

Università degli studi Roma Tre



Dipartimento di Informatica e Automazione

Corso di Controllo Digitale

Prof. Lorenzo Sciavicco Prof. Stefano Panzieri

CAN Controller Area Network

Daniele Gallinella

BOSCH



La comunicazione nei sistemi di automazione

La situazione dei sistemi di automazione industriale era, fino a poco tempo fa, notevolmente diversa da quella che ci appare oggi e che, certamente, si presenterà nel prossimo futuro.

Inizialmente avevamo un'unica unità centrale con elevata potenza di calcolo e tante stazioni periferiche, le quali fornivano informazioni solo quando il mainframe ne aveva bisogno. Esso aveva il compito di eseguire tutte le operazioni di controllo sull'intero sistema e poi ridistribuiva i compiti ad ognuno dei dispositivi periferici.

Il tutto si basava su un tipo di comunicazione punto-punto, ovvero ogni stazione era connessa direttamente all'unità centrale; è naturale che ciò portava a un numero enorme di cablaggi e a spese di installazione e manutenzione di non poco conto.

Oggi lo sviluppo dei microprocessori ha consentito ai dispositivi di poter eseguire per proprio conto le operazioni di cui necessitano, decentralizzando, così, il controllo del sistema, e facendo nascere l'esigenza di un metodo efficiente di scambio dei dati; si è tentato di raggiungere i seguenti scopi:

- riduzione dei cablaggi
- utilizzo di interfacce digitali
- standardizzazione delle interfacce
- facilità di installazione
- interoperabilità e intercambiabilità dei dispositivi connessi

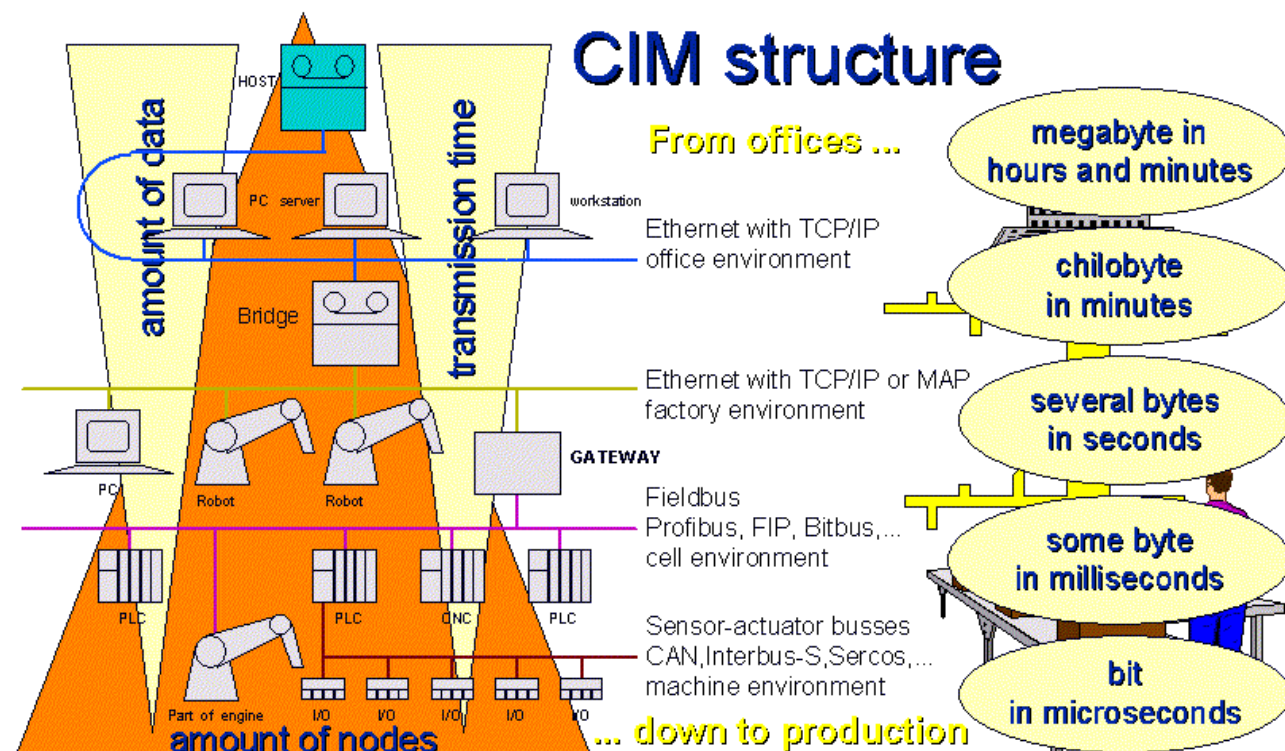


Figura uno: la piramide CIM

La piramide CIM (Computer Integrated Manufacturing) ci mostra come all'interno di un'azienda le esigenze di comunicazione siano contrastanti: tipicamente ai livelli alti, quelli riguardanti la gestione degli stabilimenti, bisogna poter trasferire grosse quantità di informazioni ad una frequenza non elevata; ai livelli bassi, campo e controllo, i dati sono di pochi byte o bit, ma essi vanno spediti molto più frequentemente (milli o micro secondi).

