

DISEÑO E IMPLEMENTACIÓN DE UNA INTERFAZ HARDWARE-SOFTWARE PARA APLICACIONES EN SIMULACIÓN DE VUELO

GUSTAVO ADOLFO LEGARDA BERMÚDEZ

JAIME ANDRÉS COLLAZOS GONZÁLEZ

UNIVERSIDAD DEL CAUCA

FACULTAD DE CIENCIAS NATURALES, EXACTAS Y DE LA EDUCACIÓN

PROGRAMA DE INGENIERÍA FÍSICA

POPAYÁN

2009

DISEÑO E IMPLEMENTACIÓN DE UNA INTERFAZ HARDWARE-SOFTWARE PARA APLICACIONES EN SIMULACIÓN DE VUELO

GUSTAVO ADOLFO LEGARDA BERMÚDEZ

JAIME ANDRÉS COLLAZOS GONZÁLEZ

Trabajo de grado elaborado para optar por el título de ingeniero físico.

Director, Mario Andrés Córdoba

Ingeniero Electrónico y de Telecomunicaciones, Universidad del Cauca

Airplane Design, University of Kansas

UNIVERSIDAD DEL CAUCA

FACULTAD DE CIENCIAS NATURALES, EXACTAS Y DE LA EDUCACIÓN

PROGRAMA DE INGENIERÍA FÍSICA

POPAYÁN

2009

Nota de Aceptación:

Ing. Mario Andrés Córdoba González
Director de trabajo de grado

Ing. Javier Cortés Carvajal
Firma del jurado

Ing. Leonairo Pencue Fierro
Firma del jurado

Fecha de sustentación, Popayán, lunes 8 de junio de 2009

Dedicatoria

A mi abuelo que fue un padre para mí, porque su fortaleza me ensaño a luchar por las metas que quiero. Siempre te llevo en mi mente y corazón.

Jaime Andrés Collazos González.

A mi familia, especialmente a mi madre mi negrita hermosa, por todo su amor, sacrificio y dedicación.

Gustavo Adolfo Legarda Bermúdez.

Agradecimientos

En primer lugar queremos agradecer a nuestras familias por su apoyo incondicional, por creer siempre en nosotros y principalmente por todo ese amor y confianza que siempre nos han brindado, este logro no es solo nuestro, también es de ustedes y su apoyo fue nuestra fuerza impulsadora

A todos nuestros amigos y profesores con los que no solo hemos evolucionado a un nivel intelectual si no también a un nivel personal y espiritual.

Finalmente agradecemos a nuestro director de trabajo de grado Mario Andrés Córdoba por abrirnos las puertas y colaborarnos en las diferentes etapas de este proyecto.

Gracias.

PD. Gracias Dios por tu fuerza y sabiduría, esperamos seas guía en esta nueva etapa de nuestras vidas.

RESUMEN

La simulación de vuelo es un importante instrumento en la formación de pilotos, investigación científica y desarrollo tecnológico, lograr la construcción de cualquiera de estos instrumentos tiene un alto costo y compleja arquitectura. Este trabajo presenta una interfaz de comunicación aplicada a simulación de vuelo, con un diseño versátil en busca de una solución alternativa para diseño y desarrollo de entrenadores de vuelo en Colombia.

El sistema desarrollado, puede ser usado para enlazar los controles e indicadores de una cabina con un software de simulación de vuelo, compuesto por dos interfaces de comunicación.

La primera interfaz, se desarrollo como un sistema de adquisición de datos denominado interfaz hardware SIES 1.0, fundamentado en los dispositivos de interfaz humana (HID) del protocolo USB 2.0, logrando de esta manera el reconocimiento como dispositivo de juego, con la capacidad de procesar 8 entradas analógicos, 32 entradas digitales y punto de mira, presenta compatibilidad con todos los software de simulación y sistemas operativos. Para su desarrollo se empleo el microcontrolador PIC18F4550, y el firmware estuvo basado en las librerías USB de Microchip.

La segunda interfaz, está compuesta de un hardware y software denominada SIES 2.0, mediante la cual, se logro tener acceso de lectura y escritura a todas las variables asociadas al Flight Simulator X de Microsoft, con capacidad de despliegue y utilización de esta información. El modulo software de SIES 2.0, controla el acceso del hardware a las variables del simulador junto a la transmisión y recepción de datos. El modulo Hardware despliega y utiliza esta información de acuerdo a los controles y actuadores correspondientes. El firmware del microcontrolador esta soportado en el protocolo de comunicación USB 2.0 y la clase CDC (Class Communication Device). El código fuente del software se desarrollo en Visual Basic 6.0 y con soporte en la librería FSUIPC de Peter Dowson.

CONTENIDO

INTRODUCCIÓN.....	14
1. MARCO TEÓRICO.....	15
1.1 Simuladores de vuelo.....	15
1.2 Software de simulación de vuelo.....	15
1.2.1 X-Plane.....	16
1.2.2 FlightGear.....	16
1.2.3 Microsoft Flight Simulator.....	17
1.3 Entrenadores de vuelo.....	17
1.3.1 Clasificación de entrenadores de vuelo.....	18
1.4 Concepto de interfaz.....	23
1.4.1 Bus Serie Universal (USB).....	23
2. ANTECEDENTES.....	31
2.1 Desarrollos no Formales.....	32
2.2 Desarrollos I+D en Universidades.....	33
2.2.1 Diseño y construcción de simuladores a partir de aviones fuera de servicio para parques de diversión.....	34
2.2.2 Interfaz 3D para un vehículo aéreo no tripulado.....	34
2.2.3 Diseño de un simulador de tren.....	34
2.2.4 Relay, Una interfaz futurista para la conducción remota.....	35
2.3 Otros desarrollos de interfaces electrónicas similares en otros campos.....	35
2.3.1 Desarrollo de una interfaz HID, para aplicación en un simulador virtual de Laparoscopia.....	35
2.3.2 RC Joystick NG2.....	36
2.3.3 Desarrollo de una estación meteorológica USB.....	36
2.3.4 Desarrollo de un sistema hardware software que mejore la funcionalidad de un ordenador personal para personas con discapacidad.....	36
3. DISEÑO CONCEPTUAL INTERFAZ DE ENTRADA DE DATOS (SIES 1.0).....	37
3.1 Funciones y características de la interfaz.....	37

3.1.1	Modulo de adquisición y procesamiento de datos.....	39
3.1.2	Compatibilidad con el sistema operativo.	39
3.1.3	Reorganización de la información y conectividad.	40
3.1.4	Velocidad de funcionamiento.....	41
3.2	Diseño preliminar.....	41
4.	DISEÑO CONCEPTUAL INTERFAZ DE SALIDA DE DATOS (SIES 2.0).....	43
4.1	Requerimientos de la interfaz de salida	44
4.1.1	Comunicación directa con el software de simulación.	45
4.1.2	Puertos de salida de datos.....	45
4.2	Diseño preliminar.....	45
5.	DESARROLLO INTERFAZ DE ENTRADA DE DATOS (SIES 1.0)	46
5.1	Descripción general de la interfaz SIES 1.0	46
5.2	Desarrollo de la interfaz SIES 1.0	46
5.2.1	Componentes utilizados.....	47
5.2.2	Implementación	49
	La estructura del <i>Report Descriptor</i> y distribución de memoria se aprecia en la tabla 5.2.....	57
5.3	Simulación del Hardware	60
5.4	Placa Base de la Interfaz SIES 1.0.....	61
6.	DESARROLLO DE LA INTERFAZ DE ENTRADA Y SALIDA DE DATOS (SIES 2.0).....	63
6.1	Software SIES (Modulo de enlace).....	63
6.1.1	Características funcionales del Software SIES.....	66
6.1.2	Base de datos para configuración de módulos.....	70
6.1.3	Etapa de configuración	73
6.2	Conexión entre el software SIES 2.0 y la tarjeta hardware SIES 2.0	79
6.2.1	Organización de los datos de transmisión y recepción.....	80
6.2.2	Procesos de comunicación.....	83
6.3	Hardware SIES 2.0.....	83
6.3.1	Características del Hardware	84
6.3.2	Firmware del microcontrolador para comunicación USB con la clase CDC.....	85
6.3.3	Diseño Hardware.....	88

7. PRUEBAS Y RESULTADOS	91
7.1 Hardware	91
Las siguientes fichas técnicas resumen los resultados de dichos procedimientos.....	91
7.2 Software.....	94
8. ANÁLISIS DE RESULTADOS	97
CONCLUSIONES.....	100
9. TRABAJO A FUTURO.....	103
GLOSARIO.....	104
BIBLIOGRAFÍA.....	108

LISTA DE FIGURAS

Figura 1.0.1. Imagen tomada del Microsoft flight simulator X.....	18
Figura 1.0.2. Imagen de un FTD nivel 7.....	22
Figura 2.1. Replica a escala de la cabina de un Boeing 747.....	32
Figura 3.1. Sistema de control de vuelo de una aeronave en un simulador.	38
Figura 3.2. Esquema del modulo para la adquisición de varias señales analógicas.	39
Figura 3.3. Estructura de los datos de acuerdo a OSI.	40
Figura 3.4. Tiempo de respuesta del sistema y tiempo de reacción del piloto.	41
Figura 3.5. Diseño preliminar de la interfaz de entrada de datos SIES 1.0.....	42
Figura 4.1. Sistema de control de pilotear una aeronave en simulación con interfaz de salida y entrada.....	44
Figura 4.2. Diseño conceptual de la interfaz de salida de datos.....	45
Figura 5.1. Interfaz SIES 1.0.....	47
Figura 5.2. Diagrama esquemático del microcontrolador 18f4550.....	49
Figura 5.3. Diagrama esquemático del circuito de la Interfaz SIES 1.0.....	51
Figura 5.4. Configuración del reloj interno del microcontrolador.....	53
Figura 5.5. Opciones de configuración de la comunicación USB en PIC18F4550.....	54
Figura 5.6. Diagrama de descriptores asociados a la interfaz SIES 1.0 y parámetros configurados..	56
Figura 5.7. Imagen de Proteus mientras se ejecuta la simulación.....	60
Figura 5.8. Diagrama del circuito impreso de la tarjeta diseñada de ambas caras para la interfaz SIES 1.0.....	61
Figura 5.9. Diseño 3D de la placa de la interfaz SIES 1.0 con todos sus componentes	62
Figura 6.1. Diseño Preliminar de la interfaz SIES 2.0.	64
Figura 6.2. Estructura del Software SIES 2.0.....	65
Figura 6.3. Ventana de bienvenida del Software SIES 2.0	65
Figura 6.4. Ventana Buscar Variables del Software SIES 2.0.....	67
Figura 6.5. Ventana principal para la configuración de variables, dispositivos de entrada y salida de datos, Software SIES 2.0.	69

Figura 6.6. Ventana para la calibración de las señales analógicas de entrada, Software SIES 2.0. ...	69
Figura 6.7. Ventana para configurar la conexión con el Hardware SIES 2.0.....	70
Figura 6.8. Estructura de la base de datos de configuración.....	71
Figura 6.9. Diagrama de Procedimientos en el Software SIES 2.0.	75
Figura 6.10. Indicador tipo 1	76
Figura 6.11. Ejemplo, Indicador tipo 2.	76
Figura 6.12. Organización de los datos de transmisión y recepción.....	81
Figura 6.13. Estructura del dato 1 para entradas digitales.	82
Figura 6.14. Estructura general de un modulo de comunicación asociado al Software SIES 2.0.	85
Figura 6.15. Diagrama de descriptores asociados a la interfaz SIES 2.0 y parámetros configurados.	87
Figura 6.16. Diagrama general del proceso de transmisión y recepción de datos.	89
Figura 6.17. Diagrama esquemático del circuito de la Interfaz SIES 2.0.....	90
Figura 6.18. Interfaz Hardware SIES 2.0.....	90

LISTA DE TABLAS

Tabla 1.1. Comparación de interfaces populares para transmisión y comunicación de datos en un PC.	24
Tabla 5.1. Características del microcontrolador	48
Tabla 5.2. Estructura del Report Descriptor	57
Tabla 6.1. Peticiones y respuestas utilizadas para la comunicación.....	83
Tabla 7.1. Conectividad.....	91
Tabla 7.2. Conectividad.....	92
Tabla 7.3. Instalación.	92
Tabla 7.4. Instalación	92
Tabla 7.5. Capacidad de Procesamiento.....	93
Tabla 7.6. Capacidad de Procesamiento.....	93
Tabla 7.7. Funcionamiento.	93
Tabla 7.8. Costo.....	93
Tabla 7.9. Estabilidad.	94
Tabla 7.10. Instalación.	94
Tabla 7.11. Reconocimiento del Hardware.....	95
Tabla 7.12. Transmisión Hardware software SIES 2.0.	95
Tabla 7.13. Escritura de las variables del FSX a través del Hardware SIES.	95
Tabla 7.14. Lectura de las variables del FSX a envió y respuesta del Hardware SIES.	96

INTRODUCCIÓN

En los últimos años con el creciente desarrollo de la tecnología, los simuladores de vuelo se han situado en una posición muy importante en la industria de la aviación, especialmente en el entrenamiento de pilotos, simulación de desastres y en el desarrollo de nuevos prototipos etc. En Colombia, la exploración en este campo es poca, Kirvit es la única empresa del país que sobresale en investigación y desarrollo en este campo.

Con el objeto de ampliar la investigación en esta área y difundir el desarrollo de entrenadores de vuelo, este trabajo plantea el diseño e implementación de un sistema de comunicación que aproveche al máximo los software de simulación disponibles al interfazar estos a una cabina de un avión. Para ello se desarrolló la interfaz hardware software SIES (Sistema Interfaz Entradas y Salidas).

El presente documento expone en primer lugar los dos primeros capítulos, el marco teórico y los antecedentes, dando unas pautas básicas para ubicar al lector en el campo de desarrollo, luego una segunda parte conformada por los capítulos 3 y 4, los cuales mencionan las funciones y requerimientos mínimos necesarios para el buen funcionamiento del sistema, además de los diseños preliminares para la implementación de cada uno de los elementos que conforman la interfaz desarrollada. Como una tercera parte, están los capítulos 5 y 6, en donde se describe la implementación de la interfaz. Describiendo los componentes que se desarrollaron y otros que se utilizaron. Una cuarta parte conformada por los capítulos 7, 8, y 9 que corresponden a las pruebas, resultados y análisis, finalmente se mencionan las conclusiones y trabajos a futuro para el presente proyecto en los capítulos 10 y 11.

1. MARCO TEÓRICO

1.1 SIMULADORES DE VUELO

Los simuladores de vuelo, son dispositivos utilizados para experimentar situaciones aéreas sin el empleo de una aeronave o nave espacial. Estos son importantes ya que permiten ahorrar tiempo y dinero en la formación de pilotos. Se utilizan en las pruebas de vuelo para investigar características como la estabilidad, el control, y comportamiento de las aeronaves o naves espaciales en diferentes escenarios, lo que permite simulaciones realistas y detalladas de todos los aspectos sin exponer a riesgo alguno el piloto.

Los simuladores de vuelo a nivel software, funcionan sobre computadoras convencionales y se caracterizan por reproducir gráficos realistas en un entorno virtual 3D y aplicación de modelos matemáticos para la simulación de la aviónica y dinámica de la aeronave. En el mercado se consiguen software de simulación de múltiples propósitos, desde aquellos que buscan el entretenimiento, hasta los que simulan con precisión cada uno de los parámetros gráficos y dinámicos de las aeronaves.

1.2 SOFTWARE DE SIMULACIÓN DE VUELO

Los simuladores de vuelo no solo han tenido una buena aceptación en el ámbito industrial por motivos profesionales o tecnológicos. La industria de los videojuegos ha jugado un papel muy importante en el desarrollo de un tipo concreto de simulador: los simuladores de vuelo para ordenador personal. Hay varios simuladores de este tipo en el mercado en función del tipo de avión que simulan, a continuación se mencionaran los referentes a aviación civil más destacados.

1.2.1 **X-Plane.**

X-Plane, es un simulador de vuelo civil, creado por Austin Meyer. Representa uno de los principales competidores de Microsoft Flight Simulator (MFS). Fue desarrollado y preparado para ser un simulador extremadamente preciso, se fundamenta en el cálculo del efecto del flujo del aire y fuerzas que actúan sobre cada uno de los elementos de superficie de los aviones simulados.

El propósito es ofrecer una experiencia de vuelo lo más realista posible. Para ello, cuenta con una gama de modelos de aviones, desde los más sencillos hasta los más grandes reactores de línea. Además cuenta con una recreación topográfica detallada de la superficie de la tierra con sus accidentes geográficos. También posee alrededor de 16.000 aeropuertos recreados en un entorno gráfico 3D para realizar las prácticas de vuelo.

Al igual que otros simuladores de esta categoría, *X-Plane* permite la conexión a Internet y el vuelo en redes gratuitas en las que se prestan servicios de control de tráfico aéreo. Las dos redes más populares son *VATSIM4* e *IVAO5*.

Asimismo, *X-Plane* cuenta con una extensa comunidad de usuarios en todo el mundo, que aportan, generalmente de manera gratuita, aviones, escenarios y otros añadidos para la mejora del simulador y la experiencia de vuelo. La Administración Norteamericana de Aviación (FAA) autorizó el uso de *X-Plane* con hardware específico para el entrenamiento de pilotos de vuelo instrumental.

1.2.2 **FlightGear**

FlightGear es un simulador de vuelo multiplataforma de código abierto. Nace de la idea de extender las posibilidades que un simulador de vuelo puede ofrecer, el problema principal de los simuladores comerciales es precisamente la falta de extensibilidad. De manera que ofrece a millones de usuarios aportar sus propias ideas para poder optimizar la simulación.

Actualmente representa una alternativa importante frente a los simuladores de vuelo comerciales. Si bien no consigue superar el alto nivel gráfico de los mejores productos comerciales, el modelo físico del vuelo y el realismo de los controles está al mismo o mayor nivel que los de sus competidores. Esto se debe a que *FlightGear* fue desarrollado desde un comienzo con un alto perfil técnico y científico.

Adicionalmente, por ser software libre son muchos los desarrollos que continuamente aportan programadores en todas las partes del mundo.

1.2.3 Microsoft Flight Simulator

Pertenece a la compañía Microsoft y fue concebido para su sistema operativo Windows. Está considerado actualmente uno de los mejores simuladores de vuelo que hay a nivel de usuario.

Flight Simulator es uno de los simuladores de vuelo más usados a nivel mundial, tanto por pilotos profesionales, aficionados y aprendices. Cuenta con instructor de vuelo, con el cual los usuarios pueden adquirir los conocimientos fundamentales para pilotar un avión. Contiene una detallada meteorología, topografía, sistema de fallos del avión y la visual interna de las cabinas de los aviones existentes en el FS.

La versión más reciente de este simulador es Microsoft Flight Simulator X (FSX).

1.3 ENTRENADORES DE VUELO

Los entrenadores de vuelo o FTD (*Flight Training Device*) por sus siglas en inglés, son dispositivos empleados en el entrenamiento de pilotos a nivel profesional, en ellos se aprende, se adquieren destrezas y se practica constantemente, sobre cómo operar los diferentes sistemas de una aeronave ya sea a nivel general o de forma específica. Existen diferentes tipos de FTD de acuerdo a su complejidad y su aplicación, se

clasifican en siete niveles, de acuerdo a los lineamientos trazados por la FAA (*Federal Aviation Administration*) en la circular AC 120 -45 para su uso a nivel profesional.

Figura 1.0.1. Imagen tomada del Microsoft flight simulator X



Fuente. Microsoft Flight Simulator X.

1.3.1 Clasificación de entrenadores de vuelo

1.3.1.1 *Entrenadores de vuelo de nivel 1*

En el nivel 1 se encuentran todos los dispositivos para entrenamiento de pilotos aprobados años atrás que aun se encuentran en uso y son conocidos como GTD (*Ground Training Devices*).

1.3.1.2 **Entrenadores de vuelo de nivel 2**

El propósito de los dispositivos de entrenamiento del nivel 2 es de permitir el aprendizaje, desarrollo, práctica de las destrezas y los procedimientos necesarios para comprender y operar los sistemas integrados de la cabina de un conjunto de aeronaves.

Los dispositivos de entrenamiento de vuelo de nivel 2 tienen las siguientes características y componentes:

- Instrumentos, equipo, paneles, sistemas y controles suficientes para el entrenamiento, localizado en una correcta configuración espacial, puede estar en una cabina o un espacio abierto. El funcionamiento de controles e interruptores deben replicar los de un aeroplano.
- El dispositivo debe simular la aerodinámica, el control de fuerzas, de tal manera que permita realizar vuelos manuales y acercamientos por instrumentos.

1.3.1.3 **Entrenadores de vuelo de nivel 3**

El propósito de los FTD de nivel 3 es el mismo de los de nivel 2, la única adición es que estos dispositivos cuentan con una cabina y con controles de navegación, visualizadores e instrumentos.

1.3.1.4 **Entrenadores de vuelo de nivel 4**

El propósito de los dispositivos de entrenamiento del nivel 4 es de permitir el aprendizaje, desarrollo, práctica de las destrezas y procedimientos necesarios para comprender y operar los sistemas integrados de una aeronave específica.

Los dispositivos de entrenamiento de vuelo nivel 4 tienen las siguientes características y componentes:

- Una cabina con réplica del panel de vuelo con los interruptores, controles e instrumentos correspondientes en una relación correcta respecto a la aeronave representada.
- Indicadores del sistema que respondan apropiadamente a los interruptores y mando de control que deben estar correctamente instalados para que el entrenamiento se realice satisfactoriamente.

1.3.1.5 **Entrenadores de vuelo de nivel 5**

El propósito de los dispositivos de entrenamiento del nivel 5 es de permitir el aprendizaje, desarrollo, práctica de las destrezas y los procedimientos necesarios para comprender y operar los sistemas integrados de una aeronave especifican en operaciones típicas de vuelo en tiempo real.

Los dispositivos de entrenamiento de vuelo de nivel 5 tienen las siguientes características y componentes:

- Una cabina con réplica del panel de vuelo con los interruptores, controles e instrumentos correspondientes en una relación correcta respecto a la aeronave representada.
- Indicadores del sistema que respondan apropiadamente a los interruptores y mando de control que deben estar correctamente instalados para que el entrenamiento se realice satisfactoriamente.
- Debe simular las capacidades aerodinámicas de la aeronave representada.
- Controles de vuelo y navegación funcionales, junto a visualizadores e instrumentos.
- Control con precisión de las fuerzas para un óptimo vuelo manual y acercamiento por instrumentos.

