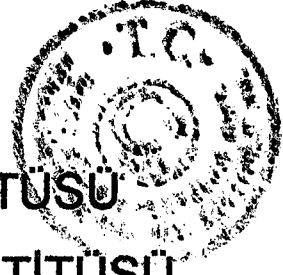


T.C.

GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ



OTOMATİK YÖNLENDİRİLEN ARAÇLAR İÇİN OTOMATİK HARİTALAMA SİSTEMİ

Ahmet BERKAY

109934

YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI

**T.C. YÜKSEK ÖĞRETİM KURULU
BİLİM, YAKAR ve FEN BİLİMLERİ ENSTİTÜSÜ**

TEZ DANIŞMANI
Doç.Dr. E. Murat ESİN

GEBZE

2001

Ahmet BERKAY'ın tez çalışması G.Y.T.E. Mühendislik ve Fen Bilimleri Sosyal Bilimler Enstitüsü Yönetim Kurulu'nun 20/8/2001 tarih ve 2001/22 sayılı kararıyla oluşturulan jüri tarafından Bilgisayar Anabilim Dalında YÜKSEK LİSANS Tezi olarak kabul edilmiştir.



JÜRİ

Üye
(Tez Danışmanı)

Doç.Dr. E. Murat ESİN

Üye

Yrd.Doç.Dr. Oğuz KUCUR

Üye

Yrd.Doç.Dr. Arif ERGİN

ONAY

GYTE Mühendislik ve Fen Bilimleri Enstitü Yönetim Kurulu'nun 13.09.2001 tarih ve 2001/24. sayılı kararı.



ÖZET



Bu tezde, Otomatik Yönlendirilen Araç (OYA) oluşturulurken çözülmesi gereken tasarım ve kontrol problemleri içerisinde yer alan “Yörünge planlama sistemi” ve “Çıkmaz yolların ve yörünge çakışmalarının belirlenmesi” problemlerinin çözüm kümesi içerisinde yer alan Otomatik Haritalandırma Sisteminin (OHS), Kontrolör Alan Ağı (KAA) üzerinden nasıl uygulanacağı pratik uygulama içerisinde açıklanmıştır.

Gerçekleştirilmek istenen uygulamanın pratik olarak uygulanabilir olması için öncelikle prototip bir OYA platformu oluşturulmuştur. Meydana getirilen OYA prototipi, X-Y koordinat sistemi içinde, ölçülebilir hareket etme yeteneğine sahip olacak şekilde dizayn edilmiştir.

Otomatik Haritalama Sistemi ve OYA kontrolü için en etkin çözüm olduğu düşünülen KAA ağının desteklediği Aygıt-Ağı (DeviceNet) bir SCADA yazılımı aracılığı ile OYA'nın OHS ve kontrol sistemleri oluşturulmuştur.

SUMMARY

In this thesis, A Automatic Navigation System (ANS), which gives the answers of the; "How Path planning system occurs?" and "Which strategies for resolving deadlocks and routing conflicts?" questions in the design and control problem encountered in an Automatic Guided Vehicle (AGV), were designed by using Controller Area Network (CAN) for AGV)

An AGV platform was made for experimental studies. The AGV platform can move in X-Y coordinate system and all the parameters of the AGV can be measured.

DeviceNet, which create the top layer of the CAN, was controlled by a Supervisory Control And Data Acquisition (SCADA) software to realize the ANS of the AGV.

TEŞEKKÜR

Bu tez konusunu seçmemde bana yardımcı olan, çalışmalarımda beni yönlendiren, tecrübelerinden büyük ölçüde faydalandığım kıymetli hocam sayın Doç.Dr. Emin Murat ESİN'e,

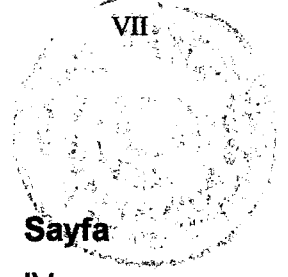
Tez çalışmam sırasında inkar edilemez desteklerinden dolayı sayın Doç.Dr.Fehmi Erzincanlı nezdinde bütün İmalat ve Tasarım bölümüne,

Madden ve manen her zaman arkamda duran, desteklerini hiçbir zaman benden eksik etmeyen ve etmeyecek olan sevgili aileme,

Ümitsizliğe düştüğüm her anda yanımda bulunan sevgili nişanlım Yük. Müh. Aynur ZEYNEL'e teşekkürlerimi bir borç bilirim.



İÇİNDEKİLER DİZİNİ



	Sayfa
ÖZET	IV
SUMMARY	V
TEŞEKKÜR	VI
İÇİNDEKİLER DİZİNİ	VII
SİMGELER VE KISALTMALAR	IX
ŞEKİLLER DİZİNİ	X
ÇİZELGELER DİZİNİ	XI
1. GİRİŞ	1
1.1. Otomatik Yönlendirilen Araçlar	1
1.1.1. OYA Tipleri	4
1.1.2. OYA İçin Yörünge Tayını	5
1.2. Tezin İçeriği	6
2. PROTOTİP OYA	8
2.1. Adam 5000/CAN Modülü	10
2.2. Pc-841 Dual Port Can Kartı	13
2.3. Ultrasonik Sensör	13
2.4. Kontrol Kartı	15
2.5. Oya Tasarımı	18
2.6. Prototip Oya'nın Görüntüsü	19
3. SÜRÜCÜLERİN KURULUMU	20
4. SCADA SİSTEMİNİN TASARIMI	24
4.1. Adam 5000/CAN İçin Geliştirilen Tool Box	25
4.2. Ohr İçin Geliştirilen SCADA Yazılımı	29
4.2.1. Yön Ayarlama	30
4.2.2. Tarama	35
4.2.3. Görüntüleme	41
4.2.4. Elle Kumanda	47
5. SONUÇ	50
KAYNAKLAR	52
ÖZGEÇMİŞ	54

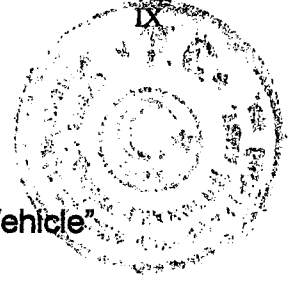
EKLER

EK-1

EK-2



SİMGELER VE KISALTMALAR DİZİNİ



AGV	Otomatik Haritalandırma Sistem "Automatic Guided Vehicle"
CAN	Kontrolör Alan Ağı (Controller Area Network)
CiA	"Otomasyonda CAN" Organizasyonu(CAN in Automation)
G/Ç	Giriş/Çıkış
SCADA	Veri Tabanlı Kontrol ve Gözetleme Sistemi (Supervisory Control And Data Acquisition)
I/O	(Giriş/Çıkış) Input/Output
UART	Genel Asenkron Alıcı Verici (Universal Asynchronous Receiver Transmitter)
LAN	Yerel Alan Ağı (Local Area Network)
KAA	Kontrolör Alan Ağı
WAN	Geniş Alan Ağı (Wide Area Network)
CSMA/CD	Carrier Sense Multiple Access/Collision Detection with
+AMP	Arbitration on Message Priority
ISO	Uluslar Arası Standart Organizasyonu (International Standard Organization)
OSI	Açık Sistem Bağlantısı (Open System Interconnection)
LLC	Mantıksal Bağlantı Kontrolü (Logical Link Control)
MAC	İletişim Araçları Ulaşım Kontrolü (Media Access Control)
SDO	Servis Veri Nesnesi (Service Data Object)
PDO	Proses Veri Nesnesi (Process Data Object)
SDS	Akıllı Dağıtık Sistem (Smart Distributed System)
LVDT	Lineer Değişken Diferansiyel Transformatör
DIP Switch	Grup ON/OFF Anahtar
PC	Kişisel Bilgisayar (Personal Computer)
OYA	Otomatik Yönlendirilen Araç
OHA	Otomatik Haritalandırma Sistem
DLL	Dinamik Bağlantı Kütüphanesi "Dynamic Link Library"
ISA	Endüstriyel Mimarlık Standartı "Industry Standard Architecture"
PID	Orantılı İntegral Türev "Proportional-Integral-Derivative"

ŞEKİLLER DİZİNİ



Şekil

1.1. Oya Tipi	4
2.1. Oya Tasarım Şeması	8
2.2. Oya Şematik Gösterimi	10
2.3. Adam 5000/Can Modülü Bağlantı Şeması.	11
2.4. Pcl-841 Kartı Can Portu Pin Bağlantıları Katlıma Topografyası	13
2.5. Ultrasonik Sensör Ölçüm Kartı	14
2.6. Polaroid Ultrasonik Sensör Ve Tarama Menzili	14
2.7. Kontrol Kartı	17
2.8. Motor Kontrolü İçin Akış Diyagramı	18
2.9. Mesafe Ölçümünü Sağlayan Encoder Kesiti Ve Ölçüleri	18
2.10. OYA'nın Fiziksel Görüntüsü.	19
3.1. Adam 5000/Can Modülü İçin Gerekli Olan Ayarlar	22
3.2. Advantech Pcl-841 Can Bus Card Setup	22
3.3. Dijital Giriş Modülünün Adam 5000/Can İçin Ayarlanması	23
4.1. New Component Ekranı	26
4.2. Programlama sırasında Girilmesi gerekli Parametreler	28
4.3. Veri Giriş Bölümlerini Object Inspector İçin Belirlenmesi	28
4.4. Onmousedown Prosedürleri	29
4.5. Oya'nın Doğrultu Kontrol Prosedürünün Akış Diyagramı	30
4.6. Oya Doğrultu Kontrol Bölümü	35
4.7. Yönlendirme Sonucu	35
4.8. Oya'ın Tarama Bölümü İçerisindeki Akış Diyagramı	36
4.9. Engelin Yer ve Konumunu Tayin Eden Tarama Bölümü	41
4.10. Oya'ın Görüntüleme Bölümü İçerisindeki Akış Diyagramı	42
4.11. Görüntüleme Bölümü İçerisinde Oya Ve Engel Çıktıları	47
4.12. Elle Kontrol Bölümü İçerisindeki Akış Diyagramı	48
4.13. Elle Kontrol Bölümü İçerisinde Oya Hareketi Ve Hareket	49

ÇİZELGELER DİZİNİ

Çizelge

3.1. PCL-841 Kartı İçin Sinir Değerler Ve Geçerli Değerler



1 GİRİŞ

Günümüz teknolojisi içerisinde uygulanan esnek ve rekabetçi üretim sistemleri, tekrarlanan işlemler sırasında meydana gelecek olan hataları yok etmek ve bu işlemlerin süresini kısaltmak amacı ile insan faktörünü devre dışı bırakan Otomatik Yönlendirilen Araç (OYA), otomasyon sistemi içerisinde sıkça kullanılmaktadır.

OYA, gerekli malzemenin, ihtiyaç duyulan yere taşınması sırasında belirli bir yörüngeye ihtiyaç duyarlar. OYA yörüngeyi belirlemek için bir çok değişik yöntemler kullanırlar.

OYA çevrelerine ve taşıdıkları yüke, bu taşıma işlemi süresince izlediği yörünge içerisinde zarar vermemek zorundadırlar. Bu sebepten dolayı izlediği yörünge içerisinde çevreleri ile devamlı olarak etkileşim halinde olmalıdırlar.

Bu tez çalışması içerisinde OYA'nın, kendi yörüngesini değişen ortama göre tekrar düzenlemesini sağlayan Otomatik Haritalama Sistemi'nin (OHS) ve kontrolünün, dağıtık veri toplama sistemlerinin kontrolünde kullanılan ağ teknolojilerinden biri olan Kontrolör Alan Ağı (KAA) üzerinde nasıl gerçekleştirdiği açıklanmaktadır.

1.1 Otomatik Yönlendirilen Araçlar

Otomatik Yönlendirilen Araçlar 1950'lerden beri kullanılmaktadır. OYA temel olarak insan faktörünü sistemden çıkaran, sürücüsüz hareket edebilen araçlar olarak tanımlanabilirler. OYA sistemleri, yapılabilmesi için çok büyük güçlere ihtiyaç duyulan, genelde tekrarlanan, basit karar verme, "insan ile kıyaslandığı taktirde" inisiyatifine sahip olması istenen alanlarda kullanılması uygun sistemlerdir.

Kullanılan ilk OYA , “sürücüsüz traktör” ismi ile Barret Electric tarafından 1953’de geliştirilmiştir. Geliştirilmiş olan bu araç, çekici bir traktörün fabrika tavanına döşenen bir kılavuz telini takip etmesi prensibine dayanmaktaydı. 1960 ve 1970’lerin başına kadar OYA bu konseptin dışına çıkmamışlardır. OYA için atılım denecek nokta ise 1970’lerin başında Avrupa’da geliştirilerek Amerika’da kullanılmaya başlanan birim taşıyıcı tipindeki OYA sayesinde yaşanmıştır. OYA tarih incelendiği zaman elektronik ve bilgisayar sektöründe ki ilerlemeye paralel olarak üç farklı nesil olduğu gözlenmektedir.

İlk OYA nesli 1950 ve 1960’ların sonuna kadar kullanılmıştır. Bu nesil içerisinde bulunan OYA tipleri, zamanının teknik imkanları sebebi ile motorlu veya elektrikli bir taşıtın yere döşenen ve taşıtın gideceği yolu belirleyen bir kılavuz teli takip etmesi mantığına dayanmaktadır. Bu nesil içerisindeki OYA ‘lar sabit bir yolu, sabit bir yörünge çerçevesinde, çevreleri ile hiçbir etkileşimleri olmadan katetmekteydiler. Bu sebepten dolayı genel proses içerisinde hiçbir esnekliğe sahip değillerdi. AGV Electronics , CCRI ve NDC Automation, Inc. firmaları tarafından üretilen OYA tipleri bu nesil için iyi birer örnek teşkil etmektedirler.

1960’ların sonundan 1970’lerin sonlarına kadar kullanılan ikinci nesil olarak adlandırabileceğimiz OYA tipleri ise yine ilk nesil gibi sabit bir kılavuzu takip ederek yol almaktadırlar. Fakat bu nesildeki OYA tipleri çevreleri ile belirli kıstaslar içerisinde etkileşim özelliğine sahiptirler. Bu etkileşim sayesinde genel sistem içerisinde belli bir esnekliğe kavuşmuşlardır. Bu tip OYA için en iyi örneği AGV Electronics , FMC ve ESKAY Corporation firmalarının ürettiği OYA tipleri oluşturmaktadır.

Elektronik ve bilgisayar teknolojisinde 1980’lerin başında meydana gelen atılım OYA neslini 3. kuşağa taşımıştır. Halihazırda kullanılan OYA tipleri bu nesil içerisinde yer almaktadır. 3. kuşak OYA tipleri çevreleri ile bire bir etkileşimli karar verme yetenekleri yüksektir. Kullanıldıkları proses içerisinde en yüksek esnekliğe sahip ola nesildir.

Günümüz teknolojisi içinde OYA geleneksel taşıma sistemleri olan fortlif ve konvoyörlerin yerini almaya aday sistemlerdir. OYA’ler yüksek güçlü piller

sayesinde enerjilendirilen, yönetilebilir motorlar sayesinde hareket ettirilebilen tekerleklerle sahip ve önceden belirlenmiş bir yörünge ile belirli bir yolu alan araçlardır (Tompkins, 1996).

OYA sahip oldukları esneklik, çok yönlülük ve yapılacak işlerin maliyetlerini düşürmelerine ek olarak bir bilgisayar ile kontrol edilmeleri sebebi ile çevreleri ile gerçek zamanlı bir etkileşime sahiptirler (Groover 1987). Bu özellikler OYA'nın her türlü otomasyon sistemi içerisine kolay adapte edilme imkanı sağlar.

Her OYA üzerinde bir kontrolör taşır. Bu kontrolör , mekanik ve elektriksel donanımı ana bilgisayara tanıtmak için bir ara yüz görevini üstlenir. Kullanılan bu kontrolörlerin programlanabilir olması OYA'nın değişen programlar içinde değişik görevleri yerine getirmesine imkan verir.

OYA sistemi otomasyon prosesi için seçilmesi ve işletilmesi aşağıdaki sonuçları meydana getirir :

Esneklik: OYA'nın sistemlerinde yörünge ve gidilecek son nokta rahatlıkla tekrar değiştirilebilir ve yenilenebilir.

Haberleşme: OYA'nın kontrol ve karar verme işlevlerini yerine getiren bilgisayarlar sayesinde bulundukları ortam ve diğer sistemler ile rahatlıkla haberleşebilir ve eş uyumluluk gösterebilirler.

Alan Tassarufu: OYA sistemlerin yapısal olarak en belirgin özelliğidir. OYA sistemlerinin sağladığı en büyük özelliklerden biri de kullandığı alanı diğer birimlerle paylaşabilmesidir.

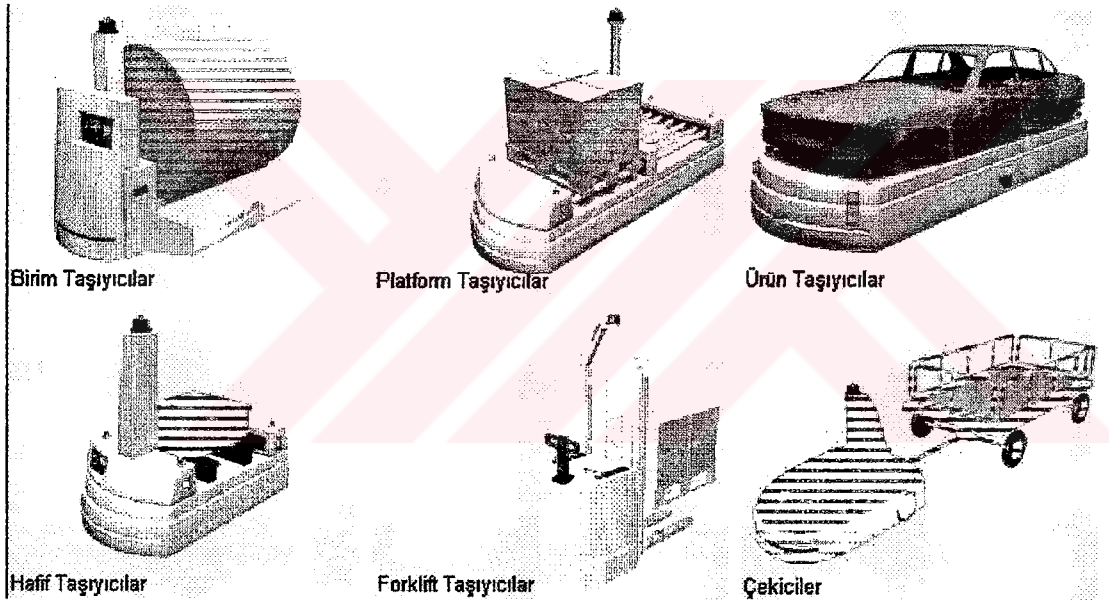
Bir OYA sistemi oluşturulurken bazı tasarım ve kontrol problemlerinin çözülmesi gerekmektedir. Çözülmesi gereken bu tasarım ve kontrol problemleri genel bir görüş altında şu şekilde sıralanabilir (Heragu 1996):

- Taşınması istenen malzeme için OYA tespiti,

- Malzeme akış ağı içerisinde, malzeme alma ve bırakma noktalarının belirlenmesi,
- OYA sayıları ve hareket alanları,
- OYA'nın yörünge ve hızlarının belirlenmesi,
- Yörünge planlama sistemi,
- Çıkmaz yolların ve yörünge çakışmalarının belirlenmesi.

1.1.1 OYA Tipleri

Endüstride çok değişik amaç ve biçimlerde OYA kullanılmaktadır. Şekil.1.1 bunlara ilişkin örnekleri göstermektedir.



Şekil 1.1. OYA tipi

Birim Taşıyıcılar: Bu tip OYA belirli tipten yüklerin kullanıldığı ortamlarda kullanılır ve taşıyacağı malzemenin özelliklerine göre tasarlanır.

Platform Taşıyıcılar: Bu tip OYA için taşıyacakları ürünün ağırlığı dışına hiçbir kısıtları yoktur çünkü bu tip OYA yük için değil yükleri yükleneceği platformları taşıyabilmek için tasarlanmıştır.

Ürün Taşıyıcılar: Bu tip OYA yörünge çevreleri ile en çok etkileşen tip OYA'dır. Çünkü bu tip OYA ürünü başlangıçtan son şeklini alıncaya kadar bir üretim bandı platformu olarak taşımak üzere tasarlanmıştır.

Hafif Taşıyıcılar: Bu tip OYA genel amaçlı tasarlanmış taşıyıcılardır. Tek kistası ağırlık olmak koşulu ile her tür malzemeyi taşımak üzere tasarlanmıştır.

Forklift: Taşıyıcılar: Bu tip OYA diğer tiplerden ayıran en önemli farkı taşıdıkları yükü altına girip kaldırarak üç boyut içerisinde hareket ettirebilme özelliğidir.

Çekiciler: Bu tip OYA diğer tiplerin aksine iletecekleri malzemeyi çekerek gidecekleri yere ulaştırırlar.

Bu tez çalışması; Hafif Taşıyıcı tipi içerisine girilebilecek prototip OYA üzerinde gerçekleştirilmiştir.

1.1.2 OYA İçin Yörünge Tayini

OYA'nın izlediği yörünge çok basit bir döngü olabileceği gibi çok karmaşık bir ağa da olabilir, izledikleri bu yörünge tekrar değiştirilebilir olmak zorundadır (Tompkins 1996). OYA sistemleri, araçların hareketlerini kontrol etmek için "tasarlanış şekillerine bağlı olarak" değişik yörünge takip etme sistemlerine sahip olabilirler. Bu sistemler içerisinde temel teşkil edecek olanlar ise aşağıda belirtilmektedir.

Elektromagnetik yörünge izleme sistemleri temel olarak yere döşenen bir kablo ve bu kablonun yaydığı radyo sinyallerinin izlenmesine dayanmaktadır. Kablo tarafından yayılan bu sinyaller OYA üzerinde bulunan antenler sayesinde algılanır ve gelen sinyallerin büyüklüğü OYA'a yön tayini için bir veri olarak değerlendirilir.

Optik yörünge izleme sistemleri ise gidilecek yol üzerine döşenen ışık kaynaklarının takibine dayanmaktadır. Folyo bant, ultraviyoleye hassas bant veya kimyasal boyalar yörünge tayininin belirlenmesi için kullanılmaktadır. Bu tip sistemlerde ise AVG gideceği yön üzerinde ışığın varlığını veya yokluğunu kontrol eder. En ekonomik, kurulumu ve kullanımı kolay yöntemdir.

Bilgisayar teknolojisinin gelişmesi ile birlikte bu tür araçlar üzerinde karmaşık kontrol birimleri kullanmak ve ileri özelliklere sahip OYA'lar gerçekleştirmek mümkün olabilmektedir. Böylece kendi yörüngesini "keşfederek" belirleyebilen, çevresinin bir tür "haritasını" (OHS, Otomatik Haritalama Sistemi) çıkartabilen, optimal yol çözümünü hesaplayabilen ve bu yol üzerinde önceden belirlenmemiş engellere rastladığında alternatif yollar deneyebilen sistemler ortaya çıkmıştır.

1.2 Tezin İçeriği

Sunulan tezde temel olarak, Bölüm 1.1'de detaylı olarak açıklanan ve Heragu tarafından ortaya konan ; bir OYA sistemi oluşturulurken çözülmesi gereken tasarım ve kontrol problemleri başlığı temel alınmıştır. Bu başlık altında bulunan "Yörünge planlama sistemi" ve "Çıkmaz yolların ve yörünge çakışmalarının belirlenmesi" adımlarının çözümü için OHS'nin, KAA üzerinden nasıl uygulanacağı pratik uygulama içerisinde açıklanmıştır.

Gerçekleştirilmek istenen uygulamanın pratik olarak uygulanabilir olması için öncelikle prototip bir OYA platformu oluşturulmuştur. Meydana getirilen OYA prototipi, X-Y koordinat sistemi içinde, ölçülebilir hareket etme yeteneğine sahip olacak şekilde dizayn edilmiştir.

Otomatik Haritalama Sistemi'ni oluşturmak için çevresel veri ve etkileri elektronik olarak OYA sistemine taşıma ihtiyacı, ultrasonik bir sensör aracılığı ile gerçekleştirilmiştir. Kullanılan ultrasonik sensörün, ölçülmesi istenen bölge içerisinde kullanılmasını sağlamak için de ultrasonik sensör hareketli hale getirilmiştir.

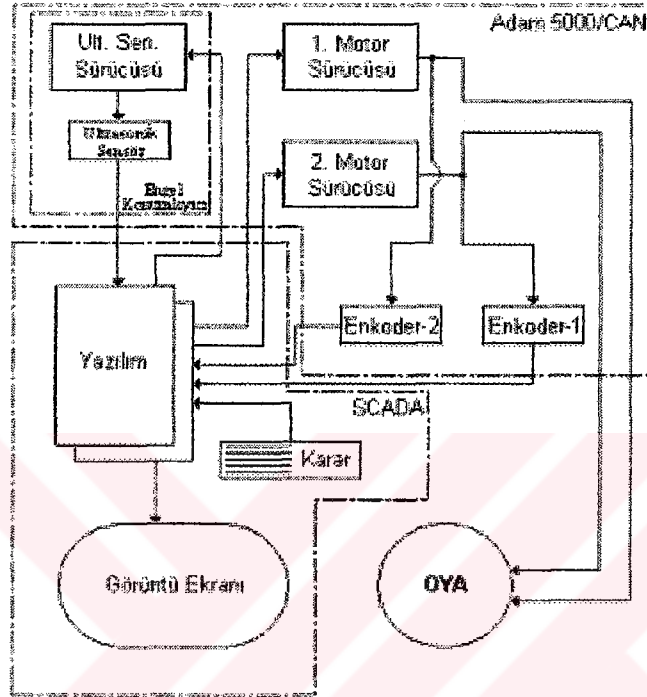
Otomatik Haritalama Sistemi ve OYA kontrolü için en etkin çözüm olduğu düşünölen KAA ağıının desteklediğı Aygıt-Ağı (DeviceNet) bir SCADA yazılımı aracılığı ile OYA'nın OHS ve kontrol sistemleri oluşturulmuştur.



2 PROTOTİP OYA



Genel anlamda bir OYA konsepti oluşturmak için ihtiyaç duyulan tasarım Şekil 2.1’de verilmektedir.



Şekil 2.1. OYA Tasarım Şeması

Bu tez çalışması içinde; OYA'ın, yörüngesini değişen ortama göre tekrar düzenlemesini sağlayan OHS'nin ve kontrolünün, (dağıtık veri toplama sistemlerinin kontrolünde kullanılan ağ teknolojilerinden biri olan) KAA, üzerinde yapılması temeline dayanmaktadır. OYA tasarım aşamaları aşağıda ifade edildiği gibi gerçekleştirilmiştir.

Adam 5000/CAN: Tez çalışmasının omurgasını oluşturan bu bölüm KAA ile gerçekleştirilmiştir. Bu işlem KAA protokolünü destekleyen ve Advantech firması tarafından üretilen Adam 5000/CAN Modülü sayesinde yapılmıştır. OYA tasarımı içerisinde bir birlerinde farklı giriş ve çıkış sinyalleri üreten ve tüketen birimlerin olması sebebi ile bu cihazlar KAA protokolünün bir üst katmanı olan DeviceNet tarafından bir ağ içerisinde bir araya getirilmişlerdir.

SCADA: OYA'yı oluşturan tüm kontrol ve ölçüm cihazlarından gelen veriler bir PC üzerinde bulunan SCADA yazılım aracılığı ile gerçekleştirilmektedir. Veri Toplama Birimi aracılığı ile toplanan tüm veriler Advantech firması tarafından üretilen PCL-841 kartı üzerinden PC'ye iletilmektedir.

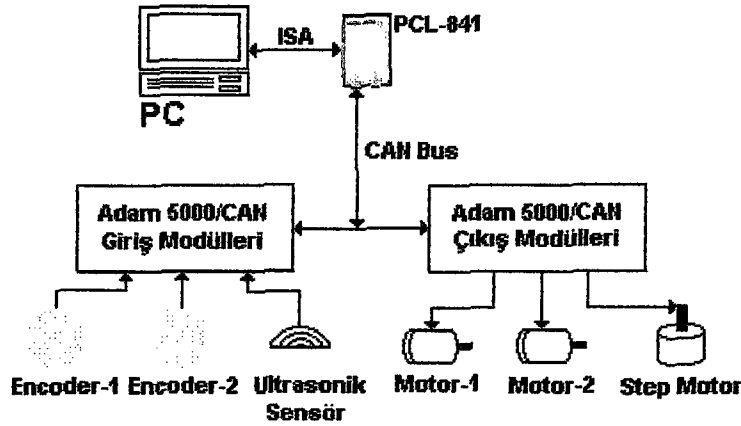
Engel Konumlayıcı: OHS'ni oluşturmak için gerekli veri bir ultrasonik sensör tarafından sağlanmaktadır. Kullanılan ultrasonik sensör, OYA tarafından gözlenmek istenen bölge içerisinde ölçüm yapılmasını sağlama amacı ile hareketli hale getirilmiştir. Ultrasonik sensör bir adım motoru aracılığı ile hareketli hale getirilmiştir.