Abbiamo così diversi tipi di reti dipendentemente dal livello al quale ci collochiamo, con caratteristiche e standard diversi.

In particolare fissiamo la nostra attenzione sulle reti di campo.

Esse consentono l'integrazione tra PLC e dispositivi di controllo con sensori e attuatori. La comunicazione deve essere snella e veloce con protocolli semplici che producano poco carico di lavoro per i microprocessori. Per questo motivo vengono realizzate funzioni che riguardano i soli primi due livelli della pila ISO/OSI, con alcune eccezioni riguardanti il livello 7 quando necessario.

Tra questo tipo di reti, tra i più noti, troviamo CAN.

Cos'è CAN

CAN (Controller Area Network) è un bus seriale di comunicazione dati progettato per applicazioni real-time. Nacque originariamente per l'industria automobilistica, ma si è diffuso presto nell'automazione industriale per le sue caratteristiche di robustezza ed affidabilità.

CAN è stato sviluppato da Robert Bosch nel lontano 1986 su richiesta della Mercedes. L'esigenza di far comunicare i molti dispositivi elettronici presenti all'interno delle automobili (Antilock Braking System, Traction Control, Air Conditioning Control, chiusura centralizzata, sono solo alcuni esempi) e la complessità di questi, avrebbe portato ad un aumento insostenibile di collegamenti dedicati ed una duplicazione dei sensori necessari a più dispositivi, con conseguente aumento dei costi di produzione e soprattutto notevole ingombro fisico.

Per questi motivi è nato CAN, il quale consente a controllori, sensori e attuatori di comunicare l'uno con l'altro ad una velocità fino a 1Mbit/sec, offrendo anche:

- bassi costi di progettazione e implementazione
- funzionamento in ambienti ostili
- facilità di configurazione e modifica
- rilevamento automatico degli errori di trasmissione

Caratteristiche

Il protocollo CAN si colloca all'interno della pila ISO/OSI nei livelli 1 fisico e 2 data link, lasciando totale libertà per il livello 7 applicazione.

Il nucleo del protocollo è nel livello data link, e nella particolare politica di accesso al mezzo di trasmissione.

- Assenza di indirizzi mittente/destinatario

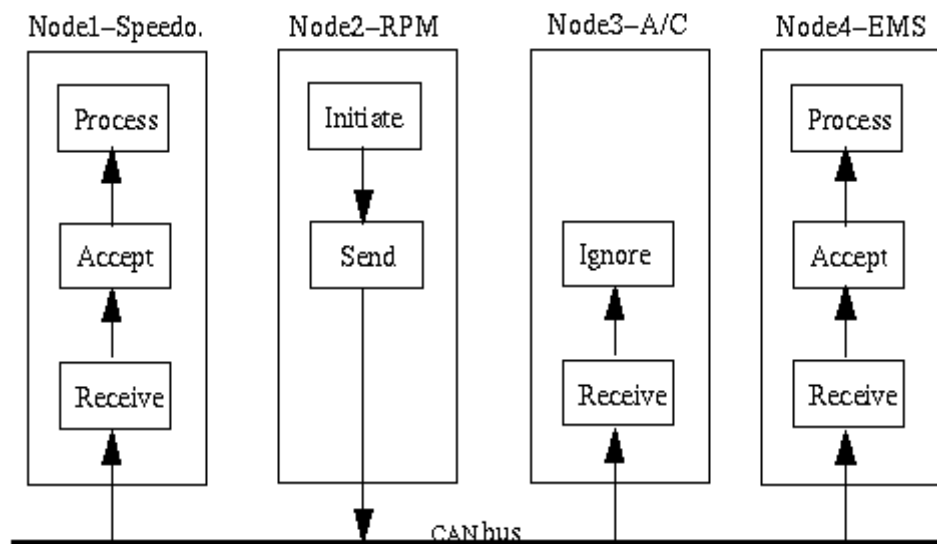


Figura due: Message Filtering

I pacchetti trasmessi da una stazione CAN non contengono indirizzi di alcun genere, al loro posto troviamo un identificatore del contenuto del messaggio (giri al minuto, temperatura motore, temperatura abitacolo ...) unico sull'intera rete.

Un nodo ricevitore può così verificare il contenuto del messaggio e filtrare i soli pacchetti a cui è interessato, ignorando gli altri. Questo modo di operare è detto Multicast.

Nella figura due viene rappresentata la situazione di un nodo che trasmette dati riguardanti i giri al minuto del motore e mentre il tachimetro e l'Engine Management System accettano il pacchetto, l'unità di gestione dell'aria condizionata lo ignora.

La caratteristica content-oriented del CAN permette un alto grado di flessibilità e modularità del sistema, consentendo che nuove stazioni che sono solo ricevitori e che hanno bisogno dei soli dati esistenti possano essere aggiunte senza alcuna modifica né all'hardware né al software.

