

Controler Area Network

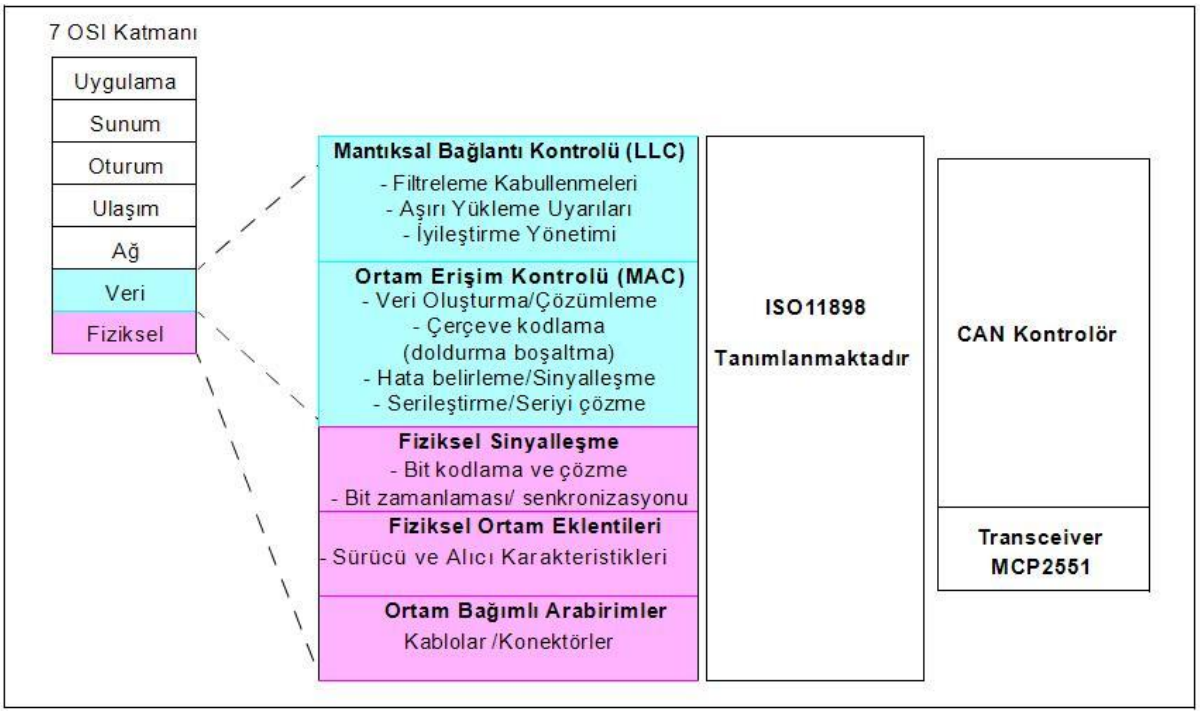
MSc. Serkan AYDIN
serkanaydin@techemail.com
Bahçelievler Erkan Avcı ATTEM Lisesi
Fen Bilimleri Enstitüsü
2005–2006 Güz Dönemi
Yüksek Lisans Mekatronik Eğitimi Bölümü

CAN (CONTROL AREA NETWORK) BUS

MSc. Serkan AYDIN*
*Marmara Üniversitesi Fen Bilimleri Enstitüsü
Göztepe/İstanbul

CAN BUS STANDART TANIMLAMALARI

CAN protokolü dağıtılmış sistemler için tasarlanmış seri haberleşme mantığına dayanan ve sinyalin (Bus) veri hattına gerilim farkı olarak gönderildiği bir protokolüdür. Bu protokol özellikle düşük maliyetli kablolamanın yapıldığı çoklu düğümlerle birbirine bağlanmış ünitelerin gerçek zamanlı kontrolü için uygundur. Standart CAN tanımlamaları OSI' nin fiziksel ve veri katmanında tanımlanmıştır. Bu nedenle standart CAN alt seviye bir protokoldür. Şekil.1' de ISO11898 standart'ında belirlenen CAN protokolünün OSI referans modelindeki tanımlamaları gösterilmektedir.



Şekil.1 CAN ve OSI Tanımlamaları [4]

Sinyalleşme sürecinin başlayışındaki sistemin davranışı, farklı düğümlerle mesaj iletimi için nasıl tanımlamaların yapılacağı, veri çerçevesinin başlıklarının nasıl çözüleceği ve sistem içerisindeki durum raporları gibi işlemlerin gerçekleştirilmesi için ayrıca CAN tabanlı OSI' nin üst seviyelerine kadar tanımlanan protokoller geliştirilmiştir. Kırkın üzerinde olan bu protokollerden belli başlı olanlarına örnek olarak

- CANKingdom
- CANOpen (CAL)
- DeviceNet
- J1939
- OSEK
- SDS

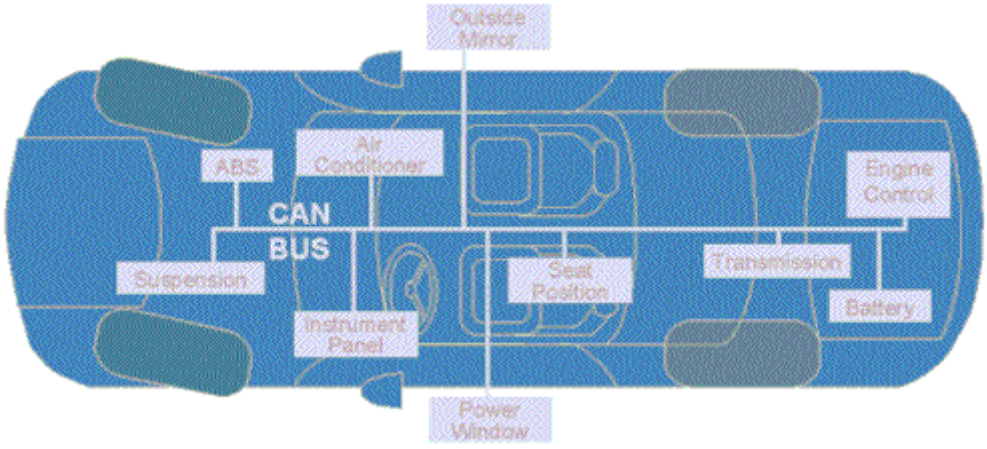
verilebilir.

CAN BUS ELEMANLARI VE UYGULAMALARI

Otomasyon endüstrisinde çok yaygın olarak kullanılan CAN protokolü ilk olarak Robert BOSCH firması tarafından 1980'in sonlarında motor kontrolü için geliştirilmeye başlanmıştır [2]. Tablo1' de ayrıca ayrıntılı bir şekilde kullanım alanları örneklerle belirtilen bu protokol özellikle otomotiv teknolojisinde yaygın bir şekilde kullanılmaktadır. Otomotiv Mühendisliği Birliği tarafından da standartlaştırılan bu protokolün Şekil2' de bir otomobil içerisinde bağlı olduğu düğümler ve ilgili üniteler örnek olarak gösterilmiştir.

