

155080

KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

**CEP TELEFONU TABANLI MOBİL SCADA OTOMASYON
SİSTEMİNİN GELİŞTİRİLMESİ**

YÜKSEK LİSANS TEZİ

Mevlüt KARAÇOR

Anabilim Dalı: Elektrik Eğitimi

Danışman: Yrd. Doç. Engin ÖZDEMİR

MAYIS 2004

KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

**CEP TELEFONU TABANLI MOBİL SCADA OTOMASYON
SİSTEMİNİN GELİŞTİRİLMESİ**

YÜKSEK LİSANS TEZİ

Mevlüt KARAÇOR

Tezin Enstitüye Verildiği Tarih : 17 Mayıs 2004

Tezin Savunulduğu Tarih : 17 Haziran 2004

Tez Danışmanı

Üye

Üye

Yrd.Doç.Dr. Engin ÖZDEMİR Doç.Dr. Bekir ÇAKIR Doç.Dr. Hakan TEMELTAŞ

(.....)

(.....)

(.....)

MAYIS 2004

CEP TELEFONU TABANLI MOBİL SCADA OTOMASYON SİSTEMİNİN GELİŞTİRİLMESİ

Mevlüt KARAÇOR

Anahtar Kelimeler: SCADA, gerçek zamanlı sistemler, J2ME Mobil uygulamalar

Özet: Modern üretim su ve atık su, doğalgaz ve petrol boru hattı sistemlerinde, sürecin denetimi ve görüntülenmesi oldukça önemlidir. Günümüzde, bu uygulama alanlarında, denetim ve görüntüleme amacıyla SCADA sistemleri kullanılmaktadır. Bu çalışmada, endüstriyel otomasyon sistemlerdeki süreç denetiminin gelişimi sunulmakta, gerçek zamanlı sistemler tanıtılmakta ve aralarındaki farklar açıklanmaktadır. Bu çalışmada, klasik SCADA sistemleri ve internet tabanlı SCADA sistemleri sunulmakta ve sistem bileşenleri ve yeni ekipmanlar tanıtılmaktadır.

Modern endüstriyel otomasyon sistemlerinde, süreç hakkındaki teknik bilgiye erişmek oldukça önemlidir. Bundan dolayı, SCADA üreticileri tarafından internet tabanlı SCADA sistemleri geliştirilmiştir. Buna rağmen, internet tabanlı SCADA sistemlerinin internet bağlantılı bilgisayar gereksinimi ve mobil olmama gibi bazı dezavantajları bulunmaktadır. Bu tez çalışmasında, internet tabanlı SCADA sistemlerinin dezavantajlarını ortadan kaldırmak amacıyla mobil tabanlı SCADA sistemi geliştirilmiştir. J2ME uygulama programı ve model vinç düzeneği kullanılarak yeni SCADA sistemi test edilmiştir. Böylece, örnek süreç cep telefonu kullanılarak her yerden denetlenebilmektedir.

DEVELOPMENT OF MOBILE PHONE BASED MOBILE SCADA AUTOMATION SYSTEM

Mevlüt KARAÇOR

Keywords: SCADA, Real-time systems, J2ME Mobile Applications

Abstract: In modern manufacturing, water and wastewater, natural gases and petroleum pipelines systems, process control and monitoring are very important. Recently, in this application areas are used SCADA system to control and monitor. The work presented here deals with evaluation of process control in industrial automation systems. In addition, real-time systems are introduced and is explained their differences. In this work, it is presented that traditional and internet based SCADA systems, are introduced components and new equipments.

In modern industrial automation systems, to access technical information is very important in about process. Therefore, internet based SCADA systems are developed by SCADA vendors. However, it has some disadvantages like that requirements of PC with connected to the internet and it is not mobile. These two disadvantages are very big paradoxes for internet based SCADA systems. Therefore, in this thesis mobile-based SCADA system is developed to solution internet based SCADA system's paradoxes. J2ME application program and model crane are produced to test a new SCADA system. Moreover, this sample system is tested with a mobile phone. Thus, process can be controlled by mobile phone in anywhere.

ÖNSÖZ VE TEŞEKKÜR

Günümüz dünyasında bir işletme için, karlılık, kaliteli üretim, verimlilik, işletme ve personel güvenliği oldukça önem taşımaktadır. İşletmenin bu gereksinimleri endüstriyel otomasyon sistemleri tarafından karşılanmaktadır. Günümüz endüstriyel otomasyon sistemlerinin temelleri 1900'lü yılların ikinci yarısından atılmıştır. Bilimin ve paralelinde teknolojinin gelişmesi sonucunda endüstriyel sistemlerde, elektromekanik ve pünomatik cihazlar, entegre devreler, programlanabilir mantık denetleyicileri, dağınık dijital sistemler ve mikro işlemciler endüstriyel süreçlerin denetiminde kullanılmaya başlanmıştır. Bu yeni denetleyicilerin süreç denetiminde kullanılmaya başlanması ile, çeşitli denetim sorunları ortaya çıkmıştır. Sistemin operatör tarafından kolayca anlaşılması, süreç hakkında çeşitli verilerin toplanması ve sürecin denetlenmesi için, 1970'lerde SCADA sistemleri kullanılmaya başlanmıştır. SCADA sistemleri, ülkemizde birçok fabrikada, enerji nakil hatlarında, petrol boru hatlarında, su ve doğalgaz şebekelerinde halen kullanılmaktadır.

1990'lı yılların ikinci yarısında ise, bir iletişim devrimi yaşanmış ve cep telefonları insanların hayatına girmiştir. Böylelikle insanlar istedikleri her yerden birbirlerine ulaşabilme imkanı kazanmıştır. Yapılan bu çalışma ile operatör, mühendis ve yöneticilerin denetlenen süreci herhangi bir yerden cep telefonu ile görsel olarak görüntülemesi ve denetlemesi sağlanmıştır.

Bu çalışma, "SCADA Sistemlerinin İncelenmesi ve PLC Üzerinden Endüstriyel Sistemlerin Denetimi" adlı EEEAG AY-40, 101E002 nolu TÜBİTAK, Araştırma Altyapısını Destekleme Projesi kapsamında gerçekleştirilmiştir.

Bana bu konuda çalışma imkanı veren danışman hocam sayın Yrd. Doç. Engin ÖZDEMİR'e ve J2ME uygulama programının yazımında yardımlarını esirgemeyen arkadaşım M. Adem ŞENGÜL'e teşekkürlerimi sunarım.

İÇİNDEKİLER

ÖZET	ii
ABSTRACT	iii
ÖNSÖZ ve TEŞEKKÜR.....	iv
İÇİNDEKİLER	v
SİMGELER DİZİNİ ve KISALTMALAR.....	vii
ŞEKİLLER DİZİNİ.....	x
TABLolar DİZİNİ	xiii
BÖLÜM 1. GİRİŞ.....	1
1.1. Süreç Yönetiminin Tarihi Gelişimi.....	1
BÖLÜM 2. GERÇEK ZAMANLI SİSTEMLER.....	6
2.1. Telemetri Sistemleri.....	8
2.2. Dağıntık Kontrol Sistemleri (DCS).....	9
2.2.1. DCS sistemlerinin özellikleri.....	10
2.3. PLC Tabanlı Sistemler	11
2.4. SCADA Sistemleri.....	13
BÖLÜM 3. SCADA SİSTEMLERİ.....	14
3.1. SCADA Sistemlerinin Tarihsel Gelişimi.....	14
3.2. SCADA Sistemlerinin Temel Yapısı	16
3.2.1 Saha ekipmanları.....	17
3.2.2 Veri ağları	17
3.2.3 Mantıksal denetim yazılımı	18
3.2.4 İnsan-makine ara yüzü (MMI).....	18
3.3. SCADA Yazılımı Seçiminde Dikkat Edilecek Noktalar	19
3.4. SCADA Sistemlerinin Uygulama Alanları	21

3.5. SCADA Sisteminin Bileşenleri.....	22
3.5.1 Ana Denetim Merkezi.....	23
3.5.2 Uzak Uç Birimi.....	23
3.5.2.1 RTU'nun yapısı ve kullanıldığı endüstriler	24
3.5.2.2 RTU'nun iletişim ve ağ yapısı	28
3.6. SCADA Sistemlerinin Gruplandırılması	31
3.6.1 Klasik SCADA sistemleri.....	31
3.6.2 İnternet Tabanlı SCADA sistemleri.....	33
BÖLÜM 4. İLETİŞİM SİSTEMLERİ	36
4.1. İletişim Ortamları.....	36
4.1.1 İletişim ortamlarının avantaj ve dezavantajları.....	38
4.2. Veri Yolu Sistemleri	41
4.2.1 Veri yolu seçim ölçütleri.....	43
4.2.2 Veri yollarının fiziksel boyutları.....	44
4.2.3 Veri yollarının fiziksel yapıları.....	47
4.3. Veri Yollarının İncelenmesi.....	51
4.3.1 ASI veri yolu.....	53
4.3.2 Interbus veri yolu	54
4.3.3 Profibus veri yolu.....	55
4.3.4 Ethernet ile gerçekleştirilen veri yolu	56
BÖLÜM 5. WinCC SCADA Yazılımı	59
5.1. Bilgisayar (Computer)	60
5.2. Etiket Yönetimi (Tag Management)	59
5.3. Editör (Editor).....	61
5.3.1 Grafik Tasarımcısı (Graphics Designer).....	61
5.3.2 Etiket Kütükleme (Tag Logging) alt programı	63
5.3.3 Kullanıcı Yönetimi (User Administrator) alt programı	65
5.3.4 Alarm Kütüğü (Alarm Logging) alt programı	66
5.4. Yardımcı Programlar	66
5.5. Siemens WinCC SCADA Yazılımının Değerlendirilmesi	68

BÖLÜM 6. ÖRNEK VİNÇ UYGULAMASININ SCADA ÜZERİNDEN DENETİMİ	70
6.1. PLC Programının Hazırlanması	71
6.2. WinCC SCADA Yazılımında Denetim Programının Geliştirilmesi ...	73
BÖLÜM 7. GELİŞTİRİLEN MOBİL TABANLI SCADA SİSTEMİ.....	88
7.1. J2ME Uygulama Programı	89
7.2. Web Sunucu Programı	92
7.3. SCADA Sunucu Programları	92
7.3.1 Uzak görüntüleme programı	93
7.3.2 Uzak denetim programı.....	94
7.4. Mobil Tabanlı SCADA Sisteminin Çalıştırılması	95
7.5. Mobil SCADA Sisteminin Ekonomik Açıdan İncelenmesi.....	98
SONUÇLAR ve ÖNERİLER.....	101
KAYNAKLAR	102
EKLER.....	106
ÖZGEÇMİŞ	130

SİMGELER DİZİNİ VE KISALTMALAR

A	: İlerleme Miktarı
TY	: Toplam Yükseklik
XY	: Anlık Konum
t	: Süre
İ	: Döngü Sayısı
DBTİM	: Dial-up Bağlantı Toplam İşletme Maliyet Gideri
GBTİM	: GPRS Bağlantı Toplam İşletme Maliyet Gideri
DBM	: Dial-up Bağlantı Maliyeti
GBM	: GPRS Bağlantı Hizmeti
PLC	: Programlanabilir Mantık Denetleyici
DDS	: Dağıntık Dijital Sistemler
MMI	: İnsan Makine Etkileşimi
RTU	: Remote Terminal Unit
DCS	: Dağıntık Kontrol Sistemleri
SCADA	: Gözetleyici Denetim ve Veri Toplama
ODBC	: Open Data Base Connectivity
OLE	: Object Link Embedded
OPC	: OLE for Process Control
SQL	: Sequential Query Language
COM	: Common Object Model
DCOM	: Distributed Common Object Model
WAP	: Wireless Access Protocol
IED	: Intelligent Electronic Devices.
RF	: Radyo Frekans
UUB	: Uzak Uç Birimi
ADM	: Ana Denetim Merkezi
DB	: Denetim Birimi
CCI	: Consultive Committee for International

PLL : Özel Kiralık Hat
ASI : Actuator Sensor Interface
DP : Distributed Processing
PA : Process Automation
FMS : Field Messaging Specification
TCP/IP : Transmission Control Protocol and Internet Protocol
WinCC : Windows Control Center
IIS : Internet Information Server
ERP : İşletme Kaynakları Planlama
API : Application Program Interface
J2ME : Java Micro Edition
Servlet : Server Applet



ŞEKİLLER DİZİNİ

Şekil 1.1. 1950'lerde süreç yönetiminin durumu.....	1
Şekil 1.2. 1960'lerde süreç yönetiminin durumu.....	2
Şekil 1.3. 1970'lerde süreç yönetiminin durumu.....	3
Şekil 1.4. 1980'lerde süreç yönetiminin durumu.....	4
Şekil 1.5. 1990'lerde süreç yönetiminin durumu.....	4
Şekil 1.6. 2000'lerde süreç yönetiminin durumu.....	5
Şekil 2.1. Gerçek zamanlı sistem bileşeni.....	6
Şekil 2.2. Gerçek zamanlı sistemlerde veri akışı.....	7
Şekil 2.3. Telemetri sisteminin temel şeması.....	8
Şekil 2.3 Örnek bir DCS sistemlerinin yapısı.....	9
Şekil 2.4. PLC'nin yapısı.....	11
Şekil 2.5. PLC tabanlı gerçek zamanlı denetim sistemi.....	13
Şekil 3.1. SCADA sisteminin temel bileşenleri.....	16
Şekil 3.2. İnsan-makine arayüzü.....	18
Şekil 3.3. SCADA sistemini temel yapısını oluşturan bileşenler.....	22
Şekil 3.4. RTU'nun veri organizasyonu.....	24
Şekil 3.5. RTU'nun prensip şeması.....	25
Şekil 3.6. Modüler olmayan RTU.....	27
Şekil 3.7. Modüler RTU.....	27
Şekil 3.8. RTU tabanlı SCADA uygulaması.....	29
Şekil 3.9. Klasik SCADA sisteminin yapısı.....	32
Şekil 3.10. Günümüz SCADA sisteminin yapısı.....	34
Şekil 4.1. SCADA sistemlerindeki veri iletişimi hiyerarşisi.....	42
Şekil 4.2. Veri yolunun kullanıldığı seviyeye göre sınıflandırılması.....	43
Şekil 4.3 Su dağıtım sisteminin otomasyonu.....	45
Şekil 4.4. Lokomotif denetim sisteminde veri yollarının kullanımı.....	46
Şekil 4.5. Veri yollarının veri gönderme kapasitelerine göre sınıflandırılması.....	47

Şekil 4.6. Birebir bağlantı.	48
Şekil 4.7. Saha veri yollarının kullanılması.	48
Şekil 4.8. Saha veri yollarının farklı fiziksel yapıları..	49
Şekil 4.9. Saha veri yollarının farklı fiziksel yapıları..	49
Şekil 4.10. Saha veri yollarında kullanılan kablo çeşitleri.....	50
Şekil 4.11. Veri yollarının sınıflandırılması.....	51
Şekil 4.12. ASI arayüzü ile algılayıcılar ve aktüatörler arasındaki bağlantı.....	53
Şekil 4.13. ASI veri yolunda ana-uydu iletişimi.	54
Şekil 4.14. Interbus veri yolu prensip şeması..	54
Şekil 4.15. Ana istasyon ile uydu istasyonlar arasındaki veri alışverişi	55
Şekil 4.16. Profibus veri yolunun yığın yapısı.....	56
Şekil 4.17. Ethernet tabanlı SCADA sistemi.....	57
Şekil 5.1. Siemens SCADA yazılımının yapısı.....	58
Şekil 5.2. Denetim merkezi (Control Center)..	58
Şekil 5.3. Bilgisayar özelliklerinin ayarlanması..	59
Şekil 5.4. Etiket yönetimi alt programı..	60
Şekil 5.5. Editör ve alt programları.	61
Şekil 5.6. Grafik Tasarımcısında canlandırma resminin hazırlanışı.	62
Şekil 5.7. Giriş/Çıkış alanları, çokgen, buton ve sabit metin nesnelerinin kullanımı.	62
Şekil 5.8. Etiket Kütükleme alt programı.	63
Şekil 5.9. Kütükleme işleminin gerçekleştirilmesi.	64
Şekil 5.10. Eğri ayarlama işleminin gerçekleştirilmesi.....	64
Şekil 5.11. Tablo işleminin gerçekleştirilmesi.....	65
Şekil 5.12. Kullanıcı Yönetimi alt programı kullanımı	65
Şekil 5.13 Alarm Kütüğü alt programı kullanımı	66
Şekil 6.1. SCADA üzerinden denetlenen model vinç düzeneği.....	69
Şekil 6.2. Test düzeneği veri akış şeması.....	70
Şekil 6.3. Simatic Manager PLC programının görünüşü.	72
Şekil 6.4. Etiketlerin (tag) oluşturulması.	73
Şekil 6.5. Verilerin arşivlenmesi.....	74
Şekil 6.6. Alarmların arşivlenmesi.....	75
Şekil 6.7. Grafik Designer'da resimlerin hazırlanışı.....	76

Şekil 6.8. Resim dosyaları arasındaki veri akışı.....	77
Şekil 6.9. C-scripts'lerinin yazılışı.....	78
Şekil 6.10. Süre tespit algoritması.....	79
Şekil 6.11. Konum denetim algoritması.....	80
Şekil 6.12. Alarm denetim algoritması.....	82
Şekil 6.13. "1.pdl" resim dosyası ekran görünümü.....	83
Şekil 6.14. Sahadan ölçülen değerlerin görüntülenmesi.....	84
Şekil 6.15. Test algoritmasının çalıştırılması.....	85
Şekil 6.16. Alarm ekranının görüntülenmesi.....	86
Şekil 6.17. Ayar değerlerinin girilişi.....	87
Şekil 7.1. Mobil tabanlı SCADA sisteminin yapısı.....	88
Şekil 7.2. Veri alışverişi ve kullanılan programlar.....	89
Şekil 7.3. Sun One Studio 4 CE programında cep telefonu programının yazılması..	90
Şekil 7.4. J2ME uygulama programına gerekli resimlerin ilave edilişi.....	90
Şekil 7.5. Hazırlanan J2ME uygulama programının WTK 20 simülasyon programında çalıştırılması.....	91
Şekil 7.6. Uzak Görüntüleme programı.....	93
Şekil 7.7. Uzak Denetleme programı.....	94
Şekil 7.8. Uzak sistem denetiminin gerçekleştirilmesi.....	95
Şekil 7.9. Uzak Denetim programı ve WinCC SCADA programının beraber çalışması.....	96
Şekil 7.10. Uzak Görüntüleme programı ve WinCC SCADA programının beraber çalışması.....	97
Şekil 7.11. J2ME uygulama programının çalışması.....	97
Şekil 7.12. Gerçekleştirilen mobil tabanlı SCADA sistemi test çalışması.....	98

TABLolar DİZİNİ

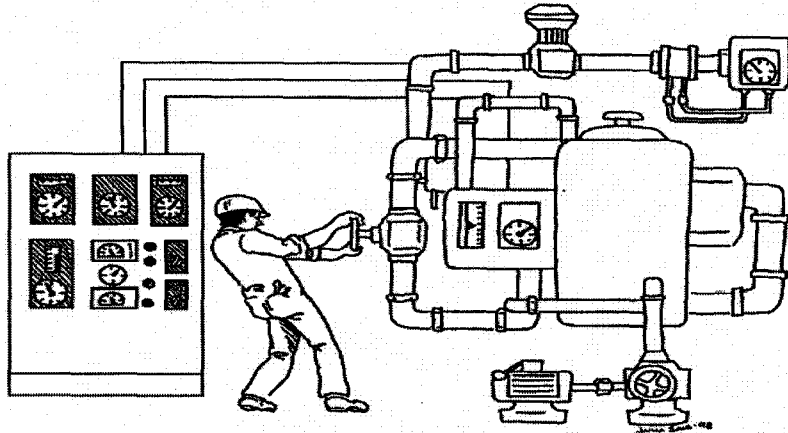
Tablo 3.1. SCADA sistemlerinin tarihsel gelişimi.	15
Tablo 3.2. RTU'nun kullanıldığı endüstriler	28
Tablo 4.1. Kablosuz veri alışverişini sağlayan iletişim ortamları.....	37
Tablo 4.2. Kablolı veri alışverişini sağlayan iletişim ortamları.....	38
Tablo 4.3. Veri yollarının uygulandığı sistemlerin fiziksel boyutları.....	44
Tablo 4.4. En yaygın kullanılan veri yolları	52
Tablo 6.1. PLC programında kullanılan giriş ve çıkış arayüz adresleri.....	71
Tablo 6.2. SCADA sisteminin PLC ile veri alışverişinde kullandığı adresler.....	71
Tablo 6.3. Kullanılan etiketler ve değişken tipleri.....	74
Tablo 7.1. SCADA'dan ASP sayfasına gönderilen parametrik değerler	93
Tablo 7.2. ASP sayfasından SCADA'ya gönderilen parametrik değerler	94
Tablo 7.3. Farklı uygulamalara ait günlük bağlantı maliyetlerinin karşılaştırılması.....	100

1. GİRİŞ

Bu bölümde endüstriyel otomasyonda kullanılan süreç yönetiminin ve sahada kullanılan cihazların (saha ekipmanları) gelişmesi adım adım ele alınmaktadır. Sürecin yönetimi, çevre birimlerinde, saha ekipmanlarında ve son kullanıcıdaki teknolojik gelişme ve eğilimlere, on yıllık periyotlarla tarihi bir bakış yapılmaktadır. Endüstriyel otomasyon sistemi tasarımındaki gelişmelere örnekler verilmektedir.

1.1. Süreç Yönetiminin Tarihi Gelişimi

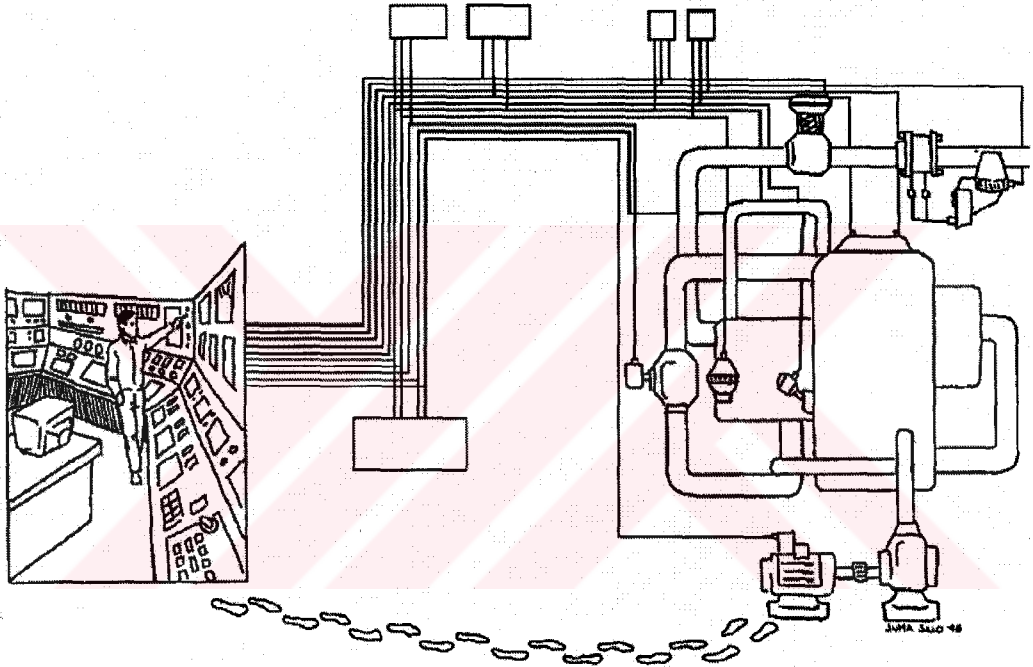
Koskinen [1], endüstriyel otomasyon sistemlerindeki gelişmeleri tanımlamış ve gelişmeleri nesillere ayrılmıştır. Süreç teknolojisindeki ve elektronik bileşenlerdeki teknolojik gelişmeler, otomasyon sistemlerindeki gelişmeleri teşvik etmiştir. 1950'lerdeki gelişim eğrisi, elektro mekanik sistemlere, standart sinyallerin yayılmasına, sürecin yakın mesafelerde görüntülenmesi ve çeşitli verilerin süreçten okunması üzerine odaklanmaktadır. Pnömatik ve analog enstrümanlar, basınç ve sıcaklık ölçümleri için çeşitli ölçüm cihazlarında kullanılmıştır. Geri beslemeli denetim, ilk olarak hidrolik sistemlerde kullanılmaya başlanmıştır. Şekil 1.1'de 1950'lerde ki süreç yönetiminin durumu görülmektedir.



Şekil 1.1. 1950'lerde süreç yönetiminin durumu.

1960'lardaki teknik gelişmelerde, dijital teknolojisindeki ve elektronik alanındaki

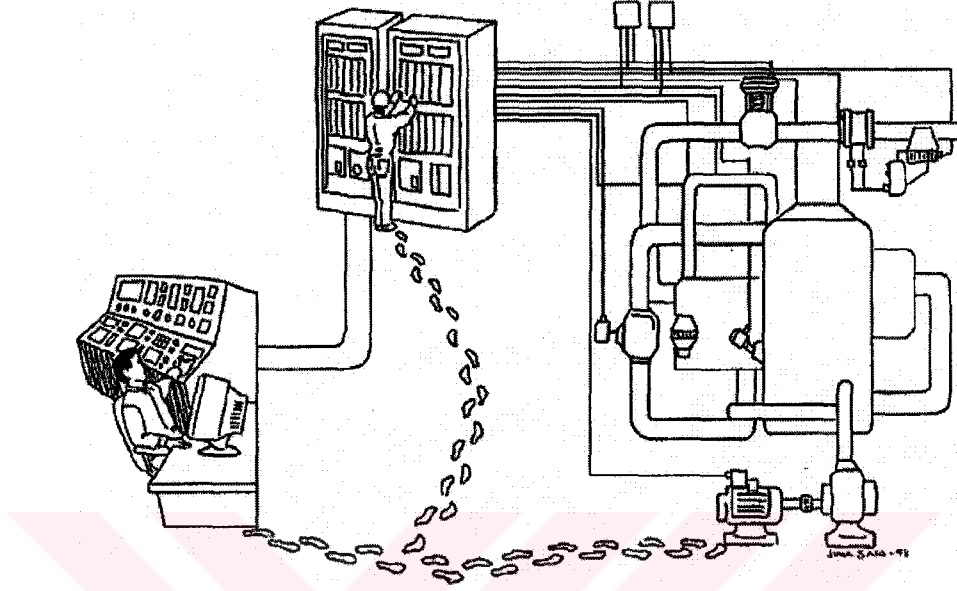
ilerlemeler özel bir yer almaktadır. Analog sinyallerin güvenilir hale gelmesi ile pünomatik ve elektrikli cihazlar, imalathanelerde büyük çapta kullanılır hale gelmiştir. Bu yıllarda süreç bilgisayarları endüstriyel uygulamalarla ilk defa tanışmış, enstrumantasyon yapıları standart bir hale getirilmiş ve merkezi denetim odası uygulamaları başlatılmıştır. Bu yıllarda, dijital teknolojisi süreç verilerinin saklanması ve işlenmesinde ilk defa kullanılmış ve süreç bilgisayarları dijital denetimde kullanılmaya başlanmıştır. Şekil 1.2’de 1960’larda ki süreç yönetiminin durumu görülmektedir.



Şekil 1.2. 1960’larda süreç yönetiminin durumu.

1970’lerin eğilimi ise, entegre devreler ve programlanabilir mantık denetleyicileridir (PLC: Programmable Logic Controller). Analog sistemlerdeki gelişmeler bu yıllarda en yüksek noktaya ulaşmıştır. Dağıtık Dijital Sistemler (DDS: Distributed Digital System) gelişmeye başlamış ve denetim odalarının yapıları gittikçe daha karmaşılaşmaya başlamıştır. Merkezi denetim sistemleri, ölçü aletleri, iletişim sistemleri ve bilgi yönetim sistemleri ile bileştirilmiştir. Dijital ve mikroişlemci teknolojisi elektronik cihazlarda ve denetim istasyonlarında kullanılmaya başlanmıştır. Bu on yıllık periyotta merkezi görüntüleme ve dağıtılmış denetim

sistemleri ile ilk defa kullanılmaya başlamıştır. Şekil 1.3'de 1970'lerdeki süreç yönetiminin durumu görülmektedir.

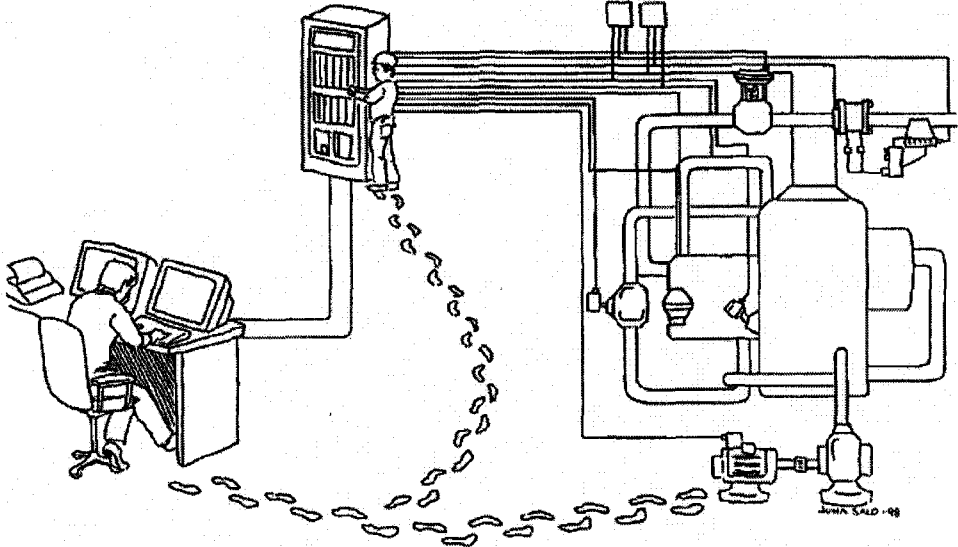


Şekil 1.3. 1970'lerde süreç yönetiminin durumu.

Dijital otomasyon sistemleri ilk olarak kağıt üretim sistemlerinde kullanılmaya başlanmıştır. Bu sistemlerde elektrikli cihazlar, sistemden bağımsız ve sisteme bir bütünün parçası olarak entegre değildi. İlk mikroişlemci tabanlı denetim sistemi bu yıllarda çeşitli uygulamalarda kullanılmaya başlanmıştır. Kağıt üretim sistemlerinde, kağıdın makaraya sarımının denetiminde ve kağıt kalitesinin kamera ile denetiminde ilk kez mikroişlemci kullanılmıştır. Ayrıca bu yıllarda akıllı ölçüm cihazlarında ve bu cihazların kalibrasyon panellerinde de mikroişlemci kullanılmaya başlanmıştır.

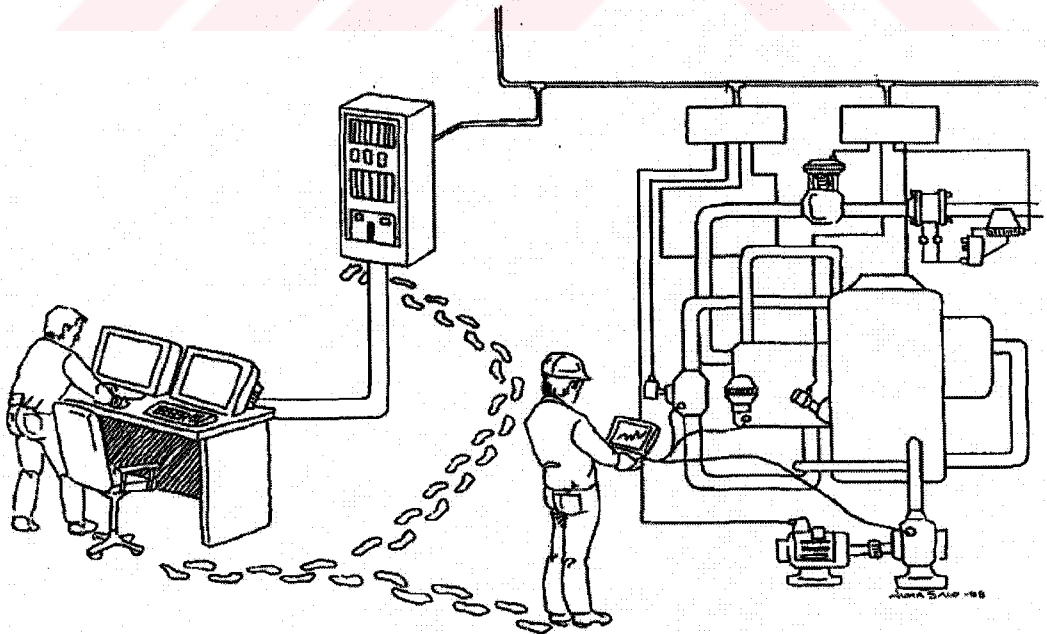
1980'li yıllarda mikroişlemci teknolojisinde büyük gelişmeler meydana gelmiştir. Video kamera ve entegre sistem çağı başlamış oldu. Süreç denetimi, merkezi bir denetime toplanmış, işlemler hiyerarşik olarak dağıtılmıştır. Yönetim sistemleri, denetim odası işlemleri, süreç istasyonları ve dağıtılmış denetim fonksiyonları olarak tanımlanmıştır. 1980'lerde denetlenen süreç ile denetim merkezi arasında iletişim için koaksiyel kablo ve optik kablo kullanılmaya başlamıştır.

Şekil 1.4'de 1980'lerde ki süreç yönetiminin durumu görülmektedir.



Şekil 1.4. 1980'lerde süreç yönetiminin durumu.

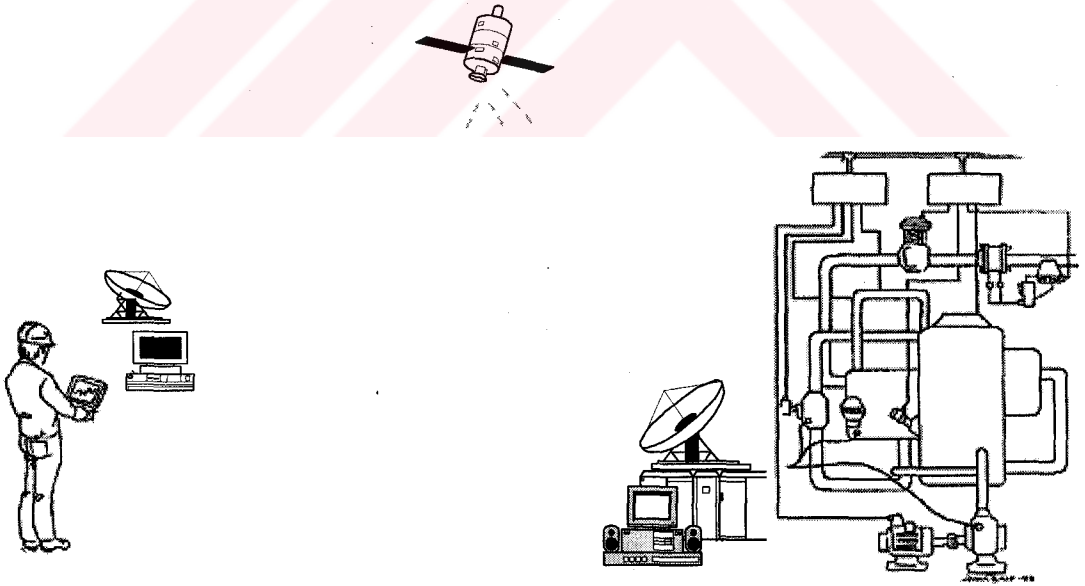
1990'larda, mikroişlemci ve elektronik bileşenlerin süreç yönetimine entegre olması giderek artmıştır. Şekil 1.5'de 1990'larda süreç yönetiminin durumu görülmektedir.



Şekil 1.5. 1990'larda süreç yönetiminin durumu.

Saha ekipmanları ve veri yolları geliştirme projeleri bu yıllarda başlamıştır. Saha ekipmanlarının yaptıkları iş ile fiyatları karşılaştırıldıklarında, fiyatlarının ucuzlamış olduğu görülmektedir. Ayrıca bu ekipmanların fiyatlarının dışında güvenilirlikleri artmış ve bakım ihtiyaçları azalmıştır. Sürecin belirli noktalardan yönetilmesi ile bu yönetim noktalarında büyük oranda bir karmaşa meydana gelmiş ve bu karmaşa sebebiyle çeşitli aksaklıkların ortaya çıkma riski oldukça artmıştır. Bilgisayar tabanlı denetim ile bu riskler oldukça azaltılmıştır.

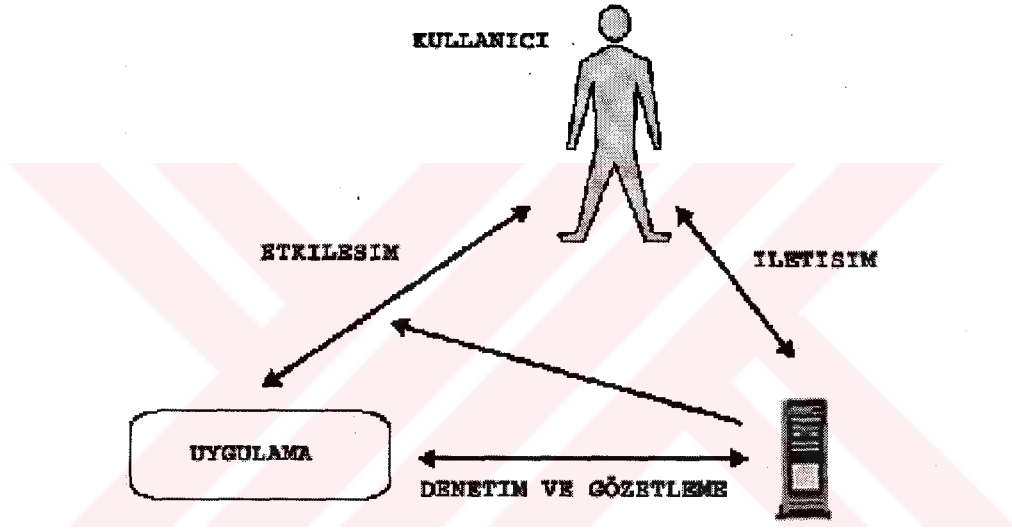
Şekil 1.6'da 2000'lerde ki süreç yönetiminin durumu görülmektedir. 2000'li yıllarda süreç denetiminde meydana gelen en önemli gelişme veri yollarında meydana gelmiştir. Sürecin yerinden denetimi yerine, şehirler arası veya ülkeler arası denetim uydu sistemlerinin kullanımı mümkün hale gelmiştir. Bunun sonucunda sistemde meydana gelen herhangi bir değişiklik merkez tarafından en kısa zamanda görülebildiği gibi, merkeze arıza durumlarında ise en kısa zamanda müdahale edebilme yeteneğini kazandırmıştır.



Şekil 1.6. 2000'lerde süreç yönetiminin durumu.

2. GERÇEK ZAMANLI SİSTEMLER

Endüstriyel otomasyonda kullanılan gerçek zamanlı (Real-Time) sistemler, uzak bir noktada veya bir fabrikada gerçekleşen işlemi bir merkezden görüntüleme ve kontrol etme amaçlı olarak kullanılan sistemlerdir. Genel olarak bütün gerçek zamanlı sistemlerde var olan bileşenler Şekil 2.1’de görülmektedir.

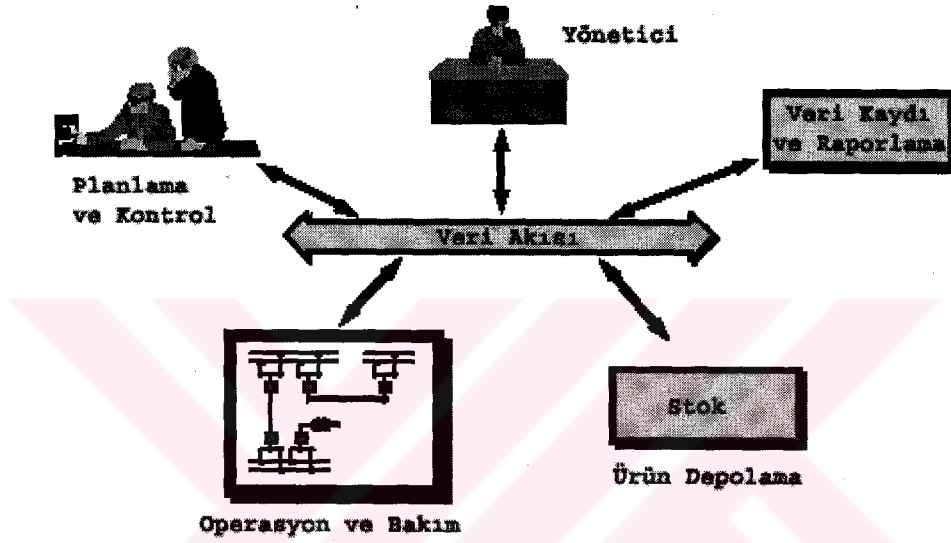


Şekil 2.1. Gerçek zamanlı sistem bileşenleri.

Gerçek zamanlı sistemler; kullanıcı, insan makine etkileşimi (MMI: Man Machine Interaction), iletişim ortamları ve uygulama olmak üzere dört temel bileşenden oluşur. Bunlar;

- i) Kullanıcı: operatör, teknisyen, bakım uzmanı veya yönetici.
- ii) MMI : uygulamadan aldığı verileri, kullanıcının anlayabileceği rapor, grafik canlandırma ve çeşitli uyarı işaretlerini üreten kısım.
- iii) İletişim Ortamları: insan ile makine arasındaki veri alışverişini gerçekleştirmek için kullanılan ortamlar.
- iv) Uygulama: insan ile makine arasındaki iletişimi gerektiren asıl amaç uygulamadır.

Şekil 2.2’de gerçek zamanlı sistemlerde veri akışı görülmektedir. Şekilde görüldüğü gibi gerçek zamanlı sistemlerde sahadan toplanan veri değerlendirilmek üzere yönetim birimi, planlama ve denetim birimi, ürün depolama birimlerine gönderilir [2]. Ayrıca sahadan toplanan veri planlama ve denetim biriminin isteklerine bağlı olarak depolanır.



Şekil 2.2. Gerçek zamanlı sistemlerde veri akışı.

Gerçek zamanlı sistemler temel olarak dört ayrı uygulama kapsamında toplanabilir.

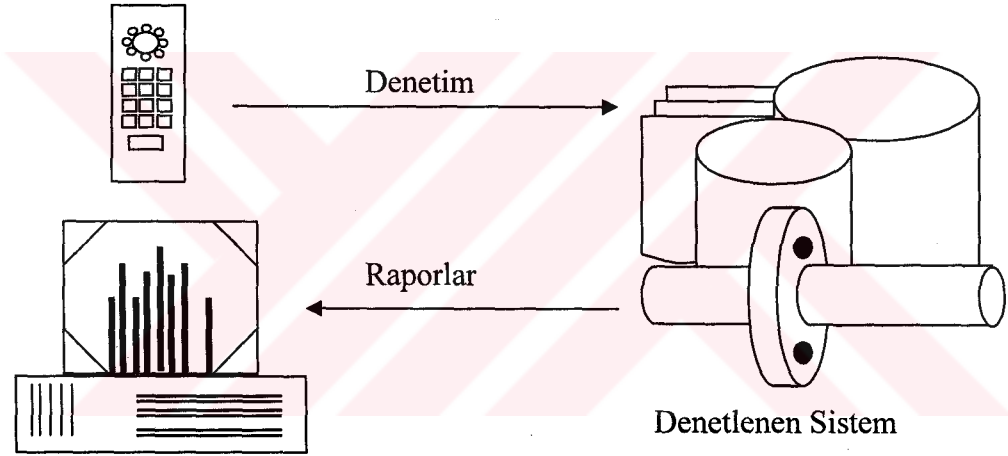
Bunlar;

- i) Telemetri Sistemleri
- ii) DCS Sistemleri
- iii) PLC Sistemleri
- iv) SCADA Sistemleri

Bu dört uygulama arasındaki en eski uygulamaya sahip olan Telemetri sistemleridir. Diğer üç sistem ise yaklaşık olarak aynı tarihsel dönemlerde kullanılmaya başlamıştır. Bu dört sistem yüzeysel olarak birbirine çok benzemesine rağmen, kullanım amacı ve kullanıldıkları alanların fiziksel boyutlarına bağlı olarak farklılıklar göstermektedir.

2.1. Telemetry Sistemleri

Telemetry kelimesi, iki kelimenin bileşirmesi ile meydana gelmiş bir terimdir. Telemetry sistemlerinde, “tele” kelimesi mesafe, “metri” de ölçüm manasına gelerek Telemetry sistemlerinin fonksiyonlarını ifade eder. Tarihsel gelişim olarak bakıldığında, telemetry sistemlerinin sanıldığı gibi aksine elektronik sistemlerden hatta elektriğin keşfinden önceleri de kullanıldığı görülmüştür. Amerikan yerlilerinin av bölgelerinde varolan çeşitli av hayvanları hakkındaki sayı ve yer bilgilerinin uzun mesafelere duman yolu ile iletmesi, telemetry sistemlerinin aslında 1900’lü yılların ikinci yarısından çok önceleri kullanıldığına en iyi örneklerden biridir. Şekil 2.3’de Telemetry sisteminin temel yapısı görülmektedir [3].