Sürücüler: Tasarlanan OYA'ı hareketli hale getirebilmek için redüktörlü doğru akım motorları kullanılmıştır. Bu tip motorların seçilmesinin temel sebebi enerjisiz durumda iken kitli kalmalarıdır. Böylelikle tasarlanan OYA, eğimli yüzeylerde rahatlıkla hareket etme imkanı kazanmaktadır. Redüktörlü motor kullanımının sağladığı diğer bir avantaj ise motorlara verilen güç ile alınan tork arasındaki bağlantıdır.

Enkoderler: Tasarlanan OYA'nın ne kadar ve ne yönde hareket ettiği, hareketi sağlayan motor milleri üzerine kuplajlanan enkoderler sayesinde ölçülmektedir. Kullanılan enkoderler bir tam dönüşte 36 sinyal verecek şekilde tasarlanmışlardır.

Güç Ünitesi: Genelde Endüstriyel OYA'lar Bölüm 1.1.1'de de incelendiği gibi güç ünitelerini kendi tasarımları içerisinde barındırmaktadırlar. Fakat güç ünitesinin tasarlanan OYA prototipi içinde yer alması sisteme ekstra yük getireceği için dışarıda bırakılmıştır. OYA enerjisini harici bir güç kaynağından kablo vasıtası ile almaktadır. Adam 5000/CAN modülü 24 V üzerinden, ultrasonik sensör 12 V üzerinden, hareketi sağlayan motorlar ve enkoderler 6 V üzerinden beslenmektedir.

Tasarlanan OYA'nın tasarım aşamaları, tez çalışmasının içeriğine dayandırılarak doldurulduğu zaman OYA prototipinin şematik gösterim şeması meydana çıkar (Şekil 2.2).



Şekil 2.2. OYA Şematik Gösterimi

Bu bölüm içerisinde geliştirilmiş olduğum OYA sistemi içerisinde kullanılan fiziksel enstrümanlar; PCL-841 haberleşme kartı, Adam 5000/CAN modülü, ultrasonik Sensör, Adam 5000/CAN modülüne ile mekanik kısım arasındaki geçiş kartı ve mekanik kısmın; prototip OYA içerisinde kullanımı hakkında detaylı bilgi verilmiştir.

2.1 Adam 5000/CAN Modülü

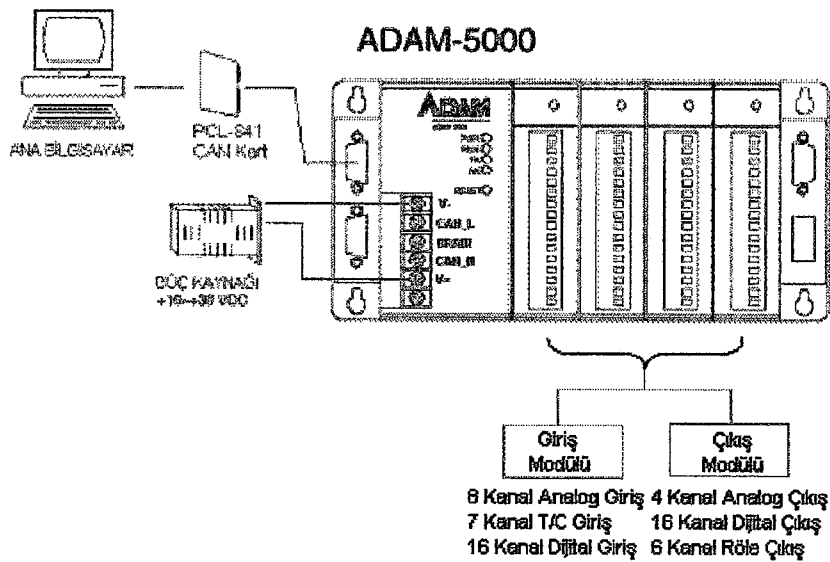
Hazırlanmış olan tezin bel kemiği, Adam 5000/CAN modülü üzerine oturtulmuştur. Adam 5000/CAN modülü özellikle geniş bir kontrol ağı içerisinde bulunan aletlerin, gerçek zamanlı uygulamaları sırasında, ihtiyaç duyacakları veri toplama ve kontrol işlevini sağlaması için Advantech firması tarafından geliştirilmiştir. Adam 5000/CAN modülü uzaktaki bir sunucu bilgisayar yardımı ile KAA protokolü üzerine oturtulmuş DeviceNet veya CANopen üst katmanları üzerinden kontrol edilebilmektedir. Adam 5000/CAN modülü veri ve kontrol sinyallerini KAA protokolü üzerinden iletilmektedir.

Adam 5000/CAN sistemleri küçük, ihtiyaca göre tekrar yapılandırılabilir esnek bir yapıya sahiptir.

Adam 5000/CAN modülünün temel kısımları aşağıdaki gibidir :

- Adam 5000/CAN ana ünitesi. Ana ünite bir CPU kartı, bir güç regülatörü, 4-slot, bir RS232 haberleşme portu ve bir çift DB9 CAN portunu kapsar.
- Giriş Çıkış Arabirimi. Adam 5000/CAN ünitesi üzerinde bulunan CPU 64 G/Ç birimini 4 slot üzerinden desteklemektedir.
- G/Ç birimleri. Adam 5000/CAN modülünün en gelişmiş kısmı G/Ç birimleridir. Bu birimler isteğe göre yer, veri alış büyüklüğü ve giriş-çıkış veri çeşidi değiştirilebilmektedir. Analog girişler birimi 16 bitlik bir çözünürlüğe sahiptirler. Dijital giriş-çıkış birimi ise programlanabilir özelliğe sahiptir.
- Yazılım desteği. Adam 5000/CAN ünitesi hem DeviceNet hem de CANopen protokollerini destekleyen DLL (Dynamic Link Library) dosyalarına sahiptir.

Şekil 2.3 Adam 5000/CAN modülünün bağlantı şemasını içermektedir.



Şekil 2.3. Adam 5000/CAN Modülü Bağlantı Şeması.

OYA tasarımı içerisinde bulunan ve birbirlerinde tamamı ile farklı giriş ve çıkış sinyalleri üreten ve tüketen birimlerin tümü Adam 5000/CAN modülü yardımı ile denetlenip kontrol edilmektedir.

Kullanılan Adam 5000/CAN ünitesi üzerindeki ilk slota yerleştirilen ADAM-5017 modülü (bkz. Ek-2), OYA kontrolünü sağlayan SCADA yazılımı aracılığı ile 0-5 volt değerleri arasında ölçüm yapacak şekilde ayarlanmıştır. Bu modül aracılığı ile OHS'nin oluşturulması için gerekli veri girişini sağlayan ultrasonik sensörün ölçüm işlemi sonucu çıktı olarak verdiği gerilim, bu modülün birinci kanalı üzerinden SCADA yazılımına aktarılmaktadır.

Adam 5000/CAN ünitesi içerisinde ikinci sırada yerleştirilen ADAM-5024 modülü (bkz.Ek-2), OYA sisteminin hareketini sağlayan doğru akım motorlarını SCADA yazılımı aracılığı ile sürmek için kullanılmaktadır. Bu sürme işlemi için gereken güç ADAM-5024 tarafından direkt olarak sağlanamamaktadır (bkz. Ek-2). Bu sebepten dolayı motorlar, tasarlanan sürücü kartı üzerindeki röleler aracılığı ile gerçekleştirilmektedir. ADAM-5024 modülünün birinci gerilim çıkışı motorları ileriye doğru sürmekte, ikinci gerilim çıkışı motorları geriye doğru sürmektedir. OYA sistemini sağa veya sola dönüşü ADAM-5024 modülünün üçüncü ve dördüncü gerilim çıkışları sayesinde sağlanmaktadır. Yön değiştirme işlemi dönecek yöndeki tekerleğin enerjilendirilmemesi, aksi taraftakinin dönüş açısını sağlayacak kadar enerjilendirilmesi ile gerçekleştirilir.

ADAM-5051 modülü (bkz.Ek-2), Adam 5000/CAN ünitesi içerisinde üçüncü slot üzerinde bulunmaktadır. Kullanılan bu modül sayesinde OYA'ı hareketli hale getiren motorların katettiği mesafeyi ölçen enkoderin gönderdiği sinyaller ölçülmektedir. ADAM-5051 modülü birinci dijital girişi sağ teker enkoderi, ikinci dijital girişi sol teker enkoderi olarak kullanılmaktadır. ADAM-5051 modülü gelen sinyalleri dijital giriş uçlarının 0 Volta çekilmesi sayesinde algılar.

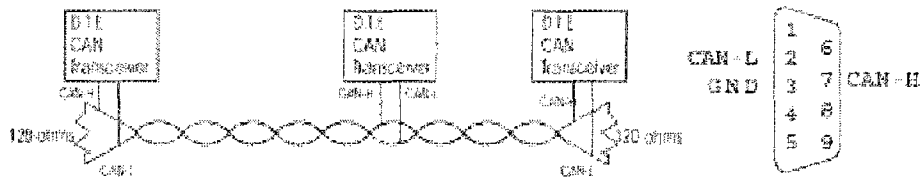
Adam 5000/CAN ünitesinin son modülü olan ADAM-5061 modülü (bkz.Ek-2) OHS sisteminde genelini yerini belirleyen polar koordinatların iki

değerinden biri olan açı değerini elde etmek için kullanılır. Açı koordinatını sağlayan adım motoru, bu modül sayesinde sürülür. Bipolar tipinde olan adım motoru dönebilmesi için gerekli olan enerjiyi bu modül üzerinden sağlar. adım motorun sargıları bu bölüm üzerinde enerjilendirilir.

2.2 PC-841 Dual Port CAN Kartı

PCL-841 kartı herhangi bir PC' yi KAA bağlayabilmek için kullanılan ISA (Endüstriyel Mimarlık Standartı "Industry Standard Architecture") tipi bir karttır. PCL-841 kartı içerisinde kurulu olan CAN kontrolörü, hata ayıklama ve veri hakemliğini iletişim sırasında otomatik olarak yapar. Bu özellikleri ile veri kayıpları ve sistem gerçekliğini sağlar. Kart üzerinde bulunan hafızanın değişik yerlerine konumlandırılmış bu kontrolörleri aynı anda ya da farklı olarak çalıştırmak mümkündür. PCL-841 kartı, 1 Mbps baud-rate hızına kadar iletişim yapabilmektedir.

CAN standartları half-duplex haberleşme standardını desteklemektedir. Adam 5000/CAN modülünün, PC ile haberleşmesini sağlayan PCL-841 kartının pin bağlantıları ağa katılım topolojisi Şekil 2.4'de verilmektedir.



Şekil 2.4. PCL-841 Kartı CAN Portu Pin Bağlantıları ve Katılım Topografyası

2.3 Ultrasonik Sensör

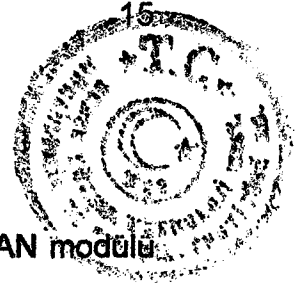
OYA sistemi içinde ultrasonik sensör olarak Polaroid ultrasonic sensöre kullanılmıştır. Fakat Polaroid Ultrasonic sensör, ölçtüğü mesafe bilgisini gerilim cinsinden vermemektedir. Bu sensörden aldığı veriyi işleyerek gerilim çıktısı olarak vermesini sağlayacak bir ara kartla desteklenmiştir.



The top section of the drawing contains three views of a mechanical component. On the left is a cross-sectional view showing a circular part with a central hole and a flange. In the middle is a side view showing the component's profile with dimensions: a total width of 1.000 (3/16), a central hole diameter of .125, and a flange thickness of .125. On the right is a top view showing the circular face with a central hole and a flange, with dimensions: a total diameter of 1.000 (3/16) and a central hole diameter of .125.

The bottom section of the drawing is a detailed cross-sectional view of a semi-circular component. It shows a semi-circular shape with a central hole and a flange. The drawing includes numerous concentric lines and radial lines, indicating a complex internal structure. The central hole is labeled with a dimension of .125. The overall shape is semi-circular, with a flat base and a curved top.

Şekil 2.6. Polaroid Ultrasonik Sensör Ve Tarama Menzili



2.4 Kontrol Kartı

OYA sistemi içerisinde ana kontrol birimi olan Adam 5000/CAN modülü ile kontrol edilecek motorlar, edcoder ve diğer elektriksel bağlantılar arasında hem Adam 5000/CAN modülünün bu aletler için gerekli gücü tek başına verememesinden hem de aletin prototip olma özelliğinden dolayı sürekli değiştirilebilecek kablo bağlantılarının daha rahat yapılabilmesi için bir ara yüz kartı tasarlanmıştır. Tasarlanmış olan bu kart ile OYA sistemi içinde kullanılacak aletler için gerekli olan güç sağlanmıştır.

Şekil 2.1 içinde verilen ve CD-56 bölgesi içerisinde gösterilen sürücü devresi sayesinde Sağ motor kontrol edilmektedir. Bu sürücü klasik bir H-bridge sürücü devresidir. AO1 ve AO2 ile belirtilen uçlar yerleştirilen ADAM-5024 modülünün (bkz Ek-2) birinci ve ikinci analog çıkışlarını ifade etmektedir.

AB-56 bölgesi içerisinde gösterilen sürücü devresi ile sol motor kontrol edilmektedir. Bu sürücü de klasik bir H-bridge sürücü devresidir. AO3 ve AO4 ile belirtilen uçlar yerleştirilen ADAM-5024 modülünün (bkz Ek-2) üçüncü ve dördüncü analog çıkışlarını ifade etmektedir.

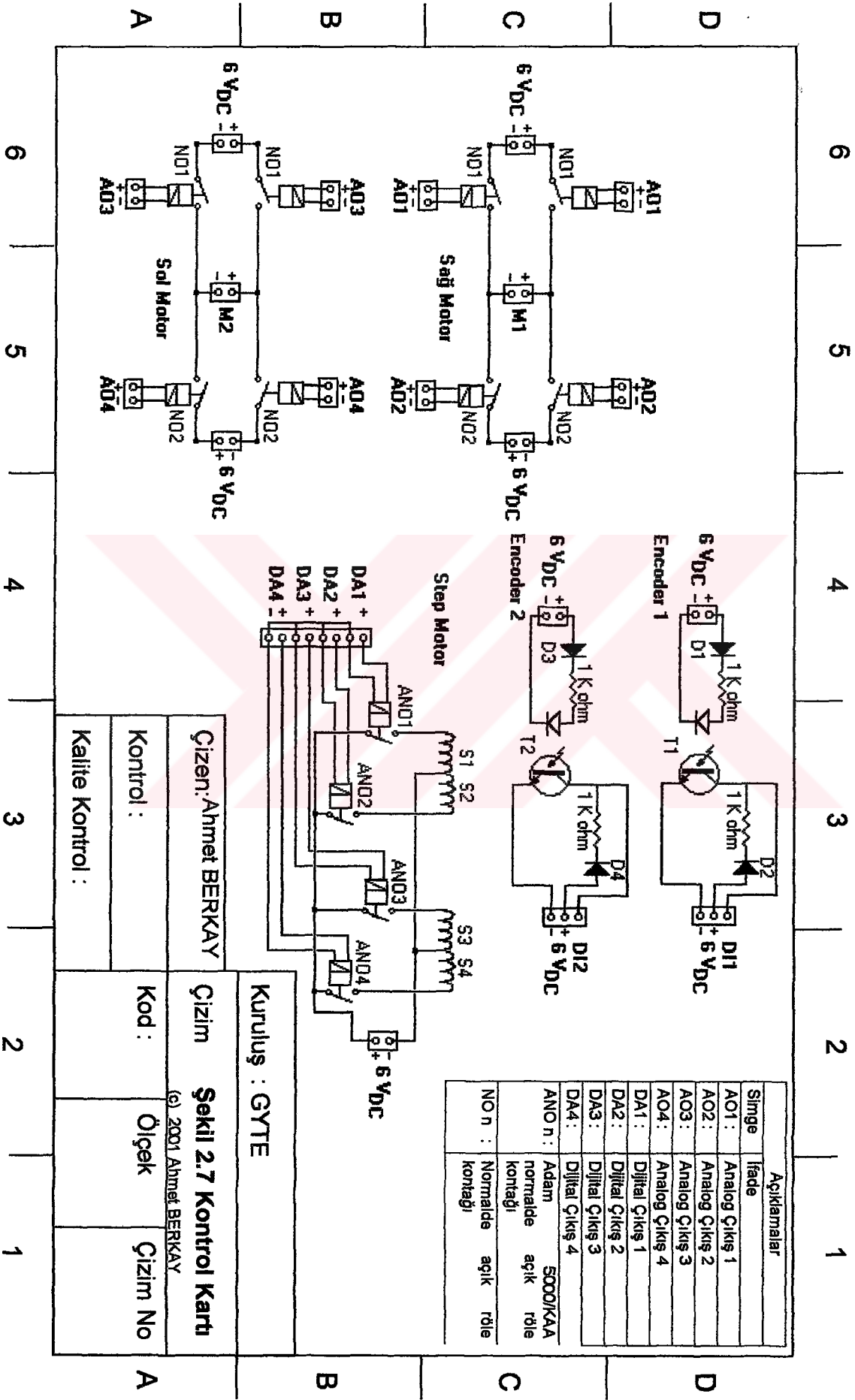
C-34 bölgesi içerisinde gösterilen optik enkoder devresi sayesinde Sol motorun kat ettiği yol ölçülmektedir. Oluşturulan bu optik-enkoder yardımı ile ADAM-5051 modülünün (bkz.Ek-2), dijital girişleri için sinyal üretilmektedir. DI1 ile ifade edilen ADAM-5051 modülünün birinci dijital girişini ifade eder.

D-34 bölgesi içerisinde gösterilen optik enkoder devresi sayesinde sağ motorun katettiği yol ölçülmektedir. Oluşturulan bu optik-enkoder yardımı ile ADAM-5051 modülünün (bkz.Ek-2), dijital girişleri için sinyal üretilmektedir. DI2 ile ifade edilen ADAM-5051 modülünün birinci dijital girişini ifade eder.

B-234 bölgesinde gösterilen devre ise ultrasonik sensörün hareketini sağlayan adım motoru sürücü devresidir. Bipolar olan motorun

döndürülebilmesi için sargı uçlarına belirli bir sıra ile gerilim uygulanır. DAI1, DAI2, DAI3 ve DAI4 ile ifade edilen uçlar ADAM-5051 modülünün (bkz. Ek-2) birinci ikinci üçüncü ve dördüncü dijital çıkışlarını ifade eder.

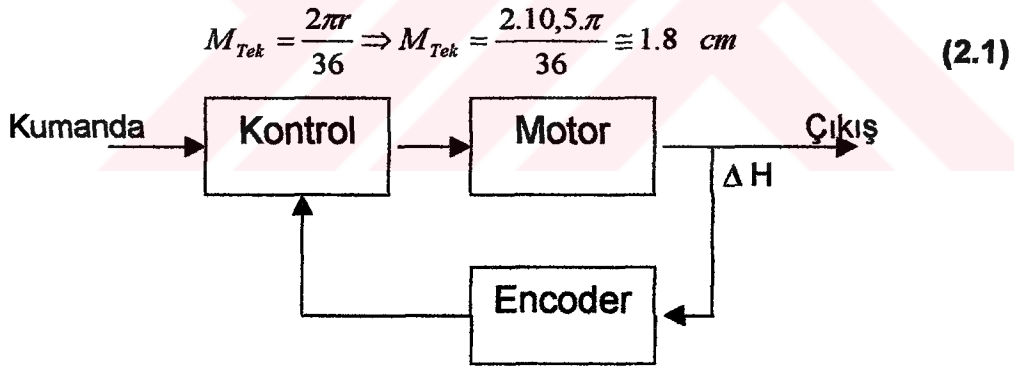




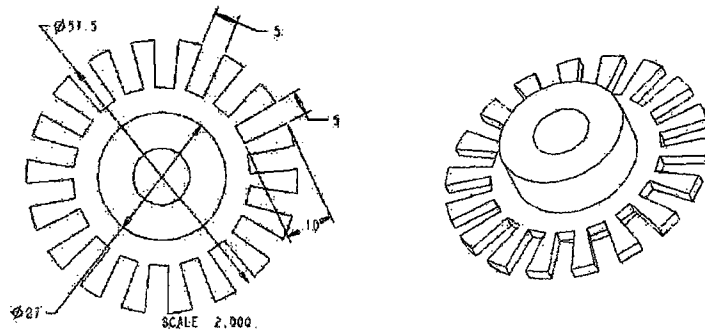
2.5. OYA Tasarımı

Geliştirilen OYA temel olarak dikdörtgen bir şase üzerine oturtulan platformun tekerlekli hale getirilmesi ile gerçekleştirilmiştir. OYA'nın hareketleri bağımsız kontrollü iki tekerlek ve bir avare tekerlek tarafından sağlanmaktadır. Kontrollü tekerlekler sırası ile 3W'lık bir motor, her tam dönüşte 36 sinyal üreten enkoder ve 10,5 cm yarıçapa sahip bir tekerlekten oluşmaktadır. OYA'nın öne ve arkaya hareketleri her iki kontrollü tekerleğin aynı hız ve miktarda döndürülmesiyle yapılırken, yön değiştirme işlemi dönülecek yöndeki tekerleğin sabit, aksi taraftakinin dönüş açısını sağlayacak kadar ilerletilmesiyle gerçekleştirilir.

Motorların tahrik ettiği tekerleklerin katettiği mesafeyi belirleyebilmek için tasarlanan encoderler (Şekil 2.9) bir tam dönüşte 36 sinyal vermektedir. Bu da; r tekerlek yarıçapı olmak üzere tek bir sinyalde ;



Şekil 2.8. Motor Kontrolü İçin Akış Diyagramı



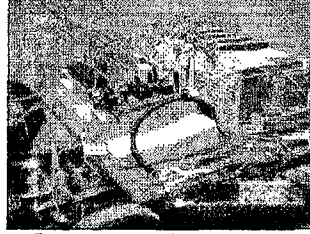
Şekil 2.9. Mesafe Ölçümünü Sağlayan Encoder Kesiti ve Ölçüleri

2.6. Prototip OYA'nın Görüntüsü

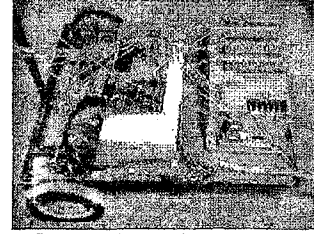
Bu bölüm içerisinde tasarım aşamaları ve teknik alt yapısı oluşturulan OYA'nın fiziksel görüntüsü aşağıda verilmektedir.



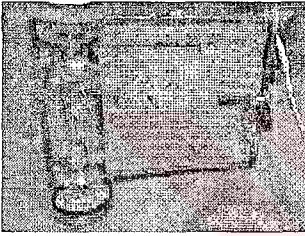
Şekil (a) Sistemin Üstten Genel Görüntüsü



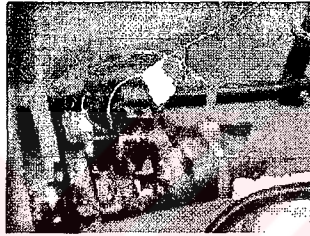
Şekil (b) Sistemin Sağdan Genel Görüntüsü



Şekil (c) Sistemin Genel Görüntüsü



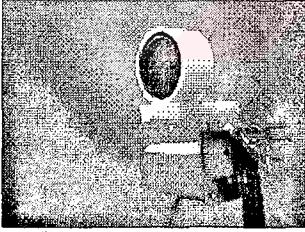
Şekil (d) Sistemin Alttan Genel Görüntüsü.



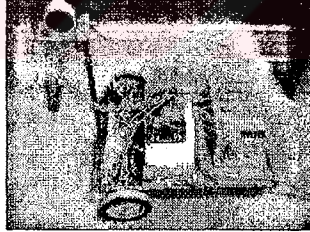
Şekil (e) Motor, Encoder, Tekerlek Bölümünün Üsten Görünüşü



Şekil (f) Motor, Encoder, Tekerlek Bölümünün Alttan Görünüşü



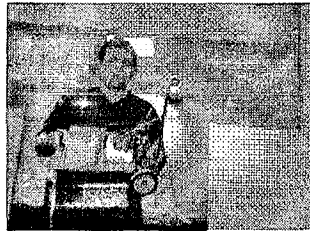
Şekil (g) Mesafe Ölçümünü Sağlayan Ultrasonik Bölüm



Şekil (ğ) Ultrasonik Sensörün Sistemdeki Yeri



Şekil (h) Sistemin Sağda Genel Görüntüsü.



Şekil (ı) Sistemin Ölçekli Genel Görüntüsü

Şekil.2.10. OYA'nın Fiziksel Görüntüsü

3 SÜRÜCÜLERİN KURULUMU

Bu bölüm içerisinde kontrol sisteminin donanım kısmının temel taşı oluşturulan Adam 5000/CAN ve PC ile bağlantısını sağlayacak PCL-841 kartının PC'ye tanıtımı açıklanmaktadır.

PCL-841 kartı aynı anda iki farklı CAN ağını destekleyebilen Advantech firması tarafından üretilen bir CAN dönüştürücü kartıdır (bkz. Ek-2). PCL-841 kullanışı ve kurulumu çok kolay bir ISA tipi kontrol kartıdır. Kartın sisteme tanıtılması sırasında dikkat edilmesi gereken belli başlı birkaç nokta dışında hiç bir ayara ihtiyaç duyulmaz, bu ayarlar yapılır yapılmaz PC'ye takılarak kullanılır hale getirilir. Bu ayarları ve kendi sistemim için geçerli ayarları Çizelge 3.1. de bulabilirsiniz.

Çizelge 3.1 PCL-841 kartı için Sınır Değerler ve Geçerli Değerler

Ayar Cinsi	Sınır Değerler	Geçerli Değerler
İmha Direnci	Kart üzerinde bulunan JP9 ve JP10 jumper'leri sayesinde kullanılacak olan portlar ya da bir başka deyişle kaç tane CAN ağının kontrol edileceğini belirlemektedir.	Gerçekleştirilen sistem içerisinde sadece bir tek CAN ağının bulunmasından dolayı JP9 açık bırakılırken JP10 her hangi bir çakışmayı engellemek için kapalı duruma getirilmiştir.
Memory Base Adres	Kart üzerinde bulunan 6 pozisyonlu DIP switch sayesinde C800 ile EF00 arasında bir değere ayarlanabilme şansına sahiptir.	Win 98 / Ayarlar / Sistem / Sistem Özellikleri / Bilgisayar Özellikleri / Giriş-Çıkış, sekmesini kullanarak yapılan kontrol sonrası default değeri DA00'a ayarlanmıştır.
Interrupt (IRQ) Ayarları	Kart üzerinde bulunan ve her port için ayrı bir ayar kısmına sahip IRQ isteği portu IRQ 3,4,5,6,7,9,10,11,12,15 kesmelerinden birine ayarlanabilme şansını sağlar	Win 98 / Ayarlar / Sistem / Sistem Özellikleri / Bilgisayar Özellikleri / Kesme İsteği, sekmesini kullanarak yapılan kontrol sonrası en uygun olan IRQ 5'e ayarlanmıştır.

PCL-841 kartı üzerinde yapına tüm ayarlar Çizelge 3.1 verilmiştir. Bu ayarlar dışında PCL841 kartına hiçbir ayar yapılmamış ve hiçbir ek yazılım

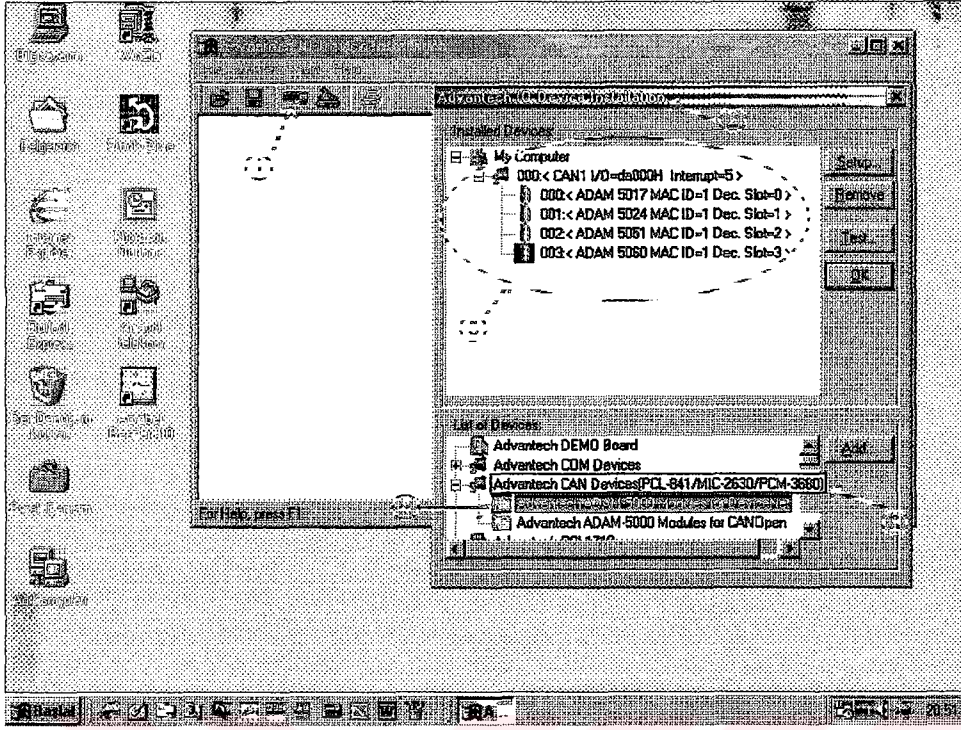
kurulmamıştır. PC içerisinde boş olan bir ISA slotuna takılarak Adam 5000/CAN ile bağlantısı Port1 üzerinden Ek2'de gösterilen pin bağlantılarına uygun bir kablo ile sağlanmıştır.