1.3.1.6 **Entrenadores de vuelo de nivel 6**

El propósito de un FTD nivel 6 es el de permitir el aprendizaje, desarrollo, practica de las destrezas en procedimientos en cabina de vuelo, maniobras simétricas, y los procedimientos necesarios para comprender y operar los sistemas integrados de una aeronave especifican en operaciones típicas de vuelo en tiempo real.

Los dispositivos de entrenamiento de vuelo de nivel 6 tienen las siguientes características y componentes:

- Una cabina con réplica del panel de vuelo con los interruptores, controles e instrumentos correspondientes en una relación correcta respecto a la aeronave representada.
- Una réplica de la cabina de la aeronave simulada con todos los elementos funcionales, controles, paneles, visualizadores e instrumentos.
- Debe simular las capacidades aerodinámicas de la aeronave representada tanto de operaciones en tierra como en vuelo.
- Control con precisión de las fuerzas para un óptimo vuelo manual y acercamiento por instrumentos correspondiente a la aeronave simulada.
- Controles para el instructor.

1.3.1.7 **Entrenadores de vuelo de nivel 7**

El propósito de un FTD nivel 7 es el de permitir el aprendizaje, desarrollo, practica de las destrezas en procedimientos en cabina de aeronaves, procedimientos y maniobras de vuelo, cuentan con todas las características necesarias para la operación de los sistemas integrados de una aeronave especifica en típicas operaciones aéreas.

Los dispositivos de entrenamiento de vuelo de nivel 7 tienen las siguientes características y componentes:

- Representación del sistema, interruptores, y controles que son requeridos del tipo de aeronave están aprobados por un programa de entrenamiento.
- Los sistemas deben responder apropiadamente y con la precisión que los hacen los controles de la aeronave simulada.
- Replica a escala completa de la cabina de la aeronave simulada.
- Correcta simulación de la aerodinámica y las características dinámicas de la aeronave.
- Correcta simulación de los efectos ambientales seleccionados, con los respectivos efectos que cause este sobre la aeronave.
- Control de fuerzas, dinámica y vuelo correspondiente aeronave simulada.
- Silla y control del instructor.

Figura 1.0.2. Imagen de un FTD nivel 7.



Fuente. <http://www.boeing.com>, 11/05/2009

1.4 CONCEPTO DE INTERFAZ

Una Interfaz se puede considerar como el nexo o medio de interconexión hardware o software que facilita la comunicación entre diferentes sistemas o dispositivos. A nivel de interfaces electrónicas, no existe un interfaz universal, existen diferentes estándares (Interfaz USB, IEEE1394, etc.) que establecen especificaciones técnicas concretas, para aplicaciones específicas o generales. En materia de hardware los términos puerto, bus de datos, slot, etc. En ocasiones hacen referencia a interfaces.

Una interfaz debe caracterizarse por ser rápida, versátil y fiable, depende de su uso, una interfaz se puede clasificar como de entrada, salida o bidireccional. Un computador emplea diferentes tipos de interfaces para la comunicación con los diferentes periféricos, hay una interfaz de salida a periféricos de despliegue visual como lo son una pantalla, o interfaces a periféricos de entrada como lo son teclados o mouse, o bidireccionales como las presentes en los dispositivos de almacenamiento de datos.

Existen múltiples interfaces, puertos o protocolos de comunicación populares en un computador los cuales se detallan en la tabla 1.1.

1.4.1 Bus Serie Universal (USB)

USB es una interfaz para transmisión de datos y distribución de energía que ha sido introducida en el mercado de PC y periféricos para mejorar las lentas interfaces serie (RS-232) y paralelo. Es un protocolo de comunicación para diferentes tipos de periféricos con la flexibilidad de mejorar de acuerdo al surgimiento de nuevas necesidades en el mercado.

Desde el punto de vista del usuario el puerto USB tiene las siguientes características.

- El USB es versátil. Es una interfaz para muchos tipos de dispositivos, en lugar de tener un tipo de conector y soporte hardware diferente para cada dispositivo él USB utiliza una interfaz para todos.

Tabla 1.1. Comparación de interfaces populares para transmisión y comunicación de datos en un PC.

INTERFACE	FORMATO	NUMERO MAXIMO) DE DISPOSITIVOS	VELOCIDAD (MAXIMA BITS/SEC)	USO TIPICO
USB	Asíncrono serial	127	1.5M, 12M, 480M	Mouse, Teclado, audio, pen-drives, cámaras fotográficas, impresora y periféricos estándar o especiales
Ethernet	Serial	1024	10G	Conexiones de Red
IEEE-1394a	Serial	64	400M	Audio y Video
IEEE-1394b	Serial	64	3.2G	Audio y Video
IEEE-488 (GPIB)	Paralelo	15	8M	Instrumentación
IrDA	infrarrojo serial	2	16M	Manos libres, impresoras
I ² c	Asíncrono serial	40	3.4M	Microcontroladores
Microwire	Asíncrono serial	8	2M	Microcontroladores
Puerto Paralelo	Paralelo	2	8M	Impresoras, scanner
RS-232	Asíncrono serial	2	20k	Modem, mouse, instrumentación
RS-485	Asíncrono serial	32	10M	Adquisición de datos y sistemas de control
SPI	Asíncrono serial	8	2.1M	Microcontroladores

Fuente. USB Complete, tercera edición- pág. 3.

- **Configuración automática.** Cuando un usuario conecta un periférico a un PC Windows detecta el periférico y carga el controlador apropiado y si este no está disponible solicita al usuario el disco de instalación.
- **Fácil de conectar.** No es necesario abrir el PC para conectar el dispositivo, cada computador tiene puertos USB que pueden ser expandidos empleando HUB (concentrador) permitiendo la conexión de múltiples dispositivos USB.
- **Simple conexión.** Los conectores USB son como una llave que no se pueden conectar de forma incorrecta. Un cable de transmisión puede tener hasta 5m de longitud. Con HUB un periférico puede estar hasta 30m del HOST, los conectores USB son pequeños y compactos en contraste a los típicos RS232 y conectores paralelo. Para asegurar la fiabilidad de las operaciones la especificación USB incluye requerimientos detallados sobre cables y conectores que deben ser empleados.
- **Plug and play.** Se puede conectar y desconectar un periférico USB cada vez que se requiera estando el sistema encendido o apagado sin daño alguno, el sistema operativo detecta cuando un periférico es conectado y está listo para ser usado.
- **No configuración por el usuario.** El usuario no tiene la necesidad de configurar el dispositivo sobre el correcto direccionamiento del puerto y las peticiones de interrupción IRQ's, no es necesaria la instalación de tarjetas de expansión o configuración de jumper en la placa madre.
- **Recursos Hardware libres para otros dispositivos.** Usando USB para muchos periféricos es posible tener líneas IRQ's libres para dispositivos que las requieran, el PC dedica una serie de direcciones de puertos y una línea IRQ para el *Host Controller*.
- **No requiere alimentación externa algunas veces.** La interfaz USB incluye una fuente de energía y línea tierra que proporciona 5 voltios nominales a partir de la fuente de energía del PC. Un periférico que requiera más de 500mA puede usar una fuente externa.

- **Velocidad.** El bus USB soporta tres velocidades, *High speed* a 480Mb/s, *Full speed* a 12Mb/s y *Low speed* a 1.5Mb/s. los *USB Host controller* en computadores recientes soportan las tres velocidades. La velocidad del bus USB describe la razón de cambio de la información en el bus, en adición de los datos el bus puede llevar señales de estado, control y chequeo de error, de todos los periféricos conectados, por tanto la tasa de transferencia de datos de un periférico individual puede ser menor que la velocidad del bus. La razón de cambio máxima teórica para un dato único transferido es alrededor de 53Mb/s a *High speed*, 1.2Mb/s a *Full speed* y 800b/s a *Low speed*.
- **Fidelidad.** La fiabilidad del USB está basada en el hardware y el protocolo para la transferencia de datos. La especificación del USB para los conductores receptores y cables aseguran una interfaz que elimina la mayor cantidad de ruido, el protocolo USB habilita la detección de errores al recibir los datos notificando al que envía sobre la retransmisión. La detección, notificación y retransmisión están en hardware y no requiere ninguna programación o intervención del usuario.
- **Bajo costo.** Incluso siendo el USB más complejo que las más recientes interfaces los componentes y cables son económicos, un dispositivo con interfaz USB es mucho más económica que su equivalente con puerto serial, paralelo o con la más reciente IEEE 1394.

1.4.1.1 **Clases USB**

Gran cantidad de dispositivos USB tienen mucho en común con otros dispositivos que desempeñan funciones similares. Todo *mouse* envía información acerca de sus movimientos y estado de sus botones, todas las unidades de disco transfieren archivos, todas las impresoras reciben datos de impresión y envían información de su estado al *HOST USB*.

Debido a que un grupo de dispositivos o interfaces comparten muchos atributos o proporcionan peticiones similares, se han definido ciertos atributos y servicios comunes en clases específicas.

Estas clases específicas sirven como una guía para desarrolladores que diseñan y programan dispositivos en la clase y para programadores que escriben *drivers* para el sistema anfitrión USB que se comunica con los dispositivos. Los sistemas operativos proveen drivers para las clases comunes, eliminando la necesidad de que los fabricantes tengan que suministrar los controladores junto a los dispositivos. Cuando un dispositivo soportado en una clase con características únicas o habilidades no incluidas en el controlador de la clase, es necesario que el fabricante suministre el controlador específico.

Las clases comunes para el protocolo USB definidas con su propia especificación son:

- **Audio**, especial para dispositivos que envían y reciben datos de audio, generalmente contienen datos codificados de voz, sonido, música, etc.
- **Communication**, especial para dispositivos como módems, teléfonos, dispositivos de red de velocidad media y puertos COM.
- **Chip/Smart Card Interface**, especial para tarjetas de llamadas telefónicas, tarjetas de regalo, tarjetas de llave de acceso, etc.
- **Content Security**, es una especificación definida para controlar el acceso a archivos de datos, música o videos transmitidos por el bus.
- **Device Firmware Upgrade**, define un protocolo para habilitar al anfitrión USB, para introducir optimizaciones o parches al firmware de los dispositivos USB.
- **Human Interface (HID)**, especial para dispositivos que son usados por humanos para controlar las operaciones de sistemas computarizados como mouse, teclado, etc.
- **IrDA Bridge**, (Infrared Data Association), está definida para el intercambio de datos a distancia corta vía infrarrojo, es un Puente entre el intercambio de datos USB e infrarrojo.
- **Mass Storage**, especial para dispositivos de transferencias de archivos en una o ambas direcciones, Típicamente usada en CD, DVD, Discos Duros, Memoria Flash, etc.

- **Printer**, especificación especial para dispositivos de impresión.
- **Still Image Capture**, especial para dispositivos de captura de imágenes (no video) y Scanners.
- **Test and Measurement**, especificación especial para dispositivos de instrumentación, donde los datos no necesitan ser almacenados en el tiempo.
- **Video**, especial para video cámaras, webcams, y otros dispositivos que envían, reciben o manipulan imágenes en movimiento.

1.4.1.1.1 Clase HID (*Dispositivos de Interfaz Humana*)

Los dispositivos de la clase de Interfaz Humana (HID) consisten principalmente de dispositivos que son usados por humanos para controlar las operaciones de sistemas computarizados. Típicos ejemplos de dispositivos de la clase HID son:

- Teclados y dispositivos señaladores como por ejemplo, mouse y joystick
- Control de paneles frontales, por ejemplo perillas, switches, botones y ejes deslizantes
- Controles que podrían ser encontrados en dispositivos tales como teléfonos, controles de realidad virtual, dispositivos de juegos o simulación, como por ejemplo guante de datos, aceleradores, volantes, timón y pedales.
- Dispositivos que no requieren la interacción con el hombre pero proveen datos en similar formato para dispositivos de la clase HID, por ejemplo un lector de código de barras, termómetro o voltímetro.

Muchos dispositivos de la clase HID incluyen indicadores, display especializados, *audio feedback* y *force-feedback*. Por lo tanto la clase HID incluye soporte para varios tipos de salidas para el usuario final.

Debido a su amplio uso e importancia de estos dispositivos en la interacción hombre – PC, la mayoría de los dispositivos USB que emplean la clase HID no necesitan de controladores, ni asistentes para su adecuada instalación y configuración, son dispositivos *Plug & Play*, de instalación automática, y compatibles con los diferentes sistemas operativos actuales, y con cualquier software que necesite acceder a estos de forma directa.

1.4.1.1.2 Clase CDC (*Communication Device Class*)

La clase de comunicación de dispositivos CDC abarca dos amplios tipos de dispositivos, los teléfonos y los dispositivos de red de velocidad media. En los teléfonos incluyendo a los análogos y módems, Adaptadores terminales ISDN y teléfonos digitales.

Los dispositivos de red incluyen módems ADSL, Cable módems, y 10BASE-T Adaptadores de Ethernet. La interfaz USB con un dispositivo de comunicación, típicamente se encarga del transporte de datos que usan aplicaciones específicas, protocolos como son V.250 para control de modem o Ethernet para el tráfico de una red local. La clase de comunicación de dispositivos es también una opción para otros dispositivos que usan como vía de acceso puertos COM.

Para el uso de dispositivos que emplean esta clase es necesaria una adecuada selección del controlador de acuerdo a la función del dispositivo y al sistema operativo a emplear, dispositivos como los módems pueden usar controladores como el *usbser.sys*. Para la habilitación de aplicaciones de acceso a dispositivos usando puertos COM (Un puerto virtual como el RS-232), puede requerir controladores específicos de diferentes fabricantes o requerir un controlador diseñado especialmente para la aplicación.

1.4.1.2 *USB vs IEEE-1394*

Otra interfaz disponible para nuevos periféricos es IEEE-1394. *Apple Computer* implementó esta interfaz y la denominó *Firewire*. Generalmente IEEE-1394 puede ser más rápida y flexible que USB pero es más costosa de implementar. Con USB, un único HOST

controla la comunicación con múltiples dispositivos, el HOST opera la mayoría de las operaciones complejas, la electrónica de los dispositivos es más sencilla y más económica. Los dispositivos IEEE-1394 pueden comunicarse con otros directamente y una única comunicación puede estar conectada a múltiples receptores. El resultado es una interfaz más flexible, pero los dispositivos electrónicos son más complejos y costosos.

IEEE-1394 es una interfaz especial para aplicaciones que requieren extremadamente velocidad y ancho de banda de comunicación para múltiples receptores. USB es la mejor para periféricos de uso común como teclados, impresoras y scanner, también como aplicaciones de baja y moderada velocidad.

2. ANTECEDENTES

Desde la antigüedad han existidos diferentes métodos militares para simular situaciones y escenarios de combate, la llegada del computador ENIAC en 1946 fue la base para la reformulación de la simulación hasta llegar a las maquinas computacionales de la actualidad en donde se puede realizar prácticamente cualquier tipo de simulación. Gracias a los avances científicos y tecnológicos la banda que separa la realidad con la simulación es cada día más estrecha, además factores como economía y seguridad convierten a la simulación una herramienta necesaria en cualquier campo de investigación.

La simulación de vuelo empezó de la mano con la aviación, ya que siempre se tuvo muy presente al riesgo al que se sometían los primeros pilotos, por tal razón diferentes modelos fueron desarrollados, como por ejemplo el “*Sander Teacher*” con un solo eje de movimiento, en 1910 se creó un simulador que fue básicamente un barril montado en un marco. El “*Link Trainer*” creado en 1930, inicialmente simulaba movimientos mecánicos y posteriormente se le adicionaron mandos de control el cual fue utilizado durante la segunda guerra mundial. Desde 1954 con la creación de un simulador de vuelo de 3 grados de libertad por “*Singer Corporation*”, el cual era una cabina montada en un marco, pasando por 1969 en donde se construyeron simuladores de aerolíneas con actuadores hidráulicos controlando cada eje de movimiento (6 ejes), hasta llegar a 1977 en donde los computadores estaban instalados en las cabinas, los simuladores de vuelo estaban más cerca a la realidad.

En la actualidad existen una gran cantidad de simuladores desarrollados por profesionales, empresas o simplemente por aficionados, además en internet se consigue información detallada de cómo construir una cabina y realizar las conexiones necesarias, es posible comprar los diferentes módulos de un simulador de vuelo, aunque comprar todos las secciones necesarias puede alcanzar unos 25000 US, también es posible encontrar descargas gratuitas de diferentes software de simulación con buenas características.

2.1 DESARROLLOS NO FORMALES

La pasión por volar siempre estará presente en el hombre, muchos son los aficionados que sueñan con volar una aeronave, e inconformes con hacerlo con un joystick y un computador han llegado al punto de fabricar replicas de aviones comerciales a escala 1:1 para hacer prácticas de vuelo simulado. En España existe una comunidad de aficionados con proyectos de construcción de entrenadores de vuelo en torno a las herramientas que en esta empresa se ofrecen, <http://www.opencockpits.com/> logrando construir replicas del avión Boeing 747 como se aprecia en la figura 2.1.

Figura 2.1. Replica a escala de la cabina de un Boeing 747.



Fuente: <http://www.opencockpits.com/>- 11/05/2009

Opencockpits ha desarrollado unas placas llamadas IOcards y un software llamado SIOC los cuales en conjunto permiten construir casi que cualquier tipo de FTD, estas tarjetas son muy conocidas por aficionados, aunque son muy completas en donde se han

tenido en cuenta muchos detalles, y habilitado una gran cantidad de funciones para facilitar la construcción de un simulador de vuelo, cabe mencionar que su arquitectura es 8 bits.

En Estados Unidos, las comunidades de aficionados a la simulación de vuelo giran en torno a <http://www.flightsim.com> que es la comunidad más grande en internet para los amantes de la simulación de vuelo y SIMKITS que es una empresa que construye y suministra partes para la fabricación de entrenadores de vuelo <http://www.simkits.com/>.

Colombia no es indiferente en este campo, dentro del territorio se pueden destacar dos empresas; Kirvit Ltda., nació en el año 2001 con el objetivo de formalizar la investigación y desarrollo en el campo de la simulación en la aviación brindando capacitación en temas de aeronáuticos y formación de pilotos virtuales. Dentro de sus desarrollos se encuentra el programa adicional para el producto de Microsoft Flight Simulator conocido como *Latin American Airports Andean Edition*, consolas básicas para entrenamiento de pilotos de aeronaves monomotor y bimotor, el *Home Flight Simulator* el cual es un equipo para entrenamiento de aeronaves tipo jet y otras aplicaciones, cabe destacar la construcción de un FTD para la fuerza aérea colombiana por un valor de \$ 200.000.00 en el año 2008.

También se encuentra SIMUTRONIK EU ubicada en la ciudad de Bogotá, se dedica al diseño y construcción de equipos electrónicos para aplicaciones industriales, entre sus desarrollos más significativos en el campo de la simulación de vuelo se destaca un *Flight Training Device* nivel 5, construido para la empresa Adevia Ltda.

2.2 DESARROLLOS I+D EN UNIVERSIDADES.

A continuación se describen diferentes trabajos que de forma directa o indirecta están relacionados con este proyecto.

2.2.1 Diseño y construcción de simuladores a partir de aviones fuera de servicio para parques de diversión.

Este trabajo, desarrollado en el año 2005 en la universidad EAFIT plantea la construcción de simuladores de vuelo para pasajeros, desde la panorámica de los aviones de transporte comerciales. Para ello, sugieren la adecuación de aviones fuera de servicio donados por compañías aéreas a diferentes parques de diversiones del país.

Su diseño, en forma general, consistió en adecuar pantallas de video para cada una de las ventanillas de la aeronave, configuración de un adecuado sistema de sonido e implementación de un sistema de control hidráulico que inclina la estructura del avión en diferentes ángulos.

2.2.2 Interfaz 3D para un vehículo aéreo no tripulado

La Agencia en Tecnología de Defensa (DTA) de Nueva Zelanda está desarrollando un control remoto para el control de una aeronave que provea el reconocimiento para unidades en el campo.

Este está soportado en una interfaz 3D de tiempo real para el uso del control y planificación de misiones. La interfaz usa un software de simulación de dominio público (*Flight Gear*) e integra datos de terreno obtenidos por *Joint Geospatial Facility* (JGSF) y datos geográficos de *Geographical Information Systems* (GIS). Adicionalmente, el diseño de la interfaz está habilitado para funcionar con el sistema virtual marítimo con el fin de ser usado a gran escala de forma multiusuario.

2.2.3 Diseño de un simulador de tren

Este trabajo desarrollado en la Universidad Pontificia Comillas de España, presenta el diseño de un simulador de tren, el cual simula la conducción de un tren de tracción eléctrica, desde la puesta en marcha de sus equipos hasta la parada total por orden del usuario o freno de emergencia, así como la apertura y cierre de puertas, se hizo todo un

desarrollo software tanto para la representación grafica, como para a lógica del vehículo, acompañado de un hardware para el control de este a partir de una cabina similar a la de un tren.

Este proyecto nació con el fin de ofrecer una herramienta para complementar la instrucción de conductores de trenes eléctricos debido a la expansión de este medio de transporte en España.

2.2.4 Relay, Una interfaz futurista para la conducción remota

El mando a distancia de vehículos, típicamente útil en las situaciones peligrosas, es una tarea difícil. Los usuarios experimentan la gran dificultad in llevar a cabo las operaciones relativamente simples, como evitar obstáculos y mantener actualizado el control del dispositivo. Esto contrasta con el rendimiento de un conductor al operar un vehículo real. Sugerimos que una razón para esto sea la falta de la realimentación del mando expresando las experiencias físicas del vehículo, la fuerza que llevan a cabo actuación sobre el cuándo se mueve. Este trabajo plantea el diseño de un radio control retroalimentado que permita percibir la fuerza y torques experimentadas cuando un automóvil es conducido. Para desarrollo del prototipo emplearon básicamente microcontroladores, interfaz USB, radio transmisores, acelerómetros y servo motores, entre otros elementos.

