

KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

**DAĞINIK OPC SİSTEMLERİNİN İNTERNET
ÜZERİNDEN DENETLENMESİ**

YÜKSEK LİSANS TEZİ

Cihan ŞAHİN

Anabilim Dalı: Elektrik Eğitimi

Danışman: Yrd. Doç. Dr. Emine DOĞRU BOLAT

KOCAELİ, 2007

KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ




**DAĞINIK OPC SİSTEMLERİNİN İNTERNET
ÜZERİNDEN DENETLENMESİ**

YÜKSEK LİSANS

Cihan ŞAHİN

Tezin Enstitüye Verildiği Tarih: 1 Haziran 2007

Tezin Savunulduğu Tarih: 2 Temmuz 2007

Tez Danışması	Üye	Üye
Yrd. Doç. Dr. Emine DOĞRU BOLAT	Doç. Dr. Fevzi BABA	Yrd. Doç. Dr. Ayşe ERGÜN AMAÇ
()	()	()

KOCAELİ,2007

ÖNSÖZ VE TEŞEKKÜR

Gelişen ve gelişmekte olan firmalar veya kurumlar giderek büyüyen rekabet dünyasında ayakta kalmak ve daha ileriye gitmek için maksimum verim minimum maliyet ilkesine göre hareket etmektedirler.

Büyüyen bu rekabet dünyasında firmaların ayakta kalabilmeleri hiç kuşkusuzdur ki alt yapılarına ve dolayısıyla da süreç denetim yapılarına bağlıdır. Bu amaçla 1970'lerde ortaya atılan bir yöntem olan gözetleyici denetim ve veri toplama (SCADA) sistemleri firmalar ve kurumlar arasında en çok kullanılan süreç denetim yapılarıdır. Gelişen süreç denetim yapıları zamanla içinde bulunduğu şartlarla yeni oluşumlar içine girmişlerdir. 1990'lı yılların ikinci yarısında ortaya atılan OPC standardı süreç denetim yöntemlerinde farklı bir yenilik olmuştur.

1990'lı yıllarda kullanımı oldukça artan diğer bir yapı ise internettir. Dünya üzerindeki uzaklıkları bir klavyenin tuşu mesafesine yaklaştıran internet süreç denetim sistemlerinde de kullanılmaya başlanmıştır. Böylelikle, insanlar istedikleri her yerden sistemlere ulaşma imkanı yakalamışlardır. Yapılan bu çalışma ile farklı noktalarda bulunan OPC yapılarının internet üzerinden denetlenmesi ve görüntülenmesi sağlanmıştır.

Çalışmalarım süresince öneri ve desteğini esirgemeyen danışman hocam Yrd. Doç. Dr. Emine DOĞRU BOLAT'a, tez konusunun oluşmasından bitimine kadar en az benim kadar emek harcayan sevgili hocam ve dostum Mevlüt KARAÇOR'a ve sevgili aileme teşekkür ederim.

İÇİNDEKİLER

ÖNSÖZ	i
İÇİNDEKİLER	ii
ŞEKİLLER DİZİNİ	iv
TABLolar DİZİNİ	vi
SİMGELER DİZİNİ VE KISALTMALAR	vii
ÖZET	viii
İNGİLİZCE ÖZET	ix
BÖLÜM 1. GİRİŞ	1
1.1. Tezin Hedefi	2
1.2. Tez Konusu ile İlgili Yapılan Çalışmalar	2
1.3. Tezin Yapısı	8
BÖLÜM 2. ENDÜSTRİYEL OTOMASYON SİSTEMLERİ VE BİLEŞENLERİ ...	4
2.1. Süreç Yönetiminin Tarihsel Gelişimi	4
2.2. Endüstriyel Otomasyon Sistem Bileşenleri	7
2.2.1 Telemetry sistemleri	8
2.2.2. PLC sistemleri	8
2.2.3. DCS sistemleri	10
2.2.4. SCADA sistemleri	11
2.2.4.1. SCADA sistemlerinin temel yapısı	14
2.2.4.1.1. Saha ekipmanları	14
2.2.4.1.2. Veri ağları	14
2.2.4.1.3. Mantıksal denetim yazılımı	15
2.2.4.1.4. İnsan-makine arayüzü (MMI)	15
2.2.4.2. SCADA sistemlerinin uygulama alanları	16
2.2.4.3. SCADA sisteminin bileşenleri	16
2.2.4.3.1. Ana denetim merkezi	17
2.2.4.3.2. Uzak uç birimi	18
2.2.4.3.2.1. RTU'nun yapısı	18
2.2.4.3.2.2. RTU'nun iletişim ve ağ yapısı	22
BÖLÜM 3. OPC NEDİR?	24
3.1. OLE	26
3.2. COM	26
3.3. DCOM	27
3.4. Genel OPC Yapısı	27
3.4.1. OPC arayüzleri	28
3.5. OPC Özellikleri	30
3.5.1. OPC veri erişimi	30
3.5.2. OPC alarm ve olaylar	33
3.5.3. OPC tarihsel veri erişimi	35
3.5.4. OPC veri değişimi	35
3.5.5. OPC XML veri erişimi	36
3.5.6. OPC güvenliği	38

BÖLÜM 4. KLASİK OPC SİSTEMİNİN S7-200 PLC İLE GERÇEKLENMESİ...	39
4.1. S7200 PC Access OPC Server Yazılımı	39
4.2. Örnek Vinç ve Taşıyıcı Bant Uygulamasının OPC Tabanlı Denetimi	42
4.2.1. Örnek vinç uygulaması	43
4.2.2. Örnek taşıyıcı bant uygulaması	54
BÖLÜM 5. DAĞINIK OPC SİSTEMLERİNİN WEB TABANLI MERKEZDEN GÖRÜNTÜLENMESİ VE DENETLENMESİ	57
5.1. Vinç Sistemi Yerel Denetim Yapısı İnternet Arayüzü.....	58
5.2. Taşıyıcı Bant Sistemi Yerel Denetim Yapısı İnternet Arayüzü	59
5.3. Web Sunucu Programı	60
5.4. Uzak Denetim Merkezi	60
5.5. Dağınık OPC Yapılarının İnternet Üzerinden Denetlenmesi ve Görüntüleme Sisteminin Çalıştırılması	64
5.6. Dağınık OPC Yapılarının İnternet Üzerinden Denetlenmesinin Ekonomik Açıdan İncelenmesi	64
BÖLÜM 6. SONUÇLAR VE ÖNERİLER	66
KAYNAKLAR	67
EKLER	70
ÖZGEÇMİŞ	80

ŞEKİLLER DİZİNİ

Şekil 2.1. 1950’lerde süreç yönetiminin durumu	5
Şekil 2.2. 1960’lardaki süreç yönetiminin durumu.....	5
Şekil 2.3. 1970 ‘lerdeki süreç yönetiminin durumu.....	3
Şekil 2.4. 2000’li yıllarda süreç denetimi... ..	7
Şekil 2.5. Temel bir telemetri sistemi	5
Şekil 2.6. Bir PLC’nin iç yapısı	9
Şekil 2.7. Dağınık Kontrol Sistemi	10
Şekil 2.8. İlk SCADA sistemleri.....	12
Şekil 2.9. Günümüz SCADA sistemleri.....	13
Şekil 2.10. SCADA sisteminin temel yapısı.....	17
Şekil 2.11. RTU’nun temel yapısı.....	20
Şekil 2.12. Modüler olmayan RTU	21
Şekil 2.13. Modüler RTU.....	22
Şekil 2.14. RTU ağ yapısı	23
Şekil 3.1. Klasik Haberleşme yapısı	24
Şekil 3.2. OPC Standardı haberleşme yapısı.....	25
Şekil 3.3. Süreç denetim mimarisi	28
Şekil 3.4. OPC Sunucu-istemci yapısı	28
Şekil 3.5. OPC İstemci/Sunucu İlişkisi.....	29
Şekil 3.6. Genel OPC mimarisi.....	30
Şekil 3.7. İstemci/cihaz arasındaki mimari	31
Şekil 3.8. OPC sunucu veri diyagram hiyerarşisi	32
Şekil 3.9. OPC İstemci/Sunucu yapısı	33
Şekil 3.10. OPC sunucu ve OPC Alarm ve Olaylar sunucu arasındaki hiyerarşik yapı.....	34
Şekil 3.10. OPC DX Hiyerarşisi	36
Şekil 3.11. OPC XML ve İnternet.....	37
Şekil 4.1. OPC sunucu ana denetim ekranı.....	39
Şekil 4.2. OPC sunucuda oluşturulan gruplar	40
Şekil 4.3. Gruplar altında oluşturulan OPC nesnesi.....	40
Şekil 4.4. OPC nesnelerinin gerçek zamanlı incelenmesi.....	41
Şekil 4.5. Klasik OPC tabanlı denetim mimarisi	42
Şekil 4.6. Model vinç uygulaması.....	43
Şekil 4.7. Test düzeneği veri akış şeması.....	44
Şekil 4.8. Örnek vinç uygulamasına ait OPC sunucu ekranı	45
Şekil 4.9. Delphi paket programında OPC nesneleri	45
Şekil 4.10. OpcSimpleClient nesnesi ve nesne özellikleri.....	46
Şekil 4.11. Sistemde kullanılacak OPC sunucunun belirlenmesi	46
Şekil 4.12. OPC sunucu haberleşme görsel yapısı.....	47
Şekil 4.13. PLC çıkışları denetim ekranı	48
Şekil 4.14. PLC girişlerine ait denetim ekranı	48

Şekil 4.15. Girişlerin denetlenmesi	49
Şekil 4.16. Grafik canlandırma ekranı	50
Şekil 4.17. Sisteme ait arşivleme yapısı.....	50
Şekil 4.18. Sisteme ait verilerin web ortamına gönderilmesi.....	51
Şekil 4.19. Vinç prototipi yerel denetim ekranı	52
Şekil 4.20. Operatör tarafından OPC sunucuya bağlanma durumu	52
Şekil 4.21. OPC sunucu üzerinden sistemin çalışma anı	53
Şekil 4.22. Laboratuvar ortamında oluşturulan model vinç uygulama prototipi	53
Şekil 4.23. Model taşıyıcı bant uygulaması	54
Şekil 4.24. Taşıyıcı bant yerel denetim ekranı.....	55
Şekil 4.25. Laboratuvar ortamında oluşturulan model taşıyıcı uygulama prototipi.....	55
Şekil 5.1. Dağınık OPC sistemlerinin internet üzerinden denetimi	57
Şekil 5.2. DOPC Haberleşme Yapısı	58
Şekil 5.3. Timer ve IdHTTP nesnelere	59
Şekil 5.4. Uzak denetim merkezi ana ekranı.....	60
Şekil 5.5. Vinç uzak denetim merkezi	62
Şekil 5.6. Taşıyıcı bant uzak denetim merkezi	63

TABLolar DİZİNİ

Tablo 5.1. Uzak denetim merkezinin ASP sayfasından aldığı ve gönderdiği veriler 61

SİMGELER DİZİNİ VE KISALTMALAR

SCADA: Gözetleyici Denetim ve Veri Toplama (Supervisory Control And Data Acquisition)

OPC : Süreç denetimi için OLE (OLE for Process Control)

OLE : Nesne gömme ve bağlama (Object Linking and Embedding)

PLC : Programlanabilir Mantık Denetleyici (Programmable Logic Controller)

DOPC : Dağıntık OPC

COM : Bileşen nesne modeli (Component Object Model)

DDE : Dinamik veri deęiřimi(Dynamic Data Exchange)

DCOM: Dağıntık COM

HMI : İnsan makine arayüzü (Human Machine Interface)

RTU : Uzak uç birimi (Remote Terminal Unit)

DCS : Dağıntık kontrol sistemleri (Distributed Control Systems)

IED : Akıllı elektronik cihazlar (Intelligent Electronics Devices)

ADM : Ana denetim merkezi

UUB : Uzak uç birimi

ERP : İşletme kaynakları planlama (Enterprise Resource Planning)

OLE : Nesne gömme ve bağlama (Object Linking and Embedding)

PDA : Kişisel sayısal yardımcı (Personal Digital Assistant)

HDA : Tarihsel veri erişimi (Historical Data Access)

SOAP : Basit nesne erişim protokolü (Simple Object Access Protocol)

XML : Geniřletilebilir işaretleme dili (Extensible Markup Language)

ID : Kimlik numarası (IDENTITY)

DLL : Dinamik kütüphane bağlantısı (Dynamic Link Library)

PHP : Kişisel web sayfası (Personal Home Page)

ASP : Aktif sunucu sayfaları (Active Server Pages)

MMI : İnsan makine etkileşimi (Man Machine Interaction)

GPS : Global konum sistemi (Global Position System)

PC : Kişisel bilgisayar (Personal Computer)

DAĞINIK OPC SİSTEMLERİNİN İNTERNET ÜZERİNDEN DENETLENMESİ

Cihan ŞAHİN

Anahtar Kelimeler: OPC, SCADA, Web Teknolojileri

Özet: Süreç denetim sistemlerinde önemli bir rolü olan SCADA sistemleri, gelişen teknolojiyle birlikte kendini yenilemektedir. Gelişen yapılarıyla birlikte, kullanıcıya getirdiği büyük avantajların yanında bazı dezavantajları da bulunmaktadır. Denetim sistemiyle sahadaki cihaz arasında yaşanan haberleşme sorunları ve yüksek maliyetler yaşanan en büyük sorunlardandır. Bu ve bunun gibi sorunların giderilmesi amacıyla yapılan çalışmaların bir ürünü olan OPC oluşumu 1990'lı yılların sonunda kullanıcıların hizmetine sunulmuştur.

Bu çalışmada, SCADA sistemleri, sistem bileşenleri, OPC standardı ve özellikleri tanıtılmıştır. Klasik uygulaması oluşturulan OPC tabanlı yerel denetim yapılarına web teknolojileri entegre edilmiştir. Böylece, dağınık OPC sistemlerinin internet üzerinden denetlenmesi ve görüntülenmesi gerçekleştirilmiştir. Kullanıcının dünyanın neresinde olursa olsun yerel noktadaki sistemleri internet üzerinden denetlemesi ve görüntülenmesi sağlanmıştır.

CONTROLLING DISTRIBUTED OPC SYSTEMS ON THE INTERNET

Cihan ŞAHİN

Keywords: OPC, SCADA, Web Technologies,

Abstract: SCADA systems having an important role on process control systems are modernized with the developing technology. Together with the improved structures, SCADA has some disadvantages beside great advantages to the users. Communication problems between control system and device in the field and high costs are the biggest disadvantages. OPC standard which is the product of striving to remove these problems is presented to the users at the end of 1990's.

In this study, SCADA systems, system components, OPC standards and their specifications are introduced. Web technologies are integrated to the classical OPC based local control structured. Consequently, controlling and monitoring of DOPC (Distributed OPC) are implemented on the internet. It is provided that the operator can control and monitor the local systems. Wherever he is in the world.

BÖLÜM 1. GİRİŞ

Gelişmiş ülkelerde, “üretim” ile “kalite” kelimeleri beraber kullanılmaktadır. Bu nedenle, üretilen ürünün kalitesinin artırılması ve verimlilik ön plana çıkan en temel problemlerdir. Aynı ortamda, verimlilik ve kalite sadece üretim sektöründe değil diğer sektörlerde de (enerji, hizmet, v.b.) en önemli temel problemdir. Günümüz rekabet dünyasında bu iki problemi en iyi şekilde çözen firmalar veya kurumlar, rakiplerini geçerek lider pozisyonlara tırmanmaktadırlar. Günümüzde bu ve buna benzer problemlerin giderilmesi hiç kuşkusuz doğru bir alt yapı yatırımdan geçmektedir. Bu nedenlerle, firmalar alt yapılarının en önemli unsuru olan süreç denetim sistemlerine büyük önem vermişlerdir. Bu amaçla, firmalar öncelikle PLC (Programlanabilir mantık denetleyici) tabanlı otomasyon sistemleri kullanmaya başlamıştır. 1973’de, SCADA sisteminin ilk olarak literatüre girmesiyle, otomasyon endüstrisinde yeni bir dönem başlamıştır

1973’de ortaya atılan SCADA sistemleri büyük firmaların vazgeçilmezleri arasına girerek başarıyla kullanılmaktadır. Firmalar için büyük önem taşıyan SCADA sistemleri günümüzde milyonlarca dolara erişen ticari bir büyüklüğe sahiptir [1].

Firmalar için önemli bir ihtiyaç haline gelen SCADA yazılım yapıları büyük bir ticari hacme ve Siemens, Mitsubishi, Hitachi gibi firmaların tekeli haline girmiştir. SCADA paket yazılımlarının, firmalara yüksek maliyet getirmesi ve aynı zamanda kullanılan denetleyici ve gözetleyici ekipmanlarda marka ve model olarak kısıtlamalara gitmesi endüstriyel SCADA sistemlerinin en büyük dezavantajlarından biridir. Bu sebeplerden firmalar yeni arayışlar içine girmiş ve 90’lı yılların ikinci yarısında OPC standardını oluşturmuşlardır.

OPC standardı sayesinde kullanıcılar, kullandıkları OPC sunucular üzerinden sahada bulunan farklı marka ve modellerde, çeşitli cihazları görüntüleyebilmekte ve denetleyebilmektedir. Standart SCADA paket yazılımlarına nazaran çok daha uygun

fiyatlarda olması ve tekelleşmeyi ortadan kaldırması, görüntüleme ve denetleme sistemlerinde OPC'yi oldukça avantajlı bir noktaya taşımıştır.

1.1. Tezin Hedefi

Bu çalışmada; süreç denetim yapıları için yeni bir oluşum olan OPC sistemlerinin internet üzerinden denetlenmesi hedeflenmiştir. Bunun için öncelikle SCADA sistemleri ve yeni bir oluşum olan OPC sistemleri tanıtılmıştır. Tanıtılan OPC sistemlerinin yerel noktada klasik uygulaması gerçekleştirilmiştir. Yerel noktada oluşturulan OPC sistemleri için, görsel bir dil kullanılarak yerel denetim yapıları gerçekleştirilmiştir. Farklı noktalarda oluşturulan OPC tabanlı denetim yapılarına internet arayüzü eklenerek uzak denetim yapısı sağlanmıştır. Farklı noktalarda bulunan OPC tabanlı denetim yapılarının internet üzerinden denetlenmesiyle DOPC (Dağılık OPC) yapısı oluşturulmuştur.

1.2. Tez Konusu ile İlgili Yapılan Çalışmalar

Yapılan literatür çalışmasında, OPC sistemleri ile ilgili oluşturulan çalışmalar arasında pek fark yoktur. Genelde OPC sistemlerinin tanıtılması ile ilgili yapılan uluslar arası bilimsel çalışmalar aşağıda özetlenmiştir.

Shimanuki (1999), Endüstriyel otomasyon sistemleri için OPC yapısını tanıtmıştır. OPC bağımsız haberleşme yapısı ile tüm otomasyon bileşenlerinin birbiriyle haberleşmesi vurgulanmıştır.

Sitao (2000), SCADA yazılımlarında yaşanan sürücü sorunu ve bunun giderilmesi amacıyla oluşturulan OPC sistemlerinden bahsetmektedir. OPC yapısının bu sorunu gidermedeki mantığı açıklanmıştır.

Li Zheng (2002), OPC yapısına neden ihtiyaç duyulduğu, özellikleri ve kullanılabilirliği hakkında yapılan bir çalışmadır.

Ding (2003), Süreç denetim sistemlerinde kullanılan çeşitli haberleşme yapılarının ortak kullanıldığı bir çalışmadır. Farklı haberleşme yapılarının kullanıldığı uygulamaların OPC sunucu üzerinden denetlenme mantığı sunulmuştur.

Hong (2003), DCOM Network Protocol mantığı kullanılarak dağıtık yapıda bulunan verilerin toplanması gerçekleştirilmiştir.

Anwer (2004), HMI (Human Machine Interface: İnsan Makine Arayüzü) ve PLC arasında veri transferinin gerçekleştirilmesi için OPC sunucuyu kullanmıştır. PLC ve HMI arasında köprü görevi gören OPC sunucu yapısıyla sisteme ait verilerin veri tabanında kaydedilmesi gerçekleştirilmiştir.

Hao (2004), OPC DX özelliği ile OPC DA özelliğini karşılaştırılmıştır. OPC DX özelliği ile endüstriyel Ethernet yapısının kullanıldığı örnek bir uygulamadan bahsetmiştir.

Persin (2004), OPC, SCADA ve MATLAB yapılarının ortak kullanılması ile oluşturulmuş bir çalışma gerçekleştirilmiştir. OPC sunucu ile sistemden elde edilen veriler SCADA yapısına gönderilerek sistemin denetlenmesi ve MATLAB yapısına sunulan verilerle de sisteme ait gerekli analizlerin yapılması sağlanmıştır.

1.3. Tezin Yapısı

Tez altı bölümden oluşmaktadır. Birinci bölümde, OPC sistemleri ile ilgili literatür taraması sunulmakta ve tezin hedefi, yapısı açıklanmaktadır. İkinci bölümde, süreç denetimin tarihsel gelişimi, otomasyon sisteminin temel bileşenleri, SCADA sistemleri ve bileşenleri anlatılmaktadır. Üçüncü bölümde, OPC sistemlerinin tanımı, tarihsel süreci, özellikleri anlatılmaktadır. Dördüncü bölümde, kullanılan OPC sunucu yapısı, klasik OPC sistemlerinin uygulanması, görsel bir dil kullanılarak yerel denetim yapısının gerçekleştirilmesi anlatılmıştır. Beşinci bölümde, farklı noktalarda dağıtık durumda bulunan OPC yapılarının internet üzerinden görüntülenmesi ve denetlenmesi gerçekleştirilmiştir. Son bölümde, gerçekleştirilen çalışmaların özeti anlatılmış, sistemin avantajları ve güvenlik konularına değinilmiştir.

BÖLÜM 2. ENDÜSTRİYEL OTOMASYON SİSTEMLERİ VE BİLEŞENLERİ

İşletmelerin beyni konumunda bulunan süreç yönetimi geçmişten günümüze büyük değişimlere uğramaktadır. Kaliteli bir süreç yönetimine sahip olan işletmeler sürekli olarak büyümekte ve değişen teknolojiyle birlikte kendini yenilemektedir. Bu sebeple işletmeler alt yapılarını oluşturan endüstriyel otomasyona geçmişten günümüze büyük önem vermektedirler. İşletmeler için önemli bir ihtiyaç olan endüstriyel otomasyon sistemleri kullandıkları yere ve ihtiyaca göre değişik yapıları bünyesinde barındırmaktadır. Bu bölümde, süreç yönetiminin tarihsel gelişimi ve endüstriyel otomasyon sistemlerinde kullanılan yapılar hakkında bilgi verilmektedir.

2.1. Süreç Yönetiminin Tarihsel Gelişimi

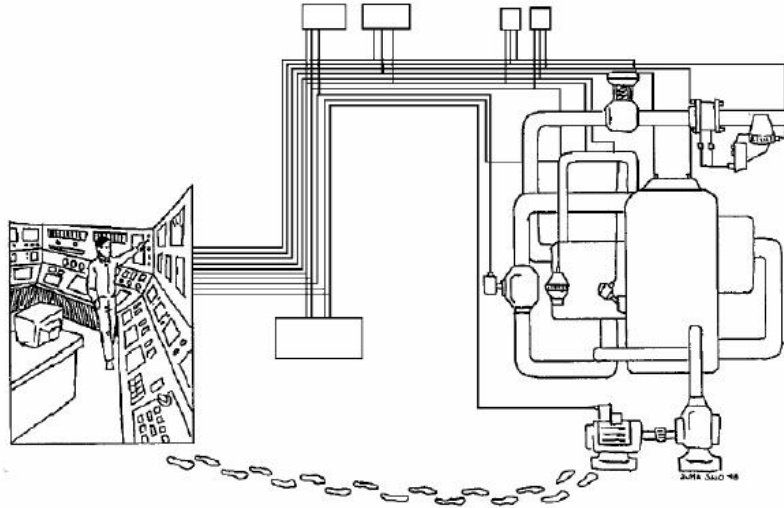
İşletmelerde, girdilerin çıktılara dönüşümündeki verimliliğin maksimum seviyelere ulaşmasında süreç yönetiminin rolü oldukça büyüktür. İşletmelerin büyüyen rekabet dünyasında ayakta kalabilmeleri için süreç yönetiminden beklentileri her geçen gün artmaktadır. Süreç yönetiminde altyapısını gelişen teknolojiyle birlikte yenileyen işletmeler günümüz rekabet dünyasında rakiplerini geçerek lider konuma gelmektedirler. Bu nedenle işletmeler, geçmişten günümüze alt yapılarını oluşturan endüstriyel otomasyon alanına oldukça yüksek harcamalar yapmaktadırlar.

Endüstriyel otomasyon sistemlerine tarihsel açıdan bakıldığında, 1950'li yıllarda, pnömatik sistemler ve analog cihazların kullanımı göze çarpmaktadır. Süreç yönetiminin en ilkel durumda olduğu bu yıllarda, süreçteki veriler yakın mesafeden görüntülenerek gerekli müdahaleler yapılmaktadır [10]. Şekil 2.1'de 1950'li yıllardaki süreç yönetiminin durumu görülmektedir [11].



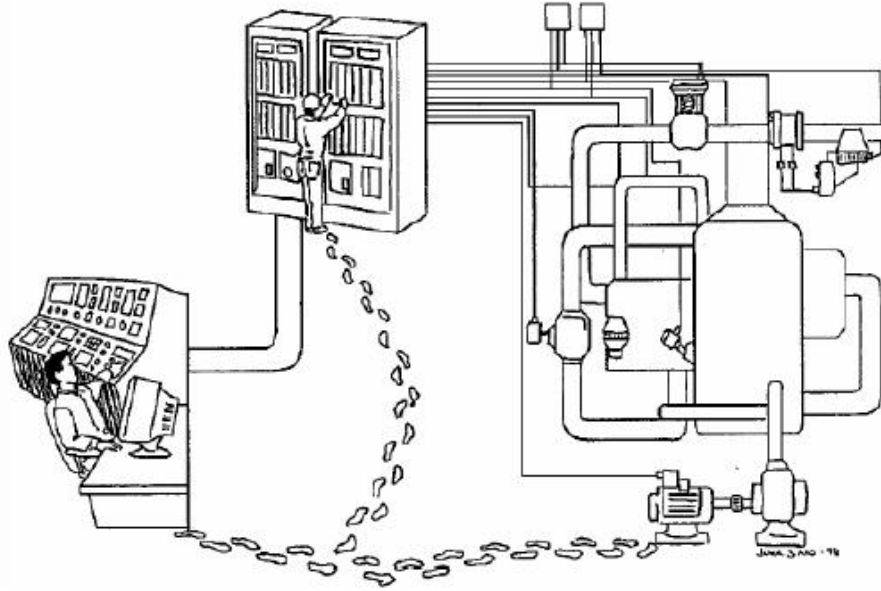
Şekil 2.1: 1950'lerde süreç yönetiminin durumu

1959-1964 yılları arasında transistörlü bilgisayarlar üretilmiş, COBOL, FORTRAN, ALGOL gibi yüksek düzeyli diller ve işletim sistemi geliştirilmiştir. Bilgisayar ve elektronik dünyasında yaşanan bu gelişmeler endüstriyel otomasyon dünyasını doğrudan etkilemiş, elektronik cihazlar ve bilgisayarlar endüstriyel otomasyon dünyasındaki yerini almışlardır. Bu gelişmelerle beraber endüstriyel otomasyon dünyasında merkezi denetim odaları oluşturulmaya başlanmıştır [10]. Şekil 2.2'de 1960'lardaki süreç denetiminin durumu görülmektedir [12].



