



UNIVERSITÀ DEGLI STUDI ROMA TRE

ANNO ACCADEMICO 1999/2000

Corso di “CONTROLLO DIGITALE”

Analisi della rete informatica:

“BOSCH - Controller Area Network (CAN)”

Francesco Scorretti

Introduzione.

Concetti.

CIM (Computer Integrated Manufacturing): modello di riferimento per la realizzazione dell'automazione industriale basata sul rilevamento, il coordinamento e la trasmissione di informazioni tra i vari sottoinsiemi mediante l'utilizzo di reti informatiche.

RETI DI CAMPO: realizzazioni di reti informatiche per il collegamento dei dispositivi di controllo, come i PLC, ed i dispositivi presenti sul campo, come i sensori e gli attuatori.

La situazione tipica attuale.

Tipicamente, sensori ed attuatori sono collegati al controllore o direttamente, mediante i moduli ingresso/uscita presenti nel suo armadio, ovvero tramite una linea seriale da armadi di I/O remoti.

Vantaggi connessi all'utilizzo di una rete di campo.

Utilizzando una rete di campo, invece, tali dispositivi diventano a tutti gli effetti dei nodi di una rete informatica. I dispositivi connessi devono essere dotati di un minimo di capacità elaborativa che gli consenta di comunicare attraverso la rete (necessità di avere sensori ed attuatori interfacciabili).

Una rete di campo differisce da una rete informatica tradizionale:

- Maggiore brevità dei messaggi;
- Multicast;
- Efficienza;
- Semplicità del protocollo.

Tuttavia, gli scopi di una rete di campo possono essere diversi. Dunque, tale categoria di reti è stata ulteriormente divisa in tre differenti classi.

1. **SENSORBUS:** bus di sensori, che operano a livello del bit, realizzano –di solito- unicamente i primi due livelli ISO-OSI. Il loro compito primario è la riduzione del cablaggio; la lunghezza tipica dei messaggi è inferiore ad un byte.
2. **DEVICEBUS:** bus di dispositivi, permettono il trasferimento di messaggi di lunghezza 16-32 bytes. Oltre ai livelli 1 e 2 talvolta implementano alcune funzioni del livello di applicazione (7). Spesso permettono la comunicazione di una semplice diagnostica.
3. **FIELDBUS:** operano a livello dei gruppi di bytes. Implementano i livelli ISO-OSI 1, 2, 7 ed un ulteriore livello 8 di "utente".

CAN si colloca tra i *bus di sensori* ed i *bus di dispositivi*. Non è previsto nel protocollo un livello di applicazione proprio: tipicamente altre reti appoggiano il loro strato 7 su CAN (cfr. *DeviceNet*).

CAN (Controller Area Network).

Introduzione.

- ❖ Tecnologia aperta per il controllo real-time (distribuito) che supporta diverse applicazioni;
- ❖ Nasce in ambiente automobilistico (controllo di motori, sensori, ABS, ...) per collegare apparecchiature elettroniche in situazioni estreme (temperature molto elevate, rumore, vibrazioni, scossoni): la possibilità di funzionare in tali ambienti ne ha permesso la diffusione anche in ambienti industriali;
- ❖ Frequenza di trasmissione: fino a 500 Kbit/s nelle più recenti implementazioni;
- ❖ Disponibili chips che implementano CAN di: Philips, Intel, Motorola, NEC, Hitachi, Siemens;
- ❖ Nel 1995, circa 5 milioni di chips circolanti;
- ❖ Indirizzi di 11 bits (*standard format*) prima, poi (*extended format*) di 29 ($2^{11}=2'048$; $2^{29}=536'870'912$): per supportare molteplici applicazioni;
- ❖ Flessibilità dell'architettura di rete: master/slave, multi-masters, punto-punto;

La tecnologia CAN si colloca sui livelli 1 e 2 (Fisico e Data Link) della pila ISO-OSI.

Inoltre, il livello Data Link viene ulteriormente suddiviso in due sottolivelli:

1. CAN-object layer:

- ❖ Seleziona i messaggi da trasmettere;
- ❖ Decide quali dei messaggi ricevuti dal CAN-transfer layer vanno usati attualmente;
- ❖ Fornisce un'interfaccia per l'applicazione verso l'hardware;

2. CAN-transfer layer;

- ❖ Tipiche funzioni Data Link: controllo dell'integrità del frame, error checking, segnalazione di errori, recupero di errori, arbitration (politica di trasmissione: "chi parla quando");

con il 2° a ridosso dello strato fisico.

