

Elettronica In

www.elettronica.in.it

€ 6,00 - Anno XXVI - n. 243 - MARZO 2020



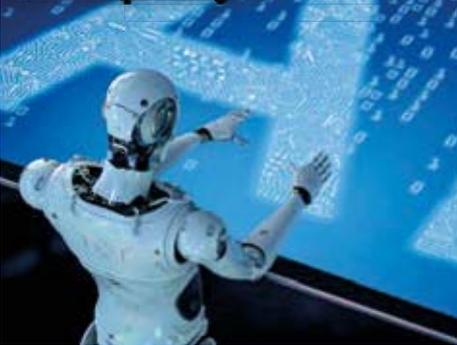
Elettronica In - Futura Group srl
Via Adige, 11
2013 Gallarate (VA) - Italy
Numero di marzo 2020 distribuito gratuitamente all'indirizzo
https://www.elettronica.in.it/flipping_demo/EI243_marzo_2020/
estratto delle pagine da 83 a 94

telecamera wireless con ESP32



- I bus per automotive
- I²C Extender
- Circuito giratore
- Domotica per tutti
- Sirena bitonale

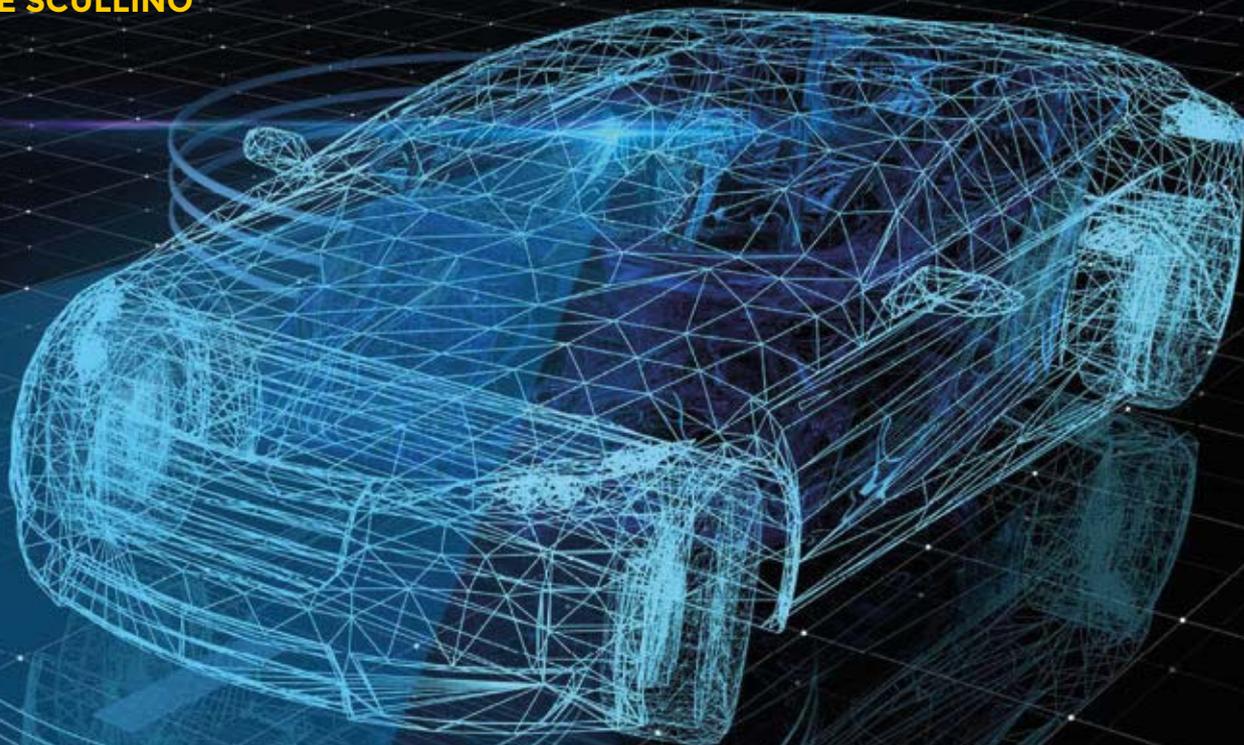
Intelligenza artificiale su Raspberry Pi



Bike tracking con SigFox



di DAVIDE SCULLINO



I BUS PER AUTOMOTIVE

Le automobili contengono sempre più elettronica, coordinata e interfacciata mediante link di comunicazione che possono essere bus o vere e proprie reti locali. Studiamole nella prospettiva dell'avvento dell'ADAS.

V

iste da sotto il cofano, le automobili dei nostri tempi sono molto diverse da quelle prodotte anche solo 30 anni fa e chi crede che l'elettronica riguardi solo l'auto elettrica, guardando bene dentro un'automobile a motore endotermico di ultima generazione arriverebbe a ricredersi. Infatti

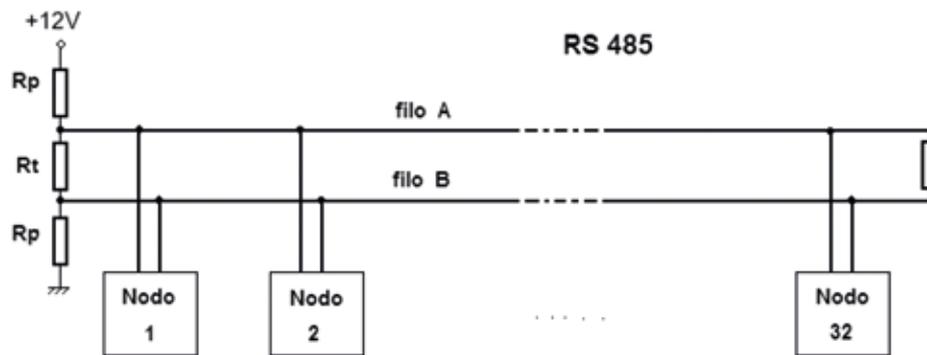
oggi una parte della meccanica viene soppiantata dall'elettronica e sensori, attuatori e microcalcolatori sono distribuiti a bordo in grande quantità; diverranno ancor di più con il crescere dei livelli di assistenza alla guida definiti da quello che ormai conosciamo come ADAS.

Le ragioni dell'elettronica a bordo delle auto sono molteplici e pesano in varia misura: qualcuna è necessaria e dettata da ragioni di sicurezza, di ecologia ed economia, qualcun'altra da questioni di marketing.

La prima comparsa dell'elettronica a bordo delle auto si deve a due fattori: ottimizzazione del funzionamento del motore e miglioramento della vita di bordo. Vediamo coinvolta l'elettronica in questi campi:

- gestione motore, per ottimizzare i consumi e ridurre le emissioni inquinanti;

Fig. 1
 Tipico bus fisico
 del CAN.



- gestione di trasmissione, sterzo e sospensioni, per accrescere il livello di sicurezza e guidabilità;
- automazione della climatizzazione e delle funzioni normalmente manuali (chiusura porte, alzacristalli ecc.);
- realizzazione di sistemi di infotainment sempre più sofisticati;
- connessioni via rete mobile.

L'elettronica dei primi tempi, limitata essenzialmente alla gestione del motore e della vita a bordo, era stand-alone, nel senso che ogni sistema viveva di vita propria, funzionava in maniera specifica ed era confinato all'ambito per il quale era stato progettato. Al crescere della complessità dei sistemi è sorta l'esigenza di creare delle connessioni standard e soprattutto versatili ed espandibili, capaci di realizzare soluzioni scalabili (ad esempio per aggiungere sensori, attuatori e funzionalità) sia di integrare, all'occorrenza, i vari sottosistemi. Sono nati così i bus per automotive, che nelle loro formulazioni iniziali erano abbastanza semplici, ma che oggi sono approdati a vere e proprie reti informatiche: un esempio per tutti è la 100BASE-T1. In questo articolo faremo una carrellata sui principali link utilizzati nell'automobile moderna conoscendone peculiarità, ambiti applicativi, passato, presente e sviluppi futuri.