Şekil 2.2: 1960'lardaki süreç yönetiminin durumu

1960'lı yılların sonlarında, mikroişlemci alanındaki gelişmeler süreç yönetimi için yeni bir devrin başlangıcı olmuştur. 1969 yılında Modicon firması tarafından röleli kumanda sistemlerinin yerine kullanılmak üzere ilk ticari PLC (Programmable Logic Controller: Programlanabilir Mantık Denetleyiciler) geliştirilmiştir [13]. Bu gelişmelerin yanında, analog sinyaller yerlerini yavaş yavaş dijital sinyallere bırakmaya başlamıştır. Endüstriyel Otomasyonda dijital sinyallerin elde edilmesiyle beraber Dağınkı Dijital Sistemler (DDS: Distributed Digital System) gelişmeye başlamış, süreç denetimi oldukça karmaşıklaşmaya başlamıştır. Şekil 2.3'de 1970'li yılların süreç yönetimi durumu görülmektedir [12].

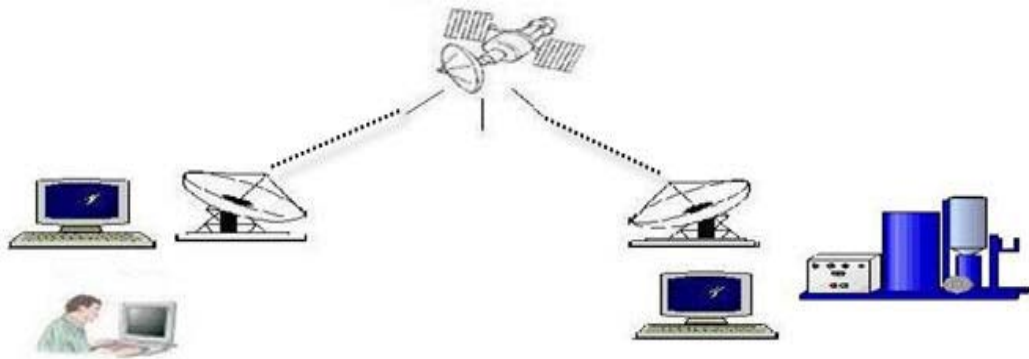


Şekil 2.3: 1970'lerdeki süreç yönetiminin durumu

1980'li yıllar süreç denetiminde video kameraların kullanılmaya başlandığı yıllardır. Denetlenen süreç ile denetim merkezi arasında iletişim koaksiyel kablolar kullanılarak sağlanmıştır. Bu yıllarda, sağlanan en büyük gelişme ise dağınkı dijital sistemlerin denetiminin tek merkezde toplanmasıdır. Bu sayede dijital sistemlerde merkezi bir denetimin yapısı gerçekleştirilmiştir.

1990'lı yıllara gelindiğinde, süreç denetiminde kullanılan mikroişlemci tabanlı cihazların birçok firma tarafından geliştirilmesi ve kullanıcıya çeşitlilik sağlamasıyla süreç denetiminde kullanımı yaygınlaşmıştır. Süreç denetiminde kullanılan bu cihazların güvenilirliğinde de büyük artışlar sağlanmıştır.

2000'li yıllara gelindiğinde ise süreç denetiminde, özellikle veri yollarında büyük devrimler yaşanmıştır. Denetlenecek sistem, yerinden denetiminin yanında uydu sistemlerinin kullanımı ile dünyanın herhangi bir noktasından denetime açık hale getirilmiştir. Bu yapı sayesinde sistemde meydana gelebilecek bir değişiklik kolayca görülebilecek ve en kısa sürede gerekli müdahale yapılabilecektir. Şekil 2.4'de 2000'li yılların süreç denetim durumu görülmektedir.



Şekil 2.4: 2000'li yıllarda süreç denetimi

2.2. Endüstriyel Otomasyon Sistem Bileşenleri

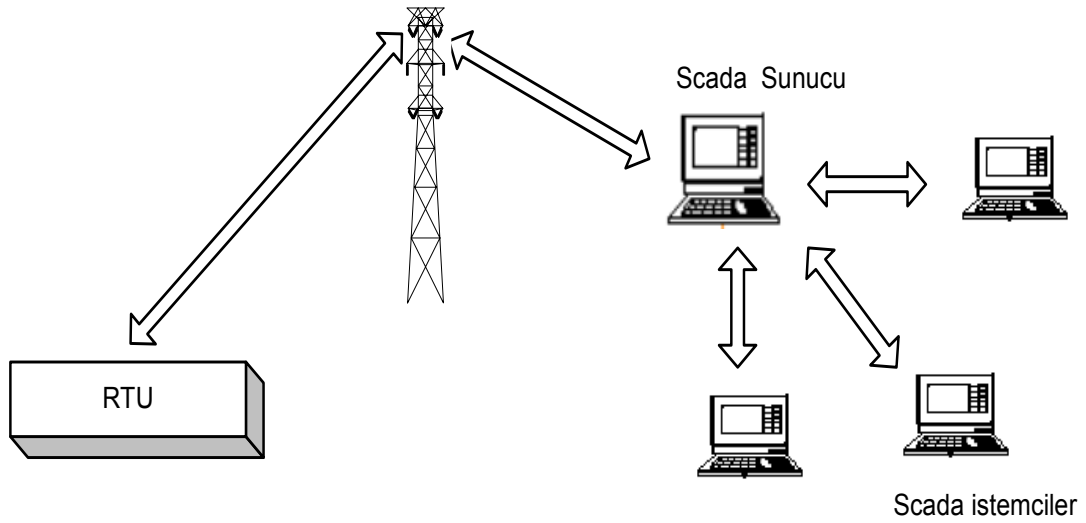
Endüstriyel otomasyon sistemleri kullanıldıkları yere ve yapıya göre çeşitli farklılıklar göstermektedir. İşletmelerin büyük maliyetler harcadığı endüstriyel otomasyon sistemlerini birbirinden ayıran en önemli fark hiç kuşkusuz ki kullandıkları haberleşme yapılarıdır. İşletmeler daha kaliteli bir süreç yönetimi yaratmak için sahadan en hızlı ve en ekonomik şekilde veri transferi elde etmek istemektedirler. Bu sebepler göz önüne alınarak gelişen endüstriyel otomasyon

sistemlerini dört başlık altında toplayabiliriz. Bunlar; Telemetri Sistemleri, PLC sistemleri, DCS sistemleri ve SCADA sistemleridir.

2.2.1 Telemetri sistemleri

Telemetri kelimesi, iki kelimenin birleştirilmesi ile meydana gelmiş bir terimdir. Telemetri sistemlerinde, “tele” kelimesi mesafe, “metri” de ölçüm manasına gelerek Telemetri sistemlerinin fonksiyonlarını ifade eder [12].

Telemetri yöntemiyle; sistem ya da tesis uzaktan kablosuz izlenme imkanına sahip olabilmektedir. Günümüzde bu yapı; büyük bir çoğunlukla RTU sistemlerinin kullanılması ile gerçekleştirilebildiği gibi, GSM vb. gibi sistemlerde kullanılarak gerçekleştirilebilmektedir. Şekil 2.5’de basitçe bir telemetri sistemi görülmektedir.



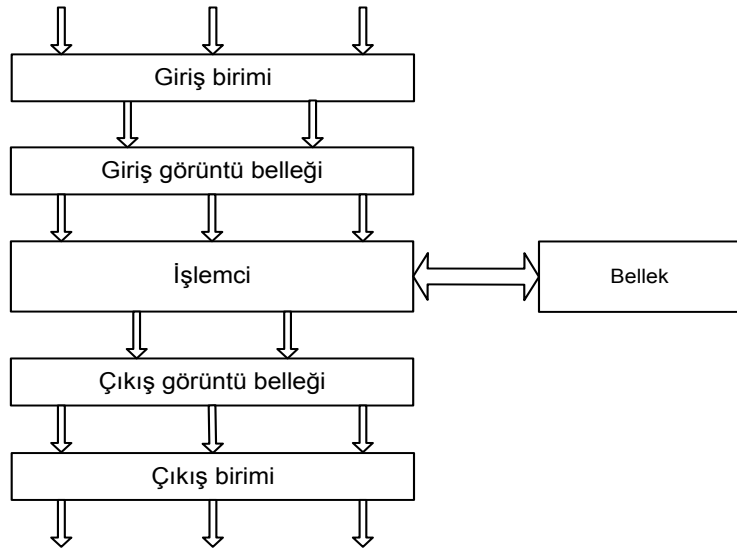
Şekil 2.5: Temel bir telemetri sistemi

2.2.2. PLC sistemleri

Programlanabilir lojik kontrolörler, endüstriyel otomasyon sistemlerinin kumanda ve denetim devreleri için uygun yapıda giriş-çıkış birimleri ve iletişim arabirimleri içeren mikroişlemci tabanlı bir denetleyicidir. İlk ticari PLC, 1969 yılında Modicon firması tarafından röleli kumanda devreleri yerine kullanılmak üzere üretilmiştir. PLC’lerin iç yapısı temel olarak Şekil 2.5’de gösterildiği gibi; bir işlemci, bellek ve

giriş-çıkış birimlerinden oluşur. Bellek, sistem programının bulunduğu sistem belleği, kullanıcı programının bulunduğu program belleği ve veri belleğinden oluşur. Sistem belleği, üretici firmanın geliştirdiği PLC işletim sisteminin yüklü olduğu bellek alanı, program belleği ise kullanıcı tarafından yazılan programın yüklendiği bellek alanıdır [13].

Veri belleği, giriş görüntü belleği ve çıkış görüntü bölümlerinden oluşmaktadır. Giriş görüntü belleği, PLC giriş noktalarındaki işaret durumlarının saklandığı bellek alanıdır. Çıkış görüntü belleği ise programın yürütülmesi sürecinde, çıkış noktalarına ait durumların saklandığı bellek alanıdır [13]. Şekil 2.6'da bir PLC'nin temel olarak iç yapısı görülmektedir [13].



Şekil 2.6: Bir PLC'nin iç yapısı

Günümüzde otomasyon dünyasının vazgeçilmezleri arasında buluna PLC'ler bir çok avantaj ve dezavantaja sahiptir [12]. Avantajları birkaç maddeyle sıralanırsa;

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.

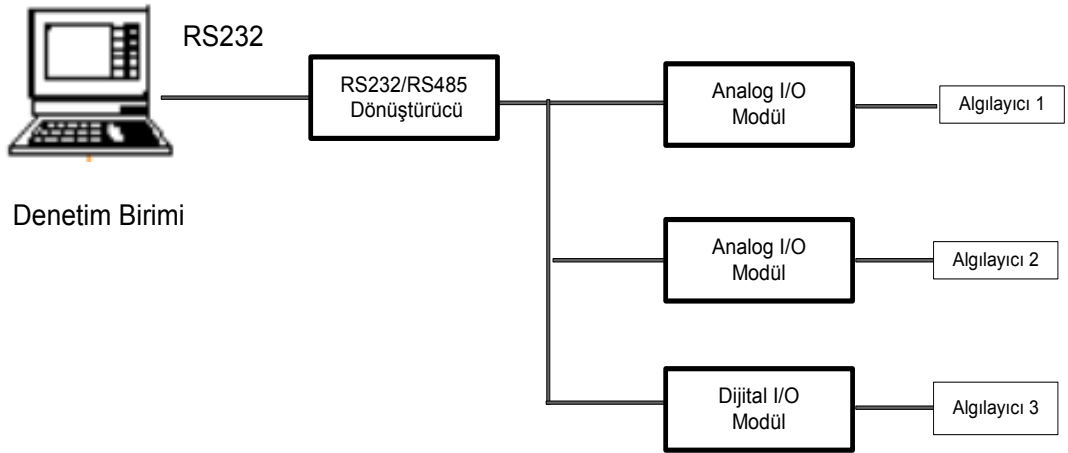
iii) PLC'ler endüstriyel ortamlarda bilgisayarlardan çok daha fazla dayanıklıdır.

Bu avantajlarının yanında birkaç maddeyle dezavantajlarını sıralarsak;

- 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.

2.2.3. DCS sistemleri

DCS (Distributed Control System) sistemleri; giriş/çıkış (I/O) birimleri, denetim modülleri ve operatör istasyonlarını yerel yapıda tümleşik olarak içeren sistemlerdir. Endüstriyel otomasyonda, sistem ile operatör istasyonu arasında haberleşme hızının milisaniyeler seviyesinde olan denetim birimleridir. Klasik SCADA sistemleri alarm tabanlı yapılar olduğundan, sahada meydana gelebilecek bir arıza denetim birimine, dakikalar hatta saatler sonra ulaşabilmektedir. Bu yapı sistemde meydana gelebilecek bir arızaya anında müdahale gerektirmeyen yerlerde güvenle kullanılmaktadır. Bazı özel sistemlerde ise, meydana gelebilecek arızanın denetim birimine milisaniyeler seviyesinde bildirilmesi ve acil müdahale edilmesi gerekebilir. Bu tip sistemlerin denetimi için, haberleşme alt yapısı oldukça hızlı olan DCS sistemleri kullanılmaktadır. DCS sistemlerinin haberleşme alt yapısında fiber optik kablolar kullanılmaktadır. Bu sebeple yüksek maliyetlere sahiptirler. DCS sistemleri bahsedilen bu özelliklerinden dolayı yerelde dağıtık olarak kullanılan denetim sistemleridir. Şekil 2.7’de örnek bir DCS sistemi görülmektedir [14].



Şekil 2.7: Dağıtık Kontrol Sistemi

DCS sistemlerinin diğer sistemlere göre bazı üstünlükleri bulunmaktadır [12]. Bunlar;

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

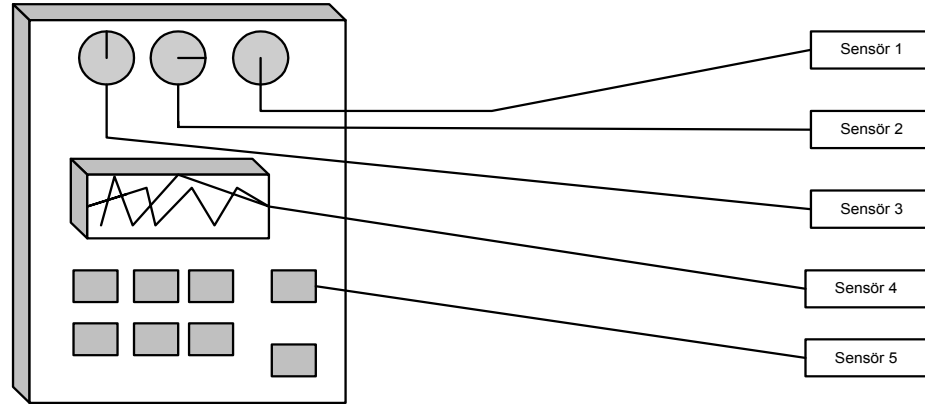
DCS sistemlerinin bu üstünlüklerinin yanında bazı dezavantajları da vardır [12]. Bunlar;

- i) DCS sistemlerine yeni birimlerin ilave edilmesi, diğer gerçek zamanlı sistemlere göre oldukça zordur. Bunun sebebi ise DCS sistemlerinin belirli bir süreci denetlemek amacıyla tasarlanmış olmasıdır. 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 yavaşlatmaktadır. Bu yavaşlama 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.

2.2.4. SCADA sistemleri

SCADA, Supervisory Control And Data Acquisition, kelimelerinin baş harflerinin yan yana gelmesi ile oluşturulmuş bir terimdir. Gözetleyici denetim ve veri toplama anlamına gelen SCADA terimi ilk olarak 1973 PICA (Power Industry Computer Applications) konferansında yayınlanmıştır [12]. Günümüzde endüstriyel otomasyon dünyasının vazgeçilmezleri arasında yerini alan SCADA sistemleri geçmişten günümüze teknolojiyle paralel olarak gelişimini sürdürmektedir. İlk SCADA

sistemleri, ölçüm cihazları, sinyal lambaları ve şeritli kart kaydedicilerinden oluşmaktadır [16]. Şekil 2.8 de ilk SCADA sistemi görülmektedir [16].



Şekil 2.8: İlk SCADA sistemleri

Şekil 2.8’de gösterilen ilk SCADA sistemlerinde, denetim panelinde bulunan yapılar (Sinyal lambası, ölçüm cihazları, anahtarlar vb.) sahada bulunan sensörlerle doğrudan iletişim halindedirler. Sisteme gerekli müdahaleler ise bir operatör tarafından panel üzerinde manuel olarak yapılmaktadır [16].

Şekil 2.8’de görülen SCADA sistemi ilkel yapıda olmasına rağmen bazı avantajları mevcuttur [16]. Bunlar;

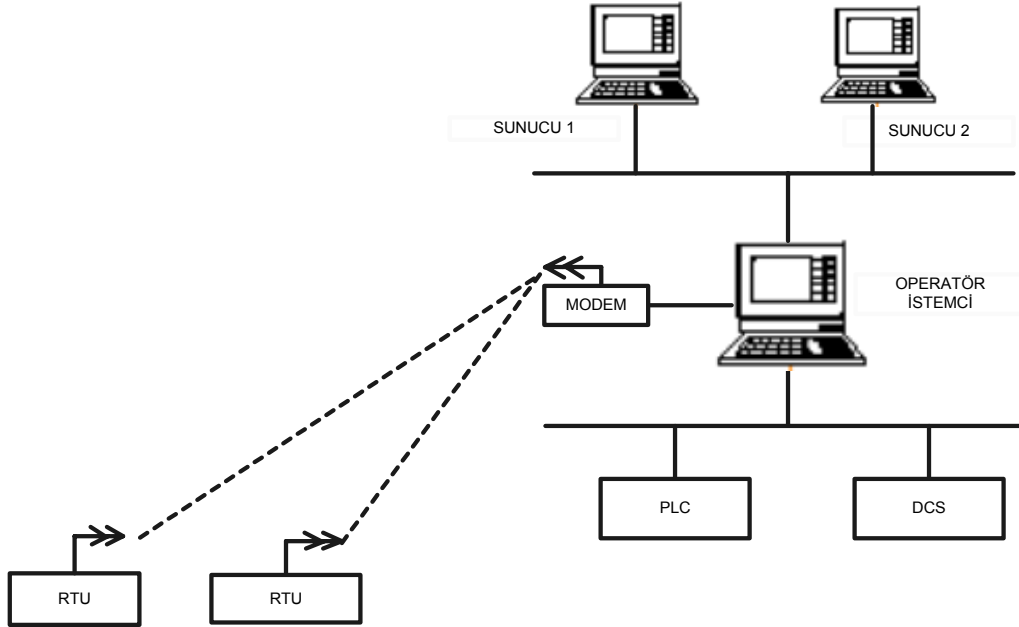
- i) Herhangi bir işlemci, RAM, ROM veya yazılım programına ihtiyacı bulunmamaktadır.
- ii) Sensörler, panel üzerinde buluna yapılar ile doğrudan bağlantı halindedirler.
- iii) Oluşturulan yapının maliyeti oldukça düşüktür.
- iv) Yapı üzerine ek cihazlar kolayca monte edilebilmektedir.

İlk SCADA sistemlerinin avantajlarının yanında bazı dezavantajları da bulunmaktadır [16]. Bunlar;

- i) Yüzlerce sensörden gelen haberleşme kabloları karışık bir yapıya sebebiyet vermektedir.
- ii) Veri tipi ve büyüklüğü oldukça küçük ve gelişmemiştir.
- iii) Gerçek zamanlı similasyon mümkün değildir.

- iv) Arşivleme yapısı oldukça yetersizdir.
- v) Operatör tarafından 24 saat boyunca gözetilmesi gerekmektedir.

Günümüz SCADA sistemleri ise veri toplama ve Telemetri sistemlerinin bir kombinasyonudur [12]. Oluşturulan bu yapı ile kilometrelerce uzaklıktaki yapılardan veriler tek merkeze toplanıp; merkezden görüntüleme, analiz yapma ve veri gönderme işlemleri gerçekleştirilebilmektedir. Kullanıcı yerel sistemlerde ağ alt yapısının kullanılmasıyla (profibus, fielbus, modbus vb.), uzak sistemlerde ise RTU alt yapısının kullanılmasıyla sistemdeki gerekli verilere ulaşmaktadır. Şekil 2.9'da günümüz SCADA sistemlerine bir örnek gösterilmektedir.



Şekil 2.9: Günümüz SCADA sistemleri

Şekil 2.9'da gösterilen günümüz klasik SCADA sisteminde; operatörün bilgilendirilmesi için belirli istasyonlara konulan istemci bilgisayar yerel ve uzakta bulunan sistemlerden gerekli verileri toplayıp, merkezde bulunan sunucu bilgisayarlara aktarmaktadır. Merkezi yapıda en az iki adet sunucu bilgisayar bulunması gerekmektedir [12]. Bunlardan biri ana sunucu olarak görev yaparken diğeri yedek sunucu olarak görev yapmaktadır.

2.2.4.1. SCADA sistemlerinin temel yapısı

Günümüz SCADA sistemleri temel olarak, saha ekipmanları, veri ağları, mantık denetim yazılımı ve insan makine arayüzü (MMI) olmak üzere dört temel kısımdan oluşmaktadır [12].

2.2.4.1.1. Saha ekipmanları

SCADA sistemlerinde saha ekipmanları, PLC, DCS ve IED (Intelligent Electronics Devices: Akıllı Elektronik Cihazlar) yapılarından oluşmaktadır. Kullanılan bu yapılar sisteme bazı avantajları kazandırmaktadır [16]. Bunlar;

- i) Çok geniş bir yapıdaki verilere kolaylıkla ulaşılmasını ve kaydedilmesini kolaylaştırmaktadırlar.
- ii) Geniş bir alan üzerinde bulunan binlerce sensör bu şekilde sistemle kolayca haberleşebilmektedir.
- iii) Operatör sistemi simülasyonla takip edebilmektedir.
- iv) Veriler herhangi bir yerden görüntülenebilmektedir.

Bu avantajlarının yanında bazı dezavantajları da bulunmaktadır [16]. Bunlar;

- i) Sistem ilk SCADA sistemlerine göre oldukça karışık bir hal almaktadır.
- ii) Farklı yeteneklerdeki operatörlere, örneğin sistemi analiz eden ve programlayan, ihtiyaç duyulmaktadır.

2.2.4.1.2. Veri ağları

Veri ağları, denetlenen sistem ile insan-makine arayüzü arasında haberleşmeyi sağlayan yapılardır. SCADA sistemlerinde veri transferi için kablolu ve kablosuz olmak üzere iki tür haberleşme yapısı kullanılmaktadır. Kablolu iletişimde koaksiyel ve fiber optik kablolar kullanılan yere göre tercih edilmektedirler. Merkezi denetleme istasyonuna uzakta bulunan yapılarla haberleşme ise, telefon hatları veya özel olarak döşenmiş haberleşme yapıları üzerinden sağlanmaktadır. Telefon hatları veya özel olarak kablo yapılarının oluşmasına izin vermeyen coğrafi bölgelerde ise iletişim radyo frekanslarıyla sağlanmaktadır.

2.2.4.1.3. Mantıksal denetim yazılımı

Mantıksal denetim yazılımları; PLC, DCS ve RTU gibi yapıların programlandığı mantıksal denetim yazılımları ve SCADA sistemini oluşturan sunucularda bulunan mantıksal denetim yazılımları diye iki kategoriye ayrılabilir. SCADA sistemini oluşturan mantıksal denetim yazılımlarında kullanıcı firmalar tarafından olması istenen temel özellikler şunlardır [16].

- i) Kullanıcı arayüzü,
- ii) Grafik ekranı,
- iii) Alarm yapıları,
- iv) PLC ve RTU arayüzleri,
- v) Sistemin ölçeklendirilmesi,
- vi) Veri erişim,
- vii) Veri tabanı,
- viii) Ağ yapısı,
- ix) Dağıtık sunucu-istemci yapısıdır.

Yukarıda verilen özelliklerin yanında günümüz SCADA yazılım üreticileri, kullanıcının dünyanın herhangi bir yerinden sistem hakkında bir bilgiye ulaşabilmesi için SCADA yazılımlarının internetle entegre çalışabilir hale getirmişleridir. İnternet tabanlı SCADA yazılımları ile diğer yazılımlar arasında bazı farklar vardır [12]. Bunlar;

- i) İnternet entegrasyonu.
- ii) İnternet tabanlı cihazların sistemde çalıştırılması.
- iii) Uzak operatör sayısının isteğe bağlı olarak kolayca artırılabilmesi.

2.2.4.1.4. İnsan-makine arayüzü (MMI)

SCADA sistemi ile operatör arasındaki ilişkiyi kuran birimdir. Operatörler, üzerlerine düşen görevleri kendilerine ait konsolları kullanarak yerine getirmektedirler. Bir operatör konsolu temel olarak, monitör, klavye, fare, yazıcı ve çiziciden oluşmaktadır. İnsan-makine arayüzünün gerçekleştirebileceği işlevler birkaç maddeyle sıralanabilir. Bunlar;

- i) SCADA yazılım programının kullanılması.
- ii) Denetlenen ve bilgi toplanan cihazların ekrandan izlenmesi, bu cihazlara ekrandan komut göndererek durumlarının takibi.
- iii) Sistemde kullanılan cihazlar hakkında detaylı bilgi (bakım tarihi, markası, üzerindeki arızalar).
- iv) Alarmları ve olayları kaydetme ve listeleme.
- v) Kullanıcının gerçekleştirdiği olayları kayıt etme ve listeleme.
- vi) Yetkili kullanıcıların şifre ve kullanıcı adıyla sisteme girmesine izin verme.

2.2.4.2. SCADA sistemlerinin uygulama alanları

SCADA sistemleri, geniş bir alana kurulmuş tesislerin birçoğunda kullanılmaktadır. Coğrafi yapısından dolayı insan gücünün istihdam edilemeyeceği işletme noktaları ile denetleme merkezi arasındaki mesafelerin kilometrelerin bulunduğu yerlerde tercih edilen sistemlerdir. SCADA sistemlerinin kullanım alanlarına genel olarak şu örnekler verilebilir [18].