Il livello fisico si occupa di trasferire bits tra nodi diversi in rispetto di proprietà elettriche.

Si descrive, in seguito, il protocollo standard (CAN v1.1: indirizzamento ad 11 bits).

Concetti base.

Caratteristiche (specifiche di progetto):

- ❖ Messaggi con priorità diverse;
 - ❖ Garanzie sul ritardo di trasmissione;
 - ❖ Flessibilità di configurazione;
 - ❖ Multicast;
 - ❖ Consistenza dei dati;
 - ❖ Rilevamento e segnalazione di errori;
 - ❖ Ritrasmissione dei messaggi corrotti;
 - ❖ Spegnimento dei nodi rotti.
- Il **CAN-transfer layer** è il kernel del CAN. Esso riceve messaggi dall'object-layer e li passa allo strato fisico affinché esso li invii (trasmissione). Inoltre, riceve messaggi dallo strato fisico e li passa all'object-layer (ricezione).
 - Il **CAN-object layer** si occupa del *message filtering*.

CAN object-layer	Data link (ISO-OSI)
CAN transfer-layer	
CAN physical-layer	Physical-layer (ISO-OSI)

Messaggi.

Le informazioni viaggiano sul bus mediante messaggi con formato ben definito di lunghezze differenti ma limitate. Quando il canale (bus) è libero ogni unità connessa può cominciare a trasmettere: politica di trasmissione “**csma/nba** (non-destructive bitwise arbitration)”.

Informazioni di routing.

In una rete CAN un nodo non ha alcuna necessità di conoscere informazioni circa la configurazione del sistema (ad esempio gli indirizzi delle stazioni); ciò implica che:

- ◆ Il sistema è flessibile: possono aggiungersi o rimuoversi nodi senza alcun aggiornamento del software o dell'hardware (trasparente al livello di applicazione);
- ◆ *Message routing*: Il tipo di contenuto di un messaggio è specificato da un identificatore (ID di 11 bits). L'ID non indica il destinatario del messaggio, ma il *significato* dei dati che contiene. In questo modo, ogni nodo della rete può decidere quali dati leggere e quali scartare (**MESSAGE FILTERING**).

- ◆ Un'ovvia conseguenza consiste nella possibilità di trasmissioni broadcast/multicast: un qualsiasi numero di nodi può ricevere simultaneamente lo stesso messaggio.
- ◆ L'ID del pacchetto è anche utilizzato per stabilire la priorità di ritrasmissione in caso di collisione.

Bit Rate.

La velocità di CAN può essere diversa in reti diverse (da 125 Kbit/s a 500 Kbit/s); in una data rete la velocità deve essere stabilita ed unica.

Priorità.

L'ID di pacchetto definisce una *priorità statica* in trasmissione (cfr. "Arbitration"). La stazione che deve trasmettere il messaggio con priorità maggiore (ID minore: lo "0" vince sull'"1". Al livello logico "0" corrisponde un livello di segnale **DOMINANT**, al livello logico "1" corrisponde un livello di segnale **RECESSIVE**. Un'eventuale sovrapposizione di segnali DOMINANT e RECESSIVE sul bus viene interpretato come uno "0" logico.) può trasmettere non appena è libero il bus.

Richiesta di dati.

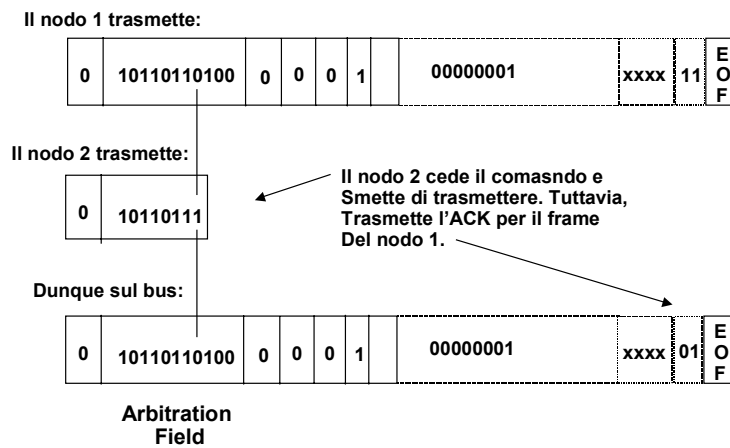
Una stazione può richiedere dei dati di un tipo specificato (da un certo ID) inviando un **REMOTE FRAME**. La stazione che può fornirli li invia in un **DATA FRAME**. Il REMOTE FRAME ed il DATA FRAME suddetti hanno lo stesso ID.