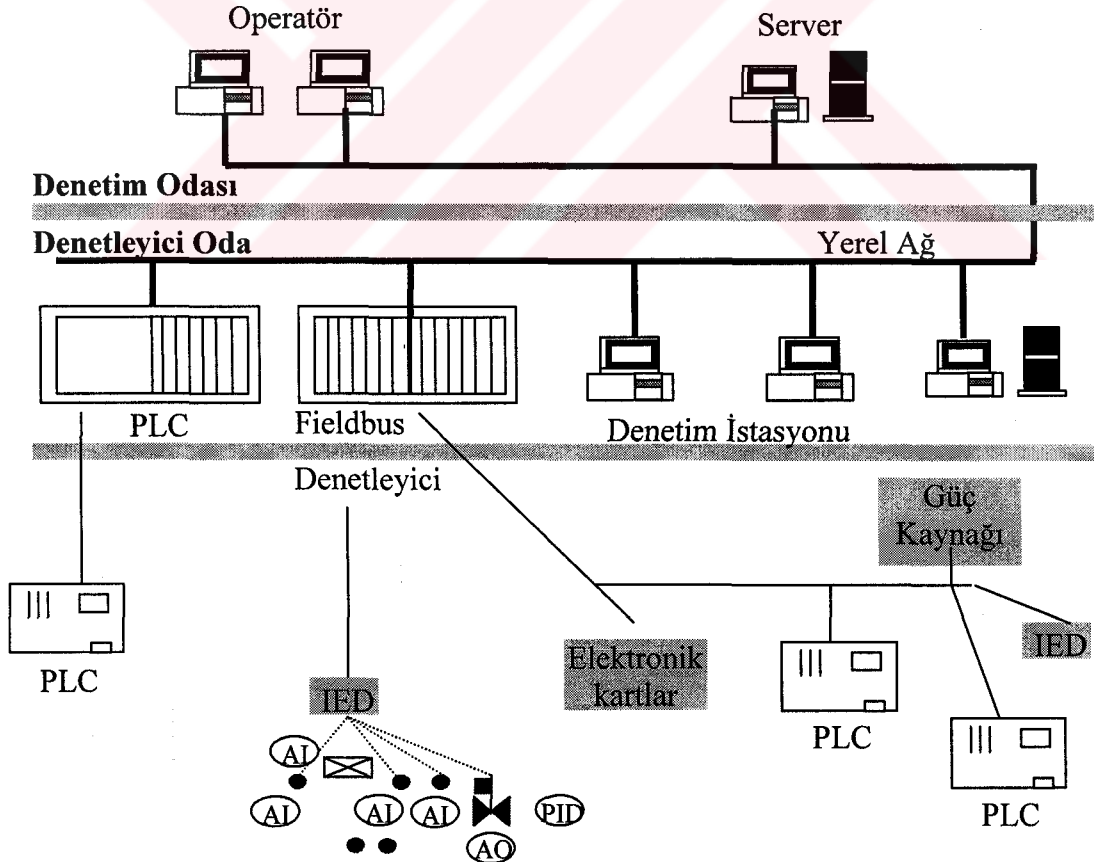


Şekil 2.3. Telemetry sisteminin temel yapısı.

Telemetry sistemleri, uzakta bulunan sistemi radyo frekansı kullanan RTU (Remote Terminal Unit) cihazları kullanarak denetler. RTU ile ilgili geniş açıklama dördüncü bölümde yapılmıştır. Denetleme ve gözetleme işleminin gerçekleştiği merkezde ise önceleri çeşitli analog ölçü aletleri ve sinyal lambaları kullanılmasına karşın günümüzde ise gelişmiş bilgisayarlar, denetlenen sistemde istenmedik durumlarda operatörleri ses ile uyarmak için sirenler ve çeşitli ikaz lambaları kullanılmaktadır.

2.2. Dağılık Kontrol Sistemleri (DCS)

Dağılık Kontrol Sistemleri (DCS: Distributed Control System), ilk olarak 1970'lerin ortalarında Honeywell ve Yokogawa ortaklığı tarafından geliştirilmiştir. DCS, sürece yönelik bir kontrol sistemidir. Sürecin bir merkezden görüntülenmesini sağlar. DCS merkez istasyonu ile Giriş/Çıkış (I/O) birimleri arasındaki mesafe oldukça kısadır. Bundan dolayı istasyon ile birimler arası iletişimde kablolu ağ yapısı ve çeşitli saha veri yolları (fiber optik, fieldbus, profibus v.b.) kullanılır. DCS operatörü saha ekipmanlarından bilgi almak istediğinde, doğrudan sisteme erişebilir ve cevap alabilir. Sahada meydana gelen olaylar doğrudan sisteme kesme (interrupt) verisi olarak iletilir. DCS sistemleri veri kaynaklarına doğrudan bağlıdır. Yedekleme sistemleri ana merkeze doğrudan paralel olarak çalışır. DCS sistemleri oldukça karmaşık süreçlerin denetiminde kullanılmaktadır. Şekil 2.3'de örnek bir DCS sisteminin yapısı görülmektedir [4].



Şekil 2.3 Örnek bir DCS sisteminin yapısı.

Şekil 2.3’de görüldüğü gibi sahadaki cihazlar veri yolu denetleyicisi üzerinden DCS merkezine bağlanır.

2.2.1. DCS sistemlerinin özellikleri

DCS sistemleri diğer gerçek zamanlı denetim sistemlerine göre bazı üstünlüklere sahiptir [5]. Bunlar ;

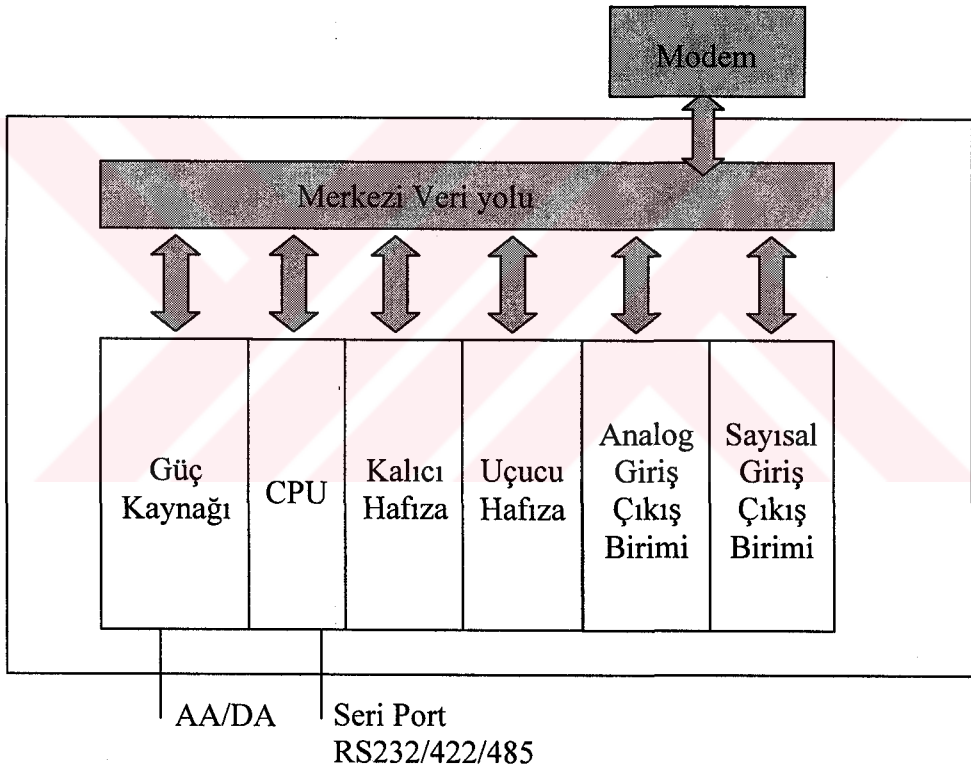
- i) Bu sistemler yüksek hızlı iletişim ağlarını kullanır. Dijital motor sürücülerini ve hızlı iletişim ağına sahip SCADA sistemleri ile çalışabilir .
- ii) DCS sistemlerinde, PC ve PLC’ler sistemi kontrol etmek amacıyla kullanılır. PC ve PLC ile DCS sistemi arasında veri iletişimi çok hızlı olduğundan, kontrol merkezi sistemdeki sıcaklık, basınç ve bunun gibi değişimlere en kısa zamanda müdahale edebilir.

DCS sistemleri bazı üstünlükleri olmasına karşın, bazı dezavantajlara da sahiptir. Bunlar;

- i) DCS sistemlerine yeni birimlerin ilave edilmesi, diğer gerçek zamanlı sistemlere nazaran oldukça zordur. Bunun sebebi ise, DCS sistemleri belirli bir süreci denetlemek amacıyla tasarlanmıştır. Bu tasarım, denetim için gerekli olan en uygun veri yollarının seçilmesi ile gerçekleştirilmiştir. İlave edilen her yeni birim, denetim merkezi ile sürecin arasındaki veri akışını yavaşlatacak etkiyi yaratmaktadır. Bu etki sebebiyle, DCS sistemlerine yeni birimlerin ilave edilmesi oldukça zorlaşmaktadır.
- ii) DCS sistemlerinde, saha ile denetim merkezi arasındaki veri akış hızı oldukça yüksektir. Bu veri akış hızını sağlayabilmek amacıyla, yüksek hızda çalışan veri yollarına ve bu sistemler ile çalışabilecek yüksek kalitede ekipmana ihtiyaç duyulmaktadır. Bu özel gereksinimleri sebebiyle DCS sistemleri oldukça maliyetli sistemlerdir.

2.3. PLC Tabanlı Sistemler

PLC, mikroişlemci tabanlı bir denetleyicidir. Birçok endüstriyel üretim sistemlerinde kullanılmaktadır. PLC, ilk olarak 60'lı yılların sonlarına doğru Modicon firması tarafından endüstriyel süreçlerin denetimi için üretilmeye başlanmıştır [6]. 1980'li yıllarda PLC teknolojisinde sekiz bitlik yapı kullanılmaya başlamıştır. Sekiz bitlik yapının kullanılması sonucunda PLC'nin tarama süresi düşürülmüş ve daha hızlı çalışır hale gelmiştir. Günümüzde ise 64 bitlik haberleşme yapabilen PLC'ler geliştirilmiştir. Şekil 2.4.'de PLC'nin temel yapısı görülmektedir.



Şekil 2.4. PLC'nin temel yapısı.

PLC beş temel bölümden oluşmaktadır. Bunlar;

- i) Güç Kaynağı, bazı PLC'lerde entegre, bazılarında ise modüler yapıda bulunmaktadır. Genellikle büyük endüstriyel sistemlerin denetlenmesinde kullanılan PLC'lerde güç kaynakları modüler yapıdadır.

- ii) CPU (Central Processing Unit: Merkezi İşlem Birimi) uygulamayı denetlemek için yazılan program bu kısımda yürütülür.
- iii) Hafıza iki kısımdan meydana gelmektedir. Kalıcı hafıza, PLC'nin işletim sistemi ile ilgili program bu kısımda saklanır. Uçucu hafıza süreçle ilgili çeşitli verilerin ve PLC programındaki çeşitli hesaplamalarda elde edilen ara değerlerin saklanmasında kullanılır. Bu hafıza tipinde bilgi kaybını önlemek için genellikle hafıza pilleri veya büyük kondansatörler kullanılmaktadır.
- iv) Analog ve sayısal giriş/çıkış birimleri, sahadaki cihaz ile PLC arasındaki fiziksel bağlantının kurulması için kullanılır. Giriş/çıkış birimi sayısı genellikle ihtiyaca bağlı olarak arttırılabilir.
- v) İletişim arayüzü, PLC'nin programlanabilmesi için bilgisayar ile PLC arasındaki bağlantıyı sağlar. İlave olarak PLC ile gerçekleştirilen ana-uydu mimarisinde her istasyon arasındaki iletişim için gerekli noktaları meydana getirir.

PLC'ler birçok avantajlara sahiptir [7-12]. Bunlar;

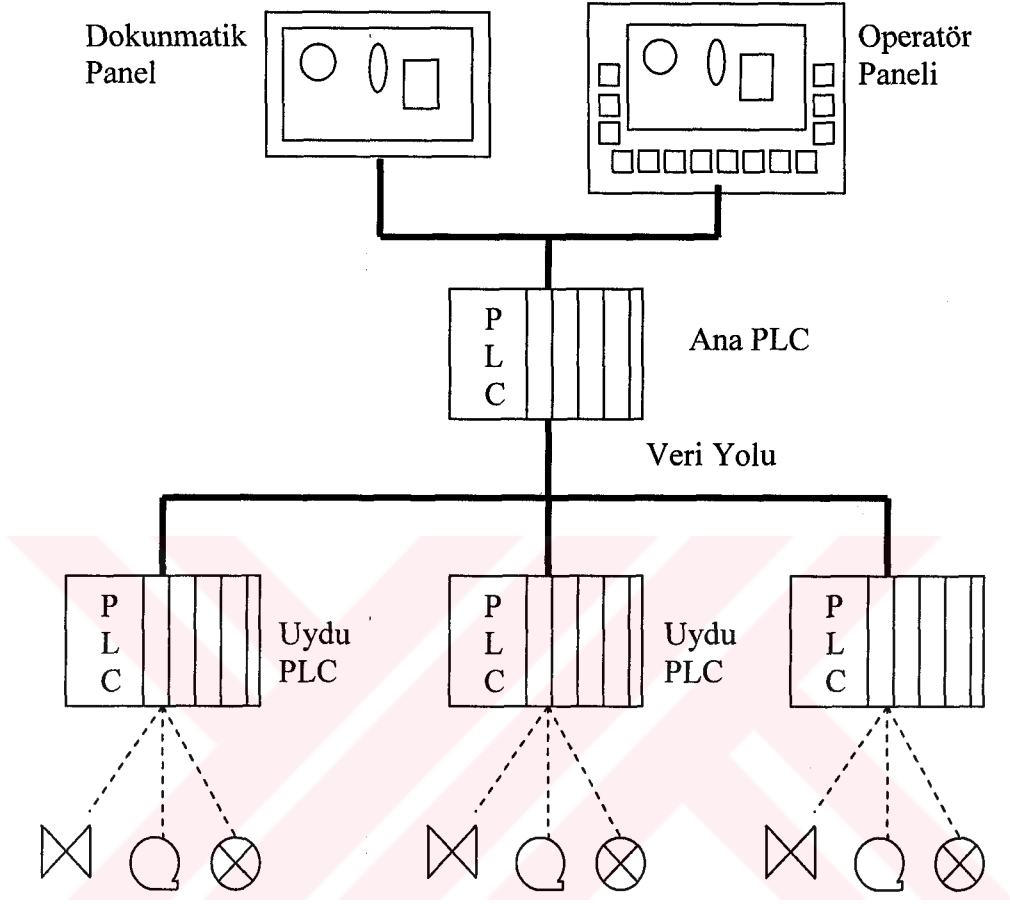
- i) PLC'ler giriş arayüzünden süreç değerlerini çok hızlı olarak okuyabilir. Aynı şekilde PLC'de varolan programdan elde ettikleri değerleri çıkış arayüzüne yazabilir.
- ii) PLC'ler sürecin çalışması esnasında bilgisayar ve diğer sistemler (SCADA, DCS ve PLC sistemleri) ile haberleşme yeteneğine sahiptir.
- iii) PLC'ler endüstriyel ortamlara bilgisayarlardan çok daha fazla dayanıklıdır.

Sahip olduğu avantajların dışında bazı dezavantajları da bulunmaktadır. Bunlar;

- i) Büyük miktarda veri alabilme kabiliyetleri düşüktür.
- ii) Veritabanına yazabilme ve okuyabilme kabiliyetleri düşüktür.
- iii) Detaylı olarak raporlama kabiliyetleri düşüktür.
- iv) Operatöre süreç hakkında bilgi gösteriminde yeterli değildir.

Şekil 2.5'de PLC tabanlı gerçek zamanlı denetim sistemi görülmektedir. Bu sistem, bir adet ana ve üç adet uydu PLC'den ve operatörün sistem hakkında bilgi edinebilmesi ve çeşitli denetim işlerini yerine getirebilmesi için kullanılan, dokunmatik panel ve operatör panelinden oluşmaktadır. Ana PLC, uydu PLC'leri denetim programına göre denetler. Ana PLC, uydu PLC'lerden elde ettiği verileri

operatör ve dokunmatik panelde çeşitli uyarı ve grafik canlandırmalar haline getirir. Böylece operatör ile makine arasındaki iletişim arayüzü gerçekleştirilir.



Şekil 2.5. PLC tabanlı gerçek zamanlı denetim sistemi.

2.4. SCADA Sistemleri

Endüstriyel otomasyondaki gerçek zamanlı sistem bileşenlerindeki en son ulaşılan nokta SCADA sistemleridir. Bir sonraki bölümde SCADA sistemleri detaylı bir biçimde anlatılmaktadır.

3. SCADA SİSTEMLERİ

Gözetleyici Denetim ve Veri Toplama (SCADA: Supervisory Control And Data Acquisition) ifadesi ilk olarak güç endüstrisinde 1971'de Arkla Energy Resources (AER) tarafından ortaya atılmış bir terimdir. "Supervisory Control and Data Acquisition" terimi ilk olarak PICA (Power Industry Computer Applications) konferansında 1973'te yayınlanmıştır. AER firması tarafından, Fisher Corporation firmasından alınan DC2 bilgisayarına ilk SCADA sistemi kurulmuştur [14].

SCADA sistemi, veri toplama ve Telemetri'nin bir kombinasyonudur. Veri toplama ve merkezden veri gönderme, analiz yapma ve daha sonra bu verilerin bir operatör ekranında gösterilmesi işlevlerini gerçekleştirir. SCADA sistemi saha ekipmanlarını görüntüler ve aynı zamanda denetler.

SCADA sistemleri alarm temellidir. Bu özellik de SCADA sistemini diğer sistemlerden ayırt eden en temel farktır. Bu yüzden SCADA sistemleri denetlenen sistemin anlık değerlerinin görüntülenmesi yerine, sahada meydana gelen herhangi bir istenmeyen durumu merkeze tarih ve saat belirterek rapor etmesi ve operatörlere gerekli uyarıları iletmesi için kullanılmaktadır.

3.1. SCADA Sistemlerinin Tarihsel Gelişimi

İlk SCADA sistemleri 1980'li yıllarda DOS işletim sistemi altında çalışan günümüz SCADA yazılımlarına göre çok daha basit sayılabilen ve insan-makine ara yüzüne sahip basit yazılımlardı. 1980'li yılların sonuna doğru Windows işletim sisteminin ortaya çıkması ile günümüz SCADA yazılımlarının ilk örnekleri ortaya çıkmaya başlamıştır. 90'lı yılların ilk çeyreğinde ise çeşitli SCADA yazılım üretici firmaları ile Microsoft arasında işbirliği anlaşmaları yapılması ve ilk NT (Network) platformu altında hizmet veren SCADA yazılımları gerçekleştirilmiştir. 90'lı yılların ortalarında ise ODBC (Open Data Base Conectivity) ve OLE (Object

Link Embeded) ile OPC (OLE for Process Control) çalışmalarına başlanmıştır. OLE otomasyonlu SCADA yazılımları hizmete sunulmuştur. Üçüncü çeyrekte ise Microsoft'un SQL'ini (Sequantial Query Language) içeren endüstriyel SQL Server yazılımı gerçekleştirilmiş ve çeşitli SCADA yazılımları COM (Common Object Model), DCOM (Disturbed Common Object Model) ve ActiveX'i desteklemeye başlamıştır. Günümüz SCADA yazılımları, Microsoft Office programlarına ve çeşitli programlara sahadan topladıkları verileri aktarabilmekte, bu verileri grafik veya analiz sonuçları olarak gözlemleyebilme imkanı sağlamaktadır. Ayrıca internet sayesinde dünyanın her yerinden saha ekipmanları durumları görüntülenebilmekte ve denetlenebilmektedir. Son gelişmeler ile birlikte cep telefonları WAP (Wireless Access Protocol) aracılığı ile saha hakkında gerekli bilgiler cep telefonu ekranına gelmektedir. Tablo 3.1'de SCADA sistemlerinin tarihsel gelişimi gösterilmiştir.

Tablo 3.1. SCADA sistemlerinin tarihsel gelişimi.

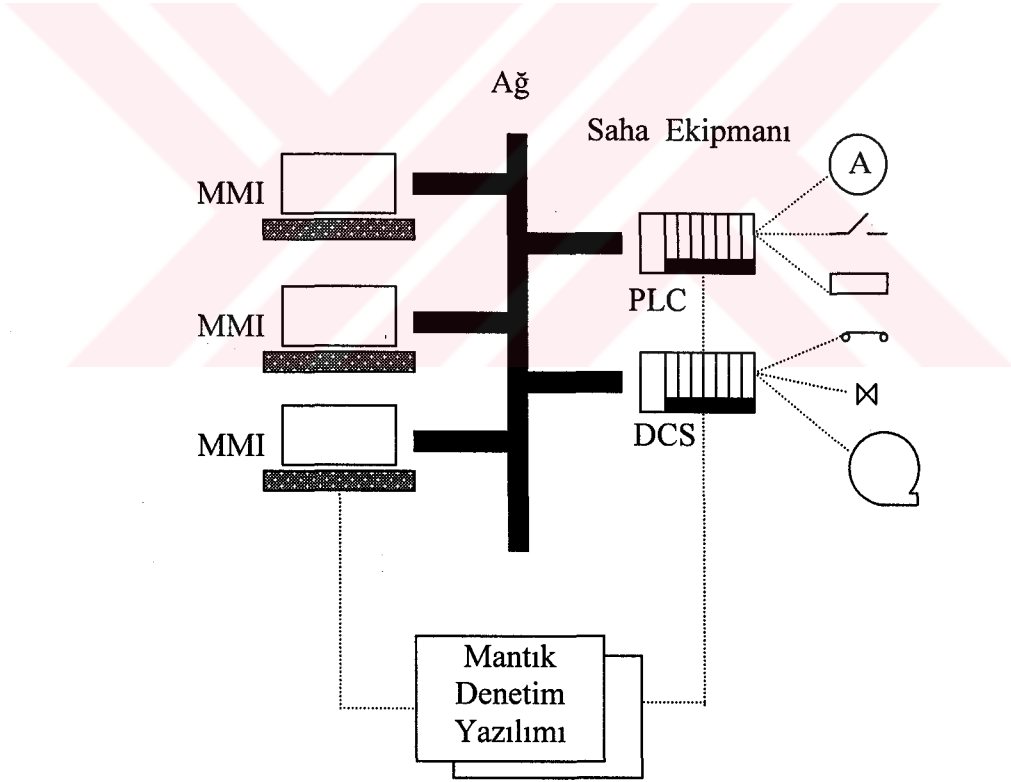
Tarih	Gelişmeler
1973	PICA (Power Industry Computer Applications) konferansında ilk uygulama denemesi tanıtılmıştır.
1980	DOS altında çalışan ticari uygulama gerçekleştirilmiştir.
1989	Windows 2.3 altında çalışan ilk Windows tabanlı MMI programı sunulur.
1990	Microsoft ile çözüm ortağı olur.
1992	NT 3.x için Microsoft'un DDE ile NetDDE bağlaşımları yaygınlaştırılır.
1994	Windows NT 3.1 altında çalışan ilk MMI programı sunulur.
1995	ODBC ve OLE ile OPC çalışmalarına başlanır.
1996	Wonderware, Microsoft'un Back Office sertifikalı ve OLE otomasyon destekli ilk endüstriyel otomasyon ürününü sunar.
1997	Microsoftun SQL'ini içeren Industrial SQL Server yazılımı için anlaşmaya varılır.
1998	COM, DCOM ve ActiveX'i desteklemeye başlar.
2000	Internet ile SCADA sisteminin birleşmesi gerçekleşir.

3.2. SCADA Sistemlerinin Temel Yapısı

Günümüz SCADA sistemleri bir çok kısımdan oluşmasına rağmen, bir SCADA sistemi dört temel kısımdan oluşmaktadır [15]. Bunlar ;

- i) Saha ekipmanları,
- ii) Veri ağları (data network),
- iii) Mantık denetim yazılımı,
- iv) İnsan-makine arayüzü (MMI).

Şekil 3.1’de SCADA sisteminin temel yapısı görülmektedir. Sahada bulunan çeşitli ölçü aletleri, pompa sistemleri, anahtar ve vanalardan alınan veriler PLC ve DCS sistemleri vasıtasıyla sunucuya iletilir. Sunucu sahadan topladığı verileri mantıksal denetim işlemlerine dönüştürerek çeşitli denetim sinyalleri üretir.



Şekil 3.1. SCADA sisteminin temel yapısı.

SCADA sistemlerinin temel bileşenleri aşağıda açıklanmaktadır.

3.2.1 Saha ekipmanları

Saha ekipmanları, Şekil 3.1’de görüldüğü gibi, sahadan veri toplama veya denetim verilerini sahadaki cihazlara gönderen ve SCADA hiyerarşik yapısında ikinci sırada olan cihazlardır. Bunlar PLC, DCS ve Akıllı Elektronik Cihazlar (IED: Intelligent Electronic Devices) olarak sayılabilir.

3.2.2 Veri ağları

Denetlenen veya görüntülenen sistem ile insan-makine arayüzündeki veri iletişimini sağlayan kanallardır. Bu kanallarda iletişimi sağlamak için çeşitli ortamlar, protokoller ve veri yolları sistemleri kullanılır. Veri ağlarında kablo, özel kiralık hat, Radyo Frekans (RF) ve uydu bağlantısı ile iletişim yaygın olarak kullanılmaktadır. Kablo ile gerçekleştirilen iletişim ortamında saha ile MMI arası mesafe kısadır. Kablolü iletişim iki temel kısımda incelenebilir. Bunlardan ilki koaksiyel kablo ile gerçekleştirilen iletişimdir İkincisi ise fiber optik kablo ile gerçekleştirilen iletişimdir.

Bu iletişim ortamlarına ilave olarak dial-up bağlantı, RF ve uydu link bağlantı ortamları da vardır. Bu bağlantı ortamları farklı durumlarda Uzak Uç Birimlerindeki (UUB) veri iletişimini kesintisiz sürdürebilmek için tercih edilmektedir. Dial-up bağlantı, UUB ile Ana Denetim Merkezi (ADM) arasındaki mesafe belirli bir değeri aştığında kablolü iletişime nazaran daha ekonomik bir hal alması sebebiyle tercih edilmektedir. Kablolü iletişim ortamlarında belirli mesafe aşımalarında veri kayıplarını önlemek için tekrarlayıcı kullanılması gerekmektedir. Tekrarlayıcıların kullanılması sebebiyle maliyet artmaktadır (örneğin şehirler arası mesafelerde fazla sayıda tekrarlayıcıya gereksinim vardır). Radyo-link ise UUB ile ADM arasındaki mesafe koaksiyel kablo kullanamayacak kadar fazla olan durumlarda tercih edilmektedir. Fakat RF’nin hava şartlarından etkilenmesinden dolayı güvenlik sebebi ile dial-up bağlantı ortamı her zaman kullanıma hazır durumda bekletilir. Üst düzey güvenlik isteyen denetim sahalarında (şehir doğal gaz denetim merkezleri, su dağıtım denetim merkezleri ve ana elektrik dağıtım denetim merkezleri gibi) veri iletişimini

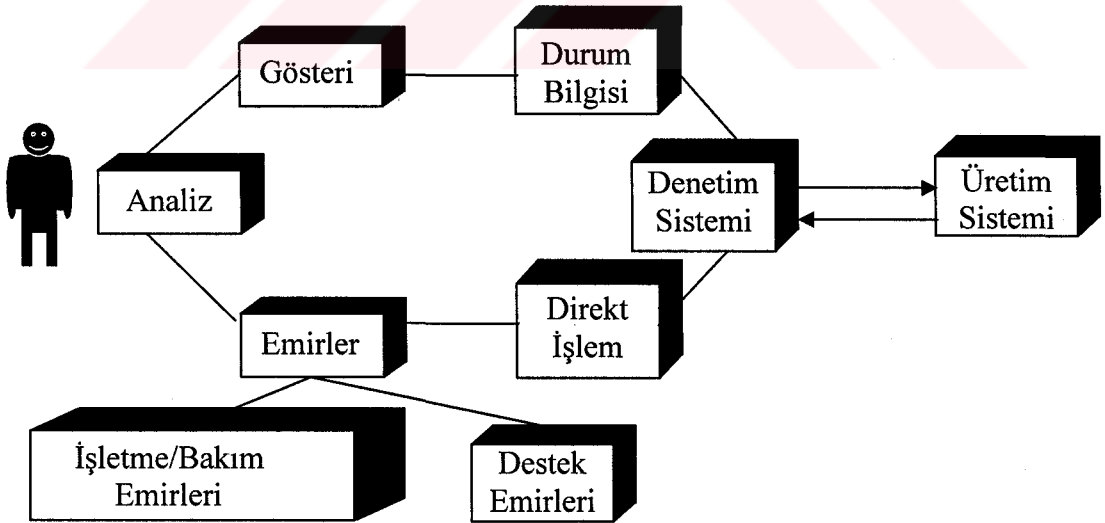
kesintisiz sağlamak için diğer iletişim ortamlarına ek olarak daima kullanıma hazır uydu yer istasyonlarını da kullanabilir. Yukarıda anlatılan iletişim ortamları sayesinde UUB ile ADM arasında kesintisiz veri iletişimi sağlanır. Bölüm 4’de iletişim ortamları ile ilgili detaylı bilgi verilmiştir.

3.2.3 Mantıksal denetim yazılımı

Mantıksal denetim yazılımı SCADA sistemlerinde iki ayrı bölümde yer alır. Birincisi PLC ve DCS sistemlerinde bulunmaktadır. İkincisi de SCADA sistemini oluşturan sunucularda bulunan sistemi denetleyen mantıksal denetim yazılımlarıdır.

3.2.4 İnsan-makine ara yüzü (MMI)

SCADA sistemlerinde birçok denetim makinelerle yapılmakla birlikte operatörlere de görevler düşmektedir. Bu görevleri aktif olarak operatör kendi konsolunu kullanarak yerine getirmektedir. Şekil 3.2’de insan-makine arayüzü görülmektedir.



Şekil 3.2. İnsan-makine arayüzü.

Bir operatör konsolunda normal olarak bulunması gereken ekipmanlar, monitör, klavye, fare, gerekiyorsa ışıklı kalem, telefon ve telsiz olarak sayılabilir.

Sistem denetim merkezinde normal olarak birden fazla operatör konsolu vardır. Bunların her biri sistemin işleyişi için üzerine düşen görevleri yerine getirir. Günümüzde iletişim teknolojisinin ilerlemesi sayesinde uzak konsollar da gerekirse işletim için kullanılabilir. Bu uzak konsollar merkeze kablo veya modem aracılığıyla bağlanabilir.

Konsollarda görevli personelin her birinin sisteme erişimi, ağ yazılımları aracılığıyla denetlenebilir. Her operatörün tanıtım ismi ve şifresi vardır. Bu sayede yetkisiz kişilerin sisteme müdahalesi önlenir. Ayrıca bahsedilen yazılım sayesinde istenen operatörlere sistemin istenen kısımlarına müdahale izni verilebilir veya engellenebilir. Örneğin A isimli operatör sadece raporlara erişebilir, uyarı, alarm mesajlarını görebilir ve sistemin işleyişine denetim ve kumanda edemez şeklinde tanımlamalar yapmak, SCADA sisteminin avantajları arasındadır.

İnsan makine arasındaki iletişimi kuvvetlendirmek için birçok duyu organlarına hitap edecek mesajlar vermek gereklidir. Bu yüzden gözlerimize hitap eden monitörlerin yanında, kulaklarımıza hitap eden sesli uyarılar da mevcuttur.

Kullanıcı kendi konsolunu kullanarak yetkili olduğu kadarıyla cihazların ve ünitelerin durumlarını izleyebilir, raporlar alabilir, tanımlama formları girebilir, hesaplamalar yapabilir, sistemin işleyişine direkt müdahale edecek program ve işleri başlatabilir, bir kısım programları devreye alabilir veya çıkarabilir.

3.3. SCADA Yazılımı Seçiminde Dikkat Edilecek Noktalar

SCADA yazılımı seçiminde dikkat edilmesi gereken birçok nokta vardır. Bu noktalar;

i) Yazılım modüler olmalıdır. Sisteme yeni üniteler eklendiği zaman kolayca sisteme entegre olabilmelidir.

- ii) İşletmede ihtiyaçlara anında cevap verebilmelidir. Bilgisayara veri gönderilmesi, verinin işlenmesi ve denetim cihazına geri gönderilmesi arasında geçen zaman çok kısa olmalıdır.
- iii) Kullanıcı kolaylığı sağlamalıdır. Yazılım fazla karmaşık olmamalı, güç üretim sistemleri ve mühendislik konusunda bilgi sahibi bir kişinin rahatlıkla üst düzey bir programlama dili bilmesine gerek kalmadan kullanabileceği basitlikte olmalı, bunun yanında istenen ihtiyaçları karşılamalıdır. Kurulduğu zaman hazır standart hesaplar, raporlar bulunmalı, yeni hesap ve raporları tasarlamak ve uygulamaya geçirmek kolay olmalıdır. Standart ana fonksiyonlar, iterasyon hesapları, mantık denetleyiciyle çalışmaları desteklemelidir.
- iv) Veri tabanı güçlü ve güvenli olmalı, standart veri değişim tiplerine ve dillerine uyumlu olmalıdır. Arka planda bir veri tabanı motoru çalışmalı bu motor sorgulamaları hızlandırmalıdır. Veri tabanında şişme meydana gelmemeli hata oluşmamalıdır. Veri tabanları alan ekleme mimarisine açık olmalı, alan tanımları değiştirildiğinde mevcut veriler bozulmadan kullanılabilir. Veri tabanında bir fiziksel hata meydana geldiğinde veri tabanının tamir edilebilme özelliği olmalıdır. Veri tabanı sorgulamalarında SQL gibi standart sorgulama dilleri kullanılabilir, sıkça kullanılan MDB ve DBF gibi standart veri tabanları ile iletişim kurabilmelidir.
- v) Çok kullanıcı mimariye açık olmalıdır. Yazılım birden fazla operatör tarafından kullanılacağı için çok kullanıcı mimariyi desteklemeli, çok kullanıcı sistemde çalışırken paylaşım ihlalleri oluşmamalıdır. Tamamen istemci – sunucu (Client-Server) mimarisine uygun çalışmalı, nesneye yönelik (Object Oriented) programlama mimarisi kullanılarak tasarlanmış olmalıdır. Uzak erişimlere izin verebilmelidir. Üst seviyede her kullanıcının istenen kaynaklara erişim hakları tanımlanabilmelidir.
- vi) Bütün bu ana kriterlerin yanında kullanılacak program iyi bir dokümantasyonla birlikte gelmeli, yetkili eğitim servisleri desteği bulunmalıdır.
- vii) Diğer üst düzey dillerle arasında bağ kurmak kolay olmalıdır.
- viii) Donanımlarla kolayca bağlantı kurabilmelidir.
- ix) Görsel ve kullanımı rahat olmalıdır. Grafikler ve tablolar simülasyonlarla desteklenerek anlatım kolaylığı sağlanmalıdır.
- x) Operatörlerin dikkatini çekebilmek için görsel yanıp sönen mesajlar verebilmeli, gerektiği durumlarda sesli uyarılarla verilen mesajın önemi artırılabilir.

Tüm bu özellikler parametrik olmalı, kolayca ayarlanabilmelidir.

3.4. SCADA Sistemlerinin Uygulama Alanları

SCADA teknolojisi, büyük alanlar üzerine dağılmış, görüntüleme ve denetim işlemleri gerektiren sistemlere uygulanır. SCADA sistemleri, merkezdeki bir operatöre, geniş sahalara dağılmış petrol veya gaz sistemleri, boru hatları v.b. sistemleri merkezi bir noktadan denetleyebilme imkanı sağlar. Örneğin, denetlenen sistemde var olan vana ve anahtar gibi cihazları açıp kapatabilme veya önceden ayarlanan değerleri değiştirme, gerçekleşen arızaları alarm olarak görüntüleyebilme gibi olanakları sağlar. SCADA sistemi tarafından denetlenen sistemin boyutları yüzlerce veya binlerce kilometrelik kollara sahip olduğu zaman bile, sistemde meydana gelen arızaların kısa zamanda tespit edilmesini sağlar. Bu tip sistemlere verilebilecek en iyi örnekler aşağıda sıralanmıştır. SCADA sistemleri;

- i) Elektrik güç şebekesindeki taleplerin, en kısa zamanda karşılanmasını gerektiren küçük hidroelektrik santral gruplarının müşteri isteklerine göre açılıp kapatılması, türbin vanalarının ayarlanması, sürekli görüntüleme gibi gereksinimleri karşılamak,
- ii) Petrol üretiminde, petrol kuyularını içeren büyük sistemlerdeki akış ölçüm ekipmanlarından verilerin toplanması, pompaların veya motorların açma kapama denetimleri, sistemden alınan bilgilerin en kısa zamanda görüntülenmesi ve sistem için gerekli denetim bilgilerinin oluşturulması gibi çeşitli işlemlerin yerine getirilmesi,
- iii) Uzak mesafelerde çeşitli noktalara yerleştirilmiş olan denetim elemanlarının merkezi bir noktadan, gaz, petrol, kimyasal madde ve su şebeke hatlarının denetlenmesinde kullanılır. Sistemde gerçekleşebilecek tehlikeli bir sızıntı ve bu gibi durumların en kısa zamanda tespit edilmesi ve gerekli önlemlerin alınabilmesi,
- iv) Binlerce kilometrelik bir sistemi oluşturan elektrik enerjisi iletim hatları, SCADA sistemi ile denetlenebilir. Bu tip sistemlerde anahtarların açılıp kapanması, hatlardaki yük değişimlerinin en kısa zamanda tespit edilerek gerekli önlemlerin alınmasını sağlamak gibi çeşitli işlemleri gerçekleştirmek için kullanılmaktadır.

Yukarıda açıklanan dört temel uygulama alanının dışında SCADA sistemleri aşağıda verilen endüstriyel alanlarda başarılı bir şekilde uygulanmaktadır. Bunlar;

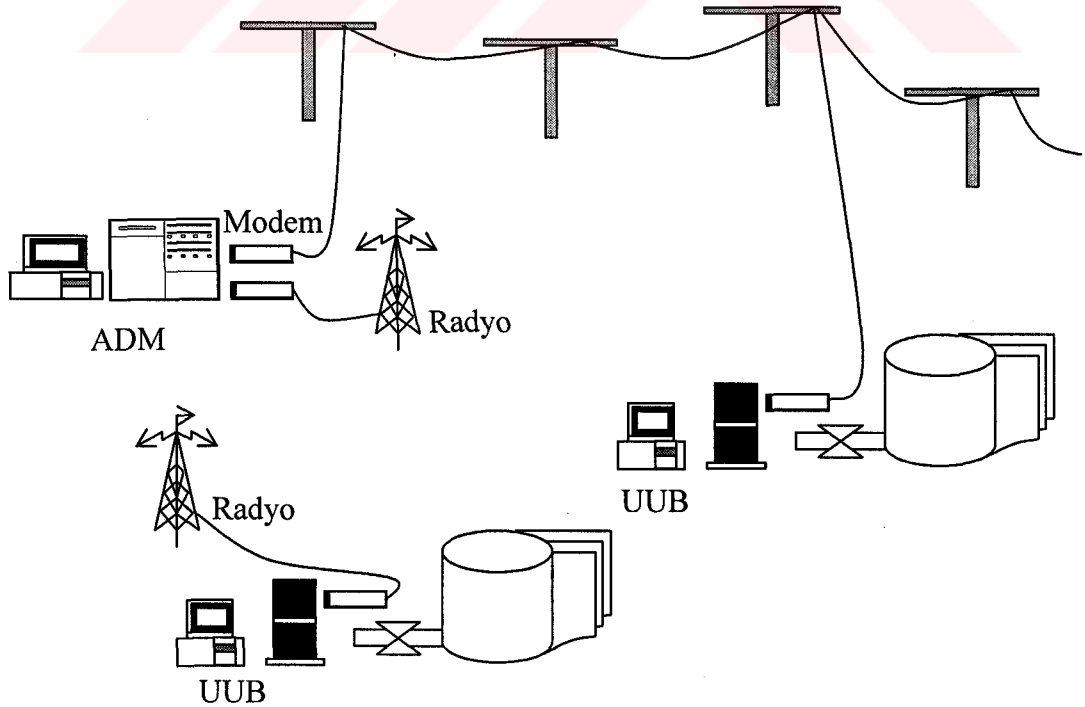
- i) Kimya endüstrisi,
- ii) Petrokimya endüstrisi,
- iii) Demirçelik endüstrisi,
- iv) Su toplama, arıtma ve dağıtım tesisleri,
- v) Hava kirliliği denetimi,
- vi) Çimento endüstrisi,
- vii) Otomotiv endüstrisi,
- viii) Trafik denetimi,
- xi) Bina otomasyonu.

3.5. SCADA Sisteminin Bileşenleri

SCADA sisteminin temel yapısı iki bileşenden oluşmaktadır. Bunlar ;

- i) Ana denetim merkezi,
- ii) Uzak uç birimi.

Şekil 3.3'de SCADA sistemini temel yapısını oluşturan bileşenler görülmektedir [16].



Şekil 3.3. SCADA sistemini temel yapısını oluşturan bileşenler.

3.5.1 Ana Denetim Merkezi

Ana Denetim Merkezi (ADM), genelde bir ana sunucudan oluşan ve sistemi denetleyen ana birimdir. İstenmedik bir duruma karşın, yedek bir sunucu ana sunucuya paralel bir şekilde çalıştırılır. ADM, sahada belirli noktalara yayılmış olan UUB'lerden veri toplar ve operatörün anlayabileceği çeşitli verilere dönüştürür. Aynı zamanda bu verilerden, ADM'de var olan program algoritması kullanılarak, sistemi denetleyecek denetim verileri üretilir. Bu denetim verileri, UUB'lere gönderilerek sistem denetlenir. SCADA sisteminin tamamı insana benzetilirse, ADM böyle bir sistemde, insan beynini temsil eder. ADM'nin sahada bulunan UUB'lerden veri toplaması 1990'ların başında dakikalarla ifade edilirken 2000'li yıllarda, iletişim ortamlarında kullanılan cihazlardaki, protokollerdeki, yazılım ve bilgisayar teknolojisindeki gelişmeler sayesinde, mili saniyelerle ifade edilmektedir.

ADM'de operatör, uzak birimlerde bulunan cihazlara, operatör giriş ve çıkış birimleri olarak da adlandırılan operatör arayüzü vasıtasıyla sisteme erişebilir. Sistem hakkında operatörü bilgilendiren cihazların tümü operatör çıkışlarıdır. Bunların en geneli bilgisayar ekranıdır. Sistem hakkındaki bilgiler ekrana çeşitli şekillerde aktarılır. Bunlar genellikle, sayısal veriler, çeşitli mesajlar, alarm bilgileri ve grafiklerdir. Yazıcılar en çok kullanılan operatör çıkışlarıdır ve sistem hakkındaki bilgilerin raporlar şeklinde alınmasını sağlar.

Bunların dışında sesli uyarı için sirenler kullanılmaktadır. Son yıllarda sistemde meydana gelen gelişmeler sayesinde otomatik şekilde operatörlerin mobil telefonlarına mesaj gönderilmesi mümkün hale gelmiştir.

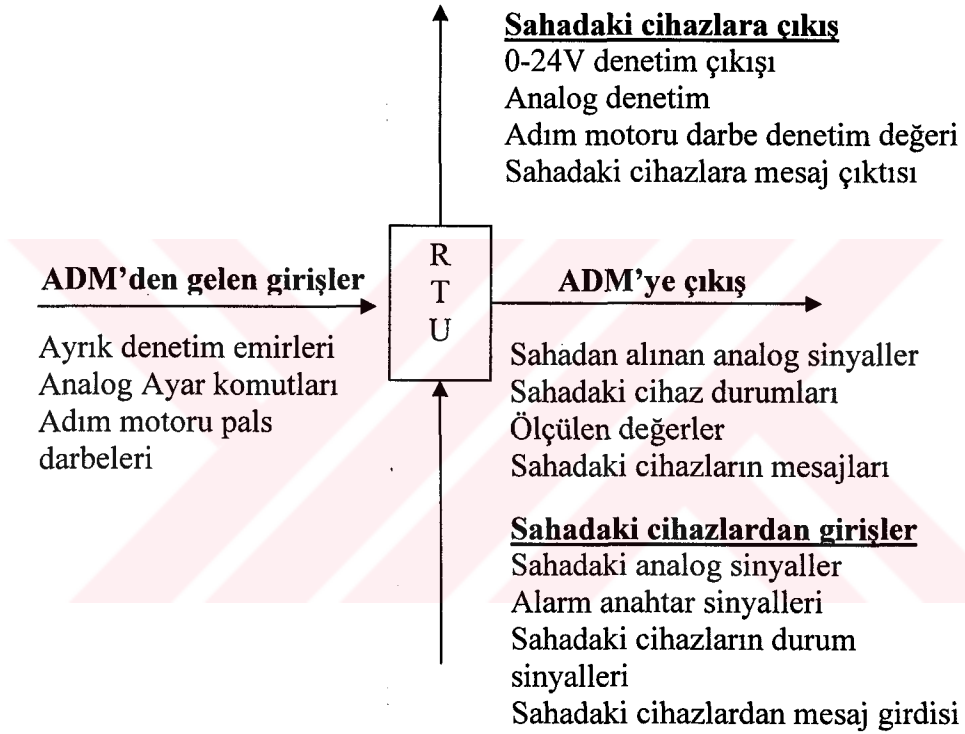
Genellikle klavye en yaygın kullanılan operatör giriş aracıdır. Bunun yanı sıra fare ışıklı kalem yaygın olarak kullanılan araçlardır. Basit sistemlerde ise giriş olarak sadece anahtar kullanılması yeterli olmaktadır [17].