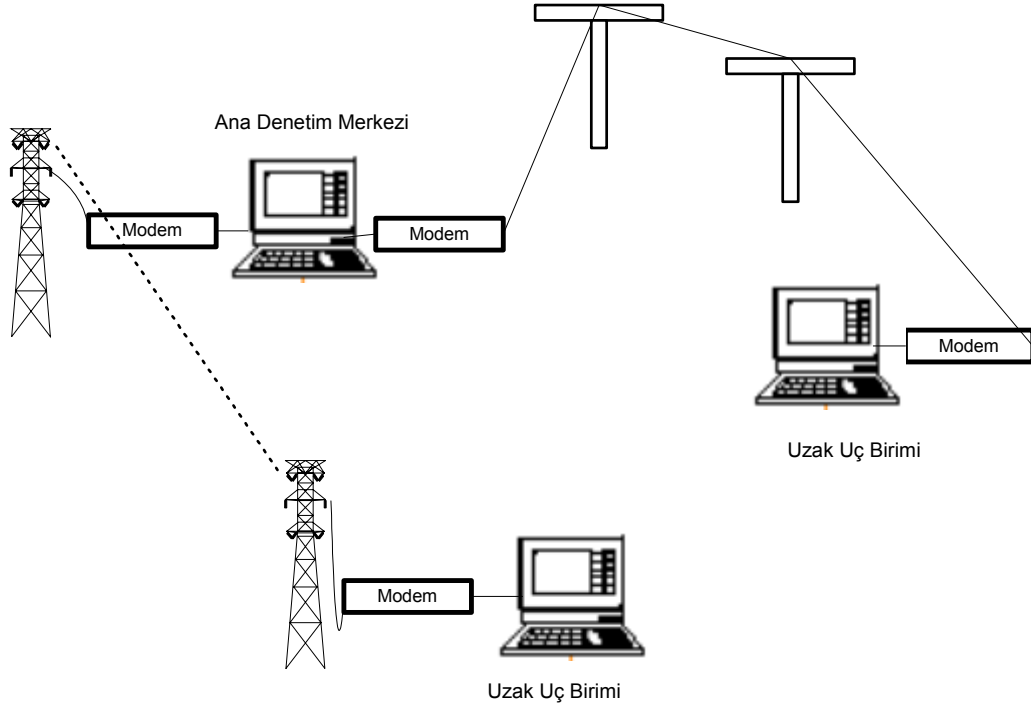
- i) Su toplama, dağıtma ve arıtma tesislerinde,
- ii) Petrokimya endüstrisinde,
- iii) Enerji üretim ve dağıtım sistemlerinde,
- iv) Güvenlik sistemlerinde,
- v) Demir-çelik endüstrisinde,
- vi) Otomotiv endüstrisinde,
- vii) Raylı taşıma sistemlerinde,
- viii) Çimento endüstrisinde,
- ix) Hava kirliliği ölçümlerinde,
- x) Bina otomasyon sistemlerinde.

2.2.4.3. SCADA sisteminin bileşenleri

SCADA sisteminin temel yapısı iki bileşenden oluşmaktadır [12]. Bunlar;

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

Şekil 2.10’da, ADM (Ana Denetim Merkezi) ve iki UUB (Uzak Uç Birimi) olan bir SCADA sistemi görülmektedir. ADM merkezi, UUB birimlerinden biriyle kablolu iletişim yöntemi kurarken kablolu iletişim imkanı olmayan diğer UUB birimiyle radyo frekansı kullanarak haberleşme sağlamaktadır. Şekil 2.10’da temel SCADA sistemine verilen örnek görülmektedir.



Şekil 2.10: SCADA sisteminin temel yapısı

2.2.4.3.1. Ana denetim merkezi

Ana Denetim Merkezi; bir ana sunucu ve ana sunucuya paralel olarak çalışan bir yedek sunucudan oluşan, oluşturulan sistemi denetleyen birimdir. Genel olarak ADM’ler geniş bir yapıda oluşturulmuş sistemlerden aldığı verileri operatörün anlayabileceği çeşitli yapılara (görsel arayüz, yazıcı çıktısı, vb.) çeviren birimdir.

ADM’nin görevleri birkaç maddeyle açıklanabilir [12] . Bunlar;

- i) Uzak uç birimlerinden bilgilerin toplanması,
- ii) Toplanmış bilgilerin yazılım programında işlenmesi,
- iii) İşlenmiş bilgi sonucuna göre denetim komutunun gönderilmesi,
- iv) Belli olaylar sonucunda alarm oluşturma,

- v) Oluşan olayları ve verileri zaman sırasına göre, alarm ve eğrisel olarak ekranlarından operatörün kullanımına sunmak,
- vi) Toplanmış ve işlenmiş bilgiyi sonuçlarında yazıcı ve çizici ile rapor çıkarma,
- vii) Yazıcı, çizici, haberleşme birimleri gibi ek donanımların denetimi gerçekleştirmek olarak sıralanabilir.

ADM'de operatör, operatör arayüzünü kullanarak yerel ve uzak birimlerde bulunan cihazlara erişebilme hakkına sahiptir. Böylece ADM'de bulunan operatör uzak yapıda oluşabilecek bir arıza durumuna çok kısa sürede müdahale ederek arızanın büyümesini engelleyebilir.

ADM'de operatöre sistem hakkında bilgi verilen cihazların tümü operatör çıkışı olarak adlandırılır [12]. Bilgisayarlar ve yazıcılar en genel operatör çıkışlarıdır. Bu yapıların dışında sesli veya ışıklı uyarı sistemleride operatör çıkışı olarak adlandırılabilir.

ADM'de operatör tarafından veri girilen yapılar ise operatör girişi olarak adlandırılır. Bunlara örnek olarak; klavye, fare ve dokunmatik paneller verilebilir.

2.2.4.3.2. Uzak uç birimi

SCADA sistemlerinde, UUB olarak PLC ve RTU (Remote Terminal Unit: Uzak Uç Birim) yapıları kullanılmaktadır.

2.2.4.3.2.1. RTU'nun yapısı

Bir SCADA sisteminde RTU; bulunduğu merkezdeki değişkenlerden elde ettiği verileri toplayan, depolayan ve bu verileri çeşitli iletişim yolları ile ADM'ye gönderen, ADM'den gelen komutları sisteme uygulayan SCADA birimidir.

SCADA sistemlerinde ilk kullanıldığında mikroişlemcisiz yapıya sahip olan RTU'lar, sadece ölçüm yaparak elde ettiği verileri merkeze gönderen ve merkezden aldığı komutları uygulayan bir yapıya sahiptiler [16]. Günümüzde mikroişlemci

tabanlı olarak üretilen RTU'lar, ADM'den aldığı veriyi sahadaki cihazların anlayabileceği analog ve dijital sinyallere dönüştürerek, sistemdeki cihaza gerekli emirleri iletebilecek bir yapıya sahiptirler. Bu yapıları sayesinde endüstriyel uygulamalarda geniş bir alanda kullanılmaktadır. Şehir şebekesi su dağıtım sistemleri, atık su denetim sistemleri ve raylı ulaşım sistemleri en temel örnek olarak verilebilir.

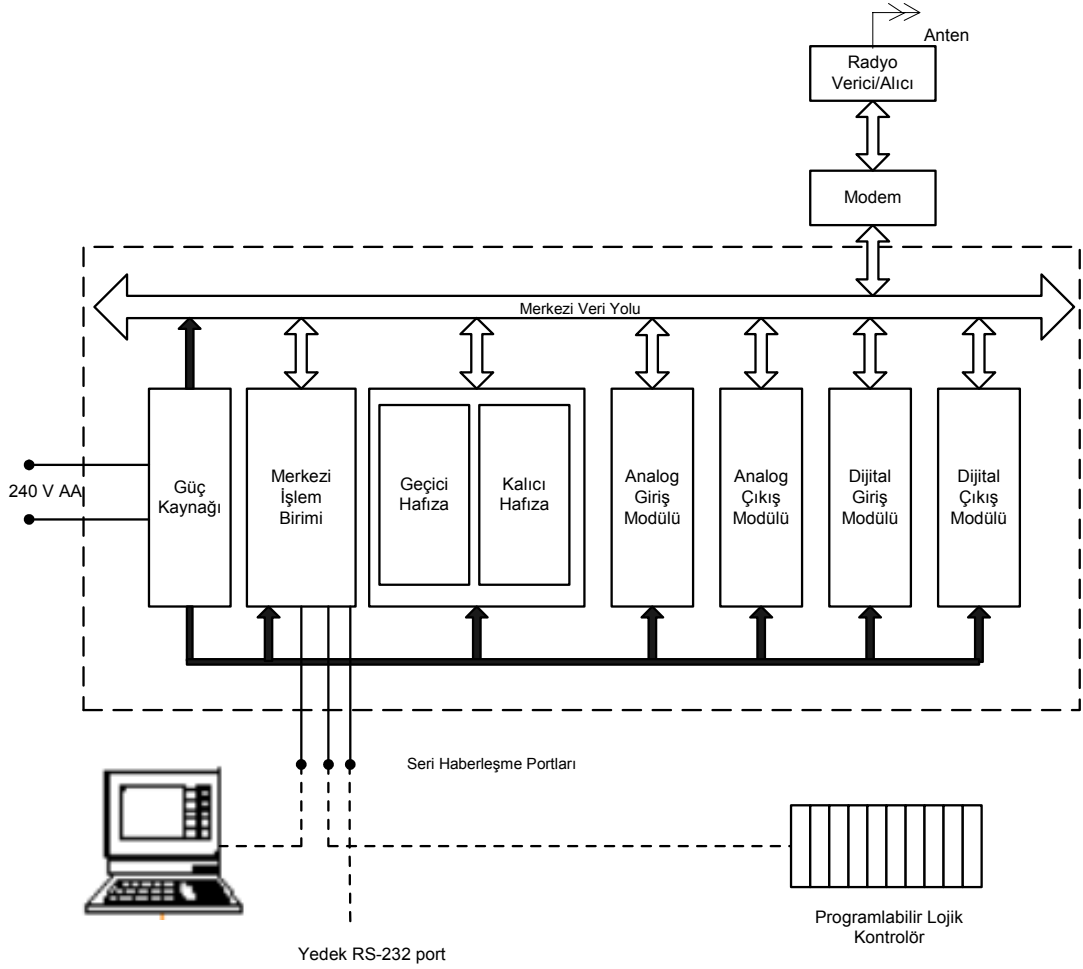
Mikroişlemcisiz RTU'ların kullanıldığı SCADA sistemlerinde, alarm durumları ve diğer bütün işlemler ana denetim merkezi üzerinden yapılmaktadır. Bu durumda, merkezin alarm kararı verip RTU'ya komut göndermesi belli bir süreyi almakta ve anında müdahale gerektiren yapılarda gecikmelere yol açmaktadır [14]. Bu ve buna benzer durumların oluşmasından dolayı mikroişlemcisiz RTU'lar zamanla yerini mikroişlemci tabanlı yapılara bırakmışlardır. Mikroişlemci RTU yapılarının avantajları birkaç maddeyle açıklanabilir . Bunlar;

- i) Merkezi birime gerek duymadan kendi başlarına karar verebilirler. Bulunduğu yerdeki durumuna göre işlenmiş programın çalışmasına olanak sağlarlar. Böylece bulunduğu yerde oluşabilecek acil bir durumda, merkezi birime gerek kalmadan sistemin çalışmasını engellerler.
- ii) RTU'lar kendi başlarına karar vermelerinden dolayı, merkezi yapıda meydana gelebilecek arıza durumundan etkilenmez, süreç denetimine devam ederler.
- iii) Mikroişlemcili RTU'lar, mekanik veya elektromekanik cihazların işlevlerini üstlenmektedir. Mekanik ve elektromekanik cihazlarda zamanla meydana gelebilecek aşınmalardan dolayı hassasiyetlerde bozulmalar olurken, RTU'larda böyle bir sorun olmamaktadır. Şekil 2.11'de mikroişlemci tabanlı bir RTU'nun temel yapısı görülmektedir [16].

Şekil 2.11'de görülen temel RTU yapısı çeşitli birimlerden oluşmaktadır. Bunlar;

- i) Güç kaynağı, RTU'ya şebekeden enerji sağlayan birimdir. Şebeke gerilimini kullanılan RTU çeşidine göre 12/24/48 V DC gerilim seviyelerine düşürmektedir[16].
- ii) Merkezi işlem birimi, RTU'nun beyni durumundadır. RTU içinde gerçekleşen tüm işlemler yöneten birimdir.

iii) Kalıcı hafıza, RTU'ya yüklenen programa göre gerekli denetim işlemlerini yapan birimdir.



Şekil 2.11: RTU'nun temel yapısı

iv) Geçici hafıza, dış dünyadan RTU'ya gelen verileri gerekli denetim işlemlerinin gerçekleşmesi için kalıcı hafızaya yönlendiren birimdir.

v) Analog giriş modülü, RTU'nun dış dünyadan analog sinyalleri aldığı modüldür. İç yapısında A/D dönüştürücü bulunmaktadır. Temel bir analog giriş modülü; 8 veya 16 girişe, 8 veya 12 bit çözünürlüğe, 4-20 mA hassasiyete, 240 Kohm- 1 Mohm giriş direncine sahiptir [16].

vi) Analog çıkış modülü, analog cihazları denetlemek için kullanılan modüldür. İç yapısında işlemciden alınan dijital sinyalleri analog sinyallere çeviren D/A dönüştürücü bulunmaktadır. Temel bir analog çıkış modülü, 8 analog çıkışa, 8 veya 12 bit çözünürlüğe, 0-20 mA, 0-20 mV çıkış değerlerine sahiptir [12-16].

vii) Dijital giriş modülü, dış dünyadan buton, anahtar ve röle gibi yapılardan AÇIK/KAPALI sinyallerinin alındığı birimdir.

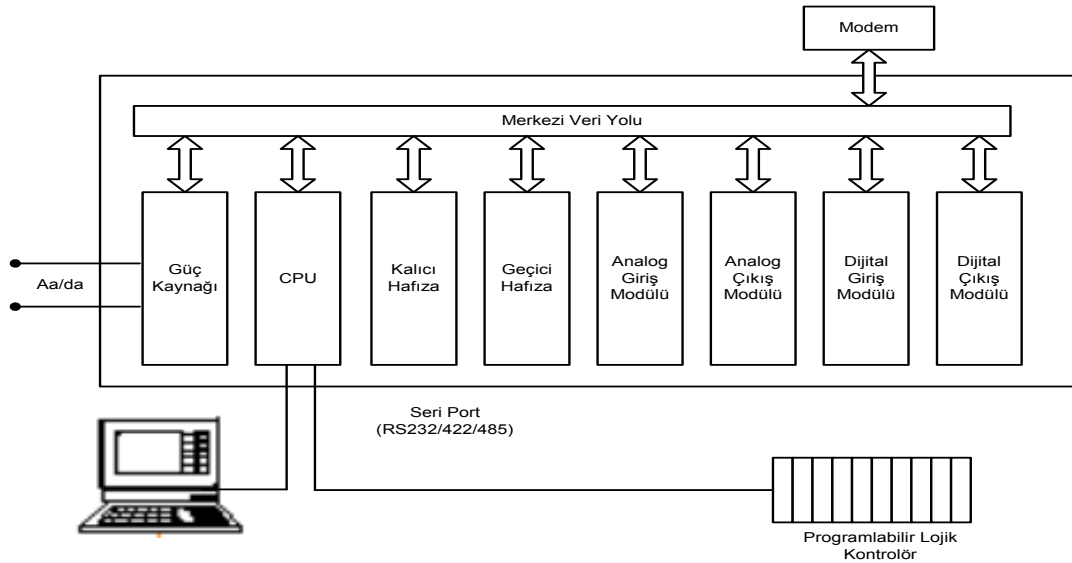
viii) Dijital çıkış modülü, RTU'nun üretmiş olduğu AÇIK/KAPALI sinyallerini kullanarak dış dünyada bulunan röle ve buna benzer yapıların denetlenmesini için kullanılan modüldür.

ix) Haberleşme arayüzü, RTU'nun ilgili cihazlarla haberleşme sağladığı birimdir.

RTU'lara yukarıdaki modüllerin yanında; GPS (Global Position System) modülü eklenebilir [12]. GPS, RTU saatlerinin uydu üzerinden ayarlanarak sistem bileşenlerinin eş zamanlı çalışması sağlar.

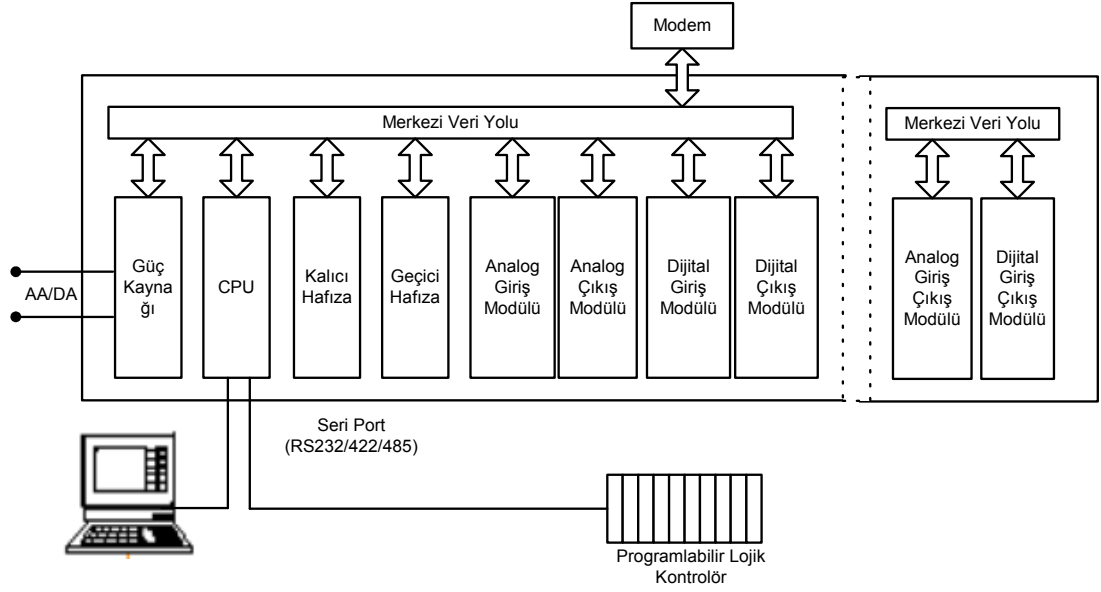
Yapısal olarak RTU'lar iki tipe ayrılırlar. Bunlar;

i) Modüler olmayan RTU (single board RTU): Bu yapıdaki RTU'lar üzerinde sabit olarak bulunan modüllerin haricinde genişletme amacıyla ilave modül takılmasına izin vermezler . Şekil 2.12'de Modüler olmayan RTU görülmektedir.



Şekil 2.12: Modüler olmayan RTU

ii) Modüler RTU: Bu yapıdaki RTU'lar üzerindeki modüllere ilave olarak, analog veya dijital modüller ilave edilmesine izin veren modüllerdir.



Şekil 2.13: Modüler RTU

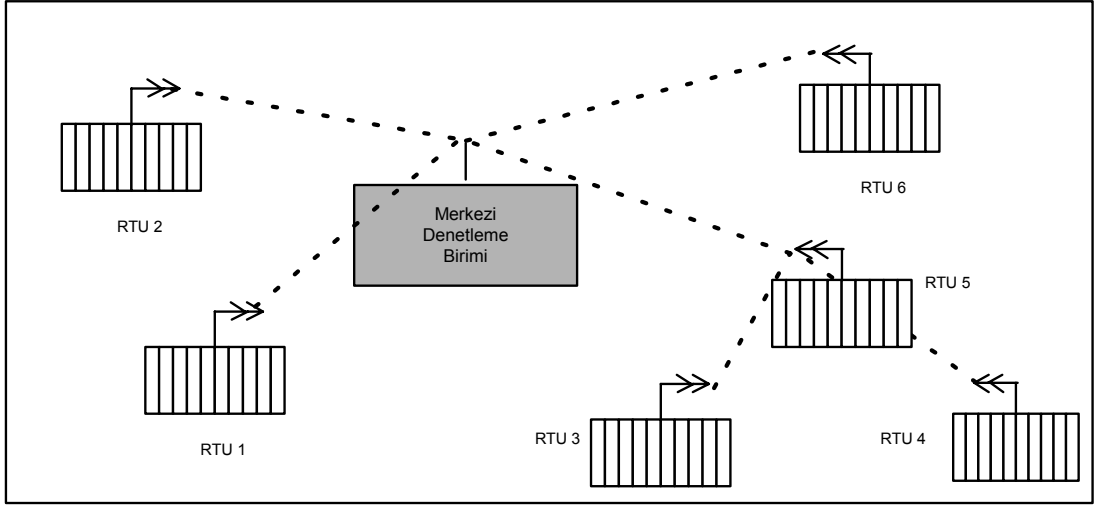
2.2.4.3.2.2. RTU'nun iletişim ve ağ yapısı

RTU'lar, ADM ve diğer RTU yapılarıyla haberleşme amacıyla entegre iletişim arayüzlere (port) sahiptir. Bu portlar, dial-up modem, radyo verici sistemleri, fiber optik hatlar ve Profibus, Fieldbus ve Modbus gibi protokolleri haberleşme amaçlı olarak kullanır [12].

RTU'ların oluşturduğu ağ yapılarında, bir RTU üzerinden diğer bir RTU'ya erişime izin vererek, RTU üzerindeki programda değişiklik veya yeniden yükleme durumlarının gerçekleşmesine olanak sağlamaktadır. Şekil 2.15'de RTU ağ yapısına örnek görülmektedir.

Uzak uç birimlerinde RTU ve PLC dışında başka cihazlar da bulunabilmektedir. Bunlara birkaç örnek verilecek olunursa;

- i) Otomatik alarm denetleyici; denetlenen sistemde daha önceden ayarlanmış bir alarm koşulu gerçekleştiğinde, siren çalma ve cep telefonuna SMS gönderebilme yeteneğine sahiptir [12].
- ii) Dairesel modem (circular modem); GSM ve diğer hücresel iletişim aygıtlarını kullanarak RTU'ya erişim imkanı sağlamaktadır [12].

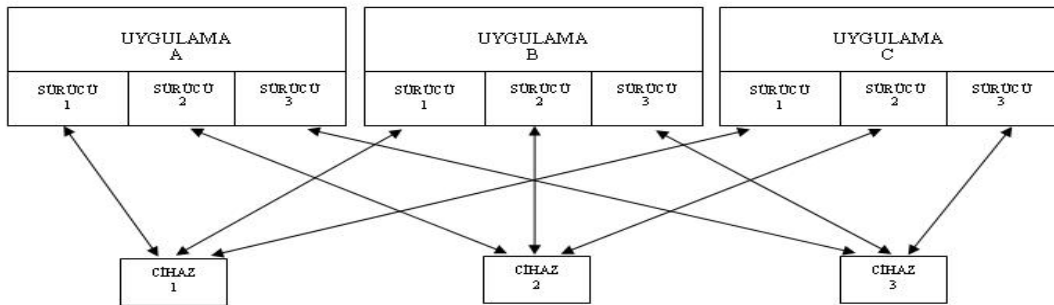


Şekil 2.14: RTU ağ yapısı

iii) Web kamera (Web CAM); sistemle entegre çalışarak sistemin uzaktan izlenmesine olanak sağlar.

BÖLÜM 3. OPC NEDİR?

Günümüzde üretim endüstrisinin vazgeçilmezleri arasında bulunan gözetleyici denetim ve veri toplama (SCADA) paket yazılımları üretim sektörüne yüksek maliyetler getirmektedir. Bu yüksek maliyetlerin başlıca nedeni, yazılımı üreten firmaların yapmış oldukları tekelleşmelerdir. SCADA paket yazılımı üreten firmalar, süreç denetiminde kullanılan cihazlardan sadece kendi üretiminde olan yapıların ilgili yazılımla haberleşmesine izin vermektedir. Bazen kendi ürettikleri cihazlar da dahil olmak üzere ilgili firmanın yazılımı ile haberleşme olanağına sahip değildirler. Örnek olarak SIEMENS'in SCADA yazılımı olan WINCC yapısı SIEMENS ailesinin S7-200 yapısındaki süreç denetim cihazlarını tanımamaktadır. Bu ve buna benzer durumlar, üretim sektörüne kullandığı yazılıma göre cihaz seçme zorunluluğu getirmekte ve bu sebeple yüksek maliyetlere neden olmaktadır. Ayrıca endüstriyel otomasyon dünyasında SCADA sistemlerinin yanında farklı ihtiyaçlar için de çeşitli yazılım uygulamaları (DCS, ERP vb.) kullanılmaktadır. Şekil 3.1'de örnek olarak bir üretim endüstrisinde kullanılan üç farklı yazılım uygulaması ve üç farklı süreç denetim cihazı arasındaki haberleşme yapısı görülmektedir. Sistemde kullanılan her bir yazılım uygulaması, sistemde bulunan her bir cihaz ile haberleşmek için ilgili cihazın sürücüsünü bulundurmak zorundadır. Yaşanan bu zorunluluk, sisteme maddi bir yük getirmekte ve veri transferinde karışıklığa da neden olmaktadır.

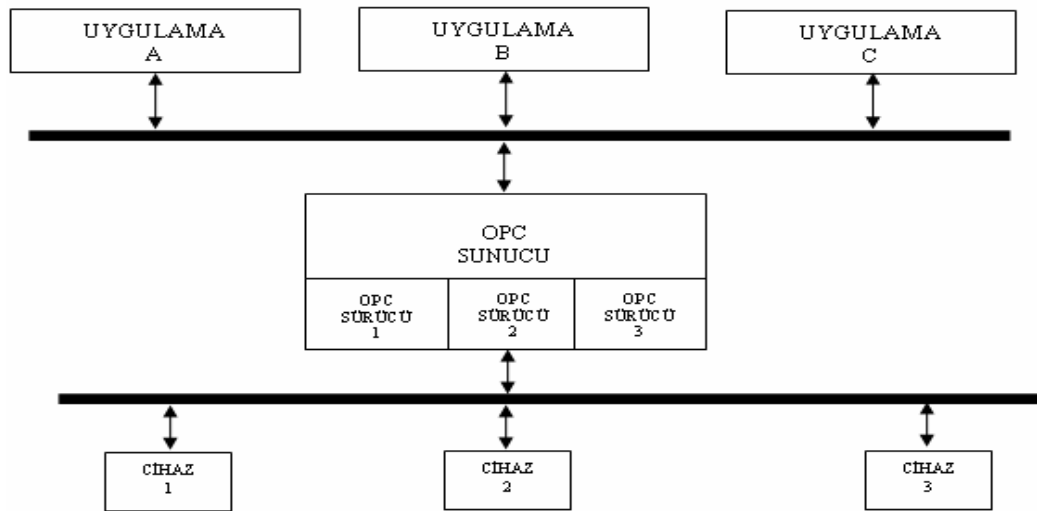


Şekil 3.1: Klasik Haberleşme yapısı

Endüstriyel otomasyonda, Şekil 3.1’de gösterilen karmaşık haberleşme yapısının yaşanması, kullanılan uygulamanın cihaz seçiminde getirdiği kısıtlamalar üretim sektörünü yeni oluşumlara yöneltmiştir. Bu sebeple, 1996 yılında Fisher-Rosemount, Rockwell Software, Opto 22, Intellution ve Intuitive Technology firmaları OPC foundation adı ile oluşturdukları organizasyonun çatısı altında birleşerek OPC standardını oluşturmuşlardır. Başlangıçta beş firma ile kurulan bu organizasyonun günümüzde 300 ün üzerinde üyesi bulunmaktadır [21].

Şekil 3.2’de görülen OPC standardındaki haberleşme yapısı incelendiğinde, OPC sunucusu sistemde kullanılan cihazlarla yazılım uygulamaları arasında doğrudan haberleşme olanağı sağlamaktadır. OPC sunucusu ile cihaz arasında veri alışverişinin gerçekleşebilmesi için ilgili cihazın OPC sürücüsünün sunucuda bulunması gereklidir. Kullanıcı bu sürücüyü cihazın üretici firmasından kolaylıkla temin edebilmektedir.

OPC, sistemde kullanılan yazılım uygulamalarıyla haberleşme olanağına sahip olabildiği gibi, görsel arayüz oluşturma, verilerin depolanması ve bunun gibi çeşitli özelliklerin, firmalar tarafından üçüncü parti yazılımlarla (Delphi, Visual basic, Visual C vb.) üretilmesine de olanak sağlamaktadır. Şekil 3.2’de OPC standardı ile elde edilen haberleşme yapısı görülmektedir.