IL CAN-BUS

Se si parla di link dati per l'automotive viene in mente prima di tutto il CAN-Bus, nato in casa Bosch per armonizzare i sistemi di gestione motore, i quali soprattutto per rispettare le sempre più stringenti normative ambientali sono cresciuti di complessità, al punto che la più elevata concentrazione di dispositivi elettronici in un'automobile è nel gruppo motopropulsore ed è lì per cercare di fare l'impossibile, ossia annullare le emissioni nocive dei motori endotermici, siano essi a ciclo Otto o a ciclo Diesel. Se pensiamo a come è complesso

oggi un motore Diesel rispetto a solo 35 anni fa, ne abbiamo un'idea; infatti il motore Diesel base era puramente meccanico e non aveva impianto elettrico, il che, abbinato alle caratteristiche intrinseche di quel tipo di propulsore, lo rendeva nettamente più affidabile del motore a benzina. Oggi attorno a un motore Diesel c'è così tanta elettronica che non solo è diventato più complesso, ma anche meno affidabile: solo per migliorare la combustione si è passati dall'iniezione meccanica a quella elettronica common-rail o iniettore-pompa aggiungendo poi tutta una serie di sensori per rilevare i parametri operativi del motore (regime di giri, fase della distribuzione, istante di iniezione, temperatura combustibile e liquido di raffreddamento del motore, temperatura e volume d'aria aspirata) e attuatori per il controllo dell'iniezione, dell'aspirazione ecc. Per non tirare in ballo quelli riguardanti il contenimento delle emissioni inquinanti, che vanno dai sistemi per abbattere CO e CO₂, a quelli per la riduzione degli NO_x (EGR e catalizzatore a riduzione selettiva) e soprattutto del particolato. Simile è il discorso che riguarda il motore a benzina. Tornando sul CAN-Bus, anche noto come ISO 15765, è stato sviluppato nel 1986 ed è un bus seriale progettato per reti interne al veicolo; la comunicazione avviene su un doppino ritorto allo scopo di offrire un'alta immunità alle interferenze elettromagnetiche. In un veicolo possono essere presenti più istanze di CAN-Bus: per esempio il controllo dei finestrini e dei sedili è normalmente eseguito da un CAN a bassa velocità, mentre la gestione del motore ed il controllo dei freni richiedono un CAN ad alta velocità.

CAN è acronimo di Controlled Area Network e offre un data-rate tipico di 1 Mbps per distanze fino a 40 m; le evoluzioni subite dal CAN-Bus hanno portato a migliorie in fatto di data-rate e oggi abbiamo anche CAN FD (che opera fino a 5 Mbps) e CAN Partial Networking. Sul piano fisico, il CAN-Bus sfrutta un link seriale a due fili riferiti a massa,



quindi una linea differenziale che come livello dei segnali è un RS485 (Fig. 1); è proprio questo che conferisce immunità nei confronti dei disturbi, inevitabilmente presenti a bordo degli autoveicoli. Il CAN è un bus che viene impiegato nella comunicazione tra la ECU e le altre centraline di bordo, come ad esempio quelle dell'ESP e dell'ABS, il body computer (BSI), le centraline di governo del cambio automatico e del servosterzo elettroidraulico (GEP) o elettrico.

Il CAN-Bus è un protocollo che definisce sia lo scambio dei dati, sia il mezzo fisico (PHY) che lo permette; quest'ultimo consiste in una connessione seriale a linea bilanciata, vale a dire che i dati viaggiano su due fili riferiti a massa, ma in opposizione, quindi quando su un filo è presente il livello logico alto sull'altro si trova quello basso; l'unità che riceve i dati ha in ingresso un amplificatore differenziale (un operazionale), il quale fornisce alla propria uscita la somma algebrica dei segnali agli ingressi invertente e non invertente. Ne deriva che se un disturbo elettrico investe i fili dei dati e si presume che induca in entrambi lo stesso livello di tensione, il differenziale annulla il disturbo sovrapposto ai fili, mentre amplifica correttamente i dati. Infatti se il disturbo entra nei due input del differenziale, quello che passa dal non-invertente viene sommato all'invertente e la somma dà zero, mentre siccome i livelli logici sono opposti, all'uscita abbiamo la somma, che è un impulso di ampiezza doppia e di polarità dipendente da quale, tra gli ingressi invertente e non-invertente è positivo. Lo standard utilizzato nel CAN-Bus è tipicamente l'interfaccia RS-485, che è una seriale bilanciata in cui le unità trasmettente e ricevente sono collegate

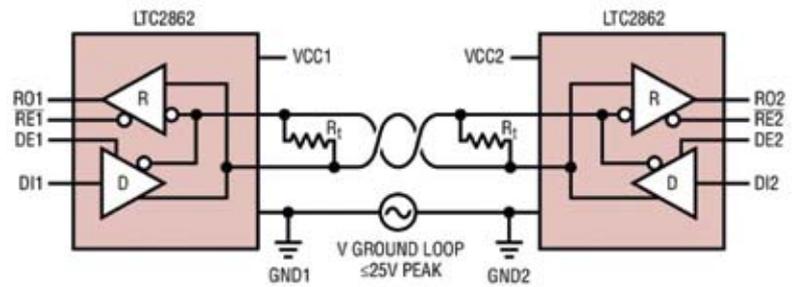


Fig. 2
Esempio di transceiver RS485.

con cavi di tipo twisted pair (doppinoritorto) per aumentare l'immunità ai disturbi. La linea RS-485 prevede dispositivi trasmettenti e riceventi e può funzionare in modo sia unidirezionale che bidirezionale; normalmente nel CAN Bus si adotta una sola linea bidirezionale e i dispositivi affacciati al doppino twistato sono transceiver, vale a dire che contengono sia un'unità trasmettente che una ricevente, attivate una alternativamente all'altra mediante segnali di controllo. La Fig. 2 mostra un esempio di linea basata su transceiver RS485. Sul piano del protocollo, ossia del formato dei dati e la loro codifica (modello ISO/OSI), lo standard descrive principalmente lo strato (layer) di scambio dati (Data Link Layer), composto dal sottostante strato (sublayer) logico Logical Link Control (LLC) e dallo strato sottostante questo che è il Media Access Control, (MAC). I protocolli di tutti gli altri layer sono lasciati alla libera scelta del progettista della rete di bordo. La Fig. 3 propone il modello ISO/OSI del CAN-Bus.

Il CAN permette di realizzare delle reti (per questo si può definirne uno stack secondo il modello ISO/OSI, che definisce l'architettura delle reti dati),