Adam 5000/CAN ise PCL-841 kartına göre daha karmaşık bir donanım ayarına ve yine PCL-841 kartından farklı olarak yazılım desteğine ihtiyaç duyar. Öncelikle Adam 5000/CAN modülünün iletişim hızını ve düğüm (node) ID'sini belirleyebilmek için yine modül üzerinde bulunan dip switch (bkz. Ek2.) aracılığı ile ayarlama yapılmış ve node ID olarak 01H, DeviceNet baudrate olarak da (iletişim mesafesinin 40 metrenin altında olmasından dolayı) 500 kbps değerlerine ayarlanmıştır.

Adam 5000/CAN modülünün kullanılacak olan PC'ye PCL-841 kartı üzerinden tanıtılması ise daha karmaşık bir yapıya sahiptir. Donanımın tanıtılma işlemi Advantech'in ürettiği "Advantech 32bit DLL Driver For Windows 95 And 98" sürücüsü sayesinde yapılmaktadır. Sürücünün kurulmasından sonra esas önemli kısım olan bu program sayesinde yapılan ayarlar ile Adam 5000/CAN modülünün PC aracılığı ile kontrol edilmesi sağlanmaktadır.

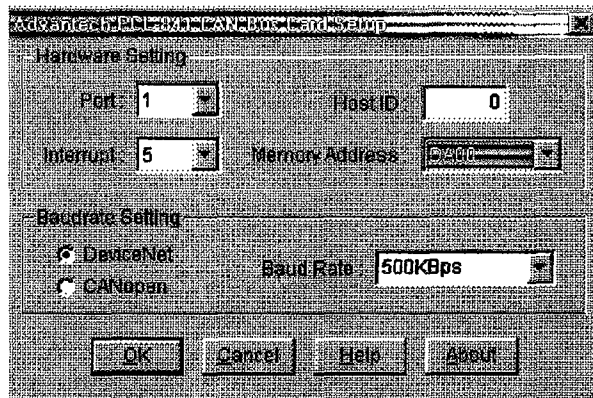
Yapılması gereken bu ayarlar incelenecek olursa:

Başlat/Programlar menüsü içerisinde bulunan Device Installation programı çalıştırıldıkta sonra karşımıza çıkan Advantech IO Device Installation çerçevesi içerisinde bulunan Install (Şekil 3.1. içindeki 1 numaralı ok ile gösterilmektedir) düğmesi yardımı ile Advantech IO Device Installation penceresine ulaşmaktayız (Şekil 3.1. içindeki 2 numaralı ok ile gösterilmektedir).



Şekil 3.1 Adam 5000/CAN Modülü İçin Gerekli Olan Ayarlar

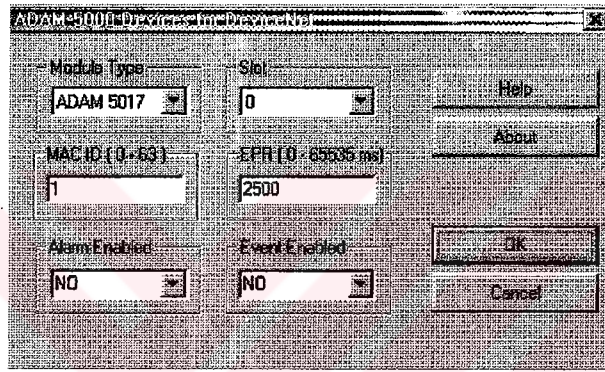
Bu uygulamadan sonra yapılması gereken işlem, Adam 5000/CAN modülünün haberleşmesi için kullanılan PCL-841 kartının parametreleri Advantech PCL-841 CAN Bus Card Setup bölümünde tanımlamaktır. Şekil 3.2’de geliştirdiğim uygulama için kullanılan orijinal değerler Advantech PCL-841 CAN Bus Card Setup ekranı içinde görülmektedir.



Şekil 3.2 Advantech PCL-841 CAN Bus Card Setup

Advantech PCL-841 CAN Bus Card Setup ayar çerçevesi içerisinde Şekil 3.2.’de tanımlanmak için sadece Host ID kısmı bulunmaktadır. Bu kısım içerisinde kullanılan PC’lerin adresleri tanımlanmaktadır.

Daha sonra Advantech IO Device Installation penceresinin alt kısmında bulunan List of Device menüsü içerisinde Advantech CAN devices (PCL-841/MIC-2630/PCM3680) seçilerek (Şekil 3.1 içinde 3 numaralı ok ile gösterilmektedir) bu sekmeden açılan alt menü içerisinde bulunan Advantech ADAM-5000 models for DeviceNet seçilir (Şekil 3.1 içinde 4 numaralı ok ile gösterilmektedir). Seçilen bu sekme sayesinde Adam 5000/CAN modülü içerisinde kullanılan giriş çıkış modülleri ve bu modüllerin özellikleri PC için tanımlanmaktadır. Örnek olarak Şekil 3.3.'de incelenen dijital giriş modülünü ele alalım:



Şekil 3.3 Dijital Giriş Modülünün Adam 5000/CAN İçin Ayarlanması

Ayar modülünün ilk bölümü olan Module Type kısmı ayarlanacak modülün kod numarasını içermektedir. İkinci ayarı olan MAC ID bölümü ise DeviceNet ağı içerisinde bulunan Adam 5000/CAN modüllerinin adresini belirlemektedir. Üçüncü ayarı olan Alarm Enabled bölümü ise sınır değerleri aşan bir değer gelmesi durumunda gerekli olan cihazları uyarmak için kullanılmaktadır. Dördüncü ayarı olan Slot bölümü ise giriş çıkış bölümünün Adam 5000/CAN modülü üzerinde hangi modüle yerleştirileceğini belirtmektedir. Beşinci ayarı olan EPR kısmı ise giriş çıkış biriminin hata sınırını belirtmektedir. Altıncı ayarı olan Event Enable kısmı ise giriş çıkış birimi üzerinde herhangi bir değişiklik olduğu zaman bu değişikliği bildirip bildirmeyeceğini belirtmektedir.

4 SCADA SİSTEMİNİN TASARIMI

Günümüz teknolojisi içerisinde veri toplama, değerlendirme ve kontrol sistemi kullanıcıları yarar-maliyet analizi sonucuna göre en uygun sistemleri tercih etmektedirler. Bu analiz içerisindeki yarar kısmı kullanıcı tarafından güçlü, gerçek zamanlı, kolay uygulanabilir, basit, kolay kullanılabilen ve esnek sistem tanımı ile doldurulmaktadır. Maliyet kısmı ise yaptığı iş gözönünde bulundurulduğu zaman aynı işi gören cihazlar arasında en ucuzu, kurulum maliyeti en düşük, uzun ömürlü ve yedek parça ihtiyacı rahat ve ucuz sağlanabilme imkanları ile doldurulmaktadır.

Bu yarar-maliyet analizi içerisinde değerlendirilmeye tabi tutulan en büyük etken ise operatör ve proses arasında bir ara yüzey oluşturan Veri Tabanlı Kontrol ve Gözetleme Sistemi'dir (Supervisory Control And Data Acquisition, SCADA). SCADA sistemleri, endüstriyel bilgisayarlar ve iş istasyonları için tasarlanmış bir gerçek-zamanlı (real-time), çok-görevli (multi-task), birçok uygulamanın gereksinimlerini karşılayan yazılım paketidir.

Standart bir SCADA yazılımı için aranan temel şartlar şunlardır:

- Kullanıcı tarafından tanımlanmış işletmeye ait parametreler ve parametre ekranlarda kullanılacak göstergeler yardımı ile işletmenin takibi (seviye, sıcaklık, basınç, sayısal sinyaller, vana ve motor durumları, sistem durumu vb...).
- Tanımlama ekranları vasıtasıyla sistem tanımlamalarının girilmesi ve işleyen sistem tanımı hakkında operatörlerin bilgilendirilmesi.
- Standart değer ekranları vasıtasıyla sistem için gerekli olan sabit verilerinin girilebilmesi olması (Setpoint, alt ve üst alarm değerleri vb.).
- PID parametrelerinin girilebilmesi ve gözetlenmesi.
- Sistem içerisinde gerçekleşen olayların gerekli değişkenler ile birlikte tutulabilmesi

- Anlık ve periyodik raporların (Üretim, Reçete, Stok vb.) alınması,
- Otomatik çalışan sisteme, SCADA ekranlarından manuel müdahale yapılabilmesi,
- Alarm ve durumların (event) gösterilmesi ve yazıcıya ve/veya veri tabanına kayıt edilmesi,
- İleri düzeyde kalite kontrol, örneğin İstatistiksel Proses Kontrol (Statistical Process Control SPC) desteği.

4.1 Adam 5000/CAN İçin Geliştirilen Tool Box

Bölüm 3.1’de kurulumunu detaylı olarak anlattığım ve Advantech firması tarafından üretilen “Advantech 32bit Dll Driver For Windows 95 And 98” sürücüsü, PC’e yüklendikten sonra, Adam 5000/CAN sistemini farklı programlama dilleri ile kontrol edilebilir kılmak için, Windows/System dizininin altına dinamik bağlantılı kütüphane (Dynamic Link Library, DLL) dosyaları yerleştirir. OYA sistemi için bu tezde geliştirmiş olan SCADA yazılımı da temel olarak bu kütüphaneleri kullanır. SCADA yazılımı Delphi programlama dili ile geliştirilmiştir.

Fakat burada geliştirilen SCADA yazılımı sonuç olarak çok özel bir uygulama için kullanılabilecek bir üründür. Bu SCADA programının geliştirme aşmaları içerisinde yukarda da bahsetmiş olduğum DLL dosyaları, (Dnet.pas ve Drivers.pas dosyalarını) sürekli tekrarlanarak kullanıldı. Bu sebepten dolayı, Adam 5000/CAN sistemi üzerinde yapılacak bir başka uygulama sırasında tekrar böyle bir zorluğa girilmemesi düşüncesi ile Adam 5000/CAN sisteminin giriş-çıkış birimlerini rahatlıkla kullanılabilmesi için Delphi programlama dili içerisinde kullanılabilecek bir Tool Box geliştirildi.

Geliştirilen “Adam 5000/CAN” Tool Box sayesinde, Delphi programlama dilinin sağladığı ve programcıya büyük bir rahatlık getiren, nesne tabanlı

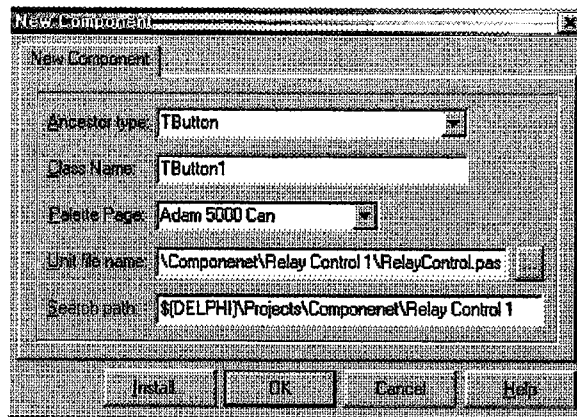
(object-oriented) programlama tekniğine uygun olarak, Adam 5000/CAN sisteminin DeviceNet protokolü aracılığı ile kontrolünü sağlar.

Geliştirmiş olduğum bu Tool Box'u (alet kutusu) daha detaylı açıklayabilmek için ADAM-5061 yani Adam 5000/CAN sitemini röle çıkışını kontrol eden RelayControl elemanı bu bölüm içerisinde açıklanmıştır.

Delphi programlama dili içerisinde yeni bir eleman yaratmak için temel olarak 5 farklı yöntem vardır.

- Mevcut kontrolleri değiştirmek (Modifying existing controls)
- Pencereleşmiş kontroller yaratarak (Creating windowed controls)
- Grafik kontrol yaratarak (Creating graphic controls)
- Pencere kontrollerini alt sınıflara ayırarak (Subclassing Windows controls)
- Görüntülü olmayan elemanlar yaratarak (Creating nonvisual components)

RelayControl elemanı, mevcut kontrolleri değiştirmek (Modifying existing controls) yöntemi kullanılarak oluşturulmuştur. Bu işlem için öncelikle Delphi programının Component/New eleman sekmesi kullanılarak New Component ekranını açılır (Şekil 4.1).

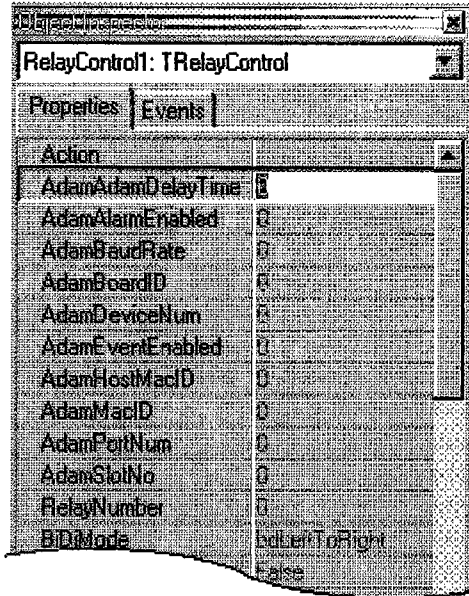


Şekil 4.1 New Component Ekranı

Açılan New Component penceresi içerisindeki ilk kısım olan "Ancestor type" kısmına, kullanım amacına en uygun özelliği gösteren, TButton

seçeneğine getirilir. Pencerenin Class Name kısmı ise ilk yaptığım ayarlardan sonra kendiliğinden otomatik olarak doldurulur. Tasarladığım elemanların hangi başlık altında ya da "Palet page" altında görüleceği ise pencerenin 3. kısmı Palet Page sayesinde gerçekleşmektedir. New Component ekranının dördüncü kısmı ise kullanacağımız pas soyadlı dosyasının (ki bu dosya ismi, oluşturulacak elemanın adını belirten kısımdır) yerini ve adını belirttiğim kısımdır. Beşinci kısım olan Search path ise kurulacak olan bu eleman için gerekli olacak DLL dosyalar ve yine bu elemanın kullanacağı ek dosyaların (sözkonusu örnek için Dnet.pas ve Driver.pas dosyaları) aranacağı araştırma kısmını oluşturmaktadır. Bu işlem tamamlandıktan sonra OK düğmesine basılarak oluşturulmak istenen eleman için gerekli olan program yazma kısmına geçilmektedir.

RelayControl elemanı Adam 5000/Can sistemi içerisindeki ADAM-5061 modülünün rölelerini kontrol edebilmek için belirli parametreler ihtiyaç duyar. Adam 5000/Can tarafından, ADAM-5061 modülünün kontrolü için ihtiyaç duyulan bu parametreleri Delphi içerisinde kullanılan nesneye dayalı programlama tekniğine uygun olarak sağlamak yazılan bu elemanın temel görevidir. Yazılmış olan bu eleman ile programlama sırasında girilen AdamDeviceNum, AdamBoardID, AdamHostMacID, AdamBaudRate, AdamPortNum, AdamSlotNo, AdamMacID, AdamAlarmEnabled, AdamEventEnabled, AdamDelayTime, AdamType gibi parametreler ile kontrol edilmek istenen rölenin KAA için ihtiyaç duyacağı parametreleri ve RelayNumber ile hangi rölenin çektireceğini ifade etmekteyim (Şekil 4.2).

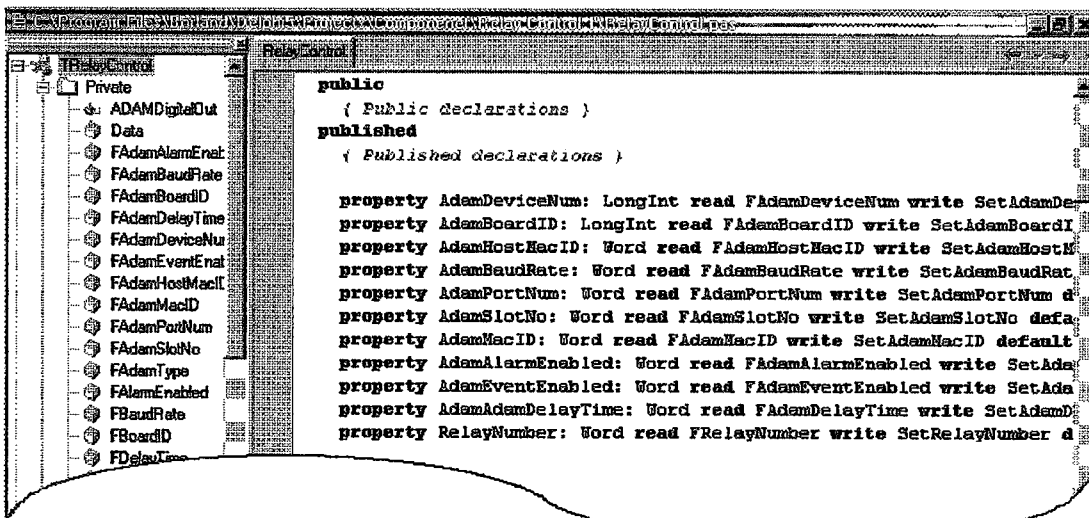


Şekil 4.2 Programlama Sırasında Girilmesi gerekli Parametreler

Yukarıda bahsettiğim verilerin girişinden sonra Delphi tarafından derlenen program çıktısında ise buton üzerinde gerçekleştirilen OnMouseDown prosödürü sayesinde verileri girilmiş olan röle çektilirilecek ve yine buton üzerinde gerçekleştirilen OnMouseUp prosödürü sayesinde ise röle bırakılacaktır.

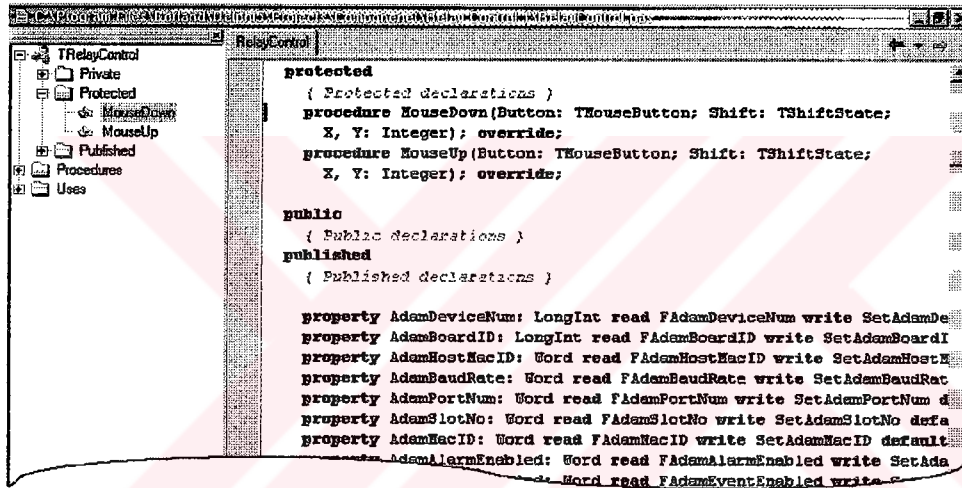
Bu elemanın geliştirme kısmına bakacak olursak istenilen bu veri giriş kısımlarının Object Inspector kısmında gösterilmesi ve OnMouseDown, OnMouseUp sırasında bu verilerin

değerlendirilerek Adam 5000/Can sistemine yollanması temel elemanın tasarlanması gereken temel kısmını oluşturmaktadır. Öncelikle veri girişi sağlanacak kısımların Object Inspector penceresi içinde kullanılabilir hale getirebilmek için, Private declarations bölümü içerisinde type'ları tanımlanır. Bu verileri Private declarations kısmında okuyacakları değişken, yazacakları değişken ve default değerleri belirlenir. (Şekil 4.3.).



Şekil 4.3 Veri Giriş Bölümlerinin Object Inspector İçin Belirlenmesi

Veri girişlerini, Object Inspector penceresi içinde kullanılır hale getirdikten sonra bu verileri değerlendirerek Adam 5000/Can sistemine yollayacak olan OnMouseDown, OnMouseUp prosedürlerini Protected declarations bölümü içerisinde tanımlandı (Şekil 4.4.). Bu tanımlama sonucunda Delphi tarafından derlenen program içerisindeki RelayControl buttonu üzerine gelip OnMouseDown işlemi yapıldığı takdirde veri giriş bölümü aracılığı ile toplanan veriler OnMouseDown prosedürü içerisine yazılan program sayesinde Dnet.pas ile Driver.pas dosyalarını kullanarak ADSDNet.DLL ve ADSAPI32.DLL dosyalarına ulaşırlar ve Adam 5000/Can sistemi ve DeviceNet prosedürünü kullanarak istenilen röleyi çektirirler.



Şekil 4.4 OnMouseDown Prosedürleri

4.2 OHR İçin Geliştirilen SCADA Yazılımı

Otomatik haritalama robotu için geliştirilen bu SCADA yazılımı sayesinde daha önceki bölümlerde tanımlanan ve açıklanan donanım kullanılarak geliştirilen OYA, bilinmeyen bir ortamda mümkün hareket koridorlarını tespit eden haritama sistemi olarak kullanılmak üzere bir yazılım ve grafik kullanıcı arayüzü geliştirilmiştir.

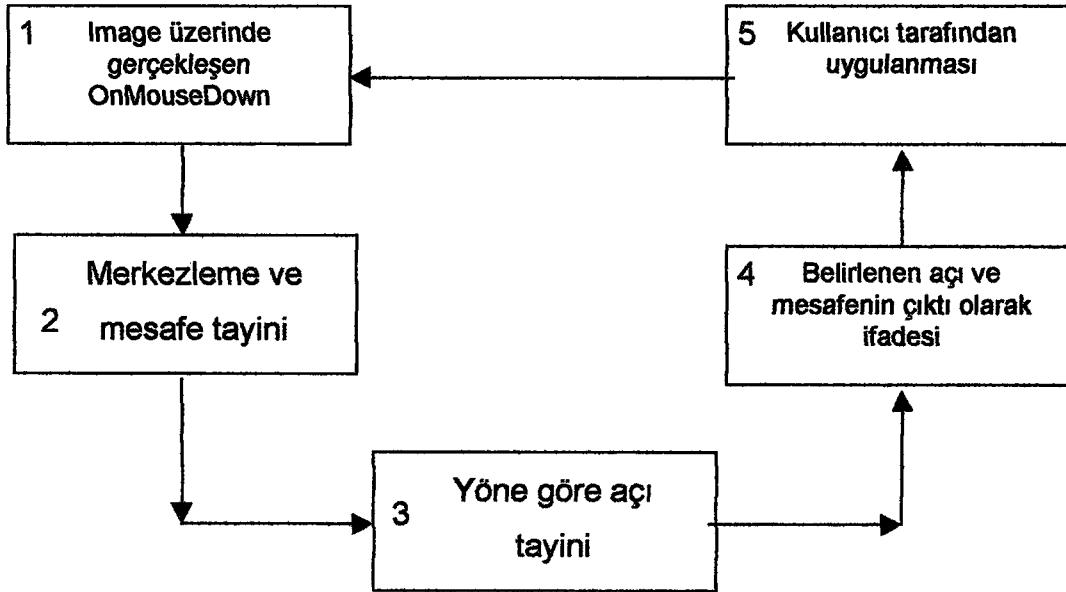
Otomatik haritalandırma robotu için geliştirilen yazılım temel olarak 4 bölüme ayrılmaktadır. Bu kısımlar sırası ile Yön Ayarlama, Tarama, Görüntüleme ve Elle kumanda kısımlarıdır. Bu bölüm içerisinde SCADA yazılımını oluşturan yukarıda saydığım birimler tek tek incelenecek ve daha

sonra sistemin bütünü hakkında Görüntüleme bölümü içerisinde bilgi verilecektir.



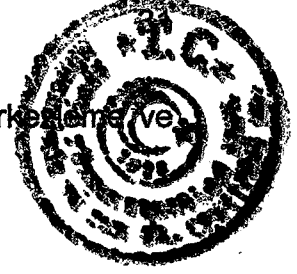
4.2.1 Yön Ayarlama

OYA'a yön vermek için kullanılan Yön Ayarlama programının çalışma prensibi temel olarak Delphi programının sağladığı OnMouseDown prosedürüne dayanmaktadır. Bir Image nesnesi içerisine yerleştirilen ve 0 derece ile AVG'nin burnu ifade edilen, ölçeklenmiş katran üzerinde istenilen açı ve mesafe noktasına Mouse yardımı ile tıklanarak belirlenir. Bu işlem sonrasında Image nesnesinin OnMouseDown prosedürü içerisinde yer alan program devreye girerek merkez olarak kabul ettiği katranın göbek noktası ile mouse'un OnMouseDown yapıldığı nokta arasında bir çizgi oluşturur ve OYA'ın burnu olarak kabul edilen 0 noktası ile yaptığı açıyı hesaplar. Fakat bu açı, kullanıcı kolaylığı sağlaması için 360 derecelik bir kadrın içinde değil de ± 180 derecelik iki eşit parçaya bölünmüştür. Böylelikle kullanıcı OYA'ın burnu temel alınarak sağa veya sola kaç derece döneceğini sağ için +180 derece sol için ise -180 derece aralığında karar verir (Şekil 4.5).



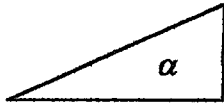
Şekil 4.5. OYA'ın Doğrultu Kontrol Prosedürünün Akış Diyagramı

Akış diyagramı içerisinde 2 ve 3 numara ile belirtilen "Merkezleme ve mesafe tayini" ile "Yöne göre açı tayini" bölümlerini açıklanırsa:



```
merkezx:=(X-107);  
merkezy:=(Y-93)*(-1);
```

Program kodu yardımı ile kadrarı oluşturan ekranın merkez noktasını sıfır olarak tayin edilir. Merkezleme yapılır.



h : gidmesafe
 α : gidaci kabul edilerek ,

```
gidmesafe:=(Sqrt(Sqr(merkezx) + Sqr(merkezy)));  
gidaci:=((merkezx/gidmesafe));  
gidaci:=Round(((ArcSin(gidaci))*90)/(pi/2));
```

Program kodu yardımı ile gidilen mesafe ve açı tayin edilmektedir.

```
If (merkezx>0) and (merkezy<0) then  
gidaci:=180-gidaci  
else If (merkezx<0) and (merkezy<0) then  
gidaci:=((90+gidaci)+90)*(-1);
```

Program kodu yardımı ile OnMouseDown olayının gerçekleştirildiği noktanın x ve y koordinatlarının işaretleri dikkate alınarak koordinat sistemi içerisinde hangi bölgede olduğu tespit edilir ve sağa veya sola kaç derece döneceğini sağ için +180 derece sol için ise -180 derece aralığında bulunması gerektiği mantığı ile açı tespit edilir ve SCADA yazılımı üzerinde bulunan "Mesafe" ve "Açı" bölümlerine yazdırılır.

Bu işlemten sonra belirlenen bu değerler kullanıcı tarafından uygun görülmesi halinde, "Hareket" ve "Dön" düğmelerine basılarak OYA'nın hareketini sağlayan motorlar Adam 5000/Can ünitesinin analog çıkışı tarafından sürülen röleler sayesinde enerjilendirilirler ve OYA hareket eder.

Akış diyagramı içerisinde 5 numara ile belirtilen "Kullanıcı tarafından uygulanması" bölümlü açıklanırsa.

