

# Plataforma embebida multipropósito para comunicación mediante protocolo MIL-STD 1553b

Ing. Cristian Pérez

Centro de Investigación y Desarrollo de Tecnologías Aeronáuticas - Fuerza Aérea Argentina  
Ruta Nac. 158 s/n, Las Higueras, Córdoba, Argentina.  
crperez@faa.mil.ar

Mg. Ing. Héctor Riso

Especialidad en Sistemas Embebidos – Instituto Universitario Aeronáutico  
Av. Fuerza Aérea Km 8 ½, Córdoba, Argentina.  
hriso@iua.edu.ar

Dr. Pedro E. Colla

Especialidad en Sistemas Embebidos – Instituto Universitario Aeronáutico  
Av. Fuerza Aérea Km 8 ½, Córdoba, Argentina.  
[pcolla@iua.edu.ar](mailto:pcolla@iua.edu.ar)

**Abstract-** El presente trabajo presenta el desarrollo de una interfaz que comunique un equipo cualquiera con el bus Muxbus mediante la utilización de una plataforma PIC y elementos COTS. Las principales características son entre otras la posibilidad de transmitir/recibir datos utilizando protocolos de amplia difusión (RS-232, Ethernet, USB, UDP, etc.) con el protocolo MIL-STD-1553b. Si bien existen placas comerciales que realizan esta función, las mismas son de alto costo por lo que el empleo del enfoque propuesto provee un instrumento de menor costo con la posibilidad de utilización en tareas de investigación y desarrollo. Por tanto es de destacar que la originalidad del trabajo no radica en innovaciones del estándar mencionado sino en el desarrollo de un equipo construido a partir de elementos comerciales y que permita implementar estándares de la industria militar ó aeroespacial, particularmente el 1553B.

**Palabras Clave:** Bus de datos, Muxbus, PIC, Controlador de Bus, Terminal Remota, Protocolo.

## I. INTRODUCCION

En un proceso tendiente a la estandarización de Sistemas Integrados de Navegación de aeronaves militares, el Departamento de Defensa de Estados Unidos publica por primera vez en 1973 el protocolo de comunicación serial MIL-STD-1553 el cual es adoptado como un estándar de la USAF (United States Air Force). Posteriormente aparecen revisiones del mismo en 1975 y 1978 dando lugar a los protocolos MIL-STD-1553a y MIL-STD-1553b [15] respectivamente, y es adoptado posteriormente por la OTAN (Organización del Tratado Atlántico Norte) como STANAG 3838 AVS [19]. Actualmente posee una creciente cantidad de aplicaciones civiles y particularmente crece día a día su utilización en subsistemas embarcados de vehículos espaciales, satélites y UAVs (Unmanned Aerial Vehicle) [12].

## II. DESARROLLOS ACTUALES

Existen diferentes fabricantes a nivel mundial (entre ellos DDC, Ballard y Systran) que producen y comercializan

distintos tipos de interfaces que soportan el protocolo MIL-STD 1553b, como AceXtreme™ MIL-STD-1553 PC/104-Plus ó USB 1553 Avionics Interface por solo citar algunas, y que pueden comunicarse con diferentes equipos mediante Ethernet, RS-232, USB, y muchos otros. DDC también comercializa circuitos integrados que poseen embebido dicho protocolo y pueden ser utilizados en diseños propios, como el HI 1565, HI 1566 [7] ó el HI 6110 [8].

## III. PROTOCOLO MIL-STD-1553B

Define las características mecánicas, eléctricas y funcionales de un bus de datos en serie. Proporciona una interfaz física de línea balanceada dual, una interfaz de red diferencial, multiplexado por división en el tiempo, protocolo de comando/respuesta half-duplex y posibilidad de conexión con hasta 31 RT (Remote Terminal) [17].

Para el presente trabajo, la unidad bajo estudio será particularmente una RT con un software embebido [13], pudiendo extenderse su aplicación a un BM (Bus Monitor) y aún a un BC (Bus Controller).

Estas unidades, RT, BC y/ó BM pueden considerarse como un sistema distribuido [3] cuyos componentes individuales operan concurrentemente, coordinando sus acciones comunicándose únicamente a través de mensajes [20], los cuales están definidos en el estándar utilizado como referencia. A su vez, la independencia a los fallos, característica típica e importante de los sistemas distribuidos, le da robustez al sistema siendo esta una propiedad muy deseada en el ámbito de aplicación aeroespacial [18].

