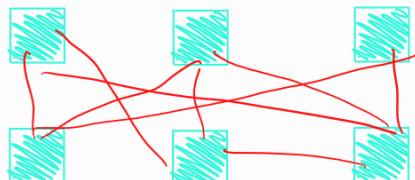
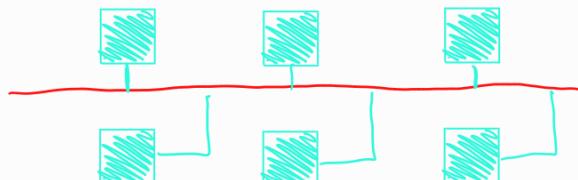


# CAN (Controlled Area Network) :

- ★ Bir otomobili insan vücutuna benzetirsek CAN Bus otomobilin sinir sistemidir.
- ★ CAN bus aracılığıyla otomobilin bünyesindeki farklı sistemler birbirleriyle iletişim kurabilir. (Sistemler  $\Rightarrow$  Electronic Control units  $\Rightarrow$  ECU  $\Rightarrow$  CAN nodes)
- ★ ECU örnekleri : Motor Kontrol Ünitesi, hava yastıkları, ses sistemi vb
- ★ CAN Bus System gibi bir sey olmasaydı ECU'lar arası iletisim, ancak kaotik bir kablolama sistemiyle gerçekleştirilebilirdi.



VS



Random Cable Connections

CAN Bus System

- ★ CAN Bus Sisteminde iki çeşit sinyal hattı kullanılır:
  - CAN High  $\Rightarrow$  yüksek
  - CAN Low  $\Rightarrow$  düşük $\Rightarrow$  Voltaj sinyali taşıyan hattır.

## Diferansiyel Sinyalizasyon :

★ Yüksek ve alçak voltajlar arasındaki farklı kullanarak veri iletimi sağlanır. Bu yöntem elektromanyetik girişimlere ve gürültüye daha dayanıklıdır, çünkü iki hat aynı anda bir sinyale maruz kalırsa aralarındaki fark değişmez.

★ Uzun kablo mesafelerinde bile sinyal bütünlüğünü korunur.



- ★ CAN Bus'ı OSI Layer'in ilk iki katmanı - Physical  $\rightarrow$  Data Layer - olarak nitelendirilebiliriz

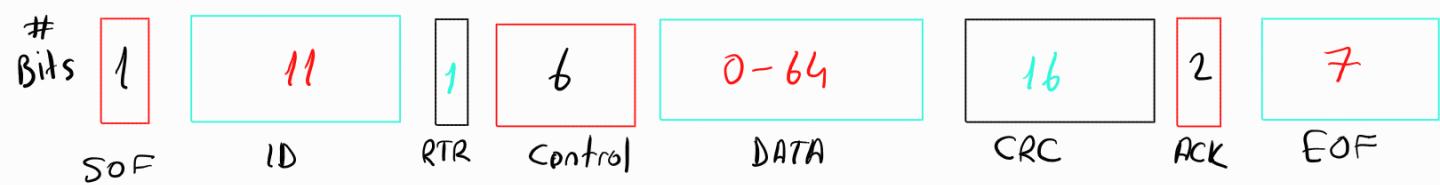


- ★ Neden CAN Bus :

1) Uygun hız  
2) Uygun價格  
3) Uygun güvenilirlik  
4) Uygun teknoloji

- 1) Basit ve Düşük maliyetli
- 2) Tamamen Merkezileştirilmiş
- 3) Son Derece Güçlü
- 4) Verimli Çalışıyor.

## Standart CAN Frame



SOF: Mesajın başlangıcı belirten alandır. "Dominant 0" ile bir düğümün diğer düğümlere veri göndermemeyi planladığını belirtir.

ID: İşlemi önceligiini belirler (niceness mantığı)

RTR: Düğüm veri talep ettiyse 1, veri gönderdiyse 0'dır.

Control: İki görevi vardır. Id bitinin 11 veya 29 bit olmasını sağlar (0-1) ve gönderilen paketin veri bytelarının sayısını belirler.