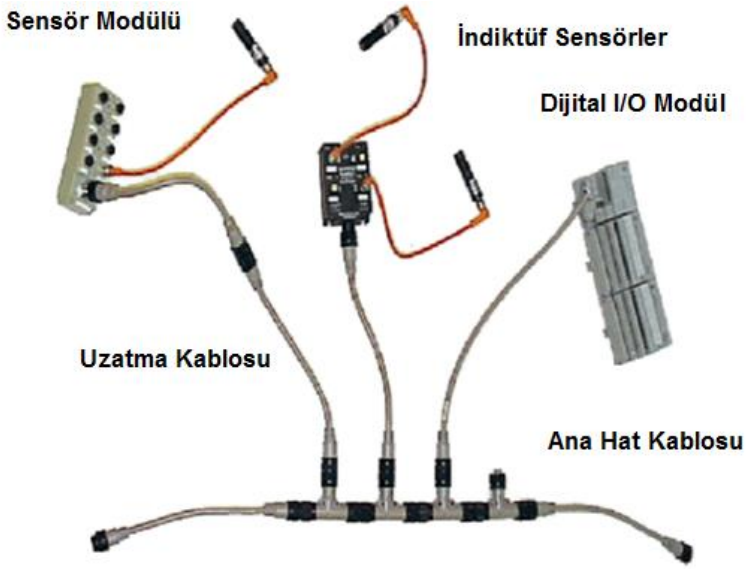
Tablo.1 CAN Protokolünün Kullanıldığı Yerler

*** Motorlu Araçlarda (arabalar, kamyonlar, otobüsler)**
— Motor Kontrol Sisteminin Haberleşmesinde (ABS, Vites Kontrol, Aktif Süspansiyon)
— Araç Ünitelerinin Kontrolü **(**Gösterge Panelleri, Işıklandırma, Havalandırma, Camlar, Otomatik Kapı Kilitleme, Hava Yastığı, Emniyet ve Bel Kemeri)
*** İş Araçlarında**
— Yük makinelerinde (Forkliftler, Traktörler, Yük Katarlarında Hidrolik Kontrolü için)
*** Trenlerde**
— Farklı Elektronik Alt Sistemler Arası Veri Transferinde (Hızlanma, Fren, Kapı Kontrol, Hata Mesajları, Sistem Kontrolü
*** Endüstri Otomasyonunda**
— Otomasyon Elemanlarının Birbirine Bağlanarak Haberleştirilmesi (Kontrol Ünitelerinin, Sensörlerin, Aktüatörlerin)
— Aşağı Yukarı Sayma İşlemlerinde ve Veri Takibinde
— Makine Kontrolünde (Baskı Makineleri, Kâğıt ve Tekstil Makineleri vs)
— Taşıma Sistemlerinde
*** Medikal Cihazlarda**
— Bilgisayarlı Görüntüleme Sistemlerinde, X-Ray Makinelerinde, Dişçi koltuklarında
*** Bina Otomasyonunda**
— Isıtma, Havalandırma, Işıklandırma, Güvenlik
— Asansör ve Yürüyen Merdivenlerde
*** Beyaz Eşyalarda**
— Bulaşık Makineleri, Çamaşır Makineleri, Kahve Makineleri
*** Ofis Otomasyonunda**
— Fotokopilerde Kâğıt Tutucu Arabirimlerinde, Kağıt Besleme ve Kağıt Ayırıcı Sistemlerinde



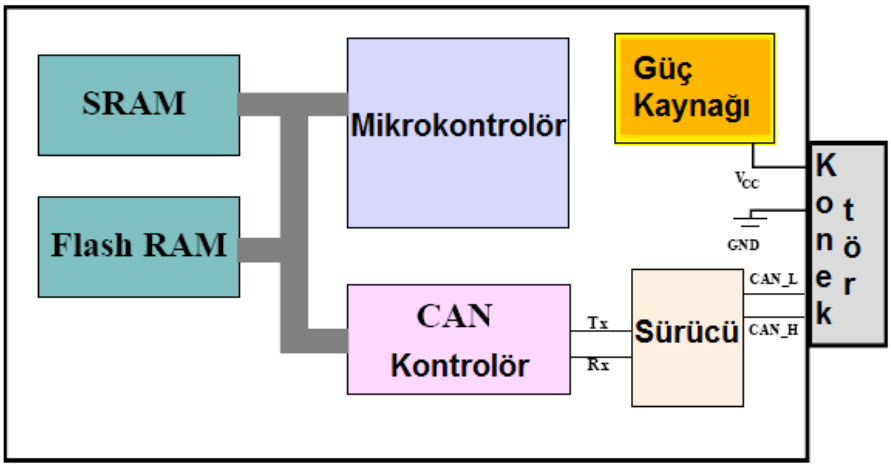
Şekil.2 Araç İçindeki Örnek Bir CAN Veri Hattı ve Düğümleri

CAN protokolü genellikle kısa mesafeler arası haberleşmede kullanılmaktadır. Şekil.3' de örnek olarak verilmiş bir CAN ağı ve elemanları gösterilmektedir. Bir CAN ağı birkaç elemandan oluşabileceği gibi yüzlerce elemandan da oluşarak kompleks bir yapı haline gelebilir. Bu ağda her cihazın belli bir ID' si (Kimlik Numarası) bulunduğundan haberleşme etkili bir şekilde yapılabilmektedir.



Şekil.3 Örnek Bir CAN Ağı ve Elemanları

Bir cihaz eğer CAN protokolü üzerinden haberleşebiliyorsa bu cihazda bir CAN modülü ya da CAN destekli bir mikrokontrolör mevcuttur. Mikrokontrolörler Motorola 65XXX veya Intel 80XXX gibi çeşitli yapılarda olabilir. CAN çerçeve yapısının oluşturulması için esas olan mikrokontrolör veya modül içerisinde bulunan yardımcı bir çevre donanımının bulunmasıdır. CAN Şekil.4' te örnek bir CAN modül blok diyagramı gösterilmiştir.

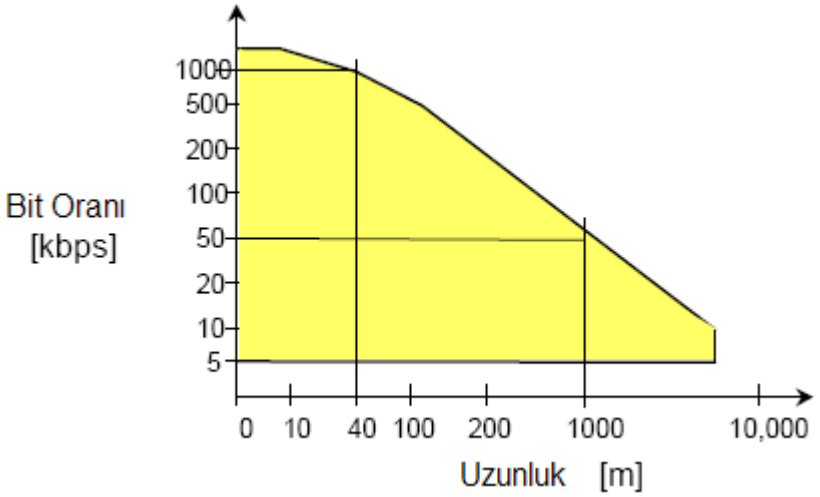


Şekil.4 CAN Modül Blok Diyagramı

Standart CAN 'de 1Mbit'e kadar veri transferi sağlanabilmektedir. Yalnız veri transfer oranı arttıkça kablo mesafesi kısalmaktadır. Kablo mesafesine bağlı olarak yaklaşık olarak aktarılabilecek veri miktarı Tablo.2' de gösterilmiştir.

Tablo.2 CAN Kablo Mesafesi-Veri Aktarım Oranı

Veri Miktarı/Zaman	Mesafe
10 Kbit/s	6.7 km
20 Kbit/s	3.3 km
50 Kbit/s	1.3 km
125 Kbit/s	530 m
250 Kbit/s	270 m
500 Kbit/s	130 m
1 Mbit/s	40 m



CAN elemanlarının kablo bağlantıları ve konektörleri belli bir standarda göre yapılmaktadır. Kablo bağlantılarında standarda uyulması ile ilgili sistemin bağlantılara bakılarak hemen tanınmasını sağlamaktadır. Şekil.5' te örnek CAN elemanları ve konektörleri gösterilmektedir.