Şekil 3.2: OPC Standardı haberleşme yapısı

Sistemde kullanılan cihazlar ile ilgili uygulamalar arasındaki haberleşmeye daha basit bir yapı kazandıran OPC standardı altyapısında, Microsoft'un en önemli istemci/dağıtıcı teknolojisi olan OLE/COM teknolojisini kullanmaktadır. Microsoft'un OLE, COM nesnelere Microsoft'un Visual Basic, Visual C++ paket programları ve Borland's Delphi paket programında mevcut olarak bulunmaktadır.

3.1. OLE

OLE, Microsoft'un farklı uygulamalar arasında veri transferinin gerçekleşmesini sağlayan DDE (Dynamic Data Exchange) nesnesinin düşük bant genişliği nedeniyle yerine getirmiş olduğu bir yapıdır [22]. Bu yapının amacı verilere servis sağlayıcılar yoluyla erişmektedir.

OLE; bir sunucu uygulamasının, Windows Registry'de saklanan verilere oluşturulan bir referansla ulaşmasına olanak verir. Ulaşılan veri sunucu uygulamasına kopyalanabilir (Object Embedding) veya orijinal dosyada saklanabilir (Object Linking) [23].

3.2. COM

COM (Component Object Model), bir istemci ya da sunucu modülünün belirli bir arabirim üzerinden iletişim kurmaları için standart bir metod tanımlayan bir teknolojidir. Burada "modül", aynı veya farklı bilgisayarlar üzerinde çalışan bir uygulama veya kütüphaneyi (DLL: Dynamic Link Library: Dinamik kütüphane bağlantısı) gösterir. Bu iki modül aynı bilgisayar üzerinde, yada bir iletişim ağı aracılığı ile birbirine bağlanmış olan farklı makinalar üzerinde çalışabilir. COM arabirimlerinin amacı iki yazılım modülü, iki çalıştırılabilir dosya, ya da bir çalıştırılabilir dosyayla bir DLL arasında iletişim kurmaktır. COM nesnelere DLL'lerde uygulanması genellikle daha basittir. Çünkü bir program ve kullandığı DLL aynı bellek adresi alanında durur [23].

3.3. DCOM

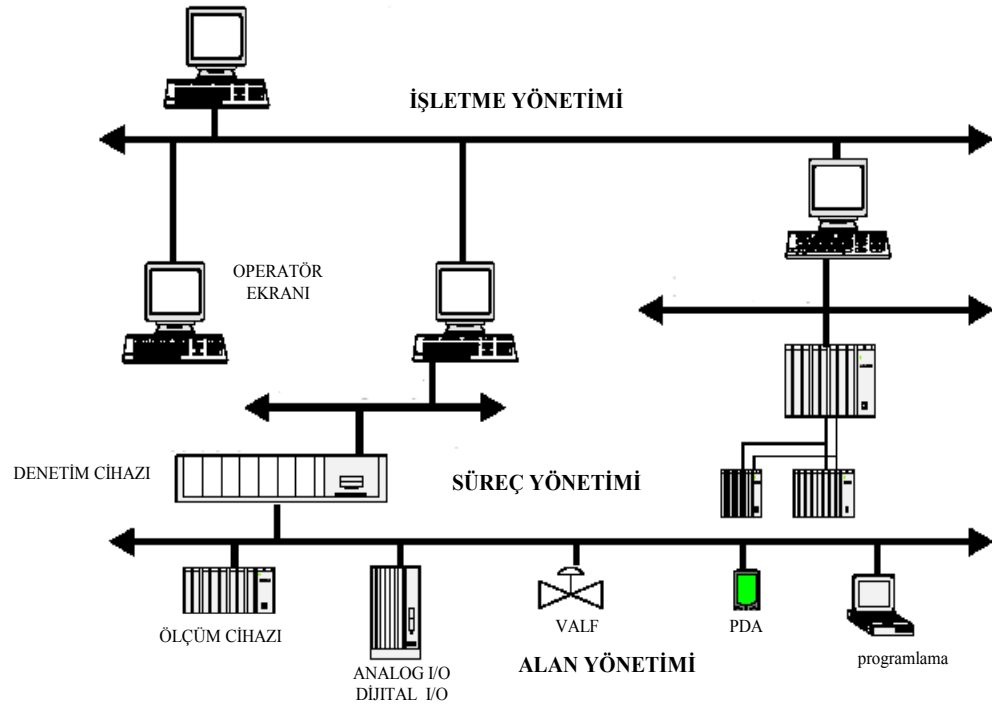
DCOM, ağ üzerindeki COM'lardan oluşmaktadır. DCOM, Network üzerindeki bilgisayarda bulunan bir uygulama olabileceği gibi bir kütüphane de olabilir. DCOM bir istemciye COM nesnelere bir başka bilgisayar üzerinde de kullanma imkanı tanıyan optimize edilmiş bir protokoldür.

3.4. Genel OPC Yapısı

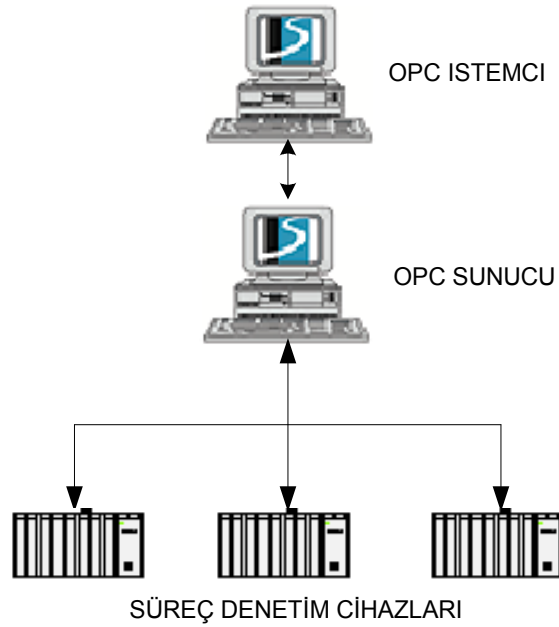
Üretim sektörünün en önemli alt yapısını oluşturan otomasyon sistemlerinden her geçen gün artan beklentiler ve kullanım alanlarındaki genişlemeler sistemin yönetimini zorlaştırmaktadır. Bu sebeple klasik bir süreç denetim yapısının yönetim şekli üç farklı katmana bölünebilir [24]. Bunlar;

- i) Alan Yönetimi (Field Managment): Sistemdeki cihazların oluşturduğu yönetim katmanıdır. Ölçü cihazları, Elektronik valfler, PDA (Personal Digital Assistant: Kişisel Sayısal Yardımcı) gibi yapılar örnek gösterilebilir.
- ii) Süreç Yönetimi (Process Managment): SCADA ve DCS gibi yapıların kullanıldığı yönetim katmanıdır.
- iii) İşletme Yönetimi (Business Managment): Süreç yönetim mimarisinin en üst katmanıdır. Sistemden elde edilen tüm bilgiler bu katmanda toplanır. Bu katmanda toplanan veriler farklı istemci uygulamalarının kullanımına sunulmaktadır. Sisteme ait arşivleme durumları da bu katmanda yapılır

Şekil 3.3'de görülen süreç denetim mimarisi kullanılan sisteme ek maliyetler ve karışık bir yapı sunmaktadır. Oluşturulan OPC standardı ise sisteme daha az maliyet ve daha basit bir yapı sunmaktadır. Şekil 3.4'de bir OPC sunucu ve istemci arasındaki ilişki görülmektedir. OPC sunucu ile ilgili OPC istemci uygulaması aynı bilgisayar üzerinde olabileceği gibi farklı bilgisayarlar üzerinde de olabilir.



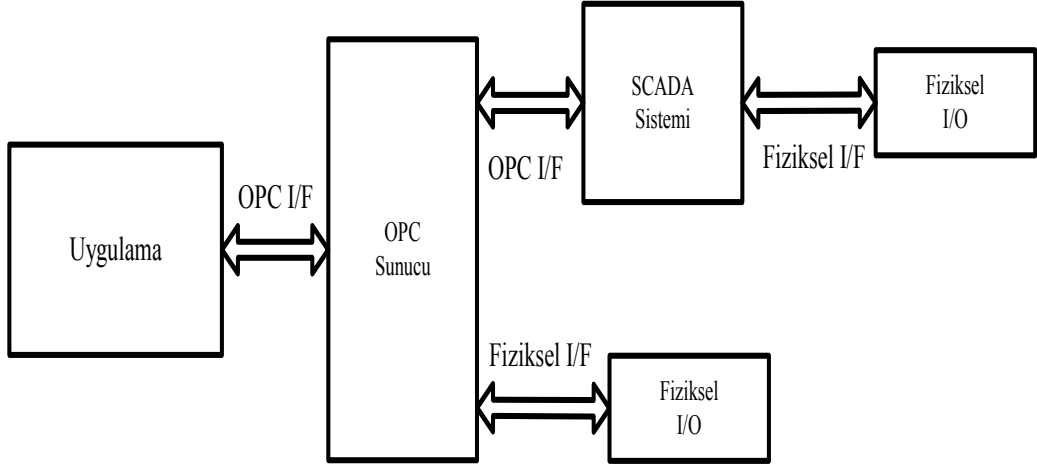
Şekil 3.3: Süreç denetim mimarisi



Şekil 3.4: OPC Sunucu-istemci yapısı

3.4.1. OPC arayüzleri

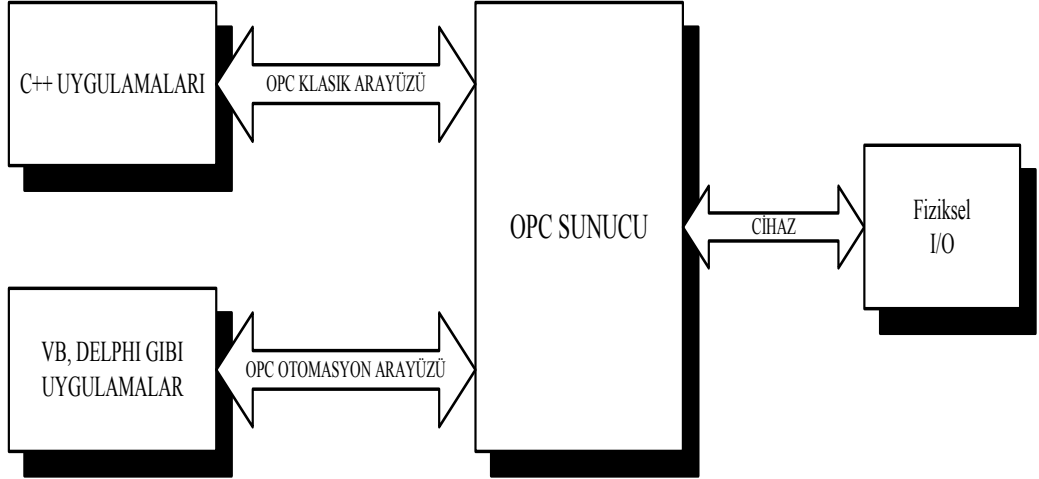
Bir OPC sunucu, gerçek dünyada kullanılan süreç denetim cihazlarından elde ettiği veriyi ilgili OPC istemcilerin (SCADA, DCS, HMI...vb) kullanımına sunmaktadır. OPC sunucu, istemcilerden ve cihazlardan iki yönlü olarak eriştiği verileri farklı istemcilerin kullanımına sunabildiği gibi; Delphi, Visual Basic, Visual C gibi görsel programlama dilleri kullanılarak oluşturulmuş özel yapıdaki uygulamaların da kullanmasına olanak sağlamaktadır. Şekil 3.7’de OPC istemci/sunucu ilişkisi görülmektedir [2].



Şekil 3.5: OPC İstemci/Sunucu İlişkisi

Sistemdeki cihazlar ile çeşitli uygulamalar arasında bir kale kapısı görevi üstelenecek olan OPC sunucu, klasik (custom) ve otomasyon (automation) ara yüzlerine sahiptir. Şekil 3.6’da genel bir OPC mimarisi görülmektedir [2].

Genel bir OPC sunucu, SCADA, DCS gibi C++ tabanlı olan istemci uygulamaları için klasik ara yüz yapısını kullanmaktadır. Otomasyon ara yüzü, görsel bir programlama yapısı ile haberleşme durumlarında kullanılan arayüzdür.



Şekil 3.6: Genel OPC mimarisi

3.5. OPC Özellikleri

OPC özellikleri; OPC topluluğu (OPC foundation) tarafından, OPC sunucunun farklı ihtiyaçların gerçekleştirilmesi amacıyla oluşturulmuş özelliklerdir. Bir OPC sunucuda, kullanılan yapının ihtiyacına göre bir veya birkaç özellik aynı anda kullanılabilir [24]. OPC yapılarının ilk oluşturulmasıyla eklenen ve OPC sunucuda olması istenen üç temel özellik vardır [2]. Bunlar;

- i) OPC Veri Erişimi (OPC Data Access)
- ii) OPC Alarm ve Olaylar (OPC Alarm and Events)
- iii) OPC Tarihsel Veri Erişimi (OPC Historical Data Access)

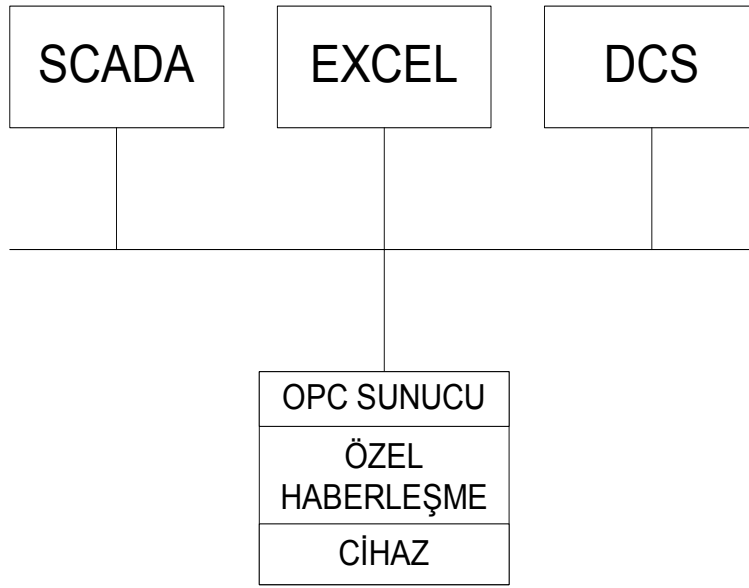
Başlangıçta üç temel özelliğe sahip OPC yapıları, günümüzde üç yeni özelliğin eklenmesiyle altı farklı özelliğe sahip olabilmektedir [21]. Bunlar;

- i) OPC Veri Değişimi (OPC Data eXchange)
- ii) OPC XML Veri Erişimi (OPC XML-DA: OPC XML Data Access)
- iii) OPC Güvenliği (OPC Security)

3.5.1. OPC veri erişimi

OPC veri erişimi (OPC Data Access); süreç denetim cihazları ile OPC istemciler arasındaki haberleşmenin sağlanması amacıyla OLE/COM arayüz protokollerinin

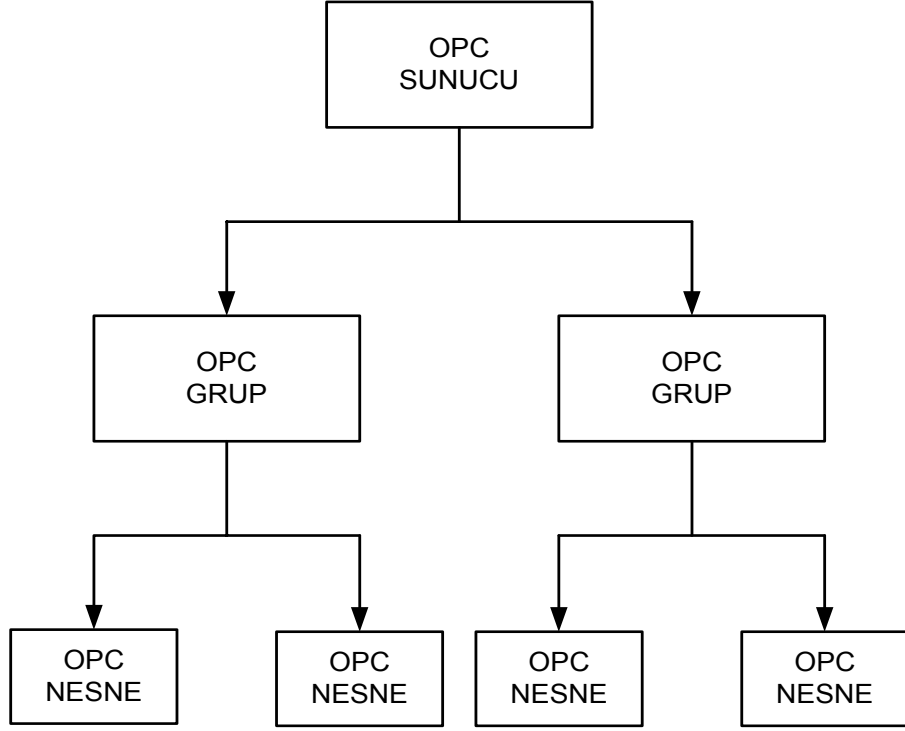
kullanılmasıyla oluşturulmuş bir özelliktir [24]. Bir OPC sunucuda olması istenen en temel özellik olan Veri Erişim özelliği, süreç denetim cihazları ve istemciler arasındaki veri aktarımını gerçek zamanlı olarak gerçekleştirmektedir [8]. Gerçek zamanlı veri aktarımı gerçekleştiren OPC veri erişim özelliği, COM ve DCOM yapılarının kullanılmasıyla geniş bir yapıdaki farklı yazılım uygulamalarına kolayca erişme olanağı sağlamalıdır. Şekil 3.7’de OPC istemciler ve süreç denetim cihazları arasındaki veri erişimine örnek görülmektedir [35].



Şekil 3.7: İstemci/cihaz arasındaki mimari

OPC Veri Erişim özelliği; OPC sunucu, OPC grup ve OPC nesnesi (OPC item) tarafından oluşturulan hiyerarşik bir yapıya sahiptir [3]. Şekil 3.8’de OPC sunucunun sahip olduğu veri diyagram hiyerarşisi görülmektedir.

OPC veri diyagram hiyerarşisi incelendiğinde, en alt yapıda OPC nesnelere, bir üst yapıda OPC nesnelere içeren OPC grupları, en üst yapıda ise OPC sunucu bulunmaktadır.

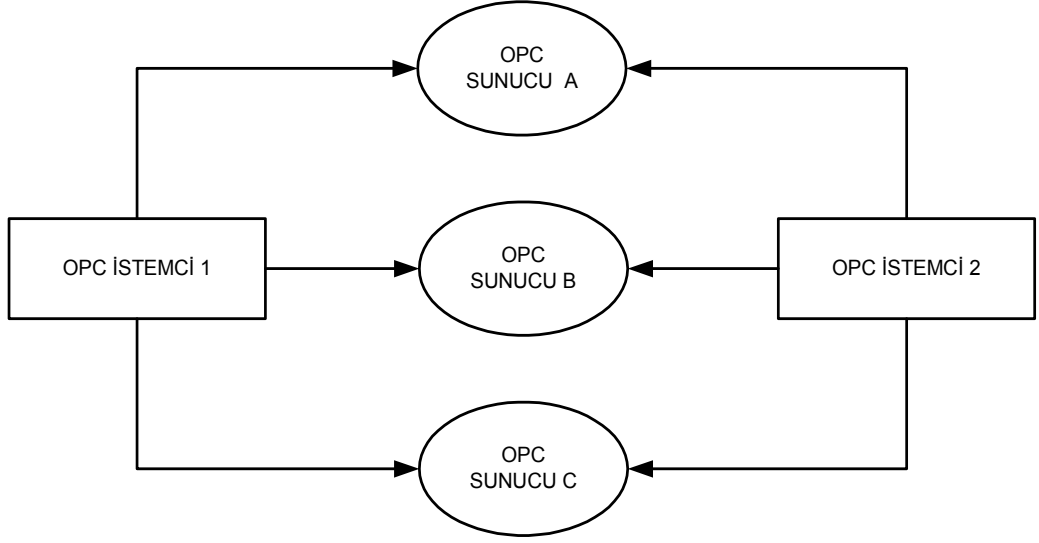


Şekil 3.8: OPC sunucu veri diyagram hiyerarşisi

Hiyerarşinin en alt yapısında bulunan OPC nesneler; sistemdeki “TAG” ları, diğer bir deyişle, dış dünyada denetlenen sistemdeki cihazlarla haberleşme noktalarını temsil etmektedir.

OPC grupları; OPC nesnelere elde ettiği verileri OPC sunuculara ileten yapılardır. Bir OPC sunucu, birden fazla OPC grubu, bir OPC grubu da birden fazla OPC nesnesi bulundurabilir [3]. Örnek olarak bir elektrikli metro SCADA sistemi incelenecek olunursa; herhangi bir istasyonun, hat akımı, hat gerilimi ve sistemdeki anahtarların durumları istasyonda bulunan operatör tarafından görüntülenip merkezde bulunan OPC sunucuya aktarılabilir. Yerel yapıdaki verilere erişen ve merkeze ileten OPC sunucusu, OPC grup olarak adlandırılabilir [3].

OPC Veri Erişim özelliği; bir OPC istemcinin birden fazla OPC sunucuyla eş zamanlı haberleşme sağlayan özelliğidir. Benzer şekilde OPC sunucunun da birden fazla OPC istemciyle haberleşmesine olanak sağlamaktadır. Şekil 3.9’da OPC istemci-sunucu yapısı görülmektedir [21].

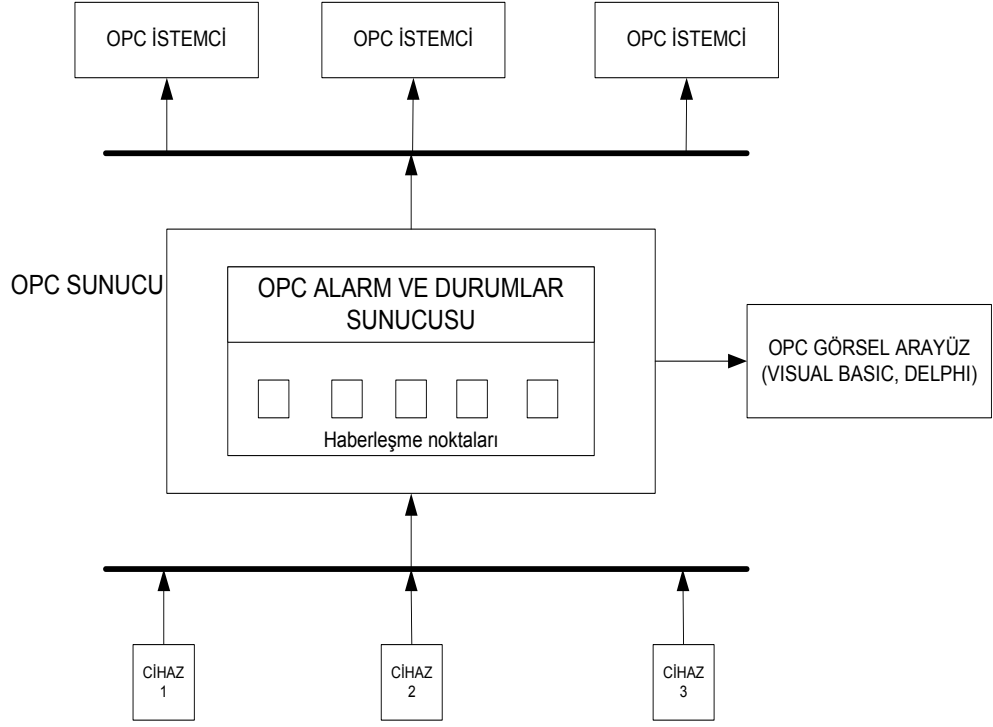


Şekil 3.9: OPC İstemci/Sunucu yapısı

3.5.2. OPC alarm ve olaylar

OPC Alarm ve Olaylar özelliği (OPC Alarm and Events); temelde OPC Veri erişim özelliği kullanılarak oluşturulan bir yapıdır. Denetlenen süreçteki alarm ve olaylarla ilgili verileri OPC istemcilere gönderen veya oluşturulan OPC görsel ara yüzünde, operatörün anlayabileceği yapıya dönüştüren OPC özelliğidir . Denetlenen sistemde sıcaklığın belirlenen sınır değerini aştığında, operatör ekranında bir metin mesajı veya ekrandaki ilgili resmin renk değiştirilerek operatörün uyarılması bu özelliğe örnek olarak gösterilebilir.

Bir OPC sunucunun sahip olduğu Alarm ve Olaylar özelliği; OPC sunucu içerisine oluşturulan; Alarm ve Olaylar sunucu yapısı tarafından gerçekleştirilmektedir. OPC Alarm ve Olaylar sunucusu; OPC sunucunun gerçek dünyadaki cihazlarla haberleşme noktalarını denetleyen yapılardır. Haberleşme noktalarında meydana gelebilecek herhangi bir değişiklik, OPC Alarm ve Olaylar sunucusu tarafından gerçek zamanlı olarak OPC sunucunun kullanımına sunulmaktadır [21]. Şekil 3.10'da OPC sunucu ve alt birimi olan OPC Alarm ve Olaylar sunucusu arasındaki hiyerarşi görülmektedir.



Şekil 3.10: OPC sunucu ve OPC Alarm ve Olaylar sunucu arasındaki hiyerarşik yapı

OPC Alarm ve Olaylar özelliği; sistemde oluşabilecek mesajları üç ayrı yapıda sınıflandırmaktadır [24]. Bunlar;

i) Temel Olaylar (Simple Events): Sistemin çalıştırılması veya durdurulması, operatörün sisteme giriş veya çıkış durumları gibi olayları bildiren mesajlar örnek olarak gösterilebilir. Herhangi bir özel müdahale gerektirmezler. Kullanıcıya temel bilgi amaçlı sunulan mesajlardır.

ii) İzlenen Olaylar (Tracking Events): İzlenen Olaylar, temel mesajlarla benzer yapıdaki mesajlardır. Oluşan bir mesaja herhangi bir özel müdahale gerektirmezler. Bu mesajları temel mesajlardan ayıran en büyük özelliği olayın kaynağını da bildirmesidir. Operatörün sistemde ayarlanmış olan bir değeri değiştirmesine ait mesajlar bu sınıfa örnek olarak gösterilebilir.

iii) Durum Olayları (Condition Events): Özel müdahalelerin gerektiği mesajlar bu sınıfta toplanabilir. Örneğin sistemde çalışan bir motorun arızalanması durumunda, operatöre bunu bildiren ve ikinci motorun sisteme alınıp alınmamasıyla ilgili onay durumunu soran mesajlar bu sınıfa örnek gösterilebilir.

3.5.3. OPC tarihsel veri eriřimi

Tarihsel Veri Eriřimi (OPC Historical Data Access (HDA)) özelliđi, süreç denetimi esnasında elde edilen geçmiş verilere erişmeye olanak sağlayan özelliştir. Bir nevi arřivleme özelliđi olan HDA sayesinde, sisteme ait geçmiş verilere erişerek, sistem hakkında ayrıntılı bilgiye sahip olunabilmektedir.

