

# FLEXRAY

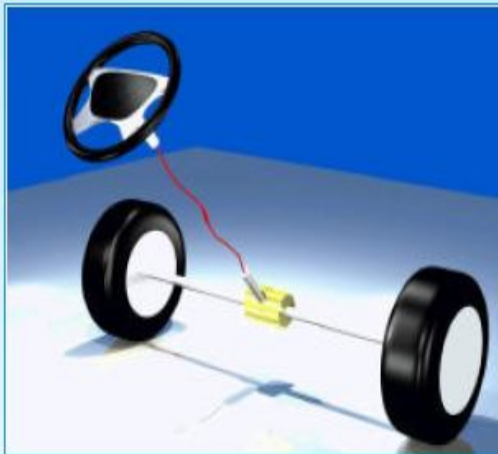
## Introducere:

Sistemul electronic din autoturisme a luat o mare amploare începând cu anul 2000. Se urmărește înlocuirea pe cât posibil a părților mecanice din mecanismul de frânare, din sistemul de direcție, deoarece acestea sunt grele, costisitoare, lichidul de frână fiind toxic în cazul în care ajunge în natura. Avantajele înlocuirii părților mecanice sau hidraulice cu cele electronice sunt asigurarea unui control mai eficient și mai rapid a frânării, de exemplu, se obține în acest fel și obținerea unui gabarit mai mic a autoturismului, și nu în ultimul rând aceste sisteme de control electronic permite implementarea așa numitului "fault-tolerance" adică toleranța la defecțiune, asigurând în acest fel și o mult mai mare siguranță.

Aceasta metodă de acționare a unor blocuri din mașina poartă numele de tehnologie x-by-wire, cu specificația ca acel X poate fi înlocuit cu : steer (direcție), break(frână).

Acest sistem s-a utilizat cu succes în realizarea avioanelor (fly-by-wire), unde greutatea este un lucru demn de luat în considerare, iar transferul de date este destul de ridicat așadar se cerea utilizarea unui nou protocol de comunicație mult mai complex.

În cazul realizării unei tehnologii de genul *x-by-wire* trebuie acordată o mare atenție software-ului care se implementează, și care comandă actuatorii și de asemenea alimentarea părții electronice care asigură comanda acestor actuatori trebuie să fie una separată față de cea de la baterie.



Până în prezent sunt dezvoltate următoarele tipuri de protocoale utilizând tehnologia x-by-wire :

- TTP – Time Triggered Protocol.
- FlexRay
- TTCAN – Time Triggered CAN

Aceste sisteme pe lângă faptul că asigură un control a unor părți a autoturismului sunt capabile să și ajute soferul în anumite situații critice.

Așadar comanda roților cu ajutorul volanului se poate realiza folosind numai fire, dar se cere un nivel de siguranță ridicat, acest nivel nu este îndeplinit de protocoalele care erau înainte : CAN, LIN. În acest fel se poate utiliza în loc de volan un joystick care poate fi folosit atât în ambele părți ale mașinii (stânga sau dreapta), opțiunile putând fi modificate prin software.

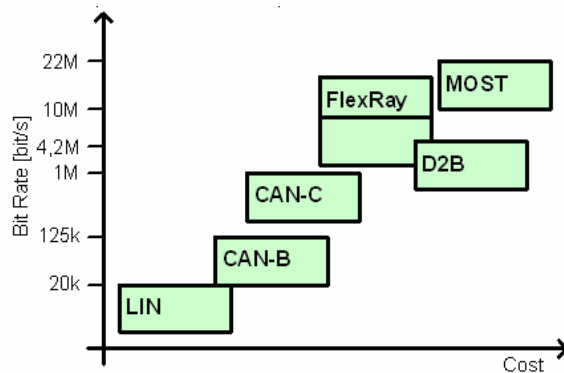
Cele mai des utilizate protocoale sunt cele TTP și FlexRay.

TTP a apărut în urma realizării unui comitet format din : PSA Peugeot Citroen, Audi, Volkswagen, Honeywell și Delphi Automotive Systems, protocol ce a luat ființă la un institut din Vienna, Austria, și se bazează pe transmisia digitală TDMA (Time Division Multiple Access).

FlexRay a luat ființă în cadrul unui concern format din : General Motors, DaimlerChrysler, BMW, Motorola, Philips Semiconductors și Bosch Automotive Group. TTCAN a fost dezvoltat de BOSCH, cei care în 1985 pun bazele protocolului CAN.

O mare deosebire între CAN și protocoalele amintite mai sus este că, CAN-ul utilizează o așa numită declanșare pe eveniment pe când celelalte protocoale folosesc o declanșare în timp. De exemplu, în cazul CAN-ului dacă un senzor sesizează că ceva nu e în regulă va trimite un mesaj de eroare pe magistrală, dar în cazul în care pot apărea mai multe defecțiuni și mai mulți senzori trimit în același moment informații magistrala se poate bloca.

Comparație între diferitele protocoale de comunicații utilizate în domeniul automotive:

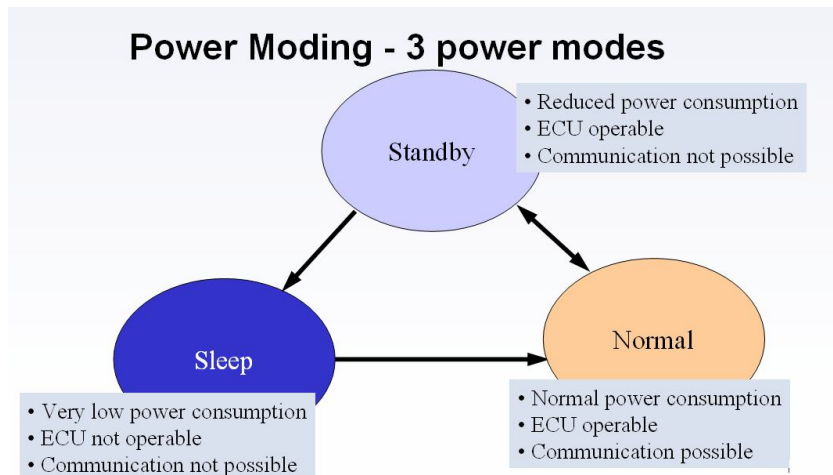


Prima versiune a FlexRay-ului apare în 2000, iar o variantă îmbunătățită a luat ființă în 2005. În momentul de față în proiectul Flexray sunt implicate 128 de firme membre. Se preconizează ca acest protocol să apară în 2006 la autoturismul BMW X5.