"Hareket" düğmesine basılmasından sonra SCADA üzerinde "Mesafe" ile belirtilen kısım içerisindeki değerin OYA'nın yapacağı hareket için ileri veya geri anlamına geçecek olan işareti kontrol edilir.

```
alfa:=StrToInt(Edit11.text);
If alfa<0 then But1MoDo(Sender);
If alfa>0 then But3MoDo(Sender);
```

Bu program işaret tespitinden sonra uygun olan analog çıkışın aktif olması için Adam 5000/Can ünitesine gerekli emri yollar. Bu emir AnalogOut prosedürü sayesinde gerçekleştirilir.

```
procedure TForm1.AnalogOut(Sender: TObject);
var ErrCode: Longint;
    ErrMsg: string;
begin
    DeviceInfoAnalogOutput(Sender);
    ControlOfCanCard(Sender);
    ErrCode := ADAMAnalogOut( FPortNum, FMacID, FSlotNo,
                             FDelayTime,
                             chan,
                             voltage );
    if ErrCode<>SUCCESS then
    begin
        ErrMsg := GetErrorMessage(ErrCode);
        Timer1.Enabled := False;
        Timer6.Enabled := False;
        Timer7.Enabled := False;
        Timer8.Enabled := False;
        Timer9.Enabled := False;
        DNReset(FPortNum); { reset CAN card }
        DNClose(FPortNum);
        MessageDlg( ErrMsg, mtError, [mbOK], 0 );
        exit;
    end;
end;
```

Bu prosedür öncelikle DeviceInfoAnalogOutput prosedürü yardımı ile Adam 5000/Can ünitesi üzerinde bulunan analog çıkışın nasıl olacağını tespit etmek için AdamDeviceNum, AdamBoardID, AdamHostMacID, AdamBaudRate, AdamPortNum, AdamSlotNo, AdamMacID, AdamAlarmEnabled, AdamEventEnabled ve AdamDelayTime değerlerini okur.

```

procedure TForm1.DeviceInfoAnalogOutput(Sender: TObject);
begin
    FDeviceNum:=      1000;
    FBoardID:=         80;
    FHostMacID:=        0;
    FBaudRate:=         500;
    FAlarmEnabled:=      0;
    FEventEnabled:=      0;
    FDelayTime:=         2500;
    FPortNum:=           1;
    FSlotNo:=            1;
    FMacID:=             1;
end;

```

Bu işlemden sonra ControlOfCanCard(Sender) prosedürü sayesinde DeviceNet içerisine girerek mesajı iletir.

```

procedure TForm1.ControlOfCanCard(Sender: TObject);
var ErrCode: Longint;
    ErrMsg: string;
begin
    ErrCode := DNOpen( FDeviceNum, FPortNum, FHostMacID , FBaudRate);
    if ErrCode=SUCCESS then
        ErrCode := DNDeviceGetParamOfReg( FDeviceNum, FMacID, FBoardID,
            FSlotNo,
            FDelayTime,
            FAlarmEnabled, FEventEnabled );
        if ErrCode=SUCCESS then
            ErrCode := DNInit( FHostMacID, FPortNum, FBaudRate );
        if ErrCode<>SUCCESS then
            begin
                DNReset(FPortNum);
                DNClose(FPortNum);
                ErrMsg := GetErrorMessage(ErrCode);
                MessageDlg( ErrMsg, mtError, [mbOk], 0 );
            end;
        end;
end;

```

Böylelikle OYA'ın hareketini sağlayan motorlar Adam 5000/CAN ünitesinin analog çıkışının aktif olmasından dolayı sürülen röleler sayesinde enerjilendirilirler ve OYA hareket eder. Fakat bu analog çıkışın aktif olan emrin yollanmasından önce motorun, yani dolayısı ile OYA'nın aldığı mesafeyi, optik enkoderin tetiklemesi sayesinde ölçen Adam 5000/Can ünitesinin dijital girişini, 1 mili saniyede bir ölçüm yapan Timer1'i aktif hale getirir. Timer1 öncelikle, DeviceInfoDigitalInput prosedürü yardımı ile Adam

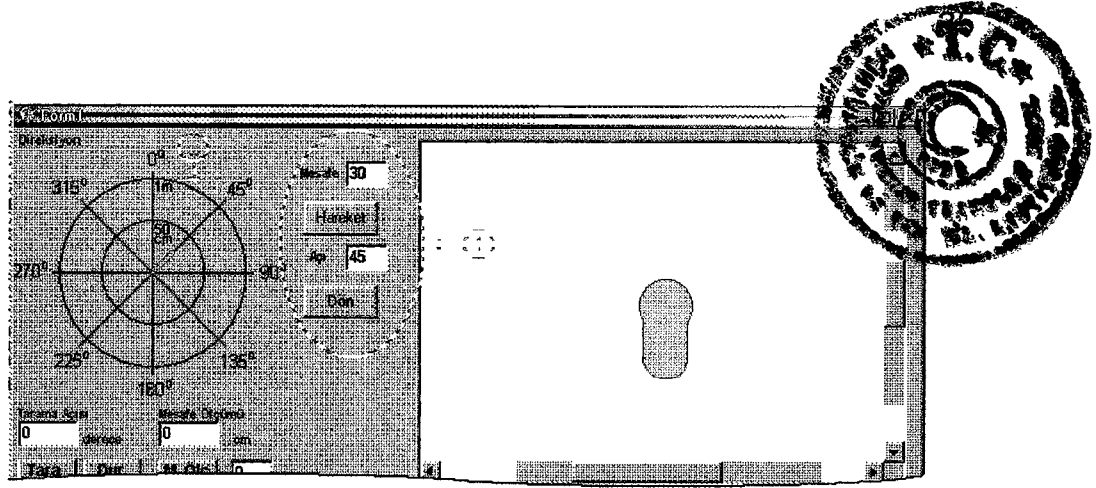


5000/Can ünitesi üzerinde bulunan dijital girişten nasıl veri alınacağını tespit eder.

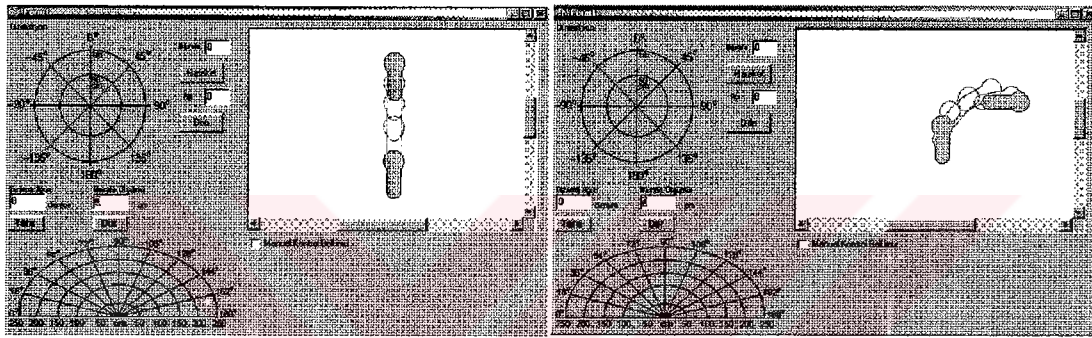
```
TForm1.DeviceInfoDigitalInput(Sender: TObject);
begin
  FDeviceNum:= 2000;
  FBoardID:= 81;
  FHostMacID:= 0;
  FBaudRate:= 500;
  FAlarmEnabled:= 0;
  FEventEnabled:= 0;
  FDelayTime:= 2500;
  FPortNum:= 1;
  FSlotNo:= 2;
  FMacID:= 1;
end;
```

Bu işlemden sonra ControlOfCanCard(Sender) prosedürü sayesinde DeviceNet içerisine girerek gerekli bilgiyi okur. Optik enkoder tarafından gönderilen darbeler böylelikle dijital giriş modülü üzerinden sayılır. Timer1 tarafından sayılan bu darbeler, her bir darbeye karşılık gelen mesafenin bilinmesi sayesinde (bkz. Bölüm 2.3.5) mesafe ölçümü yapılır. İstenilen mesafeye ulaşılması veya aşılması koşulunda OYA'ı hareket ettiren röleyi enerjilendiren analog çıkış pasif hale getirilir. Böylelikle OYA durur.

Temel olarak SCADA'nın bu bölümü kullanıcı ile direkt olarak temasta olan en önemli bölümdür. Bu bölüm, OYA'ı yönlendirmek için kullanılan temel birimdir. Kullanım kolaylığı göstermesi için tasarlanmıştır. Bu birim içerisinde kullanıcı OYA'nın gideceği yön ve mesafeyi tayin etmektedir. Bu tayin işlemi göz kararı yapılmaktadır (Şekil 4.6'da 2. okla gösterilmekte). Bu yüzden kullanıcı tarafından belirlenecek değerlerin her zaman kullanıcının istediği değerler olmayacağı varsayılarak direkt olarak OYA'a gönderilmeyip kullanıcının değerlendirmesi ve istediği değerler olduğuna karar vermesinden sonra bu değerleri OYA'a göndermesini sağlamak için Şekil 4.6'da 1. okla gösterilen kısma yollarır. Bu işlemden sonra değerler kullanıcının isteği ile OYA'a yollarır.



Şekil 4.6. OYA Doğrultu Kontrol Bölümü

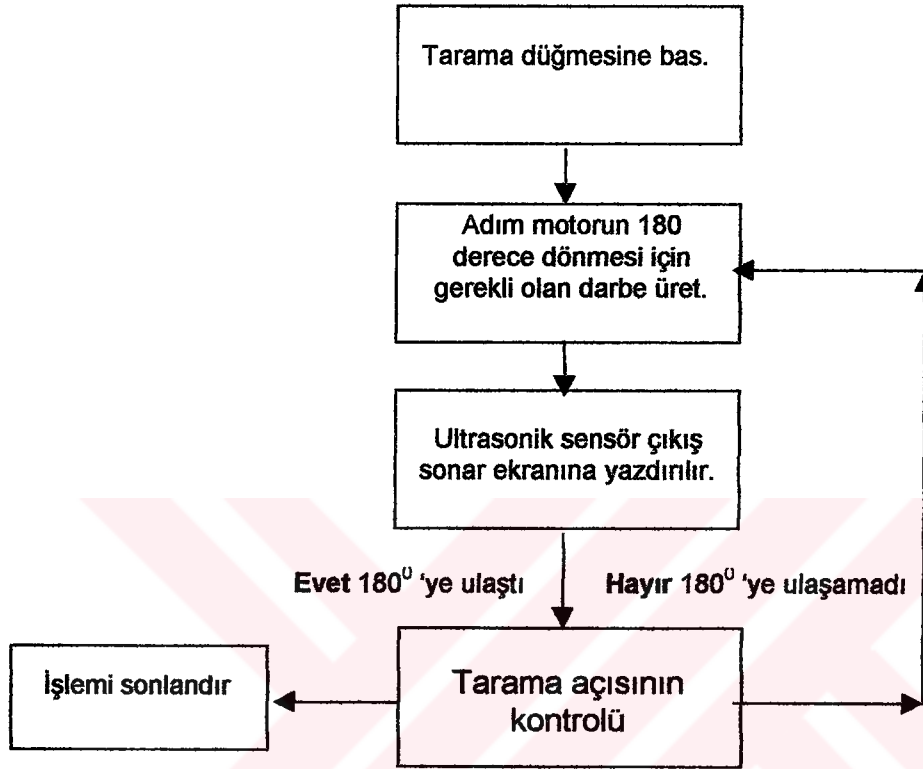


Şekil 4.7. Yönlendirme Sonucu

4.2.2 Tarama

SCADA'nın "Tarama" bölümü OYA'nın asıl işlevi olan haritalandırma özelliğinin temellerini oluşturan. Bu bölüm içerisinde; adım motoru kontrolü ve ultrasonik sensörden bilgi okuma işlemi yazılan program sayesinde koordineli olarak gerçekleştirilmektedir. "Tara" düğmesine basılması ile birlikte ilk 180 dereceyi taramasını sağlayan Timer1 aktif hale gelir çalıştırılan bu zamanlayıcı sayesinde adım motoru için gerekli olan adım frekansı oluşturulur. Bu tarama sırasında ultrasonik sensörden lineer bir çıkış yani daha kesin bir mesafe tayini sağlayabilmek için tepe motor her 18 derecede bir, 2 saniye süresince durdurulur ve iki saniye sonra ultrasonik sensör çıkışı gerilim cinsinde okunarak program içinde mesafeye döndürülür. Program içinde saklanan bu açı ve mesafe bilgisi bir sonraki 18 derecelik taramadan önce sonara ekranı üzerinde işaretlenir. Bu işlem 180 derece tamamlanıncaya kadar devam eder. 180 derecenin tamamlanmasından

sonra Tara düğmesine tekrar basılması halinde adım motoru bu kez tam zıt yönde yine yukarıda anlatılan sırayı izleyerek 180 derecelik bir tarama gerçekleştirir.



Şekil 4.8. OYA'ın Tarama Bölümü İçerisindeki Akış Diyagramı

Tarama bölümünün SCADA tarafında çalışan programlar incelenecek olursa, temel olarak iki ana proses görülür. Bunlardan birincisi sensörün adım motoru tarafından konumlanması, ikincisi de konumlanan ultrasonik sensörün engel ile arasındaki mesafeyi ölçmesi. Elde edilen bu polar koordinattaki engel konum bilgisi yine bu bölüm içerisinde işlenerek ekran çıktısı olarak verilmektedir.

Bu işlemlerden birincisini teşkil eden adım motorunu istenilen açı değerine getirilmesi prosesi, OYA sistemine bağlı olarak 180 derecelik bir kısmı kapsamaktadır. OYA hareket etmeden önce gideceği yol üzerinde ki engelleri belirlemesi amacı ile adım motoru ultrasonik sensörü 180 derecelik açı içerisinde hareket ettirir. Bu işlem için Timer6 prosedürü işletilmektedir.

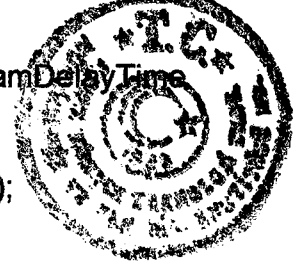


```
procedure TForm1.Timer6Timer(Sender: TObject);
begin
case I of
0: begin AdimBir(Sender);l:=1;j:=3; m:=m+1;
      k:=k+1;Edit8.text:= Floattostr(Round(((k*(180/100)))));end;
1: begin AdimIki(Sender); l:=2;j:=2;m:=m+1;
      k:=k+1;Edit8.text:= Floattostr(Round(((k*(180/100)))));end;
2: begin AdimUc(Sender); l:=3;j:=1;m:=m+1;
      k:=k+1;Edit8.text:= Floattostr(Round(((k*(180/100)))));end;
3: begin AdimDort(Sender); l:=0;j:=0;m:=m+1;
      k:=k+1;Edit8.text:= Floattostr(Round(((k*(180/100)))));end;
end;
if k=100 then begin
Sonar(Sender); Timer6.Enabled:=False; end
else if m=5 then Bekleme(Sender)
end;
```

Timer6 prosesi öncelikle adım motorunu çalışmasını sağlayan darbe jeneratörü görevini yerine getirir. Timer elemanının aralık özelliği yardımı ile saniyede kaç darbe üretileceği belirlenerek adım motorunu dönüş hızı belirlenebilmektedir (bkz. bölüm 2.1.4). Timer6, adım motoru için gönderdiği darbeleri kısır bir döngü içinde, aralık özelliği yardımı ile belirlenen frekans, Adim..... prosedürlerini çağırarak meydana getirir.

```
procedure TForm1.AdimBir(Sender: TObject);
begin
DeviceInfoDigitalOut(Sender);
ControlOfCanCard(Sender);
Data := 0;
DigitalOut(Sender);
i:=1 ;
Data := Data + i;
DigitalOut(Sender);
DNReset(FPortNum);
DNClose(FPortNum);
end;
```

Adım... prosedürleri ise öncelikle DeviceInfoDigitalOut prosedürünü çağırarak Adam 5000/Can ünitesi üzerinde bulunan röle çıkışın nasıl olacağını tespit etmek için AdamDeviceNum, AdamBoardID, AdamHostMacID, AdamBaudRate, AdamPortNum, AdamSlotNo,



AdamMacID, AdamAlarmEnabled, AdamEventEnabled ve AdamDelayTime değerlerini okur.

```
procedure TForm1.DeviceInfoDigitalOut(Sender: TObject);
begin
  FType      := 0;
  FOutBit    := 0;
  FDoState   := 0;
  FDeviceNum:= 3000;
  FBoardID:=83 ;
  FHostMacID:=0;
  FBaudRate:=500;
  FAlarmEnabled:=0;
  FEventEnabled:=0;
  FDelayTime:=2500;
  FPortNum:=1;
  FSlotNo:=3;
  FMacID:=1;
  Data :=$00;
end;
```

Bu işlemden sonra ControlOfCanCard(Sender) prosedürü sayesinde DeviceNet içerisine girerek gerekli bilginin doğruluğunu Adam 5000/Can sayesinde kontrol eder. Bu kontrol sonucunun olumlu olması durumunda adım motorunu istenilen şekilde hareket etmesi için ihtiyaç duyacağı sinyal, DigitalOut prosedürü işletilerek oluşturulur.

Timer6 adım motorunun çalışmasını sağlaması yanında adım motoruna gönderdiği darbeleri sayarak adım motorunun dönüş açısını da hesaplar. Hesaplanan bu açı değerleri tespit edilecek engelin polar koordinatının açı değerini oluşturur.

Önceden belirlenen açı değerlerine ulaşıldığı zaman (tasarlanan OYA için bu değer 18 derecede 1'dir) adım motoru, ultrasonik sensörün ölçüm bilgisinin lineer hal alabilmesi için 2 saniye süresince dönme işlemine ara verir. Bu süre sonunda ultrasonik sensörün ölçtüğü değer Bekleme prosedürü yardımı ile tespit edilir. Bekleme prosedürü öncelikle Adam 5000/Can modülü üzerindeki hangi çıkışı ne şekilde okuyacağını tanımlamak için DeviceInfoAnalogInput prosedürünün çalıştırır.

```

procedure TForm1.Bekleme(Sender: TObject);
begin
  {Analog In}
  chan:=0;
  DeviceInfoAnalogInput(Sender);
  ControlOfCanCard(Sender);
  AnalogInChan0(Sender);
  DNReset(FPortNum);
  DNClose(FPortNum);
  {Sonar}
  Sonar(Sender);
  {Digital Out}
  Timer6.Enabled:=False;
  Timer8.Enabled:=True;
end;

```

Bu prosedür yardımı ile belirlediği değerlerin gerçekliğini ControlOfCanCard prosedürü sayesinde kontrol ettikten sonra AnalogInChan0 prosedürü yardımı ile ultrasonik sensörün ürettiği değeri okur.

Bu işlemler sonucu OYA sisteminin hareket edeceği yön üzerindeki engellerin polar koordinatları tespit edilir. Bu polar koordinatlar sayesinde engelin yeri, OYA'nın konumuna göre yeri matematiksel olarak belirlenmiş olur.

Bu sonuç çıktısını SCADA yazılımı üzerinde göstermek amacı ile SCADA'nın Tarama bölümü içerisinde bir sonar ekranı oluşturulmuştur.

Bu ekran programlama bakımından image nesnesi içerisine yerleştirilerek saydam hale getirilmiş olan bir kadrandan oluşmaktadır. Oluşturulan sonar ekranının kadranı 18 derecelik kısımlara ayrılmıştır. Kadrان mesafe ölçümünü göz kararı yapabilmek için 50 cm'lik yaylar ile bölünmüştür.

Engel tarayıcı bölümden gelen polar koordinatlar sonar ekranının kadranı 0 noktası temel alınarak Sonar prosedürü yardımı ile işlenir ve yansıtılır.

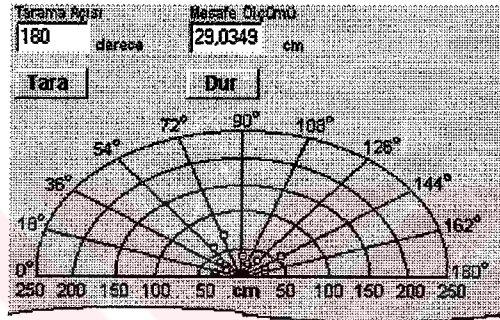
```

procedure TForm1.Sonar(Sender: TObject);
begin
  mesafe1:=strtoFloat(Edit6.text);
  aci:=strtoint(Edit8.text);
  x0:=mesafe1*(cos((aci/180)*pi));
  y0:=mesafe1*(sin((aci/180)*pi));
  {x0,y0 ölçülen gerçek değer}
  {x1,y1 ölçülen gerçek değer yuvarlanmış sonucu }
  {x,y yuvarlanmış sonucun grafiğe uygulanan değerleri }
  x1:=Round(x0);
  y1:=Round(y0);
  If aci < 91 then
  begin
    x:=140-(Round(x1/2));
    y:=100-(Round(y1/3));
    with Image1 do begin
      Canvas.Brush.Color := clRed;
      Canvas.Brush.Style := bsSolid;
      Canvas.Ellipse(x,y, x+5, y+5);
    end;
  end
  else
  begin
    x:=140-(Round(x1/2));
    y:=100-(Round(y1/3));
    with Image1 do begin
      Canvas.Brush.Color := clRed;
      Canvas.Brush.Style := bsSolid;
      Canvas.Ellipse(x,y, x+5, y+5);
    end;
  end;
  NesneninKonumu (Sender);
  Button9Click(Sender);
  nescizkon:= nescizkon+1;
  If nescizkon>=2 then
  Button12Click(Sender)
end;

```

Bu bölüm SCADA yazılımın en önemli kısmını oluşturmaktadır. SCADA yazılımının bu bölümü sayesinde OYA çevresinde olan nesnelerin yerini; yani OYA'nın ultrasonik tarayıcısına göre nesnelerin uzaklığının ve açısının

belirlendiği kısımdır. Bu kontrol SCADA yazılımı üzerinde bulunan “Tara” düğmesi sayesinde yapılmaktadır. Kullanıcı tarafından bu düğmeye basılmasından sonra ultrasonik sensörü kontrol eden adım motoru 180 derecelik bir tarama için ultrasonik sensörü hareket ettirir. Bu tarama sırasında adım motor her 18 derecede bir, iki saniyelik duraklayarak ultrasonik sensörün çıkışının lineer hale gelmesini bekledikten sonra ölçüm yapmaktadır. Yapılan bu ölçümler “Tara” düğmesinin altında bulunan sonar ekranı üzerine de, nesnenin açısı ve uzaklığı gözönüne alınarak işaretlenir (Şekil 4.9.). “Dur” düğmesi her hangi bir hata veya beklenmeyen olay karşısında tarama işlemini durdurmak için kullanılır.

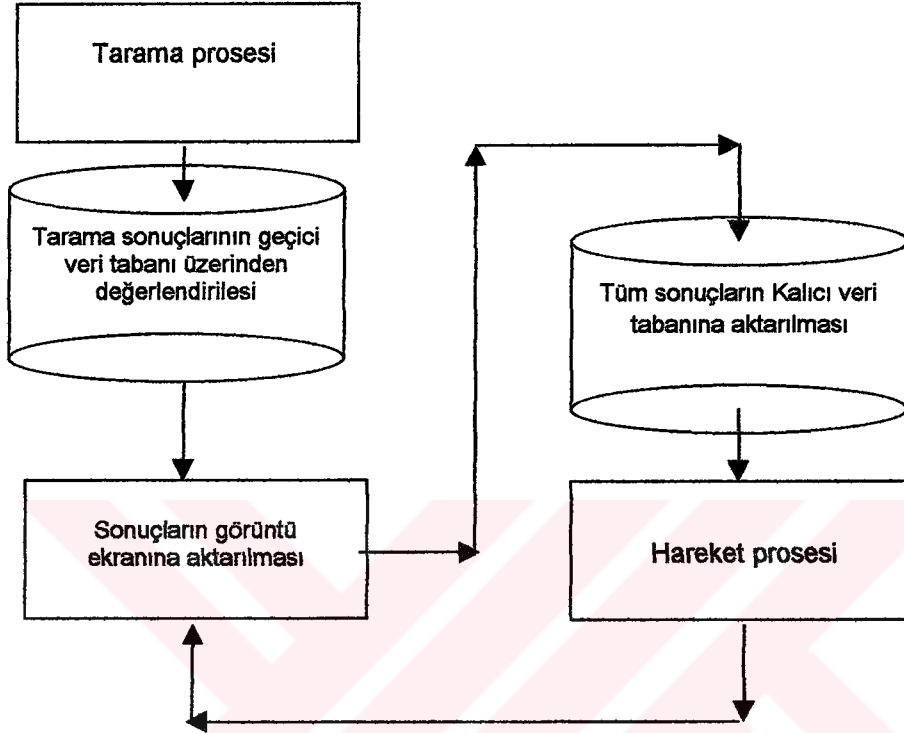


Şekil 4.9. Engelin Yer ve Konumunu Tayin Eden Tarama Bölümü

4.2.3 Görüntüleme

Görüntüleme bölümü OYA için yazılan tüm programın sonuçlarının toplandığı bölümdür. Kullanıcı tarafından belirlenen yön ve mesafe bilgileri OYA'a DeviceNet protokolü aracılığı ile Adam 5000/CAN sisteminde gönderilmesi ile birlikte Görüntüleme birimi ölçüm verilerini toplamaya başlar. Kullanıcı tarafından verilen hareket komutları bire bir görüntü ekranına ½ ölçek içerisinde yansıtılır. Tarama işleminin sonucu ise görüntü ekranına bir Access veri tabanı üzerine yazılıp tekrar değerlendirildikten sonra aktarılır. Sistem içerisindeki her hareket ve tarama olayı, OYA'ın başlangıç noktasını sıfır kabul ederek, SCADA yazılımı tarafından bir Access veri tabanına kaydedilir. Böylelikle sonradan ihtiyaç duyulabilecek veriler ölçüm sırasında kaydedilerek saklanır. Tarama işlemi ve nesne belirleme işlemi ise öncelikle değerlendirilir ve sonra ana veri tabanına aktarılır. Bu değerlendirme

sayesinde birbiri ile bağlantılı olduğu tespit edilen noktalar (örneğin bir duvar boyunca her 18 derecede bir ölçülecek veri duvarın genel silüetini oluşturma bilmek için) birleştirilirler.



Şekil 4.10. OYA'ın Görüntüleme Bölümü İçerisindeki Akış Diyagramı

Görüntüleme bölümü, SCADA sisteminin tüm sonuçlarını değerlendirip kullanıcının karar vermesi için ekran çıktısı olarak ifade eder. Bu işlevine ek olarak OYA sisteminin tüm ölçüm değerlerinin saklandığı kısım olduğundan dolayı programlama olarak en karmaşık bölümü oluşturur.

Görüntüleme bölümü temel işlevi olan ölçüm sonuçlarını yansıtırma işlemi iki kısım halinde gerçekleşir. Birinci kısmı, OYA sisteminin hareketinin yansıtılması oluşturur. İkinci kısmı, ise OYA sistemi tarafından saptanan engellerin belirlenmesi ve işleme prosesi oluşturur.

Oya sisteminin hareketleri Bölüm.2.1'de ayrıntılı olarak anlatılan enkoderler aracılığı ile sürekli olarak ölçülür (ölçme işlemin ayrıntıları Bölüm.3 içerisinde detaylı olarak anlatılmıştır). Ölçülen bu değerler

görüntüleme işlemi için başlangıç noktası olarak kabul edilen değerler üzerinden Cizim prosedürü aracılığı ile işlem görür.

```
procedure TForm1.Cizim(Sender: TObject);
begin
  aaci1:=ddegisim+aaci1;
  Araci1nSilinmesi(Sender);
  IleriGeriHareket(Sender);
  aci1salHareket(Sender);
  Araci1nCizilmesi(Sender);
end;
```

Başlangıç noktası ekranda kuzey yönünde uzanan araç çizimi ile ifade edilir. OYA hareketi sırasında, encoderin ölçebileceği en küçük değer olan 1,8 cm'lik her değer direk olarak Cizim prosedürüne aktarılır. Bu sayede OYA'nın yaptığı her hareket adım adım görüntüleme ekranına taşınmış olur.

Cizmi prosedürü sırası ile Araci1nSilinmesi, IleriGeriHareket, aci1salHareket ve Araci1nCizilmesi prosedürlerini çağırır .

Araci1nSilinmesi prosedürü sistem içerisinde yapılacak bir sonraki hareketten önce yapılmış olan bir önceki hareketi siler.

```
procedure TForm1.Araci1nSilinmesi(Sender: TObject);
begin
  with Image2 do begin
    Canvas.Pen.Color := clWhite;
    Canvas.Pen.Width:=35;
    {Drawing Car}
    Canvas.MoveTo(xx1, yy1);
    Canvas.LineTo(xx2, yy2);
    Canvas.Ellipse(xx1-4, yy1-4, xx1+4, yy1+4);
  end;
end;
```

IleriGeriHareket prosedürü OYA sisteminin her iki motorunu da birlikte hareket ettirir ve yapmış olduğu ileri geri hareketin mesafesini görüntüleme ekranına aktarır. Bu işlem sırasında IleriGeriHareket prosedürü aldığı yeni bilgiyi öncelikle başlangıç noktasına uygulayarak değişimini bir sonra gelecek veri için başlangıç noktası olarak kabul eder.

```

procedure TForm1.IleriGeriHareket(Sender: TObject);
begin
  xx11:=(Round((sin((-aaci1/180)*pi))*mmesafe1));
  yy11:=(Round((cos((aaci1/180)*pi))*mmesafe1));
  mmx:=mmx+xx11;
  mmy:=mmy+yy11; end;

```

OYA sisteminin tek bir motorunu çalıştırarak yaptığı açısal hareketler aci1salHareket prosedürü sayesinde işlenerek görüntüleme ekranına yansıtılır. OYA sistemi, tasarlanma aşamasında dönüşlerini bir tekerleğini durdurulup diğer tekerleğinin döndürülmesi sayesinde yaptırılacağı varsayılmıştır. Bu sebepten dolayı OYA sistemi her hareket edişinde belirli bir miktar ilerler. Bu ilerleyiş aci1salHareket prosedürü tarafından görüntü ekranına yansıtılır.

```

procedure TForm1.aci1salHareket(Sender: TObject);
begin
  xx1:=(Round((sin(((aaci1-180)/180)*pi))*32))+ mmx;
  yy1:=(Round((cos(((aaci1-180)/180)*pi))*32))+ mmy;
  xx2:=(Round((sin(((aaci1)/180)*pi))*7))+ mmx;
  yy2:=(Round((cos(((aaci1)/180)*pi))*7))+ mmy;
  ddegisim:=aaci1;
end;

```

OYA sistemi belirli bir noktadan istenilen noktaya hareket edip bu hareketini tamladıktan sonra OYA sisteminin yeni konumu bir Access veri tabanına kaydedilir. Tutulan bu veri tabanı sayesinde OYA'nın izlediği yol rahatlıkla bulunmaktadır.

```

procedure TForm1.Button9Click(Sender: TObject);
var
  datam:integer;
begin
  tablokal.Last;
  datam:=tablokal.fieldbyname('sayi').asinteger;
  tablokal.Insert;
  tablokal.fieldbyname('1').asinteger:=strtoint(edit13.text);
  tablokal.fieldbyname('2').asinteger:=strtoint(edit14.text);
  tablokal.fieldbyname('3').asinteger:=strtoint(edit15.text);
  tablokal.fieldbyname('4').asinteger:=strtoint(edit15.text);
  tablokal.fieldbyname('5').asinteger:=strtoint(edit17.text);
  tablokal.fieldbyname('sayi').asinteger:=(datam+1);
  tablokal.Post;

```



```
edit13.Text:='0';
edit14.Text:='0';
edit15.Text:='0';
edit16.Text:='0';
edit17.Text:='0';
edit13.SetFocus; end;
```

SCADA yazılımının Görüntüleme bölümünün gerçekleştirdiği diğer bir görev ise OHS sistemini oluşturmaktır. Oluşturulan bu OHS sistemi sayesinde engel tanımlayıcı cihazlar aracılığı ile elde edilen engel koordinatları ultrasonik sensörün OYA üzerindeki yeri referans alınarak görüntüleme ekranına yansıtılır.

