A CAN hálózat alapjai

2009.10.24

Bevezető

A CAN (Controller Area Network) egy nagybiztonságú soros kommunikációs protokoll, adatok valósidejű átvitelének hatékony támogatására. A protokoll kifejlesztésekor a cél egy olyan szabvány kialakítása volt, amely lehetővé teszi az egymással kompatibilis kommunikációs hálózatok kialakítását a következő tulajdonságokkal:

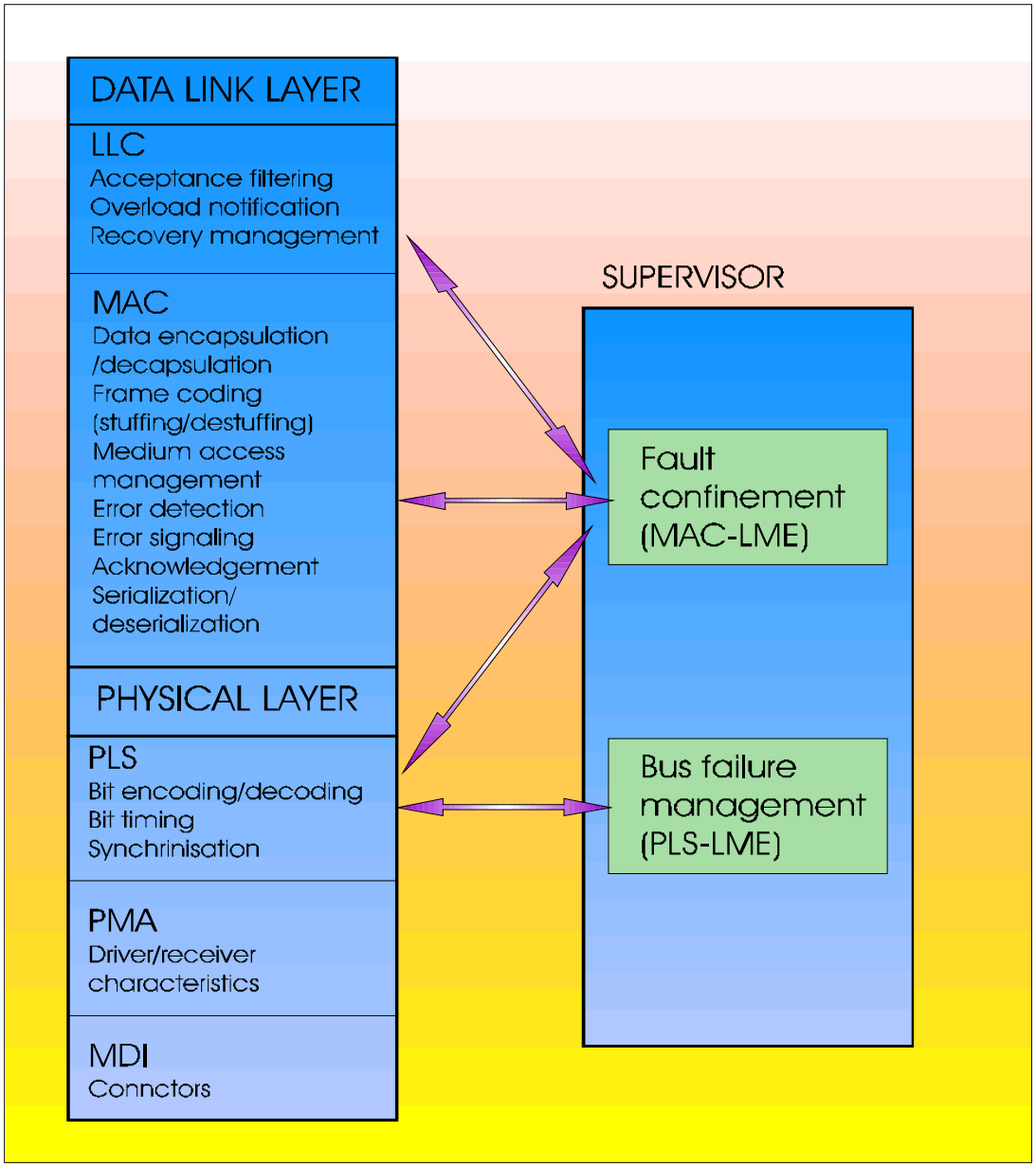
- Elődeinél gyorsabb. A CAN hálózatokban akár 1Mbit/s átviteli sebesség is elérhető
- Biztonságos. Az adatok fix formátumú, különböző, de véges hosszúságú keretekben kerülnek átvitelre. A differenciális busz megvalósításnak köszönhetően a vezetéken zavarokból származó párhuzamos jelek elnyomásra kerülnek, ami növeli az elektromágneses zavarokkal szembeni védettséget. A protokoll különböző hibafelfedő eljárásokat alkalmaz, a hibásnak értékelt kereteket automatikusan újraküldi.
- Flexibilis. A rendszerhez, annak átkonfigurálása nélkül adhatunk további node-okat.
- Eseményvezérelt
- Olcsón megvalósítható. A hálózat összes node-ja (csomópont, a rendszeren belül minden adó vagy vevő) ugyanazon a buszon kommunikál, így bármelyik node elérheti bármely másik node-ot vagy node-okat további linkek kialakítása nélkül.