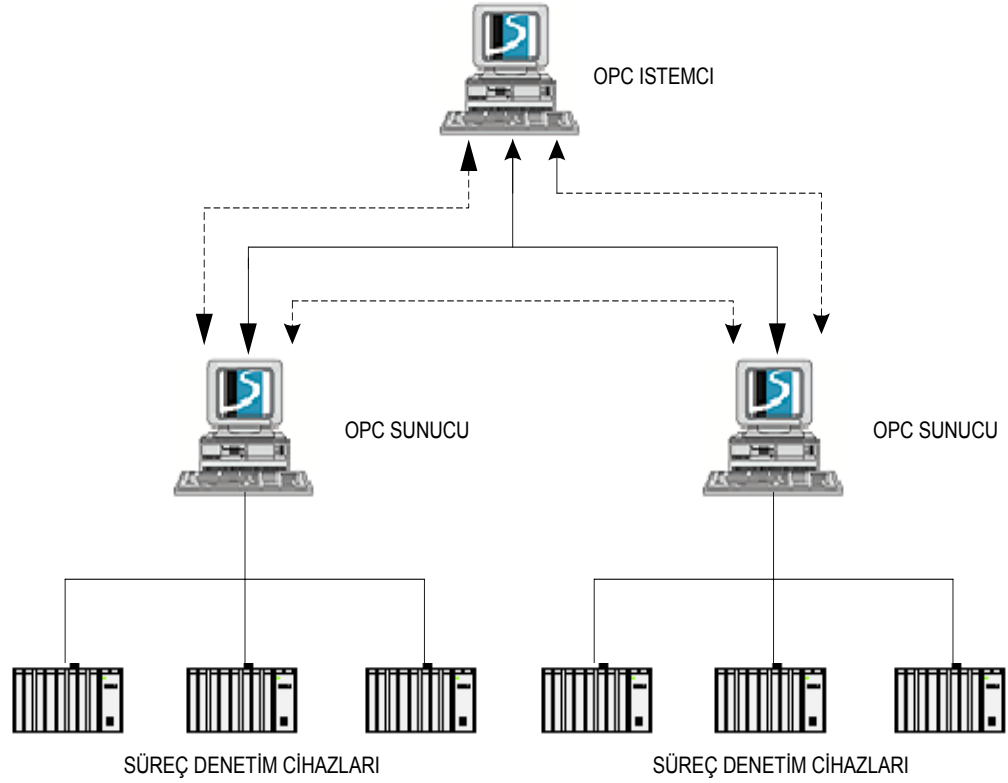
Bir OPC sunucunun denetlediđi sisteme ait geçmiş verilere erişim; OPC sunucusun alt birimi olan HDA sunucusu tarafından gerçekleştirilmektedir. HDA sunucu sayesinde; sisteme ait geçmiş bilgiler, tarih, saat ve nedenleriyle birlikte depolanarak saklanmaktadır [24].

OPC HDA özelliđi, iki farklı sunucu yapısını desteklemektedir [24]. Bunlar;

- i) Sade Eğilim veri sunucuları (Simple Trend data servers): Bu sunucu yapısı; sistemden alınan verileri, zaman, deđer ve nitelik olarak saklar. Sistemde bulunan istemci yapılarına aynı formatta sunar.
- ii) Karmařık veri sıkıřtırma ve analiz sunucuları (Complex data compression and analysis servers): Bu yapıda olan sunucular verileri sıkıřtırarak saklarlar. Arřivlenen verilere ek bilgiler eklenebilmektedir.

3.5.4. OPC veri deđiřimi

OPC Veri deđiřim özelliđi (OPC Data eXchange: OPC DX); Ethernet, Fieldbus gibi haberleřme yapılarının kullanılarak, sunucu-sunucu ve istemci-sunucu arasında veri deđiřimlerine olanak sağlayan bir özelliştir [21]. Bu sayede uygulamadaki süreç denetim cihazlarının birbirleriyle haberleřmeleri de sağlanmaktadır. Őekil 3.10'da OPC DX hiyerarřisine bir örnek görölmektedir.



Şekil 3.10: OPC DX Hiyerarşisi

OPC DX özelliği; kullanıldıkları sistemde aşağıdaki işlevsellikleri destekleyerek farklı yapıların kullanılmasına olanak sağlamaktadır [24]. Bunlar;

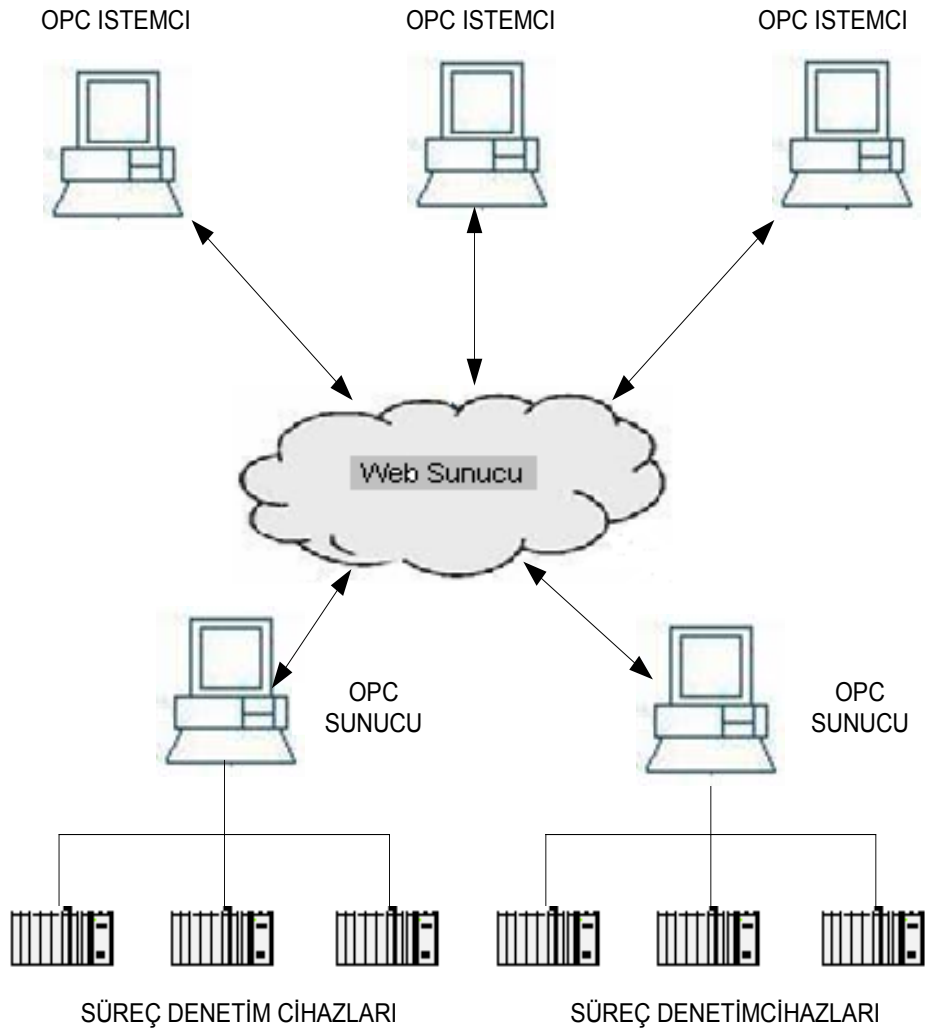
i) Sunucu-Sunucu arası haberleşme (A server to server communication): OPC DX özelliğinin en temel işlevidir. OPC sunucular arasında veri transferinin gerçekleştiği yapıdır.

ii) Uzak sunucu yapısı (A remote server configuration): Bu işlev sayesinde, OPC DX istemciler, standart DCOM ve XML tabanlı ara yüzler kullanarak, network üzerindeki OPC DX cihazlara müdahale etme yetkilerine erişebilirler.

iii) Web teknolojilerini ve farklı ağ yapıları üzerinden haberleşmenin desteklenmesi (Support for web technologies and communication over differenet network): OPC DX özelliği klasik haberleşme yapısı olarak Profibus, Fieldbus, Ethernet gibi yapıları kullanmaktadır. Bu haberleşme yapılarının yanında son yıllarda gelişmekte olan SOAP (Simple Object Access Protocol: Basit Nesne Erişim Protokolü) ve Microsoft.NET haberleşme yapılarını da desteklemektedir.

3.5.5. OPC XML veri eriřimi

OPC XML Veri Eriřimi (OPC XML-DA: OPC XML Data Access) özelliđi; XML (Extensible Markup Language) ve SOAP temelleri üzerine kurulmuř olan OPC XML-DA özelliđi; OPC sistemlerinin internet üzerinden denetimini gerekleřtirmek amacıyla oluřturulmuř bir Web Servis yapısıdır [26]. Oluřturulan bu özellik sayesinde OPC istemciler dnyanın herhangi bir noktasından verilere eriřim imkanına sahip olabilmektedir. Őekil 3.11’de OPC XML özelliđi ile elde edilen hiyerarři grlmektedir.



Őekil 3.11: OPC XML ve İnternet

3.5.6. OPC güvenliđi

OPC Güvenliđi (OPC Security); istemcilerin, sunucu ile haberleşme durumlarını denetleyen özelliktir. Bir OPC sunucu yapısında, iki farklı güvenlik ara yüzü bulunmaktadır [24]. Bunlar;

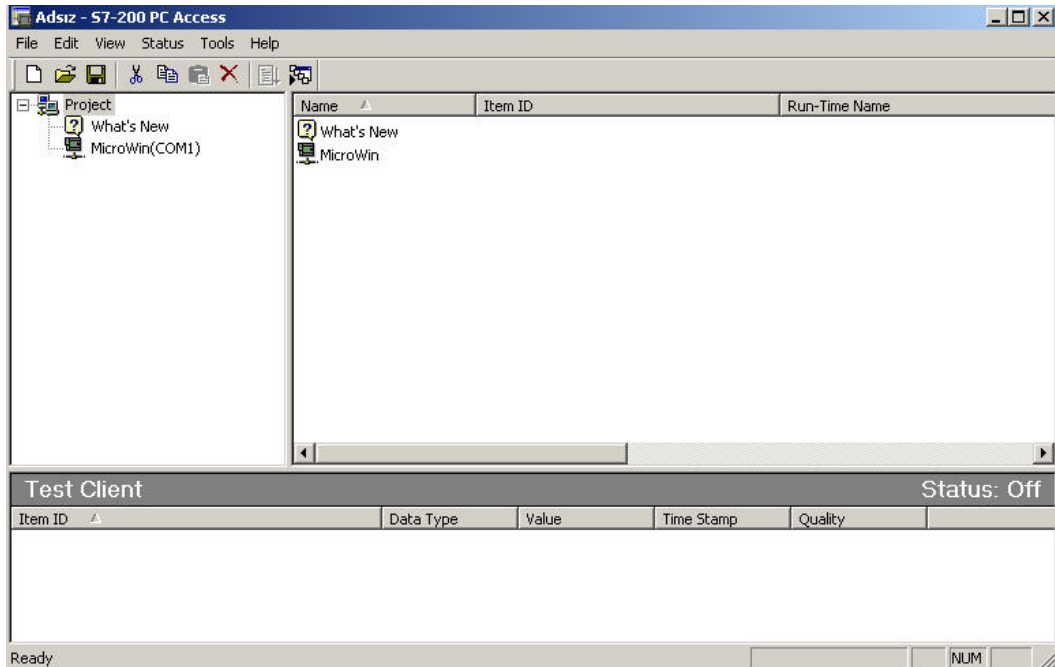
- i) IOPCSecurityNT ara yüzü; MS Windows NT veya daha sonraki platformlar üzerinde çalışan istemciler için oluşturulmuş bir ara yüzdür. Sistemdeki her bir istemciye ait özel bir kullanıcı numarası (user ID) ve şifre yoktur.
- ii) IOPCSecurityPrivate ara yüzü; MS Windows NT platformu dışında çalışan istemciler için oluşturulmuş bir ara yüzdür. Sistemdeki her bir istemciye ait kullanıcı numarası (user ID) ve şifre bulunmaktadır.

BÖLÜM 4. KLASİK OPC SİSTEMİNİN S7-200 PLC İLE GERÇEKLENMESİ

Bu bölümde, öncelikle uygulama yapısında kullanılan SİEMENS firmasına ait OPC sunucu tanıtılmaktadır. Farklı noktalarda oluşturulan OPC sistemleri ve bu sistemlere ait yerel denetim yapıları da bu bölümde ele alınmaktadır.

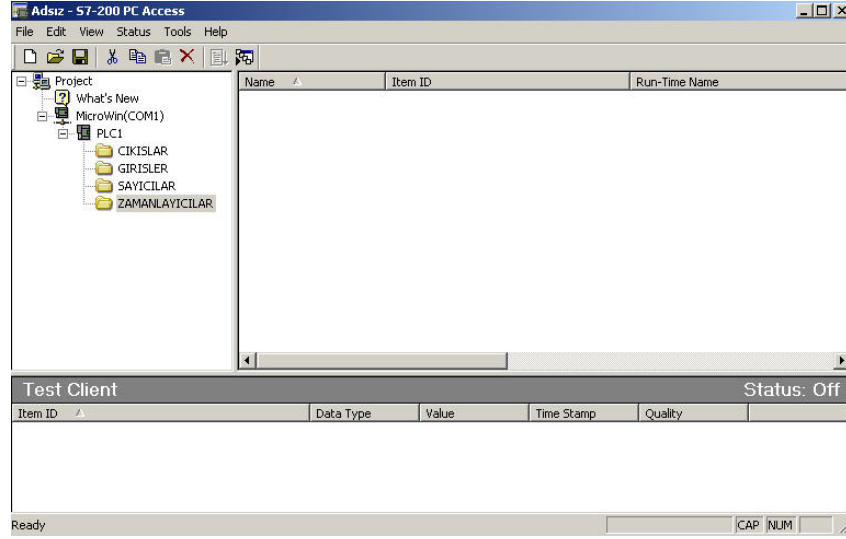
4.1. S7200 PC Access OPC Server Yazılımı

SIEMENS PC Access OPC Server yazılımı; sahada bulunan cihazlarla oluşturulan OPC nesnelere ile haberleşmektedir. OPC sunucu elde ettiği verileri ilgili programların hizmetine sunmaktadır. OPC sunucu oluşturulan her bir nesne için ayrı bir 'ID' (IDENTITY: Kimlik) yapısı atamakta, ilgili üçüncü yapıdaki program, ilgili nesneden gerekli verileri bu ID yapısı üzerinden almaktadır. Şekil 4.1'de kullanılan OPC sunucunun ana denetim ekranı görülmektedir.

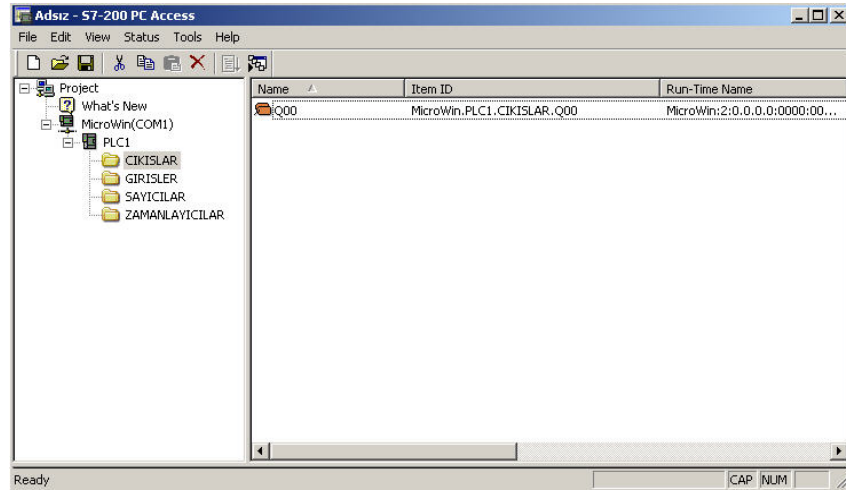


Şekil 4.1: OPC sunucu ana denetim ekranı

Şekil 4.1’de görülen ana denetim ekranında öncelikle sistemde kullanılacak PLC noktaları tanımlanmaktadır. Oluşturulacak OPC nesneleri kullanıcı tarafından oluşturulabilecek dosyalar altında gruplandırılabilceği gibi doğrudan PLC noktası altında da sıralanabilir. Şekil 4.2’de PLC1 noktası altında oluşturulmuş “CIKISLAR, GIRISLER, ZAMANLAYICILAR, SAYICILAR” adlı gruplar ve Şekil 4.3’de oluşturulan “CIKISLAR”adlı grup altında atanan “Q00” adlı OPC nesnesi görülmektedir. Bu yapı sistemde oluşturulan tüm PLC noktaları için gerçekleştirilebilmektedir.



Şekil 4.2: OPC sunucuda oluşturulan gruplar



Şekil 4.3: Gruplar altında oluşturulan OPC nesnesi

Sistemde oluşturulan her bir OPC nesnesi için gerekli olan bazı özellikler kullanıcı tarafından girilmektedir. Bunlar;

- i) veri tipi,
- ii) veri adresi,
- iii) oluşturulan nesne üzerinden yalnız veri okuma,
- iv) oluşturulan nesne üzerinden yalnız veri gönderme,
- v) oluşturulan nesne üzerinden veri gönderme ve veri okuma,
- vi) oluşturulan nesneyi tanımlayıcı sembolik isim.

Sistemde oluşturulan “PLC1” noktasına ait “CIKISLAR” grubunun altında “Q00” sembolik isimli OPC nesnesinin oluşturulmasıyla, bu nesne için oluşturulan ID yapısı “MicroWin.PLC1.CIKISLAR.Q00” olarak görünmektedir. Buradaki;

- i) MicroWin: kullanılan cihazın yazılımının ismi,
- ii) PLC1: programda oluşturulan PLC noktasına atanan sembolik isim,
- iii) CIKISLAR: PLC1 isimli PLC noktası altındaki oluşturulmuş grubun adı,
- iv) Q00: Q0.0 çıkışına atanan sembolik isim. Bu isim Q00 oluşturulabildiği gibi “CIKIS” olarak da oluşturulabilir.

Name	Item ID	Address	Data Type	EU Low
I00	MicroWin.PLC1.I00	I0.0	BOOL	0.000000
Q00	MicroWin.PLC1.Q00	Q0.0	BOOL	0.000000

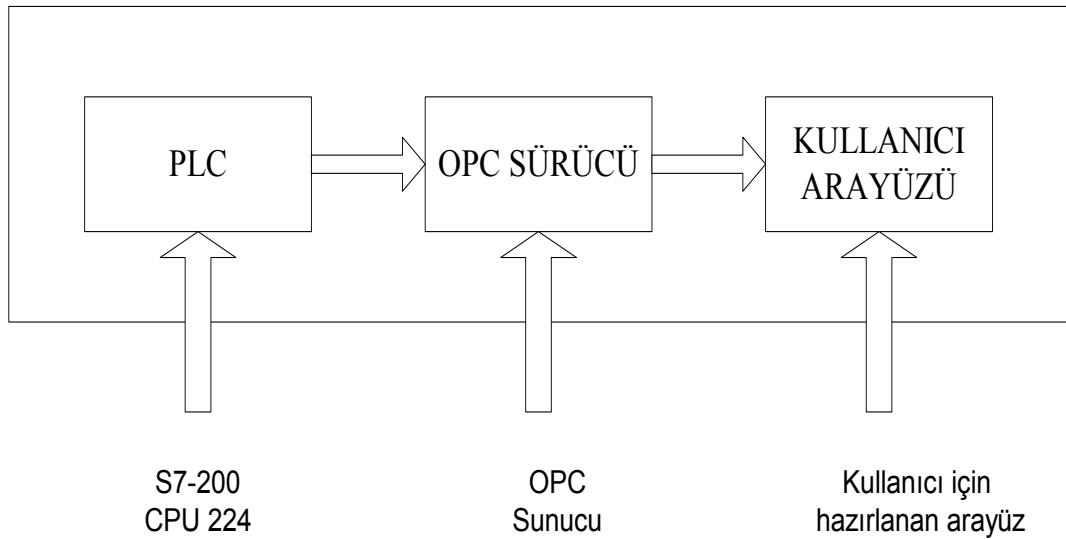
Item ID	Data Type	Value	Time Stamp	Quality
MicroWin.PLC1.I00	BOOL	1	00:41:32:750	Good
MicroWin.PLC1.Q00	BOOL	1	00:41:32:750	Good
MicroWin.PLC1.T37	INT	+2880	00:46:20:546	Good

Şekil 4.4: OPC nesnelerinin gerçek zamanlı incelenmesi

OPC sunucu kullanıcısı oluşturulan her bir OPC nesnesi ile ilgili bilgileri kullanıcı ekranında rahatlıkla görebilmektedir. Ayrıca nesnelere meydana gelen değişiklikleri gerçek zamanlı olarak da takip edebilmektedir. Şekil 4.4 de Q0.0, I0.0 ve T37 için oluşturulan OPC nesnelerinin gerçek zamanlı olarak okunan değerleri görülmektedir.

4.2. Örnek Vinç ve Taşıyıcı Bant Uygulamasının OPC Tabanlı Denetimi

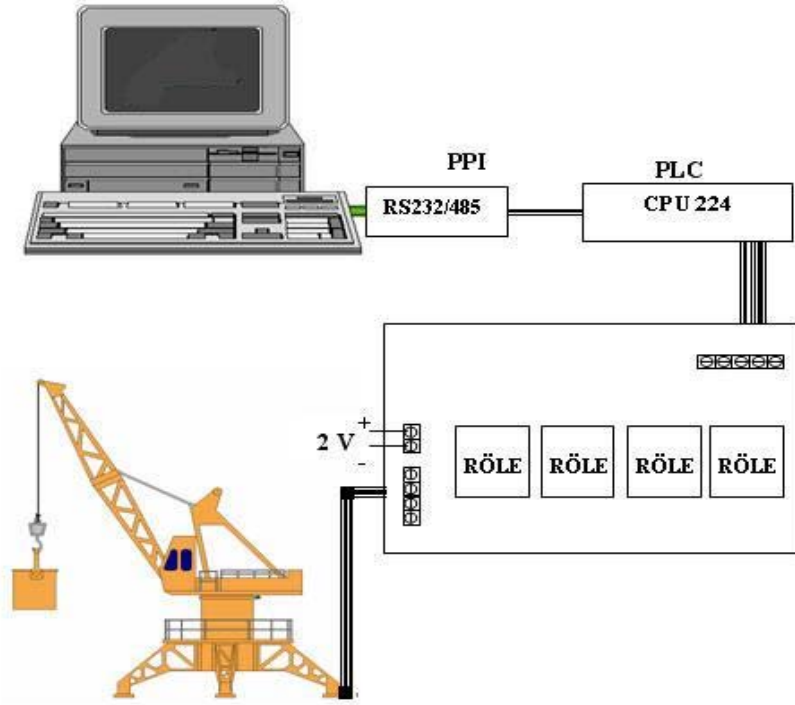
OPC sunucular cihazdan elde ettiği veriyi özel yazılım uygulamalarının (SCADA, DCS, ERP vb.) hizmetine sunabildiği gibi üçüncü parti yazılımların hizmetine de sunabilmektedir. Bu mantık temelinde, laboratuvar ortamında farklı noktalarda değişik uygulama prototipleri oluşturulmuştur. Oluşturulan bu prototiplerden, yerel denetim noktalarında kullanılan OPC sunucu sayesinde elde edilen veriler Delphi paket programında hazırlanmış olan denetim yapısının hizmetine sunulmaktadır. Oluşturulmuş yerel denetim ekranından operatörün sistemi görsel olarak takip etmesi ve sisteme gerekli müdahaleleri yapmasına olanak sağlanmıştır. Şekil 4.5’de oluşturulan yerel yapıların mimarisi görülmektedir.



Şekil 4.5: Klasik OPC tabanlı denetim mimarisi

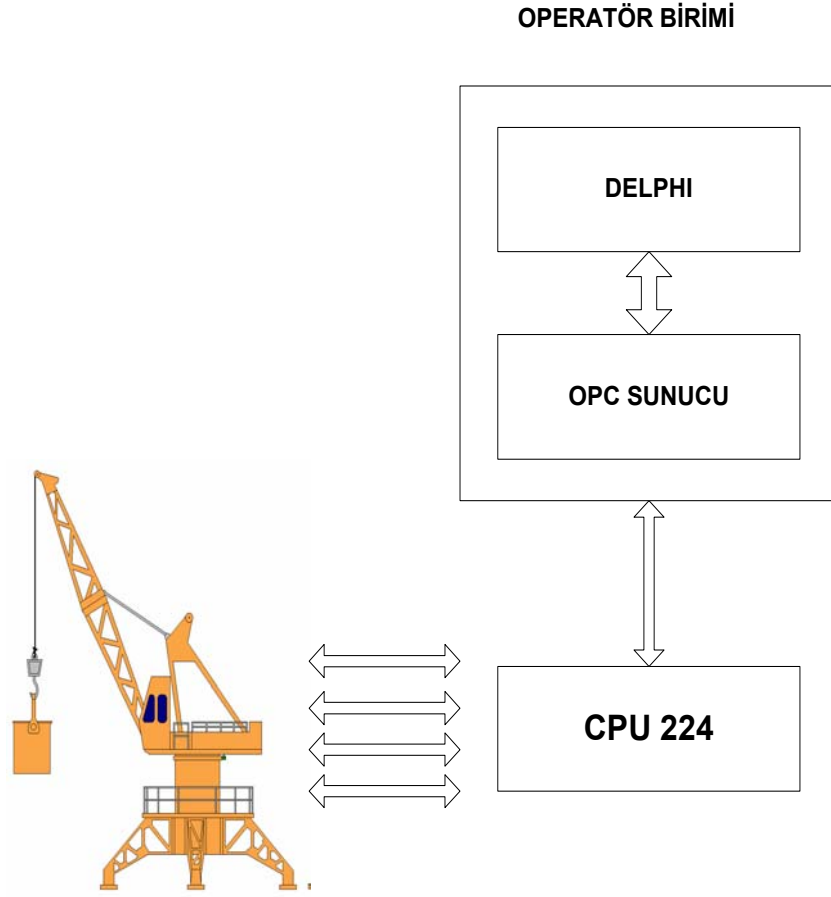
4.2.1. Örnek vinç uygulaması

OPC sistem uygulamasının gerçekleştirilmesi amacıyla Şekil 4.6’de görülen model bir vinç düzeneği hazırlanmıştır. Bu yapı, model bir vinç, Siemens S7-200 CPU 224 PLC, PPI haberleşme kablosu ve bir bilgisayardan oluşmaktadır.



Şekil 4.6: Model vinç uygulaması

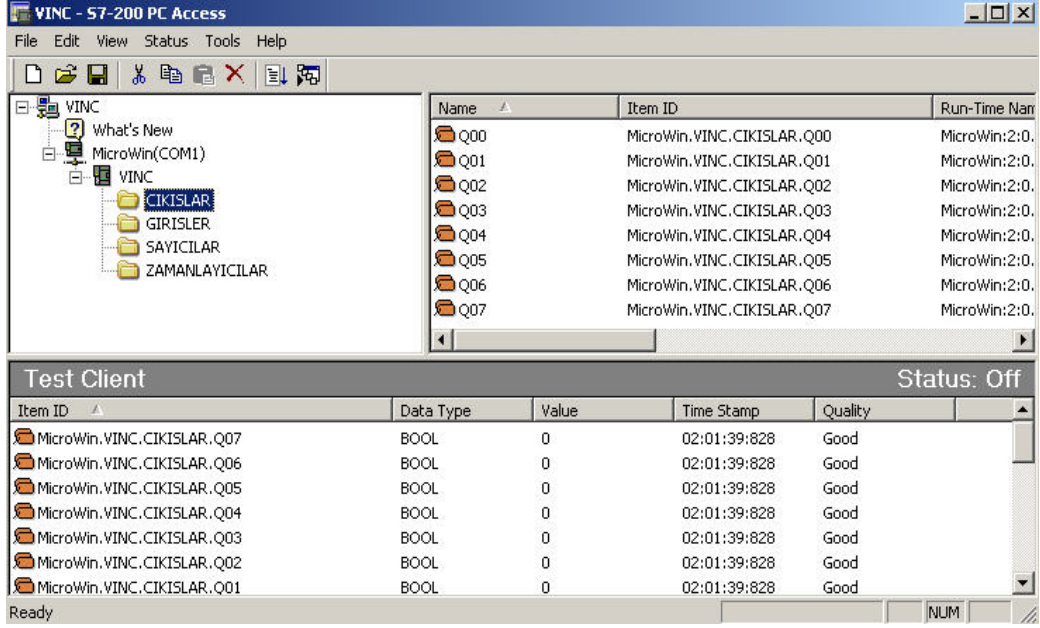
Şekil 4.6’ de gösterilen model vinç 360° kendi ekseninde ve z ekseninde yukarı aşağı hareket edebilme yeteneğine sahiptir. OPC sunucu, Delphi paket programı kullanılarak oluşturulmuş denetim ekranı ile PLC arasında veri alışverişini sağlayan bir köprü olarak görev almaktadır. Şekil 4.7’de örnek vinç uygulama prototipi ve denetim ekranı arasındaki veri akış şeması görülmektedir.