3.5.2 Uzak uç birimi

SCADA sistemlerinde Uzak Uç Birimi (UUB) olarak, RTU (Remote Terminal Unit), PLC ve IED kullanılmaktadır.

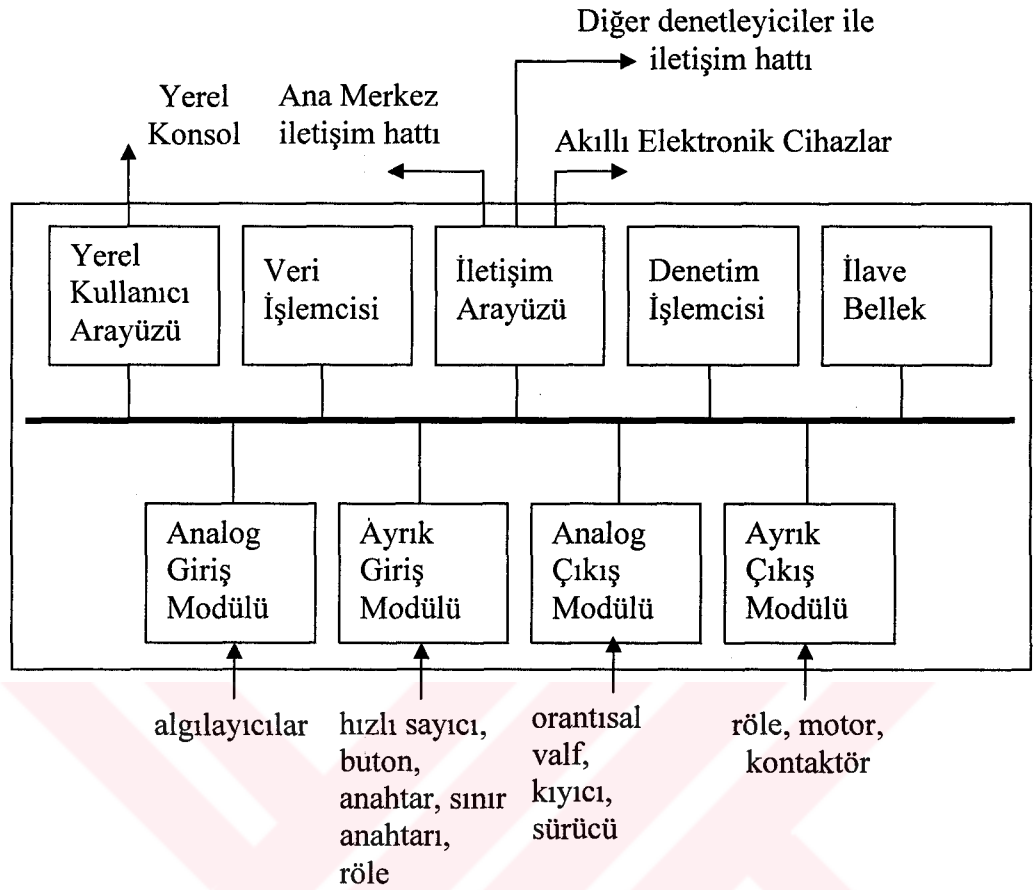
3.5.2.1 RTU'nun yapısı

Genel olarak RTU, küçük dayanıklı bilgisayar olarak çalışır. RTU, veri işleme ve UUB ile ADM arasındaki iletişimi sağlar. SCADA sistemlerinde uzak birimlerin denetlenmesinde ve görüntülenmesinde kullanılır. RTU'nun sahadan aldığı ve sahaya gönderdiği veriler, ADM'den aldığı ve ADM'ye gönderdiği veriler Şekil 3.4'de gösterilmiştir [17].



Şekil 3.4. RTU'nun veri organizasyonu.

RTU, ADM'den aldığı veriyi, sahadaki cihazların anlayabileceği analog veya ayrık sinyallere dönüştürür. ADM tarafından denetlenmek istenen sahadaki cihaza gerekli emirler RTU vasıtasıyla iletilir. Bununla beraber RTU, sahadaki çeşitli cihazlardan (algılayıcı, anahtar, röle v.b.) aldığı analog veya ayrık veriyi ADM'ye iletilmesi işlevini de gerçekleştirir. Şekil 3.5'de RTU'nun prensip şeması görülmektedir [9].



Şekil 3.5. RTU'nun prensip şeması.

Genel olarak RTU, dokuz farklı kısımdan oluşmaktadır. Bu kısımlar aşağıda tanıtılmıştır.

- i) Yerel kullanıcı arayüzü, isteğe bağlı olarak RTU'ya ilave edilebilen bir birimdir. Bu modül sayesinde, RTU'nun yanına giden bir operatör yerel bir konsol sayesinde RTU da bulunan programa kolayca müdahale edip değiştirebilir veya anlık değerleri gözleyebilir.
- ii) Veri işlemcisi, dış dünyadan RTU'ya gelen verileri veya RTU'nun veri depolama bölgesindeki verileri denetim işlemcisine gönderen işlemcidir.
- iii) İletişim arayüzü, her RTU da bulunması gereken kısımlardan biridir. İletişim arayüzü sayesinde, RTU'nun ana merkez ile iletişim kurmasını sağlar. Bunun dışında isteğe bağlı olarak diğer alt denetleyici denetim birimleri ve IED ile iletişim kurabilme yeteneğine sahip olabilir.

iv) Denetim işlemcisi, RTU'ya yüklenen programa göre gerekli denetim işlemlerini gerçekleştiren kısımdır. Denetim işlemcisinin komut işleme hızına bağlı olarak RTU'nun çalışma hızı belirlenir. Bu yüzden denetim işlemcisinin komut işleme hızı RTU seçim ölçütlerinin en önemlilerinden biridir.

v) İlave bellek, RTU'ya isteğe bağlı olarak ilave edilebilen bellektir. RTU'nun ölçümleri veya hesapladığı bazı veriler isteğe bağlı olarak bu kısımda depolanır.

vi) Analog giriş modülü, dış dünyadan RTU'nun analog sinyalleri alabilmesi için kullanılan modüldür. Dış dünyadan ölçülen analog sinyaller genellikle kullanılan algılayıcılara bağlıdır ve 0-20mA, 0-20mV, $\pm 5V$ ve $\pm 10V$ en çok kullanılan algılayıcı çıkış değerleridir.

vii) Ayrık giriş modülü, dış dünyadan RTU'nun ayrık sinyalleri alabilmesi için kullanılan modüldür. Dış dünyadan hızlı sayıcı, buton, anahtar, sınır anahtarı ve röle benzeri cihazlardan var/yok (ON/OFF) sinyallerini RTU'nun alabilmesi için kullanılan modüllerdir.

viii) Analog çıkış modülü, RTU'da var olan denetim programının ürettiği değerlere bağlı olarak dış dünyada bulunan, analog değer tabanlı cihazları denetlemek için kullanılan modüldür. Analog çıkış modülünün ürettiği sinyaller genellikle kullanılan cihazlara bağlıdır ve 0-20mA, 0-20mV, $\pm 5V$ ve $\pm 10V$ en çok kullanılan değerleridir.

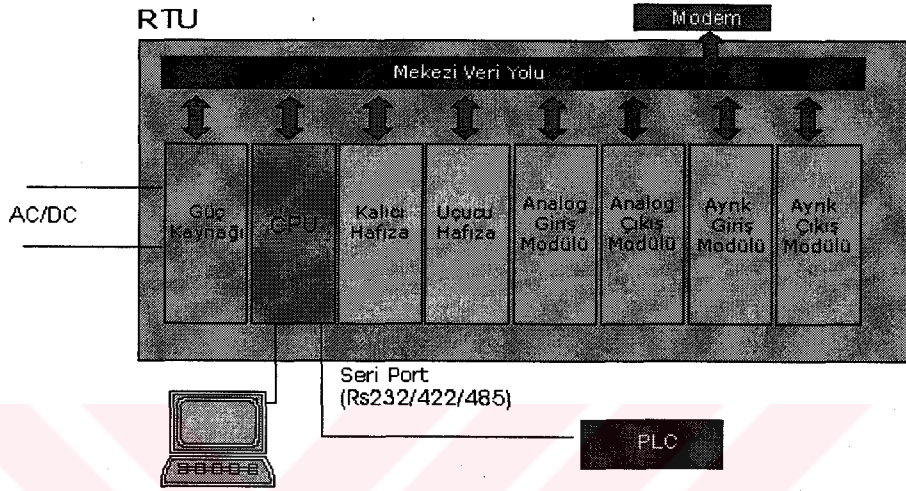
ix) Ayrık çıkış modülü, RTU'da var olan denetim programının ürettiği ON/OFF sinyallerine bağlı olarak dış dünyadaki röle, motor, kontaktör ve bunun gibi birçok cihazı denetlemek için kullandığı modüldür.

Yukarıda açıklanan modüllerin dışında bazı farklı uygulamalarda GPS (Global Position System) modülü RTU'ya entegre edilebilir. Böylece, RTU'ya GPS modülünün ilave edilmesiyle sistemdeki RTU'ların saatleri uydu üzerinden ayarlanabilir ve sistem bileşenlerinin eş zamanlı çalışması sağlanabilir.

RTU'nun programlanması PLC'nin programlanmasına oldukça benzer ve genelde Röleli Merdiven Mantığı, IEC 1131 ve Multitasking C programlama dilleri kullanılarak programlanabilir.

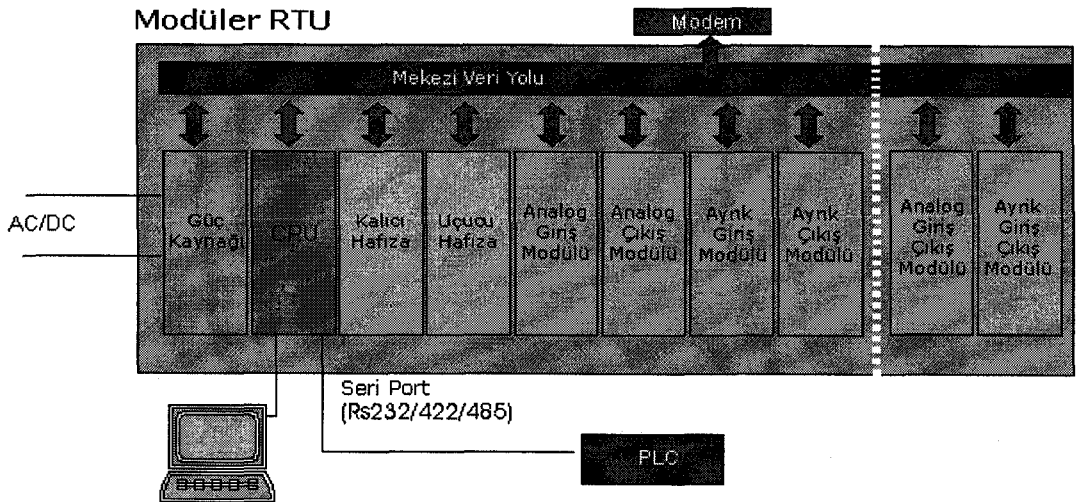
Yapısal olarak iki tip RTU vardır.

i) Modüler olmayan RTU (single board RTU): Modüler olmayan RTU yekparedir. Aynı gövdede sabit ayrı giriş ve çıkışları, analog giriş ve çıkışları ihtiva eder. Bu tip RTU'ların kapasiteleri genişletilemez. Şekil 3.6'de modüler olmayan RTU görülmektedir [18].



Şekil 3.6. Modüler olmayan RTU.

ii) Modüler RTU, bir iskelet yapıya sahiptir ve bu yapıya ayrı giriş ve çıkışlar, analog giriş ve çıkışlar talebe göre ilave edilebilir. Şekil 3.7'de modüler RTU görülmektedir.



Şekil 3.7. Modüler RTU.

Bu tip RTU yapısından dolayı, kapasiteleri genişletilebilir. Genelde RTU, 11 ile 47 arası entegre I/O birimi, 32 adete kadar entegre PID denetleyicisi ve 4 adet entegre iletişim arayüzü ile birlikte üretilmektedirler. İstendiğinde bu özellikler, ilave modüller ile kolayca arttırılabilir. RTU birçok farklı endüstriyel uygulamalarda kullanılmaktadır. Tablo3.2’de RTU’ların kullanıldığı endüstriyel uygulamalar verilmiştir.

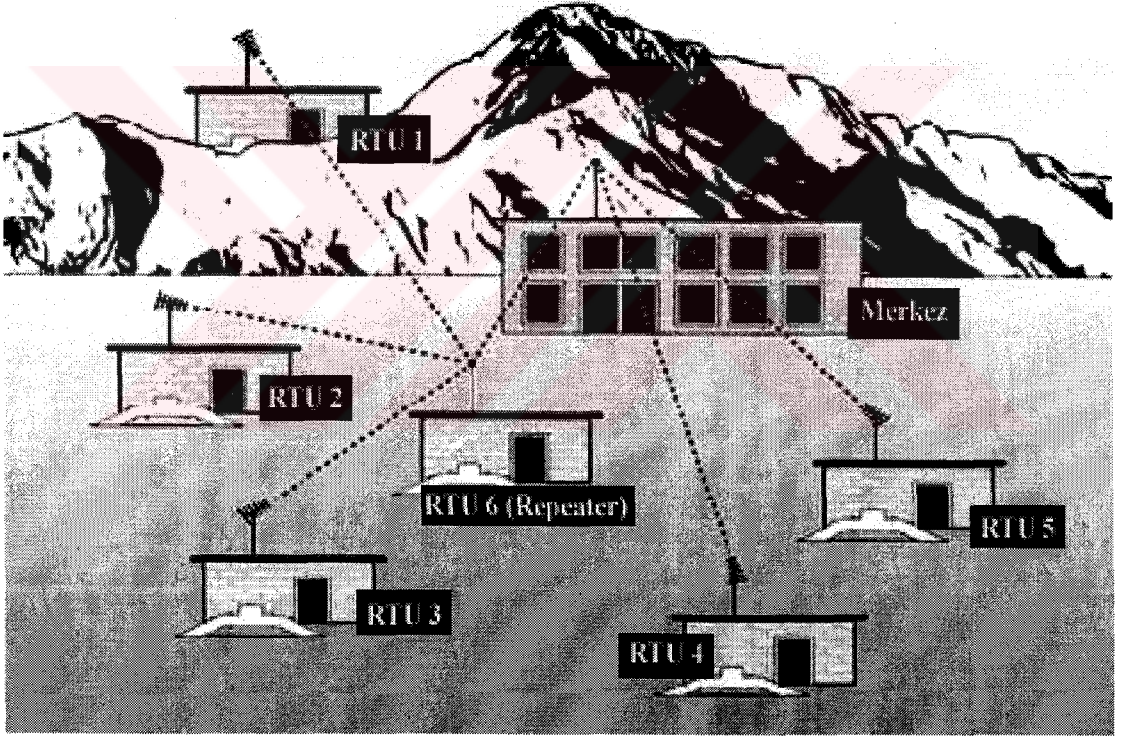
Tablo 3.2. RTU’nun kullanıldığı endüstriyel uygulamalar.

Endüstri	Uygulama
Su/Su Şebekesi	Şehir şebekesi su dağıtımı Filtre denetim sistemleri Atık su toplama sistemleri Pompa istasyonları Su kuyusu ve depo durumlarının denetimi
Elektrik Hizmetleri	Alt birimlerin denetimi ve otomasyonu Ölçüm İletim
Ulaşım	Raylı ulaşım sistemleri Hava yolu Trafik denetimi Aydınlatma denetimi
Bina Otomasyonu	Güvenlik sistemleri Aydınlatma Yangına karşı koruma sistemleri
Endüstriyel	Fabrika

3.5.2.2 RTU’nun iletişim ve ağ yapısı

RTU’ların, ADM, diğer RTU ve IED’ler ile haberleşebilmesi için entegre iletişim arayüzleri (port) bulunmaktadır. Bu portlar, kiralık hatlar, dial-up modem, radyo verici sistemleri, fiber optik hatlar, uydu iletişim sistemlerini ve Modbus, Profibus,

Fieldbus ve diğer birçok protokolü, haberleşme amaçlı olarak kullanır. RTU'da bulunan portlar, her biri farklı iletişim ortamlarını ve tekniklerini kullanabilmek için ayrı ayrı ayarlanabilir. Örneğin; bir numaralı port ADM ile haberleşme amacıyla dial-up modem ile haberleşmeye ayarlanabilirken, iki numaralı port RTU'nun bulunduğu ortamda bir denetim işlemini gerçekleştiren IED ile haberleşmek için Modbus protokolüne ayarlanabilir [9]. RTU'lardan meydana gelen bir ağ yapısında, bir RTU üzerinden diğer RTU'ya kolayca erişilebilerek bu RTU programı yeniden yüklenebilir veya gerekli değişiklikler yapılabilir. Bunun yanı sıra, ana terminal birimi, bir RTU üzerinden diğer RTU'ya erişerek gerekli görüntüleme ve denetim işlemlerini herhangi bir ilave ekipman olmadan gerçekleştirebilir. Şekil 3.8'de RTU'lar ile gerçekleştirilmiş bir SCADA uygulamasının yapısı görülmektedir.



Şekil 3.8. RTU tabanlı SCADA uygulaması.

RTU'lar ile gerçekleştirilen bu uygulamada RTU 6, RTU 1, RTU 2 ve RTU 3'ün merkez ile iletişim kurmasını sağlayan RTU'dur [19]. UUB'lerde, RTU ve PLC'lerden farklı bazı cihazlarda kullanılmaktadır. Bunlar;

i) Veri dönüştürme birimleri, SCADA sistemlerinin özel gereksinimlerini optimize eder, RTU ve PLC'nin her ikisini de destekler. İlave (Plug-in I/O) giriş çıkış birimi

ve RS-232/RS-485 birimleri monte edilebilir. Bu cihazlar RTU, PLC ve iletişim cihazlarının tam bir birleşimi ile meydana gelmiş melez cihazlardır. Uzaktaki sistemler ile radyo frekansı, fiber optik, kablolu iletişim ve diğer iletişim kanallarını kullanabilme olanaklarını sağlar [20].

ii) Web kamera (Web CAM), SCADA sistemleri ile entegre çalışabilir, cep telefonları veya PDA'da ki SCADA uygulamalarına kablosuz erişim protokollerini (Net6 Wireless Web) kullanarak erişebilir. Genellikle büyük binalarda güvenlik amacıyla veya görsel görüntülemenin istendiği yerlerde kullanılır [21].

iii) Hücresel modem (circular modem), CDPM, CDMA, GSM ve diğer hücresel iletişim teknolojilerini kullanarak RTU'lara erişim olanağı sağlar [22].

iv) Endüstriyel Ethernet, SCADA sistemlerinde yüksek seviyeli denetimin ve gerçek zamanlı denetimin gerçekleştirilmesinde oldukça başarılı olarak kullanılmaktadır. Endüstriyel Ethernet'in otomasyon sistemlerinde yaygın olarak kullanılmasının en önemli sebebi, kurulumunun kolay olması böylece yüksek işçilik maliyetinin ortadan kalkmasıdır. Ayrıca, birçok saha cihazları ile çalışabilir ve oldukça yüksek iletişim hızı sağlar [23,24].

v) Ethernet Multi-drop Modemler, bu tip cihazların kullanılması ile LAN üzerindeki farklı noktalarda bulunan cihazların herhangi bir noktadan (LAN, WAN) denetim edilebilme imkanını sağlamaktadır [25].

vi) Bus dönüştürücü Ethernet sayesinde modbus, profibus ve bunun gibi çeşitli bus sistemleri ile çalışabilen cihazlar (PLC, RTU v.b.) TCP/IP protokolüne uygun hale getirilir. Genelde bu cihazlarda kullanıcının ayarlayabileceği seri portlar mevcuttur. Kullanıcı bu portları gereksinimine göre RS 323 RS 422 veya RS 485 olarak ayarlayabilir [25].

vii) Otomatik alarm denetleyici, bu cihaz sayesinde denetlenen sistem daha önceden ayarlanan alarm koşulları gerçekleştirdiğinde cihaz denetim çıkışı, siren, cep telefonuna SMS gönderebilme yada isteğe bağlı olarak bir PLC gibi davranıp gerekli müdahaleler yapabilmektedir [26].

viii) Süreç denetleyici, süreç denetim cihazı ile basit bir döngüsel denetimden çok karmaşık yüzlerce I/O içeren sistemlerin denetimi gerçekleştirebilmektedir.

Bu tip cihazlar klasik bir PLC'nin esnekliğine ve bir PC'nin işlem yapabilme yeteneğine sahiptir. Bu tip cihazların programlanmasında C, C++, Ladder Logic, Flow Diagram, Function Block ve bunun gibi birçok PC ve PLC programlama dilleri kullanılabilir [27].

ix) Uydu veri modemleri ve sistem setleri ile kablosuz veri iletişimi sağlanmaktadır. Uydu iletişim sistemleri oldukça geniş bir alana yayılmış ve oldukça yüksek maliyetli SCADA sistemlerinde, veri sürekliliğinin kesintisiz sağlanması gereken sistemlerde kullanılmaktadır. Bu tip sistemlerin en büyük avantajı, telefon hatlarının meşgul olması, yerel elektrik şebekesinin arızalanması, özel fiber optik kablolu iletişim sistemlerinde ilk yatırım maliyetinin oldukça yüksek olması ve bakım ve fazla sayıda çalışan gereksiniminin olması ve bunun yanı sıra deprem gibi doğal afetlerde veri iletişiminin kesintisiz sağlanmasıdır. Ayrıca uydu kanal tahsisinin oldukça pahalı olması gibi dezavantajından dolayı sürekli veri iletişiminin sağlamak için kullanılması yerine daha olağan dışı durumlarda devreye alınır [28].

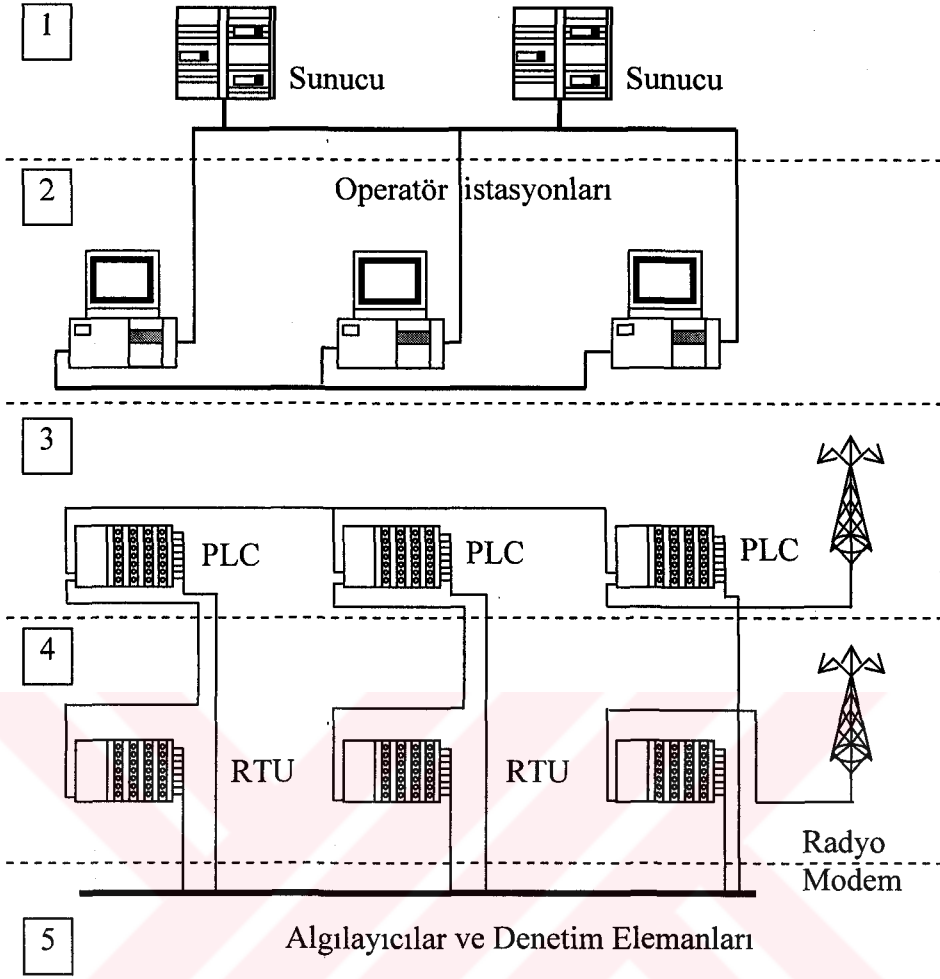
3.6. SCADA Sistemlerinin Gruplandırılması

Genel olarak SCADA sistemleri ikiye ayrılabilir. Bunlar;

- i) Klasik SCADA sistemleri,
- ii) İnternet tabanlı SCADA sistemleri.

3.6.1 Klasik SCADA sistemleri

Klasik SCADA sistemleri iki temel yapıdan günümüzde en yaygın kullanıma sahip olanıdır. En önemli avantajı aynı zamanda en önemli dezavantajıdır. Bu da klasik SCADA sistemlerinin kısıtlı sayıda kullanıcı ve belirli haberleşme noktalarından hizmet vermesidir. Örneğin sistemin bir başka ülkeden görüntülenmesi sadece uydu bağlantısı kurularak mümkün olmaktadır. Bu da oldukça maliyetli bir işlemdir. Şekil 3.9'da klasik SCADA sisteminin yapısı görülmektedir [29-30].



Şekil 3.9. Klasik SCADA sisteminin yapısı.

Klasik SCADA sistemleri genel olarak beş ayrı katmandan oluşur. Bunlar;

- i) İlk katmanda sunucu bulunmaktadır. Şekil 3.9’da görüldüğü gibi en az iki adet sunucu bulunmaktadır. Bunlardan biri ana sunucu olarak kullanılırken, diğeri yedek sunucu olarak kullanılmaktadır.
- ii) Operatör istasyonları, genelde operatörün bilgilendirilmesi için belirli istasyonlara konumlandırılan istemci bilgisayar yada ince istemcilerden (thin client) oluşmaktadır.
- iii) Fiziksel olarak ADM’nin yakınında bulunan PLC ve diğere denetleyiciler bu katmanda kullanılmaktadır ve haberleşme ortamı olarak, genelde kablo ile gerçekleştirilen veri yolları kullanılır.
- iv) Fiziksel olarak ADM’den uzakta bulunan RTU ve diğere denetleyiciler bu katmanda bulunmaktadır ve haberleşme ortamı olarak, genelde özel kiralık hatlar

veya radyo modem ile gerçekleştirilen veri yolları ve haberleşme ortamları kullanılır.
v) Bu katman klasik SCADA sisteminin ve diğer SCADA sistemlerinin genel olarak en alt katmanıdır. Algılayıcı ve çeşitli denetim cihazları bu katmanda bulunur.

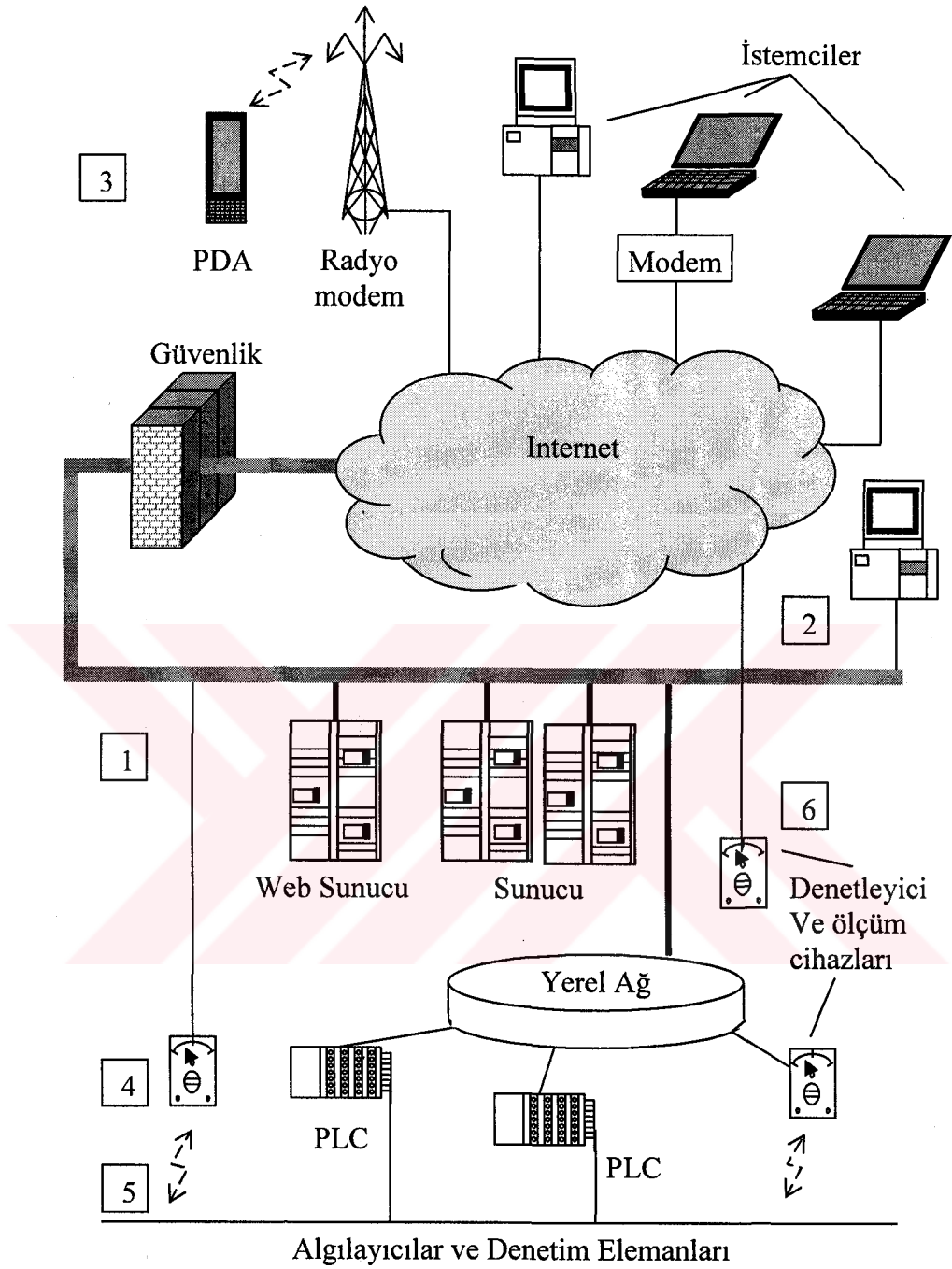
Üçüncü ve dördüncü katmanda kullanılan iletişim ortamları ve çeşitli veri yolları sonraki bölümde açıklanmaktadır.

3.6.2 İnternet tabanlı SCADA sistemleri

90'lı yılların son yarısında internet tüm dünyaya bir salgın olarak yayılmış ve oldukça fazla kullanıcı sayısına erişmiştir. Kullanıcının dünyanın herhangi bir yerindeki bilgiye kolayca ulaşabilmesi ve bu hizmetin oldukça ucuza mal olması internetin en büyük avantajlarıdır. 2000'li yılların başında SCADA yazılım üreticileri internetin bu avantajlarını göyerek, SCADA sistemlerini internete entegre çalışabilir hale getirmişlerdir. Aynı zamanda internete bağlanarak ADM ile doğrudan haberleşebilen saha cihazları üretilmiştir. Günümüz internet tabanlı SCADA sistemleri ile klasik SCADA sistemleri arasındaki en önemli farklar[31-38];

- i) İnternet entegrasyonu,
- ii) internet tabanlı cihazların sisteme entegrasyonu,
- iii) uzak operatör istasyon sayısı, isteğe bağlı olarak kolayca artırılabilirdiği gibi kolayca da azaltılabilir.
- iv) Günümüz SCADA yazılımları açık mimari yapısına sahiptir. Sahadan topladıkları verileri çeşitli stok yönetim programları ile paylaşır. Böylece yöneticilerin, sistemin bu günü ve geleceği hakkında çeşitli kararları almalarında yardımcı olur [39-48].

Şekil 3.10'da günümüzdeki internet tabanlı SCADA sisteminin yapısı görülmektedir.



Şekil 3.10. Günümüz internet tabanlı SCADA sisteminin yapısı.

Günümüz SCADA sistemleri genel olarak altı ayrı katmandan oluşur. Bunlar;

i) ilk katmanda sunucu bulunmaktadır. Şekil 3.10'da görüldüğü gibi en az üç adet sunucu bulunmaktadır. Bunlardan biri web sunucu diğeri ana sunucu olarak kullanılırken üçüncüsü yedek sunucu olarak kullanılmaktadır.

- ii) SCADA sistemlerinde dikkat edilmesi gereken en önemli nokta sistem veri güvenliğinin sağlanmasıdır. Bu genellikle ateş duvarı (fire wall) olarak adlandırılan çeşitli güvenlik yazılımlarının kullanılması ile gerçekleştirilir. Alternatif çözüm ise güvenliğin donanımsal olarak sağlanmasıdır. Özel tasarlanmış kartlar kullanarak verinin güvenliği donanımsal olarak sağlanmış olur.
- iii) Operatör istasyonları, genelde operatörün bilgilendirilmesi için belirli istasyonlara konumlandırılan istemci bilgisayar yada ince istemciden (thin client) oluşmaktadır.
- iv) Uzak operatör istasyonları, operatörün bilgilendirilmesi için ana sunucuya kullanıcı yetkisine bağlı olarak internet aracılığıyla bağlanabilen istemcilerdir. Klasik SCADA sistemlerinde operatör istasyon yeri belli ve sayısı sınırlı iken, günümüz sistemlerinde bu sayı sınırı kolaylıkla değiştirilebilir hale getirilmiştir.
- v) Fiziksel olarak ADM'nin yakınında bulunan PLC ve diğer denetleyiciler bu katmanda kullanılmaktadır. Bu katman da kullanılan PLC ve diğer denetleyiciler ile ADM arasında haberleşme ortamı olarak, genelde kablo ile kullanılarak gerçekleştirilen veri yolları kullanılır.
- vi) Fiziksel olarak ADM'den uzakta bulunan RTU, internet tabanlı cihazlar ve diğer denetleyiciler bu katmanda bulunmaktadır. Bu katman da kullanılan RTU ve diğer denetleyiciler ile ADM arasında haberleşme ortamı olarak, genelde özel kiralık hatlar veya radyo modem ile gerçekleştirilen veri yolları ve haberleşme ortamları kullanılır.
- vii) Bu katman klasik SCADA sisteminin ve diğer SCADA sistemlerinin genel olarak en alt katmanıdır. Algılayıcı ve çeşitli denetim cihazları bu katmanda bulunur.

4. İLETİŞİM SİSTEMLERİ

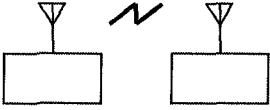
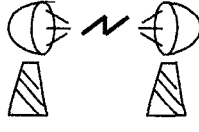
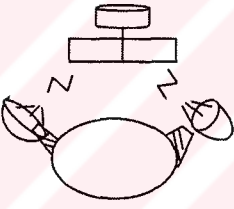
Bu bölümde SCADA sistemlerinde kullanılan, iletişim ortamları ve veri yolları incelenmektedir.

4.1. İletişim Ortamları

Orta ve büyük ölçekli SCADA sistemlerinde ADM ile UUB arasındaki iletişim, sahada kullanılan veri iletişim kabloları ile yapılamamaktadır. Bunun iki sebebi vardır. Birincisi, ADM ve UUB arasındaki mesafenin fazla olmasıdır. Diğeri ise mesafenin fazla olmasından dolayı veri tekrarlama merkezlerine ihtiyaç duyulmasıdır. Bu nedenle, kurulum ve sistemin bakım maliyeti oldukça artmaktadır. Orta ve büyük ölçekli SCADA sistemlerinde iletişim ortamları kullanılarak ADM ile UUB arasındaki veri iletişimi sağlanmaktadır. Genel olarak bu iletişim ortamları iki ana başlık altında toplanmıştır. Bunlar; kablosuz veri alışverişini sağlayan ortamlar ve telefon ve çeşitli haberleşme hatlarıdır.


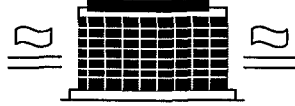
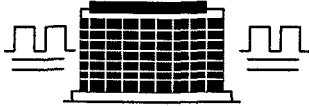
Kablosuz veri alışverişini sağlayan iletişim ortamları, VHF/UHF radyo frekansı ile iletişim, mikrodalga ile iletişim ve uydu sistemleri ile iletişimin sağlandığı ortamlardır. Mikrodalga ve uydu sistemlerinin kullanıldığı iletişim sistemlerinin, kurulum maliyetleri ve işletme giderleri oldukça fazladır. VHF/UHF radyo frekansı kullanan iletişim ortamının kurulum ve işletme giderleri uygun değerlerdedir. Bu yüzden, VHF/UHF radyo frekansı kullanan iletişim ortamı yaygın kullanım alanına sahiptir. Bu ortamın kullanımındaki en büyük problem ise radyo frekansının kullanımından önce ilgili kurumdan izin alınması gereksinimidir. Tablo 4.1'de kablosuz veri alışverişini sağlayan iletişim ortamları görülmektedir [17].

Tablo 4.1. Kablosuz veri alışverişini sağlayan iletişim ortamları.

Tip	Sembol	Tanımlama
VHF/UHF Radyo frekansı ile iletişim		VHF/UHF Radyo frekansı ile iletişimde yüksek frekanslı radyo dalgaları kullanılır. Özel antenler kullanılarak veri alışverişi sağlanır.
Mikrodalga ile iletişim		Mikrodalga ile iletişimde Giga hertz (GHz) seviyesindeki mikro dalgalar ile iletişim yapılır. Veri alışverişinde çanak antenler kullanılır ve bu antenler yüksek binalara veya kule tepelerine monte edilir.
Uydu sistemleri ile iletişim		Uydu sistemleri ile iletişimde Giga hertz (GHz) seviyesindeki sinyaller ile iletişim yapılır. Birbirleri arasında oldukça fazla mesafe bulunan birimler arasında (ülkeler arası denetim) kullanımı tercih edilmektedir.

Kablosuz veri iletişimi sağlayan ortamların dışında, varolan telefon hatları, özel kiralık hatlar (PLL: Private Liase Line) ve sayısal veri servisi (DDS: Digital Data Service) hatları kullanılarak ADM ile UUB arasındaki veri alışverişi sağlanmaktadır. Tablo 4.2’de kablolu veri alışverişini sağlayan iletişim ortamları görülmektedir.

Tablo 4.2. Kablolu veri alışverişini sağlayan iletişim ortamları.

Tip	Sembol	Tanımlama
Dialup bağlantı ile iletişim		Dialup bağlantı ile iletişimde yerel telefon hatları kullanılmaktadır.
Özel kiralık hat ile iletişim		Özel kiralık hat ile iletişim iki veya daha fazla kullanıcı 24 saat boyunca kesintisiz veri alışverişi gerçekleştirir.
Sayısal veri servisi ile iletişim		Sayısal veri servisi ile iletişimde özel kiralık hat bağlantısına benzer bir yapıya sahiptir. Farkı ise veri alışverişini sayısal ortamda gerçekleştirmesidir.

4.1.1 İletişim ortamlarının avantaj ve dezavantajları

VHF/UHF Radyo frekansı ile iletişim diğer ortamlara göre çeşitli avantajlara sahiptir. Bunlar;

- i) Coğrafi problemler sebebiyle telefon hattı ile ulaşılamayan bölgelere uzaktan erişim imkanı sağlar.
- ii) Sabit bir bağlantı ile 45 Km'lik uzak bir mesafeden veri alışveriş imkanı sağlar.
- iii) Yeni bir teknoloji olan "spread spectrum" ile çalışan cihazların kullanımı ile frekans tahsis problemi ortadan kalkmaktadır.

VHF/UHF Radyo frekansı ile iletişim, yukarıda sıralanan avantajlarına sahip olmasına rağmen çeşitli dezavantajları da bulunmaktadır. Bunlar;

- i) 20 Km'nin üstündeki mesafelerde veri tekrarlayıcıya ihtiyaç duyulmaktadır.
- ii) Kurulum maliyeti fazladır. Buna rağmen mikrodalga ve uydu sistemleri ile karşılaştırıldığında oldukça ucuza mal olmaktadır.

VHF/UHF Radyo frekansı ile iletişim sistemi kurabilmek için gerekli ekipmanlar ise; alıcı ve verici ekipman anten ve uzun mesafelerde veri tekrarlama ekipmanlarıdır.

Mikrodalga ile iletişimin diğer ortamlara göre çeşitli avantajları bulunmaktadır. Bunlar;

- i) Coğrafi olarak oldukça problemlili (tepe ve vadilere sahip) bölgelerde telefon hattı ile ulaşılamayan yerlerde kullanılır.
- ii) Sabit bir bağlantı kurulmasını sağlar. Uzun engebeli arazilerde iletişim kurulmasına imkan verir.
- iii) Sistem kiralık olmadığı için aylık servis gideri yoktur. Sadece kurulum ve bakım gideri mevcuttur.
- iv) İletişimde bekleme süresi oldukça azdır.
- v) Bir anten üzerinden birçok kanalda iletişim imkanı sağlar. Bu özelliğinden faydalanarak diğer firmalara iletişim desteği verilebilir.

Mikrodalga ile iletişim, yukarıda sıralanan avantajlarına sahip olmasına rağmen çeşitli dezavantajları da bulunmaktadır. Bunlar;

- i) Dağların içinden iletişim kurabilme imkanı yoktur.
- ii) En önemli mikrodalga iletişim frekansları ulusal iletişim komisyonu tarafından denetim altına alınmış ve belirli izinlere bağlanmıştır.
- iii) Genelde kurulum giderleri oldukça yüksektir.

Mikrodalga ile iletişim sistemi kurabilmek için gerekli ekipmanlar ise; alıcı ve verici ekipmanlar, parabolik antenler ve uzun mesafelerde kullanım ihtiyacına bağlı olarak veri tekrarlayıcı ekipmanlar.

Uydu iletişim sistemleri ile iletişimin diğer ortamlara göre çeşitli avantajları bulunmaktadır. Bunlar;

- i) Dünyanın herhangi bir yerinden UUB ile ADM arasında iletişim kurulabilir.
- ii) Sabit bir iletişim bağlantısı sağlar.

Uydu iletişim sistemleri tercih edilmeden önce UUB'nin bulunduğu yerin coğrafi özellikleri, iletişimde gönderilecek veya alınacak olan verinin boyutu ve kiralık hatlar ile aynı sistemin kurulup kurulamayacağı dikkatlice incelenmelidir. Uydu sistemleri ile veri alışverişi oldukça güvenli bir iletişim imkanı sağlamaktadır. Uydu iletişim sistemleri ile iletişimin en büyük dezavantajı oldukça yüksek kurulum ve

aylık servis gideridir. Uydu iletişim sistemi kurabilmek için gerekli ekipmanlar ise; iletişim sağlamak için gerekli uydu bağlantısı ve uydu alıcı ve vericisidir.

Dial-up bağlantı ile iletişimin diğer ortamlara göre bazı avantajları bulunmaktadır.

Bunlar;

- i) Kısa mesafeli ve sık sık veri alışveriş gerektirmeyen uygulamalarda oldukça ekonomik bir çözümdür.
- ii) Merkezi bir noktadan diğer uç birimlere dial-up bağlantı ile iletişim kurma olanağı sağlar.
- iii) Telefon firmalarına aylık bir servis ücreti ödenir bu ücret, mesafe ve UUB sayısı ile orantılıdır.
- iv) Telefon hatları "half-duplex ve full-duplex" modemler ile iletişim olanağı sağlar.

Dial-up bağlantı ile iletişim, yukarıda sıralanan avantajlarına sahip olmasına rağmen çeşitli dezavantajları da bulunmaktadır. Bunlar;

- i) Uzun mesafelerde veri alışverişi oldukça yüksek giderler meydana getirebilir.
- ii) Telefon hatlarının bulunmadığı kısımlarda kullanımı imkansızdır.
- iii) Her bağlantı noktası sisteme bağlanmak ve veri alışverişi için fazladan bir zamana ihtiyaç duyar.
- iv) Verinin güvenliği iletişim için kullanılan telefon firmasına bağlıdır.

Dial-up bağlantı ile iletişim sistemi kurabilmek için gerekli ekipmanlar ise; Telefon ve telgraf modemleridir.

Özel kiralık hat ile iletişimin diğer ortamlara göre çeşitli avantajları bulunmaktadır.

Bunlar;

- i) Ana istasyon ile sabit bir bağlantı, bu bağlantıda yüksek miktarda veri alışverişi gerektiren durumlarda iyi bir çözümdür.
- ii) Dial-up bağlantı ile taşınması güç miktarda veri özel kiralık hat (PLL) bağlantısı ile kolayca taşınabilir.
- iii) Kablosuz iletişim ortamları ile karşılaştırıldığında aylık servis gideri ve kurulum masrafları oldukça azdır.

Özel kiralık hat ile iletişimin en büyük dezavantajı ise, kiralık hatlar ile herhangi bir noktada bulunan UUB'ye erişim imkanının olmamasıdır. Çünkü özel kiralık hat bağlantısı ile iletişim telefon firmasının altyapı desteği ile sınırlıdır. Gerekli ekipman ise Bell veya CCITT (Consultive Committee for International Telephone Telegraph) modemlerdir.

Sayısal veri servisi ile iletişim diğer ortamlara göre çeşitli avantajlara sahiptir. Bunlar;

- i) Sayısal veri servisi ile iletişim yüksek miktarda verinin alışveriş halinde olduğu SCADA sistemlerinde en uygun fiyat ile verinin taşınmasını sağlar.
- ii) Sabit bir aylık servis hizmet gideri vardır. Bu gider iletişim ortamının hızına bağlı olarak değişir.