Arbitration.

Politica di trasmissione **csma/nba**.

Se 2 o più stazioni cominciano a trasmettere simultaneamente il conflitto nell'accesso al canale viene risolto confrontando gli ID dei pacchetti (non si ha collisione). Se un DATA ed un REMOTE FRAME con lo stesso ID vengono trasmessi simultaneamente, vince il DATA FRAME (il REMOTE FRAME è stato inviato per sollecitare la trasmissione dei dati, immaginando che il primo inviato fosse stato perduto): essendo uguali gli ID il confronto si fa sull'RTR bit che vale "0" per i DATA ed "1" per i REMOTE FRAMEs.

ACCESSO AL CANALE: mentre una stazione trasmette, ascolta il canale e, se rileva uno "0" avendo trasmesso un "1", essa ha perso il confronto e smette di trasmettere.



Sicurezza.

Per garantire tolleranza ai malfunzionamenti nel trasferimento dati, CAN implementa:

ERROR DETECTION: tramite *monitoring*, *CRC*, *message frame check*.
BOSCH garantisce che:

- Tutti gli errori globali vengono rilevati;
- Tutti gli errori locali in trasmissione vengono rilevati;
- Vengono rilevati più di 5 errori casuali in un messaggio;
- Errori “burst” (bits corrotti in sequenza) di lunghezza minore di 15 bits nello stesso messaggio vengono rilevati;
- La probabilità residua di non individuare un messaggio corrotto è minore di: *message error rate* * $4.7 \cdot 10^{-11}$.

Segnalazione di errori e tempo di recupero.

Ogni nodo che rilevi un errore lo segnala. La trasmissione di tutti i messaggi sul canale viene interrotta e tutti i frame vengono ritrasmessi. Il tempo di recupero è di 29 bit time (*tempo per trasmettere 29 bits = 29/bit rate*) se non ci sono errori ulteriori.

Isolamento delle stazioni guaste.

Le stazioni CAN sono in grado di distinguere malfunzionamenti casuali da malfunzionamenti permanenti. I nodi rotti vengono spenti.

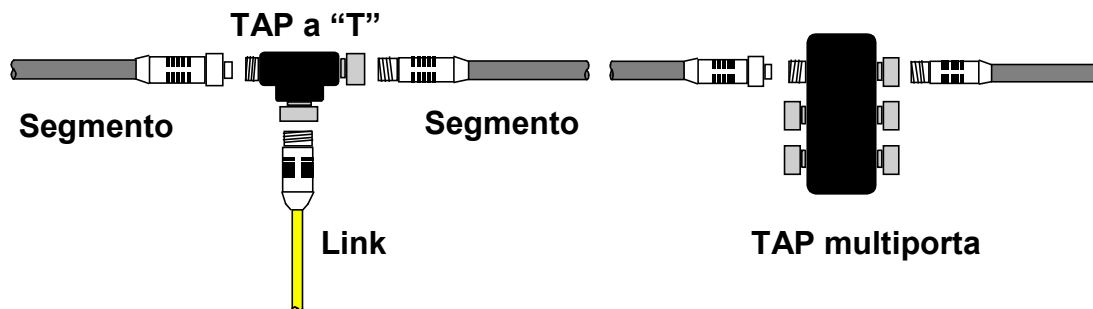
Connessioni CAN physical-layer.

CAN si avvale di un collegamento seriale su un bus. Il numero di stazioni connesse non ha un limite teorico. Praticamente, la garanzia di un delay sufficientemente piccolo limita il numero totale di unità connesse. Un altro limite può derivare dal carico elettrico: in effetti Bosch dichiara “più di 64 stazioni collegate” e “segnale ed alimentazione sullo stesso cavo (doppino doppio intrecciato)”. Segue una tabella riepilogativa:

Frequenza di trasmissione	Lunghezza bus (cavo spesso)	Lunghezza link	
		Massimo	Cumulativo
125 Kbit/s	500 m	6 m	156 m
250 Kbit/s	250 m	6 m	78 m
500 Kbit/s	100 m	6 m	39 m

Anche un cavo sottile può essere utilizzato per un segmento (lungo al più 100 m indipendentemente dalla frequenza trasmissiva).