Şekil 4.7: Test düzeneği ver akış şeması

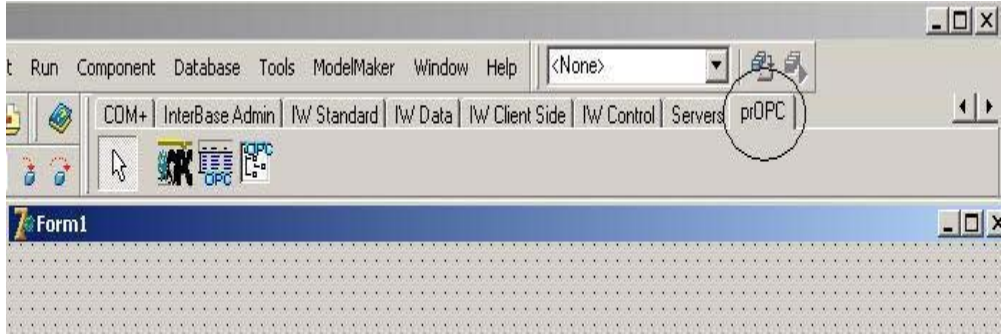
Yerel yapıdaki denetim işlevlerine ait tüm fonksiyonlar, Delphi paket programında oluşturulan yerel denetim ekranından sağlanmaktadır. Operatör biriminin, denetim ekranından sistemden veri talep etme ve müdahale etme ile ilgili tüm noktalara ait OPC nesnelere sunucu içinde önceden tanımlanmıştır. Şekil 4.8’de vinç uygulamasına ait OPC sunucu ekranı görülmektedir.

Yerel denetim ekranının oluşturulacağı Delphi paket programında ilgili OPC sunucu ile haberleşme bağlantısının kurulması için gerekli OPC bileşeninin mevcut olarak bulunması gerekmektedir. Kullanılan OPC sunucu ile Delphi paket programı arasında haberleşme görevini sağlayan bu bileşenler OPC topluluklarının internet sitelerinden kullanıcıların hizmetine sunulmaktadır.



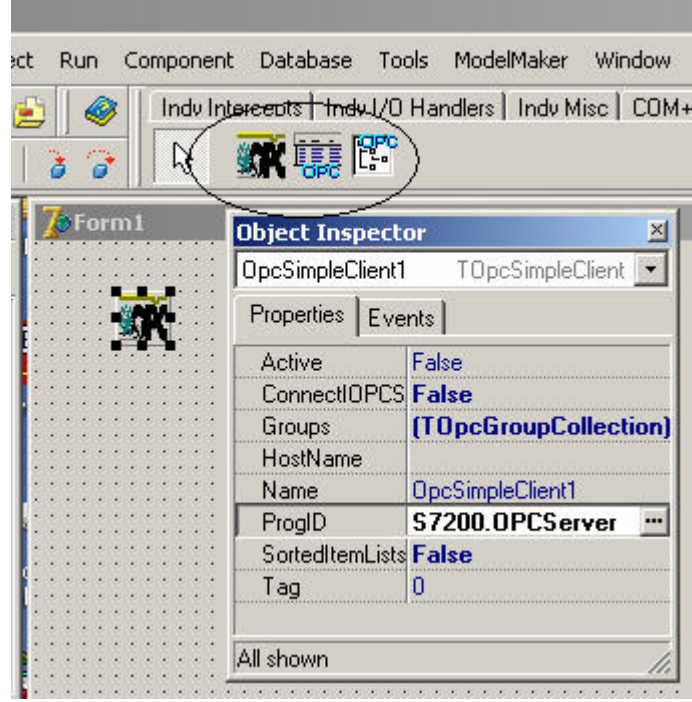
Şekil 4.8: Örnek vinç uygulamasına ait OPC sunucu ekranı

Ücretsiz olarak verilen bu bileşenin Delphi paket programına yüklenmesi ile OPC sunucu ile haberleşmeyi sağlayacak bileşenin oluşturulması sağlanmıştır. Şekil 4.9’da OPC bileşenlerinin Delphi paket program editöründen yüklenmesinden sonra bileşen paletindeki nesneleri görülmektedir.



Şekil 4.9: Delphi paket programında OPC nesneleri

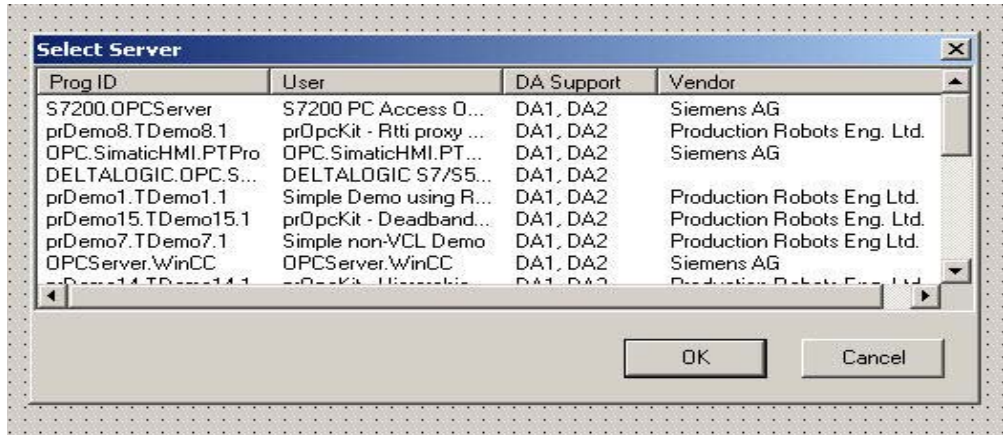
Delphi paket programı ile OPC sunucu arasında veri alışverişini sağlayacak olan OPC bileşeninin öncelikle hangi OPC sunucu ile veri alışverişinde bulunacağı belirlenmektedir. Şekil 4.10’da seçilen “OpcSimpleClient” nesnesi ve nesneye ait özellikler görülmektedir.



Şekil 4.10: OpcSimpleClient nesnesi ve nesne özellikleri

Şekil 4.10’da en önemli olan nokta nesnenin “ProgID” yapısıdır. Bu yapı OPC nesnesinin hangi OPC sunucu ile haberleşmesinin belirlendiği yapıdır. ProgID yapısı bilgisayarda yüklü konumda bulunan OPC sunuculardan seçim hakkı sunmaktadır. Şekil 4.11’de OPC nesnesinin haberleşme işlemini gerçekleştirmek için kullanılan farklı görülmektedir.

Şekil 4.11’de kullanılan yapı sayesinde kullanılmak istenen OPC sunucu kolaylıkla belirlenebilecektir.



Şekil 4.11: Sistemde kullanılacak OPC sunucunun belirlenmesi

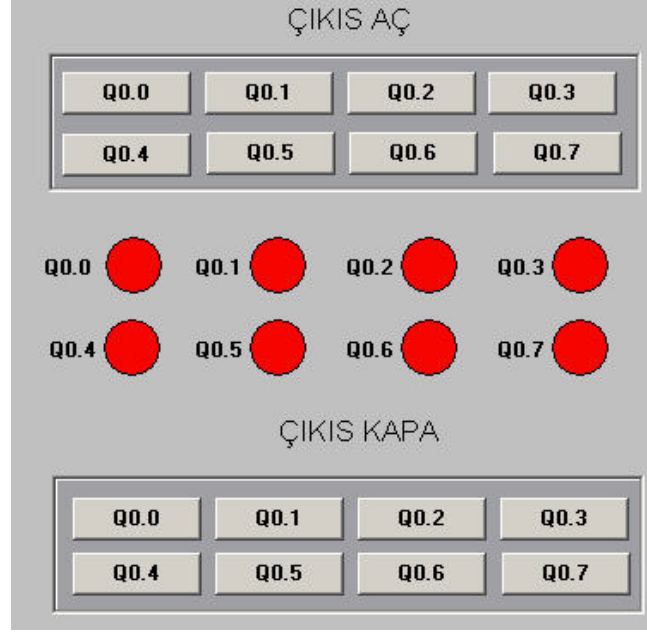
Öncelikle Delphi ortamında oluşturulan denetim programında erişilmek istenen OPC sunucu belirlenmiştir. Daha sonra operatörün OPC sunucuyu aktif hale getirmesi için gerekli olan yapı oluşturulmuştur. Benzer şekilde, operatörün OPC sunucu ile haberleşmeyi durdurması için gerekli olan yapı da oluşturulmuştur. Bu yapı sayesinde, operatörün izin vermesiyle sistem ilgili OPC sunucu ile bağlantısını gerçekleştirmekte veya haberleşme durumunu kesebilmektedir. Şekil 4.12 de OPC sunucusunun iletişime geçmesi veya iletişimi durdurmasını sağlayan kullanıcı arabirimi görülmektedir.



Şekil 4.12: OPC sunucu haberleşme görsel yapısı

OPC sunucu haberleşme izin yapısı oluşturulduktan sonra operatörün PLC çıkışlarına müdahalede bulunabileceği denetim yapısı gerçekleştirilmiştir. Bu sebeple PLC yapısının sekiz ayrı çıkışı için ayrı bir denetim butonu oluşturulmuştur. Oluşturulan butonlardan OPC sunucuda mevcut olarak bulunan ilgili çıkış nesnesinin aktif olması için gerekli fonksiyon parametresinin gönderilmesi sağlanmıştır. Benzer yapıyla oluşturulan diğer bir nokta ise operatörün ilgili çıkışı pasif hale getirmesi için gerekli komutu gönderebileceği yapı hazırlanmıştır. Şekil 4.13’de operatörün PLC çıkışları aktif veya pasif edebileceği denetim yapısı görülmektedir.

Operatör, denetim ekranından çıkışlara müdahale edebildiği gibi, operatörün sisteme ait girişlerinde durumunu denetlemesi için ayrı yapı oluşturulmuştur. Oluşturulan bu yapı ile operatörün hangi girişin aktif olduğuna dair bilgi sahibi olabilmesi sağlanmıştır. Şekil 4.14’de girişlere ait denetim yapısı görülmektedir.



Şekil 4.13: PLC çıkışları denetim ekranı



Şekil 4.14: PLC girişlerine ait denetim ekranı

Şekil 4.14’de görülen giriş denetim butonunun aktif hale getirilmesi ile yapı OPC sunucudan girişlere ait OPC nesnelерinin durumları inceleyip kullanıcıya bilgi sunmaktadır. Örnek olarak PLC’nin 0,1 ve 2 nolu girişinin aktif olduğu durumda giriş denetim butonunun aktif edilmesiyle 0,1, ve 2 nolu giriş durum sinyalleri kırmızı renkten yeşil renge dönmektedir. Şekil 4.15’de 0,1 ve 2 nolu girişlerin aktif olması ve girişlerin okunması durumunda oluşan görüntü görülmektedir.

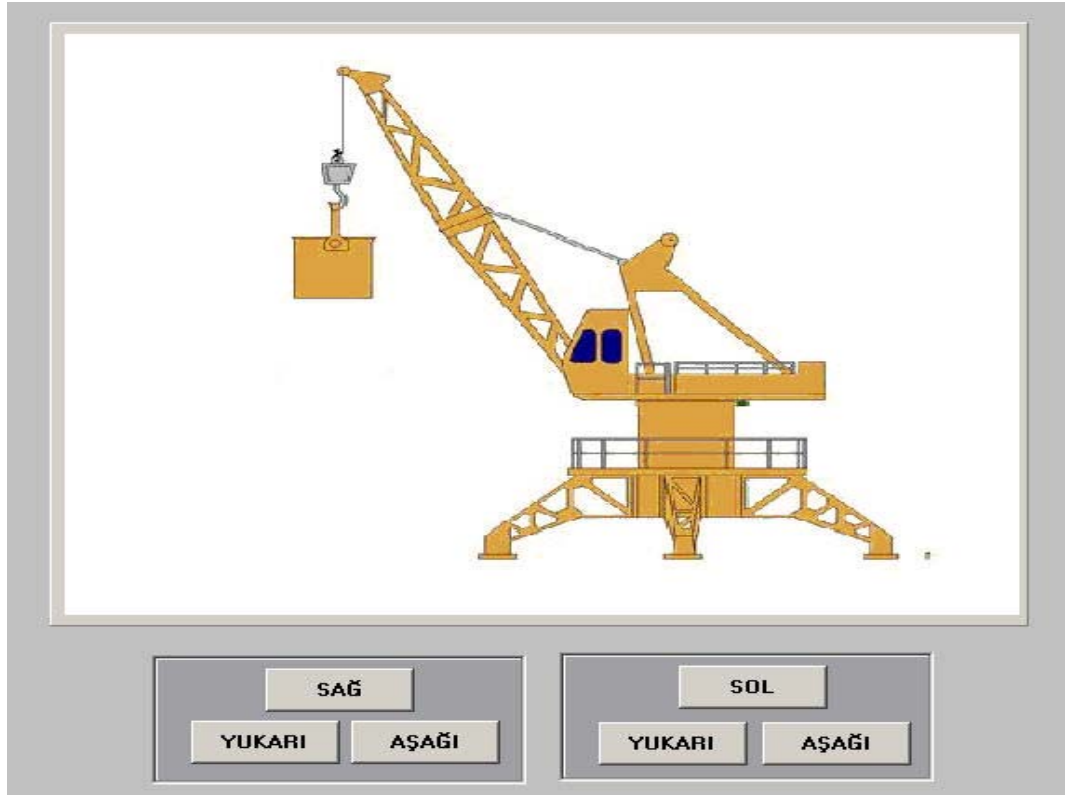


Şekil 4.15: Girişlerin denetlenmesi

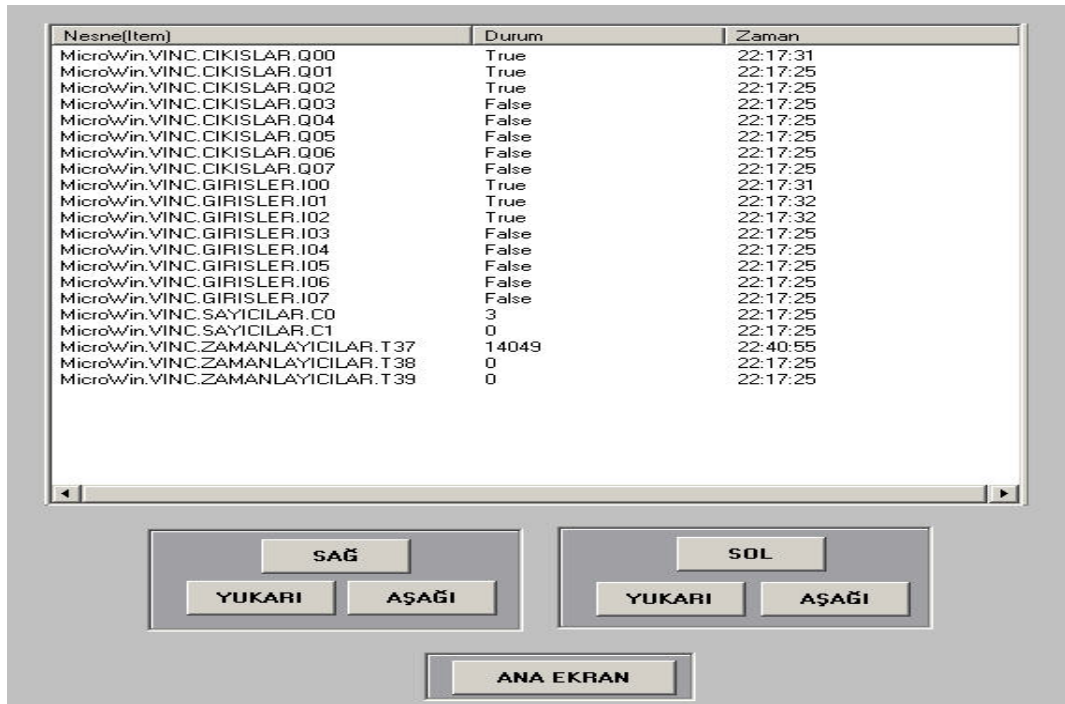
Oluşturulan denetim ekranında denetleme yapılarının oluşturulmasının ardından, uygulama prototipine (vinç sistemine) ait denetim komutları (sağ yön, sol yön, aşağı, yukarı) ve prototipin durumuna göre ekranda operatörü bilgilendirecek grafik animasyonlar gerçekleştirilmiştir. Şekil 4.16'da denetim komutları ve grafik ekranı görülmektedir.

Şekil 4.16'da görülen denetim ekranında, sisteme yapılan müdahaleye göre grafik canlandırma işleminin gerçekleştirildiği animasyonunu görülmektedir.

Denetim sistemleri için en önemli bileşenler arasında bulunan diğer bir yapı ise arşivleme birimidir. Oluşturulan bu uygulamada, sistemde meydana gelebilecek değişikliklere ait arşivleme işlevi operatörün kullanımına sunulmuştur. Bu sayede operatör sitemde bulunan OPC nesnelерinin son durumu ve bu durumun oluştuğu zaman hakkında bilgi sahibi olabilmektedir. Şekil 4.17'da arşivleme birimine ait ekran görülmektedir.



Şekil 4.16: Grafik canlandırma ekranı



Şekil 4.17: Sisteme ait arşivleme ekranı

Oluşturulan OPC tabanlı denetim yapısına son olarak internet haberleşme yapısı eklenmiştir. Eklenen bu yapı ile PLC'den elde edilen verilere internet üzerinden ulaşılması sağlanmıştır. Yereldeki operatör sistemin çalışmasıyla sistemde bulunan değişkenlere ait veriler, sanal ortama aktarılarak, sistemin uzaktan denetlenmesine olanak sağlanmıştır. Şekil 4.18'de internet üzerinden haberleşme yapısı için oluşturulan birim görülmektedir.

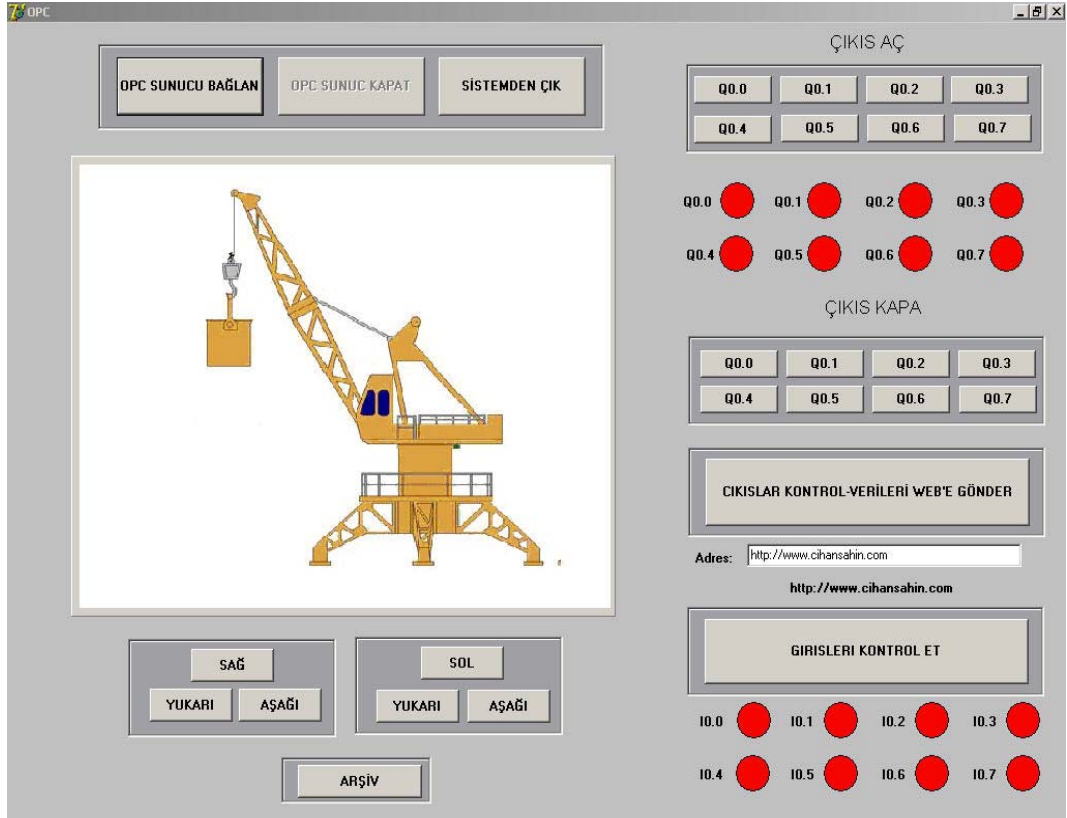


Şekil 4.18: Sisteme ait verilerin web ortamına gönderilmesi

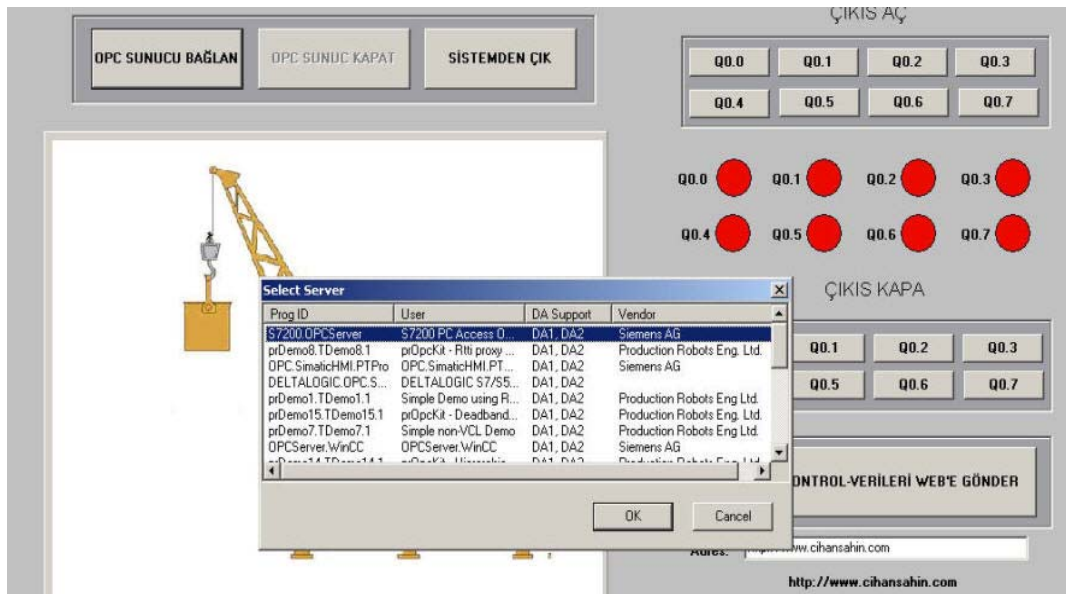
Oluşturulmak istenen yerel denetim yapısına son olarak, yerel noktada gerçekleştirilen tüm işlemlerin operatör tarafından aynı anda kullanılabilmesi amacıyla tek bir denetim ekranına taşınmıştır.. Şekil 4.19'da vinç sistemine ait yerel denetim ekranı görülmektedir.

Operatörün kullanımına sunulan bu ekran ile sisteme gerekli müdahalelerin yapılabilmesi, giriş ve çıkış durumlarının denetlenmesi, sisteme ait arşiv bilgilerine ulaşılabilmesi sağlanmıştır.

Operatör, sisteme çalışma iznini, OPC sunucuya bağlan butonunu aktif hale getirerek gerçekleştirmektedir. Delphi paket programında kullanılan bileşenin ProgID özelliği sayesinde kullanıcıya hangi OPC sunucuya veya ilgili programa erişmek isteğini sormaktadır. Şekil 4.20'de "OPC SUNUCU BAGLAN"komutuyla operatöre hangi sunucuya erişmek isteğini sormaktadır.



Şekil 4.19: Vinç prototipi yerel denetim ekranı



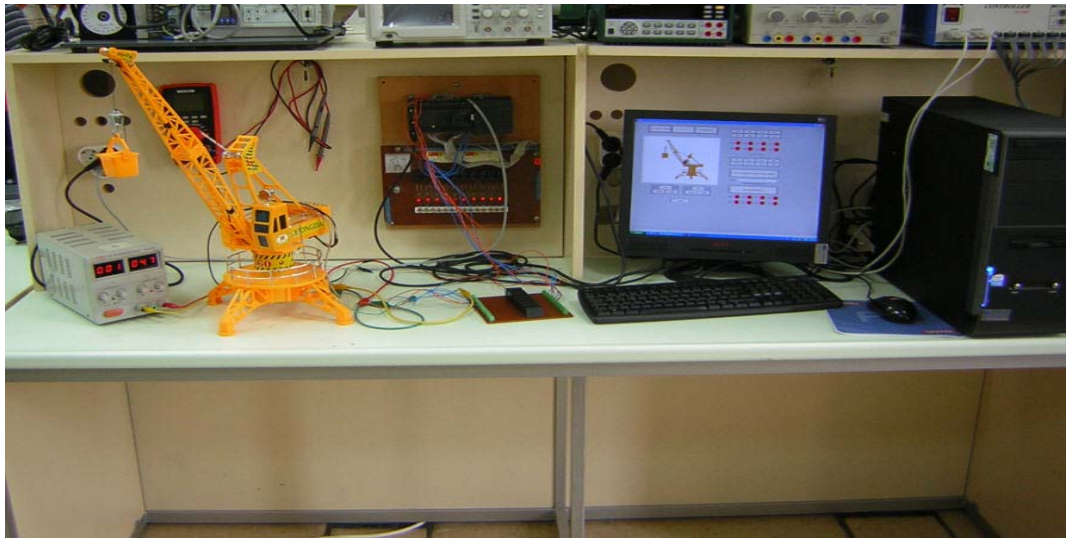
Şekil 4.20: Operatör tarafından OPC sunucuya bağlanma durumu

Örnek uygulamada kullanılan ‘‘S7200 PC Access OPC Server’’ yazılımı seçilerek sistem aktif hale getirilmiştir. Denetim yapısı, OPC sunucu üzerinden denetlenen sistem hakkında gerekli verilere sahip olup gerekli müdahale yapma yetkisine sahiptir. Sistemde kullanılan OPC sunucunun oluşturulan yapı ile haberleşmeye geçmesiyle OPC sunucuya ait haberleşme ikonları aktif olmaktadır. Ekran alt sağ köşede ikonlardan haberleşmenin sağlandığı hakkında görsel bilgiyi operatöre sunmaktadır. Şekil 4.21’da operatör tarafından OPC sunucuya bağlandıktan sonra OPC sunucu bağlantı ikonları görülmektedir.