DATA: CAN sinyallerini yanı veriyorum içeriş.

CRC: Veri bütünlüğünü - hataları ayıklama

ACK: Veriler doğru bir şekilde alındı (elde edildi) bilgisini içeriş.

EOF: CAN Frame'in sonunu belirtir,

## CAN'IN ÖRNEK KULLANIM SENARYOLARI

★ Araçtan veri almak, tahmini arıza tespiti, blackbox

## CAN Bus Data Decoding :

\* Hemen hemen her araç kendi şifreleme metodıyla CAN frame'ler oluşturur ve kendi decoderleriyle CAN帧eleri anlamlı hale dönüştürür.

★ CAN Bus'ın sağladığı temel iletişim altyapısını kullanarak belirli alanlara spesifik olarak oluşturulan protokoller var. PDF'in devamında bu protokoller not alacağım.

## CAN Bus's Sub Protocols :

J1939: Ağır iş araçları için geliştirilmiş bir protokoldür.

OBD2: Araç təşhis ve raporlama protokolüdür.

CANopen: Özellikle endüstriyel otomasyon ve kontrol uygulamalarında kullanılır.

olsa da hemen hemen her seferde kullanılır bir protokoldür.

**CAN FD:** Bir bakıma CAN 2.0 yani CAN'ın daha hızlı ve daha yüksek kapasitede veri taşıyan versiyonu denebilir.

### J1939:

- ★ SAE J1939 aslında yukarıda bahsettiğim gibi ağır iş makinelerinde ECU'ların nasıl iletişim kuracagini belirleyen standartlar grubudur.
- ★ CAN'den farklı olarak ID 11 bitten değil Can FD gibi 29 bitten oluşur,
- ★ J1939 özel veri formattları (motor devri, hız gibi) ve daha spesifik bir syntax ile ECU verilerini iletilerken, CAN ile aynı görev görülecek olsa elimizde daha genel bilgiler olur,
- ★ Belirli ECU'lar belli verileri sadece request doğrultusunda da alabilir. (on request)
- ★ SPN ve PGN parametreleri mevcuttur.
  - ↳ **SPN (Suspect Parameter Number):** Aracın bireysel parametrelerini tanımlamak için kullanılır. (motor devri ve hız gibi)
  - ↳ **PGN (Parameter Group Number):** Parametre gruplarını tanımlar ve belirli bir veri setinin içeriğini organize eder (Mesajın içeriği hakkında bilgi verir)
- ★ Birden fazla byte içeren değişkenler intel byte order yani en az anlamlı byte önce kurallı ile iletiler. Bu şekilde büyük veri blokları doğru bir şekilde iletiler. Bu özellik CAN'da mevcut değildir.
- ★ J1939 PGN'ler aracılığıyla 1785 byte' a kadar olan veri paketlerini destekleyebilir. Bu büyük veri kümelerinin parçalara bölünerek ardışık olarak iletilmesini sağlar. CAN Protokolü genellikle tek mesajla sınırlıdır ve bu tür büyük veri iletimleri için optimize edilmemiştir

## TGN vs CAN ID:

29 bit

CAN ID (extended)

bit start 9  
bit length 18

J1939 PGN

- \* CAN ID mesajın öncelliğini ve içeriğini belirlerken PGN parametre gruplarını tanımlar
- \* CAN ID to PGN converter gibi uygulamalar mevcut. 29 bitlik CAN ID'nin 8 bitlik Message ID'si belirli bir parametre grubunu gösterebilir.
- ☆ Bir aracın J1939 verilerine ulaşabilmek için CAN Logger'a 9-pin deutsch connector ile bağlanılabilir.



