo PC que se encargaría de la visualización, con FlightGear instalado. Las instrucciones para compilar la versión del código originalmente encontrada en el PC Dynamics, con Visual Studio 2015 se presenta a continuación. (Los pasos podrían ser similares en otras versiones):

1. Abrir el workspace (solución):



2. Si la versión de Visual Studio es posterior a VS9 (clic aceptar):



Al recompilar la solución, aparecerá el siguiente error fatal que impedirá que se complete:

```
c:\[...]\visual\req_res.cpp(69): fatal error C1083: No se puede abrir el archivo incluir: 'nav\dyn_ios.h': No such file or directory
```

Para solucionarlo, doble clic en dicha línea para que nos lleve a donde está el error. Cambiar la línea que menciona el depurador por la siguiente:

```
“nav\dyn_ios” -> “..\nav\dyn_ios.h”
```

Una vez cambiada la línea, ya debería de poder crearse el ejecutable. (Probar ambas versiones, relase y debug).

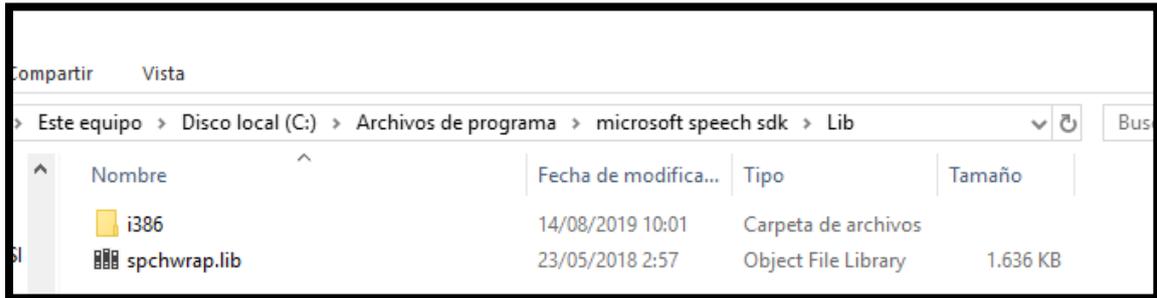
Podría aparecer también el siguiente error (seguir las instrucciones del mismo):

```
LINK : fatal error LNK1207: formato PDB incompatible en 'C:\Users\Manolo\Desktop\Seneca prueba visual\Debug\seneca.pdb'; elimine y recompile
```

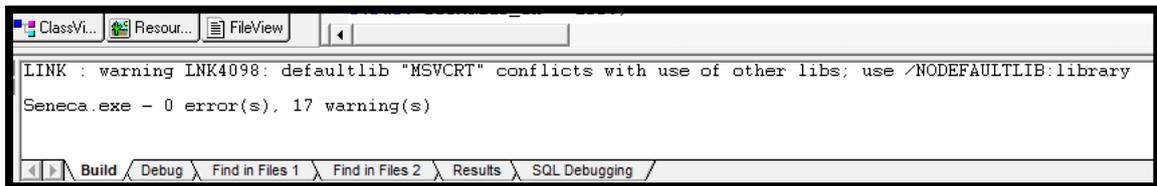
2. RECOMPILANDO EL PROGRAMA IOS

El programa IOS (Seneca.exe) está escrito en C++ y se compila, al igual que Dynamics, con VS98.

Para su compilación es necesario instalar Microsoft Speech SDK (El que está situado en el paquete de software anexo a estos manuales). A continuación, copiar el archivo **spchwrap.lib** en la ruta tal y como se observa en la imagen siguiente:



1. En primer lugar, abrir VS98 (seguir las instrucciones del apartado 1 para dynamics).
2. Abrir el Workspace del código fuente de IOS
3. Realizar las modificaciones necesarias
4. Compilar la solución 'Relase'.



El ejecutable se encontrará en la carpeta Release dentro de la carpeta donde se sitúa el workspace abierto. Nombre: **Seneca.exe** (comprobar fecha de última modificación para saber si estamos mirando en la carpeta correcta).

Se realizó una modificación de este software con éxito en la que se cambió el puerto de comunicaciones con dynamics de 12000 a 12007 para permitir ejecutar de forma simultánea el software Visual con IOS en un mismo portátil externo.

3. RECOMPILANDO LOS PROGRAMAS INSTRUMENTOS, HARDWARE, SOUND

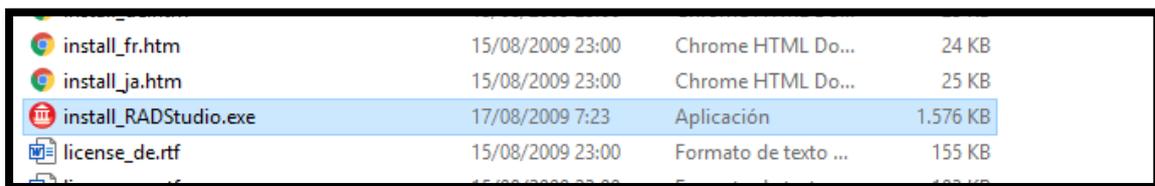
Para volver a recompilar los programas mencionados se va a emplear Embarcadero Delphi 2010 – RAD Studio Architect 2010 (Versión 7.0). Éste es el software instalado en el PC IOS y con el que fueron modificados por última vez los programas mencionados.

El software de instalación se encuentra disponible en el material anexo a este manual. Sin embargo, no se incluye la licencia, por lo que el uso ‘autorizado’ del programa es en el PC IOS donde se encuentra activado con licencia. Es posible, sin embargo, realizar pruebas con un periodo límite de 14 días antes de requerir activación.

Instalación (Windows 10):

En primer lugar, montar la imagen .ISO.

Ejecutar como administrador el instalador



install_fr.htm	15/08/2009 23:00	Chrome HTML Do...	24 KB
install_ja.htm	15/08/2009 23:00	Chrome HTML Do...	25 KB
install_RADStudio.exe	17/08/2009 7:23	Aplicación	1.576 KB
license_de.rtf	15/08/2009 23:00	Formato de texto ...	155 KB

Instalar el software con los parámetros deseados.

Durante la instalación en Windows 10 es posible que el instalador se quede congelado en el proceso de ‘Installing database pack’. En tal caso, si tras aguardar un buen rato no continúa, abrir el administrador de tareas y tratar de cerrar el proceso de instalación asociado (no el instalador general). De esta forma el instalador lo omitirá y proseguirá con la instalación del resto de componentes. Es posible que se deba reintentar la instalación varias veces hasta lograr sobrepasar este error. Si no hay forma, debe tratar de instalarse en Windows 7.

Si se llega a completar la instalación, el siguiente paso es instalar los componentes adicionales para poder trabajar con los programas del simulador que nos conciernen. Para ello copiar las siguientes carpetas a la carpeta raíz del programa:



readme.indy.txt	02/11/2009 23:02	Documento de tex...	2 KB	02/11/2009 23:02
redist_gdiplus.txt	02/11/2009 23:02	Documento de tex...	1 KB	02/11/2009 23:02
thtimer	28/08/2019 18:00	Carpeta de archivos		20/08/2019 16:24
OverbyteIcsV7	20/08/2019 16:24	Carpeta de archivos		20/08/2019 16:24
JVCL339Complete\CL211-Build3536	20/08/2019 16:22	Carpeta de archivos		20/08/2019 16:22
help	20/08/2019 16:22	Carpeta de archivos		20/08/2019 16:22

Pueden encontrarse en el material adjunto al manual. A continuación, procederemos a instalar los 3 componentes de uno en uno.

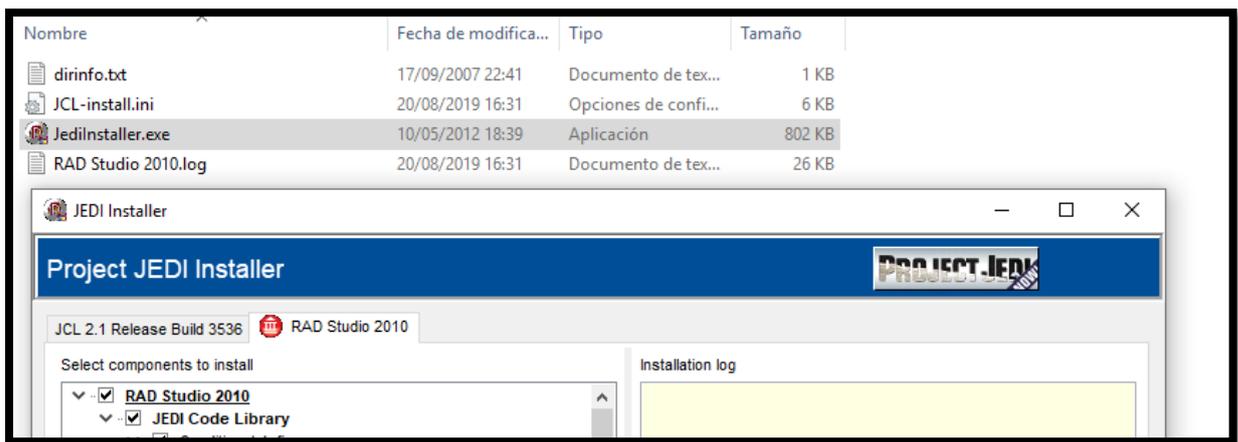
JVCL339CompleteJCL211-Build3536

Dentro de esta carpeta se encuentran dos sub-carpetas. Son dos componentes distintos, han de instalarse por separado.