#### Caracteristici generale:

- 2x10 Mbit/sec debit de informație, în cazul în care se utilizează 2 canale;
- 14 – 42 V;
- Configurații multiple (simplă, stea, hibridă);
- Transfer de date sincron și asincron;
- Flexibil;
- Transmisie pe 2 fire torsadate sau pe fibră optică;
- Toleranța la eventualele defecțiuni ce pot apărea;
- Comunicația se poate face redundant sau neredundant;
- Opțiunea de Sleep, Stand-by.



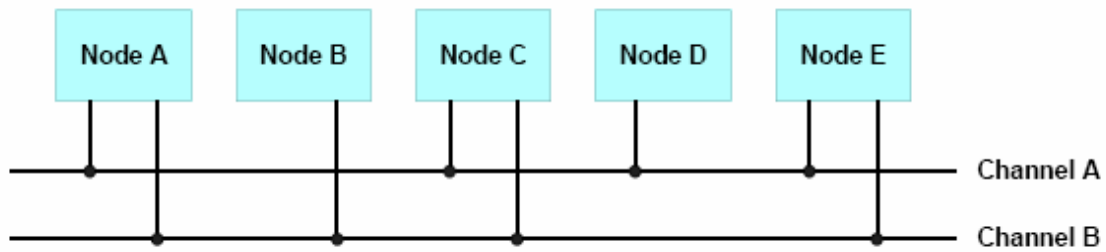
Timpul alocat transmiterii unui mesaj este împărțit în 2 părți: static și dinamic. În segmentul static, numai mesaje programate a fi transmise sunt stocate. Mesajele cruciale care pot să apară în orice moment, cel mai probabil în cazul unei defecțiuni, sunt stocate în segmentul dinamic. Acesta conține așadar mesaje de tip diagnostică, și spre deosebire de segmentul static acesta este limitat ca bandă de frecvență.

Deși Flexray este un protocol de tip *broadcast*, informațiile transmise pe magistrală nu sunt importante și nu sunt ascultate de fiecare nod. Pachetele care se transmit pe magistrală au un identificator, iar în funcție de acesta un anumit mesaj este sau nu recepționat de către un nod.

### Topologia FLEXRAY

Arhitectura se bazează pe utilizarea a cel mult 2 canale denumite Canalul A și Canalul B .

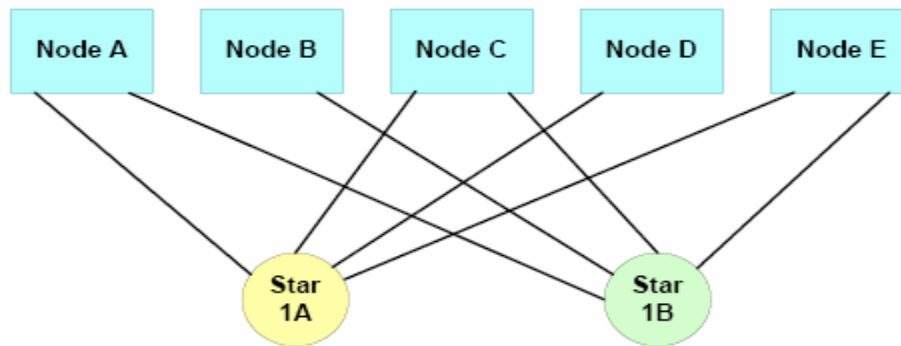
### Topologia pasivă:



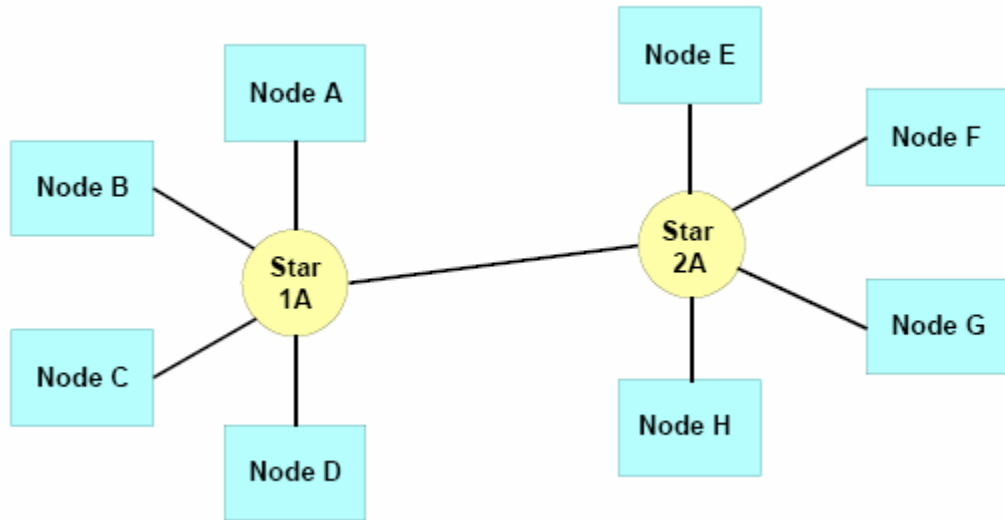
Fiecare nod poate fi conectat atât la canalul A cât și la canalul B , sau doar la unul singur. Numărul de noduri conectate la o magistrală poate să fie între 2 și 64.