Şekil.5 Örnek CAN Konektörleri ve Elemanları

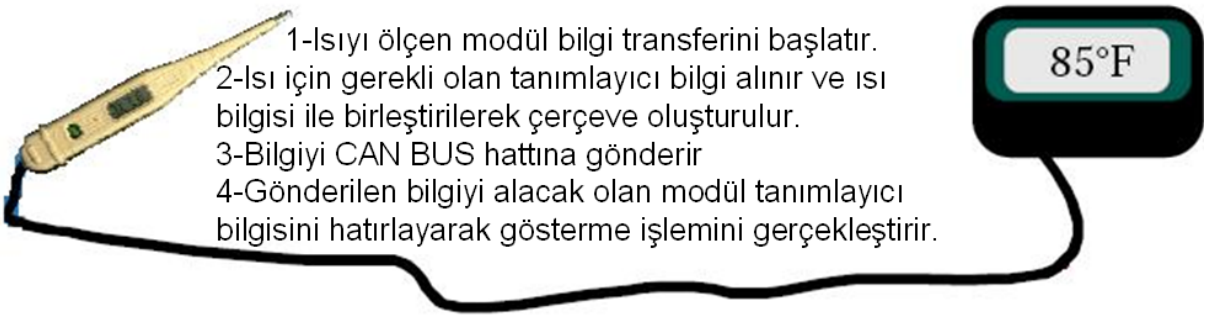
CAN BUS BİLGİ İLETİMİ

CAN protokolünde bilgi iletimi şu şekilde gerçekleşmektedir. Gönderme işlemini gerçekleştiren modülün mesajı tüm modüllere bit bit bus hattına gönderilir. Gönderici dahil alıcı olarak görev yapan tüm modüller bus hattındaki sinyalleşmede aktif durumdadır. Bütün modüller haberleşme anında hata

doğrulama yaparlar. Yanlış gönderilen mesajın tekrar gönderilmesi için istek yapılır. Gönderici hariç diğer alıcı modüller hatasız gelen mesajların bir kopyasını hafızalarına kaydederler.

Alt seviye bir protokol olması dolayısıyla, CAN ile başka bir protokolün koşturulduğu modüllerle iletişim yapılabilir. Tüm modüllerin bir kimlik bilgisi vardır. Alıcı modüller tüm doğru verileri kopyalayacağı için ağ içerisindeki modüllerde global bir veri tabanı oluşturulur. Haberleşmede kaynak veya hedefin herhangi bir ayrıcalığı yoktur.

CAN sisteminde haberleşme Şekil.6' da gösterildiği üzere şu şekilde gerçekleşmektedir. Örnek olarak verilen; ısı ölçen bir CAN donanımı kendi ID' sini ve sıcaklık bilgisini içeren mesaj paketini oluşturarak iletim hattına sinyali gönderir. Bu mesaj hatta bağlı tüm modüller tarafından alınır yalnız ilgili mesaj hangi modül tarafından işlenecekse o modül yapılacak işlemi gerçekleştirir. Şekil.6' da gösterildiği gibi gösterge aleti bus hattından gelen mesajı alıyor mesajın alındığını mesaj paketini işleyerek belirtiyor ve gösterme işlemini gerçekleştiriyor. Burada önemli olan husus ID' lere göre tanımlamaların ve işlemlerin yapılmasıdır. Her modül kendi gerçekleştirdiği işlemi ve kendisini sabit bir ID ile tanımlamaktadır.



Şekil.6 Örnek Bir CAN Sinyalleşme Süreci

CAN protokolünde farklı bus sistemleri uygulanabilmektedir. Bunlar

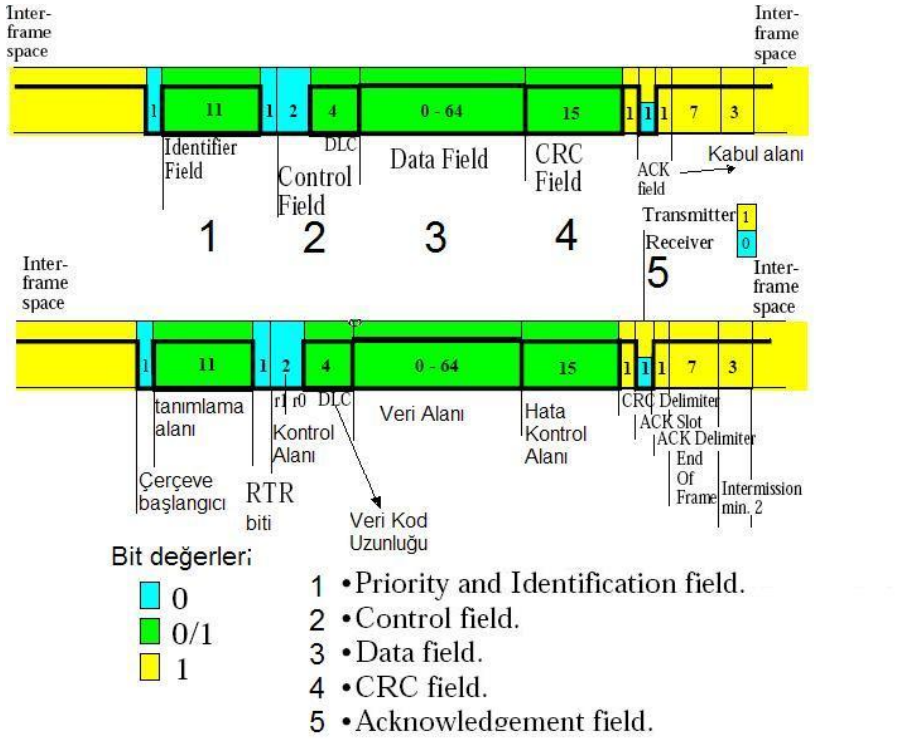
- Bit tabanlı karar verme (bitwise arbitration)
- Usta/Köle (Master/Slave)
- Papatya zinciri (Daisy Chain)
- TDMA (Zaman bölmeli çoklu erişim) olabilmektedir.

CAN MESAJI

CAN protokolünde 4 tip mesaj bulunmaktadır. Bunlar

- Veri Çerçeve
- Uzak Çerçeve
- Hata Çerçevesi
- Aşırı yük Çerçevesi' dir

Veri çerçevesi (Data Frame): En çok kullanılan çerçeve olup bilgi iletiminde kullanılmaktadır. Data çerçevesi Şekil.7 de gösterildiği gibi belli başlı 5 kısımdan oluşmaktadır.



Şekil.7 Veri Çerçevesi

Tanımlayıcı alanı CAN 2.0A standardında 11 bit iken CAN 2.0B' de 29 bit uzunluğundadır. Bunlardan 11 bit olanına standart çerçeve 29 bit olanına da genişletilmiş çerçeve denilmektedir. Karar verme (arbitration) işlemi bu kısımda yapılır. Bus hattında oluşacak çarpışma (collision) durumunda karar verme işlemi tanımlama alanının bit bit incelenmesi ile yapılır. Bu işleme arbitration adı verilmektedir. RTR biti bu kısımdadır. RTR biti uzak çerçevenin oluşturulmasında setlenir.

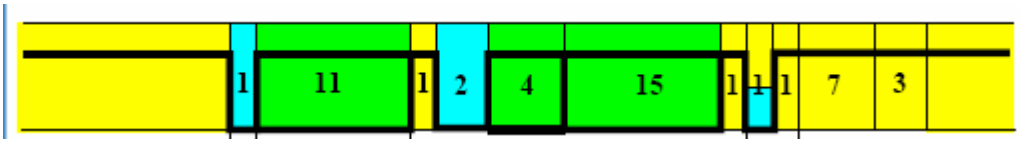
Kontrol kısmı veri alanında bulunan bilginin uzunluğunu vermektedir. (8XDLC = Veri Bit Uzunluğunu) formülü ile veri uzunluğu hesaplanır. Bu nedenle 0 dan 8 e kadar değer alır. Kontrol alanında bulunan diğer bitlerden birisi uzatılmış çerçeveyi tanımlamak için diğeri ise harici kullanım için ayrılmıştır.

