



# I BUS AVIONICI

***A cura di Marco "MarcoGT"***

In avionica, per le comunicazioni tra i vari apparati a bordo vengono utilizzati dei protocolli, alcuni creati appositamente, altri invece commerciali.

E' possibile inizialmente fare una macro distinzione:

- Bus commerciali --> COTS (Commercial Off The Shelf)
- Bus "specifici"

## ***Bus Commerciali***

Nei bus COTS rientrano:

- *Firewire*
- *AFDX*

### **Firewire**

Nome tecnico IEEE 1394, nella versione originale arriva fino a 400Mbit/s. Molto diffusa sul mercato, ora viene usata anche in ambito consumer per periferiche varie. In ambito aeronautico viene usata prevalentemente per l'IFE. Lo svantaggio principale è che viene utilizzata l'architettura Daisy Chain, ossia i devices sono collegati tra di loro a cascata o ad anello. Inoltre non è fault tolerant, e solitamente non viene utilizzata per applicazioni avioniche, anche se ovviamente dipende dai requirements (l' F-35 infatti usa Firewire).

### **AFDX**

E' un derivato dell'Ethernet e viene utilizzato principalmente in ambito civile (l'Airbus A380 usa AFDX ad esempio).

Il transmission rate di AFDX è 100Mbit/s. E' un protocollo full-duplex. AFDX è l'implementazione Airbus del protocollo ARINC664. Gli LRU (Line Replacement Unit) sono connessi tra di loro tramite due switches AFDX. Sebbene la tecnologia sia full duplex i dati sono inviati in modalità half duplex per evitare collisioni. L'avionica che ne deriva viene

chiamata IMA (Integrated Modular Avionics), ossia eventuali LRU aggiuntivi possono essere aggiunti tramite racks o cabinets. Come protocollo viene usato il famoso TCP/IP, lo stesso usato in Internet per intenderci.

## **Bus Specifici**

Nei "bus specifici" abbiamo:

- MIL-STD-1553B, chiamato generalmente MILBUS
- ARINC429

### **MIL-STD-1553B**

Utilizza un collegamento elettrico per le comunicazioni (twisted pairs in rame, schermati). Le informazioni sono inviate con l'utilizzo di un clock. Supporta una ridondanza multipla. Il controllo del bus è eseguito dal BC (Bus Controller) che comunica con gli RT (Remote Terminal); gli RT possono essere fino a 31. Gli RT eseguono operazioni legate al loro subsystem. La frequenza di clock è 1MHz e la codifica è Manchester bifase. I dati da inviare sono formattati in "words" che si dividono in *status word*, *data word* e *command word*; ognuna di esse ha una diversa formattazione. Ogni word è lunga 16bit, la larghezza di banda è pari a 1Mbit/s.

### **ARINC429**

E' un bus di tipo single source-multiple sink; è half-duplex, solitamente viene utilizzato a 100kbit/s. Viene utilizzato sia in ambito civile che in ambito militare, è utilizzato ad esempio dal Boeing 767 e dall'Airbus A330 .

La topologia fa sì che si possa trasmettere da una source verso più sink ed ognuno di questi receiver riceve il messaggio inviato. In una rete ci possono essere fino a 20 sink. La modulazione utilizzata è bipolare RTZ (Return To Zero), e i messaggi sono trasmessi utilizzando una word di 32 bit (4 byte).

Passiamo ora ad un approfondimento del MIL-STD-1553B.

### **IL BUS MIL-STD-1553B**

I cavi utilizzati con questo bus hanno un'impedenza tra i 70 e gli 85 ohm. L'industria ha comunque stabilito che i cavi twinax che vengono usati debbano avere una impedenza pari a 78 ohm.

Il ritardo di propagazione su questi cavi e' pari a 4.8ns per ogni metro di lunghezza. Il cavo e' chiamato twinaxial cable ed e' simile al cavo coassiale, ma ha due conduttori interni anziche' uno.

La larghezza di banda e' pari a 1Mbit/s. Per quanto riguarda le tensioni, sono diverse sui due conduttori: quando uno porta 0.32V, l'altro porta 0V.

1553B definisce tre "terminal":

- *Bus Controller (BC)*
- *Remote Terminal (RT)*
- *Bus Monitor (BM)*

La funzione principale del BC e' quella di controllare il flusso di dati sul bus. Inoltre il BC deve trasmettere, ricevere e coordinare il trasferimento delle informazioni sul data bus. Tutte le informazioni che fluiscono sul data bus sono del tipo command/response, ossia il BC invia un command al RT che risponde con una response.

