

**UKUROVA NİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

Mehmet CİĞER

**BİLGİSAYAR KONTROLLÜ, İNTERNET DESTEKLİ
SERA OTOMASYONU**

TARIM MAKİNALARI ANABİLİM DALI

ADANA, 2010

**ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**BİLGİSAYAR KONTROLLÜ, İNTERNET DESTEKLİ
SERA OTOMASYONU**

Mehmet CİĞER

YÜKSEK LİSANS TEZİ

TARIM MAKİNALARI ANABİLİM DALI

Bu Tez 01/02/2010 Tarihinde Aşağıdaki Jüri Üyeleri Tarafından
Oybirliği/Oyçokluğu ile Kabul Edilmiştir.

.....
Prof. Dr. Ali BAŞÇETİNÇELİK Prof. Dr. İlhami YEĞİNGİL Doç.Dr. H.Hüseyin ÖZTÜRK
Danışman Üye Üye

Bu tez Enstitümüz Tarım Makinaları Anabilim Dalında hazırlanmıştır.
Kod No:

**Prof. Dr. İlhami YEĞİNGİL
Enstitü Müdürü**

Not: Bu tezde kullanılan özgün ve başka kaynaktan yapılan bildirişlerin, çizelge ve fotoğrafların
kaynak gösterilmeden kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere
tabidir.

ÖZ

YÜKSEK LİSANS TEZİ

BİLGİSAYAR KONTROLLÜ, İNTERNET DESTEKLİ SERA OTOMASYONU

Mehmet CİĞER

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
TARIM MAKİNALARI ANABİLİM DALI

Danışman :Prof.Dr. Ali BAŞÇETİNCİLİK

Yıl: 2010, Sayfa:134

Jüri :Prof. Dr. Ali BAŞÇETİNCİLİK

:Prof. Dr. İlhami YEĞİNGİL

:Doç. Dr. H.Hüseyin ÖZTÜRK

Bu çalışmada; seralarda kullanılmakta olan otomasyon sistemlerinde görülen eksikliklerin giderilmesi ile daha esnek, daha kullanışlı ve her türlü ihtiyaca cevap verebilen bir otomasyon sistemi kurulması amaçlanmıştır. Seranın; sıcaklık, bağıl nem ve ışık gibi iklim değerleri, otomatik ve uzaktan kontrol edilebilmektedir. Bu amaçla kullanılabilecek yazılım ve bilgisayar tarafından kontrol edilebilen bir donanım tasarlanmıştır.

Üzerinde çalışılan otomasyon sisteminde; her türlü ölçme elemanı (algılayıcı, dönüştürücü) ve her türlü kontrol elemanı ile çalışabilecek program tasarlanmıştır. Bunun için; üzerinde dört adet analog, bir adet sayısal giriş ve on adet de dijital çıkış olacak şekilde çok amaçlı elektronik bir kart tasarlanmıştır. Kartın esnekliğinin sağlanabilmesi için, içerisine programlanabilir bir mikro denetleyici olan PIC16F873 elemanı kullanılmıştır.

Kontrol programı; tasarlanan elektronik kart ile haberleşecek şekilde yazılmıştır. Böylece; sisteme bağlanan tüm kartlar bilgisayar tarafından kontrol edilecektir. Herhangi bir ölçme elemanının istenilen analog girişe ve herhangi bir kontrol elemanının istenilen çıkışa bağlanabilmesine olanak sağlamak için kontrol programı ve elektronik kart esnek bir yapıda tasarlanmıştır. Kontrol programı ile toplanan ölçme bilgileri bir veri tabanında depolanmaktadır. Ayrıca, otomasyon sisteminin web ara yüzü sayesinde, ölçme bilgilerine internetten ulaşılabilmesi de mümkün olacaktır.

Anahtar Kelimeler: Sera, otomasyon, algılayıcı, kontrol, PIC

ABSTRACT

MSc THESIS

COMPUTER CONTROLLED, INTERNET-AIDED GREENHOUSE AUTOMATION

Mehmet CİĞER

**DEPARTMENT OF AGRICULTURAL MACHINERY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
UNIVERSITY OF CUKUROVA**

Supervisor :Prof.Dr. Ali BAŞÇETİNCİLİK

Year: 2010, Pages:134

Jury :Prof. Dr. Ali BAŞÇETİNCİLİK

:Prof. Dr. İlhami YEĞİNGİL

:Assoc.Prof. Dr. H.Hüseyin ÖZTÜRK

In the present study, it is aimed to build an automation system which is more flexible, useful and meeting every kind of requirements, by eliminating insufficiencies of automation systems using in greenhouses. Control of the environmental parameters such as temperature, humidity and light, of the greenhouse can be carried out automatically or remotely. Required software and computer controlled hardware are designed for this operations.

The system being developed, is designed for working with any kind of measurement (sensors, converters) and control units. Therefore, a broad spectrum electronic card and four analog, one digital input and ten digital output are designed. To obtain the flexibility, PIC16F873, a programmable micro controller unit is used with card.

Control software is written by enabling communication with the designed electronic card. In this way, all cards connected to the system, will be controlled by software or in other words by computer. Controlling software and electronic card are designed flexible, thus it is possible that any measurement unit can be connected to any desired analog input and any controlling unit to output. Collected measurement data by the controlling software has been stored in a database. In addition, its possible to reach the measurement data through web interface of the automation system.

Key Words: Greenhouse, automation, sensor, control, PIC

TEŞEKKÜR

Seralar; bitkisel üretimin iyileştirilmiş çevresel koşullarda gerçekleştirildiği, diğer anlamda; dış ortamın bitkilere olumsuz etkilerinin olabildiğince sınırlandırıldığı kontrollü ortamlardır. Seraların etkili yönetilebilmesi ve işletilebilmesi için bilgisayar – mikroişlemci kontrollü bilgi işleme ve otomasyon sistemlerine gereksinim duyulmaktadır. Seranın değişken iklim değerlerinin (sıcaklık, bağıl nem ve ışık gibi) uzaktan veya otomatik kontrollü olarak gerçekleştirilmesini sağlayan, bu sayede sistemin esnek, kullanışlı olmasını, maliyetlerin düşürülmesini amaçlayan bu çalışmanın, sera otomasyonu için önemli bir örnek olmasını umuyorum.

Seranın otomasyon sistemi için yaptığım çalışmalar arasında, sera sistemleri, elektronik, donanımı, programlama, haberleşme teknikleri ve internet konularında araştırma ve çalışmalar yer almaktadır. Tüm bu konularda önceki deneyimlerimin büyük faydası oldu. Sonuçta her türlü ölçme elemanı (algılayıcı, dönüştürücü) ve kontrol elemanı ile çalışabilecek, yetiştirilen ürüne ve alanın büyüklüğüne gerekli ayarlar yapılarak adapte olabilecek esnek ve kullanışlı bir çalışma ortaya çıkmıştır.

Bu çalışmamda, başından sonuna kadar desteğini ve zamanını esirgemeyen, her konuda çalışmalarına yardımcı olan ve yönlendiren, danışman hocam Sayın Prof.Dr. Ali BAŞÇETİNCİLİK'e; elektronik kart tasarımı, pic programlama ve tüm testlerde başından sonuna kadar yanımda olan, Y.Müh. Ali Osman YAĞLIOĞLU'na; odasını benimle paylaşan ve hatalarımı görmemde bana yardımcı olan Y.Müh. M.Yücel ÖZCAN'a ve çalışmada emeği geçen herkese en içten teşekkürlerimi saygılarımla sunarım.

İÇİNDEKİLER

SAYFA

ÖZ	II
ABSTRACT	II
TEŞEKKÜR	III
İÇİNDEKİLER	IV
ÇİZELGELER DİZİNİ	VII
ŞEKİLLER DİZİNİ	VIII
1. GİRİŞ	1
1.1. Otomasyon	2
1.2. Sera Otomasyonu	2
1.3. Otomasyonsuz Sera Sistemleri	3
1.4. Yarı Otomatik Sera Sistemleri	3
1.5. Mikroişlemci Kontrollü (PLC) Sistemler	4
1.6. Mikrodenetleyicili Sistemler	5
1.7. Bilgisayar Kontrollü Sistemler	6
2. ÖNCEKİ ÇALIŞMALAR	7
3. METERYAL ve METOD	10
3.1. Algılayıcılar	11
3.1.1. Sıcaklık Algılayıcı	12
3.1.2. Işık Algılayıcı	12
3.1.3. Bağıl Nem Algılayıcısı	13
3.2. Kontrol Sistemleri	14
3.2.1. Röle	14
3.2.2. Kontaktör	15
3.2.3. Valf	16
3.3. Sera Sistemleri	16
3.3.1. Sulama Sistemi	16
3.3.2. Isıtma Sistemi	17
3.3.3. Havalandırma Sistemi	18
3.3.4. Gübreleme Sistemi	19

3.3.5. Gölgeleme Sistemi	19
3.3.6. Soğutma Sistemi.....	19
3.3.7. Alarm Sistemi.....	20
3.4.1. Elektronik Kartta Kullanılan Entegreler	21
3.4.1.1. PIC16F873 Entegresi	21
3.4.1.2. İki Girişli AND Kapısı Entegresi (4081)	25
3.4.1.3. Regüle Entegresi (7805).....	26
3.4.1.4. max232 Seri İletişim Entegresi	27
3.5.1. Bilgisayar	29
3.5.2. Veri Tabanı.....	29
3.5.3. Programlama Dili	30
3.5.4. Otomasyon Programı.....	30
3.5.6. Otomasyon Programının İş Akışı.....	31
4. BULGULAR ve TARTIŞMA.....	33
4.1. Elektronik Kartın Tasarımı	33
4.2. Otomasyon Programının Kullanımı	36
4.2.1. Açılış Ekranı (Ana Ekran).....	36
4.2.2. İletişim Portu Tanımlama Ekranı	37
4.2.3. İşlem Tanımlama Ekranı	39
4.2.4. Giriş-Çıkış İşlemi Ekleme/Düzeltilme Ekranı.....	40
4.2.4.1. Ölçme Elemanı Tanımlama Ekranı	42
4.2.4.2. Ölçme Elemanı Karşılık Tablosu Ekranı.....	43
4.2.5. Kart-Eleman Tanımlama Ekranı	44
4.2.6. Kontrol İşlemleri	47
4.2.6.1. Otomatik Kontrol Ekranı.....	49
4.2.6.2. Elle Kontrol Ekranı	53
4.2.7. Grafik İzleme Ekranı	55
4.2.8. Program Kapatma ve Hakkımızda Ekranı.....	58
4.3. İnternet Bilgi sistemi.....	58
5. SONUÇLAR ve ÖNERİLER.....	60
KAYNAKLAR	62

ÖZGEÇMİŞ	64
EK 1. PIC PROGRAMININ KODLARI	65
EK 2. OTOMASYON PROGRAMININ KODLARI.....	73
Ana Ekranı	73
İşlem Tanım Ekranı	85
İşlem Ekle/Düzeltil Ekranı.....	88
Ölçme Eleman Tanım Ekranı	92
Ölçme Elemanı Ekle/Düzeltil Ekranı.....	94
Ölçüm Karşılık Ekranı.....	97
Ölçüm Karşılığı Ekle/Düzeltil Ekranı.....	99
Kart-Eleman Tanımları Ekranı	101
Kart-Eleman Tanımı Ekle/Düzeltil Ekranı	103
Otomatik Kontrol Tanımları Ekranı	110
Otomatik Kontrol Kuralları Ekle/Düzeltil Ekranı.....	115
Elle Kontrol Ekranı.....	119
Grafik Kriter Seçim Ekranı.....	125
EK 3. WEB PROGRAMI KODLARI	127
global.php	127
index.php	127
okunanDegerleriDoldur.php	132
sonDurumlariDoldur.php.....	133

ÇİZELGELER DİZİNİ

SAYFA

Çizelge 3.1. PIC16F873 teknik özellikleri	22
Çizelge 3.2. PIC16F873 bacak bağlantı bilgileri	24
Çizelge 3.3. 4081 AND kapısı çalışma gerilim değerleri	25
Çizelge 3.4. 4081 AND kapısı doğruluk tablosu	25
Çizelge 3.5. 7805 regüle entegresi elektriksel karakteristik tablosu.....	27

ŞEKİLLER DİZİNİ

SAYFA

Şekil 3.1. Otomasyon sistemi düzeneği	10
Şekil 3.2. NTC ısı algılayıcı.....	12
Şekil 3.3. Foto direnç (LDR).....	13
Şekil 3.4. SHS-A3 RoHS nem algılayıcı	14
Şekil 3.5. 12–220 volt röle	15
Şekil 3.6. Üç fazlı kontaktör	15
Şekil 3.7. Rüzgarsız ve rüzgarlı havalarda pencere pozisyonları.....	18
Şekil 3.8. PIC16F873 bacak bağlantıları.....	22
Şekil 3.9. PIC16F873 blok diyagramı.....	23
Şekil 3.10. 4081 AND kapısı bacak bağlantısı ve iç yapısı	25
Şekil 3.11. 7805 regüle entegresi	26
Şekil 3.12. 7805 regüle entegresi blok diyagramı.....	26
Şekil 3.13. max232 seri iletişim entegresi	28
Şekil 3.14. max232 mantıksal diyagramı.....	28
Şekil 3.15. max232 tipik çalışma devresi	28
Şekil 4.1. Elektronik kart şeması	34
Şekil 4.2. Elektronik kart baskı devre görünüşü	35
Şekil 4.3. Elektronik kart montajsız görünüşü.....	35
Şekil 4.4. Elektronik kart bitmiş önürünü	36
Şekil 4.5. Açılış ekranı.....	37
Şekil 4.6. Seri iletişim portu tanımlama.....	38
Şekil 4.7. Seri iletişim portunun varsayılan ayarları	38
Şekil 4.8. İşlem tanımlama.....	39
Şekil 4.9. İşlem tanımlama ekranı.....	39
Şekil 4.10. İşlem ekleme/düzeltilme ekranı	40
Şekil 4.11. Giriş işlemi ekleme/düzeltilme ekranı	41
Şekil 4.12. Karakter eşleşme tablosu.....	41
Şekil 4.13. Çıkış işlemi ekleme/düzeltilme ekranı.....	41
Şekil 4.14. Ölçme elemanı tanımlama	42

Şekil 4.15. Ölçme elemanı tanımlama ekranı	42
Şekil 4.16. Ölçme elemanı ekleme/düzeltilme ekranı	43
Şekil 4.17. Ölçüm karşılık tanımlama ekranına giriş	43
Şekil 4.18. Ölçme elemanı karşılık tanımlama ekranı	44
Şekil 4.19. Ölçme elemanı karşılık ekleme/düzeltilme ekranı	44
Şekil 4.20. Kart-Element tanımlama ekranına giriş	45
Şekil 4.21. Kart-Element ekleme/düzeltilme ekranı	45
Şekil 4.22. ON/OFF tipi çıkış elemanı kart ilişkisi	46
Şekil 4.23. Kademeli tip çıkış elemanı kart ilişkisi	46
Şekil 4.24. Giriş elemanı kart ilişkisi	47
Şekil 4.25. Çıkış kontrol işlemleri	48
Şekil 4.26. Otomatik kontrol işlemine giriş	48
Şekil 4.27. Elle kontrol işlemine giriş	49
Şekil 4.28. Otomatik çıkış kontrol işlemleri	50
Şekil 4.29. Kural ekleme/düzeltilme ekranı	50
Şekil 4.30. Kural ekleme/düzeltilme ekranında ölçüm işlemi seçimi	51
Şekil 4.31. Kural ekleme/düzeltilme ekranında şart seçimi	51
Şekil 4.32. Kural ekleme/düzeltilme birden fazla kıstas belirleme	52
Şekil 4.33. Kural ekleme/düzeltilme ON/OFF tipi işlem hareket seçenekleri	52
Şekil 4.34. Kural ekleme/düzeltilme ON/OFF tipi işlem seçilmiş hareket	52
Şekil 4.35. Kural ekleme/düzeltilme kademeli işlem tipi hareket seçenekleri	53
Şekil 4.36. Kural ekleme/düzeltilme kademeli işlem tipi seçilmiş hareket	53
Şekil 4.37. Elle kontrol ekranı	54
Şekil 4.38. ON/OFF tipi çıkış işlemi seçilmesi	54
Şekil 4.39. ON/OFF tipi çıkış işlemi hareket seçenekleri	55
Şekil 4.40. Kademeli tip çıkış işlemi seçilmesi	55
Şekil 4.41. Kademeli tip çıkış işlemi hareket seçenekleri	55
Şekil 4.42. Grafik izleme ekranına giriş	56
Şekil 4.43. Grafik kriterleri belirleme ekranı	56
Şekil 4.44. Grafik kriterleri işlem seçimi	57
Şekil 4.45. Grafik kriterleri başlangıç ve bitiş tarihi belirleme	57

Şekil 4.46. Grafik çizmeye başlama.....	57
Şekil 4.47. İncelenecek grafiksel gösterim	58
Şekil 4.48. Kapatma onay ve program bilgisi ekranı	58
Şekil 4.49. İnternet bilgi sistemi	59

1.GİRİŞ

Otomasyon, bir imalat dizinini, her noktasında insanın işe karışmasına gerek kalmaksızın yapan, denetleyen ve özgün olarak negatif bir geri besleme sistemi kullanan kontrol sistemidir (Yule,1989).

Otomasyon sistemleri, uzun araştırma ve tecrübeler sonucunda ortaya çıktığı için, özellikle kurulum maliyetleri, yüksek sistemlerdir. Sera otomasyon sistemlerinin de kurulum maliyetinin çok yüksek olması nedeniyle, bu sistemler tarım sektöründe fazla kullanılmamaktadır. Ayrıca; önceleri bu sistemlerin bilgisayar destekli çalışması durumunda; bu sistemleri kullanabilecek donanımlı kişilere ihtiyaç duyulmakta idi. Ancak günümüzde; hem bilgisayarların kullanımının kolaylaşması ve genişlemesi, hem de otomasyonun kurulum maliyetinin seralarda kısa zamanda karşılanıyor olması gibi sebeplerle bu sistemler yaygınlaşmaya başlamaktadır.

Günlük hayatımızın hemen hemen her noktasında kullandığımız teknolojinin tarım için de kullanılması; zaman ve maliyet açısından kaçınılmaz bir durumdur. Teknolojinin bu kadar hızlı ilerlediği günümüzde; tarım alanında da teknolojik gelişmeler kendini artan bir hızla göstermektedir. Bilgisayarlı otomasyon sistemlerinin gelişmesi ve yayılmasıyla bu ilerlemelerin çok daha hızlı bir şekilde olacağı muhakkaktır.

Taşınabilir mikrobilgisayar kullanımı ile verilerin depolanarak; bu verilerin BASIC programlı bir bilgisayarda değerlendirmesinin yapılabileceği bildirilmektedir (Gilreath, 1985).

Veri toplama; zaman alan ve bu esnada da hata oranını arttıran bir uygulamadır. Daha düşük fiyatlı, sonuçları tablolar halinde verebilen veri toplama ve analizinde tam otomasyonu sağlayacak bir aleti elde etme çalışmaları sonuçlanmış ve taşınabilir mikrobilgisayar elde edilmiştir. Taşınabilir mikrobilgisayara eklenecek bir telefonla veya mikrobilgisayarın merkezi bilgisayara direk bağlantılı hale getirilmesiyle verilerin anında merkeze ulaşması sağlanır. Bu sayede saatler süren veri toplama işi, kısa zamanda halledilmiş olur (Gilreath, 1985). Bu bilgiler ışığında; günümüzde iletişim ağının cep telefonu ya da uydu telefonları aracılığı ile tarımsal alanların hemen hemen tamamını kapsadığı düşünülürse, böyle bir sistemin kurulum

ve kullanımının daha rahat olacağı görülmektedir. Hedef sadece veri toplama değil, aynı zamanda uzaktan ölçme ve kontrol sistemleridir.

1.1. Otomasyon

Otomasyon, bir işin insan ile makina arasında paylaşılmasıdır. Toplam işin paylaşım yüzdesi otomasyonun düzeyini belirler. İnsan gücünün yoğun olduğu otomasyon sistemleri yarı otomasyon; makinanın yoğun olduğu sistemler de tam otomasyon olarak adlandırılırlar. Sanayi devriminin başlangıcının aksine, yakın geçmişimizde üretim yapmak tek başına yeterli olmaktan uzaklaşmıştır. Tüm dünyanın açık bir pazar haline geldiği rekabetçi koşullarda üretimi; hızlı, standart, güvenli, nihayet verimli kılmak, bir zorunluluk haline gelmiştir. Endüstride bu zorunluluğun karşılığı şüphesiz ki otomasyondur.

Otomasyonun genel olarak; üstünlükleri olduğu gibi olumsuzlukları da bulunmaktadır. Otomasyonda üretim standardı ve maliyet düşüşü sağlanırken, işsizliği körüklediği de bir gerçektir. Ayrıca otomasyon ile yapılan üretimlerde her şeyin esnek ve kontrol edilebilir olması, yöneticinin işine gelmektedir. Çünkü; bilgisayar ekranında sisteminin işleyişini, eğer varsa arızanın yerini, üretilen ürün miktarını v.b. fabrikayı ilgilendiren birçok bilgilere erişim ve kontrol kolaylığı sağlar. Bu faydalar da işletmeye zaman, kalite, maliyet, hız ve kâr olarak geri dönmektedir.

Bu sistemlerin en büyük olumsuzluğu, ilk kurulumunun çok maliyetli oluşudur. Bu maliyet de uzun vadede çoğu otomasyon sistemlerinde, kendini amorti etmektedir. Bir diğer zararı ise; fabrikalara giren bu otomasyon sistemleri fabrika çalışanlarının sayısında azalmaya sebep olmaktadır.

1.2. Sera Otomasyonu

Algılayıcılardan gelen verilerin bilgisayarda işlendikten sonra, gerekli olan donanımın çalıştırılmasına veya çalışmamasına karar vermekten ibaret olan sera otomasyonu, diğer otomasyonlardan farklılık göstermektedir.

Sera otomasyonu; donanım ve yazılım olarak iki kısımdan oluşmaktadır. Donanım kısmı, bilgisayar, PLC’li sistem veya mikrodenetleyicili bir sistem veya bunların karışımından oluşan bir kontrol sistemi ile bunların kontrol ettiği çıkış elemanlarından (röle, kontaktör vb.) oluşur. Yazılım kısmı ise, var olan donanımın kontrolünü ve veri alma görevini yerine getirerek, karar verme işlemini yapmaktadır.

1.3. Otomasyonsuz Sera Sistemleri

Otomasyonsuz sera sistemleri; ürün kalitesinin ve başarı oranının sera sahibinin tecrübesine bırakıldığı sıcaklık veya nem ölçmek dışında herhangi bir elektronik aygıtın kullanılmadığı sera sistemleridir.

Bu tip seralarda elektronik sistem kullanılmadığı için, maliyetler düşük olmasına karşın; radyo, televizyon vb. bilgi kaynaklarından hava tahmin raporlarının öğrenilerek, önlem almak üzere belirli zaman aralıklarında seranın kontrol edilmesi ve belirli işlemlerin yapılması gerekmektedir.

Bu tür seralar genellikle, iklim değişimlerinden etkilenmeyen bölgesel bitkiler yetiştirmek için kullanılmaktadır.

1.4. Yarı Otomatik Sera Sistemleri

Bu tür seralarda sera otomasyonunu oluşturan elemanlar birbirinden bağımsız çalışmakta ve gerekli işlemler otomatik olarak gerçekleştirilmektedir. Örneğin; toprak neminin istenilen seviyenin altında olması durumunda, sulama sistemi devreye alınabilmekte; ancak sulamanın havadaki nem ile toprak nemi bağlantılı olarak yapılması istendiğinde, bu tür sistemler yetersiz kalmaktadır. Otomasyonla belirtilen sera sistemlerinin, belirtilen zamanlarda veya şartlarda devreye girmesi sağlanabilmektedir.

Yarı otomatik sera sistemlerinde en çok zaman röleleri ve sıcaklık ayarlı termostatlar kullanılmaktadır.

1.5. Mikroişlemci Kontrollü (PLC) Sistemler

Programlanabilir Lojik Kontrolör (PLC - Programmable Logic Controller) fabrikalardaki imalat hatları veya makinaların kontrolü gibi işlemleri yapabilen, programlanma özelliği olan giriş ve çıkış uçları önceden belirlenmiş kontrol elemanıdır ki; bir nevi düşük kapasiteli bilgisayar da denilebilir.

Genel kullanımlı bilgisayarların aksine; PLC birçok girişi ve çıkışı olacak şekilde düzenlenir. Elektriksel gürültülere, sıcaklık farklarına, mekanik darbe ve titreşimlere karşı daha dayanıklı tasarlanırlar. PLC'lere denetleyeceği sistemin işleyişine uygun programlar yüklenir. PLC programları, giriş bilgilerini milisaniyeler mertebesinde hızla tarayarak, buna uygun çıkış bilgilerini gerçek zamanlıya yakın, cevap verecek şekilde çalışırlar.

PLC, bir mikroişlemcisi olup, karmaşık sistemlerin programlanmasında kullanılabilir. PC'lerden en temel farkı, sinyal girişlerinin ve çıkışlarının (Input/Output) çalışma ve işlenme şeklinde ortaya çıkmaktadır. PLC'ler ilk olarak, karmaşık röle sistemlerinin yükünü hafifletmek için çıkarılmış basit mantıksal (lojik) işlemleri yapan cihazlardı.

Günümüzde diğer programlanabilir bilgisayarlarla arasındaki farklılıklar giderek azalmaktadır. PLC bir bakıma monitörü ve klavyesi bulunmayan bir bilgisayar gibidir. Bir diğer fark ise; işletilecek verilerin gerçek ortamdan gelmesi ve sonuçların yine gerçek ortama analog veya sayısal (dijital) olarak gönderilmesidir.