Veri alanı iletilmek istenen bilgiyi içermektedir. 64 bit uzunluğundadır. Bazı CAN mikrokontrolörleri bu kısımda filtreleme ve tanımlama işlemleri yapabilmektedir.

CRC kısmında mesajda doğruluk sağlaması yapılması için gerekli olan bitleri içerir. CRC kısa tipli mesajlar için uyarlanmıştır. CAN hattında CRC ile yalnızca bir hata doğrulaması yapılabilir.

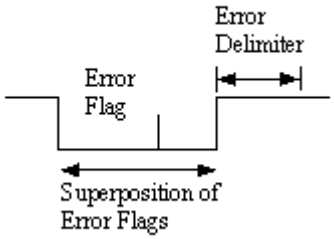
Mesajın en az bir kullanıcı tarafından alındığını kabul alanında kısımda gösterilir. Gönderici bu biti 1 yapar eğer bu bit okunmuşsa alıcı tarafından 0 yapılır.

Uzak Çerçeve (Remote Frame): Herhangi bir mesaj bir alıcı tarafından alınamadığı zaman alıcı RTR isteği ile Uzak çerçeveyi oluşturur ve bunu bus hattına gönderir. Bu mesaj alındığı zaman ilgili çerçeve tekrar gönderilir. Şekil.8' de Uzak çerçeve yapısı gösterilmektedir. Uzak çerçevede Veri kısmı bulunmamaktadır. RTR biti setlenerek uzak çerçeve yapılır.



Şekil.8 Uzak Çerçeve

Hata Çerçevesi (Error Frame): Hata çerçevesi özel bir mesajdır. Herhangi bir modül hatalı bir işlem algıladığında diğer modüllerinde bu durumu algılaması için hata çerçevesi gönderilir. Bu durumda modüllerin hata sayaçları artar. Şekil.9' da gösterilen hata çerçevesi sistemin devamlı meşgul olmaması için bayrak yapısı şeklindedir ve 6 bitten oluşan hata bayrağı (error flag) ile 8 bitten oluşan (error delimeter). Hata mesajı algılandığında gönderici mesajını tekrar gönderir.

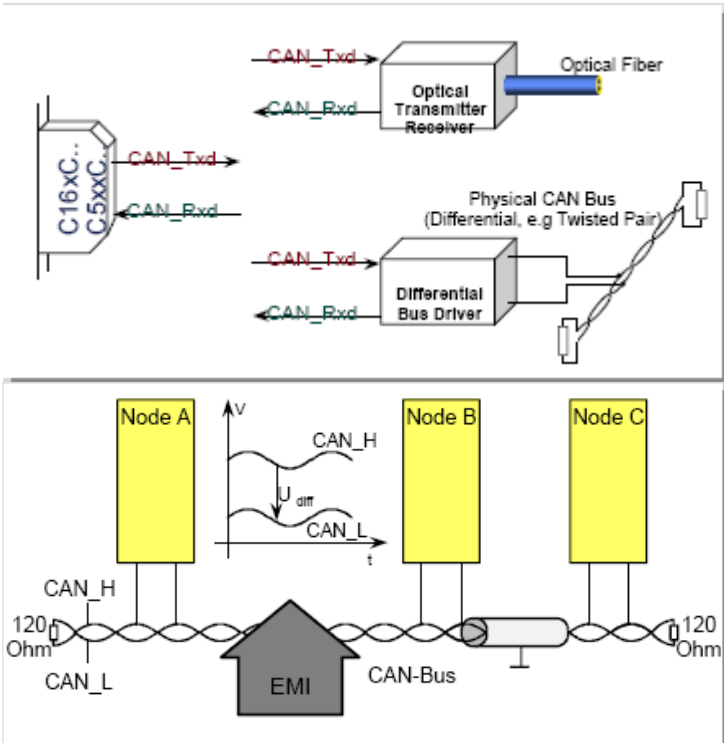


Şekil.9 Hata Çerçevesi

Aşırı yük Çerçevesi (Overload Frame): Sadece bir iki çeşit CAN kontrolünde kullanılan bu çerçeve bir modülün meşgul olduğunu belirtmek için kullanılır. Eğer bir modül aşırı yük çerçevesi yollamış işe bu modül kendisine o an için bilgi gönderilmemesini ağdaki modüllere bildirmektedir.

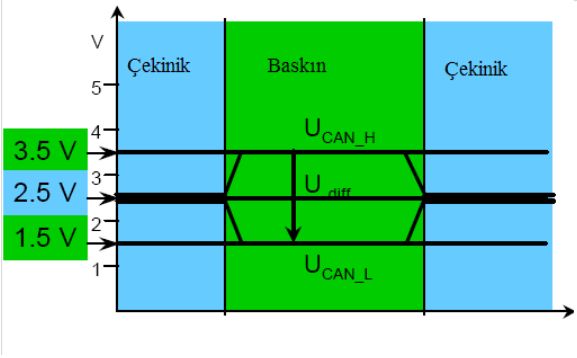
CAN SİNYALİ

CAN sinyalinde NRZ kodlaması kullanılmaktadır. Bu kodlamada sinyal bit süresi boyunca ya da segment boyunca sabittir. O seviyesi baskın 1 seviyesi çekinik olarak tanımlanmıştır. Sinyal fark sinyali olarak bus hattına iletilir. Genellikle mikrokontrolörün RX (seri alış) ve TX (seri gönderiş) uçları sürücü (transreceiver) entegrelerine bağlanılır ve CAN H, CAN L seviyeleri elde edilir. Şekil.10' da CAN sinyalinin diferansiyel olarak veri hattına uygulanması gösterilmektedir.

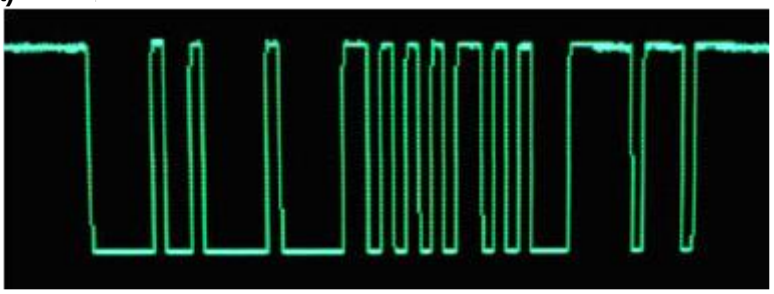


Şekil.10 CAN Sinyalinin Bus Hattına Uygulanışı [3]