- Non-destructive bitwise arbitration

Per garantire la comunicazione real-time il CAN ricorre ad un metodo di accesso al mezzo CSMA/CD modificato. Due o più stazioni che iniziano a trasmettere competono per l'accesso al bus sul loro valore di priorità determinato proprio dall'identificatore. Quello con il valore numerico più basso vince la competizione per il canale.

Ogni conflitto viene risolto in accordo con il meccanismo del wired-and, secondo il quale lo stato dominante (0 logico) sovrascrive lo stato recessivo (1 logico).

In figura tre si può notare la competizione tra tre nodi che tentano di trasmettere allo stesso istante e il segnale che si legge sul bus. Il nodo 1 perde la competizione al bit 7, mentre il nodo 3 perde al bit 3; in questo modo il nodo 2 continua a trasmettere come se fosse stato solo senza il bisogno di ritrasmissioni; i nodi perdenti diventano subito ricevitori e non tenteranno una ritrasmissione non prima che il bus sia diventato libero.

Questa politica garantisce il determinismo dell'accesso al bus, e l'assenza di periodi di inattività del canale.

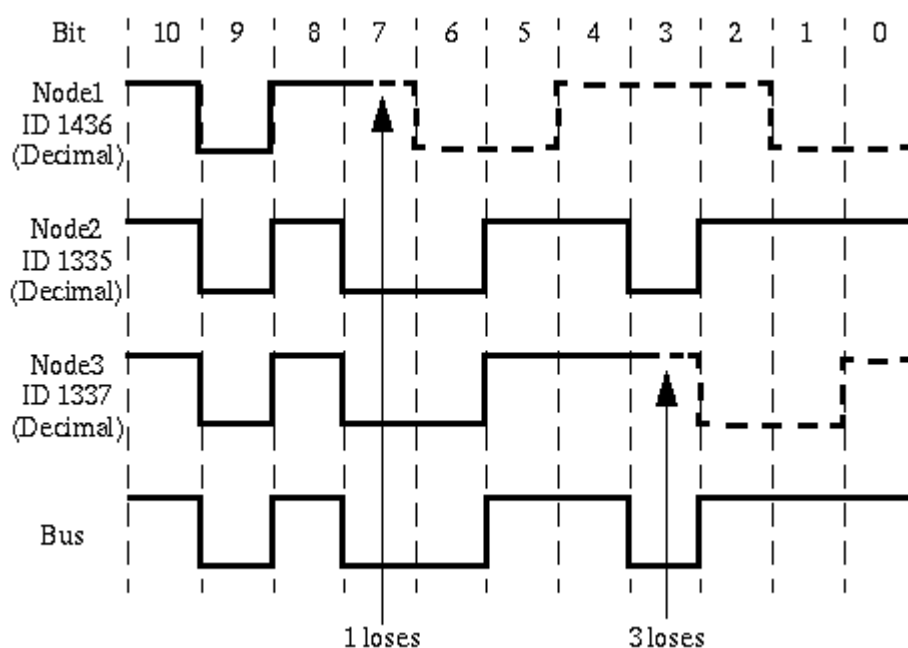


Figura tre: arbitraggio di accesso al bus

Formato dei messaggi CAN

CAN prevede 4 tipi di messaggi:

Data Frame, Remote Frame, Error Frame e Overload Frame.

Ci sono due formati per i Data Frame e Remote Frame:

- Standard CAN (Versione 2.0 A)
- Extended CAN (Versione 2.0 B)

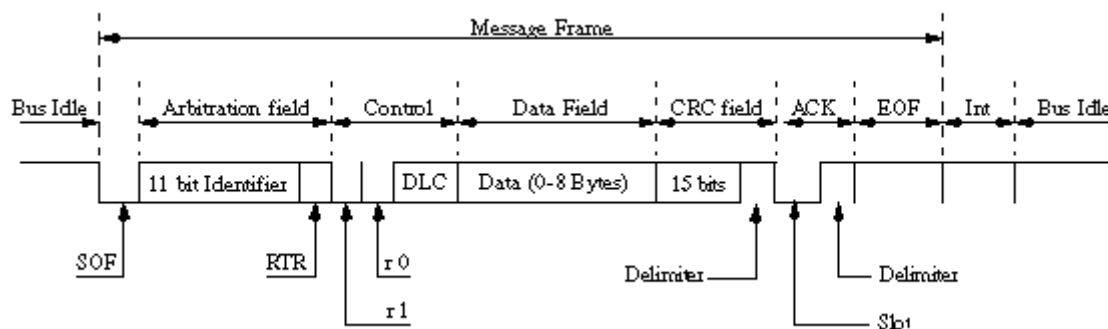


Figura quattro: formato del Message Frame per lo Standard CAN 2.0A

Lo Standard CAN 2.0 A prevede che il data frame sia strutturato come segue:

