



BME
Budapesti Műszaki és Gazdaságtudományi Egyetem

HAUT
Közlekedésautomatikai Tanszék

Járműfedélzeti rendszerek II.

6. előadás

Dr. Bécsi Tamás

LIN (Local Interconnect Network) kommunikációs hálózat

Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedésautomatikai Tanszék

- 1980-as években jelentek meg az UART alapú soros megoldások a gépjárművekben,
- Gyors elterjedés ← – alacsony költség
- Különböző megoldások a fizikai szinten
- Különböző megoldások a protokollokban
- Felmerült a szabványosítás iránti igény - >
- LIN Consortiun (Audi, BMW, DaimlerChrysler, VW, Volvo – Volcano Tech., Motorola)

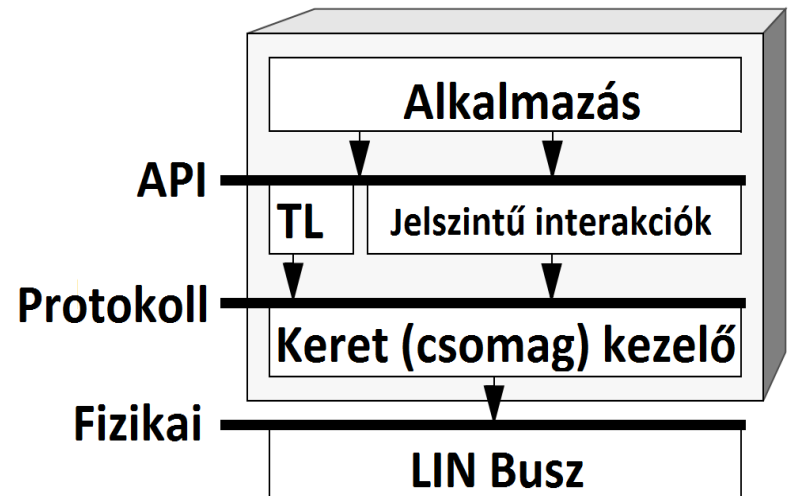
A LIN busz főbb tulajdonságai

- Egy Master (mester), több Slave (szolga) koncepció
- Alacsony költségű UART alapú integrált áramkörü megoldás, egyszerű állapotgép alapú működés
- Kvarc, illetve rezonátor nélküli szinkronizációs lehetőség a szolgák esetében
- Determinisztikus jelátvitel előre számítható jelterjedési idővel
- Alacsony költségű egyvezetékes megoldás
- 20kbit/s sebesség
- Jel érzékelésen alapuló interakció
- Újrakonfigurálhatóság
- Szállítási réteg és diagnosztikai támogatás

LIN rétegek

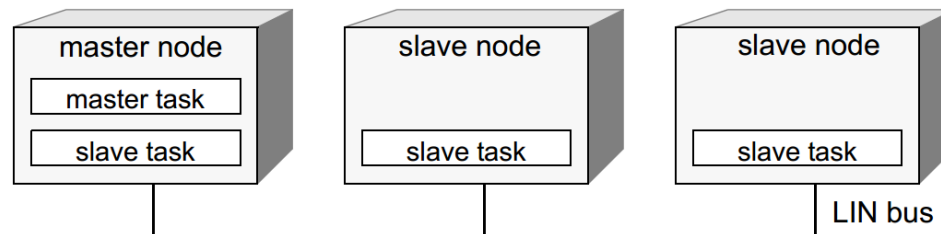
- A LIN szabvány a kommunikációt négy rétegre osztja, amelyet három interfész köt össze. A fizikai réteg és az alkalmazás között a keretkezelő és a jelfeldolgozó szolgáltat kapcsolatot.

Sokszor a jelek és keretek kezelése egyben, szoftveresen megvalósítva alkot közös interfészt az alkalmazás felé.



