

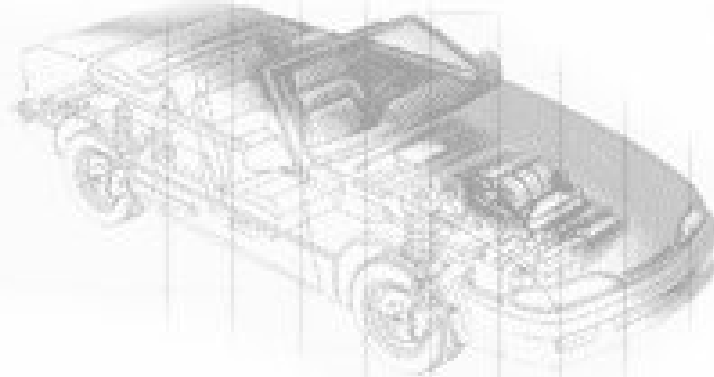


**Corso di RETI e SISTEMI per  
L'AUTOMAZIONE**

**Prof. Stefano PANZIERI**

**A.A. 2000/2001**

**LIN**



**Local Interconnect Network**

**a cura di**

**Giovanni RENDINA e Francesco RICCIARDI**

# Perché LIN?

La presenza di un numero sempre più consistente di black box e di cavi è contrastata non tanto dai costi elevati quanto dall'impossibilità di trovare una sistemazione all'interno del veicolo.

Un'architettura che utilizzi una tecnologia **multiplex** semplificherebbe decisamente il sistema di cablaggio e renderebbe così l'intero sistema **flessibile** ed **espandibile** a nuove soluzioni.

# Obiettivi di LIN

- a. standardizzazione del prodotto
- b. facilità d'uso
- c. reperibilità dei componenti
- d. basso costo (cfr. **CAN** o **J1850**)

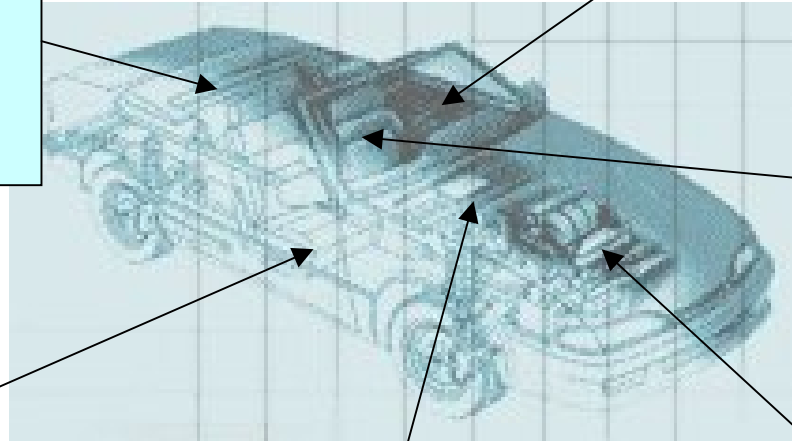
# Vantaggi di LIN

- a. affidabilità in campo automobilistico (gestione della diagnostica)
- b. elevata funzionalità ad un prezzo competitivo
- c. predisposizione a nuove tecnologie
- d. flessibilità

# Applicazioni di LIN

## COPERTURA

sensore di pioggia  
 sensore di luminosità  
 controllo luminosità  
 ...



## VOLANTE

controllo climatizzatore  
 indicatori di direzione  
 hi-fi  
 telefono...

## SEDILE

motori di posizionamento  
 pannello di controllo  
 ...

## PORTIERA

specchio retrovisore  
 chiusura portiera  
 blocco portiera  
 alzacristalli elettrico...

## CLIMATIZZATORE

pannello di controllo  
 ...

## MOTORE

sensori vari  
 ...

# Caratteristiche tecniche del sistema di comunicazione LIN

- a. implementazione a basso costo con cavo unico (enhanced ISO 9141)
- b. velocità fino a 20 Kbit/s
- c. tecnologia master/slave (il master è unico e non si richiede la gestione dell'arbitraggio)
- d. implementazione basata su un'unica interfaccia hardware (silicio a basso costo)
- e. autosincronizzazione (consistente riduzione dei costi del dispositivo hardware)
- f. tempi di latenza garantiti

