

JÁRMŰIPARBAN GYAKRAN ALKALMAZOTT FEDÉLZETI BUSZOK

1. BEVEZETÉS

A járműfedélzeti rendszerek bonyolultságának és a megoldandó feladatok komplexitásának növekedésével a huzalozási igény akár nem lineáris módon is növekszik. A kábelezés mérete több szempontból is jelentős. Földi járművek esetén elsődleges probléma az ár, ennek alapvető oka a réz magas ára és a sorozatgyártás – főleg személygépkocsi gyártás – nagy volumene. Légi járművek esetén ez szintén fontos kérdés, első sorban kis repülőeszközök esetén kritikus lehet a súly. Mindkét kategóriában jelentős szempont a megbízhatóság, amely természetesen a komplexitás növekedésével csökken.

A 90-es évek elejétől az autógyártók kifejlesztettek számos fedélzeti informatikai hálózatot a fedélzeti huzalozás egyszerűsítésére, amely a gépjármű, vasúti közlekedésben igen széles körben elterjedt.

Az elmúlt 20 évben ezek a hálózatok és a hozzá kapcsolt informatikai elemek sokat fejlődtek, áruk jelentősen csökkent, megbízhatóságuk – annak ellenére, hogy komplexitásuk nőtt – jelentősen emelkedett. Alkalmazásukhoz szabványokat és ajánlásokat dolgoztak ki, amelyek a fejlesztésben bekövetkező szoftver hibák előfordulását csökkentik.

Ez a cikk a CAN, a LIN busszal foglalkozik, érinti a további szóba jöhető megoldásokat, mint a Flexray busz és a MOST protokoll.

2. CAN BUSZ

A CAN (Controller Area Network) a BOSCH kezdte el fejleszteni 1983-ban. A protokollt 1986-ban a Detroiti SAU (Society of Automotive Engineers) kongresszuson fogadták el. Az első CAN interfész áramkört 1987-ben az Intel és a Philips adta ki. A CAN 2.0 protokollt a BOSCH 1991-ben publikálta.

Nyugodtan kijelenthetjük, hogy 1995 után tervezett és gyártott gépkocsikban már ez a busz megtalálható, de használják az intelligens épületek informatikai rendszerében, vasúti berendezésekben, terepi irányítástechnikai rendszerekben, a hajózásban és a repülésben is egyre szélesebb körben. Természetesen ez a felsorolás nem teljes.

A CAN busz alapvetően a következő tulajdonságokkal rendelkezik:

- a buszon közlekedő üzeneteknek prioritása van,
- adott üzenet időtartam (kismértékű bizonytalansággal),
- könnyű konfigurálhatóság,

- több állomás veheti az üzeneteket,
- hiba detektálás és jelzés,
- hiba esetén automatikus üzenet ismétlés,
- a hiba típusának detektálása (időszakos és folytonos hiba).

A CAN busz ezeket a funkciókat három rétegben valósítja meg. Ezek:

- fizikai réteg,
- szállítási réteg,
- objektum réteg.

A fizikai réteg definiálja a módot, hogy milyen módon jutnak el a jelek az egyik pontból a másikba.

A szállítási réteg a CAN protokoll tulajdonképpeni „magja”. A szállítási réteg felelős a bitek időzítéséért és szinkronizálásáért, az üzenet keretek (frame) összeállításáért, az arbitrációért, az üzenetek nyugtázásáért, hiba detektálás és hiba behatárolás.

Az objektum réteg felelős az üzenetek szűréséért, kezeléséért és az interfész státuszának kezeléséért.

A CAN busz többféle átviteli szabványt valósít meg. Jelenleg a leggyakrabban a CAN2.0A és a CAN2.0B szabványok a használtak.

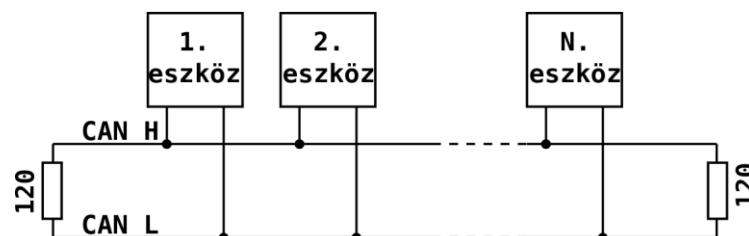
Fizikai rétegben a bit átvitelnél a jelenleg használt az ISO 11898 szabvány az elterjedt. Bár létezik az ISO 11519-2 szabvány is, de a készülék gyártók kerülnek, mert azonos költségekkel gyengébb szolgáltatást nyújt.