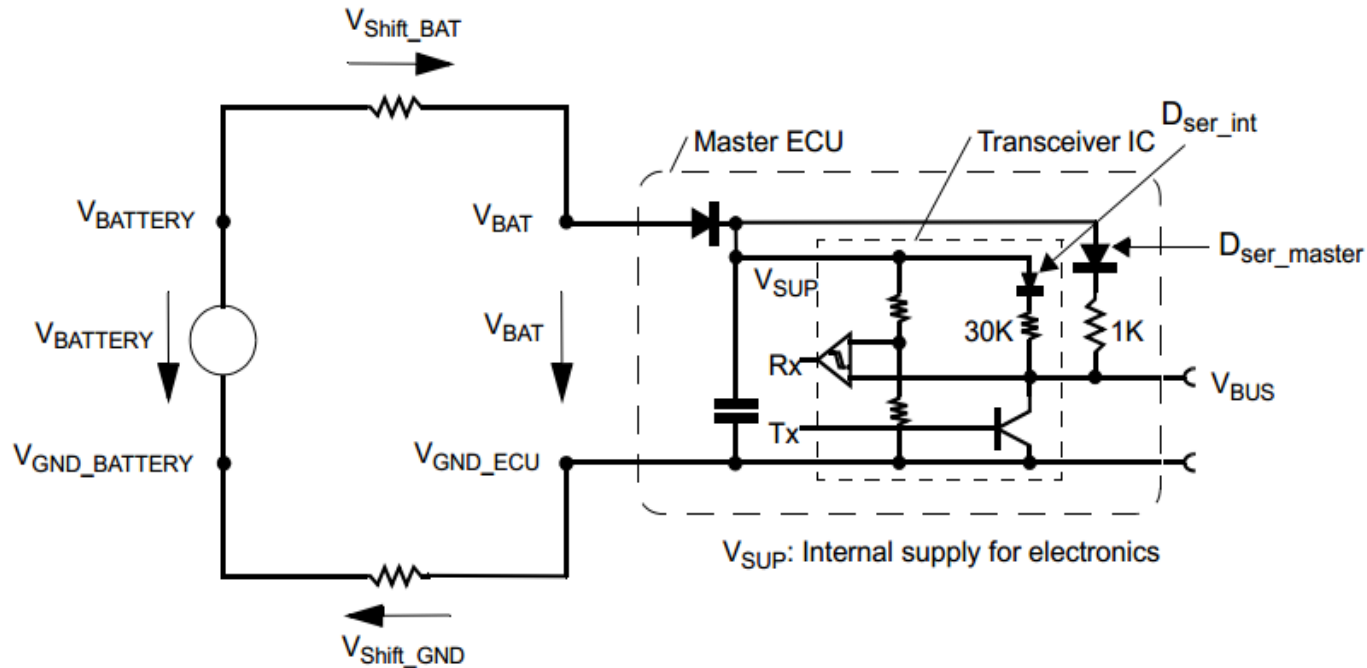
A LIN hálózat logikai felépítése

- Egy LIN hálózat egy mester és több szolga **folyamatból** áll. A mester csomópont mind mester, mind szolga feladatokat is ellát, míg a többi csomópont csak szolga folyamatokat végez.