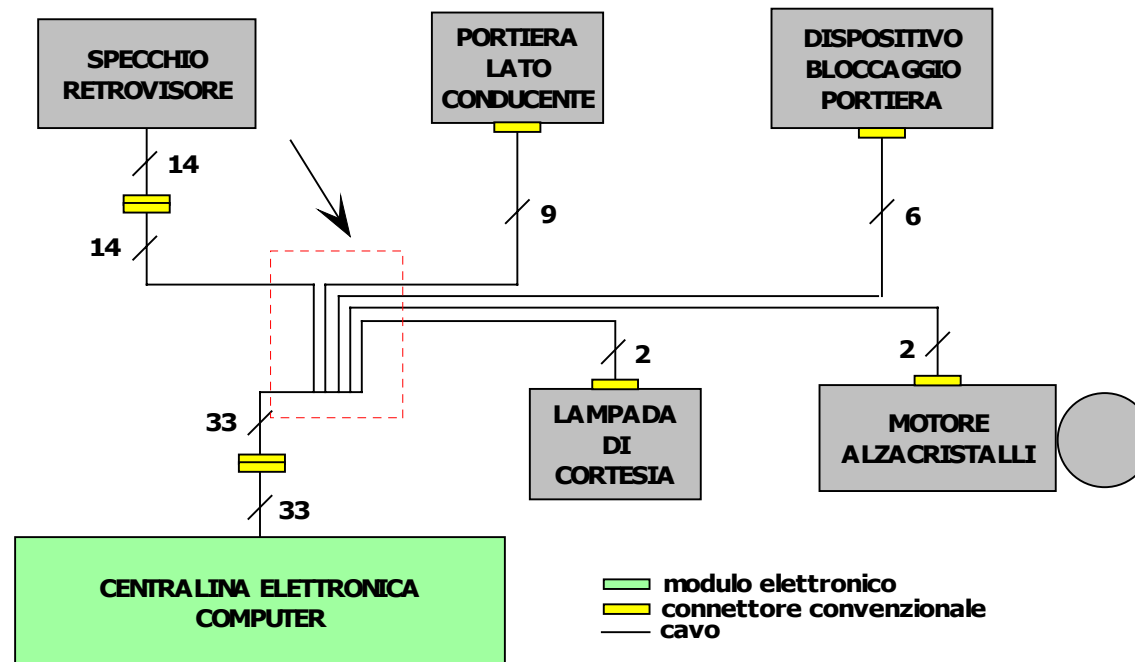
---

# Dispositivi elettronici



Le **ECU** (Electronic Control Units), oltre ad essere sempre meno ingombranti, sono applicabili direttamente ad attuatori o sensori.