### A. Tipos de palabras

El estándar contempla tres tipos diferentes de palabras todas de una longitud de 20 bits, ellas son: CW (Command Word), DW (Data Word) y SW (Status Word). De esos 20 bits los tres primeros son utilizados como señal de sincronismo y el

último como bit de paridad (el protocolo implementa paridad impar).

Para realizar la transmisión de estas palabras los datos se codifican empleando el código Manchester II bi-fase [10]. Este presenta la ventaja de tener una tasa de error de 1 palabra cada  $10^7$  transferidas lo que lo hace sumamente confiable.

### B. Estructura de un mensaje MIL-STD-1553B

Dependiendo si la comunicación es BC-RT (Controlador de Bus - Terminal Remota), RT-BC ó RT-RT, el protocolo establece la secuencia en que las palabras anteriores deben ser transmitidas y los tiempos admisibles entre ellas [15].

## IV. PROPUESTA DE DESARROLLO

Debido a la creciente utilización de este protocolo en diversas ramas y a los precios de dichas placas, surge la idea de efectuar el desarrollo de una interfaz de bajo costo con elementos COTS (*Commercial of the Shelf*), que sea una alternativa de costo modesto a los disponibles comercialmente y que se adapten a tareas de investigación y desarrollo en laboratorios en desarrollos cuya característica sea que no deban soportar condiciones ambientales extremas, permitiendo a los investigadores abordar proyectos con una inversión reducida y tener la posibilidad de realizar ajustes a la plataforma de forma de adaptarla a requerimientos específicos que pudiesen presentarse. Se utiliza al efecto una arquitectura alrededor de dispositivos PIC (*Peripheral Interface Controller*) como los comercializados por la firma Microchip.

### V. ANÁLISIS DE LA INTERFAZ PROPUESTA

La interfaz desarrollada tiene dos áreas ó regiones claramente diferenciadas, ellas son:

- Región 1: destinada a realizar la comunicación entre procesador de mensajes y el dispositivo a controlar, ó integración al mismo.
- Región 2 dedicada al procesamiento de mensajes MIL-STD 1553b.

Esquemáticamente, la representación de cada una de dichas regiones puede observarse en la Fig 1.

Ya que el propósito fundamental del presente trabajo es desarrollar un sistema embebido que capaz de manejar el protocolo MIL-STD 1553B se describirá específicamente el desarrollo de la Región 2. Dado que ésta se comunica con la Región 1 a través de dos buses paralelos (de Datos y de Estado), a que el análisis de un desarrollo particular para cada tipo de protocolo de amplia difusión que pueden utilizar los diversos equipos a controlar y a la amplia información que existe sobre ellos, es que no se describirán en el presente trabajo.

## VI. DESARROLLO

### A. Análisis de procesador de palabras

El procesador de mensajes 1553b debe implementar los siguientes bloques:

- Procesamiento central
- Interfaz a LRU (Line-Replaceable Unit)

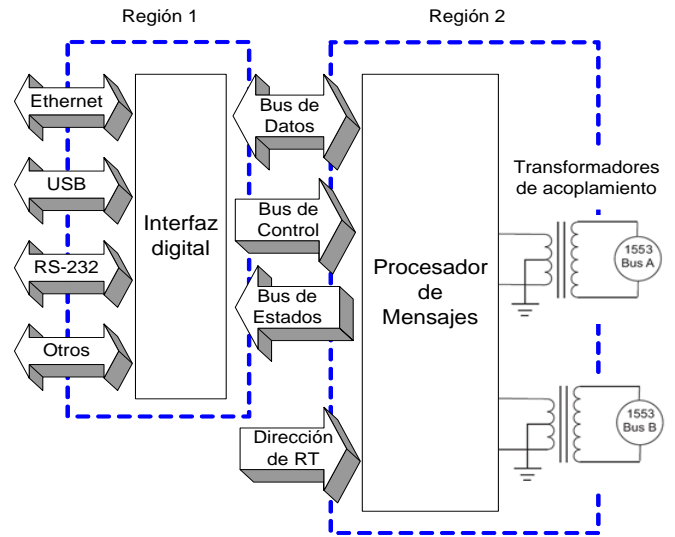


Fig. 1. Diferentes regiones de la interfaz propuesta.