Mikroişlemcilerin maliyeti daha düşük olmasına rağmen, PLC'lerin tercih edilmelerinin sebebi; elektronik tasarım için harcanacak zamanı en aza indirmesidir. Aynı zamanda endüstriyel ortamların sahip olduğu zor koşullardan (manyetik alan, büyük sıcaklık farkları, toz vb.) etkilenmeden çalışabilen hazır çözümler olmalarıdır.

Bir fabrikanın tüm otomasyon işlerini yüklenebilecek kadar giriş/çıkış sayısına sahip PLC'ler bulunmaktadır. Günümüzde geliştirilen modüler yapıdaki PLC'lere gerektiğinde ek giriş-çıkış modülleri, RS232, RS485, modem, ethernet gibi haberleşme modülleri eklenebilmektedir. Bu gibi özelliklerle mevcut yapı geliştirilebilmektedir. Ayrıca, birçok modelde süreç kontrolüne yönelik hazır ON-

OFF (AÇ-KAPA), PID, Fuzzy (Bulanık) vb. tiplerdeki kontrolörler standart olarak bulunmaktadır.

Yukarıda anlatıldığı gibi; oldukça yetenekli olmalarına karşın, yaptıkları işe göre maliyetleri göz önüne alındığında pahalı sistemlerdir. Bu nedenle, günümüzde bu sorunu aşmak için daha ucuz, daha az enerji ile iş görebilen mikrodnetleyicili sistemler, otomasyonda PLC'lerin yerini almaktadır.

1.6. Mikrodnetleyicili Sistemler

Temel olarak bir mikrodnetleyici, içerisinde mikroişlemcisi olan, hafızası olan ve giriş-çıkış uçları olan bir entegredir. Bu manada PLC ile arasındaki tek fark, çıkışlarının sayısal elektronik mantığı ile tasarlanmış olmasıdır. Bir diğer bakışla da; küçük kapasiteli bir bilgisayardır.

Günümüzde imalat teknolojileri hızla gelişen ve ucuzlayan mikrodnetleyiciler, üzerlerindeki analog digital dönüştürücüler, seri iletişim uçları ve benzeri ilaveleriyle yetenek olarak PLC'lerin ilerisine geçmiş bulunmaktadır. Bu sebeptendir ki; otomasyonlar artık PLC ile değil mikrodnetleyicilerle yapılmaktadır.

Ancak; endüstride oldukça fazla kullanım alanı olan mikrodnetleyicilerin ve PLC'lerin hafızaları ve işlem kapasiteleri karmaşık süreçler karşısında yetersiz kalabilmekte, bu yüzden de gelişen teknoloji ile birlikte daha güçlü yapıya sahip endüstriyel PC'lere geçiş söz konusu olmaktadır.

1.7. Bilgisayar Kontrollü Sistemler

Mikrodenetleyicili ve PLC’li sistemlerin gelişmiş olan bilgisayar kontrollü sistemler içinde gelişmiş mikroişlemciler barındırmaktadır. Yazılım için bellek sorunu bulunmadığından çok daha gelişmiş yazılımlar ile sistem kontrolleri ve istatistiksel veriler elde etmek, geçmişe dönük verilere ulaşmak çok daha kolay olmaktadır. Buradaki tek sıkıntı, çalışma alanının fiziki koşulları olmaktadır. Yani bilgisayarın nemli, tozlu ve kimyasal maddelerin bulunduğu alanlarda ömürlerinin çok kısa olması sebebiyle, bilgisayarı çalışan sistemin biraz dışarısına almak gerekmektedir.

Bilgisayar ile otomasyon gerçekleştirmek için, sera sistemlerinden bilgi almayı veya bu sistemlere bilgi göndermeyi sağlamak amacıyla, bazen bilgisayar üzerine takılı veya çoğunlukla dışardan bağlantılı bir I/O “giriş/çıkış” kartı veya bir kontrol kartı bulunmaktadır.

Üzerinde çalışılan otomasyon sisteminin,

- Kolay kurulabilir, esnek yapılı, bilgisayar programının kolay kullanılır olmasının yanında, uygulanabilir ve maliyetinin düşük olması,
- Seraların, sıcaklık, bağıl nem vb. çevresel faktörleri; bilgisayar kontrollü olarak düzenlenerek, her türlü sera kontrol sistemine uyum gösterecek donanım ve yazılımla tasarlanması,
- Sistemin tek bir sera ya da sera içerisinde tek bir ürün çeşidi için değil; sera içerisindeki tüm ürünlerin değişmesi halinde bile sistemin yeniden ayarlanmasıyla yeni ürünlere adaptasyonu yapılabilmesi,
- Seranın ilk kurulduğu şekliyle değil; gelecekte yapılacak iyileştirmelerde de sorunsuz bir şekilde uyum gösterebilmesi,
- Sera sisteminin uzaktan yönetilmesi için gerekli veri tabanları hazırlanarak; dünyanın her hangi bir yerinden sisteme bağlanarak; sistemin anlık değişim bilgilerine erişilebilmesi amaçlanmıştır.

2. ÖNCEKİ ÇALIŞMALAR

Konu ile ilgili yapılmış çalışmaların bazılarını şöyle sıralamak mümkündür:

Funt ve ark, "Market Model" adında bir donanım ve karar verme yöntemi geliştirildiğini bildiriyor. Amacı yetiştiricinin kendi çiftliği hakkında daha fazla bilgiye sahip olmasıdır. Bu programla yetiştirici 50 farklı ürünün fiyat belirlemesini yapabilmektedir. Faklı yıllarla karşılaştırma yapabilmekte ve belli yılların verilerini göz önüne alarak üretim planlaması yapabilmektedir. Bu işlemler tamamen donanıma yerleştirilen verilerle mümkün olabilmektedir. Bu veriler içinde farklı yıllarda ürün fiyatlarından, gübreye, depolayıcıya ödenen ücrete kadar birçok veri vardır (Funt ve ark, 1989).

Lent ve ark, iki temel bilgisayar program modeli ile sebze ve meyve yetiştiriciliğinde üretim planlaması yapılabileceğini bildirmişlerdir. Bitki için özel olarak seçilen yetiştirme ortamı ve vejetasyon periyodu boyunca bitkinin ihtiyaç duyduğu optimal şartların sağlanmasında bilgisayar sistemlerinden yararlanmışlardır (Lent ve ark. 1989).

Yaptığı bir çalışmada sera sebzeciliğinde en çok karşılaşılan sorunlardan fide harcı seçimi, hastalıkların teşhisi ve mücadele önerisi ile bitki besin elementi ihtiyacı ve gübreleme programının belirlenmesine yardımcı olmak amacıyla BASIC dilini kullanarak bir uzman sistem hazırlamıştır (Tuncay, 1990).

Kuzey Karolina Eyalet Üniversitesinde (North Carolina State University-USA) doktora çalışması süresince, toprak frezesi bıçağı kullanarak toprağın mekanik direncinin belirlenebilmesi amacıyla bilgisayar destekli elektronik bir algılayıcı tasarlanmıştır (Okursoy,1991).

Otomasyon konusunda, mikroişlemci kullanılarak da çalışmalar yapılmıştır. Cam seralara uygun mekanik bir doğal havalandırma projesinde, mikroişlemci tabanlı sıcaklık kontrolü yapılmıştır (Okursoy, 1993).

Seralarda enerji dengesi, bitki enerji dengesi, sera havalandırmasının bilgisayar simülasyonu yardımı ile analizi ve sera dizaynının optimizasyonu, sera bitkilerinin gelişiminin görüntü işleme tekniği kullanılarak belirlenmesi, seralarda

otomasyon bilgisayar kullanımı ile otomatik veri alınması ve ölçüm cihazlarının kontrolü görüntü işleme tekniği uygulamaları konularında çalışmıştır (Kaçıra, 1997).

Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü'nde yüksek lisans tezi olarak meyve fidanı çoğaltılmasında kullanılan köklendirme seralarının otomasyonu tasarlanmıştır. Yapılan bu çalışmada, meyve fidanı yetiştirmek amaçlı, köklendirme seralarında kullanılmak üzere otomasyon sistemi oluşturulmuştur. Otomasyonda internet ve internet haberleşme teknikleri kullanılmıştır. Üzerinde çalışılan otomasyon sisteminin uygulanabilir olması ve maliyetinin düşük olması amaçlanmıştır. Köklendirme serasının, sıcaklık ve bağıl nem gibi çevresel faktörleri bilgisayar kontrollü olarak düzenlenmiştir (İnan, 2002).

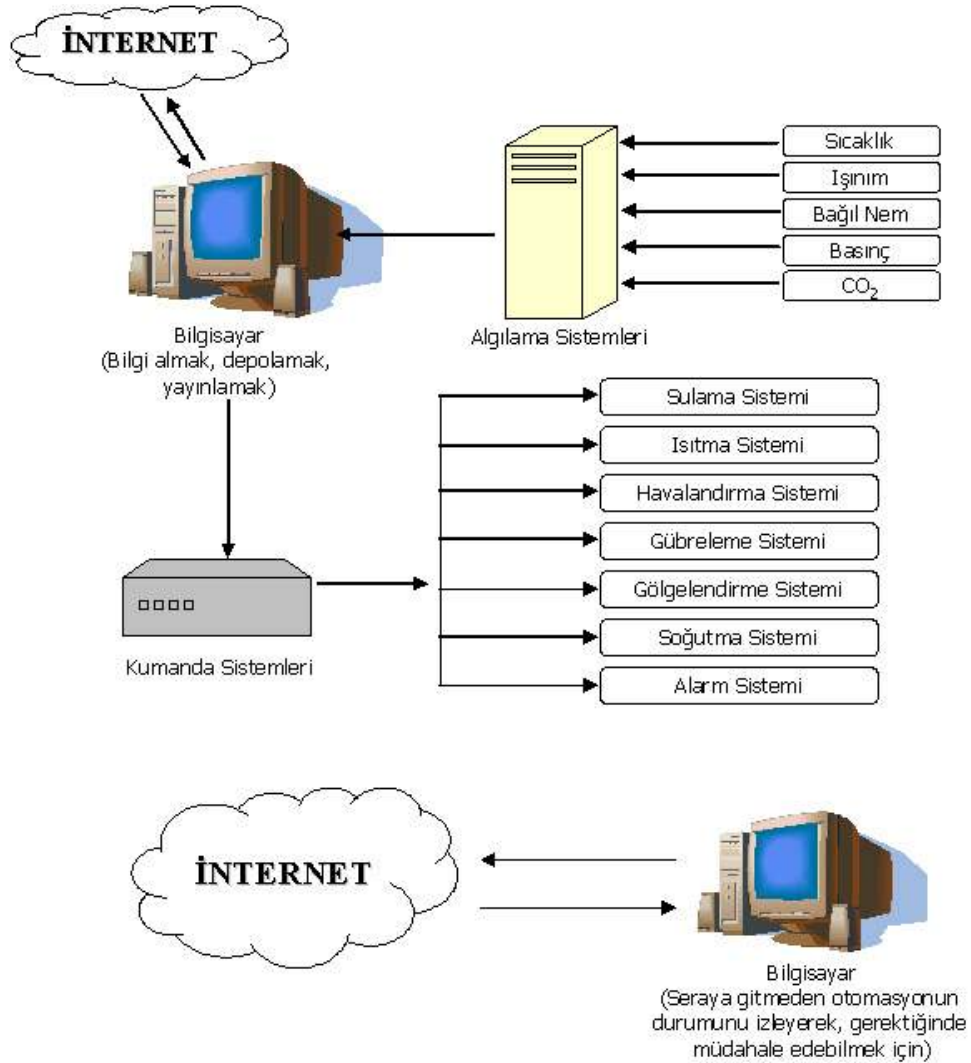
A.A.Yelken tarafından seralarda iklim kontrolüne yönelik bir bilgisayar programı geliştirilmiştir. Programın hazırlanmasında Visual Basic programlama dili kullanılmış olup geliştirilen programda seralarda iklim kontrolü için, iç ve dış ortamdaki; sıcaklık, bağıl nem, CO₂, güneş ışıınımı ve hava hızı gibi çevresel etmenler ölçülmektedir. Sera iç ortamında ölçülen değerler bitkinin en uygun koşullarda gelişmesi için programa girilen çevresel etmen değerleriyle karşılaştırılır. Karşılaştırma sonucunda, kontrol edilmesi gereken iklim etmenine bağlı olarak ısıtma, havalandırma, nemlendirme, aydınlatma gibi gerekli iklimlendirme işlemi gerçekleştirilmektedir (Yelken, 2003).

Kürklü A. ve Çağlayan N., sera otomasyon sistemlerinin geliştirilmesine yönelik bir çalışma yayınlamıştır. Bu çalışmada seralarda kullanılan otomasyon sistemleri, iklim ve sulama-gübreleme sistemleri şeklinde genel olarak incelenmiş, ülkemizdeki seraların teknolojik seviyelerinin ve üreticinin konfor seviyesinin yükseltilmesine katkı sağlayabilecek laboratuvar bazlı bir iklim kontrol çalışmasının sonuçları verilmiştir. Seralarda kullanılan otomasyon sistemleri genellikle ya geri beslemeli (feed-backward) veya ileri beslemeli (feed-forward) kontrol yöntemlerini kapsayan uzman sistemlerini (expert systems) içermektedir. Geri beslemeli sistemlerde, değişkenin algılanmasından sonra kontrol sistemi devreye girerken, ileri beslemeli sistemlerde değişkenin değişim eğilimi önceden tahmin edilerek, kontrol sisteminin gerekli tepkiyi vermesi sağlanmaktadır. Sıcaklık, bağıl nem, ışık, rüzgâr, yağmur gibi bitki yetiştiriciliğinde önemli parametrelerin kontrolü ve tehlike

sınırlarında uyarı amaçlı geri beslemeli bir otomasyon sistemi laboratuvar ortamında bir sera maketi üzerinde yapılmış ve denenmiştir. Sera maketi üzerine bağlı sıcaklık, bağıl nem, rüzgar, yağmur ve ışık algılayıcılarından alınan analog sinyaller bir ADC (analogue to digital convertor) kartı kullanılarak sayısal sinyallere dönüştürülmüştür. Bu sayısal sinyaller aynı zamanda bir gösterge vasıtasıyla görüntülenmiştir. Seranın kontrolü daha önceden girilen ayar değerlerine göre otomatik olarak sağlanabildiği gibi, bir bilgisayar tarafından da izlenebilmektedir (Kürklü ve Çağlayan, 2005).

3. MATERYAL ve METOD

Otomasyon sistemi, iklimsel büyüklükleri, uygun algılama sistemleri ile algılayarak, sayısal veriye dönüştürür ve bilgisayara depolar. Kalibre edilmiş verileri aynı zamanda internet üzerinden erişime sunar. Bilgisayarın ve üzerine kayıtlı yazılımın diğer görevi ise, önceden belirtilmiş şartlar ile verileri karşılaştırarak gerekli kontrol sinyallerini oluşturmak ve ilgili sistemin otomatik olarak devreye girmesini ya da devreden çıkmasını sağlamaktır. Aşağıda böyle bir sistemin şematik resmi görülmektedir (Şekil 3.1).



Şekil 3.1. Otomasyon sistemi düzeneği

3.1. Algılayıcılar

Sera içerisindeki fiziksel büyüklükleri algılamak ve bu fiziksel büyüklükleri kontrol edebilmek amacıyla gerekli verileri sağlamak için algılayıcılar kullanılır. Bu elemanlardan alınan analog ya da sayısal bilgiler ise; otomasyonu sağlayan sisteme aktarılır. Algılayıcılar ile sera içerisindeki; sıcaklık, bağıl nem, basınç, toprak nemi, yaprak nemi, rüzgar hızı ölçülebilmektedir. Algılayıcılardan alınan verilerin çoğu analog veriler olduğundan, verilerin sayısal sistemler tarafından kullanılabilir hale getirilmesi için dönüştürücüler kullanılır.

Otomatik kontrol sistemlerinin çalışmasındaki ilk aşama; kontrole taban oluşturacak temel verilerin elde edilebilmesidir. Otomasyon uygulamalarında genel olarak; elektriksel büyüklükler ya doğrudan ölçülerek ya da fiziksel büyüklüklerden dönüştürücüler yardımıyla dönüştürülerek elde edilebilmektedir.

Algılayıcılar ile dönüştürücüler, çoğu zaman iç içe bir yapı oluşturmakla beraber, ayrı ayrı yapılar içinde de bulunabilmektedir. Dönüştürücüler, işlevsel özelliklerine göre genellikle transmitter, algılayıcı, detektör, cell, gaugar, toplayıcı ve prob olarak ifade edilebildiği gibi, bazı büyüklüklerin önüne "ölçer" ifadesi getirilerek; ivme ölçer, debi ölçer ve hız ölçer (takometre) gibi adlarla da bildirilebilmektedir.

Algılayıcılar sıcaklık, basınç, uzaklık, hız, ivme, sıvı akışı, ışık yoğunluğu, gerilim, akım, direnç, kuvvet ve tork gibi değerlerin ölçülmesini sağlayacak elektriksel davranış değişikliklerini gösteren elemanlardır. Yapısal özelliklerine de bağlı olarak; dönüştürücülerin verdikleri sinyaller, analog, dijital veya kodlanmış dijital sinyal karakterinde olabilmektedir.

Sera otomasyon sistemi için; sıcaklık, ışık ve bağıl nem gibi çevresel faktörlerin algılanması yanında, toprak nemi, rüzgar hızı, sera dış sıcaklığı, sera iç sıcaklığı, toprak sıcaklığı, ışık şiddeti, toprak ıslaklığı ve yaprak ıslaklığının da ölçülmesi gerekebilir. Ölçülmek istenilen değerler için algılayıcı sayısını arttırmak mümkündür.

3.1.1. Sıcaklık Algılayıcı

Sera içi iklim kontrolünde en önemli etkenlerden biri, sıcaklık olduğundan en sık ölçülen büyüklük olmuştur. Hemen hemen her kimyasal, biyolojik işlem ve reaksiyonlar sıcaklığa bağlı olduğundan, sıcaklık değerinin kritik olduğu durumlarda, uygun olmayan sıcaklıklardan sonuçlanan önemli hasarlar meydana gelmektedir.

Bu durumlar göz önüne alınarak, metal ve yarı iletkenlerin sıcaklık karşısında direnç değiştirmeleri prensibine dayanarak, termorezistif dönüştürücüler imal edilmiştir. Termistörler iki tiptir; birincisi negatif sıcaklık katsayılı NTC termistör, diğeri ise pozitif sıcaklık katsayılı PTC termistördür.

NTC'ler oldukça büyük bir negatif sıcaklık katsayılarına sahip dirençlerdir. Demir grubu oksitlerinden imal edilen NTC'lerin dirençleri metallerin aksine artan sıcaklıkla çok büyük bir azalma gösterirler. Direncin sıcaklığa böyle büyük ölçüde bağlı olması sayesinde, bu elemanlardan sıcaklığın elektriksel olarak ölçülmesinde yararlanılmaktadır (Şekil 3.2).

PTC'ler, sıcaklığın artmasıyla büyüyen elektriksel bir dirençtirler. Direnç sıcaklık karakteristiğinin hızla yükselmesi, bu elemanın sıcaklık ölçme elemanı olarak kullanılmasına imkân sağlar. Böylece, bu elemanlar az bir zahmetle ölçme ve kontrol uygulamaları için kullanılmaktadırlar.



Şekil 3.2. NTC ısı algılayıcı

3.1.2. Işık Algılayıcı

Seralar için ışık gereksinimi bitkiye göre değişim gösterebildiği gibi, bir bitkinin ışığa olan gereksinimi de farklı zamanlarda yine farklı düzeylerde olabilmektedir. Bitkiler gelişme dönemlerinde bol ışığa, buna karşılık dinlenme

dönemleri süresince daha az ışığa gereksinim duyarlar. Sera içerisindeki ışık miktarını ölçmek, uygun şekilde ayarlamak ve veri tabanı dosyası olarak kaydetmek seralarda yetiştirilen ürünlere ait olan araştırmalarda çok faydalı olacaktır. Bu sebeple sera içerisindeki ışık şiddetini ölçmek için; sera içerisine elektriksel karakteristiği ışık ile değişen foto dirençler yerleştirilir. Yapılan küçük bir devre ile algılayıcı olarak bu dirençler kullanılarak sera içerisindeki ışık ve buna bağlı olarak dalga boyu gibi veriler algılanmaktadır (Şekil 3.3).



Şekil 3.3. Foto direnç (LDR)

3.1.3. Bağlı Nem Algılayıcı

Bağlı nem oranı, bitki yaprakları ve ortam havası arasındaki buhar basıncı farkını etkilediğinden, terleme işlemi için önemli olmaktadır (Başçetinçelik ve Öztürk, 1998).

Sera içerisindeki bağlı nemi veya toprak nemini ölçmek için nem algılayıcıları kullanılır. Havadaki nemi ölçmek için basit bir kapasitif devre kullanılmaktadır. Bu devrenin çalışma mantığı, havadaki nem miktarı ile birlikte, devrenin kapasitif değerinin değişmesini sağlayan iki metal levha arasına havanın nemini tutacak bir madde yerleştirilmesidir. Bu değişim ADC devre ile sayısal değerlere dönüştürülüp, gerekli olan işlemler gerçekleştirilir. Toprak nemi için ise; toprak içerisine belirli aralıklarla daldırılan metal problemlerin kapasitif değişimi ölçmesi sağlanır (Şekil 3.4).



Şekil 3.4. SHS-A3 RoHS nem algılayıcı

3.2. Kontrol Sistemleri

Bilgisayarın ve kendisine bağlı olan elektronik sistemin düşük gerilim ile çalışması ve sera sisteminin yüksek gerilim veya üç faza ihtiyaç duyması sebebiyle, bilgisayar sistemi ile sera sistemlerinin kontrol edilmesinde, yüksek gerilim ya da akımda çalışabilen kontrol elemanları gereklidir. Bu kontrol elemanları sera sistemi ile bilgisayar sistemi arasında geçiş elemanı olarak görülür ki; bunlara sürücü devre elemanları denir.

Kontrol sistemi, sera otomasyon sisteminin kas sistemini oluşturduğundan, gerçekleştirilen otomasyon sisteminin her serada çalışmasını sağlamak amacıyla; sürücü elemanların tamamını kontrol edebilecek şekilde rölelerden oluşan bir devre grubu tasarlanmıştır. Bu röle grubu, sürülmesi olası devre elemanlarının (motor, fan ve valfların) çalışmasını sağlayacak şekilde tasarlanmıştır.

3.2.1. Röle

Röle, içerisinde bobin ve yumuşak demir göbekten oluşan elektromekanik bir parçadır ve düşük gerilim ile yük kontrolü yapıldığı durumlarda kullanılmaktadırlar. Bilgisayar ve elektronik sistem, standart olarak 5 volt ile 24 volt arasında doğru akım ile çalışmakta, ancak kontrolü yapılacak çoğu cihaz 220 volt alternatif akım ile çalıştığından, otomasyon sistemi içinde röle ve benzeri elemanların kullanılması

gereklidir. Kullanılacak olan rölelerin seçiminde rölenin bobin akım ve gerilimi ile kontak akım ve gerilim değerleri göz önünde bulundurulmalıdır (Şekil 3.5).



Şekil 3.5. 12–220 volt röle

3.2.2. Kontaktör

Rölelerin kontak akımının, geriliminin ya da kontak adedinin yetersiz kaldığı durumlarda, kullanılan kontaktörlerin çalışma prensipleri röleler ile aynıdır. Kontaktörler özellikle 3 fazı aynı anda açıp-kapatma özelliğine sahip olmaları ve kontak akımlarının yüksek olması sebebiyle tercih edilirler. Tek faz ile çalışan sistemler, çok fazla akım çekiyorsa, çektiği akım miktarına göre kontaktör seçimi yapılmalıdır (Şekil 3.6).



Şekil 3.6. Üç fazlı kontaktör

3.2.3. Valf

Sera sulama sisteminde ve ısıtma sisteminde sıvıların akışının kontrol edilmesinde kullanılan valfler, 220 volt alternatif akım ve 24 volt doğru ve alternatif akım ile çalışan modelleri olan elektromekanik parçalardır.

3.3. Sera Sistemleri

Bitkilerin ihtiyacı olan maddelerin bitkilere verilmesi ve gerekli olan iklim ortamının oluşturulabilmesi için; sera otomasyon sistemlerine ihtiyaç vardır. Elektrik enerjisi ile çalışması veya elektrik enerjisi ile kontrol edilmesi gereken sistem; sıcaklık, bağıl nem, ışık gibi parametrik değerleri bitkinin ihtiyaç duyduğu şekilde ayarlar.

3.3.1. Sulama Sistemi

Sera içindeki bitkilerin su ihtiyaçlarını karşılamak amacıyla kurulan sulama sistemi; bitkinin su ihtiyacı olup olmadığını, algılayıcılardan gelen bilgiler doğrultusunda tespit ederek; sulama işlemini otomatik olarak başlatır veya sona erdirir.

Otomatik kontrollü sulama sistemlerinde veri kazanım ünitesiyle topraktan, bitkiden veya çevreden elde edilen bilgiler mikroişlemcilerde işlenmekte ve yorumlanmaktadır. Mikroişlemciden çıkan sulama kararına göre de sulama işlemi gerçekleştirilir.

Sulamada yüksek kullanım etkinliği sağlayan algılayıcılar ve mikroişlemcilerle bütünleşik otomatik kontrollü sulama uygulamaları başlatılmasıyla, sulama yönünden bitki-su-toprak üçlüsü arasında kararlı bir denge oluşturulmuştur. Böylece, sulama kararlarının alınmasına kaynak sağlayacak çeşitli değişkenler, etkin biçimde ve kısa sürede algılanarak, sulamanın otomatik olarak yapılması sağlanır.