A busz sebessége 40kbit/s -tól 1Mbit/s tartományban használható (ISO 11898 esetén). Az áthidalható maximális távolság az átviteli sebességtől függ. 40 kbit/s sebességnél 1km, 1Mbit/s sebességnél 40m. Ennek egyszerű fizikai okai vannak, erre később kitérünk.

A busz jellegű kialakítás miatt az átvitel félduplex.

A CAN busz vezetékezése szimmetrikus, jelátvitel differenciális és a két végén 120 Ohmos lezárást igényel.

Alacsonyabb sebességeknél a lezárás hiánya nem kritikus. Ezt az alkatrészgyártók az interfész áramkörök esetén a jelfelfutási sebesség korlátozásával is elősegítik.



1. ábra. CAN vezetékezés

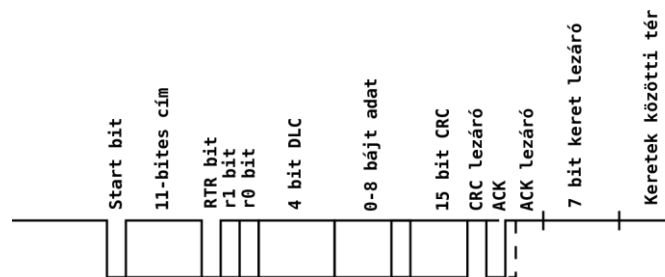
A két vonal a differenciális adatátvitel miatt ellenkező jelszinteken dolgozik a CAN_L vezetéken a domináns jelszint a logikai '0' a CAN_H vezetéken a domináns jelszint a logikai '1'.

A CAN busz többféle keretet kezel. Ezek:

- adatkeret (data frame)
- standard keret
- kibővített (extended) keret
- vezérlési (remote) keret
- hiba keret (error frame)
- „túlfutás” keret (overload frame)

2.1. Standard adatkeret

A standard adatkeret felépítése a 2. ábrán látható. Az ábra a CAN_L vezeték jelalakját mutatja. A CAN_H vezeték jelalakja logikai jeleket tekintve ennek „tükörképe”.

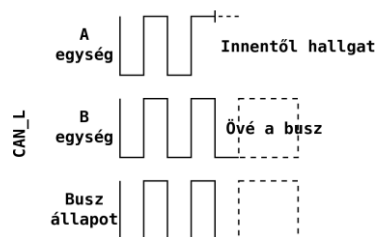


2. ábra. Standard adatkeret

Az adatkeret egy egy bit hosszúságú domináns bittel kezdődik. Ezt követi a 11 bites cím mező, amelyet az RTR bit követ. Az RTR bit mondja meg, hogy a kérdéses keret adatkeret-e, vagy vezérlési keret-e. Az R1, R0 bitek jelenleg továbbfejlesztési célra fenntartott bitek. Ezt követi a 4 bites DLC mező, amely megmondja, hogy hány adatbájtt fog az ezt követő adatmezőben jönni. Az adatmező hossza 0-8 bájt lehet. Az adatmező után a 15 bites CRC mező következik, amit a CRC lezáró recesszív bit követ. Ezután jön a az ACK mező. Ha ez a mező egy domináns bit, akkor az átvitel sikeres, ha 1.5 bit hosszúságú, akkor az átvitelben hiba történt. Ha recesszív marad a keretet senki nem vette. Az ACK mezőt egy 7 bit hosszúságú recesszív jelszintű mező követi, ez a keret lezáró. Ezután a keretek közötti „terület” jön, ami minimum 3 recesszív bit.

A standard adatkeretről beszéljük meg az átvitel néhány jellegzetességét.