### Topologia activă:

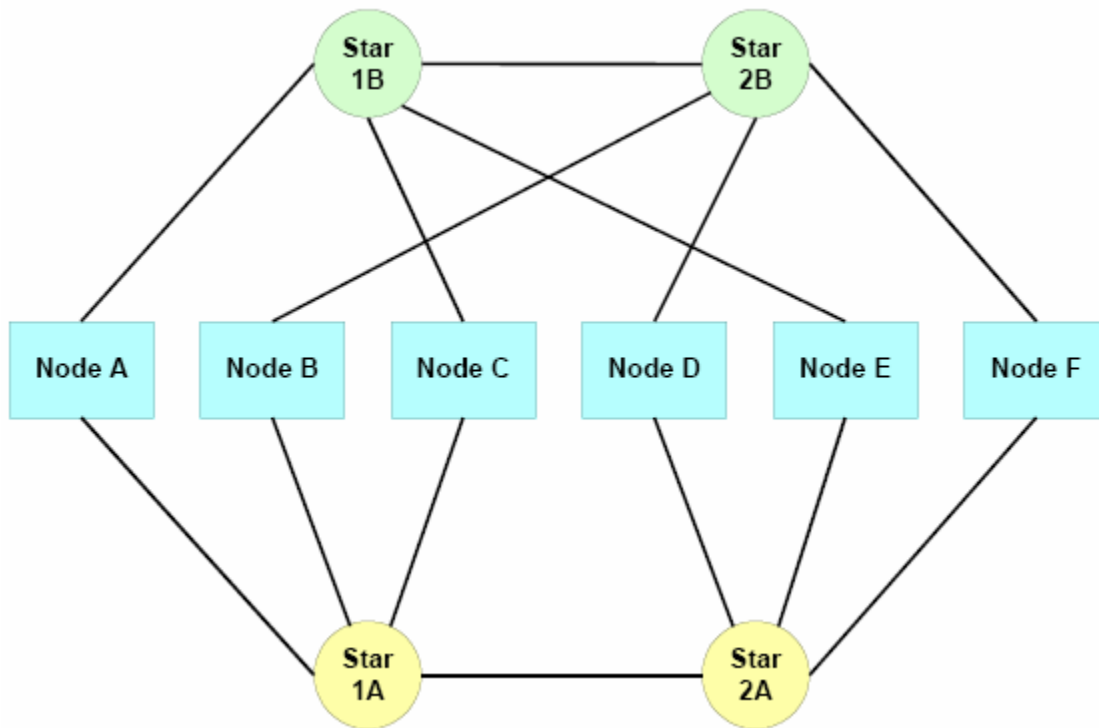
Acest tip de topologie permite conectarea mai multor noduri în formă de conexiune stea.