A CAN az ISO OSI modell alapján rétegekre bontható. A szabvány csak az alsó két réteget definiálja, azok funkciói megegyeznek az OSI rétegek funkcióival.

A Can-en belül mindkét réteget további alrétegek képezik.Az adatkapcsolati rétegen belül két alréteget definiáltak:

- Logic Link Control (LLC)
  Logikai kapcsolatvezérlő
- Medium Acces Control (MAC)
  Átviteli közeg hozzáférés vezérlő
  A fizikai rétegen belül három alréteget definiáltak:
- Physical Signaling (PLS)
  Fizikai jelek képzése
- Physical Medium Attachment (PMA)
  Adó és vevő áramkörök
- Medium dependent Interface (MDI)
  Az átviteli közeg csatlakoztatása.

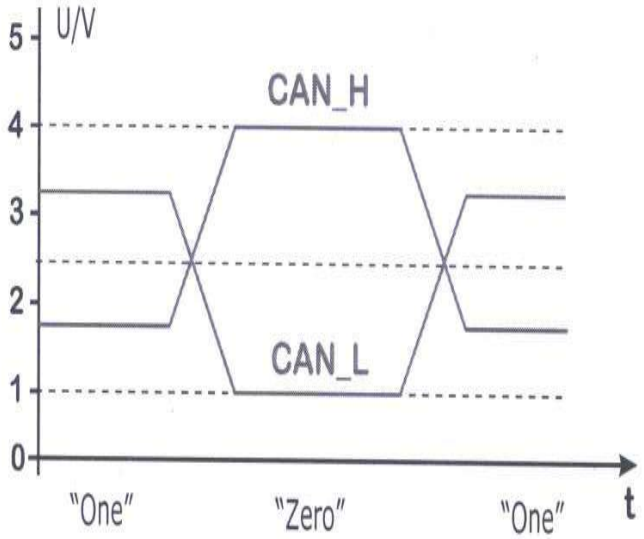
A rétegek vezérlését és azok állapotának felügyeletét a supervisor látja el. Nem képezi egyik réteg részét sem, ez helyett az egyes alrétegekkel van kapcsolatban, azok állapotát felügyeli, vezérli.



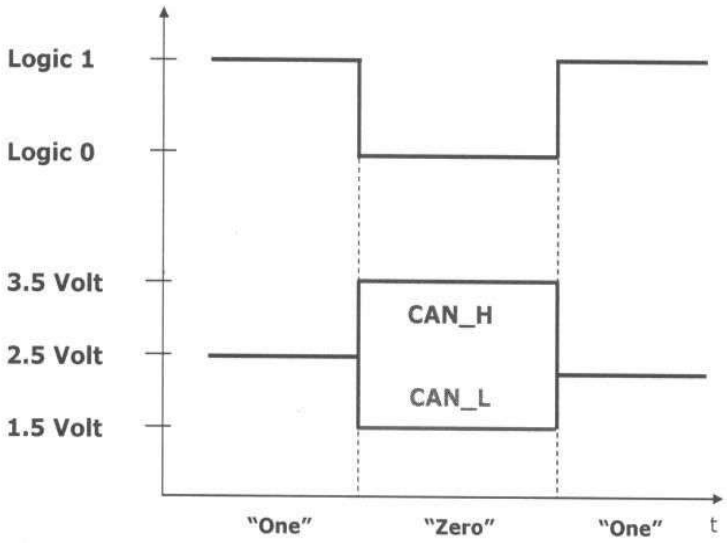
A CAN rétegeinek felosztása

Fizikai réteg:

- Az ISO 11898-2 szabványban definiált nagy sebesség- változat.
- (1 Mbaud sebességig)
- Az ISO 11898-3 szabványban definiált alacsony sebesség- változat.
- (125 Kbaud sebességig)
- Az ISO 11898-1 szabványban fizikai és adatkapcsolati rétegeket definiáltak



Alacsony sebességű CAN



Nagy sebességű CAN

Alacsony sebességű 125Kbaud sebességű szabvány (bal oldalon) és a nagy sebességű szabvány (jobb oldalon) szintjei a fizikai rétegben.

Ahol **a „logikai 0” értékek a domináns**, míg **a „logikai 1” értékek recesszív** szintet jelölik.

A buszmeghajtások egyik bevált módszere a huzalozott-ÉS (aktív magas szinteknél) vagy huzalozott-VAGY (aktív alacsony szinteknél) kapcsolat megvalósítása a csomópontok között nyitott kollektoros vagy nyitott drain meghajtással.

A differenciált CAN busz is ehhez hasonló megoldást nyújt fizikai szinten, de logikai domináns / recesszív szintre fordítás után ugyanezt a logikai műveletet kapjuk. A huzalozott-ÉS logikának jelentős szerepet szántak a keret-azonosítók prioritás rendjének dinamikus vezérlésére (ID arbitráció), továbbá az üzenetek vételi nyugtázására.

