

# KONTROLÖR ALAN AĞI İÇİN DELPHİ PROGRAMI ARACILIĞI İLE GELİŞTİRİLECEK OLAN YAZILIMLARDA KULLANILMAK ÜZERE HAZIRLANAN DELPHİ BİLEŞENLERİ

Ahmet BERKAY<sup>1</sup>Murat ŞEKER<sup>2</sup>E. Murat ESİN<sup>3</sup><sup>1,2,3</sup>Bilgisayar Mühendisliği Bölümü

Gebze Yüksek Teknoloji Enstitüsü, 41400, Çayırova/Gebze, Kocaeli

<sup>1</sup>e-posta: aberkay@bilmuh.gyte.edu.tr<sup>2</sup>e-posta: mseker@bilmuh.gyte.edu.tr<sup>3</sup>e-posta: emesin@bilmuh.gyte.edu.tr

Anahtar sözcükler: Kontrolör Alan Ağı, Delphi, Delphi Bileşenler

## ÖZET

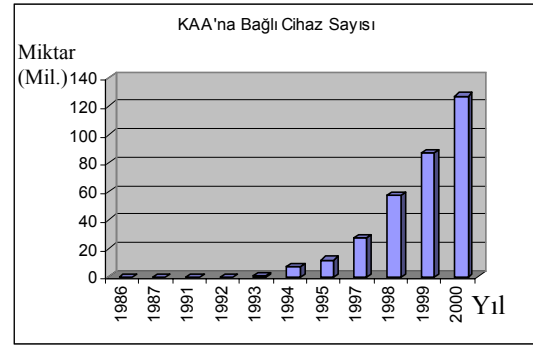
Bu çalışma içerisinde Avrupa Otomotiv Endüstrisinin gelecek 15 yıl içerisinde kullanması kesin olan ve hali hazır da üretilen araçların %90'ında kullanılan Kontrolör Alan Ağı (KAA) incelenecek ve söz konusu olan teknolojinin gelişimi irdelenerek, söz konusu ağı için delphi programı aracılığı ile hazırlanacak olan yazılımlarda kullanılmak üzere geliştirilmiş olan delphi bileşenleri açıklanacaktır.

Kontrollör Alan Ağı sahip olduğu çoklu-yönetici "multi-master" yapısından dolayı çevik ve geniş sistemleri güçlü yöneticilere ihtiyaç duymadan oluşturabilmektedir. Oluşturduğu ağlar da, gelişmiş hata tespit mekanizmaları ve hatalı mesajları doğrulama yeteneği ile %100'lük veri güvenliğini de garanti etmektedir. Bu özelliklerine ek olarak 15 den fazla Kontrolör Alan Ağı cipi üreticisi sayesinde son kullanıcı için hem maddi hem de teknolojik yeterlilik açısından uygun bir alternatiftir.

## 1. GİRİŞ

Kontrolör Alan Ağı (KAA) temelde seri bir haberleşme veri yoludur. KAA, özellikle akıllı cihazların oluşturduğu akıllı sistemler ve/veya alt sistemlerin yüksek veri iletim hızında haberleşme ihtiyacını karşılamak için geliştirilmiştir. KAA, saniyede 1 Megabite kadar çıkan veri iletişim hızlarında bile çok mükemmel bir hata tespit ve onaylama mekanizması ile çalışmaktadır.

KAA tarihinin dönüm noktalarını da gösteren Şekil.1'den de anlaşılacağı gibi, çok eski bir tarihi geçmişe sahip olmasa da, KAA çok hızlı olarak gelişmiş ve yayılmıştır. KAA ilk defa, Şubat 1986 yılında düzenlenen Society of Automotive Engineers (SAE) kongresinde Robert Bosch GmbH tarafından açıklanmıştır. Gördüğü büyük ilgi karşısında 1997 yılında Intel ve Philips firmaları tarafından ilk KAA cipleri üretilmiştir. Fakat KAA'nda asıl atılım 1991 yılında Bosch tarafından KAA 2.0 versiyonunun açıklanması ile gerçekleşmiştir.