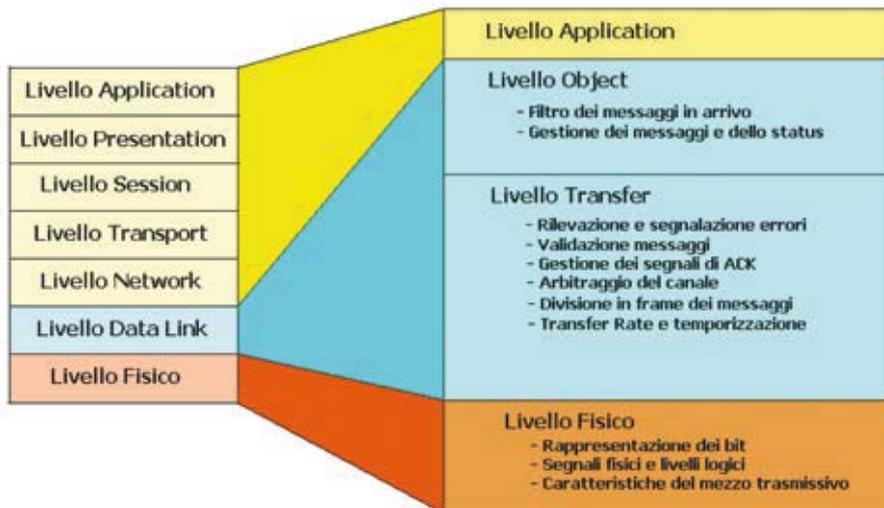
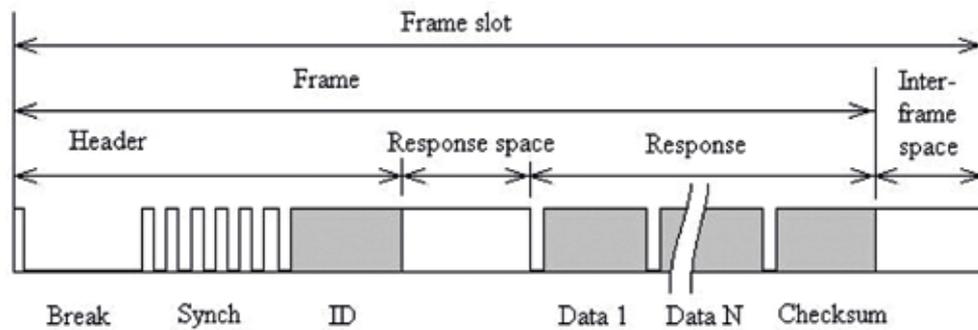


Fig. 3
Architettura CAN-Bus.

Fig. 4
 Tipico frame del protocollo LIN.



quindi con dispositivi (nodi) che comunicano tra loro e non con un master che interroga le periferiche e le coordina; questo impone che il protocollo detti alcune regole utili a evitare che in caso di trasmissione simultanea, i dati vengano a sovrapporsi e quindi vadano perduti.

La regola principale è che i dati viaggiano secondo un modello basato su bit dominanti (0 logico) e bit recessivi (1 logico); se un nodo trasmette un bit dominante e un altro un bit recessivo, il bit dominante prevale, come avviene dell'operazione logica AND vista in questo stesso capitolo. Quando vengono a trovarsi contemporaneamente sulla rete un bit recessivo e uno dominante, si verifica una collisione e solo il bit dominante va in rete (le collisioni sono invisibili). In pratica il bit dominante determina l'impulso sulla linea bilanciata e quello recessivo non produce effetto, quindi ogni volta che si presenta una differenza di potenziale tra i due fili di segnale, tutta la rete la interpreta come bit dominante.

Nel caso di un bus a linea bilanciata come l'RS485, per stabilire la priorità di comunicazione si applica lo schema CSMA/BA (Carrier Sense Multiple Access/Bitwise Arbitration): se due o più dispositivi trasmettono i loro pacchetti dati contemporanea-

mente, si applica un meccanismo di arbitrato basato sulla priorità per decidere a quale dispositivo permettere di proseguire la trasmissione. Durante la trasmissione, ogni nodo in trasmissione controlla lo stato del bus e confronta il bit ricevuto con il bit trasmesso: se riceve un bit dominante mentre viene trasmesso un bit recessivo, interrompe la trasmissione (perde l'arbitrato). L'arbitrato viene eseguito durante la trasmissione del pacchetto dei dati di identificazione del nodo; i nodi che incominciano contemporaneamente a trasmettere inviano un ID dominante a 0 binario, che incomincia con il bit alto. Non appena il loro ID è rappresentato da un numero più grande (quindi a priorità minore) i nodi stessi inviano un bit 1 (recessivo) e aspettano la risposta di uno 0 (dominante), quindi interrompono la trasmissione. Al termine dell'invio degli ID, tutti i nodi sono tornati allo stato di OFF, e il messaggio con la priorità corrente massima può liberamente transitare.

I messaggi scambiati in rete, detti "frame" iniziano con un bit di start (Start Of Frame, SOF); possono essere di quattro tipi:

- Data frame; sono i dati che il nodo trasmette;
- Remote frame; richiede la trasmissione di un determinato identificatore.
- Error frame; trasmesso da un nodo che ha rilevato un errore, per comunicarlo alla rete;
- Overload frame: introduce un ritardo fra data frame e/o remote frame.

Per quanto riguarda i Data frame, ossia i dati scambiati sul CAN, i messaggi possono avere due formati:

- Base frame format, caratterizzato da 11 bit di ID;
- Extended frame format, con 29 bit di ID.

Lo standard CAN deve riconoscere almeno il primo, mentre il secondo è opzionale; se è presente ma non gestito, deve essere comunque tollerato. Il frame base permette 2^{11} (2048) tipi di messag-

Fig. 5
 Connessione LIN di un alternatore.



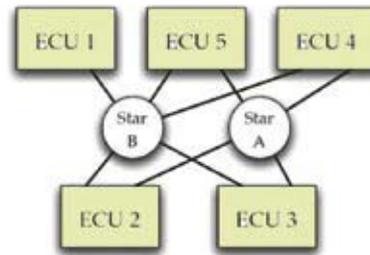
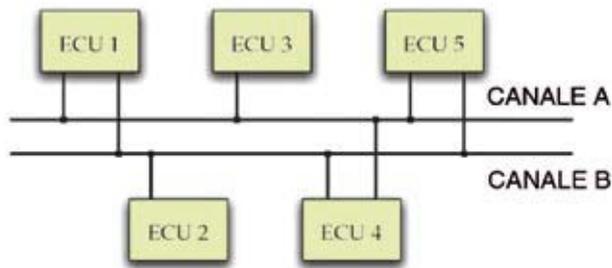


Fig. 6
La topologia di rete di un cluster FlexRay può essere di tipo a bus oppure a stella.

gio, ma da specifiche Bosch se ne possono usare solo 2031; l'extended frame format arriva a 2^{29} (536.870.912) tipi di messaggio.

La connessione fisica non è universale ma può cambiare in base al dispositivo; ha comunque sempre due contatti per i dati e una massa di riferimento. I livelli in gioco sono di $\pm 12V$.

Il CAN-Bus è normalmente presente sul connettore OBDII, che permette la diagnosi della centralina (ECU o Body Computer) principale del veicolo; le relative connessioni sono le seguenti:

- pin 6 = CAN High;
- pin 14 = CAN Low;
- pin 4 = massa impianto veicolo;
- pin 16 = positivo di batteria veicolo.

IL BUS LIN

Il Local Interconnection Network è un bus utilizzato per connessioni locali di sensori e attuatori di sottosistemi di bordo con le centraline principali, ovvero la ECU o il body computer; per esempio interfaccia la ECU con il modulo regolatore di carica dell'alternatore. Il bus è a bassa velocità di comunicazione, pensato per collegare i sensori e gli attuatori di un singolo sottosistema del veicolo in modo semplice e soprattutto economico. Alcuni esempi sono: dispositivi di controllo degli automatismi delle porte (alzavetri e chiusura centralizzata), regolazione specchietti elettrici, lettura dei sensori del motore. Quindi il controller del LIN è un dispositivo di interfaccia tra la centralina principale e i sensori e attuatori, che rappresentano le parti periferiche del sistema.

La comunicazione avviene su un solo filo rispetto a massa, come nell'One-wire; un dispositivo Master gestisce la comunicazione sul bus e deve avere una temporizzazione precisa, mentre i dispositivi Slave si sincronizzano con le stringhe di dati inviate dal master. La comunicazione è quindi asincrona e le unità slave utilizzano l'intestazione (header) dei messaggi inviati dal master per sincronizzare il proprio baud-rate; ecco perché il master deve

avere un clock preciso, mentre per gli slave è ammessa una tolleranza del 14%.