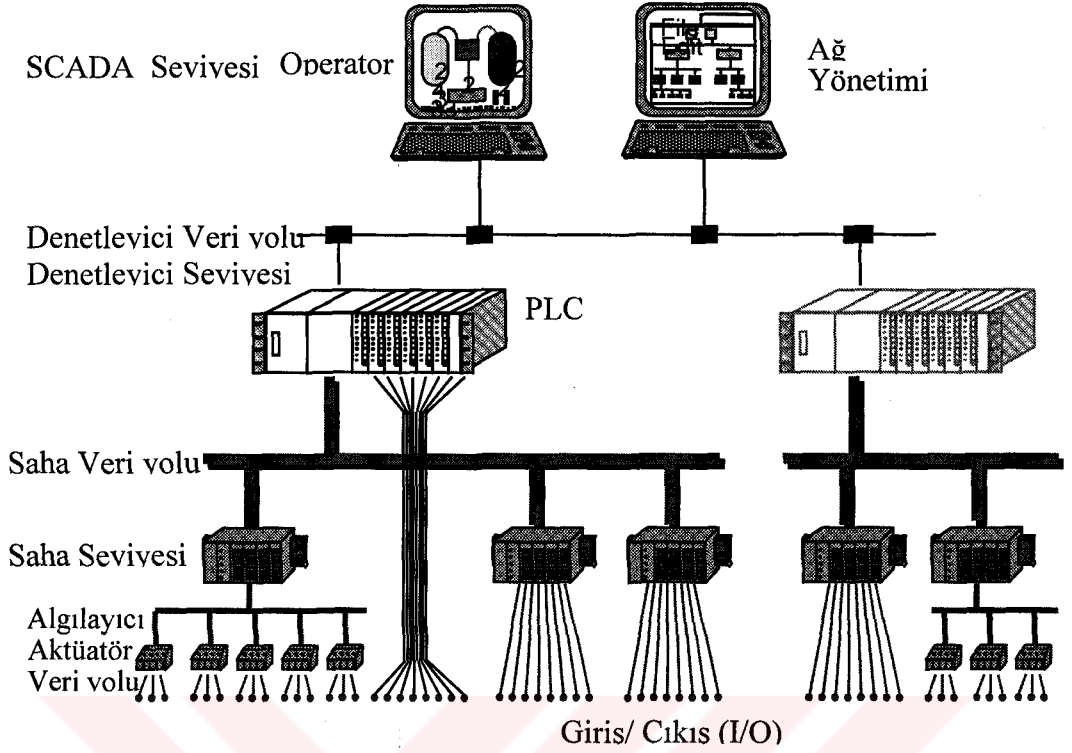
En büyük dezavantajı aylık ödenen servis gideridir. Taşınan verinin güvenliği iletişim hizmetini sağlayan firma tarafından sağlanır.

4.2. Veri Yolu Sistemleri

SCADA sistemlerinde, çeşitli kademelerde ADM ile veri alışverişinde bulunan cihazlar ve bilgisayarlar bulunmaktadır. Bir sistemin SCADA otomasyonu üç ayrı seviyede incelenmiştir. Bunlar, SCADA seviyesi, denetleyici seviyesi ve saha seviyesidir. Bu üç seviyede bulunan çeşitli cihaz ve ekipmanlar aşağıda sıralanmıştır.

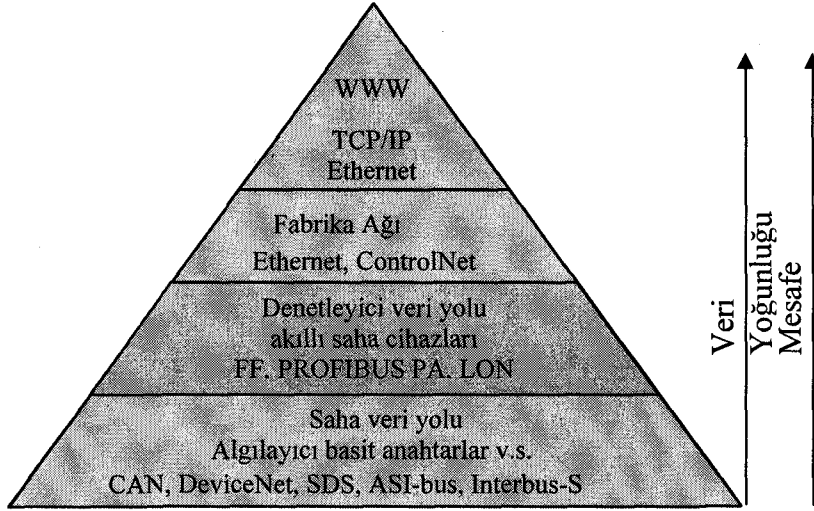
- i) Saha seviyesinde, algılayıcılar, aktüatörler, röleler ve çeşitli sinyal lambaları kullanılmaktadır.
- ii) Denetleyici seviyesinde, PLC, DCS ve IED'ler kullanılmaktadır.
- iii) SCADA seviyesinde ise, SCADA otomasyon ağına bağlı operatör bilgisayarları, SCADA sunucusu ve yedekleme üniteleri bulunmaktadır.

Şekil 4.1'de SCADA sistemlerindeki veri iletişim hiyerarşisi görülmektedir [17].



Şekil 4.1. SCADA sistemlerindeki veri iletişimi hiyerarşisi.

Şekil 4.2’de veri yolunun kullanıldıkları seviyeye göre sınıflandırılması verilmekte ve bu sınıflardaki mesafe ve veri yoğunluğunun değişimi gösterilmektedir [49]. Şekil 4.2’de üçgenin üst tarafına çıkıldıkça, cihazlar arası haberleşmede kullanılan veri yoğunluğu ve cihazlar arası mesafe artmaktadır. Saha seviyesindeki ekipmanlar ile gerçekleştirilen iletişim veri yoğunluğu en düşük olan iletişimdir. Aynı zamanda bu seviyedeki cihazlar arası mesafe diğer seviyedeki cihazlara nazaran daha düşük olmak zorundadır.



Şekil 4.2. Veri yolunun kullanıldığı seviyeye göre sınıflandırılması.

4.2.1 Veri yolu seçim ölçütleri

SCADA sistemi kurulmadan önce, sistemin ihtiyaçları çok iyi bir şekilde tespit edilmelidir. Bu ihtiyaçlar (sistemin hızı, haberleşilecek istasyon sayısı, UUB ile ADM arasındaki mesafe) ve gelecekte gerçekleştirilmesi olası endüstriyel sistemin genişletilme çalışmaları dikkate alınarak veri yolunun seçilmesi gerekmektedir. Endüstriyel sistem kurulumunda, sahadaki cihazların denetimi ve bu cihazların, kendi aralarında ve merkezi terminal birimi ile gerçekleştirdikleri haberleşmede kullanılan veri yollarının seçimi önemlidir. Veri yolları seçiminde dikkat edilmesi gereken ölçütler aşağıda verilmiştir.

- i) Endüstriyel sistemin fiziksel boyutları,
- ii) Endüstriyel sistemin sahadaki dağılımı,
- iii) Kablolamanın azaltılması,
- iv) Endüstriyel sistemin modülerliğini artırması,
- v) Hata gerçekleştiğinde arıza yerinin tespit ve bakım kolaylığını sağlaması,
- vi) Sistemin yeni cihazların entegrasyonuna açık olması.

Yukarıda maddeler halinde verilen seçim ölçütlerinin SCADA sisteminin tasarımı aşamasında dikkatlice uygulanması halinde çeşitli faydalar sağlanmaktadır. Bunlar;

- i) Kablolamanın azalması ile kurulum maliyeti azalmaktadır. Ayrıca kurulumda daha az sayıda kablo ile iletişimin sağlanabilmesi sebebiyle, saha seviyesinde veya diğer seviyelerde bulunan kablolarda meydana gelebilecek herhangi bir arıza daha kısa zamanda giderilebilmektedir. Böylece, arıza sebebiyle çalışması beklemeye alınan sistemin üretim kayıpları minimuma indirilmektedir.
- ii) Seçilen veri yoluna bağlı olarak, sistemin kurulumu en kısa zamanda tamamlanarak çalışır duruma gelmesi sağlamaktadır.
- iii) Endüstriyel bir işletme veya hizmet sistemleri (elektrik üretim, iletim veya dağıtım, su iletim veya dağıtım v.b.) belirli zaman aralıklarıyla bakım ve sistemi oluşturan ekipmanlarda modernizasyon gereklidir. Bu yüzden kullanılan veri yolunun sisteme ilave edilecek ekipmanların ihtiyacını karşılayabilecek şekilde olması gerekmektedir.

4.2.2 Veri yollarının fiziksel boyutları

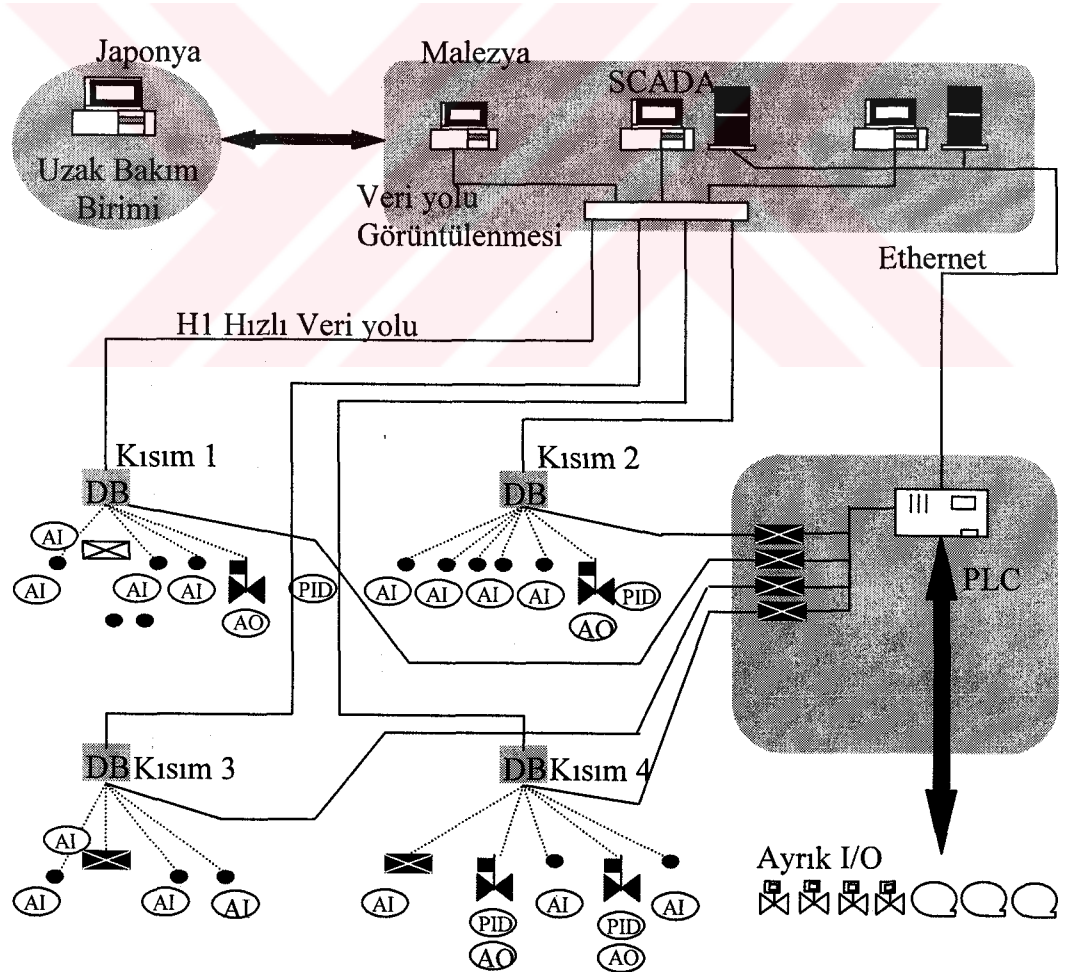
Veri yollarının fiziksel boyutları kullanıldığı sisteme göre değişmektedir. Veri yollarının fiziksel boyutları Tablo 4.3'de sunulmaktadır [49].

Tablo 4.3. Veri yollarının uygulandığı sistemlerin fiziksel boyutları.

Fiziksel Boyut	Kullanım Alanları
1 km .. 1000 km	İletim ve Dağıtım Sistemleri Enerji dağıtım ve iletim sistemlerinde, su, gaz ve petrol dağıtım sistemlerinde
1 km .. 5 km	Enerji Üretim Tesisleri Hidroelektrik, termik, nükleer, rüzgar ve güneş enerjisi santrallerinde
50 m .. 3 km	Endüstriyel Sistemler Demir çelik üretim tesislerinde, kimya, kağıt, yiyecek ve bunların dışındaki birçok üretim tesislerinde
500m .. 2 km	Bina Otomasyonu Bina enerji denetim sistemlerinde, yangın ve güvenlik bildirim sistemlerinde
1 m .. 1 km	Üretim Robotları Esnek üretim hücrelerinde ve robotlarda
1 m .. 800 m	Taşıtlar Lokomotif, tren, tramvay, uçak ve uzay araçları, taksi, kamyon ve diğer taşıtlarda

Tablo 4.3’de görüldüğü gibi uygulama alanlarına bağlı olarak sistemin fiziksel boyutları değişim göstermektedir. Sistemin fiziksel boyutuna bağlı olarak da kullanılan veri yolu değişim göstermektedir. Aşağıda veri yollarının kullanıldığı iki örnek verilmiştir. Bu örneklerden ilki en büyük fiziksel boyutlandırmaya ait olan sistemlerden su dağıtım sisteminin otomasyonu diğeri ise, en küçük fiziksel boyutlandırmaya sahip olan sistemlerden lokomotif denetim sistemidir.

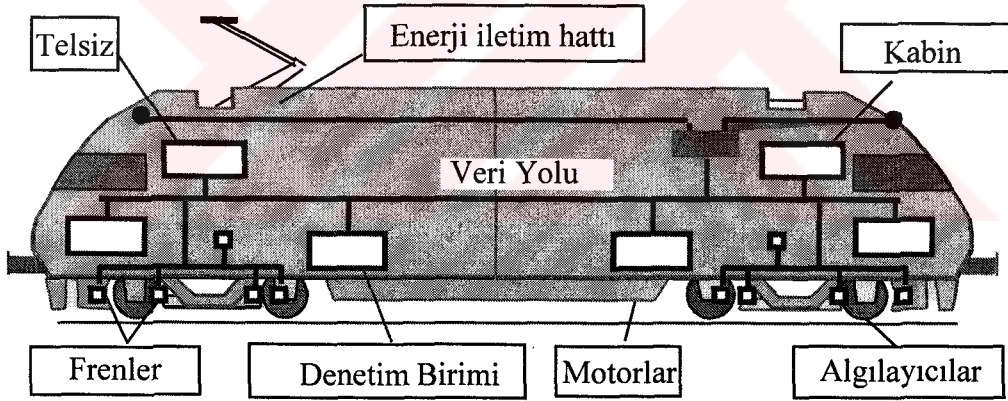
Şekil 4.3’de Japon Kaneka şirketi tarafından Malezya için hazırlanmış olan su dağıtım sisteminin otomasyonu görülmektedir [49]. Tasarlanan bu sistem dört ayrı seviyeden oluşmaktadır. Saha seviyesinde çeşitli analog giriş ve çıkış birimleri bulunmaktadır. Bu birimler, çeşitli vana ve pompaları denetlemek, su debisi ve miktarı hakkında ADM’yi bilgilendirmek amacıyla kullanılır.



Şekil 4.3. Su dağıtım sisteminin otomasyonu.

Denetleyici seviyesinde ise PLC ve IED kullanılmaktadır. PLC'ler sahadaki pompa ve vanaları denetlemek için kullanılmakta ve ADM ile iletişimde ise ethernet kullanılmaktadır. IED'ler ise DB (denetim birimi)'lerde vana denetiminde kullanılmaktadır. DB'lerin ADM ile iletişimde, Sinec H1 veri yolu kullanılmaktadır. SCADA seviyesinde, sunucu ile diğer istemciler arasında veri alışverişinde ethernet kullanılmaktadır. Japonya ve Malezya arasında veri alışverişinde İnternet kullanılmaktadır.

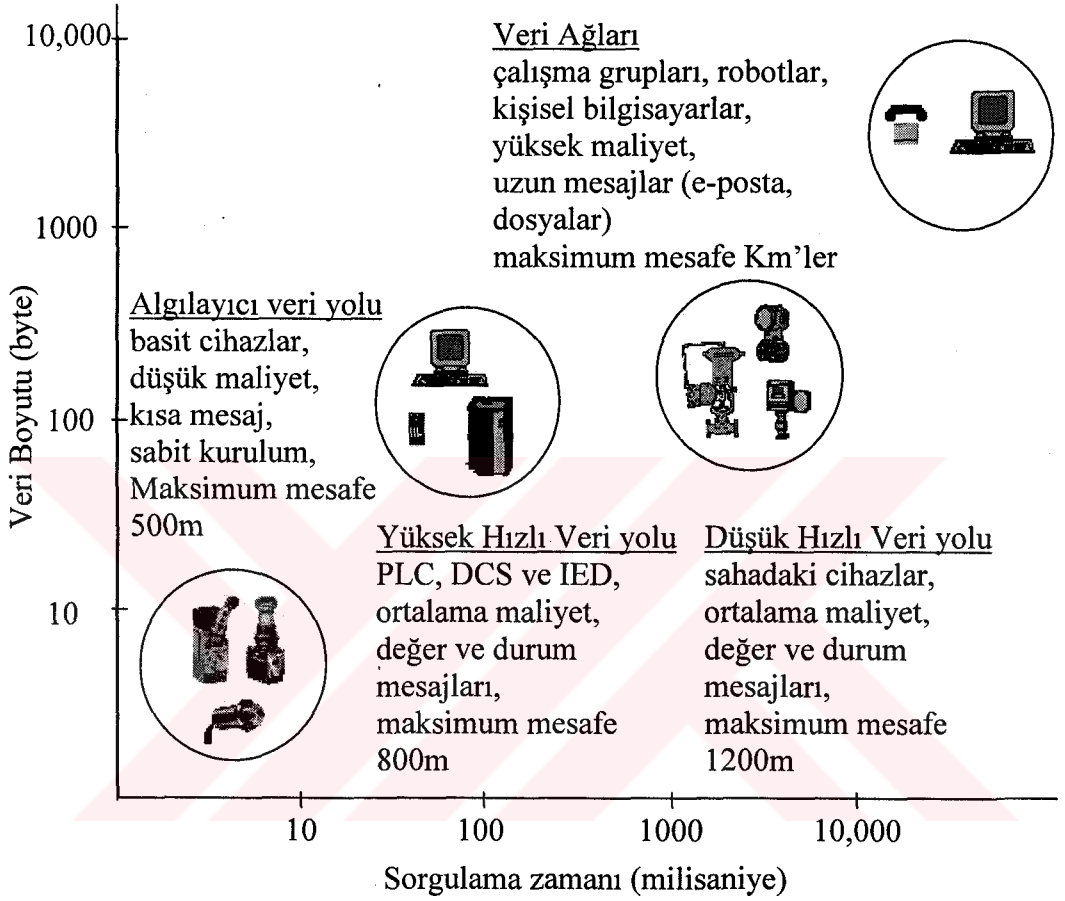
Şekil 4.4'de lokomotif denetim sisteminde veri yollarının kullanımı görülmektedir. Lokomotiflerde ve diğer taşıtlarda kullanılan veri yolları kısa mesafe iletişimde kullanılan en yüksek hızlı veri yoludur. Çünkü, lokomotifte algılayıcılar tarafından ölçülen değerler en kısa zamanda ADM'ye iletilmeli ve ADM'de gerekli denetim değerlerini üretip lokomotifi en kısa zamanda denetlemelidir.



Şekil 4.4. Lokomotif denetim sisteminde veri yollarının kullanımı.

Kullanıldığı sisteme göre veri yollarının veri gönderme kapasiteleri ve iletişim hızları değişmektedir. Algılayıcılardan veri toplamak için kullanılan veri yolları en düşük iletişim hızına ve veri gönderme kapasitesine sahip olan veri yollarıdır. Kurulum maliyeti düşük, basit cihazlarla çalışabilen ve kısa mesafede veri iletişimini destekleyen bir yapıya sahiptir. Düşük hızlı veri yolları, sahada var olan cihazlar ile iletişim sağlamak için kullanılır. Düşük hızlı veri yollarının maliyeti ortalama bir değerdedir ve 1200 metreye kadar tekrarlayıcı kullanılmadan iletişim desteği sağlar.

Yüksek hızlı veri yolları PLC, DCS ve IED'ler ile ADM arasında iletişim sağlamak için kullanılır. Ortalama bir kurulum maliyetine sahiptir. Şekil 4.5'de veri yollarının veri gönderme kapasitelerine göre sınıflandırılması gösterilmiştir.

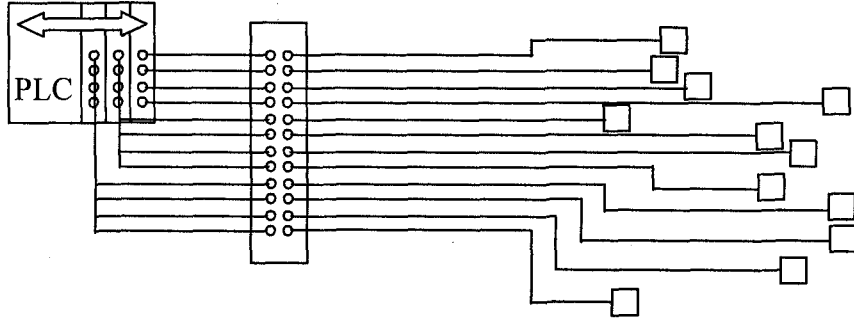


Şekil 4.5. Veri yollarının veri gönderme kapasitelerine göre sınıflandırılması.

4.2.3 Veri yollarının fiziksel yapıları

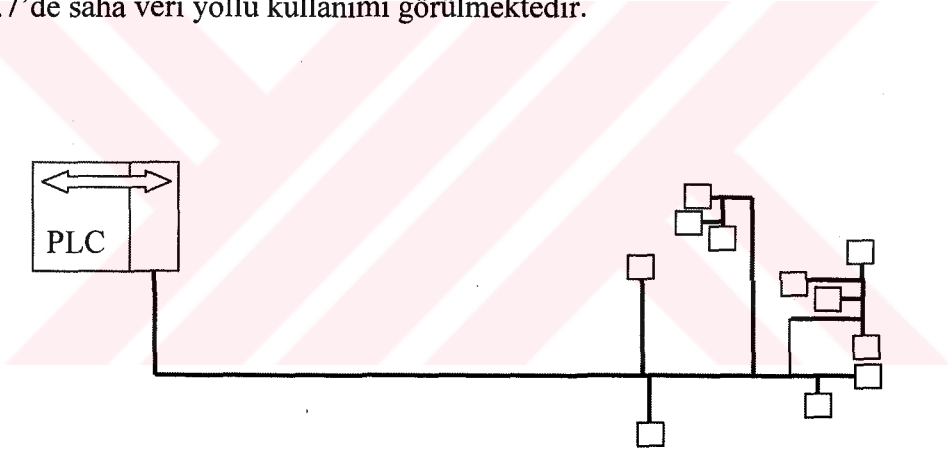
Veri yolları, birebir bağlantı, yıldız, tek hat, dal ve ağaç veri yolu fiziksel yapıları olmak üzere beş farklı fiziksel yapıdan oluşmaktadır. Şekil 4.6'da birebir bağlantı şekli görülmektedir. Birebir bağlantı fiziksel yapısı, saha veri yollarının ekonomik olmadığı zamanlarda yaygın olarak kullanılmış fakat saha veri yollarının uygulama maliyetleri azaldıkça terk edilmiş bir tekniktir. Birebir bağlantı, bir ana denetleyici ve uydu cihazlardan oluşmaktadır. Ana denetleyici ile her uydu cihaz arasındaki bağlantı bağımsızdır. Bunun sonucunda kablolama maliyetleri artar. Ayrıca iletişim

kablolarının oldukça fazla olmasından dolayı meydana gelebilecek arızaların giderilme süresi oldukça artmaktadır.



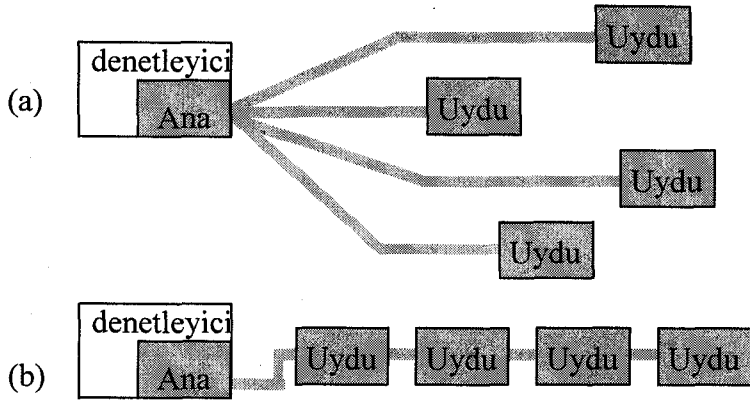
Şekil 4.6. Birebir bağlantı.

Şekil 4.7’de saha veri yolu kullanımı görülmektedir.



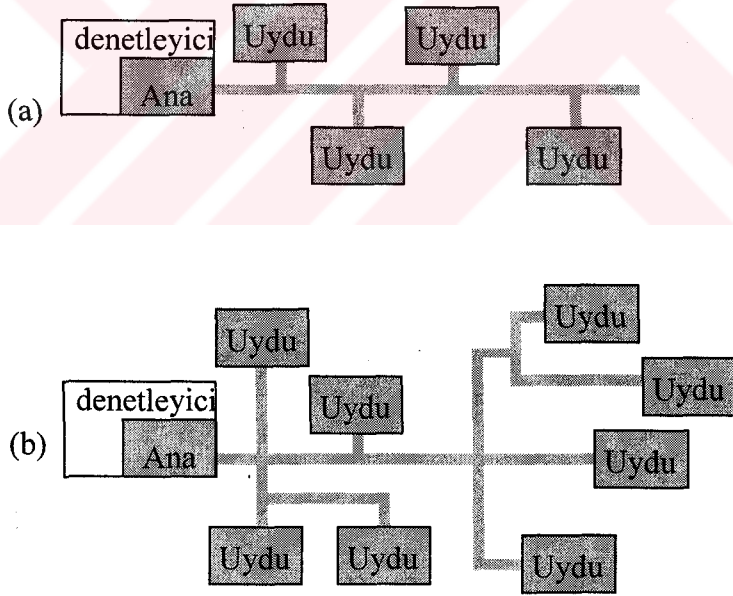
Şekil 4.7. Saha veri yollarının kullanılması.

Şekil 4.7’den de anlaşılacağı üzere sahadaki her cihazın ayrı ayrı ana denetleyiciye bağlanarak iletişimin gerçekleştirilmesi yerine, sahadaki belirli noktalara kurulu iletişim araçları vasıtası ile gerçekleştirilir. Şekil 4.8’de saha veri yollarının farklı fiziksel yapıları görülmektedir.



Şekil 4.8. a) Yıldız veri yolu fiziksel yapı,
b) Tek hat veri yolu fiziksel yapı.

Yıldız veri yolu ve tek hat veri yolu en yaygın olarak kullanılan veri yolu fiziksel yapılarıdır. Şekil 4.9'da dal ve ağaç fiziksel yapısındaki veri yolları görülmektedir.



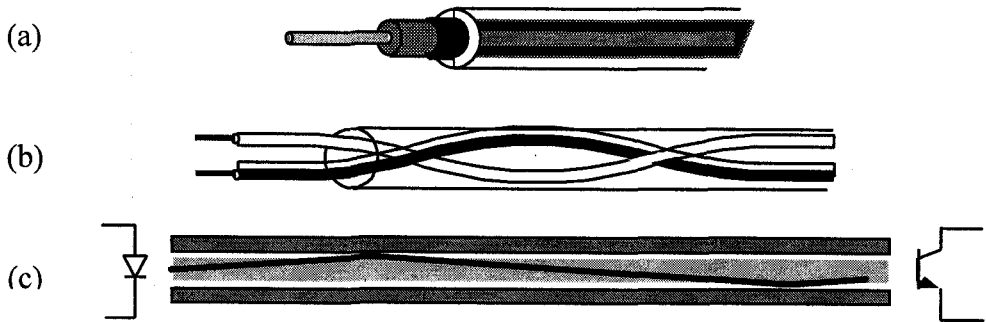
Şekil 4.9. a) Dal fiziksel yapısındaki veri yolları,
b) Ağaç fiziksel yapısındaki veri yolları.

Dal ve ağaç fiziksel yapısında fazla miktarda dağıtım istasyonu kullanılması nedeniyle kurulum maliyeti artar. Buna rağmen yıldız ve tek hat fiziksel yapısına

göre daha üstün bir veri sürekliliği sağlama kapasitesine sahiptir. Bu iki fiziksel yapıda her uydu istasyon hattına ait veri yolunda meydana gelebilecek arızada sadece o uydu istasyon devre dışı kalır. Diğer uydu istasyonlar ile ana istasyon arasındaki veri iletişimi sürdürülebilir.

Farklı fiziksel yapılarda çeşitli kablolar kullanılmaktadır. Bu kablo çeşitleri, kullanılan veri yolunun iletişim hızı, çevresel etkenler (manyetik gürültü, ısı ve nem), iletişim mesafesi ve maliyet etkenleri göz önüne alınarak seçilir. Ekranlı kablo, saha veri yolunun iletişim hızı 10 Mhz ile 100 MHz arası olduğu durumlarda ve düşük kayıp gerektiren veri yollarında uygun bir seçimdir. Kullanılan kablo ekran ile koruma altında olmasından dolayı, manyetik gürültü ve çevreden gelebilecek çeşitli darbelere karşı diğer kablo çeşitlerine nazaran daha dayanıklıdır. Kullanılan ekranın bu avantajlarına rağmen en büyük dezavantajı kablonun esnekliğini azaltmasıdır. Ekransız çift damarlı kablo ise, saha veri yolunun iletişim hızı 10 Mhz ile 100 MHz arası olduğu durumlarda ve ortalama hizmet mesafelerinde oldukça uygun bir seçimdir. Ekranlı kablonun tersine oldukça esnek olup sahadaki fiziksel yapıya kolayca uygulanabilir.

Veri yollarının kurulumunda genelde ekranlı kablo, ekransız çift damarlı kablo ve fiber kablo kullanılmaktadır. Şekil 4.10'da ve yollarında kullanılan kablo çeşitleri görülmektedir.

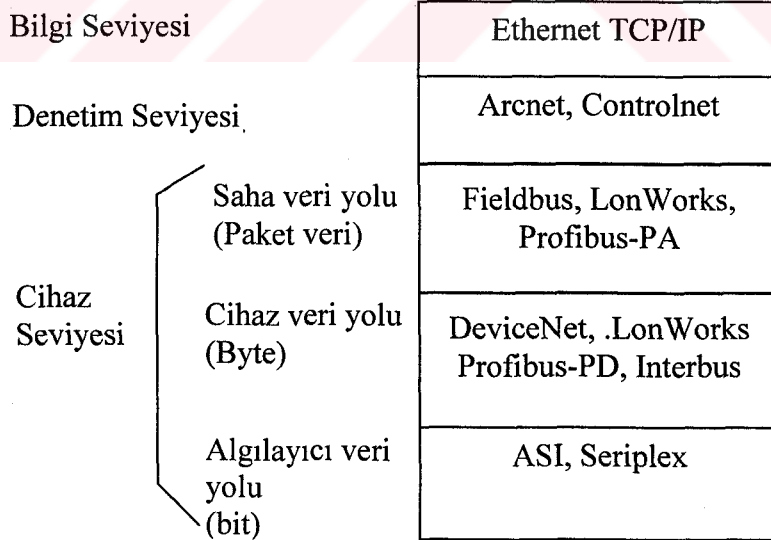


Şekil 4.10. a) Ekranlı kablo,
b) Ekransız çift damarlı kablo,
c) Fiber kablo.

Veri iletişim hızının çok yüksek (400 Mhz) olduğu veri yollarında, en iyi çözüm fiber optik kablodur. Manyetik gürültüden etkilenmemesi, kablonun oldukça hafif olması ve çok yüksek hızlarda iletişim sağlaması gibi avantajları vardır. Buna rağmen kurulum maliyetinin çok yüksek olması sebebiyle zorunlu olmadığı durumlarda kullanımı tercih edilmez.

4.3. Veri Yollarının İncelenmesi

Veri yollarının kullanıldığı seviyelere göre sınıflandırılması Şekil 4.11'de görülmektedir [50]. Veri yollarının sınıflandırılmasında en alt noktada algılayıcı veri yolu bulunmaktadır. Algılayıcı veri yolu, sahadan alınan bit tipindeki (on/off) verilerin denetim merkezine iletiminde kullanılır. Hızlı ve etkili bir yapıya sahiptir. Fakat sadece basit makinelerin denetiminde kullanılır. ASI (Actuator Sensor Interface) Avrupa'da, Amerika'da ise Seriplex'en en yaygın kullanım alanına sahip algılayıcı veri yoludur.



Şekil 4.11. Veri yollarının kullanıldığı seviyelere göre sınıflandırılması.

Cihaz veri yolları ise, daha büyük cihazların denetlemesi ve sahadan alınan analog veya sayısal verilerin denetim merkezine iletiminde kullanılır. DeviceNet, programlanabilir denetleyiciler ile iyi uyum sağlayabilen bir veri yoludur. Buna rağmen, DeviceNet, en fazla atmış dört farklı nokta üzerinde çalışabilir[50]. LONWork veri yolu oldukça büyük mesafeler arasında veri iletişimini sağlayabilir. Aynı zamanda binlerce nokta üzerinden veri alıp gönderebilme yeteneğine sahiptir. Bütün bu üstünlüklerinin yanı sıra, oldukça karmaşık bir yapıya sahip olması ve veri iletişim hızının düşük olması gibi dezavantajları da bulunmaktadır.

Saha veri yolları ise, sahada bulunan birçok farklı özelliklerdeki (farklı iletişim hızı ve farklı karmaşık yapıyı içeren cihazlar) ve farklı mesafelerde bulunan cihazların iletişiminde kullanılır. Denetim seviyesi veri yollarına en iyi örnek, Allen-Bradley firması tarafından geliştirilmiş olan ControlNet'dir.

En üst seviyede yer alan bilgi seviyesindeki veri yolları, Internet ve Intranet bağlantısını TCP/IP standardı ile sağlar. Son yıllarda Ethernet, endüstriyel dünya tarafından büyük ölçüde kabullenilmiştir. Tablo 4.4'de en yaygın kullanıma sahip veri yolları görülmektedir.

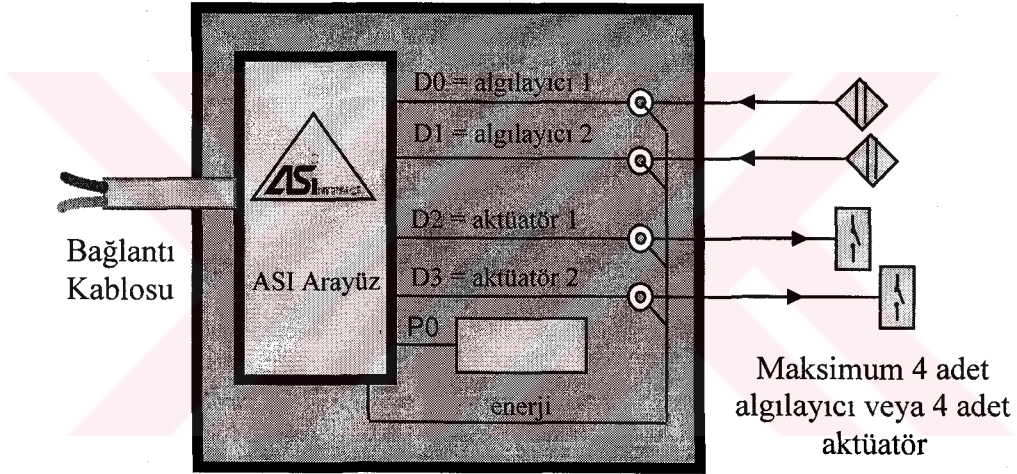
Tablo 4.4. En yaygın kullanılan veri yolları.

Veri yolu	Pazar Payı	Kullanılan Alan	Destekleyen Firma
CANs	25%	Otomotiv, süreç denetimi	CiA, OVDA, Honeywell
Profibus	26%	Süreç denetimi	Siemens, ABB
LON	6%	Bina sistemleri	Echelon, ABB
Ethernet	50%	Fabrika veriyolu	alle
Interbus	7%	Üretim	Phoenix Contact
Fieldbus	7%	Kimya endüstrisi	Fisher-Rosemount, ABB
ASI	9%	Bina sistemleri	Siemens
Modbus	22%	Noktadan noktaya	Birçok kuruluş
ControlNet	14%	Fabrika veriyolu	Rockwell

Tabloda verilen veri yolları birçok firma tarafından desteklenmekle birlikte birkaç büyük destekleyici firmaya sahiptir. Siemens ve ABB, veri yollarını destekleyen en büyük iki firmadır [50,51].

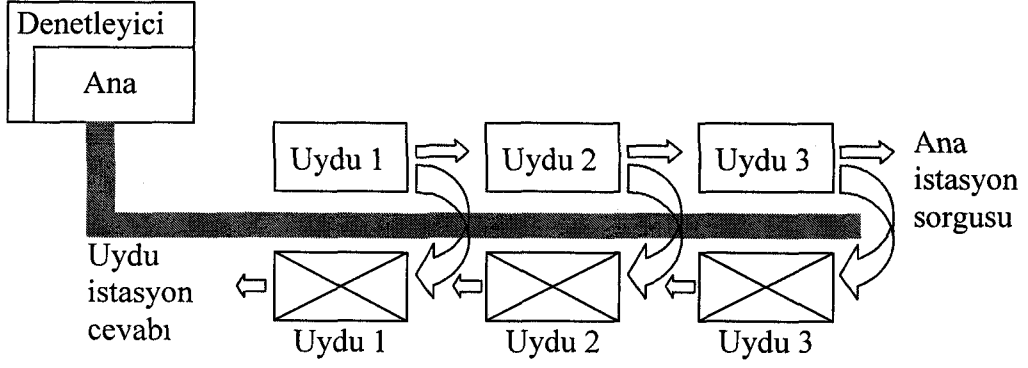
4.3.1. ASI veri yolu

ASI, algılayıcı temelli arayüz (Actuator Sensor Interface) kelimelerinin kısaltmasından meydana gelmiş ve bina otomasyonunda kullanılan veri yoludur. Aynı kablodan hem veri hem de enerji taşınmasına izin verir. Şekil 4.12’de ASI arayüzü ile algılayıcılar ve aktüatörler arasındaki bağlantı şekli görülmektedir.



Şekil 4.12. ASI arayüzü ile algılayıcılar ve aktüatörler arasındaki bağlantı.

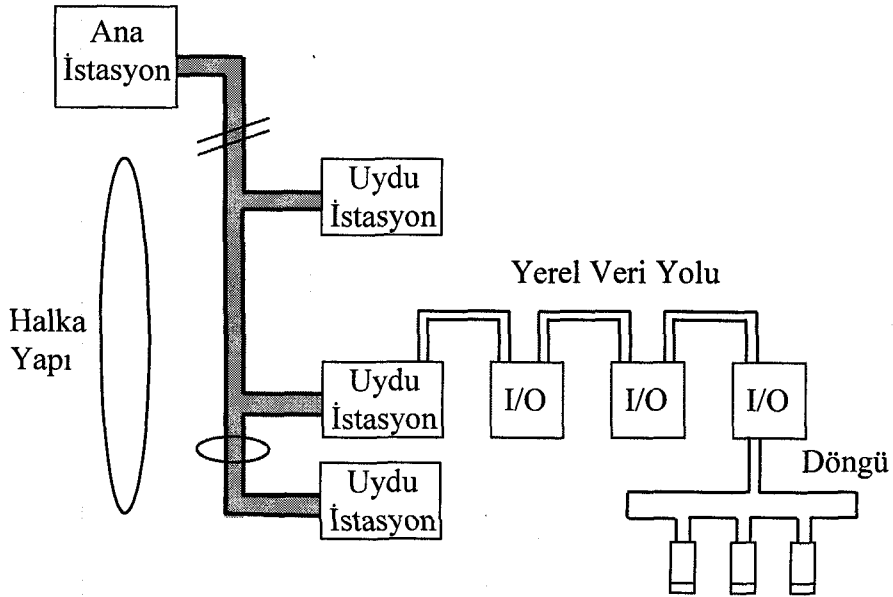
ASI veri yolunun temelinde ana-uydu prensibi bulunmaktadır. Bir hat boyunca bir ana istasyona otuz bir adet uydu istasyon bağlanabilmektedir [52]. ASI veri yolunda, çevirim hızı 5 ms’den daha küçüktür. Her uydu istasyona dört adet ayrık giriş ve dört adet ayrık çıkış bağlanabilir ve ilave edilen her uydu istasyon otomatik olarak adreslenir. Şekil 4.13’de ASI veri yolunda ana-uydu iletişimi görülmektedir.



Şekil 4.13. ASI veri yolunda ana-uydu iletişimi.

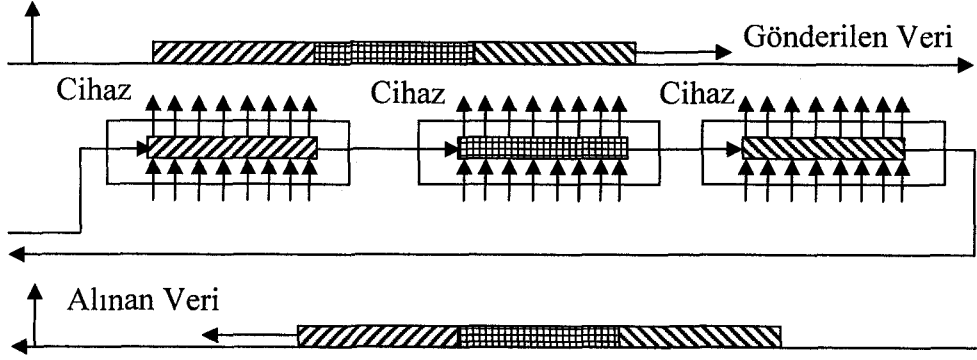
4.3.2 Interbus veri yolu

Interbus veri yolu genelde üretim tesislerinde kullanılmakta ve yapısal olarak halka (ring) temellidir [53]. Şekil 4.14’de Interbus veri yolu prensip şeması görülmektedir. 4096 adet giriş ve çıkış birimi Interbus vasıtasıyla merkezden denetlenebilir. İletişim hız 500 Kbit/sn’dir. İki istasyon arası mesafe maksimum 400 m ve Interbus veri yolunun fiziksel boyutu ise maksimum 13 km’dir. Sisteme entegre olan döngülerin fiziksel boyutları 50 m’yi aşmamalıdır. Çevrim zamanı giriş çıkış sayısına bağlı olarak maksimum iki mili saniyedir.



Şekil 4.14. Interbus veri yolu prensip şeması.

Şekil 4.15’de ana istasyon ile uydu istasyonlar arasındaki veri alışverişi görülmektedir. Interbus veri yolunda ana istasyon her uydu istasyon için göndereceği veriyi bir katar halinde ard arda gönderir.



Şekil 4.15. Ana istasyon ile uydu istasyonlar arasındaki veri alışverişi.

4.3.3 Profibus veri yolu

Otomasyon sistemlerinde kullanılan Profibus veri yolunun üç farklı çeşitli bulunmaktadır [54]. Bunlar; PROFIBUS-DP, PROFIBUS-PA ve PROFIBUS-FMS’dir.

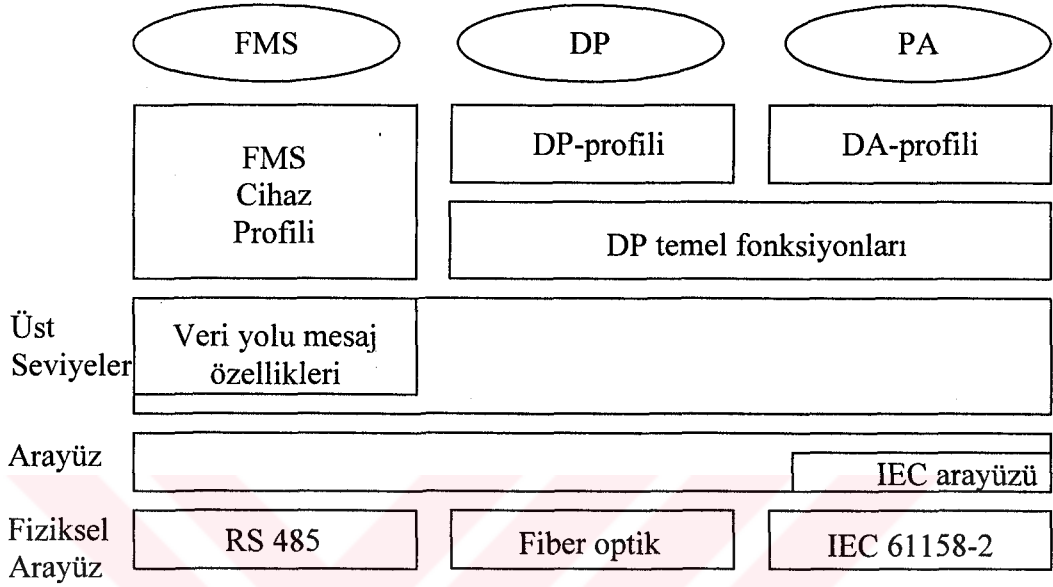
i) PROFIBUS-DP (Distributed Processing): merkezden uzakta olan I/O sistemleri ile PLC arasındaki iletişimi sağlamak için tasarlanmıştır. Temelde bir ana istasyon ve onun denetimi altında bulunan uydu istasyonlardan oluşan bir yapıya sahiptir.

ii) PROFIBUS-PA (Process Automation): Sahadaki cihazlar için enerji taşıyan kablolardan veri iletişiminin de gerçekleşmesini sağlar. Algılayıcılar ve aktüatörler ile ADM arasında iletişimi sağlayan Profibus veri yoludur, veri iletişimde IEC 61158-2 standardını kullanır.

iii) PROFIBUS-FMS (Field Messaging Specification): Eş düzeyler arası iletişim sağlar ve genişletilebilir bir mimariye sahiptir.