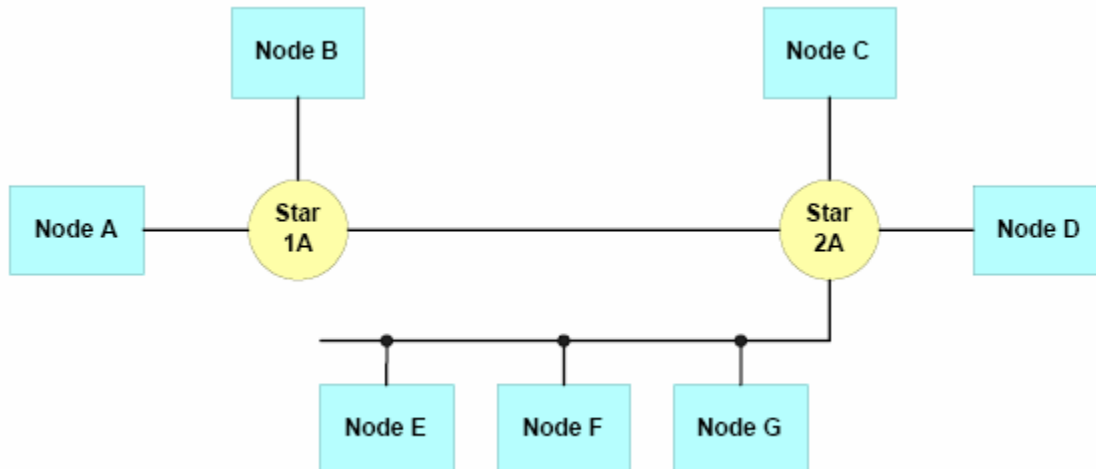
*Topologie stea cu două canale*



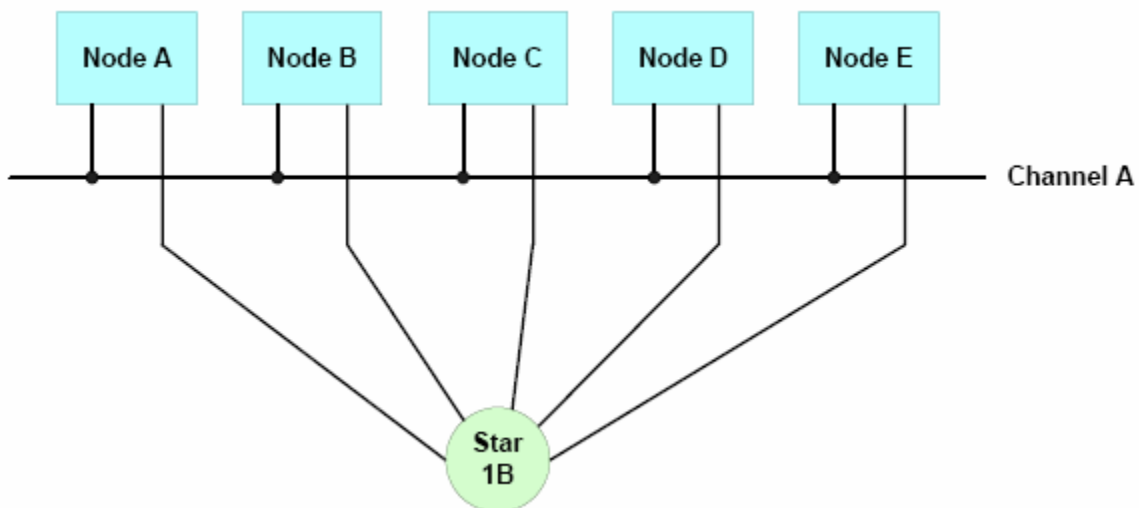
*Topologie stea cu un singur canal*



*Topologie stea completă*



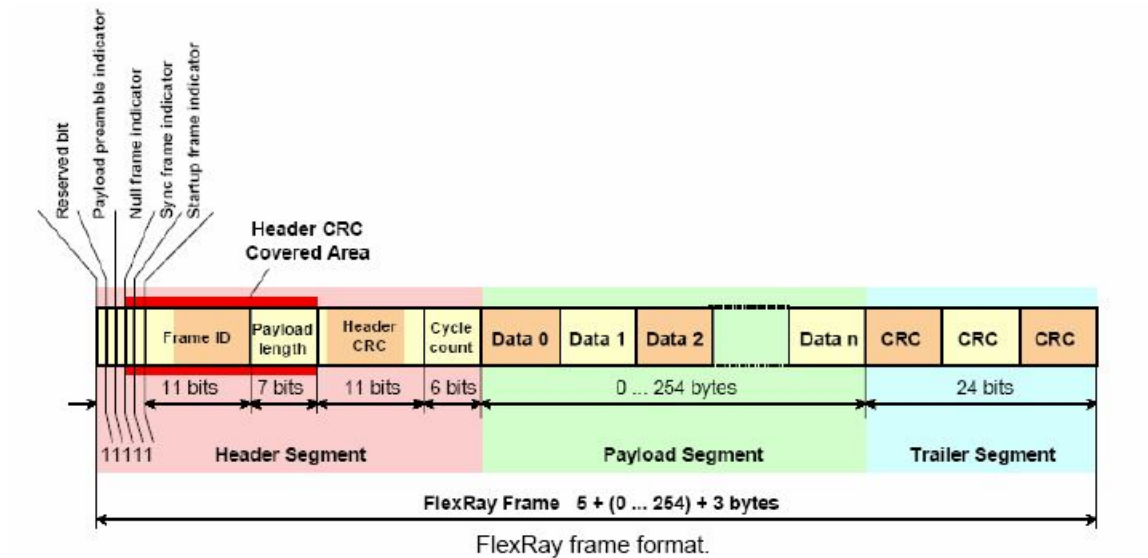
*Topologie hibridă cu un singur canal*



*Topologie hibridă cu 2 canale*

Fiecare nod are identificatorul lui unic. Nodurile conțin un master, un controler pentru comunicație, până la 2 „gardieni” de magistrală și un driver. Nodurile pot fi configurate să transmită pe unul sau pe două canale.

## Formatul de date



După cum se observă formatul de date este împărțit în 3 :

- antetul segmentului (Header Segment)
- datele propriu zise (Payload Segment)
- codul de eroare (Trailer Segment).

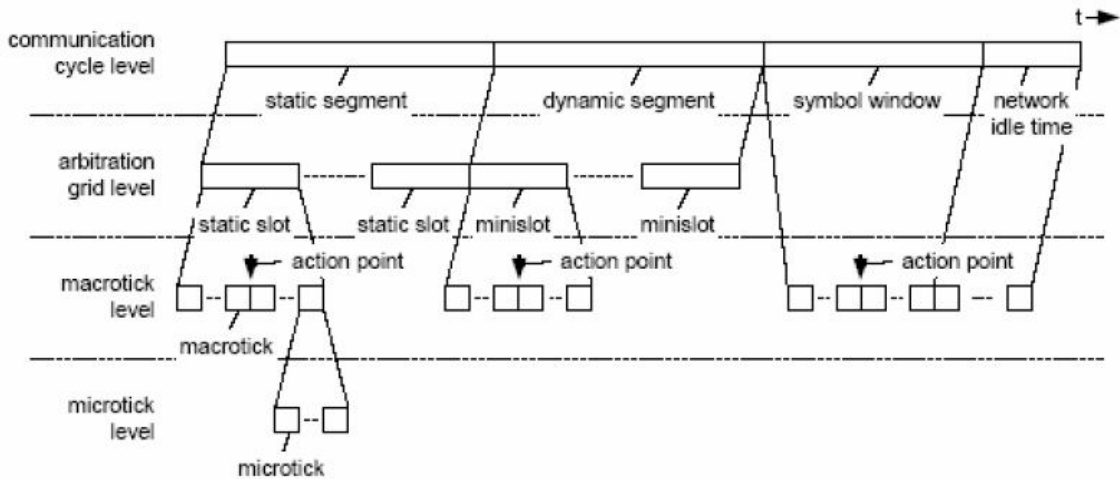
Nodul transmite formatul de date pe magistrală astfel încât antetul este citit prima dată, urmat de datele propriu zise, iar ultima parte se transmite codul de eroare.

Antetul segmentului conține 5 bytes, și anume:

- Bitul rezervat (Reserved bit)- acesta este momentan nealocat, fiind păstrat pentru versiunile viitoare.
- Indicatorul preambulului de date (The payload preamble indicator)- indică dacă segmentul de date conține sau nu un vector opțional
- Indicatorul de nul (The null frame indicator) – arată dacă formatul de date este nul sau nu
- Indicatorul de sincronizare (Sync frame indicator)- indică faptul că acel frame este sau nu un format de date de sincronizare
- Indicatorul de start (Startup frame indicator)- indică faptul că acel frame este unul de start. Aceste tipuri de frame-uri ocupă un loc important în procesul de începere a comunicației.
- ID-ul frame-ului- definește slotul în care va fi transmisă informația. Acest ID este folosit o singură dată de fiecare canal într-un ciclu de comunicație.
- Lungimea încărcăturii (Payload length) – indică jumătate din numărul de bytes de date ce va fi transmis.
- Antetul CRC (Header CRC) – conține un cod ciclic redundat care se compune din indicatorul de sincronizare, indicatorul de start, frame ID, lungimea încărcăturii.
- Numărătorul de cicluri (Cycle count)- definește starea nodurilor la un moment dat.