- Motor del protocolo
- Modulación / Demodulación - Conversión serie / paralelo – paralelo / serie
- Transmisión/Recepción

Esquemáticamente la relación entre éstos es mostrada en la Fig 2.

Entre los diferentes módulos existe comunicación ya sea a través de registros ó de la información a transmitir/transmitida en el bus y/ó señales de control.

Si bien existen chips que reúnen los requerimientos de transmisión/recepción especificados por el estándar militar 1553b, integrados en una sola pastilla de reducido tamaño, capaces de operar en rangos extendidos de temperatura y con bajo consumo de energía, como alternativa a estos, se propone un circuito que puede realizar la misma tarea que los procesadores mencionados, pero construido íntegramente con elementos COTS, lo que permite la construcción de los mismos

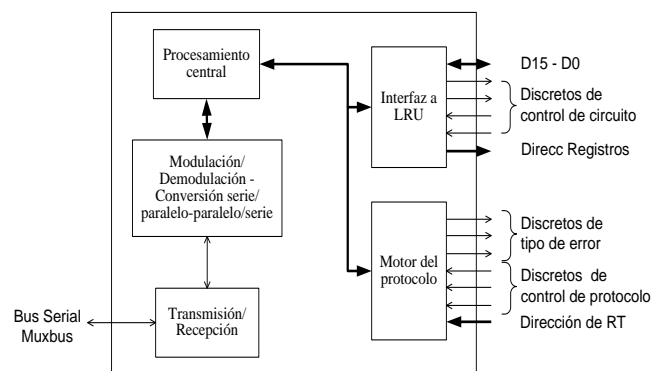


Fig. 2. Bloques del circuito propuesto.

para tareas de investigación y desarrollo, pudiendo llegar a fabricarse en el mismo laboratorio en el que se utilizará, integrados o no a la RT desarrollada/controlada.

Analizando la interfaz en forma aislada del equipo con el que se conectará, queda claro que debe pagarse un costo en lo referente a tamaño, ya que las dimensiones son mayores que la de las placas comerciales. No obstante, un desarrollo como el propuesto puede ser integrado en la misma placa del dispositivo ó RT con el que se desea establecer la comunicación lo que elimina la necesidad de conectores pudiendo llegar a equipararse las dimensiones del conjunto Placa comercial + RT con las del desarrollo propio integrado.

Como un caso extremo, la flexibilidad es tal que esta podría ejecutarse directamente en la Interfaz Digital (Región 1) mediante resultar embebida directamente en el dispositivo que controla el procesador de palabras. La utilización de un sistema comercial no permite este tipo de enfoques.

### B. *Procesamiento Central*

Este bloque es el corazón del circuito, vincula los diferentes bloques mencionados anteriormente y realiza diferentes tareas de procesamiento, comunicación y control.

Cuando el BC envía información hacia una terminal, todas escuchan pero una sola se identifica como receptora del mismo (excepto en modo *broadcast*). Por ello la primera tarea que debe realizar es analizar la dirección del destinatario y compararla con la suya. En caso que sea para esa unidad se procesa el resto del mensaje lo que determina si recibirá información ó si deberá transmitirla, así como cantidad de palabras involucradas, la subdirección de destino o modo de operación, y control de bit de paridad.

En caso que se reciban datos estos son encolados en una FIFO (*First In, First Out*) y re direccionados inmediatamente a la Interfaz LRU (no se analiza la información que llegó). Luego que todos han sido recibidos, se genera el SW y es enviado al BC. El tiempo transcurrido desde el arribo del último bit y el comienzo del envío del SW debe ser menor de 12  $\mu$ s [15].

Cuando se ha recibido una solicitud de datos éste se comunica con la Interfaz LRU a fin de requerirlos y una vez que dispone de ellos arma las palabras según el formato definido por el estándar y las envía.

Dado que las DW tienen definido un campo de datos de 16 bits de longitud, puede transmitirse más de un dato en forma conjunta. Este empaquetado también es realizado por el procesador central, el cual al mismo tiempo rellena con ceros las posiciones que no contengan datos y luego se arma el SW. Finalmente se transmite dicho SW en primer lugar y a continuación la cantidad de DW requeridas.