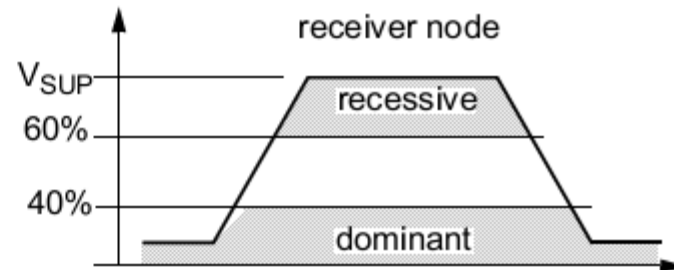
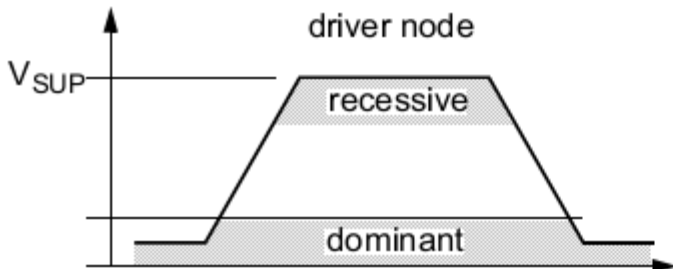
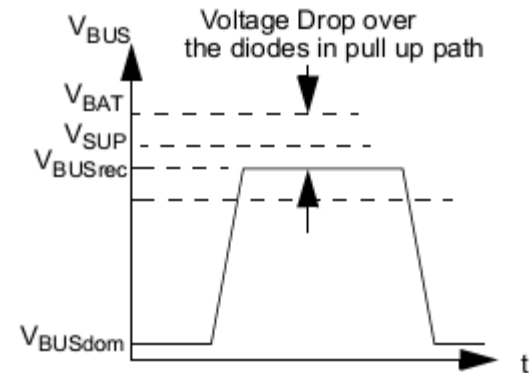
LIN Fizikai réteg

- „Egy vezetékes”
- Echo



Jelszintek

- V_{BUSrec} Tipikusan 12V
- Meghajtás 20%-80%
- Fogadás 40%-60%

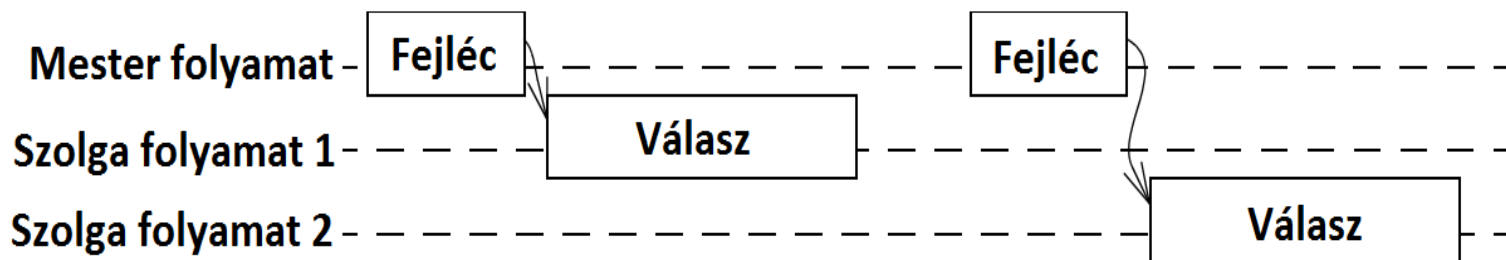


LIN Keretek

Frames

Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedésautomatikai Tanszék

- Egy adatkeret a mesterfolyamat által szolgáltatott fejlécből (header) és a szolgafolyamat által szolgáltatott válaszból (response) áll.