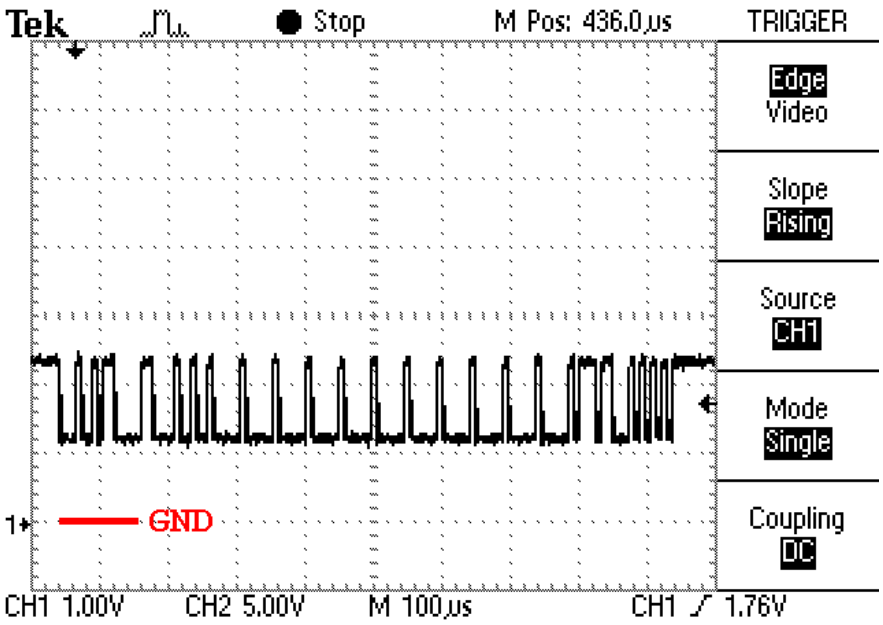
Bus hattına cihazlar çift kablo üzerinden CAN H ve CAN low uçları ortak olacak şekilde bağlanmaktadır. 120 OHM' luk dirençlerle sonlandırılan hatlarda uygun osilaskop görüntüsü Şekil.11' de verilmektedir.



a)



b)



c)

Şekil.11 a) Sinyal Seviyesi **b)**CAN Anlg. Osilaskop Görüntüsü **c)**12H ID' li Djtl. Osilaskop Görüntüsü [1]

CAN BUS' TA KARAR VERME (ARBITRATION)

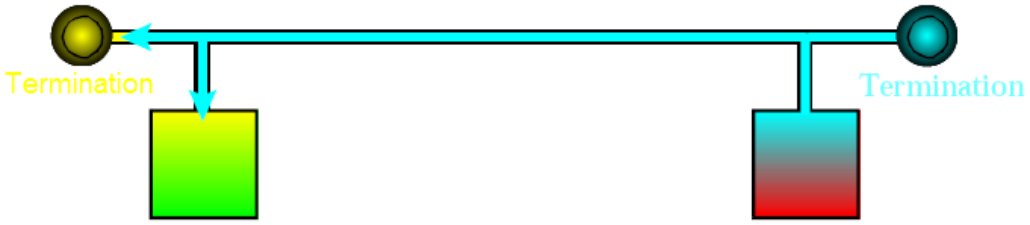
Bit tabanlı olarak yapılan karar verme işlemi mesajın tanımlayıcı alanında yapılmaktadır. Bu durum çarpışma anında hangi modülün iletim hattına mesaj göndereceğini belirlemektedir. Hat erişimi mesaj önceliğindeki yapıya göre çözümlenmektedir. CSMA/CR (Carrier Sense Multiple Access / Collision Resolution) olarak hat çözümlemesi yapılır. Hat erişimindeki karışıklıklarda çerçeve yapısına zarar verilmez. Gecikme zamanında belirsizlik yoktur. CAN sisteminde düğüm adresi yoktur. Bütün düğümler bütün mesajları alır gelen mesajın kullanılıp kullanılmayacağına kendisi karar verir.



1: Yeşil düğüm boş bus hattına bilgi göndermek için çekinik sinyalle (Lojik 1) başlayarak çerçevesini gönderiyor.



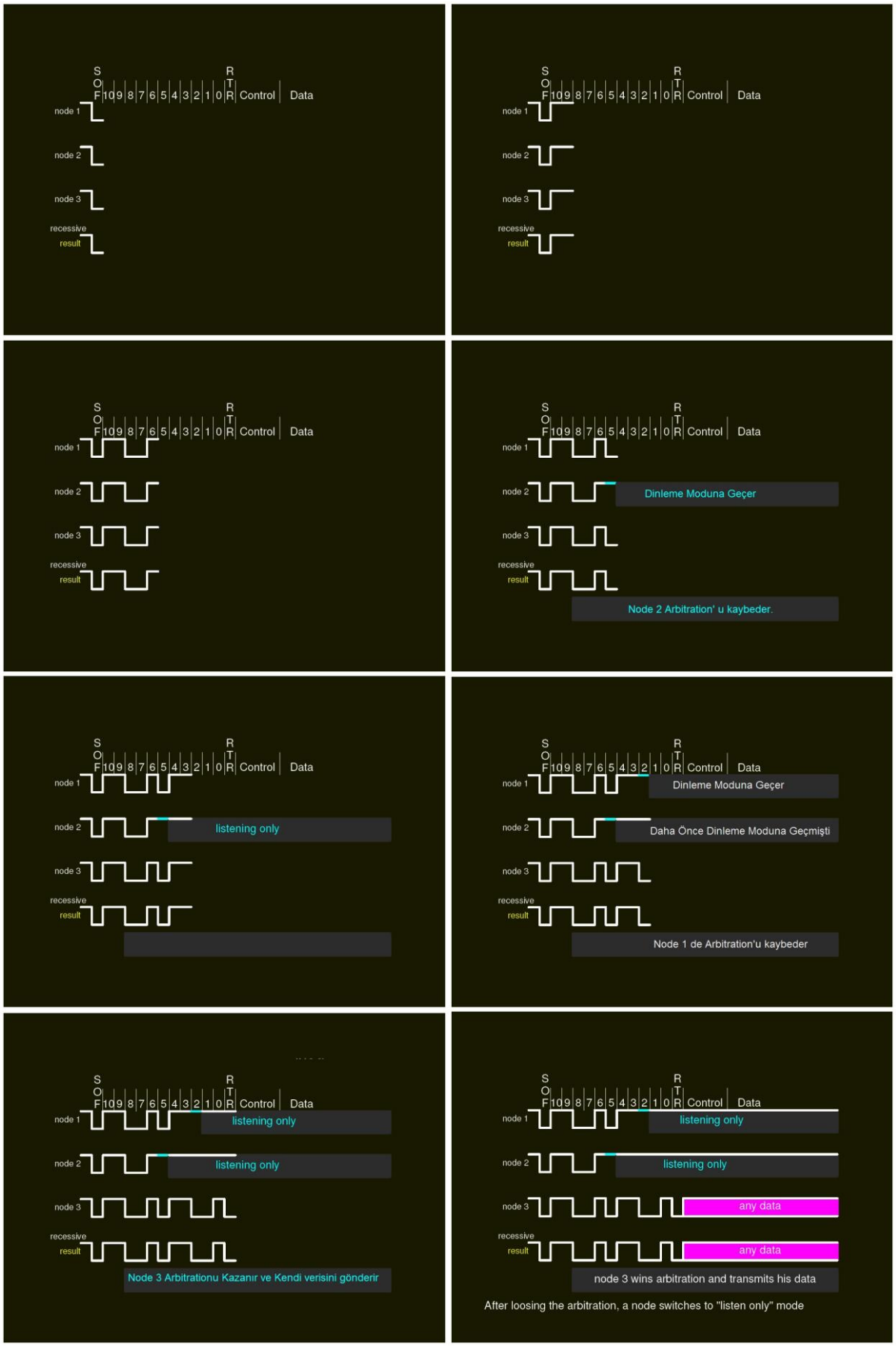
2: Daha yeşil düğüm gönderdiği sinyali diğer düğümlere ulaşmadan kırmızı düğüm baskın (0) sinyali hatta gönderiyor



3: Yeşil düğüm baskın sinyali görünce dinleme moduna geçiyor. Veriyi gönderen düğüm sinyal en son noda erişine kadar bekliyor ve veri gönderme işlemi tamamlanıyor.