Nombre	Fecha de modificación	Tipo	Tamaño
jcl	20/08/2019 16:22	Carpeta de archivos	
javcl	20/08/2019 16:23	Carpeta de archivos	

Dentro de ambas carpetas, acceder a la carpeta 'bin' para encontrar los ejecutables de instalación:

JCL:



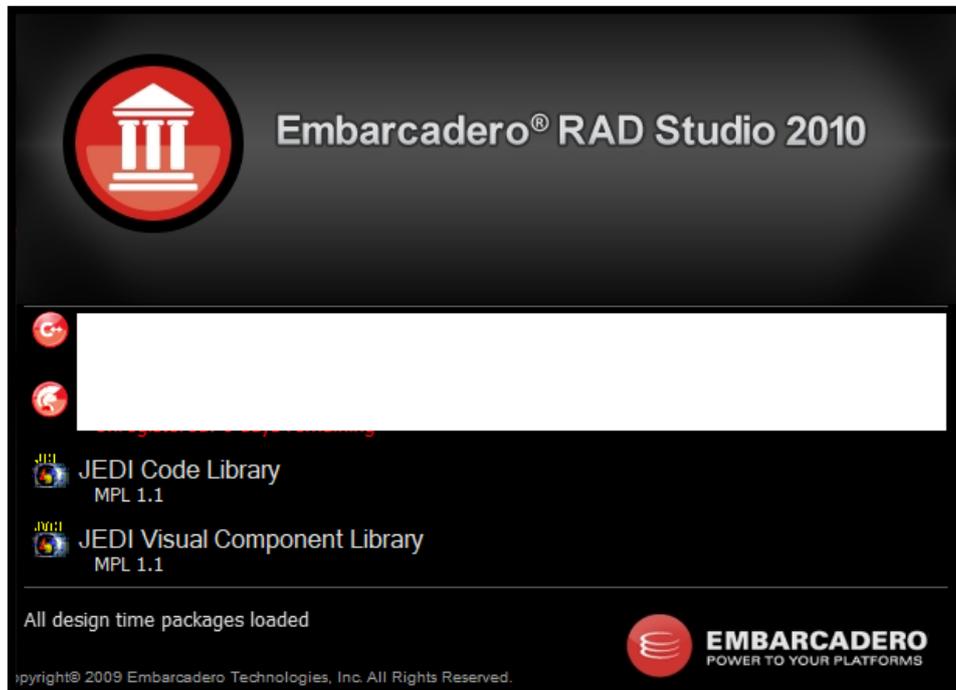
Seguir los pasos de los instaladores. Clic INSTALL comprobando que los parámetros están configurados de forma adecuada.

JVCL:

Nombre	Fecha de modifica...	Tipo	Tamaño
ApplicationHookPlugin	20/08/2019 16:22	Carpeta de archivos	
Data	20/08/2019 16:22	Carpeta de archivos	
DataPlugin	20/08/2019 16:22	Carpeta de archivos	
MDIPlugin	20/08/2019 16:22	Carpeta de archivos	
BaseTest.mdb	04/08/2004 19:48	Microsoft Access ...	196 KB
bin.txt	24/12/2007 16:34	Documento de tex...	0 KB
grass.bmp	09/08/2006 20:00	Archivo BMP	30 KB
human.bmp	09/08/2006 20:00	Archivo BMP	11 KB
JVCLCmdStarter.exe	10/05/2012 18:39	Aplicación	31 KB
JVCLInstall.exe	10/05/2012 18:39	Aplicación	1.179 KB
JVCLInstall.ini	20/08/2019 16:32	Opciones de confi...	1 KB
PhoneList.csv	09/09/2008 21:29	Archivo de valores...	1 KB
TestData.csv	24/12/2007 16:34	Archivo de valores...	58 KB

Seguir los pasos del instalador.

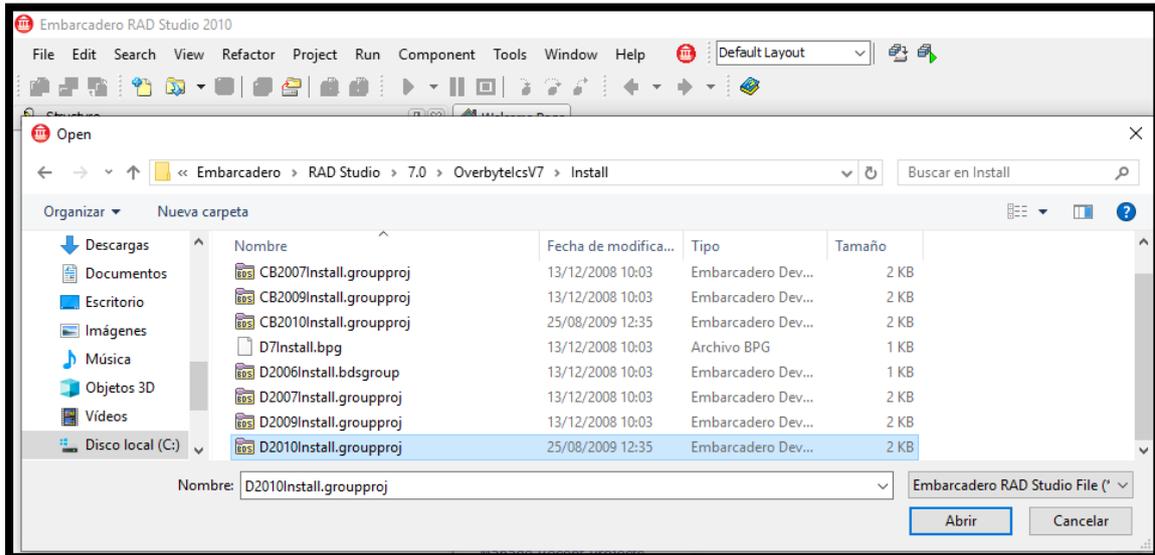
Ejecutar Delphi Embarcadero. En la pantalla de carga inicial deben aparecer ambos complementos y sus respectivos iconos.



Instalación de **OverbytelcsV7**:

Abrir Embarcadero Delphi.

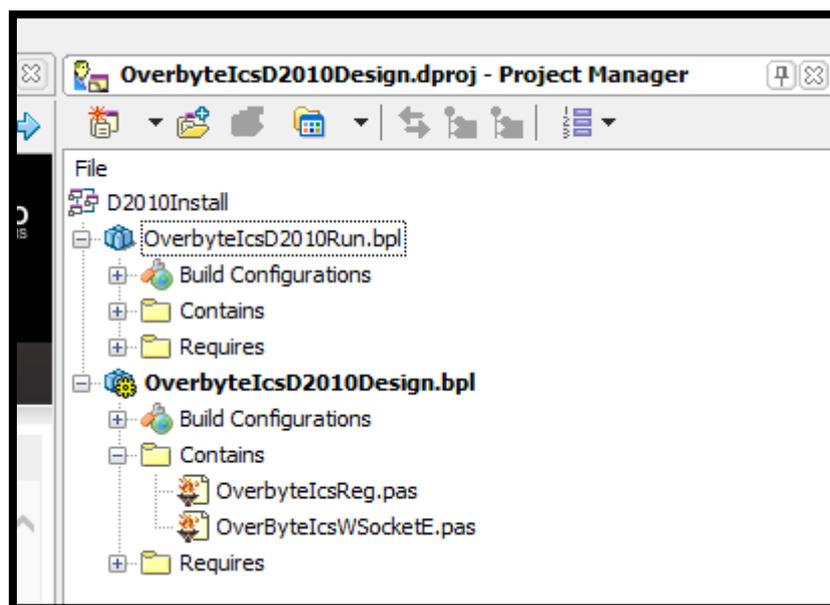
Abri el 'Project Group' que se observa en la imagen:



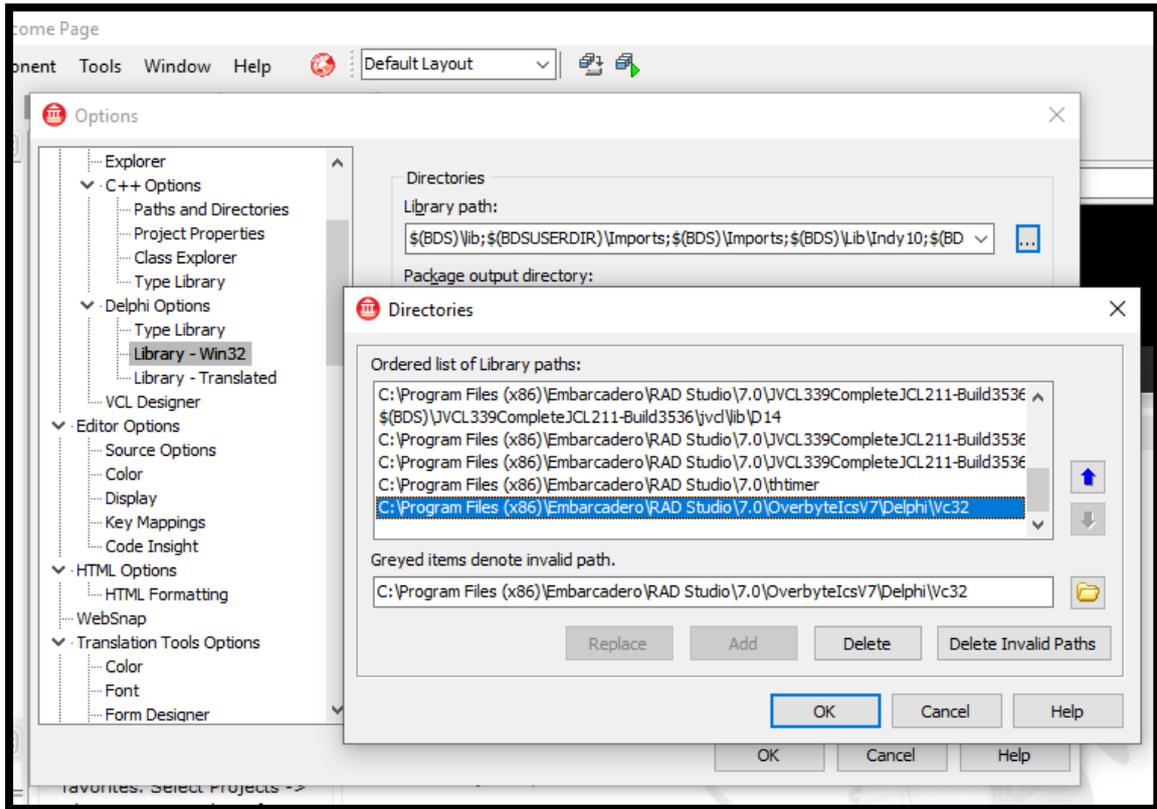
A continuación, observar la columna derecha de la interfaz.

Clic con botón derecho a OverbytelcsD2010Run.bpl y seleccionar 'Build'

Clic con botón derecho a OverbytelcsD2010Design.bpl y seleccionar 'Install'.