Il BC e' l'elemento chiave del bus system. Sul bus possono esserci piu' di un BC, ma solo uno puo' essere attivo nello stesso momento. Il controllo da parte del BC include la trasmissione di comandi agli RT ad intervalli prestabiliti (una sorta di polling). I comandi possono includere data, request for data (incluso lo status).

Il BC ha anche l'autorita' di modificare il flusso di dati se ci sono dei cambiamenti nell'ambiente in cui sta operando (per esempio una failure).

Il RT e' un dispositivo progettato per interfacciarsi con i subsystem tramite il bus 155B. L'interfaccia puo' essere embedded nel subsystem stesso.

Il RT riceve e decodifica i comandi dal BC, rileva gli errori e reagisce a questi errori. Il RT deve essere in grado di gestire sia errori di protocollo (missing word, per esempio) che errori elettrici (distorsioni, per esempio).

Su un singolo data bus possono esserci fino a 31 RT; ogni RT puo' avere fino a 31 subaddresses. Il RT non puo' trasmettere se non interpellato dal BC.

Il BM "ascolta" tutti i messaggi e li registra. Il BM e' sostanzialmente un bus passivo che raccoglie i dati per analisi sia real time che a posteriori.

I messaggi che sono trasmessi sul MILBUS sono word; esistono, per il MILBUS ovviamente, 3 word diverse:

- *Command word*
- *Data word*
- *Status word*

La *command word* e' lunga 20 bit e' suddivisa nel seguente modo:

Bit 0-2: sync waveform

Bit 3-7: indirizzo del RT

Bit 8: T/R bit (bit utilizzato per indicare se siamo in transmit o receive)

Bit 9-13: subaddress mode

Bit 14-18: data word count/mode code

Bit 19: parity bit

*Data word* (20 bit)

Bit 0-2: sync waveform  
Bit 3-18: data  
Bit 19: parity bit

*Status word* (20 bit)

Bit 0-2: sync waveform  
Bit 3-7: RT address  
Bit 8: Message error  
Bit 9: Instrumentation  
Bit 10: Service Request  
Bit 11-13: Reserved  
Bit 14: Broadcast command  
Bit 15: Busy  
Bit 16: Subsystem Flag  
Bit 17: Dynamic Bus Control  
Bit 18: Terminal Flag  
Bit 19: Parity

Le *command word* sono inviate dal BC al RT per sollecitare l'invio/ricezione di dati o lo svolgimento di qualche operazione.

Senza entrare nel dettaglio di ogni singolo bit, spieghiamo i principali:

*RT address*: come dice il nome e' l'indirizzo del RT a cui inviare il messaggio. NB: il messaggio viene spedito a tutti gli RT, sara' poi compito di ogni singolo RT verificare che il suo address corrisponda con quello specificato nel messaggio; se sono uguali significa quindi che il messaggio era destinato a lui, ed agira' di conseguenza.

*T/R*: indica l'azione richiesta al RT da parte del BC.

*Data Word/Count Mode*: indicano la quantita' di dati che devono essere inviati (o ricevuti) al RT.

Le *status word* sono trasmesse da un RT in risposta ad un comando (non broadcasted, però). Le *status word* contengono quindi informazioni sugli errori (eventuali) contenuti nel messaggio ricevuto dallo stesso RT che invia la *status word*. Le *status word* sono trasmesse da un RT solo dopo aver ricevuto un comando da un BC. A cosa serve questo? A verificare che il corretto RT risponda alla *command word* e che nessun altro RT risponda (per sbaglio).

Di seguito degli "scambi" di messaggi tra BC e RT.

### 1) *BC to RT Transfer*

Il BC invia un receive command seguito dalle data words. Il RT, dopo la ricezione e validazione del messaggio, trasmette una status word al BC. Command e Data word sono inviate, dal BC, senza intervalli di interruzione.

Receive Command -- Data Word -- Data Word ..... Data Word ++ Status Word --> Next Command

### 2) *RT to BC Transfer*

Il BC invia un transmit command al RT; l'RT, dopo la ricezione e validazione, invia una status word al BC seguito dalle data words.

Transmit Command ++ Status Word ++ Data Word -- Data Word ... Data Word --> Next Command

### 3) *RT to RT Transfer*

BC riceve un receive command da RT A e un transmit command da RT B. RT B, dopo ricezione e validazione, trasmette una status word seguita dalle data words; terminato l'invio delle data words da parte di RT B, RT A invia una status word.

Receive Command Transmit Command ++ Status Word Data Word -- Data Word .... Data Word ++ Status Word --> Next Command

Questi sono gli "schemi" principali: ne esistono ovviamente degli altri, come ad esempio il broadcast.

**Marco "MarcoGT"**

**Copyright 2011 Md80.it**

## **BIBLIOGRAFIA**

**I. Moir, A. Seabridge "Military Avionics Systems", Wiley**