# Un'architettura convenzionale

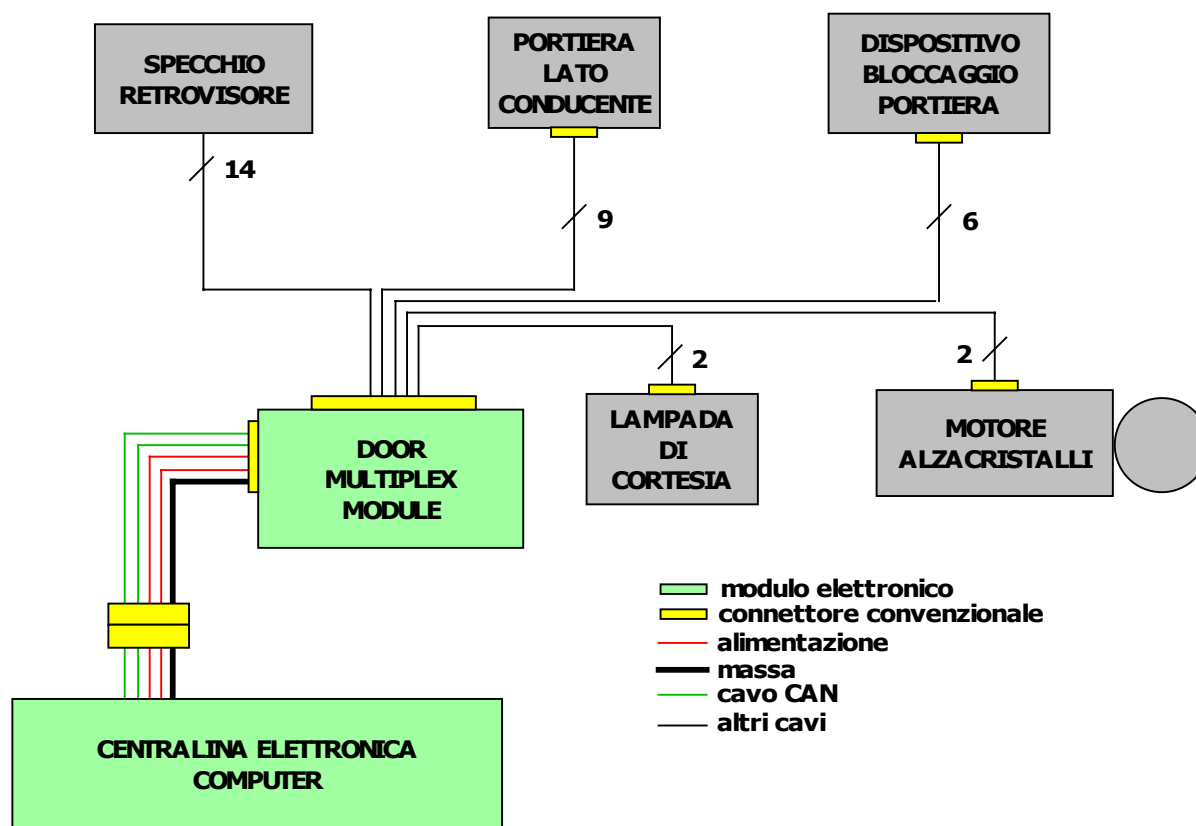




In questo caso la tecnologia **multiplex** è completamente assente.

Questa architettura, prevedendo la presenza di un esiguo numero di dispositivi elettronici e di una grande quantità di cavi, non si rivela affatto flessibile ad eventuali cambiamenti.

# Un'architettura multiplex centralizzata



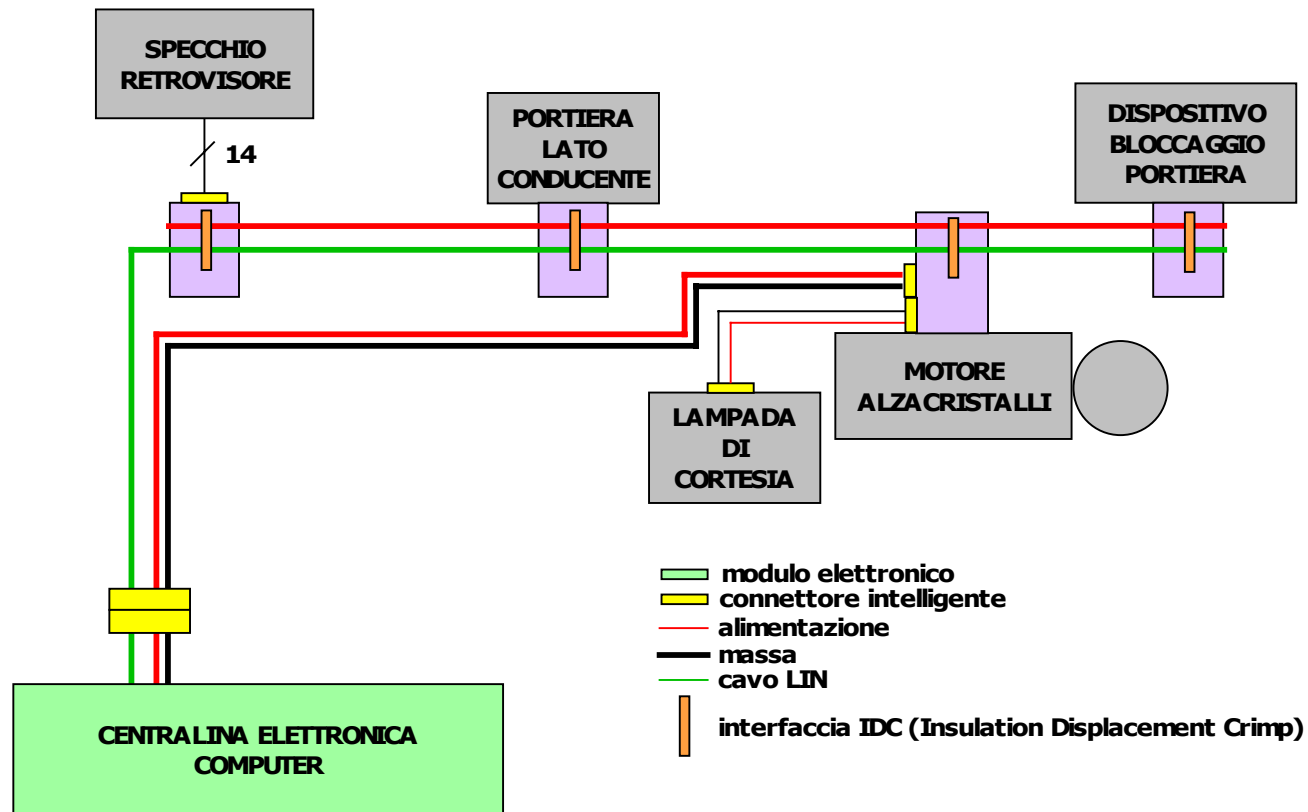
In questo caso sono presenti gli stessi carichi controllati, però, attraverso un modulo **multiplex centralizzato**.