2.3 OTROS DESARROLLOS DE INTERFACES ELECTRÓNICAS SIMILARES EN OTROS CAMPOS

2.3.1 Desarrollo de una interfaz HID, para aplicación en un simulador virtual de Laparoscopia.

Dado que la cirugía abdominal requiere un alto grado de destreza en el médico. En el mercado existen diversos equipos para realizar este entrenamiento a un elevado costo,

por este motivo investigadores en la universidad de Queen's, Reino Unido construyeron una interfaz a partir del desarrollo de dispositivos de Interfaz Humana HID, para ser usado con un software de simulación virtual y realizar prácticas de cirugía.

2.3.2 RC Joystick NG2

Este proyecto es básicamente una interfaz entre un radio de control de UAV (aeronave no tripulada) y un computador vía USB, el cual permite que el PC reconozca al radiocontrol como un dispositivo de juego

2.3.3 Desarrollo de una estación meteorológica USB

Consistió en la construcción de una tarjeta hardware que fuera capaz de adquirir y procesar la información proveniente de diversos sensores, además de controlar la comunicación USB con el *host* y un software que diera una adecuada visualización de la información adquirida.

2.3.4 Desarrollo de un sistema hardware software que mejore la funcionalidad de un ordenador personal para personas con discapacidad

Este trabajo consistió en el desarrollo de un hardware y software con funcionalidades especiales que proporcionaba a una persona discapacitada mayor libertad y autonomía para el acceso a ordenadores personales, mejorando su calidad de vida.

3. DISEÑO CONCEPTUAL INTERFAZ DE ENTRADA DE DATOS (SIES 1.0)

Un avión tiene una serie de indicadores, los cuales dan la información del estado actual de la aeronave, estos datos son utilizados por el piloto para tomar decisiones y ejercer maniobras de diferente tipo según se requiera. El piloto tiene a su disposición los mandos de control para cambiar el estado de la aeronave, es decir con estos controles realiza las diferentes maniobras sobre su nave aérea, creándose de esta manera un sistema en lazo cerrado.

Cuando se lleva este sistema a la simulación aparecen 3 partes muy importantes. El primero es el computador; sobre el cual se soporta el software de simulación y al mismo tiempo despliega información visual de la aeronave, de segundo están las replicas de los mandos de control; las cuales captan la información sobre las maniobras que el piloto ejerce al avión, y por último esta la interfaz (Objetivo de este proyecto) la cual toma todos los datos de los mandos de control, los procesa convirtiéndolos a un formato que el computador entienda siendo reconocidos por el software de simulación. En la figura 3.1 se observa el sistema de control de vuelo de una aeronave en un simulador.

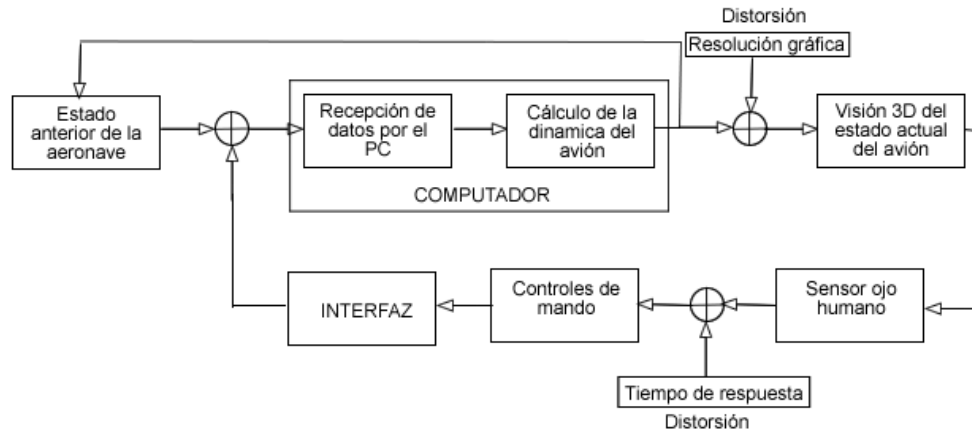
Como el objetivo del presente proyecto es la construcción de la interfaz, no se va a profundizar en los mandos de control ni en cómo funciona el computador en conjunto con el software de simulación.

3.1 FUNCIONES Y CARACTERÍSTICAS DE LA INTERFAZ.

La interfaz debe ser capaz de adquirir, procesar, encriptar y enviar la información perteneciente a los mandos de control de la aeronave hasta el software de simulación. Con

el fin de que la interfaz funcione correctamente debe cumplir al menos los siguientes requerimientos.

Figura 3.1. Sistema de control de vuelo de una aeronave en un simulador.



Fuente. Autores.

- Tener un modulo de adquisición y procesamiento de datos.
- Compatible con el sistema operativo en donde este el software de simulación.
- Reorganizar la información en un formato legible para el sistema operativo.
- Tener conectividad con el computador.
- Su velocidad de funcionamiento total debe ser por lo menos 2 veces mayor que la velocidad de respuesta del sistema a simular.

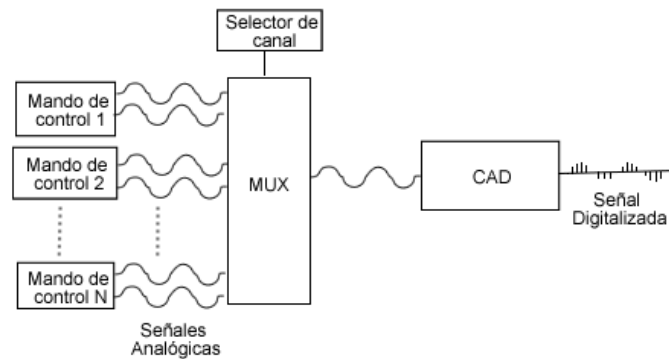
3.1.1 Modulo de adquisición y procesamiento de datos.

El modulo de adquisición y procesamiento de datos esta dividido en 2 partes, la que concierne a la adquisición de señales analógicas y las señales binarias.

Para la adquisición de las señales analógicas es necesario un CAD (convertidor analógico digital), el cual toma la señal analógica proveniente de los controles de mando y la convierte en un numero binario, además es necesario un multiplexor que sea capaz de tomar varias señales y pasarlas a través del CAD, esto es debido a que los mandos de control tienen como respuesta más de una señal analógica. La cantidad de mandos de control estará limitada por la capacidad de la interfaz.

Para la adquisición de las señales binarias la interfaz debe tener un puerto con varias entradas en el cual se pueda hacer seguimiento de las señales on/off, como por ejemplo las de diferentes switch.

Figura 3.2. Esquema del modulo para la adquisición de varias señales analógicas.



Fuente. Autores.

3.1.2 Compatibilidad con el sistema operativo.

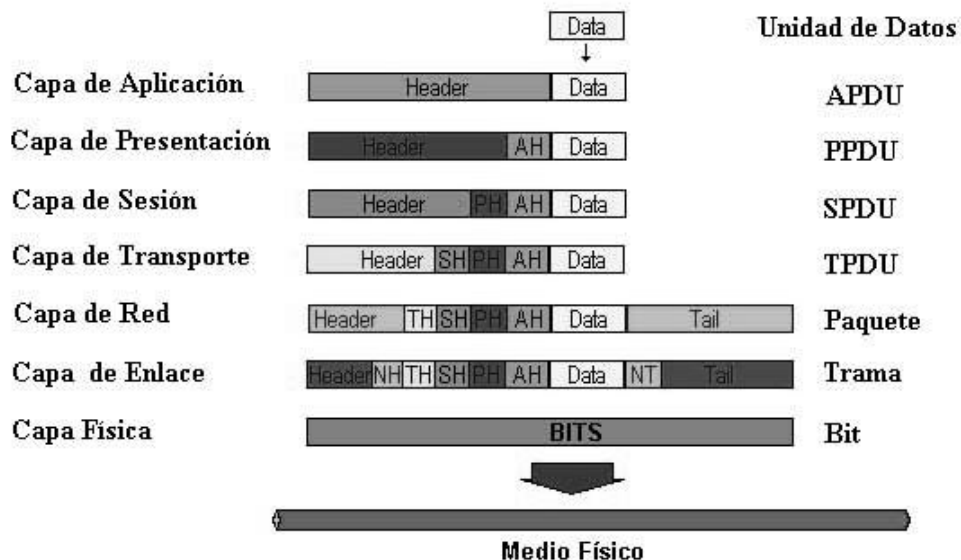
Para asegurar la compatibilidad de la interfaz con el sistema operativo presente en el computador es necesaria la implementación de un protocolo de comunicación sobre la

interfaz, esto obliga la utilización de un dispositivo programable y que tenga memoria en donde se pueda grabar las instrucciones para la ejecución del protocolo.

3.1.3 Reorganización de la información y conectividad.

De acuerdo al bus de comunicación que se utilice, para asegurar la comunicación es necesario escoger un protocolo de comunicación adecuado para organizar los datos provenientes de los mandos de control, con el cual el computador proceda a procesar la información, es decir que al momento de recibir la información esta se direcciona al software de simulación de una manera libre de errores al mismo tiempo que se hace un control de flujo entre la interfaz y el computador, para que realice las operaciones solicitadas en la simulación por el piloto. Independiente del protocolo de comunicación que se vaya a implementar, los datos deben estar organizados de acuerdo a la norma OSI que es el marco de referencia para la definición de arquitecturas de interconexión de sistemas de comunicaciones.

Figura 3.3. Estructura de los datos de acuerdo a OSI.

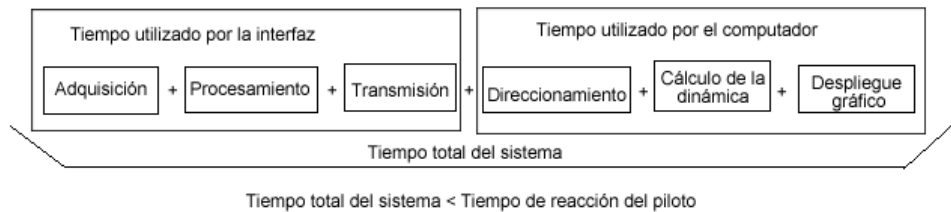


Fuente. <http://albertjh.cymaho.com/> - 11/05/2009

3.1.4 Velocidad de funcionamiento

Para complementar el realismo que los mandos de control y el software de simulación brindan a la simulación es necesario que el tiempo que la interfaz tarda en adquirir, procesar, encriptar (dar formato) y enviar la información sea muy corto, de tal forma que el tiempo total de la respuesta del sistema (Simulador) sea menor que el tiempo de reacción del piloto.

Figura 3.4. Tiempo de respuesta del sistema y tiempo de reacción del piloto.



Fuente. Autores.

3.2 DISEÑO PRELIMINAR

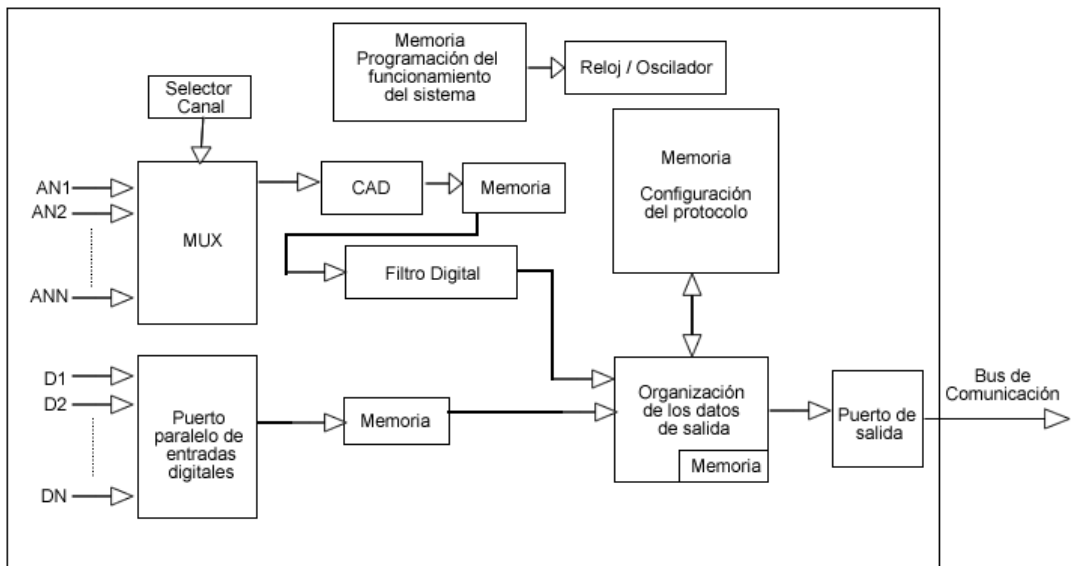
Teniendo en cuenta las características y requerimientos de la interfaz mencionada anteriormente, se considero el diseño de la interfaz sin especificar componentes hardware y software como el de la figura 3.5, de tal manera que permitiera modelar el sistema para determinar los elementos necesarios para su implementación.

En la Figura 3.5 se pueden observar otros elementos necesarios para el buen funcionamiento de la interfaz, como lo son:

- **Memorias:** Para almacenar el estado de los datos de entrada, el programa principal de la interfaz, la configuración del protocolo a implementar y para almacenar los datos de salida a espera de que el bus de transmisión este libre.

- Filtro Digital: Este filtro se plantea para las señales analógicas con el fin de eliminar el ruido provocado por la conexión entre los mandos de control y la interfaz.
- Reloj / Oscilador: se utiliza para garantizar la sincronización del sistema.

Figura 3.5. Diseño preliminar de la interfaz de entrada de datos SIES 1.0.



Fuente. Autores.

4. DISEÑO CONCEPTUAL INTERFAZ DE SALIDA DE DATOS (SIES 2.0)

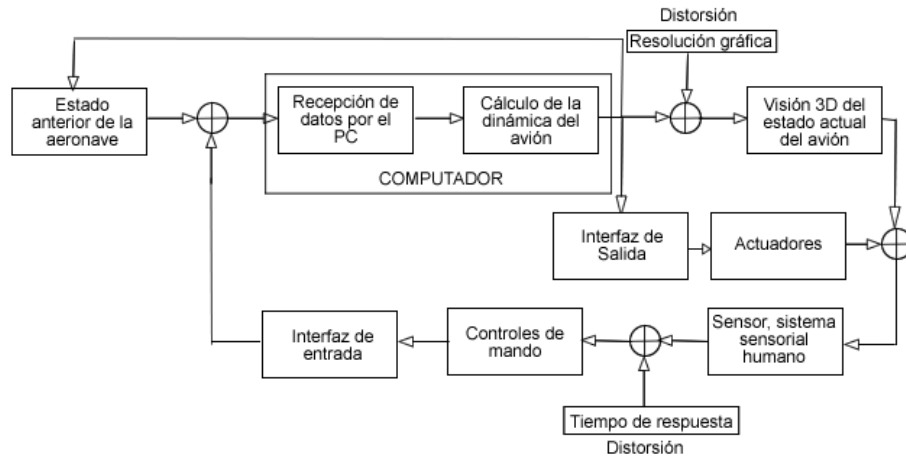
Un simulador de vuelo cuenta con los mandos de control para manipular el estado del avión, pero el piloto se vale de diferentes indicadores como luces, letreros, mensajes que le brindan información sobre el estado actual de la aeronave y de acuerdo a esto tomar una decisión para proceder, por ejemplo una luz puede indicar si el tren de aterrizaje sigue afuera.

De acuerdo a lo ya expuesto, se hace necesaria la construcción de una interfaz de salida, que brinde la información (datos numéricos y tangibles) sobre el estado de la aeronave, de tal forma que el usuario pueda almacenar esta información y utilizarla según la aplicación que desee elaborar, por ejemplo; si se desea construir una plataforma *Full Motion* (plataforma que simula los movimientos espaciales de la aeronave), entonces la interfaz brinde la información de los ángulos *pitch*, *roll*, *yaw*, la velocidad vertical y cualquier otra que puede necesitar para su construcción.

En la figura 4.1 se puede apreciar un elemento nuevo aparte de la interfaz de salida, que son los actuadores, estos de acuerdo a los datos suministrados por la interfaz de salida brindaran al piloto una información sensorial más completa, optimizando el realismo en la simulación de volar cualquier aeronave. No se hará énfasis en este elemento ya que no son el propósito del presente proyecto.

Con la inserción de una interfaz de salida, el sistema de control de vuelo de una aeronave en un simulador cambiaria de la siguiente forma.

Figura 4.1. Sistema de control de pilotear una aeronave en simulación con interfaz de salida y entrada.



Fuente. Autores.

4.1 REQUERIMIENTOS DE LA INTERFAZ DE SALIDA

Para garantizar el adecuado funcionamiento de la interfaz de salida debe cumplir por lo menos con los siguientes requerimientos:

- Compatibilidad con el sistema operativo.
- Comunicación directa con el software de simulación.
- Tener puertos de salida de datos.
- El tiempo de respuesta del sistema debe ser menor que el tiempo de respuesta del piloto.

La necesidad de compatibilidad con el sistema operativo y la velocidad del sistema se sustentan en el numeral 3.1.2 desarrollo conceptual.

4.1.1 Comunicación directa con el software de simulación.

Con el propósito de tener acceso a la información que posee el software de simulación es necesario tener un modulo que permita esta comunicación. La función de este modulo seria básicamente preguntarle al usuario que tipo de información desea conocer, dirigirse al software de simulación extraer esta información para luego enviarla a un puerto de salida y finalmente sea desplegada en el formato que el usuario más le convenga.

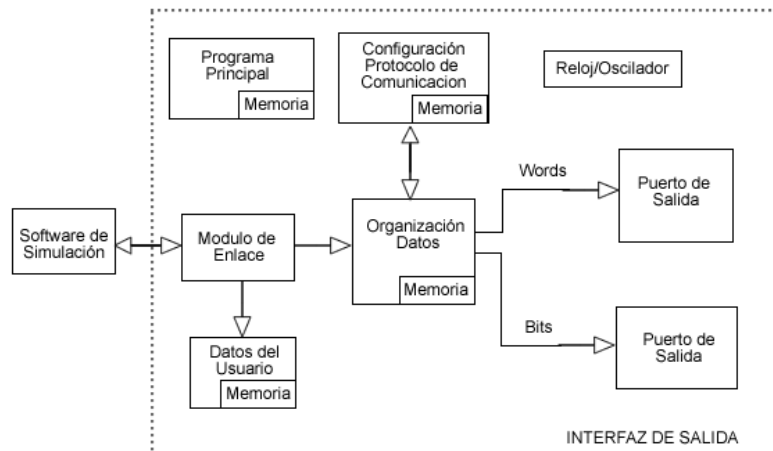
4.1.2 Puertos de salida de datos

Debido a que el usuario puede requerir por ejemplo la información sobre el estado de una luz (*on/off*) o el valor de la velocidad vertical (continuo) es necesario que la interfaz de salida tenga la capacidad de manipular estos dos tipos de datos. En otras palabras la interfaz debe ser capaz de entregar la información en *bits* y *Word* (8 bytes).

4.2 DISEÑO PRELIMINAR

Según los requerimientos planteados anteriormente sin especificar el software o hardware empleado se plantea el siguiente diseño conceptual

Figura 4.2. Diseño conceptual de la interfaz de salida de datos.



Fuente. Autores.

5. DESARROLLO INTERFAZ DE ENTRADA DE DATOS (SIES 1.0)

Los simuladores de vuelo en los cuales se soporta parte de este trabajo, son software comerciales que funcionan sobre sistemas operativos como Windows, Linux o MacOS, compatibles con los dispositivos comunes como teclados, mouse, joystick y con muchos dispositivos específicos que existen en el mercado como lo son TrackIR, consolas de mandos de vuelo, set de palancas, etc.

Partiendo del diseño preliminar figura 3.5 y de la compatibilidad necesaria con el sistema operativo mencionado en el capítulo diseño conceptual, fue necesario que la interfaz SIES 1.0 estuviera soportada en un protocolo de comunicación convencional.

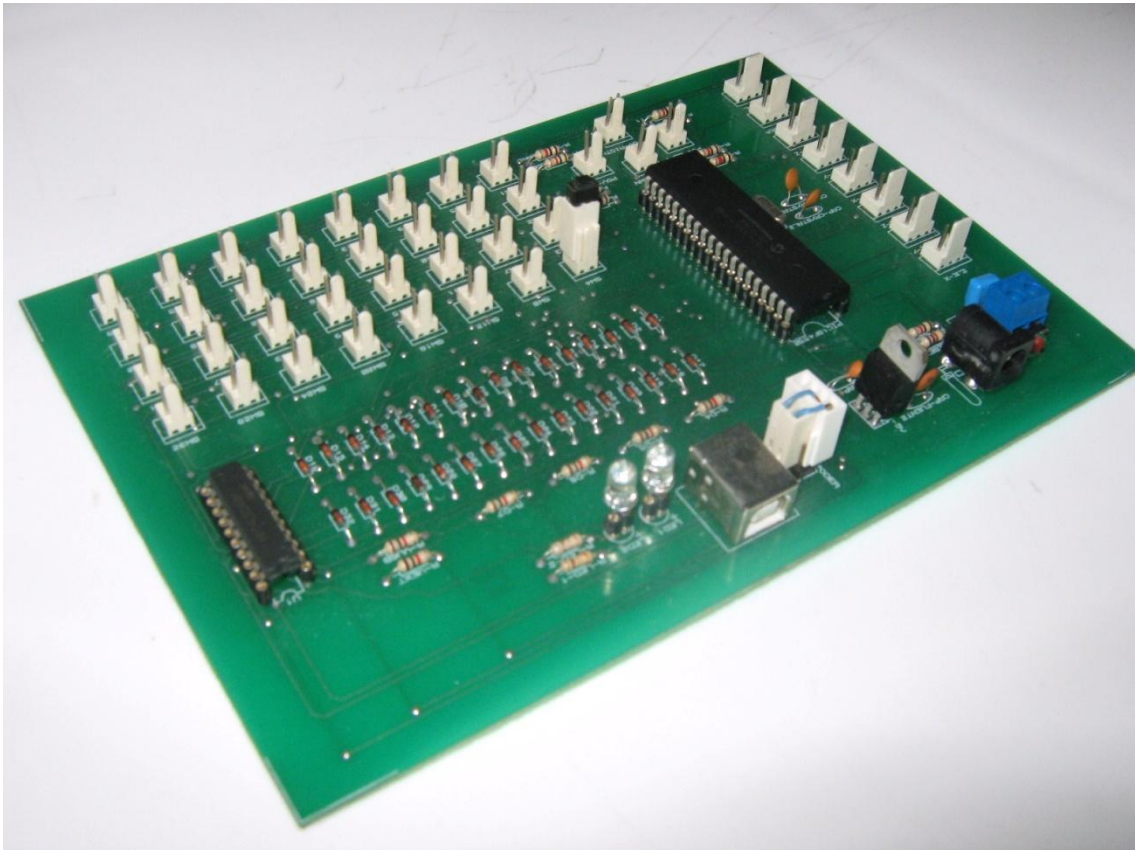
5.1 DESCRIPCIÓN GENERAL DE LA INTERFAZ SIES 1.0

La interfaz SIES 1.0, tiene conexión USB 2.0, fácil de usar, rápida (*Full Speed*), conectividad *Plug & Play*, de instalación automática, esta soportada mediante la especificación HID y emplea los drivers comunes incluidos en los sistemas operativos convencionales (Windows, Linux, Mac), es reconocida como un dispositivo de juego de 8 ejes analógicos (*X, Y, Z, Rot X, Rot y, Rot Z, Dial, Slider*), 32 botones de estado *on/off* y control del punto de mira (*up, down, left, right*) mediante cuatro botones.