Arbitráció

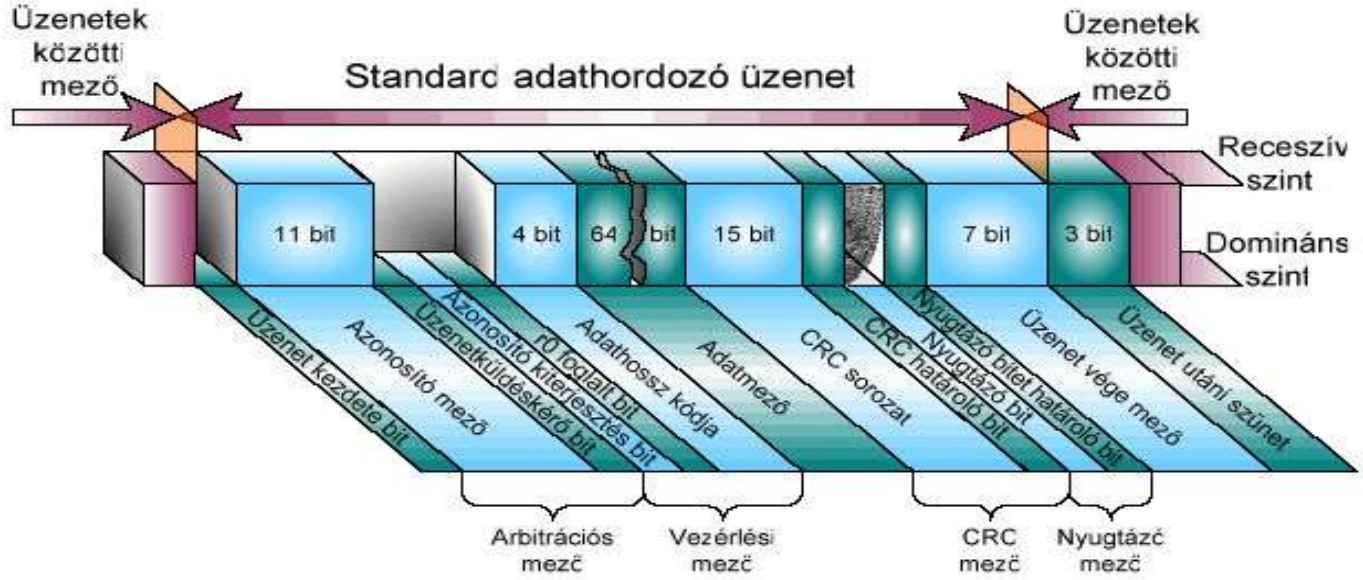
A CAN hálózat keret protokollja CSMA/CA elvű, azaz ütközés elkerülést valósít meg, keret-azonosítók prioritásos figyelésével. Az alacsonyabb sorszámú keretazonosítóval rendelkező üzeneteknek van magasabb prioritása, amit a vétel során a csomópontok a sajátjukkal összehasonlítva, már a töredék bitekkel is eldönthetik, van-e prioritásjoguk adást folytatni

Az adatkapcsolati réteg

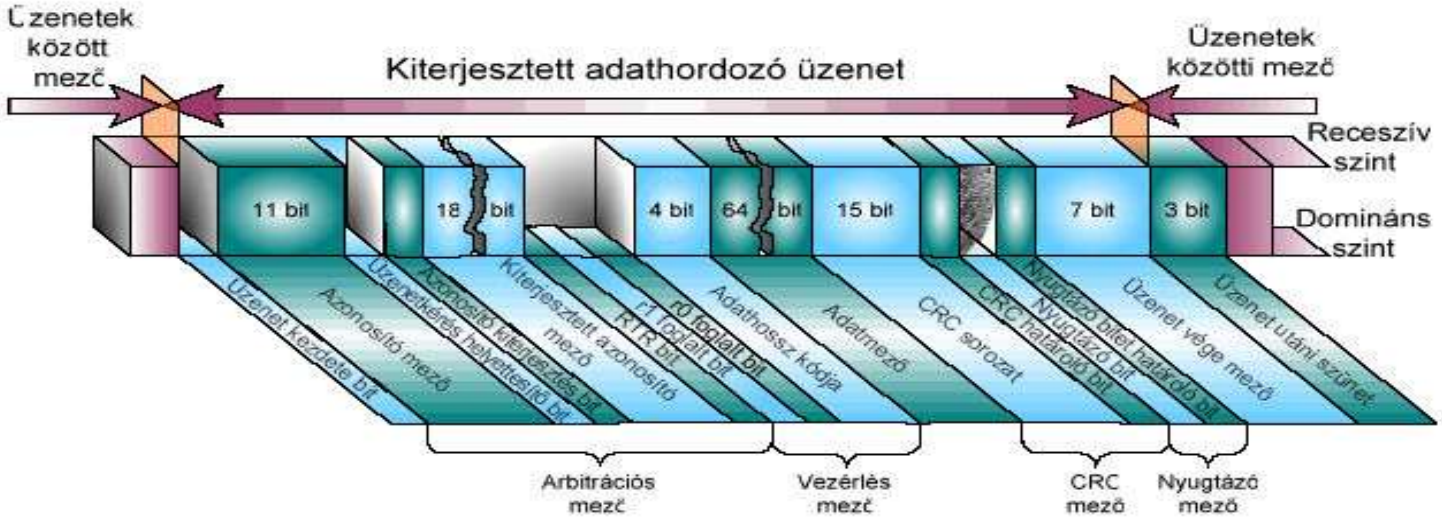
Keretekbe foglalt üzenetekkel valósul meg a kommunikáció a CAN buszon, melyek típusai szerint lehetnek:

- Adat keretek (Data frame)
- Hiba keretek (Error frame)
- Lekérdező keretek (Remote frame)
- Túlterheltséget jelző keretek (Overload frame)

A remote és overload keretek ritkábban használt típusok. A túlterhelés keret csupán az adó oldalnak küldött üzenete a lassú működésű vevőtől. A távlekérdezés (remote frame) kerete nem tartalmaz adatmezőket, csupán keretező mezőket az igényelt azonosítójú üzenet elkéréséhez.