Per **SEGMENTO** si intende una porzione del bus che può terminare o in un **TAP** (transceiver o bus driver) o su un'impedenza da 121 Ω . Per **LINK** si intende il collegamento (sottile) che va dal segmento ad un sensore. La figura che segue mostra quanto detto.



Tipica sezione di rete CAN.

Sleep mode.

Per ridurre il consumo di corrente elettrica, una stazione CAN può "assopirsi" (nessuna attività interna, disconnessione dai bus drivers). La stazione esce dallo sleep mode tramite un wake-up causato da qualche attività sul bus o da condizioni interne del sistema.

Per svegliare nodi in sleep mode esiste un messaggio dedicato di **wake up**: "11111101111" che deve essere inviato con l'ID più piccolo possibile.

Trasferimento di messaggi.

Tipi di messaggi.

Le informazioni viaggiano su frames di quattro tipi:

1. **DATA FRAME** (trasporta dati da una sorgente ad un ricevente);
2. **REMOTE FRAME** (viene inviato da una stazione per ottenere un DATA FRAME con lo stesso ID);
3. **ERROR FRAME** (trasmesso da qualsiasi stazione che rilevi un errore sul bus);
4. **OVERLOAD FRAME** (serve ad aggiungere un ritardo extra tra il REMOTE ed il DATA FRAME);

Data Frame.

START OF FRAME (1 bit)	
ARBITRATION FIELD (12 bits)	IDENTIFIER (11 bits)
	RTR BIT (1 bit)
CONTROL FIELD (6 bits)	Riservati (2 bits)
	DATA LENGTH CODE (4 bits)
DATA FIELD (0 → 8 bytes, variabile)	
CRC FIELD	CRC SEQUENCE (16 bits)
	CRC DELIMITER (1 bit)
ACK FIELD (2 bits)	ACK SLOT (1 bit)
	ACK DELIMITER (1 bit)
END OF FRAME (7 bits)	

Un **DATA FRAME** è composto di sette campi:

1. **START OF FRAME**: denota l'inizio del frame. È un singolo bit settato a "0".
2. **ARBITRATION FIELD**:
 - a. **IDENTIFIER**: 11 bits, trasmessi dal più significativo (MSB) al meno significativo (LSB). I primi sette bits **NON** devono essere tutti "1";
 - b. **RTR BIT** (Remote Transmission Request BIT): vale "0" se il frame è DATA, "1" se è REMOTE.
3. **CONTROL FIELD**:
 - a. **2 bits riservati** per espansioni future;
 - b. **DATA LENGHT CODE**: 4 bits ($2^4=16$, ma si utilizzano le sole combinazioni "0000" → "1000"), indicano la lunghezza (in bytes) dei dati contenuti nel campo DATA FIELD.

4. **DATA FIELD**: contiene i dati che vengono trasferiti col "DATA FRAME".
5. **CRC FIELD**:
 - a. **CRC SEQUENCE**;
 - b. **CRC DELIMITER**: bit che vale "1".
6. **ACK FIELD**: il trasmittente invia "11"; il ricevente, se ha ricevuto un frame con CRC valida, invia "01" al trasmittente (in caso di accesso simultaneo al canale, l'ack del frame viene fatto anche dal nodo che ha perso l'uso del bus).
7. **END OF FRAME**: consta di sette bits impostati ad "1".

Remote Frame.

Il REMOTE FRAME ha gli stessi campi del DATA FRAME tranne il campo DATA FIELD. L'RTR BIT è impostato a "1".

Error Frame.

Consiste di due campi:

- Il primo è l'**ERROR FLAG SUPERPOSITION**;
- Il secondo è l'**ERROR DELIMITER**.