5.2 DESARROLLO DE LA INTERFAZ SIES 1.0

A continuación se describe el desarrollo de la interfaz SIES 1.0, en primera instancia se describen los componentes utilizados, la implementación y pruebas realizadas sobre esta.

Figura 5.1. Interfaz SIES 1.0



Fuente. Autores

5.2.1 Componentes utilizados

Retomando la figura 3.5 del diseño preliminar en la cual se distingue los componentes esenciales de la interfaz SIES 1.0, Multiplexor, convertidor análogo a digital, memoria de programa y puertos de entrada y salida.

Existen múltiples dispositivos programables para el desarrollo de la interfaz SIES 1.0, pero se optó por un microcontrolador, ya que cuenta con todos los componentes necesarios, es económico y de fácil acceso, además de que los desarrolladores ya contaban con conocimientos previos para realizar aplicaciones en este tipo de dispositivos.

Dentro de los microcontroladores comerciales que cuentan con CPU USB interna se destacan: Microchip PIC 18F4550, Cypress EZ-USB, Cypress encoré II, Freescale MC68HC908JB16 y Freescale MCF5482 ColdFire.

5.2.1.1 **Microcontrolador PIC18f4550 de Microchip**

El microcontrolador PIC18f4550 producido por microchip fue elegido para el desarrollo de la interfaz SIES 1.0 debido a las características técnicas especificadas en la tabla 5.1, bajo costo, Firmware base desarrollado por Microchip para aplicaciones USB optimizado a USB 2.0 programable en lenguaje C mediante el MPLAB 8.0 y el *toolsuit* C18 3.2.

Tabla 5.1. Características del microcontrolador

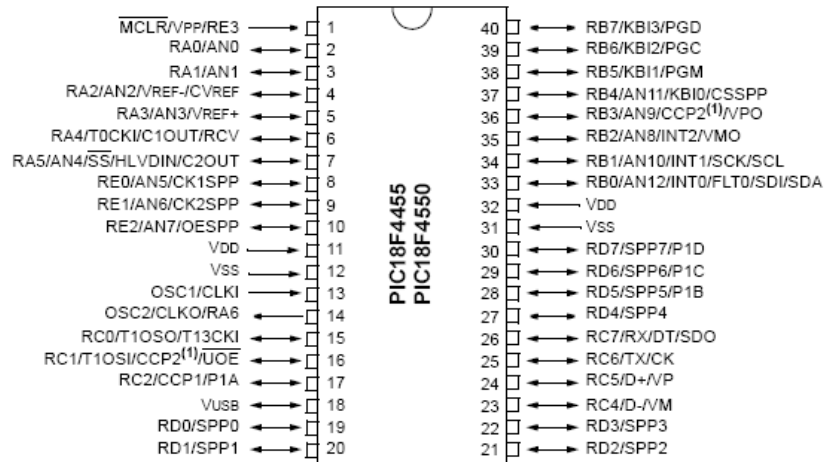
Características USB	<ul style="list-style-type: none">• Compatibilidad USB 2.0• Soporte de velocidades Low-speed (1.5 Mb/s) y Full-speed (12 Mb/s)• Soporte de Transferencias de Control, Interrupt, Isochronous y Bulk.• Soporte de 32 endpoint (16 endpoint bidireccionales)• 1-Kbyte de memoria RAM de acceso dual• Transceiver USB integrado
Características Generales	<ul style="list-style-type: none">• 2 módulos comparación y captura PWM• USART optimizada• 3-Wire SPI• I²C Master /Slave• 13 Canales analógicos y convertidor análogo a digital de 10 bits• 35 pines de entrada y salida

Fuente, PIC18F2455/2550/4455/4550 Data Sheet.

5.2.1.2 **Selección de transductores**

Para capturar los cambios en cada uno de los mandos de control analógicos, se opto por emplear potenciómetros de desplazamiento lineal, para todos los mandos de dos estados se opto por pulsadores e interruptores, adicionalmente un alto porcentaje de dispositivos de uso profesional para la simulación de vuelo usan potenciómetros para el censado de movimiento de los controles de mando.

Figura 5.2. Diagrama esquemático del microcontrolador 18f4550



Fuente PIC18F2455/2550/4455/4550 Data Sheet.

5.2.1.3 Herramientas software

- Microchip Mplab 8.0
- Mplab IDE, es el editor de código para desarrollar el *firmware* de los microcontroladores microchip. C18 Versión 3.2, es el *toolsuit* adicional que permite programar los microcontroladores PIC de la serie 18 en lenguaje C.
- PROTEUS, es un programa para simular circuitos electrónicos complejos integrando inclusive desarrollos realizados con microcontroladores de varios tipos, en una herramienta de alto desempeño con unas capacidades graficas impresionantes. Presenta una filosofía de trabajo, en donde se arrastran componentes de una barra e incrustándolos en la aplicación,

5.2.2 Implementación

Como se menciona en el marco teórico la especificación HID comprende todos los dispositivos de interfaz humana, por ejemplo el joystick es un dispositivo HID compatible

con todos los software de simulación de vuelo, es un dispositivo común y por ende los controladores HID incluidos en los sistemas operativos abarcan este dispositivo.

Los joystick comerciales cuentan con 4 ejes, 4 botones y punto de mira, el cual permite una interacción aceptable con el simulador de vuelo para controlar el avión, pero no suficiente respecto a la cantidad de controles y mandos que presenta un avión real, motivo por el cual los usuarios de estos dispositivos necesitan apoyo constante del teclado del computador.

Con el objetivo de contar con una interfaz de mayor capacidad, aprovechando la compatibilidad de los joystick con los software de simulación de vuelo y sistemas operativos, se planteó que la interfaz SIES 1.0 sea reconocida y tenga el funcionamiento de un joystick con la mayor cantidad de prestaciones permitas por los controladores. Estas prestaciones son 8 ejes analógicos (*X, Y, Z, Rot X, Rot y, Rot Z, Dial, Slider*), 32 botones de estado *on/off* y control del punto de mira (*up, down, left, right*) mediante cuatro botones.

Hay que resaltar que muchos simuladores de vuelo permiten que los controles de sus aeronaves sean manejados mediante la asignación a diferentes elementos de varios joystick, por esta razón la interfaz SIES 1.0 será de este tipo y con estas características.

Para sustentar el proceso de desarrollo, inicialmente se describirá el diseño del hardware de la interfaz SIES 1.0, posteriormente el *firmware* del microcontrolador, las simulaciones realizadas y por último imágenes del dispositivo terminado.

5.2.2.1 **Diseño hardware**

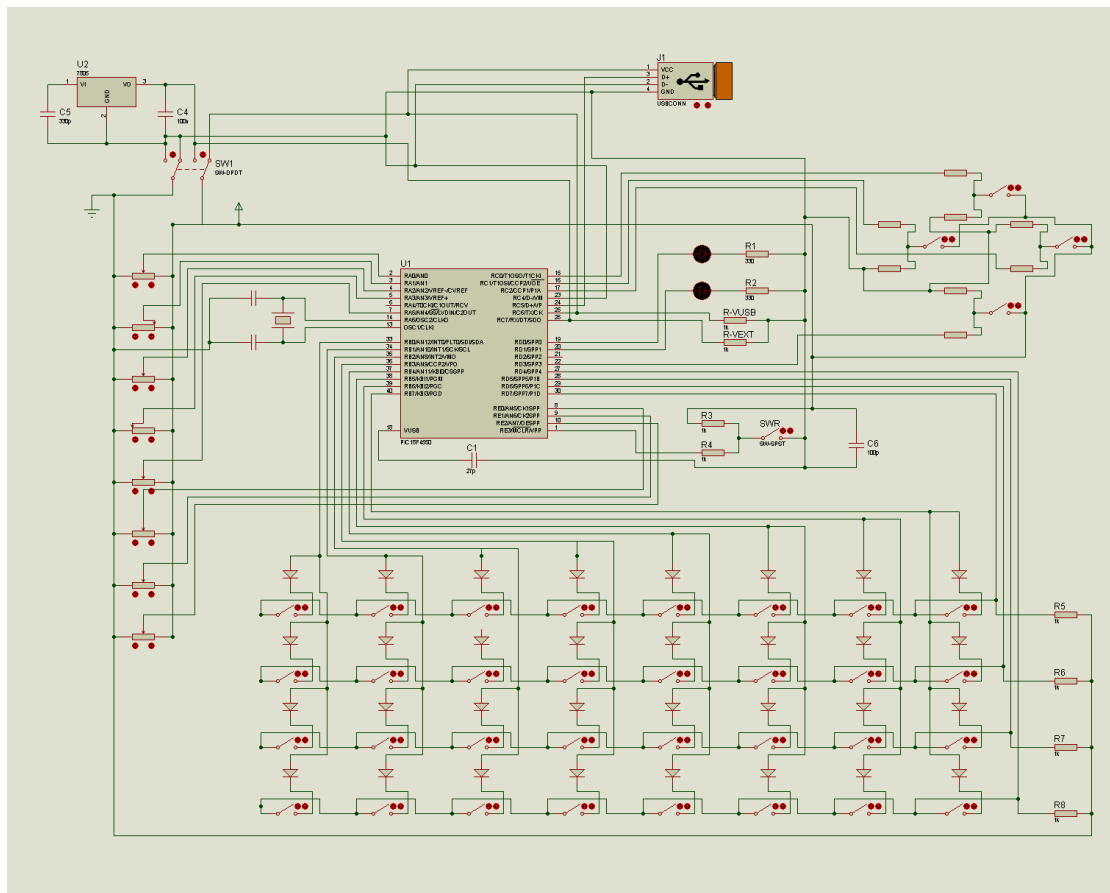
De acuerdo a los requerimientos, 8 ejes, 36 botones, se realizo el diseño del hardware de la interfaz SIES 1.0 teniendo en cuenta componentes adicionales para el monitoreo de estado de la comunicación USB.

Teniendo en cuenta que el microcontrolador tiene 35 pines de entrada y salida se diseño un teclado matricial de 8 filas 4 columnas a partir de pulsadores y diodos de rápida conmutación (1N914) de igual forma para la lectura del estado de 32 botones. Para los

ocho ejes analógicos se usaron las canales en el rango (AN0 – AN7), para los cuatro botones del punto de mira se asigno un pin a cada botón.

El diagrama esquemático fue realizado empleando la herramienta Isis de Proteus. Ver figura 5.3.

Figura 5.3. Diagrama esquemático del circuito de la Interfaz SIES 1.0.



Fuente. Autores.

Los dos diodos LED que se aprecian en el diagrama, son empleados para mostrar el estado de la comunicación USB.

Teniendo en cuenta que la interfaz SIES 1.0 se puede adaptar a las necesidades del usuario y por tanto cambiar los transductores utilizados, se dejó opcional una fuente externa de energía, la cual sería empleada en los casos que impliquen un consumo de corriente mayor a 500 mA, en caso contrario se usa la fuente de energía incluida en el puerto USB.

5.2.2.2 *Firmware del Microcontrolador*

El firmware del microcontrolador de Microchip MCHPUSB ver 1.3, es un conjunto de librerías y bloque de instrucciones de programa en lenguaje C que es usado para crear aplicaciones USB, este es grabado en la memoria ROM y establece la lógica de toda la electrónica del microcontrolador.

Como se aprecia en la figura 5.3, en el diagrama hardware de la interfaz SIES 1.0, el microcontrolador gobierna toda la electrónica, el procesamiento de datos y la comunicación de los mismos, motivo por el cual el firmware debe contar con un equilibrio entre las funciones asociadas a la transmisión de datos USB, y las funciones de adquisición y procesamiento de datos, la descripción de la configuración de los diferentes componentes se aprecia a continuación.

5.2.2.2.1 *Configuración de los puertos de entradas y salidas*

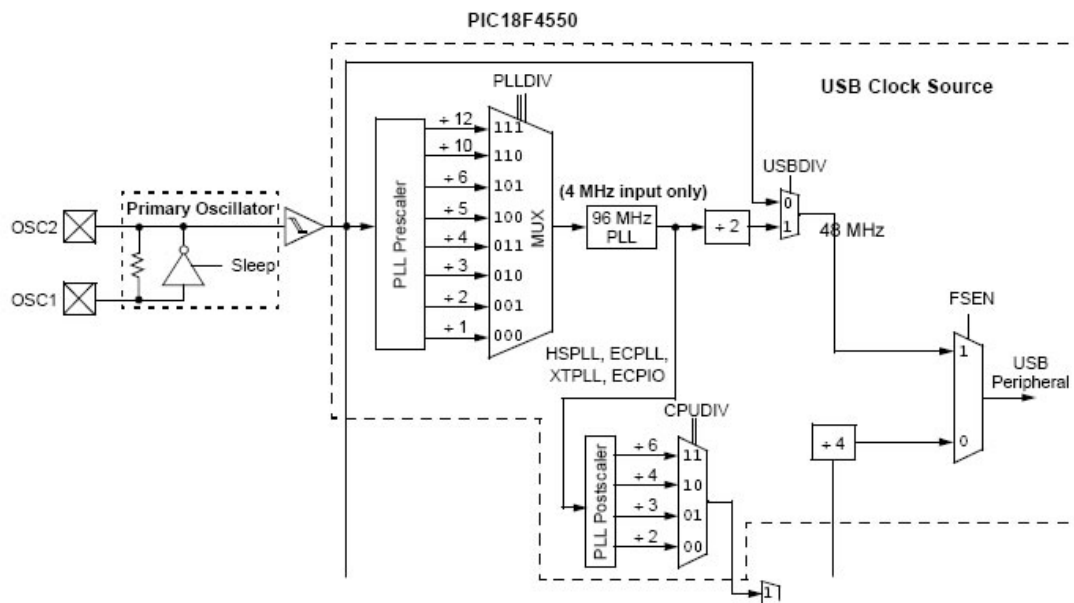
En concordancia al hardware diseñado, en firmware del microcontrolador se habilitaron 8 entradas analógicas para la adquisición de datos mediante el CAD (Convertidor Análogo Digital), se habilitaron 12 pines para la adquisición de datos de la matriz de 32 botones, se habilitaron 4 entradas para los botones de control del punto de mira, 2 salidas a indicadores LED para mostrar el estado de la conexión y transferencia de datos USB.

5.2.2.2.2 Configuración global USB

El módulo USB del microcontrolador requiere de una frecuencia de reloj específica para operar correctamente, concretamente 48 MHz son necesarios para operar a *full speed* o 6 MHz cuando se va a utilizar a *low speed*.

Para el desarrollo del módulo de comunicaciones USB se empleó un cristal de 20 MHz como oscilador externo, se fijó el *prescaler* en 5, para obtener los 4 MHz necesarios con el fin de obtener 96 MHz a partir del módulo *PPL*, posteriormente se fijó el *postscaler* en 2, y se obtuvieron los 48 MHz de frecuencia de reloj interno USB para configurar obtener una velocidad de operación *full speed* (12 Mb/s).

Figura 5.4. Configuración del reloj interno del microcontrolador.



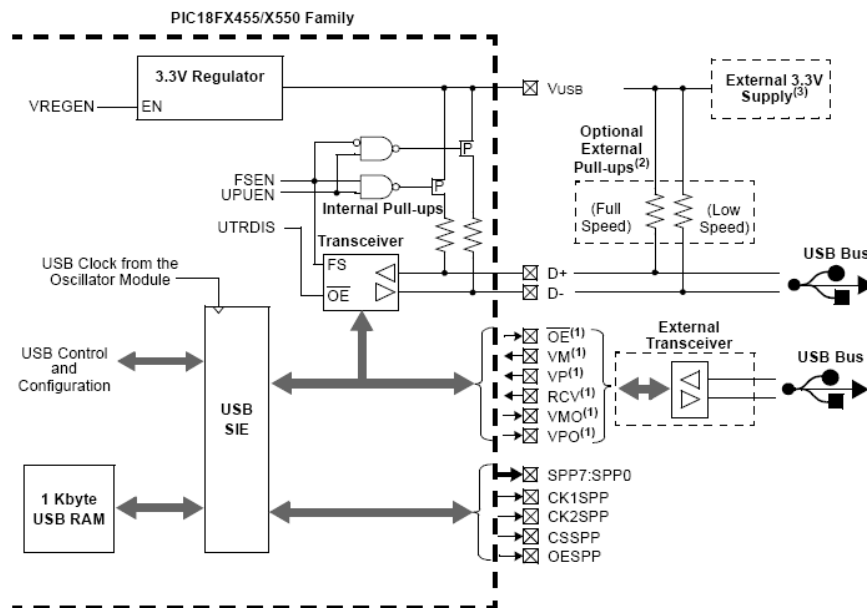
Fuente, PIC18F2455/2550/4455/4550 Data Sheet

Para configurar la comunicación USB del microcontrolador se empleó el módulo transmisor – receptor (*transceiver*) USB 2.0 integrado, para su funcionamiento se ajustaron

las resistencias *pull-up* internas mediante el registro UCFG (*USB Configuration Register*). El microcontrolador también cuenta con la opción de conectar un *transceiver* externo, aunque aumenta los componentes hardware, ver figura 5.5.

Respecto a la configuración de energía, el diseño hardware presenta dos opciones, alimentación directa del puerto USB o por una fuente externa, para ello en la configuración del firmware se habilitaron 2 pines para obtener la información de la fuente de energía empleada e informar al *host* USB. De acuerdo al diseño hardware, empleando los potenciómetros lineales y la matriz de interruptores el consumo de corriente no supera los 350 mA,

Figura 5.5. Opciones de configuración de la comunicación USB en PIC18F4550



Fuente, PIC18F2455/2550/4455/4550 Data Sheet.

5.2.2.2.3 Configuración de descriptores USB

Los descriptores son esenciales para la instalación e identificación del dispositivo, son una estructura de datos diseñada para informar todos los atributos del dispositivo conectado y de esta manera seleccionar el controlador adecuado, fue fundamental el valor asignado a

cada uno de los parámetros ya que se usarían los controladores genéricos presentes en los sistemas operativos convencionales.

Son seis descriptores, involucrados en el desarrollo de la interfaz SIES 1.0. Existe una jerarquía entre ellos y la existencia del siguiente depende de los parámetros del descriptor anterior, a continuación se describen en el orden correspondiente:

- **Device Descriptor:** describe la información general del dispositivo.
- **Configuration Descriptor:** contiene la información general de configuración del dispositivo.
- **Interface Descriptor:** describe las características de la interfaz del dispositivo, uso de memoria asociada y la declaración del uso de la especificación HID.
- **Endpoint Descriptor:** describe la información del uso de memoria y ancho de banda requerido por el dispositivo
- **HID Descriptor:** la función de este descriptor es confirmar el uso de la clase HID, especificar la existencia y tamaño de otros descriptores asociados a la clase.
- **Report Descriptor:** La función de este descriptor es informar las características específicas del dispositivo a partir de los parámetros establecidos en la clase HID, estas características definen las aplicaciones que tiene el dispositivo y define buffer de memoria requerido, informando el uso para cada dato de la trama enviada.

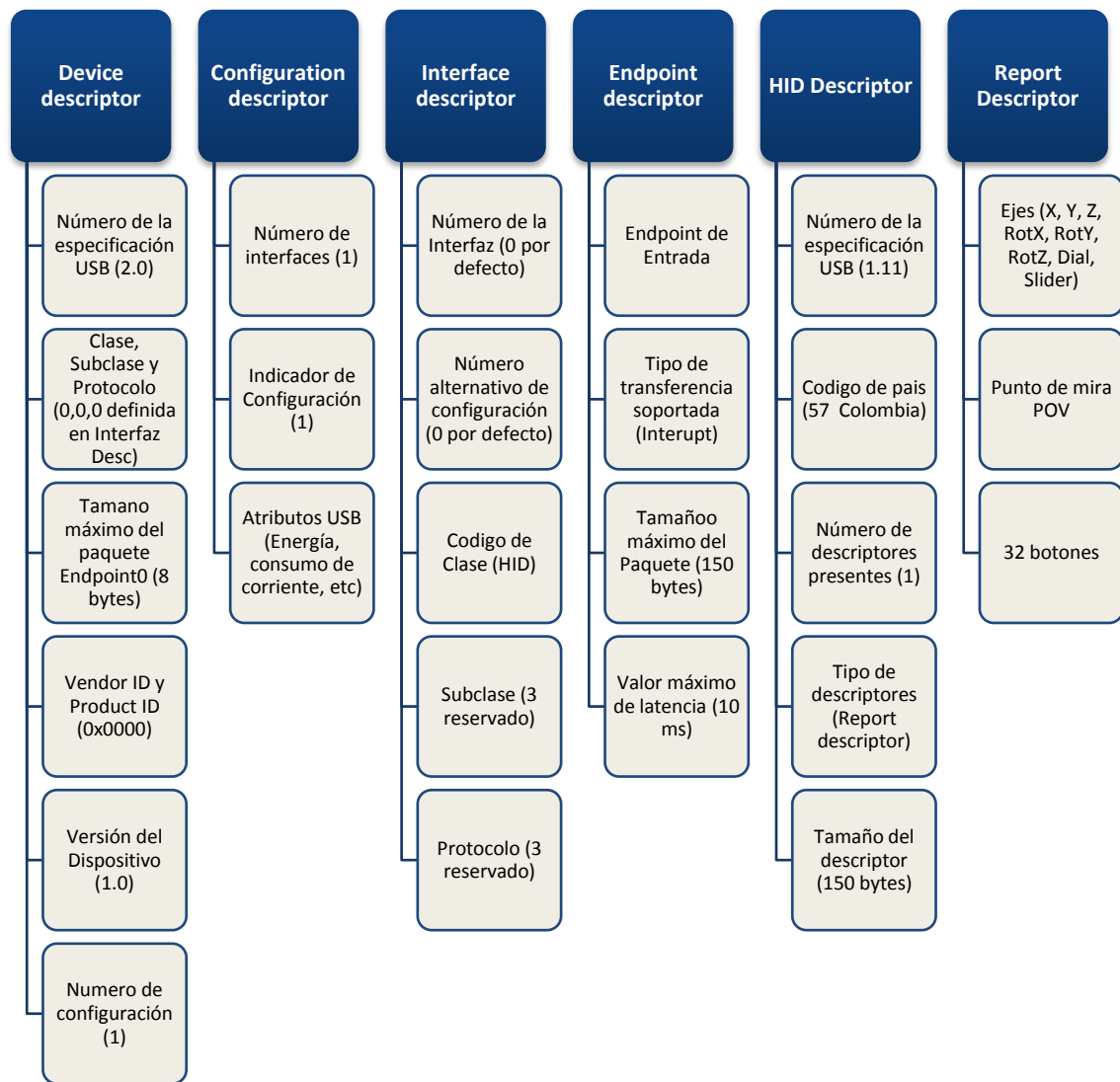
De esta manera se logra que el dispositivo sea reconocido como un periférico mas del computador y el sistema operativo reconozcas y aplicaciones propias del dispositivo permitiendo que esté disponible para cualquier aplicación que requiera sus servicios, como lo es el software de simulación de vuelo.