Profibus veri yolu, en yaygın kullanılan fiziksel arayüz olan RS 485 1.5 Mbit/s iletişim hızını destekler. Profibus arayüzünde 8051, NEC V25, 80186 ve 68302 işlemci ve mikro denetleyicileri kullanılır. Profibus veri yolunun dünya çapındaki en

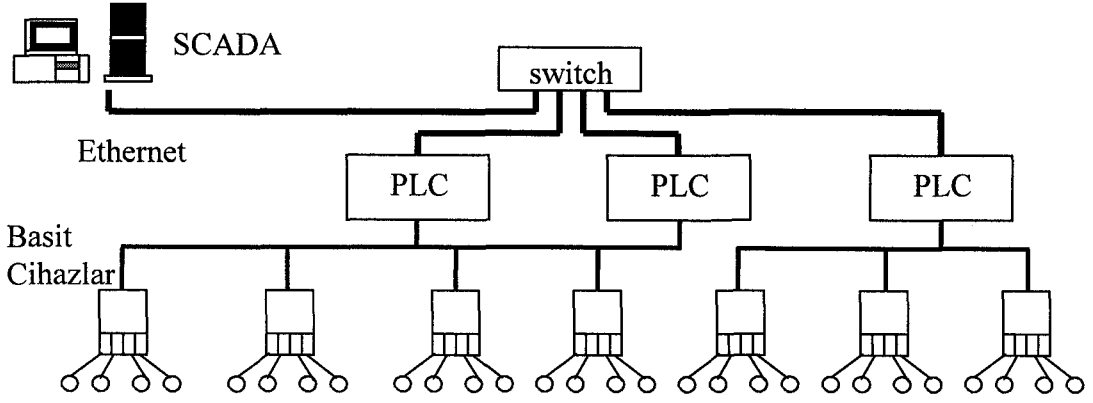
büyük destekleyicisi Siemens'dir. Profibus veri yolunun yığın yapısından da görüldüğü gibi RS 485, fiber optik ve IEC 61158-2 fiziksel arayüzünü kullanılır. Şekil 4.16'de Profibus veri yolunun yığın yapısı görülmektedir.



Şekil 4.16. Profibus veri yolunun yığın yapısı.

4.3.4 Ethernet ile gerçekleştirilen veri yolu

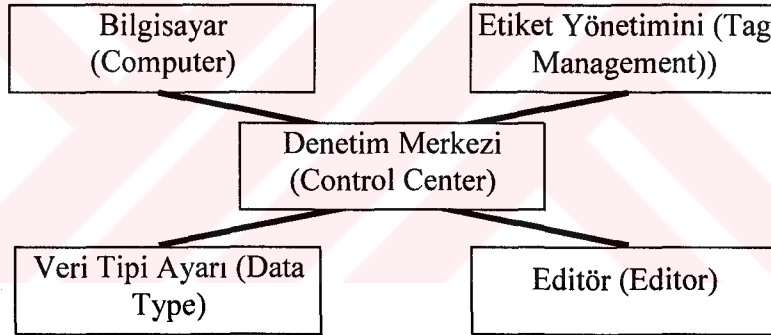
Ethernet yerel istemci sunucu uygulamalarında oldukça iyi performans ortaya koymuştur. Bu nedenle üretici firmalar ethernetin endüstriyel ortamlarda kullanımı üzerinde çalışmaya başlamış ve en sonunda Endüstriyel Ethernet'i saha uygulamaları için geliştirmişlerdir. Endüstriyel ethernetin ofis uygulamalarında kullanılan ethernetten tek farkı endüstriyel şartlarda çalışabilecek fiziksel dayanıklılıkta üretilmesidir. Her ikisi de iletişimde TCP/IP (Transmission Control Protocol and Internet Protocol) protokolü kullanır. Fiyatının ucuz olması, kurulum maliyetinin çok düşük olması ve yüksek nitelikli personel gerektirmemesi endüstriyel ethernetin en önemli üstünlükleridir. Şekil 4.17'de ethernet tabanlı SCADA sistemi görülmektedir.



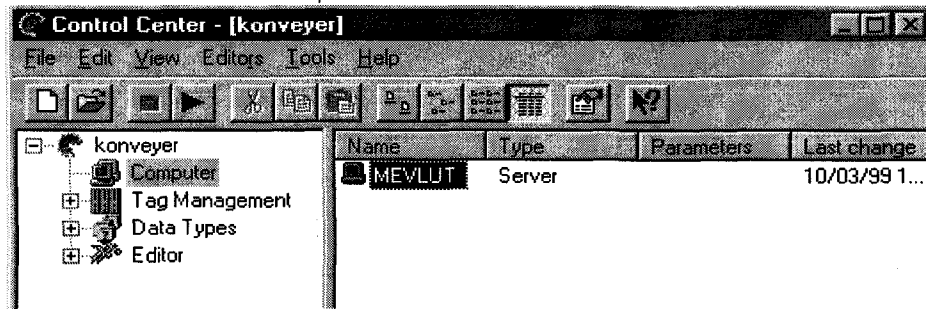
Şekil 4.17. Ethernet tabanlı SCADA sistemi.

5. WinCC SCADA YAZILIMI

Bu bölümde, uygulama çalışmasında kullanılan Siemens firması tarafından geliştirilen WinCC SCADA yazılımı anlatılmaktadır. Siemens WinCC (Windows Control Center) SCADA yazılımı ve diğer SCADA yazılımları, bilgisayar, etiket yönetimini, veri tipi ayarı ve editör alt programlarından oluşan bir yapıya sahiptir [57-64]. Bu dört ana kısım, denetim merkezi (Control Center) tarafından denetlenir. Şekil 5.1'de Siemens WinCC SCADA yazılımının yapısı, Şekil 5.2'de ise denetim merkezi görülmektedir.



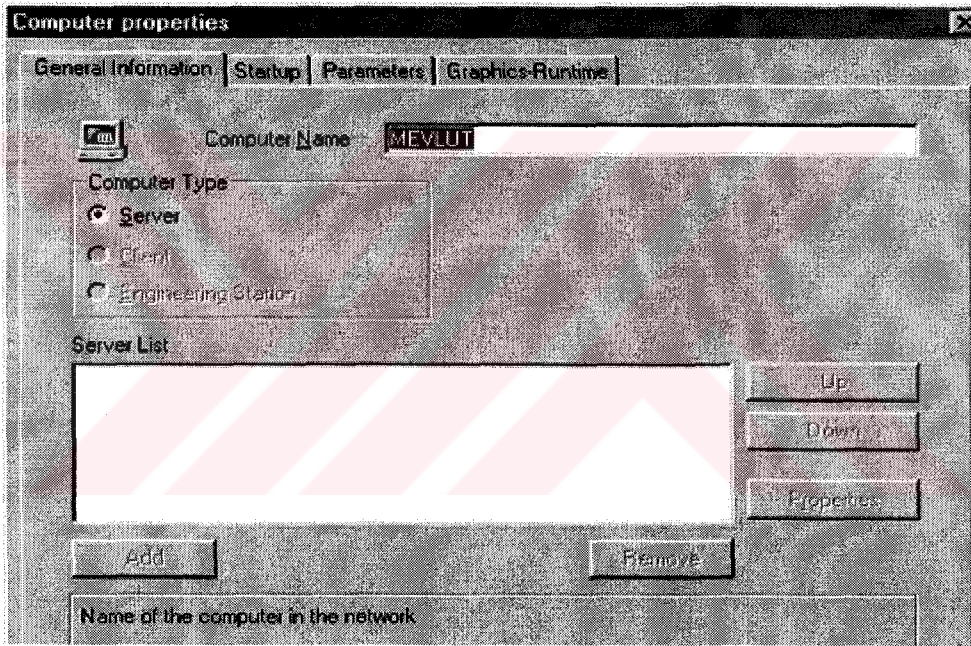
Şekil 5.1. Siemens SCADA yazılımının yapısı.



Şekil 5.2. Denetim merkezi (Control Center).

5.1. Bilgisayar (Computer)

Bu kısımda, hazırlanan programın çalıştırılacağı bilgisayar ve ilk olarak hangi sayfanın gösterime alınacağı, gösterimdeki sayfaya ait sayfanın boyutlarını sabitleme veya kullanıcıya bağlı olarak ayarlanabilme yetkilendirilmesi yapılmaktadır. Ayrıca, SCADA yazılımı çalıştırıldığında raporlama, alarm arşivlenmesinin başlatılması, etiket yönetiminin ve kullanıcı yetkilendirme programının çalıştırılması ile ilgili ayarlarda bu kısımda yer almaktadır. Şekil 5.3’de bilgisayar özelliklerinin ayarlanması görülmektedir.

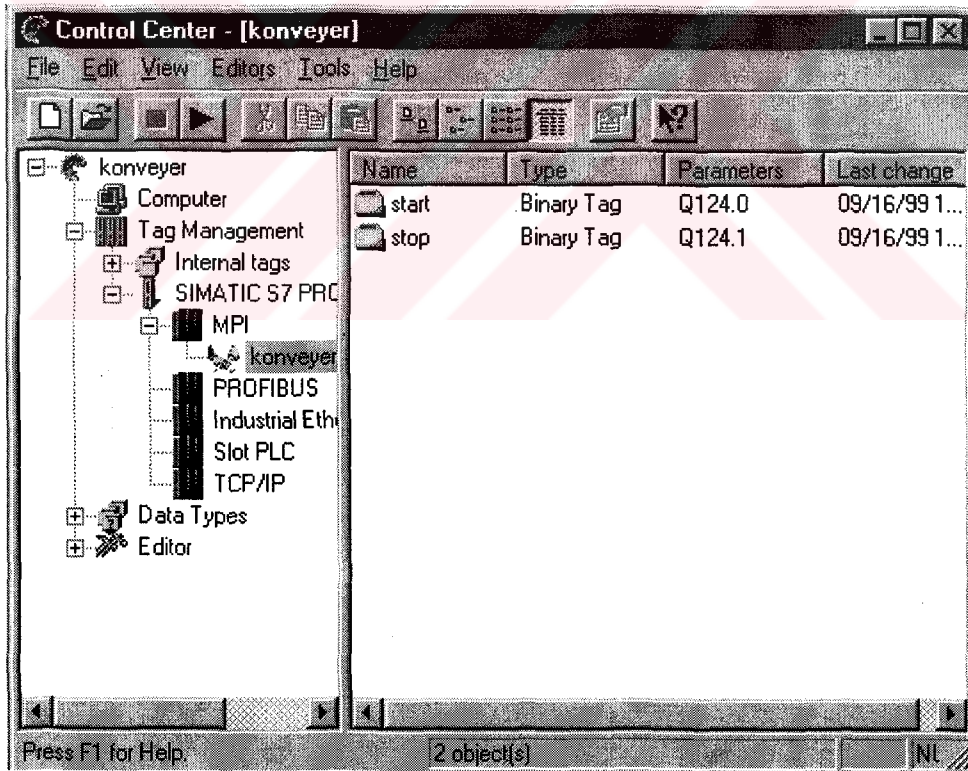


Şekil 5.3. Bilgisayar özelliklerinin ayarlanması.

5.2. Etiket Yönetimi (Tag Management)

Bu alt programda PLC’ler ile SCADA yazılımı arasındaki veri alış-verişi için gerekli veri yolu ayarlamaları bulunmaktadır. Bu veri yolları ile SCADA yazılımının dış dünya ile haberleşmesi sağlanır. Etiket yönetimi çeşitli iletişim arayüzlerini içerir bunlar ;

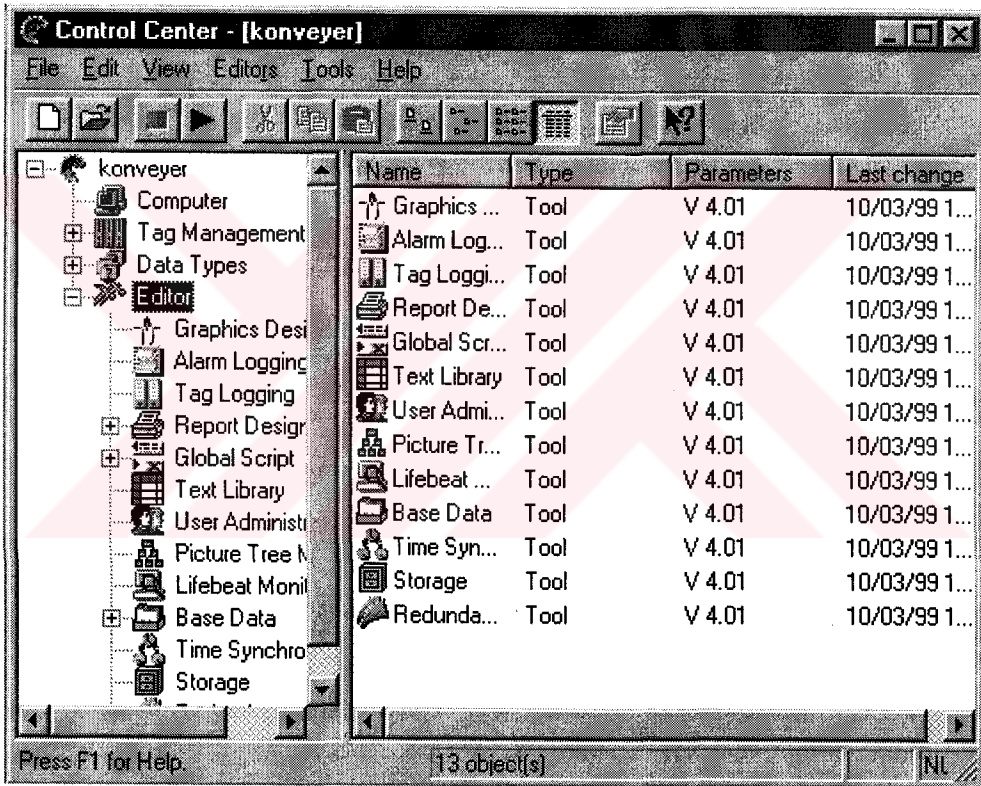
SIMATIC S7, SIMATIC S5, SIMATIC TI drivers, PROFIBUS DP, PROFIBUS FMS, Industrial Ethernet, TCP/IP, OPC-Server, OPC-Client, DDEServer ve ActiveX. İsteğe bağlı olarak Allen Bradley PLC'ler (DF1, DH, DH+, DH485, ControlNet, DeviceNet, Ethernet), Modicon (Modbus ASCII, RTU, Modbus+, Modbus Ethernet), GE (SNP, SNPX), Mitsubishi, Omron, Telemecanique ve diğer PLC'ler ile haberleşme sağlanabilir. WinCC, OPC istemci (OPC Client) ve diğer uygulamalar için bir arayüz görevi gören OPC Sunucu (OPC Server) içerir. WinCC, OPC Sunucuyu kullanarak MES/EPR sistemlerine veri gönderebilir ve bu sistemlerden veri alabilir. Ayrıca, PLC'lerin hangi adresinden veri aktarımı yapılacaksa, bu adres ile SCADA yazılımı arasındaki, gerekli köprüyü oluşturacak olan etiket bu kısımda oluşturulur. Şekil 5.4'de etiket yönetimi alt programı görülmektedir.



Şekil 5.4. Etiket yönetimi alt programı.

5.3. Editör (Editor)

Bu kısımda, insan makine arayüzünün gerçekleştirilmesi için gerekli olan editör alt programları bulunmaktadır. Bu alt programlar kullanılarak, gerekli grafik canlandırma, rapor ve alarmlar hazırlanır. Editörde, Graphics Designer, Alarm Logging, Tag Logging, Report Designer, Global Script, Text Library ve User Administrator alt programları bulunmaktadır. Şekil 5.5’de Editör ve alt programları görülmektedir.

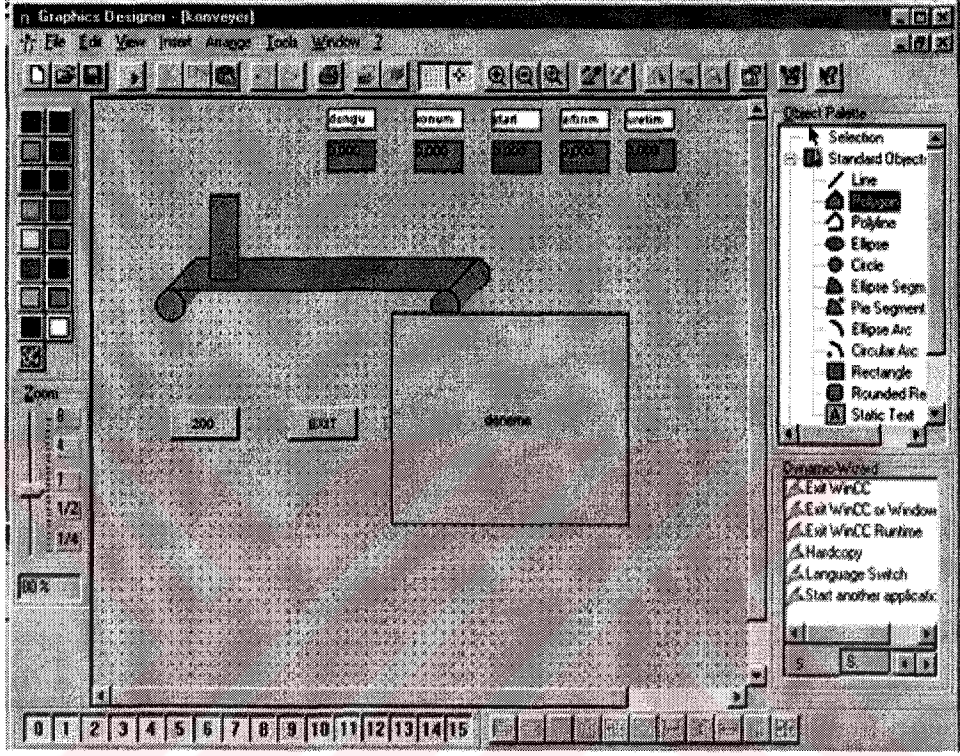


Şekil 5.5. Editör ve alt programları.

5.3.1 Grafik Tasarımcısı (Graphics Designer)

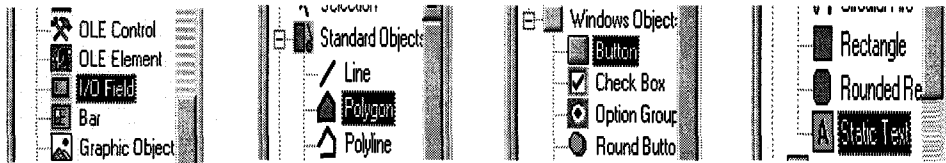
Grafik Tasarımcısı, denetlenen sistem için hazırlanan SCADA yazılımı çalıştırıldığında operatörün gösterimine sunulacak olan, canlandırma resimlerinin hazırlandığı alt

programdır. Şekil 5.6'da grafik tasarımcısında canlandırma resminin hazırlanışı görülmektedir.



Şekil 5.6. Grafik Tasarımcısında canlandırma resminin hazırlanışı.

Grafik Tasarımcısında, canlandırma resminde kullanılmak üzere nesnelere bulunmaktadır. Şekil 5.7'de giriş/çıkış alanları, çokgen, buton ve sabit metin nesnelerinin kullanımı görülmektedir.



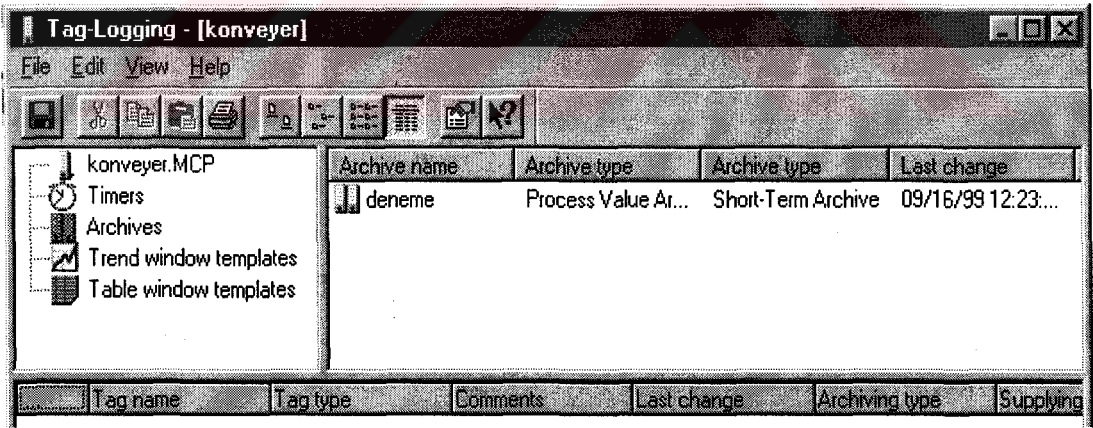
Şekil 5.7. Giriş/Çıkış alanları, çokgen, buton ve sabit metin nesnelerinin kullanımı.

Grafik Tasarımcısında bulunan bileşenler;

- i) Giriş/Çıkış alanları (I/O Fields), program içinde üretilen veya dış dünyadan alınan verileri doğrudan kullanıcıya aktarmak için kullanılır. Burada kullanılan veriler ikili, kelime, ondalıklı ve katar (binary, word, double word, float ve string) tipinde olabilir.
- ii) Çokgen (Ploygon), canlandırma resimlerinde çok kenarlı şekilleri çizmek için kullanılır.
- iii) Buton (Button), bir canlandırma resminden diğerine, çeşitli verileri etiket'lere (tag) göndermede ve bunun gibi bir çok işlemlerde kullanılır.
- iv) Sabit metin (Static Text), bir canlandırma resmine metin yazılması işlemi için kullanılır.

5.3.2 Etiket Kütükleme (Tag Logging) alt programı

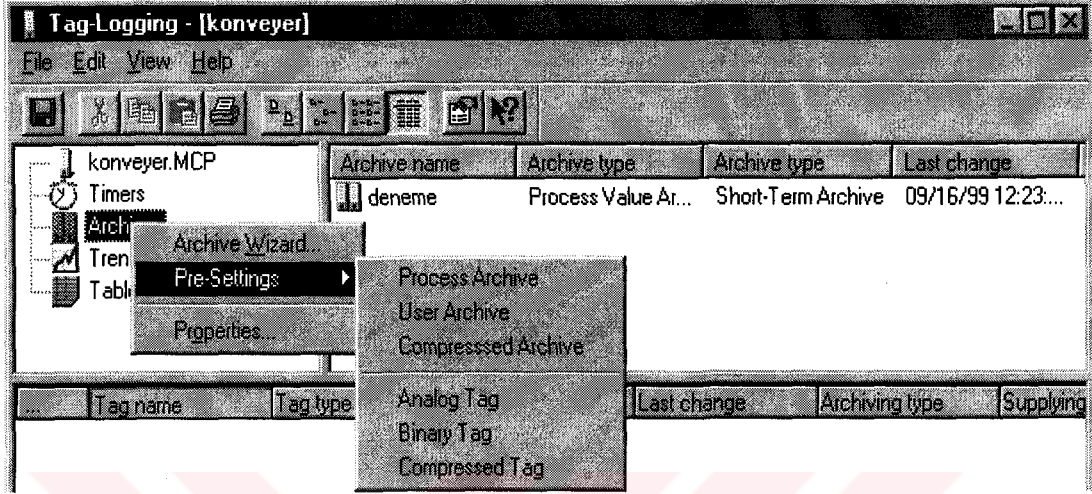
Kontrol edilen sistemden alınan verilerin kullanıcıya eğriler şeklinde aktarılması için kullanılan alt programdır. Şekil 5.8'da Etiket Kütükleme alt programı görülmektedir.



Şekil 5.8. Etiket Kütükleme alt programı.

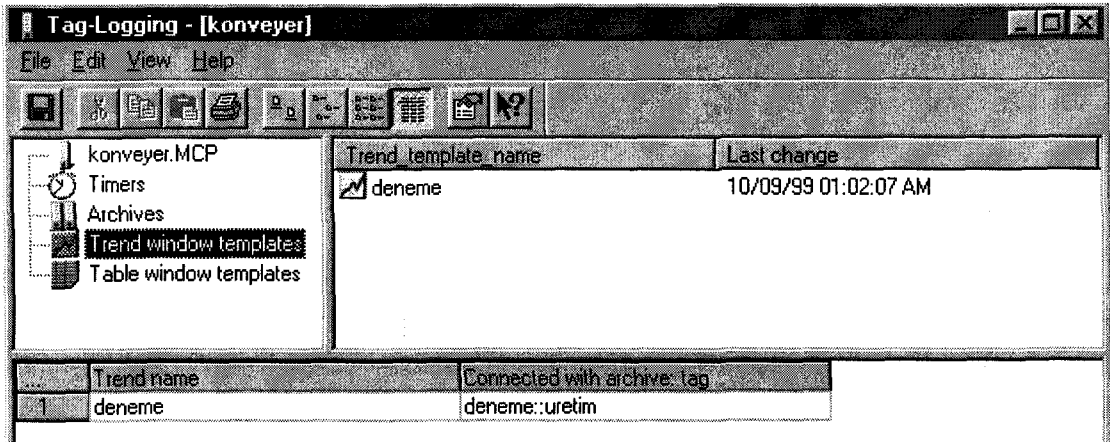
Etiket Kütükleme'de kullanılan veriler arşivlenerek sistem hakkındaki veriler gelecekte kullanılmak üzere depolanır. Bu veriler, periyodik aralıklarla depolanabildiği gibi olaya bağlı olarak da depolanabilir. Aşağıda Şekil 5.9'da, bir eğri (trend) oluşturabilmek için

ilk olarak arşivleme işleminin gerçekleştirilmesi ve bu işlemin nasıl yapıldığı görülmektedir.



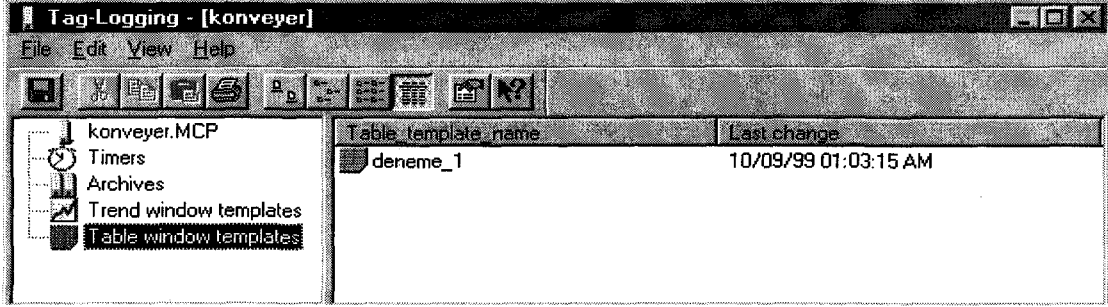
Şekil 5.9. Kütükleme işleminin gerçekleştirilmesi.

Daha sonra, eğri ayarlamalarının gerçekleştirilmesi gerekmektedir. SCADA sisteminin çalışması esnasında ekranda eğri görünmesi isteniyorsa gerekli ayarlamalar bu kısımda yapılır. Şekil 5.10'da eğri ayarlamasının nasıl yapıldığı görülmektedir.



Şekil 5.10. Eğri ayarlama işleminin gerçekleştirilmesi.

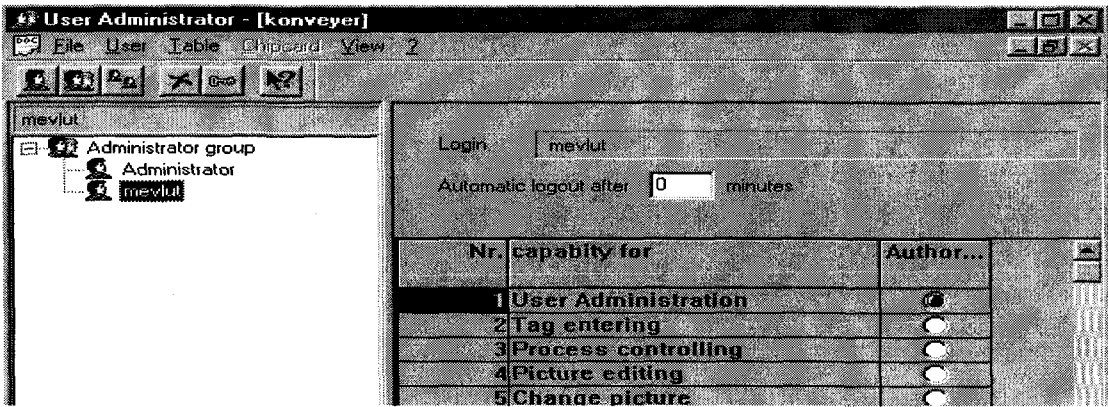
SCADA sisteminin çalışması esnasında ekranda istenilen verilerin bir tabloda görünmesi isteniyorsa gerekli ayarlamalar bu kısımda yapılır. Tablolar ile gerekli ayarlamalar Şekil 5.11’de görülmektedir.



Şekil 5.11. Tablo işleminin gerçekleştirilmesi.

5.3.3 Kullanıcı Yönetimi (User Administrator) alt programı

Kullanılan verilerin, canlandırma resminin üstünde gösterimi esnasında, belirli verilerin değiştirilmesi veya kullanıcıların bu verilere müdahalesi belirli yetkilerle kısıtlanmıştır. Şekil 5.12’de Kullanıcı Yönetimi alt programının kullanımı görülmektedir.

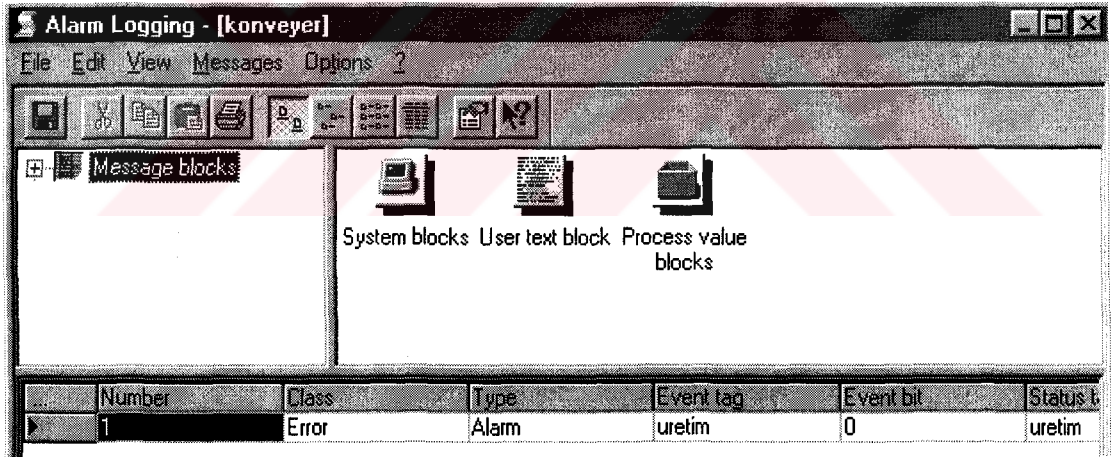


Şekil 5.12. Kullanıcı Yönetimi alt programı kullanımı.

Yetkilendirme, yetkisi bulunmayan kullanıcıların sisteme verebileceği hataları ortadan kaldırmak veya tüm kullanıcılar tarafından erişilmesi istenilmeyen verilerin saklanması sağlamaktadır. Bu alt programda yetkilendirme seviyeleri çeşitli gruplardan oluşmaktadır. Bunlar; Yönetici, etiket girişi, proses kontrolü, resim denetimi, arşiv başlatma, arşiv durdurma, proje yönetimi ve denetimi gibi guruplardan oluşmaktadır.

5.3.4 Alarm Kütüğü (Alarm Logging) alt programı

SCADA sisteminin çalışması esnasında sahadan toplanan verilerde bir veri sınır değerini üstüne çıktığında veya altına indiğinde kullanıcıyı ve sistemi uyarma işlemini gerçekleştirmek için bu alt program kullanılır. Şekil 5.13'de Alarm Kütüğü alt programının kullanımı görülmektedir



Şekil 5.13 Alarm Kütüğü alt programı kullanımı.

5.4. Yardımcı Programlar

Aşağıda tanıtılmış olan yardımcı programlar SCADA sisteminin ihtiyaçları göz önüne alınarak sisteme kurular. Böylece gereksiz yere sistemin maliyetinin artmasının önüne geçilmiş olunur. Yardımcı programlar şunlardır.

- i) PLC yedekleme programı, WinCC ile bağı olan SIMATIC S7 300 ve S7 400 PLC'lerin kurulum ayarları otomatik olarak yedeklenir. Bu yedekleme sayesinde PLC'lerde herhangi bir arıza meydana geldiğinde WinCC otomatik olarak yedek PLC'leri devreye alabilir.
- ii) Sunucu yedekleme programı ile, bir ana sunucuda çalışan SCADA programı yedek bir sunucuda eş-zamanlı olarak çalışır. Bu sayede ana sunucuda gerçekleşen veri toplama ve veri gönderme işlemleri yedek sunucuda yedeklenir, ve ana sunucunun herhangi bir arıza sonunda devre dışı kalması ile yedek sunucu en kısa zamanda devreye girerek sistemin gözetlenmesinde ve kontrol edilmesinde süreklilik sağlar.
- iii) Çoklu sunucu ve istemci mimarisi programı, WinCC'de sunucu yedekleme özelliği genişletilerek büyük alanlara yayılmış uygulamalarda PC'ler TCP/IP kullanarak birbirlerine bağlanabilir. WinCC istemcileri üst seviyede bulunan sunuculara eş-zamalı olarak erişip veri alabilir veya gönderebilir.
- iv) Web sunucu programı, WinCC internet teknolojisi ile uzak bir birimdeki PC'ye web sunucu aracılığıyla ulaşabilmektedir. Aynı şekilde uzak bir birimdeki istemci standart bir internet gezgini (Internet Explorer, Netscape) kullanılarak gerekli verileri okuyabilir ve yazma işlemlerini gerçekleştirebilir. WinCC standart olarak Microsoft Internet Information Server (IIS)'i kullanır.
- v) Haberci (Messenger) programı, sistem çalışırken önceden belirlenen veriler, ekran görüntüleri ve işlemin kritik değerleri belirlenen servis elemanlarına, yöneticilere veya mühendislere elektronik mektup (e-mail) olarak gönderme özelliğine sahiptir.
- vi) Gardiyan (Guardian) programı, çalışma esnasında Video Yönetim Sisteminin (Video Management System) Active X kontrole gömülmesi işlemine denir. Bir video kamera ile operatörün erişmesinin oldukça güç olacağı ve kritik değerlerin gözlenmesi istenen üretim sahasından alınan görüntülerin WinCC ortamına aktarılması sağlanır.
- vii) ERP entegrasyon programı, günümüzde otomasyon sistemleri, işletme kaynakları planlama (ERP) sistemleri ile ortaklaşa çalışabilme yeteneğine sahiptir. WinCC' de bu yeteneğe sahip bir SCADA yazılımıdır. MES/ERP sistemleri merkezi bir veritabanı üzerinden kontrol edilen sistemin bilgilerine erişir ve bu bilgileri kontrol eder. Genelde Oracle, Informix, Ingres ve Microsoft SQL ile merkezi bir veri tabanı uygulaması gerçekleştirilir. Veritabanlarının çoğu ODBC/SQL (Open Data Base Connectivity) /

(Structured Query Language) uygunluđuna sahiptir. Bu standart arayüzler veritabanı tablolarına erişilmesine ve sistem analiz bilgilerinin sistemin çalışması halinde bile sorgulama yapılmasına izin verir. WinCC bu standart arayüzleri destekleyerek MES/ERP sistemleri ile uygun bir çalışma ortamını sağlamış olur.

5.5. Siemens WinCC SCADA Yazılımının Deđerlendirilmesi

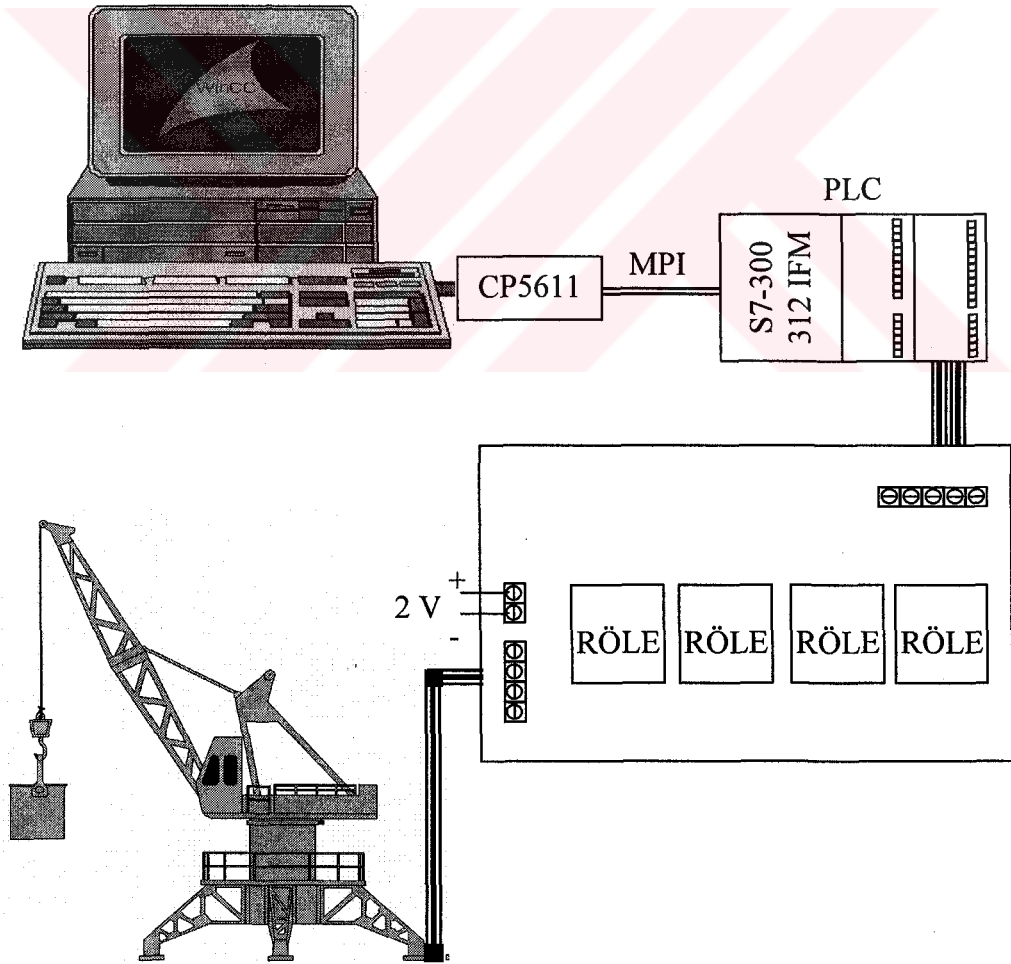
WinCC oldukça iyi bir yapıya sahip, grafik editör, alarm, etiket kütükleme, raporlama ve özel bir senaryo oluşturabilme özelliđi (picture program, alarm, tag logging, reports, global scripts) gibi farklı SCADA fonksiyonlarına sahiptir. Bu sayede, büyük projelerin sistematik bir şekilde düzenli ve daha kısa bir zamanda sonuca ulaştırılmasını sağlar. WinCC oldukça açık bir yapıya sahiptir, böylece standart olmayan işlemler C-Scripts'leri kullanarak farklı API'ler (Application Program Interface) aracılığıyla SCADA sistemlerine yeni fonksiyonlar ilave edilebilir. Kullanıcı yetkilendirme (User Administration) fonksiyonu ile farklı bölüm ve fonksiyonlara sistem operatörlerinin erişmeleri, kullandıkları yetkiye bađlı olarak sınırlandırılmıştır. WinCC'nin, Siemens tarafından üretilen SIMATIC WinAC, SIMATIC STEP 7 ve diđer bazı mühendislik programları ve MES/ERP yönetim ve stok yönetimi programları ile ortak çalışabilmesi büyük avantajlarından sayılabilir. Bunun gibi birçok avantajlarına sahip olan WinCC SCADA paketinin bazı dezavantajları da vardır. Bunlar;

- i) Öğrenilmesi zaman almaktadır.
- ii) Programın büyük bir yapıya sahip olması sebebi ile küçük projelerin hayata geçirilmesi uzun sürmektedir.
- iii) WinCC uygulamalarında kullanılan bilgisayarlar oldukça yüksek niteliklere sahip olmasını gerektirmektedir.

6. ÖRNEK VİNÇ UYGULAMASININ SCADA ÜZERİNDEN DENETİMİ

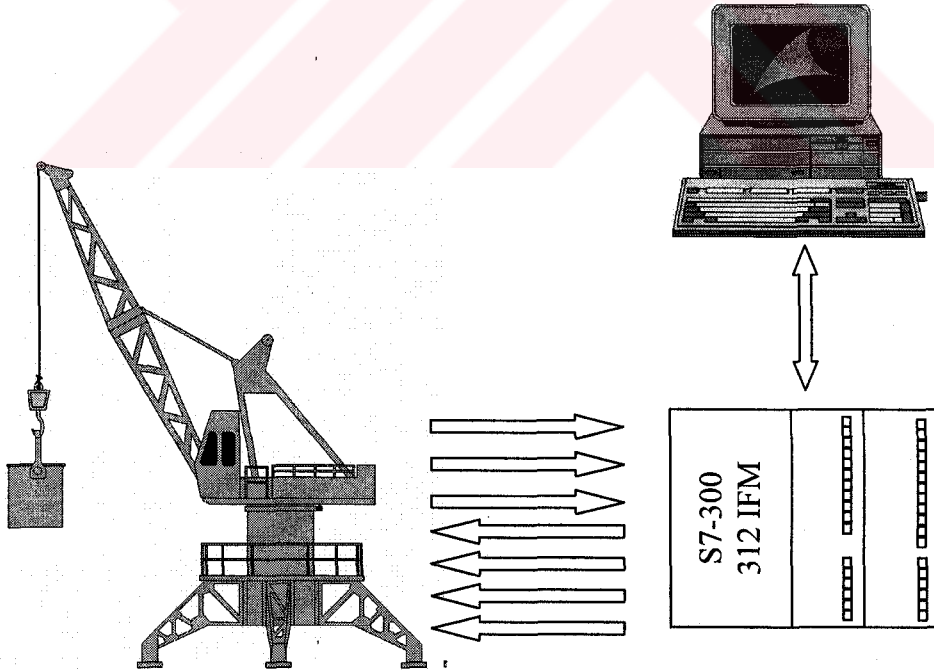
Bu bölümde, örnek bir test sistemi hazırlanarak SCADA üzerinden denetiminin gerçekleştirilmesi incelenmektedir.

SCADA sistem uygulamasını gerçekleştirmek amacıyla Şekil 6.1’de görülen model bir vinç düzeneği hazırlanmıştır. Bu düzenek, model bir vinç, Siemens S7-300 CPU 312 IFM, CP5611 veri haberleşme kartı ve bir bilgisayardan oluşmaktadır.



Şekil 6.1. SCADA üzerinden denetlenen model vinç düzeneği.

Model vinç 360° kendi etrafında dönme ve y ekseninde hareket edebilme özelliklerine sahiptir. Bu özellikler, vincin sahip olduğu iki adet doğru akım motoru kullanılarak gerçekleştirilir. Siemens S7-300 CPU 312 IFM programlanabilir mantık denetleyicisi ve SM334 (4Ix2O) I/O kartı kullanılmıştır. PLC, sunucu ile test düzeneği arasındaki veri alışverişini sağlayan bir köprü olarak görev yapmaktadır. Denetleme işlemlerinin büyük kısmı bilgisayardaki SCADA yazılımı içinde kullanılan küçük program parçacıkları vasıtasıyla gerçekleştirilmiştir. Sunucu ile PLC arasındaki veri alışverişi, MPI bus protokolü ve CP 5611 MPI bus kartı kullanılarak gerçekleştirilmiştir. PLC ile sunucu arasındaki veri alışveriş hızı 187.5 kb/s'dir. Hazırlanan test düzeneğinde, model vinçten PLC'ye üç adet bilgi girişi bulunmaktadır. Bunlar; model vincin kovanının üst seviyeye ulaştığını bildiren sınır anahtarının durumu, model sistemin çektiği akım ve uygulanan gerilim değeridir. PLC'den model vince dört adet denetim çıkışı bulunmaktadır. Bunlar; model vincin iki adet motorunun yön denetimi için kullanılan dört adet denetim çıkışıdır. Şekil 6.2'de test düzeneğindeki veri akış şeması görülmektedir.



Şekil 6.2. Test düzeneği veri akış şeması.

6.1. PLC Programının Hazırlanması

Model vincin denetlemesi için yazılan PLC programı oldukça basit bir yapıya sahiptir. Bunun sebebi model vincin denetiminin PLC tarafından değil SCADA sunucusu tarafından gerçekleştirilmesidir. PLC programında kullanılan giriş ve çıkış adresleri Tablo 6.1’de görülmektedir.

