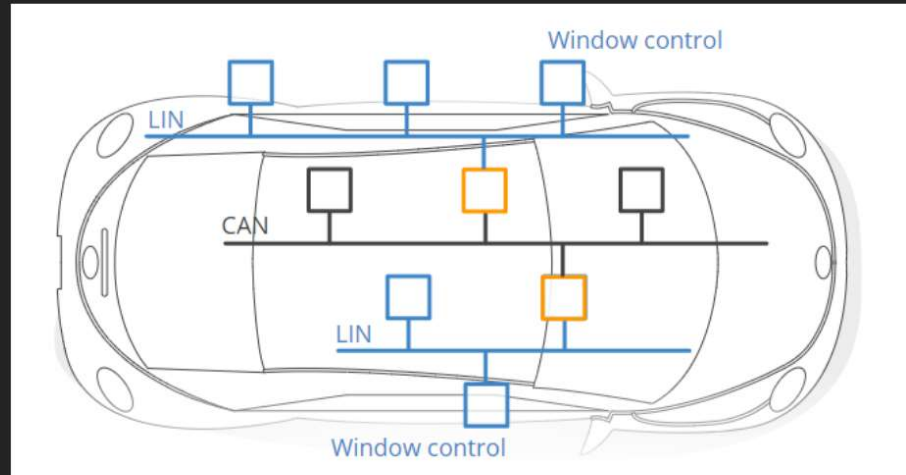


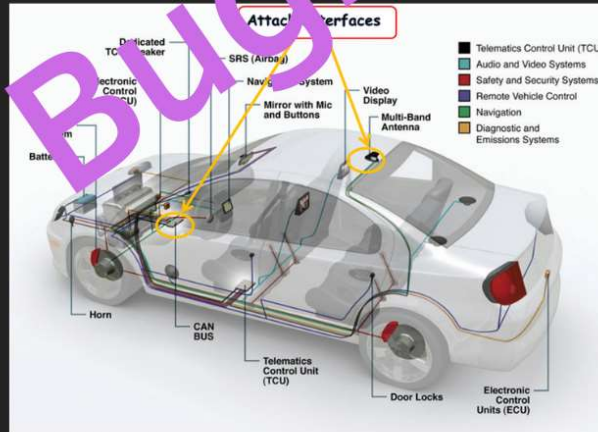
CAN BUS HABERLEŞMESİ VE UYGULAMALARI

Buğra ER



02 CAN BUS

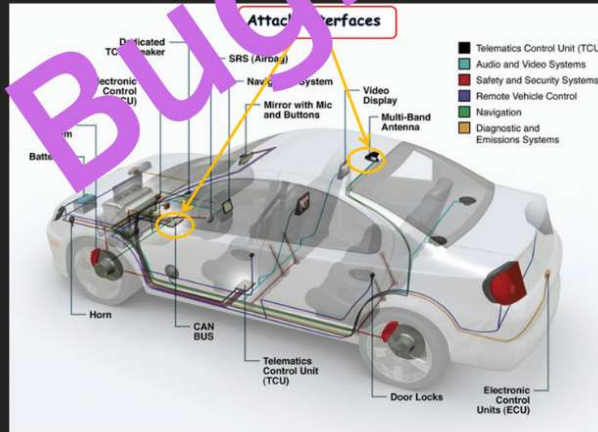
- Açılımı "Control Area Network Bus" yani "Kontrol Alan Ağı Veri Yolu" dur.
- 1980 li yıllarda Robert BOSCH tarafından otomotiv de kablo yumağı yerine bir kablodan yazılım kontrollü veri transferini sağlamak amacıyla geliştirilmiştir.
- Robert Bosch, 1986 yılının Şubat ayında, CAN (Controller Area Network) seri veriyolu sistemini SAE (Society of Automotive Engineers) Detroit kongresinde tanıttı.
- 1987 ortalarında Intel, Philips gibi yarı iletken üreticileri CAN'lerini piyasaya sunmaya başladılar. Böylelikle başta otomobiller olmak üzere her türlü taşıtlardan sanayi ürünlerine kadar birçok alanda CAN en yaygın kullanılan veriyolu protokolüne geldi.