Questa architettura solitamente utilizza un canale CAN per il trasferimento dati.

Con questo approccio si riduce il numero di cavi anche se la flessibilità ad eventuali cambiamenti rimane decisamente limitata.

Una modifica apportata ad un carico qualsiasi potrebbe richiedere interventi sull'intero modulo multiplex centralizzato.

# Un'architettura multiplex distribuita

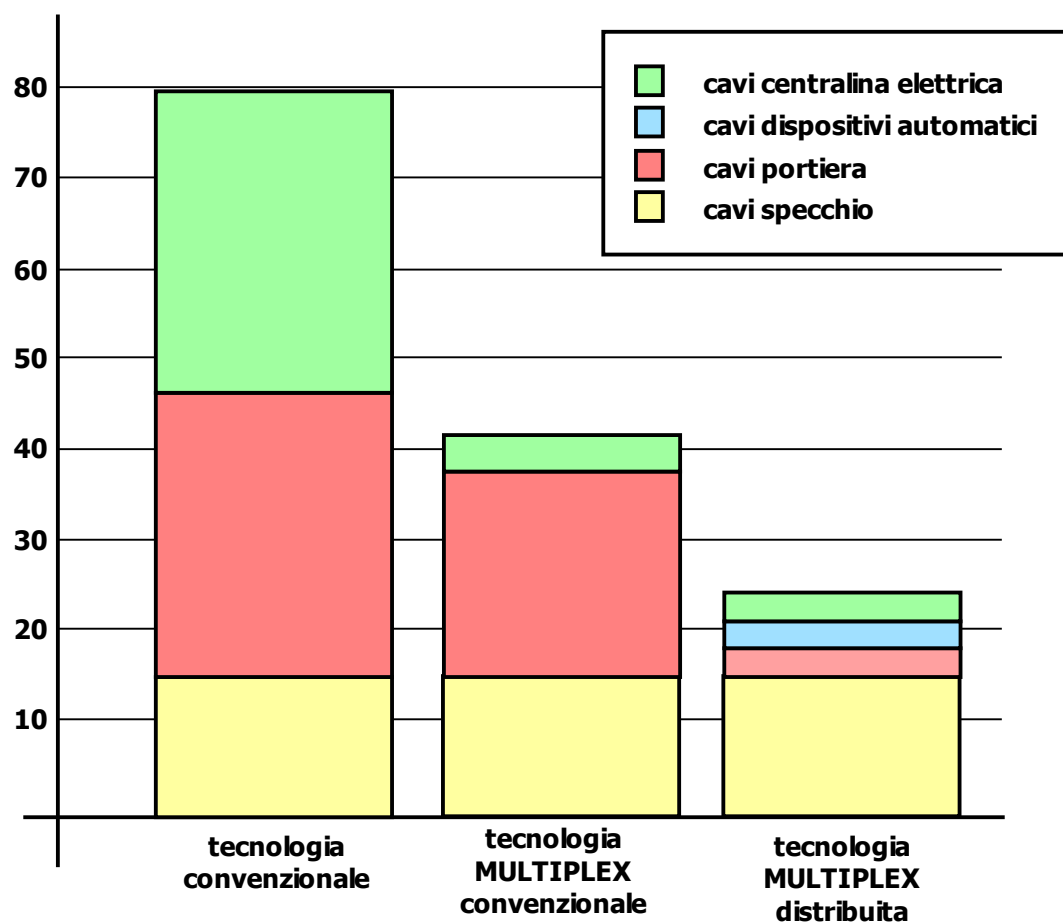


In questo caso gli stessi carichi sono gestiti attraverso un'architettura **multiplex distribuita** che utilizza **connettori intelligenti**.