Sera sulama sistemi, sulama suyunun bitki köklerinden veya bitki üzerinden dolaştırılmak suretiyle oluşturulur. Buna göre; sulama sistemi, damlama sulama ya da yağmurlama sistemlerinin biri ya da bitki türüne göre her ikisinden de oluşabilir. Sulama suyunun otomatik olarak açılıp kapanabilmesi için elektrik ile çalışan valf kullanılır. Sulama valfinin açılması ile birlikte, basınçlı su sistemde belirtilen süre boyunca sulama işlemini gerçekleştirir.

3.3.2. Isıtma Sistemi

Isıtma sistemi özellikle kış aylarında büyük önem kazanmaktadır. Farklı türlerde olabilecek olan ısıtma sistemlerinde, genellikle su ısıtmalı sistemler kullanılmaktadır. Seralarda kullanılan ısıtıcı yüzeyleri oluşturan borular yetiştirme masalarının altına, toprak altına, sera tabanına, yan duvarlar boyunca ve tabana yakın, sera duvarları boyunca da taban ve tavana yakın yerleştirilirler (Titiz, 2004). Toprak sıcaklığı, toprak sıcaklık probu kullanılarak ölçülür ve sıcaklık istenen değere ulaştığı zaman ısıtma sistemi devre dışı bırakılır. Isıtma sistemi, sıcak su tankına en direkt olarak bağlanır. Sıcaklık yeterli düzeye ulaşmayacak olursa, devreye elektrikli ısıtıcı girerek, istenen sıcaklık seviyesine ulaşmaya kadar çalıştırılır. Tüm bu işlemler; otomasyon programındaki parametrik değerler çerçevesinde gerçekleştirilebilir.

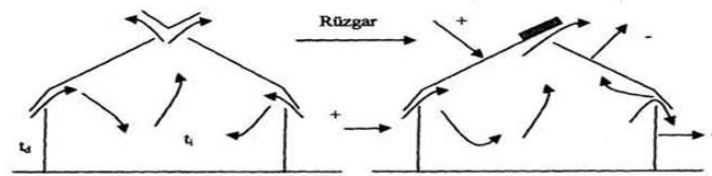
Tüm sistemi dolaşan sıcak su, sıcaklığını seraya ya da bitki köklerine aktardıktan sonra; ısıtıcı kazan içerisine tekrar geri dönmektedir. Isıtma sistemlerinde elektrikli ısıtıcı dışında, kömür, doğalgaz, güneş enerjisi ve diğer alternatif ısıtma kaynaklarının, programa tanıtılması ile kullanmak mümkün olmaktadır. Isıtma sistemi yetiştirilen bitkiye göre farklı şekillerde tasarlanabilir. Isıtma sisteminin devreye girebilmesi için dış sıcaklığın, iç sıcaklığın ve toprak sıcaklığının belirtilen değer dışına çıkmış olması gerekmektedir.

3.3.3. Havalandırma Sistemi

Sera içerisindeki sıcaklığın veya bağıl nemin istenen değerden yüksek olması durumunda, soğutma sistemi devreye girmeden önce veya havalandırılmanın serada yapılacak çalışmalar için gerekli görüldüğü hallerde, havalandırma kapaklarının açılması suretiyle içerideki sıcaklık veya bağıl nem miktarının düşürülmesi veya gerçek anlamda havalandırmanın yapılması sağlanmış olur (Başçetinçelik ve Öztürk, 1997).

Havalandırma sistemi, her serada farklı şekillerde tasarlanabilir. Havalandırma yapılması düşünülen sera için, havalandırma kapakları mekanik bir sistem tarafından açılıp-kapatılabilecek şekilde tasarlanmalıdır. Böylece kapakları açıp kapatacak mekanik sisteme elektrikle veya hava ile çalışan bir kontrol sistemi ile etki edilebilecektir. Tasarlanan kart üzerinde yer alan elektronik sistem tarafından kontrol edilebilen elektrik motoru veya hidrolik bir sistem mekanik sistemi kontrol edecektir. Havalandırma kapakları yüzdelik değerliklerle açılıp-kapatılabilmektedir.

Kapakların tam olarak açık veya kapalı olma durumlarını kontrol eden sınır anahtarları konularak, havalandırma sistemi açma ya da kapatma anında sistemin sınırların dışına çıkması veya mekanik zorlama yapması engellenecektir. Kapakların, rüzgâr hızının yüksek olduğu durumlarda, sistem tarafından otomatik olarak kapatılması gerekeceğinden, programda izin verilmiş olmasına rağmen, kapak kontrollerinin uzun süreli el ile kontrol şeklinde tutulması tavsiye edilmez (Şekil 3.7).



Şekil 3.7. Rüzgarsız ve rüzgarlı havalarda pencere pozisyonları (Hakgören ve Kürklü, 2007)

3.3.4. Gübreleme Sistemi

Sera içerisinde gübreleme işlemlerinin otomatik olarak yapılması da mümkündür. Bu işlem için uygulanacak yöntemlerden birisi, bitkiye verilecek gübrenin ya da ilacın, eğer bitkinin köküne ya da toprağa verilmesi gerekiyor ise, homojen dağılımı sağlanarak damlacık sulama sistemiyle birlikte bitkiye verilmesidir. Diğer yol ise, eğer gübre ya da ilaç bitkinin yapraklarına verilecek ise; yine homojen dağılım sağlanarak fiskiyelerle sulama sistemi ile bitkiye verilebilir. Her iki durumda da sistem, sulama sistemiyle birlikte düşünülmeli ve homojenliğin sağlanabilmesi için gerekli hazırlama tankları sisteme eklenmelidir. Burada kullanılacak malzemeler sulama sisteminde tıkanmaya sebep olabilecek ise, bunun içinde az miktarda bazı açıcı malzemelerle sistemdeki tıkanmaların engellenmesi sağlanmalıdır.

3.3.5. Gölgeleme Sistemi

Seralardaki sıcaklığı doğrudan etkileyen ışık miktarı, özellikle sıcak yaz aylarında, gölgeleme sistemi ile kontrol altına alınabilir. Gölgeleme sistemi, basit bir mekanik düzenek ve elektrik motorları ile gerçekleştirilerek, seradaki ısının soğutma sistemi devreye girmeden önce azaltılmasını sağlamaktadır.

Bu sistem, ışık miktarı ve sıcaklık parametreleri kontrol edilerek, otomasyon sistemi tarafından kontrol edilir.

3.3.6. Soğutma Sistemi

Sera içindeki sıcaklığın belirtilen değerlerden yüksek olması durumunda, soğutma işlemini gerçekleştirecek olan sistemdir. Genel olarak suyun yoğunlaşması yöntemiyle çalışan bu sistem, ıslak pad ve havalandırma fanından oluşur. Soğutma sistemi devreye girmeden önce, dış ortam sıcaklığı ve rüzgâr hızı kontrol edilir. Önce

havalandırma kapakları açılarak sera içerisi soğutulmaya başlanır. Eğer yeterli soğutma işlemi yapılamıyorsa, soğutma sistemi devreye girmelidir.

3.3.7. Alarm Sistemi

Sera sistemlerinin tam olarak çalışmaması durumunda, alarm sistemi devreye girmektedir. Alarm sistemi, sesli uyarı vermek, program üzerinden bilgi vermek veya belirtilecek farklı işlemleri de gerçekleştirebilmesi için tasarlanmıştır. Alarm sisteminin etkin bir şekilde çalışması, sera sistemlerinin tasarlanmasına ve sistemlerden bilgi alınmasına bağlıdır. Bu durum, otomasyon sisteminin verimliliğini arttırmaktadır. Sağlıklı bir bilgi akışı ile sera sistemi arızalarının bulunması ve giderilmesi daha kolay olur.

Alarm sistemi sera içerisindeki güvenliğin sağlanması amacıyla da kullanılır. Bu sistem sayesinde, belirtilen saatler dışında sera içerisine girilmesi durumunda, sistem otomatik olarak devreye girerek, sesli uyarı veya diğer uyarma yöntemleri etkin duruma getirilebilir.

3.4. Elektronik Kart ve PIC Programı

Sistem için hazırlanan elektronik kartın temelde iki işlevi vardır. Birinci işlevi, algılayıcılardan elde edilen bilginin bilgisayara ulaştırılmasıdır. Burada bilgi akışı sayısal olduğundan, eğer algılayıcı analog ise bilgi sayısal olarak dönüştürülerek aktarılır. İkinci işlevi ise; bilgisayardan gelen kontrol sinyallerine uygun olarak, çıkış cihazlarının açılıp-kapanmaları için gerekli kontrol sinyallerini sağlamaktır.

Elektronik kart birinci işlevi olan ölçme değerlerini bilgisayara gönderirken, öncelikle bu isteğin bilgisayardan gelmesini bekler. Ana programda kart ile elemanlar arasındaki bağlantı ilişkisi tanımlanırken, elemanların ne kadar süre aralıklarla okunacağı programda belirtilir. Buna göre de bilgisayar bir adresleme mantığı yardımıyla (kaç numaralı kartın, kaç numaralı ucu) o anda ölçüm değerini bilmek istediği algılayıcının bilgisini iletişim hattına verir. Sistemde birden fazla kart varsa, bu istek tüm kartlara iletilir. Karttaki program, kodu çözerek mesajın kendisi

ile ilgili olup olmadığını anlar. Eğer bilgi kendisi için ise, ilgili portun üzerindeki bilgiyi okur. Bu okuma esnasında algılayıcı analog bile olsa, bilgi sayısala dönüşmüş olur. Bu sırada bilgisayar, gönderdiği bilginin cevabını almak için beklemektedir. Basit bir network şeklinde tasarlanmış olan iletişim hattına okunan bilgi verilir. Hattın karşı tarafındaki bilgisayar da bu bilgiyi okuyarak istediği veriyi almış olur. Bu haberleşme de cevap, belirlenen sürede gelmezse bilgisayar tekrar aynı isteği yineler ve 3 kez erişilemez durumu oluşursa da mesajla hata bilgisini verir.

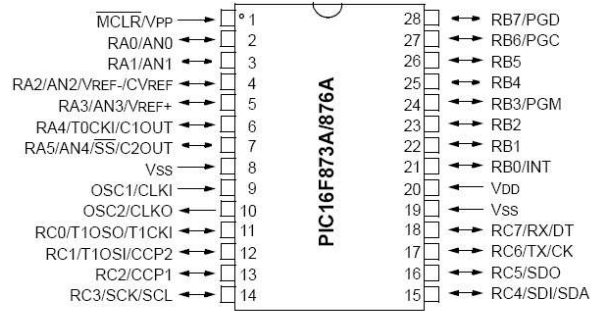
Elektronik kart, ikinci işlevi olan kontrol sinyallerini bilgisayardan alıp kontrol aletlerine aktarırken, öncelikle bu yönde bir bilginin bilgisayardan gelmesini bekler. Ana programda çıkış elemanlarının yapacağı işler tasarlanırken, açma kapama ya da belirli açılarda açık kalma şartları verilmiştir. Bu yapılandırma da bilgisayar kartlardaki algılayıcılardan bilgiler aldıktan sonra, bu şartları kontrol eder. Bu kontrol sonucunda pozisyonu değişmesi gereken çıkış elemanı belirlenirse, bu elemanla ilgili açma ya da kapatma komut bilgisi iletişim hattına verilir. Bu bilginin yapısı yine algılayıcılardaki adres mantığına benzer (kaç numaralı kartın, kaç numaralı ucu, ON ya da OFF pozisyonu) şeklindedir. Sistemde birden fazla kart varsa bile bu komut tüm kartlara iletilir. Karttaki program kodu çözerek, mesajın kendisi ile ilgili olup olmadığını anlar. Eğer, bilgi kendisi için ise; gelen komutu istenilen port ucuna uygular. Böylece, ilgili çıkış elemanı açılarak ya da kapatılarak zaman fonksiyonunun da yardımıyla on/off ya da yüzdelikli açma sağlanmış olur.

3.4.1. Elektronik Kartta Kullanılan Entegreler

3.4.1.1. PIC16F873 Entegresi

Elektronik kartın en önemli elemanı, mikrodenetleyici olan PIC16F873 entegresidir. Kontrol ve kumandalar için gerekli PIC yazılımı yüklenerek algılayıcı ve kumandaların yönetimini gerçekleştirir.

28-Pin PDIP, SOIC, SSOP

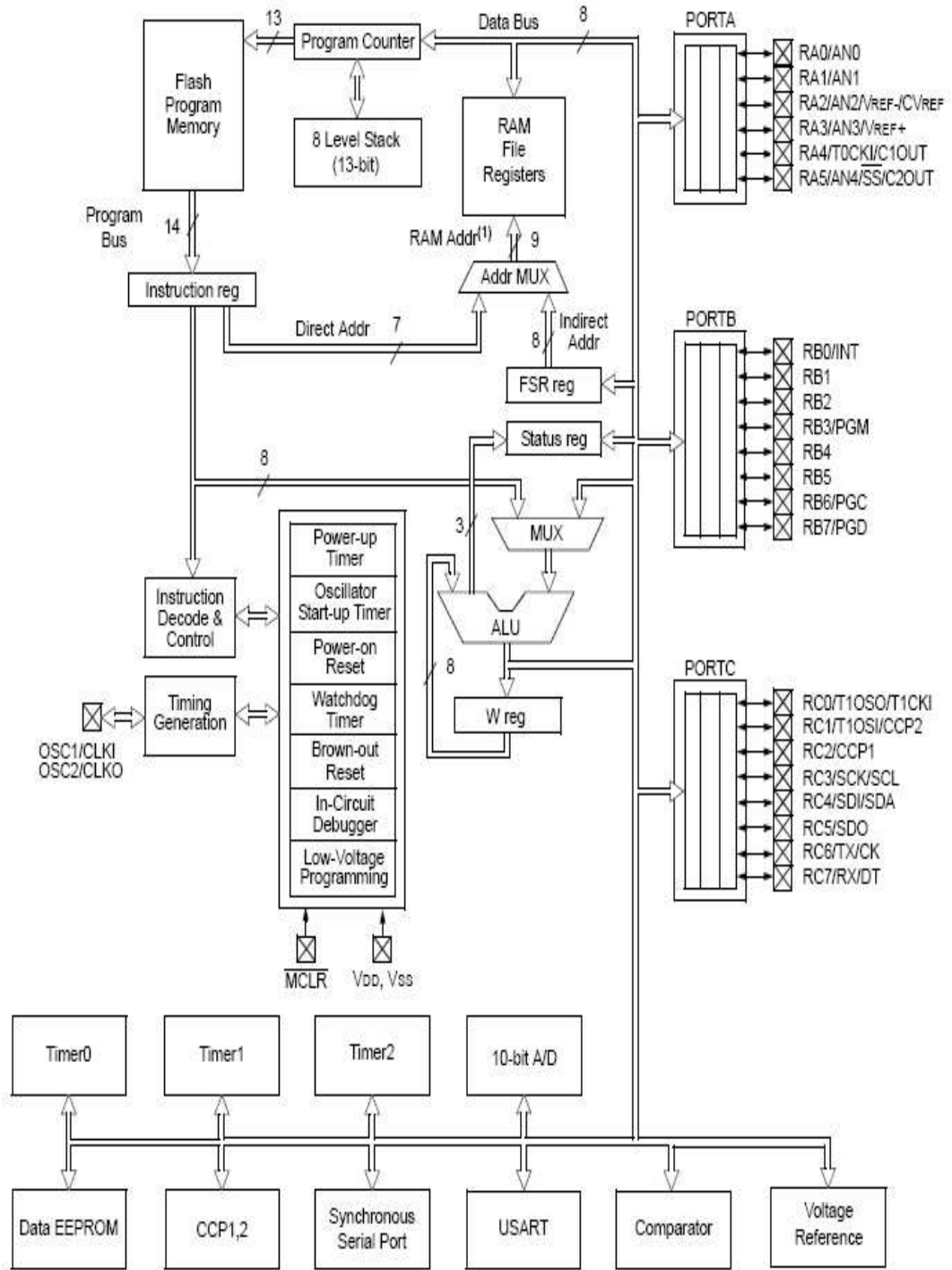


Şekil 3.8. PIC16F873 bacak bağlantıları

Çizelge 3.1. PIC16F873 teknik özellikleri

Teknik özellikler	PIC16F873 için değeri
Çalışma Frekansı	DC - 20 MHz
RESET	POR, BOR (PWRT, OST)
FLASH Hafıza Kapasitesi (14-bit)	4K
RAM Hafıza Kapasitesi (byte)	192
EEPROM Hafıza Kapasitesi (byte)	128
Kesme adedi	14
I/O Portları	Ports A,B,C
Timer	3
Darbe Genişliği Modülasyonu (PWM)	2
Seri İletişim	MSSP, USART
10-bit Analog-Digital Dönüştürücü	5 giriş kanalı
Analog Karşılaştırıcı	2
Komut Seti	35 komut
Paket tipi	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN

PIC16F873 entegresinin bacak bağlantıları (Şekil 3.8), teknik özellikleri (Çizelge 3.1), blok diyagramı (Şekil 3.9) ve bacak bağlantı bilgileri (Çizelge 3.2) ayrıntılı olarak verilmiştir.



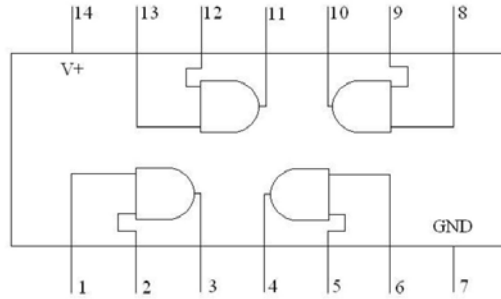
Şekil 3.9. PIC16F873 blok diyagramı

Çizelge 3.2. PIC16F873 bacak bağlantı bilgileri

Pin Adı	Pin Num.	I/O/P Tipi	Tampon Tipi	Tarif
OSC1/CLKIN	9	I	ST/CMOS(3)	Dahili osilatör kristal girişi ve harici osilatör girişi
OSC2/CLKOUT	10	O	—	Dahili kristal osilatör girişi ve osc1 deki clock sinyalinin ¼ oranı ile çıkışı (komut periyodu çıkışı)
MCLR/VPP	1	I/P	ST	Ana RESET ucu
RA0/AN0	2	I/O	TTL	PORTA iki yönlü GİRİŞ/ÇIKIŞ portudur. RA0, ayrıca analog input AN0 olarak kullanılabilir. RA1, ayrıca analog input AN1 olarak kullanılabilir RA2, ayrıca analog input AN2 olarak kullanılabilir. Extra olarak negatif analog gerilim referans girişi olarak kullanılabilir. RA3, ayrıca analog input AN3 olarak kullanılabilir. Extra olarak pozitif analog gerilim referans girişi olarak kullanılabilir. RA4 Open Drain bağlantılıdır. Ayrıca timer0 clock girişi olarak kullanılabilir. RA5, ayrıca analog input AN5 olarak kullanılabilir. RA5 aynı zamanda slave moda senkron iletişim seçme ucudur.
RA1/AN1	3	I/O	TTL	
RA2/AN2/VREF	4	I/O	TTL	
RA3/AN3/VREF+	5	I/O	TTL	
RA4/T0CKI	6	I/O	ST	
RA5/SS/AN4	7	I/O	TTL	
RB0/INT	21	I/O	TTL/ST(1)	PORTB iki yönlü GİRİŞ/ÇIKIŞ portudur ayrıca tüm uçları dahili PULL-UP yapısındadır. RB0 aynı zamanda kesme ucu olarak kullanılabilir. RB3 alçak gerilim programlama ucu olarak kullanılabilir. Sinyal değişim kesmesi olarak kullanılabilir. Sinyal değişim kesmesi olarak kullanılabilir. Sinyal değişim kesmesi olarak kullanılabilir. Ayrıca seri programlama clock ucudur. Sinyal değişim kesmesi olarak kullanılabilir. Ayrıca seri programlama data ucudur.
RB1	22	I/O	TTL	
RB2	23	I/O	TTL	
RB3/PGM	24	I/O	TTL	
RB4	25	I/O	TTL	
RB5	26	I/O	TTL	
RB6/PGC	27	I/O	TTL/ST(2)	
RB7/PGD	28	I/O	TTL/ST(2)	
RC0/T1OSO/T1CKI	11	I/O	ST	PORTC iki yönlü GİRİŞ/ÇIKIŞ portudur. RC0 aynı zamanda timer1 osilatör ucu olarak kullanılabilir. RC1 aynı zamanda timer1 osilatör giriş ucu, capture2 giriş ucu, PWM2 çıkış ucu olarak kullanılabilir. RC2 ucu capture giriş, karşılaştırıcı çıkış, PWM1 çıkış ucu olarak kullanılabilir. RC3 senkron seri iletişim clock ucu . RC4, SPI moda data giriş ucu. RC5, SPI moda data çıkış ucu. RC6 , seri asenkron haberleşmede gönderme ucu RC7 , seri asenkron haberleşmede alma ucu
RC1/T1OSI/CCP2	12	I/O	ST	
RC2/CCP1	13	I/O	ST	
RC3/SCK/SCL	14	I/O	ST	
RC4/SDI/SDA	15	I/O	ST	
RC5/SDO	16	I/O	ST	
RC6/TX/CK	17	I/O	ST	
RC7/RX/DT	18	I/O	ST	
VSS	8, 19	P	—	- Besleme bağlantısı
VDD	20	P	—	+ Besleme bağlantısı

3.4.1.2. İki Girişli AND Kapısı Entegresi (4081)

Sistemde kullanılan seri iletişim modeli bir ring ağ gibi çalışmaktadır. AND kapısının (Şekil 3.10) buradaki görevi, kartların çıkış ucundan farklı zamanlarda gelecek olan seri bilgilerin tek hat üzerinden bilgisayara gönderilmesinin sağlanmasıdır. Bu işlem için AND kapısının girişlerinden bir uc 1 iken, diğer uctan gelecek bilginin aynen çıkışa ulaşması mantığı kullanılmıştır (Çizelge 3.4). İki girişli 4081 entegresinin çalışma gerilimleri Çizelge 3.3'te verilmiştir.



Şekil 3.10. 4081 AND kapısı bacak bağlantısı ve iç yapısı

Çizelge 3.3. 4081 AND kapısı çalışma gerilim değerleri

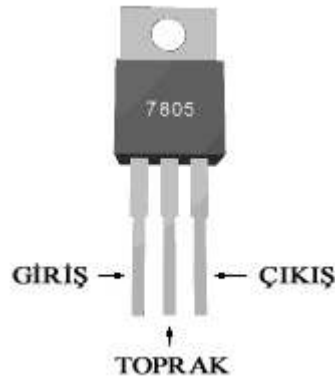
Çalışma gerilimi	3V ile 15V arasında
Maksimum 0 gerilimi	1.5V @ çalışma gerilimi 5V ise
	2.7V @ çalışma gerilimi 9V ise
	3.6V @ çalışma gerilimi 12V ise
	4.0V @ çalışma gerilimi 15V ise
Minimum 1 gerilimi	3.5V @ çalışma gerilimi 5V ise
	6.3V @ çalışma gerilimi 9V ise
	8.4V @ çalışma gerilimi 12V ise
	11.0V @ çalışma gerilimi 15V ise

Çizelge 3.4. 4081 AND kapısı doğruluk tablosu

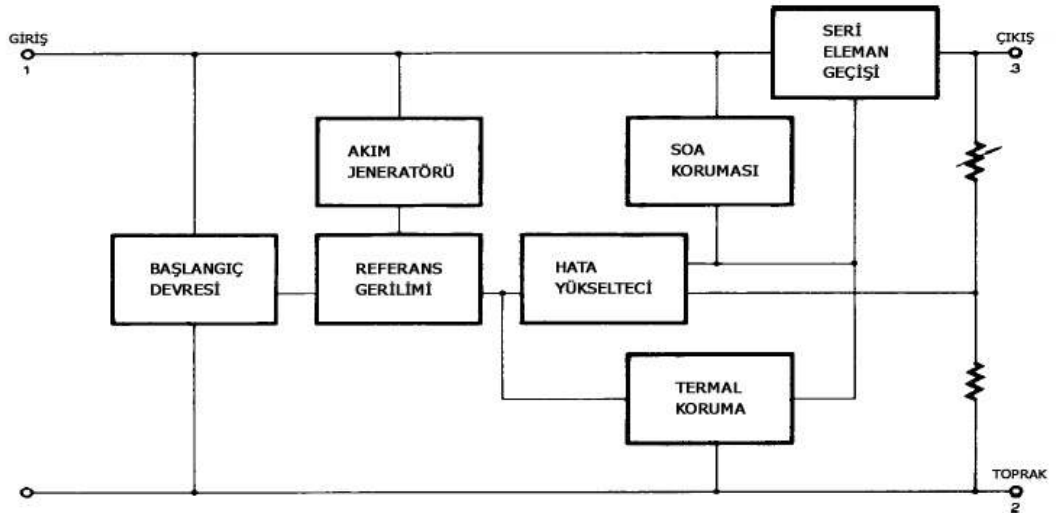
A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1

3.4.1.3. Regüle Entegresi (7805)

Girişten gelen DC gerilimi elektronik kartın çalışma gerilimi olan DC 5V'a çevirmek için kullanılmıştır (Şekil 3.11). 7805 entegresinin blok diyagramı Şekil 3.12'de gösterilmektedir. Bu entegre regüle entegresi olduğundan, çalışma aralıkları ve elektriksel karakteristiklerinin bilinmesi ayrıca önem kazanmaktadır. Çizelge 3.5'te elektriksel karakteristik tablosu ayrıntılı bir şekilde görülmektedir.