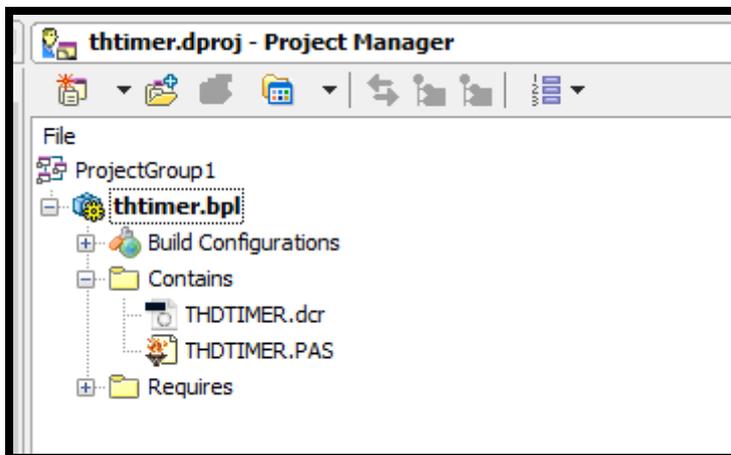


A continuación, añadir la siguiente carpeta a la librería, abrir las ventanas según la siguiente imagen: (Menú tools – options). La carpeta a añadir es OverbytesV7\Delphi\Vc32.

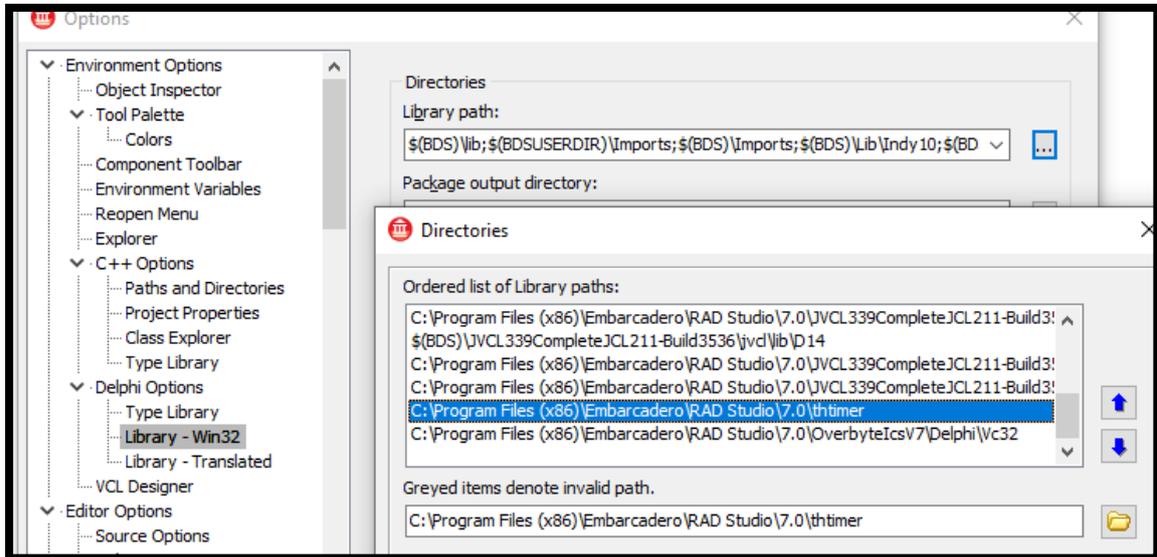


Instalación de thtimer:

Abrir el pack en la carpeta 'thtimer' y realizar un install del package como se realizó antes con overbytes. Botón derecho y 'Install'.

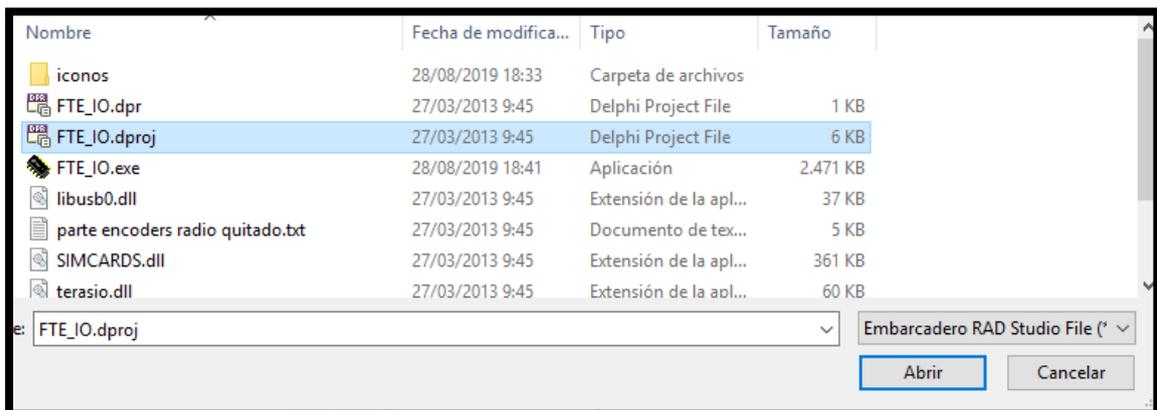


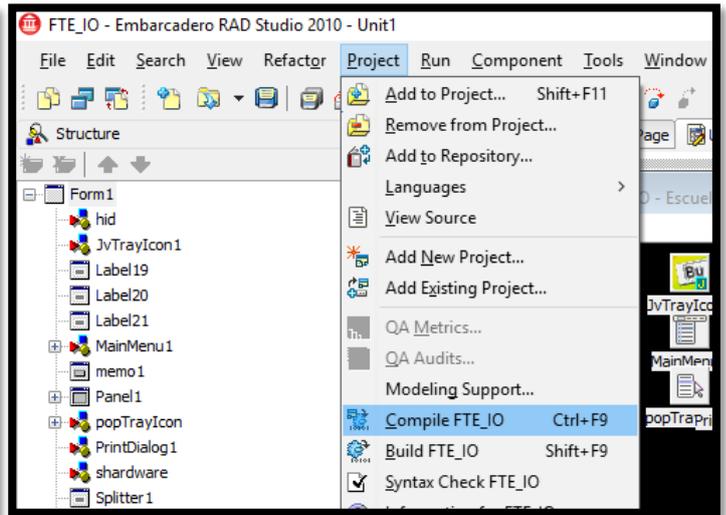
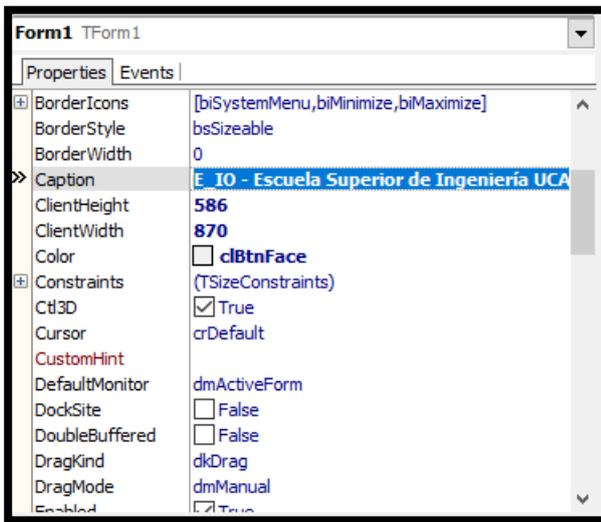
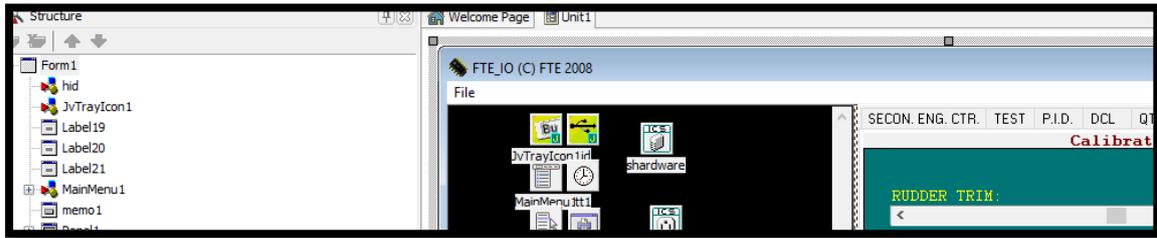
A continuación, añadir la carpeta a la librería igual que se hizo para overbytes:



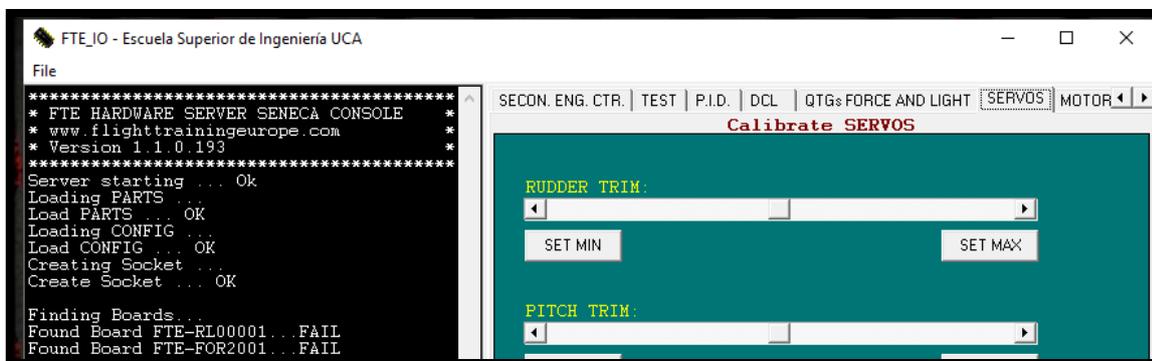
Una vez realizados todos estos pasos, el programa ya debería estar listo para recompilar los programas que funcionan en el simulador. Vamos a realizar un ejemplo con FTE_IO.exe, donde vamos a cambiarle el título a la ventana del programa.

Copiamos la carpeta del programa Hardware a un lugar para pruebas. A continuación, abrimos el archivo de proyecto Delphi.





Tras la compilación, pueden salir advertencias y 'hints' pero NO deben aparecer errores. Si aparecen errores, leer con atención el error para tener una pista de qué componente no se ha instalado correctamente.