La velocità di comunicazione (data-rate o bit-rate che dir si voglia) sul bus LIN non supera i 19.200 bps e questo limita il massimo numero di dispositivi gestibili, almeno se si desidera una risposta in tempo breve, tuttavia il LIN nasce per l'interrogazione di sensori e il comando di attuatori che non richiedono tempi di reazione particolarmente stretti. Il LIN supporta anche la comunicazione a 2.400 e 9.600 bps.

Nello standard LIN ogni dispositivo connesso al bus è chiamato *nodo* e il complesso dei nodi su un bus prende il nome di *cluster*. La specifica prevede un numero massimo di 16 nodi per cluster e una lunghezza del bus non superiore a 40 metri, condizione che non viene mai raggiunta in un automezzo, anche perché di norma il sottosistema è abbastanza vicino ai sensori o attuatori correlati. Per quanto riguarda il protocollo di comunicazione, il bus LIN ha molte caratteristiche in comune con il CAN, in particolare la modalità di comunicazione che è di tipo broadcast. Non esiste una comunicazione punto-punto tra master e slave, ogni messaggio inviato è ricevuto da tutti i nodi della rete. Si elimina così il problema di conflitti e di sincronizzazione. Il bus è perciò deterministico, poiché l'invio di un messaggio dipende esclusivamente dalle

Fig. 7
Topologia di rete Flexray ibrida single-channel.

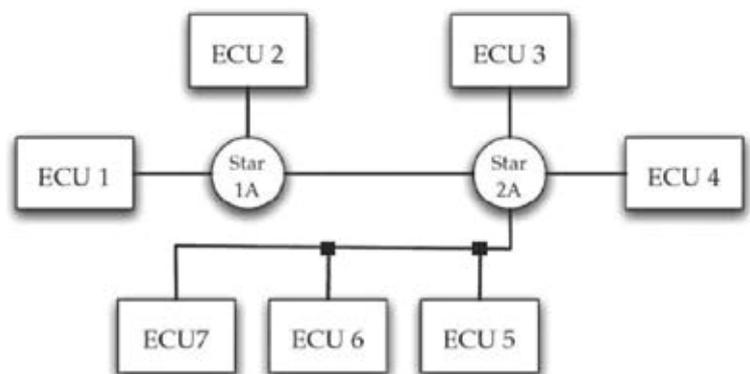
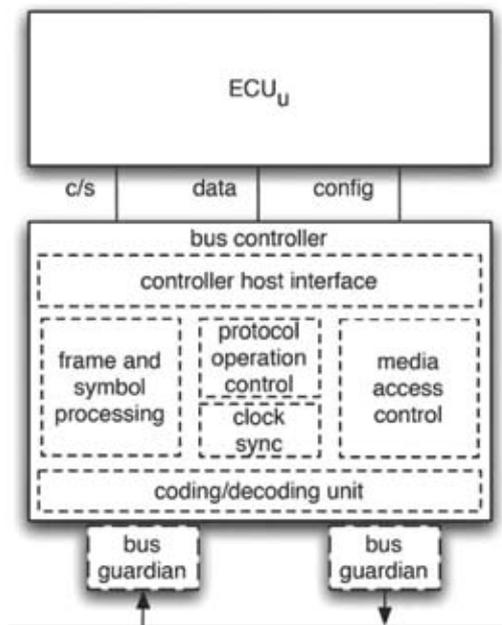


Fig. 8
Struttura completa di un nodo FlexRay: l'unità di controllo elettronico è collegata al controller del bus tramite le linee di configurazione, controllo e dati.

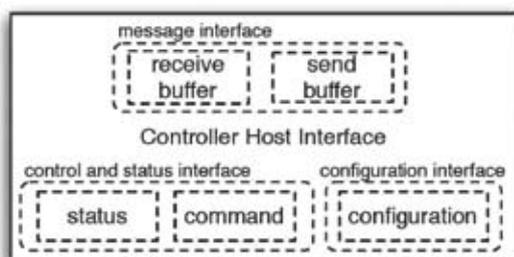


temporizzazioni del master. Anche qui il messaggio si chiama frame ed ha un proprio identificativo PID che ha la struttura di **Fig. 4**: la specifica prevede 64 PID (0x00...0x3F) di cui gli ultimi quattro riservati per comandi specifici del protocollo.

La tipica connessione del LIN è un connettore a due poli, dei quali spesso è collegato solo quello di segnale, perché come massa viene utilizzata quella dell'involucro del sensore o attuatore da leggere; almeno quando il tipo di segnale lo permette perché non soffre del cammino di corrente sulle masse dell'automobile nel punto dove si va a effettuare la connessione LIN (**Fig. 5**). Il connettore può anche avere tre poli.

Il segnale è tipicamente un'onda rettangolare unidirezionale dell'ampiezza di 12V e a vuoto, ossia senza collegamento alle unità Master, sulle Slave si rileva una frequenza dell'ordine di 4,8 kHz (assenza di dati). Il bus LIN è basato su un'interfaccia UART a 8 bit e supporta un'architettura di tipo Single Master/Multi Slave, vale a dire che un'unità master può gestire più unità slave, che nello specifico sono tipicamente sensori o attuatori.

Fig. 9
Controller host interface.



Con questa architettura, l'aggiunta o la rimozione di un nodo comporta delle conseguenze soltanto sul nodo Master: i nodi Slave non vengono influenzati in alcun modo. Un'altra caratteristica del bus LIN è che il nodo Master è in grado di sincronizzare tutti gli Slave tramite un segnale di clock incorporato, pertanto non è necessario un filo di clock o un oscillatore esterno.

Ciò che rende pratico ed economico il LIN è che a livello fisico richiede soltanto un filo alimentato a 12 V (la massa del veicolo costituisce il conduttore di ritorno) e questo semplifica i cablaggi.

RETE FLEXRAY

Questo bus di comunicazione è una vera e propria rete nata per sistemi distribuiti. È stato sviluppato dal consorzio FlexRay fondato nel 1999, da BMW, Bosch, DaimlerChrysler, Freescale Semiconductor, General Motors, NXP Semiconductors e Volkswagen, cui si sono aggiunti Fiat, National Instruments, Renault e Fujitsu Microelectronics Europe GmbH. Il FlexRay dispone di due canali di comunicazione, ciascuno in grado di fornire una velocità pari a 10 Mbps, che possono essere utilizzati insieme in modo da introdurre un meccanismo di ridondanza tale da garantire velocità e affidabilità (tolleranza verso i guasti) oppure indipendentemente l'uno dall'altro, permettendo di raggiungere una velocità aggregata pari a 20 Mbps. Il FlexRay è un protocollo di comunicazione flessibile (adattabile a molte situazioni) e fault-tolerant (ossia capace di recuperare gli errori senza che il sistema si blocchi). Un sistema FlexRay è costituito da alcune ECU, ciascuna con un controller che gestisce l'accesso a uno o due canali di comunicazione. Tale sistema è il cluster FlexRay.

Anche per il FlexRay è definito il concetto di cluster e un cluster consiste in uno (*single-channel*) o due canali (*dual-channel*) e può essere realizzato tramite un bus, una rete a stella o con una configurazione ibrida. Nella topologia dual-channel, ciascuna ECU può essere connessa sia ad un solo canale che ad entrambi.

Alcuni esempi di configurazione dei cluster FlexRay sono illustrati in **Fig. 6**.

Nella "stella" il punto di connessione centrale è detto *star coupler*. Una rete ibrida è una combinazione di una rete a bus e di una a stella, pertanto un canale è realizzato come un bus e l'altro come una stella. Un esempio di tale configurazione è mostrato in **Fig. 7**.