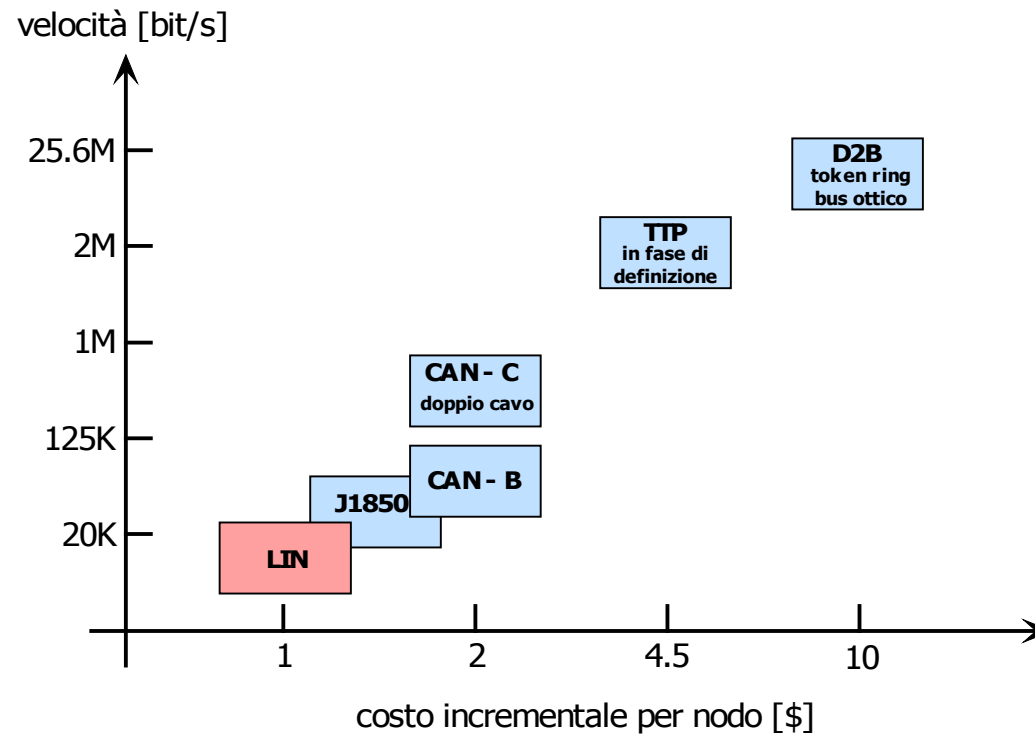
Le funzioni del precedente modulo multiplex centralizzato sono distribuite sui quattro connettori intelligenti per mezzo del protocollo LIN.

Questo approccio prevede la presenza di un elevato numero di dispositivi elettronici supportati da un esiguo impianto di cablaggio (si veda il grafico successivo).

# Confronto tra architetture

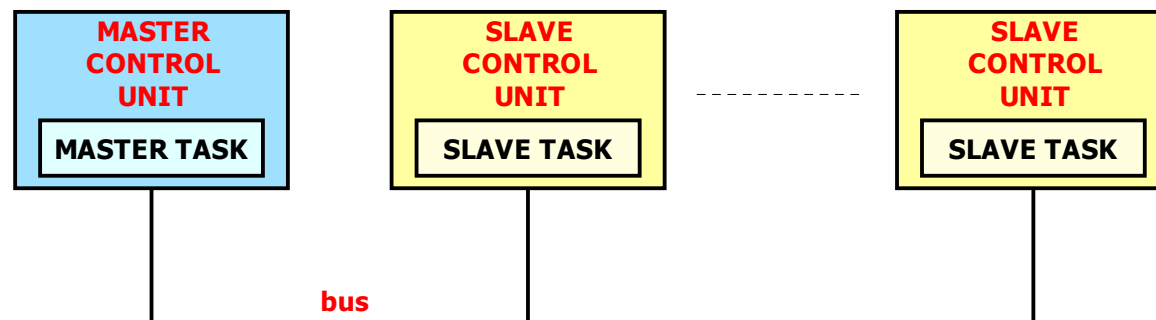


# Confronto tra ABS (Automotive Bus Systems)



# il MASTER (1)

- ha il controllo completo tanto del bus su cui vengono immessi i dati quanto del protocollo di comunicazione
- determina il momento in cui avviare il trasferimento di messaggi sul canale





---

## il MASTER (2)

- invia il **SYNC BREAK** (sequenza predefinita di byte che precede un frame)
- invia il **SYNC BYTE** (per sincronizzarsi con gli slave)
- invia l'**ID-field** (campo di 6 bit contenente informazioni sul mittente, i destinatari, la tipologia di messaggio e la lunghezza del campo dati)
- riceve i bit di **WAKEUP BREAK** dagli slave quando questi richiedono di comunicare ed il canale è inattivo
- effettua il monitoraggio dei **DATA-BYTE** e del **CHECK-BYTE**, verificandone rispettivamente la coerenza e la correttezza.