- Un campo Start Of Frame (SOF). È un bit dominante (0 logico) che indica l'inizio di un message frame. Il rilevamento di un bit dominante durante il Bus Idle è sempre interpretato come un SOF.
- Un Arbitration Field, contenente 11 bit di identificatore e il Remote Transmission Request (RTR) bit. Quest'ultimo quando è settato a 0 indica che il frame è un Data Frame, se settato a 1 indica che è un Remote Frame. Il Remote Frame è una richiesta da parte di un nodo del Data Frame corrispondente (avente lo stesso identificatore) e non presenta un campo dati.
- Un Control Field contenente 6 bit: 2 bit riservati per usi futuri e 4 bit di Data Length Code (DLC). Questo indica il numero dei byte nel Data Field seguente, esso può variare da 0 a 8 byte.
- Il CRC Field contenente 15 bit di cyclic redundancy check code e un bit recessivo come delimitatore.
- Il campo Ack di 2 bit; il primo è lo Slot Ack che è trasmesso come recessivo, ma è sovrascritto con un bit dominante da ogni stazione che riceve correttamente il messaggio; il secondo bit è recessivo e svolge il compito di delimitatore.
- Il campo End Of Frame (EOF) che consiste di 7 recessive bit.

Lo standard CAN 2.0 B prevede un identificatore di 29 bit, per offrire compatibilità con altri protocolli di comunicazione seriale usati in USA.

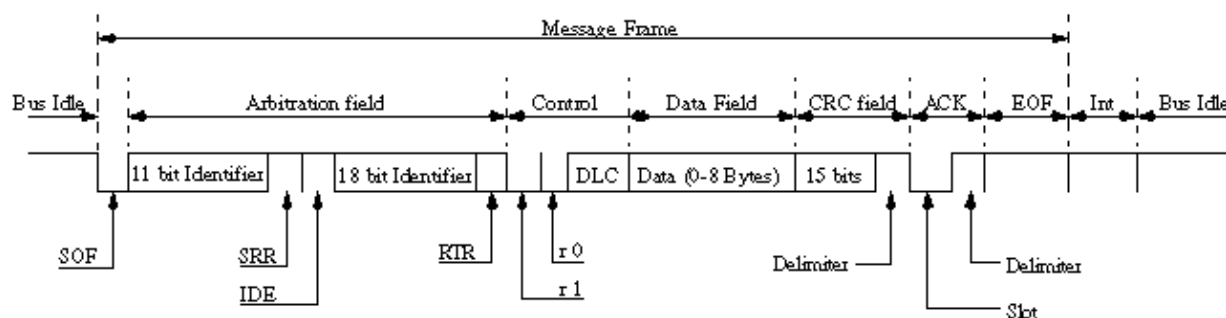


Figura cinque: formato del Message Frame per l' Extended CAN 2.0B

L'identificatore è diviso in Base ID lungo 11 bit per garantire la compatibilità con la versione A e in Extension ID di 18 bit.

Il Substitute Remote Request (SRR) bit è spedito sempre recessive in modo che in caso di competizione con un frame della versione A quest'ultimo vinca sempre.

L' Identifier Extension (IDE) bit è spedito recessive se il nodo sta spedendo messaggi nel formato Extended o dominante se spedisce messaggi nel formato Standard.

Esistono tre tipi di controllori CAN: i 2.0A che sono capaci di spedire solo messaggi di formato Standard, restituendo errore nel caso ricevano in formato Extended; i 2.0B passive che sono in grado di spedire solo in formato Standard, ma possono ricevere in formato Extended ignorandoli; i 2.0B che possono funzionare in entrambe le modalità.

Rilevamento degli errori

Il rilevamento e la segnalazione degli errori insieme all'isolamento dei guasti rendono CAN veramente affidabile, garantendo la correttezza delle informazioni e la loro coerenza sull'intero sistema.

Il processo di segnalazione degli errori si articola nelle seguenti fasi:

- Un controller CAN rileva un errore (in trasmissione o in ricezione)
- Un Error Frame viene immediatamente trasmesso
- Il messaggio incriminato viene ignorato da tutti i nodi
- Viene aggiornato lo stato del controller CAN
- Il messaggio viene ritrasmesso, eventualmente competendo con altri.

Un errore può essere rilevato in 5 modi, 3 dei quali a livello del messaggio e 2 a livello del singolo bit:

- Bit Stuffing Error; normalmente un nodo in trasmissione inserisce dopo 5 bit consecutivi della stesa polarità un bit di polarità opposta; ciò è chiamato bit stuffing. Un nodo che riceve più di 5 bit consecutivi di segno uguale rileverà un errore di questo tipo.
- Bit Error; un nodo in trasmissione ascolta sempre il bus per verificare la corrispondenza con ciò che sta trasmettendo: se esso ascolta un bit diverso dal suo (e non ci troviamo nell'Arbitration Field né nell'Ack Slot) verrà segnalato un errore
- Checksum Error; ogni nodo ricevente ricalcola il CRC in base a ciò che ha ricevuto, e se non corrisponde a quello inviato dal mittente viene segnalato un errore
- Frame Error; viene segnalato questo tipo di errore quando vengono violati alcuni campi fissi del pacchetto (bit che devono essere spediti sempre dello stesso tipo).
- Acknowledgement Error; se il trasmettitore non rileva alcun riscontro al frame appena inviato.