Le stazioni CAN possono essere:

- "ERROR ACTIVE";
- "ERROR PASSIVE";

- Cosa è un "**ERROR FLAG**"? Una stringa di 6 bits: se viene inviata da una stazione ERROR ACTIVE, prende il nome di "ACTIVE ERROR FLAG" e vale "000000"; se, viceversa, viene inviata da una stazione ERROR PASSIVE, prende il nome di "PASSIVE ERROR FLAG" e vale "111111". Una stazione CAN che rileva un errore sul bus, lo rende noto inviando un ERROR FLAG. Mentre una stazione ERROR ACTIVE lo fa certamente, una stazione ERROR PASSIVE si limita a provarci (una stazione di questo tipo necessita di 3 bit time di idle del bus per terminare correttamente un ERROR FRAME). Poiché tale EF viola la codifica "bit stuffing", tutte le stazioni che rilevano tale errore trasmettono anche loro un EF. Sul bus possono, dunque, essere presenti più EF consecutivi (max 2) seguiti da un unico ERROR DELIMITER. Questi EF accodati costituiscono il campo ERROR FLAG SUPERPOSITION.

L'**ERROR DELIMITER** è una stringa di 8 bits settati ad "1".

Overload Frame.

Serve per garantire un rudimentale controllo di congestione. Anche esso contiene due campi:

- **OVERLOAD FLAG**;
- **OVERLOAD DELIMITER**.

Ci sono due ragioni per cui una stazione può generare un OVERLOAD FRAME:

1. Condizioni interne della stazione stessa, che necessita di un delay prima di ricevere il nuovo frame. In questo caso, il pacchetto (OF) viene inviato "non appena possibile". Una stazione non può inviare più di due pacchetti di OVERLOAD in sequenza.
2. La stazione rileva uno "0" durante la lettura dell'INTERMISSION (parte dell'INTERFRAME SPACE). In questo caso il pacchetto (OF) viene inviato un bit time dopo aver rilevato lo "0".

- **OVERLOAD FLAG**: è una stringa di 6 bits: "000000";
- **OVERLOAD DELIMITER**: è una stringa di 8 bits: "11111111".

Interframe Spacing.

INTERFRAME SPACE non è un periodo di tempo tra due frame, bensì un insieme di bit (un pacchetto speciale inviato dall'ultimo nodo che ha trasmesso) che separa tra loro frames in sequenza. In disaccordo con quanto detto: OVERLOAD FRAMES ed ERROR FRAMES non sono preceduti dall'INTERFRAME SPACE come anche non lo sono due OVERLOAD FRAMES consecutivi.

Il pacchetto ha due (tre) campi:

- **INTERMISSION** (3 bits settati ad "1"; quando sul bus passano i bits di INTERMISSION, nessuna stazione può trasmettere DATA o REMOTE FRAME: l'unica azione ammessa è la segnalazione di condizioni di overload);
- **BUS IDLE** (di lunghezza arbitraria [il campo ha tanti bits quanto vale $(frequenza_di_cifra) \cdot (periodo_di_bus_idle)$], è composto di bits "1": ogni stazione che deve trasmettere lo può fare, la presenza di un bit "0" sul canale viene interpretata come uno START OF FRAME);
- **SUSPEND TRASMISSION** (solo per stazioni ERROR PASSIVE; tale campo, se presente, è posto tra gli altri due. Una stazione ERROR PASSIVE, dopo l'invio di un messaggio, trasmette INTERMISSION seguito da 8 bits "1" prima di trasmettere un nuovo frame o di liberare il bus. Se durante tale periodo un'altra stazione comincia a trasmettere, questa si arresta).

Protocollo.

Definizione di TRASMETTITORE/RICEVITORE.

TRASMETTITORE: Una stazione che origina un messaggio è detta TRASMETTITORE del messaggio. Tale stazione permane in questo stato finché il bus è libero (idle) o perde l'ARBITRATION.

RICEVITORE: Una stazione è detta RICEVITORE se non sta trasmettendo ed il bus è occupato.

Message Validation.

Trasmettitore e ricevitore considerano VALIDO (buono) un frame in istanti di tempo diversi:

- **TRASMETTITORE:** Il trasmettitore considera valido un messaggio se ciò che ha trasmesso coincide con ciò che ha rilevato sul bus (dal primo all'ultimo bit). Se ciò non accade, il frame viene ritrasmesso automaticamente non appena il bus è libero (se qualche stazione deve trasmettere un messaggio con priorità più alta, viene ritrasmesso secondo le regole usuali di ARBITRATION).
- **RICEVITORE:** Il ricevitore considera valido un messaggio se non riscontra errori tra START OF FRAME e END OF FRAME.

Codifica.