A fin de poder realizar todas estas tareas y dentro de tiempos establecidos por el protocolo se ha seleccionado un micro controlador PIC de Gama Mejorada de la familia 18FXXXX, tipo RISC (Reduced Instruction Set Computer) de alta performance. Este ofrece como cualidades importantes para este bloque en particular una velocidad de CPU de 16 MIPS (millones de instrucciones por segundo), 35 pines de entrada/salida más 1 pin solo de entrada, más de 1,5 KB de RAM y diferentes niveles de prioridad para interrupciones, todas ellas explotadas al máximo a fin de poder cumplir con el estándar.

Inicialmente todo el código fue realizado en lenguaje C [11], debido a que es un lenguaje débilmente tipificado (sus funciones/métodos pueden aceptar, en algunos casos, argumentos de diferente tipo de los que fueron definidos) de medio nivel pero con muchas características de bajo nivel que permite generar códigos fáciles de escribir y legibles [22]. No obstante existían funciones que demoraban en ejecutarse más tiempo del aceptado por el estándar 1553b (por ej. el tiempo de respuesta a un mensaje debe efectuarse antes de los 12  $\mu$ s de recibido el mismo).

Debido a ello, las tareas en donde el tiempo de ejecución es crítico fueron programadas en Assembler, lenguaje de bajo nivel, desarrollado especialmente para la programación de microprocesadores, micro controladores y otros circuitos integrados [1][2], reduciéndose la cantidad de ciclos de reloj necesarios para ejecutarlas y por ende el tiempo asociado a ello.

De esta forma, utilizando programación en C y Assembler en forma conjunta se optimizó el rendimiento del PIC para que fuera compatible con los requerimientos de tiempo de respuesta del protocolo.

Si bien la pastilla posee otras características como tamaño de Memoria de Programa y RAM que exceden las necesidades del bloque, no pudo usarse un PIC de una gama menor (tal como la familia 16FXXX) dado que su velocidad de procesamiento no se ajustaba a las requeridas en este bloque.

### C. *Interfaz a LRU*

En caso de recepción, este bloque toma la información binaria que previamente ha sido procesada y enviada por el bloque Procesamiento Central y la reenvía en forma paralela (bus de 16 bits) hacia el dispositivo a controlar (no analizado en este trabajo), quien es el destinatario de dicha información.

En caso de transmisión pone a disposición del bloque Procesamiento Central la información requerida (en caso de contarse con ella) a fin de que sea armada la DW y enviada hacia el bus. Esta información puede ser de distinto origen, como información de sensores, memorias, estado del equipo, registros, etc.

También este bloque posee discretos para control de la interfaz, como selección de chip, habilitación de lectura escritura, e información de datos disponibles desde el procesador central y un bus para direccionamiento de registros.

### D. *Motor del protocolo*

Este bloque interactúa con el bloque Procesamiento Central a través de la información contenida en diferentes registros (de status, de control e informe de error) y con el medio a través de los siguientes bloques de señales discretas:

- De control (entradas): destinado a establecer entre otras cosas el modo de operación, restauración del circuito, y señal de reloj.
- De estado (salidas): estas trabajan en forma conjunta con la Interfaz a LRU indicando si la información que este módulo contiene es correcta, si posee error, si existe nueva información disponible, etc.

- De Dirección (entradas): destinado exclusivamente a asignar a la LRU la dirección que se desea (en caso de hacerse por hardware).

Como puede observarse de la descripción de los dos bloques anteriores, la carga de procesamiento que éstos tienen es reducida si se la compara con la del bloque *Procesamiento Central*, ya que se limitan a transmitir información procesada o por procesar (dependiendo si la interfaz recibió o debe enviar datos respectivamente) e interactuar con algunas señales discretas que la comunican con el medio. Por ello para implementar los bloques Interfaz LRU y Motor del Protocolo basta con utilizar un micro controlador PIC de Gama Media de la familia 16FXXX, el cual posee una velocidad de CPU de 5 MIPS, 22 pines de entrada/salida y 368 Bytes de RAM, capacidades suficientes para las tareas que debe realizar.

Como en el subsistema anterior también la programación se realizó en C y Assembler.