SECCIÓN XX: ASPECTOS GENERALES DEL CÓDIGO FUENTE

1. CÓDIGO DYNAMICS

1.1. Funciones de transmisión y recepción de datos entre ordenadores

Las funciones que realizan el intercambio de datos entre los programas, mediante las conexiones ethernet, se encuentran en el archivo **etherhost.cpp**.

Las funciones utilizadas en este archivo relacionadas con los sockets se encuentran en el header winsock.h, un header obsoleto, pero al tratarse esta de la versión compilada con Visual Studio 6, no existe ningún problema. Este header evolucionó a winsock2.h y más adelante a Windows.h que incluye a ambos.

El código se compone en total de 46 archivos .cpp y 30 headers .h. Así mismo contiene los archivos de configuración config.dat y qtgdata.dat.

Dentro de dicho archivo se encuentran las siguientes funciones, que se dividen en dos grupos:

Funciones de conexión general:

`initialise_socket_fns()`

- Función de inicialización de sockets.

`initialise_socket(void)`

- Función de inicialización de sockets.

`initialise_ethernet(void)`

- Al ser llamada inicializa las funciones anteriores.

`close_ethernet(void)`

- Ejecuta EtherCleanup()

`close_sock(void)`

- Cierra el socket si hay algún error.

`EtherCleanup()`

- Ejecuta la anterior.

Funciones específicas:

ET_init_visual()

- Inicia el socket para comunicación con el PC Visual

ET_ex_visual_data()

- Intercambio de datos con el PC Visual

ET_init_IOS()

- Inicia el socket para comunicación con el PC IOS

ET_ex_ios_data()

- Intercambio de datos con el PC IOS

ET_init_hardware()

- Inicia el socket para comunicación con el PC Hardware (Programa FTE_IO)

ET_ex_hardware_data()

- Intercambio de datos con el PC Hardware (Programa FTE_IO)

ET_init_sound()

- Inicia el socket para comunicación con el PC Hardware (Programa FTE_sound)

ET_ex_sound_data()

- Intercambio de datos con el PC Hardware (Programa FTE_sound)

ET_init_screens()

- Inicia el socket para comunicación con el PC Screens

ET_ex_screens_data()

- Intercambio de datos con el PC Screens

ET_init_screens1()

- Inicia el socket para comunicación con el PC Screens

ET_ex_screens_data1()

- Intercambio de datos con el PC Screens

Funciones de inicialización. Estructura general:

```

int ET_init_visual()
{
    int ret;
    char ptr = 0;

    unsigned long non_blocking = 11;

    host_ptr = gethostbyname(SE_config.Vname);

    VIS_Socket_desc = socket(PF_INET, SOCK_DGRAM, 0);

    if (setsockopt(VIS_Socket_desc, SOL_SOCKET, SO_BROADCAST, &ptr, sizeof(int)) == -1)
    {
        printf("setsockopt error");
        return 0;
    }

    VIS_Dest.sin_family = PF_INET;
    VIS_Dest.sin_port = htons( (int)SOCK_DISCARD );
    bcopy(host_ptr->h_addr, &VIS_Dest.sin_addr, host_ptr->h_length);

    if (connect(VIS_Socket_desc, (struct sockaddr *) &VIS_Dest, sizeof(VIS_Dest)) == -1)
    {
        printf("connect error");
        return 0;
    }

    ret = ioctlsocket(VIS_Socket_desc, FIONBIO, &non_blocking);

    if (ret == -1)
    {
        printf("ioctl error");
        return 0;
    }

    FD_ZERO(&Fdset);
    Timeout.tv_sec = 0;
    Timeout.tv_usec = 0; /* select timeout values */

    return 1;
}

```

Función que recopila los datos de conexión del ordenador al otro extremo, identificándolo mediante el nombre que se introduce como parámetro de línea de comandos al ejecutar el programa. Ver capítulo de software propio de Dynamics.

Creación del socket específico para esta conexión. (En este ejemplo es la de Visual-Dynamics)

Asignación de direcciones al socket, se utiliza la estructura declarada al principio del archivo VIS_Dest, del tipo sockaddr_in

Conexión del socket mediante los datos de direcciones asignados

SOCK_DGRAM es un parámetro definido al principio del archivo que representa el puerto de conexión. Cada conexión se realiza puerto a puerto, por lo que cada función de datos tendrá un puerto distinto.

Funciones de transmisión de datos. Estructura general:

```
void ET_ex_hardware_data()
{
    int bytes_sent;
    int rec_flag;
    static float no_rec_time = 0.0;

    bytes_sent = send(H_W_Socket_desc, (const char *) IO_Sen_out, sizeof(sen_out_struct), 0);

    rec_flag = FALSE;
    while(( recv(H_W_Socket_desc, (char *) &recv_sen_in, sizeof(sen_in_struct), 0)) > 0)
    {
        memcpy(IO_Sen_in, &recv_sen_in, sizeof(sen_in_struct));
        rec_flag = TRUE;
        //PARA LAS SCREENS
        IO_Sen_out->alt1_cap_baro = IO_Sen_in->alt1_cap_baro;
        IO_Sen_out->alt2_cap_baro = IO_Sen_in->alt2_cap_baro;
        IO_Sen_out->alt3_fo_baro = IO_Sen_in->alt3_fo_baro;
        IO_Sen_out->pilot_hsi_obs = IO_Sen_in->pilot_hsi_obs;
        IO_Sen_out->pilot_hsi_atp = IO_Sen_in->pilot_hsi_atp;
        IO_Sen_out->copilot_hsi_obs = IO_Sen_in->copilot_hsi_obs;
        IO_Sen_out->copilot_hsi_atp = IO_Sen_in->copilot_hsi_atp;
        IO_Sen_out->nav2_obs = IO_Sen_in->nav2_obs;
        IO_Sen_out->pilot_ias_set = IO_Sen_in->pilot_ias_set;
        IO_Sen_out->copilot_ias_set = IO_Sen_in->copilot_ias_set;
        IO_Sen_out->battery_on = IO_Sen_in->elec_battery_switch;
        IO_Sen_out->pack25[0] = IO_Sen_in->pack25[0];
    }
    if (!rec_flag)
    {
        no_rec_time += delta_time;
        if (no_rec_time > COMS_BREAK_TIME)
        {
            printf("\n===== > HARDWARE COMS FAIL");
            no_rec_time = 0.0;
        }
    }
    else
        no_rec_time = 0.0;
#ifdef HARNESS_INPUT
    SE_update_from_sen_kbd();
#endif
}
}
```

Paquete enviado. Se invoca como char (bytes), es la estructura que contiene los datos a enviar mediante esta función preestablecida (Send())

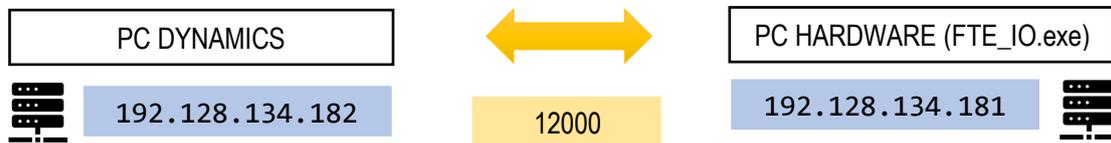
(const char *) IO_Sen_out, sizeof(sen_out_struct), 0);

while((recv(H_W_Socket_desc, (char *) &recv_sen_in, sizeof(sen_in_struct), 0)) > 0)

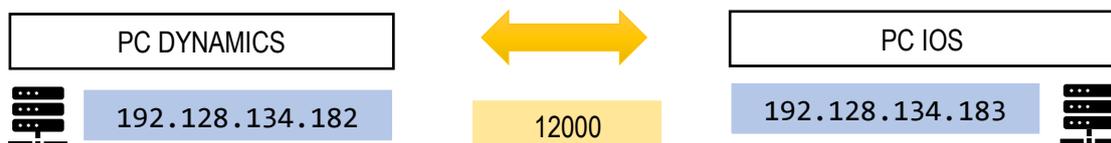
memcpy(IO_Sen_in, &recv_sen_in, sizeof(sen_in_struct));
rec_flag = TRUE;

Paquete recibido. Los bytes recibidos se arreglan como estructura preestablecida, y se copian a una estructura con la operación memcpy().

1.2. Datos enviados entre ordenadores:



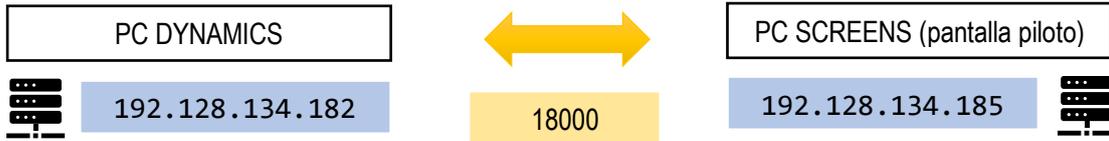
Dynamics INPUT	Dynamics OUTPUT
<p>Nombre de la estructura de datos: IO_Sen_in Nombre del tipo de estructura: sen_in_struct En numerosas partes del código se almacenan datos en la estructura a enviar. La estructura <code>sen_in_struct</code> viene declarada en <code>\Seneca\DAT_STRU\IN_HEAD.h</code>. Tiene una extensión significativa, pero en el código vienen organizadas las variables con suficiente claridad.</p> <p>Longitud en bytes: 372 bytes</p>	<p>Nombre de la estructura de datos: IO_Sen_out Nombre del tipo de estructura: sen_out_struct En numerosas partes del código se almacenan datos en la estructura a enviar. La estructura <code>sen_out_struct</code> viene declarada en <code>\Seneca\DAT_STRU\OUT_HEAD.h</code>. Tiene una extensión significativa, pero en el código vienen organizadas las variables con suficiente claridad. A continuación, algunos detalles relevantes:</p> <p>Longitud en bytes: 480 bytes</p>