Şekil 1 KAA'na Bağlı Cihaz Sayısı ve Tarihsel Gelişimi

1992 yılında ise Mercedes-Benz henüz ISO tarafından standardı oluşturulmamasına rağmen KAA ciplerinin kullandığı araçları satışa sunmuştur. 1994 ve 1995 yıllarında ise ISO, 11898 ve geliştirilmiş 11898 standartlarını açıklayarak KAA genel kabul görmesine olanak sağlamıştır. ISO 11898 standardı temel olarak üç ana kısımda oluşmaktadır; birinci kısmı KAA protokolünü; ikinci kısmı yüksek hızlı (hatasız-tolerans) fiziksel seviyeyi; üçüncü kısmı ise düşük hızlı (hata-toleranslı) fiziksel seviyeyi tanımlamaktadır.

1995 yılından sonra KAA ciplerinin kullanımı katlamalı bir artış göstererek çoğalmış ve yaygınlaşmıştır. Genel KAA uygulamalarını gösteren Tablo 1. incelendiği takdirde bu artışın sebebi ve KAA'nın önemi daha iyi anlaşılacaktır. Tablo 1'den de anlaşılacağı gibi KAA kendini kanıtlamış ve insan hayatı ile ilgili olan çok önemli medikal uygulamalarda dahi rahatlıkla kullanılmaktadır.

## 2. KAA PROTOKOLÜ

KAA protokolü ISO 11898 uluslararası standartla tanımlanmıştır. Buna ek olarak KAA ciplerinin uygunluk testleri ISO 16845 tarafından tanımlanmış ve bir birlerinin yerlerine kullanılabilmesi garanti altına alınmıştır. KAA protokolü ISO Referans Katmanlarının ilk ikisi olan Fiziksel seviye (Physical

Layer) ve Veri Bağlantı (Data Link Layer) seviyesine karşılık gelmektedir.

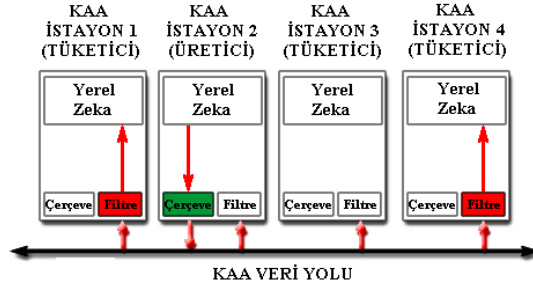
Tablo 1. Genel KAA Uygulamaları

Kullanım Alanı	Açıklama
Motorlu Araçlar	<ul style="list-style-type: none"> <li>- Yüksek hızda hatasız iletişime ihtiyaç duyan birimler. ABS, vites ayarı, motor kontrolü, devir-sıcaklık ayarı, hidrolik kontrolü vb...</li> <li>- Düşük hızda iletişime ihtiyaç duyan birimler. Açılır tavan, elektrikli camlar, hava yastığı, emniyet kemeri, merkezi kilit sistemi vb...</li> </ul>
İş Araçları	<ul style="list-style-type: none"> <li>- İnşaat araçları, forklif, traktörler vb...</li> <li>- Elektrikli trenler, hidrolik kontroller vb...</li> </ul>
Trenler	<ul style="list-style-type: none"> <li>- Özellikle hızlı veri alışverişlerine ihtiyaç duyan hız ayarı, fren sistemi, kapı kontrolü ve tren sinyalizasyon sistemi.</li> </ul>
Endüstri Otomasyon Cihazları	<ul style="list-style-type: none"> <li>- Kontrolör, sensör ve ivmelendiriciler gibi otomasyon cihazları için iyi bir bağlantı aracıdır.</li> <li>- İlk yol vermede, cihazlar için önemli bir program tutucu ve kalkış araçlarının haberleşmesi için uygundur.</li> <li>- Bir çok akıllı cihazın belirli bir sistem içerisinde yada alt sistemler olarak çalışmasına imkan tanır</li> <li>- Robot manipülatörler ve taşıma sistemlerinde</li> </ul>
Medikal Uygulama.	<ul style="list-style-type: none"> <li>- Bilgisayarlı tomografiler, röntgen cihazları ve diş hekimliği alanında.</li> </ul>
Ev Ürünleri	<ul style="list-style-type: none"> <li>- Çamaşır makinesi, bulaşık makinesi ve hatta kahve makinelerinde bile.</li> </ul>

Bu başlık altında KAA'nda veri alışverişinin, gerçek zamanlı veri iletişimi, mesaj çerçeve yapısı ve hat tespit ve işleme özellikleri detaylı bir şekilde açıklanacaktır.