Possono essere aggiunti i cosiddetti Guardian Bus, che consentono di incrementare la tolleranza



ai guasti, giacché permettono di fronteggiare le principali cause di guasti che si possono presentare in una topologia a bus o a stella. I Guardian Bus sono dispositivi posizionati tra il controller del bus e il canale nel caso di configurazione a bus, mentre sono posizionati nel centro stella nella topologia a stessa; essi disconnettono la ECU dal canale quando non è consentito loro di trasmettere dati.

Nelle ECU che supportano il FlexRay, il controller del bus è collegato all'interfaccia di I/O del microcontrollore; il controller si interfaccia mediante:

- porta di controllo/stato (c/s);
- porta dati (data);
- porta di configurazione (config).

La struttura è mostrata nella **Fig. 8**. Il controller del bus è strutturato in sei componenti principali:

- interfaccia con l'host, la quale interfaccia la ECU con il bus controller;
- controllo operativo protocollo, che gestisce tutti i comandi del protocollo FlexRay e relativi stati;
- controllo di accesso al mezzo; ha il compito di programmare l'assemblaggio dei messaggi e la trasmissione;
- elaborazione simboli e frame, che gestisce i messaggi ricevuti e separa l'header dai dati effettivi;
- unità di codifica e decodifica, che esegue fisicamente l'accesso in scrittura e lettura al bus e inoltre decodifica i messaggi ricevuti e codifica quelli da trasmettere;
- sincronizzazione del clock, che ha il compito di fornire un orologio locale per l'intero sistema, che deve essere sincronizzato con tutti gli altri.

Il blocco controller host interface (CHI) fornisce l'interfaccia tra il controller del bus e la ECU e consta di (**Fig. 9**):

- interfaccia messaggi per la porta dati
- interfaccia di configurazione per la porta config
- interfaccia di controllo e stato per la porta c/s.

Nell'interfaccia messaggi ci sono due buffer: uno di trasmissione (SB) e uno di ricezione (RB). Nel primo, l'host deposita il payload dei messaggi che devono essere inviati sul bus scrivendo tramite la porta dati. Oltre a questo buffer, è utilizzato un puntatore a questa area di memoria (SBP) ossia un contatore utilizzato come indirizzo per accedere all'SB. Quando l'host scrive dalla porta dati, i dati sono memorizzati all'indirizzo a cui punta l'SBP e quest'ultimo è incrementato di un'unità.

Dal secondo buffer l'host legge i dati che sono

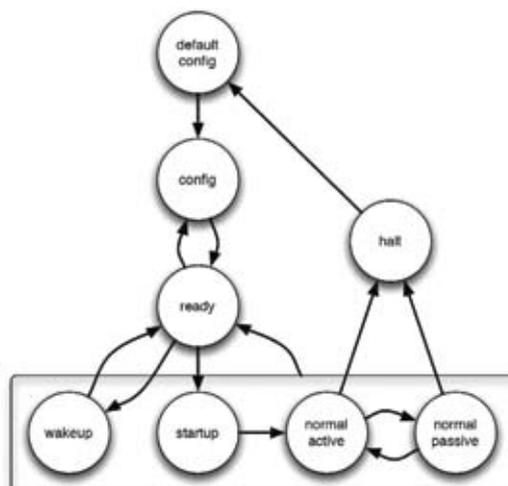
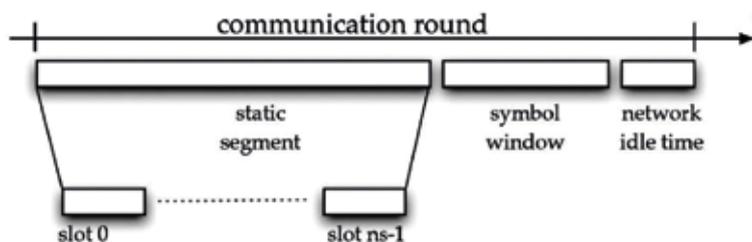


Fig. 10
La macchina a stati che descrive il funzionamento di un nodo FlexRay.

stati ricevuti; anche qui è previsto un puntatore (RBP) all'area di memoria RB. L'interfaccia di configurazione conserva alcuni parametri fondamentali scritti solo durante la fase di avvio, attraverso la porta di config. Infine, l'interfaccia c/s fornisce all'host utili informazioni circa lo stato di ricezione dei frame. Inoltre, l'host può interagire con lo stato del bus controller inviando dei comandi tramite la porta c/s.

Passiamo al Protocol Operation Control (POC), scopo del quale è rispondere ai comandi dell'host o a particolari condizioni, come stati di errore. Il POC imposta gli altri componenti nella condizione operativa appropriata. All'avvio, il POC parte dallo stato di default config, per poi passare alla configurazione del nodo nello stato config; il controller può essere configurato tramite la porta config solo in tali stati. Dopo la configurazione, il POC entra nello stato ready. A seconda dell'attività del canale di comunicazione, il controller andrà in wake-up o in startup; in quest'ultimo stato, il controller si integra nel cluster. Finché non si verifica alcun errore, esso rimane nello stato normal active, mentre in caso contrario si porta in normal passive e cerca di reintegrarsi nel cluster. Invece se si verifica un fatal error, il controller blocca qualsiasi attività del nodo, mandandolo nello stato halt. Inoltre, all'host è con-

Fig. 11
Struttura di una communication round FlexRay: la presenza di un segmento statico con un numero fisso di slot garantisce una comunicazione con tempi deterministici.



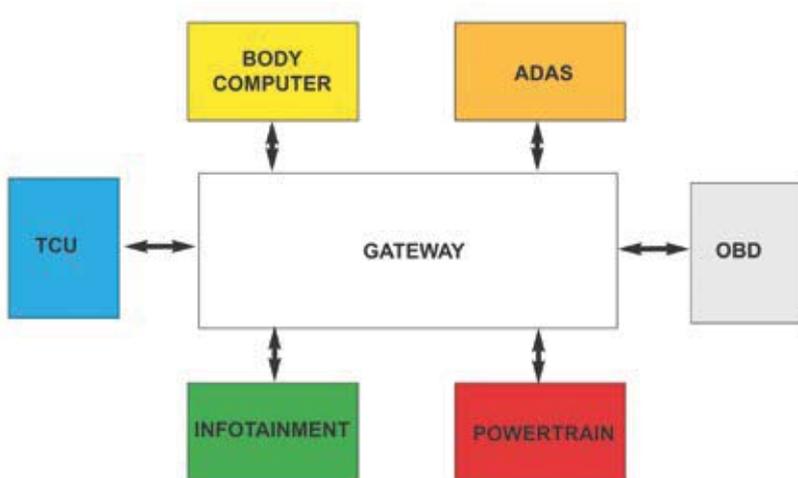


Fig. 12
Schema a blocchi
di un sistema
100BASE-T1 con
TCU.

cesso di modificare lo stato del controller. In ogni caso, il controller deve decidere se tale comando può o non può essere eseguito, in base allo stato attuale; per esempio, un comando di halt inviato dall'host è consentito solo se il nodo si trova in *normal active* o *normal passive*.

Il funzionamento del nodo FlexRay è definito dalla macchina a stati che vedete in **Fig. 10**.

Passiamo al controllo di accesso al mezzo, ossia alla rete: il media access control (MAC); questo componente gestisce la trasmissione dei dati dall'host verso il bus, programma l'accesso in scrittura al bus e aggiunge l'header al payload.