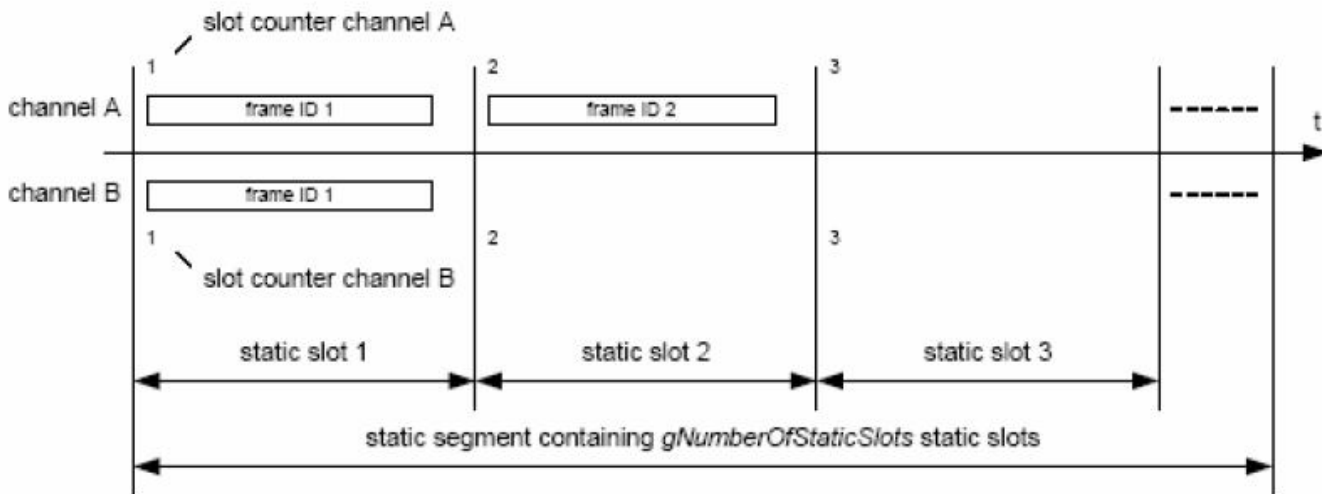
Datele propriu zise (payload segment) conțin 255 bytes de date. Pentru datele transmise în segmentul dinamic primii 2 bytes pot fi utilizați opțional ca și câmp identificator a mesajului. În cazul segmentului static primii 13 bytes pot fi folosiți opțional ca și vector de administrare a rețelei. Codul de eroare conține 24 de biti CRC.

Ciclul de comunicație



Ciclul de comunicație cuprinde segmentul static, dinamic, fereastra de simboluri și timpul în care rețeaua este inactivă (network idle time).

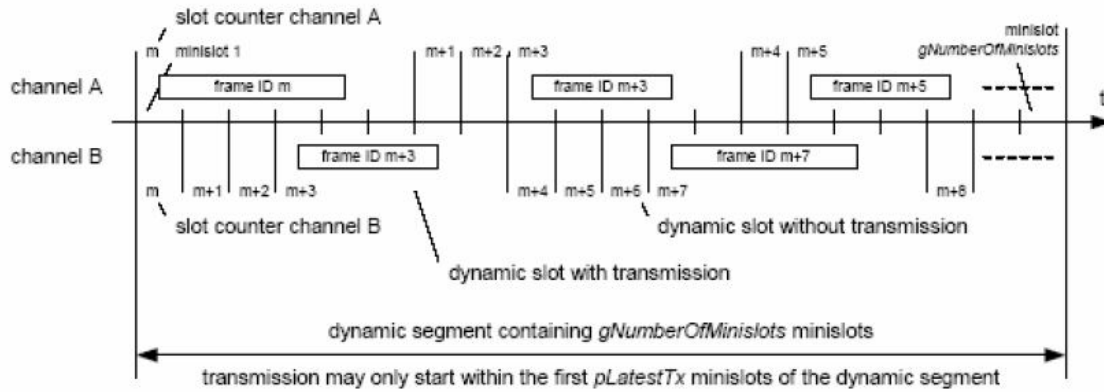
*Segmentul static* este împărțit în felii egale de timp. Figura de mai jos prezintă stările posibile în cazul în care avem un singur nod .



După cum se observă din figură în primul interval de timp (slot 1) nodul transmite date către canalul A și B. În al doilea interval acesta transmite un cadru de date numai în canalul A, iar în ultimul cadru nu se transmite nimic.