KAA üzerinde veriler ağa yayın şeklinde iletilmektedir. Bu yöntem de, verinin ağ üzerindeki tüm cihazlara ulaşmasını sağlamaktadır. Söz konusu olan veri yayını sebebi ile veriler, mesaj temelli bir protokol sayesinde aktarılmaktadır. Fakat mesajlar ulaşılacak istasyon veya istasyonları tanımlamazlar sadece veriyi taşımakla yükümlüdürler. Mesajların ayrılması ise mesaj tanımlayıcıları sayesinde gerçekleştirilmektedir. Mesaj tanımlayıcıları tüm sistem için tektir ve sadece içerik belirtmezler mesaj

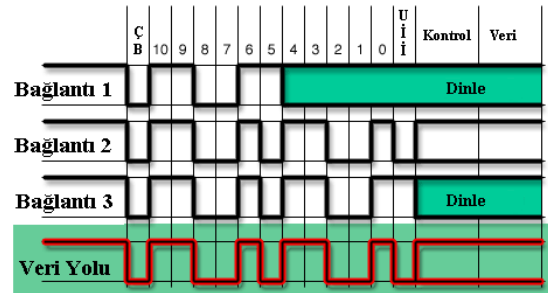
önceliğini de ihtiva ederler. Bu da veri yoluna giriş için yarışan istasyonlar için çok önem taşımaktadır. KAA üzerindeki temel veri iletişimi Şekil 2'de gösterilmektedir.



Şekil 2 KAA Üzerindeki Veri İletişimi

KAA yapılandırma esnekliğini ve yüksek dereceli sistemler oluşturabilme yeteneğini içerik temelli adresleme yapısına borçludur. Yine bu sebepten dolayı var olan KAA'na yeni bir bağlantı noktası eklemek için donanımsal yada yazılımsal değişikliğe ihtiyaç duyulmaz. Bu özellik ise KAA'na modüler bir yapılanma şansı tanımaktadır. Çünkü verinin kim tarafından üretildiği KAA içinde ki istasyonlar için hiçbir şey ifade etmez.

Gerçek zamanlı işlemlerde, mesajların aciliyeti ve önceliği çok ani değişiklikler gösterebilmektedir. Örneğin seyir halindeki bir aracın motor gücündeki değişiklik aracın diğer tüm parametrelerine göre çok daha fazla bir önem taşımaktadır ve süratle sistem içinde iletilmelidir. Bu sebepten dolayı mesaj öncelikleri mesaj tanımlayıcısı ile belirtilmiştir. Mesajların öncelik sırası sistem kurulurken belirlenmelidir. Çünkü KAA mesaj öncelik sırasının dinamik olarak değiştirilmesine izin vermez. KAA sistemlerinde en düşük binary sayıya sahip olan mesaj belirleyicisi en büyük önceliğe sahip olur. KAA veri yoluna girme şekli ve önceliği Şekil 3'de gösterilmektedir.

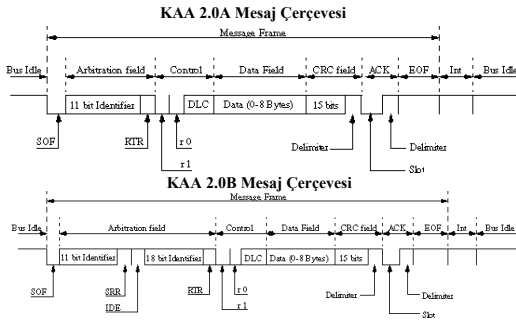


Şekil 3 KAA Veri Yoluna Girme

Veri yoluna giriş çakışmaları ise tanımlayıcılar üzerinde bulunan ve her istasyonun veri yolunu bit seviyesinde takibi sonucunda değerlendirebildiği akıllı-bit hakemliği sayesinde çözümlenmektedir. Veri yoluna giremeyen bağlantılar otomatik olarak daha yüksek önceliğe sahip birer alıcı durumuna geçerler ve veri yolu uygun seviyeye gelene kadar dinlemeye başlarlar.