Si bien podría utilizarse en este caso un PIC de mayores prestaciones (tal como un 18FXXX) lo cual mejoraría la performance de los dos bloques, hay que tener en cuenta que el regulador de performance (“*cuello de botella*”) de la interfaz está en el Procesador Central, por lo que el rendimiento global de la misma no se modificaría sustancialmente.

De allí surge que la razón de utilizar en esta instancia dos micro controladores diferentes para implementar los tres bloques anteriores es por una limitación en la velocidad de los mismos (en particular del primero) y para poder realizar todas las tareas necesarias sin pérdida de datos y cumplir con los requerimientos máximos de tiempo establecidos por el estándar entre requerimientos y respuestas, teniendo presente que por cada operación se requiere un mínimo de cuatro ciclos de reloj.

#### E. Modulación/Demodulación - Conversión serie/paralelo-paralelo/serie

Este es un bloque de función múltiple que cumple el propósito de modular la información a enviar por el bus y de modular la que se encuentra presente en él. Se continúa realizando la comunicación mediante código Manchester II bi-fase.

Por otro lado, el procesamiento en el bloque Procesador Central se realiza de a 16 bits pero el bus Muxbus es serializado, por lo que es necesario realizar dichas conversiones con cada palabra recibida/transmitida.

El proceso se realiza utilizando una serie de registros de desplazamiento de entrada serial y salida paralela de 8 bits tal como el 74HCT164 [4] que operan con un reloj a una velocidad de 2 MHz combinados con compuertas And y cerrojos 74HCT373 [5] también de 8 bits.

La razón de utilizar esa frecuencia, el doble de la que opera el bus, es debido a la utilización del código Manchester II (el cual presenta siempre un cambio de nivel lógico en el medio de un bit) [10], existiendo por tanto dos niveles lógicos por cada bit de información transmitida/recibida, excepto para los bits de sincronismo que es un código Manchester inválido. Por tanto el operar este circuito a la frecuencia mencionada presenta varias ventajas importantes tales como:

- Diferenciar bits de información del sistema de los bits de sincronismo.
- En caso de detectar bits de sincronismo diferenciar CW y SW de DW, dado que el sincronismo de los primeros se encuentra invertido respecto del último.
- Realizar un conteo de los bits recibidos para identificar el dato propiamente dicho (16 primeros bits) y bit de paridad para determinar si es una palabra válida o no.
- Informar al bloque Procesador Central cuando tiene información válida disponible.
- Toda la actividad se realiza mediante hardware (minimizando el tiempo necesario de procesamiento por software) por lo que no se sobrecarga al bloque Procesador Central con tareas adicionales.

#### F. Transmisión/Recepción

A fin de adaptar la tensión en modo diferencial presente en el bus (1 a 14 Vpp) con los niveles TTL (Transistor-Transistor Logic) a los cuales trabajan los componentes del circuito, existe una electrónica adicional destinada a ello. Para ello se emplea un circuito integrado tipo DS75176 [6] y un arreglo de transistores push-pull para adecuar la tensión al nivel requerido por el resto del circuito.

Este bloque además contiene un transformador de acoplamiento el cual ofrece aislamiento eléctrico entre el circuito y el bus, adaptación de impedancia y rechazo en modo común. Ello puede realizarse con un transformador tipo PM-DB27XXX [9]. La selección de uno en particular dependerá principalmente de las relaciones de transformación necesarias y unidades deseadas por pastilla.

#### VII. CÓDIGO DEL PROGRAMA

Como ya se mencionó anteriormente, los lenguajes utilizados en la programación han sido C y assembler. A modo de ejemplo, se muestran dos secciones de código. La primera empleada para determinar la dirección del destinatario, que se debe realizar tan rápido como los datos arriban a la RT (programado en assembler) ya que en caso de coincidir con la de esta se desencadena todo el proceso de tratamiento de la información y generación de respuesta. En el segundo caso los datos de una palabra presente en el bus han sido latcheados y direccionados a dos puertos del PIC para ser guardados. Como los datos no se perderán hasta no resetear los latches y el tiempo no es crítico como en el caso anterior (pues se dispone de 20 µs hasta tener que procesar otro dato), la programación puede hacerse en C.

```
//-----
// Chequeo dirección propia y Broadcast
//-----
paridad_check = 0x00;
cont_paridad = 0x00; //Reinicio del
contador de paridad
#asm
    btfsc PORTC,0; //Control de bit q
    verific direcc propia
    goto direccion_mia_ok;
```

```

    btfscc PORTC,1;      //Control de bit q
verif broadcast
    goto direccion_broadcast
    goto principal      //Si no es ninguna
espero otra trama
    direccion_mia_ok:
    direccion_broadcast:
#endasm

//-----
// Almacenamiento de Datos
//-----
TRAMA_RECIBIDA[0]=PORTB; //Almaceno
datos del latches
TRAMA_RECIBIDA[1]=PORTD;
bit_clear(PORTA,3);      //Reseteo
latches
    bit_set(PORTA,3);