## OBD2:

- ☆ Kisaca aracın kendi sorunlarını tespit etme sistemiidir.
- ☆ Aracınızda gördüğünüz motor arıza lambası bu sistemin bir ürünüdür. yetkili servise gittiğinizde OBD2 Scanner kullanıp hata arama işlemini başlatabilir. (Diagnostic Trouble Codes - DTC)
- ☆ OBD2 16 pin connector kullanılır.
- ☆ OBD2 connector direksiyonun yakınında ama kapak ve panellerin arkasında gizli olabilir.
- ☆ 16 pin arac kapatılyken bile veri aktarımı yapabilir.
- ☆ OBD2 pin düzeni kullanılan protokole göre değişebilir. (En yaygın protokol ISO 15765'tir ve bu yüzden pin6 (CAN-H) ve pin 14 (CAN-L) bağlanır.)

## OBD Connector - type A vs type B:

- ☆ Type A arabalarda bulunurken type B ağır iş makinelерinde yaygındır.
- ☆ Type A - 12V , Type B - 24V

★ Type B kablosu her Tyre A soketine olur ama Type A kablosu hiçbir type B soketine uyumaz

★ OBD2 2008 sonrasında çikan neredeyse tüm otomobilin kullanıldığı bir standarttır. Belirli yillarda ise belirli bölgelerde direkt zorunluluk haline gelmiştir.

★ Güncel elektrikli arabalar OBD2'yi desteklememektedir çünkü arabalar daha zayıf sistemlere sahip olduğunda bu teknoloji yetersiz kalmaktadır. OBD3 ile bu durum çözülmektedir.

11 bit	# bytes	Mode	pID	A	B	C	D	unused
--------	---------	------	-----	---	---	---	---	--------

ID

★ ID'de requestler "7DF" ile başlarken response mesajları "7E8-7EF" arası değişir. Genelde 7E8 ile başlar.

★ Mode kısmı "01-0A" arası değişir. OBD2'da 0 yerine 4 olur. DTC, show current data gibi işlevleri mevcuttur.

★ Her moda spesifik pID'ler vardır. "0D" aracın hızını göstermektedir.

★ A B C D direkt dатадır.

★ Data yine araca özel decoderlarla okunabilir

0

## UDS (Unified Diagnostic Services):

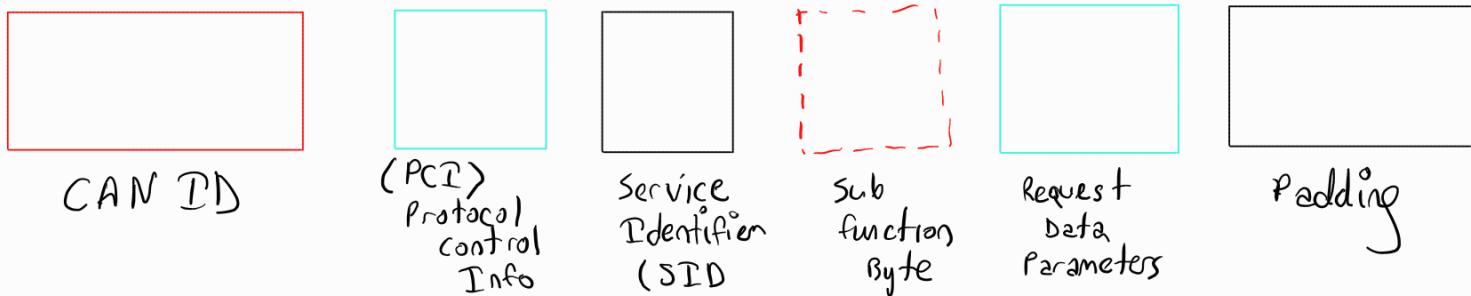
★ Temelde OBD2 ile aynı amaca hizmet eder

★ Server - Client modeli ile çalışır. Client => Tester Tool, Server => Testing vehicle.

★ OBD2 bağlantısından ECU'nun desteklenmesi halinde UDS requestleri atılabilir.

★ Sıri numarası, ECU'lar için yazılım güncelleneleri, DTC'ler, sıkılık gibi veriler alınabilir ve işlemleri yapılabilir.

★ Veriler local serverdan python API servisleriyle cebilip grafana dashboardda okunabilir.