A CAN busz esetén nincsen kitüntetett eszköz mindenki kezdeményezhet adatátvitelt. Amikor valamelyik eszköz adatátvitelt kezdeményez, akkor megvizsgálja, hogy a busz szabad-e (ezt elvileg a protokollgép tudja), ha igen megkezdje az átvitelt, ha nem, akkor várakozik. De előfordulhat az az eset, hogy két egység is egy időben kezdi az adást. Ekkor mindkettő ad addig, míg valahol a címmezőben az aktuális címbit az egyik egységnél nem lesz domináns és a másik egységnél nem lesz recesszív. Ekkor a domináns címbitet adó egység folytatja az átvitelt a recesszív bitet adó egység pedig leválik a buszról és csak hallgat. Tehát az alacsonyabb címetek adó egységnek van prioritása.



3. ábra. Arbitrációs versenyhelyzet

Ezt a tulajdonságot a repülésben alkalmazott CAN busznál kihasználják.

A CAN üzenetek úgynevezett multicast vagy ATM (Anyone to Many) üzenetek. Ez azt jelenti, hogy a CAN üzenet címrésze nem annak az állomásnak a címét adja meg, akinek az üzenet szól, hanem az üzenetet azonosítja. Akinek az üzenetre szüksége van az azt feldolgozza. Ez azonban azt jelenti, hogy a buszon lévő minden eszköz veszi az üzenetet és nyugtázza a vételt.

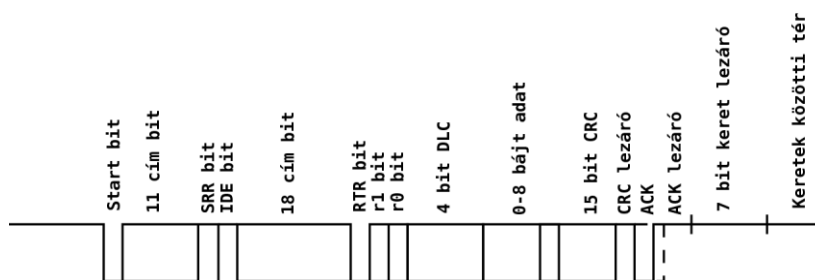
A sikeres nyugtázás egy egy bit szélességű domináns bit. Ha valamely állomás hibásan veszi az üzenetet, akkor az negatív nyugtázást küld, ami egy másfél bit hosszúságú domináns jel.

Ez az a működési jellemző, amely korlátozza a busz hosszát a sebesség függvényében. Ha a nyugtázás negatív, vagy nem érkezett nyugtázás a küldő újraküldheti az üzenetet.

2.2. Kibővített adatkeret

A 4. ábra a kibővített adatkeret felépítését mutatja. A start bit után a standard keretben szereplő 11 bites cím következik, ezt a cím tartományt tekintik az ID28-ID18 biteknek. További megszorítás, hogy a cím felső 7 bitje (ID28-ID22) nem lehet mind recesszív értékű. Ezután a recesszív SSR és az IDE bitek jönnek. Ezután még egy 18 bit hosszúságú cím mező következik. A keret további felépítése megegyezik a standard keret felépítésével.

Meg kell jegyeznünk, hogy a kibővített keretet nagyon kevés helyen alkalmazzák.



4. ábra. kibővített adatkeret

2.3. Vezérlési keret

A vezérlési keret felépítésében teljesen megegyezik egy olyan adatkeret felépítésével, amely 0 adatbájtot tartalmaz. Az alapvető különbség az, hogy az RTR bit értéke recesszív. A DLC mező értéke indifferens.

A vezérlési keretet általában kiszolgáláskérésre használják. Például egy egység adatokat kér a többiektől.

2.4. Hibakeret

A CAN adatkeretben van egy biztonsági intézkedés. Ez előírja azt, hogy egymás után nem lehet öt egyforma bitnél több. Ha mégis szükség lenne arra, hogy több egyforma bitet adjunk, akkor a CAN protokoll gépe egy ellentétes bitet szűr be a bitfolyamba, amelynek az értéke nem számít, csak biztonsági szerepe van. Például ha öt '0' értékű bit jött a buszon, akkor mindenképpen egy '1' -es értékű bit következik.

Ezt a tulajdonságot használják ki az úgynevezett hibakeretben. Ha egy átvitel esetén ötnél több egyforma bit jelenik meg, akkor hiba történt és az átvitel megszakad. Tehát a hibakeret minimális mérete hat bit, a maximális mérete a szuperpozíciók miatt 12 bit.