Normál adatkeret 11 bites (ID) azonosító hosszal (CAN2.0A)



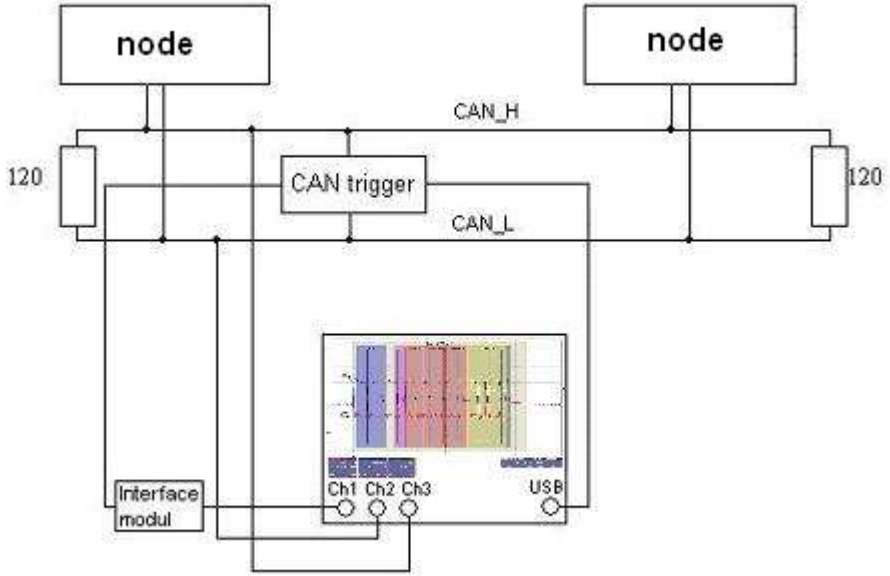
Kiterjesztett adatkeret 29 bites (ID) azonosító hosszal (CAN2.0 B)

A kiterjesztett adatkeret 29 bites ID-je az „azonosító kiterjesztés bit” aktív szintje utáni 18 biten ékelődik a keretbe további két vezérlő bittel. Egy rendszeren belül 32 csomópont lehet szabványos buszmeghajtók esetén, és akár 64-128 darab lehet alkalmazás specifikus meghajtók esetén.

A CAN hálózat minden node-ja névlegesen megegyező bitsebességgel kommunikál. A megengedett sebességbeli eltéréseket a vevők az adó jeléhez szinkronizálva küszöbölik ki. Mivel a CAN NRZ kódolást használ, a pontos szinkronizáció biztosítására csak véges számú megegyező értékű bit követheti egymást. Erre a problémára kínál megoldást a bit beékelés művelete (bit stuffing), amivel maximálható az egymást követő megegyező értékű bitek száma (minden öt azonos bit után beékelődik egy ellentétes logkai szintű stuff bit). A CAN adat és remote keretben start of frame, az arbitrációs mező, a kontroll mező, az adatmező és a CRC mező bit beékeléssel kódoltak. A maradék mezők, CRC határoló, nyugtázási mező, end of frame fix formátumúak, ezért nem kerülnek bitbeszúrásos kiegészítésre, kódolásra.