Tablo 6.1. PLC programında kullanılan giriş ve çıkış arayüz adresleri.

Adres	Açıklama
I124.0	Üst sınır anahtarı
PIW256	Sistemin çektiği akım değerinin ölçümü
PIW258	Sisteme uygulanan gerilim değerinin ölçümü
Q0.0	Motor 1’in yukarı yönde hareketi
Q0.1	Motor 1’in aşağı yönde hareketi
Q0.2	Motor 2’in sağ yönde hareketi
Q0.3	Motor 2’in sol yönde hareketi

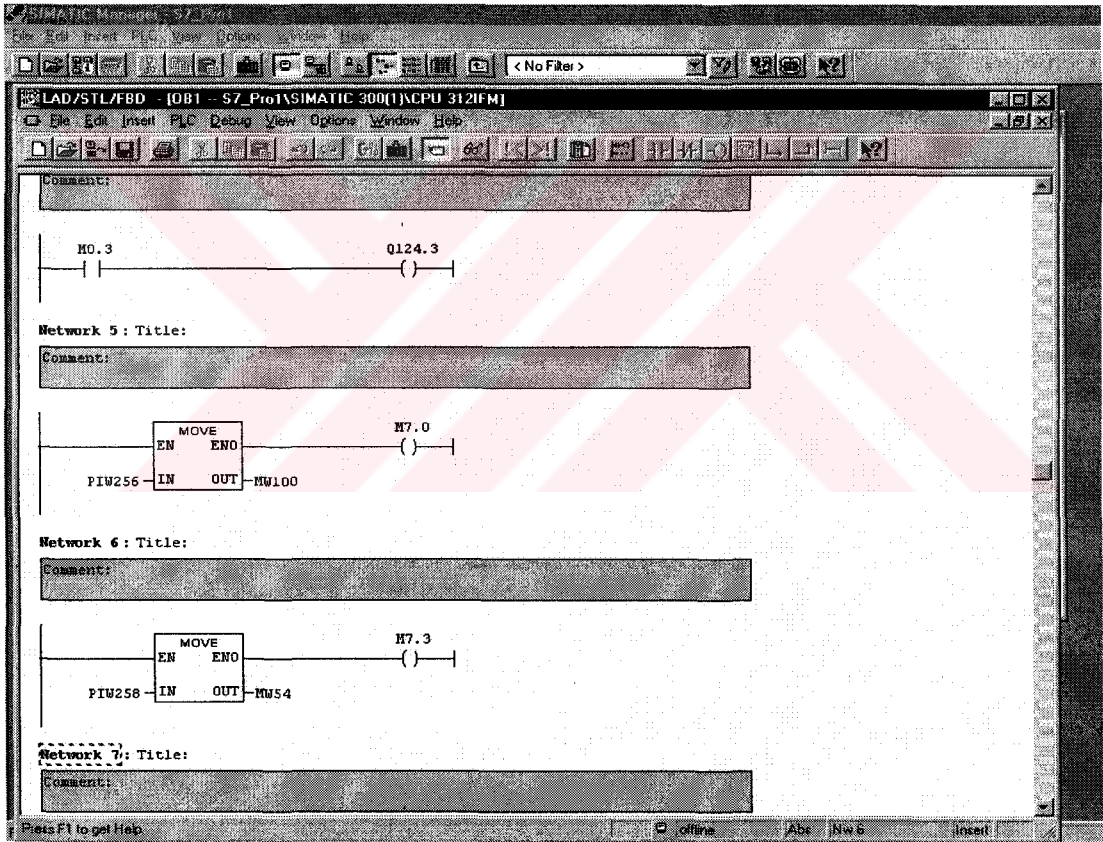
PLC ile SCADA sistemi arasındaki veri alışverişinde kullanılan adresler Tablo 6.2’de görülmektedir.

Tablo 6.2. SCADA sisteminin PLC ile veri alışverişinde kullandığı adresler.

PLC I/O Adresleri	SCADA Görüntü ve Denetim Adresleri	Açıklama
I124.0	I124.0	Üst sınırdaki kullanılan sınır anahtarı
PIW256	MW50	Sistemin çektiği akım değerinin ölçümü
PIW258	MW54	Sisteme uygulanan gerilim değerinin ölçümü
Q0.0	M0.0	Motor 1’in yukarı yönde hareketi
Q0.1	M0.1	Motor 1’in aşağı yönde hareketi
Q0.2	M0.2	Motor 2’in sağ yönde hareketi
Q0.3	M0.3	Motor 2’in sol yönde hareketi

Tablo 6.2’de her iki sütunda görülen adresler PLC’ye aittir. Aralarındaki fark ise SCADA sisteminin bazı adreslere doğrudan erişmesi bazılarına ise dolaylı olarak erişmesidir. SCADA sistemi, Q0.0, Q0.1, Q0.2 ve Q0.3 PLC çıkış modülü adreslerine, hazırlanan PLC programında bulunan M0.0, M0.1, M0.2 ve M0.3 hafıza adreslerini kullanarak erişmekte ve I124.0 PLC giriş modülü adresine bağlı üst sınır anahtarına ise PLC programını kullanmadan doğrudan erişmektedir.

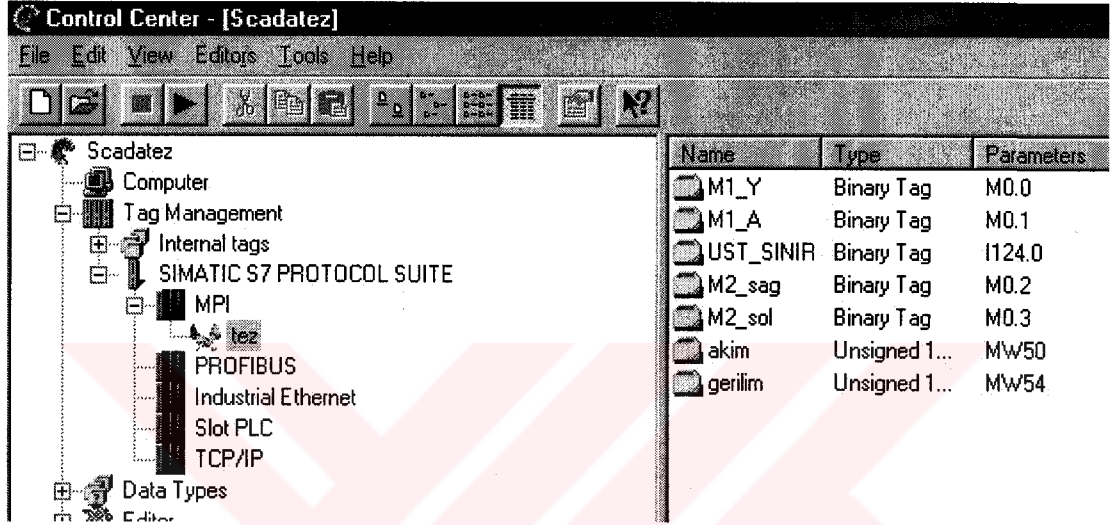
Şekil 6.3’de test düzeneğini denetlemek için yazılan Simatic Manager’deki PLC programı görülmekte ve bu programın tamamı EK A’da verilmektedir.



Şekil 6.3. Simatic Manager PLC programının görünüşü.

6.2. WinCC SCADA Yazılımında Denetim Programının Geliştirilmesi

WinCC SCADA yazılımında, denetim programı beş ayrı aşamada hazırlanmıştır. İlk aşamada, PLC ile SCADA sunucusu arasındaki veri alışverişini gerçekleştirmek için kullanılacak olan etiketler belirlenir. Şekil 6.4’de etiketlerin (tag) oluşturulması görülmektedir.



Şekil 6.4. Etiketlerin (tag) oluşturulması.

Bu aşamada ilk olarak kullanılacak olan sürücü sisteme ilave edilir. SCADA sisteminde UUB olarak, Siemens S7-300 CPU 312 IFM PLC kullanılmış ve sürücü olarak SIMATIC S7 PROTOCOL SUITE seçilmiştir. Sürücü program sisteme kurulduktan sonra, sistem etiketleri MPI klasörünün altında oluşturulan bağlantı üzerinden gerçekleştirilir. Oluşturulan bağlantının ismi “tez” olarak seçilmiştir. Bu bağlantının altında ise yedi adet etiket oluşturulmuş ve her bir etiket PLC’deki veri alışveriş adresleri ile ilişkilendirilmiştir. Yeni bir etiket oluşturulurken dikkat edilmesi gereken birkaç önemli nokta vardır. Bunlar;

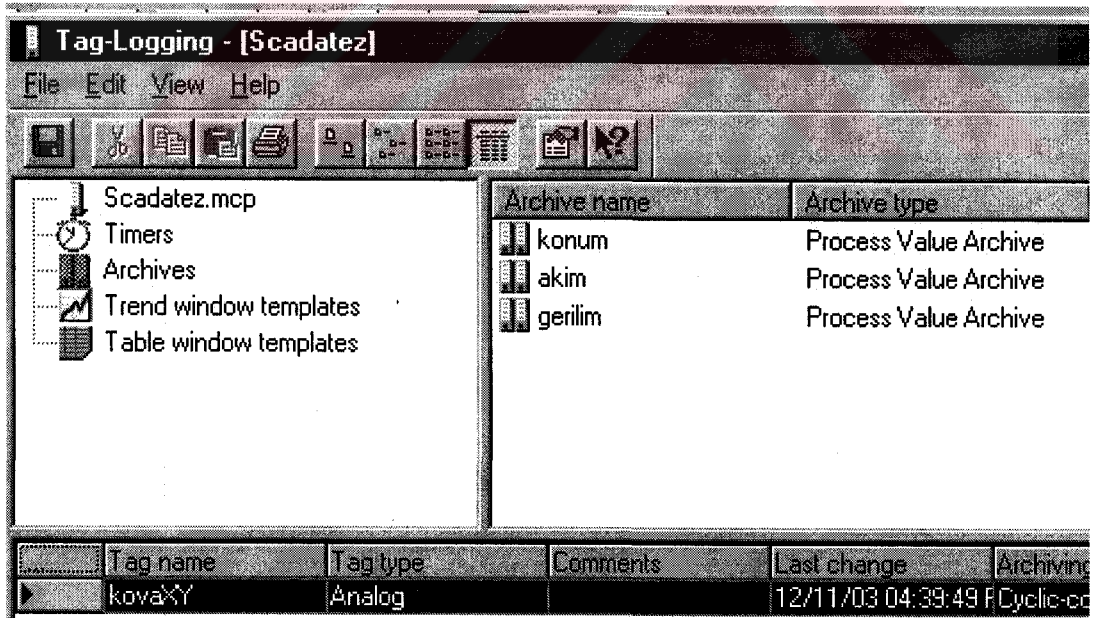
- i) PLC’de erişilecek adresin etiket ile ilişkilendirilmesi.
- ii) PLC’den alınacak veya PLC’ye gönderilecek olan veri tipinin belirlenmesi.

Tablo 6.3’de hazırlanan düzeneği denetlemek için kullanılan etiketler ve değişken tipleri görülmektedir.

Tablo 6.3. Kullanılan etiketler ve deęişken tipleri.

SCADA Görüntü ve Denetim Adresleri	Etiket	Deęişken Tipleri	Açıklama
I124.0	UST_SINIR	Binary	Üst sınır anahtarı
MW50	akim	Unsigned 16	Sistemin çektięi akım
MW54	gerilim	Unsigned 16	Sisteme uygulanan gerilim
M0.0	M1_Y	Binary	Motor 1'in yukarı yönde hareketi
M0.1	M1_A	Binary	Motor 1'in aşağı yönde hareketi
M0.2	M2_sag	Binary	Motor 2'in sağ yönde hareketi
M0.3	M2_sol	Binary	Motor 2'in sol yönde hareketi

İkinci aşamada, sahadan alınan verilerin arşivlenmesi ve çeşitli eęri ve tabloların oluşturulması gerçekleştirilmiştir. Şekil 6.5'de verilerin arşivlenmesi görülmektedir.

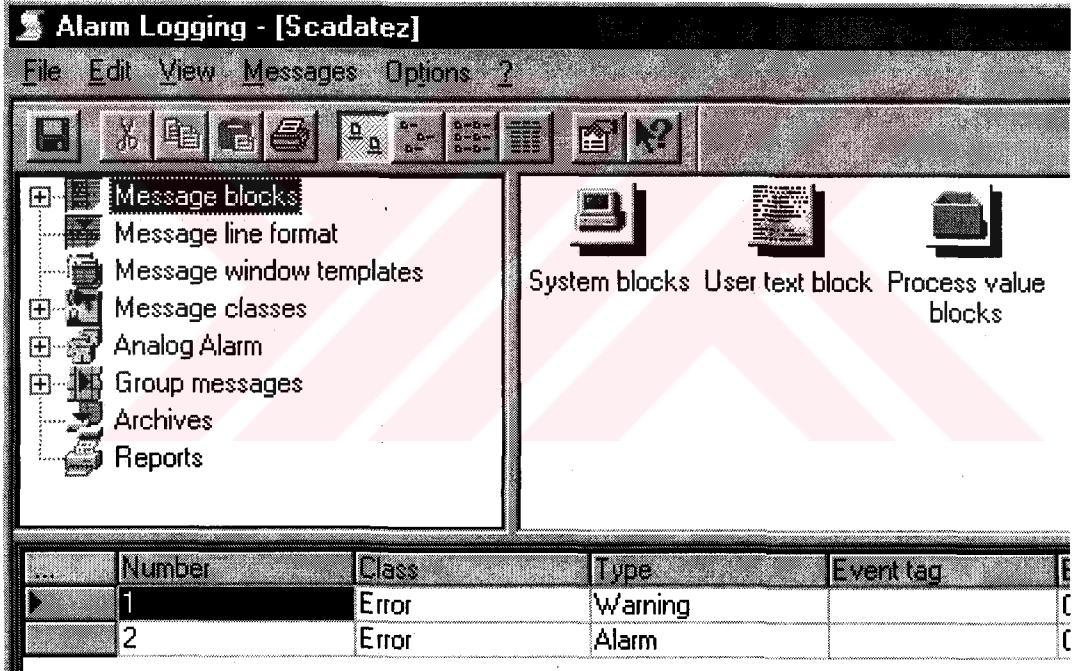


Şekil 6.5. Verilerin arşivlenmesi.

Denetim programında üç ayrı arşiv oluşturulmuştur. Bunlardan iki tanesi gerilim ve akım deęerlerini, üçüncüsü ise kovanın y eksenindeki yüksekliğinin piksel cinsinden

arşivlenmesi için kullanılmıştır. Bu aşamada aynı zamanda arşivlenen bu değerlerin grafik olarak görüntülenmesi işlemi de gerçekleştirilebilmektedir. Görüntülenmek istenen eğriler, “Trend window templates” klasörünün altına yeni grafik ilave edilerek gerçekleştirilir. Hazırlanan programda sistem akım, gerilim ve kovanın pozisyonu eğri olarak görüntülenmektedir.

Üçüncü aşamada ise, sistemde meydana gelebilecek alarm durumları belirlenmiştir. Daha sonra bu durumların sistemin çalışması esnasında görüntülenmesi ve gerçekleşen alarmları arşivleme işlemleri için gerekli olan ayarlar gerçekleştirilmiştir. Şekil 6.6’da alarmların arşivlenmesi gösterilmiştir.

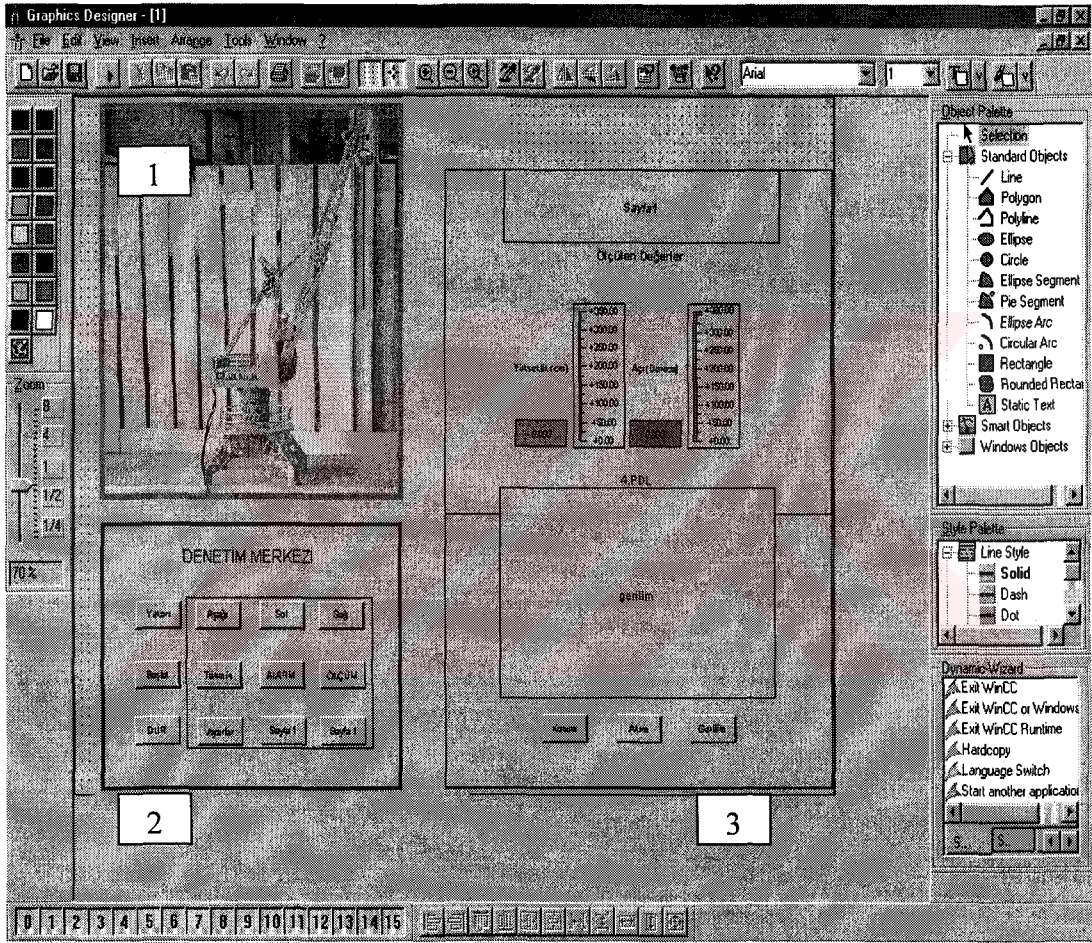


Şekil 6.6. Alarmların arşivlenmesi.

Sistem alarm koşulu olarak motor gerilim değeri 2.5 volta ayarlanmıştır. Eğer motorlara uygulanan gerilim değeri 2.2V’un üzerine çıkarsa uyarı mesajı aktif hale gelir ve operatöre “2V Dikkat sınır durumu aşıldı” mesajı iletilir. Motorlara uygulanan gerilim değeri 2.5V’un üzerine çıkarsa alarm mesajı aktif hale gelir ve operatöre “2.5V Dikkat sınır durumu aşıldı” mesajı iletilir ve sistemin çalışması

durdurulur. Sistemin tekrar çalışmaya hazır duruma gelebilmesi için motorlara uygulanan gerilim seviyesinin 2.5V'un altına indirilmesi gerekir.

Dördüncü aşamada ise, çeşitli grafik animasyonlar, alarm ve grafiklerin SCADA ekranında gösterilmesi için Grafik Designer'da resimler hazırlanmıştır. Şekil 6.7'de kullanıcı ekranı için resimlerin Grafik Designer'da hazırlanışı görülmektedir.



Şekil 6.7. Grafik Designer'da resimlerin hazırlanışı.

Grafik Designer'da kullanıcı ekranında gösterilmek üzere altı adet resim hazırlanmıştır. Bunlar;

i) Ana kullanıcı ekranını içeren "1.pdl" adlı resim dosyası üç bölüme ayrılmıştır. Birinci bölümde SCADA sistemi tarafından denetlenen model vince ait grafik animasyon bulunmaktadır. İkinci bölümde, motor 1 ve motor 2'nin aşağı/yukarı ve

sağ/sol yönlerinde hareket etme emirlerinin verildiği butonlar, ölçülen değerlerin, alarm arşivinin görüntülenmesini, ayar değerlerinin girilmesi ve kovanın konumunun belirlenmesini sağlayan resimler ile bağlantıları gerçekleştiren butonlar bulunur. Üçüncü bölümde ise, sisteme ait gerilim, akım ve konum bilgilerinin eğri ve grafik olarak gösterildiği bar grafik ve eğriler bulunmaktadır. Aynı zamanda bu bölümde sistemin çalışması esnasında meydana gelen alarmlar ekranda anında görüntülenir ve operatör uyarılır.

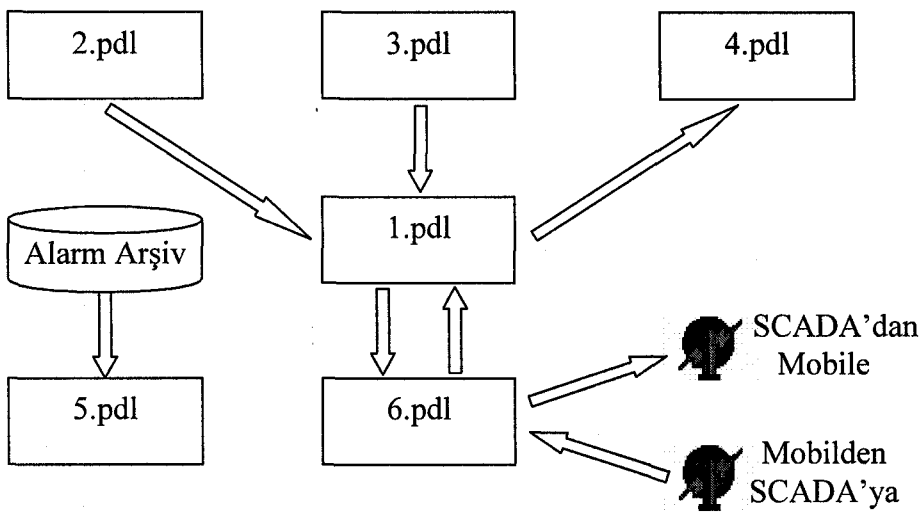
ii) “2.pdl” adlı resim dosyası, grafik animasyon ile hareketli hale getirilen vince ait kova iniş ve çıkış zaman sabiti ve kova iniş/çıkış süresi, bu resim dosyasında ayarlanır.

iii) “3.pdl” adlı resim dosyası, kova konumunun belirlenmesini sağlayan test algoritmasını içeren resim dosyasıdır.

iv) “4.pdl” adlı resim dosyası, sisteme ait gerilim, akım ve konum bilgilerinin eğri ve grafik olarak gösterildiği bar grafik ve eğrilerin bulunduğu resim dosyasıdır.

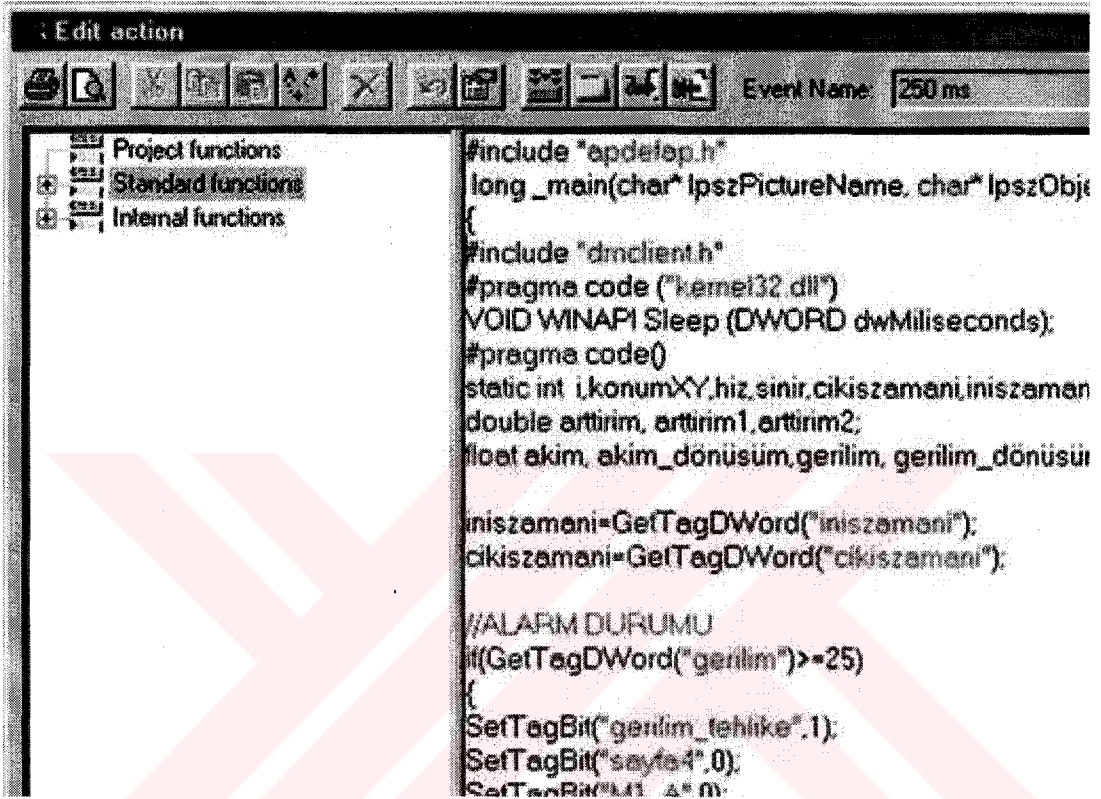
v) “5.pdl” adlı resim dosyası, alarm arşivinin görüntülediği resim dosyasıdır.

vi) “6.pdl” adlı resim dosyası, uzak denetim ve görüntüleme programlarının çalıştırıldığı, uzak istemciye gönderilen verilerin ve uzak istemciden alınan verilerin görüntülenme işleminin gerçekleştiği resim dosyasıdır. Şekil 6.8’de hazırlanan resim dosyaları arasındaki veri akışı görülmektedir.



Şekil 6.8. Resim dosyaları arasındaki veri akışı.

Son aşamada ise, grafik animasyonları denetlemek ve çeşitli mantık denetimlerini gerçekleştirmek için c-scripts program parçaları yazılmıştır. Şekil 6.9'da c-scripts'lerinin yazılışı görülmektedir.



```
#include "apdetop.h"
long _main(char* lpszPictureName, char* lpszObjName)
{
#include "dmclient.h"
#pragma code ("kernel32.dll")
VOID WINAPI Sleep (DWORD dwMilliseconds);
#pragma code()
static int i,konumXY,hiz,sinir,cikis zamani, inis zaman
double artirim, artirim1, artirim2;
float akim, akim_dönüsüm, gerilim, gerilim_dönüsüm

inis zaman=GetTagDWord("inis zaman");
cikis zaman=GetTagDWord("cikis zaman");

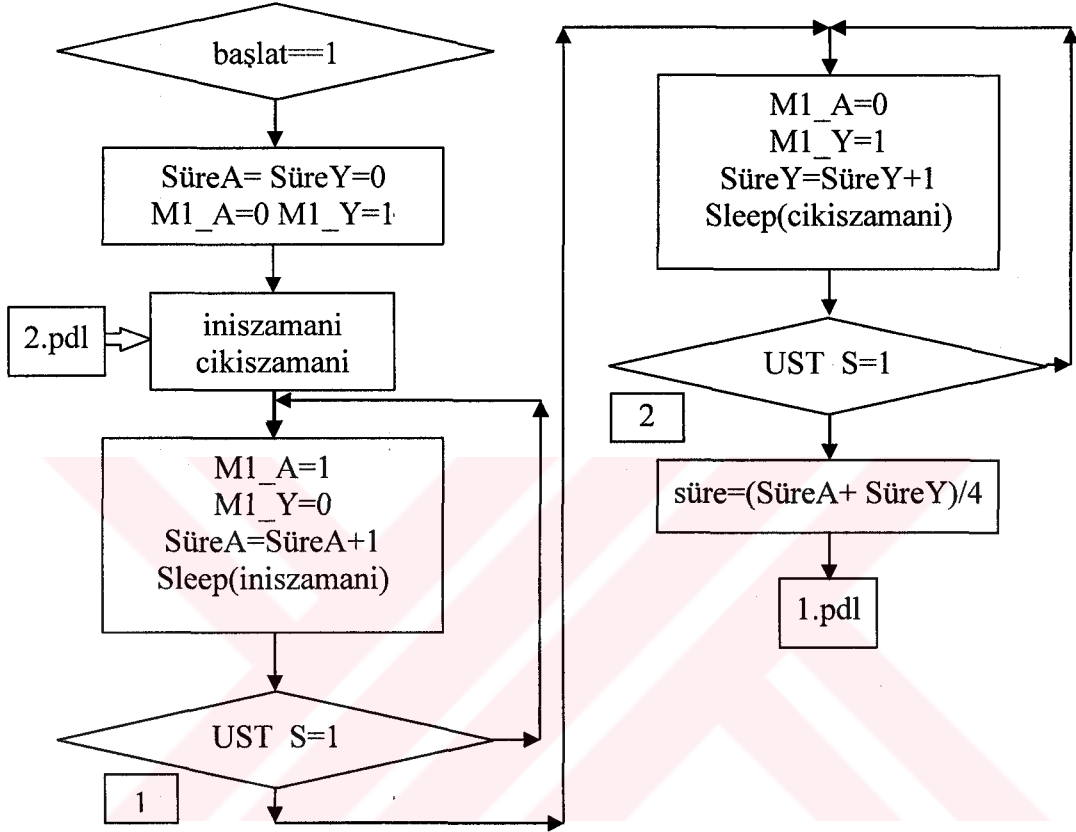
//ALARM DURUMU
if(GetTagDWord("gerilim")>=25)
{
SetTagBit("gerilim_tehlike",1);
SetTagBit("sayfa4",0);
SetTagBit("M1", 1);
}
```

Şekil 6.9. C-scripts'lerinin yazılışı.

C-scripts'ler ANSI C programlama dili kullanılarak hazırlanmıştır. C programlama dilinin kullanılması sayesinde Windows işletim sistemine ait WINAPI'ler (Windows Application Program Interface) kolayca kullanılabilmiştir. C-scripts'ler kullanılarak yazılan sistemi denetleyen denetim algoritmaları aşağıda verilmiştir.

i) Kovanın iniş/çıkış süresi tespit edildiği algoritma, "3.pdl" resim dosyasının içindedir. Hız tespit algoritması çalıştırdıktan sonra elde edilen süre değeri "1.pdl ve 2.pdl" resim dosyasında bulunan süre etiketine atanır. Süre tespit algoritması, kovanın üst sınır anahtarından aşağı noktaya veya aşağı noktadan üst sınır anahtarının olduğu noktaya çıkış veya iniş zamanının tespit edilmesi işlevini gerçekleştirir. Hız tespit algoritmasına ait c-script yazılımı EK B'de verilmiştir. Süre tespit sayfasına gidebilmek için "1.pdl" resim dosyasında bulunan tahmin butonuna

basılır. Süre tespit algoritması, “3.pdl” resim dosyasında bulunan başlat butonuna basılması ile çalışmaya başlar. Şekil 6.10’da süre tespit algoritması görülmektedir.

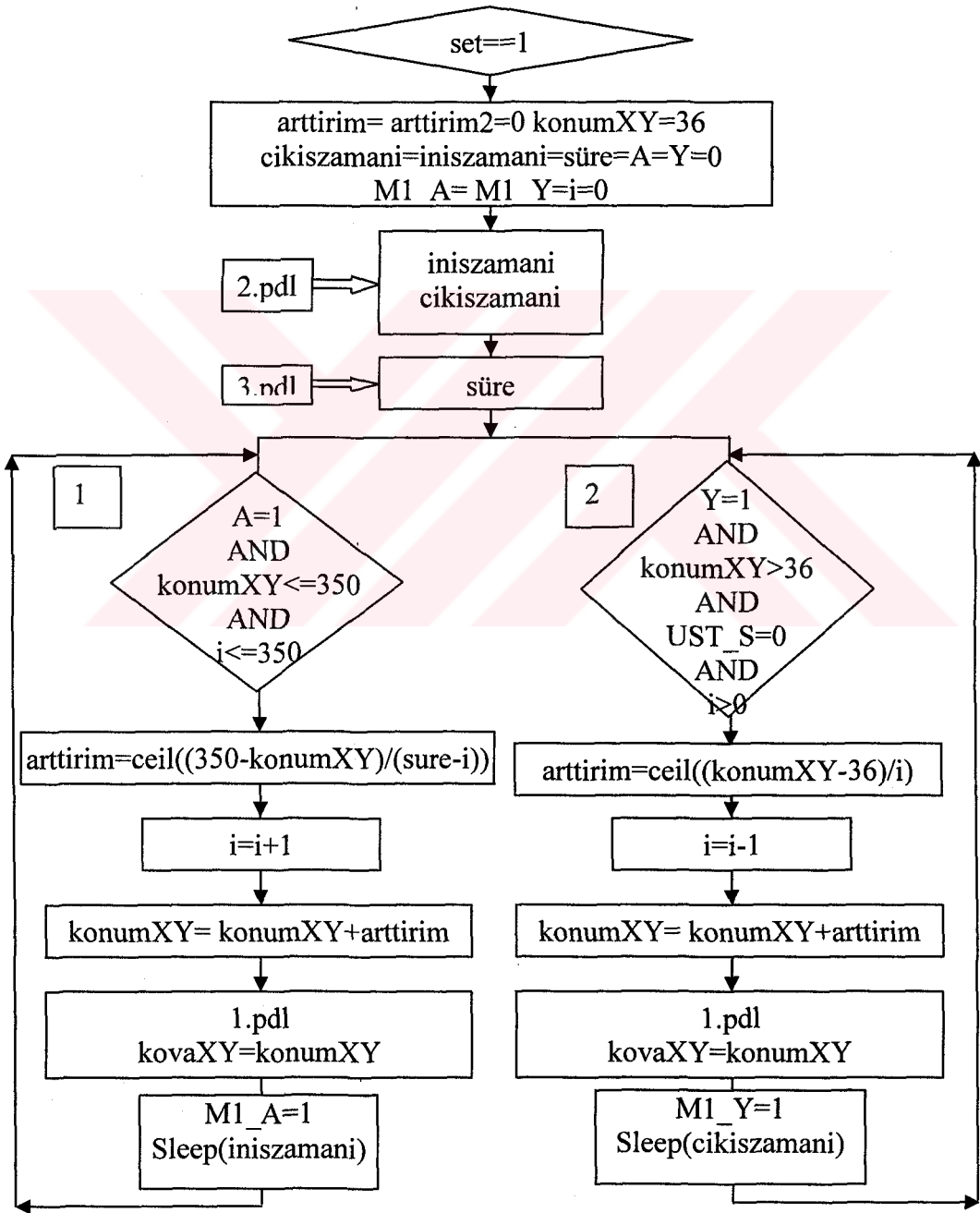


Şekil 6.10. Süre tespit algoritması.

Algoritmanın çalışmaya başlaması ile ilk olarak SüreA, SüreY ve M1_A etiketlerine sıfır değeri, M1_Y etiketine ise 1 değeri atanır. Motor 1’in yukarı çıkış ve aşağı iniş zamanlarının hesaplanmasında kullanılan zaman sabit çarpanları “2.pdl” resim dosyasından alınarak “iniszamani” ve “cikiszamani” etiketlerine atanır. Daha sonra, motor 1 yukarı yönde kovayı çekmeye başlar ve kova üst sınır noktasına geldiğinde motor 1’in yukarı yönde çalışması sona erer ve motor 1 aşağı yönde hareket etmeye başlar. Kova aşağı inip tekrar üst sınıra gelmesi için geçen süre bir döngü içinde ölçülür. Süre ölçüm işleminin sonlandırılması veya devam edilmesi 1. döngü ile denetlenir. Kovanın tekrar üst sınıra gelmesi için geçen süre SüreA etiketine kayıt edilir. Motor 1’in devir yönü tekrar değiştirilir ve kova aşağı yönde hareket etmeye

başlar ve kovanın tekrar üst sınıra gelmesi için geçen süre SüreY etiketine kayıt edilir. Bu süre 2. döngü kullanılarak denetlenir. Kovanın ortalama iniş veya çıkış süresi “SüreA” ve “SüreY” etiketine atanan değerlerin toplamının dörtte birine eşittir.

ii) Ortalama iniş çıkış süresi tespit edildikten sonra kovanın denetim işlemini gerçekleştiren “konum denetim algoritması” çalıştırılır. Şekil 6.11’de konum denetim algoritması görülmektedir.



Şekil 6.11. Konum denetim algoritması.

Kova konum denetim algoritması, “1.pdl” resim dosyasındaki “Başlat” butonuna basılması ile çalışmaya başlar. Süre denetim algoritması işlevini tamamladıktan sonra, kova üst seviyede çalışmaya hazır durumda bekler. Böylece test düzeneğinde kullanılan vincin kova konumu ile SCADA canlandırma resmindeki kovanın konumu eşleştirilmiş olur. Aşağı butonuna basıldığında kova aşağı yönde hareket eder. Kova konumu y ekseninde 350 piksele eşit olduğunda konum denetim algoritmasının çalışması durur. Kova aşağı yönde hareket ederken aşağı butonu devre dışı bırakılırsa hem test düzeneğinde hem de canlandırma resminde kovanın hareketi durur ve kovaya ait konum bilgisi piksel cinsinden kayıt edilir. Böylece tekrar aşağı butonuna veya yukarı butonuna basıldığında kova kaldığı konumdan aşağı veya yukarı yönde çalışmaya devam eder. Yukarı butonuna basıldığında hem canlandırma resminde hem de test düzeneğindeki kova yukarı yönde hareket eder. Kovanın konumu y ekseninde 36 piksele ulaştığında kovanın hareket etmesi durur. Kovanın canlandırma resmi üzerindeki konumunun (XY) belirlenmesindeki en önemli problem, kovanın yol aldığı mesafe (x) ile süre (t) birbirine bölündüğünde sonucun ondalıklı sayı çıkmasıdır. Bu problemi aşmak için ilerleme miktarı tavana yuvarlanmıştır. Her döngüde (i) ilerleme miktarı tekrar hesaplanmış sonuç tamsayı çıkana kadar bir üstüne yuvarlanmıştır. Bir üst sayıya yuvarlamadan kaynaklanan hatalı konum tespiti, kova ilerlemesini sürdürdükçe bu fark giderek kapanacak ve canlandırma resmindeki kovanın konumu ve test düzeneğindeki kovanın konumu çakışacaktır. Eşitlik 6.1 ve 6.2’de kova aşağı yönde hareket ederken kovanın konumunun hesaplanmasında kullanılan formüller görülmektedir. Eşitlik 6.3 ve 6.4’de kova yukarı yönde hareket ederken kovanın konumunun hesaplanmasında kullanılan formüller görülmektedir.

$$A = \frac{TY(350 - XY)}{t - i} \quad (6.1)$$

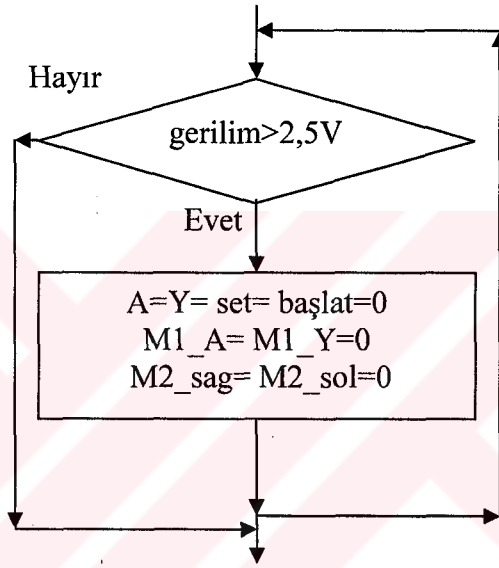
$$XY = XY + A \quad (6.2)$$

$$A=TY(\underline{XY-36}) \quad (6.3)$$

$$XY=XY-A \quad (6.4)$$

Konum denetim algoritmasına ait c-script yazılımı EK C’de verilmiştir.

iii) Son algoritma ise alarm denetim algoritmasıdır. Alarm denetim algoritması, test düzeneğini korumak amacıyla kullanılmıştır. Şekil 6.12’de alarm denetim algoritması görülmektedir.



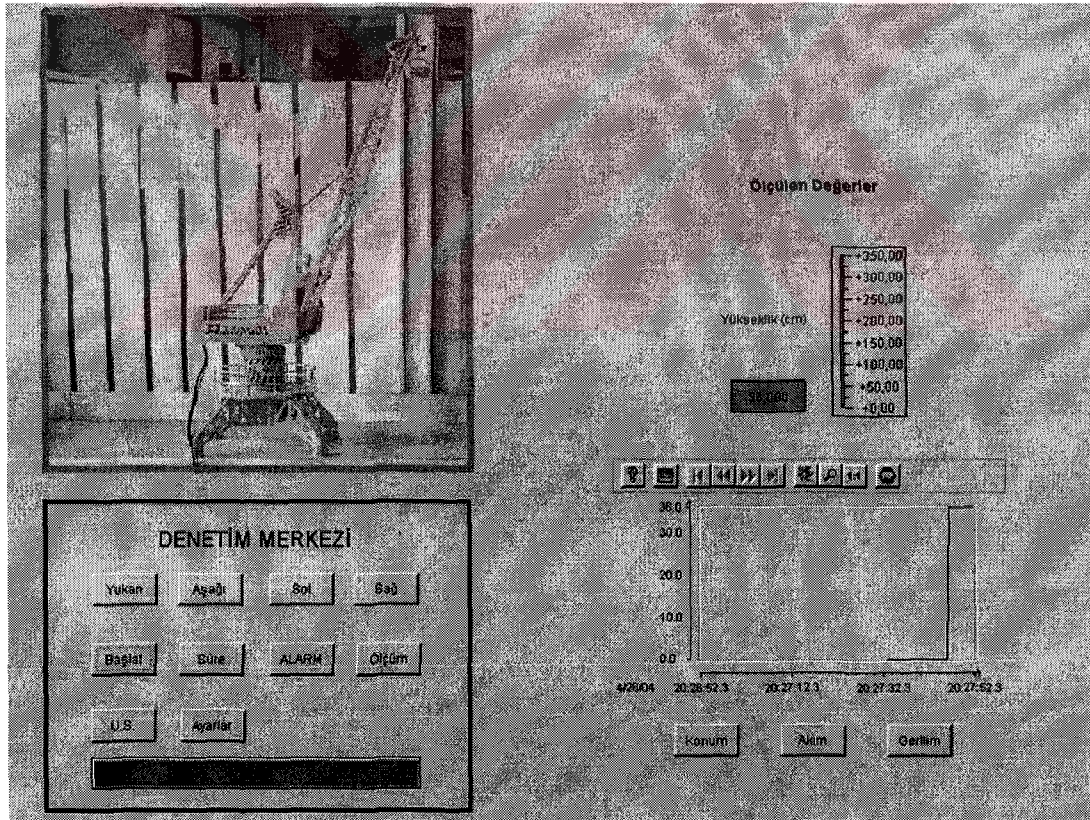
Şekil 6.12. Alarm denetim algoritması.

Alarm denetim algoritması “gerilim” etiketinden aldığı değeri denetler. Alarm denetim algoritmasına ait c-script yazılımı Ek D’de verilmiştir. Değer 2.5 voltun üzerinde ise motor 1 ve motor 2’yi devre dışı bırakır. Aynı zamanda canlandırma resminin denetimini sağlayan konum denetim ve süre tespit algoritmalarının çalışmasını devre dışı bırakır. Alarm durumu gerçekleştiğinde, kullanıcı ekranının sol üst kısmında alarm ile ilgili uyarı mesajı görünür hale gelir. Kullanıcının gerekli düzeltmeleri alması sağlanır.

Yukarıda açıklanan denetim algoritmaları ile hem canlandırma resminin hem de test düzeneğinin denetimi gerçekleştirilir. Test düzeneğini denetlemek için kullanılan diğer c-script’leri EK E’de verilmiştir. Buraya kadar yapılan SCADA sistemi, klasik

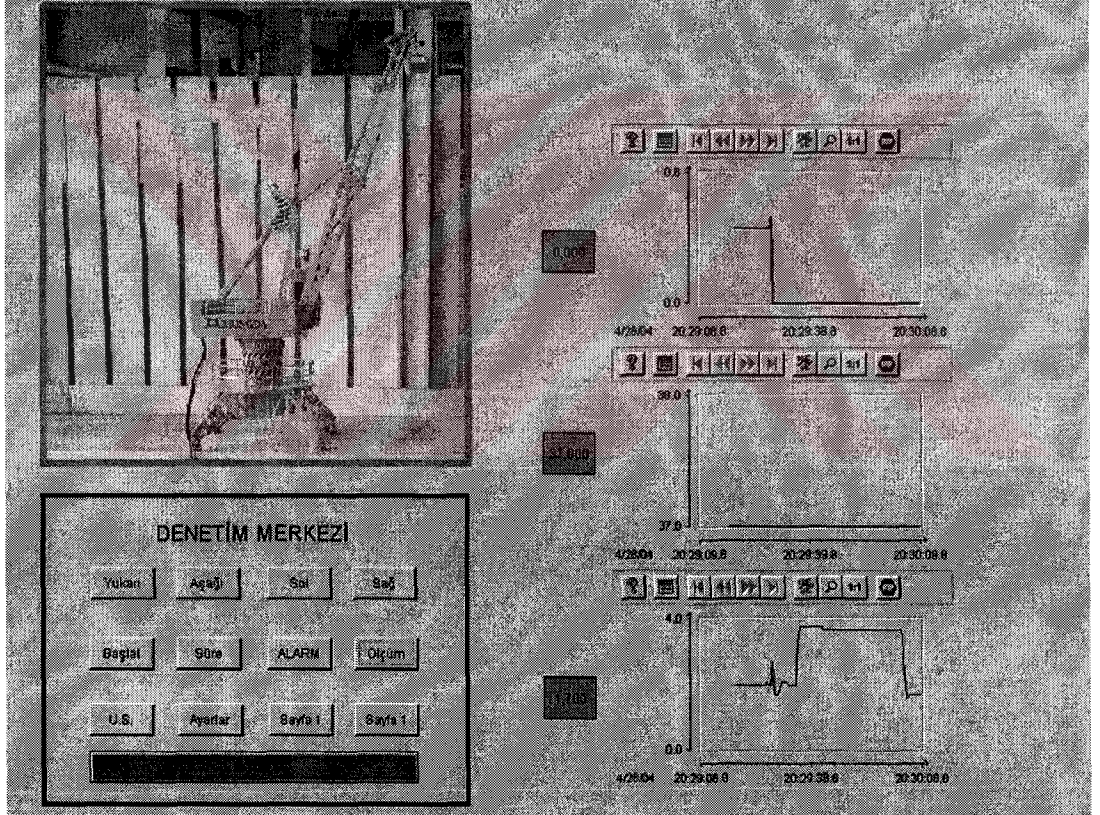
SCADA sistemi yapısındadır ve uygulamada kullanılan SCADA sistemleri ile aynı fonksiyonlara sahiptir.