```

### VIII. RESULTADOS

A lo largo del proceso de desarrollo se realizaron pruebas sobre los distintos bloques a fin de verificar que cumplan individualmente con las especificaciones fijadas. En algunos casos estas no se cumplieron inicialmente por lo que se debieron realizar modificaciones en el diseño (como lo explicado previamente en VI.B donde debió migrarse de una programación exclusivamente en C a una conjunta entre C y Assembler).

Una vez integrados todos los bloques constitutivos de la interfaz, se le enviaron desde un equipo que funcionaba como BC diferentes palabras y se analizaron las respuestas que ésta generaba. Dicho análisis consistió en verificar que se responda con las palabras correctas (contenido y cantidad de las mismas), tiempos mínimos y máximos de respuesta, período de cada bit transmitido y amplitud de la onda de tensión, dando los resultados esperados.

A modo de ejemplo, en la Fig. 3 puede visualizarse la forma de onda en las salidas del transformador de acoplamiento con respecto a masa. Ambos canales están en 5 V/div y la escala de tiempo es de 5  $\mu$ s/div. Claramente se observa que están invertidas una respecto a la otra y ambas componen la señal del código Manchester II bi-fase.

La amplitud total de la onda en modo diferencial es de 10 Vpp lo que se encuentra dentro de los límites establecidos en el estándar.

La trama total tiene entonces una longitud de 20  $\mu$ s, lo que corresponde a 20 bits transmitidos a 1 MHz. En la misma puede notarse la siguiente distribución de bits: 1 a 3 de sincronismo, 4 a 8 dirección de RT (es este caso 21, 10101B), 9 de transmisión (correspondiendo un 1), 10 a 14 subdirección (se escogió también 10101B a fin que la forma de onda sea igual a la dirección y se puedan visualizar con mayor facilidad), 15 a 19 cantidad de palabras a transmitir (establecido en 2, 00010B), 20 bit de paridad (impar), correspondiendo para los datos anteriores un 1.

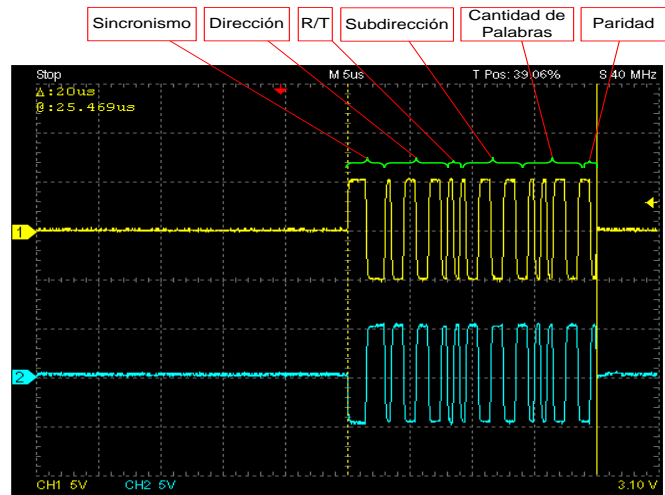


Fig. 3. Señal a la salida del transformador de acoplamiento.

### IX. CONCLUSIONES

De lo expuesto anteriormente se concluye que es altamente probable poder implementar una Interfaz MIL-STD-1553 con elementos no especializados de bajo costo y alta disponibilidad, por lo que el objetivo principal del trabajo ha sido cumplido.

Los prototipos desarrollados hasta el momento reemplazan funcionalmente a las comerciales toda vez que los requisitos de sostener ambientes rigurosos puedan ser dispensados.