A protokoll tulajdonságai

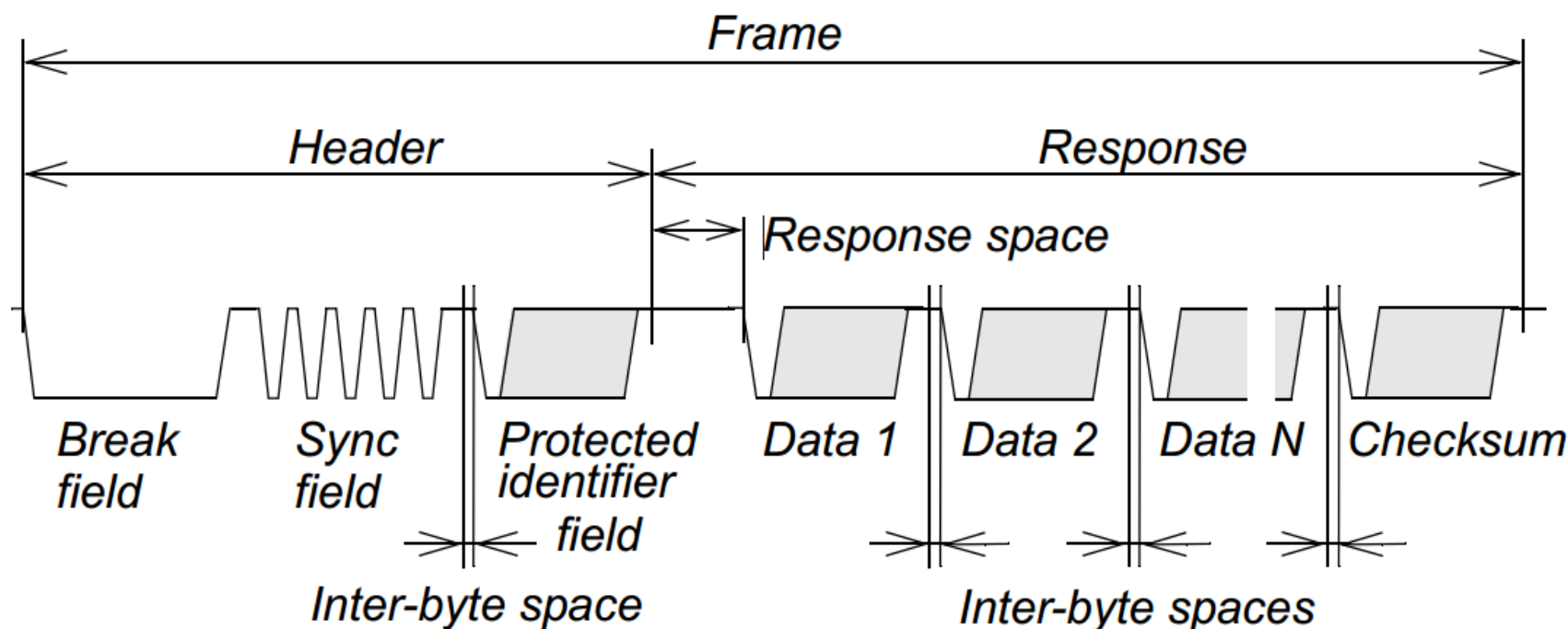
- Ez a protokoll a következő előnyös tulajdonságokat eredményezi:
 - **Flexibilitás:** A LIN hálózathoz tetszőlegesen lehet (szolga-) csomópontokat hozzáadni a többi szolgacsomópont szoftveres vagy hardveres átkonfigurálása nélkül.
 - **Üzenetcímzés:** A CAN-hez hasonlóan az üzenet címzettjét a keret azonosítója definiálja.
 - **Multicast:** Egyszerre több csomópont is felhasználhatja a keretek adatait.

LIN Keretek általános felépítése

Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedésautomatikai Tanszék

- Fejléc felépítése:
 - Break Signal (Legalább 13 bit hosszúágú)
 - Szinkronizációs mező
 - Keretazonosító
- Válasz felépítése:
 - Adatmező
 - Ellenőrzőösszeg