Nel protocollo FlexRay, a ciascuna ECU è concesso di trasmettere un messaggio solo all'interno di un certo intervallo temporale, chiamato slot. In ogni slot, il MAC verifica se la relativa ECU può trasmettere; in caso affermativo, il MAC importa il payload dal buffer di trasmissione (SB) e genera l'header del messaggio, poi l'unità di codifica/decodifica segnala quando il messaggio può essere scritto. Quanto al componente frame and symbol processing (FSP), gestisce la ricezione dei dati; l'unità di codifica/decodifica segnala se è ricevuto un frame valido ricevuto. Dopo la ricezione, il messaggio viene suddiviso in header e payload, che viene copiato nel buffer di ricezione (RB). Inoltre l'FSP fornisce lo stato all'host dopo ogni slot, ossia l'informazione relativa all'ora locale per la sincronizzazione e allo stato di ricezione del frame (per esempio, se è stato ricevuto un frame errato nello slot precedente). L'unità coding/decoding (CODEC) è il componente che fisicamente accede al bus. Quando il MAC segnala al CODEC di volere trasmettere un messaggio, quest'ultimo aggiunge il campo CRC per il controllo degli errori. A questo punto il messaggio

viene codificato con gli opportuni valori di tensione. Per la ricezione dei frame, il CODEC è in ascolto sul bus. Quando arriva un messaggio, innanzitutto viene verificata l'integrità dei dati, tramite controllo del CRC e se tutto risulta corretto, è notificato ad FSP l'arrivo del messaggio.

Infine, il componente di sincronizzazione del clock esegue due modi di sincronizzazione e genera un'unità per la scansione del tempo locale, detta *macrotick*. In questo modo il controller dispone di una base temporale per la schedulazione dei messaggi e degli interrupt verso l'host

Sincronizzazione del clock FlexRay

Il componente di sincronizzazione del clock prevede due modi di sincronizzazione e genera un'unità per la scansione del tempo locale, detta *macrotick*, utilizzata dal controller per schedulare i messaggi e gli interrupt verso l'host. La sincronizzazione è finalizzata a rendere uguale la lunghezza di tutti i *macrotick* nel cluster, evitando così che due nodi trasmettano nello stesso momento; l'oscillatore di clock può derivare ad esempio a causa del calore e lo standard FlexRay prevede una massima deviazione di $\pm 0,15\%$ il che implica che il clock di due nodi possa differire al massimo dello 0,3%. La sincronizzazione viene effettuata tramite messaggi di *sync* e la correzione è effettuata secondo due criteri: correzione dell'offset e del rate.

Scheduling

Il protocollo FlexRay utilizza una schedulazione dei messaggi di tipo time-triggered: diversamente da un sistema event-triggered (come ad esempio il CAN-Bus) questo garantisce tempi di trasmissione deterministici, il che è fondamentale per i sistemi critici. L'accesso al bus è programmato tramite uno schema TDMA (Time-Division Multiple Access), in cui la scrittura sul bus è consentita solo entro gli slot (a ciascuna ECU ne è assegnato un certo numero). Ogni controller conosce i propri slot di trasmissione ma non quelli degli altri controller. La schedulazione di tutti i nodi è detta *communication round* (**Fig. 11**) e contiene un segmento statico con un numero fisso di slot, una *symbol window* e un tempo di inattività della rete. La *symbol window* è utilizzata per trasmettere messaggi speciali come i simboli di wakeup. Durante il periodo di inattività della rete, il componente per la sincronizzazione del clock calcola ed esegue la correzione. Dopo l'accensione di un controller è richiesta una strategia di avvio perché il suo clock non risulta sincronizzato con gli altri. Inoltre, è richiesta una



procedura per il riavvio dopo un fatal error, poiché lo stato halt può essere lasciato solo passando per il default config. Per soddisfare tali condizioni, all'interno di un cluster FlexRay sono previste alcune ECU, dette *coldstart node*, capaci di inviare messaggi broadcast contenenti l'informazione di sincronizzazione. Dopo l'accensione, un algoritmo di elezione sceglie il nodo leader fra tutti i coldstart node, che invia in broadcast l'informazione del tempo agli altri controller, che così possono integrarsi nei cluster sincronizzando i loro clock.

ETHERNET IN AUTO

Nell'automotive ha preso piede la tendenza a utilizzare la rete Ethernet. Ideata dalla Xerox nel 1976, attualmente costituisce lo standard (IEEE 802.3 e sue varianti) più utilizzato e più semplice, tanto che viene adottato anche per realizzare reti tra computer sfruttando il protocollo di comunicazione TCP/IP (Transmission Control Protocol/Internet Protocol) cioè lo stesso di Internet, nel quale ogni elemento è identificato da un indirizzo composto da gruppi di tre cifre separati da punti (per esempio 192.168.1.2 nello standard IPv4).

Dagli iniziali 3 Mbit/s la velocità di comunicazione della ethernet è stata portata a 10 Mbit, poi a 100 Mbit/s e successivamente a 1 e 10 Gigabit. La più recente implementazione è la 100BASE-T1, specifica per l'automotive; classificata dall'IEEE come 802.3bw e nota anche come 100BASE-T1 (in passato era BroadR-Reach) è una Ethernet a 100 Mbps che consente di aumentare la quantità

effettiva di dati attraverso l'utilizzo dei principi di base della sovrapposizione e degli schemi specifici di codifica e scrambling, che permettono di ridurre le interferenze elettromagnetiche (EMI), il peso del cablaggio, i costi e le dimensioni rispetto agli standard Ethernet 10BASE-T e 100BASE-TX.

Il 100BASE-T1 è un nuovo protocollo di comunicazione a livello fisico (PHY) che consente velocità di 100 Mbps su un cavo a singolo doppino ritorto non schermato, come quello del CAN-Bus; fino al suo avvento, l'ethernet non era molto diffusa nelle auto, anche se alcuni veicoli utilizzano il 100BASE-TX per gli strumenti diagnostici di bordo (OBD). Tuttavia quest'ultimo non ha avuto successo perché richiede due cavi a doppino ritorto e non soddisfa i severi limiti per le emissioni irradiate di Classe 5 del Comité international spécial des perturbations radioélectriques (CISPR). Invece il 100BASE-T1 consente data-rate effettivi di 100 Mbps con cavi lunghi anche 15 metri. Il profilo di emissione 100BASE-T1 è conforme al metodo stripline CISPR 25 Classe 5 Allegato G e ad altri standard di emissione automobilistici come il TC8 di Open Alliance. Il 100BASE-T1 è in grado di abilitare la comunicazione di dati audio, video, connected car, firmware/software e dati di calibrazione all'interno di veicoli utilizzando la raccolta AVB (Audio Video Bridging) di protocolli Ethernet su cavo a singolo doppino ritorto non schermato. La raccolta di standard AVB presenta latenza ridotta e deterministica, nodi sincronizzati e shaping del traffico, tutti aspetti importanti per scambiare vari tipi di informazioni

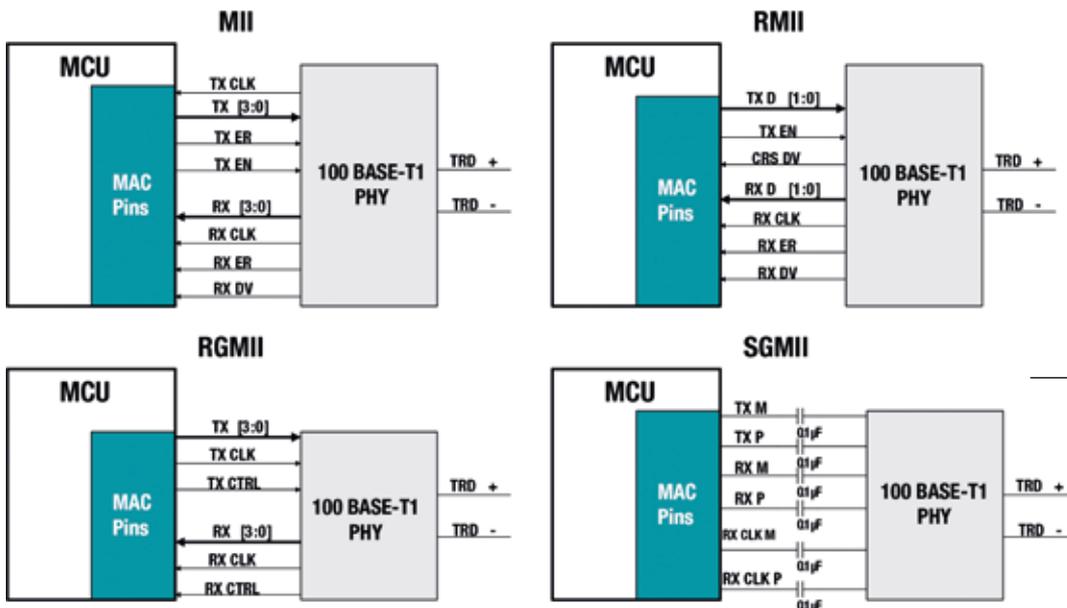


Fig. 13
Vari tipi di MII disponibili.