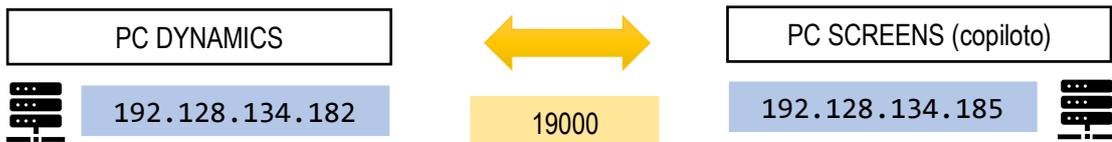
Dynamics INPUT	Dynamics OUTPUT
<p>Nombre de la estructura de datos: ios2dyn pasada a la función <code>IOS_get_data_from_ios()</code>; Nombre del tipo de estructura: ios_to_dyn Ubicación del tipo de estructura: <code>\NAV\DYN_IOS.h</code> Longitud en bytes: 552 Bytes <i>Se recibe el buffer en bytes con tamaño igual a MAX_PACKET_SIZE (1472)</i></p>	<p>Nombre de la estructura de datos: IO_dyn_ios Nombre del tipo de estructura: dyn_to_ios Ubicación del tipo de estructura: <code>\NAV\DYN_IOS.h</code> Longitud en bytes: 1208 bytes</p>



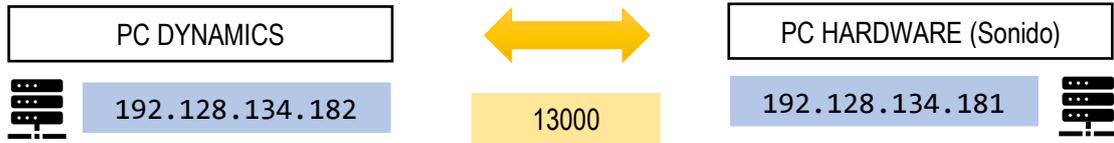
En amarillo, el puerto de comunicaciones UDP. En azul, la dirección IP configurada del adaptador ethernet activo del PC. Solo se configura la dirección IP, la máscara de subred debe aparecer automáticamente, el resto de parámetros se dejan en blanco (Gateway, DNS, etc.)



INPUT	OUTPUT
<p>Nombre de la estructura de datos: recv_sen_qtg pasada tras recibirse como: <code>memcpy(IO_Sen_qtg,&recv_sen_qtg, sizeof(sen_qtg_struct))</code> Nombre del tipo de estructura: sen_qtg_struct Ubicación del tipo de estructura: ...\DAT_STRU\In_head.h Longitud en bytes: 12 bytes</p>	<p>Nombre de la estructura de datos: IO_Sen_screens Nombre del tipo de estructura: sen_screens_struct Ubicación del tipo de estructura: ...\DAT_STRU\OUT_HEAD.h Longitud en bytes: 268 bytes</p>



INPUT	OUTPUT
N/A	<p>Nombre de la estructura de datos: IO_Sen_screens Nombre del tipo de estructura: sen_screens_struct Ubicación del tipo de estructura: ...\DAT_STRU\OUT_HEAD.h Longitud en bytes: 268 bytes</p>



INPUT	OUTPUT
N/A	Nombre de la estructura de datos: ET_tx_snd Nombre del tipo de estructura: dyn_to_snd Ubicación del tipo de estructura: ... \DAT_STRU\OUT_HEAD.h Longitud en bytes: 268 bytes


```
Buffer_ptr = (RTX_Data + sizeof(RTX_header));
```

El puntero ahora señala la posición del buffer tras el header.

- Bucle while
- Bucle switch

Dentro de estos bucles se analiza el buffer de arriba abajo y se interceptan los datos cotejándolos con un código hexadecimal. Se guardan en las variables de memoria apropiadas.

El bucle while está activo hasta que la longitud marcador llegue a la longitud final del mensaje.

A continuación, el puntero se invoca como char, de forma que el SWITCH selecciona el CASE según el contenido leído para la posición de memoria que marca el puntero. Si se busca la definición de los parámetros que definen cada CASE, se podrá observar que consisten en códigos hexadecimales.

Hay varios datos interceptados, se van a comentar los más relevantes:

COLL_RES_CODE – Detección de colisión

Cuando aparece esta cadena, se considera que el avión se ha estrellado, por eso aparece el comando:

```
dyn_state = DYN_HALT;
```

TFB_RES_CODE – Respuesta (feedback) del terreno del escenario.

```
tfb_res = (RTX_tfb_res *) Buffer_ptr; - Almacena los datos del mensaje
```

Ejecuta la función: **Terrain_feedback_response()** con el buffer **tfb_res**

Ésta función contiene `terrain_height = tfbr->set[0].height;` donde se obtiene la elevación del terreno.

1.3. Parámetros que definen la aeronave

Se van a incluir algunos parámetros generales de mayor relevancia. La relación completa de parámetros se debe consultar, continuar y analizar en el Excel adjunto a este manual.

Fichero FMS_S3FM.h

Aerodinámica:

Nombre	Valor	Función	Comentarios
kx	1.9000	calc_FM	$I_{xx} = k_x^2 m$
ky	1.3700	calc_FM	$I_{yy} = 6 k_y^2 m$
kz	2.2500	calc_FM	$I_{zz} = k_z^2 m$
Sw	18.9800	calc_FM	Superficie el ala
b	11.8600	calc_FM	Envergadura del ala
cbar	1.6000	calc_FM	Cuerda promedio
St	3.6000	calc_FM	Superficie del estabilizador horizontal
lt	5.2200	calc_FM	NOT CHECKED: Distancia efectiva desde el centro de masas al estabilizador horizontal
dCd_cowl_flap	0.0050	calc_FM	Coefficiente de resistencia del Cowl Flap
dCd_gear	0.0300	calc_FM	Coefficiente de resistencia asociado al tren de aterrizaje
dCM_gear	-0.0050	calc_FM	Coefficiente de momentos referido al tren de aterrizaje
Cla_w	5.1000	calc_FM	Coefficiente de sustentación del ala
CM0	-0.1000	calc_FM	Coefficiente de momentos
alpha_0_w	-0.0350	calc_FM	Ángulo de sustentación nula (α_0)
deda	0.3000	calc_FM	
Cla_t	4.5000	calc_FM	Coefficiente de sustentación del estabilizador de cola
alpha_s_t	-0.2500	calc_FM	Componente del ángulo de ataque... ¿cuál?
k_pitch_damp	1.5000	calc_FM	
elev_max	15.0000	calc_FM	Grados. Máximo ángulo del elevador.
rudd_max	25.0000	calc_FM	Grados. Máximo ángulo de deflexión del timón de cola.

Motor:

thrust_scale	0.7000	calc_FM	
eng_min_speed	60.0f	calc_FM	rad/s - engines don't generate any power below this speed
eng_shutdown_rate_limit	100.0000	calc_FM	// rad/s per second. Limites para dx->user[X_L_RPM]
eng_shutdown_t	0.5000	calc_FM	Script de apagado del motor
k_f_bhp_mix_rich	-0.2000	calc_FM	
f_mix_bhp_rich_max	2.0000	calc_FM	allows 100% rich before power reduction
idle_passage	0.1300	calc_FM	Propulsion forces & moments
t_throt	0.2000	calc_FM	Propulsion forces & moments

t_rpm	0.6660	calc_FM	Propulsion forces & moments
max_boost	33.0000	calc_FM	dx->user[X_R_BOOST]
max_rpm	2800.0*RPM2RPS	calc_FM	Propulsion forces & moments
max_bhp	220.0000	calc_FM	BHP Máximo.INSIDE: factor for low speed/MAP running (straight line bhp laws don't apply at low rpm or MAP)
t_boost	0.2000	calc_FM	
primer_max_ff	8.0000	calc_FM	
T_ff	0.5000	calc_FM	fuel flow damping
K_mag_fail	0.9900	calc_FM	mag fail bhp scale factor.INSIDE: factor for low speed/MAP running (straight line bhp laws don't apply at low rpm or MAP)
bhp_mag_fail	1.0000	calc_FM	Usado en corte de magnetos. bhp drop for mag fail
l_bhp_factor	0.0010	calc_FM	Usado en corte de magnetos. CJ scale factor for mag drop cases
r_bhp_factor	0.0015	calc_FM	Usado en corte de magnetos. CJ scale factor for mag drop cases
t_EGT	1.0000	calc_FM	EGT time constant
t_CHT	25.0000	calc_FM	CHT time constant
t_OilT	50.0000	calc_FM	OilT time constant
OilT_0	75.0000	calc_FM	OilT nominal temp (deg F)
OilT_man_P	5.0000	calc_FM	OilT/manifold pressure relation
OilT_Vu	-0.5000	calc_FM	OilT/speed relation
dOilT_Vu_cowl	-0.5000	calc_FM	OilT/speed effect of cowl flap
d_Vu_OilT	10.0000	calc_FM	OilT/speed nominal speed
t_OilP	4.0000	calc_FM	OilP time constant
OilP_0	30.0000	calc_FM	OilP nominal pressure (psi)
OilP_man_P	1.0000	calc_FM	OilP/manifold pressure relation
Qstart	35.0f	calc_FM	starter motor torque
beta_hi	90.0*D2R	calc_FM	Blade model
beta_lo	12.0*D2R	calc_FM	Blade model
rpm_hi	$2800.0 * (2*PI)/60.0$	calc_FM	
rpm_lo	$1000.0 * (2*PI)/60.0$	calc_FM	
CTmax	0.1000	calc_FM	Simple blade stall model
Jprop	2.0000	calc_FM	Blade model
prop_rad	0.9650	calc_FM	m
prop_area	2.9300	calc_FM	m^2. Cálculo de TpropW y QpropW en Wind AXES

Piloto automático:

flash1_rate	1.0000	KFC150_exec_FDL	
flash2_rate	2.0000	KFC150_exec_FDL	
ap_Ktheta_p	-10.0000	KFC150_exec_FDL	pitch_servo_dem
ap_Ktheta_i	0.0000	KFC150_exec_FDL	pitch_servo_dem
ap_Kphi_p	10.0000	KFC150_exec_FDL	
ap_Kphi_i	0.5000	KFC150_exec_FDL	
ap_Kpsi_p	4.0000	KFC150_exec_FDL	*fd_roll
ap_theta0	4.0*D2R	KFC150_exec_FDL	theta_dem
ap_Khdot_p	0.0300	KFC150_exec_FDL	theta_dem
ap_Kalt_p	0.0050	KFC150_exec_FDL	hdot_dem
ap_Kalt_i	0.0100	KFC150_exec_FDL	hdot_dem
ap_Kalt_d	0.0000	KFC150_exec_FDL	hdot_dem
ap_phi_max	20.0*D2R	KFC150_exec_FDL	*fd_roll
ap_theta_max	(+)15*D2R	KFC150_exec_FDL	*fd_pitch
ap_theta_min	(-)10*D2R	KFC150_exec_FDL	*fd_pitch