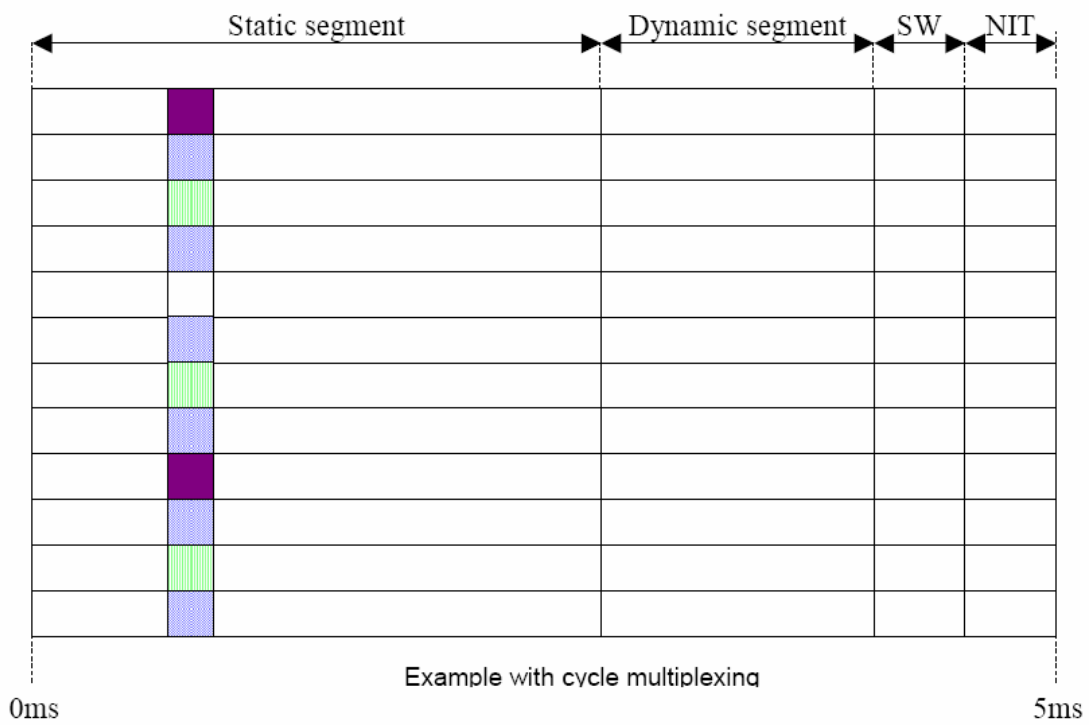
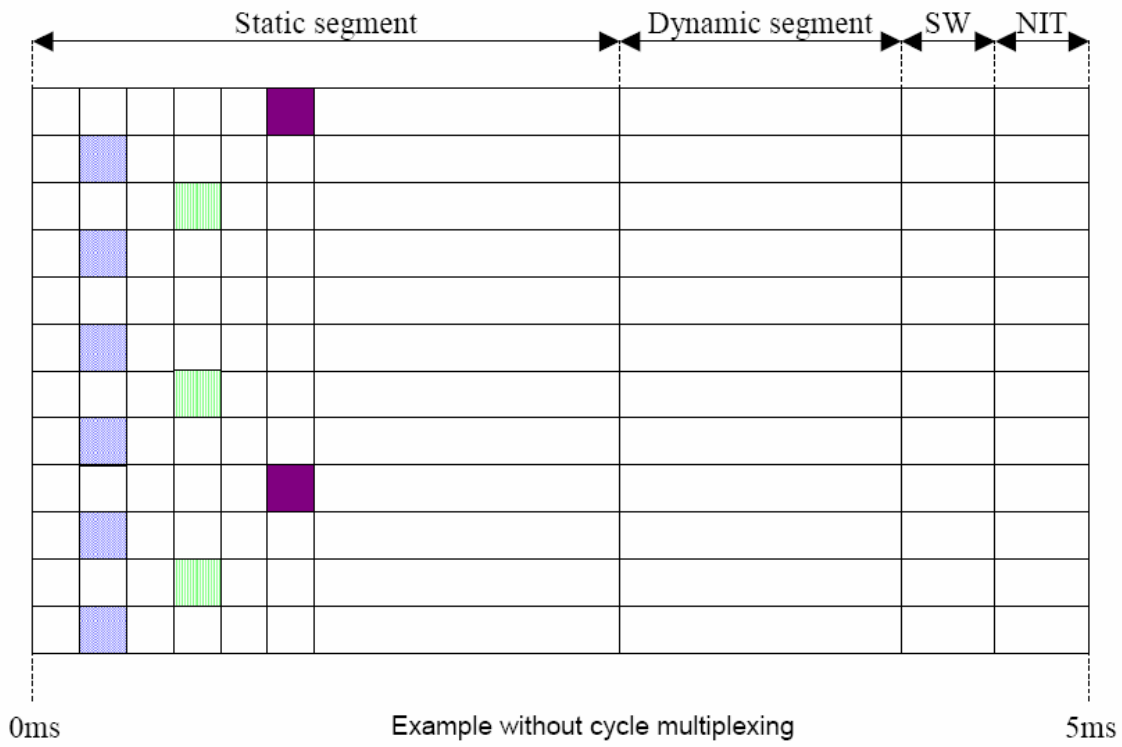
*Segmentul dinamic* este împărțit în *mini-slot-uri* a căror durată se poate modifica. Pentru a asigura o transmisie ordonată și programată se folosesc două numărătoare de cadre de timp (slot counter), câte una pentru fiecare canal. În cazul segmentului static aceste numărătoare pot fi incrementate simultan pe când în cazul segmentului dinamic acestea pot fi incrementate independent.



Fereastra de simboluri este de asemenea împărțită în durate de timp bine definite, aceasta este opțională, și are rolul de a transmite anumite simboluri de exemplu *medium access test symbol*.

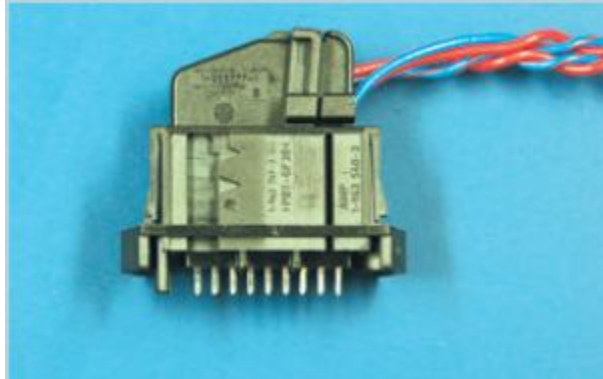
*Network idle time* este un interval de timp în care nodurile calculează și aplică anumite corecții asupra clock-ului, pentru a produce o sincronizare mai precisă.