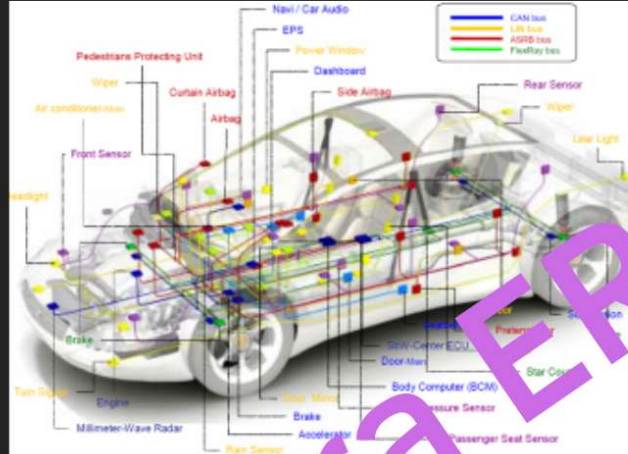
Buğra ER

03 CAN BUS

- Güvenliğin çok önemli olduğu gerçek zamanlı uygulamalarda kullanılır. Öyleki istatistiksel olasılık hesapları sonucunda bir asırda bir tane tespit edilemeyen mesaj hatası yapabileceği hesaplanmıştır.
- Uygulama alanı yüksek hızlı ağlardan düşük maliyetli çoklu kablolu sistemlere kadar genişler.
- CANBUS asansörler, otomobil elektroniği, akıllı motor kontrolü, robot kontrolü, akıllı sensörler, makine kontrol birimleri, kaymayı engelleyici sistemler, trafik sinyalizasyon sistemleri, akıllı binalar ve laboratuvar otomasyonu gibi uygulama alanlarında maksimum 1 Mbit/sn lik bir haber veri iletişimi sağlar.



Buğra ER

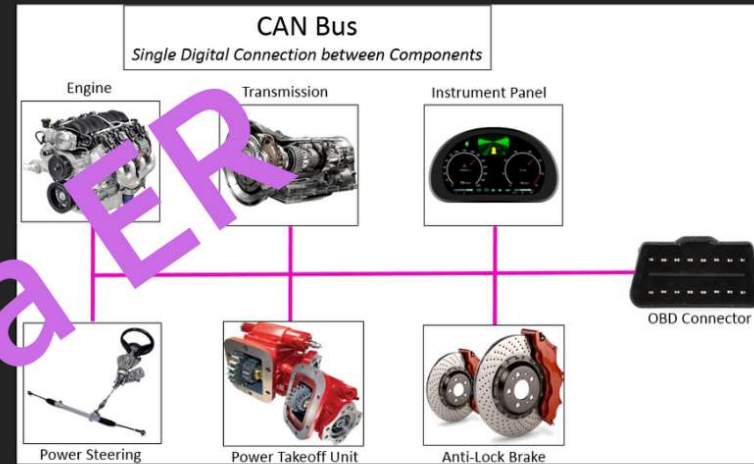
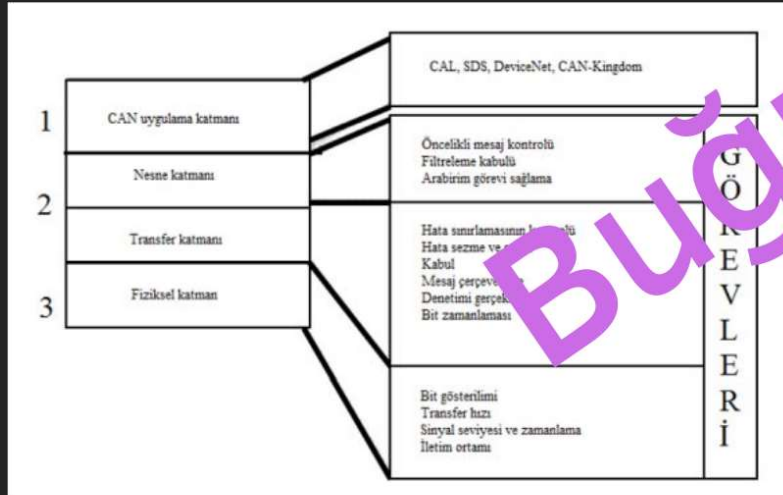


CAN BUS

- İletişim hızı 40m de 1Mbit/sn iken 1km uzatıldıkça 40Kbit/sn ye düşmektedir.
- CAN diğer protokollerden farklı olarak adres temelli değil mesaj temelli çalışmaktadır. Her mesaja özgü bir ID numarası vardır. Mesajlar diğer evler ile iletilirler.