Şekil 3.11. 7805 regüle entegresi



Şekil 3.12. 7805 regüle entegresi blok diyagramı

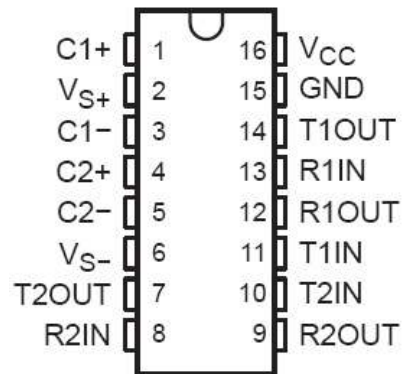
Çizelge 3.5. 7805 regüle entegresi elektriksel karakteristik tablosu

Parametre	Şartlar		Durum			Birim
			Min.	Typ.	Max	
	TJ =+25 oC		4.8	5.0	5.2	V
Çıkış Gerilimi	5.0mA ≤ Io ≤ 1.0A, PO ≤ 15W VI = 7V to 20V		4.75	5.0	5.25	
Boşta Regülasyon	TJ=+25 oC	VO = 7V to 25V	-	4.0	100	mV
		VI = 8V to 12V	-	1.6	50	
Yükte Regülasyon	TJ=+25 oC	IO = 5.0mA to1.5A	-	9	100	mV
		IO =250mA to 750mA	-	4	50	
Sukunet Akımı	TJ =+25 oC		-	5.0	8.0	mA
Sukunet Akımı Değişimi	IO = 5mA to 1.0A		-	0.03	0.5	mA
	VI= 7V to 25V		-	0.3	1.3	
Çıkış Gerilim Gürültüsü	f = 10Hz to 100KHz, TA=+25 oC		-	42	-	µV/VO
Parazit Bastırma	f = 120Hz VO = 8V to 18V		62	73	-	dB
Kesimde Çıkış Gerilimi	IO = 1A, TJ =+25 oC		-	2	-	V
Çıkış Direnci	f = 1KHz		-	15	-	mΩ
Kısa Devre Akımı	VI = 35V, TA =+25 oC		-	230	-	mA
Tepe Akımı	TJ =+25 oC		-	2.2	-	A

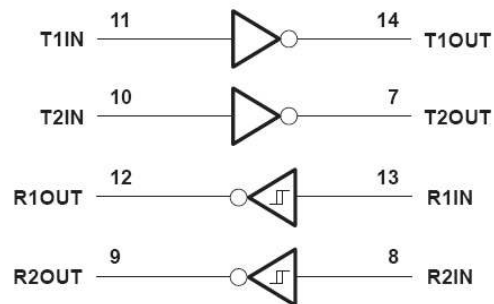
3.4.1.4. max232 Seri İletişim Entegresi

Seranın büyüklüğüne bağlı olarak sistemde birden fazla elektronik kart kullanılması mümkündür. Sera büyüklüğü göz önüne alınarak, iletişim protokolü olarak seçilen seri iletişim elektronik kartta max232 entegresi (Şekil 3.13) ile sağlanmıştır. Bu entegre, seri 0-5 volt aralığında olan seri iletişim sinyallerini -12 volt, +12 volt şekline dönüştürerek bilginin daha uzaklara ve hatasız iletimini sağlar. Ayrıca 0-5 volt olan pic16f873 seri iletişimini bilgisayardaki +12 volt, -12 volt sistemine dönüştürmüş olur.

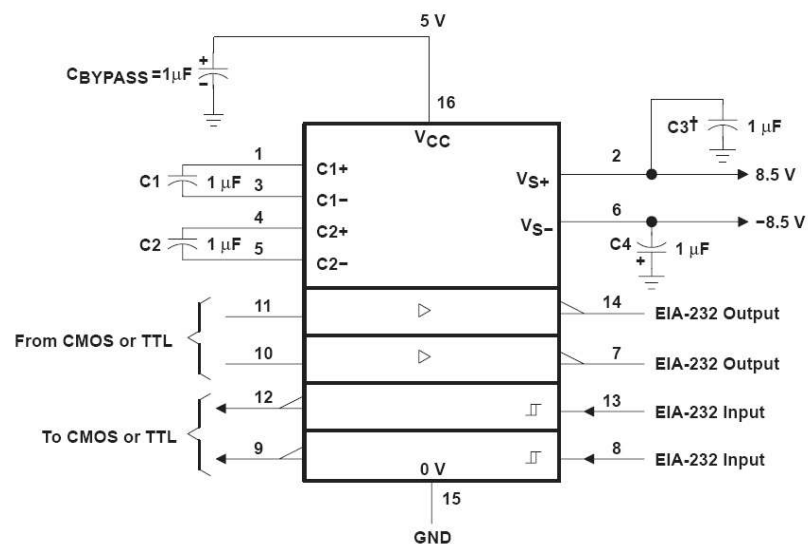
max232 entegresinin çalışma yapılırken mantıksal yapısının bilinmesi gerekmektedir. (Şekil 3.14). Çalışması ile ilgili en genel kullanımı Şekil 3.15'te gösterilmiştir.



Şekil 3.13. max232 seri iletişim entegresi



Şekil 3.14. max232 mantıksal diyagramı



Şekil 3.15. max232 tipik çalışma devresi

3.5. Bilgisayar ve Otomasyon Programı

3.5.1. Bilgisayar

Sistemin kontrolü ve karar vermesini sağlayacak en önemli bölümü bilgisayardır. Bilgisayar, sera sistemini tüm gün boyunca takip edecek ve gerekli kararları vererek, kontrol sistemlerinin devreye girmesi için gerekli sinyalleri üretecektir. Bilgisayarın 7/24 çalışacağı düşünüldüğünde, donanımsal olarak bir kişisel bilgisayar yerine, sunucu özellikli bir bilgisayar seçilmesi isabet olacaktır. Sunucular, 7/24 çalışılacağı varsayımı üzerine (hiç kapanmayacak gibi) tasarlanmış dayanıklı bilgisayarlardır. Seçilecek sunucumuzun Windows işletim sistemi çalıştırılabilecek herhangi bir sunucu olması yeterli olacaktır. İşletim sisteminin Windows olarak belirlenmesi, kullanımının kolay olması ve günümüzde en yaygın şekilde kullanılan işletim sistemi olmasıdır.

Sunucu ile elektronik kart arasında iletişim seri iletişim portu ile yapıldığından, sunucu üzerinde seri iletişim portunun (COM) olması tercih nedeni olabilir. Seri iletişim portu olmayan bir sunucuda ise mutlaka USB portu bulunmalıdır. USB portunu COM porta çevirebilen dönüştürücüler ile kullanabileceğimiz seri iletişim portu elde etmek mümkün olacaktır.

3.5.2. Veri Tabanı

Veri tabanı; sistemin içerisinde oluşan tüm hareketlerin depolandığı gerektiğinde tekrar çağırılarak, bu veriler üzerinde inceleme olanağı sağlayan disk alanlarıdır.

Geliştirdiğimiz otomasyon sisteminde veri tabanı olarak MySQL kullanılmıştır. MySQL veri tabanı ücretsiz bir veri tabanı olmasının yanında, bir veri tabanından beklenen temel işlemleri yeri getirmedeki hızı ile de ön plana çıkmaktadır. Ayrıca, uygulamanın web ile bağlantılı olması da MySQL veri tabanı kullanılmasında etkili olmuştur.

3.5.3. Programlama Dili

Otomasyon programının yazımında görsel bir programlama dili olan Delphi kullanılmıştır. Dilin seçiminde; iletişim portlarında sorunsuz çalışan araçlarının bulunması, grafik ekrandaki yazım kolaylığı ve veri tabanı ile sorunsuz çalışması göz önünde bulundurulmuştur.

3.5.4. Otomasyon Programı

Otomasyonun beyni olan yazılımı bilgisayar ile elektronik kart arasındaki köprü olarak nitelendirebiliriz. Yazılım, tanım, kontrol ve kumanda olmak üzere 3 bölümden oluşmaktadır.

Tanımlamalar

Bu bölümde otomasyon içerisinde uygulanacak olan giriş ve çıkış işlemleri ile bu işlemlerin elektronik kartla olan bağlantıları belirlenir. Örneğin sıcaklık algılama işlemi tanımlanmış ise, bu sıcaklık algılama işlemini hangi eleman ile ölçüleceği, bu elemandan gelen verilerin hangi sıcaklık değerine eş değer olacağı gibi belirlemelerden sonra, bu elemanın kaç numaralı elektronik kartın, hangi portuna bağlı olduğunun yazılıma tanıtılması işlemleri bu bölümde yapılmaktadır.

Bu tanıtım işlemleri doğru yapılmadığında yazılım, gelen değerleri doğru şekilde değerlendiremeyecektir.

İşlem ve elemanların yazılıma tanıtılma işlemlerinden sonra, yapılması gereken son tanımlar otomatik kontrolün nasıl olması ile ilgili şartların belirlenmesidir. Bu şartlar, serada yetiştirilmek istenilen ürüne göre değişim göstereceğinden, ürünler belirlenirken mutlaka ziraat mühendisinden destek alınmalı ve onun gözetiminde yapılmalıdır.

Kontrol

İşlemler, işlem elemanları ve şartların oluşturulmasıyla otomasyon yazılımı çalışmaya başlayacaktır. Yazılım, tüm tanımlı algılama elemanlarından periyodik olarak bilgi istemekte, algılayıcılardan gelen değerleri bağlı bulunduğu veri tabanına kayıt etmektedir.

Kayıt edilen bu veriler ile seranın olması gereken değerleri (oluşturulmuş şartlara göre) kontrol sistemi her defasında karşılaştırır. Karşılaştırma sonucunu kumanda sistemine aktarır.

Kumanda

Kontrol işleminde varılan sonuçlar neticesinde, açılması ya da kapanması gereken kumanda elemanları, bu bölüm vasıtasıyla harekete geçirilir. Bu bölüm sayesinde, tüm kumanda elemanları açık ya da kapalı kalmaları gerektiği kadar kalırlar.

3.5.6. Otomasyon Programının İş Akışı

Program, açılışta halen var olan veri tabanındaki durumu görüntüler. Burada ilk kez kullanılıyor ise, tüm alanlar boş olacaktır. Birinci yapılması gereken iş, seraya yerleştirilen kartları ve bu kartlara bağlı elemanların programa tanıtılmasıdır. Bu tanıtım yapılırken; algılayıcılar için okuma zaman aralıkları, ölçüm elemanının tipi seçilmeli ya da girilmelidir. Daha önce hiç kullanılmamış bir ölçüm elemanı kullanılıyor ise, bu eleman için ölçüm-değer karşılığı değerlerinin (bir tür kalibrasyon) girilmesi de gerekir. Bu tanımlamalarda hem giriş hem de çıkış elemanları tanıtılır.

Eleman tanıtımının ardından sera kontrol şartlarının sisteme girilmesi işlemi yapılır. Programın asıl işlevi buradadır. Burası her türlü kontrol şartını girmeye yönelik olacak şekilde ve oldukça esnek tanımlanmıştır. Buradaki tanımlamalar “örneğin; sıcaklık 25 °C’nin üzerinde ise birinci havalandırma penceresini aç” gibi olacaktır. Esnek yapısı sebebiyle bitki türüne ve her türlü çevre şartlarına göre şart oluşturmaya izin verir.

Kumanda şartlarının oluşturulmasının ardından, program çalışma aşamaları işler ve sırasıyla algılayıcılardan bilgi okuma, okunan bilgileri istenilen birimlere dönüştürme, elde edilen verileri otomasyon şartları ile karşılaştırma, karşılaştırma sonucunda çıkış elemanlarının önceki durumlarını değiştirecek bir sonuç olup olmadığının tespiti ve son olarak konumu değişecek çıkış elemanına gerekli kontrol sinyalinin gönderilmesi döngüsünü çalıştırır. Aynı zamanda sistem, elde ettiği veriler

ile kontrol çıkışlarını bilgi ekranı üzerinden verir. Bu bilgi ekranı ayrıca yazılan web yazılımı sayesinde internet üzerinden izlenebilir.

Diğer taraftan program; el ile kontrolün bilgisayar üzerinden yapılmasına da imkân verir. Program üzerinde elle kontrol seçeneği seçilerek ve istenilen çıkış cihazına direk bilgi gönderilerek, açma-kapama işlemleri yapılabilir. Bu sayede de seranın içerisine girmeksizin yapılması gereken işlemler varsa; bu işlemlerin bilgisayar üzerinden yapılması sağlanmış olur.

4. BULGULAR ve TARTIŞMA

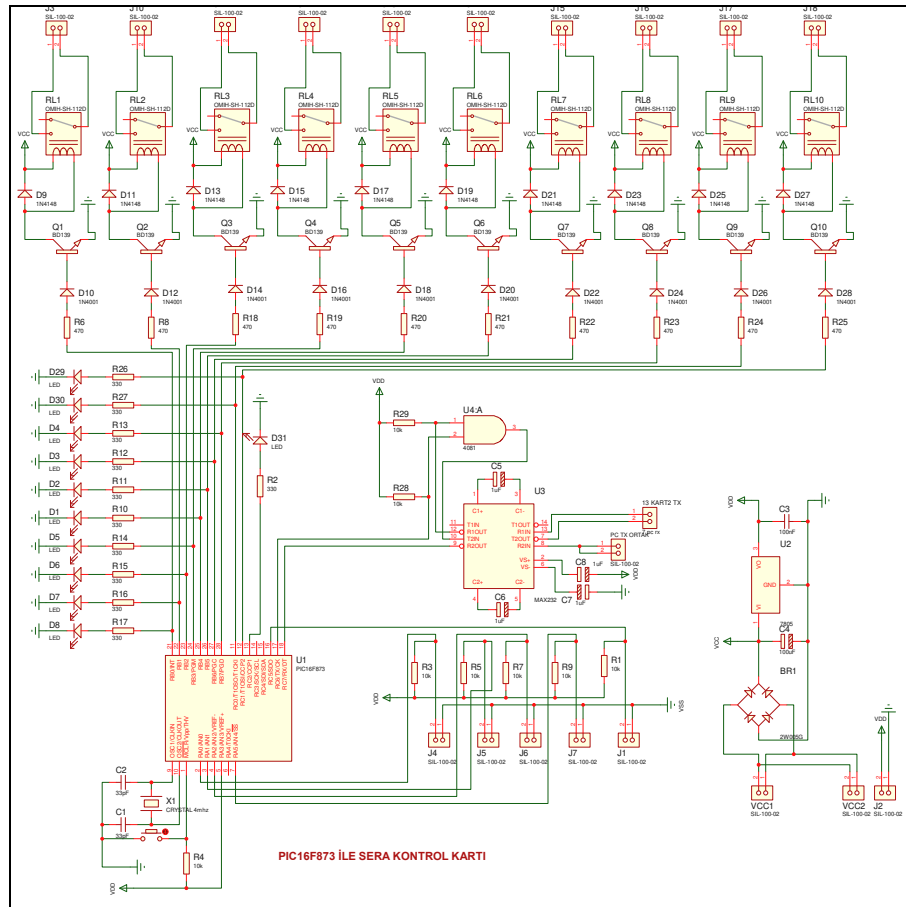
Bir sera içerisinde yapılabilecek otomatik kontroller için tasarlanacak sistemde, iletişimin nasıl yapılacağı, hangi fiziksel büyüklüklerin ölçülebileceği ve bunlara karşılık hangi cihazların, nasıl kontrol edilebileceği konularında çalışılmıştır. Bu çalışma neticesinde, mesafe ve fiziksel alanın olabilecek iklim durumları da göz önüne alındığında, seri iletişimle haberleşen bir sistem tasarlanmasına karar verilmiştir. Sistemin merkezinde bir bilgisayar konularak hem esnek programlanabilmesi ve hemde internet bağlantısının kolay yapılabilmesi özelliklerinden faydalanılmıştır. Ağ yapısında tasarlanan ve eklenebilir özellikli olan sistemin üzerinde, programlanabilen cihazlar kullanılarak, sistemin esnek ve modüler olması program ve elektronik kart yardımıyla sağlanmıştır.

4.1. Elektronik Kartın Tasarımı

Ortam içerisindeki değerlerin bilgisayara aktarılması ve bilgisayardan gelen isteklerin de kumanda sistemlerine aktarılmasını için hazırlanan elektronik kartın şeması (Şekil 4.1) incelendiğinde; temelde 5 bölümden oluştuğu görülecektir.

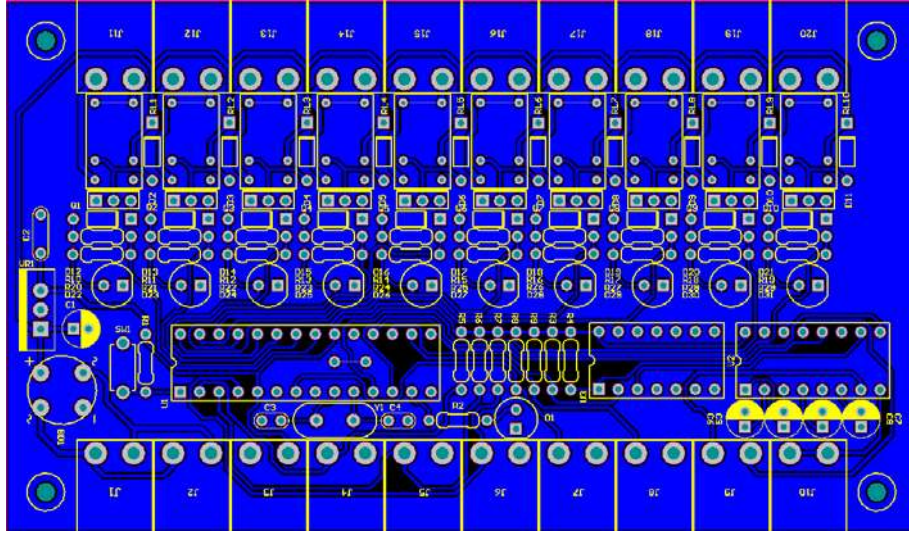
Ana kısım pic16f873 mikrodenetleyici entegresinin olduğu kısımdır ki; gerekli yazılım buraya kaydedilmektedir. İkinci kısım, ledlerden oluşan ve sistemde nerenin aktif olduğunun görülmesi için tasarlanmış bir kısımdır. Üçüncü kısım, basit bir besleme devresidir. Burada gerekli olan 5 voltun üretilmesi sağlanmaktadır. Dördüncü kısım, max232 entegresi ile oluşturulmuş seri iletişim kısmıdır. Sonuncu kısım ise, rölelerden oluşan çıkış elemanlarını kontrol kısmıdır.

Sistemin asıl kısmı pic16f873 içerisindeki assembler ile yazılmış programdır. Bu programda aşağıda verilmiş olup, bölüm başlarında ya da açıklayıcı olacağı düşünülen bölümlerde açıklama satırları eklenmiştir.

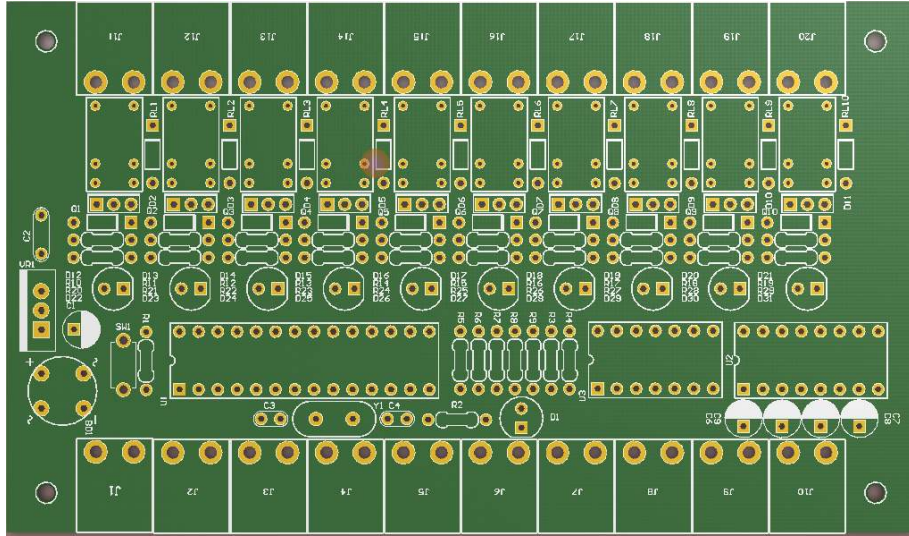


Şekil 4.1. Elektronik kart şeması

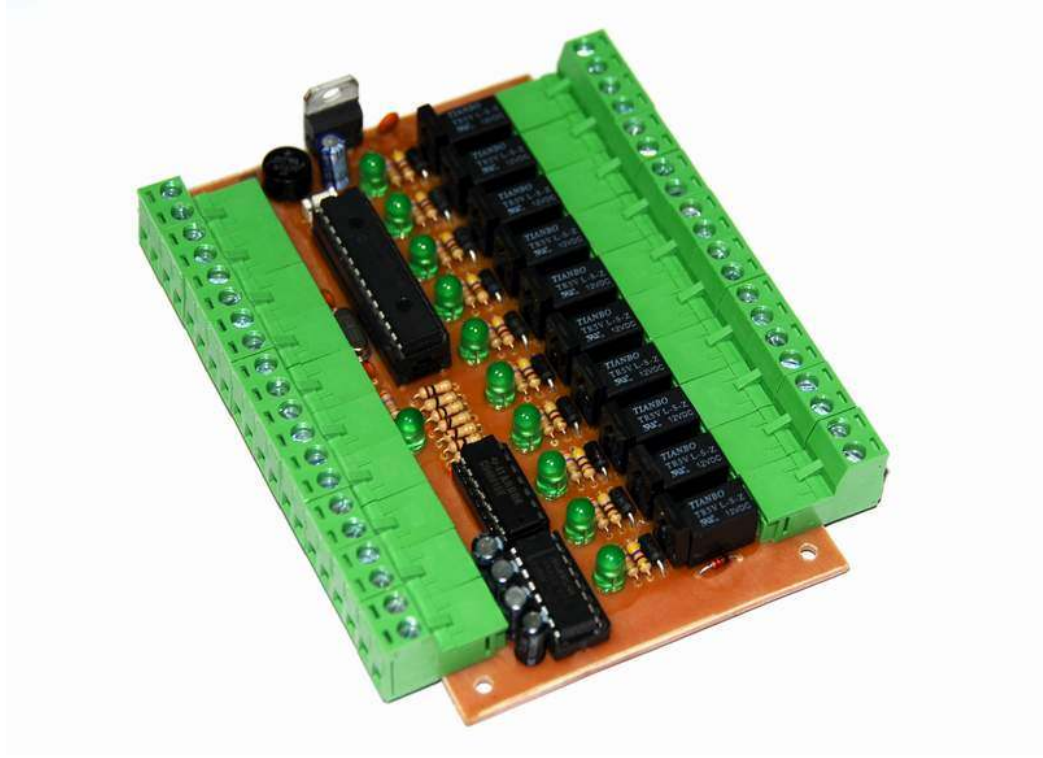
Kartın yapımında öncelikle; devrenin şemasının bakır plakete üzerine yerleştirilebilmesi için, baskı devresi tasarlanmıştır (Şekil 4.2). Baskı devresi tasarlanan kartın üstten görünen montajsız hali Şekil 4.3'te verilmiştir. Tasarım ütü metoduyla plakete aktarılmıştır. Tuz ruhu ve perhidrol karışımından oluşan asit yardımıyla, fazla bakır çıkartılarak baskı devre elde edilmiştir. Elektronik kartın montaj edilmiş önürünü Şekil 4.3'te görülmektedir.



Şekil 4.2. Elektronik kart baskı devre görünüşü



Şekil 4.3. Elektronik kart montajsız görünüşü



Şekil 4.4. Elektronik kart bitmiş önürünü

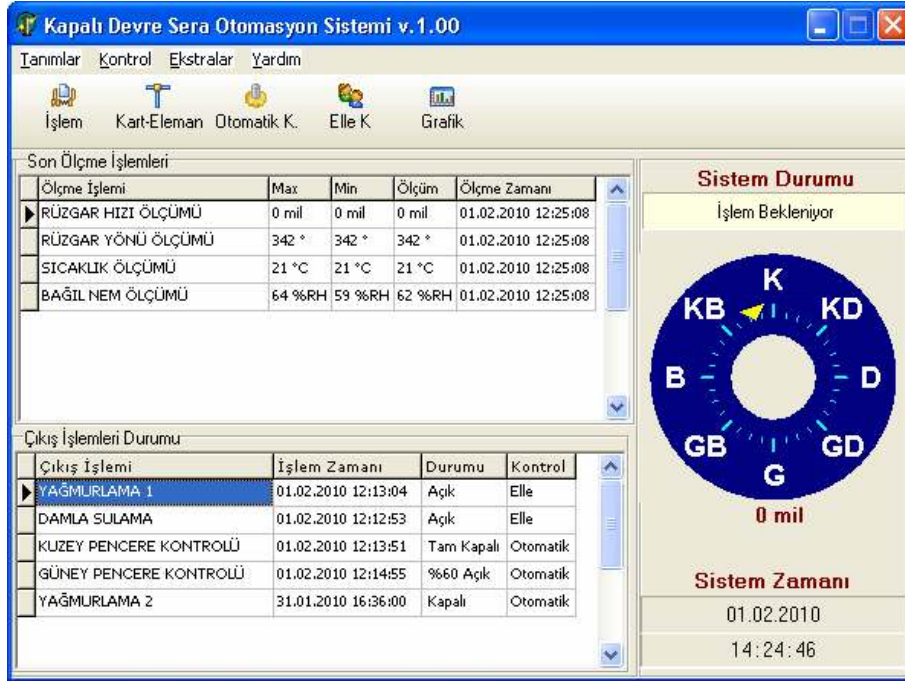
4.2. Otomasyon Programının Kullanımı

Bu bölümde, kapalı devre sera otomasyon sisteminin kontrol ve ayarlamalarının yapıldığı programın kullanımı anlatılmaktadır. Program çalıştırıldıktan sonra ilk açılış ekranı Şekil 4.5’de verilmiştir.