CAN üzenetek vizsgálata LeCroy oszcilloszkóp segítségével_

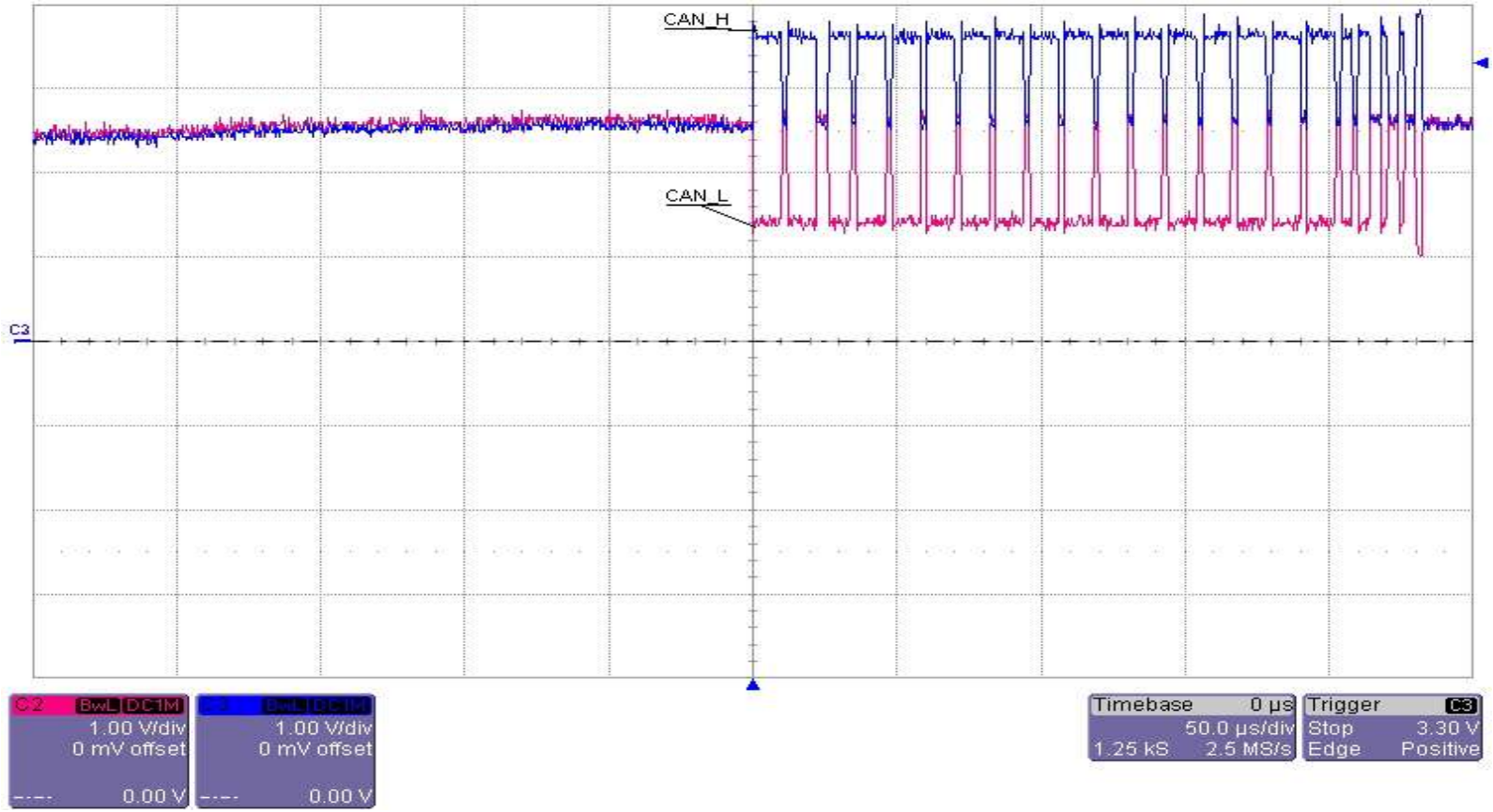
Az üzenetek részletesebb, pontosabb megfejtéséhez CAN Bus TD trigger modul használata volt célszerű. Ez a modul csak a egyes Lecroy oszcilloszkópokkal kompatibilis, jelen összeállításban használt oszcilloszkóp főbb adatai: 4 csatornás, 350Mhz-es sávszélességgel rendelkezik, csatornánként 2.5GS/s maximális mintavételezési sebességű. Az erre csatlakoztatott CAN trigger modul olyan, a Lecroy által fejlesztett eszköz, melynek segítségével könnyedén analizálható a CAN busz kommunikáció a beágyazott CAN kontrollerek segítségével. A modul egy 32 bites, 64Mhz-es mikrokontrollert és két Philips SJA1000 CAN vezérlőt tartalmaz. Tartozik hozzá egy CAN interface modul, amely segítségével rá tudjuk csatlakoztatni az oszcilloszkópra.