A hibakeret lehet:

- passzív. Ekkor a hibajelzés recesszív bitfolyammal történik.
- aktív. Ekkor a hibajelzés domináns bitfolyammal történik.

2.5. Túlfutás keret

A túlfutás keret (overload frame) akkor kerül alkalmazásra, ha egy eszköz az adat keret feldolgozásával valamilyen okból nem végez és ezért időt kér.

A túlfutás keret a keretek közötti tér (inteframe space) minimálisan kötelező 3 bites recesszív bitfolyama helyett egy hat bites domináns bitsorozatot ad, amit egy nyolcbites recesszív bitsorozat követ.

2.6. A CAN hibakezelése

A CAN hibakezelése igen kifinomult. Ötféle hibadetektálást valósítottak meg a protokollban, ezek:

1. Bit hiba. Az az egység, aki a kérdéses bitet adja, azt vissza is olvassa. Ha a visszaolvasott érték nem azonos az adottal, akkor bithiba keletkezett. Ez alól kivétel az arbitrációs mezőben adott recesszív bit (lásd korábban) és az ACK mezőben az ACK lezáró. Ekkor az adó egy passzív hibakeretet ad.

2. Bitsorozat hiba. Ha a keretben ötnél több egyforma bit követi egymást, akkor bit sorozat hiba történt.

3. CRC hiba. Az adat és vezérlési kereteket egy 15 bites CRC mező egészíti ki. A CRC generátor polinomja: $X^{15}+X^{14}+X^{10}+X^8+X^7+X^4+X^3+1$

4. Formai hiba. Ez akkor következik be, ha a kötött helyű és értékű bitek helyén egy, vagy több illegális értékű bit található.

5. ACK hiba. Ez akkor következik be, ha az ACK mezőben se pozitív, se negatív ACK nem történik. Nevezetesen recesszív jelszint tapasztalható a buszon.

2.7. A CAN busz alkalmazási formái

A CAN buszt eredetileg úgynevezett esemény által kiváltott üzenet küldésre szánták tervezői (event triggered CAN, röviden ET CAN). Ez azt jelenti, hogy valamilyen változás a monitorozott rendszerben egy üzenet küldését eredményezi.

Bár maga a CAN protokoll nagyon biztonságosan működik, a feldolgozó elektronika, illetőleg a működtető szoftver hibázhat, például kicsúszhat az időből. Ezért a járműiparban a kritikus helyeken a monitorozott rendszer kritikus értékeit meghatározott időnként elküldik. Ezt az alkalmazási formát idő által kiváltott üzenet küldésnek nevezik (time triggered CAN, vagy röviden TT CAN).

Vannak olyan állapotok, amelyek esetén mindenképpen szükséges az üzenetek folyamatos küldése. Ha az állapot megszűnik, akkor nincs szükség erre az ismeretre. Ezt az alkalmazást esemény által kiváltott - időzített üzenetnek hívjuk (event-time triggered CAN, röviden ETT CAN).

2.8. A CAN busz a repülésben

A CAN busz alkalmazása a repülésben gyakorlatilag megjelenésétől kezdve történik. A fokozott biztonsági követelményeknek megfelelően az alkalmazásoknak egy részébe az úgynevezett egy-az-egy (peer-to-peer, PTP) kapcsolatot hoznak létre, amely logikailag ellentmond a vezetékezés csökkentésének. Azonban a szabványosítás és az egyéb előnyök, például a diagnosztika miatt célszerű az alkalmazása. Az alkalmazások másik részében a hagyományos ATM jellegű megoldások szerepelnek.

Kihasználva a CAN busz arbitrációs tulajdonságait az üzenetek prioritásait a címekhez kötik. A z 1. táblázat megadja az üzenetek jellegét és egyéb tulajdonságait az úgynevezett CANaerospace szabványban.