La figura 5.6, contiene un diagrama con la información general descrita en cada uno de los descriptores del dispositivo, los datos en paréntesis corresponden a los valores asignados en cada uno de los parámetros.

5.2.2.2.4 Report descriptor

El *report descriptor* describe las características específicas de la interfaz SIES 1.0, el uso de los ejes *X, Y, Z, Rot X, Rot Y, Rot Z, Dial, Slider, POV* y 32 botones se especifica en este descriptor, para su desarrollo se estudio la especificación HID 1.11 y se usaron las *HID Usage Tables* que contiene la lista usos e ítems para el diseño del *Report descriptor*.

Figura 5.6. Diagrama de descriptores asociados a la interfaz SIES 1.0 y parámetros configurados.



Fuente. Autores.

La estructura del *Report Descriptor* y distribución de memoria se aprecia en la tabla 5.2.

Tabla 5.2. Estructura del Report Descriptor

byte	0	1	2	3	4	5	6	7
0	Low X							
1	High X							
2	Low Y							
3	High Y							
4	Low Z							
5	High Z							
6	Low Rx							
7	High Rx							
8	Low Ry							
9	High Ry							
10	Low Rz							
11	High Rz							
12	High Dial							
13	Low Dial							
14	High Slider							
15	Low Slider							
16	0	0	0	0	Hat Switch			
17	Botón 1	Botón 2	Botón 3	Botón 4	Botón 5	Botón 6	Botón 7	Botón 8
18	Botón 9	Botón 10	Botón 11	Botón 12	Botón 13	Botón 14	Botón 15	Botón 16
19	Botón 17	Botón 18	Botón 19	Botón 20	Botón 21	Botón 22	Botón 23	Botón 24
20	Botón 25	Botón 26	Botón 27	Botón 28	Botón 29	Botón 30	Botón 31	Botón 31

Fuente, los autores.

5.2.2.2.5 Modulo de adquisición y procesamiento de datos

Para la adquisición de datos analógicos se diseño una función de adquisición de datos a partir del convertidor análogo a digital de 10 bits incluido en el microcontrolador ,se configuraron los registros ADCON0, ADCON1, ADCON2, teniendo en cuenta los tiempos de conversión y adquisición de datos y la frecuencia de operación.

Para el procesamiento de datos se diseñó un filtro digital de media móvil optimizado para minimizar el tiempo de cálculo que se describe más adelante, se diseñó una función para organizar el buffer de salida a partir de los datos filtrados de cada eje y de los datos de los 36 botones teniendo en cuenta la estructura del *Report Descriptor*.

Finalmente para el envío del buffer de salida por el bus USB, se usaron las funciones de transmisión de datos HID.

5.2.2.2.6 Filtro digital implementado

Como todo circuito electrónico, la interfaz SIES 1.0 es susceptible al ruido electrónico, los puntos vulnerables del diseño son los 8 entradas analógicas. Teniendo en cuenta que las señales provienen de cambios generados en los mandos de vuelo interfazados manipulados por un piloto, las señales muestreadas son niveles DC, motivo por el cual se decidió diseñar un filtro digital de media móvil (*Moving Average*).

El concepto de este filtro consiste en tomar N muestras, sumarlas y dividir las por N. El promedio es considerado móvil ya que se calcula cada vez que se obtiene una nueva muestra teniendo en cuenta las N-1 muestras anteriores.

$$y(n) = \frac{1}{N} [x(n) + x(n - 1) + \dots + x(n - N + 1)] \dots \dots \dots (1)$$

$$y(n) = \frac{1}{N} \sum_{l=0}^{N-1} x(n - l) \quad (2)$$

La implementación de este filtro requiere N-1 locaciones de memoria adicionales más los ciclos de máquina generados por la suma y la división, para optimizar recursos se emplea la fórmula recursiva (4)

$$y(n) = y(n - 1) - \frac{y(n-1)}{N} + \frac{x(n)}{N} \quad (3)$$

$$y(n) = \left(1 - \frac{1}{N}\right)y(n - 1) + \frac{x(n)}{N} \quad (4)$$

Comparando las dos ecuaciones (2) y (4), la última solo requiere dos multiplicaciones ya que $1/N$ es constante, y solo dos locaciones de memoria son necesarias, es la forma más eficiente de aproximar un filtro *moving average*.

Para el diseño del filtro se tuvieron en cuenta todas las funciones implementadas para la comunicación USB, el filtro debe funcionar sin obstaculizar las tareas USB, para ello se implementaron punteros a variables globales, para almacenar el dato anterior de cada señal, adicionalmente se implementó una función para que en la etapa de iniciación de la interfaz SIES 1.0 tome las N muestras iniciales y determine el primer valor promediado, y a partir de este se calcule el valor de la siguiente muestra empleando (4).

Cabe resaltar que este filtro es utilizado principalmente por la característica que presenta el ruido, aleatorio y generalmente de valor medio nulo, cuando se promedian varias mediciones el ruido tiende a cancelarse.

El valor N es conocido como constante de promediación del filtro, en la etapa de iniciación del filtro son necesarias N muestras para el cálculo del primer valor y a partir de este emplear (4), esto genera un retardo τ .

$$\tau = N(13T_{AD} + 2\mu s) \quad (5)$$

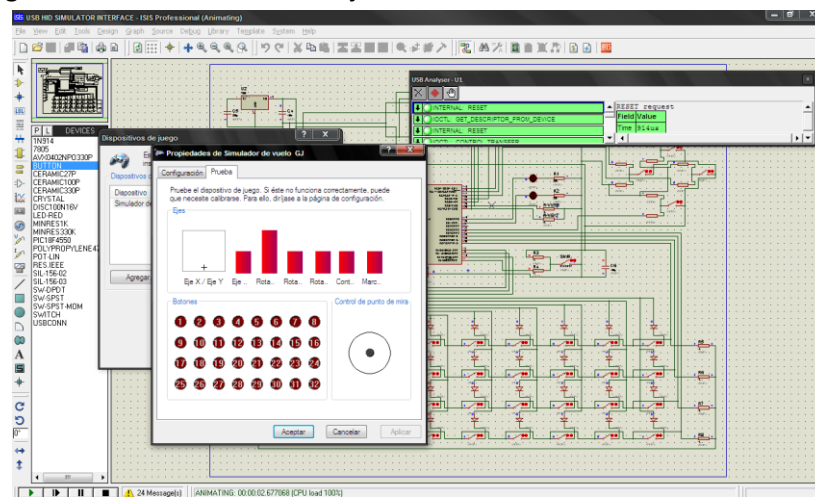
Los 13 TAD se deben a que 11 TAD son requeridos para la conversión Análogo a digital y 2 TAD el tiempo que transcurre entre cada conversión, El microcontrolador trabaja a 12 MIPS así que el retardo debido a las operaciones matemáticas es aproximadamente $2 \mu s$. El valor de TAD es $0,75 \mu s$.

Después de haber iniciado el microcontrolador y haber promediado el primer valor, el retraso ocasionado por el filtro es únicamente el generado por la operación matemática (4), ya que solo es adquirida una muestra para hacer el promedio; para el diseño del filtro se empleo un $N = 15$, debido a que este es aplicado a cada uno de las 8 entradas analógicas.

5.3 SIMULACIÓN DEL HARDWARE

Para el diseño hardware y simulación se empleo la aplicación ISIS del software Proteus versión 7.2 producido por Labcenter, se empleo esta herramienta ya que simula toda la electrónica y el firmware del microcontrolador ya que cuenta con un modulo USB virtual integrado, esta herramienta fue muy útil para el desarrollo de la interfaz SIES 1.0 ya que mediante as pruebas relazada permitieron optimizar el firmware, fueron más de 300 simulaciones realizadas las que permitieron ajustar el firmware y el hardware de la interfaz SIES 1.0.

Figura 5.7. Imagen de Proteus mientras se ejecuta la simulación.



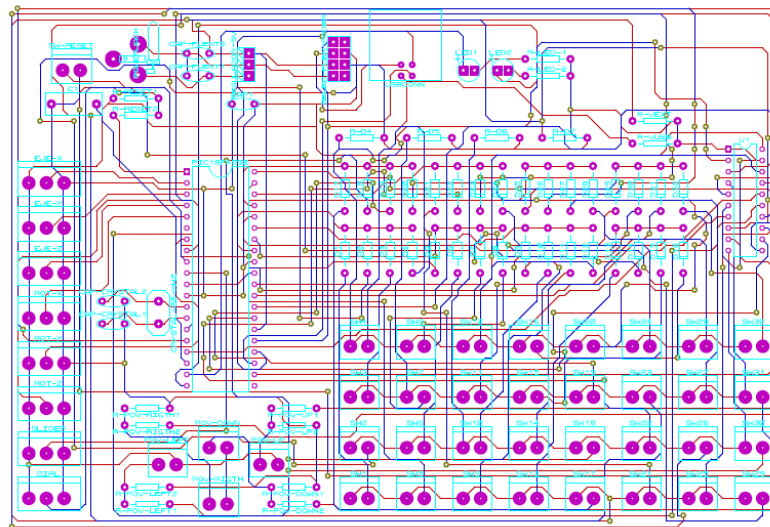
Fuente. Autores.

La diferencia entre el resultado obtenido en simulación y en las pruebas finales del hardware real se basaron en la instalación, en simulación la instalación requería la asistencia del usuario, mientras que en las pruebas del hardware real, la instalación fue automática y no requería intervención del usuario.

5.4 PLACA BASE DE LA INTERFAZ SIES 1.0

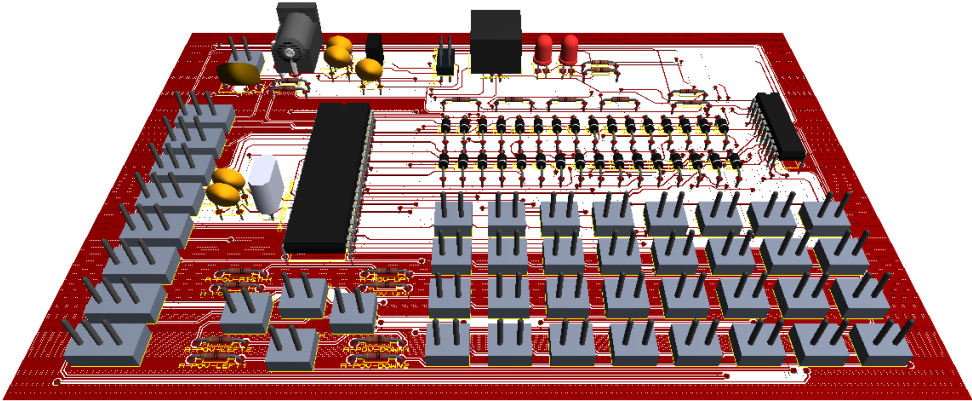
La placa base del hardware se diseño con la aplicación Ares incluida en el software Proteus, a partir del diseño simulado se diseño la placa base.

Figura 5.8. Diagrama del circuito impreso de la tarjeta diseñada de ambas caras para la interfaz SIES 1.0.



Fuente. Autores.

Figura 5.9. Diseño 3D de la placa de la interfaz SIES 1.0 con todos sus componentes



Fuente. Autores.

6. DESARROLLO DE LA INTERFAZ DE ENTRADA Y SALIDA DE DATOS (SIES 2.0)

La interfaz SIES 1.0 tiene compatibilidad universal con los software de simulación y casi todos los sistemas operativos debido a que funciona y realiza una comunicación bidireccional mediante la clase HID, sin embargo no tiene el soporte para solicitar información específica del software de simulación, como por ejemplo; el estado de las diferentes luces del avión. Se podría decir que es un intercambio de datos entre la interfaz SIES 1.0 y-PC mas no entre interfaz SIES 1.0 y-software. En otras palabras el tener una gran compatibilidad limita el intercambio de información porque cada software de simulación utiliza direcciones de variables diferentes para construir la dinámica del sistema.

Debido a que cada software de simulación tiene diferentes direcciones como mecanismos de acceso para cada una de sus variables, no es posible construir una interfaz que acceda a las variables de cualquier software de simulación, es por esta razón que se decidió trabajar con uno específico (*Flight Simulator X (FSX)*).

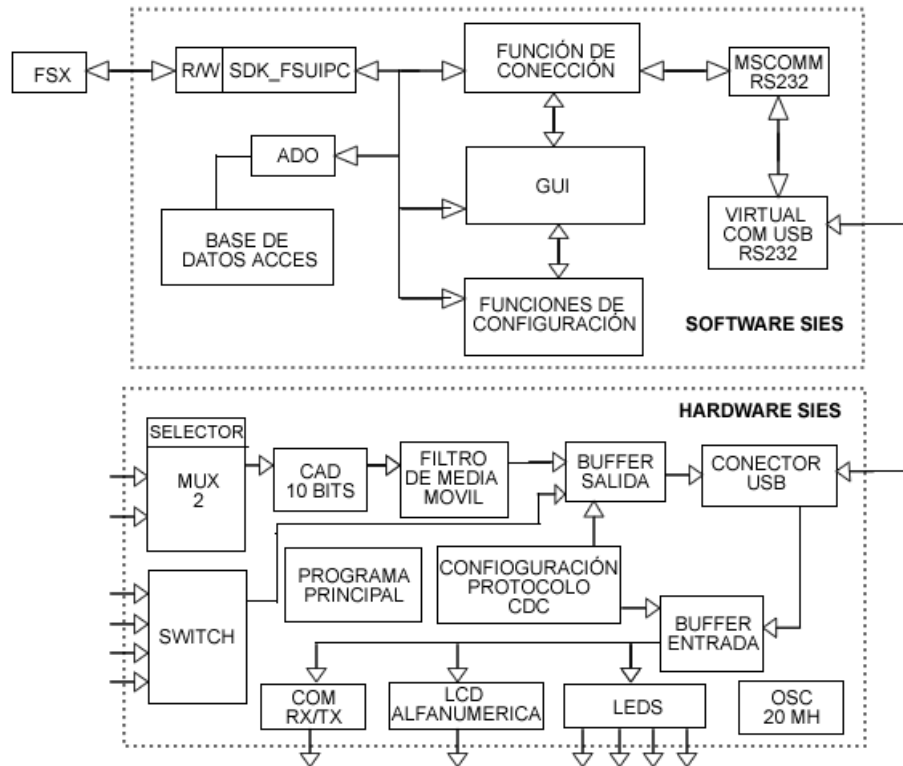
Para la construcción de la interfaz de salida SIES 2.0 fue necesaria la elaboración de un software que permitiera la comunicación entre FSX y el hardware de la interfaz SIES 2.0, de tal forma que la interfaz de salida de datos estaría compuesta por un software y un hardware. De acuerdo al diseño preliminar de la interfaz de salida de datos, se realizo el esquema de la configuración de la figura 6.1, donde se especifica cada uno sus componentes, los cuales se describirán a continuación.

6.1 SOFTWARE SIES (MODULO DE ENLACE)

Dentro del esquema presentado en la figura 4.2 correspondiente al diseño conceptual, el Software SIES 2.0 es el que corresponde al modulo de enlace. El Software SIES 2.0

principalmente ejecuta dos tareas, acceder a las variables del software de simulación y comunicarse con un hardware pertinente por medio de un protocolo de comunicación.

Figura 6.1. Diseño Preliminar de la interfaz SIES 2.0.

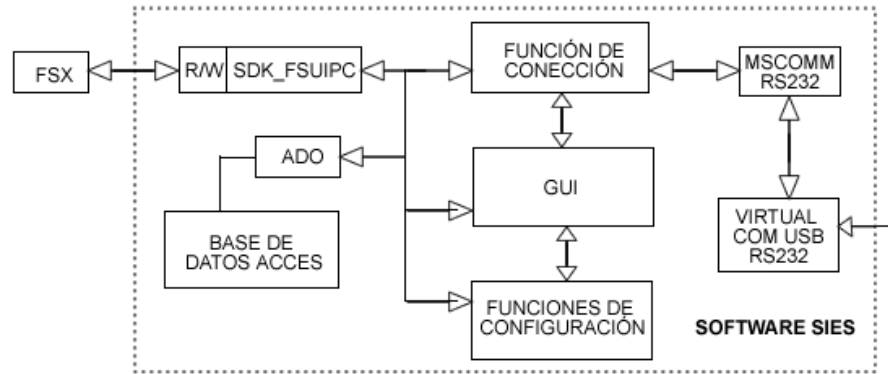


Fuente. Autores.

El Software SIES 2.0 se desarrollo con la herramienta de programación Visual Basic 6.0, ya que es un buen gestor para manejo de datos, facilita la comunicación PC - hardware y porque los desarrolladores del presente proyecto ya estaban familiarizados con este programador.

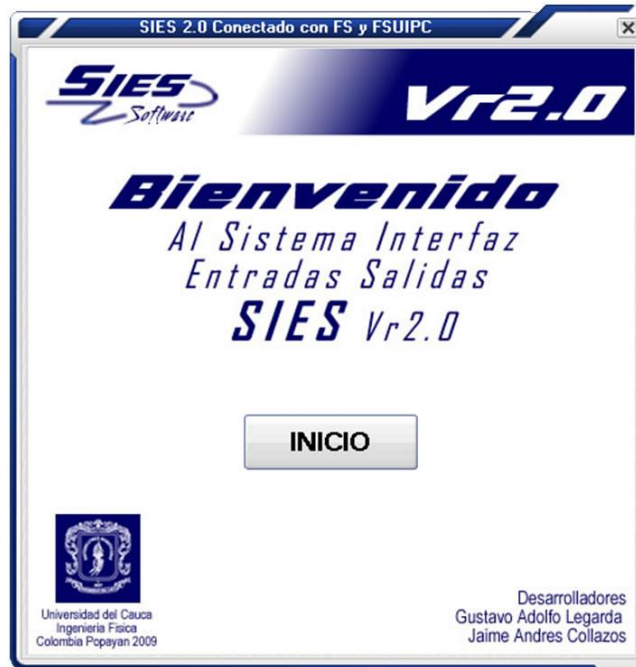
En la figura 6.2 se puede observar la estructura fundamental del programa SIES.

Figura 6.2. Estructura del Software SIES 2.0.



Fuente. Autores.

Figura 6.3. Ventana de bienvenida del Software SIES 2.0



Fuente. Autores.

6.1.1 Características funcionales del Software SIES

A continuación se mencionaran las características más relevantes y se explicaran en qué consisten cada una de ellas. Para profundizar sobre su funcionamiento se recomienda leer el manual de usuario.

6.1.1.1 Accesibilidad a las variables del FSX.

El Software SIES 2.0 tiene la capacidad de acceder a cada una de las variables del FSX, de tal forma que es posible leer o cambiar los valores de parámetros como por ejemplo estados de los mandos de control, indicadores o hasta el mismo clima dentro del FSX.

Para lograr leer y escribir sobre las variables del FSX, se agrego al proyecto del Software SIES un archivo llamado *FSUIPC.bas* de Peter Dawson, en el cual están las funciones que acceden a los archivos *kernel32*, y *user32* pertenecientes a la carpeta *system32* del sistema operativo Windows, logrando escribir y leer los estados de las variables del FSX

6.1.1.2 Bidireccionalidad de SIES 2.0

Debido a que es posible escribir en las variables del FSX, fue posible que SIES se desarrollara pensando en la bidireccionalidad, es decir que fueran capaz de obtener la información del FSX y enviarla a la tarjeta hardware como si se tratase de una interfaz de salida, u obtener la información de los mandos de control y escribir directamente en las variables del FSX como una interfaz de entrada. Este hecho da la opción de que a través de SIES se puedan conectar diferentes módulos de un simulador de vuelo y que estos funcionen de manera independiente. Por ejemplo: se puede construir un modulo *flaps* que cuente con las palancas de control además de luces que indiquen su estado actual.