Oluşturulan OHS sisteminin bir diğer özelliği ise toplanan bu engel koordinatlarının değerlendirilmek üzere bir Access veri tabanına aktarılmasıdır. Bu değerlendirme sayesinde birbiri ile bağlantılı olduğu tespit edilen noktalar (örneğin bir duvar boyunca her 18 derecede bir ölçülen veri sayesinde duvarın genel silüetini oluşturabilmek için) belirli bir mantık çerçevesinde birleştirilir.

```
function TForm1.deger(j:integer;k:integer;ek:integer):integer;
var
  datafark,datax1,datax2:integer;
begin
  tablokal.Locate('sayi',j, [loPartialKey]);
  datax1:=tablokal.fieldbyname(inttostr(k)).asinteger;
  tablokal.Locate('sayi',(j-ek), [loPartialKey]);
  datax2:=tablokal.fieldbyname(inttostr(k)).asinteger;
  datafark:=datax1-datax2;
  if datafark < 0 then
    datafark:=datax2-datax1;
  if k=1 then begin
    datax11:=datax1;
    datax12:=datax2;
  end;
  if k=2 then begin
    datay11:=datax1;
    datay12:=datax2;
  end;
  result:=datafark;
end;
```



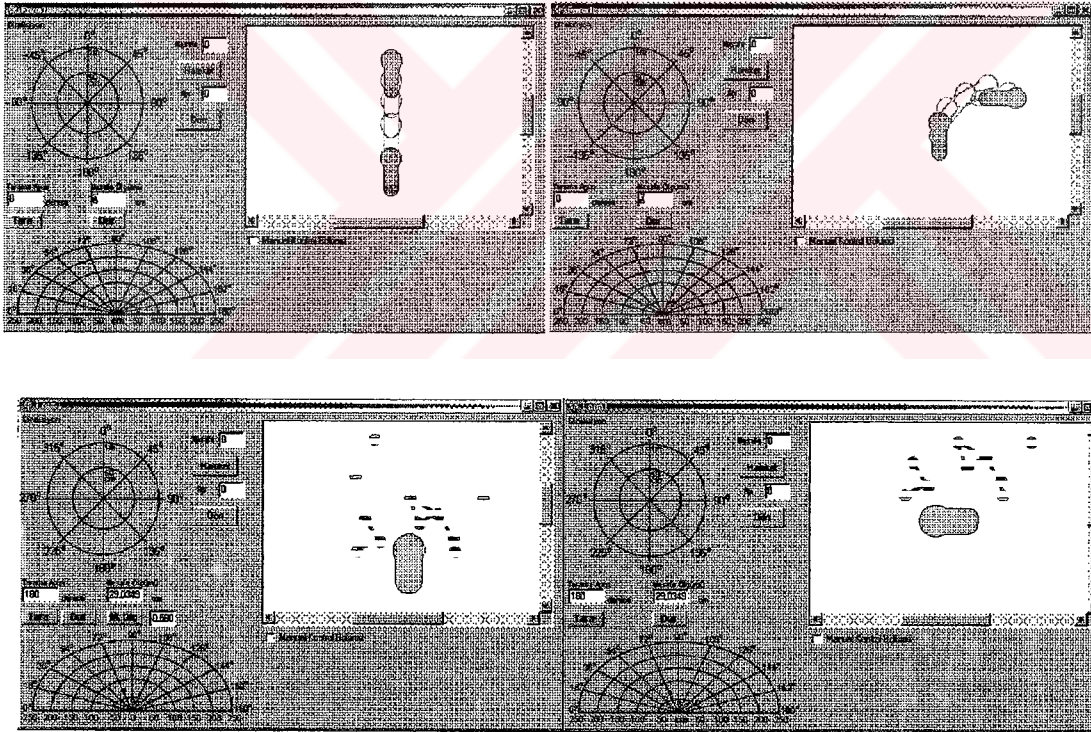
Bu değerlendirme işlemi temel olarak kıyaslama mantığına dayanır. Tarama sırasında her 18 derecede bir alınan engel koordinat bilgisi sıra ile Access veri tabanına aktarılır. Aktarılan bu engel noktalarının koordinatlarını oluşturan X ve Y değerleri birbirinden çıkarılarak farkları belirlenir. Bu fark deneysel çalışmalar sonucunda bulunan sabit bir değer altında ise belirlenen bu iki engel noktası birleştirilir. Bu karşılaştırma işlemi her bir engel noktası için 180 derecelik tarama sonucu elde edilen diğer tüm veriler arasında yapılır.

Değerlendirme prosesi bitirildikten sonra o tarama için toplanan tüm bilgiler OYA'nın konum bilgisi ile beraber kalıcı veri tabanına aktarılır.

```
procedure TForm1.Button13Click(Sender: TObject);
var
son:integer;
begin
tablokal.First;
tablotoplam.Last;
son:=tablotoplam.fieldbyname('sayi').asinteger;
while not tablokal.Eof do begin
son:=son+1;
tablotoplam.Insert;
tablotoplam.fieldbyname('sayi').asinteger:=son;
tablotoplam.fieldbyname('sayi1').asinteger:=tablokal.fieldbyname('sayi').asinteger;
tablotoplam.fieldbyname('1').asinteger:=tablokal.fieldbyname('1').asinteger;
tablotoplam.fieldbyname('2').asinteger:=tablokal.fieldbyname('2').asinteger;
tablotoplam.fieldbyname('3').asinteger:=tablokal.fieldbyname('3').asinteger;
tablotoplam.fieldbyname('4').asinteger:=tablokal.fieldbyname('4').asinteger;
tablotoplam.fieldbyname('5').asinteger:=tablokal.fieldbyname('5').asinteger;
tablotoplam.Post;
tablokal.next;
end;
tablokal.close;
tablokal.Open;
tablokal.First;
while not tablokal.Eof do begin
tablokal.delete;
end;
end;
```

Geliştirilen SCADA yazılımı sayesinde kullanıcı görmediği bir alanın haritasını çıkaracağı için Görüntüleme bölümü kullanıcıya bulunduğu alan

hakkında bilgi verecektir. Kullanıcı öncelikle OYA'nın konumu hakkında görsel tarama sayesinde bilgi alacak ve gideceği yön ve mesafeyi bu konum bilgisini referans alarak belirleyecektir. Kullanıcı bu tarama işlemi sonucunu ve bu tarama işlemine göre OYA'nın bulunduğu konumu Görüntüleme bölümü içerisinde gözleyecektir. Görüntüleme bölümü üç farklı ölçüm sisteminden aldığı sonuçları yorumlayarak oluşturulmaktadır. Görüntüleme sistemi motorlar üzerinde bulunan enkoderler sayesinde ölçülen (gidilen) mesafeyi; görüntüleme ekranını fazla büyütmemek için yarı yarıya düşürterek görüntüleme ekranına yansıtmaktadır. Görüntüleme ekranı OYA'nın hareketlerini $\frac{1}{2}$ zoom out görüntü açısı ile göstermektedir. Dönme hareketi ve engellerin tespiti için de yine aynı görüntüleme açısı ile görüntü vermektedir.

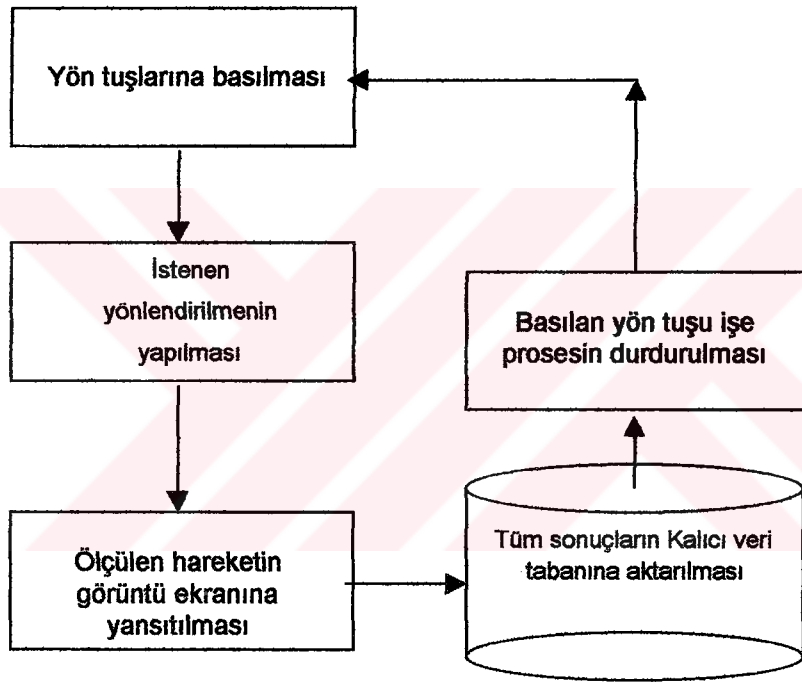


Şekil 4.11. Görüntüleme Bölümü İçerisinde OYA ve Engel Çıktıları

4.2.4 Elle Kumanda

OYA'nın kontrolünü sağlayan SCADA yazılımı içerisindeki Elle Kontrol bölümü direkt olarak OYA'nın hareketlerini göz kararı denetlemek için

geliştirilmiştir. Belirli limit değerler olmadan kontrol imkanı sağlamaktadır. Elle kumanda sonucunda OYA tarafından gerçekleştirilen hareketler görüntü ekranına ve veri tabanına kaydedilmektedir. Bu bölüm içerisinde Tarama kısmına müdahale edecek bir elle kumanda kısmı bulunmamaktadır. Bu bölüm içerisinde bulunan yön tuşları sayesinde direk olarak OYA'nın kontrolünü sağlayan bir program icra edilmektedir. Bu yön tuşları sayesinde OYA'a yaptırılan hareketler yine bu bölüm içerisinde bulunan mesafe ve açı değer gözlem bölümleri içerisine yansıtılmaktadır. OYA'nın hareketleri görüntü ekranına ve veri tabanına yansıtılmaktadır.

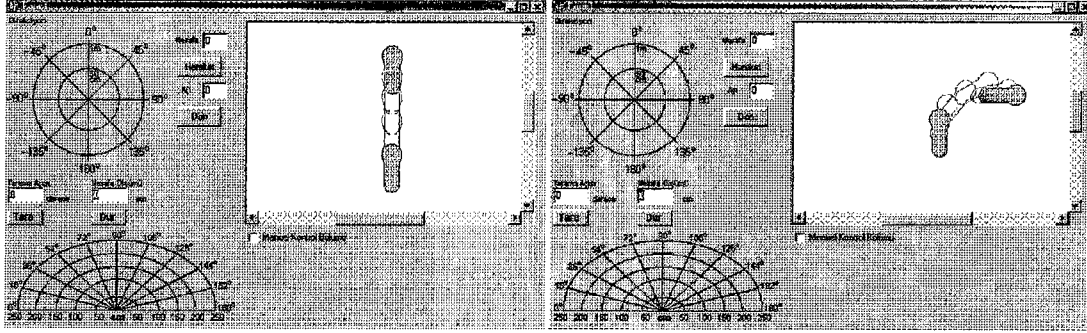


Şekil 4.12. Elle Kontrol Bölümü İçerisindeki Akış Diyagramı

SCADA'nın Elle kumanda bölümü tamamen Yön Ayarlama bölümünde kullanılan yazılım sayesinde yapılmaktadır. Fakat bu kez yapılan hareket için bir sınır değeri belirlenmediği için hareket yön verme düğmelerine basılı tutulduğu müddetçe devam eder. Encoderlerden gelen bilgi bu kez yapılan hareketin türüne göre açı veya mesafe olarak Elle Kumanda bölümüne ve görüntüleme ekranına aktarılır.

Her SCADA yazılımında olduğu gibi OYA kontrolü için geliştirilen SCADA yazılım içerisinde de elle kontrolü sağlayan bir bölüm bulunmaktadır.

Bu bölüm normal şartlar altında SCADA yazılımının içinde gözükmez. Elle kumda kısmı "Manuel Kontrol Bölümü" seçeneği (CheckBox) aktif hale getirilerek kullanılabilen opsiyonel bir seçenek olarak SCADA yazılımı içerisinde bulunmaktadır. Bu elle kontrol seçeneği beklenmeyen durumlar, OYA'ın takılma veya geçersiz işlem yürütme hallerinde kullanmak için SCADA içerisine katılmıştır



Şekil 4.13. Elle Kontrol Bölümü İçerisinde Oya Hareketi ve Dönüşü

5 SONUÇ

Bu tez çalışmasında sonucunda; tasarlanan OYA prototipi sayesinde, kendi yörüngesini “keşfederek” belirleyebilen ve çevresinin bir tür “haritasını” çıkartan Otomatik Haritalama Sistemi için kullanışlı bir mantık geliştirilmiştir.

Geliştirilen OHS algoritması sayesinde OYA'nın takip edeceği yol üzerinde bulunan engeller tespit edilerek, OYA'ya izleyeceği yörüngeyi, belirlenen engel noktalarının sınırlandırdığı koridor esas alınarak, tekrar düzenleme imkanı sağlanmıştır. Temel olarak OYA, OHS yardımı ile ulaşmak istediği noktaya gidebilmesi için kendine bir koridor belirlemesi için gerekli veri oluşturulmuştur.

OHS'ni oluşturabilmek için gerekli olan kontrol algoritması, tasarlanan bir SCADA aracılığı ile sağlanmıştır. Geliştirilen SCADA yazılımı sayesinde OYA kontrol ve karar verme süreci için gerekli bilgi ölçümü ve aktarımı gerçekleştirilmiştir.

SCADA yazılımı için gerek duyulan veri akış ihtiyacı , dağıtık veri toplama sistemlerinin kontrolünde kullanılan ağ teknolojilerinden biri olan Kontrolör Alan Ağı üzerinde gerçekleştirilmiştir. Kullanılan KAA sayesinde, OYA prototipi üzerinden gerçek zamanlı veri aktarımı ve kontrolü sağlanmıştır. KAA'nın bu tip bir uygulamanın duyduğu ihtiyaçlara ortaya konulan öngörüler çerçevesinde cevap verdiği gözlenmiştir.

OHS algoritmasının, OYA'nın kendi yörüngesinin çevre koşullarına göre baştan tanımasını sağlayacak mantığı oluşturduğu tespit edilmiştir.

Bu bilgi birikimine dayanılarak mevcut teknoloji, OYA görevleri için iyi bir platform oluşturmaya adaydır.

Kontrol bilgisayarı ile veri bağlantısının telsiz teknolojilerle desteklenmesi halinde daha geniş bir alanda ileri görevlerde kullanılabileceği düşünülmektedir.



Ayrıca sistem, yazılımının geliştirilmesiyle tamamen bağımsız hareket edebilecek bir görev aracına da dönüştürülebilecektir.



KAYNAKLAR



- 1- Advantech, ADAM 5000/CAN User Manual, Taiwan, 1997.
- 2- Advantech, Help on: ADAM 5000 DeviceNet Driver, 1999.
- 3- Advantech, Total Solution for PC-Based Industrial Automation, Solution Guide IA Vol.81, 1999.
- 4- Bosch, R., CAN Specifications Version 2.0, BOSCH, Stuttgart, 1991.
- 5- Dunkin, A., Automated Guided Vehicle Systems, Industrial Engineering 1994
- 6- CAN in Automation , <http://www.can-cia.de/>, 1999.
- 7- Fredriksson, L.B., Controller Area Networks and The Protocol CAN For Machine Control Systems, Kvaser AB, <http://www.kvaser.se>, Sweden, 1996.
- 8- Goover, M., Automatic Production Systems and Computer Integrated Manufacturing, Kulwer Academic Publisher, 1987
- 9- Hammond, G., AGVS at Work, IFS Publishing Com, 1986
- 10- Heragu, S., Facility Design, PWB, Publishing Com, 1996
- 11- Honeywell, Smart Distributed System (SDS) , <http://www.honeywell.com/sensing/prodinfo/sds/> , 1999.
- 12- Kvaser AB, Kvaser CAN Kingdom Pages (CAN), <http://www.kvaser.se/canking/index.htm>, Sweden, 1999.
- 13- Lennartsson, K. and Fredriksson, L.B., Fundamental Parts in SDS, DeviceNet and CAN Kingdom, Kvaser AB, Sweden, 1996.
- 14- McLaughlin, R.T. and Khoh, S.B., Controller Area Network (CAN) For Industrial Applications, <http://www.warwick.ac.uk/devicenet/indus.htm>.
- 15- Miller Freeman Inc, CAN Solutions Directory 1998, <http://www.directories.mfi.com/embedded/can/> , 1998.
- 16- Newcastle University Robotics and Automation Group (NURAG), The CANOpen Protocol, <http://www.ncl.ac.uk/~nrauto/canproto.htm> , 1997.
- 17- Nilsson, S., Controller Area Network - CAN Information, <http://www.algonet.se>, 1997.
- 18- Olsson, G., Piani, G., Computer Systems for Automation and Control, Prentice Hall Inc, New York, 1992.

19- Rigby, W. H. And Dalby, T., Computer Interfacing a Practical Approach to Data Acquisition and Control, Prentice-Hall Inc., New Jersey, 1-5, 152-173, 1995.

20- Schofield, M. J., Controller Area Network – CAN Application Layers, <http://www.compulink.co.uk/>, 1998.

21- Tang, K. H., Introduction to DeviceNet, University of Warwick, <http://www.warwick.ac.uk/devicenet/dnet1.htm> , 1999.

22- Tompkins, J. A., Facilities Planning, John Wiley and Sons, 1996.

23- Wierzbicki, J., Programming The 8250 UART, <http://www.perplexed.com>, <http://www.byterunner.com/16550.html>.



ÖZGEÇMİŞ

Ahmet BERKAY, 01 Haziran 1976 tarihinde Kuzey Kıbrıs Türk Cumhuriyetinin Magosa şehrinde dünyaya geldi. İlk, orta ve lise eğitimini Gazi Magosa Türk Maarif Koleji'nde tamamladı. 1993 yılında Kocaeli Üniversitesi, Elektrik Mühendisliği Bölümünde girdi. 1998 yılında Kocaeli Üniversitesi Elektrik Mühendisliği Bölümünden mezun oldu. Aynı yıl Gebze Yüksek Teknoloji Enstitüsü'nün yüksek lisans sınavını kazandı. Halen bu kurumda araştırma görevlisidir.



EK-1

1. KONTROLÖR ALAN AĞI (KAA)

Kontrolör Alan Ağı (KAA) özellikle “akıllı” aletler ile, algılayıcı (sensor) ve/veya eylemlendirici (actuator) gibi “sıradan” aletlerin bir sistem veya alt-sistemler için de uyumlu olarak haberleşmesi için geliştirilmiş seri veri yolu sistemidir. Daha kesin olarak Kontrolör Alan Ağını “KAA” dağıtık elektriksel modüller üzerinden haberleşebilen çok kanallı seri haberleşme sistemi olarak ifade edilir.

Günümüz teknolojisi içerisinde KAA'nın güvenilir, hızlı ve yaygınlaşmakta olan bir teknoloji olduğunu görülmüştür. KAA, henüz ülkemizde pek yaygınlaşmamasına rağmen özellikle Avrupa'da yaygınlaşmaktadır ve 500'den fazla üretici KAA uyumlu malzeme üretmektedir. Bu yüzden KAA'e endüstriyel uygulamalarda geleceğin teknolojisi de denilebilir.

Kontrolö Alan Ağı'nın seçim sebebi aşağıda açıklanmaktadır :

- Geniş olarak kabul edilmiş bir standarttır.
 - Avrupa otomotiv endüstrisi tarafından yıllarca kullanılan yerleşik bir standarttır.
 - Amerika'da ise endüstriyel alanda giderek kabul gören bir standarttır.
- Karalıdır.
 - Bir çok sıra dışı şart karşısında uygulamasındaki farklılık çok azdır.
 - Ayarlanması kolaydır.
 - Hata bulma kapasitesi yüksektir.
 - İki telli bağlantı için hata toleransı oldukça düşüktür.
- Yazılım ve donanım olarak geniş bir desteğe sahiptir.
 - Uygulama katmanları ve sürücüler için bir çok yazılım geliştirilmiştir.

- KAA veri yolu denetleyicileri ve geliştirme kitleri bulunmaktadır.
- KAA kullanıcıları gurupları tarafından sürekli geliştirilmektedir.



1.1. KAA Uygulamaları

Günümüzde KAA, mikro kontrolörler içerisinde kullanılabildiği gibi akıllı cihazlar üzerinde açık bir haberleşme sistemi olarak da kullanılmaktadır.

Orijinal KAA seri veri yolu sistemi otomobiller üzerinde kullanılmak için geliştirilmiştir fakat zaman içinde önce endüstriyel uygulamalar sonrada binalar, gemiler, robotlarda ve en son olarak ta tıbbî alanda kullanılmaya başlamıştır. Bu yaygınlaşmada en önemli sebep ise KAA sisteminin çok güvenli oluşudur.

1.1.1. Araçlardaki Seri Haberleşme İhtiyacı

Taşıtlarda seri haberleşme her biri farklı gereksinim ve amaca sahip 4 temel uygulama şeklinde bulunmaktadır. Seri haberleşme ağı, motor zamanlaması, transmisyon, şase ve frenleri kontrol eder. Veri oranı, gerçek zamanlı uygulamalar uygun olarak, 200kBit/s - 1Mbit/s arasındadır.

Ağın birer parçası ve taşıtı daha konforlu hale getirilmesini sağlayan, şase elektroniği ve genel elektronik sistem (havalandırma, ışıklar, merkezi kilit, emniyet kemerleri ve ayna ayarları gibi) genellikle 50kBit/s veri oranında bulunmaktadır.

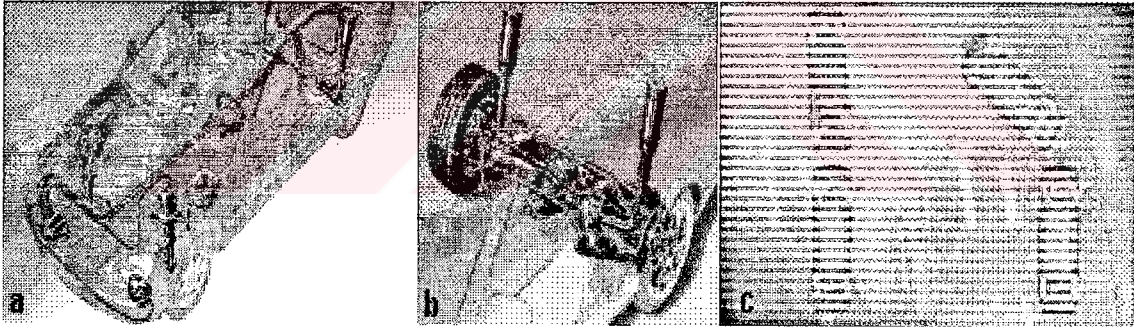
1.1.2. KAA'in Taşıtlarda Kullanımı

Günümüzde kullanılan taşıtlar çok sayıda değişik amaca hizmet eden elektronik kontrol sistem bulunmaktadır. Fakat kullanılan bu elektronik kontrol sistemleri eş zamanlı ve gerçek zaman ölçeğinde kontrol edilip haberleşme ihtiyacı duymaktadır. Önceleri bu sistemler arasındaki bağlantı

veri sinyallerinin direkt olarak taşınması ile gerçekleştirilmiş fakat araları sayı ve kontrol ihtiyacı giderek karşılanamaz duruma gelmiştir. Bu sebepler ve gelecek için yapıla öngörüler Bosch'un, Kontrolör Alan Ağını (Kaa, Controller Area Network "CAN") geliştirmesine sebep olmuştur. Bosch'un geliştirdiği bu ağ uygulaması daha sonra uluslar arası bir standart (ISO 11898) haline gelmiş ve birçok yarı iletken üreticisi tarafından kullanılmıştır.

Kullanılan bu sistem ile otomobil içerisinde yer alan tüm aletler, merkez Cpu'ya ek bir yük getirilmeden bir birleri ile rahatlıkla haberleşebilmektedirler. Yine otomobil içerisinde oluşan elektromagnetik dalgalanmalardan etkilenen veri bu sistem ile korunup düzeltilmektedir.

Kaa'nın taşıtlardaki kullanımı için en iyi örneği Mercedes-Benz oluşturmaktadır. Mercedes-Benz yarış otomobillerinden ağır iş makinelerine kadar tüm ürün yelpazesi içerisinde dağıtık veri toplama sistemlerinin kontrolü için KAA kullanmaktadır (Şekil 1.1).



Şekil 1.1 Mercedes-Benz Fren ve Tekerlek Aksamı için Kullandığı KAA

1.1.3.KAA'in Endüstride Kullanımı

Araçlardaki veri yolu sistemi ile endüstriyel uygulamalardaki field bus sistemleri karşılaştırıldığı zaman çok büyük benzerlikler gözlenmektedir. Her iki alanda da düşük maliyete, yüksek elektromagnetik şartlara, gerçek zamanlı uygulamalara ihtiyaç duyulmaktadır. KAA sisteminin Mercedes-Benz's "S" Class ve US commercial vehicle'in kullanılması sonucu piyasada

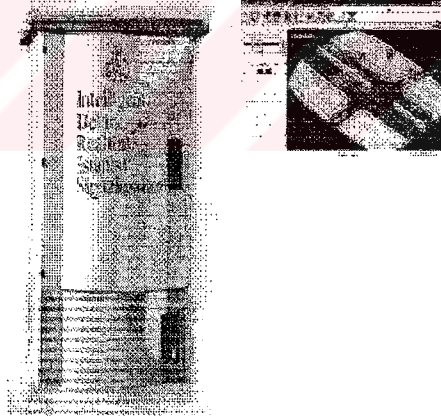


bulunan diğer şirketlerin ilgisi de bu alan kaymaya başlıyor. Bu öncülüğle sonra tekstil sektörü, özel amaçlı makine üreticileri, asansör üreticileri ve son olarak tıbbi cihaz üreticileri bu sektöre kaymışlardır.

Özellikle tekstil sektörü KAA sistemi için bir öncü durmandadır. 1990'ların başında ilk olarak üretilen KAA sistemine sahip tekstil makinesinden sonra sektör içerisinde KAA kullanımı o kadar yaygınlaşmıştır ki artık KAA hakkında önemli bir otorite olan "KAA Textile Users Group" oluşumuna kadar ulaşmıştır.

KAA sistemi özellikle tıbbî makine üreticileri arasında güvenilir ve sağlam yapısından dolayı hızlı bir yaygınlaşma göstermiştir.

KAA sisteminin kullanıldığı en önemli endüstriyel uygulamalar arasında Mercedes-Benz ve Hundai firmalarının üretim bantları sayılabilir. Türkiye'de KAA sisteminin kullanıldığı en önemli proje ise İstanbul belediyesi tarafından gerçekleştirilen akıllı trafik kontrol uygulamasıdır.



Şekil 1.2 İstanbul Belediyesi Tarafından Gerçekleştirilen KAA Sistemi

1.1.4. KAA Ürünleri

Dünya çapında KAA ürünlerinin standardizasyonunu; 1992 martında uluslar arası, kullanıcı ve üretici organizasyonu olarak kurulan CAN in Automation (CiA) sağlar. Ticari bir kuruluş olmayan CiA, 250 üyesi ile

pazarlama, teknik eğitim, standartlar ve çeşitli geliştirme faaliyetlerinde bulunmakta, KAA hakkında dünya çapında forumlar ve seminerler düzenlemektedir.



Bu organizasyonun desteklediği yüzlerce çeşitlilikte KAA ürünleri vardır. Örneğin:

- ISA (PC uyumlu), PCMCIA (Laptops), paralel portlar ve Ethernet için adaptörler.
- KAA donanımlı sensörler (limit, fotoelektrik göz...).
- KAA donanımlı eylemciler (hidrolik, pünomatik, motor, röleler...).
- OKAA tekrarlayıcılar (repeaters).