Rövidítés	Jelleg	Leírás	Cím	Prioritás
EED	ATM	Vészhelyzeti adatcsatorna	0-127	Legmagasabb
NSH	PTP	Magas prioritású adatcsatorna	128-199	
UDH	ATM/PTP	Magas prioritású felhasználói adatcsatorna	200-299	
NOD	ATM	Normál prioritású adatcsatorna	300-1799	
UDL	ATM/PTP	Alacsony prioritású felhasználói adatcsatorna	1800-1899	
DSD	ATM/PTP	Hibakeresési adatcsatorna	1900-1999	Legalacsonyabb
NSL	PTP	Alacsony prioritású adatcsatorna	2000-203	

1. táblázat. CAN üzenetek kategóriái és a címtartományok

A szabvány a légijármű számos paraméterére kiterjed. A következő táblázat néhány paraméter azonosítóját mutatja be.

Cím	Paraméter	Adattípus	Dimenzió
303	Bólintás értéke	float short	fok
304	Dőlés mértéke	float short	fok
305	Elfordulás mértéke	float short	fok
800-807	Hidraulika nyomás	float short	hPA
1039	GPS sebesség	float short	m/s

2. táblázat. Néhány CAN üzenet címe példaként

Az üzenetek Time Triggered módon kerülnek továbbításra. A legrövidebb időszel 12.5ms, amely 80kHz frekvenciának felel meg, a leghosszabb 1s. Az üzenetek szigorú időszelben (time slot) kerülnek adásra, így csökkentve az ütközések számát. A rendszer felépítése lehetőséget nyújt a redundancia kezelésére is.

A szabvány előírja a fizikai réteg kialakítását is, definiálja a felhasználható átviteli közeget, a csatlakozókat és EMC problémák csökkentésére az eszközök galvanikus leválasztását.

3. LIN BUSZ

LIN (Local Interconnection Network) 1999 publikálta a LIN konzorcium. Alapvetően egy nagyon olcsó kommunikációs lehetőséget próbáltak biztosítani intelligens szenzorok és végrehajtók számára. Ezt a célt tökéletesen elérték.

Az busz alapvető jellemzői:

- egy busz mester és maximálisan 15 szolga tehető egy vonalra,
- egyvezetékes busz (persze földvezeték kell),
- sebessége 1 – 20 kbit/s, a leggyakrabban használt sebességek 2.4 kbit/s, 9.6 kbit/s és a 19.2kbit/s,
- a busz maximális hossza 40m,
- sebességét önmaga szinkronizálja (nem tartalmaz kvarcot a kontroller),
- az üzenet hossza 2, 4 és 8 bájt lehet,
- van hibadetektálás,
- a fizikai réteg ISO9141,
- van lehetőség alvó üzemmódra.

A LIN busz előnyei:

- könnyű a használata (a CAN busz használatához képest),
- az alkatrész gyártók biztosítanak eszközöket,

- olcsóbb a CAN busz elemeinél (kevesebb mint fele a CAN elemek árának),
- csökken a vezetékezés mértéke,
- könnyű bővíteni.

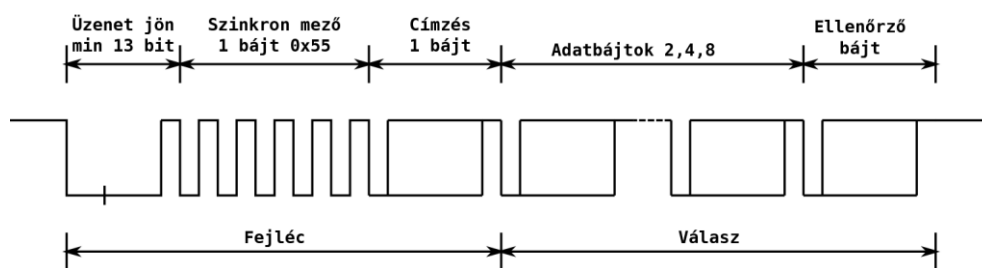
Itt meg kell jegyeznünk, hogy a LIN busz nem helyettesíti a CAN buszt, hanem lehetőséget ad a CAN csomópont intelligens bővítésére.

3.1. LIN protokoll

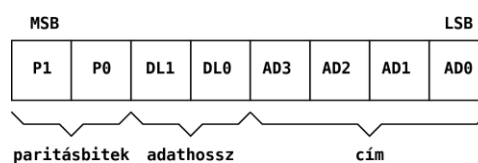
A LIN busz úgynevezett egy mesteres busz. A mester feladata a keretek indítása.