CAN BUS;

- > Nesne Katmanı (Object Layer)
- > İletim Katmanı (Transfer Layer)
- > Fiziksel Katman (Physical Layer)
 - olmak üzere üçe ayrılır.

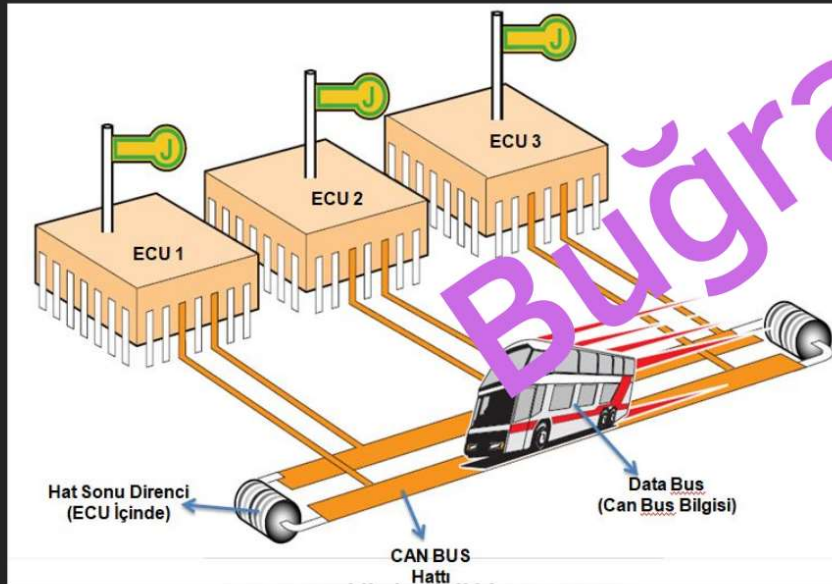


Nesne Katmanının Görevi

- Hangi mesajın transfer edileceğini tespit etmek,
- İletim katmanında hangi mesajın alınacağına karar vermek,
- Donanımla ilgili uygulamaya arayüz sağlamaktır.

İletim Katmanının Görevi

- İletim katmanının başlıca görevi transfer protokolüdür.
- İletim katmanı yeni bir mesajı yollamadan önce iletim hattının boş olmasına dikkat eder. Aynı zamanda iletim hattından veri alınmasından da sorumludur.
- Ayrıca senkron iletişim için veri transferi sırasında bit zamanlamasını bazı parametrelerini göz önünde bulundurur.



Fiziksel Katmanın Görevi

- Fiziksel katman, üniteler arasında veri haberleşmesi sırasındaki tüm elektriksel kısımdır.

CAN BUS SİSTEMİNİN GENEL ÖZELLİKLERİ

- Mesaj Önceliği
- Kayıp Zaman Güvenliği
- Yapılandırma Esnekliği
- Senkronizasyonlu çoklu kabul: Aynı veri birçok ünite tarafından alınabilir.
- Sistemdeki veri yoğunluğunu kaldırabilme
- Çok efendili (Multi master) çalışma
- Hata tespiti ve hataya ilişkin sinyaller üretme
- Mesaj yollanmasında hata oluşması halinde mesajın iletim hattının (BUS) boş olduğu bir anda mesajın otomatik olarak tekrar yollanması
- Ünitelerde oluşan geçici ve kalıcı hataları ayırt edebilme ve özerk olarak kalıcı hatalı üniteleri kapatabilme