Şekil.12' de karar verme olayı bit tabanlı olarak aşama aşama incelendiğinde 3 tane düğüme bağlı CAN modüllerinin aynı anda hatta erişmek istedikleri durum gösterilmektedir. 2. düğüme bağlı modül dinleme moduna geçene kadar üç Modülün ID kısımlarındaki lojik sinyal seviyeleri aynıdır. 2. modülün dinleme moduna geçmesi; bu modülün çekinik olarak ifade edilen Lojik 1 sinyalini gönderirken diğer modüllerin baskın olan Lojik 0 olarak ID kısımlarının 5. bitini bus hattına göndermesidir. Daha sonra ID kısmının 2 bitine gelindiğinde 1. modül aynı şekilde çekinik seviye ile hatta erişmek isterken 3. modül hatta baskın seviye ile erişmiştir. Bu şekilde 1. düğümde hatta dinleme moduna geçmiştir. Hattı ele geçiren 3. modül bilgi gönderme hakkını kazanırken diğer modüller gönderici modülün mesajını almak ve bir kopyasını oluşturmak için dinleme moduna geçmiştir.

(CSMA/CR [Carrier Sense Multiple Access / Collision Resolution]) olarak adlandırılan çarpışma yönetimine göre CAN protokolü hat erişim çözümlemesini yapmaktadır. Çarpışma anı bit tabanlı karar verme sistemine göre çarpışma gerçekleştiği an çözümlenmektedir. Örneğin Ethernet protokolünde çarpışma anı gerçekleştiğinde çarpışan taraflar rasgele belirlenen bekleme sürelerine göre beklemekte takrar hatta erişmektedir. CAN bus sisteminde ise çözümleme işlemi gerçek zamanlı olarak yapılmakta baskın olan ID numaralı modül hatta bilgi gönderme hakkını kazanmaktadır.



Şekil.12 CAN BUS Arbitration Karar Verme

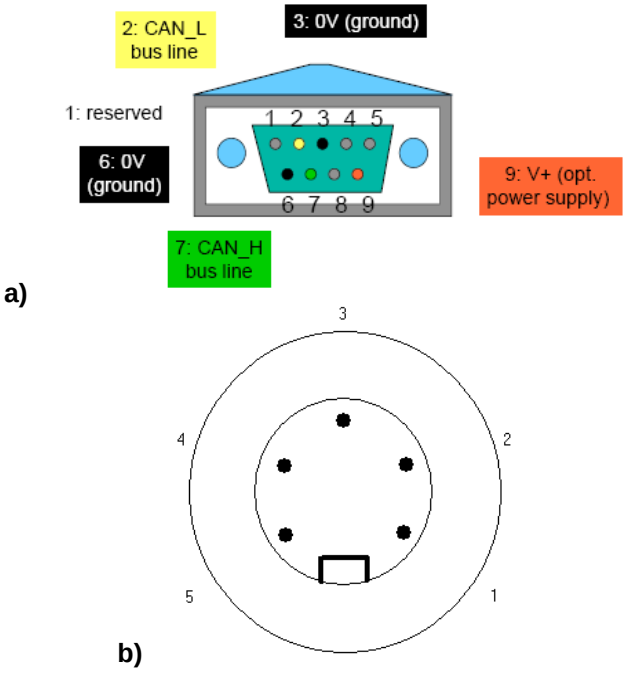
CAN BİT ZAMANLAMASI

Bit zamanlaması ve senkronizasyon çok önemli bir unsurdur. Senkronizasyon işlemi hattın uzunluğundan hattın düzgün sonlandırılmasına, seçilecek olan kontrolörlerin kaydedici yapısından kullanılan osilatöre kadar geniş bir hesaplama gerektiren süreçtir. Bu nedenle genelde kesin sonuç elde etmek için tolerans aralıkları geniş bırakılarak limitler fazla zorlanmamaktadır.
http://www.kvaser.com/can/protocol/timing_calc.htm adresinde bit zamanlaması için web tabanlı uygun bir hesaplama uygulaması gösterilmektedir.

CAN KABLO BAĞLANTILARI

CAN kablolaması karışıklığa neden olmamak için belli bir standarda göre yapılmaktadır. Tablo.3'de ve Şekil.13'de kablo bağlantı isimleri ve şekilleri gösterilmektedir.

9-PİN DSUB KONEKTÖR			**5-PİN MİNİ C**		**6-PİN Alman DT04-6p**	
1	-	Boş	Kanal	Çıplak	Negatif Gerilim	Siyah
2	CAN_L	CAN_L Sinyali	V+	Kırmızı	CAN_H	Beyaz
3	CAN_GND	CAN Şase	V-	Siyah	Harici Sinyal Şasesi	Sarı
4	-	Boş	CAN_H	Beyaz	Harici Başlangıç	Gri
5	(CAN_SHLD)	**Opsiyonel** CAN kalkan	CAN_L	Mavi	Pozitif Gerilim	Kırmızı
6	(GND)	**Opsiyonel** Harici Şase			CAN_L	Mavi
7	CAN_H	CAN_H Sinyali				
8	-	Hata hattı içim ayrılmış				
9	CAN_V+	**Opsiyonel** CAN gerilimi				



Şekil.13 a) 9-Pin DSUB Erkek 9' lu konektör. **b)** 5-Pin Erkek Mini C [1]

Veri Hattı Hata Modları:

ISO 11898 CAN kablolarında meydana gelebilecek hatalar şu şekilde tanımlamıştır.

Bunlar

1. CAN_H kesildi
2. CAN_L kesildi
3. CAN_H akü voltajına bağlandı
4. CAN_L toprak gerilimine bağlandı
5. CAN_H toprak gerilimine bağlandı
6. CAN_L akü voltajına bağlandı
7. CAN_L CAN_H kablosuna bağlı
8. CAN_H ve CAN_L aynı noktada kesildi
9. Sonlandırmada bağlantı problemi
10. olarak tanımlanmıştır.

Hata mesajları için 1, 6 ve 9 numaralı pinlerin kullanılması tavsiye edilmektedir.

CAN SİSTEMİNDE HATA BELİRLEME

Hata tespit işlemi CAN sisteminin en önemli özelliklerinden biridir. CAN sisteminde hata oluşması durumunda mesajın tekrar gönderilmesi ile hatanın çözümlemesi hedeflenmiştir. Bütün CAN modüllerin sinyalleşme sürecinde hata çözümleme işlemini gerçekleştirir. Bir hata oluştuğunda hatayı tespit eden modül hata mesajı ile durumu diğer modüllere gönderiri. Bu esnada hat trafiği messgul olur. Hatayı tespit eden modüller gelen mesajı göz ardı eder ve mesajın tekrar gönderilmesi için dinlemede kalırlar. Her modülün iki tane sayıcısı bulunmaktadır. Bunlar Gönderici Hata Sayıcısı ve Alıcı Hata Sayıcısı' dır. Bu sayıcıların artması ve azalması bazı kurallara dayanmaktadır. Esasen gönderici eğer bir hata algıladıysa alıcılardan daha hızlı olarak kendi gönderici hata sayıcısını 1 arttırır sonra alıcılar alıcı hata sayıcısını 1 arttırmaktadır. Eğer bir modülün hata sayıcısı ciddi bir sayıya ulaştıysa bu modül "error passive" moduna geçmektedir. Bu modülün hat erişim trafiğinde daha bulunmayacağı anlamına gelmektedir. Hata sayıcılarının kullanılması sadece hata tespiti için değil ayrıca güvenilirlik işlemi içinde gerçekleştirilmektedir.

Hata belirileme işlemi genel olarak 5 çeşit yapılmaktadır. Bunlar

1. Bit İzleme (Bit Monitoring)
2. Bit Doldurma (Bit Stuffing)
3. Çerçeve Kontrol (Frame Check)
4. Alındı Cevap Kontrolü (Acknowledgement Check)
5. CRC Kontrolü (Cyclic Redundancy Check) 'dır.