A keret egy úgynevezett break karakter küldésével indul, ez minimum 13 bit hosszúságú logikai '0' szint, amit egy logikai '1' zár. Ezután az aszinkron soros adatátvitelnek megfelelő logikai bitkódolás szerint jönnek az adatok. Tehát egy '0' értékű start bit 8 adatbit és végül egy logikai '1' értékű stop bit. A LIN keret felépítése az 5. ábrán látható.

A break karakter után egy 55h értékű szinkron karakter következik, majd a cím bájtt, lásd 6. ábra.



5. ábra. LIN keret



6. ábra. A cím bájtt

A cím bájtt bitjeinek jelentése:

- P1 és P0 paritásbitek, ahol:
 $P0 = \text{not} (AD1 \text{ exor } AD3 \text{ exor } DL0 \text{ exor } DL1)$
 $p1 = AD0 \text{ exor } AD1 \text{ exor } AD2 \text{ exor } DL0$
- DL1 és DL0 adathossz kódok

0	0	2 bájt
1	0	2 bájt
0	1	4 bájt
1	1	8 bájt
- AD3 – AD0 címbitek

A címzés után a szolgák küldik adataikat (a mester is működhet szolgaként). A küldött adatmennyiség lehet – mint azt feljebb láttuk - 2, 4 és 8 bájt. Az adatok végén egy ellenőrző összeg kerül átvitelre.

Az ellenőrző összeg csak az adatbájtokat ellenőrzi. Az ellenőrző összeg képzése a következő példán követhető:

Adott 4 adatbájtunk: 4Ah, 55h, 93h, E5h

1. $4Ah + 55h = 9Fh$,
2. $9Fh + 93h = 132h$, de ez túlsordult, az átvitelt hozzáadjuk az eredmény alsó bájtjához, ezért 33h,
3. $33h + E5h = 118h$, itt is túlsordult, tehát 19h,
4. ezt bitenként negáljuk, így az eredmény E6h

A LIN busz a következő hibákat képes detektálni:

- paritás hiba a címmezőben,
- bit hiba, az adó nem azt olvassa vissza, amit adott,
- ellenőrző összeg hiba,
- a szolga nem válaszol,
- rossz szinkron bájt.

Látható, hogy a LIN elég biztonságosan működik, bár összehasonlítva a CAN busszal ez a biztonsági szintjel alacsonyabb.

3.2. A LIN busz javasolt felhasználása a repülésben

Egyértelmű, hogy sebesség és megbízhatósági kérdések miatt kritikus területeken a LIN busz nem használható. Viszont kiegészítő rendszereknél, mint járulékos műszerezés, klíma berendezések kezelése, hangosítás és kabinvilágítás kezelésére, bináris szenzorok kezelésére alkalmazható olcsósága és egyszerűsége miatt.

4. EGYÉB ALTERNATÍVÁK

Az autóiparban alkalmazott buszok közül még kettő jöhet szóba mindkettő nagy teljesítményű, nagy megbízhatóságú eszköz. Ezek a Flexray busz és a MOST.

4.1. Flexray

A Flexray egy skálázható flexibilis nagy sebességű kommunikációs rendszer. A vezető autóipari cégek által létrehozott konzorcium fejlesztette ki. Ez a konzorcium 2009-ben megszűnt.

A Flexray kiválóan alkalmas:

- fedélzeti gerinchálózatok kialakítására, amelyhez különböző alhálózatok kapcsolódhatnak, különböző avionikai rendszerek közti távolsági kommunikációra,

- elosztott intelligenciájú rendszerek közötti kommunikáció megvalósítására,
- biztonságkritikus rendszerek kommunikációs hálózatok megvalósítására, például elektronikus kormányvezérlés.

Jellemzői:

- a fizikai réteg 2, vagy redundáns 4 vezetékes sodrott érpár,
- az áthidalható távolság maximálisan 24m,
- sebessége 10 Mbit/s,
- a buszon több mester lehet,
- felépítése lehet busz és csillag (ettől kezdve bármi).