KAA protokolü, sadece tanımlayıcı uzunluklarının farklı olduğu iki tip mesaj çerçeve formatının desteklemektedir. Desteklediği bu iki mesaj çerçevesi arasındaki tek fark ise bu mesaj çerçevelerinin tanımlayıcılarıdır (ID). Standart formata (KAA 2.0A) ID uzunluğu 11 bit, uzatılmış formatta (KAA 2.0B) ise ID uzunluğu 29 bittir. Şekil 4, KAA 2.0A ve KAA 2.0B çerçevelerinin ayrıntılarını vermektedir.

İletilecek mesaj çerçevesi 7 ana kısma ayrılmıştır. Standart formattaki mesaj (start of frame, "ÇB") biti ile başlar bu biti, mesajın bir veri çerçevesi mi yoksa veri içermeyen (remote frame) istek çerçevesi mi olduğunu belirten (remote transmission request, "UII") bitini de içeren tanımlayıcı "arbitration field" biti takip eder. Kontrol alanı "control field", mesajın Standart formata mı yoksa, extended formattaki olduğunu belirleyen ve kendinden sonra gelecek bitleri tekrar düzenleyen (son 4 biti ise veri alanı içindeki verilerin sayısını belirler) IDE (identifier extension) bitini içerir. Veri alanının (data field) büyüklüğü 0-8 bit aralığında olam zorundadır. Bu biti çerçeve güvenliği için hata kontrolü yapan "CRC field" biti takip etmektedir. "ACK" biti ise verinin tam olarak gönderildiğini kontrol için kullanılmaktadır. Mesajın sonu "end of frame" ile bitirilmekte ve mesajın bittiğini göstermektedir.



Şekil 4 KAA 2.0A ve KAA 2.0B Mesaj Çerçeveleri

İki Mesaj formatı arasındaki farklar aşağıdaki gibidir :

Versiyon 2.0B de hakem alanı iki belirteç bit alanı içerir. İlki (Ana Belirteç), versiyon 2.0A ile uyumlu 11 bit uzunluktadır. İkinci alan (Belirteç Ek), 18 bit uzunluktadır ve toplam olarak 29 bit uzunluk olur.

İki mesaj formatı arasındaki fark bir Belirteç Ek (IDE) biti kullanılarak yapılır.

Bir vekil uzak istek biti (SRR) Hakem alanındadır. Standart veri çerçevesi ve gelişmiş veri çerçevesi arasında hakemlik gerektiği zamanlarda SRR biti daima bir pasif bit olarak iletilir. Eğer her iki mesaj da aynı 11 bit belirtece sahipse standart veri çerçevesi daima önceliklidir.

Diğer veri yolu sistemlerinden farklı olarak, KAA protokolü bilgilendirme mesajları içermez bunun yerine hat oluştuğu zaman sinyaller oluşmaktadır.

Hata tespiti için KAA protokolü mesaj işleme seviyesinde üç ayrı mekanizma kullanır:

**Cyclic Redundancy Check (CRC) :** CRC çerçeve içindeki bilgiye, iletim sonuna kontrol bitleri ekleyerek koruyuculuk yapar. Alıcı konumu ise bu bitler test edilerek kendisi için tanımlanmış alıcı bitleri ile karşılaştırılırlar. Eğer aralarında benzerlik bulunmaz ise CRC hatası meydana gelir.

**Çerçeve Kontrolü :** Bu kontrol sırasında veri çerçevesinin orijinal büyüklüğü ve kısımları temel alınarak iletilen mesaj kontrol edilir. Bu kontrol sonucunda oluşan hata "format errors" olarak adlandırılır.