SCADA sunucuda WinCC programı çalıştırıldığında ilk olarak Şekil 6.13'deki "1.pdl" resim dosyası ekrana gelir. Şekil 6.13'deki ekran görüntüsü ana kullanıcı ekranıdır ve bütün denetim ve görüntüleme işlemleri bu ekran üzerinden gerçekleştirilir. Ekranın sağ alt köşesinde "Denetim Merkezi" bulunmaktadır. Denetim merkezinde bulunan butonlar kullanılarak, model vincin kovanı aşağıya, yukarıya, sağa ve sola hareket ettirilebilir. Operatör denetim merkezi üzerinde bulunan başlat, süre, alarm, ölçüm, uzak sistem ve ayarlar butonlarını kullanarak denetim algoritmalarını çalıştırabilir ve akım, gerilim ve pozisyon değerlerini görüntüleyebilmektedir.



Şekil 6.13. "1.pdl" resim dosyası ekran görünümü.

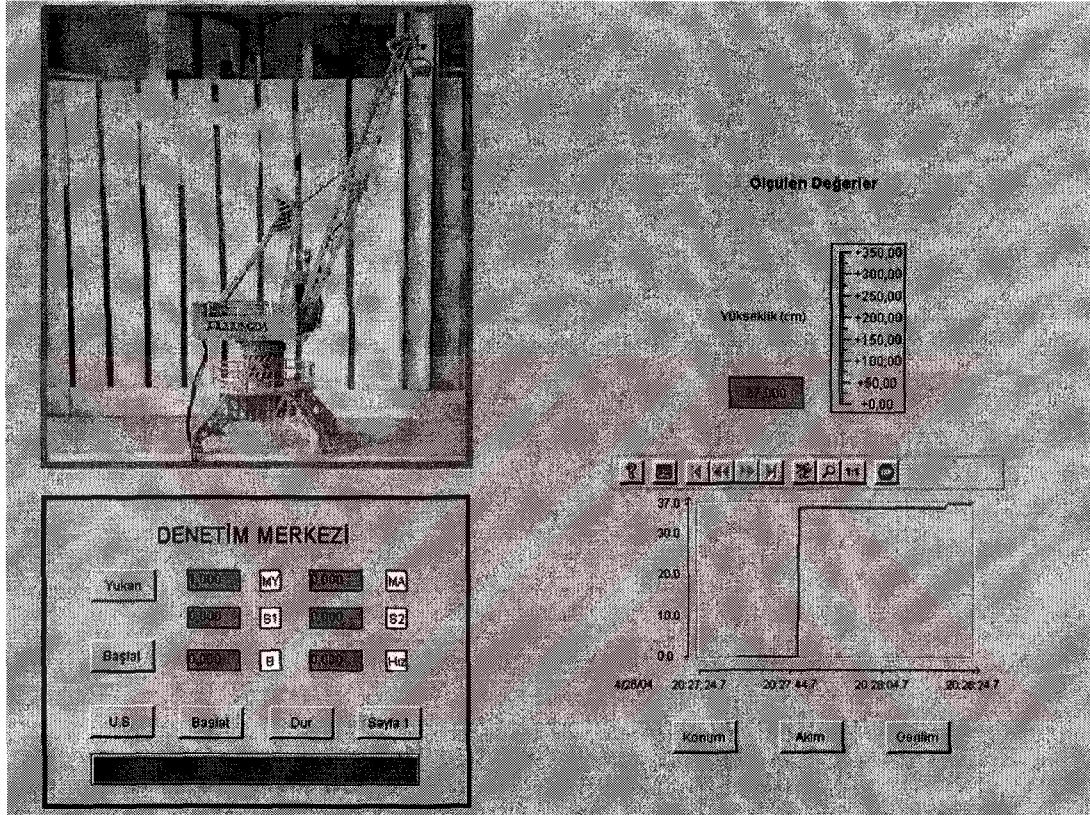
Şekil 6.14’de sahan ölçülen değerler görülmektedir. “1.pdl ve 4.pdl” resim dosyalarının beraber çalışması ile Şekil 6.14’deki ekran görüntüsünü meydana gelir. Ölçüm ekranını görüntüleyebilmek için denetim merkezinde bulunan “Ölçüm” butonuna basılması gerekir. Böylece, Şekil 6.14’deki ekrana görüntüsü elde edilir. Tekrar Şekil 6.13’deki ana ekran görüntüsüne elde edebilmek için denetim merkezindeki “Sayfa 1” butonuna basılması gerekir. Kullanıcı ekranının sağ tarafında, sırasıyla akım, pozisyon ve gerilim değerleri görülmektedir. Bu değerler, geçmişe dönük inceleme yapılabilmesi amacıyla “akım”, “pozisyon” ve “gerilim” arşivlerinde depolanmaktadır. Şekil 6.14’de görüntülenen akım, pozisyon ve gerilim değerleri, amper, piksel ve volt birimleri cinsindedir.



Şekil 6.14. Sahan ölçülen değerlerin görüntülenmesi.

Şekil 6.15’de test algoritmasının çalıştırıldığı ekran görüntüsü görülmektedir. “1.pdl ve 3.pdl” resim dosyalarının beraber çalışması ile Şekil 6.15’deki ekran görüntüsü meydana gelir. Test ekranını görüntüleyebilmek için, denetim merkezinde bulunan

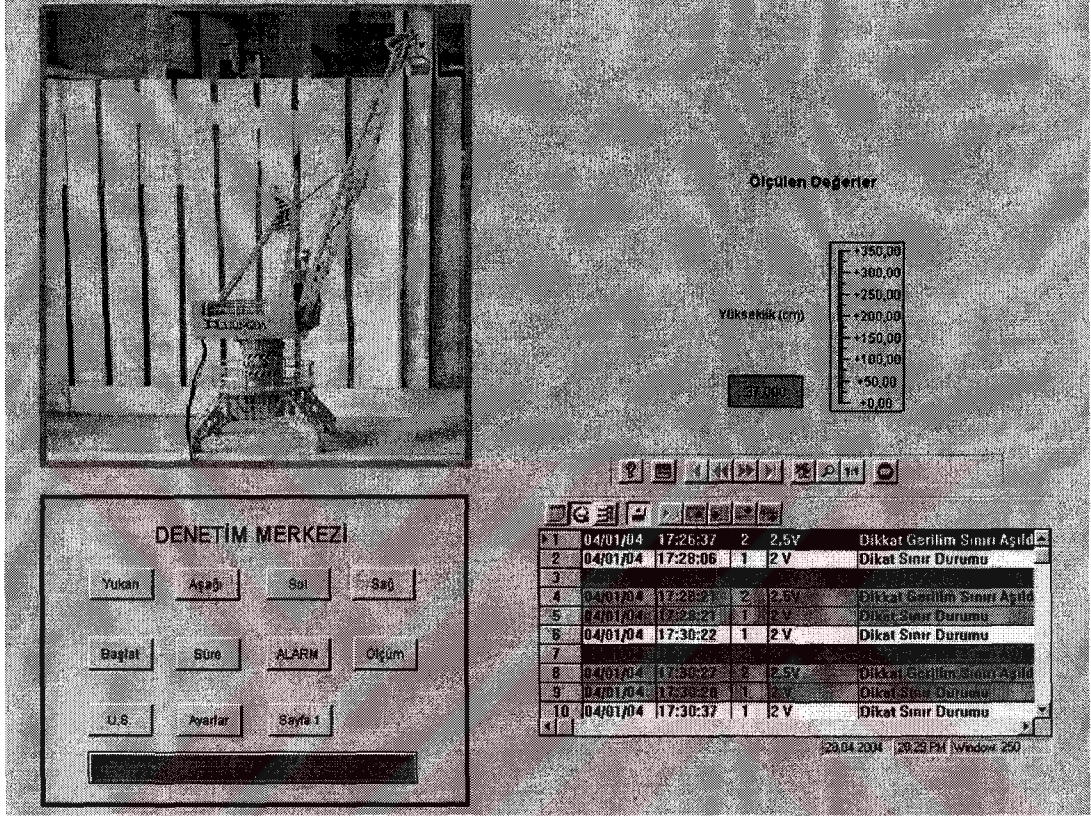
“Test” butonuna basılması ile Şekil 6.15’deki ekrana görüntüsü elde edilir. Ekranın sağ tarafı, ana ekran görüntüsü ile aynı olurken, sol alt köşesinde bulunan denetim merkezi bileşenleri değiştirilerek, test algoritmasının çalıştırılması için gerekli olan bileşenler burada görüntülenir. “Başlat” butonuna basılarak, test denetim algoritması çalıştırılır ve böylece sistemi denetlemek için gerekli olan değerler elde edilmiş olur.



Şekil 6.15. Test algoritmasının çalıştırılması.

Şekil 6.16’da hazırlanan test düzeneğinde meydana gelen alarmların görüntülediği ekran görüntüsü görülmektedir. “1.pdl ve 5.pdl” resim dosyalarının beraber çalışması ile Şekil 6.16’deki ekran görüntüsü meydana gelir. Alarm ekranını görüntüleyebilmek için denetim merkezinde bulunan “Alarm” butonuna basılması gerekir ve böylece alarm ekranı görüntülenir. Anlık ve geçmişe dönük olarak meydana gelen alarmlar, ekranın sağ alt köşesinde görüntülenir. Meydana gelen alarmlar ileriki bir zamanda incelenmek üzere “alarm” adlı arşive kaydedilir.

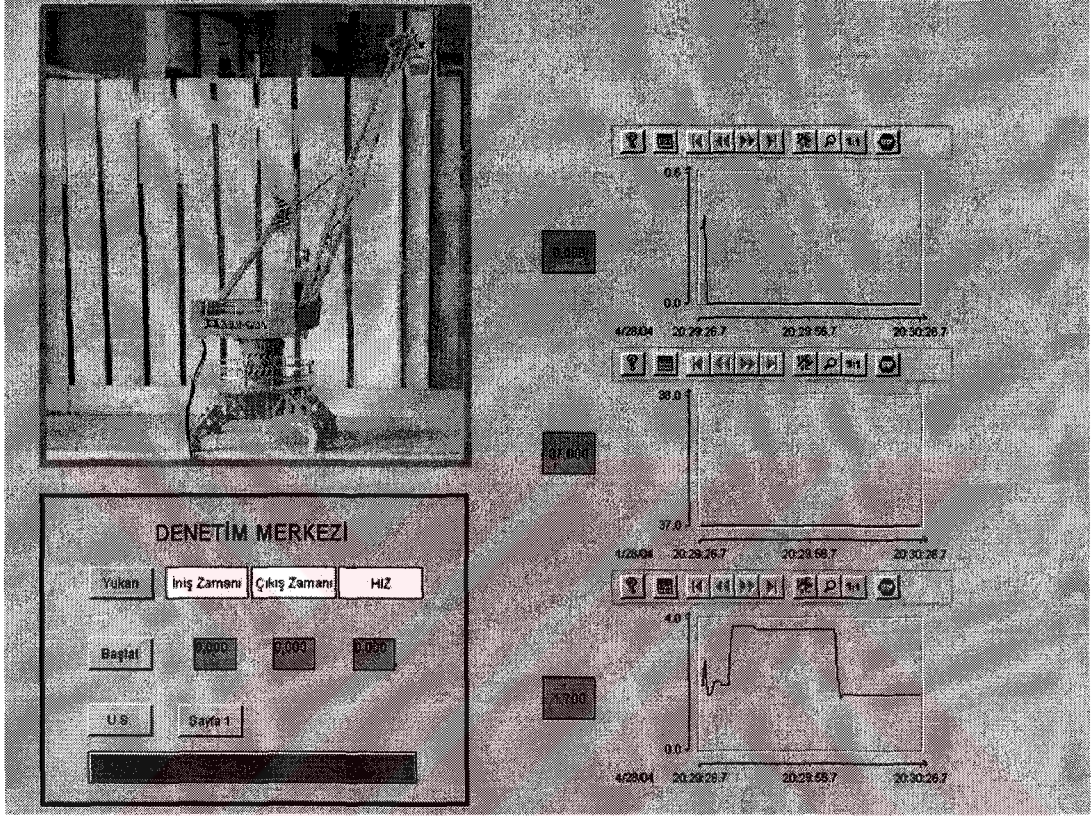
sistemde meydana gelen alarm ve arıza durumları arşivden incelenerek, arıza nedenleri araştırılabilir.



Şekil 6.16. Alarm ekranının görüntülenmesi.

Şekil 6.17’de ayar değerlerinin girildiği ekran görüntüsü görülmektedir. “1.pdl ve 2.pdl” resim dosyalarının beraber çalışması ile Şekil 6.17’deki ekran görüntüsü meydana gelir. Ayar değerlerinin girildiği ekranı görüntüleyebilmek için denetim merkezinde bulunan “Ayarlar” butonuna basılması gerekir. Böylece, Şekil 6.17’deki ekran görüntüsü elde edilir. Denetim merkezinde bulunan, “inişzamani”, “cikiszamani” ve “hız” metin kutularının altında bulunan I/O alanlarından, kova iniş, çıkış zamanı ve hız ayar değerleri girilir. I/O alanlarından girilen değerler “inişzamani”, “cikiszamani” ve “hız” etiketlerine atanır ve konum denetim algoritması bu değerleri kullanarak kova konumunu tespit eder. Ayar değerleri girildikten sonra “Sayfa 1” butonuna basılarak ana kullanıcı ekranı tekrar

görüntülenir. Böylece, hazırlanan beş ayrı resim dosyasıyla model vincin denetlenme ve görüntülenme işlemleri gerçekleştirilmiştir.

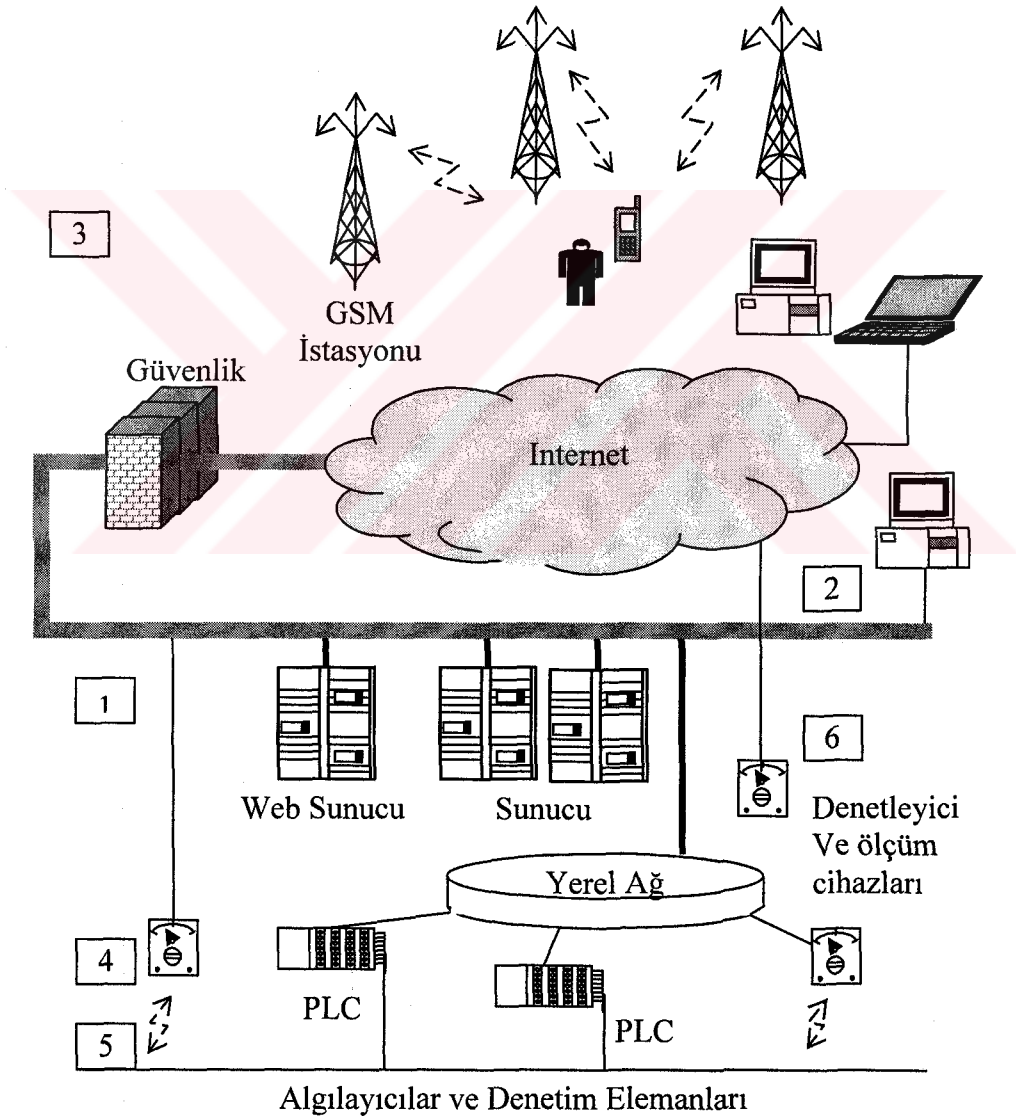


Şekil 6.17. Ayar değerlerinin girilişi.

Geliştirilen mobil tabanlı SCADA sistemi ile ilgili ekran görüntüleri bir sonraki bölümde detaylı biçimde açıklanmaktadır.

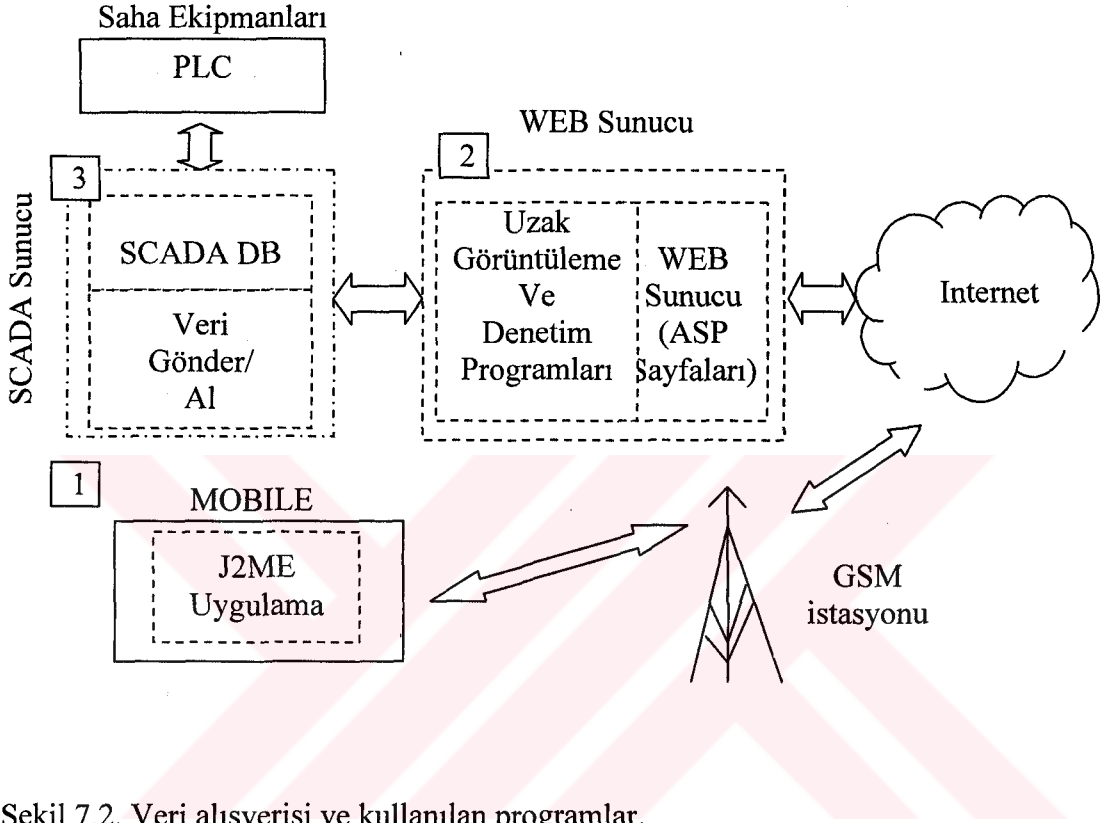
7. GELİŞTİRİLEN MOBİL TABANLI SCADA SİSTEMİ

Bölüm 3.8’de açıklanan klasik ve günümüz SCADA sistemleri yapısına bir yenilik olarak, SCADA sistemlerinin cep telefonları aracılığıyla görüntülenmesini ve denetlenmesini sağlayacak yeni mobil tabanlı bir yapı geliştirilmiştir. Şekil 7.1’de mobil tabanlı SCADA sisteminin yapısı görülmektedir.



Şekil 7.1. Mobil tabanlı SCADA sisteminin yapısı.

Mobil tabanlı SCADA sisteminde istemci olarak bilgisayar yada ince istemciler kullanılabilirdiği gibi cep telefonları da kullanılabilir. Şekil 7.2’de sahadaki sistem ile cep telefonu arasındaki veri alışverişi ve kullanılan programlar görülmektedir.



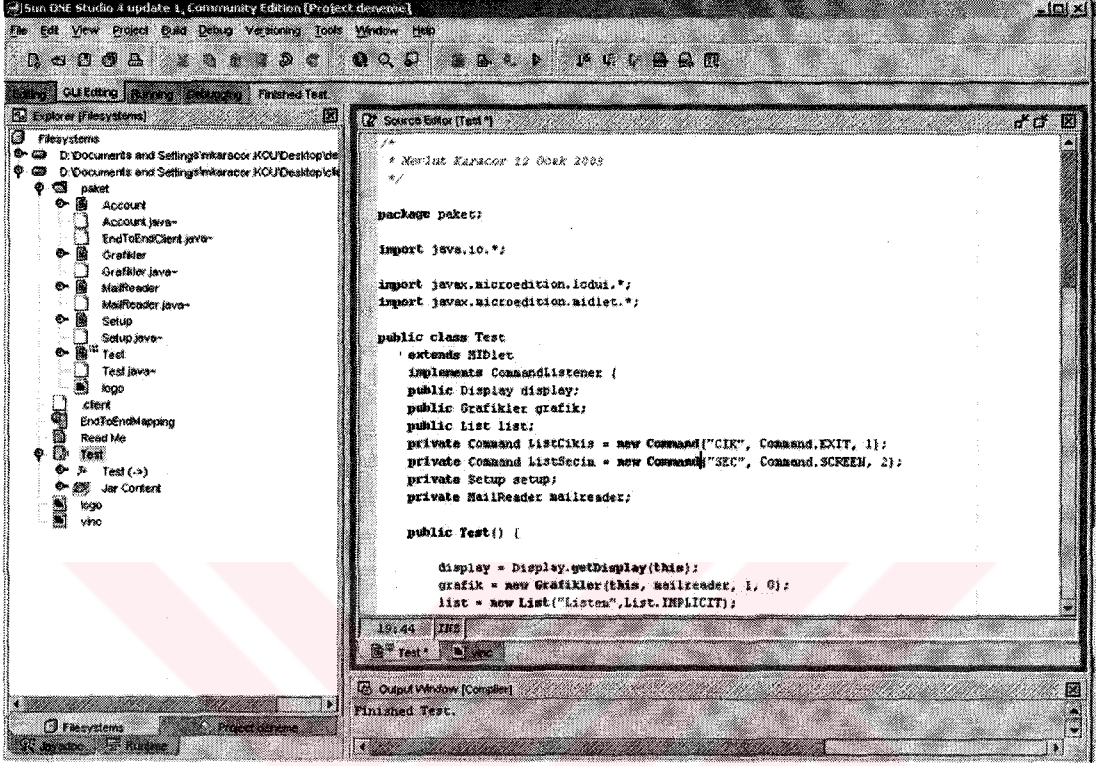
Şekil 7.2. Veri alışverişi ve kullanılan programlar.

Mobil tabanlı SCADA uygulamasını gerçekleştirmek için üç ayrı kısımda çalışmak üzere tasarlanmış üç ayrı program ve bir web sayfası hazırlanmıştır. Bu kısımlar ve programlar aşağıda açıklanmaktadır.

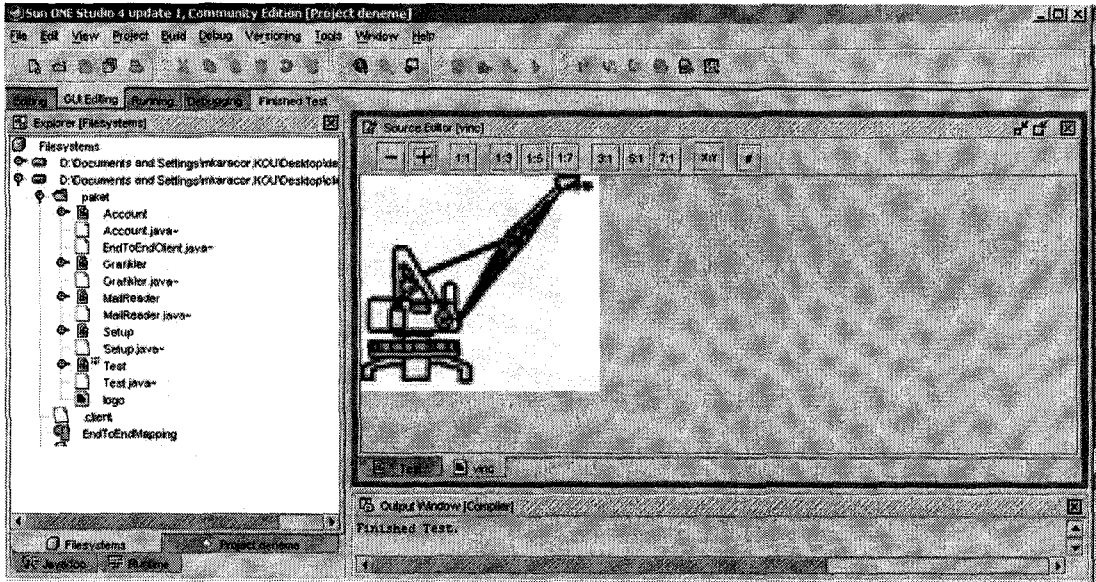
7.1. J2ME Uygulama Programı

Cep telefonundaki J2ME uygulama programı, GPRS veya WAP destekli, aynı zamanda JAVA uyumlu herhangi bir cep telefonunda çalıştırılmak üzere yazılmıştır. Bu programı yazmak için Sun Microsystems firmasına ait Sun One Studio 4 CE programı kullanılmıştır. J2ME (Java Micro Edition) uygulama programı, GPRS veya WAP protokolü kullanarak Internet üzerinde bulunan web sayfasına bağlanır. Gerekli olan verileri bu sayfadan şifre denetimi yapıldıktan sonra alır veya SCADA

sistemine web sayfası üzerinden gönderir. Şekil 7.3 ve Şekil 7.4’de J2ME uygulama programının hazırlanışı görülmektedir.



Şekil 7.3. Sun One Studio 4 CE programında cep telefonu programının yazılması.



Şekil 7.4. J2ME uygulama programına gerekli resimlerin ilave edilmesi.

Şekil 7.5’de hazırlanan J2ME uygulama programının WTK 20 simülasyon programında çalıştırılması görülmektedir.



Şekil 7.5. Hazırlanan J2ME uygulama programının WTK 20 simülasyon programında çalıştırılması.

Hazırlanan J2ME uygulama programı Java program kodları EK F’de verilmiştir. J2ME programı ile SCADA test düzeneği ile ilgili, web sayfasından altı adet farklı değişkenden çeşitli parametrik değerleri alınır ve web sayfasına üç adet parametrik değer gönderilir. J2ME uygulama programında, web sayfasından SCADA sisteminin ekran görüntüsü yerine, parametrik değerlerin alınması ve web sayfasına parametrik değerlerin gönderilmesinin aşağıdaki faydaları vardır. Bunlar;

i) Cep telefonu ile web sunucu arasında sadece 9 baytlık parametrik veri alışverişi gerçekleştirilir. 6 baytlık veri cep telefonunda SCADA ekranın oluşturulması için, 3 baytlık veri ise cep telefonu ile test düzeneğinin denetlenmesi için kullanılmıştır.

ii) Parametrik değerlerin kullanılması ile alışverişte kullanılan verinin boyutu azalmış ve uygulama programı daha kısa sürede SCADA sistemi hakkında bilgi toplayabilme imkanına kavuşmuştur.

iii) Kullanılan parametrik deęerler “1,0,37,2,0.5,1,1,1,1” gibi sayısal deęerlerdir. Kullanılan web sayfası, bilgisayar veya sistem dıřı bir cep telefonundan grntlense bile grntlenen veriler herhangi bir anlam ifade etmemektedir. Web sayfasından grntlenen verilere gerekli anlamı, hazırlanan J2ME uygulama programı vermektedir. Buda sistemin gvenlięi aısından olduka nemlidir. Bylece, J2ME uygulama programı olmadan sistemin grntlenebilme olasılıęı ortadan kalkar.

iv) Aktarılan verinin boyutunun kk olması sebebiyle GPRS zerinden sisteme baęlanan kullanıcıların telefon operatrlerine dedięi cret olduka azalmaktadır. nk, GPRS sisteminde telefon operatrleri aktarılan verinin boyutuna baęlı olarak cretlendirme yapmaktadırlar.

7.2. Web Sunucu Programı

Cep telefonu ile veri alışverişini gerekleřtirmek amacıyla aktif bir web sayfası hazırlanmıřtır. Aktif web sayfası herhangi bir server side script (ASP, PHP v.b.) veya kullanıcı sayısının fazla olması durumunda yksek performans gsteren Java Servlet (Server Applet) kullanılarak hazırlanabilir. SCADA sistemini denetlemek amacıyla ASP kullanılarak dinamik bir web sayfası hazırlanmıřtır. Hazırlanan web sayfası SCADA sistemi ile cep telefonu arasında bir kpr olma zellięine sahiptir. J2ME uygulama programından aldıęı verileri Uzak Denetim programı vasıtasıyla SCADA sistemine, uzak grntleme programından aldıęı veriyi J2ME uygulama programına gnderir. Bu veri trafięi ift ynl bir kprye benzemektedir. ASP ile hazırlanan web sayfası aldıęı veya gnderdięi verileri kendine ait bir veri tabanı zerinden gerekleřtirir. Kullanıcının web sayfasındaki verileri grntleyebilmesi iin kullanıcı ismi ve parolasını doęru olarak girmesi gerekir. Kullanıcı ismi ve parolası denetlendikten sonra kullanıcı, sistem bilgilerine ulařma imkanı kazanır.

7.3. SCADA Sunucu Programları

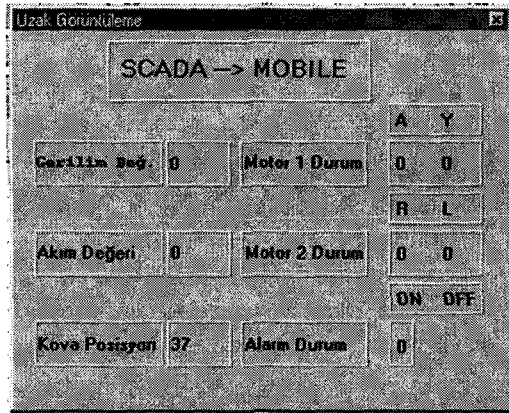
SCADA sunucusunda cep telefonundan grntleme ve denetleme iřlemlerini gerekleřtirmek amacıyla iki program yazılmıřtır.

7.3.1 Uzak Görüntüleme programı

Uzak görüntüleme programı SCADA veritabanından okuduğu altı adet veriyi, ASP sayfasına gönderme işlemini gerçekleştirir. Uzak görüntüleme programının, ASP sayfasına gönderdiği parametrik verilerin açıklaması Tablo 7.1’de verilmiştir. Şekil 7.6’da Delphi 6.0 da hazırlanan Uzak Görüntüleme programı görülmektedir.

Tablo 7.1. SCADA’dan ASP sayfasına gönderilen parametrik değerler.

Değişken Adı	Parametre	Açıklama
M1	0	Motor 1 Stop
M1	1	Motor 1 aşağı
M1	2	Motor 1 yukarı
M2	0	Motor 2 Stop
M2	1	Motor 2 sağ
M2	2	Motor 2 sol
konum	36-350	Kova konumu
gerilim	0-5	Sisteme uygulanan gerilim
akim	0-1	Sistemin çektiği akım
alarm	0	Alarm durumu yok
alarm	1	Alarm durumu gerçekleşti



Şekil 7.6. Uzak Görüntüleme programı.

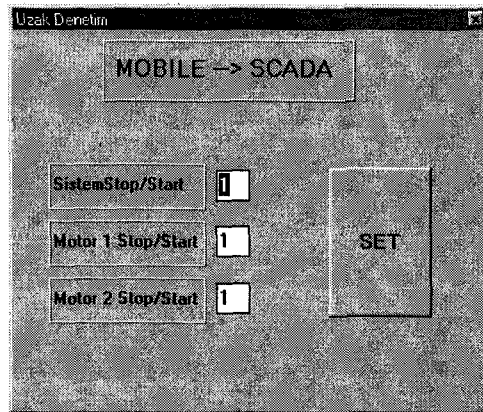
Görüntüleme amacıyla hazırlanan program Borland firmasına ait Delphi 6.0'da hazırlanmıştır. Görüntüleme programıyla WinCC veri tabanı arasındaki bağlantı MADO (Microsoft ActiveX Data Object) bileşeni ile sağlanmıştır.

7.3.2 Uzak Denetim programı

Uzak Denetim programı ASP sayfasından aldığı üç adet veriyi SCADA veritabanına kaydeder. Uzak Denetim programının, ASP sayfasından aldığı parametrik verilerin açıklaması Tablo 7.2'de verilmiştir. Şekil 7.7'de Delphi 6.0 da hazırlanan Uzak Denetleme programı görülmektedir.

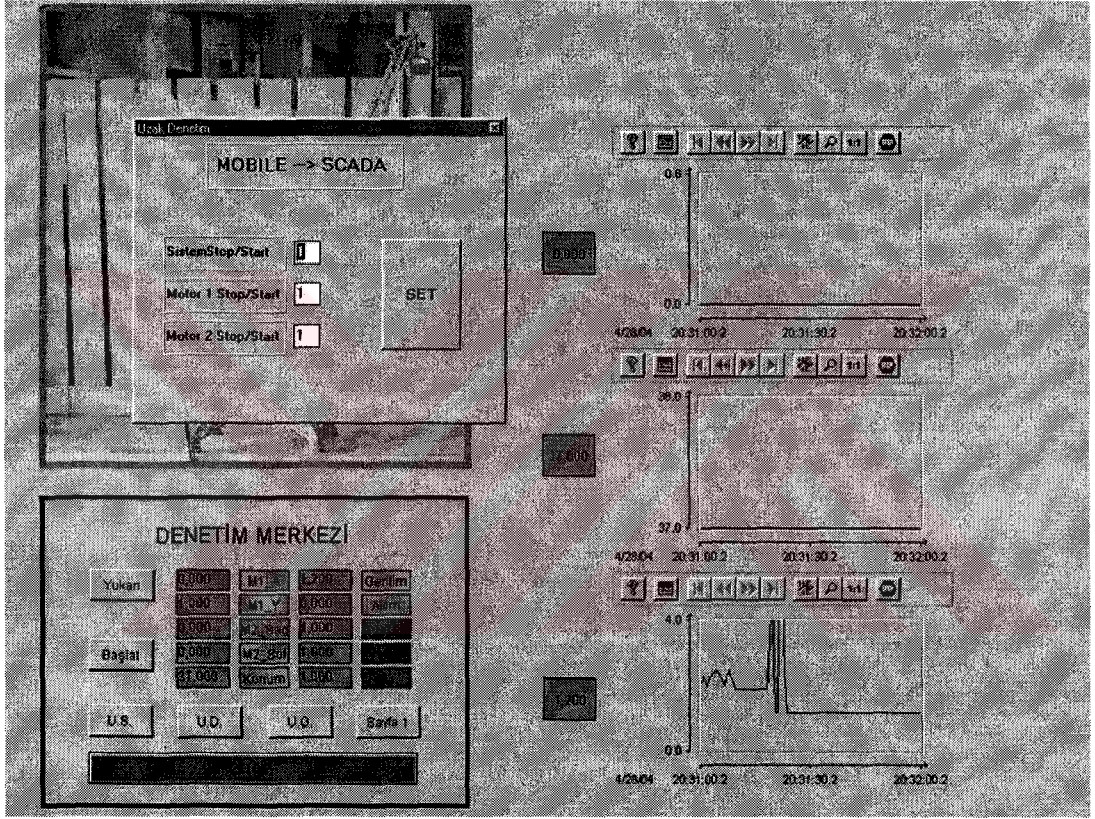
Tablo 7.2. ASP sayfasından SCADA'ya gönderilen parametrik değerler.

Değişken Adı	Parametre	Açıklama
Sistem	0	Sistem Stop
Sistem	1	Sistem Start
M1	0	Motor 1 Stop
M1	1	Motor 1 Start
M2	0	Motor 2 Stop
M2	0	Motor 2 Stop



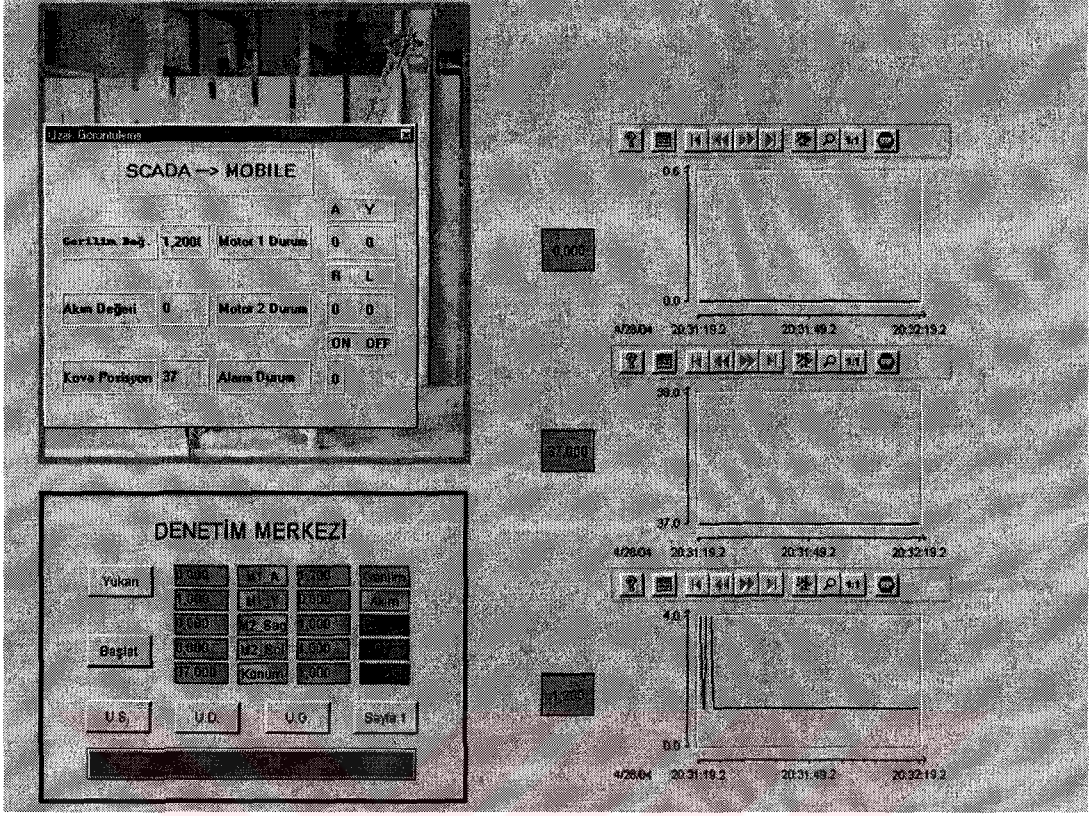
Şekil 7.7. Uzak Denetleme programı.

ve alarm durumu ile ilgili veriler hazırlanan ASP sayfasına gönderilmeye başlar. “UD” butonuna basılmasıyla uzak denetim programı çalışmaya başlar ve ASP sayfasında alınan veriler SCADA yazılımına aktarılır. Şekil 7.9’da Uzak Denetim programı ve WinCC SCADA programının beraber çalışması görülmektedir. Motor 1’e, motor 2’ye veya sisteme müdahale etmek için, uzak denetim programındaki I/O alanlarına 1 değeri yerine 0 değeri yazılarak “SET” butonuna basılması gerekir. Böylece, sistem veya motorlar uzaktan durdurulabilir veya tekrar devreye alınabilir.



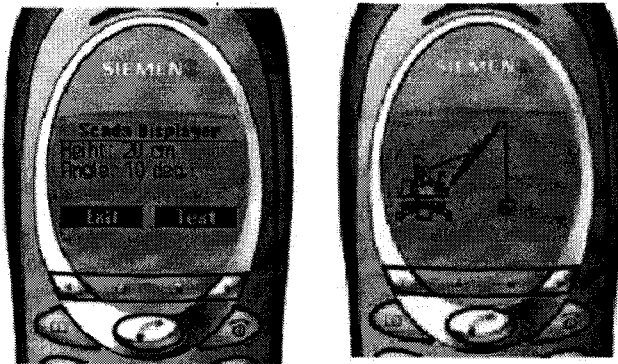
Şekil 7.9. Uzak Denetim programı ve WinCC SCADA programının beraber çalışması.

Şekil 7.10’da Uzak Görüntüleme programı ve WinCC SCADA programının beraber çalışması görülmektedir. Uzak görüntüle programı üzerinden ASP sayfasına gönderilen verilere kullanıcının herhangi bir değişiklik yapabilme olanağı yoktur. Sistem gönderilen verilere doğrudan erişerek otomatik olarak veri gönderme işlemini gerçekleştirir.



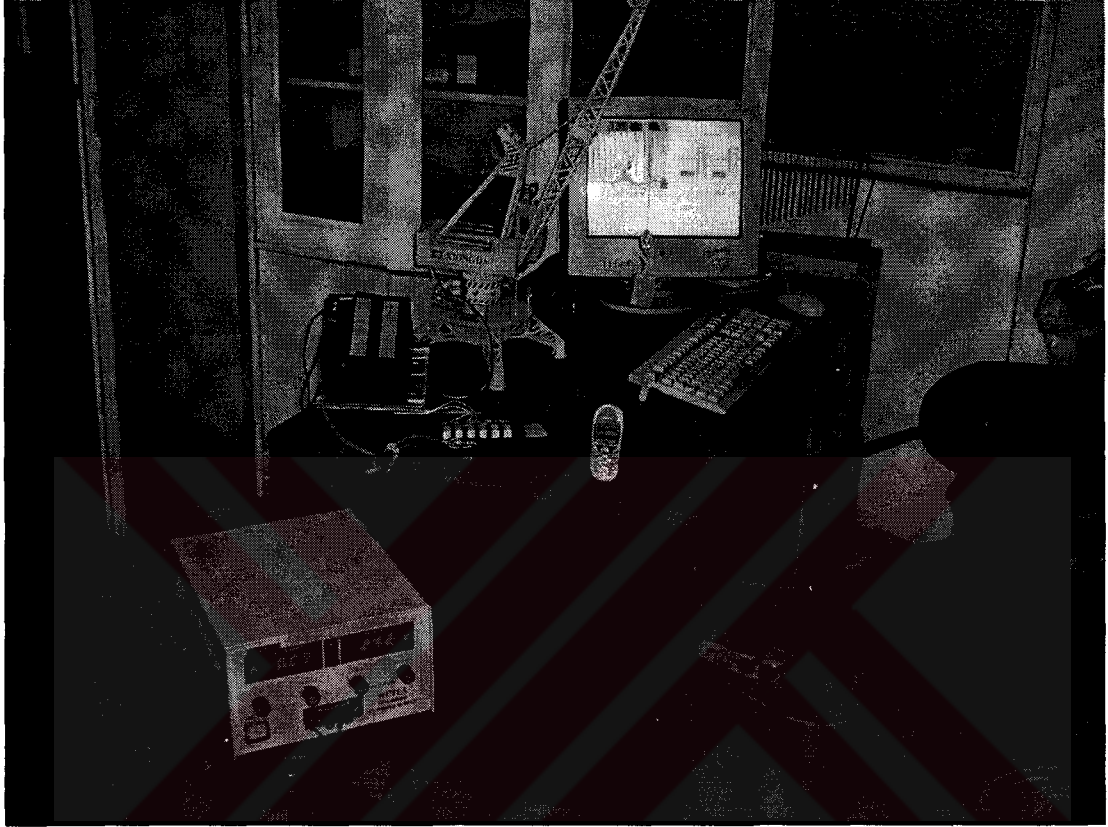
Şekil 7.10. Uzak Görüntüleme programı ve WinCC SCADA programının beraber çalışması.