CiA'na üye firmaların KAA ürünlerinin listesi Çizelge 2.1.de verilmiştir

Çizelge 1.1 KAA Ürünleri Listesi.

KAA Protokol Kontrolör İşlemcileri	
Advantech	ADAM5000 Series
Motorola	Motorola's KAA Microcontrollers
Motorola	MC68HC08AZ0/16/24/32 & MC68HC908AZ60 MSKAA08 On-Chip Controller Area Network
Motorola	68HC12 16-Bit Microcontrollers with On-Chip KAA Support
Motorola	68376 32-Bit Microcontroller with On-Chip KAA Support
Motorola	MPC555 Microcontroller with Flash EEPROM
Motorola	Motorola OSEK
NEC Electronics	μPD78(F)081x 8-Bit Microcontrollers with DKAA
NEC Electronics	μPD78(F)082x 8-Bit Microcontrollers with DKAA
NEC Electronics	μPD78(F)094x 8-Bit Microcontrollers with DKAA
NEC Electronics	Atomic 32-Bit RISC Microcontrollers
Philips Semiconductors	8xC592/8xC598 8-Bit Microcontroller with On-Chip KAA

Philips Semiconductors	SJA1000 Stand-alone KAA Controller and XA-C3 CMOS Single-Chip 16-Bit Microcontroller
Philips Semiconductors	XA-C3 CMOS Single-Chip 16-Bit KAA 2.0B Microcontroller
Siemens	C164CI General-Purpose High-Performance MCU with OTP on Chip
Siemens	C167CR Highly Integrated Microcontroller with On-Chip KAA Module
Siemens	C505C Microcontroller with Complete KAA Capability
Siemens	C515C Multipurpose Microcontroller with On-Chip Full KAA Module
Siemens	SAE 81C90/91 Full KAA Controller
ST Microelectronics	ST10F167 Microcontroller with KAA 2.0B
ST Microelectronics	ST725x 8-Bit Microcontroller Family with KAA 2.0B
KAA İletici Çipleri	
Alcatel Microelectronics	MTC-3054 KAA Interface Chip
Philips Semiconductors	82C250/82C251/82C252 KAA Transceivers
Donanım Araçları	
Dearborn Group	Gryphon Ethernet-to-KAA
Hitex Development Tools	Toolbox for KAA Applications
I+ME Actia	KAA-WinEMU and NetTest III
I+ME Actia	Advanced KAA Technology
Ixxat Automation GmbH	Analyzing Tools, Interfaces, and Higher Layer Protocols
Nsi SA	Traminator KAA
Vector KAAtech, Inc.	KAAalyzer
Vector Informatik GmbH	Vector KAA Products
Yazılım Araçları	
Keil Software	C51 Compiler Version 5.50
Keil Software	C166 Compiler Version 3.0
Tasking	C166, KAA, and Tasking: The Integrated Solution
Weidmüller Connex GmbH & Co.	DIAtool Software

Weidmüller Connex GmbH & Co.	DIAvis/Wizcon
Kart-Seviyesi Ürünler	
Arcom Control Systems	AIM104-KAA PC/104 Single Channel KAA Controller with DeviceNet Tools
National Instruments	KAA/DeviceNet Interfaces
National Instruments	AT-KAA, AT-KAA/2 Interfaces for ISA
National Instruments	PCI-KAA, PCI-KAA/2 Interfaces for PCI
Phytec	PHYTEC KAA Single Board Computers and Software
SBS GreenSpring Modular I/O	IP-DeviceNet Interface IndustryPack Module for ISA, PCI, CompactPCI, VME, and PC/104
Arabirimler ve Algılayıcılar	
Balluff	BIS-C-603-U2-DN ID Systems Interface
Balluff	Micropulse - BTL Linear Position Transducers
RM MICHAELIDES Software & Electronic GmbH	KAAview Interpreting and Control Unit to Connect KAA Fieldbuses to PCs
KAA Arabirimler, Kontrolör ve Sürücüler	
Contemporary Controls	EXTEND-A-BUS Series Fieldbus Extender for DeviceNet or Smart Distributed System
Infranor	SMTBD Digital Servo Amplifier
Wago Corporation	WAGO I/O System

1.2. Kontrolör Alan Ağına (KAA) Genel Bakış

KAA, her KAA bağlantı noktasının (node) veri iletme ve yine her KAA bağlantı noktasının bu veriyi veri yolu içerisinde alma hakkına anlamını gelen çok efendili (multi-master) seri veri yolu sistemidir. Gerçek zamanlı seri veri yolları, uluslar arası standartlar içerisinde ISO 11898 konu başlığı altında bulunmaktadır ve KAA sistemin en alt iki katmanı ISO/OSI referans modeli içerisinde yer almaktadır. KAA ağı içerisinde abone veya istasyon adresleri yerine polarize edilmiş mesajlar iletilir. Bir verici yollayacağı mesajı KAA ağının her bağlantı noktasına yollar yani mesajını ağa gönderir



(broadcasting). Her alıcı kendisine iletilen bu mesajın çerçeve adresinden, bu mesajı kabul edip etmeyeceğine karar verir. Yukarda sayılan özellikleri herhangi bir cihaza uygulamak, KAA ara yüzey elektroniği sayesinde çok rahat ve ekonomik bir çözümdür.

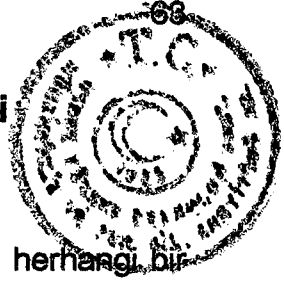
Her KAA mesajı 0 ile 8 bit arası bir kullanıcı bilgisi ile iletilir. Bu kullanıcı bilgisi bölümlere ayırma metodu ile daha da büyütülebilmektedir. Bir KAA sistemi için maksimum iletim hızı 1 Mbit/s'dır. Bu hız 40 metreye kadar olan mesafeler için kullanılabilir. Mesafe uzunluğu ile iletişim hızı ters orantılı olarak azalmaktadır. 500 metre için iletişim hızı 125 kbit/s düşerken 1 km için hız 50 kbit/s olmaktadır. KAA, teorik olarak bir tek ağda 2032 aygıtı bağlayabilir (bir nokta bir belirteç varsayılarak). Bununla birlikte donanımın pratikteki limitine bağlı olarak bir ağda sadece 110 nokta bağlanabilir. Bu özelliklerine ek olarak, hata sınırlama ve hata tespiti özelliği, onu gürültülü çevrelerde daha güvenilir yapmaktadır.

Mikrokontrolörlerle haberleşmeyi sağlayacak düşük maliyetli kontrol çiplerine, KAA veri link tabakalarının adaptasyonu 1989'dan beri gerçekleştirilmektedir. Günümüzde 50 değişik KAA kontrol çipi 15 değişik firma tarafından üretilmektedir. Hemen hemen Avrupa'da üretilen tüm yolcu, yük ve off-road araçlarında KAA çipleri kullanıldığından dolayı üretici firmalar tarafından KAA çiplerinin en az 10 yıl daha üretileceği açıklanmaktadır [CiA konferans,1999]. 1999 baharı itibarı ile dünya üzerinde 150 milyon adet bağlantı noktası kurulmuştur.

1.3. Kontrolör Alan Ağının (KAA) Teknik Özellikleri

Bu bölüm içerisinde Kontrolör Alan Ağının (KAA) teknik verileri ve standartları genelden özel doğru incelenmiştir.

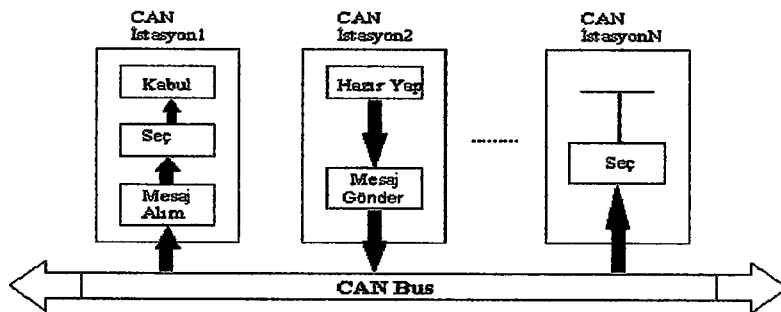
1.3.1. Kontrolör Alan Ağı'nın Veri Değişim Prensipleri



KAA ağı içerisinde bir veri iletildiği zaman iletilen bu veri herhangi bir iletilme adresi veya gideceği istasyon adresi barındırmaz bunun yerine tüm ağ tarafından çözümlenebilecek bir tanımlayıcıya sahiptir. Bu tanımlayıcı iletilen mesajın içeriği ile birlikte önceliğini de belirlemektedir. Bu öncelik ise bir çok cihaz, ağ içindeki veri yoluna giriş için yarışırken çok önemli bir belirleyici haline gelir.

Her hangi bir CPU istasyonu başka bir istasyona veya başka istasyonlara veri yollamak istediği zaman yollamak istediği veriyi sadece KAA çipine iletir. CPU sadece iletişimi başlatması (Make ready) yeterlidir geriye kalan belirleyicinin eklenmesi ve mesajın derlenmesi işlemi KAA çipi tarafından yapılır. KAA çipi tarafından yollanan (Send Message) verinin veri yoluna ulaşmasından sonra KAA ağı üzerindeki diğer tüm istasyonlar bir alıcı (Receive Message) durumuna gelirler. İletilen veri daha sonra bütün istasyonlar tarafından doğru olarak alınır ve bir uygunluk testinden (Select) geçirilirler. Test edilen bu veri uygun olan imzaya sahip ise istasyon tarafından işlenirler (Accept), fakat test edilen veri uygun olan imzayı taşıyor ise ret edilirler.

KAA ağına herhangi bir yeni donanım eklemek söz konusu olduğu zaman kullanılmakta olan yazılım veya donanımda her hangi bir değişiklik yapma ihtiyacı duyulmaz. Çünkü iletişim protokolü eklenecek olan yeni cihazlar için herhangi bir fiziksel gidici adresine ihtiyaç duymaz.



Şekil 1.3. İstasyonların KAA Ağı İçerisindeki İletişim Protokolü

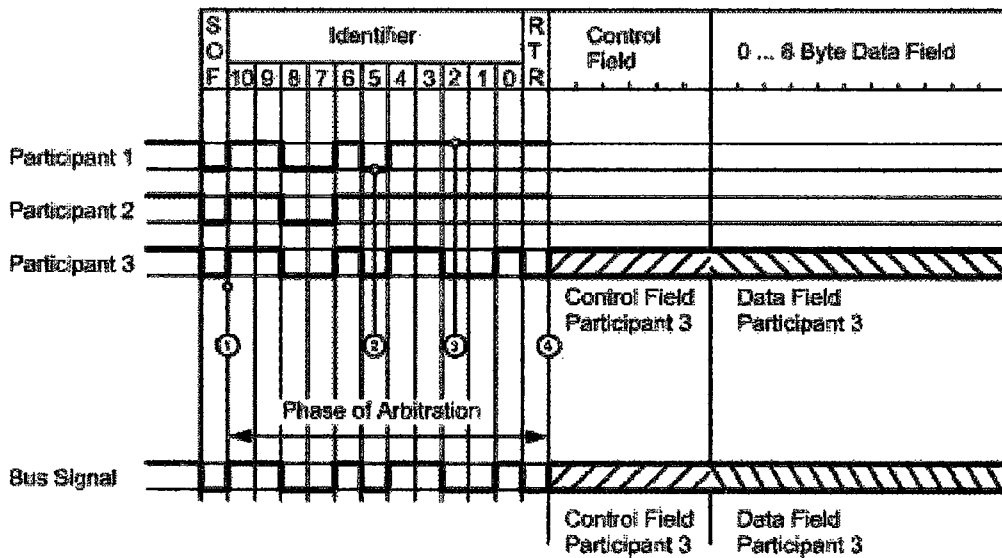


1.3.2. Yıkıcı Olamayan Hakemlik Prensibi

Gerçek zamanlı bir proses içerisinde veri çok çabuk olarak ilerlemek zorundadır. Bir çok istasyon sürekli olarak ve hemen hemen aynı anda veri iletmek isteği halinde olduğu düşünülürse bu iletim için sadece 1 Mbit/s bir veri yolu değil hızlı bir veri yoluna ihtiyaç vardır.

Veri iletimi içerisinde iletilmesi acil olan veriler sistem içerisinde belirleyicileri sayesinde ayrılırlar. Öncelikler sistem dizaynı sırasında belirlenmektedir sistem kurulduktan sonra dinamik olarak belirlenemezler. En düşük binary sayıya sahip belirleyici en yüksek önceliğe sahiptir.

Bir bağlantı noktasının bir başka bağlantı noktasının veri iletim çerçevesinin engellememesi için veri iletim çerçevesinin belirleyici bölgesi RTR biti iletişim sırasında veri yolşu sinyali temel alınarak izlenir (Şekil 1.4). Bu izlem sırasında dominan veri iltim bitine sahip iletişim veri çerçevesi alıcılara yollanır. Diğer veri çerçevesi kesilerik tekrar yollanmak üzere bekletilir.



Şekil 1.4. Yıkıcı Olamayan Hakemlik Prensibi



Veri yoluna giriş çakışmaları, veri seviyesini bit ,bit ,bit ,bit ,bit istasyonların gözlemciliğinde belirleyiciler içinde bulunan akıllı-biçimlenmiş sayesinde çözülmektedir. Veri iletim yolu içerisinde yer kapma ,yer kapma kaybeden bütün istasyonlar daha yüksek bir öncelik olarak tekrar iletili haline gelmekte ve veri yolu içerisinde yer bulana kadar denemeye devam etmektedirler.

1.3.3. Veri Yolu Tahsisinin Etkinliği

Veri yolu tahsis sisteminin etkinliği, seri veri yolu sistemine sahip mevcut bir çok uygulama temel alınarak saplanmıştır. Hangi veri yolunun hangi uygulama için uygun olduğunu belirlemek, literatürdeki veri yolu tahsis metotlarını sınıflandırarak rahatlıkla yapılabilir. Bu seçim aşağıdaki sınıflar arasında yapılmaktadır.

Veri yolu tahsisi belirli bir takvime bağlanır. Tahsis her katılımcı için o an veri yoluna ihtiyacı olup olmadığına bakılmaksızın, maksimum sürede ve sıra ile yapılır (Örnek olarak, token slot veya token passing).

Veri yolu tahsis ihtiyaca göre belirlenir. Tahsis katılımcının veri yoluna girme isteği göz önüne alınarak meydana getirilir (Örnek olarak, CSMA, CSMA/CD, flying master, round robin veya bitwise arbitration).

KAA sistemlerinde veri yolu tahsisi tamamen istasyonun veri transfer etme isteğine dayanmaktadır. Bu da KAA sistemlerinde, veri yolu tahsisi ihtiyaca göre belirlenmesi anlamına gelmektedir.

Veri yolu tahsisi yada bir başka değişle veri yoluna giriş şekilleri sıralanacak olursa :

- Yıkıcı olmayan veri yoluna giriş. Bu metot ile veri yolu, isteğin gelmesi ile birlikte hemen yada belirli bir zamanlama göz önüne alınarak sadece bir istasyon için ayrılır. Bu da her istasyon tarafından yapılacak olan veri yolu girişini için belirli veri yolu tahsisi olması anlamına gelir.

(Örnek olarak, token slot, token passing, round robin veya bitwise arbitration).

- Yıkıcı olan veri yoluna giriş. Bu metot ile bir veya daha fazla istasyonun veri yolu girişi için yapacakları eş zamanlı istekler tüm isteklerin iptal edilmesi ile sonuçlanır yani veri yolu tahsisi diye bir şey söz konusu değildir (Örnek olarak, CSMA/CD veya Eternet).

KAA ağı içerisinde oluşan bütün iletme isteklerini ; en düşük veri transfer oranı belirleme baskısı altında ; KAA protokolü aracılığı ile değişik istasyonlar tarafından gerçekleştirilen eşzamanlı veri yolu girişi istekleri karşısında her bir veri yolu girişi isteği için belirli bir veri yol tahsisi garanti enden , bir veri yolu tahsis metodu ile yerine getirmek zorundadır. Akıllı-bit hakemliği metodu, iletişim bekleyen bir çok istasyonun iletişim isteklerini çakışmaması için sadece mesajın tanımlayıcısını kullanmaktadır. Bunu da herhangi bir veri yolu giriş isteği periyodunda en az 13 (standard format) veya 33 (extended format) bir periyodu kullanarak yapmaktadır. CSMA/CD metodu tarafından uygulanan akıllı mesaj hakemliği yöntemine nazaran yıkıcı olamayan hakemlik metodu çakışma için gerekli olan çözümü, gerekli olmayan mesajları iletmemekle sağlanmalıdır.

Yıkıcı olmayan veri yolu girişi bir ileri sınıflandırmaya tabi tutulursa, sistem içerisindeki kontrol mekanizması bir (merkezleştirilmiş) veya birden fazla (merkezleştirilmemiş) olasılığı göz önünde tutularak,

- Merkezleştirilmiş veri yolu giriş kontrolü.
- Merkezleştirilmemiş veri yolu giriş kontrolü.

olarak ikiye ayrılabilir.

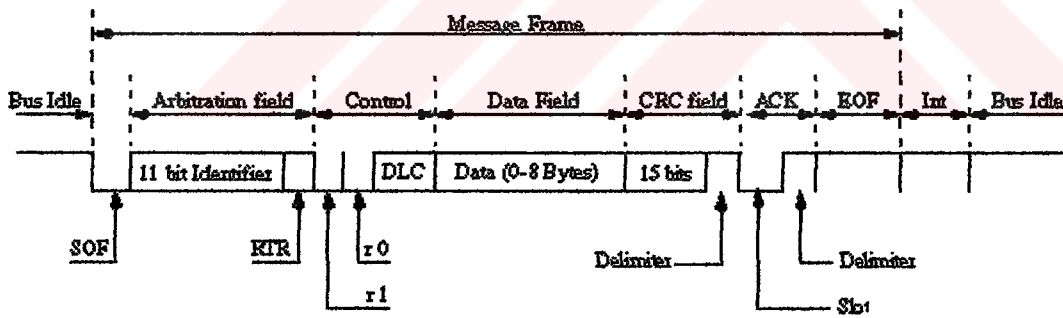
Belirli istasyonlar ile dizayna edilmiş bir haberleşme sistemi master istasyon (merkezleştirilmiş veri yolu giriş kontrolü) devre dışı kalsa dahi etkinliğini devam ettirebilecek bir strateji sağlamalı. Bu tanım, hat yönetimi stratejisi açısından gerçekleştirilmesi pahalı ve zor olmasının yanında merkez istasyonun görevlerini başka bir istasyona devretmesi çok zaman alması gibi



bir çok dezavantaja sahiptir. Bu sebeplerden dolayı ve merkez istasyonun (dolayısı ile tüm haberleşme sisteminin) güvenilirliği problemlerinden kaçınmak için KAA protokolü merkezleştirilmemiş veri yolu girişi kontrolü ile gerçekleştirilmektedir. Bütün önemli haberleşme mekanizmaları, haberleşme sisteminin yüksek ihtiyaçlarını karşılaya bilme yolunun tek çözümü olarak veri yolu girişi kontrolüne sahiptir.

Özet olarak KAA' in, merkezleştirilmemiş veri yolu girişi kontrolü ile yıkıcı olmayan veri yolu girişi sayesinde, mümkün olan en düşük veri yolu oranında en yüksek yararlı veri oranı ile trafiğe bağlı bir veri yolu tahsisi yerine getirdiğini söyleyebiliriz. Veri yolu tahsisinin etkinliği veri yolunun istasyonlar tarafından iletim isteği gelmeyinceye kadar kullanılmaması ile yükseltilmektedir.

Bu istekler mesajın tüm sistem için önemi göz önüne alınarak sıraya konmaktadır. Bu özellikle aşırı yüklenme durumunda yapılmaktadır.



Şekil 1.5. KAA 2.0A Mesaj Çerçevesi

1.3.4. Mesaj Çerçeve Şekli

KAA protokolü iki mesaj çerçeve şeklini desteklemektedir. Desteklediği bu iki mesaj çerçevesi arasındaki tek fark ise bu mesaj çerçevelerinin tanımayıcılarıdır (ID). Standart formata ID uzunluğu 11 bit, uzatılmış (extended) formatta ise ID uzunluğu 29 bittir. İletilecek mesaj çerçevesi 7 ana kısma ayrılmıştır.

Standart formattaki mesaj "start of frame" biti ile başlar bu biti, mesajın bir veri çerçevesi mi yoksa veri içermeyen "remote frame" istek çerçevesi mi olduğunu belirten "RTR" (remote transmission request) bitini de içeren tanımlayıcı "arbitration field" biti takip eder. Kontrol alanı "control field", mesajın Standart formatını 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 kontrolu 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.

1.3.5. Hata Tespit ve İşaretlenmesi

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.

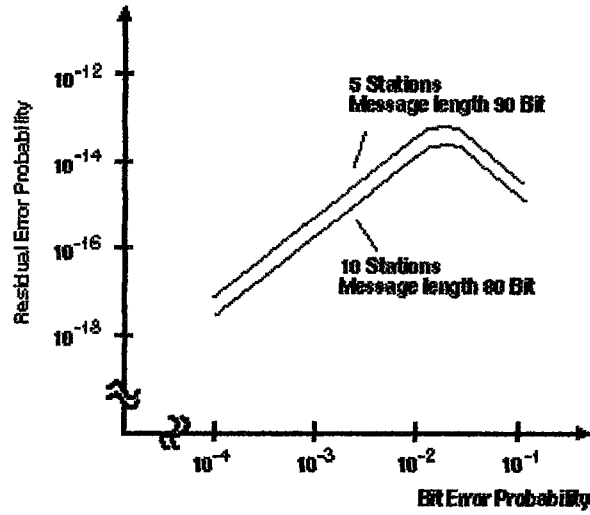
- **Monitoring** : Bu yöntemde iletilen mesajların hatalarını kontrol etmek için veri yolu sinyalleri izlenir. İletim yapan her node yine veri yolu seviyesini kontrol eder yani gönderilen bit ve alınan bit arasındaki farkı gözler. Bu iletim sırasında oluşacak bütün global ve yerel hataları tespit eder.
- **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 iletilme pozisyonuna "automatic repeat request" geçer. Bir kural olarak tekrar iletilme durumu hata tespitinden 23 bit periyodu sonra başlar. Bazı özel durumlarda bu periyot 31'e çıkar.

1.3.6. KAA Protokolünde Veri Güvenliği

Bu bölüm içerisindeki amaç ; ver gerçekliğinin yeteri derece yüksek olup olmadığı veya hata yapma olasılığının yeterinde düşük olup olmadığını tespit etektir. Artan hata olasılığı veri gerçekliğinin azalışını statik olarak ölçümü ile gözlenebilir.



Şekil 1.6. Artan Hat Olasılığı ve Bit Hata Olasılığı Karşılaştırılması

Artan hata olasılığının hesaplanması için sistem içinde oluşan hataları sınıflandırılmalı ve tüm iletim yörüngesi modellenmelidir. Eğer KAA' in artan hata olasılığı:

Mesaj uzunluğu 80 90 arasında olarak kabul ederek, bit hata olasılığının bir fonksiyonu olarak tanımlanırsak ve örnek olarak 5 veya 10 node için hata oranını 1/1000 olarak belirlersek maksimum bit hata olasılığının yaklaşık olarak 0,02 olur.

1.3.7. Extended Formatında KAA Mesajı

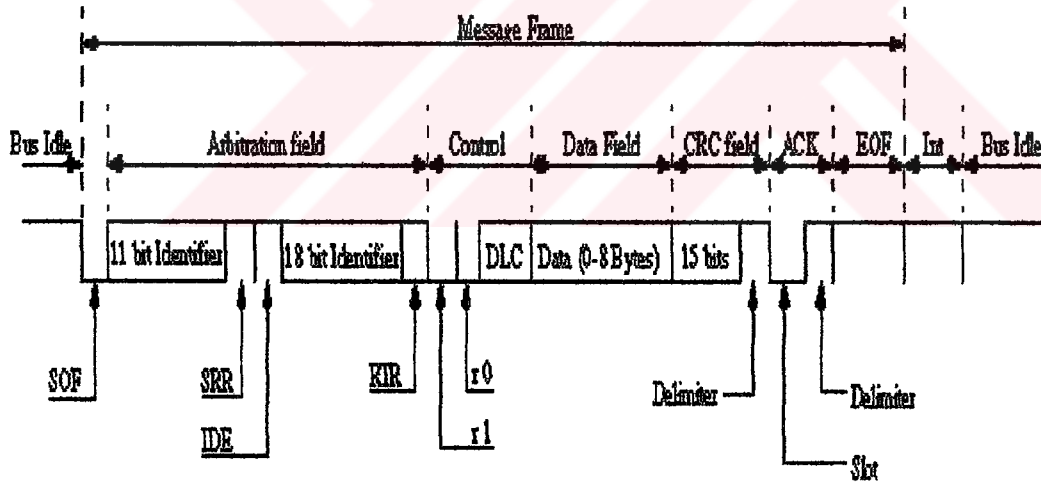
KAA protokolü daima iki tip mesaj formatı kullanır : Standard KAA (Version 2.0A) ve Extended KAA (Version 2.0B). Her iki mesaj tipide aynı veri yolu üzerinde kullanılabilir. Fakat Standart formatında ki mesajın her zaman için extended formatında ki mesaja karşı önceliği bulunmaktadır.

Extended formatını destekleyen KAA kontrolörleri standart formatta mesajda gönderip alabilmektedir. Fakat standart formatta mesaj alıp verebilen kontrolörler Extended formata mesaj alıp yollayamazlar. Bu tip

kontrolörlerde Extended formatındaki mesajlar hata olarak algılanamaz sadece ret edilirler.

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 uzunluğundadır. İkinci alan (Belirteç Ek), 18 bit uzunluğundadır 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.



Şekil 1.7. KAA 2.0B Mesaj Çerçevesi.

2. DEVICE NET

KAA protokolünün geliştirilmesinden sonra bu protokolü temel alarak değişik amaçlara hizmet eden üst protokoller geliştirilmiştir. Bunlara arasında:

- CANopen
- CAN Kingdom
- DeviceNet

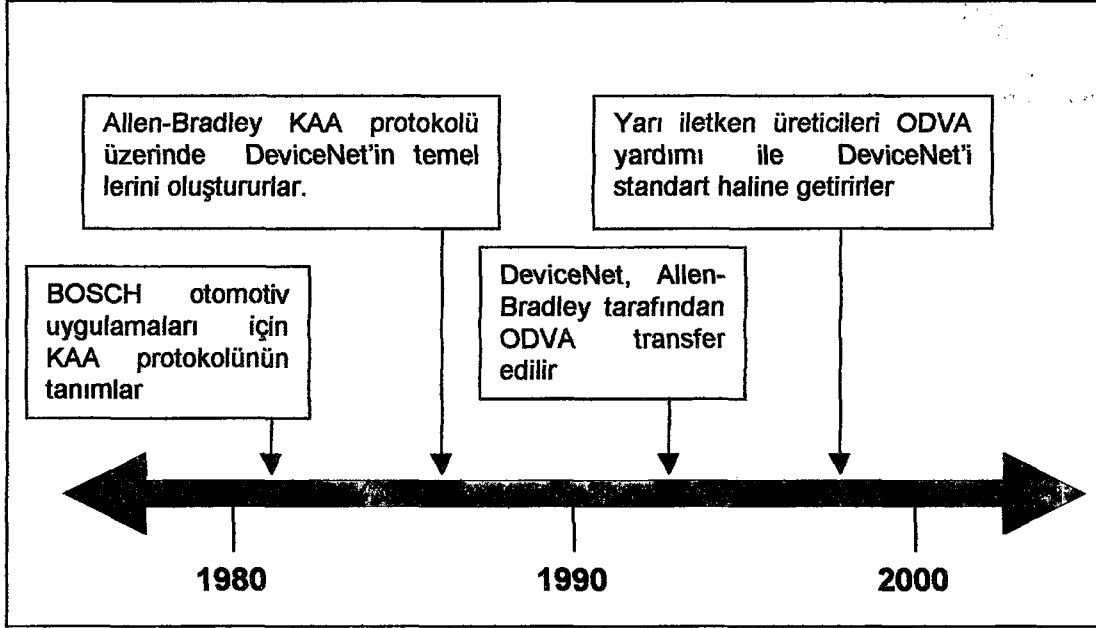
sayılabilir. Kullanılan sistem ve ihtiyaçlarını göz önüne alarak DeviceNet protokolünün daha uygun olduğuna karar verilmiştir. Bu bölüm içerisinde DeviceNET protokolünü ve Adam 5000/CAN sistemi üzerine nasıl uygulandığını inceleyeceğim.

2.1. DeviceNet'in Tarihçesi

1980'lerin başında önemli bir otomobil yedek parça üreticisi olan Bosch tarafından daha ucuz maliyete sahip, elektriksel gürültüye dayanıklı ve güvenilir bir haberleşme protokolü olan KAA protokolü geliştirilir. Geliştirilen bu protokolü kullanan Allen-Bradley nesne-tabanlı (object-oriented) dizaynı destekleyen ve daha gelişmiş bir haberleşme içeriğine sahip Araç-Ağı (DeviceNet) protokolünü oluştururlar .