**ACK hatası :** Mesaj çerçevesi içerisindeki bu bit ise verinin tam olarak gönderildiğini kontrol için kullanılmaktadır. Yani kontrol, veri çerçevesi içerisinde gerçekleşir ve en etkili kontrol metodudur.

Bu kontrol mekanizmasına ek olarak KAA protokolü verileri bit seviyesinde de kontrol eden iki mekanizmaya sahiptir.

**Monitoring :** Bu yöntemde iletilen mesajların hatalarını kontrol etmek için veri yolu sinyalleri izlenir. İletim yapan her bağlantı yine veri yolu seviyesini kontrol eder yani gönderilen bit ve alınan bit arasındaki farkı gözler. Bu iletim sırasında oluşacak bütün global ve yerel hataları tespit eder.

**Bit stuffing :** Özel bitler kodları bit seviyesinde test edilmektedir. Bit kodlamada maksimum etkinlik sağlayan NRZ (non-return-to-zero) kodlama, KAA tarafından kullanılan bit ifade şeklidir.

Tek bir istasyonda bile yukarıdaki mekanizmalar sonucu bir veya birden fazla hata tespit edilirse yapılan iletim "error flag" yollanarak iptal edilir. Bu da mesajın başka bir istasyon tarafından alınmasını ve ağ içerisindeki veri doğruluğunun bozulmasını engeller.

Eğer her hangi bir istasyonun mesaj yollama isteği her hangi bir hata durumundan dolayı iptal edilirse yollayıcı tekrar iletme pozisyonuna "automatic repeat request" geçer. Bir kural olarak tekrar iletme durumu hata tespitinden 23 bit periyodu sonra başlar. Bazı özel durumlarda bu periyot 31'e çıkar.

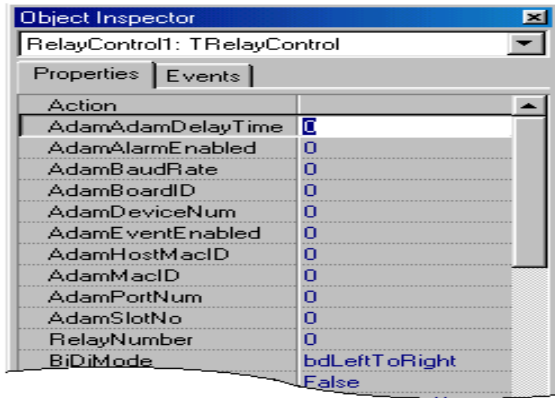
KAA'nın hata tespit oranı örneklenirse; KAA ile donatılmış bir cihazın, 500 kbps'lik KAA veri yolu hızı ve %25 veri yolu yoğunluğu ile yılda 2000 saat çalıştırılması sonucunda her 1000 yılda 1, tespit edilemeyen hata ortaya çıkacağı hesaplanmıştır. KAA'nın hata tespit oranının kalitesini açıklamak için verilen bu reel örnek sayesinde KAA'nın hata tespit oranının ne kadar etkin bir sistemi olduğu daha iyi ortaya çıkar.

### 3. DELPHİ BİLEŞENLERİ

Kontrolör Alan Ağı için delphi programı aracılığı ile hazırlanacak olan yazılımlarda kullanılmak üzere geliştirilmiş olan delphi bileşenleri nasıl hazırlandığı ve geliştirilen örnek uygulamanın Advantech firması tarafından geliştirilen ADAM 5000/CAN modülü üzerindeki uygulaması bu bölüm içerisinde açıklanacaktır.

Delphi programlama dili içerisinde yeni bir eleman yaratmak için temel olarak; Mevcut kontrolleri değiştirmek (Modifying existing controls), Pencereleşmiş kontroller oluşturmak (Creating windowed controls), Grafik kontrol oluşturmak (Creating graphic controls), Pencere kontrollerini alt sınıflara ayırmak (Subclassing Windows controls) ve Görüntülü olmayan elemanlar oluşturmak (Creating nonvisual components) olmak üzere 5 farklı yöntem vardır.