1. Bit İzleme (Bit Monitoring):

Bütün modüller gönderilen sinyal seviyelerini kontrol etmektedir. Eğer bir bitin seviyesi diğer bitlerin seviyelerinden farklı ise hata mesajı ilgili hatayı tespit eden modül tarafından gönderilmektedir. Yalnız kara verme anında bu tip bir mesaj gönderilmemektedir.

2. Bit Doldurma (Bit Stuffing)

Eğer arka arkaya 5 bit aynı seviyede gelmişse farklı seviyede bir bit 6 bit olarak bu gruba eklenmektedir. Alıcılar bu ilave biti kaldırmaktadır. Burada amaç DC bir seviye oluşmuş izleniminin ortadan kaldırılmasıdır. Bu özellik alıcılara hata tespiti için ayrıca bir şans tanımaktadır. Eğer ardı ardına 5 ten fazla lojik seviye algılanırsa "stuff error" sinyali üretilir.

3. Çerçeve Kontrol (Frame Check)

CAN çerçevesinde bazı standart kısımlar bulunmaktadır. Bunlar CRC Delimiter, ACK Delimiter, End of Frame ve Intermission kısmlarıdır. Bunlar için ayrıca bir çerçeve kontrolü yapılmaktadır. Eğer CAN

kontrolörü bunların içlerinde çerçevenin formunu bozacak bir değerle karşılaşırsa “form error” sinyali üretmektedir.

4. Alındı Cevap Kontrolü (Acknowledgement Check)

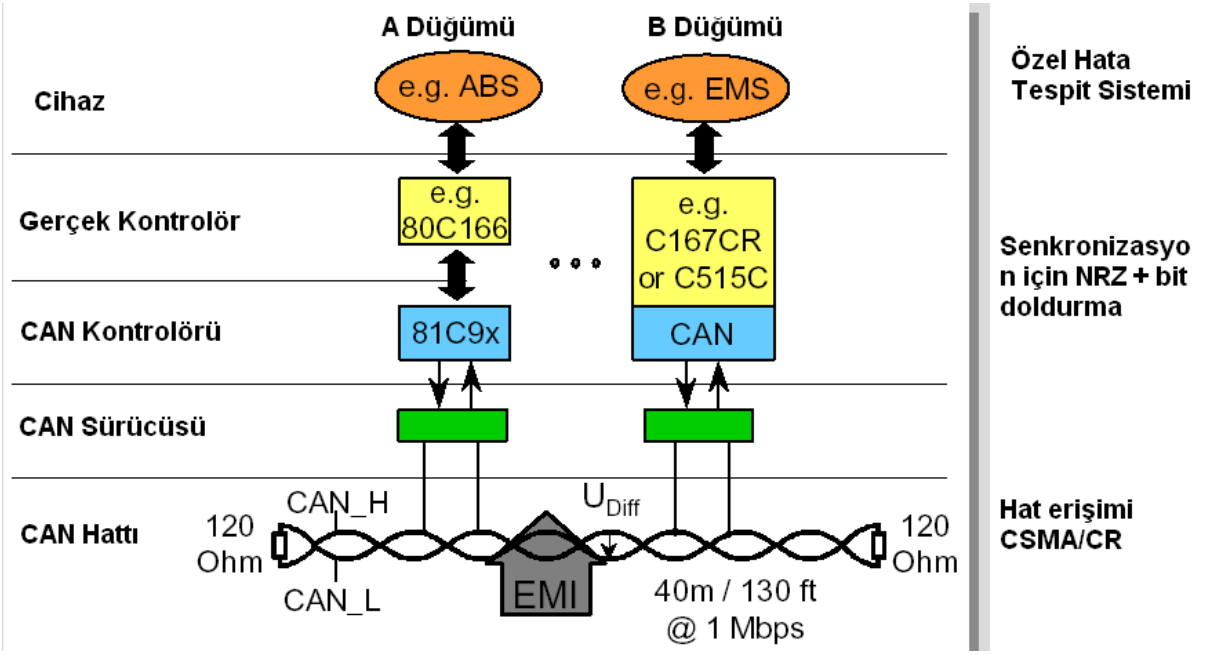
Gönderici modül hatta mesaj gönderdiğinde Acknowledgement bayrağını Lojik 1 yani çekinik sinyale seviyesine getirir. Eğer bir modül bu mesajı okursa ilgili bayrağı Lojik 0 baskın sinyal seviyesine getirmektedir. Eğer gönderici bunun tersi bir olayla karşılaşırsa yani gönderdiği mesaj aynen kendisine gelirse “ acknowledgement error” sinyali üretilir.

5. CRC Kontrolü (Cyclic Redundancy Check)

Her gönderilen mesaj paketine 15 bitlik CRC kontrolü uygulanmaktadır. İletim sırasında sinyale seviyelerinde oluşabilecek hatalar nedeniyle mesaj bozulursa bu CRC ile tespit edilmektedir. Böyle bir durumda kontrolör “CRC hata” sinyali üretmektedir.

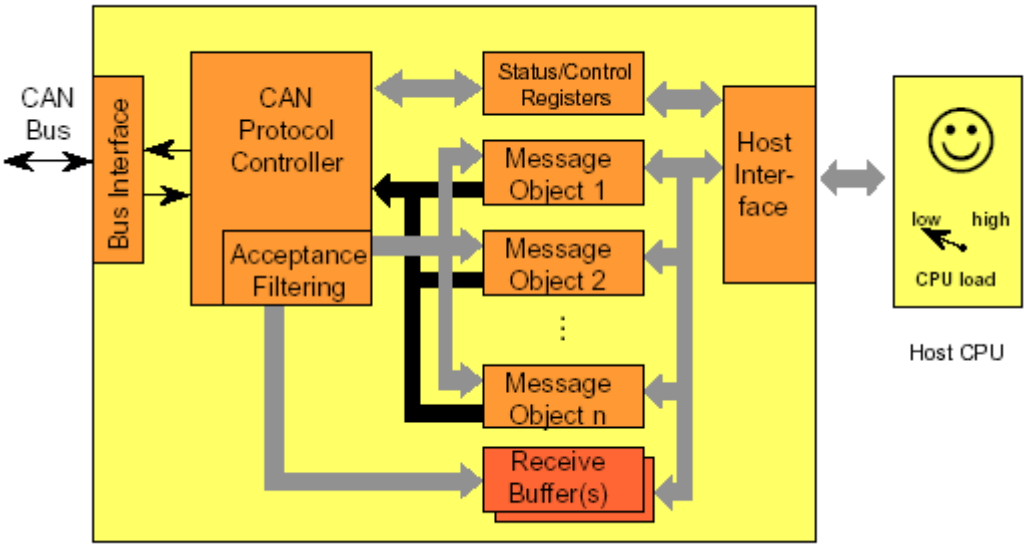
CAN UYGULAMASI

CAN Bus’ta herhangi bir düğüme farklı bir cihaz bağlanabilmektedir. Modüllerin haberleşme kısmı bir CAN kontrolör tarafından yapılırken ölçüm, yönetim kontrol gibi işlemler ev sahipliği yapan gerçek kontrolör tarafından yapılmaktadır. Sinyal seviyelerinin ayarlanması işlemi ise sürücü (transreceiverlar) tarafından gerçekleştirilir. Şekil.14’ de bir araç içerisinde bulunan ABS ile Elektro Manyatik Sensörün CAN Bus üzerinden bağlantısı gösterilmektedir. Bu uygulamada 8051 tabanlı mikrodentleyiciler kullanılmıştır.