★ PCTI'nın altında UDS ile doğrudan bir alakası yoktur. CAN Bus üzerinden UDS requestlerinin atılması sağlanır. Tek bir CAN frame'e sığmayan mesajlar hakkında veri barındırır. 1-3 byte arası yer kaplar

★ SID kısaca yapacağınız işlem hakkında bilgi verir. Bir nevi fonksiyon

★ Sub function byte SID'in alt fonksiyonudur ve kimi zaman boş olabilir.

★ UDS sadece CAN Bus'a spesifik çalışmaz, LIN Bus, IP gibi protokollerin de altında çalışabılır.

★ OBD2'nin son yıllarda yetersiz kalmasıyla UDS'in önemi artmaktadır. WWH-OBD (UDS based) OBD on UDS gibi teknolojiler OBD2'nin yerini alacaktır.

### CAN open :

★ Temelde Motion oriented makinelere için geliştirilse de kullanım alanları gün geçtikçe artmıştır.

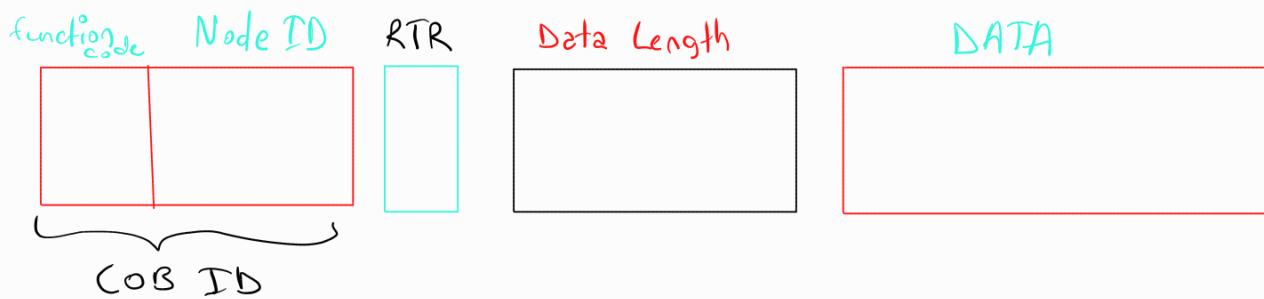
★ 6 temel konsepti vardır:

→ **Communication Models:** Master / Slave, Client / Server, Consumer / Producer gibi iletişim modelleri mevcuttur

→ **Communication Protocols:** Protokoller iletişim için kullanılır. Mesela node ayarları için SDO, gerçek zamanlı data iletimi için PDO kullanılır

→ **Device States:** Noduları temsil eden "mesaj" gibi durumlar bütünüyle

- **Object Dictionary**: Her cihaz OD adında entrylerle config gibi bilgilerini tutar.
- **Electronic Data Sheet**: OD entryleri için standart dosya formatıdır.
- **Device Profiles**: I/O ve Motion Control gibi standartları açıklar.

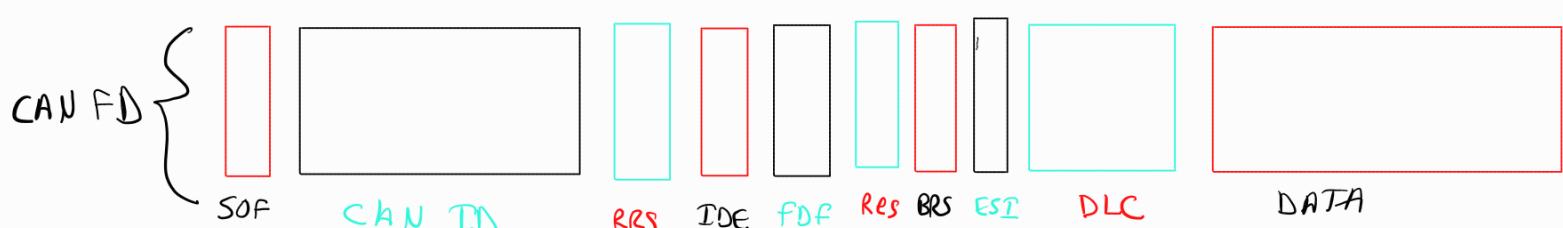
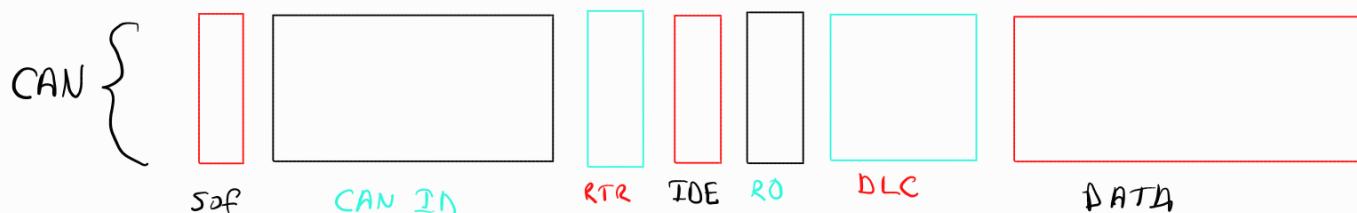