Şekil 4.21: OPC sunucu üzerinden sistemin çalışma anı

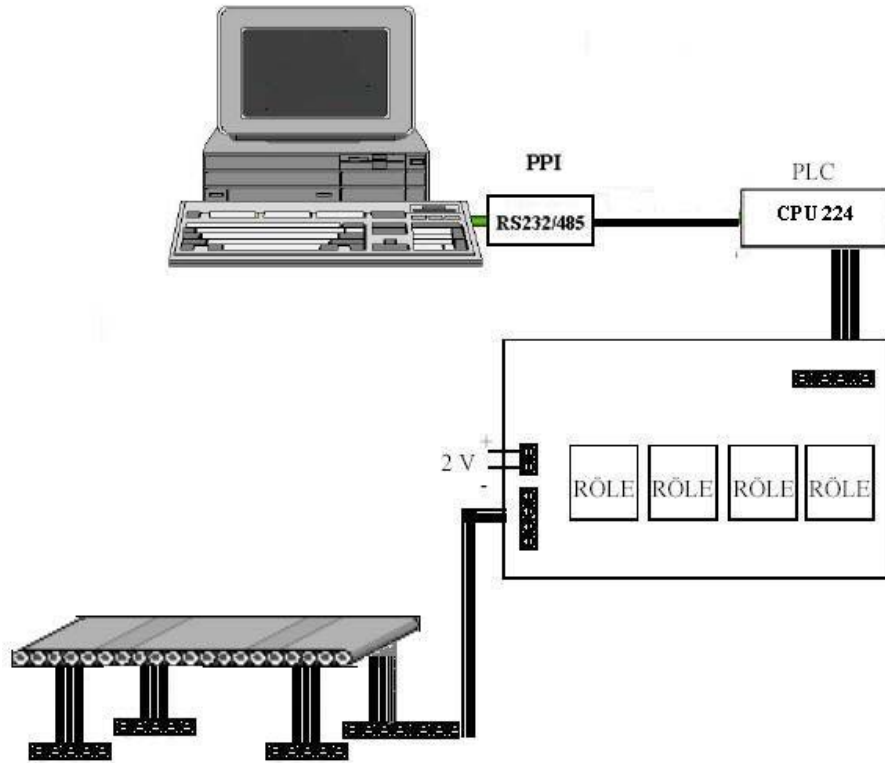
Laboratuar ortamında oluşturulan vinç uygulamasına ait prototip Şekil 4.22 de görülmektedir.



Şekil 4.22: Laboratuar ortamında oluşturulan model vinç uygulama prototipi

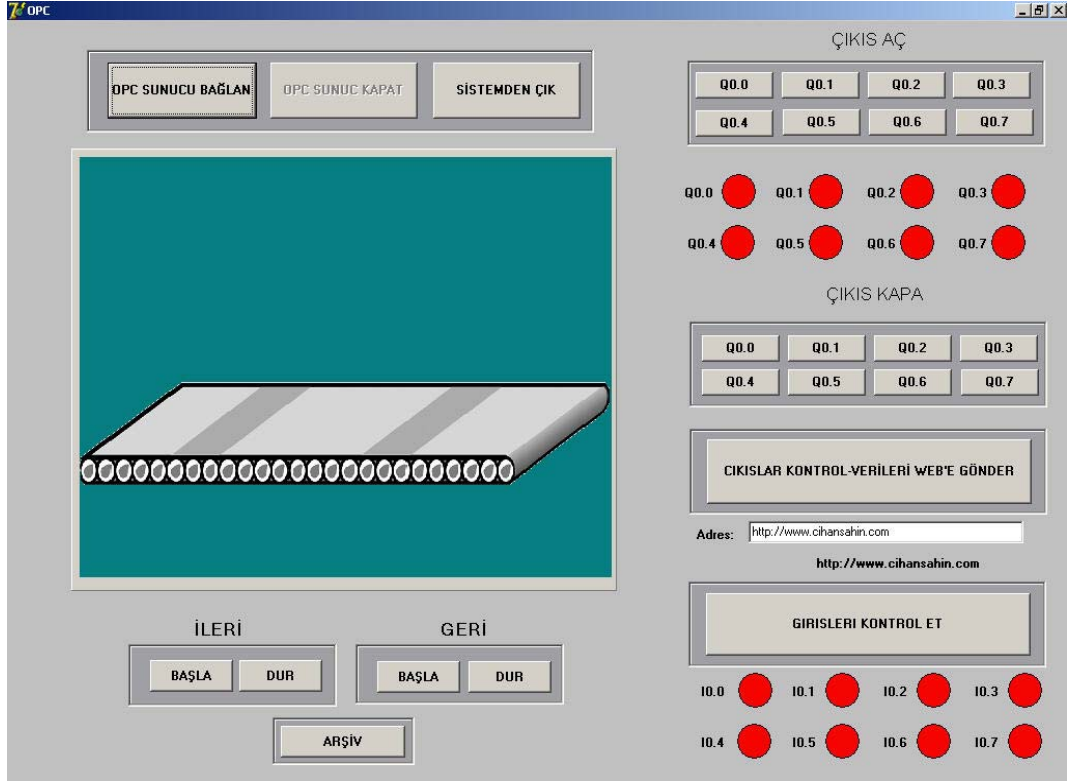
4.2.2. Örnek taşıyıcı bant uygulaması

Laboratuvar ortamında oluşturulan taşıyıcı bant uygulaması, örnek vinç uygulaması ile benzer mantıkta yapılmıştır. Örnek prototip model bir taşıyıcı bant, Siemens S7-200 CPU 224 PLC, PPI haberleşme kablosu ve bir bilgisayardan oluşmaktadır. Şekil 4.23’de model taşıyıcı bant uygulaması görülmektedir.



Şekil 4.23: Model taşıyıcı bant uygulaması

Oluşturulan sistemde taşıyıcı bandın ileri ve geri yönde hareket etmesi ve durdurulması sağlanmıştır. Şekil 4.24’de yerel denetim ekranı ve Şekil 4.25’de taşıyıcı bant uygulamasının laboratuvar ortamında oluşturulan prototipi görülmektedir.



Şekil 4.24: Taşıyıcı bant yerel denetim ekranı

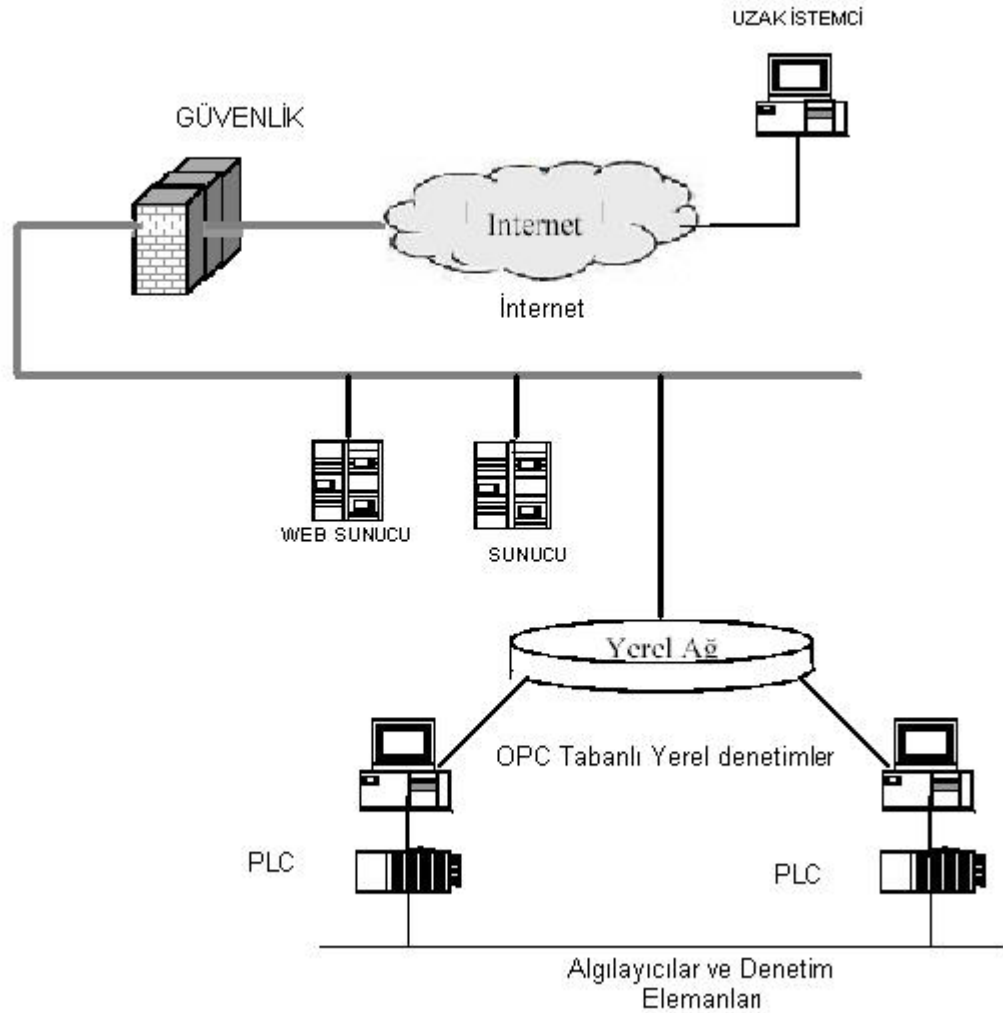


Şekil 4.25: Laboratuvar ortamında oluşturulan model taşıyıcı uygulama prototipi

Oluřturulan yerel denetim ekranından taşıyıcı bant sisteminin operatör denetimi gerçekleştirilmiştir. Vinç prototip sistemiyle benzer şekilde verilerin internete sunulmasıyla sistemin uzaktan denetlenmesi sağlanmıştır.

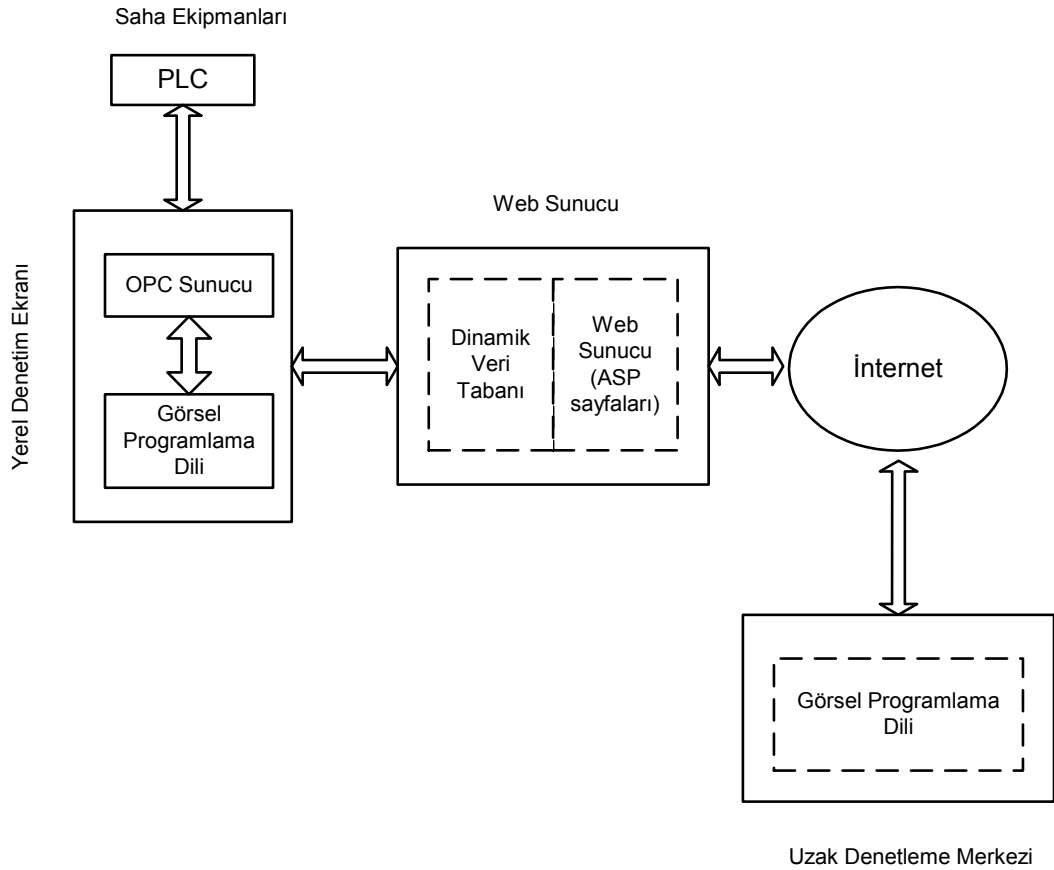
BÖLÜM 5. DAĞINIK OPC SİSTEMLERİNİN WEB TABANLI MERKEZDEN GÖRÜNTÜLENMESİ VE DENETLENMESİ

Bu bölümde, bölüm 4’de açıklanan ve klasik uygulaması yapılan farklı noktadaki OPC sistemlerinin internet üzerinden görüntülenmesi ve denetlenmesini sağlayan bir yapı geliştirilmiştir. Şekil 5.1’de oluşturulan Dağınık OPC (DOPC) sisteminin yapısı görülmektedir.



Şekil 5.1: Dağınık OPC sistemlerinin internet üzerinden denetimi

Gerçekleştirilen yapı ile yerel noktalardan elde edilen veriler uzak noktalardaki istemcilerin bilgisine sunulmaktadır. İstemci tarafından yapılan müdahaleler de ilgili yerel denetim noktasına komut olarak sunulmaktadır. Şekil 5.2’de sahadaki OPC sistemleriyle uzak denetim merkezi arasındaki veri alışverişi görülmektedir



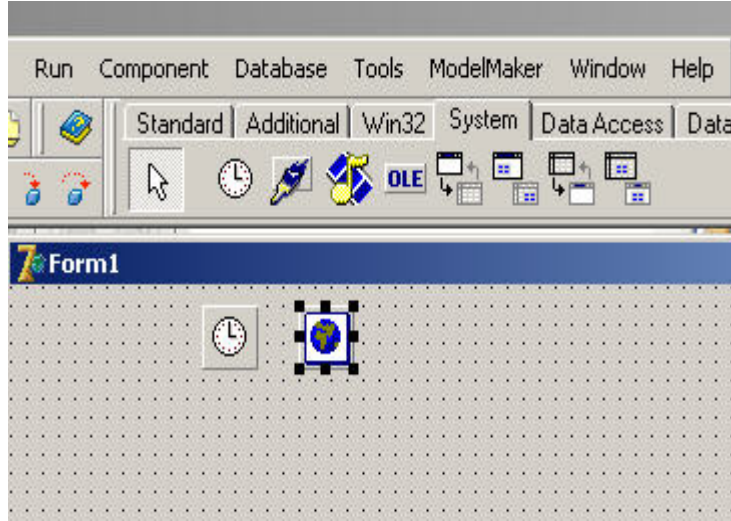
Şekil 5.2: DOPC Haberleşme Yapısı

Dağınık OPC uygulamasını gerçekleştirmek için, üç ayrı kısımda çalıştırılmak üzere üç ayrı program ve dinamik bir web sayfası hazırlanmıştır.

5.1. Vinç Sistemi Yerel Denetim Yapısı İnternet Arayüzü

Bu yapı bölüm 4’de klasik OPC sistemi kullanılarak oluşturulmuş örnek vinç uygulamasının uzaktan denetimi için gerçekleştirilmiştir. Örnek uygulamaya, web arayüzü kodları yazılarak sistemin gerekli verileri internet ortamına aktarması sağlanmıştır. Ayrıca uzak denetim merkezinden gelen komutların oluşturulan yerel

denetim yapısında aktif olmasını gerçekleştiren bölümdür. Oluşturulan yapı SCADA sistemlerinin alarm tabanlı felsefesine dayanarak gerçekleştirilmiştir. Yerel denetim yapısında operatörün oluşabilecek değişiklikleri isteğe bağlı olarak sanal ortama aktarması sağlanmıştır. Ayrıca operatörden bağımsız olarak, sistemin belli zaman aralıklarıyla denetlenip internet ortamına veri akışının sağlanması gerçekleştirilmiştir. Borland firmasına ait Delphi 7.0'da hazırlanmış yerel denetim yapısına Şekil 5.3'de görülen Timer ve IdHTTP nesneleri entegre edilmiştir. Kullanılan bu nesnelere sayesinde sistemin belli zaman aralıklarla web sunucudaki veritabanına veri göndermesi sağlanmaktadır. Benzer şekilde uzaktan yapılan müdahalelerinde sisteme uygulanmasını gerçekleştirilmiştir.



Şekil 5.3: Timer ve IdHTTP nesneleri

5.2. Taşıyıcı Bant Sistemi Yerel Denetim Yapısı İnternet Arayüzü

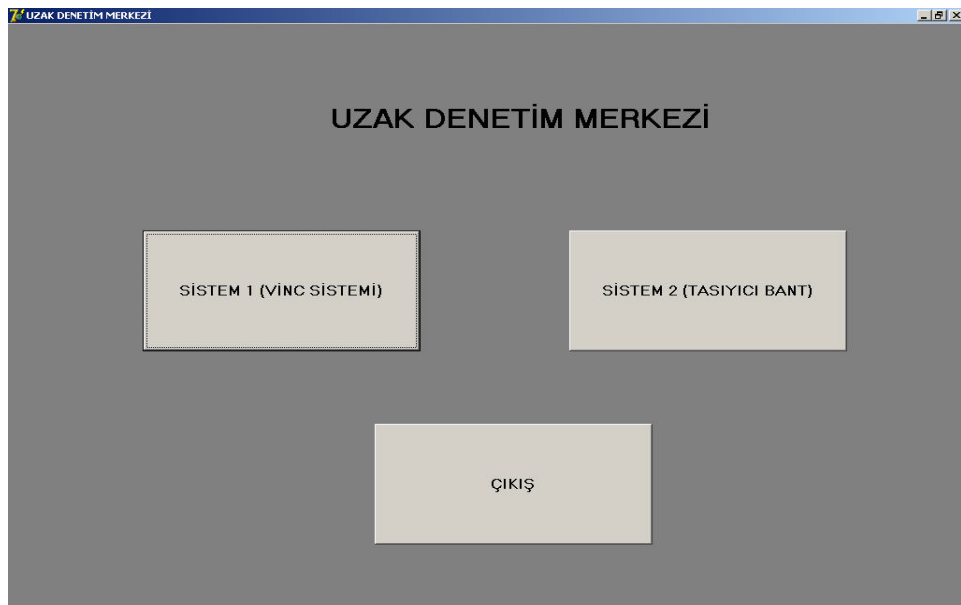
Bölüm 4'de klasik OPC sistemi kullanılarak oluşturulmuş taşıyıcı bant sistemine, internet arayüzü ilave edilerek hazırlanmış bir yapıdır. Vinç sisteminde kullanılan Timer ve IdHTTP nesnelere kullanılarak sistemin belli zaman aralıklarla internete veri göndermesi gerçekleştirilmiştir. Ayrıca, uzak denetim merkezinden gönderilecek komutların yerel denetim noktasında işlenmesini gerçekleştirmektedir.

5.3. Web Sunucu Programı

Yerel yapıdaki sistemle uzak denetim merkezindeki yapının veri alışverişini gerçekleştirmek amacıyla aktif bir web sayfası hazırlanmıştır. Aktif web sayfası herhangi bir server script (ASP, PHP v.b) kullanılarak hazırlanabilir. Yerel OPC sistemini denetleme amacıyla ASP kullanılarak dinamik bir web sayfası hazırlanmıştır. Hazırlanan web sayfası OPC tabanlı yerel denetim yapısıyla uzak denetim yapısı arasında bir köprü olma özelliğindedir. ASP ile hazırlanan web sayfası aldığı verileri kendine ait bir veri tabanına üzerinden kaydeder. Web sayfasındaki verileri kullanıcı harici kimsenin görüntülememesi için kullanıcı ismi ve parola güvenliği konmuştur. Bu sayede verilerin başkası tarafından görüntülenmesi engellenmiştir. Veri görüntüleme ise sayfaya gönderilen sorgu işlemi sonucunda gerçekleştirilmektedir.

5.4. Uzak Denetim Merkezi

Uzak denetim merkezi, kullanıcının internet üzerinden yerel noktadaki yapıları görüntülemesi ve müdahalelerde bulunması için Borland firmasına ait Delphi 7.0 kullanılarak yazılmış bir programdır. Şekil 5.4'de uzak denetim merkezinin ana sayfası görülmektedir.



Şekil 5.4: Uzak denetim merkezi ana ekranı

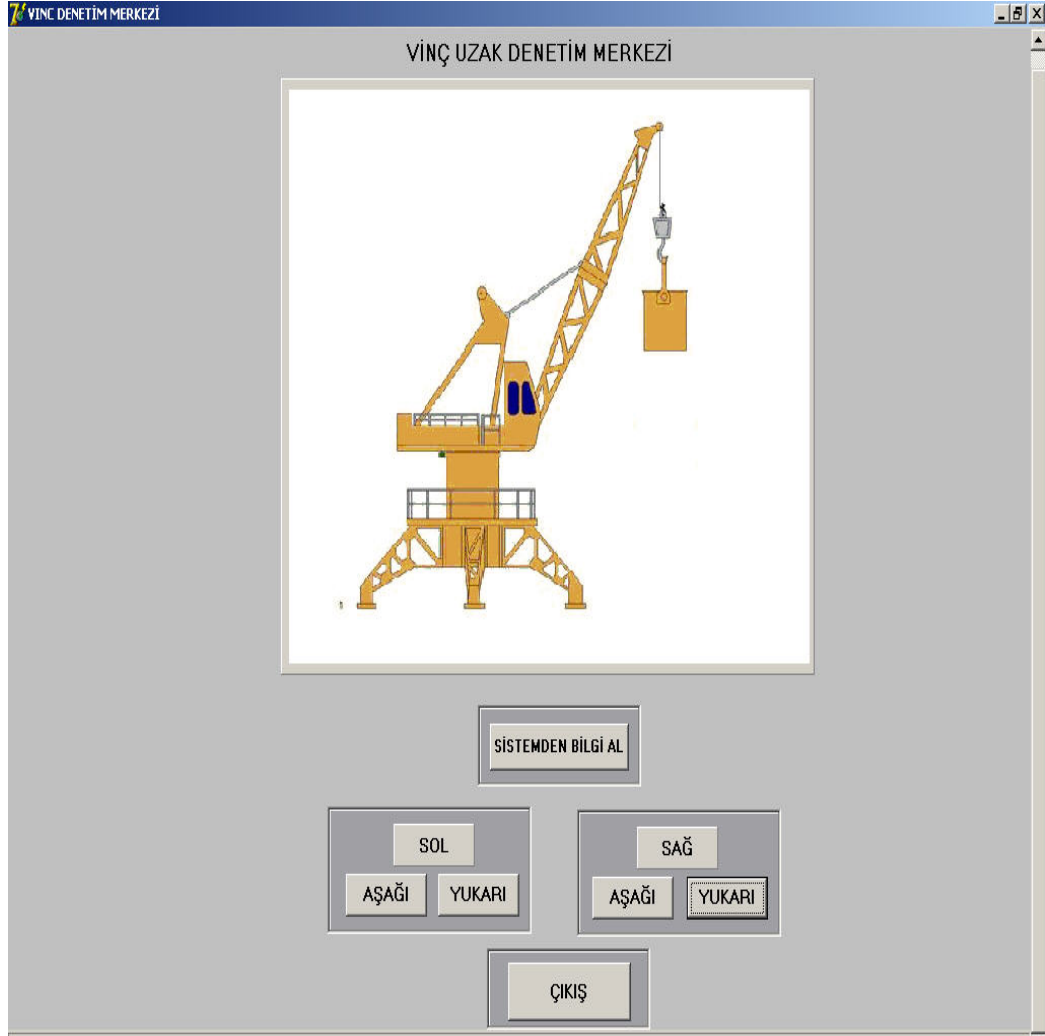
Uzak denetim merkezi, yerel noktalardan vinç sistemine ait 4 adet, taşıyıcı bant sistemine ait 3 veriyi, ASP sayfasından alma ve aynı yapılara gerekli verileri gönderme işlemini gerçekleştirmektedir. Uzak denetleme merkezinin ASP sayfasından aldığı ve sorgulara verdiği cevaplara ait parametrik veriler Tablo 5.1’de görülmektedir. Uzak denetim merkezi aldığı verilerde 1/0 durumunu inceleyerek sistem hakkında gerekli durumu grafiksel olarak kullanıcının hizmetine sunmaktadır.

Tablo 5.1: Uzak denetim merkezinin ASP sayfasından aldığı ve gönderdiği veriler

Vinç denetim sistemi		
Değişkenin Adı	Parametre	Açıklama
L0	1	Vinç Sağ Yön
L1	1	Vinç Aşağı
L2	1	Vinç Yukarı
L3	1	Vinç Sol Yön
Taşıyıcı Bant denetim sistemi		
Değişkenin Adı	Parametre	Açıklama
B0	1	Bant İleri
B1	1	Bant Geri
B2	1	Bant Start

Oluşturulan uzak denetleme merkezinde, kullanıcıya birden fazla yerel denetim noktası hakkında, bilgi sahibi olması ve müdahale yetkisi verilebilmektedir. Bu yapılar oluşturulan uzak denetim merkezinin alt sayfaları şeklinde hazırlanmış ve kullanıcının hizmetine sunulmuştur. Uzak denetim merkezinin alt sayfası olarak oluşturulmuş “VINC UZAK DENETİM MERKEZİ” ekranı Şekil 5.5 de görülmektedir.