Şekil.14 CAN BUS Uygulaam Örneği [3]

CAN Mikrodenetleyicileri Basit ve Tam olmak üzere iki çeşittir. Basit CAN denetleyicilerinde aşırı CPU yükü oluşmaktadır. Bu eksikliği gidermek amacıyla Tam (Full) CAN kontrolörleri geliştirilmiştir. Bu kontrolörlerde veri hattından gelen mesajları kaçırmamak ve CPU yükünü azaltmak için Mesaj Merkezi görevi gören kaydediciler kullanılmaktadır. Böylelikle birden fazla CAN mesajının kaydedicilere alınmasıyla daha performanslı haberleşmenin gerçekleştirilmesi sağlanmıştır. Şekil.15’de Tam CAN Kontrolü blok şeması verilmiştir.



Şekil.15 Full CAN Kontrolör Blok Şeması [3]

CAN BUS PROTOKOLÜNÜN ARTILARI VE EKSİLERİ

Bu protokol düşük maliyetli kablolamanın yapılacağı dağınık sistemler için geliştirilmiştir. Kablo maliyetinin ağ yapısına bağlı olarak düşük olması büyük bir avantaj sağlamaktadır. Yalnız ağın bus şeklinde olmasından dolayı kablo kopmalarında sistemin tamamen kullanılmaz hale gelmesine neden olabilir. Sonlandırma işleminin uygun yapılamaması karar verme işleminde lojik seviyelerin tam çözümlenememesinden kaynaklanan hatalara neden olmaktadır. Sistemin mesafeye bağlı olarak bant genişliği düşmektedir. Maksimum 1Mbit/saniye veri iletimi yaklaşık 50 m kablo mesafesine kadar sağlanabilmektedir. Ağda dolaşan veri bütün modüller tarafından okunduğundan geniş bir veritabanı sistem içinde oluşturulmaktadır.

KAYNAKLAR

[1] http://www.Kvaser.com (Erişim Tarihi: Aralık 2005)
[2] http://www.Bosch.com (Erişim Tarihi: Aralık 2005)
[3] http://www.Siemens.com (Erişim Tarihi: Aralık 2005)
[4] http://www.Microchip.com (Erişim Tarihi: Aralık 2005)

EKLER

DS80C410, DS80C411

Network Microcontrollers with Ethernet and CAN

The DS80C410/DS80C411 network microcontrollers offer the highest integration available in an 8051 device. Peripherals include a 10/100 Ethernet MAC, three serial ports, an optional CAN 2.0B controller, 1-Wire® Master, and 64 I/O pins. The DS80C410 and DS80C411 also include 64KB internal SRAM for user application storage and network stack.

To enable access to the network, a full application accessible TCP IPv4/6 network stack and OS are provided in the ROM. The network stack supports up to 32 simultaneous TCP connections and can transfer up to 5Mbps through the Ethernet MAC. Its maximum system clock frequency of 75MHz results in a minimum instruction cycle time of 54ns. Access to large program or data memory areas is simplified with a 24-bit addressing scheme that supports up to 16MB of contiguous memory.

To accelerate data transfers between the microcontroller and memory, the DS80C410 and DS80C411 provide four data pointers, each of which can be configured to automatically increment or decrement upon execution of certain data pointer-related instructions. High-speed shift, normalization, accumulate functions and 32-bit/16-bit multiply and divide operations are optimized by the DS80C410/DS80C411 hardware math accelerator.

The High-Speed Microcontroller User's Guide and the High-Speed Microcontroller User's Guide: Network Microcontroller Supplement should be used in conjunction with this data sheet. Download both at: www.maxim-ic.com/user_guides

Part Number and Description	Internal Program Memory	Internal MOVX SRAM	Serial Ports	16-bit Timers	Data Pointers	10/100 Ethernet MAC	1-Wire Master	Watchdog	Power Fail Reset & Interrupt	Power Mgmt. Mode	Supply Voltage (V)	Package
DS80C410 Network Microcontrollers with Ethernet and CAN	64kB ROM	73.5kB	3	4	4	Yes	Yes	Yes	Yes	Yes	1.8 3.3	100/LQFP

Key Specifications: Networked Microcontrollers

Part Number	Internal Program Memory	Internal MOVX SRAM	Serial Ports	16-bit Timers	Data Pointers	10/100 Ethernet MAC	1-Wire Master	Watchdog	Power Fail Reset & Interrupt	Power Mgmt. Mode	Supply Voltage (V)	Price @ 1k
DS80C410	64kB ROM	73.5kB	3	4	4	Yes	Yes	Yes	Yes	Yes	1.8 3.3	$8.93 @ 1k
DS80C411												$8.00 @ 1k

Key Features

- High-Performance Architecture
  - Single 8051 Instruction Cycle in 54ns
  - DC to 75MHz Clock Rate
  - Flat 16MB Address Space
  - Four Data Pointers with Auto-Increment/ Decrement and Select-Accelerate Data Movement
  - 16/32-Bit Math Accelerator
- Multitiered Networking and I/O
  - 10/100 Ethernet Media Access Controller (MAC)
  - Optional CAN 2.0B Controller
  - 1-Wire Net Controller
  - Three Full-Duplex Hardware Serial Ports
  - Up to Eight Bidirectional 8-Bit Ports (64 Digital I/O Pins)
- Robust ROM Firmware
  - Supports Network Boot Over Ethernet Using DHCP and TFTP
  - Full, Application-Accessible TCP/IP Network Stack
  - Supports IPv4 and IPv6
  - Implements UDP, TCP, DHCP, ICMP, and IGMP
  - Preemptive, Priority-Based Task Scheduler
  - MAC Address Can Optionally Be Acquired from IEEE Registered DS2502-E48
- 10/100 Ethernet MAC
  - Flexible IEEE 802.3 MII (10/100Mbps) and ENDEC (10Mbps) Interfaces Allow Selection of PHY
  - Low-Power Operation
  - Ultra-Low-Power Sleep Mode with Magic Packet® and Wake-Up Frame Detection
  - 8kB On-Chip Tx/Rx Packet Data Memory with Buffer Control Unit Reduces Load on CPU
  - Half- or Full-Duplex Operation with Flow Control
  - Multicast/Broadcast Address Filtering with VLAN Support
- Full-Function CAN 2.0B Controller
  - 15 Message Centers
  - Supports Standard (11-Bit) and Extended (29-Bit) Identifiers and Global Masks
  - Media Byte Filtering to Support DeviceNet™, SDS, and Higher Layer CAN Protocols
  - Autobaud Mode and SIESTA Low-Power Mode
- Integrated Primary System Logic
  - 16 Total Interrupt Sources with Six External
  - Four 16-Bit Timer/Counters
  - 2x/4x Clock Multiplier Reduces Electromagnetic Interference (EMI)
  - Programmable Watchdog Timer
  - Oscillator-Fail Detection
  - Programmable IrDA Clock

Applications

- Data Converters (Serial-to-Ethernet, CAN-to-Ethernet)
- Environmental Monitoring
- Equipment
- Home/Office Automation
- Industrial Control and Automation
- Network Sensors
- Remote Data-Collection
- Transaction/Payment Terminals
- Vending

Application Notes

- App Note 3478: DS80C400/410/411 Flash Memory Selection - DS80C410, DS80C411
- App Note 3550: Using the IAR Compiler for the DS80C400 - DS80C410, DS80C411
- App Note 3664: Expanding TINI's IO Capability - ds80c410, ds80c411
- App Note 3398: DS80C400/DS80C410/DS80C411 Network Boot - DS80C410, DS80C411
- App Note 2935: Design Considerations for CAN Bus and Asynchronous Serial - ds80c410, ds80c411
- App Note 3421: Interfacing 8051-based Microcontrollers to an SCI Port - DS80C410
- App Note 3553: CAN-to-Ethernet Using the DS80C400 - DS80C410