★ PDO ve SDO protokollerini burada da kullanabiliyor.

### CAN FD:

★ CAN'in modern ihtiyaçlara cevap vermemesiyle beraber hem data boyutu hem data gönderme hızı açısından günümüzde daha tercih edilebilir bir protokoldür. Daha az error içermesi ve daha az meraflı olması da kabası.

★ CAN FD'den farklı olarak her frame'de 64 data byte'i bulunur. geleneksel CAN'de bu 8 byte'dır.

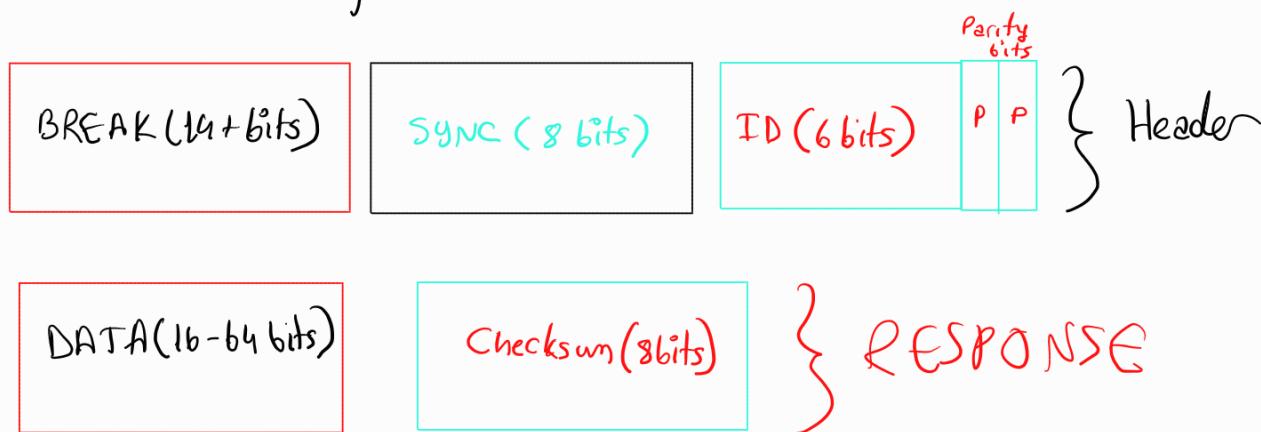


★ RTR vs RRS: CAN'da RTR remote frame'leri tensil eder. CAN FD'de direkt Remote Frame'ler yok. RRS her zaman 0

- ★ r0 vs FDF: Temelde aynı seyler ama FDF'ye sonra gelen 3 bit error frame görevi görüyor. r0 => 0 iken FDF => 1
- ★ res: r0 ile aynı rolü oynar
- ★ BRS: Bit hızı anahtarı, paketin gönderme hızını ayarlayan bitdir. 0 iken 1mbit/s, 1 iken 5mbit/s
- ★ ESI: Hata durum göstergesidir. 0 ise hata aktifdir 1 ise değildir.