- **CAN sistemlerinde veriler paketler halinde iletilir. Burada iki tip paketleme yapılır ve bu paketlemelerin özel adları vardır.**
 - > **11 bit tanımlayıcıya sahip olanlar CAN2.0A diğer adıyla STANDART CAN, 29 bit tanımlayıcıya sahip olanlara ise CAN2.0B diğer adıyla EXTENDED (Geliştirilmiş) CAN denir. Aralarındaki temel fark ise tanımlanacak mesaj sayısıdır.**
 - **Standart CAN'de $2^{11} = 2048$ mesaj tanımlanır.**
 - **Extended CAN'de $2^{29} = 536.870.912$ mesaj tanımlanır.**
 - > **Bu bilgilerin tutulduğu alana mesaj ID'si denir. Mesaj önceliğini tanımlamada buradaki sayı dikkate alınır.**
 - **Veri yolunda mesajlar çerçevelere bölünerek iletilmektedir. İki farklı mesaj türü vardır.**
 - > **Bunlar veri çerçevesi (Message Frame) ve istek çerçevesi (Remote Transmit Request Frame).**
- Farkları ise veri çerçevesinin en fazla 8 byte uzunluğa kadar veri taşıyabilmesi, istek çerçevesinde ise belli bir mesaja ait verinin istenmesidir.**

BASE FRAME (TEMEL ÇERÇEVE)

- Her çerçeve SOF (Start Of Frame) sinyali ile başlar.
- > Bu sinyal 1 bitliktir dominanttır.
- Bunun ardından 12 bitlik denetim alanı gelmektedir.
- > İlk 11 biti mesaj ID alanıdır ve bu alandaki ID değeri ile mesajlar etiketlenir.
- > Denetim alanındaki son bit RTR diye adlandırılır ve özel anlamı vardır.
- Bu bit 0 (dominant) ise gönderilen çerçeve veri çerçevesidir ve veri alanında, ID alanında tanımlanan mesaja ait veri vardır.
- Bu bit 1 ise çerçeve istek çerçevesidir ve veri alanı yoktur.
- Bu çerçevenin ID alanındaki değer ile belirlenen mesaja ait veri ilgili düğümlerden istenmektedir.
- ID alanındaki ilk 7 bit ardışıl olarak res. sıralanmaz.





Denetim Alanı

- Denetim alanından sonra kontrol alanı gelmektedir.
- Bu alanın ilk bitini 0F diye isimlendirilir ve bu çerçevenin 11 bitlik ID alanına sahip 2.0A çerçevesi olduğunu belirten dominant bir bittir.
- Bu bitin ardından bir bitlik kullanılmayan rezerve alan gelmektedir.
- .Daha sonra 4 bitlik DLC diye isimlendirilen bir alan gelir.
- > DLC alanı gönderilen verinin kaç byte olduğunu söyler.

Kontrol Alanı

- Kontrol alanını veri alanı takip etmektedir.
- Veri alanı en fazla 8byte olabilmektedir.

Buğra ER



Veri Alanı

- Veri alanını CRC alanı takip eder.
- Bu alan 16 bitliktir ve 15 bitlik CRC (Cyclic Redundancy Check) bilgisi ile resesif CRC Delimiter bitinden oluşmaktadır.
- CRC alanı gönderilen SOF alanından CRC alanına kadar gönderilen verinin doğru olup olmadığının anlaşılması için bir değerdir.
- Veriyi gönderen düğüm veri üzerinde bir takım işlemler yaparak 15 bitlik CRC değerini hesaplar ve çerçeveye ekler.
- Alan düğüm veriyi aldığı anda gönderenin yaptığı işlemler ile aynı işlemleri yapar ve CRC yi tekrar hesaplar.
- Alınan ve gönderilen CRC bitleri eşitse veri doğru gönderilmiştir.
- Alıcı düğümlerden en az 1 tane bile veriyi yanlış aldıysa veri tekrar gönderilmelidir.

Buğra ER

CRC Alanı

- CRC alanını ACK alanı takip eder.
- Bu alan 2 bitlidir.
- > İlk bitini gönderici resesif olarak gönderir.
- > Eğer veri en az bir alıcı tarafından doğru alınmışsa alıcı yola dominant biti yazar.
- > Böylece gönderici mesajın en az bir alıcı tarafından alındığını anlar.
- > Eğer gönderici dominant biti okuyamassa ACK işleminde kaynaklı bir hata olduğuna kanaat getirir ve veriyi tekrar yollar.
- > Bu alanın ikinci bitini ise ACK delimiter olarak kullanılır ve resesiftir.
- > Eğer gönderen İstek Çerçevesi yollamışsa, alıcı da iletim hattının boş bir anında cevabını göndericiye yollar.