Un Frame Error è costituito da un Error Flag ed un Error Delimiter.

L'Error Flag è lungo 6 bit dello stesso segno e viola volontariamente la regola dello bit stuffing in modo che tutte le altre stazioni rilevino un errore e spediscono anch'esse un Error Flag. Per questo motivo il campo Error Flag nel pacchetto è di lunghezza variabile (max 12 bit) dato dalla sovrapposizione di tutti gli Error Flag spediti.

A seguito c'è un Error Delimiter costituito da 8 bit recessive.

CAN offre un meccanismo di auto isolamento dei guasti unico nel suo genere. Esso è in grado di distinguere tra condizioni di guasto transitorie (sbalzi di voltaggio, condizioni esterne di disturbo), e guasti permanenti (cattive connessioni, cavi rotti).

Ogni controller mantiene due registri error count uno di trasmissione e uno di ricezione: essi sono inizialmente settati a 0 e vengono incrementati ogni qualvolta si presenta una situazione di errore (+1 per un errore in ricezione, +8 per un errore in trasmissione).

A sua volta ogni stazione può trovarsi in tre stati:

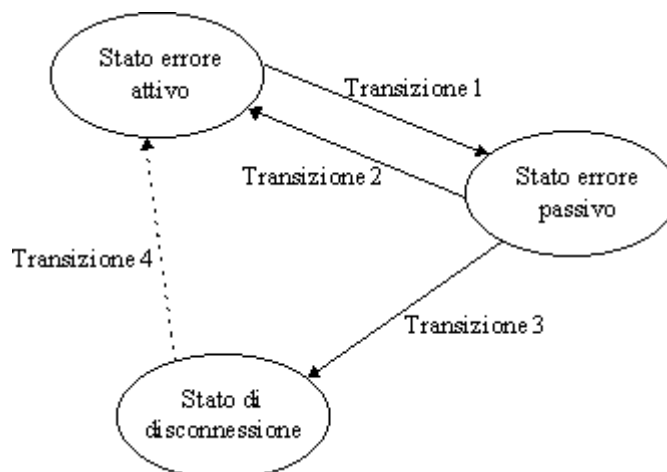


Figura sei: diagramma a stati finiti per l'autodiagnostica

- Error Active. Nessuno dei due contatori ha superato il valore di 127. Il nodo è nel pieno delle sue funzionalità e decrementa di 1 i contatori ogni volta che riceve un messaggio andato a buon fine. Quando è in questo stato il nodo che rileva un errore spedisce un Error Flag costituito da 6 bit dominanti in modo da interrompere sempre la trasmissione.
- Error Passive. Almeno uno dei due contatori ha superato 127. Il nodo è ancora in grado di eseguire tutte le sue funzioni, ma è probabile che esso presenti dei disturbi o condizioni di guasto. Per questo quando esso rileva un errore spedisce un Error Flag di 6 bit recessive che vengono interpretati come errore solo se nessuna stazione sta spedendo un suo proprio messaggio (i bit recessive contrariamente vengono sovrascritti).

- **Bus Off.** Se uno dei contatori supera 255 il nodo si stacca dal bus e non partecipa alla comunicazione lasciando gli altri nodi nella possibilità di continuare a scambiarsi informazioni. Se ciò accade certamente la stazione presenta un problema permanente che necessita di un intervento esterno per ripristinare il perfetto funzionamento. Alcune implementazioni consentono al nodo di tornare Error Active dopo che esso abbia ricevuto 128 occorrenze di 11 bit recessive consecutivi (128 messaggi andati a buon fine), altre necessitano di un reset hardware.

L'alta affidabilità del CAN può essere misurata con la capacità di identificare dati corrotti da guasti di trasmissione. La probabilità residua d'errore è una misura statistica di questo: specifica la probabilità che un messaggio sia corrotto e non diagnosticato tale da nessuno sulla rete. Questa probabilità dovrebbe essere così piccola che nessun messaggio errato rimanga irrilevato su tutto il ciclo di vita del sistema.

È stato calcolato che su una rete operante a 1 Mbit/sec, con un'utilizzazione media del bus del 50%, una lunghezza media dei messaggi di 80 bit, e un tempo di lavorazione di 8 ore al giorno per 365 l'anno, un errore irrilevato capiterà ogni 1000 anni.