Allen-Bradley geliştirdikleri bu protokolü diğer satıcılara ulaştırılabilmek için Open DeviceNet Vendors Association (ODVA) kurarlar ve bu dernek sayesinde Device net standartları oluşturulur.

ODVA 1990'ların başına kadar genişleyerek üye sayısını artırır bu yaygınlaşma sonucu National Semiconductor Manufacturing Organization (SEMI) DeviceNet bir standart olarak kabul eder.



Şekil 2.1. DeviceNet'in Tarihçesi

DeviceNet protokolünün büyük bir hızla yaygınlaşması aslında geleneksel kontrol sistemlerinin çok karışık bir haberleşme kablolamasına sahip olması yatmaktadır. DeviceNet protokolü ise kontrolöre kendine bağlanacak cihazlar için bir gövde yapısı sunar. Bu da kablolama için daha ucuz maliyet ve montaj kolaylığı sağlar.

2.2. DeviceNet'in Tanımı

DeviceNet, limit anahtarları, foto sensörler, motor sürücüler, bar cod okuyucuları ve operatör ara yüzeyleri gibi endüstriyel cihazları bir ağ içerisinde düşük donanım maliyeti ile birbirine bağlayabilen düşük maliyetli bir haberleşme linkidir.

Farklı üreticilerin farklı amaçlar için sunduğu cihazların direk olarak bir ağ içerisinde bağlanması gelişmiş bir haberleşme sağlar. DeviceNet açık bir ağ standardıdır. Bu yüzden dolayı DeviceNet için geliştirilen donanım veya yazılımın lisanslaşmasına gerek yoktur. DeviceNet protokolü KAA protokolü

üzerine inşa edildiği için bütün KAA çipleri ile kullanılabilir. DeviceNet'in genel özelliklerini kısaca sıralayacak olursak

Çizelge 2.1 DeviceNet'in Genel Özelliklerini

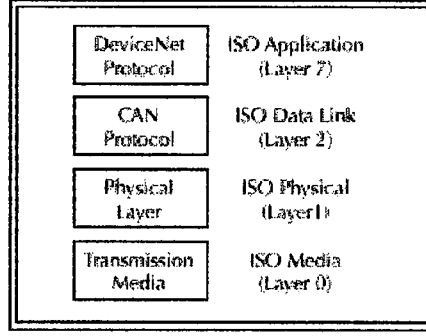
Ağ Büyüklüğü	64 bağlantı noktasına kadar.	
Ağ genişliği	Baut Rate	Mesafe
	125 kbps	500 m
	250 kbps	250 m
	500 kbps	100 m
Veri Paketleri	0-8 bit arası	
Veri Yolu Topolojisi	Lineer. Yani güç ve sinyaller aynı kablo üzerinden iletilir.	
Veri Yolu Adresleme	Multi-Cast : Peer-to-Peer (one-to-many); Multi-Master : Özel durumlar için Master/Slave special	
Ağ özelliği	Ağ içerisindeki cihazlar enerjileri kesilmeden veya ağın enerjisi kesilmeden söküp takılabilir.	

2.3. DeviceNet'in Özellikleri

DeviceNet'in özellikleri endüstriyel kontrol sistemlerinin bir birleri arasındaki haberleşmeyi tanımlar. Bu özellikler 3 kısımda incelenebilir :

- DeviceNet'in Fiziksel katmanı ve iletişimi (DeviceNet Physical Layer and Media).
- DeviceNet içerisinde kullanılan KAA.
- DeviceNet haberleşme protokolü ve uygulamaları.

DeviceNet KAA'nı kendi içerisinde barındırmasından dolayı DeviceNet içerisinde kullanılan KAA, syntax'ı veya veri hareketinin formunu belirlerken DeviceNet uygulama katmanı ise semantics veya veri hareketinin anlamını tanımlar.



Şekil 2.2. Uygulama Katmanları İçerisinde DeviceNet'in Konumu

DeviceNet içerisindeki haberleşme ;

- Peer-to-Peer veri değişim ki bu her hangi bir DeviceNet ürününün sistem içerisinde mesaj üretilip yok edebileceği anlamını gelmektedir.
- Master/Slave operasyonları ki bu tip operasyonlar Peer-to-Peer veri değişiminin doğal bir alt kümesidir.
- DeviceNet ürünü sistem içerisinde bir Client, Server veya her ikisi gibi davranabilir.

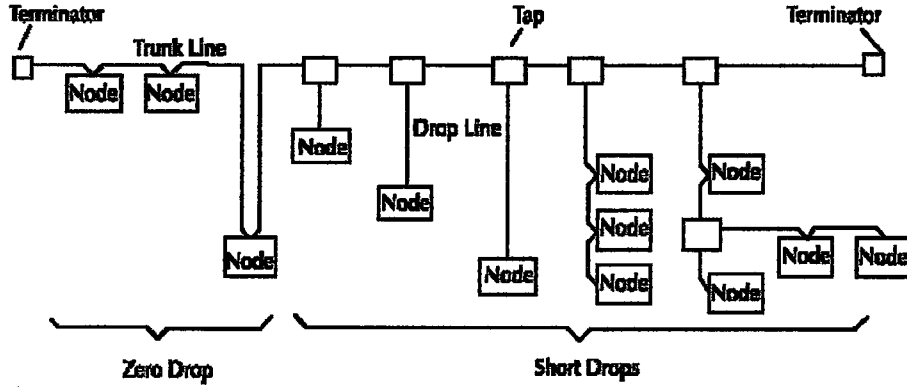
şeklinde olabilmektedir.

2.3.1. DeviceNet'in Fiziksel Katmanı ve İletişimi

DeviceNet prosedürü üzerinde uygulanabilecek topolojinin bir örneği Şekil 1.10'da gösterilmektedir. Topoloji üzerinde bulunan trunkline-dropline sayesinde hem güç hem de sinyaller taşınmaktadır. End-to-end ağ mesafesinin (Çizelge 2.3), data rate ve kablo kalınlığına göre değişme göstermektedir. Aletler ihtiyaç duyduğu güç ve bir biri ile kuracakları haberleşmeyi aynı kablo üzerinden yaparlar. Ağ içerisine eklenecek yeni nodalar ağın gücü kesilmeden de yapılabilir.

Ağ içerisine ihtiyaç duyulduğu takdirde, ekstra güç üniteleri DeviceNet ağı içerisinde (ki bu özellik diğer KAA protokolü destekli ağlar için geçerli değildir) istenilen noktada eklenebilmektedir. Trunkline için akım sınırı 8

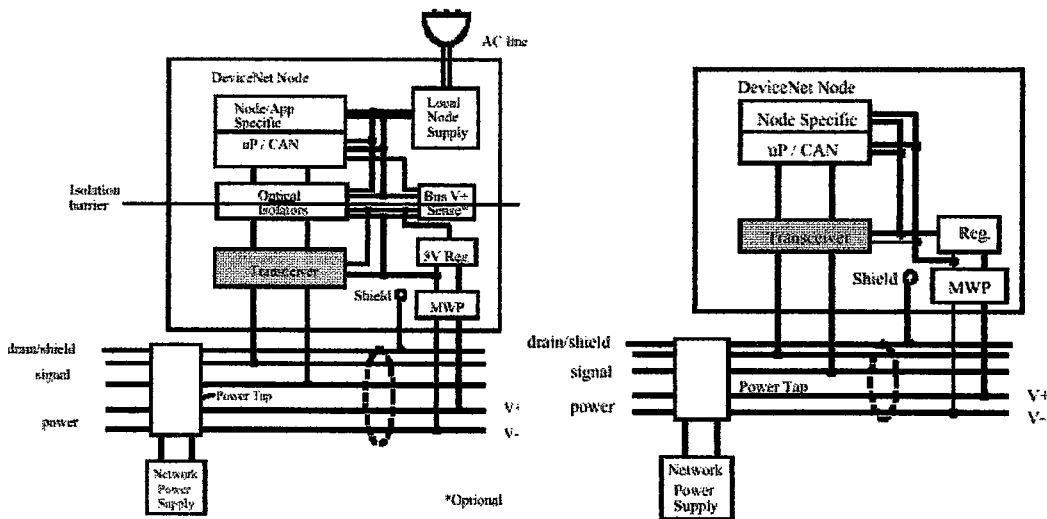
amperdir. Sistem içersine eklenmek isteyen selenoid valflar veya sürücüler gibi yüksek güçlü cihazlar optik izolasyonla rahatlıkla gerçekleştirilebilmektedir.



Şekil 2.3. DeviceNet Üzerinde Oluşturulabilecek Örnek Topoloji

Çizelge 2.2. End-to-end Ağ Mesafesinin ile Data/rate-Kablo İlişkisi

Data Rates	125	250	500
Thick Trunk Length	500	250	100
Thin Trunk Length	100	100	100
Maximum Drop	6 m	6 m	6 m
Cumulative Drop	156	78 m	39 m

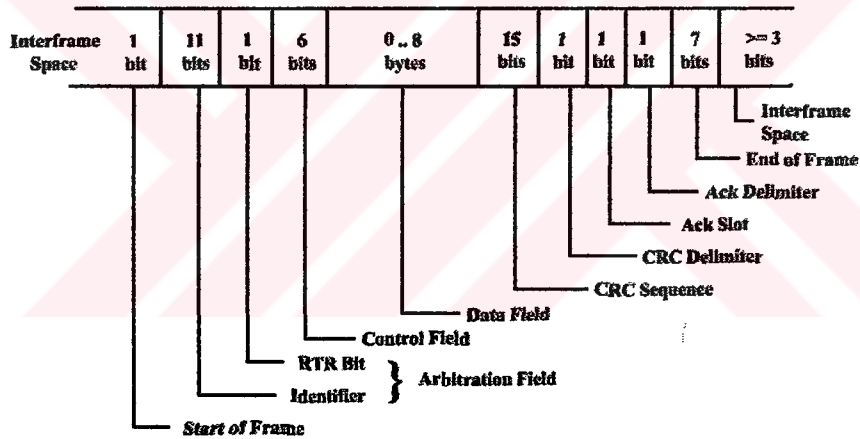


Şekil 2.4. İzole Edilmiş ve İzole Edilmemiş Fiziksel Katman

2.3.2. DeviceNet İçerisinde Kullanılan KAA

DeviceNet veri link seviyesi tamamen KAA tarafından tanımlanmaktadır ve hiçbir ekleme çıkartma yapmadan bir KAA chipi tarafından işenebilmektedir. KAA protokolü dominant (logic 0) ve recessive (logic 1) olmak üzere, iki veri yolu durumu belirlemektedir. Veri yoluna gönderilen her iletim isteği dominant durumunda olur. Veri yolu ancak hiçbir iletim isteği olmadığı zaman recessive durumuna gelebilir.

KAA protokolünün bölüm 2.1.2.4'de tanımlandığı gibi çeşitli mesaj çerçevelerine sahiptir. Veri DeviceNet üzerinden, standart veri çerçevesini kullanarak taşınır. Bunun dışında kalan diğer veri çerçeveleri, istisnalar dışında, kullanılmaz. Veri çerçevesinin formatı Şekil 2.5 de verilmektedir.



Şekil 2.5. Veri Çerçevesinin Formatı.

Bir veya birden fazla node'dan ağa eş zamanlı veri girişi yapılmaya çalışıldığı zaman, akıllı-bit yıkıcı olmayan tahsis mekanizması sayesinde potansiyel çakışmalar veri yada veri yolu genişliğinde bir kayıp olamadan engeller. DeviceNet, her iki KAA protokolü standardı içerisinde bulunan, içinde tanımlayıcı ve RTR biti olan hakem alanını tanımaz. Bu yüzden dolayı DeviceNet RTR bitini herhangi bir işlem için kullanmaz. Bunun dışında kalan tüm KAA özellikleri DeviceNet içinde geçerlidir.



2.3.3. DeviceNet Haberleşme Protokolü ve Uygulamaları

Bu bölüm içerisinde sırası ile DeviceNet 'in kullandığı bağlantılar, mesaj protokolü ve haberleşme ile ilgili nesneler açıklanmaya çalışılmıştır.

Bileşik DeviceNet standartları veya uygulamaya yönelik özel nesneler kullanan uygulamalar Device Profile ismi verilmektedir. Device Profile ağ içerisinde kullanılacak cihazı tam olarak tanımlamaya yararmaktadır.

DeviceNet strobed, polled, cyclic, change-of-state and application-triggered data movement'i desteklemektedir. Kullanıcı, kullanılacak cihazların ve uygulamanın gereklerine göre master/slave, multi-master ve peer-to-peer uygulamalarında birini veya onların bir bileşkesini seçebilir. Veri hareket tipinin seçimi direk olarak sistemin cevap verme süresi ile ilişkilidir.

2.3.3.1 Bağlantılar

DeviceNet haberleşme protokolü bağlantı tasarısı üzerine kurulmuştur. Ağ içine eklenen bir cihaz diğerleri ile haberleşmesi için muhakkak ağa bağlanması gerekir. DeviceNet içerisine her hangi bir bağlantı yapabilmek için aletin Unconnected Message Manager (UCMM) veya Unconnected Port tarafında tanımlanması gerekir. Unconnected Message Manager (UCMM) veya Unconnected Port, bilgiyi bir node'dan bir başka noda taşıya bilme özelliğini veya fazladan I/O bağlantısını sağlayan, Explicit Messaging Connection tanımlayabilmek için kullanılır. Bu bağlantı gerçekleşikten sonra I/O verisi ağ içerisinde istenilen her hangi bir cihaza taşınabilir. Bu uygulamadan sonar ağ içerisindeki her cihaz 11 bitlik bir belirleyiciye sahip olmuş olur.

Oluşturulan bu 11 bitlik KAA tanımlayıcısı bu andan itibaren kullanılan cihazın ID konumunda olur. DeviceNet, 11 bitlik KAA tanımlayıcılarını 4 guruba böler. Bu guruplardan ilk üçünün mesaj kısmı, 6 biti MAC ID ve geriye

kalan kısmı ise mesaj ID'si olmak üzere, iki alana bölünmüştür. Üçüncü gruptaki mesajlar ise offline haberleşme için kullanılır.

IDENTIFIER BITS										HEX RANGE	IDENTITY USAGE
10	9	8	7	6	5	4	3	2	1		
0										000-3ff	Message Group 1
1	0									400-5ff	Message Group 2
1	1									600-7bf	Message Group 3
1	1	1	1	1						7c0-7ff	Message Group 4
1	1	1	1	1	1	1	1	x	x	7f0-7ff	Invalid CAN Identifiers
10	9	8	7	6	5	4	3	2	1	0	

Şekil 2.6. DeviceNet tarafından 4'e Ayrılan Tanımlayıcı Tipleri

Kullanılacak Olan cihaz Client, Server veya her ikisinde olabilir. Client veya Server olarak kullanılan cihazlar üretici veya tüketici olabilirler. Tipik Client cihazlar istek üretirler ve yanıt tüketirler. Tipik Server cihazlar ise istek tüketirler ve yanıt üretirler. DeviceNet bu işlemler için bir çok varyasyon üretir. Bazı bağlantılar ise hem bir Server hem de bir Client durumundadır ve sadece mesaj tüketirler veya sadece mesaj üretirler. . Bu tip bağlantılar sadece cyclic veya change-of-state tipi mesajlar kullanabilir. Cyclic veya change-of-state tipi mesajlar banda genişliğine büyük ölçüde azaltırlar. Dizaynlarından dolayı DeviceNet sistemi içerisindeki her hangi bir node kendi tanımlayıcılarını yönetmekle sorumludur.

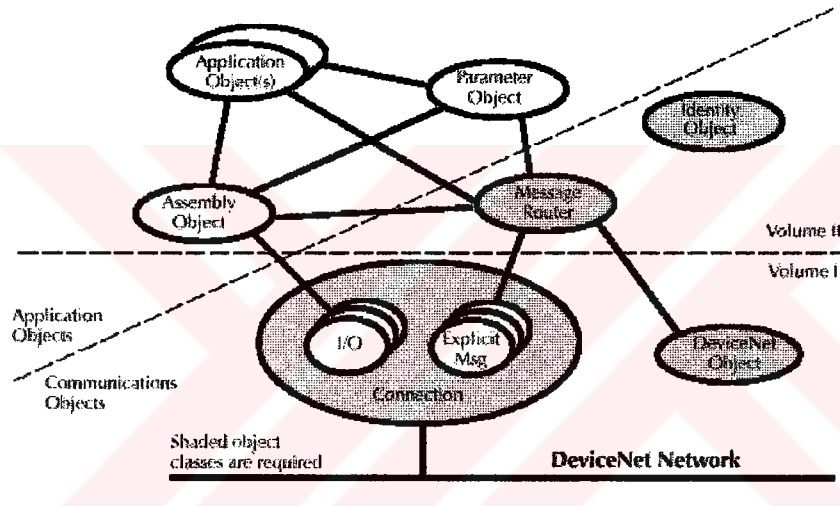
2.3.3.2 Object Model

Object Model, DeviceNet ürünleri için Attributes (data), Services (methods or procedures) and Behaviors organizasyonu ve yerine getirmesi için kalıp sağlar. Model, dört sayıya sahip her Attributes için adresleme planı sağlar. Bu dört sayıyı Node Address (MAC ID), Object Class Identifier, Instance Number ve Attribute Number oluşur. Bu dört seviyedeki adres DeviceNet ağı içerisinde veriyi bir yerden bir yere taşıyan Explicit Messaging

Connection için bir bağlaç olarak kullanılmaktadır. Bu dört adresleme elemanı için sınırlar Çizelge 2.3.'de verilmektedir.

Çizelge 2.3. Adresleme Elemanı İçin Sınırlar

Adre	En	En
Node	0	63
Class	1	65535
Insta	0	65535
Attrib	1	255



Şekil 2.7. DeviceNet İçin Object Model

Identity Object: DeviceNet ürünleri tipik olarak tek Identity Object örneğine sahiptir. Bu örnek Vendor ID, Device Type, Product code, revision, status, serial number, product name ve state sembolleri içerir. Bu özellik Get_Attribute_Single and a Reset ile kullanılır.

Message Router Object: DeviceNet ürünleri tipik olarak tek Message Router Object örneğine sahiptir. Message Router Object, ürünün Explicit Messages'inin diğer bir object'e geçmesi için bir araçtır.

DeviceNet Object: DeviceNet ürünleri tipik olarak tek DeviceNet Object örneğine sahiptir. Bu örnek Node Address, MAC ID, baudrate, Bus-Off

action, Bus-Off counter, the allocation choice ve master's MAC ID niteliklerine sahiptir. Bu özellik Get_Attribute_Single ile kullanılır.

Assembly Object(s): DeviceNet ürünleri tipik olarak bir veya birden fazla Assembly Object(s) örneğine sahiptir. Bu object'in temel amacı, değişik uygulamaların obje'lerinde değişik Attributes'ları (data), tek bir mesaj tarafından taşınan tek bir Attributes için gururlandırmaktadır.

Connection Objects: DeviceNet ürünleri en az iki DeviceNet Object örneğine sahiptir. Her bağlantı objesi, DeviceNet ağı üzerindeki iki node arasındaki sanal bağlantının son noktasını belirtir.

Parameter Object: Opsiyonel olan Parameter Object, cihaz içinde kullanılan ayarlanabilen bir parametredir. Parameter Object, her tür veri girişi sırasında konfigürasyon araçları için standart haline getirir. Parameter Object tanımı içerisinde değerler limitler ve aralıklar girmektedir.

2.3.3.3 Mesaj

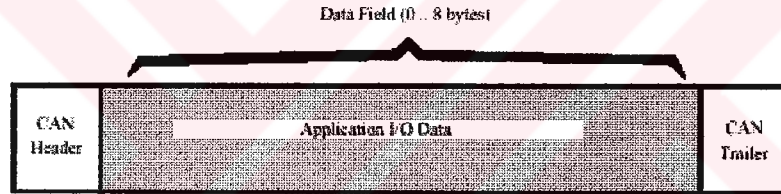
DeviceNet uygulama seviyeleri, belirleyicilerin nasıl atanacağını, KAA veri bölgesinin veriyi nasıl taşıyacağını ve ne anlama geldiğini anlayabilmek için nasıl kullanılacağını tanımlar. Bir haberleşme ağı içerisinde verinin nasıl akacağı kritik bir karardır. Eski haberleşme teknolojilerinde mesajların kararlılığı özel bir kaynak ve gönderici tarafından gerçekleştirilirdi. DeviceNet daha etkili bir yöntem olan ve veri için tanımlayıcı alanlara ihtiyaç duyan paketleri içeren Producer-Consumer Model'ini kullanır. Bu tanımlayıcılar bir çok öncelikli katman için, I/O verisi daha etkili bir transfer ve bir çok tüketicinin olmasını ifade eder.

Bütün cihazlar ağ içerisinde üretilen mesajları dinlerler. Cihaz mesaj üzerinde kendinin anlayabileceği bir veri olduğu takdirde bu mesajı tüketirler. Bu üretici-tüketici modeli ile ağ içerisinde bulunan mesajın kaynağı ve alıcısının hiçbir önemi kalmamış olur. Bir kontrolör tarafından üretilen mesaj

bu model sayesinde aynı anda bir çok aynı işi yapan cihaz tarafından daha az bant genişliği kullanılarak işlenebilmektedir.

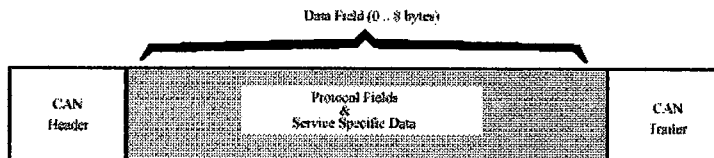
DeviceNet, I/O Messaging ve Explicit Messaging olmak üzere iki değişik tip mesaj tanımlar.

I/O messages tipi mesaj zaman-kritikli ve kontrol-yönlendirilmeli veri içermektedir. Bu tip mesaj kalıbı, üretici uygulama ile bir veya birden fazla tüketici uygulaması arasında özle amaçlı haberleşme yolu sağlar. Sinyal boyunca veya multi-cast bağlantılar boyunca alış veriş yaparlar ve genellikle yüksek öncelik tanımlayıcıları vardır. I/O Messaging tipinde 8 bitlik veri alanı içerisinde her hangi bir protokol uygulanmaz. Bu olayın tek istisnası parçaları ayrılmış I/O Messaging içindir. Bu tipte 1 bit parçalama için kullanılmaktadır.



Şekil 2.8. I/O Messaging Formatı

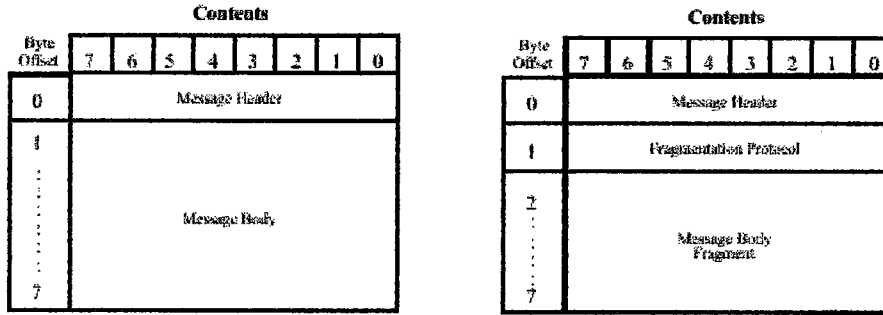
Explicit Messaging tipi mesaj ise iki cihaz arasında çok-amaçlı ve point-to-point tipi haberleşmeyi sağlamak için kullanılmaktadır. Bu tip mesajlar tipik istek / yanıt-yönlendirilmeli ağ haberleşmesini sağlar. Explicit Messaging mesaj tip genellikle düşük öncelikli tanımlayıcılar için kullanılmaktadır.



Şekil 2.9. Explicit Messaging Formatı

Fragmentation services ise 8 bitten daha uzun mesajlar için kullanılmaktadır. I/O Messaging tipinde bu işle için sadece en öndeki tek bitini

ayırır. Bölme işlemi için herhangi bir sınır yoktur. Explicit Messaging mesaj tipide bu bölme işlemini kullanabilmektedir. Bu bölme esnekliği DeviceNet ağı içerisinde daha karmaşık ve güçlü cihazların kullanılabilmesine olanak sağlar. Yine object-oriented dizayn ve adresleme planı sayesinde daha alt protokoller ve ağ modellerine ihtiyaç duymadan rahatlıkla genişleyebilmektedir.

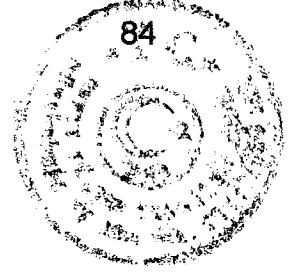


Şekil 2.10. Parçalanmamış ve Parçalanmış Explicit Messaging Formatı

2.3.3.4 Tekrar Tanımlanan Master/Slave Bağlantı Şekli

DeviceNet cihazlar arasında dinamik bir bağlantı şekline imkan sağlayan bir güçlü bir uygulama katman protokolü sağlamasına rağmen bu güçlü uygulamayı kullanamayan yada kullanamayan cihazlarda bulunmaktadır. Bu yüzden dolayı DeviceNet bu cihazların ihtiyaç duyduğu Master/Slave bağlantı şekli için tekrar tanımlanan Master/Slave bağlantı şeklini geliştirmiştir.

Birçok cihaz kendi çıktılarını düzenleyebilmek için diğer cihazların verisini değerlendirmek zorundadır. Bu cihazlar genelde kendi çıkışlarını ayarlaya bilmek için girdi verisine veya işlemek için çıktı verisine ihtiyaç duyarlar. Tekrar tanımlanan Master/Slave bağlantı şeklini ise bu cihazların zaman içerisindeki duyacakları veri ihtiyacını karşılamak için kullanılmaktadır.



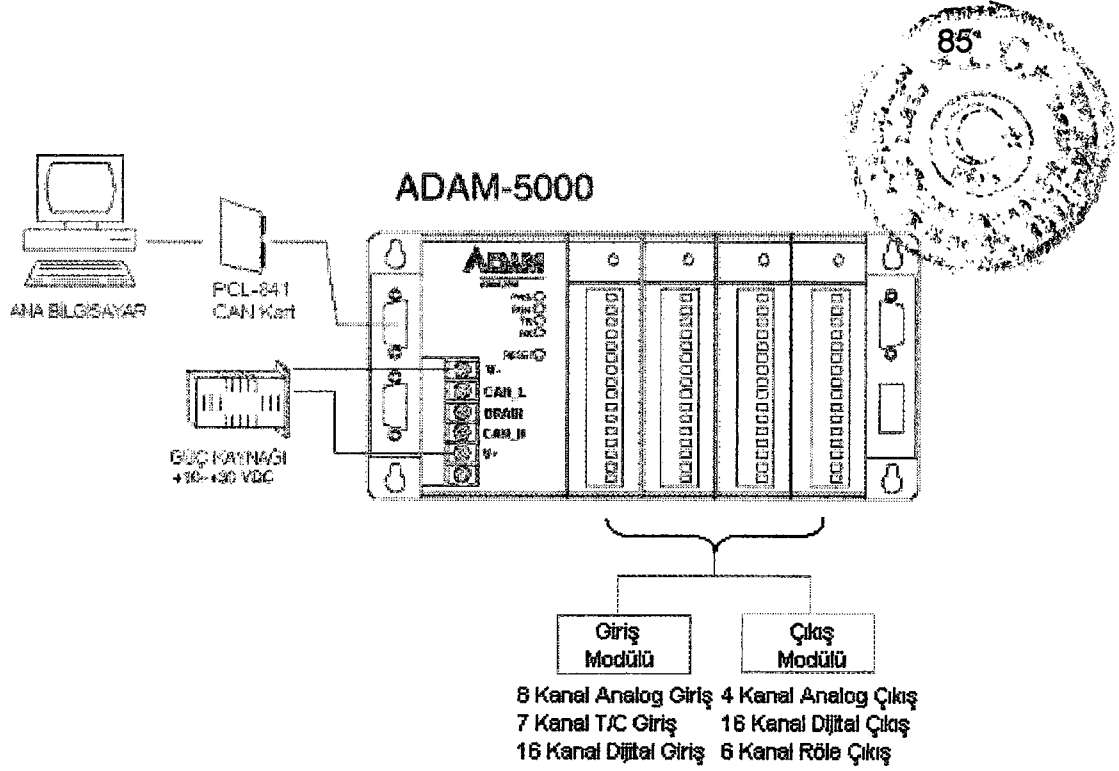
EK - 2

1. ADAM 5000/CAN MODÜLÜ

Adam 5000/CAN modülü özellikle geniş bir kontrol ağı içerisinde bulunan aletlerin, gerçek zamanlı uygulamaları sırasında, ihtiyaç duyacakları veri toplamak ve kontrol işlevini sağlaması için geliştirilmiştir. Adam 5000/CAN modülü uzaktaki bir sunucu bilgisayar yardımı ile DeviceNet ve CANopen protokolleri üzerinden kontrol edilebilmektedir. Veri ve kontrol sinyalleri KAA protokolü üzerinden iletilmektedir. Adam 5000/CAN sistemleri küçük, ihtiyaca göre tekrar yapılandırılabilir esnek bir yapıya sahiptirler. Adam 5000/CAN modülünün temel kısımları aşağıdaki gibidir :

- Adam 5000/CAN ana ünitesi. Ana ünite bir CPU kartı, bir güç regülatörü, 4-slot, bir RS232 haberleşme portu ve bir çift DB9 CAN portunu kapsar.
- Giriş Çıkış Arabirimi. Adam 5000/CAN ünitesi üzerinde bulunan CPU 64 G/Ç birimini 4 slot üzerinden desteklemektedir.
- G/Ç birimleri. Adam 5000/CAN modülünün en gelişmiş kısmı G/Ç birimleridir. Bu birimler isteğe göre yer, veri alış büyüklüğü ve giriş-çıkış veri çeşidi değiştirilebilmektedir. Analog girişler birimi 16 bitlik bir çözünürlüğe sahiptirler. Dijital giriş-çıkış birimi ise programlanabilir özelliğe sahiptir.
- Yazılım desteği. Adam 5000/CAN ünitesi hem DeviceNet hem de CANopen protokollerini destekleyen DLL (Dynamic Link Library) dosyalarına sahiptir.