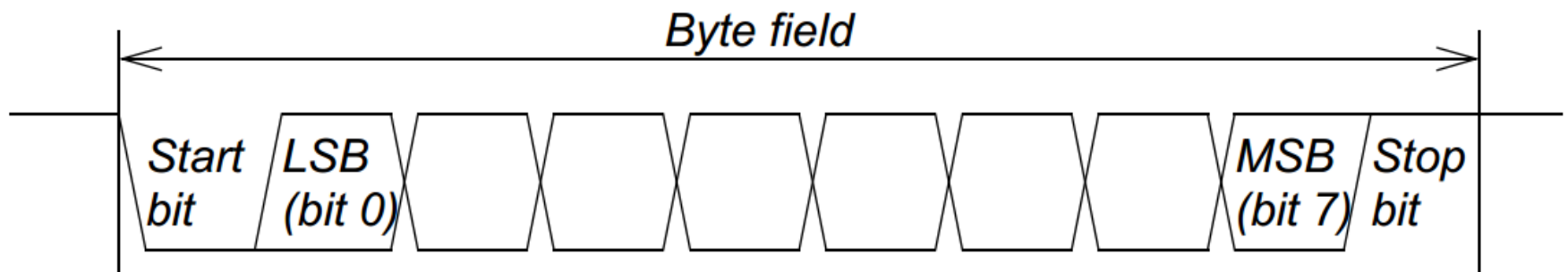
LIN Keretek általános felépítése



LIN keret byte mezőjének formátuma

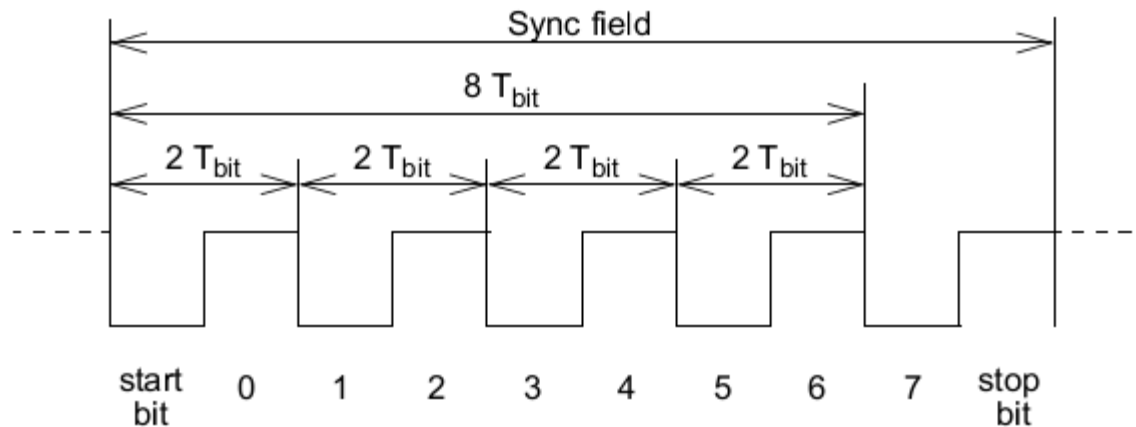
Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedésautomatikai Tanszék

- Start bit (Domináns 0)
- LSB-MSB adat
- Stop bit (Recesszív 1)
- (Plusz inter-byte space)



Szinkronizáció

- A szinkronizáció technikailag egy '0x55' adat byte kiküldését jelenti
- A szolga node így a lefutó élek távolságából meghatározhatja a „bithosszt”



Azonosító mező 1

- A *védett azonosító* (protected identifier) mező egy olyan byte mező, amely két almezőből áll: a keretazonosítóból és a hozzá tartozó paritásból. A keretazonosító hat bitből áll, így 0..63 értéket vehet fel, és háromféle csoportba sorolható:
 - 0..59 (0x3B) alkalmazandó a jeltovábbító keretekhez;
 - 60 (0x3C) mester igény (master request) és 61 (0x3D) szolgaválasz (slave response) a diagnosztikai keretek azonosítói; illetve
 - 62 (0x3E) és 63 (0x3F) fenntartott azonosítók későbbi protokollbővítéshez.

Azonosító mező 2

- A paritás a keretazonosító bitjeiből számítandó a következő módon:

$$P0 = ID0 \oplus ID1 \oplus ID2 \oplus ID4$$

$$P1 = \neg(ID1 \oplus ID3 \oplus ID4 \oplus ID5)$$

Adatmező

- Egy keret egytől nyolc byte-ig terjedő adatot szállíthat, de a keret hossza előre definiált kell, hogy legyen. A LIN keret utolsó eleme az ellenőrzőösszeg, amely két alapvető formájú lehet:
 - Klasszikus ellenőrzőösszeg, amely az adatbyte-ok átvitelével együtt vett összege invertálva; diagnosztikai keretek, illetve a LIN 1.x protokollt használó szolgák esetében alkalmazandó.
 - Továbbfejlesztett ellenőrzőösszeg (Enhanced checksum), amely a védett azonosítót is beleveszi az összeg számolásába, amelyet a LIN 2.x protokollt alkalmazó eszközök használnak.
- Az ellenőrzőösszegek használata értelemszerűen keretenként előre definiált kell, hogy legyen.