ACK Alanı

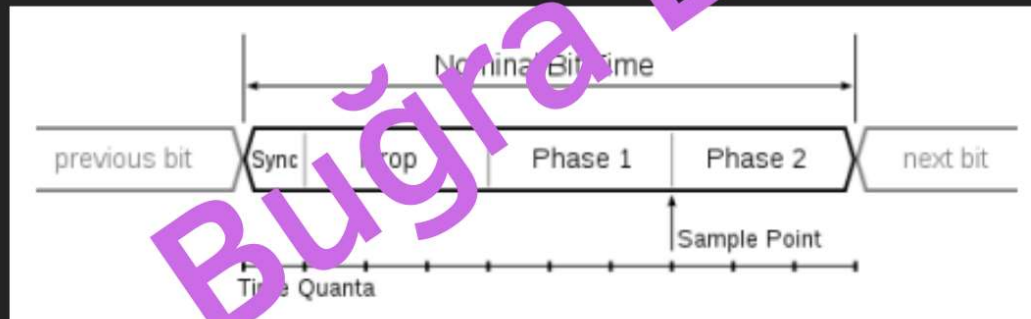
- **ACK Alanının arkasından çerçevenin sonlandırıldığına belirten 7 bitlik EOF alanı gelir.**
- **Bu alandaki bitler resesiftir.**
- **Daha sonra ise çerçeveler arasında boşluk bırakmak amacıyla 3 bitlik INT alanı gelmektedir ve bitleri resesiftir.**
- **Böylece temel çerçevede bir mesaj gönderilmiş ve alınmış olur.**



Buğra ER

BIT TIME BİT SÜRESİ

CAN protokolünde bit hızı diğer seri protokolden farklı olarak direk olarak baud rate önbölücüsünü kurarak ayarlanmaz.



CAN BUS HATA ÖNLEME

- CAN protokolünün beş adet hata önleme yöntemi vardır.
- Herhangi bir hata oluştuğunda gönderici veriyi tekrar gönderir böylece işlemcinin olaya müdahale etmesine gerek kalmaz.
- Hata önleme yöntemlerinden 3 tanesi çerçeve düzeyinde 2 tanesi ise bit düzeyindedir.
- Çerçeve düzeyinde olanlar çerçeve formatı kontrolü CRC kontrolü ve ACK (onay) kontrolü dür.
- Bit düzeyinde olanlar ise bit kontrolü ve bit stuffing kontrolüdür.

- CAN çerçeve formatında birden fazla alan olduğunu ve roller arasındaki ilişkilerini anlattık.
- Alıcı veriyi aldıktan sonra verinin formatını kontrol eder ve çerçeve yapısı ile uyumlu olup olmadığını karşılaştırır.
- Alınan veride eksik alan varsa veri reddedilir ve veriyi yoluna hata çerçevesi bırakılır.
- Bu sistem doğru formatta veri alımını sağlar.
- Çerçeve düzeyindeki ikinci hata önleme yöntemi ise CRC kontrolüdür.

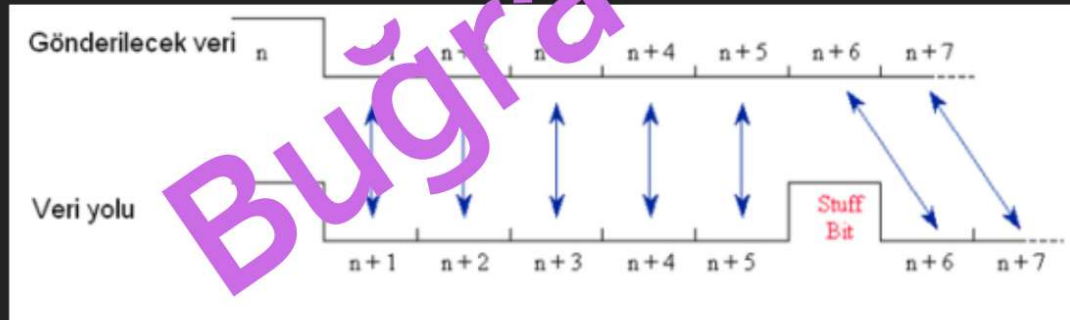
- **SOF bitinden CRC bitlerinin başına kadar olan bitler bir takım işlemlerden geçirilerek CRC kodu üretilir.**
- **Alıcıda bu CRC kodu alınan veri ile karşılaştırılır ve alınan bitlerin doğru olup olmadığı kontrol edilir.**
- **Format kontrolünden sonra alıcının bitleri kontrol etmesi ile formata uyan fakat hatalı mesajların önüne geçilmiş olur.**
- **Çerçeve düzeyindeki son hata kontrolü ise ACK mesajının göndericiye ulaşmamasıdır.**
- **ACK bitinin anlamı alınan mesajı alıcının onaylamasıdır.**