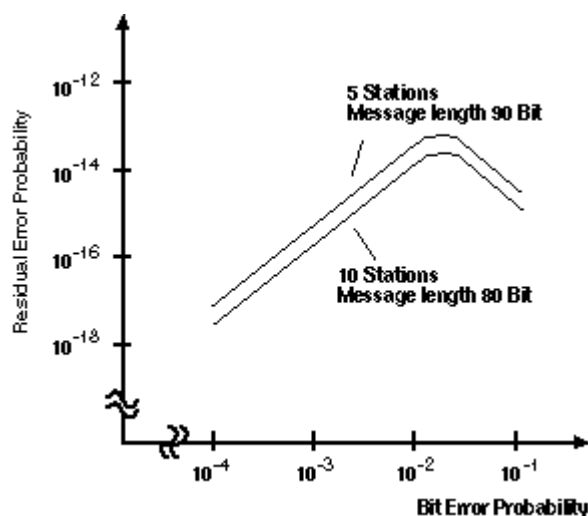


Figura sette: probabilità residua di errore come funzione della probabilità di errore sul bit

Overload Frame e Interframe Spacing

Un Overload Frame è simile ad un Error Frame poiché anch'esso consiste di un Overload Flag e di un Overload Delimiter.

Esso può essere generato da due condizioni:

1. Quando un ricevitore ha bisogno di più tempo per processare i dati correnti prima che altri vengano ricevuti

2. Il rilevamento di un bit dominante durante l'Intermission Field

Un nodo trasmetterà l'overload flag subito dopo l'EOF a causa della condizione 1, o subito dopo il bit che ha provocato la condizione 2. In questo modo tutti gli altri nodi diagnosticheranno una condizione di overload e spediranno anch'esse un overload flag. A seguito della sovrapposizione dei flag ci saranno 8 bit recessive di delimitare.

Un Overload Frame non richiede la ritrasmissione del frame che ha causato la condizione di overload.

L'Interframe Spacing separa i Data Frame e i Remote Frame da qualsiasi tipo di frame precedente, mentre gli Error Frame e gli Overload Frame non sono preceduti da nulla. Esso consiste in tre bit recessive che costituiscono l'Intermission Field dopo i quali si entra nella condizione di Bus Idle. Ora il bus è considerato libero ed ogni nodo può cominciare ad inviare un frame.

CAN physical layer

Il livello fisico del CAN è stato standardizzato in accordo con ISO 11898.

Il cavo trasmissivo deve essere un doppino intrecciato sia schermato sia non schermato. L'impedenza che termina il cavo deve essere di 120 ohm.

La lunghezza massima del bus dipende dalla velocità trasmissiva usata. È richiesto che il fronte di salita di un bit possa viaggiare lungo tutto il bus e poi tornare indietro prima che il bit sia campionato.

Lunghezza del bus (metri)	Max bit rate (bit/sec)
40	1 Mbit/s
100	500 Kbit/s
200	250 Kbit/sec
500	125 Kbit/s
6K	10 Kbit/s

Implementazioni del CAN

Non c'è uno standard per come i controller CAN devono essere implementati o come loro dovrebbero comunicare con i loro microcontroller.

Ci sono due principali strategie di implementazione per i controller CAN: le sostanziali differenze tra le due sono su come vengono filtrati i messaggi, su come vengono bufferizzati e su come avvengono le risposte ai Remote Frames.

Basic CAN

Il Basic CAN è un controller più economico. Normalmente è un chip a parte rispetto al microcontrollore a cui si interfaccia e per questo motivo più ingombrante. Esso possiede dei buffer di ricezione e trasmissione gestiti con politica FIFO; un messaggio può essere ricevuto su un buffer mentre il microcontrollore sta leggendo su un altro buffer; se arriva un messaggio quando tutti i buffer sono pieni, il più vecchio viene conservato, il che significa che possono andare perse delle informazioni nell'eventualità che il microcontrollore non sia veloce abbastanza.

Un messaggio è inviato scrivendolo in un buffer di trasmissione.

I messaggi a cui il nodo è interessato sono filtrati usando due registri che operano sull'identificatore del messaggio: se l'identificatore supera il controllo di tali maschere viene registrato in un buffer.

Ogni bit della maschera può essere settato ad 1, a 0 o Don't Care; la scelta va fatta molto accuratamente perché ogni messaggio che supera tale controllo viene processato dal microcontrollore e il filtraggio finale verrà effettuato via software, generando carico di lavoro non necessario.

Il Basic CAN non supporta la risposta automatica ai Remote Frames, ma l'applicazione software dovrà gestirli, garantendo da una parte la correttezza della risposta, ma un extra carico sul processore.

Full CAN

Il Full CAN è un controller più performante e più costoso del Basic. Esso costituisce un unico chip con il microcontrollore sul quale è montato.

Esso ha un insieme di buffer chiamati mailboxes i quali al momento dell'inizializzazione del sistema sono settati in trasmissione o in ricezione e ad ognuno viene assegnato un identificatore; questo significa che ad ogni messaggio compete il proprio buffer specifico.

Quando viene ricevuto un messaggio vengono controllati tutti i buffer di ricezione cercando quello avente l'identificatore in questione: se viene trovato significa che il messaggio ha un contenuto rilevante per il nodo e viene memorizzato, altrimenti viene scartato in quanto non interessante. In questo modo il filtraggio avviene interamente senza che se ne occupi il processore.