Per tutti i bits dei campi: START OF FRAME, ARBITRATION FIELD, CONTROL FIELD, DATA FIELD e CRC SEQUENCE è adottata la regola di codifica

BIT STUFFING: il trasmettitore che rileva 5 bits uguali consecutivi nella stringa che sta trasmettendo, gli accoda un bit complementare (allo stesso modo un ricevitore che rileva 5 bits uguali elimina quello immediatamente successivo).

Gli altri campi dei pacchetti DATA FRAME e REMOTE FRAME (ACK FIELD ed END OF FRAME) sono fissati e non vengono, pertanto, codificati.

I pacchetti **ERROR FRAME** e **OVERLOAD FRAME** non vengono codificati.

Errori.

Si riscontrano cinque tipologie di errori (non mutuamente esclusive):

- **BIT ERROR**: Il bit “ascoltato” sul bus è diverso da quello trasmesso;
- **STUFF ERROR**: Un ricevitore rileva 6 bits uguali consecutivi che sarebbero dovuti essere codificati tramite *bit stuffing*;
- **CRC ERROR**: Ricezione di un pacchetto con CRC diverso da quello calcolato localmente;
- **FORM ERROR**: Un campo a “valori imposti” contiene uno o più bits non ammissibili;
- **ACK ERROR**: Si ha quando un trasmettitore non rileva “0” durante l’ACK SLOT.

Segnalazione di errori.

Quando una stazione rileva un errore lo segnala tramite un **ERROR FRAME** (ACTIVE o PASSIVE).

Quando l’errore rilevato è un CRC ERROR viene inviato l’ERROR FRAME subito dopo l’ultimo bit di ACK DELIMITER; per tutti gli altri errori, la trasmissione dell’ERROR FRAME comincia subito dopo il bit errato.

Isolamento dei guasti.

Una stazione può trovarsi in uno dei seguenti tre stati:

- ✓ **ERROR ACTIVE**;
- ✓ **ERROR PASSIVE**;
- ✓ **BUS OFF**: non ha alcuna influenza sul bus;

Su ogni unità, allo scopo di isolare i guasti, sono implementati due contatori:

- 1) **TRANSMIT ERROR COUNT**;
- 2) **RECEIVE ERROR COUNT**;

Tali contatori vengono incrementati secondo le seguenti regole:

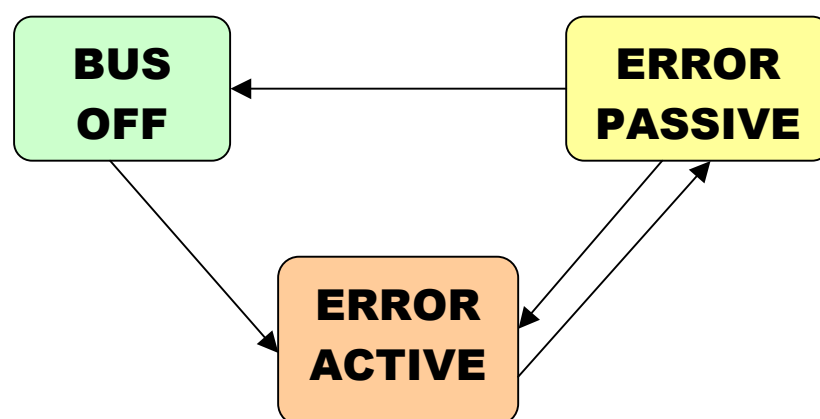
- a) Se un ricevitore rileva un errore, il suo RECEIVE ERROR COUNT viene incrementato di 1;
- b) Se un ricevitore rileva uno “0” dopo aver trasmesso un ERROR FLAG, il suo RECEIVE ERROR COUNT viene incrementato di 8;
- c) Se un trasmettitore invia un ERROR FLAG il suo TRANSMIT ERROR COUNT viene incrementato di 8;
- d) Se un trasmettitore rileva un BIT ERROR mentre invia un ACTIVE ERROR FLAG o un OVERLOAD FLAG, il suo TRANSMIT ERROR COUNT viene incrementato di 8;
- e) Se un ricevitore rileva un BIT ERROR mentre invia un ACTIVE ERROR FLAG o un OVERLOAD FLAG, il suo RECEIVE ERROR COUNT viene incrementato di 8;