Mecánica del vuelo (estabilidad y control):

Yv	-1.0000	calc_FM	Principal Derivatives. side force as a result of Sideslip
Lv	-1.0000	calc_FM	Principal Derivatives. Roll as a result of Sideslip
Lp	-0.5000	calc_FM	Principal Derivatives. Roll Damping
Nv	1.0000	calc_FM	Principal Derivatives. yaw Daming as a result of Sideslip
Nr	-0.5000	calc_FM	Principal Derivatives. yaw Daming
Yp	0.0000	calc_FM	Secondary Derivatives. side force as a result of Roll Rate
Yr	0.0000	calc_FM	Secondary Derivatives. side force as a result of yaw Rate
Lr	0.1000	calc_FM	Secondary Derivatives. Roll as a result of yaw Rate
Np	0.0000	calc_FM	Secondary Derivatives. yaw as a result of Roll Rate
Yzeta	0.0000	calc_FM	Control Derivatives. side force as a result of rudder
Lzeta	0.0000	calc_FM	Control Derivatives. roll force as a result of rudder
Nzeta	-0.1250	calc_FM	Control Derivatives. yaw force as a result of rudder
Yxsi	0.0000	calc_FM	Control Derivatives. side force as a result of Aileron
Lxsi	-0.2500	calc_FM	Control Derivatives. roll force as a result of Aileron
Nxsi	0.0000	calc_FM	Control Derivatives. yaw force as a result of Aileron

Tren de aterrizaje:

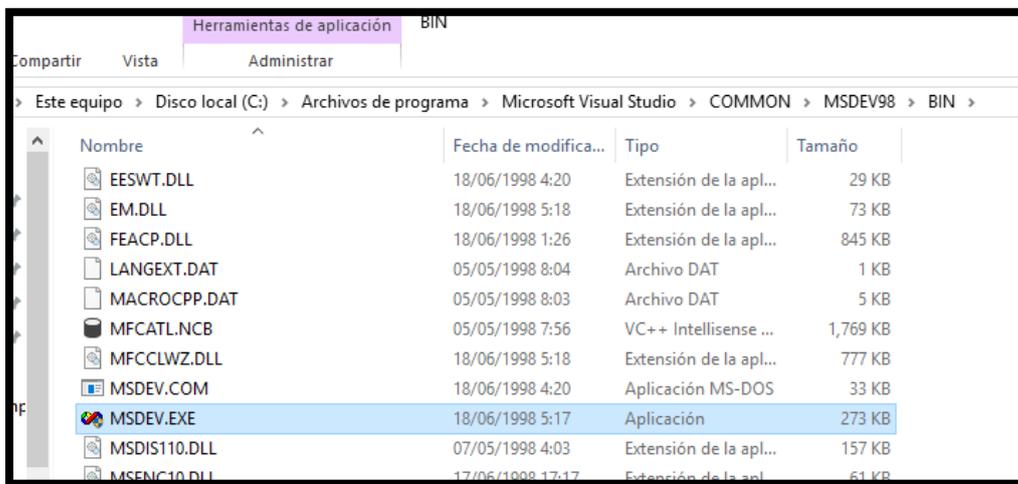
K_nose_steer	0.0500	calc_FM	NG = nose gear, RG = right gear, LG = left gear
t_N_gear_1	1.0000	calc_FM	
t_L_gear_1	1.1000	calc_FM	
t_R_gear_1	1.1200	calc_FM	
max_gear_rate_1	10.0000	calc_FM	Tope maximo de velocidad de bajada de las ruedas parece
t_N_gear_2	0.5000	calc_FM	Mismo que lo anterior pero para despliegue de emergencia parece
t_L_gear_2	0.5500	calc_FM	
t_R_gear_2	0.5700	calc_FM	
max_gear_rate_2	10.0000	calc_FM	Tope maximo de velocidad de bajada de las ruedas parece. (caso de emergencia)
NG_k	160000.0000	calc_FM	Parámetro asociado a compresión
RG_k	40000.0000	calc_FM	Parámetro asociado a compresión
LG_k	40000.0000	calc_FM	Parámetro asociado a compresión
NG_b	150000.0000	calc_FM	
RG_b	15000.0000	calc_FM	
LG_b	15000.0000	calc_FM	
NG_x	2.0000	calc_FM	Coordenada
RG_x	-1.0000	calc_FM	Coordenada
LG_x	-1.0000	calc_FM	Coordenada
NG_y	0.0000	calc_FM	Coordenada
RG_y	2.0000	calc_FM	Coordenada
LG_y	-2.0000	calc_FM	Coordenada
NG_xl	2.1000	calc_FM	Coordenada
RG_xl	2.0000	calc_FM	Coordenada
LG_xl	2.0000	calc_FM	Coordenada
NG_preload	0.0000	calc_FM	
RG_preload	0.0000	calc_FM	
LG_preload	0.0000	calc_FM	

SECCIÓN XIX: INTRODUCCIÓN DE UN NUEVO PC PARA ESTUDIOS

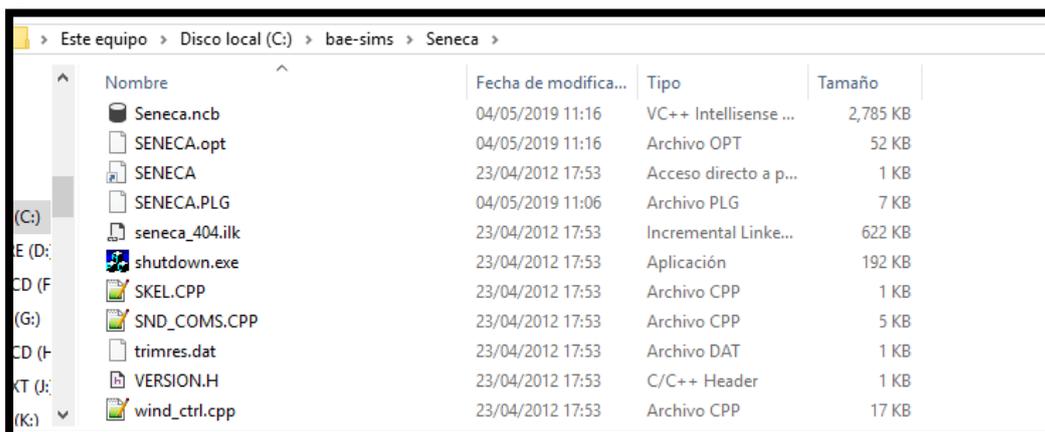
1. CONEXIÓN DEL NUEVO PC AL EQUIPO

Pasos para añadir un ordenador al simulador que reciba datos en tiempo real mediante conexión por protocolo UDP. Revisar los capítulos “Recompilando Dynamics” y “generalidades del código”.

Recordatorio: Visual Studio 98 se ha comprobado que funciona perfectamente en Windows 10, sin necesidad de instalarlo, con compatibilidad a Windows XP SP 2 y ejecutado como Admin. Debe estar ‘instalado’ de la siguiente forma y con la siguiente ruta:



El workspace con el que trabajaremos debe estar situado tal y como se muestra en la siguiente figura:



El ejecutable resultante aparecerá en la carpeta C:\Seneca.

Importante: antes de modificar/añadir/eliminar cualquier cosa de un PC del simulador, es obligatorio crear una copia de seguridad de lo que se vaya a manipular.

En primer lugar, es necesario modificar el archivo HOSTS de Windows del ordenador Dynamics donde se ejecuta el programa que vamos a compilar. Dicho archivo se encuentra en la carpeta System32, drivers,

etc. En dicho archivo se encontrará una lista de direcciones IP con un nombre asociado a cada una. Son las direcciones IP de todos los PC's que forman el simulador, y tendremos que añadir una nueva línea con una nueva IP que tendremos que configurarle como IPv4 al ordenador a añadir, así como un nombre que debemos apuntar para después. Se edita el archivo con Notepad, se guarda y se reemplaza.

El siguiente paso es abrir el Workspace del programa con VS98. Los archivos que se editarán para crear una nueva conexión son:

- Seneca.cpp
- Etherhost.cpp
- Config.h

Abrimos el archivo Seneca.cpp con el VS98.

Al ejecutar el programa final, lo hacemos con un acceso directo en el que hay una serie de parámetros de línea de comandos. Dichos parámetros son precisamente los nombres de los PC's que están declarados en el archivo HOSTS editado previamente, anteceditos con un guión y una letra identificadora. Si quisiéramos añadir un nuevo PC, aparte del nuevo PC que hay en esta versión adjunta del Workspace, en el acceso directo tendríamos que añadir un nuevo guión con el nombre del nuevo PC y una letra identificadora que no esté ya repetida. En este caso ya hay un nuevo PC que se nombró en HOSTS, llamado XSender. La letra identificadora sirve para lo siguiente:

```

        if ( argv[i][0]=='-' && argv[i][1]=='X') //AÑADIDO POR MANOLO
        {
            strcpy(SE_config.Xname,&argv[i][2]); //AÑADIDO POR MANOLO
            SE_config.senderr = TRUE; //AÑADIDO POR MANOLO
        }
    } //End of check command line arguments
}

```

Si nos vamos a la función `void command_line_decode(int argc, char *argv[])` veremos como se ha añadido un nuevo IF, donde el código lee la línea de comandos, detecta la letra aquí especificada y copia el resto de la palabra del parámetro en una estructura llamada SE_config tipo sen_config. Por lo que aparte, tendremos que modificar dicha estructura (config.h) para que tengamos el nombre del nuevo pc guardado también al leerse la línea de comandos.

```

typedef struct
{
    unsigned visual      :1;
    unsigned instructor  :1;
    unsigned sound      :1;
    unsigned awe32      :1;
    unsigned hardware    :1;
    unsigned screens     :1; //ULTIMA MODIFICACION DE FTE
    unsigned senderr     :1; //AÑADIDO POR MANOLO

    char      Vname[15];
    char      Iname[15];
    char      Sname[15];
    char      Hname[15];
    char      Cname[15];
    char      Xname[15]; //AÑADIDO POR MANOLO
}sen_config;

```

Se puede observar que, si en la línea de comandos encuentra el parámetro para la letra identificadora, entonces copia la palabra y declara TRUE la configuración. Esto es para que se ejecuten las funciones socket y de envío de datos asociadas a este nuevo PC, como se ve a continuación.