Mobil tabanlı SCADA uygulama programı, sistemi test etmek için 228Kb kullanıcı hafızasına sahip ve Java uyumlu Siemens M50 cep telefonuna yüklenmiştir. Şekil 7.11'de Siemens M50 cep telefonunda J2ME uygulama programının çalışması görülmektedir.



Şekil 7.11. J2ME uygulama programının çalışması.

Şekil 7.12’de gerçekleştirilen mobil tabanlı SCADA sistemi görülmektedir. Mobil tabanlı SCADA sistemi, bilgisayar, cep telefonu, PLC, güç kaynağı ve model vinç kullanılarak test edilmiştir.



Şekil 7.12. Gerçekleştirilen mobil tabanlı SCADA sistemi test çalışması.

7.5. Mobil Tabanlı SCADA Sisteminin Ekonomik Açından İncelenmesi

Klasik bir SCADA sistemini, mobil tabanlı bir SCADA sistemine dönüştürebilmek için yüksek maliyetli bir yatırım gerekmemektedir. Mobil SCADA uygulaması için;

i) Web sunucu, SCADA sistemi ile mobil cihazlar (cep telefonu ve PDA) arasındaki veri trafiğini gerçekleştirmek için gereklidir. Yedi gün yirmi dört saat çalışabilecek bir sunucu maliyeti ortalama 2000\$’dır.

ii) Web sunucunun Internet ortamı ile veri alışverişini gerçekleştirebilmesi için Internet bağlantısı gerekmektedir. 128 kbit’lik internet bağlantısı maliyeti aylık ortalama 42 \$’dır.

iii) Hazırlanan J2ME uygulama programını çalıştırmak için Java uyumlu bir cep telefonu gereklidir. Uygulama Siemens M50 cep telefonu ile gerçekleştirilmiş ve kullanıcı başına ortalama maliyeti 100 \$ civarındadır.

iv) Son maliyet ise GSM operatörünün alt yapısının kullanımından kaynaklanan işletim gideridir. Kullanıcı başına ortalama maliyet 0,06 cent/kbyte'tır.

Kullanılan cihazlara ve hizmetlere ait giderler dolar cinsinden fiyatlandırılmıştır. İlk iki gider firma tarafından son iki gider ise kullanıcılar tarafından karşılanan giderlerdir. Mobil tabanlı bir SCADA sistemi, internet tabanlı bir SCADA sistemi ile karşılaştırıldığında çok daha ekonomiktir.

Günlük dial-up bağlantı toplam işletme maliyet (DBTİM) gideri, saat başına ortalama dial-up bağlantı maliyeti (DBM) 71 cent olarak alındığında aşağıdaki gibi bulunur;

$$DBTİM=24*DBM \quad (7.1)$$

Eşitlik 7.1'den günlük toplam maliyet 17 \$ olarak bulunur.

ADSL modem ile 128 kbit'lik internet bağlantısı günlük işletim gideri ise yaklaşık 1.41\$'dir. Yapılan çalışmada ortalama 10 sn'yede bir 9 byte'lık sisteme ait veri alışverişi gerçekleştirilir. Buna göre, günlük işletim gideri aşağıdaki gibi bulunur.

$$GBTİM=(9/1024)*10*6*60*24*GBM \quad (7.2)$$

Mobil SCADA sisteminde, veri alışverişinde GPRS internet bağlantısı kullanılmıştır. GPRS internet bağlantı hizmetinde ücretlendirme kbyte cinsinden alınan ve gönderilen verinin boyutuna bakılarak gerçekleştirilir. Aycell GSM operatöründe GPRS hizmeti (GBM) kbyte başına 67 cent ve Aria GSM operatöründe ise kbyte başına 85 cent'dir. Mobil cihaz ile sisteme yirmi dört saat süresince bağlanıldığında toplam 760 kbyte veri alışverişi gerçekleştirilmiştir. Eşitlik 7.2'den GPRS bağlantı toplam işletme maliyet (GBTİM) gideri hesaplandığında Aycell GSM operatöründe 50 cent, Aria GSM operatöründe 65 cent'dir.

Sonuç olarak kullanıcı işletim gideri açısından bakıldığında, mobil SCADA sistemi dial-up bağlantıya göre 26 ila 33 ve ADSL modem ile bağlantıya göre 2,2 ila 2,8 kat daha ekonomiktir. Tablo 7.3’de mobil istemci ile internete bağlantılı bilgisayara ait günlük işletim maliyetlerin karşılaştırılması verilmektedir.

Tablo 7.3. Farklı uygulamalara ait günlük bağlantı maliyetlerinin karşılaştırılması.

İstemci	Günlük Maliyet	Açıklama
Bilgisayar, dial-up bağlantı.	17 \$	26-33 kat daha ekonomik.
Bilgisayar, ADSL modem ile bağlantı.	1.41\$	2,2-2,8 kat daha ekonomik.
Cep telefonu, GPRS ile bağlantı.	50-65 Cent	Mobil tabanlı SCADA sistemi.

8. SONUÇLAR

Tez çalışmasında, günümüz SCADA sistemlerinin ulaştığı son noktalar açıklanmış ve ilk olarak Java uyumlu bir cep telefonu istemci olarak kullanılmış ve mobil tabanlı örnek bir SCADA uygulaması gerçekleştirilmiştir. Mobil tabanlı SCADA sistemi ile, yerel bir bilgisayardan veya internete bağlı bir bilgisayardan bağımsız olarak, uzak noktalardan veya yerel noktalardan sahadaki sürecin kolayca görüntülenmesi ve denetlenmesi sağlanmıştır.

Aynı zamanda, Mobil tabanlı SCADA sistemini gerçekleştirmek için var olan GSM baz istasyonlarının kullanılmasından dolayı, görüntüleme ve denetleme işlemleri için herhangi bir ağ kurulum maliyeti yoktur. Veri alışverişi GPRS veya WAP üzerinden yapılmakta ve bu yüzden iletişim masrafları, klasik modem bağlantıyla karşılaştırıldığında oldukça düşüktür. Bu maliyet hazırlanan test uygulaması için, 24 saat internete bağlı bir bilgisayar ile GPRS üzerinden SCADA sistemine bağlı bir cep telefonunun iletişim giderleri arasında yaklaşık 26 ila 33 katlık bir oran bulunmaktadır. ADSL modem ile bağlantıya göre mobil SCADA 2,2 ila 2,8 kat daha ekonomiktir.

Mobil istemci ile SCADA sistemi arasındaki veri alışverişi parametrik olarak gerçekleştirilmiştir. Veri aktarımında kullanılan web sayfası, bilgisayar veya sistem dışı bir cep telefonundan görüntülense bile, veriler herhangi bir anlam ifade etmemektedir. Web sayfasından görüntülenen verilere, sistemin güvenliği açısından, gerekli anlamı hazırlanan J2ME uygulama programı vermektedir. Böylece, J2ME uygulama programı olmadan sistemin görüntülenebilme olasılığını ortadan kaldır. Uygulama programı cep telefonuna, kablo, infared veya internet vasıtasıyla yüklenebilmektedir.

KAYNAKLAR

1. KOSKINEN, P., 2000. Decision-making process on field technology for process management, University of Oulu, 48-52, Oulu, Finland.
2. CEGRELL, T., 1997. Integrated Information and Control Systems, DA/DSM 97 Europe Conference, Amsterdam.
3. ScanData Remote monitoring and control, 1994. Telemetry & SCADA Handbook p.1-6.
4. BURCKHART, J., 1998. Detector control system. LEB98 Workshop, Rome, Italy.
5. BLEVINS, T., KINNEY, T. PELUSO, M., 1997. How Fieldbus can influence your next Project. Control Engineering, 43, 133, Barrington.
6. MORLEY, D., 1995. Ladder logic, sci-fi wise. Manufacturing Systems, 13, 14, Wheaton. ISSN/ISBN: 0748948X
7. Western Area Power Administration, 1997. How programmable logic controllers open new automation opportunities. Electrical World, 211, 46, New York. ISSN/ISBN: 00134457
8. GYORKI, J., 1995. PLCs drive standard buses. Machine Design, 67, 83, Cleveland. ISSN/ISBN: 00249114
9. Anonim, 1996. What's better, PLCs or PCs?. Machine Design, 68, 164, Cleveland. ISSN/ISBN: 00249114
10. LANGNAU, L., 1995. PC vs. PLC: Selection rules are changing. Material Handling Engineering, 50, 31, Cleveland. ISSN/ISBN: 00255262
11. EFFNER, J., 1998. PLC vs. PC machine controls. FDM, Furniture Design & Manufacturing, 70, 40, Chicago. ISSN/ISBN 01928058
12. LANGNAU, L., 1998. To PC or to PLC?. Material Handling Engineering, 53, 26, Cleveland. ISSN/ISBN: 00255262
13. DAVID, T., 2000. Back to basics on PLCs. Electrical Apparatus, 53, 39, Chicago. ISSN/ISBN 01901370
14. HOWARD, J., 1988. An Economical Solution to SCADA. Communications, 25, 5.

15. ALLEN, K., YUNG, L., 2000. Inheritance and Polymorphism in real time monitoring and control systems. *Journal of Intelligent Manufacturing*, 11, 285.
16. BOYER, S., 1993. SCADA: Supervisory Control and Data Acquisition. Instrument Society of America, Research Triangle.
17. Synchrony, 2001. Trends in SCADA for Automated Water Systems.
18. Asergei Electrical Engineering, 2002. <http://home.delfi.ee/asergei/eng/rtu.htm>.
19. Frank Electric Corp, <http://www.frankelectric.com/remote.html>.
20. PC in bridge, 2002. <http://www.iclinks.com/Products/ICL4300/ICL4300.html>.
21. Live Video, <http://www.sagedesignsinc.com/>
22. SMITH, L., WAYNE, B., 1993. RTUs Slave for Supervisory Systems. *IEEE Computer Applications in Power*, 27, 32.
23. LEKAS, S., 1999. Making Ethernet work. Special Section, Sensors Online.
24. CLARKSON, M., 1999. February 1999 Shop Floor To The Top Flor Enterprise Wide Connectivity. Special Section, IOtech Inc.
25. Data Comm for Business Inc., <http://www.dcbnet.com>.
26. Bentek Systems SCADA & Telemetry Solutions., <http://www.scadalink.com/>.
27. ICL4300, <http://www.iclinks.com/Products/IntelligentRTU/ICL4300.html>.
28. Globalstar, <http://www.qualcomm.com>.
29. STEPHAN, J. Scada Systems In Wastewater Treatment. Principal Process-Logic.
30. BOYER, S., 2002. The long arm of the SCADA. *InTech*, 49, 57, Durham. ISSN/ISBN: 0192303X
31. PARK, J., 2002. Modern SCADA technology. *Energy Markets*, 7, 48, Houston. ProQuest document ID: 123618861
32. Anonim., 1998. Java programmable SCADA products. *Pipeline & Gas Journal*, 225, 107, Dallas. ISSN/ISBN: 00320188
33. Anonim, 1996. PC-based SCADA gains ground. *Control Engineering*, 43, 82, Barrington. ISSN/ISBN: 00108049
34. COATES, T., 2003. HMI for stand-alone or integrated OEM applications. *Control Engineering*, 50, 26. ISSN/ISBN: 00108049

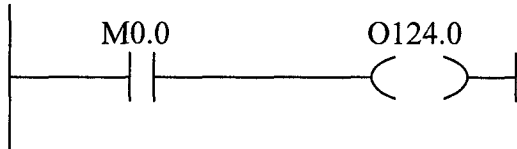
35. JOHNSON, D., SCADA software raises the stakes. Control Engineering, 43, 80, Barrington. ISSN/ISBN: 00108049
36. Anonim, 1997. Soft logic and SCADA. Control Engineering, 44, 84, Barrington. ISSN/ISBN: 00108049
37. MELVIN, H., SHEARER, A., 2000. GPS World. 11, 30, Cleveland. ISSN/ISBN: 10485104
38. STROTHMAN, J., 2002. HMI/SCADA meets the Web--and likes it!. InTech, 49, 31, Durham. ISSN/ISBN: 0192303X
39. Factory Link 7, <http://www.tecnomatix.com/>.
40. Paradym-31, <http://www.synapsesystem.com/paradym31.html>.
41. WizFactory, <http://www.conlab.com.au/axeda.html>.
42. Cimplicity Plant Edition, <http://www.gefanuc.com/>.
43. Genesis32, <http://www.iconics.com/>.
44. Intellution Dynamics, <http://www.intellution.com/>.
45. LabView, <http://www.ni.com/>.
46. HMI/SCADA Paragon, www.nematron.com/.
47. FactoryFloor Software, <http://www.opto22.com/>.
48. RSView32, <http://www.software.rockwell.com/>.
49. KIRRMANN, H., 2003. Field Bus Types, EPFL Industrial Automation. ABB Research Center, Baden, Switzerland.
50. KIRRMANN, H., 2003. Field Bus Standarts, EPFL Industrial Automation. ABB Research Center, Baden, Switzerland.
51. PINTO, J., 2000. Fieldbus - Conflicting "Standards" Emerge, but Interoperability is Still Elusive. Design Engineering, UK.
52. LIAN, S., 2003. ASI Master Bus Controller. University of Queensland, Australia
53. INTERBUS, <http://www.interbusclub.com/>
54. PROFIBUS, <http://www.profibus.org>.

55. PINTO, J., 2000. How do I catch the Fieldbus? . ISA Worldbus Journal - Supplement to InTech.
56. FOUNDATION Fieldbus Technical Overview, <http://www.fieldbus.org>.
57. Siemens SIMATIC HMI WinCC Web Navigator V1.1 Manual Edition, Nisan 2001.
58. Siemens SIMATIC HMI WinCC Manual Volume 1/2, Ağustos 1999.
59. Siemens SIMATIC HMI WinCC Manual Volume 2/2, Ağustos 1999.
60. Siemens WinCC Configuration Manual Manual Volume 1, Eylül 1999.
61. Siemens WinCC Configuration Manual Manual Volume 2, Eylül 1999.
62. Siemens WinCC Configuration Manual Manual Volume 3, Eylül 1999.
63. Siemens WinCC Communication Manual Manual 1, Eylül 1999.
64. Siemens WinCC Communication Manual Manual 2, Eylül 1999.
65. GOMEZ, J., 2002. Survey of SCADA SYSTEMS and visualization of a real life process. Linköping University, Sweden.

EK A

Siemens Simatic S7-300 CPU 312 IFM PLC yazılımı.

NETWORK 1



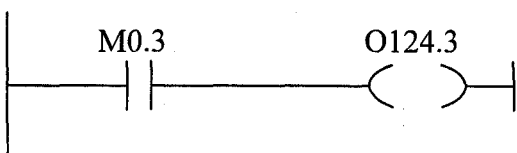
NETWORK 2



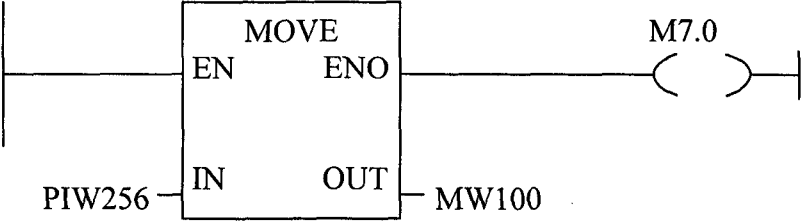
NETWORK 3



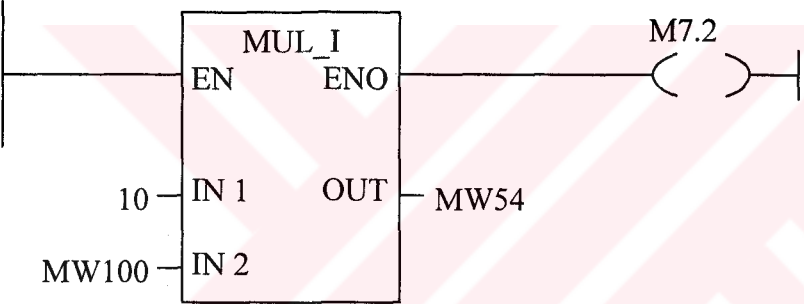
NETWORK 4



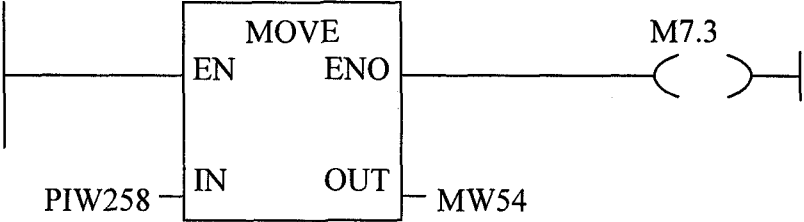
NETWORK 5



NETWORK 6



NETWORK 7



EK B

Hız Tespit Algoritması.

```
#include "apdefap.h"
long _main(char* lpszPictureName, char* lpszObjectName, char*
lpszPropertyName)
{
#include "dmclient.h"
#pragma code ("kernel32.dll")
VOID WINAPI Sleep (DWORD dwMilliseconds);
#pragma code()
static int i,k;
double artirim,artirim1,hiz;

if(GetTagBit("Sayfa3_baslat")==1)
{

// Birinci Asma Kova Yukarı
if (( GetTagBit ("yedek") ==0) & (GetTagBit ("yedek1") == 0) & (GetTagBit
("yedek2") ==0) & (GetTagBit ("yedek3") ==0) & (GetTagBit ("yedek4") ==0) &
(GetTagBit("yedek5")==0)&(GetTagBit("yedek6")==0))
{
SetTagBit("M1_Y",1);
if(GetTagBit("UST_SINIR")==1)
{
SetTagBit("M1_Y",0);
SetTagBit("yedek",1);
}
}

// İkinci Aşama

if ((GetTagBit ("yedek") ==1) & (GetTagBit ("yedek1") == 0 ) & (GetTagBit
("UST_SINIR")==1))
{
SetTagBit("M1_A",1);
SetTagBit("yedek1",1);
}

if((GetTagBit ("yedek1") ==1 ) & (GetTagBit ("yedek2") == 0 ) & (GetTagBit
("UST_SINIR ") == 0))
{
artirim=0;
for(i=0;i<2000;i++)
{
Sleep(100);
if (GetTagBit("UST_SINIR")==1)
```

```

        {
            SetTagBit("yedek2",1);
            SetTagDWord("sayac1",i);
            SetTagBit("M1_A",0);
            break;
        }
        SetTagDWord("sayac1",i);
    }
}

// Üçüncü Aşama
if((GetTagBit ("yedek2") ==1) & (GetTagBit ("yedek3") == 0 ) & (GetTagBit
("UST_SINIR") ==1))
{
    SetTagBit("M1_Y",1);
    SetTagBit("yedek3",1);
}
if ((GetTagBit ("yedek3") == 1) & (GetTagBit ("yedek4") == 0) & (GetTagBit
("UST_SINIR")==0))
{
    for(i=0;i<2000;i++)
    {
        Sleep(100);
        if (GetTagBit("UST_SINIR")==1)
        {
            SetTagBit("yedek4",1);
            SetTagDWord("sayac2",i);

            artirim=GetTagDWord("sayac1");
            artirim1=GetTagDWord("sayac2");

            hiz=artirim1+artirim;
            SetTagDWord("HIZ",hiz/4);
            SetTagBit("M1_Y",0);
            break;
        }
        SetTagDWord("sayac2",i);
    }
}
}
return 200;
}

```

EK C

Konum Denetim Algoritması.

```
#include "apdefap.h"
long _main(char* lpszPictureName, char* lpszObjectName, char*
lpszPropertyName)
{
#include "dmclient.h"
#pragma code ("kernel32.dll")
VOID WINAPI Sleep (DWORD dwMilliseconds);
#pragma code()
static int i,konumXY,hiz,sinir,cikis zamani,inis zamani;
double arttirim, arttirim1,arttirim2;
float akim, akim_dönüşüm,gerilim, gerilim_dönüşüm;
```

```
inis zamani=GetTagDWord("inis zamani");
cikis zamani=GetTagDWord("cikis zamani");
```

```
//Alarm Durumu
```

```
if(GetTagDWord("gerilim")>=25)
{
SetTagBit("gerilim_tehlike",1);
SetTagBit("sayfa4",0);
SetTagBit("M1_A",0);
SetTagBit("M1_Y",0);
SetTagBit("M2_sag",0);
SetTagBit("M2_sol",0);
SetTagBit("Y",0);
SetTagBit("A",0);
}
if(GetTagDWord("gerilim")<=24)
{
SetTagBit("gerilim_tehlike",0);
}
```

```
// Akım Dönüşüm Rutini
```

```
if(GetTagBit("set")==1)
{
akim=GetTagDWord("akim");
akim_dönüşüm=akim/20;
SetTagFloat("akim_dönüşüm",akim_dönüşüm);
}
```

```
// Gerilim Dönüşüm Rutini
```

```

if(GetTagBit("set")==1)
{
    gerilim=GetTagDWord("gerilim");
    gerilim_dönüsüm=gerilim/10;
    SetTagFloat("gerilim_dönüsüm",gerilim_dönüsüm);
}

```

// Aşağı Denetim

```

if((GetTagBit("A")==1))
{
    SetTagBit("M1_A",0);
    if((konumXY<=350))
    {
        SetTagBit("M1_A",1);
    }
    konumXY=GetTagDWord("kovaXY");
    hiz=GetTagDouble("HIZ");
    arttirim=350/(hiz);
    arttirim1=3500/(hiz);
    if((arttirim1-arttirim*10)>0)
    {
        arttirim=arttirim+1;
    }
    if(hiz>=350)
    {
        arttirim=1;
    }
    i=GetTagDWord("i");
    for (i;i<=350;i++)
    {

```

//Alarm Durumu

```

if(GetTagDWord("gerilim")>=25)
{
    SetTagBit("gerilim_tehlike",1);
    SetTagBit("sayfa4",0);
    SetTagBit("M1_A",0);
    SetTagBit("M1_Y",0);
    SetTagBit("M2_sag",0);
    SetTagBit("M2_sol",0);
    SetTagBit("Y",0);
    SetTagBit("A",0);
}

```

```

if(GetTagDWord("gerilim")<=24)
{

```

```

        SetTagBit("gerilim_tehlike",0);
    }

// Akım Dönüşüm Rutini
    if(GetTagBit("set")==1)
    {
        akim=GetTagDWord("akim");
        akim_dönüşüm=akim/20;
        SetTagFloat("akim_dönüşüm",akim_dönüşüm);
    }

// Gerilim Dönüşüm Rutini

    if(GetTagBit("set")==1)
    {
        gerilim=GetTagDWord("gerilim");
        gerilim_dönüşüm=gerilim/10;
        SetTagFloat("gerilim_dönüşüm",gerilim_dönüşüm);
    }

    konumXY=konumXY+arttirim;
    SetTagDouble("i",i);
    arttirim2=(arttirim-1);
    arttirim1=(3500-10*arttirim*i)/(hiz-i);

    if((arttirim1-(arttirim2*10))==0)
    {
        arttirim=arttirim-1;
    }

    if((konumXY>=350))
    {
        SetTagDouble("kovaXY",350);
        break;
    }

    Sleep(iniszamani);

    if((GetTagBit("A")==0))
    {
        SetTagDouble("kovaXY",konumXY);
        break;
    }

    SetTagDouble("kovaXY",konumXY);
}
}

```

```
// Yukarı denetim
```

```
if(GetTagBit("Y")==1)
```

```
{  
    SetTagBit("M1_Y",0);  
    if((konumXY>=36))  
    {  
        SetTagBit("M1_Y",1);  
    }  
}
```

```
if(GetTagBit("UST_SINIR")==1)
```

```
{  
    SetTagBit("M1_Y",0);  
}
```

```
konumXY=GetTagDWord("kovaXY");
```

```
hiz=GetTagDouble("HIZ");
```

```
arttirim=350/(hiz);
```

```
arttirim1=3500/(hiz);
```

```
if((arttirim1-arttirim*10)>0)
```

```
{  
    arttirim=arttirim+1;  
}
```

```
i=GetTagDWord("i");
```

```
for (i;i<=350;i--)
```

```
{
```

```
//Alarm Durumu
```

```
if(GetTagDWord("gerilim")>=25)
```

```
{  
    SetTagBit("gerilim_tehlike",1);  
    SetTagBit("sayfa4",0);  
    SetTagBit("M1_A",0);  
    SetTagBit("M1_Y",0);  
    SetTagBit("M2_sag",0);  
    SetTagBit("M2_sol",0);  
    SetTagBit("Y",0);  
    SetTagBit("A",0);  
    break;  
}
```

```
if(GetTagDWord("gerilim")<=24)
```

```
{  
    SetTagBit("gerilim_tehlike",0);  
}
```

```
// Akım Dönüşüm Rutini
```

```
if(GetTagBit("set")==1)
{
    akim=GetTagDWord("akim");
    akim_dönüşüm=akim/20;
    SetTagFloat("akim_dönüşüm",akim_dönüşüm);
}
```

```
// Gerilim Dönüşüm Rutini
```

```
if(GetTagBit("set")==1)
{
    gerilim=GetTagDWord("gerilim");
    gerilim_dönüşüm=gerilim/10;
    SetTagFloat("gerilim_dönüşüm",gerilim_dönüşüm);
}
```

```
konumXY=konumXY-arttirim;
SetTagDouble("i",i);
arttirim2=(arttirim-1);
arttirim1=(3500-10*arttirim*i)/(hiz-i);
```

```
if((arttirim1-(arttirim2*10))==0)
{
    arttirim=arttirim-1;
}
```

```
if((konumXY<=35))
{
    SetTagDouble("kovaXY",35);
    SetTagBit("M1_Y",0);
    break;
}
```

```
Sleep(cikiszamani);
```

```
if((GetTagBit("Y")==0))
{
    SetTagDouble("kovaXY",konumXY);
    SetTagBit("M1_Y",0);
    break;
}
```

```
SetTagDouble("kovaXY",konumXY);
```

```
}
}
```



```

// Başlatma Rutini

if((GetTagBit("UST_SINIR")==0)&(GetTagBit("s2")==1))
{
    SetTagBit("M1_Y",1);

//Alarm Durumu

    if(GetTagDWord("gerilim")>=25)
    {
        SetTagBit("gerilim_tehlike",1);
        SetTagBit("sayfa4",0);
        SetTagBit("M1_A",0);
        SetTagBit("M1_Y",0);
        SetTagBit("M2_sag",0);
        SetTagBit("M2_sol",0);
        SetTagBit("M1_Y",0);
        SetTagBit("s2",0);
        SetTagBit("Y",0);
        SetTagBit("A",0);
    }

    if(GetTagDWord("gerilim")<=24)
    {
        SetTagBit("gerilim_tehlike",0);
    }
}

if((GetTagBit("UST_SINIR")==1)&(GetTagBit("s2")==1))
{
    SetTagBit("M1_Y",0);
    SetTagBit("s2",0);
}

return 1024;
}

```

EK D

Alarm Denetim Algoritması.

```
if(GetTagDWord("gerilim")>=25)
{
    SetTagBit("gerilim_tehlike",1);
    SetTagBit("sayfa4",0);
    SetTagBit("M1_A",0);
    SetTagBit("M1_Y",0);
    SetTagBit("M2_sag",0);
    SetTagBit("M2_sol",0);
    SetTagBit("Y",0);
    SetTagBit("A",0);
}

if(GetTagDWord("gerilim")<=24)
{
    SetTagBit("gerilim_tehlike",0);
}
```

EK E

SCADA yazılımında kullanılan diğer C-script kodları.

Akım Dönüşüm Rutini

```
if(GetTagBit("set")==1)
{
    akim=GetTagDWord("akim");
    akim_dönüsüm=akim/20;
    SetTagFloat("akim_dönüsüm",akim_dönüsüm);
}
//
```

Gerilim Dönüşüm Rutini

```
if(GetTagBit("set")==1)
{
    gerilim=GetTagDWord("gerilim");
    gerilim_dönüsüm=gerilim/10;
    SetTagFloat("gerilim_dönüsüm",gerilim_dönüsüm);
}
//
```

Motor Sol

```
#include "apdefap.h"
void OnLButtonDown(char* lpszPictureName, char* lpszObjectName, char*
lpszPropertyName, UINT nFlags, int x, int y)
{
    SetTagBit("M2_sol",1); //Return-Type :BOOL
}

```

```
#include "apdefap.h"
void OnLButtonUp(char* lpszPictureName, char* lpszObjectName, char*
lpszPropertyName, UINT nFlags, int x, int y)
{
    SetTagBit("M2_sol",0); //Return-Type :BOOL
}

```

Motor Sağ

```
#include "apdefap.h"
void OnLButtonDown(char* lpszPictureName, char* lpszObjectName, char*
lpszPropertyName, UINT nFlags, int x, int y)
{
    SetTagBit("M2_sag",1); //Return-Type :BOOL
}

```

```
}
```

```
#include "apdefap.h"
```

```
void OnLButtonUp(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName, UINT nFlags, int x, int y)
```

```
{  
    SetTagBit("M2_sag",0); //Return-Type :BOOL  
}
```

Motor Yukarı

```
#include "apdefap.h"
```

```
void OnLButtonDown(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName, UINT nFlags, int x, int y)
```

```
{  
    if(GetTagDWord("gerilim")>=25)  
    {  
        SetTagBit("Y",0);  
    }  
}
```

```
if(GetTagDWord("gerilim")<=24)
```

```
{  
    SetTagBit("Y",1);  
}
```

```
#include "apdefap.h"
```

```
void OnLButtonUp(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName, UINT nFlags, int x, int y)
```

```
{  
    SetTagBit("Y",0); //Return-Type :BOOL  
}
```

Motor Aşağı

```
#include "apdefap.h"
```

```
void OnLButtonDown(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName, UINT nFlags, int x, int y)
```

```
{  
    if(((GetTagDouble("kovaXY")<=350)&(GetTagDWord("gerilim")<=24))  
    {  
        SetTagBit("A",1);  
    }  
}
```

```
if(((GetTagDouble("kovaXY")>=350)|((GetTagDWord("gerilim")>=25))
```

```
{  
    SetTagBit("A",0);  
}
```

```

#include "apdefap.h"
void OnLButtonUp(char* lpszPictureName, char* lpszObjectName, char*
lpszPropertyName, UINT nFlags, int x, int y)
{
    SetTagBit("A",0);
    SetTagBit("M1_A",0);    //Return-Type :BOOL

    if(GetTagDouble("kovaXY")>=350)
    {
        SetTagBit("M1_A",0);
        SetTagBit("A",0);
    }
}

```

Başlat

```

#include "apdefap.h"
void OnLButtonDown(char* lpszPictureName, char* lpszObjectName, char*
lpszPropertyName, UINT nFlags, int x, int y)
{
    SetTagDWord("kovaXY",36);    //Return-Type :BOOL

    if(GetTagDWord("gerilim")<=24)
    {
        SetTagBit("s2",1);
        SetTagBit("set",1);
    }

    if(GetTagDWord("gerilim")>=25)
    {
        SetTagBit("s2",0);
        SetTagBit("set",0);
    }
}

```

Tahmin

```

#include "apdefap.h"
void OnLButtonDown(char* lpszPictureName, char* lpszObjectName, char*
lpszPropertyName, UINT nFlags, int x, int y)
{
    if(GetTagDWord("gerilim")<=24)
    {
        SetTagBit("sayfa3",1);
        SetTagDWord("kovaXY",37);
        SetTagBit("i",0);
    }

    if(GetTagDWord("gerilim")>=25)

```

```
{
    SetTagBit("sayfa3",0);
}
```

Ayarlar

```
#include "apdefap.h"
void OnLButtonDown(char* lpszPictureName, char* lpszObjectName, char*
lpszPropertyName, UINT nFlags, int x, int y)
{
    SetTagBit("sayfa2",1);
}
```

Ölçüm

```
#include "apdefap.h"
void OnLButtonDown(char* lpszPictureName, char* lpszObjectName, char*
lpszPropertyName, UINT nFlags, int x, int y)
{
    SetTagBit("sayfa4",1);    //Return-Type :BOOL
}
```

Ölçüm Sayfa 1

```
#include "apdefap.h"
BOOL _main(char* lpszPictureName, char* lpszObjectName, char*
lpszPropertyName)
{
    if(GetTagBit("sayfa4")==1)
    {
        return 1;
    }
    if(GetTagBit("sayfa4")==0)
    {
        return 0;
    }
}
```

Konum

```
#include "apdefap.h"
void OnLButtonDown(char* lpszPictureName, char* lpszObjectName, char*
lpszPropertyName, UINT nFlags, int x, int y)
{
    SetTagBit("konum_buton",1);
    SetTagBit("akim_buton",0);
    SetTagBit("gerilim_buton",0);
}
```

Akim

```
#include "apdefap.h"
void OnLButtonDown(char* lpszPictureName, char* lpszObjectName, char*
lpszPropertyName, UINT nFlags, int x, int y)
{
    SetTagBit("konum_buton",0);
    SetTagBit("akim_buton",1);
    SetTagBit("gerilim_buton",0);
}
```

Gerilim

```
#include "apdefap.h"
void OnLButtonDown(char* lpszPictureName, char* lpszObjectName, char*
lpszPropertyName, UINT nFlags, int x, int y)
{
    SetTagBit("konum_buton",0);
    SetTagBit("akim_buton",0);
    SetTagBit("gerilim_buton",1);
}
```

Alarm

```
#include "apdefap.h"
void OnLButtonDown(char* lpszPictureName, char* lpszObjectName, char*
lpszPropertyName, UINT nFlags, int x, int y)
{
    SetTagBit("sayfa5",1);    //Return-Type :BOOL
}
```

Alarm Sayfa 1

```
#include "apdefap.h"
void OnLButtonDown(char* lpszPictureName, char* lpszObjectName, char*
lpszPropertyName, UINT nFlags, int x, int y)
{
    SetTagBit("sayfa5",0);    //Return-Type :BOOL
}
```

EK F

Gerçekleştirilen J2ME uygulama programı.

```
/* Copyright © 2003 Sun Microsystems, Inc.
```

```
* All rights reserved. Use is subject to license terms.
```

```
*/
```

```
package paket;
```

```
import java.io.*;
```

```
import javax.microedition.lcdui.*;
```

```
import javax.microedition.midlet.*;
```

```
public class Test
```

```
    extends MIDlet
```

```
    implements CommandListener {
```

```
    public Display display;
```

```
    public Grafikler grafik;
```

```
    public List list;
```

```
    private Command ListCikis = new Command("CIK", Command.EXIT, 1);
```

```
    private Command ListSecim = new Command("SEC", Command.SCREEN, 2);
```

```
    private Setup setup;
```

```
    private MailReader mailreader;
```

```
    public Test() {
```

```
        display = Display.getDisplay(this);
```

```
        grafik = new Grafikler(this, mailreader, 1, 0);
```

```
        list = new List("Listem", List.IMPLICIT);
```

```
        setup = new Setup(this);
```

```
        Liste();
```

```
    }
```

```
    public void startApp() {
```

```
        display.setCurrent(grafik);
```

```
    }
```

```
    public void Liste(){
```

```
        ListSecim = new Command("SEC", Command.SCREEN, 1);
```

```
        ListCikis = new Command("CIK", Command.EXIT, 2);
```

```
        list.insert(0,"GENEL KONTROL",null);
```

```
        list.addCommand(ListCikis);
```

```
        list.addCommand(ListSecim);
```

```
        list.setCommandListener(this);
```

```
    }
```

```
    public void warn(String message) {
```

```
        Alert exception = new Alert("Warning", message, null, AlertType.WARNING);
```

```
        exception.setTimeout(Alert.FOREVER);
```



```

        display.setCurrent(exception,list);
    }

    public void pauseApp() {
    }
    public void destroyApp(boolean unconditional) {
    }
    public void FormuKapat(){
        destroyApp(false);
        notifyDestroyed();
    }
    public void commandAction(Command c, Displayable d) {
        if(c == ListSecim){

            if(list.getSelectedIndex()==0){
                setup.DegerAyarla("1");
                new MailReader(this, setup.getAccount()).start();

            }else if(list.getSelectedIndex()==1){
                setup.DegerAyarla("1");
                new MailReader(this, setup.getAccount()).start();
            }
        }else if(c == ListCikis){
            display.setCurrent(grafik);
        }
    }
}

```

```

/* Copyright © 2003 Sun Microsystems, Inc.
 * All rights reserved. Use is subject to license terms.
 */

```

```

package paket;

```

```

import javax.microedition.lcdui.*;
import javax.microedition.rms.*;

```

```

public class Setup
    extends Form
    implements CommandListener {
    private TextField user;
    private TextField password;
    private TextField serverUrl;
    private Account account;
    private Command okCommand = new Command("TAMAM",
        Command.SCREEN, 1);
}

```

```

private Command cancelCommand = new Command("iPTAL", Command.EXIT,
2);
private Test test;
protected volatile RecordStore accountRecord = null;
public Setup(Test test) {
    super("Ayarlar");
    this.test = test;
    user = new TextField("Kullanici ismi :", "", 100, TextField.ANY);
    password = new TextField("Kullanici sifresi :", "", 100, TextField.ANY |
TextField.PASSWORD);
    serverUrl = new TextField("Server adresi :", "http://", 100, TextField.ANY);
    append(user);
    append(password);
    append(serverUrl);
    addCommand(okCommand);
    addCommand(cancelCommand);
    setCommandListener(this);
    load();
}

public void load() {
    account = new Account();
    try {
        accountRecord = RecordStore.openRecordStore("mailaccount", true);

        if (accountRecord.getNumRecords() != 0) {
            byte[] userBytes = accountRecord.getRecord(1);
            byte[] passwordBytes = accountRecord.getRecord(2);
            byte[] serverUrlBytes = accountRecord.getRecord(3);

            if (userBytes != null) {
                account.user = new String(userBytes);
            }

            if (passwordBytes != null) {
                account.password = new String(passwordBytes);
            }

            if (serverUrlBytes != null) {
                account.serverURL = new String(serverUrlBytes);
            }
        }
    } catch (RecordStoreException rse) {
        test.warn(rse.toString());
    }
}

public void DegerAyarla(String istekdegeri){
    account.istek = istekdegeri;
}

```

```

    }

    public void showForm() {
        user.setString(account.user);
        password.setString(account.password);
        serverUrl.setString(account.serverURL);
        test.display.setCurrent(this);
    }

    public void close() {
        try {
            if (accountRecord.getNumRecords() == 0) {
                accountRecord.addRecord(account.user.getBytes(), 0,
account.user.length());
                accountRecord.addRecord(account.password.getBytes(), 0,
account.password.length());
                accountRecord.addRecord(account.serverURL.getBytes(), 0,
account.serverURL.length());
            } else {
                accountRecord.setRecord(1, account.user.getBytes(), 0,
account.user.length());
                accountRecord.setRecord(2, account.password.getBytes(), 0,
account.password.length());
                accountRecord.setRecord(3, account.serverURL.getBytes(), 0,
account.serverURL.length());
            }
            accountRecord.closeRecordStore();
        } catch (RecordStoreException rse) {
            test.warn(rse.getMessage());
        }
    }

    public Account getAccount() {
        return account;
    }

    public void commandAction(Command c, Displayable d) {
        if (c == okCommand) {
            account.user = user.getString();
            account.password = password.getString();
            account.serverURL = serverUrl.getString();
            close();
            test.display.setCurrent(test.grafik);
        } else { // cancelCommand
            test.display.setCurrent(test.grafik);
        }
    }
}

```

```

/* Copyright © 2003 Sun Microsystems, Inc.
 * All rights reserved. Use is subject to license terms.
 */
package paket;
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import javax.microedition.io.*;
import java.io.*;

public class MailReader
    extends Thread{

    private String str;
    private Test test;

    private Account account;
    public boolean TekrarOnayi=true;
    private boolean BaglantiDurumu = false;

    public MailReader(Test test, Account account) {

        this.test = test;
        this.account = account;
    }

    public void run() {
        Form progressForm = new Form("Veri transferi");
        progressForm.append("Sisteme giriliyor...\n");
        test.display.setCurrent(progressForm);
        progressForm.append("Veriler transfer ediliyor...");

        try {

            while(TekrarOnayi==true){
                postData(account.user, account.password, account.istek,
account.serverURL);
                if (BaglantiDurumu==false) {
                    test.warn("Server ile baglanti kurulamadi!");
                    return;
                }
                read(str);
                try{Thread.sleep(10000);}catch(InterruptedException e){}
            }
        } catch (Exception e) {
            test.warn("Server ile baglanti kurulamadi!");
        }
    }
}

```

```

}

private void read(String GelenVeri) throws IOException {

    int sayi;
    sayi=Integer.parseInt(GelenVeri);
    Grafikler grafik;
    grafik = new Grafikler(test, this, 2, sayi);
    test.display.setCurrent(grafik);
}

private void PostData(String kul, String sif, String ist, String serUrl)throws
IOException{
    HttpURLConnection http = null;
    InputStream IStr = null;
    String url = serUrl+"?kullanici="+kul+"&sifre="+sif+"&istek="+ist;
    try{
        http = (HttpURLConnection)Connector.open(url);
        http.setRequestMethod(HttpURLConnection.GET);
        IStr = http.openInputStream();
        ReadServerData(http, IStr);

    }finally{
        if (IStr != null)
            IStr.close();
        if (http != null)
            http.close();
    }
    if(BaglantiDurumu==false)
        test.warn("Server ile baglanti kurulamadi...");
}

private void ReadServerData(HttpURLConnection ht, InputStream Is) throws
IOException{
    if(ht.getResponseCode() == HttpURLConnection.HTTP_OK){
        int length = (int)ht.getLength();

        if(length != -1){

            byte ServerData[] = new byte[length];
            Is.read(ServerData);
            str = new String(ServerData);

        }else{

            ByteArrayOutputStream bStrm = new ByteArrayOutputStream();
            int ch;
            while ((ch = Is.read()) != -1)
                bStrm.write(ch);

```

```

        str = new String(bStrm.toByteArray());
        bStrm.close();
    }
    BaglantiDurumu = true;
    }
}
}
}

```

```

* Grafikler.java

```

```

*

```

```

* Created on 06 Eylül 2003 Cumartesi, 23:12

```

```

*/

```

```

package paket;

```

```

import javax.microedition.lcdui.*;

```

```

import javax.microedition.midlet.*;

```

```

public class Grafikler extends Canvas implements CommandListener{

```

```

    private Test test;

```

```

    private MailReader mailreader;

```

```

    private Setup setup;

```

```

    private int GrafParam;

```

```

    private String message = "M&A PRODUCT";

```

```

    private Command checkCommand = new Command("TAMAM",
Command.SCREEN, 1);

```

```

    private Command menuCommand = new Command("ANAMENU",
Command.SCREEN, 1);

```

```

    private Command cikisCommand = new Command("CIKIS", Command.EXIT, 3);

```

```

    private Command ayarCommand = new Command("AYARLAR",
Command.SCREEN, 2);

```

```

    private int ServerDegeri;

```

```

    public Grafikler(Test test, MailReader mailreader, int GrafParam, int
ServerDegeri)

```

```

    {

```

```

        this.test = test;

```

```

        this.mailreader = mailreader;

```

```

        this.GrafParam = GrafParam;

```

```

        this.ServerDegeri = ServerDegeri;

```

```

        setCommandListener(this);

```

```

        setup = new Setup(test);

```

```

    }

```

```

    public void paint(Graphics g) {

```

```

        switch(GrafParam){

```

```

            case 1:

```

```

                addCommand(menuCommand);

```

```

                addCommand(cikisCommand);

```

```

        addCommand(ayarCommand);
        try{
            Image im = Image.createImage("/logo.png");
            g.drawImage(im, 45, 23, Graphics.VCENTER | Graphics.HCENTER);
        }catch(Exception e){}
        g.drawString("Industrial Solutions",0,47,Graphics.TOP|Graphics.LEFT);
        break;
    case 2:
        addCommand(checkCommand);
        try{
            Image im =Image.createImage("/vinc.png");
            g.drawImage(im, 1, 1, Graphics.TOP | Graphics.LEFT);
        }catch(Exception e){}
        g.drawRect(63, (60-ServerDegeri), 10, 10);
        g.fillRect(63, (60-ServerDegeri), 10, 10);
        break;
    case 3:
        break;
    default:
        g.drawString("Hello Canvas",0,0,Graphics.TOP|Graphics.LEFT);
    }
}
}
public void commandAction(Command c, Displayable d) {
    if (c == cikisCommand) {
        test.FormuKapat();
    } else if (c == menuCommand) {
        test.display.setCurrent(test.list);
    }else if (c == checkCommand) {
        mailreader.TekrarOnayi=false;
        test.display.setCurrent(test.list);
    }else if(c == ayarCommand){
        setup.showForm();
    }
}
}
}

```

```

/* Copyright © 2003 Sun Microsystems, Inc.
 * All rights reserved. Use is subject to license terms.
 */

```

```

package paket;

```

```

public class Account {
    public String istek = "";
    public String user = "";
    public String password = "";
    public String serverURL = "";
}

```

ÖZGEÇMİŞ

1977'de Eskişehir'de doğdu. İlk, orta ve lise eğitimini Eskişehir'de tamamladı. 1995 yılında girdiği Kocaeli Üniversitesi Teknik Eğitim Fakültesi Elektrik Eğitimi Bölümü'nden 2000 yılında Elektrik Öğretmeni olarak mezun oldu.

2000 yılından beri Kocaeli Üniversitesi Teknik Eğitim Fakültesi Elektrik Eğitimi Bölümü'nde Araştırma Görevlisi olarak görev yapmaktadır.