- f) Qualche nodo tollera sequenze di più di 7 bits "0" consecutivi dopo aver inviato un ERROR FLAG od un OVERLOAD FLAG. Dopo aver rilevato il 14° "0" consecutivo (dopo la trasmissione di un ACTIVE ERROR FLAG o un OVERLOAD FLAG) o dopo aver rilevato l'8° "0" consecutivo (dopo la trasmissione di un PASSIVE ERROR FLAG) tutti i trasmettitori incrementano il proprio TRANSMIT ERROR COUNTER di 8 e tutti i ricevitori incrementano il proprio RECEIVE ERROR COUNTER di 8;
- g) Quando la trasmissione di un messaggio è andata a buon fine, il trasmettitore diminuisce di 1 il proprio TRANSMIT ERROR COUNT (se non valeva 0);
- h) Quando la ricezione di un messaggio è andata a buon fine, il ricevitore diminuisce di 1 il proprio RECEIVE ERROR COUNT (se era compreso tra 1 e 127) o lo imposta ad un valore compreso tra 119 e 127 (se era maggiore di 127) o lo lascia invariato (se valeva 0);

E modifica il proprio status in queste occasioni:

- Un nodo diviene **ERROR PASSIVE** se uno dei suoi due contatori assume valore maggiore o uguale a 128: quando ciò avviene, il nodo in questione invia un ACTIVE ERROR FLAG;
- Un nodo diviene **BUS OFF** quando il suo TRANSMIT ERROR COUNT assume valore maggiore o uguale a 256;
- Una stazione torna ad essere **ERROR ACTIVE** quando entrambi i suoi contatori assumono valori minori o uguali a 127;
- Un nodo che si trova nello stato di **BUS OFF** può tornare nello stato **ERROR ACTIVE** (con entrambi i contatori settati a 0) dopo aver monitorato per 128 volte la presenza di una stringa di 11 "1" sul bus.

NOTA: la presenza di contatori con valore maggiore di 96 indica che il bus è molto disturbato.



Extended version (CAN v1.2).

Questa sezione vuole evidenziare le differenze tra la vecchia (1.1) e la nuova specifica di CAN (1.2).

Uniformazione dei livelli alla pila ISO-OSI.

(cfr. pagg. 3, 4).

Il livello	<i>CAN object layer</i>	diventa	<i>sottolivello LLC</i> ;
Il livello	<i>CAN transfer layer</i>	diventa	<i>sottolivello MAC</i> ;

Entrambi (LLC e MAC) sono sottolivelli standard del livello 2 (data link) della pila ISO-OSI. Il cambiamento è riferito alla sola nomenclatura: la sostanza è rimasta immutata.

Trasferimento di messaggi ed indirizzamento.

(cfr. pag. 9).

Nella versione estesa di CAN i pacchetti DATA e REMOTE FRAME hanno ognuno due diversi formati (che differiscono per la lunghezza dell'ID):

1. *STANDARD FRAME* (ID di 11 bits);
2. *EXTENDED FRAME* (ID di 29 bits);

Per mantenere la compatibilità con la precedente versione i pacchetti *STANDARD FRAME* sono identici a quelli descritti nelle prime pagine.

Formato di pacchetto: il campo "ARBITRATION FIELD".

(cfr. pagg. 6, 9).

Il campo *ARBITRATION FIELD* per uno *STANDARD FRAME* è quello già visto (pag. 9; 11 bits di ID + 1 bit RTR); in un pacchetto *EXTENDED*, viceversa, il campo *ARBITRATION* è composto di:

- **29 bits di IDENTIFIER**: divisi in 11 bits di *BASE ID* (disposti secondo il protocollo standard con i primi 7 bits non tutti a "1"; su di essi si stabiliscono le priorità) e 18 bits di *EXTENDED ID*;
- **1 bit SRR** (Substitute Remote Request): è un bit settato ad "1" che occupa la posizione fisica dell'RTR nel pacchetto standard (si trova immediatamente dopo gli 11 bits di *BASE ID*);
- **1 bit IDE** (Identifier Extension): tale bit è posto immediatamente dietro l'SRR, nel caso di *EXTENDED FRAME* è impostato ad "1" e seguito dall'*EXTENDED ID*, nel caso di *STANDARD FRAME* è impostato a "0" e coincide con uno dei 2 bits riservati del campo *CONTROL FIELD*;
- **1 bit RTR** (come nel CAN standard, per stabilire se il frame è DATA o REMOTE, è l'ultimo bit del campo *ARBITRATION*).