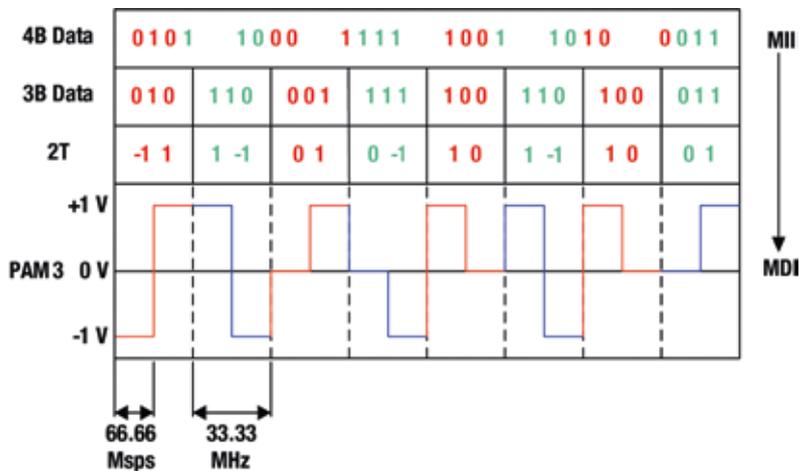


Fig. 14
Conversione di dati
GPHY da MII a MDI.

nei sistemi automotive, e consentire il trasporto di diverse tipologie di dati con varie priorità (bassa velocità di trasmissione dati e alta priorità oppure velocità elevata e bassa priorità, nonché sincronizzazione temporale). La possibilità di trasportare, oltre ai dati per la gestione elettronica del veicolo, informazioni audio e video consente di integrare nella gestione di sistema sia l'infotainment, sia l'ADAS.

La ripartizione della larghezza di banda è gestita dall'AVB; l'impostazione predefinita assegna 75 Mbps ai flussi audio e video e i restanti 25 Mbps ai flussi di dati. Tuttavia, nelle applicazioni ADAS, i dati video grezzi forniti dalle telecamere possono richiedere larghezze di banda superiori a 1 Gbps, a meno che il video non venga compresso prima della trasmissione; ciò viene ottenuto integrando a bordo delle telecamere dei CODEC in grado di comprimere la banda, spesso integrati in microcontrollori dotati di interfaccia Ethernet sia per eseguire la compressione video sia per fornire un

layer MAC (Media Access Control) per le comunicazioni Ethernet. In generale tutti i dispositivi che si affacciano alla rete 100BASE-T1 devono avere un MAC address. Il 100BASE-T1 in combinazione con l'AVB può trasmettere sia dati audio che video, aprendo quindi opportunità per l'Ethernet nel campo dell'infotainment e dell'ADAS.

La **Tabella 1** mostra la larghezza di banda necessaria per l'audio in base alla frequenza di campionamento e alla profondità espressa bit per due canali; appare come il 100BASE-T1 risponde ai requisiti di larghezza di banda per audio a 44,1 kHz, 48 kHz e persino a 96 kHz campionati con profondità in bit fino a 32 campione. Con un'ampiezza di banda di 75 Mbps di AVB abbinata al 100BASE-T1 dovrebbe essere possibile gestire una coppia di canali video nel campo dell'infotainment.

Applicazioni Connected Car

Ma il 100BASE-T1 non si limita alle applicazioni AVB: una connessione fondamentale all'interno del veicolo è l'unità di controllo telematica (TCU), che controlla la comunicazione wireless da e verso il veicolo. La comunicazione dalla TCU al gateway (che interconnette i sottosistemi dell'autoveicolo, come si vede nella **Fig. 12**) consente l'accesso al web, facilitando gli aggiornamenti software/firmware via etere per varie ECU, ma anche interagendo con le infrastrutture stradali.

La maggior parte delle auto dispone di una porta OBD 100BASE-TX per la lettura dei dati diagnostici e l'aggiornamento o il flashing del software/firmware. Collegando varie ECU in un veicolo tramite 100BASE-T1 a un gateway centrale con una porta OBD, gli aggiornamenti possono essere eseguiti più rapidamente rispetto a bus di comunicazione come CAN, CAN FD e FlexRay. Il 100BASE-T1 può inoltre facilitare la calibrazione di una ECU in prossimità del completamento della produzione.

Codifica dei dati

Il 100BASE-T1 opera con un esclusivo schema di codifica 4-bit to 3-bit (4B3B), 3-bit to 2-ternary (3B2T) e a modulazione di ampiezza degli impulsi a tre livelli (PAM3) per ottenere emissioni ridotte rispetto al Fast Ethernet. Il 100BASE-T1 prescinde dal MAC in quanto la Media Independent Interface (MII) esistente non è cambiata rispetto all'Ethernet comune. Al momento sono disponibili quattro xMII principali per 100BASE-T1:

- **MII**; è un'interfaccia dati a 4 bit (controlli di ricezione e trasmissione, clock di ricezione e trasmissione);

Bit rate (Mbps)	Sampling rate (kHz)	Profondità campionamento (bit/sample)
1.4112	44.1	16
2.1168		24
2.8224		32
1.536	48	16
2.304		24
3.072		32
3.072	96	16
4.068		24
6.144		32

Tabella 1
Larghezza di banda
audio per due canali.



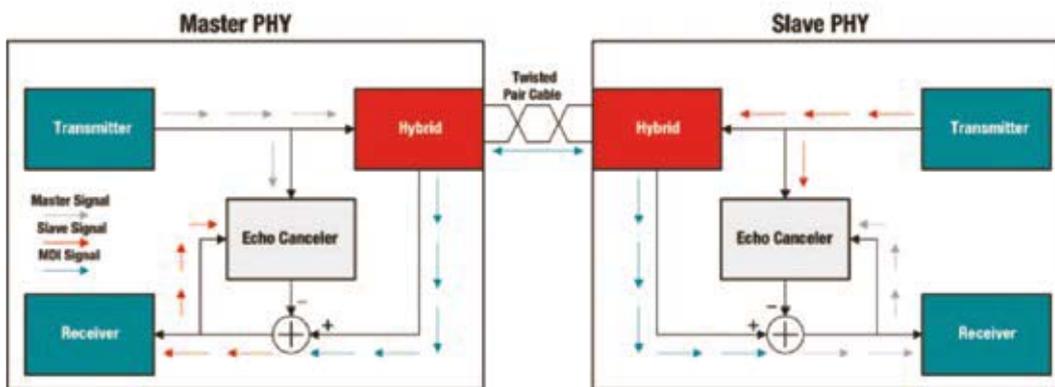


Fig. 15
Cancellazione dell'eco e schema a blocchi ibridi.