Buğra ER

- Gönderici CRC bitlerini gönderdikten sonra ACK bitini resesif olarak gönderir.
- Alıcılardan en az bir tanesinin hattaki resesif olan ACK bitini dominant bitle ezmesi beklenir.
- Eğer zaman aşımı sonucu ACK biti göndericiye ulaşmamışsa ACK bitinde hata olduğu şeklinde yorumlanır.
- Bunun sonucu göndericide hata oluşur ve ACK onayını alıncaya kadar aynı mesajı tekrar gönderir.

Buğra ER

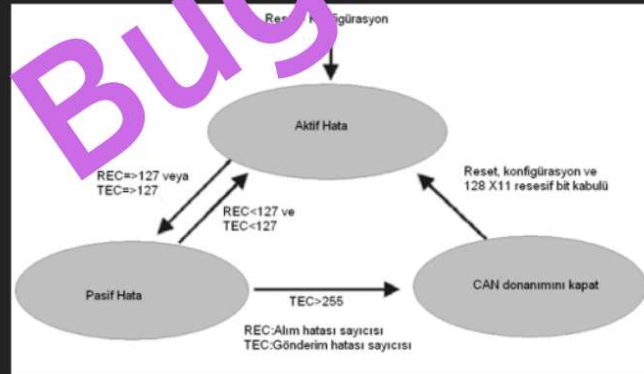
- Bit düzeyindeki hatalardan ilki bit stuffing hatasıdır.
- Gönderici ile alıcı arasında saat darbeleri gönderilmez.
- Bunun yerine veri yolundaki CANL ve CANH hatlarındaki lojik değişimler ile senkronizasyon sağlanır.
- Bunun sonucu olarak aynı lojik seviyeden (dominant) 5 ten fazla bitin ard arda gelmesi senkronizasyonun bozulduğu anlamına gelir ve alıcıda hataya sebebiyet verir.
- Bunu engellemek için gönderici aynı 5 seviyeden sonra 1 arş seviyeden bir bit göndererek iletişime devam eder.
- Bu olay aşağıdaki şekilde gösterilmiştir.



Buğra ER

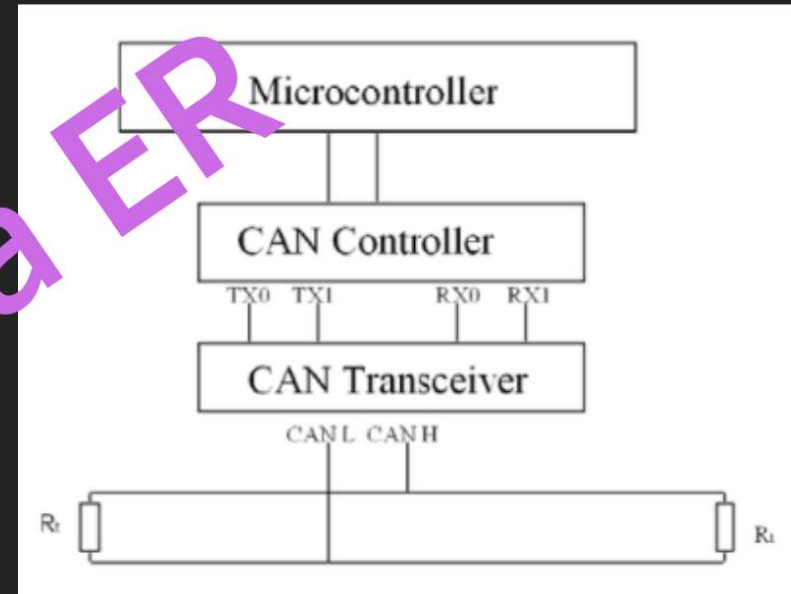
- Sonuç olarak herhangi bir düğüm herhangi bir anda hata mesajı oluşturmak istediğinde veri yoluna 6 adet dominant bit yazar ve hataya sebebiyet verir.
- Bit düzeyindeki diğer hata ise bit kontrolüdür.