4.2. MOST

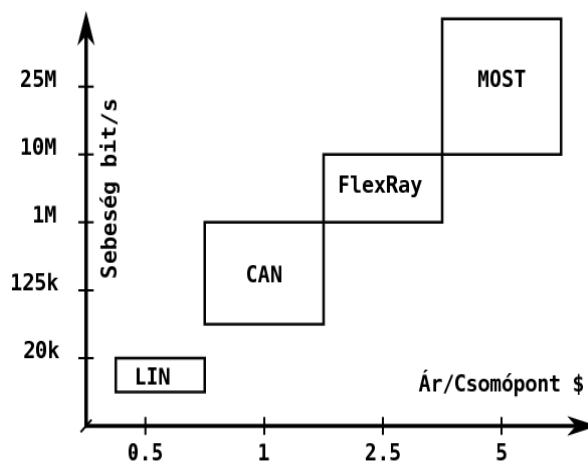
A MOST protokoll egy multimédiás optikai hálózat. Jellemzői:

- a fizikai átvivő közeg optikai kábel, vagy réz,
- sebessége:
- MOST25 ~25Mbit/s (Európa),
- MOST50 ~50Mbit/s (Japán, USA, nem optikai),
- MOST150 ~150Mbit/s (Európa kísérleti fázis).
- gyűrű topológia egy mester maximálisan 64 szolgál,
- TCP/IP alapú kommunikáció.

A MOST -ot alapvetően nem vezérlésre fejlesztették ki, hanem kommunikációs célokra. Érdekes kérdés, hogy lehetne-e ezt a protokollt járműirányításra is alkalmazni.

5. ÖSSZEFOGLALÁS

Az általános felhasználású fedélzeti buszok a járműiparban több mint húsz éve bevezetésre kerültek. A szoftver és hardver elemek megbízhatósága rendkívüli módon megnőtt. Számos új valósídejű operációs rendszert fejlesztettek, amelyek támogatják a fedélzeti információs rendszerek működését és folyamatos diagnosztikát készítenek a megbízhatóság növelése és a karbantarthatóság optimalizálása érdekében. Az alkalmazott eszközök ára is jelentősen csökkent az utóbbi időben. Erre mutat példát a következő diagram.



7. ábra. Árak és a sebesség csomópontonként

Meglévő légi járműveken alkalmazhatók pótlólagos műszerezésre:

- speciális meteorológiai adatgyűjtésre,
- légifényképezés vezérlésére,
- kameramozgatásra,
- egyéb mérőberendezések kezelésére,
- ejtőberendezések irányítására,
- járulékos és utólagos műszerezés informatikai rendszerének kialakítására és vezérlésére.

Ebben a cikkben egy közepes és egy alacsony „igényű” fedélzeti buszt mutattunk be és érintettünk további két rendszert. Látható, hogy a követelményeknek megfelelően találhatunk olyan technológiát, amely kielégíti az igényeket.

Ez a cikk nem törekedett az összes alkalmazott fedélzeti busz ismertetésére. Célunk az volt, hogy rávilágítsunk, hogy lehetséges a földi járművekben alkalmazott megoldások gazdaságosan és megbízhatóan alkalmazhatók a repülésben is.

FELHASZNÁLT IRODALOM

- [1] BOSH CAN Specification 2.0 Part A
- [2] BOSH CAN Specification 2.0 Part B
- [3] http://www.aviationtoday.com/av/issue/feature/CAN-Bus-in-Aviation_31468.html
- [4] <http://en.wikipedia.org/wiki/CANaerospace>
- [5] Jürgen Klüser: „CAN-based Protocols in Avionics” Application Note AN-ION-1-0104
- [6] http://www.interfacebus.com/Design_Connector_LIN_Bus.html
- [7] <http://www.lin-subbus.de/index.php?pid=4&lang=en>
- [8] <http://www.flexray.com/>
- [9] www.eskorea.net/2010/support/data/FlexRay2.0.pdf
- [10] http://en.wikipedia.org/wiki/Media_Oriented_Systems_Transport
- [11] <http://www.telos.de/most/>
- [12] Aviation Standards: Canaerospace, Arinc 661, Arinc 825, Arinc 429, Allied Standards Avionics Architecture Council, ISBN-10: 115569285
- [13] Ákos Demeter “Improving safety of CAN bus systems by redundancy” Master Thesis mtm ws2010/11
- [14] Milos Roland, Veres Bálint: “Hálózatok a járműelektronikában” XXV Jubileumi Kandó Konferencia 2009.