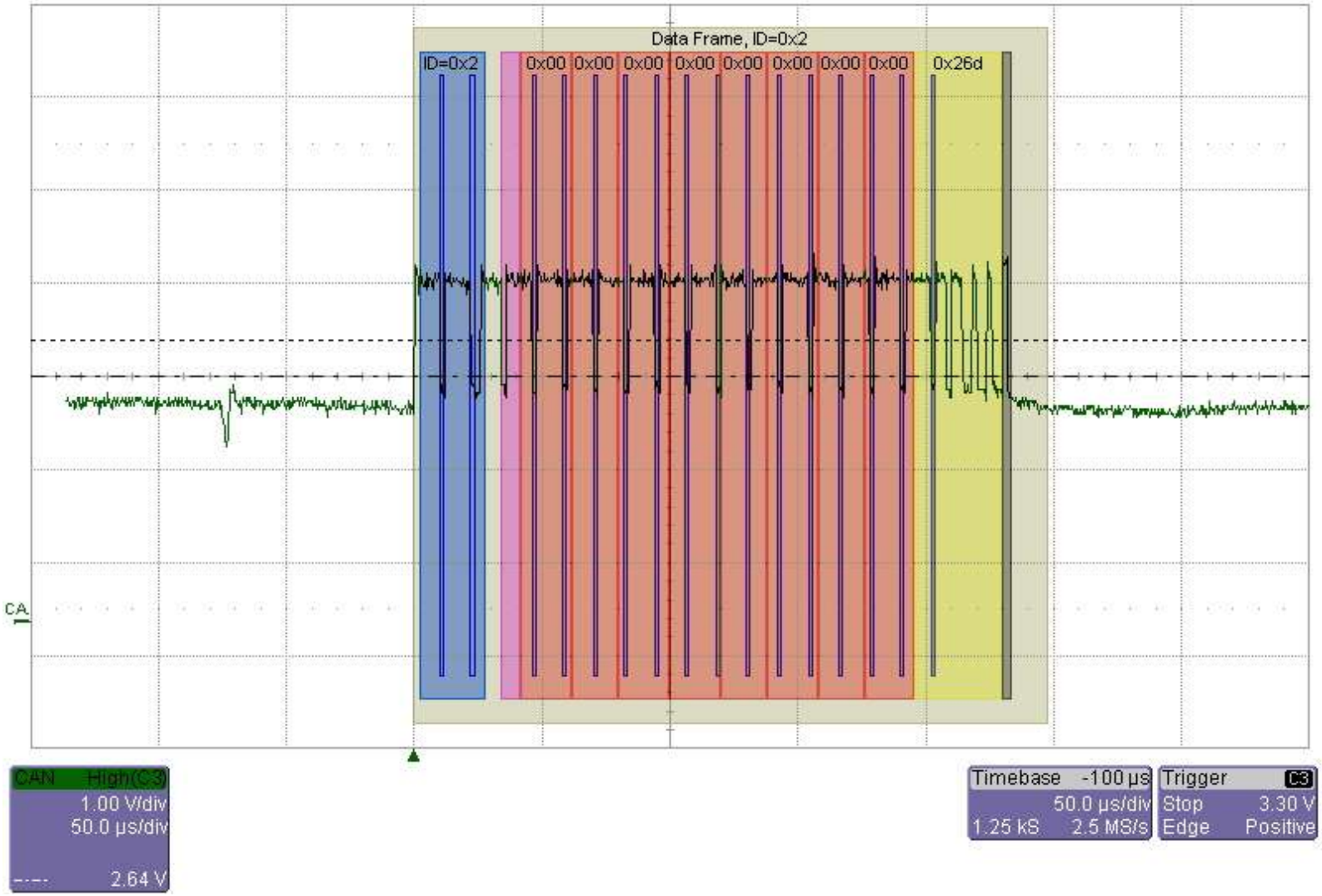
A mérés összeállítása

A „nyers” CAN jel az oszcilloszkópon:

A hármas csatornán (CAN_H) a triggerelést pozitív élre van állítva, a triggerszint pedig 3,3V. **Receszív állapotban mindkét vezeték közelítőleg Ut/2 feszültségű, míg domináns állapotban a CAN_H jelű vezeték Ut felé, a CAN_L vezeték 0V felé mozdul el, növelve ezzel a differenciális feszültséget.** Az oszcilloszkóp beállításai 50μs/div és 1V/div. Az adatátviteli sebesség 500kbit/s, így a bitidő 2μs. További segítséget nyújthat a bitek megfejtésénél, hogy a bit beékelés szabályainak megfelelően nem lehet ötnél több egymást követő megegyező bit, ami 10μs-nak felel meg.