- Veri yolu boşaldığında düğümlerin mesaj göndermek için veri yolunu mesajların ID lerine göre ele geçirdiğini söylemiştik.
- Her düğüm veri yoluna yazdığı biti tekrar geri okuyarak kendisinin gönderdiğinden daha önemli bir mesaj varmı diye bakar.
- Eğer daha önemli mesaj varsa geri çekilerek veri yolunun boşalmasını bekler.
- Böylece veri yolunu bir düğüm kazanmış olur ve bitleri göndermeye devam eder.
- Aynı zamanda gönderdiği bitleri geri okur.
- Eğer veri yolunu kazanan düğüm gönderdiği seviyeden farklı bir seviye okursa hata oluşturur.

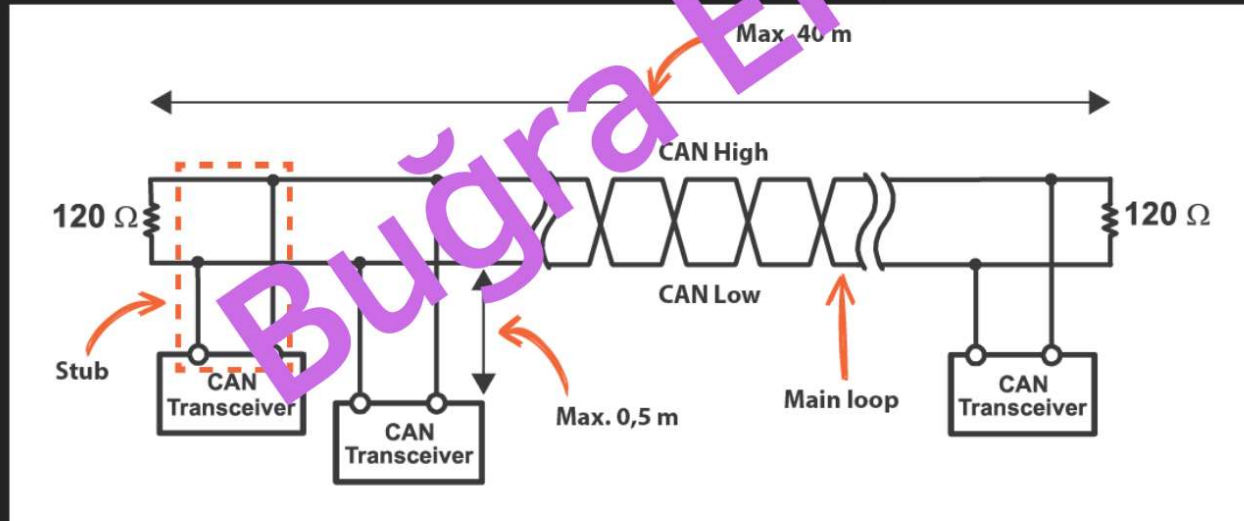


- CAN yapısının temelinde her mesajın farklı bir ID ile etiketlendiğini söylemiştik.
- Mesajları etiketlerken bu kurala dikkatle uymamız gerekmektedir.
- Yoksa farklı düğümlerin aynı ID ye ait farklı veriler göndermesi hata CRC hatasına sebep olacaktır.
- CAN donanımı oluşan hatalara göre hata durumları arasında geçiş yapmaktadır.
- İki adet hata sayıcısı vardır.
- Bunlar göndericide oluşturulan hataları ve alınan hataları sayarlar.
- Herhangi bir sayıcı 127 ve büyük bir değere ulaşırsa donanım pasif hata moduna girer.
- Bu modda gelen hata çerçevelerini cevaplamaya devam eder fakat hata oluşturduğunda dominant bitler yerine resesif bitler gönderir.