Şekil 1.1 Adam 5000/CAN modülünün bağlantı şemasını içermektedir.



Şekil 1.1 Adam 5000/CAN Modülü Bağlantı Şemasını.

1.1. Adam 5000/CAN Modülünün Fiziksel Özellikleri

Adam 5000/CAN Modülünün çalışması için tasarlandığı nominal çevre şartları Çizelge 1.1. verilmiştir.

Çizelge 1.1 Adam 5000/CAN Modülünün Çalışma Koşulları.

Özellik	Sınırlar
Saklama Koşulları	-25 °C ile +70 °C arası
Çalışır halindeki ortam sıcaklığı	-25 °C ile +70 °C arası
Çalışma halinde ortamın nemi	%5 ile % 95 arası
Atmosfer	Yanıcı gazlar dışında her türlü atmosfer ortamında

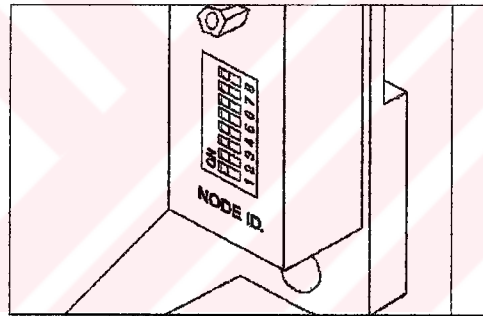
Fakat dikkat edilmesi gereken en önemli ortam koşulu ise elektriksel gürültüdür. Ölçümlerde Nemlilik oranı %30'un altına düşmesi halinde elektriksel gürültünün daha da şiddetli etki gösterdiği görülmüştür (Advantech, Adam 5000 manual)



Adam 5000/CAN Modülü sistemi standart endüstriyel regülesiz 24 V_{DC} güç kaynağı için tasarlanmıştır. +10 - +30 V_{DC} aralığındaki herhangi bir güç ünitesini kabul eder. Güç kaynağı en fazla 5 V ripple ile sınırlıdır.

Adam 5000/CAN Modülü, uygulama katmanında iki protokolü destekler, DeviceNet ve CANOpen. Default haberleşme protokolü DeviceNet'tir.

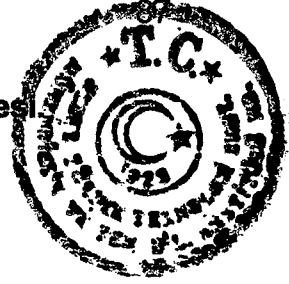
Baudrate ve ağ adresini set etmek için Adam 5000/CAN Modülü üzerindeki 8 bitlik Şekil 1.2.'de gösterilen DIP Switch kullanılır. 1 ila 6. bitler ağ adresini (Nokta ID) set etmek için kullanılır. 0'dan 63'e kadar (00h – 3Fh) adres kullanılabilir. ON = 1, OFF = 0 ifade eder. Bir ağdaki master için 00h adresi ayrılmıştır. 7. ve 8. bitler baudrate set etmek için kullanılır.



Şekil 1.2. Adam 5000/CAN Modülü- Dip Swich.

Çizelge 1.2 Adam 5000/CAN Modülü için Ağ Adresi Set Tablosu.

Nokta ID	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6
01h	ON	OFF	OFF	OFF	OFF	OFF
02h	OFF	ON	OFF	OFF	OFF	OFF
03h	ON	ON	OFF	OFF	OFF	OFF
.						
.						
3Fh	ON	ON	ON	ON	ON	ON



Çizelge 1.3 Adam 5000/CAN Modülü için Baudrate Set Çizelgesi

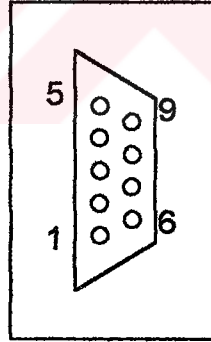
Bit 7	Bit 8	DeviceNet Baudrate	CANOpen Baudrate
OFF	OFF	125 kbps	20 kbps
ON	OFF	250 kbps	125 kbps
OFF	ON	500 kbps	500 kbps
ON	ON	-	1 Mbps

Adam 5000/CAN Modülü 10-30 V_{DC} gerilimde çalışmaktadır. Adam 5000/CAN Modülü güç besleme kabloları en az 2 mm² olmalıdır . Adam 5000/CAN Modülü ile sunucu bilgisayar arasında iletişimi sağlayacak olan kablolama ise elektriksel gürültüyü minimuma çekmesi için muhakkak twisted pair tipi ekranlanmış kablo ile yapılmalıdır (Advantech Adam 5000 manual).

Adam 5000/CAN Modülü ile sunucu bilgisayar arasında iletişimi sağlayacak olan CAN portu (Çizelge 1.4) ve RS-232 portu (Çizelge 1.5.) için gerekli olan pin bağlantıları aşağıda verilmektedir.

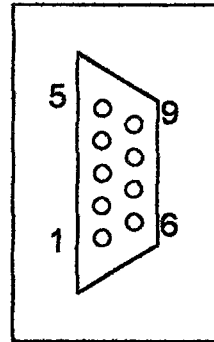
Çizelge 1.4 Adam 5000/CAN Modülü CAN Portu Pin Bağlantıları

Pin No.	Tanım
Pin 1	Kullanılmıyor
Pin 2	CAN_L
Pin 3	GND
Pin 4	Kullanılmıyor
Pin 5	Kullanılmıyor
Pin 6	Kullanılmıyor
Pin 7	CAN_H
Pin 8	Kullanılmıyor
Pin 9	Kullanılmıyor



Çizelge 1.5 Adam 5000/CAN Modülü RS-232 Portu Pin Bağlantıları

Pin No.	Tanım
Pin 1	Kullanılmıyor
Pin 2	Gönderilen Veri (TxD)
Pin 3	Alınan Veri (RxD)
Pin 4	Kullanılmıyor
Pin 5	GND
Pin 6	Kullanılmıyor
Pin 7	Kullanılmıyor
Pin 8	Kullanılmıyor
Pin 9	Kullanılmıyor



1.2. Adam 5000/CAN Modülünün Çalışma Özellikleri

Adam 5000/CAN modülü özellikle geniş bir kontrol ağı içerisinde bulunan aletlerin, gerçek zamanlı uygulamaları sırasında, ihtiyaç duyacakları veri toplamak ve kontrol işlevini, çoklu Giriş/Çıkış birimleri sayesinde gerçekleştirmektedir. bir CPU kartı, bir güç regülatörü, 4-slot, bir RS232 haberleşme portu ve bir çift DB9 CAN portunu kapsar.

CPU Adam 5000/CAN modülünün kalbi sayılabilir. CPU birimi temel olarak :

- G/Ç birimleri için veri akış ve kontrol sistemini .
- Termokupulların lineerleştirilmesi.
- Haberleşme yazılımı ve emir komutlarının işlenmesini.
- Kalibrasyon yazılımı ve emir komutlarının işlenmesini.
- Alırım görüntüleme.
- Sistem parametrelerinin EEPROM'da tutulması için gerekli yönetim.
- Veri aktarımı .
- Diagnosis

İşlevlerini yerine getirmektedir.

1.2.1. Diagnosis

Her Adam 5000/CAN modülü çalışması sırasında, yürütmesini istediğimiz görevlerin temel parametrelerini görsel olarak ifade edebileceği 4 lede (PWR, RUN, TX, RX) sahiptir.Yine bu ledler sayesinde Adam 5000/CAN modülü self-testlerini gerçekleştirirken meydana çıkacak hataları kullanıcıya bildirmektedir.

1.2.2. Katlı İzolasyon ve Watchdog Time

Elektriksel gürültü sisteme çeşitli yollardan girebilmektedir. Fakat Adam 5000/CAN modülü için elektriksel gürültü kaynağı olabilecek G/Ç birimleri, güç kaynağı ve haberleşme portunun topraklama bağlantısı ayrı ayrı izole edilmiştir. G/Ç birimleri 3000 V_{DC} , Haberleşme bağlantıları 2500 V_{DC} ve güç ünitesi de 3000 V_{DC} kadar yalıtılmış durumdadır. Bu yalıtım sayesinde kısa devre ve elektriksel gürültünün azaltılması konusunda tedbir alınmıştır. Bu yalıtımlara ek olarak Adam 5000/CAN modülünün CPU birimi bir watchdog timer tarafından izlenir ve hatalara karşı CPU otomatik olarak resetler.

1.2.3. Sistem Konfigürasyonu ve Kalibrasyonu

Adam 5000/CAN modülü ile Analog giriş birimlerini, çeşitli seviyelerde giriş için, akım giriş için ve termokuplör girişi için rahatlıkla programlanabilir. Bütün bu işlemler G/Ç modüllerin parametreleri, iletişim hızı, HI-LO alarm birimleri ve kalibrasyonları rahatlıkla uzaktaki sunucu bilgisayar aracılığı ile yapılabilmektedir. Yapılan bu ayarlamalar direkt olarak Adam 5000/CAN modülü üzerinde bulunan EEPROM a yazılarak daha sonra oluşacak elektrik kesintilerinden etkilenmemiş hale getirilir. Adam 5000/CAN modülü haberleşmesini açık haberleşme protokolleri olan CANopen ve DeviceNet protokolleri ile gerçekleştirmektedir (Detaylı bilgi Bölüm 2'de verilmektedir). Adam 5000/CAN modülü için default protokol DeviceNet protokolüdür.

1.2.4. Durum-Değiştî Fonksiyonu

Adam 5000/CAN modülü gerçek zamanlı uygulamaları gerçekleştirebilmek için durum-değiştî fonksiyonunu kullanmaktadır. Bu fonksiyon sayesinde Adam 5000/CAN modülü sitemde her hangi bir değişiklik olduğu zaman bu değişikli bulundu ağ içerisinde yayar (broadcast yapar). Örneğin Adam 5000/CAN modülü üzerinde bulunan dijital girişlerden biri çalışır konumdan (on), kapalı konuma (off) geçtiği zaman Adam 5000/CAN modülü

bu bilgiyi bağılı bulunduğu ağ yayınlarak (broadcast) ağ üzerindeki bütün node'lara ve dolayısı ile sunucu bilgisayara iletmış olur. Bu özelliği sayesinde Adam 5000/CAN modülü haberleşme etkinliğini artırırken gerçek zamanlı uygulamalar için gereken ihtiyacı sağlamış olur.

1.2.5. Adam 5000/CAN Modülünün Teknik Özellikleri

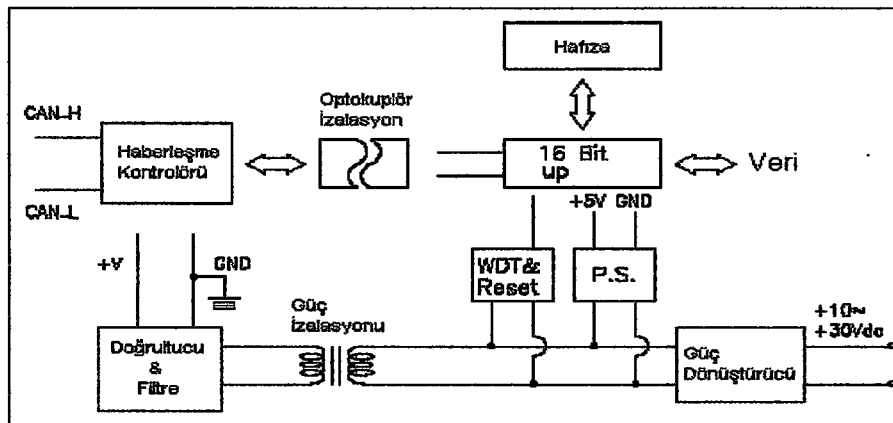
Bu bölüm içerisinde Adam 5000/CAN modülü için verilen teknik özellikler belirtilmektedir (Advantech, Adam 5000 manual) .

Çizelge 1.6. Adam 5000/CAN modülünün İşlemcisi için Teknik Veriler

CPU	80188, 16 bit mikroişlemci
SRAM	32 KB
ROM	128 KB
Giriş Tarama Süresi	DG:0,33 msec AG :0,1 sec
G/Ç Birimi Sayısı	4 slot
Watchdog Timer	Evet
Güç Tüketimi	1,2 W

Çizelge 1.7. Adam 5000/CAN modülünün İşlemcisi için Teknik Veriler

CAN portu	1 Adet, Giriş ve çıkış için 2 ayrı terminal
RS-232 portu	1 Adet
Veri Yolu (Bus)	CAN 2.0a (ISO-11898)
Veri Yolu (Bus) Girişi	CSMA/CD
Veri Transfer Oranı	DeviceNet : 125k, 250k, ve 500k bps
Max. Haberleşme	1000 m @ 20 kbps
Max bağlantı noktası	64 bağlantı noktasına göre
Mesaj Çerçevesi	11 bitlik ID paketi, 8 bite kadar veri paketi
Olaylara cevap verme	500 kbps'da Max. 1ms (Bağlantı sayısından bağımsız)
Haberleşme Protokolü	Master/Slave tipi haberleşme
Veri Güven	CRC üzerinden 15 bitlik denetimi, bit-stuffing ve Bus



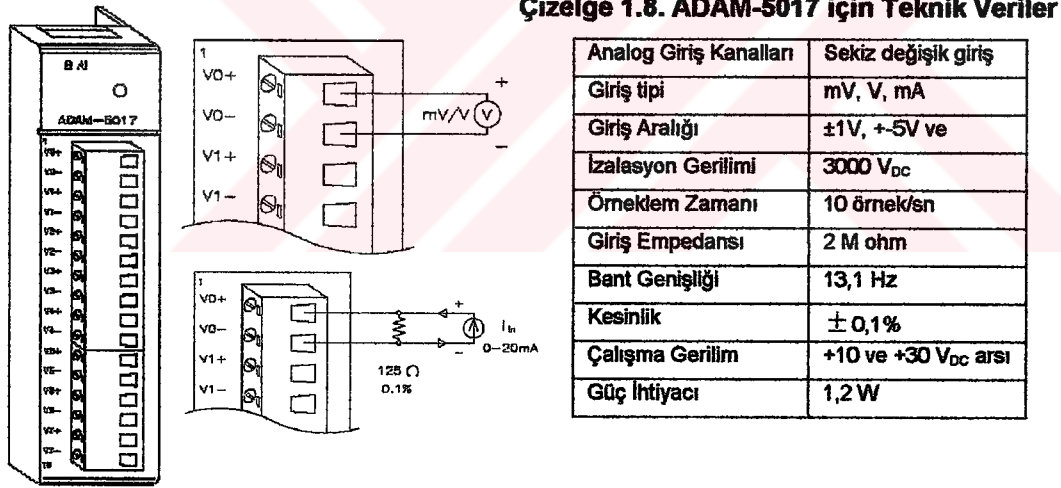
Şekil 1.3. Adam 5000/CAN Modülü Blok Şeması

1.3. Adam 5000/CAN Giriş Çıkış Modülleri

Bu bölüm içerisinde Adam 5000/CAN modülünün kullandığı Giriş Çıkış modüllerinin tiplerini ve teknik verilerini açıklayacağım.

1.3.1. ADAM-5017, 8 Kanallı Analog Giriş Modülü

Adam-5017 16-bitlik, 8 kanallı, analog giriş değişimini ölçebilen, programlanabilir bir gerilim ölçme aralığına sahip bir Analog giriş kartıdır. Mili volt girişlerini ($\pm 150\text{mV}$, $\pm 500\text{mV}$), voltaj girişlerini ($\pm 1\text{V}$, $\pm 5\text{V}$ ve $\pm 10\text{V}$) ve akım girişlerini ($\pm 20\text{mA}$, 125Ω) aralığında ölçebilmektedir. Modül ölçtüğü verileri, ana bilgisayara mühendislik birimleri (mV, V, mA veya $^{\circ}\text{C}$) olarak yollar. Analog girişler ve modülün kendi arasında $3000\text{ V}_{\text{DC}}$ 'lik bir optik izolasyon bulunmaktadır.

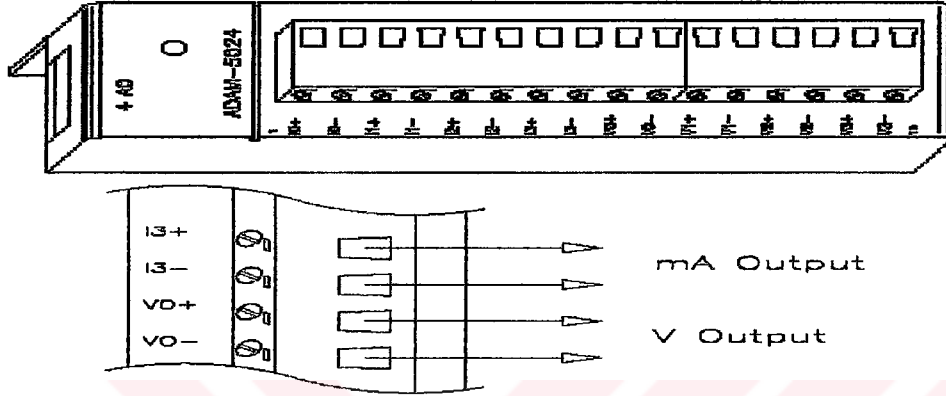


Şekil 1.4. ADAM-5017 Fiziksel ve Bağlantı Şekli

1.3.2. ADAM-5024, 4 Kanallı Analog Çıkış Modülü

Adam 5000/CAN modülü, sunucu bilgisayar tarafından, CAN ara yüzeyi sayesinde aldığı dijital veriyi ana ünite tarafından kontrol edilen bir D/A konvörtör kullanarak analog çıkış haline getirir ve ADAM-5024 modülü üzerinden dışarıya verir.

Dönüştürme oranını ve başlangıç akımını kullanıcı Adam 5000/CAN modülü yazılım ayarları yardımı ile kontrol edebilir. Yüksek değıştirme oranı çıkış gerilimlerinin daha çabuk değışmesini sağlar. Yine Adam 5000/CAN modülü yazılım ayarları yardımı ile çıkış akım veya gerilim olarak sağlanabilir. Analog çıkış 3000 V_{DC} 'lik bir optik izolasyon sağlayarak . Beslediğı ekipmanları kısa devreye karşı korur.



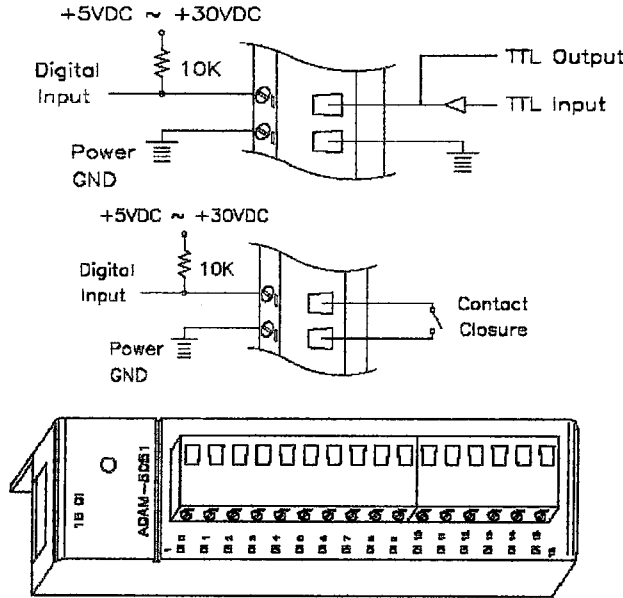
Şekil 1.5. ADAM-5024 Fiziksel ve Bağlantı Şekli

Çizelge 1.9. ADAM-5024 için Teknik Veriler

Analog Çıkış	4 Adet
Çıkış tipi	V, mA
Çıkış Aralığı	0-20 mA, 4-20 mA,
İzolasyon Gerilimi	3000 V _{DC}
Çıkış Empedansı	0,5 ohm
Doğruluk	13,1 Hz
Kesinlik	± 0,1 % Akım çıkışı
Sıfır Ayan	± 30 μ V ^D C , V
Güç İhtiyacı	2,5 W

1.3.3. ADAM-5051, 16 Kanallı Dijital Giriş Modülü

16 kanal sayısal giriş modülüdür. Bu modül, yüksek güçteki DC veya AC hatlar ve TTL lojik sinyalleri arasındaki arabirimleri izleyebilir ve kontrol edebilir.



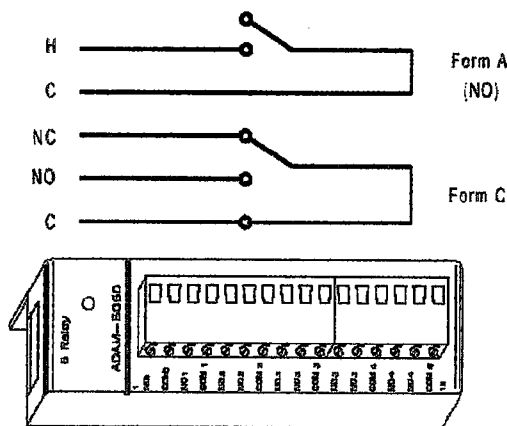
Şekil 1.6. ADAM-5051 Bağlantı ve Fiziksel Şekli

Çizelge 1.10. ADAM-5051 için Teknik Veriler

Kanalları	16 Adet
Dijital Giriş	Lojik 0 : Max. +1 V
Güç İhtiyacı	0,3 W

1.3.4. ADAM-5061, 6 Kanallı Röle Çıkış Modülü

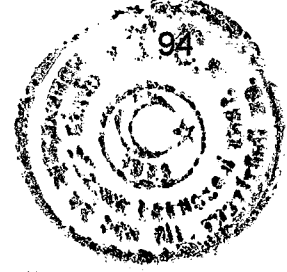
16 kanal sayısal giriş modülüdür. Bu modül, yüksek güçteki DC veya AC hatlar ve TTL lojik sinyalleri arasındaki arabirimleri izleyebilir ve kontrol edebilir.



Çizelge 1.11. ADAM-5061 için Teknik Veriler

Röle Çıkış	6 Adet , 2 Adet A
Kontak Oranları	AC : 125 V @ 0,6
Röle Çekme	3 mili saniye
Röle bırakma	1 mili saniye
Toplam	10 mili saniye
Güç İhtiyacı	0,7 W

Şekil 1.7. ADAM-5061 Bağlantı ve Fiziksel Şekli



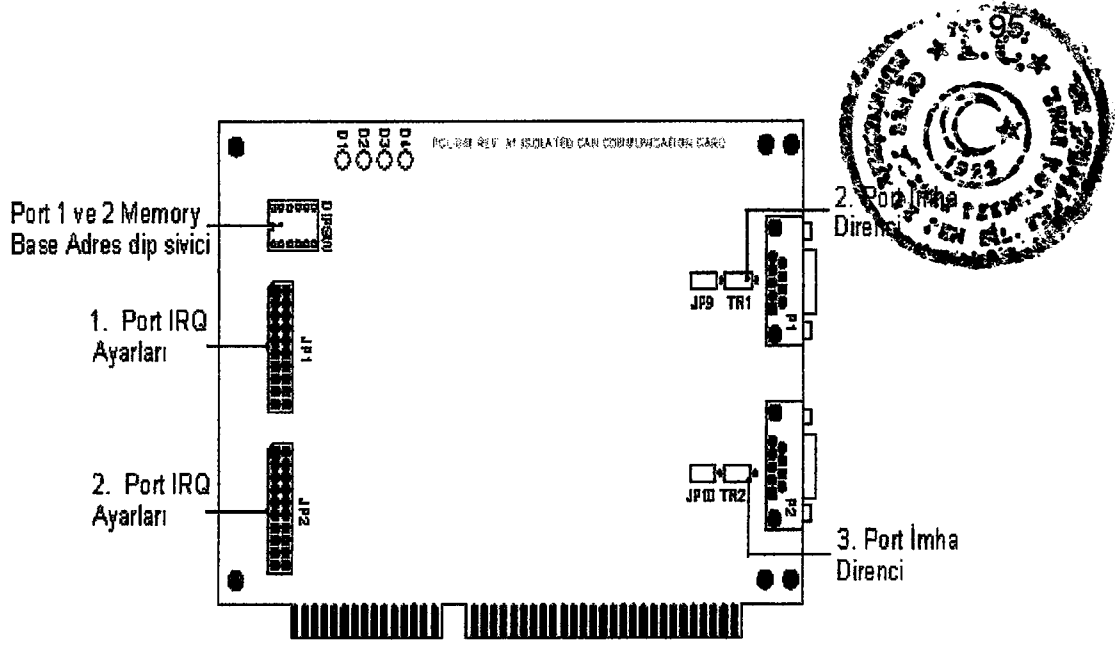
2. PC-841 DUAL PORT CAN KARTI

PCL-841 kartı herhangi bir PC' yi Kontrolör Alan Ağına bağlayabilmek için kullanılmaktadır. PCL-841 kartı içerisinde kurulu olan CAN kontrolör sayesinde, hata ayıklama ve veri hakemliğini iletişimi sırasında otomatik olarak gerçekleştirmektedir. Bu özellikleri ile veri kayıpları ve sistem gerçekliğini sağlar. On-board kontrolör üzerindeki hafızanın değişik yerlerine konumlandırılmıştır bu kontrolörleri aynı anda yada farklı olarak çalıştırmak mümkündür. PCL-841 kartı, 1 Mbps baud-rate hızına kadar iletişim yapabilmektedir.

PCL-841 kartı temele olarak bir memory adresi olarak iş görmektedir. Böylelikle CAN kontrolörüne direkt bir erişim rahatlıkla sağlanır. Bu yöntemle PCL-841 kartı kullanılacak olan PC için bir RAM olarak görüldüğü için ayrı bir yazılım kullanılmadan PC' ye tanıtılabilir. PCL-841 kartı kısa devre ve elektriksel gürültü için optik izolasyona sahiptir.

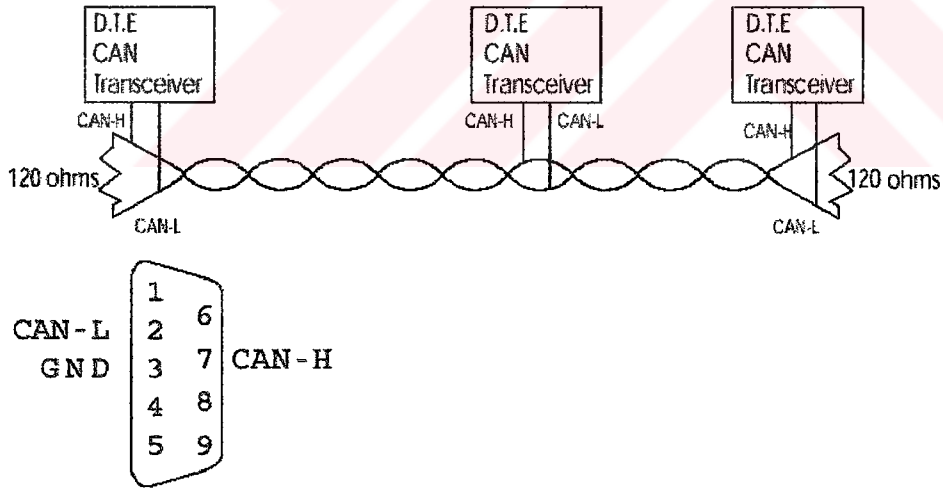
PCL-841 kartının genel özellikleri aşağıdaki gibidir:

- İki ayrı CAN ağı ile aynı anda iletişim kurabilir.
- 1 Mbps'a kadar iletim hızına sahiptir.
- C800H'dan EF00H'a kadar 40 değişik adres ayarlamasına sahiptir.
- 1000 V_{DC} 'ya kadar optik izolasyona sahiptir.
- Seçilebilir IRQ ayarı. IRQ 3,4,5,6,7,9,10,11,12,15.
- CAN kontrolörüne hızlı ulaşımı sağlayan direkt memory mapping özelliğine sahip.
- CAN kontrolör : 82C200 ; CAN transmitter : 82C250



Şekil 2.1 PCL-841 Kartı

CAN standartları half-duplex haberleşme standardını desteklemektedir.



Şekil 2.2. PCL-841 Kartı CAN Portu Pin Bağlantıları ve Katilolama Topografyası