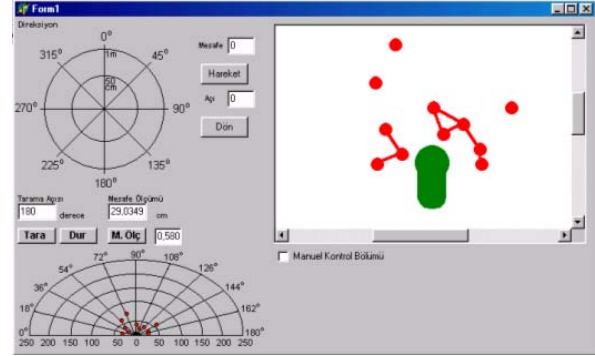
Geliştirilmiş olan Adam 5000/CAN bileşeni oluşturulurken mevcut kontrolleri değiştirmek yöntemi kullanılarak modül üzerinde bulunan dijital giriş, dijital çıkış, analog giriş, analog çıkış birimlerinin her biri için ayrı bir bileşen meydana getirilmiştir. Geliştirilen bu bileşenler yine Delphi programı tarafından sunulan derleme imanı ile bir araya getirilmiş ve kullanıcının hizmetine sunulmuştur. Şekil-5’de dijital çıkış birimi için hazırlanmış olan bileşenin ekran görüntüsü görülmektedir.



Şekil 5 Dijital Çıkış için Oluşturulan Bileşen

Geliştirilen Adam 5000/CAN Delphi bileşenlerinin bir araya getirilerek oluşturduğu Delphi Tool Box (bileşen kutusu) sayesinde, Delphi programlama dilinin sağladığı ve programcıya büyük bir rahatlık getiren, nesne tabanlı ( object-oriented ) programlama tekniğine uyum olarak, Adam 5000/CAN sisteminin DeviceNet protokolü aracılığı ile kontrolü sağlanmıştır. Geliştirilmiş olan tool box yardımı ile hazırlanmış olan Adam 5000/CAN kontrolünde kullanılan Veri Tabanlı Kontrol ve Gözetleme Sistemi'nin (Supervisory Control And Data

Acquisition, SCADA) örnek ekran çıktısı Şekil-6 görülmektedir.



Şekil 6 Örnek SCADA Uygulaması

### 3. SONUÇ

Gerçekleştirilen bu çalışma sayesinde KAA sistemlerinin kontrolü sırasında genel amaçlı olarak kullanılacak bir Delphi bileşeni oluşturulmuştur. Oluşturulan bu bileşen yardımı ile günümüzde ve gelecekte çok sık olarak kullanılacağı aşikar olan KAA uygulamaları için yeni bir geliştirmeye aracı ortaya çıkarılmıştır

### KAYNAKLAR

- [1] Berkay, A., Şeker, M., Esin, M., E., “An AGV Application with Controller Area Network”, Proceedings Of The Third International Conference On Mathematical & Computational Application, Selçuk Üniversitesi, Vol 3, Sep 4-6, 2002, pp127.
- [2] Şeker, M., Yıldırım, E., S., Berkay, A., “Use of Error Correction Network in Robot Path Planning”, Proceedings Of The Third International Conference On Mathematical & Computational Application, Selçuk Üniversitesi, Vol 3, Sep 4-6, 2002, pp211.
- [3] Berkay, A., “Automatic Navigation System For Automatic Guided Vehicle”, GYTE Yüksek Lisans Tezi, January 2001, 56-72.
- [4] CiA Draft Standard ., “CANopen Application Layer and Communication Profile”, CiA, Nov 1999.
- [5] International Standard Organization., “Road Vehicles - Controller Area Network (CAN) – Part 3: Fault Tolerant medium access unit”, ISO, ISO/CD11898-3, Nov 2001, ISO TC 22/SC 3/WG.
- [6] Hartwich, F., Bassemir, A., Robert Bosch GmbH, “The Configuration of the CAN Bit Timing”, Proceedings 6th International CAN Conference, Vol 6, Turin, June , 1999, pp211.