4.2.1. Açılış Ekranı (Ana Ekran)

Açılış ekranının en üstünde programın menüleri bulunmaktadır. Menülerin altında ise, bu menülere ait kısa yol tuşları yer almaktadır.

Açılış ekranının içeriğinde ölçme elemanlarına ait bazı bilgiler görülebilmektedir. Bunlar; ölçme işleminin adı, en düşük, en yüksek ve ortalama ölçme değerleri ve ölçme işleminin en son yapıldığı zaman bilgisi bulunmaktadır.



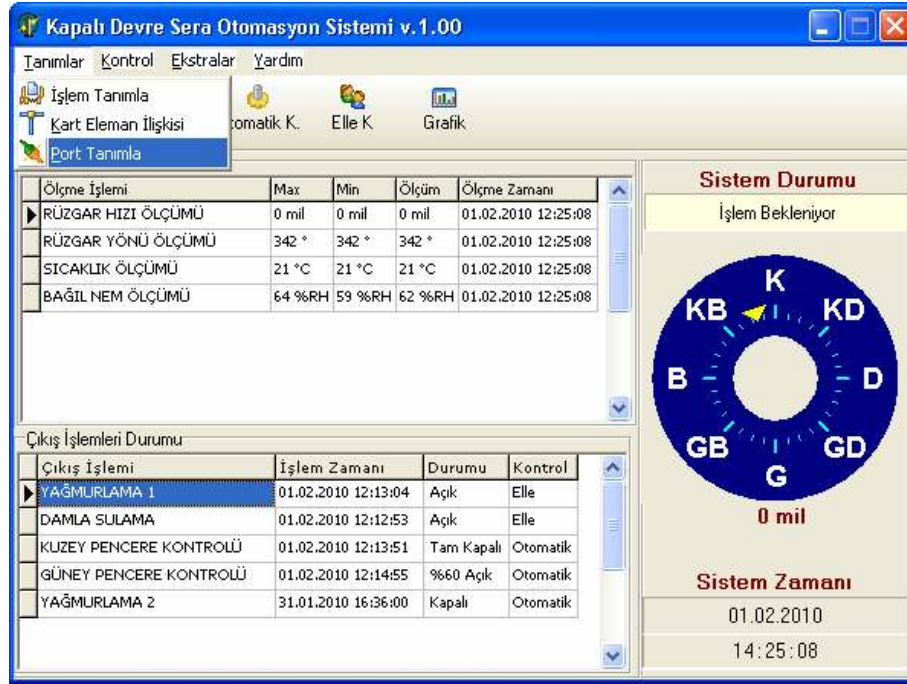
Şekil 4.5. Açılış ekranı

Açılış ekranının diğer bir içeriği de çıkış elemanlarının durum bilgileridir. Burada da ilgili çıkış elemanlarının hangi durumda olduğu ve bu duruma ne zaman getirildiği bilgisi görülebilmektedir.

Açılış ekranı içeriğinin sağ tarafında sistemin çalışma durumu, sistemin tarih ve saati, daha kolay anlaşılabilmesi için de rüzgâr yönü ve şiddeti ekstra olarak gösterilmektedir.

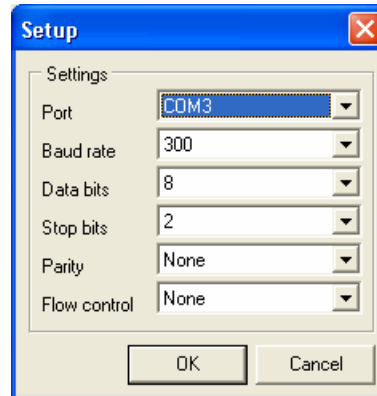
4.2.2. İletişim Portu Tanımlama Ekranı

Otomasyon sisteminde kullanılan elektronik kartlar ile bilgisayar arasındaki haberleşmede seri iletişim portu (COM) kullanılmıştır. Varsayılan ayarların dışında bir ayarlama yapılacaksa “Tanımlar” menüsünden “Port Tanımla” seçeneği (Şekil 4.6) seçilmelidir.



Şekil 4.6. Seri iletişim portu tanımlama

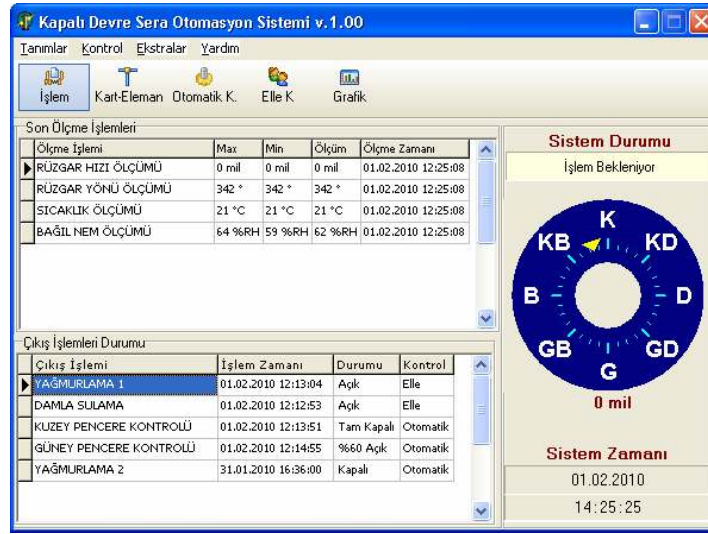
Seri iletişim portunun varsayılan ayarları Şekil 4.7’de görülmektedir. Bu ayarlar sistemin normalde çalışması gereken ayarlardır. Ancak bilgisayarlardaki port durumlarına göre değişkenlik gösterebilir.



Şekil 4.7. Seri iletişim portunun varsayılan ayarları

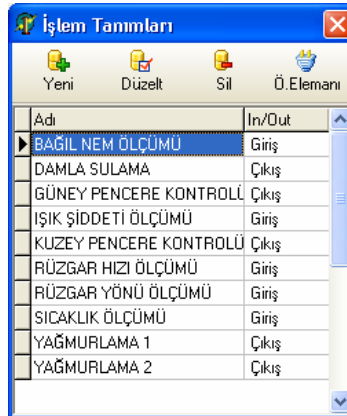
4.2.3. İşlem Tanımlama Ekranı

Sistemde var olması istenen ölçme işlemlerinin tanımlanabilmesi için açılış ekranındaki kısa yol tuşlarından “İşlem” (Şekil 4.8) ya da “Tanımlar” menüsünden “İşlem Tanımla” (Şekil 4.6) seçeneği kullanılabilir.



Şekil 4.8. İşlem tanımlama

İşlem tanımlama ekranında, tüm tanımlama ekranlarında olduğu gibi, “Yeni”, “Düzeltil”, “Sil” butonlarının yanı sıra sadece giriş elemanlarına ait olarak “Ö.Elemanı” (Ölçme Elamanı Tanımlama) butonu da bulunur (Şekil 4.9).



Şekil 4.9. İşlem tanımlama ekranı

4.2.4. Giriş-Çıkış İşlemi Ekleme/Düzeltilme Ekranı

Yeni bir işlem tanımlarken (Şekil 4.10) işlem adı, giriş (In) işlemi mi, çıkış (Out) işlemi mi olacağı, çıkış işlemi ise ne tür bir çıkış işlemi olacağı, giriş işlemi ise ölçülen değerin birimi ve kontrol aralıkları; dakika, gün ya da saat olarak belirlenmelidir. Bir işlem tanımlandıktan sonra; sistem içerisinde kullanılmak istenmediği ya da ihtiyaç kalmadığı durumlarda silmek yerine pasif yapılabilmesi için aktifliğinin de belirlenmesi gerekmektedir. “Kaydet” butonuyla yapılan değişiklik ya da tanımlar hafızada saklanır. “Vazgeç” butonu da yapılan işlemi iptal ederek bir önceki ekrana döner.

Şekil 4.10. İşlem ekleme/düzeltilme ekranı

Aşağıda bir giriş işleminin tanımlanmasına örnek verilmiştir. (Şekil 4.11). Burada ek olarak, ölçme birimi belirlenirken yardımcı olması açısından, gerektiğinde kullanılmak üzere karakter eşlem tablosu da birim alanının sağına yerleştirilmiştir. Bu butona basıldığında Şekil 4.12’de görülen karakter eşlem tablosu açılır. Buradan istenilen karakter seçilerek birim alanında kullanılabilir. Şekil 4.13’da da bir çıkış işlemi tanımlanmasına ait örnek görülmektedir.

İşlem Ekle/Düzeltil

Kaydet Vazgeç

İşlem Adı : RÜZGAR HIZI ÖLÇÜMÜ

In/Out : In Giriş

Birimi : mil

Kontrol Aralığı : 5 dk

Aktif : Evet

Şekil 4.11. Giriş işlemi ekleme/düzeltilme ekranı

Karakter Eşlem

Yazı Tipi: Times New Roman Yardım

Kopyalanan karakter: ° Seç Kopyala

☒ Gelişmiş Görünüm

U+00B0: Degree Sign Tuş vuruşu: Alt+0176

Şekil 4.12. Karakter eşlem tablosu

İşlem Ekle/Düzeltil

Kaydet Vazgeç

İşlem Adı : YAĞMURLAMA 1

In/Out : Out On/Off

Birimi :

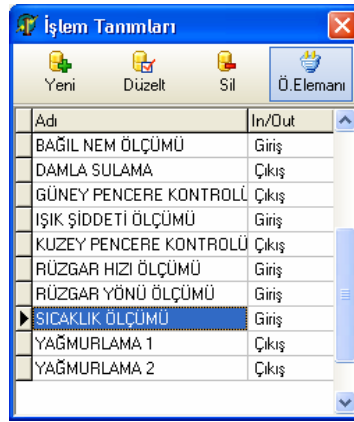
Kontrol Aralığı : 5 dk

Aktif : Evet

Şekil 4.13. Çıkış işlemi ekleme/düzeltilme ekranı

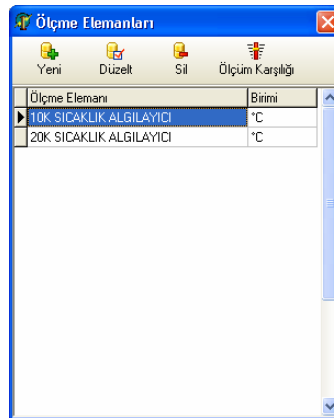
4.2.4.1. Ölçme Elemanı Tanımlama Ekranı

Bir giriş işleminin ölçümlerde kullanılabilmesi için; bu giriş işlemine ait ölçme eleman ya da elemanlarının belirlemesi gerekir. İşlem tanımlama ekranında ilgili giriş işlemi seçilip ‘Ö.Elemanı’ (Ölçme Elemanı) butonuna (Şekil 4.14) basıldığında, seçili giriş işlemi için kullanılabilecek ölçme elemanları tanımlama ekranı gelir (Şekil 4.15).



Şekil 4.14. Ölçme elemanı tanımlama

Burada da “Yeni”, “Düzeltil”, “Sil” butonlarına ek olarak ilgili elemanın kalibrasyon tablosunun oluşturulabileceği “Ölçüm Karşılığı” butonu bulunmaktadır.



Şekil 4.15. Ölçme elemanı tanımlama ekranı

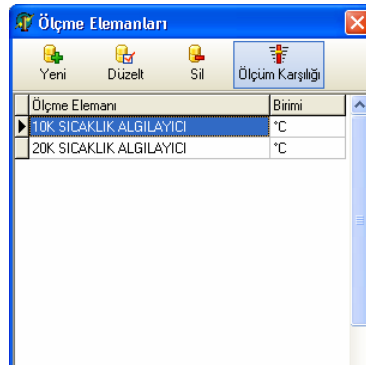
Ölçme elemanı eklenirken ya da düzenlenirken elemanı tanıttıcı bir isim, eleman ile ilgili açıklamalar ve de elemanın resmi istenildiğinde sisteme kayıt edilebilmektedir (Şekil 4.16).



Şekil 4.16. Ölçme elemanı ekleme/düzeltilme ekranı

4.2.4.2. Ölçme Elemanı Karşılık Tablosu Ekranı

Ölçme elemanı eklendikten sonra yapılacak iş; ölçme elemanı ile ilgili kalibrasyon tablosunun oluşturulmasıdır. Kalibrasyon tablosu oluşturulurken, ölçme elemanı tanımlama ekranında bulunan “Ölçüm Karşılığı” butonu kullanılarak (Şekil 4.17) karşılık tanımlama ekranına geçilir.



Şekil 4.17. Ölçüm karşılık tanımlama ekranına giriş

Bu ekranda seçilmiş ölçme elemanının adı, resmi ve elemana yapılmış olan açıklama otomatik olarak gelir. Burada bu elemanın alabileceği her bir değer için, kalibrasyon karşılıkları girilir (Şekil 4.18). Şekil 4.19’de karşılıkların eklenip düzenlenmesi görülmektedir.



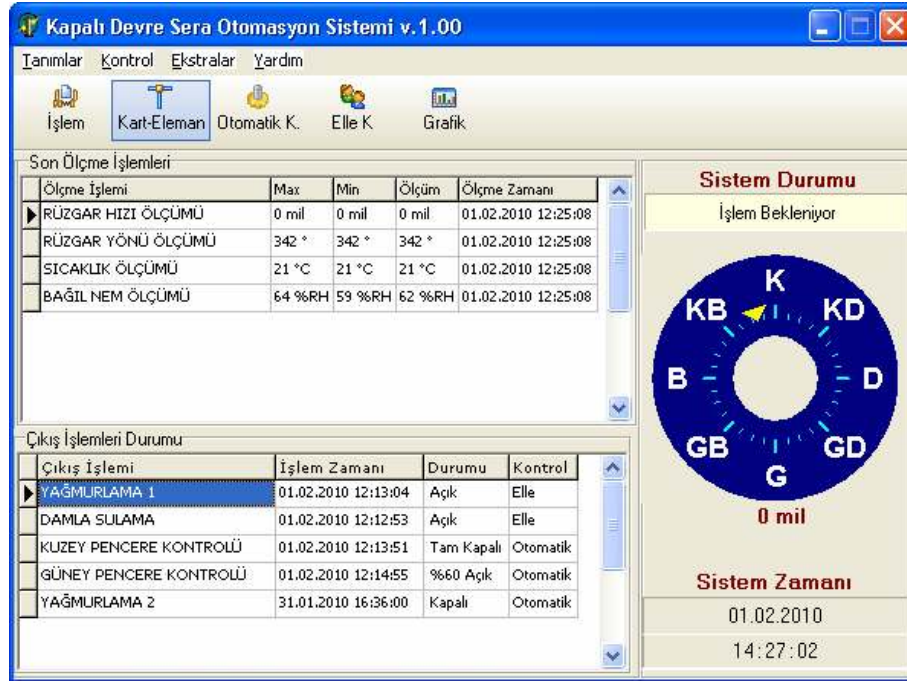
Şekil 4.18. Ölçme elemanı karşılık tanımlama ekranı



Şekil 4.19. Ölçme elemanı karşılık ekleme/düzeltilme ekranı

4.2.5. Kart-Eleman Tanımlama Ekranı

İşlem tanımlamaları yapıldıktan sonraki aşama, her bir elemanın kaç numaralı kart üzerinde takılı olduğunun sisteme tanıtılmasıdır. Bunun için açılış ekranında bulunan “Kart-Eleman” butonu (Şekil 4.20) seçilmeli ya da “Tanımlar” menüsünden “Kart-Eleman İlişkisi” seçeneği (Şekil 4.6) seçilmelidir.



Şekil 4.20. Kart-Eleman tanımlama ekranına giriş

Bu ekranda her bir elemanın kaç numaralı kartın kaç numaralı portuna ilişkilendirileceği belirlenir. Kartın üzerinde 4 adet analog giriş (1..4 portlar) 1 adet sayısal giriş (5 nolu port) bulunduğundan, analog girişlere istenilen analog elemanlar ilişkilendirilebilir. Tek sayısal giriş ise, rüzgâr hızı ölçümü için ayrılmış durumdadır. Kart üzerinde 10 adet de analog çıkış tasarlanmıştır (6..15 portlar).

Şekil 4.21. Kart-Eleman ekleme/düzeltilme ekranı

Örnek olarak; bir çıkış elemanı seçilmiş olsun. Bu çıkış elemanı da ON/OFF tipi bir çıkış kontrol elemanı olsun. Bununla ilgili olarak ayarlarının nasıl yapılacağı Şekil 4.22’de gösterilmiştir.

The screenshot shows a software window titled 'Kart-Eleman Ekle/Düzeltil'. It has a blue title bar with a close button. Below the title bar, there are two buttons: 'Kaydet' (Save) and 'Vazgeç' (Cancel). The main area contains three dropdown menus: 'Kart No' (Card No.) with value '2', 'Port No' (Port No.) with value '7', and 'İşlem' (Operation) with value 'DAMLA SULAMA'. To the right of these is a text box labeled 'Açıklama:' (Description:). The window has a standard Windows XP-style interface.

Şekil 4.22. ON/OFF tipi çıkış elemanı kart ilişkisi

Eğer kademeli bir çıkış elemanı seçilmiş ise, bu eleman için gerekli olan kart ve port tanımı ile bu elemanın maksimum kapanması ya da açılması için gerekli süre saniye cinsinden belirtilmelidir (Şekil 4.23).

The screenshot shows the same 'Kart-Eleman Ekle/Düzeltil' window. In this configuration, the 'İşlem' (Operation) dropdown is set to 'KUZEY PENCERE KONTROL'. Below the main settings, there is a section for 'Max. Açılma veya Kapanma Süresi:' (Maximum Opening or Closing Time:) with a value of '30 sn'. Below this, there are additional dropdowns for 'Kart No' (Card No.) set to '1' and 'Port No' (Port No.) set to '8'. The window has a standard Windows XP-style interface.

Şekil 4.23. Kademeli tip çıkış elemanı kart ilişkisi

Eğer işlem bir giriş işlemi ise; kart ve port tanımlarına ek olarak girişte kullanılan ölçme elemanının da belirlenmesi gerekmektedir (Şekil 4.24). Ölçme

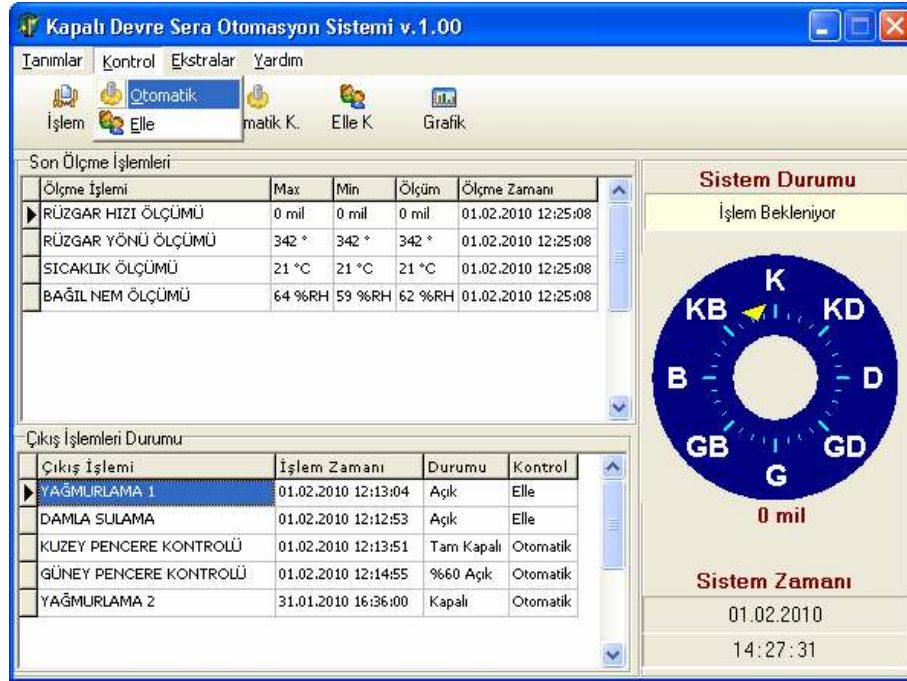
elemanı seçildiğinde seçilen elemana ait açıklamalar ve elemanın resmi otomatik olarak gelmektedir.

Şekil 4.24. Giriş elemanı kart ilişkisi

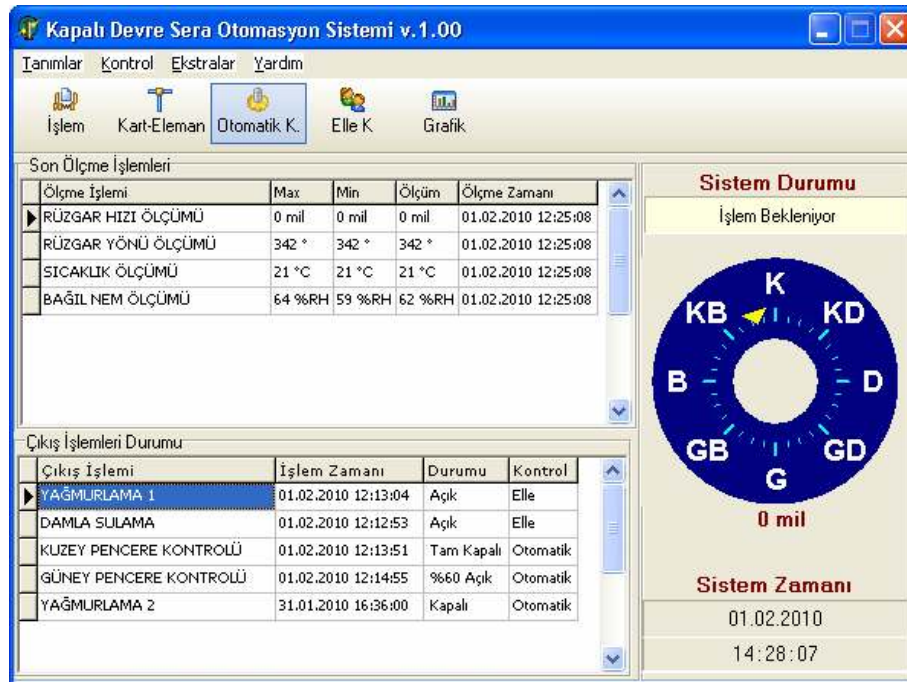
4.2.6. Kontrol İşlemleri

Eleman tanımlamalarından sonraki aşama ise; seranın ve bu sera içerisindeki ortama göre kontrol elemanlarının koordinasyonunun belirlenmesidir. Bunun için açılış ekranında “Kontrol” menüsü (Şekil 4.25) kullanılır. Kumanda elemanlarını sistemimizde; otomatik ve elle olmak üzere iki şekilde kontrol etmek mümkündür. Eğer, kumanda edilmek istenilen eleman bilgisayar tarafından alınan ortam şartlarına göre kontrol edilsin isteniliyorsa, açılış ekranında bulunan “Otomatik K.” (Otomatik Kontrol) butonu (Şekil 4.26) kullanılmalı ya da “Kontrol” menüsünden “Otomatik” (Şekil 4.25) seçeneği seçilmelidir.

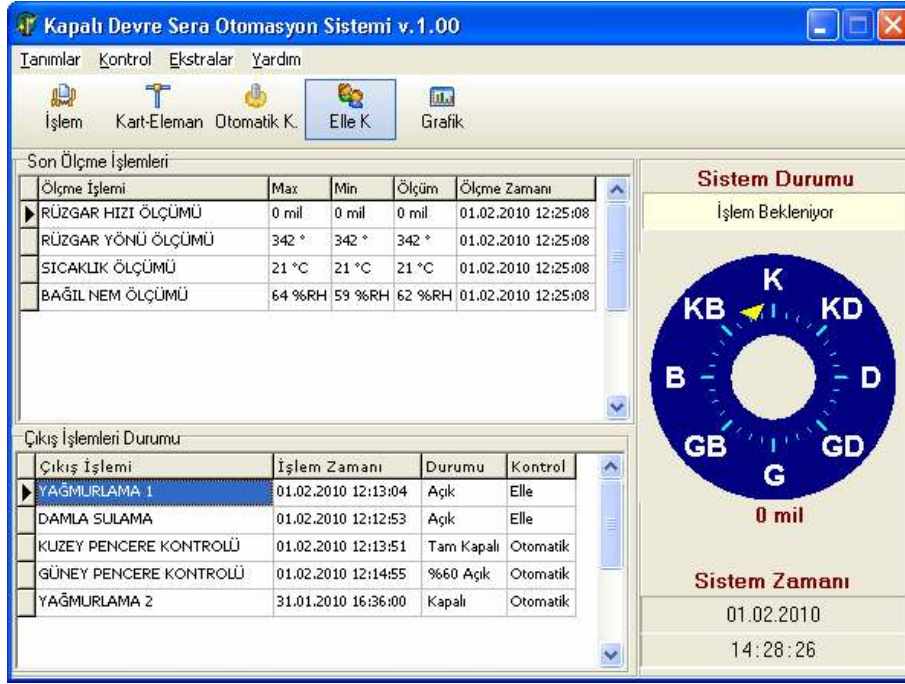
Kumanda işlemi elle kontrol edilmek isteniyorsa, açılış ekranında bulunan “Elle K.” (Elle Kontrol) butonu (Şekil 4.28) kullanılmalı ya da “Kontrol” menüsünden “Elle” (Şekil 4.25) seçeneği seçilmelidir.