Időzítés

- A LIN mester folyamata folyamatosan karbantartja az időzítési táblát (schedule table). Így minden keret egy meghatározott keret pozícióban keletkezik (frame slot). A keret pozíciók hossza megegyezik a leghosszabb keret elméleti hosszával, plusz egy keret-közi állandó idővel. Mivel a közeghozzáférést egyedül a mesterfolyamat vezérli, ezért konfliktus csak speciális esetben alakulhat ki.

Feltétel nélküli keret (Unconditional frame)

Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedésautomatikai Tanszék

- Egyszerű jeleket szállítanak
- Keretazonosítójuk 0 és 59 közötti értékeket vehetnek fel
- Átvitelük a hozzájuk tartozó frame slot-ban történik
- A keret fejlécét a mester folyamat írja
- Az adatbeírásra kötelezett szolga folyamat kitölti az adatmezőket
- A fogadásra kötelezett szolgák értelmezik

Eseményvezérelt keret (Event triggered frame)

Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedésautomatikai Tanszék

- Célja a LIN hálózat sávszélességének optimalizálása
- Több szolga figyelése történhet egyszerre ritkán fellépő események esetében
- A mester a feltételes keret fejlécét kitölti
- A válaszadó szolgák egyike feltölti
- Ekkor ritkán konfliktus fordulhat elő
- A konfliktust a mester érzékeli és kezeli