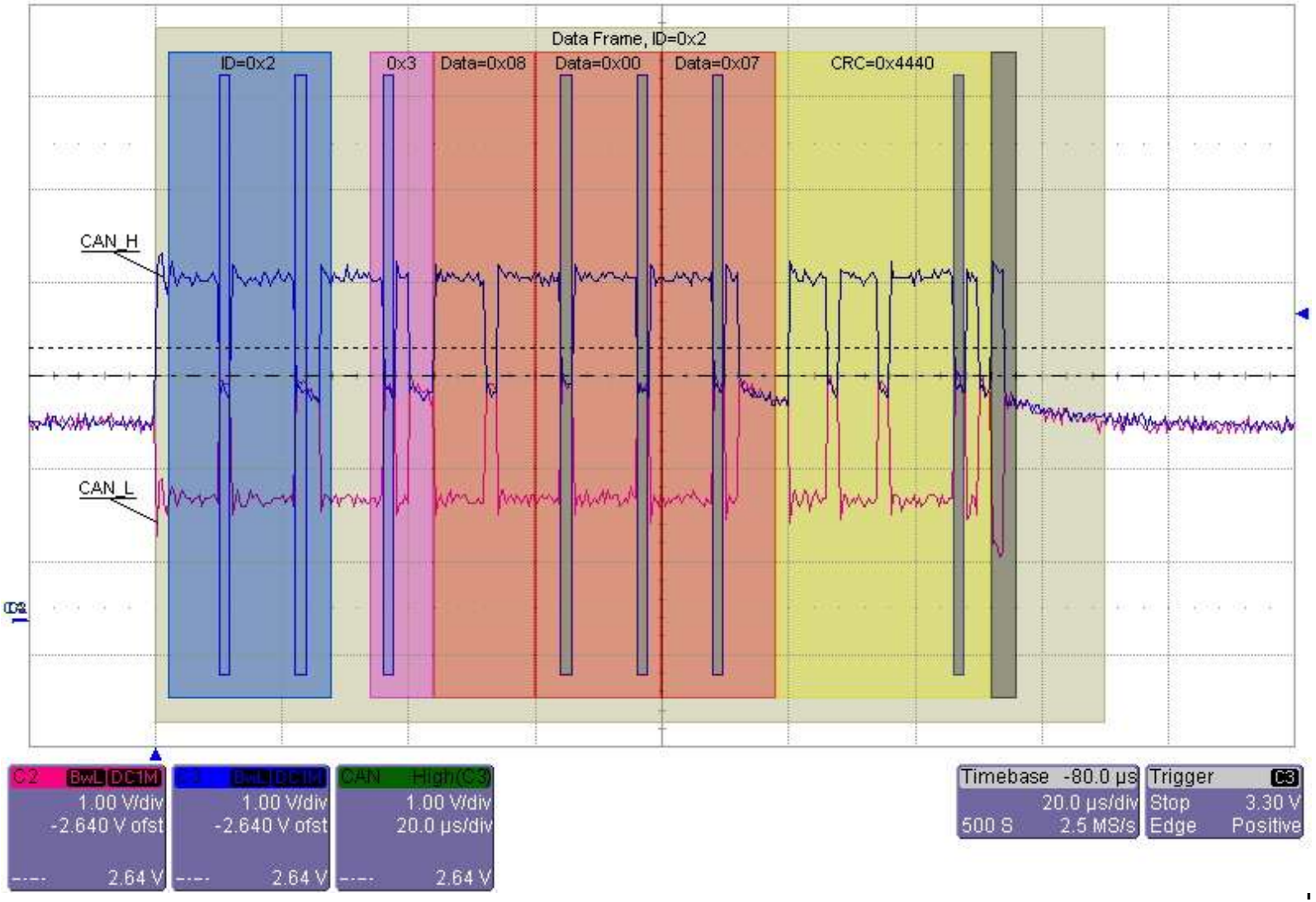


A CAN trigger modul által értelmezett CAN jel oszcilloszkópon:



Látható, hogy a modul könnyebbé teszi az üzenet megfejtését azáltal, hogy dekódolja az üzenetet és különböző blokkokra bontja azt. Ezáltal egyértelművé válik, hogy melyik bit melyik mezőhöz tartozik. (jelenleg csak a szinkron üzenet látható)

Ezt követően először a CAN_L és CAN_H jelet is megjelenítjük, majd a szinkronüzenetben a DLC (az adatmező hossza byte-ban) értékét 3-ra állítjuk, az adatbyte-ok első byte-jának értékét beállítjuk 8-ra, a harmadik byte-ját pedig 7-re és felnagyítjuk a jelet:



Látható, hogy a modul értelmezte a beállított értékeket és ezeket az egyes mezők fölött megjelenítette. Szintén jól láthatóak a stuff-bitek. Mivel az oszcilloszkóp beállítása 20μs/div, a bit idő pedig 2μs, ezért minden egyes bit láthatóvá vált és így könnyedén le lehet ellenőrizni, hogy melyik bit melyik mezőhöz tartozik.

A D domináns bitet, az R receszív bitet, az S a stuff biteket jelöli:

Szürke - Start Of Frame
1 domináns bit.
Kék - Arbitration field
11 bit identifier
jelen példában az értéke 2
00000000010.
1 bit RTR
Az értéke 0, tehát adatkeretről van szó
Szürke - 2 bit fenntartott bitek
Értékük 00
Lila - Kontroll mező
4 bit DLC
értéke 3
000011

Piros - Adatmező 3 byte
0x08 0x00 0x07
000010000000000000000111

Sárga - CRC mező
értéke 0x4440
15 bit
A 15db CRC bitet követi a CRC határoló receszív bit.
Sötétszürke - Egy db domináns ACK bit (a vevőktől érkező nyugtázó bit) és az azt lezáró receszív bit.
Szürke - Hét db. receszív bit képezi a keret végét

A CRC kód:

A CAN rendszeren belül alkalmazott fejlett hibafelfedés részeként az elküldött keretek polinom kódolással előállított CRC kódot tartalmaznak
Az eljárás elve, hogy az adatfolyamot redundáns bitekkel egészítik ki oly módon, hogy a kiegészített jelsorozat által reprezentált polinom maradék nélkül osztható legyen egy a vevő számára is ismert generátor polinommal.
A generátor polinom:

$$X^{15} + X^{14} + X^{10} + X^{8} + X^{7} + X^{4} + X^{3} + 1$$

Ez alapján a vételi oldalon a vett jelsorozat generátor polinommal való osztásával következtethetünk az esetleges bittévesztésekre.

A nem rendszeres hibák helyreállítása a hibás üzenetek automatikus újraküldésével történik, az ismétlődő hibák kiküszöbölése a hibás csomópont (node) kikapcsolásával történik.