Şekil 4.25. Çıkış kontrol işlemleri



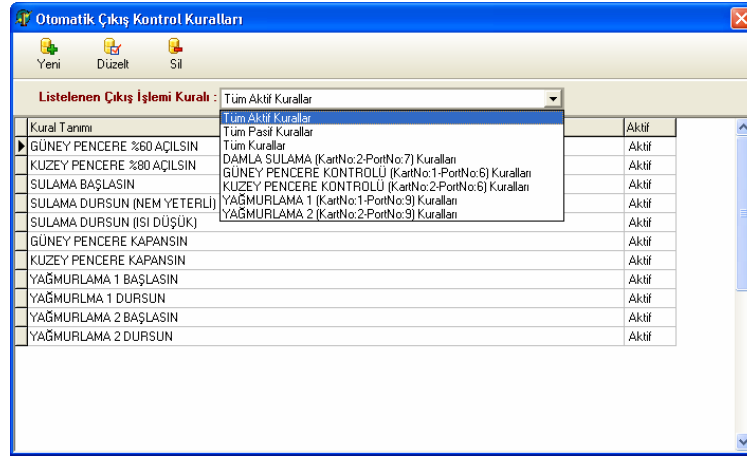
Şekil 4.26. Otomatik kontrol işlemine giriş



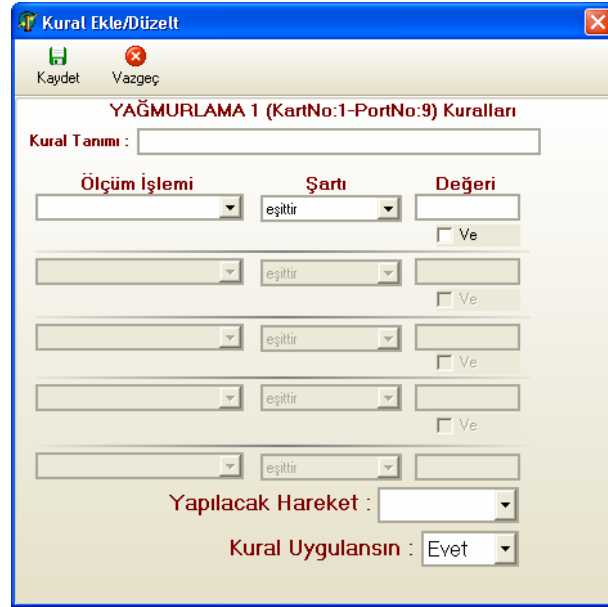
Şekil 4.27. Elle kontrol işlemine giriş

4.2.6.1. Otomatik Kontrol Ekranı

Otomatik çıkış kontrol işlemleri ekranının içeriğinde, tüm aktif kuralların bir listesi görülür. Ayrıca, tüm tanımlama ekranlarımızda olduğu gibi, otomatik kontrol ekranında da “Yeni”, “Düzeltil”, “Sil” butonları bulunmaktadır. Ekleme ve düzeltme işlemlerinin yapılabilmesi için, “Listelenen Çıkış İşlemi Kuralı” olarak, belirtilmiş listeden ilgili çıkış işlemi seçilmelidir (Şekil 4.28).



Şekil 4.28. Otomatik çıkış kontrol işlemleri



Şekil 4.29. Kural ekleme/düzeltilme ekranı

Kural ekleme/düzeltilme ekranında, eklenecek kuralın tanımı ile ölçme işleminin (Şekil 4.30) değerinin herhangi bir şarta (Şekil 4.31) bağlı olup olmayacağı belirlenebilir. Bu işlem ve şartlar, 1 ile 5 tane aynı işlem ya da ayrı işlem olmak üzere, duruma göre (örneğin, sıcaklık ölçümü 10 °C ile 20 °C arasında ve hava bağıl nemi %50 RH ise gibi.) bir kural yazılabilmektedir. Bunun için bir şart belirlendikten sonra, eğer devamına bir şart daha yazılmak isteniliyorsa, o şartın altında bulunan “Ve” işaret kutusu işaretlenmelidir. İşaretlenen her “Ve” kutusu ile altındaki şart

yazma kutuları aktif hale gelecektir (Şekil 4.32). Ayrıca bu ekranda, çıkış elemanına yaptırılmak istenilen hareket ile oluşturulmuş bu kuralın uygulanıp uygulanmayacağı belirlenebilmektedir. Şekil 4.33’de ON/OFF tipi çıkış işlemine ait seçilebilecek ve Şekil 4.34’de seçilmiş hareket görülmektedir. Aynı şekilde Şekil 4.35’de kademeli tip çıkış elemanına ait seçilebilecek ve Şekil 4.36’da seçilmiş hareket görülmektedir.

Şekil 4.30. Kural ekleme/düzeltilme ekranında ölçüm işlemi seçimi

Şekil 4.31. Kural ekleme/düzeltilme ekranında şart seçimi

Şekil 4.32. Kural ekleme/düzeltilme birden fazla kıstas belirleme

Şekil 4.33. Kural ekleme/düzeltilme ON/OFF tipi işlem hareket seçenekleri

Şekil 4.34. Kural ekleme/düzeltilme ON/OFF tipi işlem seçilmiş hareket

Şekil 4.35. Kural ekleme/düzeltilme kademeli işlem tipi hareket seçenekleri

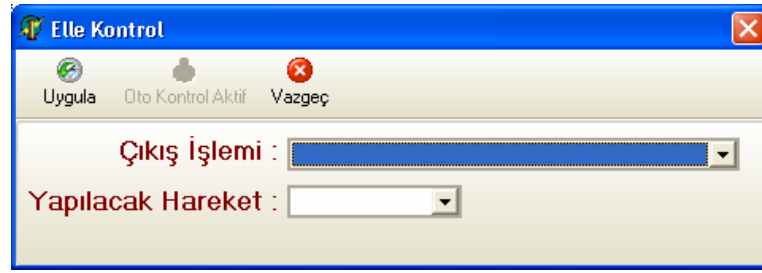
Şekil 4.36. Kural ekleme/düzeltilme kademeli işlem tipi seçilmiş hareket

4.2.6.2. Elle Kontrol Ekranı

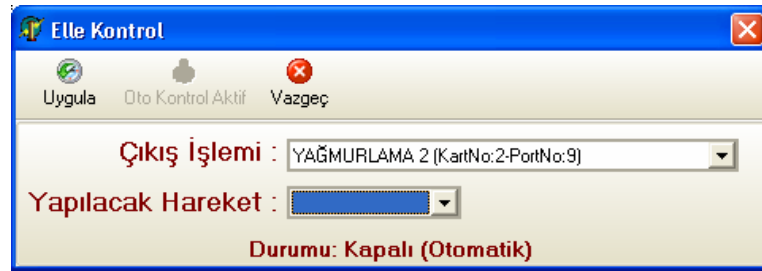
Elle çıkış kontrol işlemleri ekranının içeriğinde, çıkış işleminin seçilebileceği seçim listesi ile yapılması istenilen hareketin seçilebileceği seçim listesi bulunmaktadır (Şekil 4.37). Bu seçimler yapıldıktan sonra, istenilen hareketin yapılabilmesi için; “Uygula” butonu kullanılmalıdır. Seçilen işlem ve hareket uygulandıktan sonra “Oto Kontrol Aktif” butonu pasif durumdan aktif duruma gelir.

(İşlemin yeniden otomatik kontrole geçirilebilmesi için) İlgili çıkış işleminin kontrolü yeniden otomatik olarak yaptırılmak istenildiğinde “Oto Kontrol Aktif” butonu kullanılmalıdır. Bu buton kullanılmadığı takdirde, ilgili çıkış işlemi hiçbir şekilde otomatik olarak kontrol ettirilemeyecektir.

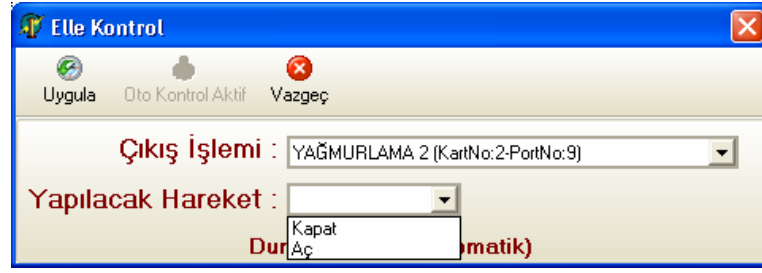
Herhangi bir çıkış işlemi seçildiğinde, yapılacak hareket listesi işlemin ON/OFF (Şekil 4.38) veya kademeli tip (Şekil 4.40) çıkış işlemi olmasına göre değişecektir. (Şekil 4.39. ve 4.41). Ayrıca çıkış işlemi seçildiğinde içeriğin en altında seçilmiş işlemin durumu ve bu duruma ne şekilde (Otomatik, elle) geldiği bilgisi görülecektir.



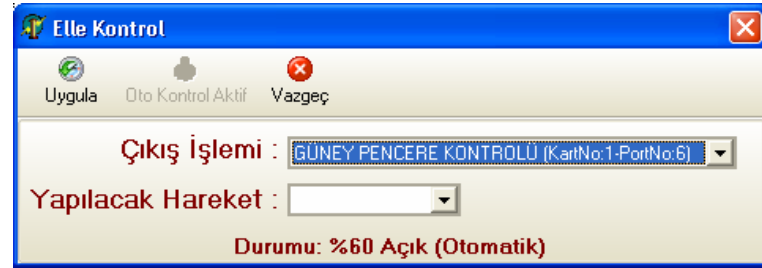
Şekil 4.37. Elle kontrol ekranı



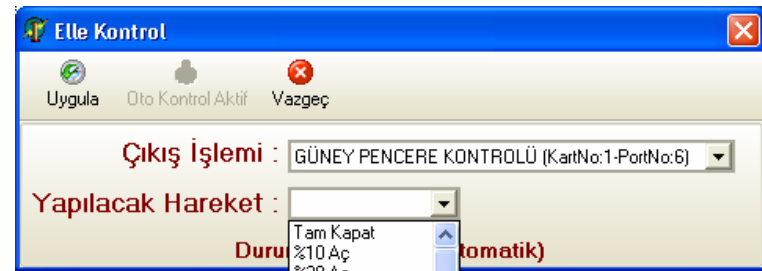
Şekil 4.38. ON/OFF tipi çıkış işlemi seçilmesi



Şekil 4.39. ON/OFF tipi çıkış işlemi hareket seçenekleri



Şekil 4.40. Kademeli tip çıkış işlemi seçilmesi



Şekil 4.41. Kademeli tip çıkış işlemi hareket seçenekleri

4.2.7. Grafik İzleme Ekranı

Sistemdeki iklimsel şartların zamana göre değişimlerinin grafiksel olarak izlenebileceği ekrana giriş yapabilmek için; açılış ekranında bulunan “Grafik” butonu (Şekil 4.42) ya da “Ekstralar” menüsünden “Grafik” seçeneği seçilmelidir.

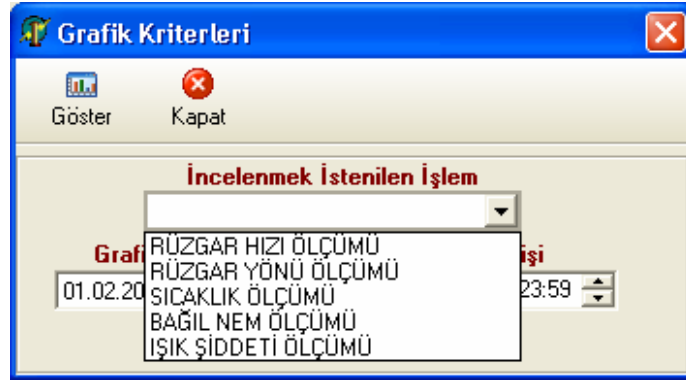


Şekil 4.42. Grafik izleme ekranına giriş

Grafik izleme ekranına giriş yapıldıktan sonra, karşımıza grafik izleme kriterlerinizi belirleyebileceğimiz ekran gelmektedir (Şekil 4.43).

Şekil 4.43. Grafik kriterleri belirleme ekranı

Grafik kriterlerinde; ilk önce incelenecek giriş işlemi seçilir (Şekil 4.44). Daha sonra işlemin inceleneceği tarih ve/veya saat (Şekil 4.45) aralığı belirlenir. Grafiğin çizdirilebilmesi için “Göster” butonuna (Şekil 4.46) basılır.



Şekil 4.44. Grafik kriterleri işlem seçimi



Şekil 4.45. Grafik kriterleri başlangıç ve bitiş tarihi belirleme



Şekil 4.46. Grafik çizmeye başlama

İstenilen kriterlere göre belirlenmiş grafik incelenmek üzere ekrana gelir (Şekil 4.47).



Şekil 4.47. İncelenecek grafiksel gösterim

4.2.8. Program Kapatma ve Hakkımızda Ekranı

Program kapatılırken karşımıza her defasında programın kapatılıp kapatılmayacağına dair bir sorgu ekranı gelir (Şekil 4.48). Burada programın yazarı ve iletişim bilgisi de yer almaktadır. Eğer program kapatılmak isteniyorsa, “Kapat” butonu, programın çalışmasına devam edilmek isteniyorsa “Devam Et” butonu kullanılmalıdır.

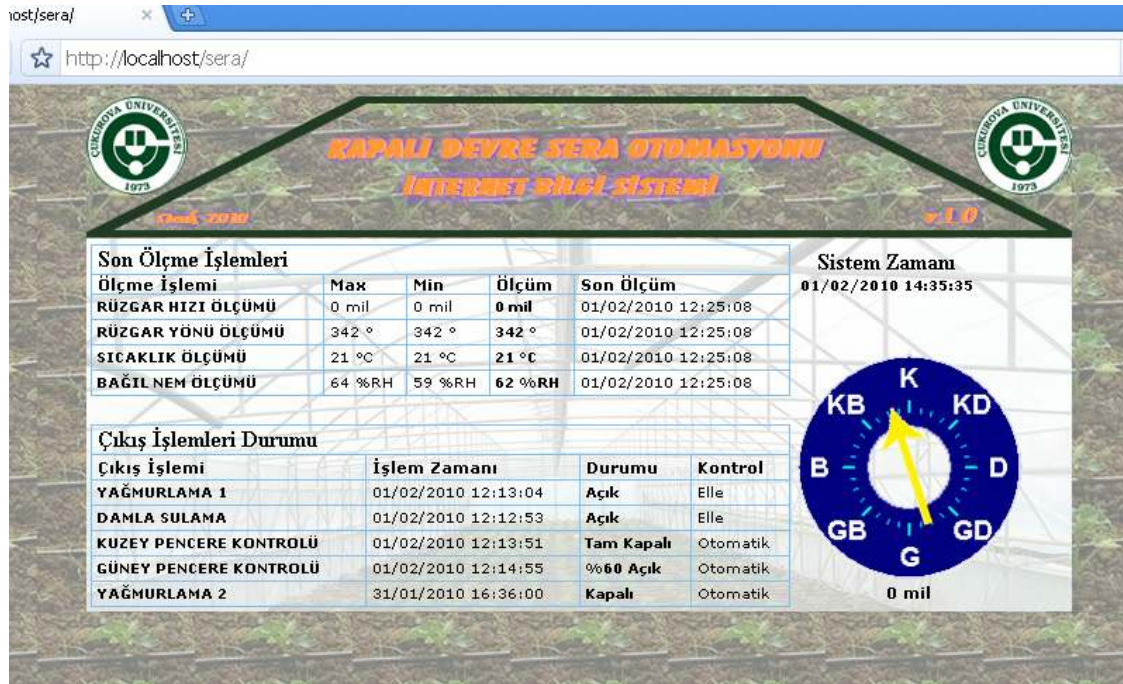


Şekil 4.48. Kapatma onay ve program bilgisi ekranı

4.3. İnternet Bilgi sistemi

Otomasyon programı, algılayıcılardan gelen bilgiler ile kumanda sistemlerini harekete geçirir. Otomasyon programı üzerinden yapılan tüm işlemler anlık olarak veri tabanında saklanır. Veri tabanında saklanan bu veriler; aynı anda internette

yayınlamak üzere internet bilgi sistemine aktarılır. İnternet bilgi sistemine belirlenecek bir web adresi ya da kontrol bilgisayarının IP adresini yazarak ulaşmak mümkündür. Sisteme ulaşan kullanıcıların bilgilere erişebilmesi için, kullanıcı adı ve şifrelerinin olması gereklidir. İnternet bilgi sistemine her internet kullanıcısı erişim yapabilir. En son ölçülmüş olan iklim verileri ile kumanda sistemlerinin en son durumları görülebilir (Şekil 4.49).



Şekil 4.49. İnternet bilgi sistem

5. SONUÇLAR ve ÖNERİLER

Otomasyon sistemlerinin çok yüksek maliyetli olması ve kullanımındaki teknik eleman ihtiyacı, tarımsal alanda otomasyon kullanılması önündeki en büyük engelleri oluşturmaktadır. Geliştirilen otomasyon sistemi, hem düşük maliyeti, hem de kullanıcı dostu yazılımı ile tarım alanında ekonomik kurulum yapılarak uzun vadede yapılabilecek yenilemelere uyumlu olabilecektir. Kullanıcı dostu yazılımı sayesinde de branş dışında, herhangi bir teknik elemana ihtiyaç duyulmaması sağlanmıştır. Ürün için gerekli iklim şartlarının verileri tanıtıldığında, sistem ürüne uygun otomasyon işlemlerini gerçekleştirecektir.

Yapılan çalışmalar göstermiştir ki; seralar için sistem tasarlamak, öncelikle seracılık hakkında yeterli bilgi sahibi olmak gerekliliğini ortaya koymuştur. Bu aşamada halen var olan sistemlerin incelenmesi de sistemin nelerden oluşması gerektiği ve gelecekte sisteme nelerin eklenebileceği konularında fikir vermesi açısından önemlidir. Ancak; bu konuda bazı kuruluşların kurulu sistemlerini araştırmacılara incelettirmek istemedikleri gerçeği ile karşı karşıya kalınmıştır.

Geliştirilen elektronik kartın tasarım aşamasında tek başına elektronik devre elemanlarının ideal ve tüm sistemlere uyumlu olamayacağı anlaşılmıştır. Tüm sistemlerle uyumlu olabilmesi için, kartın gelen verileri, doğru bir şekilde yönlendirmesi, beklemesi ve zamanında işlemleri yerine getirmesi gerekmektedir.

Tasarım aşamasında sistemin ucuz, çok bulunan, arıza halinde tamiri mümkün ve kolay olacak şekilde tasarlanmasına çalışılmış; ancak burada programlanabilen parça kullanılmadığında sistemin karmaşık ve sıkıntılı olacağı görülmüştür. Bu sebeple tasarlanan kartta bir mikrodenetleyici kullanılarak, sistemin kalan kısmının hem efektif hem de çok bulunan ve ucuz malzemelerden oluşması sağlanmıştır.

Sistem tasarımıdaki en büyük sıkıntı ise; algılayıcıların aldıkları bilgileri istenilen büyüklüğe dönüştürürken yaşanan kalibrasyon problemleridir. Bu problemin temeli ise, elektronik malzeme satıcılarının sattıkları parçalarda ucuzluk konusunu ön plana çıkarmaları sebebiyle, standart dışı ya da katalogu mevcut

olmayan malzeme satmalarıdır. Katalog değerleri olan malzemeler için bu işlem sıkıntı olmaktan çıkacaktır.

Otomasyon sisteminin bilgi deposu olarak kullanılabilecek veri tabanı ile ilişkisi, sera ile ilgili tüm bilgilerin toplanabileceğini göstermiştir. İyi tasarlanmış bir veri tabanı ile geçmişe yönelik çalışmalar yapılarak, gelecekte yapılması gerekenler ve alınması gereken kararlar, çok daha kolaylıkla verilebilecektir.

Veri tabanında biriken bilgiler ile çok daha farklı uygulamalar tasarlayabilmek mümkün olduğundan, bu uygulamaların ne şekilde olabileceği seraların imkânları ölçüsünde sistemin hayata geçirilmesini mümkün kılmaktadır. İstenilen bilgilere hızlı ve zamanında erişilebilmesini otomasyonun ayrılmaz bir parçası olarak görmekteyiz.

Sistemin tasarlanmasında günümüz teknolojilerinden en iyi şekilde faydalanmak suretiyle, işlerin daha hızlı, doğru ve verimli hale getireceği düşünülmüştür. Bu sebeple sistemin internet bağlantısı için, seranın coğrafi şartlarının bir sıkıntı oluşturmaması amacıyla kablolu yerine kablosuz ya da uydu bağlantılı erişimler kullanılabilir. Ayrıca sisteme internet üzerinden elle kontrol da eklenerek seraya gelmeden bazı kontrollerin uzaktan yapılması sağlanabilir. Yine seraya uygun noktalara kameralar yerleştirilerek hem bitkilerin gelişiminin ve sera içi şartlarının izlenmesi, hem de güvenliğinin sağlanması konusunda yardımcı olacaktır. Bir diğer husus ise; bitkilerin manyetik alandan etkilenmeleri ile ilgili araştırmalardan faydalanarak, eğer sorun çıkmayacaksa, algılayıcılar, kontrol kartları, kameralar gibi yan aparatların tamamı kablosuz çalışır hale getirilerek, sistem kurulumu çok pratik hale dönüştürülebilir.

KAYNAKLAR

- BAŞÇETİNÇELİK, A., ÖZTÜRK H.H. 1997. Seralarda Ortam Kontrolü, 2. Seracılık Sempozyumu, Simav, Kütahya.
- BAŞÇETİNÇELİK, A., ÖZTÜRK, H. H., 1998. Kontrollü Ortamlarda Yapılan Bitki Denemeleri İçin İklim Etmenlerinin Ölçülmesi. 2. Sebze Tarımı Sempozyumu 28-30 Eylül 1998 Bildiri Kitabı, Gaziosmanpaşa Ü. Ziraat Fakültesi Bahçe Bitkileri Bölümü, S: 349-354, Tokat.
- DAŞDEMİR Y., 2002, Veritabanları ve SQL, Delphi ile Veritabanı Uygulamaları Geliştirme, Türkmen Kitabevi, İstanbul
- FUNT, R, 1989, Management Decision Making Software, Marked, Model International Society For Horticultural Science Apr., 51-56
- GILREATH, J., 1985, Description of a Basic Program For Data Collection Using a Portable Microcomputer, American Society For Horticultural Science Apr, 301
- İNAN, S.A., 2002, Meyve Fidanı Çoğaltılmasında Kullanılan Köklendirme Seralarının Otomasyonu, Yüksek Lisans Tezi, SDÜ, Bahçe Bitkileri Anabilim Dalı, Isparta.
- KAÇIRA, M., SHORT, T. H., STOWELL, R.R., 1997. Modeling Natural Ventilation Designs for Greenhouses in Mediterranean Climates. Presented at the ISHS Symposium, Antalya, Turkey.
- KARAGÜLLE İ., PALA Z., 2001, Borland Delphi 5, Türkmen Kitabevi, İstanbul.
- KÜRKLÜ A, ÇAĞLAYAN N. 2005. Sera Otomasyon Sistemlerinin Geliştirilmesine Yönelik Bir Çalışma, Akdeniz Üniversitesi Ziraat Fakültesi Dergisi, 2005, 18(1), 25-34, Antalya
- HAKGÖREN F., KÜRKLÜ A. 2007. Sera Planlaması, Akdeniz Üniversitesi Ziraat Fakültesi, S: 89, Antalya
- MICROCHIP, 2001, PIC16F87X Data Sheet 28/40-Pin 8-Bit CMOS FLASH Microcontrollers
- OKURSOY, R. 1993. Design of a Microprocessor Based Temperature Control System. Doğa-Tr. Journal of Agricultural and Forestry. Sayı: 17. Sayfa : 977-985. Tübitak, Ankara.

- PHP DOCUMENTATION GROUP, 2009, PHP Klavuzu
- SUN MICROSYSTEMS, 2008, MySQL 5.1 Referans Klavuzu
- TİTİZ, S., 2004. Modern Seracılık Yatırımcıya Yol Haritası, Antalya Sanayici ve İşadamları Derneği (ANSİAD) yayınları, Antalya, S:21
- TUNCAY, Ö, 1990, Sera Sebze Yetiştiriciliğinde Karşılaşılan Bazı Sorunların Çözümüne Yönelik Bir Bilgisayar Programı, Yüksek Lisans Tezi, E.Ü, Bahçe Bitkileri Anabilim Dalı, Bornova
- YELKEN, A.A., 2003, Seralarda İklim Kontrolüne Yönelik Bir Bilgisayar Programı, Yüksek Lisans Tezi, ÇÜ, Tarım Makinaları Anabilim Dalı, Adana
- YULE, D., 1989, Temel Bilim Ansiklopedisi, Milliyet, İstanbul, 313, 314

ÖZGEÇMİŞ

1977 yılında Adana’da doğdu. İlkokul, Ortaokul ve Lise öğrenimini Adana’da tamamladı. 1996 yılında, Mersin Üniversitesi, Erdemli Meslek Yüksek okuluna başladı, 1998 yılında mezun oldu. Aynı yıl Mersin Üniversitesi, Bilgisayar Mühendisliği Bölümüne dikey geçiş yoluyla kabul edildi. Mühendislik fakültesinden 2001 yılında mezun oldu. Mezun olduğu yıl Mersin Üniversitesinde Uzman kadrosunda göreve başladı. Halen Mersin Üniversitesinde Mühendis kadrosunda çalışmaktadır.

2003 yılında Çukurova Üniversitesi Fen Bilimleri Enstitüsü Tarım Makinaları Anabilim Dalı’nda yüksek lisans eğitime başladı. Ailevi nedenlerden dolayı tez aşamasında bırakmak zorunda kaldı.

Yüksek lisans eğitimini tamamlamak amacıyla 2009 yılında af kapsamından yeniden başladı.