Una vez desarrollado el software y hardware, contando con todos los materiales, circuitos y herramientas necesarias, el tiempo de construcción de nuevas placas y carga de software en las mismas resultó consistentemente en el orden de una jornada completa de un especialista.

Como trabajo futuro se encuentran la implementación de la misma en un micro controlador más potente (a fin de concentrar en uno solo las tareas realizadas en los dos PICs mencionados en VI.B a VI.D) con el objeto de optimizar el desarrollo reduciendo dimensiones, componentes y consumo de energía.

También se encuentra bajo análisis la opción de migrar a la tecnología FPGA (*Field Programmable Gate Array*) ya que de esta forma, al poder sintetizar hardware, se está dando un paso hacia adelante muy importante si se decidiera construir una pastilla propia personalizada a un proyecto específico.

Sobre las implementaciones futuras que hagan prever un carácter casi definitivo del desarrollo se realizarán test mas específicos para lograr su validación y verificación para su utilización en laboratorios.

Adicionalmente, se debe realizar la adecuación y certificación necesarias en el diseño para que cumpla con normas ambientales si se desea su utilización en sistemas embarcados, tales como la RTCA DO 160 E (Environmental Conditions and Test Procedures for Airborne Equipment) y RTCA DO 178 (Software Considerations in Airborne Systems and Equipment Certification).

### REFERENCIAS

- [1] J. Angulo Usategui, I. Angulo Matinez, "Microcontrolador PIC – Lenguajes Pbasic y ensamblador", 3° Ed., Mc Graw Hill (2003).
- [2] M. Balch, "Complete digital design", McGRAW-HILL (2003).

- [3] G. Coulouris, J. Dollimore, T. Kindberg, "Distributed systems. Concepts and designs", 3<sup>o</sup> Ed., China Machine Press (2001).
- [4] Data Sheet 74HCT164 - 8-bit serial-in, parallel-out shift register: NXP Semiconductors.
- [5] Data Sheet 74HCT373 - Octal D-type transparent latch; 3-state: NXP Semiconductors.
- [6] Data Sheet DS75176B/DS75176BT Multipoint RS-485/RS-422 Transceivers: National Semiconductor (1998)
- [7] Data Sheet HI-1565, HI-1566: Holt Integrated Circuits INC. (2009).
- [8] Data Sheet HI 6110: Holt Integrated Circuits INC. (2010).
- [9] Data Sheet PM-DB2745S: Holt Integrated Circuits INC. (2003).
- [10] R. Gagliardi, "Introduction to Communications Engineering", 2<sup>o</sup> Ed., Hardcover (1988).
- [11] B. Gottfrien, "Programación en C", McGraw-Hill (1991).
- [12] J. Kim, J. Han, "Development of a distributed OBDH system with MIL-STD-1553B and its application". In: Aerospace Conference Proceedings, 2000 IEEE, pp. 243-248 vol. 5, Big Sky, MT , USA (2000)
- [13] Q. Li, C. Yao, "Real-Time concepts for embedded systems", CMP Books (2003).
- [14] "MIL-STD-1553 Designers's guide sixth Edition", DDC – Data Device Corporation (1998).
- [15] "MIL-STD-1553B with notice 1, 2, 3, and 4 RETYPED VERSION", Ballard Technology (1996).
- [16] E. Palacios, F. Remiro, L. López, "Microcontrolador PIC – Desarrollo de proyecto", 2<sup>o</sup> Ed., Alfaomega
- [17] SBS Technologies Incorporated: An Interpretation of MIL-STD-1553B. Doc PN: 500-553815-00 (2004)
- [18] "Software considerations in airborne systems and equipment certification RTCA/DO-178B", RTCA Inc. (1992).
- [19] F. Stone, "1553 Overview", In: Buses for Instrument: VXI and Beyond, pp. 8/1 – 8/7, London , UK (1989).
- [20] A. Tanenbaum, M. Van Steen, "Distributed systems - Principles and paradigms", 2<sup>o</sup> Ed., Prentice Hall (2007).
- [21] J. Zhang, "Design issues for MIL-STD-1773 optical fiber avionics data buses" In: Aerospace and Electronic Systems Magazine, IEEE, pp. 25--32. (2002).
- [22] K. Zurell, "C Programming for Embedded Systems", R&D Books (2000).