Lo stesso in trasmissione, il messaggio viene memorizzato nel buffer che gli compete; in questo modo può essere attuata anche una politica di selezione del messaggio da trasmettere, favorendo il più prioritario e non una semplice FIFO come nel Basic.

Quando viene ricevuto un Remote Frame, il controller CAN verifica se esiste un buffer di trasmissione con lo stesso identificatore: se sì, il Data Frame corrispondente viene subito inviato senza chiamare in causa il microcontrollore e snellendo di conseguenza i suoi compiti. Ciò, però, può portare ad un invio di dati ormai vecchi che non sono stati ancora aggiornati, ma che risiedevano nel buffer. Diviene così molto importante la frequenza con cui i dati nei buffer vengono aggiornati.

Serial Link I/O devices

Gli SLIOs sono controller CAN che non implementano nessuna funzione di microcontroller. Essi sono slaves amministrati da un master controller che risiede da qualche altra parte sulla rete.

Implementazioni disponibili

Il mercato offre un'ampia gamma di dispositivi CAN di ogni tipo descritto. Ne viene proposta una lista non completa dei produttori più importanti:

Intel

- AN82527. Un Full CAN controller che supporta entrambi Standard (V2.0A) e Extended (V2.0B) formats. Configurabile per interfacciamento a CPU a 8 o 16 bit.
- 80196CA. Microcontroller con montato l'82527 CAN controller.

Motorola

- 68HC05X4, 68HC05X16. Microcontrollers con Basic CAN controllers integrati per Standard (V 2.0A) format. I suffissi X4 and X16 si riferiscono al numero dei Kilobytes della RAM sul chip.

National Semiconductor

- COP684BC, COP884BC. Microcontrollers con Basic CAN controllers integrati per Standard (V 2.0A) and 2.0B passive formats
- MM57C360, MM57C362. Serial Linked I/O devices. (SLIOs)

Philips

- PCA82C200. Un Basic CAN controller che supporta Standard CAN (V2.0A).
- P80CE598 (ROM less version), P83CE598 (ROM version), P87CE598 (EPROM/One Time Programming version). Single chip, 8 bit microcontrollers (80C51) con un chip 82C200 CAN controller.
- 82C150. Serial Linked I/O device (SLIO) con CAN controller per V2.0A e V2.0B passive.

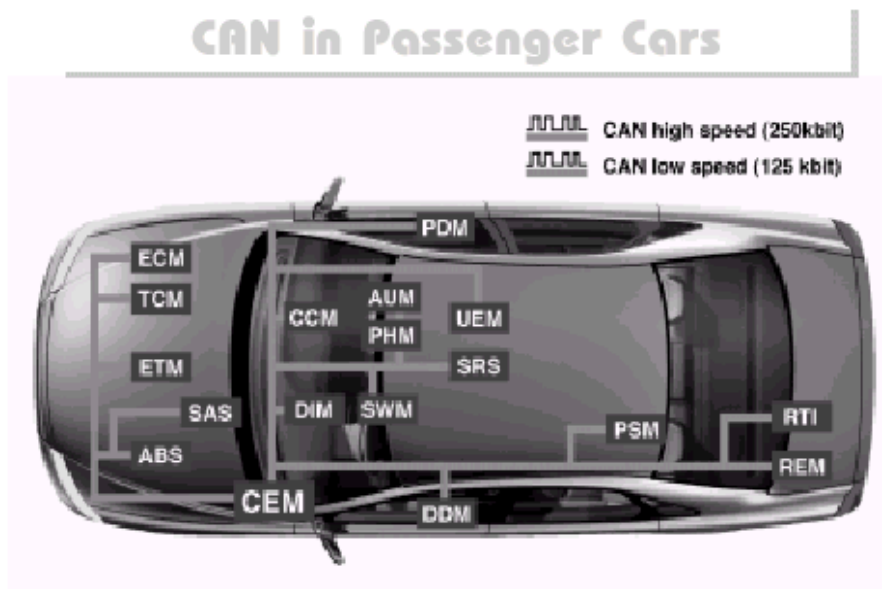
Siemens

- SAE 81C91. Uno stand alone Full CAN controller (V 2.0A and 2.0B passive).
- SAE 81C90. Specifiche identiche al C91, ma senza le due porte parallele.
- 80C167CR. Un 16 bit microcontroller con montato un Full CAN controller. Supporta Standard CAN (V 2.0A) and Extended CAN (V 2.0B).

Applicazioni industriali del CAN



CAN è stato originariamente sviluppato per essere usato nelle automobili, ed infatti è soprattutto in questo campo che troviamo un suo largo utilizzo.