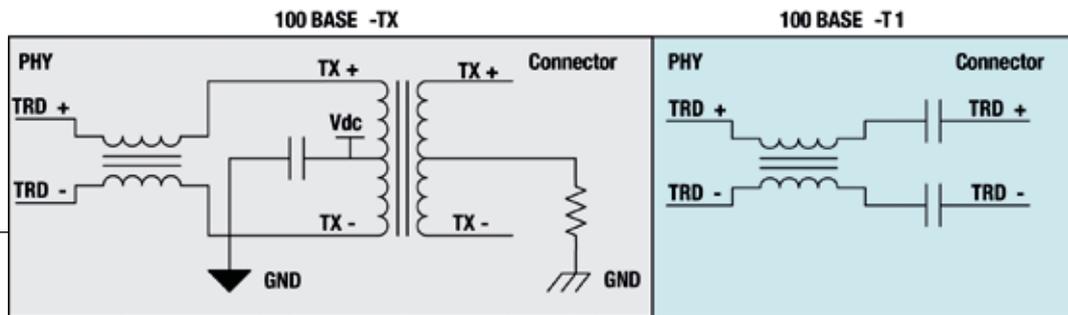
- **Reduced Media Independent Interface (RMII);** è un'interfaccia dati a 2 bit, controlli di ricezione e trasmissione, riferimento clock singolo;
- **Reduced Gigabit Media Independent Interface (RGMII);** interfaccia dati a 4 bit, con controlli di ricezione e trasmissione, clock di ricezione e trasmissione;
- **Serial Gigabit Media Independent Interface (SGMII);** prevede un percorso di ricezione a due fili Low-Voltage Differential Signaling (LVDS) nonché un percorso di trasmissione anch'esso a due fili LVDS.

Per LVDS si intende l'utilizzo di impulsi di bassa tensione ma differenziali, ossia su linea bilanciata, che permettono di elevare l'immunità ai disturbi e il data-rate raggiungibile; per il funzionamento vale quanto spiegato sull'interfaccia fisica del CAN Bus. La Fig. 13 mostra i vari tipi di comunicazione tra MAC e PHY a seconda dell'interfaccia. Dopo aver ricevuto i dati dal MAC, il PHY (transceiver) esegue la codifica, lo scrambling e la serializzazione dei dati. Questi processi preparano i dati per il front-end analogico del PHY, che li trasmette quindi sul doppino verso l'unità destinataria. Ad esempio, un PHY 100BASE-T1 che comunica con un MAC tramite RGMII riceverà quattro bit paralleli con clock a 25 MHz (100 Mbps in totale). Il PHY converte questi quattro bit in tre bit e aumenta la frequenza di clock a 33 1/3 MHz per mantenere la velocità in bit di 100 Mbps (se un frame non è divisibile per tre, il PHY aggiunge bit di riempimento per abilitare la conversione corretta; il partner di collegamento rimuove questi bit di riempimento prima di trasferirli al MAC). Usando ciascun gruppo di tre bit, il PHY genera una coppia ternaria (2T) in base alla specifica mappa dei simboli. Infine, il vettore della coppia ternaria (TA, TB) viene trasmesso utilizzando la modulazione di ampiezza

dell'impulso a tre livelli (PAM3) a una frequenza fondamentale di 66 2/3 MHz. La Fig. 14 mostra i dati convertiti da MII alla Medium Dependent Interface (MDI) attraverso il PHY: osservando il segnale PAM3 si nota come ogni periodo di 33 1/3 MHz rappresenti tre bit di dati e quindi trasferisca i dati a 100 Mbps. Questo segnale viene inviato utilizzando tre livelli di tensione (+1 V, 0 V e -1 V) con meno di 2,2 V da picco a picco (se misurati con terminazione differenziale 100 Ω). Il PAM3 viene trasmesso sull'MDI, che include il connettore per il cavo e il cavo a singolo doppino ritorto stesso, nonché tutti i componenti passivi esterni utilizzati per il filtraggio passa-basso aggiuntivo e la reiezione in modalità comune (CM). Il connettore MDI e i cavi a doppino ritorto non schermato devono soddisfare specifiche come l'attenuazione di riflessione, la perdita per conversione e la tolleranza ai guasti, definite nella Sezione 96.8 della specifica MDI dello standard 100BASE-T1.

I protocolli Ethernet hanno specifiche differenti per lo scrambling, la serializzazione e la codifica dei dati in base alla loro applicazione di destinazione. Per confronto, il 100BASE-TX trasmette i dati a 125 MHz utilizzando il Multi-Level Transmit (MLT-3). La frequenza fondamentale è superiore a quella del 100BASE-T1 (66 2/3 MHz) e richiede un doppino ritorto dedicato per la trasmissione e la ricezione. Il 100BASE-T1 richiede solo circa 33,3 MHz di larghezza di banda, quasi la metà del 100BASE-TX. In pratica, è possibile utilizzare un cavo di qualità inferiore (riduzione dei costi) offrendo al contempo un filtraggio migliore per migliorare emissioni e immunità, entrambi fondamentali nelle applicazioni automobilistiche. Inoltre, l'efficienza spettrale aumenta quando si utilizza il metodo di 100BASE-T1 di mappatura del segnale 3B2T e la modulazione PAM3 sulla codifica 4B5B di 100BASE-TX in MLT-3, diminuendo quindi le emissioni attraverso

➔ Fig. 16
Confronto fra
isolamento CC
100BASE-TX e
100BASE-T1.



la riduzione della larghezza di banda necessaria per inviare la stessa quantità di dati.

Interfaccia TX/RX a singolo doppino

Il 100BASE-T1 è un'interfaccia fisica full duplex che consente la trasmissione e la ricezione sullo stesso doppino, a differenza di 10BASE-T e 100BASE-TX, dove trasmissione e ricezione avvengono su doppietti dedicati. I supporti condivisi riducono il peso complessivo dei cavi nel veicolo, riducendo i consumi. Il full duplex fisico è realizzato attraverso i principi di sovrapposizione. I PHY 100BASE-T1 presentano ibridi integrati e utilizzano la cancellazione dell'eco per rimuovere il proprio segnale trasmesso ed estrarre le informazioni ricevute da un partner di collegamento; allo scopo, un PHY funge da master e l'altro da slave. Quando due PHY 100BASE-T1 si connettono tra loro avviano un processo di addestramento che porta sia il dispositivo in prova (DUT, Device Under Test) che il partner di collegamento a trasmettere informazioni alla stessa frequenza e con la stessa fase. La Fig. 15 schematizza la cancellazione ibrida e dell'eco all'interno di ciascun PHY.

Prima che la segnalazione PAM3 esca dalla scheda o vi entri, vengono implementati dei condizionamenti mirati a isolare l'MDI per evitare loop di massa e offset CC del driver, ridurre il rumore CM (modo comune) e ridurre le emissioni irradiate,

mantenendo un'elevata immunità.

Per quanto riguarda il disaccoppiamento in CC, i PHY 100BASE-TX utilizzano in genere trasformatori con presa centrale (sul lato PHY) collegata a una tensione CC dipendente dal PHY. I trasformatori utilizzano anche la terminazione Bob Smith (presa centrale, sul lato del connettore, collegata a massa attraverso un resistore) per contribuire a migliorare il filtraggio del rumore CM.

Il 100BASE-T1 ha un approccio più semplice, basato sull'utilizzo di soli due condensatori che forniscono l'isolamento CC e riducono le dimensioni della soluzione rispetto a un'applicazione con un trasformatore.

La Fig. 16 mostra un'implementazione tipica del circuito 100BASE-T1. Invece nella Fig. 17 è possibile vedere la schematizzazione del tipico sistema 100BASE-T1 al completo, partendo dal microcontrollore e arrivando al doppino ritorto che consente la comunicazione con i sottosistemi di bordo.

CONCLUSIONI

Bene, con questo abbiamo terminato la nostra carrellata sui bus di comunicazione utilizzati nelle automobili moderne e che certamente equipaggeranno anche le auto del futuro, a prescindere da quale ne sarà la motorizzazione. □

➔ Fig. 17
Implementazione tipica
della 100BASE-T1.