Nos movemos ahora a void main (int argc, char *argv[])

```
if(SE_config.senderr == TRUE)
{
    /*testing*/
    ET_init_SIMULINKoutputs(); //AÑADIDO POR MANOLO
    ET_init_SIMULINKscr(); //AÑADIDO POR MANOLO
    ET_init_SIMULINKhdw(); //AÑADIDO POR MANOLO_12_04
    /*testing*/
    printf("\n SENDER");
}
```

En esta función hay que añadir un nuevo IF donde se llame a las funciones de inicio de los sockets de transferencia de datos para el nuevo PC, las cuales tienen siempre el prefijo ET_init_.

A continuación, bajamos a la función void ethernet_comms(void) en el mismo archivo.

Aquí añadimos las nuevas funciones de transferencia de datos, las cuales tienen siempre el prefijo ET_ex_.

```
if(SE_config.senderr == TRUE) //AÑADIDO POR MANOLO
{
    /*testing*/
    ET_ex_SIMULINKoutputs(); //AÑADIDO POR MANOLO
    ET_ex_SIMULINKscr(); //AÑADIDO POR MANOLO_12_04
    ET_ex_SIMULINKhdw(); //AÑADIDO POR MANOLO_12_04
    /*testing*/
}
```

Abrimos ahora el archivo etherhost.cpp. En este archivo es donde están declaradas todas las funciones ET_ex y ET_init, así como todo lo relacionado con los sockets.

Se va a comentar a continuación la conexión y transmisión de datos utilizada para añadir un ordenador que reciba los datos de la estructura OUTPUTS, de tipo 'OUTPUTS', definida en el header Fmgdefs.h.

```
int ET_init_SIMULINKoutputs() //AÑADIDO POR MANOLO
{
    int ret;
    char ptr = 0;

    unsigned long non_blocking = 11;

    host_ptr = gethostbyname(SE_config.Xname);

    SENDER_Socket_desc = socket(PF_INET, SOCK_DGRAM, 0);

    if (setsockopt(SENDER_Socket_desc, SOL_SOCKET, SO_BROADCAST, &ptr, sizeof(int)) == -1)
    {
        printf("setsockopt error");
        return 0;
    }

    SENDERR_Dest.sin_family = PF_INET;
    SENDERR_Dest.sin_port = htons( (int)19500 );
    bcopy(host_ptr->h_addr, &SENDERR_Dest.sin_addr, host_ptr->h_length);

    if (connect(SENDER_Socket_desc, (struct sockaddr *) &SENDERR_Dest, sizeof(SENDERR_Dest)) == -1)
    {
        printf("connect error");
        return 0;
    }

    ret = ioctlsocket(SENDER_Socket_desc, FIONBIO, &non_blocking);

    if (ret == -1)
    {
        printf("ioctl error");
        return 0;
    }

    FD_ZERO(&Fdset);
    Timeout.tv_sec = 0;
    Timeout.tv_usec = 0;

    return 1;
}
```

La primera variable que se utiliza es el nombre del Host, que el programa ha leído de la línea de comandos, y que ahora buscará su dirección IP en el archivo HOSTS que editamos al principio.

```
/*-----*/
/* Locals */
/*-----*/

SOCKET sd;

static sockaddr_in dest;
static sockaddr_in H_W_Dest; /* Does this have to be global?? */
static sockaddr_in IOS_Dest; /* Does this have to be global?? */
static sockaddr_in VIS_Dest; /* Does this have to be global?? */
static sockaddr_in SND_Dest; /* Does this have to be global?? */
static sockaddr_in SCR_Dest;
static sockaddr_in SCR_Dest1;

static sockaddr_in SENDERR_Dest; //AÑADIDO POR MANOLO
static sockaddr_in SENDERR_Dest2; //AÑADIDO POR MANOLO_12_04
static sockaddr_in SENDERR_Dest3; //AÑADIDO POR MANOLO_12_04

static fd_set Fdset;
static timeval Timeout;
static hostent *host_ptr;

static int socket_open = 0;
static int H_W_Socket_desc; // Connection socket descriptor for hardware
static int IOS_Socket_desc; // Connection socket descriptor for instructor
static int VIS_Socket_desc; // Connection socket descriptor for Visuals
static int SND_Socket_desc; // Connection socket descriptor for sound
static int SCR_Socket_desc; // Connection socket descriptor for hardware
static int SCR_Socket_desc1; // Connection socket descriptor for hardware
static int SENDER_Socket_desc; //AÑADIDO POR MANOLO
static int SENDER_Socket_desc2; //AÑADIDO POR MANOLO_12_04
static int SENDER_Socket_desc3; //AÑADIDO POR MANOLO_12_04
```

Esta función utiliza las estructuras con sufijo `_Socket_desc` y `_Dest`, por lo que al principio del archivo tendremos que declararlas para cada función que creamos.

```

/*-----*/
/* Locals */
/*-----*/

SOCKET sd;

static sockaddr_in dest;
static sockaddr_in H_W_Dest; /* Does this have to be global?? */
static sockaddr_in IOS_Dest; /* Does this have to be global?? */
static sockaddr_in VIS_Dest; /* Does this have to be global?? */
static sockaddr_in SND_Dest; /* Does this have to be global?? */
static sockaddr_in SCR_Dest;
static sockaddr_in SCR_Dest1;

static sockaddr_in SENDERR_Dest; //AÑADIDO POR MANOLO
static sockaddr_in SENDERR_Dest2; //AÑADIDO POR MANOLO_12_04
static sockaddr_in SENDERR_Dest3; //AÑADIDO POR MANOLO_12_04

static fd_set Fdset;
static timeval Timeout;
static hostent *host_ptr;

static int socket_open = 0;
static int H_W_Socket_desc; // Connection socket descriptor for hardware
static int IOS_Socket_desc; // Connection socket descriptor for instructor
static int VIS_Socket_desc; // Connection socket descriptor for Visuals
static int SND_Socket_desc; // Connection socket descriptor for sound
static int SCR_Socket_desc; // Connection socket descriptor for hardware
static int SCR_Socket_desc1; // Connection socket descriptor for hardware
static int SENDER_Socket_desc; //AÑADIDO POR MANOLO
static int SENDER_Socket_desc2; //AÑADIDO POR MANOLO_12_04
static int SENDER_Socket_desc3; //AÑADIDO POR MANOLO_12_04

```

El siguiente parámetro que hay que modificar es el puerto, para esta primera nueva función se utilizó el puerto 19500 como se puede ver. Simplemente especificamos un puerto que esté libre.

Por último, creamos (copiamos y modificamos) la función tipo ET_ex, para la cual usaremos la misma estructura _Socket_desc que antes, y especificaremos que datos queremos enviar. En este caso, los datos son los contenidos en la estructura outputs, la cual, más arriba está declarada como externa (se declara y se usa en el archivo Dynamics.cpp), pero también hay que especificar el header donde se declara el tipo de estructura, por lo que aparece Fmgdefs.h como #include. Por último no olvidar escribir el tipo de estructura anteriormente mencionado en sizeof().

```

void ET_ex_SIMULINKoutputs() //AÑADIDO POR MANOLO
{
    int bytes_sent;

    /*bytes_sent = send(SENDER_Socket_desc, (const char *) &sendtest,sizeof(sendtest), 0);*/
    /*sendtest = sendtest + 1;*/ // Contador para pruebas

    bytes_sent = send(SENDER_Socket_desc, (const char *) &outputs,sizeof(outputs), 0);

    /*
    OUTPUTinbox.a = OUTPUTinbox.a + 0.01;
    OUTPUTinbox.b = OUTPUTinbox.b + 0.01;
    OUTPUTinbox.c = OUTPUTinbox.c + 0.01;
    */
}

```

```
/*-----*/  
/* External Variables */  
/*-----*/  
  
extern dyn_to_ios IO_dyn_ios;  
extern OUTPUTS outputs;  
  
#include "FLT DY\Fmgdefs.h" //AÑADIDO 1/04
```

A partir de aquí, si no se ha olvidado nada, el código debería poder compilarse mediante Rebuild All, resultando en 0 errores y 2 warnings. El ejecutable se encontrará en la carpeta C:\Seneca\ con nombre de archivo seneca_DYN404.exe. Tan solo tendremos que copiar este ejecutable, cambiarle el nombre y pegarlo en el ordenador del simulador Dynamics, en la misma carpeta donde se encuentra la versión anterior, y crear el acceso directo correspondiente con los parámetros tal y como se explicó al principio.

Los datos que se estarán transmitiendo (la estructura outputs), si fuese a ser recibida por otro programa compilado con .cpp, bastaría con recibir los bytes, y decodificarlos usando una estructura declarada de forma similar en dicho programa.

En este caso, los datos están siendo recibidos en Simulink en un ordenador portátil, el cual recibe los paquetes UDP en tiempo real y los interpreta de la siguiente manera: Los 27 primeros datos son float simples, ocupan 4 bytes, luego hay lo que parece ser una matriz, un vector float y un vector de enteros. Hasta ahora, lo que se ha comprobado que funciona es la recepción de los primeros 27 datos, mediante un bloque de Simulink al que se le configura el parámetro de tipo de datos recibidos como 'single' (4 bytes), decodificando correctamente los datos float de C++. Se está trabajando para comprobar si el resto de datos de esta estructura se ha recibido correctamente estableciendo una longitud de mensaje excedente. Además, debe de haber una forma mucho más precisa de capturar todos los datos de distinto tipo a la vez, pero por ahora, la forma más sencilla descubierta de obtenerlos ha sido esta.

Como último recurso, para obtener cualquier dato que se pueda encontrar en el código, siempre quedará declarar una estructura nueva en algún lugar, guardarle el parámetro a capturar invocándolo como un float o entero, y transmitir dicha estructura por un puerto distinto. Pero hasta ahora se ha optado por no añadir más cosas al código y usar lo que ya se encuentra en él.