*Multiplexarea ciclurilor* permite ca un ECU să poată transmite/recepționa diferite cadre de date în același timp. Această funcție a Flexray-ului permite o mai bună optimizarea a benzii de frecvență.

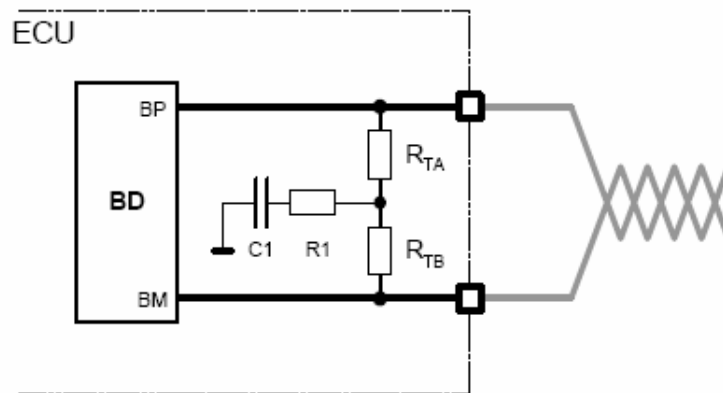


Caracteristicile mecanice:

Transmisia se realizează diferențial pe 2 fire sau prin fibră optică. Conectorul poate să arate ca în figura de mai jos:

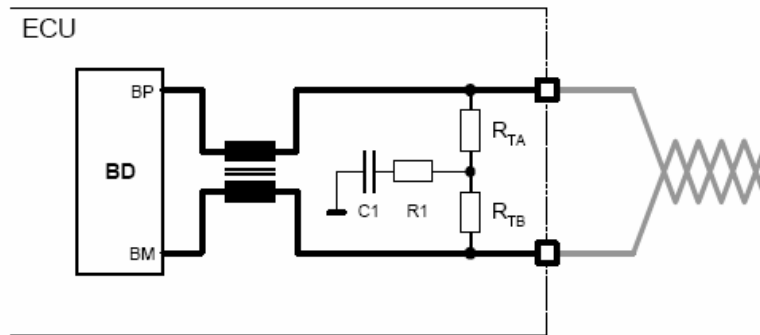


Terminatorul utilizat:

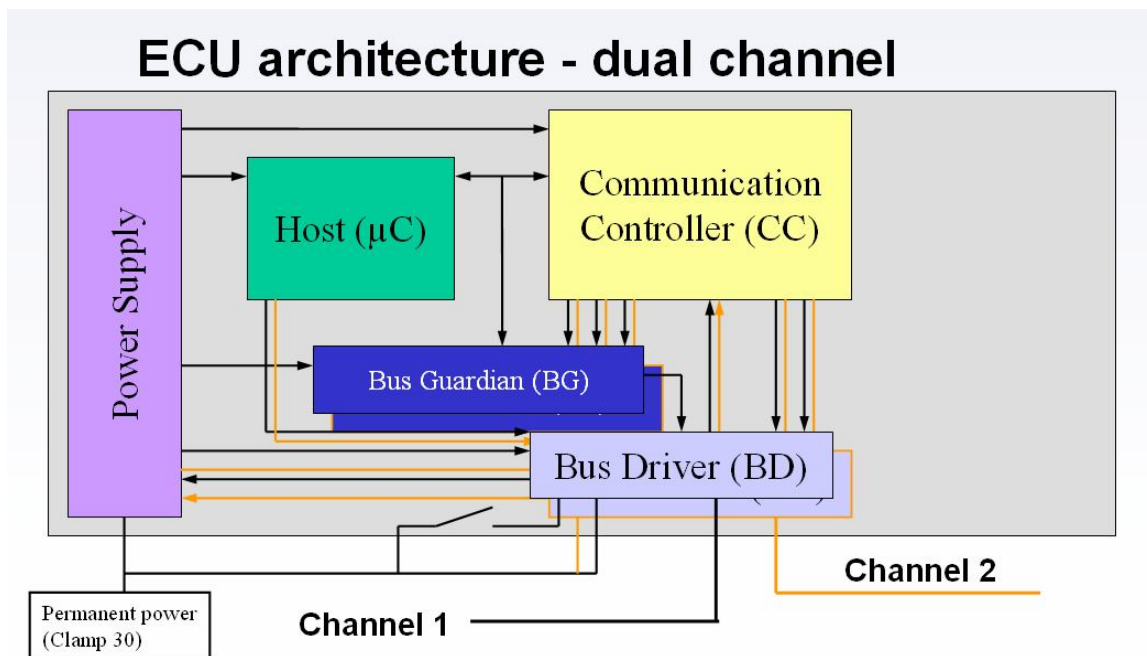


Name	Description	Typ	Unit
$R_1$	Resistor	< 10	$\Omega$
$C_1$	Capacitor	4700	pF
$2 \times  R_{TA} - R_{TB}  / (R_{TA} + R_{TB})$	Matching of termination resistors	$\leq 1$	%

Pentru o mai bună compatibilitate electromagnetică se utilizează următoarea schema:

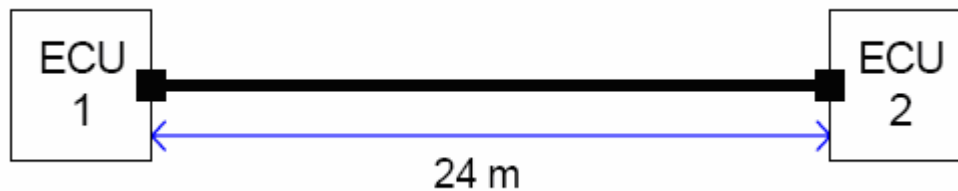


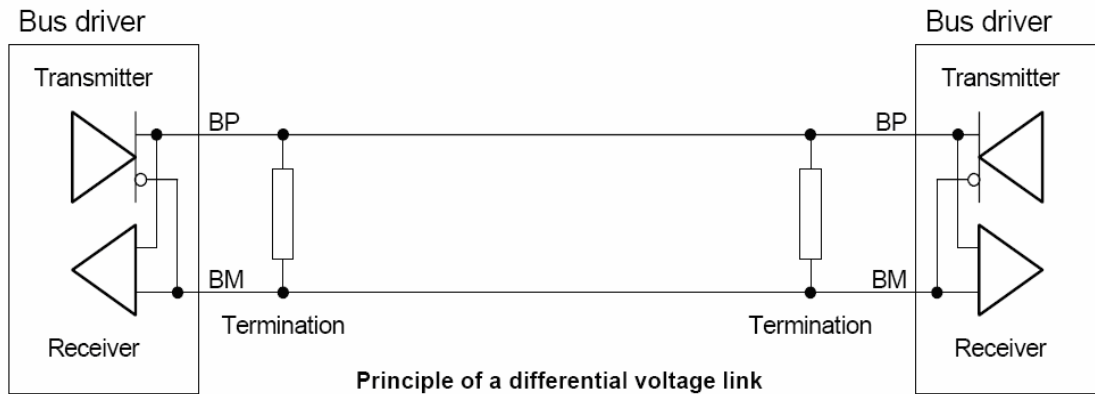
Schema bloc a unui ECU:



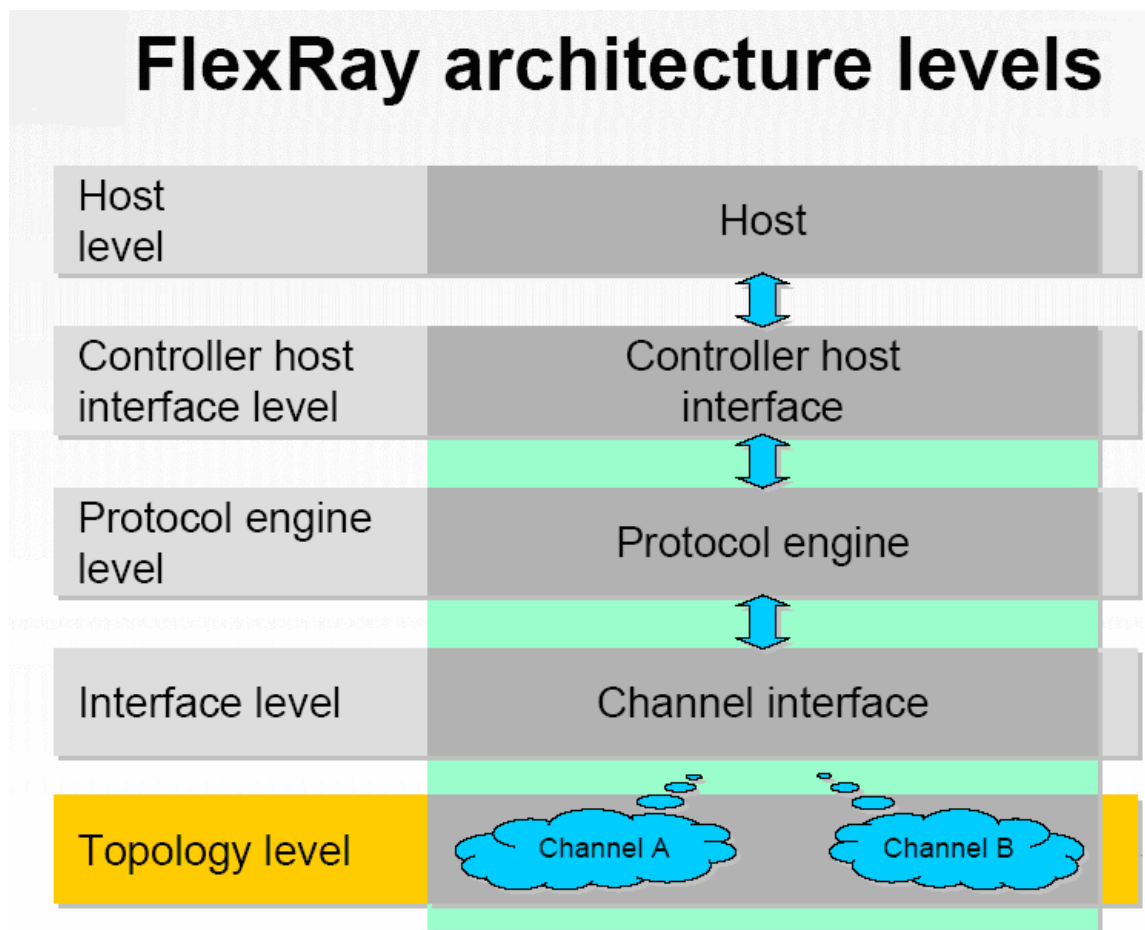
Transmisia începe numai când se primește accesul din partea BG (Bus Gardian). Acest acces este primit în cazul segmentului dinamic sau dacă a venit vremea nodului să trimită ceva pe magistrală. Dacă se dorește a se transmite ceva pe magistrală, trebuie anunțat din timp, adică în ultimul cadru de date. Acest lucru este necesar pentru a se asigura faptul ca magistrala nu este ocupată.

Dimensiunea maximă a cablului dintre 2 ECU este de 24m:





**Nivele arhitecturii FlexRay:**



Nodurile dintr-o rețea trebuie să se sincronizeze la începerea comunicației. Este foarte important ca un nod care intră mai târziu în rețea să nu perturbe celelalte noduri. Numai un nod care aparține ambelor canale este în măsură să înceapă această sincronizare.

În starea dinamică un nod master începe sincronizarea clock-ului, trimițând un SOC(Start of Cycle) la celelalte noduri. Algoritmul pentru starea statică este ceva mai complicat. Fiecare mesaj a oricărui nod conține și informații referitoare la sincronizare.