- Eğer gönderim hata sayısı 255 i geçerse donanım kapatılır ve hattaki iletişime karışmaz ve etkilenmez.
- Haberleşmeyi tekrar başlatmak için işlemci bir olaya müdahale etmesi gerekmektedir.
- CAN donanımını resetler ve tekrar konfigüre eder.
- Bu mekanizmalar düğüm hata durumuna düştüğünde ard arda hata mesajları göndererek veri yolunu meşgul etmesini önlemek içindir.

Bağlantı noktaları düğüm olarak isimlendirilirler ve mikrodenetleyici,CAN kontrolcüsünden oluşmaktadırlar.Bizim işlemcilerimizde bu CAN kontrolcüsü dahil edilmiştir fakat harici entegre olarak kullanılabilir.



CAN kontrolcüsü CAN veri yoluna direk bağlanır. Bu veri yolu iki telden oluşmuş, iki tarafı 120 ohm dirençlerle sonlandırılmış bir veri yoludur.



CAN BUS ENTEGRELERİ

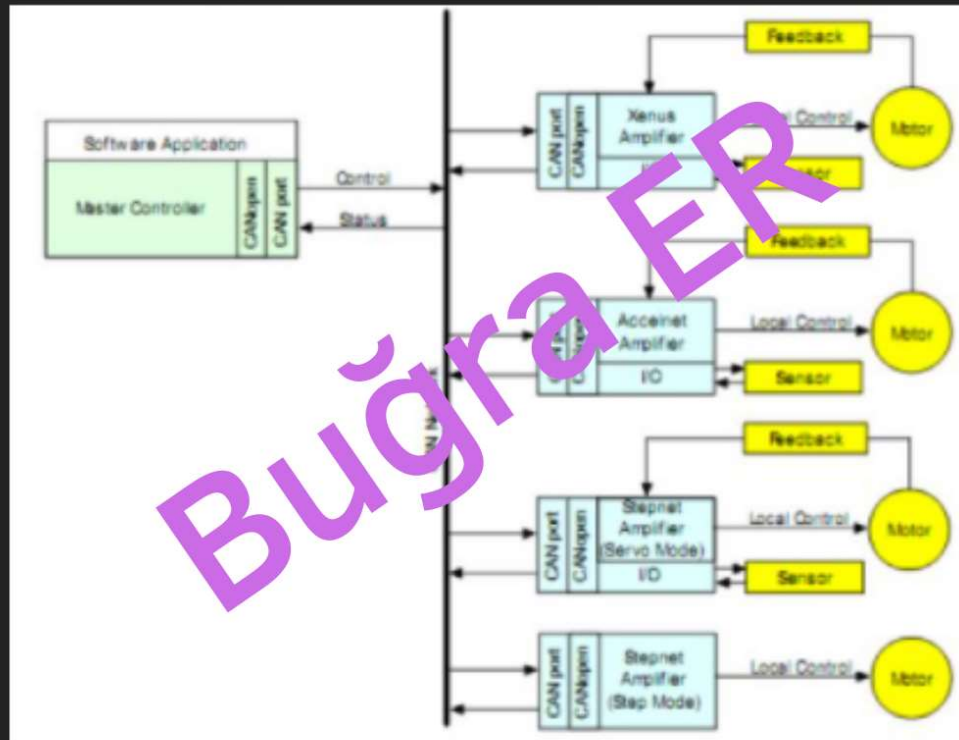
MCP2551

SN65HVD230

MCP2515



Buğra ER



Kablo seçimi

ISO 11898'e göre CAN-BUS. CAN-BUS kabloları (Controller Area Network (CAN)) hızlı dijital bilgi aktarımı ve alış verişi için kullanılmaktadır. Besleme zincirleri ve yüksek esneklik gerektiren uygulamalar için uygundur.



Buğra ER

Elektrik Elektronik Mühendisi
Arge Mühendisi

Dinlediğiniz için teşekkür ederim