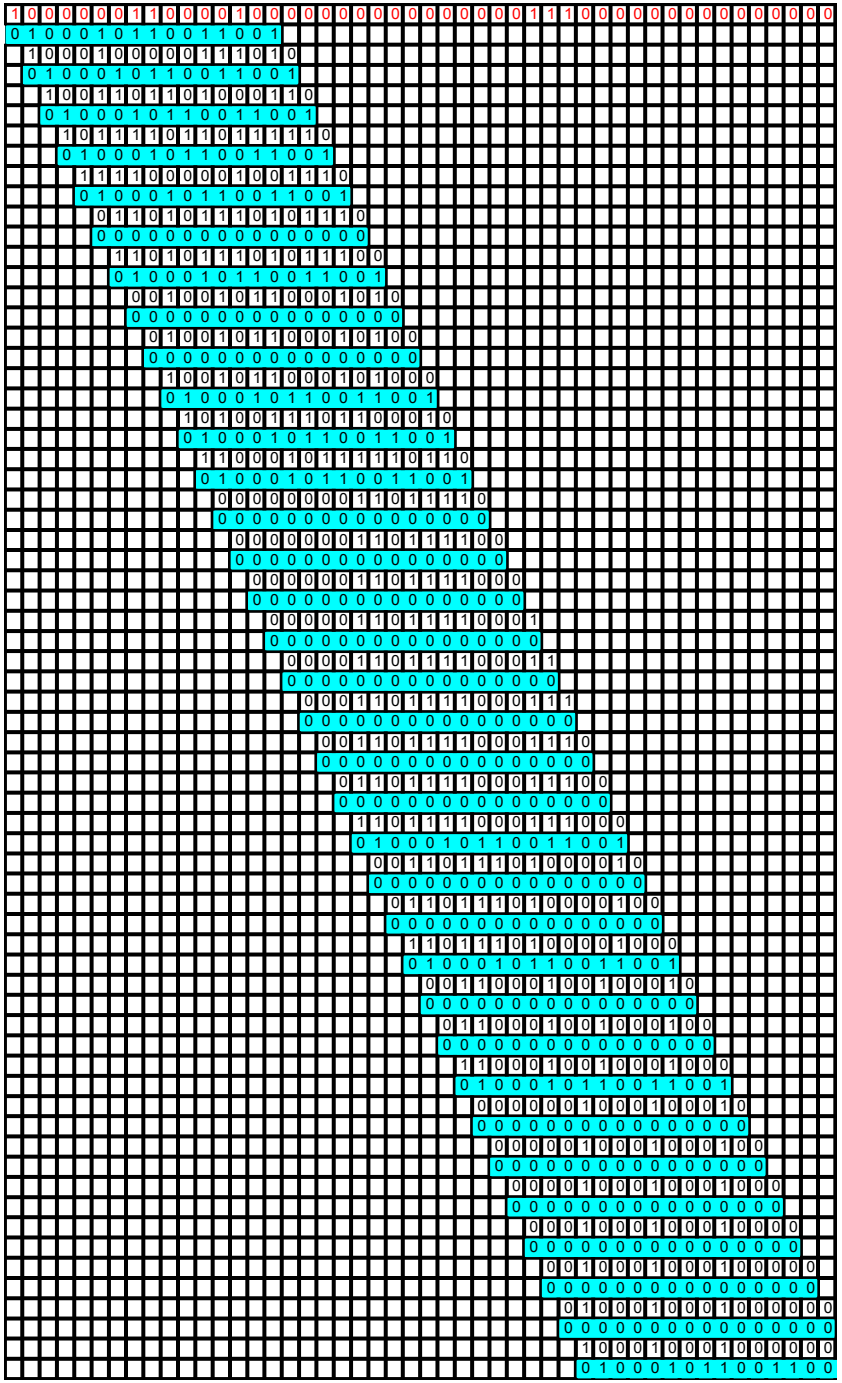
A CRC kódolás hatásfoka:

- Száz százalékban felismeri a 15 bitnél nem hosszabb hibacsomókat.
- Száz százalékban felismer 5 vagy annál kevesebb véletlenszerűen elhelyezkedő hibát.
- Száz százalékban felismeri a páratlan számú meghibásodott bitet tartalmazó üzenetet hibáját.

(rossz minőségű folyamatos átvitel esetén is 1000 évenként feltételez egy felfedezetlen hibát)

Az ellenőrző összeg kiszámítása bonyolultnak tűnhet, de Peterson és Brown bebizonyította, hogy algoritmusok alkalmazása helyett shift regiszterből szerkeszthető egy hardver az ellenőrző összeg kiszámítására. A gyakorlatban a maradékképzést osztás helyett shift regiszter alkalmazásával végzik.

- 15 nullát írunk a bitsorozat mögé.
- beléptetjük az első 14 bitet a shift regiszterbe.
- a sorozatot egy bittel balra shifteljük, és a 15. bit helyére a bitsorozat következő bitjét léptetjük.
- Megvizsgáljuk, hogy a 15. bit a bitsorozat utolsó bitje volt e. Az ellenőrzés eredményétől függően átlépjük a következő lépést, vagy beállítjuk a vége jelzést. Ez a döntés a művelet leállítására szolgál.
- Az első bit vizsgálata. Ha 0, akkor nem osztható a generátorral, átlépjük a következő műveletet.
- Ha az első bit egy volt, modulo 2 összeadással kivonjuk a generátor polinomot a shift regiszter tartalmából úgy, hogy a maradék a shift regiszterben tárolódjon.
- A vége jelzés vizsgálatával ellenőrizzük, hogy szükséges e még folytatni a műveletet. Ha a vége jelzés aktív, a shift regiszter a maradékot tartalmazza.
  Ha a vége jelzés nem aktív a folyamatot a 3. ponttól kell folytatni, míg a bitsorozat végére nem érünk.



Tehát a CRC kód:

100010001000000