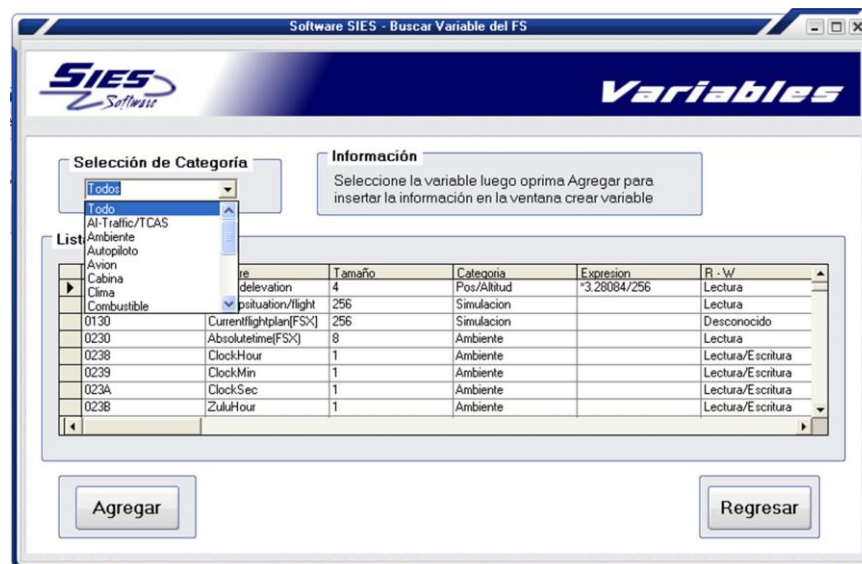
6.1.1.3 Banco de datos de 1391 variables del FSX

Para leer o escribir sobre cualquiera de las variables del FSX, es necesario conocer su dirección y tamaño en bytes, pero esta tarea puede ser algo engorrosa debido a la cantidad de variables que pueden existir en el FSX, es por esta razón que para el desarrollo del Software SIES 2.0, se realizó una pequeña base de datos con 1391 variables, y se enlazo al Software SIES, además se realizó una respectiva interfaz de usuario que facilitara la búsqueda de las variables a utilizar. Ver figura 6.4.

6.1.1.4 Configuración de variables, dispositivos de entrada y salida de datos.

Mediante una pequeña base de datos llenada por el usuario, y teniendo en cuenta el hardware que está conectado al PC, con SIES es posible realizar la configuración de una gran variedad de módulos especiales para la construcción de un entrenador de vuelo de cualquier modelo de avión existente en el software de simulación FSX.

Figura 6.4. Ventana Buscar Variables del Software SIES 2.0.



Fuente. Autores.

En el momento de construir cualquier modulo de un simulador de vuelo, pueden aparecer una gran cantidad de controles e indicadores, es por eso que para la realización del Software SIES 2.0 se realizo una clasificación de estos elementos, y se resumieron en una serie de tablas, en donde se recopila información necesaria para la adecuada configuración de cada uno de ellos teniendo en cuenta también el hardware conectado, esta información es suministrada por el usuario, pero la aplicación cuenta con una serie de opciones que facilitan la creación de cada uno de estos elementos. En el numeral 6.1.3 se explica con más detalle cada uno de los diferentes tipos de controles e indicadores. Ver figura 6.5.

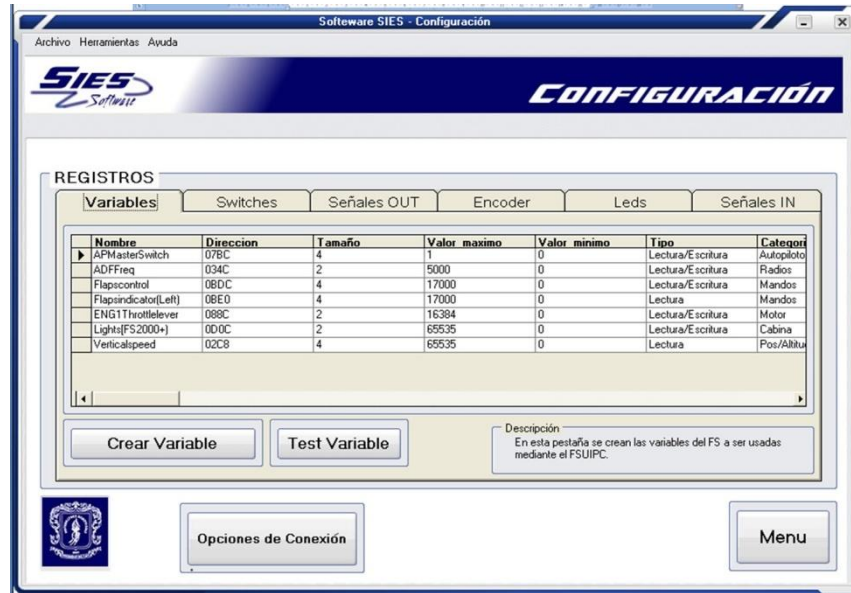
6.1.1.5 *Calibración de señales de entrada*

En ocasiones es posible que el rango de valores para manipular una variable enviado por el hardware sea menor que el propio de la variable, es decir, el hardware puede enviar valores entre 0 y 1024, pero la variable varía entre 0 y 16000, es por esto que es necesario asignarle una función al control de entrada de tal forma que su rangos abarquen los de la variable a manipular, en el manual de usuario hay mas información detallada para la calibración de estos tipos de controles. Ver figura 6.6.

6.1.1.6 *Opciones de conexión*

La comunicación se realiza mediante el protocolo RS232, pero esta se realiza de manera automática, es decir el usuario no tiene que configurar nada referente a la comunicación, solo debe escoger el puerto en el cual está conectada la tarjeta software y la velocidad de conexión (9600 - 19200), también está la opción de realizar un test en el cual se puede ver la información de transmisión. Ver figura 6.7.

Figura 6.5. Ventana principal para la configuración de variables, dispositivos de entrada y salida de datos, Software SIES 2.0.



Fuente. Autores.

Figura 6.6. Ventana para la calibración de las señales analógicas de entrada, Software SIES 2.0.



Fuente. Autores.

Figura 6.7. Ventana para configurar la conexión con el Hardware SIES 2.0



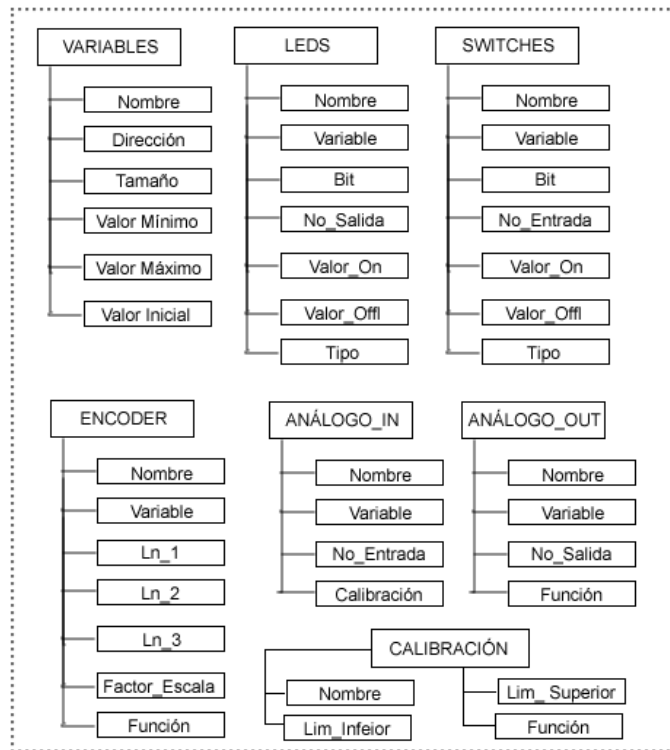
Fuente. Autores.

6.1.2 Base de datos para configuración de módulos

De acuerdo al diseño conceptual planteado en la figura 6.2, es necesaria la construcción de una base de datos la cual almacene la información del usuario en la cual se indique cuales variables quiere leer o escribir en el FSX. Para su construcción se utilizo la herramienta *Access de Microsoft*.

La base de datos cuenta con 6 tablas que en conjunto, almacenan información necesaria para la construcción de los diferentes módulos que se pueden encontrar en un entrenador de vuelo. En la figura 6.8 se puede observar la estructura de la base de datos.

Figura 6.8. Estructura de la base de datos de configuración.



Fuente. Autores.

6.1.2.1 **Tabla Variables**

En esta tabla se almacena toda la información de las diferentes variables del FSX de las cuales se desea tener acceso, como es la dirección, tamaño, valor máximo y mínimo, además del nombre que el usuario quiera colocarle a cada una de las variables para la identificación del Software SIES 2.0 como se verá a continuación.

6.1.2.2 **Tabla Leds**

La tabla *leds* almacena toda la información correspondiente a los tipos de indicadores que son de la forma *on/off*. El parámetro variable hace referencia a la variable sobre el cual el indicador funcionara, la parámetro nombre es para que el usuario le asigne una distinción

a ese indicador. El parámetro tipo, es para distinguir que la clase a la cual el indicador pertenece, las demás casillas se utilizan de acuerdo al tipo de indicador.

6.1.2.3 **Tabla Switches**

La tabla *switches* es la homóloga de la tabla *leds*, con la diferencia que la información existente allí es para los controles de tipo *on/off*.

6.1.2.4 **Tabla Encoder**

La tabla *Encoder* es para almacenar toda la información correspondiente al uso de los *encoder*, es por esta razón que existen tres parámetros llamados Ln_1, Ln_2, Ln_3 que hacen referencia a los tres conectores que tiene este control. El parámetro *factor_escala* es para indicar la magnitud de cambio por cada paso de giro del *encoder* y el parámetro *función* se utiliza para normalizar el valor de la variable.

6.1.2.5 **Tabla Analogo_Out**

Analogo_Out es la tabla que almacena toda la información necesaria para que SIES pueda leer el valor de las variables especificadas por el usuario y enviarlas a la tarjeta hardware.

6.1.2.6 **Tabla Analogo_In**

Analogo_In es la tabla que almacena toda la información necesaria para que SIES pueda leer los datos de la tarjeta hardware y escribirlos en las variables especificadas por el

usuario, tiene un parámetro especial que se llama Configuración, este da la información para la calibración en software de los transductores conectados en la tarjeta hardware.

6.1.2.7 *Tabla Calibración*

Almacena toda la información necesaria para calibrar los transductores, conectados a las entradas analógicas de la tarjeta Hardware SIES 2.0.

En la sección 6.1.3 Configuración se explicara cómo se uso la información de la base de datos en las diferentes funciones desarrolladas en el Software SIES 2.0.

6.1.2.8 *Enlace base de datos a SIES*

Para enlazar la base de datos con el Software SIES 2.0 se utilizo un control de datos *ADO*, este control permite tener acceso a cada una de las tablas expuestas en la figura 6.8, además se utilizo el control llamado "*Data Grid*" para poder dar al usuario un despliegue de cada una de las tablas y tener la opción de modificarlas a través del Software SIES 2.0.

6.1.3 *Etapas de configuración*

Antes de realizar cualquier comunicación entre el FSX y la tarjeta hardware, es necesario determinar las variables que son de interés por el usuario, escoger el puerto de salida o entrada de datos de la tarjeta hardware, y con esta información, estructurar la información de tal forma que la tarjeta sea capaz de interpretarlos y realizar cada una de las tareas que el usuario requiere. En otras palabras configurar a través del Software SIES el comportamiento del Hardware SIES 2.0.

La configuración se realizó con la información existente en la base de datos, debido a que se pueden presentar diversos tipos de controles e indicadores así como la forma en la cual se pueden usar las variables del FSX, fue necesario realizar una variedad de funciones que adquirieran, e interpretaran los datos de la base de datos para establecer si los dispositivos estaban correctamente configurados y así organizar la información de una manera eficiente para enviarla a la parte hardware. En la figura 6.9 se describe los procedimientos que se ejecutan en el Software SIES.

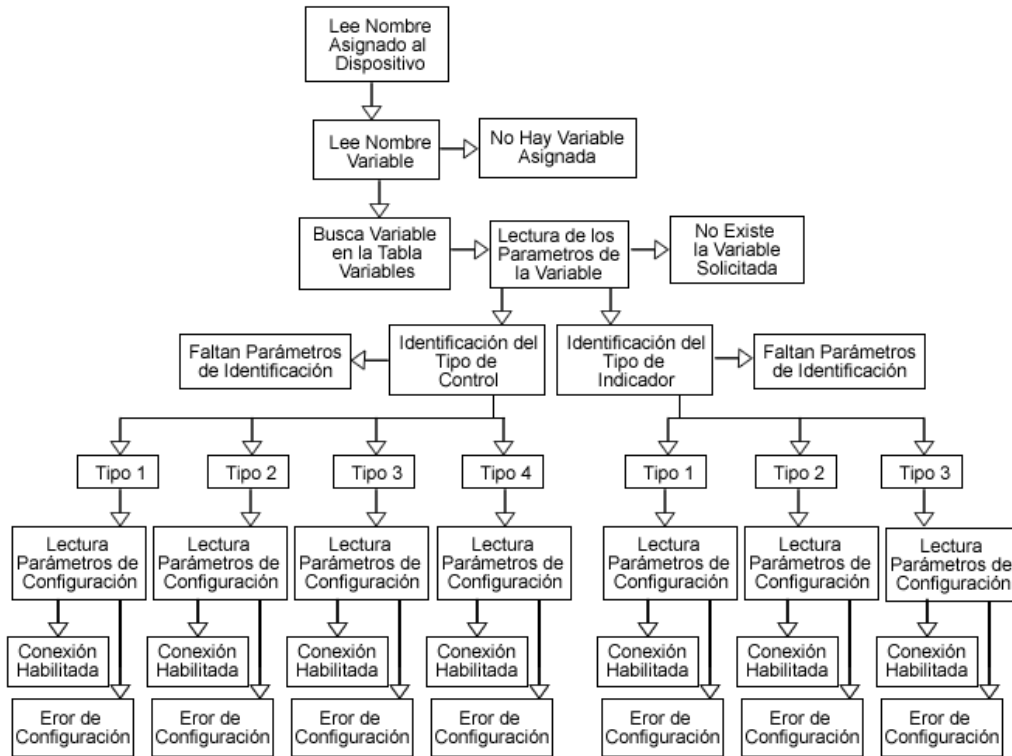
Cada función recorre un camino diferente en el diagrama de expuesto en la figura 6.9, esto depende del tipo de indicador o control con la que esté relacionada. En el caso de la función Leds, esta tiene 3 opciones de recorrido ya que tiene relacionados los indicadores tipo 1, 2 y 3. A continuación se describirán los tipos de indicadores y controles que se pueden encontrar en un FTD y se mencionan las funciones que se utilizaron para su adecuada configuración.

6.1.3.1 **Indicadores**

Para la configuración de los indicadores se tuvo en cuenta cuatro situaciones diferentes, de acuerdo a como se leería la información de las variables.

- En el caso en que solo se requiera conocer el estado de un bit de la variable. Se llamara *tipo 1*.
- Si se desea saber cuando la variable tiene valores exactos (Valores establecidos por el usuario), se denominara *tipo 2*.
- Cuando se quiere saber si el valor de la variable esta dentro de un rango especifico. Será entonces *tipo 3*.
- Para el caso en donde se quiera saber el valor actual de la variable se le llamara indicador tipo 4.

Figura 6.9. Diagrama de Procedimientos en el Software SIES 2.0.

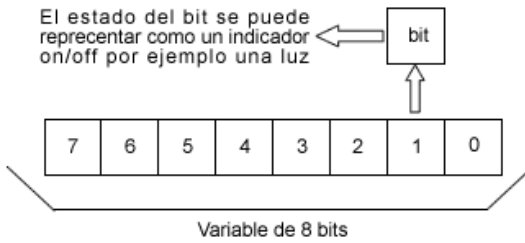


Fuente. Autores.

6.1.3.1.1 Indicador tipo 1

Cuando se trata de indicadores *on/off*, por ejemplo saber si la luz del panel de control esta encendido, el cual se puede representar con una luz encendida o apagada, FSX almacena esta información en un bit de una variable, de tal manera que si se tienen 8 indicadores de este tipo entonces el FSX utiliza una variable de un byte para guardar su estado.

Figura 6.10. Indicador tipo 1



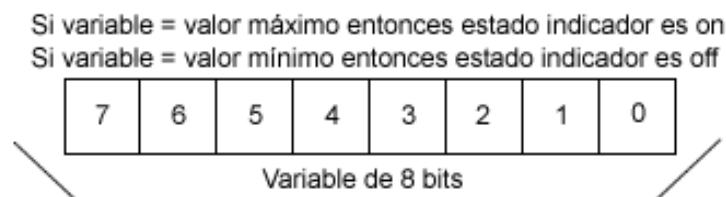
Fuente. Autores.

6.1.3.1.2 Indicador tipo 2 y tipo 3

Existen otro tipo de indicadores que también son de la forma *on/off*, pero que el FSX no utiliza un *bit* para almacenar su estado como en el caso anterior, si no por el contrario utiliza toda una variable, esto se utiliza cuando el usuario desea saber si la variable tiene un valor en específico, por ejemplo; si se quiere un indicador del tren de aterrizaje el cual indique cuando está adentro o afuera, se utilizarían los valores máximo y mínimo de la variable correspondiente.

Al igual que el indicador tipo 2 el tipo 3 hace referencia cuando el usuario desea saber si una variable se encuentre en un determinado rango, por ejemplo en el caso de los *flaps*, los cuales dependiendo el avión pueden tener intervalos de 10, 15 o 30 grados, y es necesario saber en qué posición se encuentran.

Figura 6.11. Ejemplo, Indicador tipo 2.



Fuente. Autores.

6.1.3.1.3 *Función Leds*

Debido a que los indicadores del tipo 1, 2 y 3 funcionan de la forma on/off, se utilizó una función llamada *Leds* que determinara a qué tipo de indicador pertenece, si está correctamente configurado y que parámetros hacen falta de acuerdo a la información existente en las tablas *Leds* y *Variables*.

6.1.3.1.4 *Indicador tipo 4*

Tiene el nombre de indicador pero no es de la forma on/off. El indicador de tipo 4 es de naturaleza analógica y se usa cuando el usuario desea tener acceso al valor de una variable, por ejemplo en el caso de los ángulos *pitch*, *roll* y *yaw* los cuales se deben estar leyendo constantemente y enviarlos a los actuadores de una plataforma *full motion*.

6.1.3.1.5 *Función Analogo_Out*

Esta función con la información de la tabla *Analogo_out* verifica que la información suministrada por el usuario sea correcta, es decir, que no tenga salidas repetidas, que las variables tengan una dirección y su respectiva normalización del valor de las variables de tipo 4.

6.1.3.2 **Controles.**

Como ya se había mencionado el Software SIES 2.0 se desarrolló de tal forma que permitiera tanto la lectura y escritura de las variables del FSX, esto obliga a la creación de funciones para los diferentes tipos de controles que se pueden encontrar en un entrenador de vuelo.

En un FTD se pueden encontrar 3 tipos de controles.

- Los que cambian el valor de un bit de una variable serán de tipo 1.

- Así como se manipula un bit de una variable, también es posible manipular todos los bits, pero en este caso se pueden presentar 2 opciones:
 - Cuando se modifica el valor de la variable de manera discreta será control tipo 2.
 - Cuando se modifica el valor de la variable de manera analógica se llamará control de tipo 3.

6.1.3.2.1 Control tipo 1

Al igual que existen variables que almacenan información de diferentes indicadores, también es posible tener controles que modifiquen esos parámetros, por ejemplo así como se tiene un indicador del encendido del piloto automático hay un control para este.

6.1.3.2.2 Control tipo 2

Este control tiene una particularidad y es el hecho de poder ser manipulado por *switches* o *encoders*, por ejemplo si se desea un control para sacar el tren de aterrizaje es suficiente un switch (on/off), pero en el caso de un módulo de radio, el control de la frecuencia debe ser un *encoder*, el cual pueda asignarle varios valores discretos a la variable. Para la verificación de la adecuada configuración de estos controles, se hicieron 2 funciones, una para *switches* y otra para *encoders*.

6.1.3.2.3 Función Switches

Esta función verifica que la información de la tabla *switches* este correcta, identificando el tipo de control que se está configurando.

6.1.3.2.4 *Función Encoders*

Esta función al igual que la función *switches* verifica todo lo que tiene que ver con el tipo de control 2, pero en este caso referente a los *encoder*. Para esta función se tuvo en cuenta el funcionamiento de este tipo de controles, los cuales necesitan 3 líneas de conexión para su adecuado funcionamiento.

6.1.3.2.5 *Control tipo 3*

El tipo de control 3 seguramente es el más conocido, con este se pueden manipular variables correspondientes al timón, acelerador, pedales y otros elementos de la misma naturaleza (Analógicos), su control por lo general se realiza con potenciómetro.

6.1.3.2.6 *Función Analogo_In*

Para la configuración de los controles tipo 3 se desarrollo esta función, la cual lee los datos existentes en la tabla *Analogo_In* y verifica que estén correctos.

En el manual de usuario se explican de qué manera se configuran cada uno de estos tipos de indicadores y controles, además se dan unos ejemplos de cómo combinándolos se puede lograr la configuración adecuada para la creación de algunos módulos de un simulador de vuelo.

6.2 **CONEXIÓN ENTRE EL SOFTWARE SIES 2.0 Y LA TARJETA HARDWARE SIES 2.0**

Como se mencionó anteriormente, el software esta desarrollado con Visual Basic 6.0, para la transmisión de datos se emplean los módulos de comunicación serial del RS-232, lo cual implica la selección de un puerto y su velocidad de transmisión

Para el desarrollo del hardware se busco la no dependencia de un puerto RS-232 instalado en el computador, para ello, se tuvo en cuenta la clase CDC del protocolo USB, mediante la cual es posible realizar la comunicación serie RS-232 con el protocolo USB. Para ello se diseña un dispositivo USB que es reconocido como un puerto COM con todas sus propiedades,

Esto permite realizar una comunicación de datos a nivel software como si el dispositivo fuera un puerto RS-232, pero a nivel hardware la comunicación de datos se hace mediante el bus USB con transferencias del tipo *CONTROL Y BULK*.

6.2.1 Organización de los datos de transmisión y recepción.

Al igual que en la interfaz SIES 1.0 en donde los datos de los controles se organizaban según el *Report Descriptor*, en la interfaz 2.0 también es necesario, con la diferencia de que la clase CDC no cuenta con un *report descriptor* así se tenga una conexión entre el software y el hardware, por tal razón es indispensable establecer la estructura en la cual la información viajara a través del bus de datos. Este paso es muy importante ya que esto brinda las siguientes ventajas:

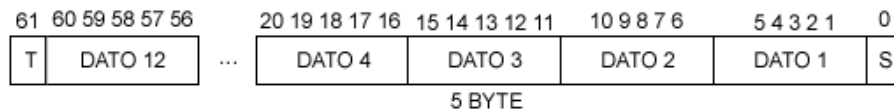
- Aprovecha de forma eficiente el bus de datos. Ya que es posible compactar la información y realizar una corrección de errores.
- Ofrece compatibilidad e independencia entre el Software SIES 2.0 y el Hardware SIES 2.0. Esto implica que el Software SIES 2.0 intercambia la información en un formato establecido de tal forma que independiente del hardware que se conecte, el lenguaje es el mismo.
- Mediante software es posible manipular las diferentes entradas y salidas del hardware.
- Posibilidad de mejoras en el futuro. Respetando el formato de los datos es posible realizar hardware más robusto sin necesidad de cambiar el Software SIES 2.0.