EK 1. PIC PROGRAMININ KODLARI

```
; Sera kontrol kartı programı. 300 baud da seri haberleşme yapar.  
; Veri 8 bitlik data şeklinde olacak.  
; bit7:0 off ya da giriş, 1:on  
; bit 654: kartno  
; bit 3210: eleman(port) no  
; port no da: 1234 analog giriş, 5 giriş yelolçer, 6-15 çıkış  
; yel olcme altprogramı aktif
```

```
list      p=16f873
```

```
#include      "P16F873.INC"
```

```
; Değişkenler
```

```
cblock      0x20
```

```
geldat1,geldat11,analog,eee,zd1,zd2,zd3,zd4,kartno,say,sure,bitti,datx;
```

```
endc
```

```
; Başlangıç vektörleri tarifi.
```

```
org         0h
```

```
goto       start
```

```
org         4h
```

```
goto       saydir
```

```
; Ana Program başlangıç ayarları kısmı
```

```
start      BCF      STATUS, RP0    ;  
           BCF      STATUS, RP1    ; Bank0  
           clrw      ; Clear W.  
           movwf     PORTA ; PORTA zero  
           movwf     PORTB ; PORTB zero  
           movwf     PORTC ; PORTC zero .
```

```
; Port Setup kısmı:
```

```
bcf        STATUS,RP0    ; Select Bank 0  
bsf        RCSTA,SPEN    ; Enable USART.  
bsf        RCSTA,CREN    ; Enable Receive
```

```
bsf        STATUS,RP0    ; Select Bank 1  
movlw      0x06  
movwf      ADCON1
```

```
movlw      0x3F  
movwf      TRISA          ; porta giriş  
CLRF       TRISB          ; portb çıkış
```

```
;seri iletişim ayarları kısmı:
```

```
movlw      0xCF            ; BRG değeri 300 baud @ 4MHz  
movwf      SPBRG          ; Write it to register.  
movlw      0xA0            ; CSRC/TXEN (Internal clock, 8 bit mode, Async operation, High
```

```
Speed)
```

```
movwf      TXSTA          ; Write to TX control register.
```

```

        bcf     STATUS,RP0    ; Select Bank 0
        bcf     STATUS,RP1    ; Select Bank 0
        call    swait

;adc ayarları kısmı:
        bcf     STATUS,RP1    ;
        bsf     STATUS,RP0    ; Select Bank 1

        movlw   0xFF
        movwf   TRISA
        movlw   0x0
        movwf   TRISA
        bcf     STATUS,RP0
        bsf     PORTA,0x0
        bcf     PORTA,0x0
        bsf     PORTA,0x0
        bcf     PORTA,0x0

        bsf     STATUS,RP0
        movlw   0xFF
        movwf   TRISA

        movlw   0xF0
        movwf   TRISC

        movlw   b'00000111'    ;dahili sinyal 1/2 oranı
        movwf   OPTION_REG    ; yelolcer zaman için timer0 ayar

        movlw   0x02
        movwf   ADCON1
        bcf     STATUS,RP0
        bcf     STATUS,RP1    ; Select Bank 0

        movlw   0x01
        movwf   datx

BASLA  movlw   10h            ;Kart no 1
        movwf   kartno
        CLRF    PORTB
        BSF     PORTC,2        ; test için kart çalışıyor anlamında

        bcf     PIR1,RCIF      ; Clear the interrupt flag

BAS1   call     getc           ;gelen datayı oku
        movwf   geldat1
        call    swait

        movf    geldat1,w
        andlw   h'70'
        subwf   kartno,w
        btfss   STATUS,2
        goto    BAS1

        bsf     PORTB,5        ;test için data alındı mı

```

```

        movf    geldat1,w
        andlw   0Fh
        movwf   geldat11

intest   nop                                ;giriş elemanları değeri mi

test1    movf    geldat11,w
        sublw   0x1
        btfss   STATUS,2
        goto    test2
        call    do_conv1
        goto    gonder

test2    movf    geldat11,w
        sublw   0x2
        btfss   STATUS,2
        goto    test3
        call    do_conv2
        goto    gonder

test3    movf    geldat11,w
        sublw   0x3
        btfss   STATUS,2
        goto    test4
        call    do_conv3
        goto    gonder

test4    movf    geldat11,w
        sublw   0x4
        btfss   STATUS,2
        goto    testyel
        call    do_conv5
        goto    gonder

testyel  movf    geldat11,w
        sublw   0x5
        btfss   STATUS,2
        goto    out1
        goto    yeloku

;*****RUZGAR HIZI OLCME KISMI*****

yeloku   clrf    bitti
        ;MOVLW    0XFE
        ;MOVWF    TMR0
        clrf    TMR0                        ;Timer 0 başlangıç sayısı 00
        movlw   b'10100000'                ;interruptlar açık, Timer 0 interrupt aktif
        ;Timer 0 zaman aşımı 0

        movwf   INTCON
        clrf    analog
        movlw   .7
        movwf   sure

clkkon   btfss   PORTC,5

```

```

        goto    clkkon1
        goto    clkkon2
clkkon1 btfss   bitti,1
        goto    clkkon
        goto    yelbitti
clkkon2 call    bekle
        btfss   PORTC,5
        goto    clkkon

brr      btfsc   PORTC,5
        goto    brr1
        goto    brr2
brr1     btfss   bitti,1
        goto    brr
        goto    yelbitti
brr2     call    bekle
        btfsc   PORTC,5
        goto    brr
        incf    analog,f
        btfss   bitti,1
        goto    clkkon
        goto    yelbitti

yelbitti bcf     INTCON,7
        goto    gonder

saydir   bcf     INTCON,T0IF ;Timer 0 zaman aşımını tekrar 0 yap
                                ;değilse tekrar int alt pr. çalışır
        rlf     datx,f
        movwf   PORTB
        decfsz  sure,f
        goto    saydev2
        goto    saydev1
saydev2  ;MOVLW    0XFE
        ;MOVWF    TMR0
        clrf    TMR0
        retfie

saydev1  clrf    TMR0
        bsf     bitti,1
        clrf    PORTB
        retfie

;*****ADC OLCME KISMI*****

do_conv1 movlw    0xC1
        movwf   ADCON0
        goto    adc_olc

do_conv2      movlw  0xC9
        movwf   ADCON0
        goto    adc_olc

do_conv3      movlw  0xD1

```

```

        movwf ADCON0
        goto   adc_olc

;do_conv4      BSF    PORTB,4
;              movlw  0xD9
;              movwf  ADCON0
;              goto   adc_olc

do_conv5      movlw  0xE1
              movwf  ADCON0
              goto   adc_olc

adc_olc call   swait

              bsf    ADCON0,0x2
wait_eoc      btfsc  ADCON0,0x2
              goto   wait_eoc

              movf   ADRESH,W
              movwf  analog
              return

;*****CIKIS KONTROL KISMI*****

out1      nop

o1_t6      movf   geldat11,w
           sublw  06h
           btfss  STATUS,2
           goto   o1_t7
           btfss  geldat1,7
           goto   o1_t60
           bsf    PORTB,0
           goto   BAS1
o1_t60     bcf    PORTB,0
           goto   BAS1

o1_t7      movf   geldat11,w
           sublw  07h
           btfss  STATUS,2
           goto   o1_t8
           btfss  geldat1,7
           goto   o1_t70
           bsf    PORTB,1
           goto   BAS1
o1_t70     bcf    PORTB,1
           goto   BAS1

o1_t8      movf   geldat11,w
           sublw  08h
           btfss  STATUS,2
           goto   o1_t9
           btfss  geldat1,7

```

	goto	o1_t80
	bsf	PORTB,2
	goto	BAS1
o1_t80	bcf	PORTB,2
	goto	BAS1
o1_t9	movf	geldat11,w
	sublw	09h
	btfss	STATUS,2
	goto	o1_ta
	btfss	geldat1,7
	goto	o1_t90
	bsf	PORTB,3
	goto	BAS1
o1_t90	bcf	PORTB,3
	goto	BAS1
o1_ta	movf	geldat11,w
	sublw	0Ah
	btfss	STATUS,2
	goto	o1_tb
	btfss	geldat1,7
	goto	o1_ta0
	bsf	PORTB,4
	goto	BAS1
o1_ta0	bcf	PORTB,4
	goto	BAS1
o1_tb	movf	geldat11,w
	sublw	0Bh
	btfss	STATUS,2
	goto	o1_tc
	btfss	geldat1,7
	goto	o1_tb0
	bsf	PORTB,5
	goto	BAS1
o1_tb0	bcf	PORTB,5
	goto	BAS1
o1_tc	movf	geldat11,w
	sublw	0Ch
	btfss	STATUS,2
	goto	o1_td
	btfss	geldat1,7
	goto	o1_tc0
	bsf	PORTB,6
	goto	BAS1
o1_tc0	bcf	PORTB,6
	goto	BAS1
o1_td	movf	geldat11,w
	sublw	0Dh
	btfss	STATUS,2
	goto	o1_te
	btfss	geldat1,7

```

        goto    o1_td0
        bsf     PORTB,7
        goto    BAS1
o1_td0  bcf     PORTB,7
        goto    BAS1

o1_te   movf    geldat11,w
        sublw   0Eh
        btfss   STATUS,2
        goto    o1_tf
        btfss   geldat1,7
        goto    o1_te0
        bsf     PORTC,0
        goto    BAS1
o1_te0  bcf     PORTC,0
        goto    BAS1

o1_tf   movf    geldat11,w
        sublw   0Fh
        btfss   STATUS,2
        goto    BAS1
        btfss   geldat1,7
        goto    o1_tf0
        bsf     PORTC,1
        goto    BAS1
o1_tf0  bcf     PORTC,1
        goto    BAS1

y_durum      GOTO  BAS1

,***** SERI GONDER VE AL ALTPROGRAMLARI*****

,*****SERI DATA GONDERME KISMI*****

gonder   movf    analog,w
        call     putc
        call     swait
        goto     BAS1

;W ye yuklenmis bir byte seri porttan gonderme altprogrami
;W = character

putc     bcf     STATUS,RP0          ; Select Bank 0.
        movwf    TXREG              ; Write it!
        bsf     PORTC,5

        bsf     STATUS,RP0          ; Select Bank 1
        movf    TXSTA,W             ; Peek transmit STATUS
putc1    btfss   TXSTA,1             ; Skip if TXbuffer empty
        goto    putc1               ; Try again
        bcf     STATUS,RP0          ; Select Bank 0.

        bcf     PORTC,5
        return

```

;*******SERI DATA OKUMA KISMI*******

;Seri porttan bir byte seri veri okuma altprogrami

;Okunan data W registerine aliniyor

```
getc    bcf    STATUS,RP0          ; Select Bank 0.
getc1   btfss  PIR1,RCIF           ; Skip if RC int flag set
        goto   getc1              ; Try again
        movf   RCREG,W             ; Read the geldat1acter
        bcf    PIR1,RCIF           ; Clear the interrupt flag
        return
```

```
hang    clrwdt                     ; Clear WDT in case it is enabled.
        goto   hang
```

;BEKLEME PROGRAMI 1

```
swait
        movlw  0xFF
        movwf  zd1
sw2     movlw  0xFF
        movwf  zd2
sw3     nop
        decfsz zd2,f
        goto   sw3
        decfsz zd1,f
        goto   sw2
        return
```

;BEKLEME PROGRAMI 2

```
bekle   movlw  .60
        movwf  zd1
bek1    nop
        decfsz zd1,f
        goto   bek1
        return
        end
```

EK 2. OTOMASYON PROGRAMININ KODLARI

Ana Ekran:

```
unit anaFormFrm;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ExtCtrls, CPort, Menus, StdCtrls, Grids, DBGrids, Mask, RzEdit, RzButton,
  jpeg, RzLabel, _GClass, AbCompas, RXClock, RzPanel, dxCntner, dxEditor,
  dxExEdtr, dxEdLib, ToolEdit;

type
  TanaForm = class(TForm)
    MainMenu1: TMainMenu;
    Tanmlar1: TMenuItem;
    KartElemanTanmla1: TMenuItem;
    PortTanmla1: TMenuItem;
    ComPort: TComPort;
    GroupBox1: TGroupBox;
    ElemanTanmla1: TMenuItem;
    Timer1: TTimer;
    beklemeSuresi: TTimer;
    lmElemanTanmla1: TMenuItem;
    ruzgarYonu: TAbCompass;
    portGosterge: TPanel;
    GroupBox3: TGroupBox;
    GroupBox2: TGroupBox;
    cikiselemanList: TDBGrid;
    testBilgi: TGroupBox;
    olcumElemanList: TDBGrid;
    sysZaman: TRxClock;
    Label1: TLabel;
    Label2: TLabel;
    RzToolbar1: TRzToolbar;
    RzToolButton1: TRzToolButton;
    RzToolButton2: TRzToolButton;
    RzToolButton3: TRzToolButton;
    RzToolButton4: TRzToolButton;
    ruzgarHizi: TLabel;
    sysTarih: TPanel;
    Ekstralar1: TMenuItem;
    Grafik1: TMenuItem;
    Hakknda1: TMenuItem;
    Hakknda2: TMenuItem;
    Kontroll1: TMenuItem;
    ElleKontroll1: TMenuItem;
    RzToolButton5: TRzToolButton;
    procedure ComPortStatus;
    procedure FormActivate(Sender: TObject);
    procedure ComPortAfterClose(Sender: TObject);
    procedure ComPortAfterOpen(Sender: TObject);
```

```

procedure PortTanmla1Click(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure KartElemanTanmla1Click(Sender: TObject);
procedure ComPortRxChar(Sender: TObject; Count: Integer);
procedure ElemanTanmla1Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
function TarihToMySQL(tarih:string): string;
procedure olcumDegerleriniYaz;
procedure lmElemanTanmla1Click(Sender: TObject);
procedure cikiselemanListDrawColumnCell(Sender: TObject;
  const Rect: TRect; DataCol: Integer; Column: TColumn;
  State: TGridDrawState);
procedure RzToolButton4Click(Sender: TObject);
procedure olcumElemanListDrawColumnCell(Sender: TObject;
  const Rect: TRect; DataCol: Integer; Column: TColumn;
  State: TGridDrawState);
procedure Hakknda2Click(Sender: TObject);
procedure ElleKontrol1Click(Sender: TObject);

private
  { Private declarations }
public
  { Public declarations }
  comPortOkunan:Integer;
  simdi:TDateTime;
end;

var
  anaForm: TanaForm;

implementation

uses dataModul, elemanTanimFrm, kartElemanTanimFrm, olcumElemaniTanimFrm,
  cikisTanimFrm, grafikFrm, grafikSecFrm, about, elleKontrolFrm, bekleFrm;

{$R *.DFM}
procedure TanaForm.olcumDegerleriniYaz;
begin
  ShowMessage (inttostr(comPortOkunan));
  if comPortOkunan <> -1 then begin
    with d.Query2 do begin
      close;
      sql.Clear;
      sql.Add('insert into olcumdegerleri set
tanimliElamanID='+d.Query1.FieldName('id').AsString+',
olcumDegeri='+inttostr(comPortOkunan)+'',
olcumZamani='"+TarihToMySQL(datetimetostr(simdi))+"'");
      ExecSQL;
    end;
    d.Query1.Next;
  end;
end;

```

```

function TanaForm.TarihToMySQL(tarih:string):string;
begin
    TarihToMySQL := copy(tarih,7,4)+'-'+copy(tarih,4,2)+'-'+copy(tarih,1,2)+' '+trim(copy(tarih,11,9));
end;

procedure TanaForm.ComPortStatus;
begin
    if ComPort.Connected then begin
        portGosterge.Color := clLime;
        portGosterge.BevelInner := bvRaised;
        portGosterge.Caption := 'İşlem Yapılıyor';
    end else begin
        portGosterge.Color := clInfoBk;
        portGosterge.BevelInner := bvLowered;
        portGosterge.Caption := 'İşlem Bekleniyor';
    end;
end;

procedure TanaForm.FormActivate(Sender: TObject);
begin
    sysTarih.Caption := datetostr(date);
    ComPortStatus;

end;

procedure TanaForm.ComPortAfterClose(Sender: TObject);
begin
    ComPortStatus;
end;

procedure TanaForm.ComPortAfterOpen(Sender: TObject);
begin
    ComPortStatus;
end;

procedure TanaForm.PortTanmla1Click(Sender: TObject);
begin
    ComPort.ShowSetupDialog;

    try
        ComPort.Open;
    except
        ComPort.Close;
    end;

    if ComPort.Connected then ComPort.Connected := false;

    ComPortStatus;
end;

procedure TanaForm.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    aboutform.ShowModal;
    Action := canone;
end;

```

```

procedure TanaForm.KartElemanTanmla1Click(Sender: TObject);
begin
    d.kartelemanDQ.open;
    kartelemanTanim.ShowModal;
    d.kartelemanDQ.close;

end;

procedure TanaForm.ComPortRxChar(Sender: TObject; Count: Integer);
var
    Okunan:string;
begin
    ComPort.ReadStr(Okunan,Count);
    comPortOkunan := ord(Okunan[1]);
end;

procedure TanaForm.ElemanTanmla1Click(Sender: TObject);
begin
    d.elemanDQ.open;
    elemanTanim.ShowModal;
    d.elemanDQ.close;
end;

procedure TanaForm.Timer1Timer(Sender: TObject);
var
    sayac,hesaplananOlcumDegeri,donguSayisi,kuralID:Integer;
    islemID,query:string;
    okumaTamam:bool;
    beklemeZamani:real;
begin
    Timer1.Enabled:=false;
    sayac := 0;
    if ComPort.Connected then begin
        while (sayac<30000) do begin
            Application.ProcessMessages;
            Sleep(20);
            sayac := sayac+1;
            if not anaForm.ComPort.Connected then begin
                anaForm.ComPort.Connected := true;
                anaForm.Timer1.Enabled := false;
                Break;
            end;
        end;
    end else begin
        anaForm.ComPort.Connected := true;
        anaForm.Timer1.Enabled := false;
    end;

    if (sayac>0) and (ComPort.Connected) then begin
        Timer1.Enabled:=true;
        exit;
    end;
end;

```

```

simdi := Now;
with d.timerQuery do begin
  close;
  sql.Clear;
  sql.Add('select et.*, e.adi, DATE_ADD(e.sonkontrol,INTERVAL (e.kontrolAraligi) MINUTE )
AS okumaZamani from elemantanim et,eleman e where et.elemanID=e.id and e.inOu="0" and
e.aktif="1"');
  open;
  try
    if ComPort.Connected then begin
      islemID := "";
      while not eof do begin
        if (simdi>=FieldByName('okumaZamani').AsDateTime) or
(FieldByName('okumaZamani').AsString="") then begin
          if islemID=FieldByName('id').AsString then
            donguSayisi := donguSayisi+1
          else begin
            donguSayisi := 1;
            islemID:=FieldByName('id').AsString;

            comPortOkunan := -1;
            ComPort.WriteStr(chr(FieldByName('cNo').AsInteger));

            sayac := 0;
            while (comPortOkunan=-1) and (sayac<15) do begin
              Application.ProcessMessages;
              Sleep(200);
              sayac:=sayac+1;
            end;
            end;

            if donguSayisi>5 then begin
              donguSayisi := 0;
              Next;
            end;

            if comPortOkunan<> -1 then begin
              hesaplananOlcumDegeri := comPortOkunan;
              with d.timerUpdate do begin
                close;
                sql.Clear;
                sql.Add('select * from olcumkarsilik k, elemantanim e WHERE
e.id='+d.timerQuery.FieldByName('id').AsString+' and k.olcumElemanID=e.olcumElemanID and
k.olculenDeger='+inttostr(hesaplananOlcumDegeri));
                open;

                if d.timerUpdate.RecordCount > 0 then hesaplananOlcumDegeri :=
d.timerUpdate.FieldByName('olcumKarsiligi').AsInteger;

                close;
                sql.Clear;
                sql.Add('insert into olcumdegerleri set
tanimliElemanID='+d.timerQuery.FieldByName('id').AsString+',

```

```

olcumDegeri='+inttostr(hesaplananOlcumDegeri)+'
olcumZamani="'+TarihToMySql(datetimetostr(simdi))+""";
    ExecSQL;

    close;
    sql.Clear;
    sql.Add('UPDATE eleman SET sonKontrol="'+TarihToMySql(datetimetostr(simdi))+""
WHERE id='+d.timerQuery.FieldByName('elemanID').AsString);
    ExecSQL;
    end;
    d.timerQuery.Next;
end;

end else
    d.timerQuery.Next;
end;

close;
sql.Clear;
islemID := 'select e.id,e.adi,Min(od.olcumDegeri) as "enDusuk",Max(od.olcumDegeri) as
"enYuksek",round(sum(od.olcumDegeri)/count(od.olcumDegeri)) as "ortalama" from elemantanim et
'+
    ' inner join eleman e ON et.elemanID=e.id and e.inOu="0" and e.aktif="1"'+
    ' inner join olcumdegerleri od ON od.tanimliElemanID=et.id and
od.olcumZamani="'+TarihToMySql(datetimetostr(simdi))+""'+
    ' group by e.id';
sql.Add(islemID);
open;
while not eof do begin
    with d.timerUpdate do begin
        close;
        sql.Clear;
        islemID := 'INSERT INTO elemanortalama SET '+
            'eleman='+d.timerQuery.FieldByName('id').AsString+
            ', olcumZamani="'+TarihToMySql(datetimetostr(simdi))+""'+
            ', enYuksek='+d.timerQuery.FieldByName('enYuksek').AsString+
            ', enDusuk='+d.timerQuery.FieldByName('enDusuk').AsString+
            ', ortalama='+d.timerQuery.FieldByName('ortalama').AsString;
        sql.Add(islemID);
        ExecSQL;
    end;
    Next;
end;
d.sonOlcumler.Refresh;

Close;
sql.clear;
islemID := 'select e.id,e.adi,eo.ortalama as ortalama,e.birimi as birimi, eo.olcumZamani as
olcumZamani from eleman e '+
    'inner join elemanortalama eo ON eo.eleman=e.id and eo.id=(select max(id) from
elemanortalama where eleman=e.id) '+
    'where e.aktif="1" and (e.id=1 or e.id=2) '+
    'group by eo.eleman';

SQL.Add(islemID);

```

```

open;

while not eof do begin
    if FieldByName('id').AsInteger=1 then ruzgarHizi.Caption :=
FieldByName('ortalama').AsString + ' '+FieldByName('birimi').AsString;
    if FieldByName('id').AsInteger=2 then ruzgarYonu.ValueShould :=
FieldByName('ortalama').AsInteger;

    next;
end;

// Kural Kontrolleri Başlangıcı

Close;
sql.clear;
SQL.Add('SELECT * from cikiskurallari WHERE aktif="1"');
open;

while not Eof do begin
    query := 'SELECT '+
        'o1.eleman as eId_1'+
        ', o1.ortalama as d_1'+
        ',e.*';

    for sayac:=2 to 5 do begin
        if FieldByName('elemanID'+inttostr(sayac)).Text <>" then begin
            query := query + ', o'+inttostr(sayac)+'eleman as eId_'+inttostr(sayac)+
                ', o'+inttostr(sayac)+'ortalama as d_'+inttostr(sayac);
        end;
    end;

    query := query + ' from '+
        ' elemanortalama as o1'+
        ',elemantanim as e';

    for sayac:=2 to 5 do begin
        if FieldByName('elemanID'+inttostr(sayac)).Text <>" then begin
            query := query + ', elemanortalama as o'+inttostr(sayac);
        end;
    end;

    query := query + ' WHERE '+
        'o1.eleman='+FieldByName('elemanID1').Text+' and '+
        'o1.id=(select max(id) as id from elemanortalama WHERE
eleman='+FieldByName('elemanID1').Text+') and '+
        'o1.ortalama'+FieldByName('sart1').Text+FieldByName('deger1').Text+
        ' and e.id='+FieldByName('cikisElemanID').Text;

    for sayac:=2 to 5 do begin
        if FieldByName('elemanID'+inttostr(sayac)).Text <>" then begin
            query := query +

```

```

        ' and o'+inttostr(sayac)+'.eleman='+FieldByName('elemanID'+inttostr(sayac)).Text+'
and '+'
        'o'+inttostr(sayac)+'.id=(select max(id) as id from elemanortalama WHERE
eleman='+FieldByName('elemanID'+inttostr(sayac)).Text+') and '+'

'o'+inttostr(sayac)+'.ortalama'+FieldByName('sart'+inttostr(sayac)).Text+FieldByName('deger'+inttostr
(sayac)).Text;
    end;
end;

with d.timerUpdate do begin
    close;
    sql.Clear;
    sql.Add(query);
    open;

    if d.timerUpdate.RecordCount >0 then begin
        while not Eof do begin
            with d.timerUpdate2 do begin
                close;
                sql.Clear;
                sql.Add('select * from cikisDurum WHERE kuralID=0 and
tanimliElemanID='+d.timerUpdate.FieldByName('id').AsString+' and id=(select max(id) from
cikisDurum WHERE tanimliElemanID='+d.timerUpdate.FieldByName('id').AsString+' );
                open;

                kuralID :=1;
                if d.timerUpdate2.RecordCount = 0 then begin
                    close;
                    sql.Clear;
                    query := 'SELECT max(islemZamani) as islemZamani FROM cikisDurum WHERE
tanimliElemanID='+d.timerUpdate.FieldByName('id').AsString+' and
cikisData='+d.timerQuery.FieldByName('cikisData').AsString+' and id=(select max(id) from
cikisDurum WHERE tanimliElemanID='+d.timerUpdate.FieldByName('id').AsString+' );
                    sql.Add(query);
                    open;
                end else kuralID := 0;
            end;

            if (kuralID>0) and (d.timerUpdate2.FieldByName('islemZamani').Text='') then begin
                comPortOkunan := -1;
                if d.timerQuery.FieldByName('cikisData').AsInteger=1 then begin
                    ComPort.WriteString(chr(FieldByName('cNo').AsInteger));
                end else if d.timerQuery.FieldByName('cikisData').AsInteger=2 then begin
                    ComPort.WriteString(chr(FieldByName('cNo').AsInteger+128));
                end else if d.timerQuery.FieldByName('cikisData').AsInteger=3 then begin
                    ComPort.WriteString(chr(FieldByName('cNoKapat').AsInteger+128));
                end;
                sayac :=0;
                while (sayac<100) do begin
                    Application.ProcessMessages;
                    Sleep(10*FieldByName('maxKapanmaSuresi').AsInteger);
                    sayac:=sayac+1;
                end;
                ComPort.WriteString(chr(FieldByName('cNoKapat').AsInteger));
            end else if d.timerQuery.FieldByName('cikisData').AsInteger>3 then begin