Şekil 5.5’de görülen vinç uzak denetim merkezinde “kullanıcı” sistemde olan değişikliklerden grafik animasyonları yardımıyla haberdar olmaktadır. Oluşturulan sistem belli zaman aralıklarıyla kendini yenilemektedir. Sistemde herhangi bir değişiklik olduğunda duruma uygun grafiksel animasyon sunulmaktadır. Ayrıca oluşturulan sistem, operatör tarafından isteğe bağlı olarak internet üzerinden en son veriye erişerek o anki durum hakkında grafiksel olarak bilgi sahibi olması sağlanmıştır

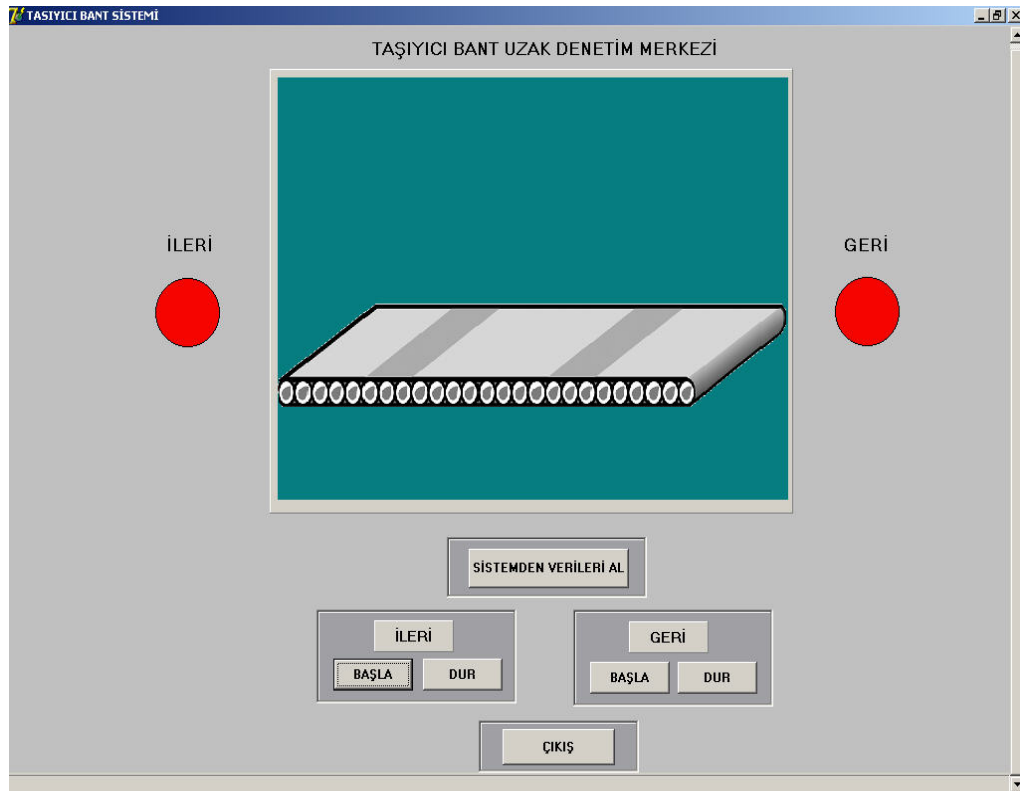


Şekil 5.5: Vinç uzak denetim merkezi

Oluşturulan “VINC UZAK DENETİM MERKEZİ” ekranında kullanıcının sistem hakkında bilgi sahibi olmasının yanında, yerel noktadaki sisteme müdahale etme izni de tanınmıştır. Uzak denetim merkezindeki kullanıcı, sisteme yaptığı müdahale ile veri tabanındaki değerleri değiştirmektedir. Yerel noktadan çekilen bu veriler oluşturulan basit program parçacıkları yardımı ile analiz edilip yerel noktadaki sistem için gerekli çıkışları vermektedir.

Yerel denetim merkezlerinden taşıyıcı bant sistemi içinde gerçekleştirilen uzak denetim merkezi ekranı, “VINC UZAK DENETİM MERKEZİ” benzer mantıkla gerçekleştirilmiştir. Uzak denetim merkezinin bir diğer alt sayfası olarak hazırlanan

taşıyıcı bant uzak denetim merkezi, yerel yapıdaki sisteminin, uzaktan görüntülenmesi ve denetlenmesi sağlamaktadır. Farklı noktalarda bulunan sistemlerinin internet üzerinden uzak denetim merkezleriyle haberleşmesi için ASP de oluşturulan ortak web sayfası kullanılmıştır. Vinç ve taşıyıcı bant sisteminin aynı web sayfasını kullanmalarına rağmen veri transferinin çakışmaması amacıyla her biri için ayrı bir dinamik veri tabanı oluşturulmuştur. Taşıyıcı bant uzak denetim merkezi denetim ekranından veriler belli zaman aralıklarıyla çekilmektedir. Gerçekleştirilen sistemde verilerin belli zaman aralıklarıyla otomatik olarak çekmesi ve sayfanın duruma göre kendini yenilemesi sağlanmıştır. Ayrıca kullanıcın yerelde bulunan sistemden gerekli verileri isteğe bağlı olarak almasında sağlanmıştır. Kullanıcıya sistem hakkında bilgi sahibi olmasının yanında yerel yapıya uzaktan müdahale etme izni de tanınmıştır. Şekil 5.6'da Taşıyıcı bant Uzak denetim merkezi görülmektedir.



Şekil 5.6: Taşıyıcı bant uzak denetim merkezi

Oluşturulan uzak denetim merkezi ve yerel denetim merkezlerinde meydana gelebilecek her bir değişiklik sistemlere ait dinamik veri tabanlarını son duruma göre değiştirmektedir. Uzak denetim ve yerel denetim yapılarının dinamik veri tabanından

belli zaman aralıklarıyla veri çekme işlemini gerçekleştirmesiyle son durum hakkında ilgili grafik animasyonunu aktif olmaktadır. Bu yapı sayesinde yerel ve uzaktan sisteme yapılan her müdahale tüm birimlerden görülmektedir.

5.5. Dağılık OPC Yapılarının İnternet Üzerinden Denetlenmesi ve Görüntüleme Sisteminin Çalıştırılması

Dağılık OPC yapılarının internet üzerinden denetlenmesi için yerel yapıda bulunan denetim programlarının ve uzak denetim merkezi için hazırlanan programların çalıştırılması gerekmektedir. Yerel yapıda hazırlanan denetim yapıları OPC sunucu üzerinden sistemden aldığı verileri internet arayüzü kullanarak web ortamında oluşturulan dinamik veri tabanına aktarmaktadır. Uzak denetim programı çalıştırıldıktan sonra karşımıza Şekil 5.4’de bulunan ekran çıkmaktadır. Bu ekrandan hangi sistem uzaktan denetlemek isteniyorsa onun seçim işlemi gerçekleştirilir. Denetlenmek istenen sayfanın seçilmesiyle beraber sayfa kodlarına gömülen zamanlayıcı çalışmaktadır. Ayarlanan süre sonunda ASP sayfasından çektiği verilere göre sistem hakkında kullanıcıya bilgi vermektedir. Kullanıcı bu ekranda görülen komutlardan hangisini aktif ederse ilgili butonun altına atanan fonksiyonlar sayesinde sisteme ait parametrik veriler ASP sayfasına gönderilmeye başlar. ASP sayfasına gönderilen veriler yerel denetim yapılarından da çekilerek sistemde gerekli değişiklik yapar. Bu sayede dünyanın herhangi bir yerinden yerel yapıdaki sistemin denetlenmesi ve görüntülenmesi gerçekleştirilmiş olur. Uzak denetim merkezlerindeki denetim ekranlarında “SİSTEMDEN VERİ AL” butonuyla da uzak merkezdeki operatör denetim programını çalıştırmaya başladıktan sonra ayarlanan zamanı beklemeden ASP sayfasındaki verilerin çekilmesi sağlanmıştır.

5.6. Dağılık OPC Yapılarının İnternet Üzerinden Denetlenmesinin Ekonomik Açıdan İncelenmesi

OPC tabanlı denetim yapılarının internet üzerinden denetlenmesi için donanımsal ve yazılımsal malzemelere ihtiyaç vardır. Kullanıcı firma tarafından temin edilmesi gereken malzemeler aşağıdaki gibi sıralanabilir.

- i) Web Sunucu, yerel denetim yapılarıyla uzak denetim merkezleri arasında veri trafiğini gerçekleştirmek için gereklidir. Yedi gün yirmi dört saat çalışabilecek bir sunucunun maliyeti ortalama 2500-3000\$ civarındadır.
- ii) Web sunucunun internet ortamı ile veri alışverişini gerçekleştirebilmesi için internet bağlantısı gerekmektedir. 256 Kb/sn'lik internet bağlantısının maliyeti aylık 33\$'dır.
- iii) Sistemde kullanılacak OPC sunucunun maliyeti ortalama 500\$'dır.

BÖLÜM 6. SONUÇLAR

Bu tez çalışmasında, endüstriyel süreç denetimine tarihsel bakış yapılmış, günümüz dünyasında geldiği noktalar açıklanmıştır. Küreselleşen dünyada, minimum maliyet maksimum hizmet politikasının egemen olduğu günümüz koşullarının endüstriyel süreç yönetim sistemlerini nasıl değişik yapılara yönlendirdiği anlatılmıştır.

Gelişen süreç yönetim sistemlerinde yeni bir yapı olan OPC sistemlerinin oluşumu, gelişimi, özellikleri ve günümüzdeki yeri anlatılmıştır. Süreç yönetim sistemlerinde, yeni bir arayüz olan OPC sistemlerinin kullanılmasıyla yerel denetim yapıları gerçekleştirilmiş herhangi bir SCADA yazılımı kullanılmaksızın, görsel programlama dili kullanılarak yerel SCADA yapıları elde edilmiştir.

Günümüz küreselleşen dünyasında, artık vazgeçilmezler arasında olan diğer bir yapı da internettir. İnsan hayatına oldukça kolaylıklar getiren internet, zamanla süreç yönetim sistemlerinin vazgeçilmezleri arasına da girmiştir.

Günümüz dünyasının internet yapısının dağıtık OPC yapılarına entegrasyonu gerçekleştirilerek, farklı noktalarda bulunan OPC yapılarının internet üzerinden tek merkezden denetlenmesi ve görüntülenmesi sağlanmıştır.

Yerel denetim noktası ile uzak denetim merkezi arasındaki veri alışverişi parametrik olarak gerçekleştirilmiştir. Veri aktarımında kullanılan web sayfası, sistem dışı bir bilgisayar tarafından görüntülense bile, veriler 1 ve 0 yapılarından oluştuğundan herhangi bir anlam ifade etmemektedir. Web sayfasından alınan 1 ve 0 yapılarından oluşan verilere gerekli anlamları Delphi'de hazırlanan uygulama programı vermektedir. Böylece, oluşturulan bu programlar olmaksızın başka bir yapı tarafından sistemin görüntülenmesi ve denetlenmesi ortadan kalkmaktadır.

KAYNAKLAR

- [1] Multimedia Tutorial of OPC, <http://www.matrikon.com/tutorial.asp> (**Ziyaret Tarihi: 06 Haziran 2005**)
- [2] Shimanuki, Y. , “OLE for process control (OPC) for new industrial automation systems”, *IEEE Systems, Man, and Cybernetics Conference*, SMC 1999, Vol.6, pp.1048-1050.
- [3] Sitao, W., Qingquan, Q., “Using device driver software in SCADA systems”, *IEEE Power Engineering Society Winter Meeting*, 2000, Vol. 3, pp 2046-2049.
- [4] Li Zheng, Nakagawa, H., “OPC (OLE for process control) specification and its developments”, *IEEE 41st SICE Annual Conference*, 2002, Vol. 2, pp. 917-920.
- [5] Ding, Z.Q., Aendenroomer, A., He, H. Goh, K.M., “OPC based device management and communication in a distributed control application platform”, *IEEE Industrial Informatics Conference*, INDIN 2003, pp 107-111.
- [6] Hong, X., Hua, W.J, ShiQuan, Z., “The distributed data integration and performance evaluation in power automation system”, *IEEE Power Engineering Society Winter Meeting*, 2000, pp.468-471.
- [7] Anwar, M.R., Anwar, O., Shamim, S.F., Zahid, A.A., “Human Machine Interface Using OPC (OLE for Process Control)”, *IEEE Engineering, Sciences and Technology*, Student Conference, 2004, pp. 35-40.
- [8] Hao, X., Hou, S., “OPC DX and industrial Ethernet glues fieldbus together”, *IEEE Control, Automation, Robotics and Vision Conference*, ICARCV 2004, 562-567 Vol. 1, pp. 562-567.
- [9] Persin, S., Tovornik, B., “Real-time implementation of fault diagnosis to a heat exchanger”, *Elsevier Control engineering practice* , 2005, Vol. 13, pp. 1061-1069.
- [10] KOSKINEN, P., “Decision-making process on field technology for process management”, *University of Oulu Department of Electrical Engineering*, 48-52, Oulu, Finland.
- [11] Technical History, <http://americanhistory.si.edu/collections/scienservice/120001.htm>.,(**Ziyaret Tarihi: 18 Kasım 2006**)
- [12] Karaçor, M. “Cep telefonlu tabanlı mobil SCADA otomasyonunun geliştirilmesi”, Yüksek lisans Tezi, *Kocaeli Üniversitesi Fen Bilimleri Enstitüsü*, İzmit, (2004)

- [13] Kurtalan, S., “PLC ile endüstriyel otomasyon”, Dördüncü basım, **Birsen Yayınevi**, 2-6, (2005)
- [14] Tan, K.K., Lee, T.H., Soh, C.Y., “Remotely operated experiment for mechatronics: monitoring of DCS on the Internet”, **IEEE Advanced Intelligent Mechatronics International Conference**, 2001, Vol. 2, pp 1106-1111.
- [15] HOWARD, J., “An Economical Solution to SCADA Communications”, **Pipe line industry**, 1988, Vol. 68, pp41-43.
- [16] Clarke, G., Reynders, D., “Practical SCADA for Protocols”, **Elsevier Newnes**, 2-5, (2004).
- [17] ALLEN, K., YUNG, L., 2000. “Inheritance and Polymorphism in real time monitoring and control systems”, **Journal of Intelligent Manufacturing**, 11, 285.
- [18] Ghosh, S.K., “Changing role of SCADA in manufacturing plant”, **IEEE Power Engineering Society Winter Meeting**, 2000, Vol. 3, pp.1565-1566.
- [19] BenteK Systems SCADA & Telemetry Solutions, <http://www.scadalink.com/> (Ziyaret tarihi: 25 Kasım 2006).
- [20] SMITH, L., WAYNE, B., 1993. “RTUs Slave for Supervisory Systems” **IEEE Computer Applications in Power**, 27, 32.
- [21] What is the OPC Foundation?, http://www.opcfoundation.org/Default.aspx/01_history.asp?MID=AboutOPC (Ziyaret tarihi: 20 Ocak 2005).
- [22] Hadlich, T., Szczepanski, T., 2000, “OPC-Making the Fieldbus Interface Transparent” [online], http://www.opcfoundation.org/Archive/White_Papers/OPC_Making_the_Fieldbus_Interface_Transparent.pdf (Ziyaret Tarih: 03 Ekim 2005).
- [23] Cantu, M., “Delpi 7 Uygulama Geliştirme Kılavuzu”, Birinci baskı, Duygu Arbatlı Yağcı, **Alfa/Aktüel Kitabevi**, 486-487, 2003.
- [24] Fajt, M., “Information agents in process automation”, Master’s thesis, **Helsinki University of Technology, Department of Automation and Systems Technology Automation Technology Laboratory**, Helsinki, 8-10, 2003.
- [25] Shi, Y., Lu, J., “A framework for distributed fieldbus remote supervisory control system”, **IEEE American Control Conference**, 2005, Vol. 1, pp. 66-72.
- [26] Kapsalis, V., Charatsis, K., Georgoudakis, M., Papadopoulos, G., “Architecture for Web-based services integration”, **IEEE Industrial Electronics Society Annual Conference**, IECON 2003, Vol.1, pp. 871.

- [27] Blanco, Paulo Mrcello P. A., Poli, Marco A., Barretto, Marcos R. Pereira, “OPC and CORBA in Manufacturing Execution System: A Review”, *IEEE Emerging Technologies and Factory Automation Conference*, ETFA 2003, Vol. 2, pp.50-57.
- [28] Venzke, M., Weyer, C., Turau, V., “Application specific vs. standard Web service interfaces for the vertical integration of fieldbus systems”, *IEEE Third International Workshop on Intelligent Solutions in Embedded Systems*, 2005, pp. 162.
- [29] Ling, Z., Yu, J., “The design of SCADA based on industrial Ethernet”, *IEEE 4th World Congress on Intelligent Control and Automation*, 2002. Vol. 4, pp. 2786-2789.
- [30] Santori, M., “OPC: OLE for Process Control”, *Real-Time Magazine*, Vol.97, pp 78-81.
- [31] Shim, M., Park, S., Yoo, D., Kim, J., Yi, M., “A Study on the flexible and efficient instrument control software generation”, *Proceedings of the 7th Korea International Symposium*, KORUS 2003.
- [32] Roengruen, P., Suesut, T., Tipsuwanporn, V., Kongratana, V., Kulphanich, S., “Design of PLC networks using remote I/O module based on controllerarea network” *IEEE Canadian Conference on Electrical and Computer Engineering*, 2001, Vol. 2, pp. 1023-1027.
- [33] García, R. G., Gele, E., Strohmeier, A., “A Software Architecture for Industrial Automation”, *Seventh IEEE International Enterprise Distributed Object Computing Conference*, 2003, pp.315.
- [34] Romero, C. E., Shan, J., “Development of an artificial neural network-based software for prediction of power plant canal water discharge temperature”, *Elsevier Expert Systems with Applications*, 2005, Vol. 29, pp. 831-838.
- [35] Santos, R. A., Normey-Rico, J. E, Gómez, A. M., Arconada, L. F. A., Moraga, C. F., “OPC based distributed real time simulation of complex continuous processes”, *Elsevier Simulation Modeling Practice and Theory*, 2005, Vol. 13, pp. 525-549.
- [36] Georgoudakis, M., Kapsalis, V., Koubias, S, Papadopoulos, G., “Advancements, trends and real-time considerations in industrial Ethernet protocols”, *IEEE Proceedings of the 4th World Congress on Digital Object Identifier*, INDIN 2003, pp.112-117.

EK-1

Yerel Denetim Yapıları Delphi 7 Kodları.

```
unit Unit1;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, prOpcClient, IdBaseComponent, IdComponent, IdTCPConnection,  
IdTCPClient, IdHTTP, StdCtrls, ComCtrls, ExtCtrls;
```

```
type
```

```
TForm1 = class(TForm)  
  OpcSimpleClient1: TOpcSimpleClient;  
  IdHTTP1: TIdHTTP;  
  BAGLAN: TButton;  
  ARSIV: TListView;  
  CIKIS00: TButton;  
  CIKIS00AKTIF: TButton;  
  GIRISKONTROL: TButton;  
  Shape1: TShape;  
  Shape2: TShape;  
  Shape3: TShape;  
  Shape4: TShape;  
  Shape5: TShape;  
  Shape6: TShape;  
  Shape7: TShape;  
  WEBGONDER: TButton;  
  Edit1: TEdit;  
  Edit2: TEdit;  
  Edit3: TEdit;  
  Edit4: TEdit;  
  Edit5: TEdit;  
  Edit6: TEdit;  
  Edit7: TEdit;  
  Edit8: TEdit;  
  StatusBar1: TStatusBar;  
  Edit9: TEdit;  
  procedure BAGLANClick(Sender: TObject);  
  procedure CIKIS00Click(Sender: TObject);  
  procedure CIKIS00AKTIFClick(Sender: TObject);  
  procedure GIRISKONTROLClick(Sender: TObject);  
  procedure WEBGONDERClick(Sender: TObject);
```

```

private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;

implementation

  {$R *.dfm}

// OPC SUNUCU BAGLAN

procedure TForm1.BAGLANClick(Sender: TObject);
var
  Grup: TOpcGroup;
  c: Integer;
begin
  Grup:= Client.Groups[0];
  Grup.Items.Clear;
  Client.GetAllItems(Grup.Items);
  ItemList.Items.BeginUpdate;
  try
    for i:= 0 to Group.Items.Count - 1 do
      with ItemList.Items.Add do
        begin
          Caption:= Group.Items[c];
          for j:= 0 to 2 do
            Subitems.Add("")
          end
        end
      finally
        ItemList.Items.EndUpdate
      end;
    end;
end;

// Q0.0 ÇIKIŞININ PASİF EDİLMESİ

procedure TForm1.CIKIS00Click(Sender: TObject);
var
  d: Integer;
  I: Integer;
  c: integer;
  Grup: TOpcGroup;
begin
  I:=0;
  Grup:= Client.Groups[0];
  d:= Grup.Item[0]

```

```
c:=Group.ItemValue[I];
shape2.Brush.Color:=clred;
end;
```

```
//Q0.0 ÇIKIŞININ AKTİF EDİLMESİ
```

```
procedure TForm1.CIKIS00AKTIFClick(Sender: TObject);
d: Integer;
I: Integer;
c: integer;
Grup: TOpcGroup;
begin
I:=1;
Grup:= Client.Groups[1];
d:= Grup.Item[1]
c:=Group.ItemValue[I];
shape2.Brush.Color:=clgreen;
```

```
//GIRISLERIN SORGULANMASI
```

```
procedure TForm1.GIRISKONTROLClick(Sender: TObject);
Var
i,c: Integer;
Group: TOpcGroup;
//
begin
//0 nolu giriş
i:=0;
c:= Group.ItemValue[i];
if c='1' then
begin
shape1.Brush.Color:=clgreen;
end
else
begin
shape1.Brush.Color:=clred;
end;

//1nolu giriş
i:=1;
c:= Group.ItemValue[i];
if c='1' then
begin
shape2.Brush.Color:=clgreen;
end
else
begin
shape2.Brush.Color:=clred;
end;
```

```
//2 nolu giriş
i:=2;
c:= Group.ItemValue[i];
if c='1' then
begin
shape3.Brush.Color:=clgreen;
end
else
begin
shape3.Brush.Color:=clred;
end;
```

```
//3 nolu giriş
i:=3;
c:= Group.ItemValue[i];
if c='1' then
begin
shape4.Brush.Color:=clgreen;
end
else
begin
shape4.Brush.Color:=clred;
end;
```

```
//4 nolu giriş
i:=4;
c:= Group.ItemValue[i];
if c='1' then
begin
shape5.Brush.Color:=clgreen;
end
else
begin
shape5.Brush.Color:=clred;
end;
```

```
//5 nolu giriş
i:=5;
c:= Group.ItemValue[i];
if c='1' then
begin
shape6.Brush.Color:=clgreen;
end
else
begin
shape6.Brush.Color:=clred;
end;
```

```

//6 nolu giriş
i:=6;
c:= Group.ItemValue[i];
if c='1' then
begin
shape7.Brush.Color:=clgreen;
end
else
begin
shape7.Brush.Color:=clred;
end;

//7 nolu giriş
i:=7;
c:= Group.ItemValue[i];
if c='1' then
begin
shape8.Brush.Color:=clgreen;
end
else
begin
shape8.Brush.Color:=clred;
end;
end;

//çıkış durumlarının web ortamına aktarılması
procedure TForm1.WEBGONDERClick(Sender: TObject);
var
  c,c1,c2,c3,c4,c5,c6,c7: Integer;
  link:=string;
str:=string;
begin
c:=0;
L0:= Grup.ItemValue[c];
Edit1.Text:=IntToStr(L0);
c:=1;
L1:= Grup.ItemValue[c];
Edit2.Text:=IntToStr(L1);
c:=2;
L2:= Grup.ItemValue[c];
Edit3.Text:=IntToStr(L2);
c:=3;
L3:= Grup.ItemValue[c];
Edit4.Text:=IntToStr(L4);
c:=4;
L4:= Grup.ItemValue[c];
Edit5.Text:=IntToStr(L4);
c:=5;
L5:= Grup.ItemValue[c];

```

```

Edit6.Text:=IntToStr(L5);
c:=6;
L6:= Grup.ItemValue[c];
Edit7.Text:=IntToStr(L6);
c:=7;
L6:= Grup.ItemValue[c];
Edit8.Text:=IntToStr(L7);
end
  try
    try
      StatusBar1.SimpleText := ";
      link:=
Edit9.Text+'?kullanici=isim&sifre'+Edit1.text'+Edit2.text'+Edit3.text'+Edit4.text'+
Edit5.text'+Edit6.text'+Edit7.text'+Edit8.text';
      str := IdHttp1.Get(link);
      if str = 'OK:' then
        StatusBar1.SimpleText := 'Veriler internete gönderildi...';
      finally
        Durum := False;
      end;
    except
      StatusBar1.SimpleText := 'İnternete bağlanırken hata oluştu.';
    end;
  end;
end.

```

EK-2

Uzak Denetim Yapıları Delphi 7 Kodları.

```
//Ana ekran kodları
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;

type
  TForm1 = class(TForm)
    Button1: TButton;
    Button2: TButton;
    StaticText101: TStaticText;
    Button3: TButton;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1; implementation

uses Unit2, Unit3;

{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
begin
  Form2.Visible:=true;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
  Form3.Visible:=true; end;
procedure TForm1.Button3Click(Sender: TObject);
begin
```

```

CLOSE;
end;
end.

//Vinç uzak denetim merkezi kodları
unit Unit2;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, IdBaseComponent, IdComponent,
  IdTCPConnection, IdTCPClient, IdHTTP, ComCtrls;

type
  TForm2 = class(TForm)
    Button1: TButton;
    Button2: TButton;
    Panel1: TPanel;
    Panel4: TPanel;
    Image1: TImage;
    Panel9: TPanel;
    Button3: TButton;
    Panel2: TPanel;
    Panel3: TPanel;
    Button4: TButton;
    Button5: TButton;
    Panel5: TPanel;
    Button6: TButton;
    Panel6: TPanel;
    IdHTTP2: TIdHTTP;
    Edit5: TEdit;
    StatusBar1: TStatusBar;
    StaticText101: TStaticText;
    IdHTTP1: TIdHTTP;
    Edit6: TEdit;
    Timer1: TTimer;
    procedure Button1Click(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button5Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button6Click(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

```



```

end;
var
  Form2: TForm2;
  Durum: Boolean;
  L0,L1,L2,L3,L4,L5,L6,L7,L8:integer;
implementation

uses Unit1;

{$R *.dfm}

procedure TForm2.Button1Click(Sender: TObject);

begin
Image1.Picture.LoadFromFile(ExtractFilePath(Application.EXENAME)+'SAGYUK
ARI.BMP');
  L0:=1;
  L1:=0 ;
  L2:=1;
  L3:=0;
  L4:=0;
  L5:=0;
  L6:=0;
  L7:=0;
end;

procedure TForm2.Button3Click(Sender: TObject);
begin

Image1.Picture.LoadFromFile(ExtractFilePath(Application.EXENAME)+'SAGASAG
I.BMP');
  L1:=1;
  L2:=1;
end;

procedure TForm2.Button5Click(Sender: TObject);
begin
Image1.Picture.LoadFromFile(ExtractFilePath(Application.EXENAME)+'SOLASAG
I.BMP');
  L1:=1 ;
  L3:=1;

end;
procedure TForm2.Button4Click(Sender: TObject);
begin
Image1.Picture.LoadFromFile(ExtractFilePath(Application.EXENAME)+'SOLYUKA
RI.BMP');
  L0:=1;
  L3:=1;

```

```

end;

procedure TForm2.Button2Click(Sender: TObject);
begin
CLOSE;
end;

procedure TForm2.Button6Click(Sender: TObject);
var
  c,c1,c2,c3,c4,c5,c6,c7: Integer;
  link:=string;
str:=string;
begin
Edit1.Text:=IntToStr(L0);
Edit2.Text:=IntToStr(L1);
Edit3.Text:=IntToStr(L2);
Edit4.Text:=IntToStr(L4);
Edit5.Text:=IntToStr(L4);
Edit6.Text:=IntToStr(L5);
Edit7.Text:=IntToStr(L6);
Edit8.Text:=IntToStr(L7);
end
  try
    try
      StatusBar1.SimpleText := "";

link:=Edit9.Text+'?kullanici=isim&sifre'+Edit1.text'+Edit2.text'+Edit3.text'+Edit4.
text'+Edit5.text'+Edit6.text'+Edit7.text'+Edit8.text';

      str := IdHttp1.Get(link);
      if str = 'OK:' then
        StatusBar1.SimpleText := 'Veriler internete gönderildi...';
      finally
        Durum := False;
      end;
    except
      StatusBar1.SimpleText := 'İnternete bağlanırken hata oluştu.';
    end;
  end;
end;

procedure TForm2.Timer1Timer(Sender: TObject);
begin
Button6.Click;
end;

end.

```

ÖZGEÇMİŞ

1980 yılında İskenderun'da doğdu. İlkokul, ortaokul ve liseyi İskenderun'da tamamladı. 1999 yılında girdiği Kocaeli Üniversitesi Teknik Eğitim Fakültesi Elektrik Eğitimi Bölümünden 2004 yılında mezun oldu. Aynı yıl Kocaeli Üniversitesi Fen Bilimleri Enstitüsü Elektrik Eğitimi Anabilim Dalında yüksek lisans eğitimine başladı.