6.2.1.1 **Características generales**

De acuerdo a la capacidad máxima del microcontrolador escogido para el hardware de la interfaz SIES 2.0 se utilizó un buffer de 62 bytes, los cuales están compuestos por 1 byte de inicio llamado "S", un *byte* de fin de paquete llamado "T", 60 *bytes* que transportan la información de los controles en el caso de recepción o 60 *bytes* cuando se transmite la información de los indicadores, los cuales están divididos en 12 partes de 5 bytes llamados DATOS.

En la figura 6.12 se puede observar la estructura establecida para los datos tanto de recepción como de transmisión, mas sin embargo también es necesario establecer algunas normas que expliquen cómo usar esta estructura.

Figura 6.12. Organización de los datos de transmisión y recepción.



Fuente. Autores.

6.2.1.2 **Byte de inicio (s) y fin (t) de trama**

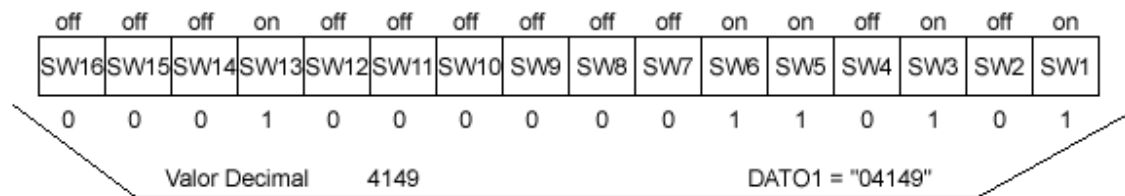
Para identificar cuando empieza la trama del dato y cuando termina se establecieron como identificadores los caracteres "s" y "t" respectivamente. Con estos parámetros es fácil identificar cada una de las posiciones de los datos 1 al 12.

6.2.1.3 **Bytes 1 al 5, DATO 1**

16 *switch* o 16 **LED** se pueden representar como un número binario de 16 dígitos en el que cada dígito representa su estado *on/off*, a su vez este número binario tiene una representación decimal que va desde 0 a 65535, para la organización del DATO 1 se

utilizo un *byte* para cada dígito decimal, es decir 5 *bytes*. De tal forma que los *byte* 1 al 5 llevan la información correspondiente al estado de las 16 entradas en el caso de recepción o 16 salidas en el caso de la transmisión.

Figura 6.13. Estructura del dato 1 para entradas digitales.



Fuente. Autores.

6.2.1.4 **Bytes 6 al 60, DATOS 2 al 12**

Al igual que el *DATO 1* los *DATOS 2 al 12* están compuestos por 5 *bytes* y también representan valores numéricos del 0 al 65535, con la diferencia que cada conjunto de *bytes* (*DATO*) representa el valor de una sola señal, estos datos se utilizan para almacenar la información correspondiente a variables de múltiples estados.

6.2.1.5 **Carácter "n", DATO vacío**

El hecho de que un *DATO* tenga el valor "00000" no implica que no tenga información, ya que una variable puede tener valor cero, entonces un dato está vacío es etiquetado como tal al contener como información la cadena de caracteres "nnnnn", de esta manera cuando se lea este carácter, se entienda que el *DATO* no tiene información de alguna variable.

6.2.2 Procesos de comunicación

Para controlar y garantizar la comunicación entre el Software y Hardware SIES 2.0, se plantearon 2 procesos en los cuales el hardware debe responder a diferentes peticiones, las cuales se describen a continuación.

Tabla 6.1. Peticiones y respuestas utilizadas para la comunicación

	Petición	Descripción	Respuesta	Descripción
Proceso 1	nserie	Esta petición le solicita al hardware el serial.	serial	Es el serial que identifica el hardware.
			*NAK	Esta palabra es enviada cuando no se reconoce la petición.
Proceso 2	dato	Este es el paquete que contiene la información del dato de transmisión y a la vez se utiliza como petición del dato de recepción.	dato	Es el paquete que lleva la información del dato de recepción.
			*NAK	Esta palabra es enviada cuando no se reconoce la petición.

Fuente. Autores.

* Cuando se reciben 5 NAK consecutivos se considera que existe un error en la comunicación.

6.3 HARDWARE SIES 2.0

El objetivo del Hardware SIES 2.0, es de capturar, procesar y transmitir cada una de las señales asociadas a los sensores del hardware a su vez recibir y procesar los datos enviadas por el Software SIES 2.0 para ejecutar acciones sobre los actuadores o indicadores correspondientes.

Adicionalmente lo que se busca con el hardware es permitir al usuario la creación de diferentes módulos que ejecuten diferentes tareas dentro de un entrenador de vuelo, ya sea un modulo de control de radio, luces, flaps, combustible, etc.

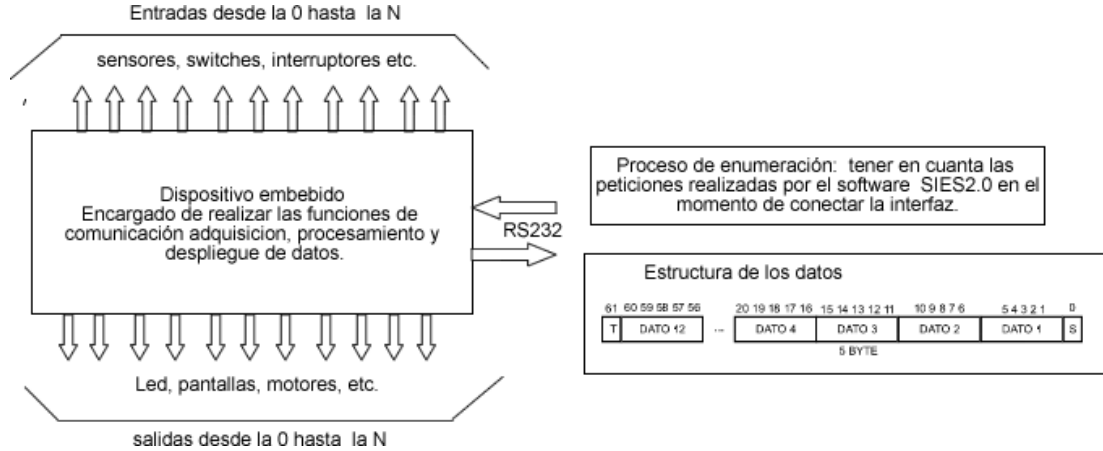
Para ello se plantea una libertad en la elección de los componentes del hardware siempre y cuando el modulo implementado cumpla los siguientes requisitos

- La comunicación mediante protocolo RS232.
- Las velocidades de comunicación en la tarjeta y PC iguales (19200 b/s).
- Exactitud en la estructura de los datos de transmisión y recepción.
- Respuesta a las peticiones y estructura de datos congruente a lo mencionado en el numeral 6.2.1.
- Tiempo de respuesta menor a 0.05 ms.
- Enumeración precisa y congruente de los puertos de entrada y salida.

6.3.1 Características del Hardware

El hardware implementado no está asociado a un modulo de un simulador de vuelo en particular, es un diseño general que pretende servir de base para cualquier modulo a desarrollar, cuenta con 4 entradas de señales analógicas, 8 entradas digitales, 8 salidas digitales, una pantalla alfanumérica de 32 caracteres, conectividad USB 2.0.

Figura 6.14. Estructura general de un modulo de comunicación asociado al Software SIES 2.0.



Fuente. Autores.

6.3.1.1 Selección de transductores y componentes utilizados

Al igual que en el hardware de la interfaz de entrada, también se utilizó el microcontrolador PIC18f4550 de microchip, potenciómetros lineales, *switches*, LED, LCD alfanumérica de dos líneas de 16 caracteres, y componentes necesarios para el funcionamiento de estos.

Como herramienta de programación y depuración se empleó *Mplab IDE 8.0* de *Microchip*, C18 como lenguaje de programación, *Proteus de Labcenter* como herramienta de simulación y depuración.

6.3.2 Firmware del microcontrolador para comunicación USB con la clase CDC.

Al igual que en la interfaz SIES 1.0, con la cual se logró desarrollar un dispositivo equivalente a un joystick USB de 8 ejes 32 botones y Punto de mira, se desarrolló el firmware del microcontrolador a partir de las librerías MCHPUSB ver 1.3

6.3.2.1 Configuración de los puertos de entradas y salidas

En concordancia, a la descripción del hardware diseñado, se habilitaron 4 entradas analógicas para la adquisición de datos mediante el CAD, se habilitaron 8 pines digitales para la adquisición del estado de diferentes *switches* o encoder, 6 salidas digitales habilitadas para indicadores LED y seis pines para el control de una pantalla LCD alfanumérica de 32 caracteres (16x2), adicionalmente dos indicadores LED para mostrar el estado de conexión y transferencia USB.

6.3.2.2 Configuración global USB

La configuración USB se realizó igual que en la interfaz SIES 1.0 descrita en el numeral 5.2.2.2.2.

6.3.2.3 Configuración de Descriptores USB

Teniendo en cuenta que el hardware diseñado hace parte de los dispositivos de soportado en la clase CDC (*Class Communication Device*) mediante la cual se permite emular la comunicación RS-232 mediante un dispositivo USB, fue necesario realizar una configuración diferente de los descriptores USB a la realizada con la interfaz de entrada SIES 1.0, esto es muy importante ya que los descriptores son fundamentales en el proceso de instalación y enumeración del dispositivo.

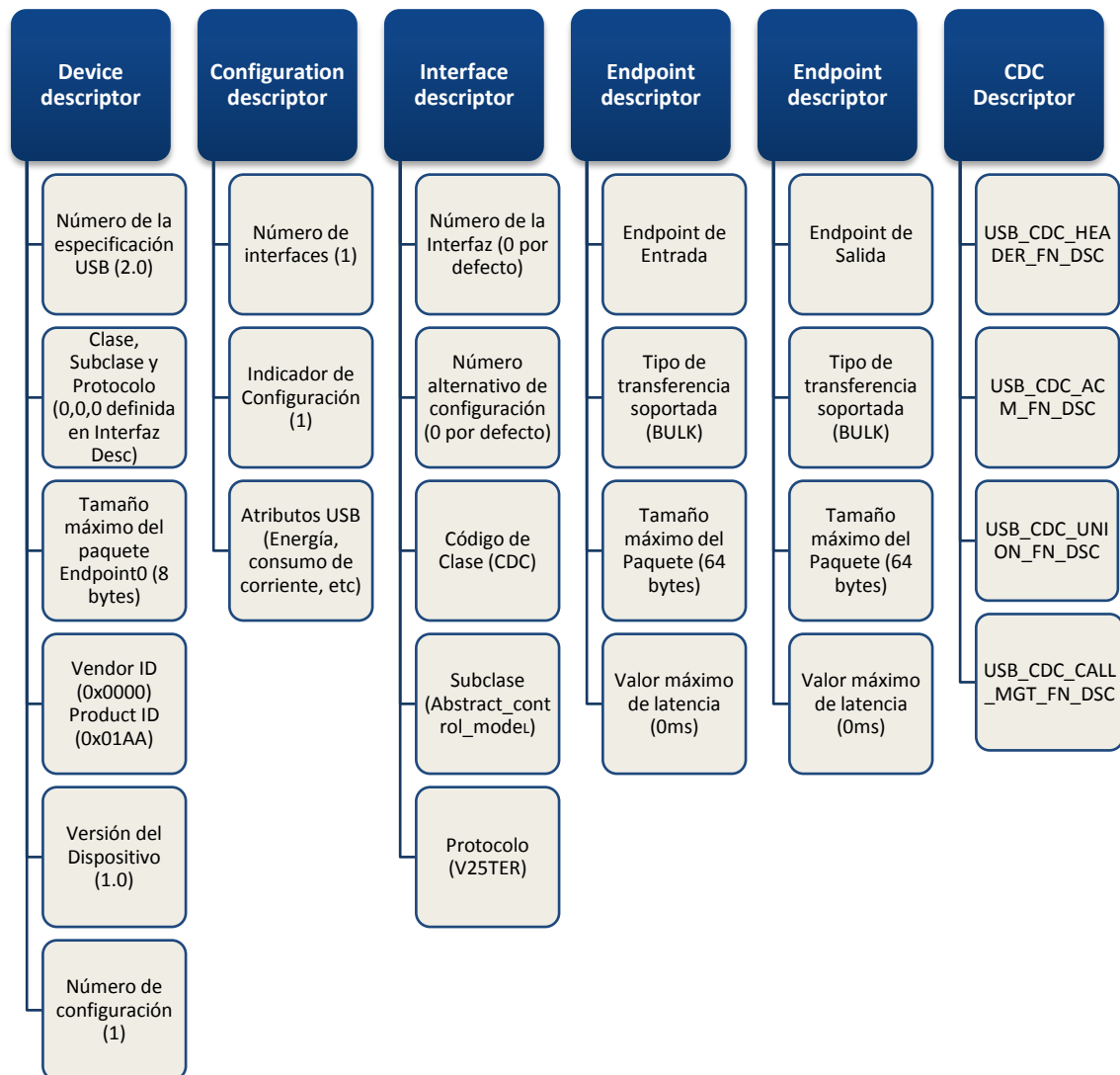
Los Descriptores declarados son:

Device Descriptor, Configuration Descriptor, Interface Descriptor, Endpoint Descriptor y CDC Descriptor.

6.3.2.4 **Modulo de adquisición y procesamiento de datos**

La adquisición, procesamiento y filtrado de señales en la etapa de adquisición se realizo de la misma forma como se indica en los numerales 5.2.2.2.5 y 5.2.2.2.6 de la interfaz SIES 1.0

Figura 6.15. Diagrama de descriptores asociados a la interfaz SIES 2.0 y parámetros configurados.



Fuente. Autores.

6.3.2.5 **Modulo de Transmisión de Datos**

Para la transmisión de datos se usaron las funciones propias de la clase CDC teniendo en cuenta las recomendaciones del artículo AN956 (*Migrating Applications to USB from RS-232 UART with Minimal Impact on PC Software*) de *Microchip Technology Inc.* Adicionalmente fue necesario la implementación de una maquina de estados para cada uno de los procesos de la comunicación.

Para la organización y encriptación del buffer de datos fue necesario implementar funciones de organización de datos a si mismo fue necesaria la implementación de funciones para interpretar los datos recibidos y procesarlos de acuerdo a los indicadores y actuadores pertinentes.

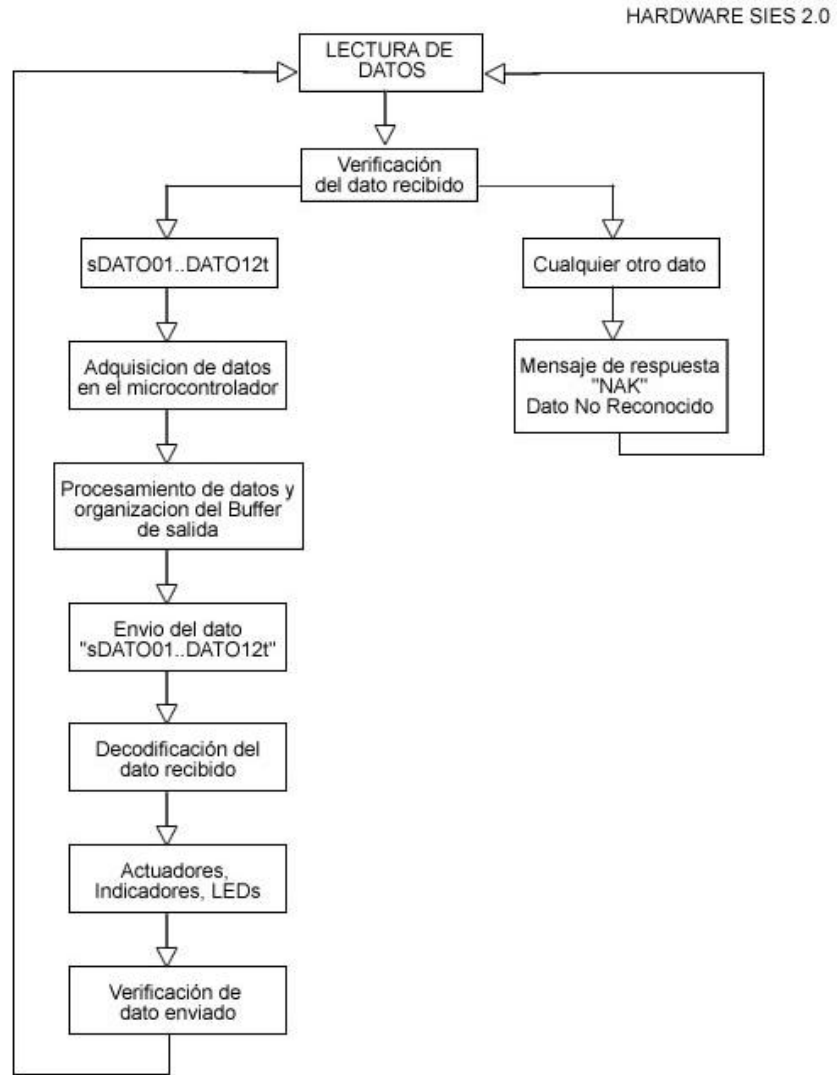
Todas las funciones de comunicación y el buffer de datos se implemento de acuerdo a los lineamientos trazados en el numeral 6.2.1.

6.3.3 **Diseño Hardware**

De acuerdo a las capacidades del microcontrolador, se realizo el diseño del hardware de la interfaz SIES 2.0 teniendo en cuenta 4 ejes analógicos, 8 entradas digitales, 6 salidas digitales y un display alfanumérico; se diseño el hardware de la interfaz SIES 2.0 y simulación mediante la herramienta *Proteus de Labcenter Electronics*.

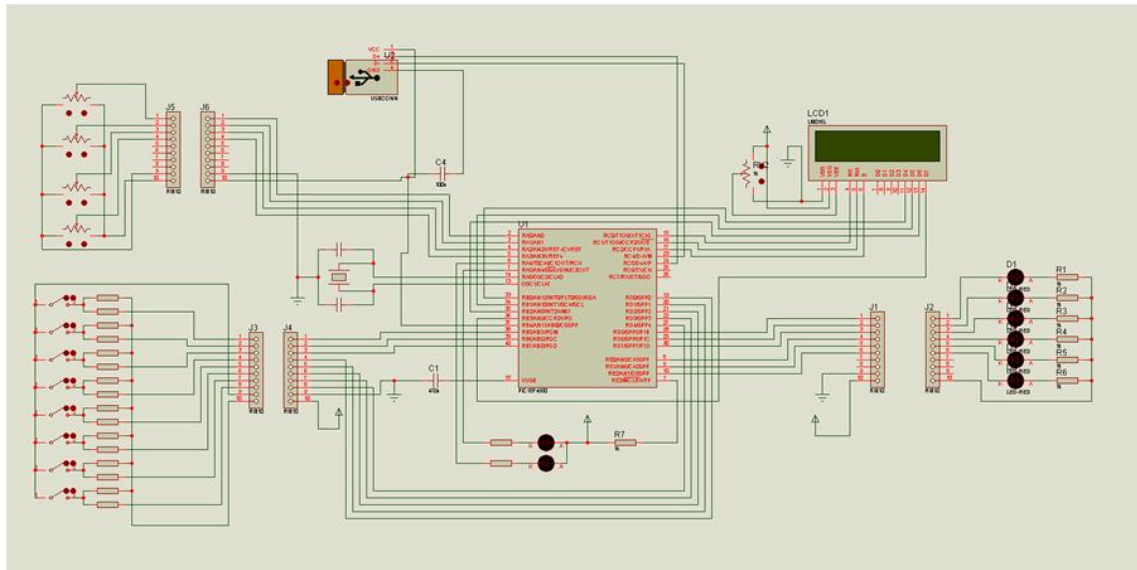
La simulación del sistema realizada en *Proteus*, fue satisfactoria en la medida que se verifico el funcionamiento del diseño final de la interfaz, se logro verificar el comportamiento de todos los componentes, sin embargo, la simulación del dispositivo USB mediante el *hub virtual* de *Proteus* presento conflictos en la transmisión de datos empleando CDC, motivo por el cual las pruebas de comunicación fueron realizadas directamente en el circuito implementado, el cual fue exitoso.

Figura 6.16. Diagrama general del proceso de transmisión y recepción de datos.



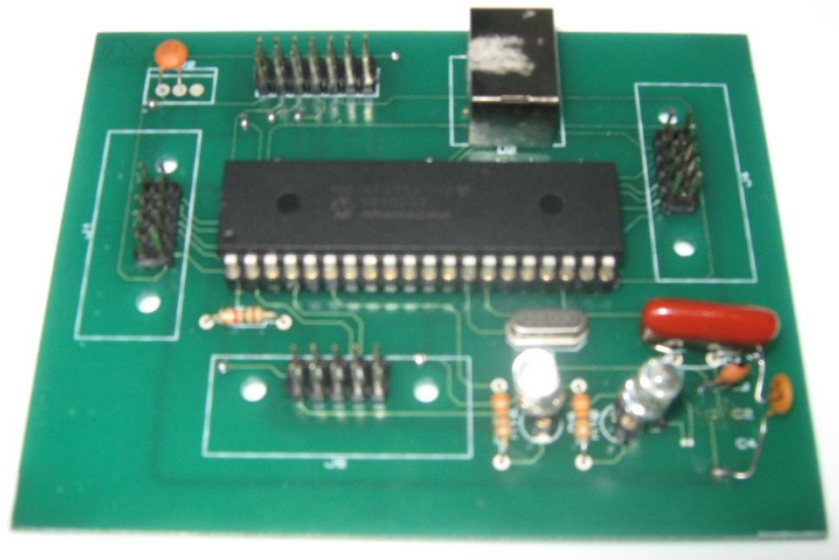
Fuente. Autores

Figura 6.17. Diagrama esquemático del circuito de la Interfaz SIES 2.0.