## LIN Bus (Local Interconnect Network):

- ★ CAN Bus'a ek olarak üretilmiştir.
- ★ Daha düşük hız ve güvenilirliğe sahip olsa da son derece hesaplı bir protokoldür.
- ★ Silcekle, camlar ıdima gibi işlen gücünü minimumda gerektiren işler için kullanılır.
- ★ 12V'luk tek kablo kullanılır.
- ★ Gelecekte ciddi siber güvenlik sorunlarına yol acabilir, tabii bu problemler çözülmeyece...
- ★ CAN Bus'ta birden fazla master/slave varken, LIN Bus'ta bir master ve sayısız kölesi olabilir. Master tekdir.



- ★ BREAK: Sof görevi görür.
- ★ SYNC: Default 0x55 değerine sahiptir. Her düğümün senkronize olmasını sağlar.
- ★ ID: Gönderilen her LIN mesajı için ID görevi görür. Aynı zamanda header'a hangi düşüğün tepki vereceğini belirler.
- ★ DATA: 16-64 bitslik veri alanıdır.
- ★ Checksum: 8 bitslik bir kontrol toplamıdır.

★ Checksum, LIN frame'in geçerliliğini söylemek

## ★ LIN frame çeşitleri

- Unconditional
- Event Triggered
- Sporadic
- Diagnostic
- User Defined
- Reserved

## CAN DBC :)

★ Basitçe CAN Database, Raw Can Bus datayı decode eden bir text file

## CAN errors :)

★ CAN'in en güçlü noktalarından biri error handling'dır.

★ 6-9-12 bitlik error flagleri olabilir.

★ 5 tip CAN bus error vardır.

1-) Bit Error : Beklenmediği gibi mantıksal error, elektriksel gürültü / bağlantı problemlerinden kaynaklanabilir.

2-) Stuffing Err : CAN, her 8 bitte bir doldurma biti ekler ve bu bit mantıksal olarak yanlış olabilir. İletim hızı olması gerekliden yüksekse karşılaşılabilir.

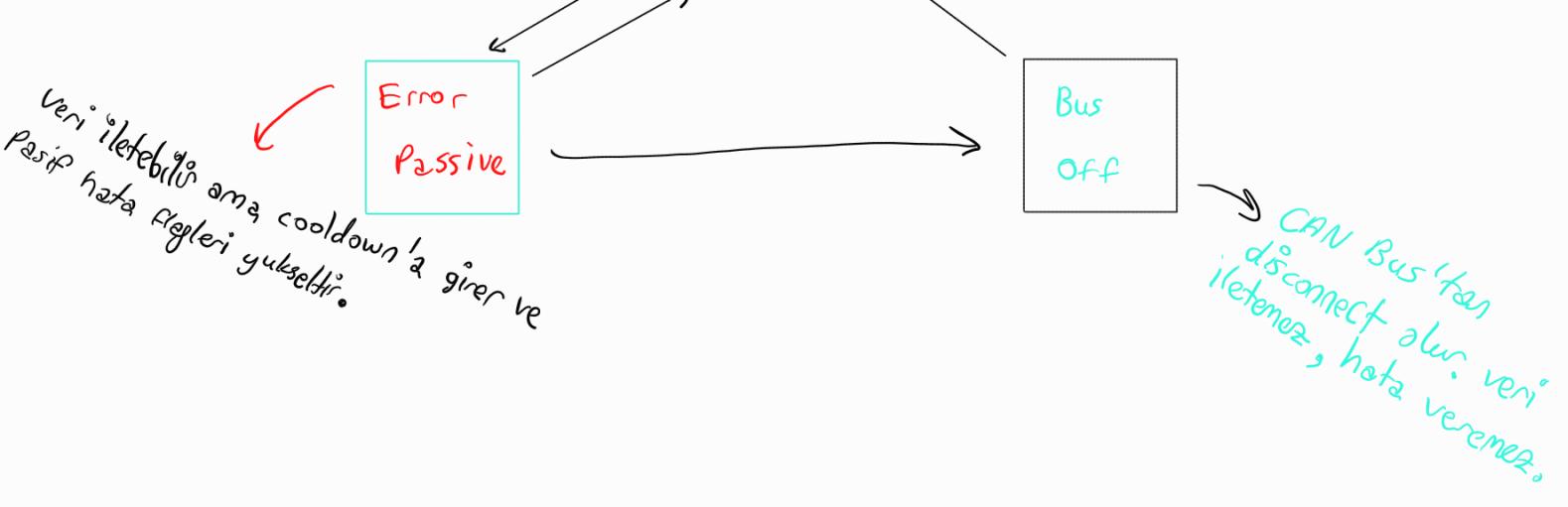
3-) Form error : frame başı veya sonu beklenileneden farklıdır.