En el siguiente vídeo se puede ver un ejemplo de un 'replay' de un log de datos capturados. Es decir, en el simulador se capturan los datos, se guardan a un archivo timeseries de MATLAB, y luego en cualquier otro momento se reproducen con el mismo sample rate que fueron capturados en tiempo real.

<https://www.youtube.com/watch?v=qJOAnu8VtGc>

SECCIÓN XX: INTERFAZ DE COMUNICACIÓN CON FLIGHTGEAR

Otra de las mejoras posibles al simulador gracias a conocer cómo funciona la transmisión de datos, es la posibilidad de utilizar otro software ya existente para la renderización del escenario como reemplazo del Visual antiguo. Esto se ha logrado con el software libre FlightGear, aunque se conoce que también es posible con X-Plane. De esta forma, podemos utilizar estos simuladores de vuelo como generadores del entorno mientras el modelo de vuelo y todo lo asociado al avión sigue siendo tarea de nuestro simulador.

Para lograr esto con FlightGear, se van a describir a continuación los pasos, sin embargo, es preciso que el lector ya esté familiarizado con las conexiones ethernet y protocolo UDP con el que trabaja el simulador, siendo capaz de realizar lo descrito en la sección anterior de este manual.

Sobre FlightGear:



FlightGear es un simulador de vuelo gratuito que es compatible con la última versión de Windows. En el desarrollo de este manual se ha utilizado la versión "FlightGear 2018.3.2" sin embargo es muy probable que una versión posterior funcione de manera similar.

En primer lugar, instalar este software siguiendo las instrucciones que se muestren en su web. Es recomendable instalarlo fuera de la carpeta típica de Archivos de Programa.

Las posibilidades de comunicación de FlightGear se comentan a continuación, para más detalle, consultar su documentación oficial y su wiki en internet.

Protocolo nativo:

Se trata de un protocolo de comunicaciones en el que FlightGear se comunica con un programa externo mediante la red con un paquete de datos prefijado. Este paquete de datos, tanto de salida como de entrada contiene multitud de variables con un tipo de datos definido. Por tanto, el software externo a FlightGear debe adaptarse para poder recibir y enviar esos datos de la misma forma que lo hace FlightGear. Un ejemplo a poner en práctica de forma fácil con este protocolo puede estudiarse en Simulink, cargando el ejemplo: "<https://www.mathworks.com/help/aeroblks/hi-20-project-with-optional-flightgear-interface.html>"

Protocolo genérico:

Este protocolo de comunicaciones es el que va a utilizarse para nuestro caso. Es completamente distinto del anterior. En este caso, nosotros creamos el paquete de datos que queremos que FlightGear reciba y emita por la red. Se van a describir los pasos para lograr el objetivo de esta sección:



Se recomienda utilizar el programa Notepad++ así como tener conocimientos básicos sobre archivos .xml y su sintaxis. Se considera a partir de aquí que FlightGear se ha instalado con éxito y puede ejecutarse mediante su lanzador con GUI sin problemas. Para más información de este protocolo de comunicaciones consultar http://wiki.flightgear.org/Generic_protocol

En primer lugar, navegar a la carpeta:

[RUTA DE INSTALACIÓN].\FlightGear 2018.3.2\data\Protocol\

Y crear un archivo .XML mediante Notepad++, el nombre de este archivo será el nombre de nuestro protocolo de comunicaciones.

Por ejemplo, se ha creado para este ejemplo un archivo llamado fromvisual.xml

El contenido de un .XML para la tarea que estamos tratando sería el siguiente:

```

1  <?xml version="1.0"?>
2  <PropertyList>
3
4  <generic>
5  <input>
6      <line_separator>newline</line_separator>
7      <var_separator>tab</var_separator>
8      <binary_mode>>true</binary_mode>
9      <byte_order>host</byte_order>
10
11
12  <chunk>
13      <node>RUTA A LA VARIABLE A MODIFICAR</node>
14      <type>TIPO DE DATO</type>
15      <format>%f</format>
16  </chunk>
17
18
19  </input>
20
21
22  <output>
23      <line_separator>newline</line_separator>
24      <var_separator>tab</var_separator>
25      <binary_mode>>true</binary_mode>
26      <byte_order>host</byte_order>
27
28  <chunk>
29      <node>RUTA DE LA VARIABLE A EXTRAER</node>
30      <type>float</type>
31      <format>%f</format>
32  </chunk>
33
34  </output>
35  </generic>
36
37  </PropertyList>

```

Donde cada subsección CHUNK es un parámetro a añadir en el paquete de datos de comunicación. A continuación se va a definir un .XML para nuestro ejemplo:

```

<?xml version="1.0"?>
<PropertyList>
  <generic>
    <input>
      <line_separator>newline</line_separator>
      <var_separator>tab</var_separator>
      <binary_mode>>true</binary_mode>
      <byte_order>host</byte_order>
      <chunk>
        <node>/environment/visibility-m</node>
        <type>float</type>
        <format>%f</format>
      </chunk>
      <chunk>
        <node>/orientation/pitch-deg</node>
        <type>float</type>
        <format>%f</format>
      </chunk>
      <chunk>
        <node>/orientation/roll-deg</node>
        <type>float</type>
        <format>%f</format>
      </chunk>
      <chunk>
        <node>/orientation/heading-deg</node>
        <type>float</type>
        <format>%f</format>
      </chunk>
      <chunk>
        <name>altitude-ft</name>
        <type>double</type>
        <node>/position/altitude-ft</node>
      </chunk>
      <chunk>
        <name>latitude-deg</name>
        <type>double</type>
        <node>/position/latitude-deg</node>
      </chunk>
      <chunk>
        <name>longitude-deg</name>
        <type>double</type>
        <node>/position/longitude-deg</node>
      </chunk>
    </input>
  </generic>
</PropertyList>

```

Para saber dónde encontrar los 'nodos' o parámetros, en el menú en partida DEPURAR, clic en VISOR DE PROPIEDADES INTERNAS y navegar por el árbol de parámetros.

Una vez creado éste .XML, creamos la parte correspondiente a nuestro código Dynamics.

Se va a utilizar el puerto 12008 y la misma función de transmisión que emplea para VISUAL.

```

/*-----SEGUNDO VISUAL 14_08-----*/
int ET_init_SENDERvisual()
{
    int ret;
    char ptr = 0;

    unsigned long non_blocking = 11;

    host_ptr = gethostbyname(SE_config.Xname);

    SENDER_Socket_descVis = socket(PF_INET, SOCK_DGRAM, 0);

    if (setsockopt(SENDER_Socket_descVis, SOL_SOCKET, SO_BROADCAST, &ptr, sizeof(int)) == -1)
    {
        printf("setsockopt error");
    }

    SENDERR_Visual.sin_family = PF_INET;
    SENDERR_Visual.sin_port = htons( (int)12008 );
    bcopy(host_ptr->h_addr, &SENDERR_Visual.sin_addr, host_ptr->h_length);

    if (connect(SENDER_Socket_descVis, (struct sockaddr *) &SENDERR_Visual, sizeof(SENDERR_Visual)) == -1)
    {
        printf("connect error");
    }

    ret = ioctlsocket(SENDER_Socket_descVis, FIONBIO, &non_blocking);

    if (ret == -1)
    {
        printf("ioctl error");
    }

    FD_ZERO(&Fdset);
    Timeout.tv_sec = 0;
    Timeout.tv_usec = 0; /* select timeout values */

    return 1;
}

void ET_ex_visual_data()
{
    int bytes_sent;
    int bytes_sent_tosender;
    int num_bytes;
    int rec;
    static float no_rec_time = 0.0;

    /* Need to make sure we do not send too much data, else
       receiving PC slows down */

    // format RTX_Data for send purposes. Return no bytes to send
    num_bytes = RR_encode_vis_buffer();

    bytes_sent = send(VIS_Socket_desc, (const char *) &RTX_Data, MAX_PACKET_SIZE, 0);
    bytes_sent_tosender = send(SENDER_Socket_descVis, (const char *) &FGOutput, sizeof(FGProtocol), 0);

    rec = FALSE;
    while (( recv(VIS_Socket_desc, (char *) &RTX_Data, MAX_PACKET_SIZE, 0)) > 0)
    {
        rec = TRUE;
    }

    if (rec)
    {
        //Since both send and receive use the same buffer, we must NOT try
        //and decode RTX_Data if nothing was received (since that would
        //be decoding the output data!

        RR_decode_vis_buffer();
        no_rec_time = 0.0;
    }
    else
    {
        no_rec_time += delta_time;
        if (no_rec_time > COMS_BREAK_TIME)
        {
            printf("\n===== > VISUAL COMS FAIL");
            no_rec_time = 0.0;
        }
    }
}

```

Donde la estructura enviada se define como:

```
typedef struct {
    float vis;
    float theta;
    float phi;
    float psi;
    double altitude;
    double latitude;
    double longitude;
} FGProtocol; //FGPROTOCOL
```

Y le hemos asignado los datos en el archivo REQ_RES.cpp dentro de las funciones Set_own_craft_req() y Set_sph_own_craft_req():

```
/*FGPROTOCOL*/
FGOutput.vis =(float)IOS_atis.Visibility ;
FGOutput.theta =(float)outputs.theta * R2D ;
FGOutput.psi =(float)outputs.psi * R2D ;
FGOutput.phi =(float)outputs.phi * R2D ;
//FGOutput.altitude=(double)-states.D * m2ft ;
FGOutput.altitude=(double)-states.D * m2ft;
FGOutput.latitude=(double)states.latitude * R2D ;
FGOutput.longitude=(double)states.longitude * R2D ;
/*FGPROTOCOL*/
```

Una vez preparado y recompilado el código, ejecutamos flightgear creando un archivo .bat en el escritorio con el siguiente contenido:

E:

cd E:\Programas\FlightGear 2018.3.2\

SET FG_ROOT=E:\Programas\FlightGear 2018.3.2\data

.\bin\vgfs --generic=socket,in,120,,12008,udp,fromvisual --enable-terrasync --airport=LEJR --runway=02 -
-altitude=2000 --timeofday=noon --fdm=null --disable-panel

Donde ha de ajustarse la ruta del programa de forma adecuada, y el contenido de la línea de comandos deben estar redactados en una misma línea.