Fuente. Autores.

Figura 6.18. Interfaz Hardware SIES 2.0.



Fuente Autores

7. PRUEBAS Y RESULTADOS

En este capítulo se describen todas las pruebas, resultados obtenidos y un análisis de los mismos, de las dos interfaces de comunicación desarrollada, SIES 1.0 y SIES 2.0. Se realizaron pruebas de conectividad, instalación, capacidad de procesamiento, funcionamiento, estabilidad, y costo.

7.1 HARDWARE

A continuación se describen las pruebas realizadas a nivel hardware como son conectividad, instalación, capacidad de procesamiento, funcionamiento, estabilidad y costo. Ambas tarjetas fueron probadas en 6 equipos de cómputo con diferentes sistemas operativos y software de simulación.

Las siguientes fichas técnicas resumen los resultados de dichos procedimientos.

Tabla 7.1. Conectividad

Interfaz	Protocolo de Comunicación	Clase o Subclase Asociada	Velocidad de Conexión
Interfaz de entrada SIES 1.0	USB	HID	Full Speed 12 Mb/s
Interfaz SIES 2.0	USB	CDC	Full Speed 12 Mb/s como dispositivo USB, 19200 b/s en Tx de datos emulación RS-232

Fuente. Autores.

Tabla 7.2. Conectividad

Interfaz	Velocidad de transmisión de datos estimada	Tamaño del paquete de datos	Corrección de Errores
Interfaz de entrada SIES 1.0	80 Kbits por segundo	20 bytes	CRC5
Interfaz SIES 2.0	19.2 Kbits por segundo	62 bytes	CRC5 y Validación del paquete NAK

Fuente. Autores.

Tabla 7.3. Instalación.

Interfaz	Sistemas Operativos Compatibles	Necesidad de Archivo de Instalación	Necesidad de Software Adicional
Interfaz de entrada SIES 1.0	Windows, Linux, MacOS	Instalación automática y desentendida	NO
Interfaz SIES 2.0	Windows	Necesario Archivos de Instalación e Intervención del Usuario	Software SIES 2.0, FSUIP

Fuente. Autores.

Tabla 7.4. Instalación

Interfaz	Software de Simulación Compatibles	Tipo de Dispositivo Reconocido
Interfaz de entrada SIES 1.0	Con todos los software de simulación del mercado	Dispositivo de juego con 8 ejes, 32 botones y punto de mira
Interfaz SIES 2.0	Microsoft Flight Simulator	Dispositivo de comunicación SIES

Fuente. Autores.

Tabla 7.5. Capacidad de Procesamiento

Interfaz	Entradas Digitales	Entradas Analógicas
Interfaz de entrada SIES 1.0	36	8
Interfaz SIES 2.0	16	11

Fuente. Autores.

Tabla 7.6. Capacidad de Procesamiento.

Interfaz	Salidas Digitales	Salidas Analógicas	Soporte de Calibración
Interfaz de entrada SIES 1.0	0	0	software
Interfaz SIES 2.0	16 -36	11 - 12.	software

Fuente. Autores.

Tabla 7.7. Funcionamiento.

Interfaz	Switches	LED	An_OUT	An_IN
Interfaz de entrada 1.0	Asignación sin inconvenientes y funcionamiento sin errores	No Aplica	No aplica	Asignación sin inconvenientes y funcionamiento sin errores
Interfaz 2.0	Asignación sin inconvenientes y funcionamiento sin errores	Asignación sin inconvenientes y funcionamiento sin errores	Asignación sin inconvenientes y funcionamiento sin errores	Asignación sin inconvenientes y funcionamiento sin errores

Fuente. Autores.

Tabla 7.8. Costo.

Interfaz	Inversión en materiales para el desarrollo
Interfaz de entrada SIES 1.0	\$ 350.000
Interfaz SIES 2.0	\$ 300.000

Fuente. Autores.

Tabla 7.9. Estabilidad.

Tiempo de Funcionamiento	Activación de más de 4 Elementos	Desconexión de un Elemento en Tiempo de Ejecución	Conflictos con otros Dispositivos del PC	Tiempo de Funcionamiento
Se conecto y se utilizo por 24 horas continuas sin presentar ningún problema de comunicación	Puede tener todos los elementos activados sin presentar conflicto	El sistema no presento fallas al momento de desconectar controles.	No Presenta conflictos	Se conecto y se utilizo por 24 horas continuas sin presentar ningún problema de comunicación
Se conecto y se utilizo por 24 horas continuas sin presentar ningún problema de comunicación	Puede tener todos los elementos activados sin presentar conflicto	El sistema no presento fallas al momento de desconectar indicadores o controles.	No Presenta conflictos	Se conecto y se utilizo por 24 horas continuas sin presentar ningún problema de comunicación

Fuente. Autores.

7.2 SOFTWARE

Con el fin verificar el correcto funcionamiento del software SIES 2.0 desarrollado se realizaron una serie de pruebas las cuales se resumen en las siguientes tablas-

Tabla 7.10. Instalación.

Prueba	No. de equipos de prueba	Pruebas satisfactorias	Pruebas erróneas	Descripción
Instalación	8	8	0	La instalación es asistida por el usuario.
Desinstalación	8	8	0	La instalación es asistida por el usuario.
Nota: El Software SIES solo inicia completamente cuando el Flight Simulator y el FSUIPC de Petter Dowson están en ejecución.				

Fuente. Autores.

Tabla 7.11. Reconocimiento del Hardware.

Prueba	Respuesta
Selección de puerto invalido	Mensaje de Error, numero de Puerto invalido
Selección de Puerto en uso	Mensaje de Error, El Hardware no ha sido reconocido
Selección del Puerto Correcto	Mensaje de confirmación, Hardware Habilitado

Fuente. Autores.

Tabla 7.12. Transmisión Hardware software SIES 2.0.

	Dato Enviado del Software	Dato Recibido del Hardware	Dato esperado del Hardware
Test de Transmisión de Datos	sdato...datot	sdato...datot	sdato...datot
	#\$%"\$\$%	NAK	NAK
	Sdato...#\$&%	NAK	NAK
	&\$#..datot	NAK	NAK
Nota: sdato...datot hace referencia al formato de dato establecido en el numeral 6.2.1, conformado por: un byte de inicio (s), un byte de fin de trama (t) y 12 paquetes de datos de 5 bytes.			

Fuente. Autores.

Tabla 7.13. Escritura de las variables del FSX a través del Hardware SIES.

Control	Variable Asociada	Posición Mínima	Respuesta FSX	Posición máxima	Respuesta FSX	Observación
Piloto automático	APMasterSwitch	0	off	1	on	Switch tipo 2
Flap 0º	Flapscontrol	0	No cambio	1	Flaps a 0º	Switch tipo 3
Landing light	Lights[FS2000+)	0	Off	1	On	Switch tipo 1
Acelerador	ENG1Throttlelever	0	Mínimo	1024	Máximo	Entrada analógica
Radio	ADFFreq	000	No cambio	100 -010 - 001	Incremento o decremento de la variable	Encoder
Nota: Todas las pruebas fueron satisfactorias.						

Fuente. Autores.

Tabla 7.14. Lectura de las variables del FSX a envío y respuesta del Hardware SIES.

Indicador	Variable Asociada	Respuesta a FSX	Posición Mínima	Respuesta FSX	Posición máxima	Observación
piloto Automático	APMasterSwitch	off	apagado	on	encendido	Led tipo 2
Flap 0 -15	Flapsindicator (Left)	Flaps en movimiento o fuera de la posición 0º-15º	off	Flaps en movimiento entre la posición 0º-15º	On, dentro del rango (0-5470)	Led tipo 3
Landing light	Light[FS2000+)	off	apagado	on	encendido	Led tipo 1
Velocidad vertical	Verticalspeed	min	0	Max	65535	Salida analógica
Nota: Todas las pruebas fueron satisfactorias.						

Fuente. Autores.

8. ANÁLISIS DE RESULTADOS

Como se aprecia en el capítulo anterior, se desarrollaron dos interfaces para aplicaciones en simulación de vuelo denominadas SIES 1.0 y SIES 2.0, diseñadas inicialmente como interfaz de entrada y de salida.

La interfaz SIES 1.0, teniendo en cuenta los resultados de conectividad e instalación, se puede definir como un sistema de adquisición de datos para múltiples aplicaciones, el reconocimiento como dispositivo de juego USB tipo Joystick, permite que tenga compatibilidad con todos los software de simulación, no solo de vuelo sino de cualquier tipo de vehículo, adicionalmente la mayoría de lenguajes de programación tiene acceso a este tipo de periféricos, *LabView de National Instruments* tiene un instrumento virtual especial para adquirir datos desde este dispositivo, muy utilizado en especial para sistemas de control.

La flexibilidad de selección de transductores y ajuste del rango de operación mediante calibración software, descrita en las tablas de capacidad de procesamiento, permite flexibilidad y un amplio rango de aplicaciones. Dentro de la simulación de vuelo, los ocho ejes analógicos se usan generalmente en controles como timón direccional, timón de profundidad, alerones, control de mezcla de combustible, acelerador, controles del radio; el punto de mira, permite controlar el ángulo de visión rotando la cámara 360 grados, los 32 botones permiten ser configurados a múltiples acciones y comandos como encendido de motores, flaps, control del tren de aterrizaje, piloto automático, luces, frenos, etc.

Todas estas características permite a esta interfaz ser usada en aplicaciones específicas, la adaptación a un radio control para simuladores de aviones radio controlados, o a los mandos de una aeronave fuera de servicio para la construcción de un FTD de nivel inferior son ejemplos de algunas aplicaciones.

Se puede decir que la interfaz SIES 1.0, es un dispositivo muy versátil, 8 ejes analógicos, 32 botones o interruptores, punto de mira, compatibilidad con cualquier software de simulación y sistema operativo hacen de esta una herramienta apropiada en el área de

simulación. Sin embargo el flujo de información es unidireccional, es decir, es una interfaz de captura de datos y en ciertas aplicaciones es necesario complementar la interacción piloto aeronave con estímulos como movimiento, vibración, rigidez, luces, y demás, necesarios especialmente en FTD de nivel superior.

Por otra parte, la interfaz hardware – software SIES 2.0, pensada inicialmente como interfaz de salida, tiene un flujo de información bidireccional, pero a diferencia de la interfaz SIES 1.0 está diseñada específicamente para los simuladores de vuelo desarrollados por Microsoft y el sistema operativo Windows XP.

El objetivo de la interfaz SIES 2.0 es de retroalimentar el sistema accediendo a las diferentes variables del Flight Simulator y enviando la información a los diferentes actuadores del sistema.

El módulo Software de SIES 2.0 es un puente entre el hardware y el *Flight Simulator (FS)*, permitiendo la retroalimentación del sistema. Su capacidad de acceder a las variables del FS, transmitir la información al hardware, depurar los datos recibidos por el mismo y finalmente modificar las variables del FS, convierten al Software SIES 2.0 en una herramienta idónea en la simulación de vuelo.

Las pruebas de este fueron desarrolladas sin inconvenientes y sin alteraciones en la estabilidad del sistema, el diseño amigable, y los menús que orientan al usuario permiten la configuración del sistema, realizar las pruebas de lectura y escritura entre el FSX y el Hardware SIES 2.0 tal como se describe en los resultados de la tabla 7-13 y 7-14.

Esta prueba no solo permitió verificar el adecuado funcionamiento del software, si no también del hardware, lo que demostró un óptimo funcionamiento de las etapas de transmisión, recepción y procesamiento de datos en ambos módulos.

De la interfaz SIES 2.0 implementada, se puede decir que contiene todos los tipos de controles e indicadores presentes en un entrenador de vuelo y configurables con el software SIES 2.0.

Respecto a la pantalla LCD y LEDs, son usados para visualizar el valor de las variables leídas del FSX, la LCD representa a todos los actuadores que pueden ser usados como

sistemas hidráulicos, rozamiento, fricción en los controles, y demás actuadores de los cuales no se profundiza en este documento.

De acuerdo a lo anterior se podría decir que la interfaz hardware software SIES 1.0 y SIES 2.0 las cuales son compatibles, son interfaces de comunicación que puede ser empleadas para soportar toda la electrónica de diferentes módulos de un entrenador de vuelo de cualquier nivel.

CONCLUSIONES

- Se desarrollo con éxito la interfaz hardware – software para aplicaciones en simulación de vuelo, a partir del diseño e implementación de dos dispositivos USB 2.0 y una aplicación software. Estos permiten controlar los mandos de vuelo principales de una aeronave en cualquier software de simulación y controlar totalmente todos los parámetros asociados a las variables de *Microsoft Flight Simulator*.
- Se diseño e implemento la interfaz de entrada SIES 1.0 utilizando el microcontrolador PIC 18f4550 de microchip, el cual realiza la conversión de 8 señales analógicas a digitales con filtrado digital, lectura del estado de 36 interruptores simultáneamente, transmisión de datos mediante el protocolo de comunicación USB 2.0 con una velocidad *Full Speed* y soporte en la clase HID (*Human interface Device*). Esta configuración permitió que el dispositivo desarrollado presentara conectividad *plug and play*, funcionalidad con cualquier software de simulación y compatibilidad con diferentes sistemas operativos.
- Se desarrollo una interfaz (SIES 2.0) compuesta de un modulo hardware y un modulo software con las siguientes características: El modulo software es capaz de acceder a las variables del FSX, transmitir la información al hardware, depurar los datos recibidos por el mismo y finalmente modificar las variables del FSX. Un modulo hardware dotado de controles e indicadores con comunicación USB 2.0, velocidad *Full Speed* y soporte en la clase CDC (*Class Communication Device*).
- Se desarrollo e implemento satisfactoriamente el firmware del microcontrolador PIC18f450 a partir de las librerías suministradas por microchip, obteniendo dos dispositivo de adquisición, procesamiento y depuración los datos Reconocidos, como dispositivo de interfaz Humana HID y Puerto de comunicación virtual RS-232 con conectividad USB 2.0.
- Se logro desarrollar una interfaz bidireccional (SIES 2.0) especial para Microsoft Flight Simulator X, con la cual se puede controlar todos los mandos, clima, fallos y acceder la información de estos mismos parámetros permitiendo tener una interfaz

hardware software especial para diseñar modelos para la construcción de entrenadores de vuelo.

- Se realizó un análisis y clasificación de los diferentes controles e indicadores que se pueden presentar en un simulador de vuelo, para seleccionar los transductores adecuados y dar a la interfaz capacidad para estos tipos de controles e indicadores además de compatibilidad con los transductores correspondientes..
- Se logró diseñar y simular la interfaz SIES 1.0 en su totalidad tanto a nivel firmware como electrónico empleando la aplicación *ISIS de Proteus v7.2* de forma satisfactoria.
- Se logró diseñar y simular la interfaz SIES 2.0 en tanto a nivel firmware como electrónico empleando la aplicación *ISIS de Proteus v7.2*, con la limitante de incapacidad de simulación de la comunicación de datos mediante el puerto RS-232 generado virtualmente, debido a problemas del HUB virtual del simulador, aunque esto no implicó inconvenientes ya que las pruebas de comunicación a nivel hardware fueron satisfactorias
- Se implementó la calibración software en las interfaces, SIES 1.0 y SIES 2.0, gracias a la adecuada configuración de los descriptores del protocolo USB y al análisis del sistema, obteniendo más estabilidad ante fallos hardware y respuesta lineal
- La realización de las respectivas pruebas con los programas *HIDtest* y *USB Command Verifier* fueron satisfactorias verificando la correcta configuración de cada uno de los descriptores del protocolo USB y de la clase HID.
- Las pruebas de instalación, conectividad, estabilidad verificaron el óptimo funcionamiento de la interfaz desarrollada.
- A partir de la interfaz realizada se cuenta con la electrónica necesaria para la construcción de un FTD nivel 5, para entrenamiento de pilotos tanto de aviación civil como militar, tomando como software de simulación la Microsoft Flight Simulator X. aportando en el desarrollo de nuevos dispositivos de simulación en Colombia.

- Con este trabajo y toda la investigación que implico, se puede argumentar que suministro a los desarrolladores bases en el diseño y análisis de procesos ingenieriles, en el protocolo de comunicación USB, creación de base de datos y manejo *SQL*, experiencia en el manejo de herramientas como *MPLAB IDE*, *Proteus*, *Visual Basic* entre otras siendo estas pilares para el desarrollo de nuevas aplicaciones en este tipo de tecnología.

9. TRABAJO A FUTURO

En busca de perfeccionar el sistema, y en miras de lograr la capacidad de administrar al 100% la electrónica y lógica de un FTD Nivel 7, se plantean los siguientes trabajos a futuro.

- Interfazar SIES 1.0 y SIES 2.0 con los mandos de vuelo de una aeronave para el desarrollo de un entrenador de vuelo.
- Implementación de una plataforma *full motion* a partir de los datos obtenidos del Microsoft Flight Simulator con la interfaz SIES 2.0, como segundo paso para la construcción de un FTD nivel 7.
- Modificar el software SIES 2.0 de tal forma que cumpla con los siguientes requerimientos:
 - Compatibilidad con diferentes sistemas operativos.
 - Compatibilidad con otros software de simulación.
 - Capacidad de comunicación con varias tarjeas hardware al mismo tiempo.
- Diseñar e implementar tarjetas hardware de aplicaciones específicas que cumplan con los requerimientos de cada uno de los módulos de los diferentes controles e indicadores presentes en la cabina de un avión para facilitar la construcción y supervisión del funcionamiento de un entrenador de vuelo.

GLOSARIO

ADO: es el control de Visual Basic 6.0 que permite crear conexiones entre controles de enlace de datos y proveedores de datos.

CAD: Convertidor Análogo a Digital.

Class: un dispositivo USB está organizado entre clasificaciones tales como HID, audio u otras basadas en las características, peticiones soportadas y protocolos de datos.

Data Grid: es un control de Visual Basic 6.0 de tipo hoja de cálculo que muestra una serie de filas y columnas que representan registros y campos de un objeto.

Descriptor: la información sobre un dispositivo USB es almacenada en segmentos de su ROM (memoria de solo lectura). Estos segmentos son llamados *descriptors*.

Device class: un método de organizar funciones comunes y protocolos para dispositivos que tengan funciones similares- por ejemplo, comunicación, audio, display, etc.

Device descriptor: paquete de información que describe el dispositivo, por ejemplo, el vendedor, ID del producto, firmware versión, etc.

Encoder: es un dispositivo, circuito, transductor, algoritmo, o dispositivo que convierte información de un formato o tipo a otro. Específicamente en este documento, el encoder hace referencia a un dispositivo con el cual se identifica la dirección de giro de un rotor de doce estados.

Endpoint descriptor: descriptor estándar USB, describe el tipo y capacidades del canal o *pipe* de comunicación USB.

FAA (Federal Aviation Administration): es la entidad gubernamental responsable de la regulación de todos los aspectos de la aviación civil en los Estados Unidos.

Firmware: es el programa que se graba en la memoria ROM del microcontrolador, establece la lógica de más bajo nivel que lo controla. Se considera parte del hardware por estar integrado en la electrónica del dispositivo, pero también es software, ya que proporciona la lógica y está programado bajo algún tipo de lenguaje de programación.

HID (Human Interface Device): acrónimo especificando una clase específica de dispositivo o el tipo de dispositivo conocido como dispositivo de interfaz humana (HID), o HID class devices; por ejemplo un guante de datos.

HID descriptor: información acerca del dispositivo USB es almacenada en segmentos de su ROM (memoria de solo lectura), estos segmentos son llamados descriptors.

Host Controller: es el controlador principal de la transmisión y recepción de datos

HOST: un computador con puerto USB, como lo opuesto a un dispositivo conectado respecto a él.

Hub: un dispositivo USB que contiene uno o más puertos USB.

Pitch: ángulo de Euler que hace referencia al cabeceo.

Plug & play: se refiere a la capacidad de un sistema informático de configurar automáticamente los dispositivos al conectarlos. Permite poder enchufar un dispositivo y utilizarlo inmediatamente, sin preocuparte de la configuración.

Report descriptor: campos de datos específicos transferidos entre un dispositivo y un driver.

Resistencias pull-up: son resistencias conectadas entre los cables de transmisión y líneas de tensión con el fin asegurar un voltaje válido cuando no se esté recibiendo ningún dato, o identificar la velocidad de transmisión.

Roll, ángulo de Euler que hace referencia al alabeo

SQL (Structured Query Language): es un lenguaje de programación de administración, consulta y acceso a base de datos.

Transceiver: es la unidad encargada de transmisión y recepción de datos en un dispositivo.

Yaw: ángulo de Euler que hace referencia a la guiñada.

BIBLIOGRAFÍA

AXELSON, Jan. USB Complete. 3ra edición. Madison USA: Lakeview Research, 2005.

COMPAQ Hewlett- Packard, Intel, Microsoft, NEC, Philips, Universal Serial Bus Specification , Revisión 2.0, USA: USB-IF, abril 2000, <http://www.usb.org>.

GOTTFRIED, Byron. Programación en C. 1ra edición. España: McGraw-Hill, 1991.

HYDE, John. USB Designed by example. Intel, 2da edición.

KUO, Sen M. LEE, Bob. TIAN, Wenshun. Real - Time Digital Signal Processing, 2da edición. USA: Wiley, 2006.

MICROCHIP. MCHPFSUSB Firmware User's Guide. USA: Microchip technology, 2007.

MICROCHIP. Migrating Applications to USB from RS-232 UART with Minimal Impact on PC Software. 2004

MICROCHIP. MPLAB C18 C Compiler User's Guide. USA: Microchip technology, 2005.

MICROCHIP. PIC18F2455/2550/4455/4550 Data Sheet. USA: Microchip technology, 2004.

MICROSOFT CORPORATION. Microsoft Visual Basic 6.0 manual del programador: 1ra edición. España: McGraw-Hill, 1998.

USB. Implementer's Forum. Device Class Definition for Communication Devices (CDC), versión 1.1, USA: USB – IF, 1999.

USB. Implementer's Forum. Device Class Definition for Human Interface Device (HID), versión 1.11, USA: USB – IF, 2001.

USB. Implementer's Forum. HID Usage Tables, versión 1.2, USA: USB – IF, 2005.