4-) ACK error : ACK sinyali alınmadıysa ortaya çıkar.

5-) CRC error : CRC beklener değerinin dışındaysa karşılaşılır. Veri bütünlüğünde problem olabilir.

Error  
Active

default düşündür. Bu durumda hala veri iletimini gerçekleştirebilir. yükseliş algılandığında "Aktif hata bayrakları"

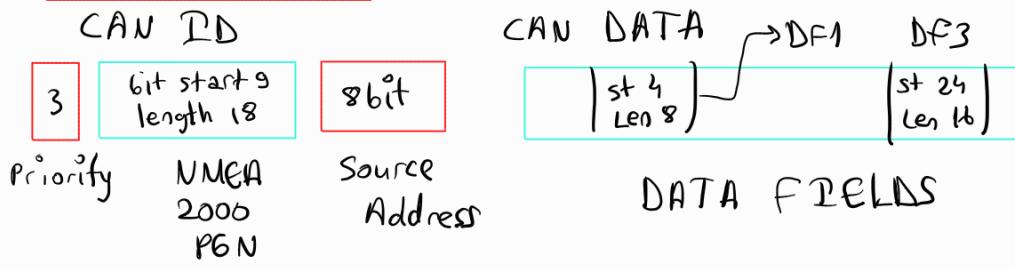


## LIN Bus Errors:

- 1-) Checksum Error
- 2-) Receive Error
- 3-) Synchronization Error
- 4-) Transmission Error

## NMEA 2000:

- ★ Deniz araçları için özelleştirilmiş CAN protokolüdür.
- ★ Güç kaynakları direkt nodelara entegredir. Bu durum kablolama maliyetini azaltır.
- ★ Önceliği versiyonu NMEA 0183'ür ve ona göre hemen hemen her ekipman daha iyi dir.
- ★ Banyak farklı connector çeşidi vardır.
  - T connector (M12 5-pin)
  - Engine Connector (J1939 9-pin)
  - CAT 9-pin
  - Deutsch DT06-6P
- ★ Kilit noktası Fast Packet özelliğidir.
- ★ PGN konsepti burada da kullanılır.



## TSOBUS:

- ★ Tarım teknolojileri için kullanılır.
- ★ Farklı tarım makinesi üreticilerinin ekipmanlarının birbirleriyle uyumlu çalışma-sını sağlar.
- ★ Makineler arası veri paylaşımı ve koordinasyonu sağlar.
- ★ Yeni nesil tarım teknolojileriyle entegre olabilir ve genel tarım verimini artırır.

## CCP (Can Calibration Protocol):

- ★ ECU'ların kalibrasyonunu sağlamak için kullanılır.
- ★ CAN Bus aracılığıyla parametre okuyup yazabılır, hata tespiti yapabilir.

## XCP (Universal Measurement and Calibration Protocol):

- ★ CAN ve diğer protokoller üzerinden ölçüm ve kalibrasyon verilerinin iletimini sağlar.
- ★ Esnek bir yapıdadır ve birçok iletişim hızı/veri tipiini destekler.
- ★ CCP / XCP protokolü single master/multi-slave konseptini benimsenir.
- ★ CAN vs LIN arasındaki ilişki gibi bir UDS vs CCP / XCP ilişkisi mevcuttur.