Eseményvezérelt keret konfliktuskezelés

Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedésautomatikai Tanszék

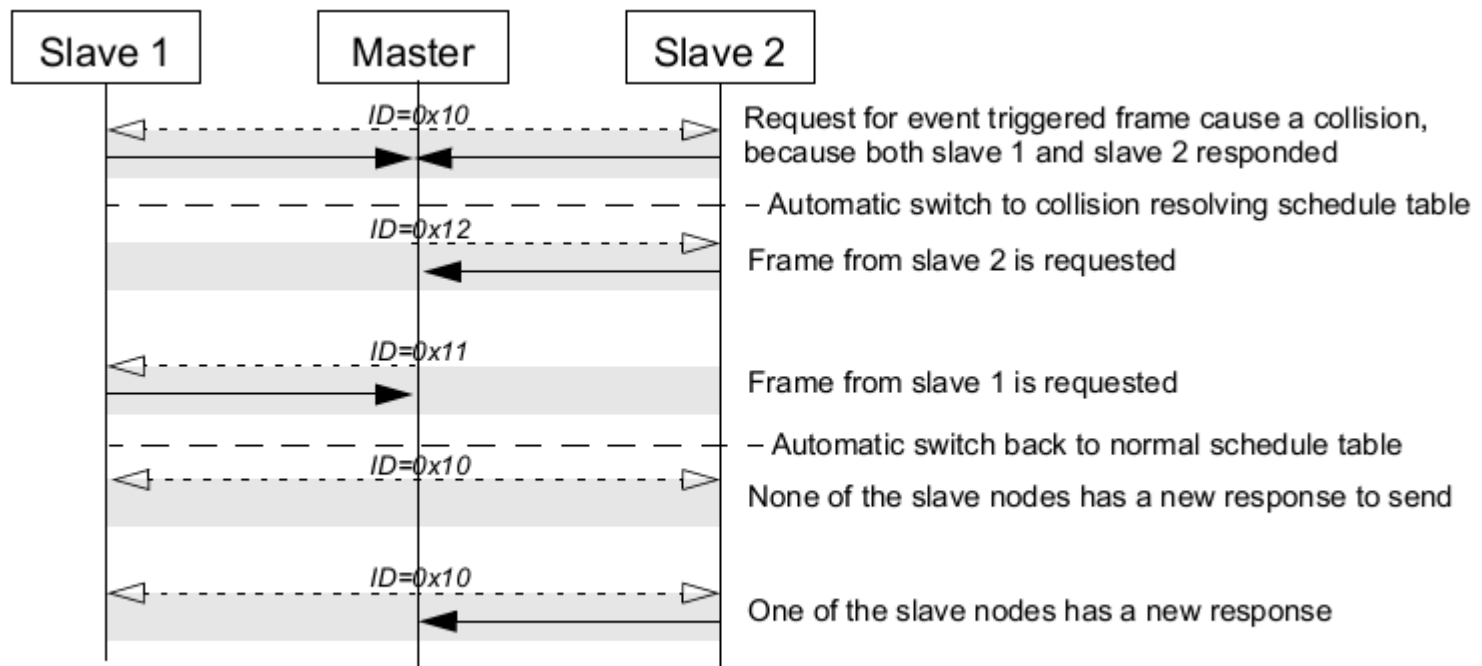


Figure 2.10: Event-triggered frame example.

Szórványosan megjelenő keret (Sporadic frame)

Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedésautomatikai Tanszék

- Olyan adatokat hordoznak, amelyek nem ütemes rendszerességgel kell, hogy megjelenjenek a hálózaton
- Több ilyen keret foglalhatja el ugyanazt a frame pozíciót az időzítési táblában
- Időzítésükről a mester folyamat dönt

Diagnosztikai keret (Diagnostic Frames)

Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedésautomatikai Tanszék

- A szállítási réteg információját továbbítják a hálózaton
- Két azonosítója lehet:
 - 60, mester igény (master request)
 - 61, szolga válasz (slave response)

Fenntartott keretek (Reserved Frames)

Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedésautomatikai Tanszék

- A LIN2.x hálózatban a 62,63 azonosítójú keretek nem használhatóak
- Későbbi protokollbővítésre vannak fenntartva

Jelek kezelése

- Egy keretben két típusú adat kerülhet továbbításra:
 - Jel (Signal), amely egy **skalár**, vagy **byte tömb**, amely a keret adatrészébe van csomagolva. a jelek pozíciója egy adott keretazonosítón belül állandó.
 - **Diagnosztikai üzenetek**, amelyek két fenntartott azonosítójú keretben közlekedhetnek.

Skalár jelek

- Hosszuk 1-16 bit között mozoghat
 - Az 1 bites jelet boolean jelnek nevezzük
 - A 2-16 bitesek unsigned-ként kezelendők
- Minden jelnek egy szolgáltatója van, de azt több keretben is küldheti.
- Minden jelhez tartozik egy kezdeti érték, amely a jel első írásáig érvényesnek tekintendő
- A jelek kódolása LSB-MSB sorban történik
- A jelek átnyúlhatnak a kerethatárokon

Byte array jelek

- 1-8 byte hosszúak lehetnek
- Minden esetben kötöttek a keret byte határaihoz

A hálózat leírása

az LDF (LIN Description File)

Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedésautomatikai Tanszék

```
<LIN_description_file> ::=
LIN_description_file ;
<LIN_protocol_version_def>
<LIN_language_version_def>
<LIN_speed_def>
(<Channel_name_def>)
<Node_def>
(<Node_composition_def>)
<Signal_def>
(<Diag_signal_def>)
<Frame_def>
(<Sporadic_frame_def>)
(<Event_triggered_frame_def>)
(<Diag_frame_def>)
<Node_attributes_def>
<Schedule_table_def>
(<Signal_groups_def>)
(<Signal_encoding_type_def>)
(<Signal_representation_def>)
```

Vége

Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedésautomatikai Tanszék

Köszönöm a figyelmet!