Ma anche in altri settori esso viene largamente apprezzato per i suoi particolari vantaggi:

- bassi costi di progettazione e implementazione
- operatività in condizioni critiche

- facilità di configurazione e modifica
- rilevamento automatico degli errori
- diagnosi centralizzata dei guasti
- ottima affidabilità e sicurezza

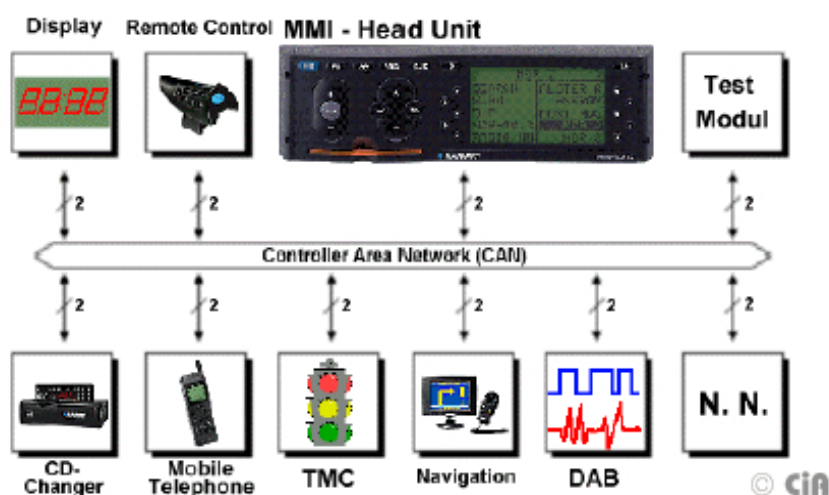
La maggior parte delle case automobilistiche fa uso di reti CAN.

A cominciare dalla Daimler-Mercedes Benz, per continuare con Audi, BMW, Renault, Saab, Volkswagen, Volvo.

Diversi tipi di CAN vengono usati all'interno dell'auto; una high-speed per la gestione del motore e una low-speed per l'elettronica interna come il controllo degli sportelli

Un terzo tipo di rete connette tutti gli apparati di intrattenimento presenti all'interno dell'abitacolo come la radio, il compact disc, il cellulare, il sistema di navigazione satellitare

Entertainment in Motor Vehicles



Altre applicazioni le troviamo nell'elettronica marina, nel controllo degli ascensori, nelle macchine agricole, nelle fotocopiatrici, perfino in alcuni giocattoli.

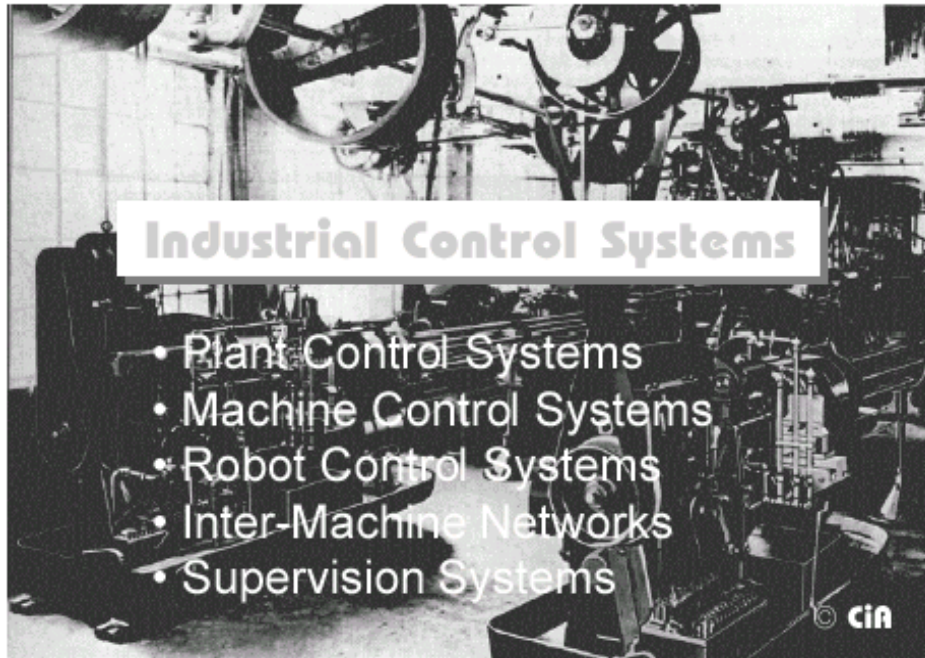
Discorso a parte merita il settore dell'industria medica, dove il CAN è apprezzato moltissimo per le sue funzionalità riguardanti la sicurezza e l'affidabilità che in questo campo sono le più rigorose.

Per questo non è raro che apparecchiature quali macchine per raggi X, facciano uso del CAN, così come altre apparecchiature speciali come i telescopi.

Ugualmente i sistemi di controllo della produzione industriale.

Moltissime aziende manifatturiere ricorrono al CAN. Il settore tessile è stato uno dei pionieri ed ha aperto la porte al CAN per entrare nelle linee di produzione.

Ora è presente anche in macchine per la stampa, macchine per l'imballaggio, macchine per la lavorazione del legno, celle di produzione, centri per test di qualità e tanti altri.



Un altro importante campo di applicazione nell'automazione industriale è il controllo di robot. ABB, Bosch, Engel, Kuka e altri hanno già implementato reti CAN.