## Io SLAVE

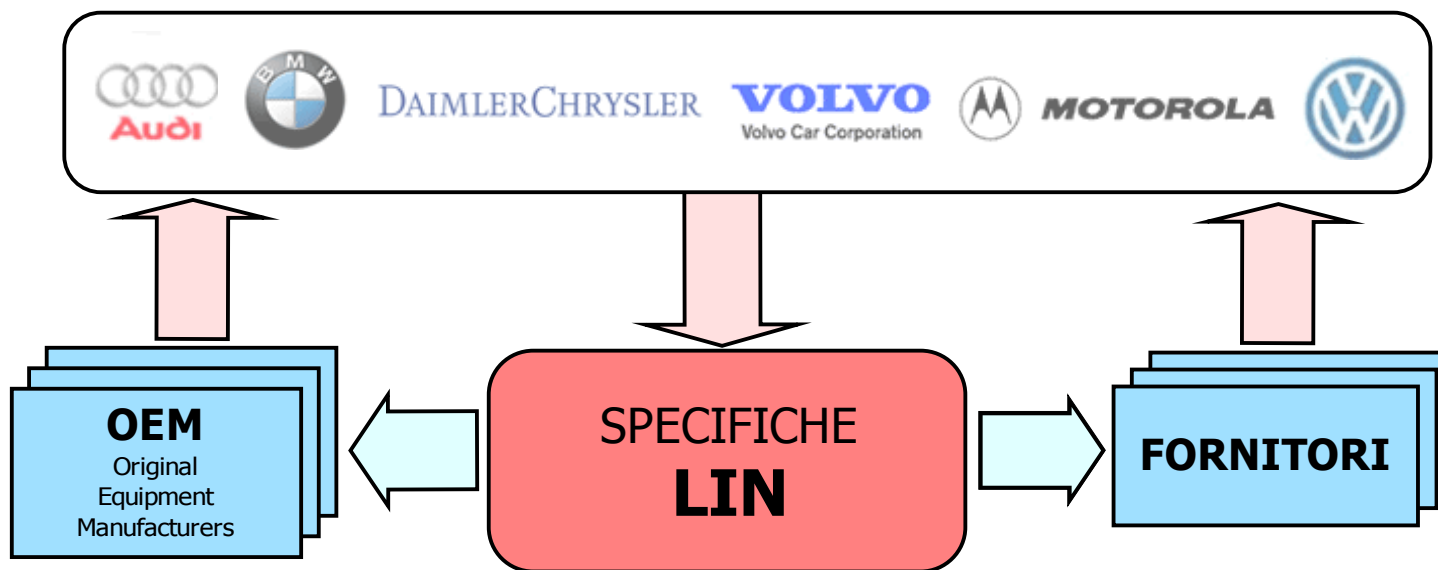
- è uno dei nodi (da 2 a 16) presenti sul canale che può ricevere o trasmettere dati quando il master lo ritiene opportuno
- attende il **SYNC BREAK** da parte del master
- si sincronizza sul **SYNC BYTE** emesso dal master
- analizza i bit l'**ID-field**
- in base al contenuto dell'**ID-field** determina cosa fare

**N.B.** Un nodo può fungere contemporaneamente sia da master che da slave!

# La strategia di mercato

1. Proporre una soluzione a **basso costo**
2. Minimizzare le **dimensioni** dei dispositivi ed il livello di **resistenza termica**
3. Dimostrare l'**efficienza** del prodotto effettuando test tecnici sulla molteplicità dei dati presenti sul canale

# LIN ed il mercato



# Bibliografia

1. **SAE Technical Paper Series “LIN Bus and its Potential for use in Distributed Multiplex Applications”** (2001-01-0072)
2. **Introduction to LIN** by Motorola Group (Hans Christian von der Wense, Munich)
3. **LIN Consortium** [http:// www.lin-subbus.org](http://www.lin-subbus.org)
4. **TTProtocol** [http:// www.ttpforum.org](http://www.ttpforum.org)
5. **J1850 Protocol**  
[http://developer.intel.com/design/intarch/papers/j1850\\_wp.htm](http://developer.intel.com/design/intarch/papers/j1850_wp.htm)