```

```

ComPort.WriteStr(chr(FieldByName('cNoKapat').AsInteger+128));
sayac :=0;
while (sayac<100)do begin
    Application.ProcessMessages;
    Sleep(10*FieldByName('maxKapanmaSuresi').AsInteger);
    sayac:=sayac+1;
end;
ComPort.WriteStr(chr(FieldByName('cNoKapat').AsInteger));

sayac :=0;
while (sayac<100)do begin
    Application.ProcessMessages;
    Sleep(10);
    sayac:=sayac+1;
end;
ComPort.WriteStr(chr(FieldByName('cNo').AsInteger+128));

sayac:=FieldByName('maxKapanmaSuresi').AsInteger;
beklemeZamani := round((sayac-3)*10/100*10);
sayac :=0;
while (sayac<100)do begin
    Application.ProcessMessages;
    Sleep( round(beklemeZamani)) ;
    sayac:=sayac+1;
end;
ComPort.WriteStr(chr(FieldByName('cNo').AsInteger));
end;

sayac :=0;
while (sayac<150) do begin
    Application.ProcessMessages;
    Sleep(20);
    sayac := sayac+1;
end;

with d.timerUpdate2 do begin
    close;
    sql.Clear;
    query := 'insert into cikisDurum set
tanimliElemanID='+d.timerUpdate.FieldByName('id').AsString+',
cikisData='+d.timerQuery.FieldByName('cikisData').AsString+',
kuralID='+d.timerQuery.FieldByName('id').AsString+',
islemZamani="'+TarihToMySql(datetimetostr(simdi))+""';
    sql.Add(query);
    ExecSQL;
end;
end;
Next;
end;
end;
end;
Next;
end;

```

//Kural KOntrolleri Bitisi

```

d.sonDurumlar.Refresh;

Timer1.Enabled:=true;
if ComPort.Connected then ComPort.Connected := false;
end;
except
  ShowMessage('Port Açılmadı');
  Timer1.Enabled:=true;
end;
end;
end;

procedure TanaForm.ImElemanTanmla1Click(Sender: TObject);
begin
  with d.Query2 do begin
    close;
    sql.Clear;
    sql.Add('select et.*,e.adi from eleman e '+'
      'inner join elemantanım et ON e.id = et.elemanID '+'
      'WHERE inOu="1" and aktif="1" order by e.adi,et.kartNo,et.portNo');
    open;

    cikisKuralTanim.elemanID.items.Clear;
    cikisKuralTanim.elemanID.Values.Clear;
    cikisKuralTanim.elemanID.AddItemValue('Tüm Aktif Kurallar','-1');
    cikisKuralTanim.elemanID.AddItemValue('Tüm Pasif Kurallar','-2');
    cikisKuralTanim.elemanID.AddItemValue('Tüm Kurallar','-3');
    while not Eof do begin
      cikisKuralTanim.elemanID.AddItemValue(fieldbyname('adi').Text+ '
      ('+KartNo: '+fieldbyname('kartNo').Text+'-PortNo: '+fieldbyname('portNo').Text+')
      Kuralları',fieldbyname('id').Text);
      Next;
    end;
  end;

  cikisKuralTanim.elemanID.ItemIndex := 0;
  cikisKuralTanim.elemanID.OnChange(Sender);

  d.cikisKurallariDQ.open;
  cikisKuralTanim.ShowModal;
  d.cikisKurallariDQ.close;
end;

procedure TanaForm.cikiselemanListDrawColumnCell(Sender: TObject;
  const Rect: TRect; DataCol: Integer; Column: TColumn;
  State: TGridDrawState);
var
  deger :string;
begin
  deger := "";
  if datacol=2 then begin
    if Column.Field.AsString='0' then
      deger:='Giriş Elemanı'

```

```

else if Column.Field.AsString='1' then
    deger:='Kapalı'
else if Column.Field.AsString='2' then
    deger:='Açık'
else if Column.Field.AsString='3' then
    deger:='Tam Kapalı'
else if Column.Field.AsString='4' then
    deger:='%10 Açık'
else if Column.Field.AsString='5' then
    deger:='%20 Açık'
else if Column.Field.AsString='6' then
    deger:='%30 Açık'
else if Column.Field.AsString='7' then
    deger:='%40 Açık'
else if Column.Field.AsString='8' then
    deger:='%50 Açık'
else if Column.Field.AsString='9' then
    deger:='%60 Açık'
else if Column.Field.AsString='10' then
    deger:='%70 Açık'
else if Column.Field.AsString='11' then
    deger:='%80 Açık'
else if Column.Field.AsString='12' then
    deger:='%90 Açık'
else if Column.Field.AsString='13' then
    deger:='Tam Açık'
else
    deger:='Tanımsız';
cikisolemanList.Canvas.TextRect(Rect,Rect.Left+4,Rect.top+2,deger);
end;
end;

procedure TanaForm.RzToolButton4Click(Sender: TObject);
begin
    grafikSec.basTarih.DateTime := date;
    grafikSec.bitTarih.DateTime := date;
    grafikSec.basZaman.Time := StrToTime('00:00');
    grafikSec.bitZaman.Time := StrToTime('23:59');

    with d.Query2 do begin
        close;
        sql.clear;
        sql.add('select * from eleman WHERE inOu="0"');
        open;

        grafikSec.elemanID.items.Clear;
        grafikSec.elemanID.Values.Clear;
        First;
        while not Eof do begin
            grafikSec.elemanID.AddItemValue(fieldbyname('adi').Text,fieldbyname('id').Text);
            Next;
        end;
    end;
    grafikSec.showmodal;
end;

```

```

procedure TanaForm.olcumElemanListDrawColumnCell(Sender: TObject;
const Rect: TRect; DataCol: Integer; Column: TColumn;
State: TGridDrawState);
begin
if (datacol>0) and (datacol<4) then begin
if Column.Field.AsString="" then
olcumElemanList.Canvas.TextRect(Rect,Rect.Left+4,Rect.top+2,'-');
end;
if datacol=4 then begin
if Column.Field.AsString="" then
olcumElemanList.Canvas.TextRect(Rect,Rect.Left+4,Rect.top+2,'Hiç Ölçüm Yok');
end;
if datacol=3 then begin
if d.sonOlcumler.fieldbyname('id').AsInteger = 1 then
ruzgarHizi.Caption := Column.Field.AsString;
if d.sonOlcumler.fieldbyname('id').AsInteger = 2 then
ruzgarYonu.ValueShould := d.sonOlcumler.fieldbyname('ort').AsInteger;
end;
end;

procedure TanaForm.Hakknda2Click(Sender: TObject);
begin
aboutform.ShowModal;
end;

procedure TanaForm.ElleKontrol1Click(Sender: TObject);
begin
with d.Query2 do begin
close;
sql.Clear;
sql.Add('select et.*,e.adi from eleman e '+'
'inner join elemantanim et ON e.id = et.elemanID '+'
'WHERE inOu="1" and aktif="1" order by e.adi,et.kartNo,et.portNo');
open;
elleKontrol.elemanID.items.Clear;
elleKontrol.elemanID.Values.Clear;
while not Eof do begin
elleKontrol.elemanID.AddItemValue(fieldbyname('adi').Text+ '
('+KartNo:+'fieldbyname('kartNo').Text+'-
PortNo:+'fieldbyname('portNo').Text+')',fieldbyname('id').Text);
Next;
end;
end;
elleKontrol.elemanID.ItemIndex := -1;
elleKontrol.cikisData.ItemIndex := -1;
elleKontrol.elemanBilgi.Caption := "";
elleKontrol.otoKontrol.Enabled :=false;

elleKontrol.ShowModal;
end;
end.

```

İşlem Tanım Ekranı

unit elemanTanimFrm;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
RzPanel, RzButton, ExtCtrls, Grids, DBGrids;

type

```
TelemanTanim = class(TForm)
  DBGrid1: TDBGrid;
  RzToolBar1: TRzToolBar;
  RzToolButton1: TRzToolButton;
  RzToolButton2: TRzToolButton;
  RzToolButton3: TRzToolButton;
  PlugInsaaaa: TRzToolButton;
  procedure RzToolButton1Click(Sender: TObject);
  procedure RzToolButton2Click(Sender: TObject);
  procedure RzToolButton3Click(Sender: TObject);
  procedure DBGrid1DrawColumnCell(Sender: TObject; const Rect: TRect;
    DataCol: Integer; Column: TColumn; State: TGridDrawState);
  procedure PlugInsaaaaClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
```

var

elemanTanim: TelemanTanim;

implementation

uses elemanEkleFrm, dataModul, olcumElemaniTanimFrm;

{ \$R *.DFM }

procedure TelemanTanim.RzToolButton1Click(Sender: TObject);

begin

```
  elemanEkle.adi.Text := '';
  elemanEkle.inOut.ItemIndex := 0;
  elemanEkle.ekleDuzelt.Caption := 'ekle';
  elemanEkle.aktif.ItemIndex := 0;
  elemanEkle.birimi.Text := '';
  elemanEkle.inOut.OnChange(Sender);
  elemanEkle.showmodal;
```

end;

procedure TelemanTanim.RzToolButton2Click(Sender: TObject);

begin

```
  elemanEkle.adi.Text := d.elemanDQ.fieldbyname('adi').Text;
  elemanEkle.inOut.ItemIndex := d.elemanDQ.fieldbyname('inOu').AsInteger;
  elemanEkle.birimi.Text := d.elemanDQ.fieldbyname('birimi').text;
```

```

if elemanEkle.inOut.ItemIndex = 0 then begin
  if (d.elemanDQ.fieldbyname('kontrolAraligi').AsInteger mod (60*24)) = 0 then begin
    elemanEkle.zaman.ItemIndex := 2;
    elemanEkle.kontrolAraligi.Value :=
d.elemanDQ.fieldbyname('kontrolAraligi').AsInteger/(60*24);
  end else if (d.elemanDQ.fieldbyname('kontrolAraligi').AsInteger mod 60) = 0 then begin
    elemanEkle.zaman.ItemIndex := 1;
    elemanEkle.kontrolAraligi.Value := d.elemanDQ.fieldbyname('kontrolAraligi').AsInteger/60;
  end else begin
    elemanEkle.zaman.ItemIndex := 0;
    elemanEkle.kontrolAraligi.Value := d.elemanDQ.fieldbyname('kontrolAraligi').AsInteger;
  end;
end else begin
  elemanEkle.kontrolAraligi.Text := "";
  elemanEkle.zaman.ItemIndex := -1;
end;

if d.elemanDQ.fieldbyname('aktif').AsString = '1' then
  elemanEkle.aktif.ItemIndex := 0
else
  elemanEkle.aktif.ItemIndex := 1;

elemanEkle.inOut.OnChange(Sender);
elemanEkle.ekleDuzelt.Caption := 'duzelt';
elemanEkle.showmodal;

end;

procedure TelemanTanim.RzToolButton3Click(Sender: TObject);
begin
  if d.elemanDQ.RecordCount = 0 then exit;

  with d.Query1 do begin
    close;
    sql.Clear;
    sql.Add('select * from elemantanim where elemanID='+d.elemanDQ.fieldbyname('id').AsString);
    open;

    if d.Query1.RecordCount > 0 then begin
      ShowMessage('Bu işlemin Kart Elaman tanımı yapılmış olduğundan
Silinemez'+#13#10+'Öncelikle tanımlardan silinmeli'+!!!);
      exit;
    end;

    close;
    sql.Clear;
    sql.Add('select * from cikiskurallari where
elemanID1='+d.elemanDQ.fieldbyname('id').AsString+' or '+'
'elemanID2='+d.elemanDQ.fieldbyname('id').AsString+' or '+'
'elemanID3='+d.elemanDQ.fieldbyname('id').AsString+' or '+'
'elemanID4='+d.elemanDQ.fieldbyname('id').AsString+' or '+'
'elemanID5='+d.elemanDQ.fieldbyname('id').AsString+' or '+'
'cikisElemanID='+d.elemanDQ.fieldbyname('id').AsString);
    open;
  end;
end;

```

```

        if d.Query1.RecordCount > 0 then begin
            ShowMessage('Bu işlem Çıkış Elemanları Kurallarında kullanılmış olduğundan
Silinemez'+#13#10+'Öncelikle kurallardan silinmeli'+!!!');
            exit;
        end;

        if (MessageDlg('Seçili işlemi Sistem dışı Bırakmak istediğinizden emin misiniz?', mtConfirmation,
[mbYes, mbNo], 0) = mrNo) then exit;

        close;
        sql.Clear;
        sql.Add('DELETE FROM elemanortalama WHERE
eleman='+d.elemanDQ.fieldbyname('id').AsString);
        ExecSQL;

        close;
        sql.Clear;
        sql.Add('select * FROM olcumeleman WHERE
elemanID='+d.elemanDQ.fieldbyname('id').AsString);
        open;

        while not Eof do begin
            with d.Query2 do begin
                close;
                sql.Clear;
                sql.Add('DELETE FROM olcumkarsilik WHERE
olcumElemanID='+d.Query1.fieldbyname('id').AsString);
                ExecSQL;
            end;
            Next;
        end;

        close;
        sql.Clear;
        sql.Add('DELETE FROM olcumeleman WHERE
elemanID='+d.elemanDQ.fieldbyname('id').AsString);
        ExecSQL;

        close;
        sql.Clear;
        sql.Add('DELETE FROM eleman WHERE id='+d.elemanDQ.fieldbyname('id').AsString);
        ExecSQL;
        end;
        d.elemanDQ.Refresh;
    end;

procedure TelemanTanim.DBGrid1DrawColumnCell(Sender: TObject;
const Rect: TRect; DataCol: Integer; Column: TColumn;
State: TGridDrawState);
var
deger :string;
begin
    deger :='';
    if datacol=1 then begin

```

```

    if Column.Field.AsString='0' then
        deger:='Giriş'
    else if Column.Field.AsString='1' then
        deger:='Çıkış';
    DBGrid1.Canvas.TextRect(Rect,Rect.Left+4,Rect.top+2,deger);
end;

end;

procedure TelemanTanim.PlugInsaaaaClick(Sender: TObject);
begin
    if d.elemanDQ.FieldByName('inOu').AsInteger = 1 then begin
        ShowMessage ('Çıkış Elemanı Seçilmiş');
        exit;
    end;

    with d.olcumElemanDQ do begin
        close;
        sql.Clear;
        sql.add('select *,e.birimi from olcumelemanı o, eleman e WHERE e.id=o.elemanID and
        elemanID='+d.elemanDQ.fieldbyname('id').Text+' order by o.elemanBilgi');
        open;
        olcumElemanTanim.ShowModal;
        d.olcumElemanDQ.close;
    end;
end;

end.

```

İşlem Ekle/Düzeltilme Ekranı

```

unit elemanEkleFrm;

interface

uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
    StdCtrls, RzCmboBx, Mask, RZEdit, RzPanel, RzButton, ExtCtrls,shellapi;

type
    TelemanEkle = class(TForm)
        RzToolBar1: TRzToolBar;
        RzToolButton1: TRzToolButton;
        RzToolButton2: TRzToolButton;
        pnlArama: TRzPanel;
        Label2: TLabel;
        Label3: TLabel;
        inOut: TRzComboBox;
        ekleDuzelt: TLabel;
        adi: TRzEdit;
        Label1: TLabel;
        kontrolAraligi: TRzNumericEdit;
        zaman: TRzComboBox;
        Label4: TLabel;
        aktif: TRzComboBox;
    end;

```

```

olcumElemanLbl: TLabel;
Label5: TLabel;
birimi: TRzEdit;
charMap: TRzBitBtn;
hareketDurumu: TRzComboBox;
procedure RzToolButton2Click(Sender: TObject);
procedure RzToolButton1Click(Sender: TObject);
procedure adiKeyPress(Sender: TObject; var Key: Char);
procedure inOutChange(Sender: TObject);
procedure charMapClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  elemanEkle: TelemanEkle;

implementation

uses dataModul;

{$R *.DFM}

procedure TelemanEkle.RzToolButton2Click(Sender: TObject);
begin
  close;
end;

procedure TelemanEkle.RzToolButton1Click(Sender: TObject);
var
  query,aktiflik:string;
  hesaplananZaman:real;
begin
  if (ekleDuzelt.Caption <>'ekle') and ( ((d.elemanDQ.fieldbyname('inOu').AsInteger=0) and
(inOut.ItemIndex=1)) or ((d.elemanDQ.fieldbyname('inOu').AsInteger=1) and (inOut.ItemIndex=0)) )
  then begin
    ShowMessage('Giriş işlemi Çıkış, Çıkış işlemi Giriş olarak değiştirilemez!!!');
    inOut.SetFocus;
    exit;
  end;

  if adi.Text = " then begin
    ShowMessage ('İşlem Adı Belirtilmemiş!');
    adi.SetFocus;
    exit;
  end;

  if hareketDurumu.ItemIndex < 0 then begin
    ShowMessage ('Hareket Durumu Belirtilmemiş!');
    hareketDurumu.SetFocus;
    exit;
  end;

```

```

if inOut.ItemIndex = 0 then begin
  if birimi.Text = " then begin
    ShowMessage ('Birimi Belirtilmemiş!');
    birimi.SetFocus;
    exit;
  end;

  if kontrolAraligi.Value = 0 then begin
    ShowMessage ('Kontrol Aralığı Belirtilmemiş!');
    kontrolAraligi.SetFocus;
    exit;
  end;

  if zaman.ItemIndex < 0 then begin
    ShowMessage ('Kontrol Aralığı Belirtilmemiş!');
    zaman.SetFocus;
    exit;
  end;
end;

with d.Query1 do begin
  if (ekleDuzelt.Caption <>'ekle') and (aktif.Value='0') and
  (d.elemanDQ.fieldbyname('inOu').AsInteger=0) then begin
    close;
    sql.Clear;
    sql.Add('select * from cikiskurallari where
    elemanID1='+d.elemanDQ.fieldbyname('id').AsString+' or '+'
    'elemanID2='+d.elemanDQ.fieldbyname('id').AsString+' or '+'
    'elemanID3='+d.elemanDQ.fieldbyname('id').AsString+' or '+'
    'elemanID4='+d.elemanDQ.fieldbyname('id').AsString+' or '+'
    'elemanID5='+d.elemanDQ.fieldbyname('id').AsString);
    open;

    if d.Query1.RecordCount > 0 then begin
      ShowMessage('Bu işlem Çıkış Elemanları Kurallarında kullanılmış olduğundan Pasif
      Yapılamaz!!!');
      aktif.SetFocus;
      exit;
    end;
  end;

  if (ekleDuzelt.Caption <>'ekle') and (aktif.Value='0') and
  (d.elemanDQ.fieldbyname('inOu').AsInteger=1) then begin
    close;
    sql.Clear;
    sql.Add('select * from cikiskurallari where
    cikisElemanID='+d.elemanDQ.fieldbyname('id').AsString);
    open;

    if d.Query1.RecordCount > 0 then begin
      ShowMessage('Bu işlem Çıkış Elemanları Kurallarında kullanılmış olduğundan Pasif
      Yapılamaz!!!');
      aktif.SetFocus;
      exit;
    end;
  end;
end;

```

```

    end;
end;

close;
sql.Clear;
if ekleDuzelt.Caption ='ekle' then
    query := 'INSERT INTO eleman SET '
else
    query := 'UPDATE eleman SET ';

if inOut.ItemIndex = 0 then begin

    hesaplananZaman := kontrolAraligi.Value;
    if zaman.ItemIndex = 1 then
        hesaplananZaman := kontrolAraligi.Value*60
    else if zaman.ItemIndex = 2 then
        hesaplananZaman := kontrolAraligi.Value*60*24;
    end else begin
        hesaplananZaman := 0;
        birimi.Text := '';
    end;

if aktif.ItemIndex = 0 then
    aktiflik := 'I'
else
    aktiflik := '0';

    query := query + ' adi="'+adi.Text+'", inOu="'+inttostr(inOut.ItemIndex)+'",
hareketDurumu="'+hareketDurumu.Value+'", birimi="'+birimi.Text+'", aktif="'+aktiflik+'",
kontrolAraligi='+floattostr(hesaplananZaman);

    if ekleDuzelt.Caption ='duzelt' then query := query + ' WHERE
id='+d.elemanDQ.fieldbyname('id').AsString;
    SQL.Add(query);
    ExecSQL;
end;
d.elemanDQ.Refresh;
close;
end;

procedure TelemanEkle.adiKeyPress(Sender: TObject; var Key: Char);
begin
    if (Key='i') or (Key='İ') then Key := 'I';
    if (Key='ı') or (Key='I') then Key := 'I';
end;

procedure TelemanEkle.inOutChange(Sender: TObject);
begin
    hareketDurumu.items.Clear;
    hareketDurumu.Values.Clear;
    if inOut.ItemIndex = 0 then begin
        hareketDurumu.AddItemValue('Giriş','0');
        hareketDurumu.ItemIndex := 0;
        birimi.Enabled := True;
    end;
end;

```

```

charMap.Enabled := True;
kontrolAraligi.Enabled := True;
zaman.Enabled := True;
end else begin
hareketDurumu.AddItemValue('On/Off','1');
hareketDurumu.AddItemValue('Kademeli','2');
birimi.Enabled := False;
charMap.Enabled := False;
kontrolAraligi.Enabled := False;
zaman.Enabled := False;
try
hareketDurumu.ItemIndex := d.elemanDQ.fieldbyname('hareketDurumu').AsInteger -1;
except
hareketDurumu.ItemIndex := -1;
end;
end;
end;

procedure TelemanEkle.charMapClick(Sender: TObject);
var
Dosya:string;
begin
Dosya :=ExtractFilePath(Application.ExeName)+'charmap.exe';
ShellExecute(elemanEkle.Handle, nil, PChar(Dosya), nil, nil, SW_SHOWNORMAL);
end;
end.

```

Ölçme Eleman Tanım Ekranı

```

unit olcumElemaniTanimFrm;

interface

uses
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
Grids, DBGrids, RzButton, ExtCtrls, RzPanel;

type
TolcumElemanTanim = class(TForm)
RzToolBar1: TRzToolBar;
RzToolButton1: TRzToolButton;
RzToolButton2: TRzToolButton;
RzToolButton3: TRzToolButton;
DBGrid1: TDBGrid;
RzToolButton4: TRzToolButton;
procedure RzToolButton2Click(Sender: TObject);
procedure RzToolButton1Click(Sender: TObject);
procedure RzToolButton4Click(Sender: TObject);
procedure RzToolButton3Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;

```

```

var
    olcumElemanTanim: TolcumElemanTanim;

implementation

uses dataModul, olcumElemaniEkleFrm, olcumKarsilikFrm;

{$R *.DFM}

procedure TolcumElemanTanim.RzToolButton2Click(Sender: TObject);
begin
    olcumElemaniEkle.elemanBilgi.Text := d.olumElemanDQelemanBilgi.Text;
    olcumElemaniEkle.aciklama.Text := d.olumElemanDQaciklama.Text;
    if FileExists(d.olumElemanDQresim.Text) then begin
        olcumElemaniEkle.olumElemanResmi.Picture.LoadFromFile(d.olumElemanDQresim.Text);
        olcumElemaniEkle.resimYolu.Caption := d.olumElemanDQresim.Text;
    end else begin
        olcumElemaniEkle.olumElemanResmi.Picture := nil;
        olcumElemaniEkle.resimPaneli.Caption := 'Resim Yok';
        olcumElemaniEkle.resimYolu.Caption := '';
    end;
    olcumElemaniEkle.ekleDuzelt.Caption := 'duzelt';
    olcumElemaniEkle.showmodal;

end;

procedure TolcumElemanTanim.RzToolButton1Click(Sender: TObject);
begin
    olcumElemaniEkle.elemanBilgi.Text := '';
    olcumElemaniEkle.aciklama.Text := '';
    olcumElemaniEkle.olumElemanResmi.Picture := nil;
    olcumElemaniEkle.resimPaneli.Caption := 'Resim Yok';
    olcumElemaniEkle.ekleDuzelt.Caption := 'ekle';
    olcumElemaniEkle.showmodal;

end;

procedure TolcumElemanTanim.RzToolButton4Click(Sender: TObject);
begin
    if d.olumElemanDQ.RecordCount > 0 then begin
        olcumKarsilik.elemanBilgi.Caption := d.olumElemanDQelemanBilgi.Text;
        olcumKarsilik.aciklama.Text := d.olumElemanDQaciklama.Text;
        if FileExists(d.olumElemanDQresim.Text) then begin
            olcumKarsilik.olumElemanResmi.Picture.LoadFromFile(d.olumElemanDQresim.Text);
        end else begin
            olcumKarsilik.olumElemanResmi.Picture := nil;
            olcumKarsilik.resimPaneli.Caption := 'Resim Yok';
        end;
    end;
    olcumKarsilik.DBGrid1.Columns[1].Title.Caption := 'Karşılığı ('+
d.olumElemanDQbirimi.Text+')';

    with d.olumKarsilikDQ do begin

```

```

        close;
        sql.Clear;
        sql.Add('select * from olcumkarsilik WHERE
olcumElemanID='+d.olcumElemanDQ.fieldbyname('id').AsString+' order by olculenDeger');
        open;
        olcumKarsilik.ShowModal;
        close;
    end;
end;
end;

procedure TolcumElemanTanim.RzToolButton3Click(Sender: TObject);
begin
    if d.olcumElemanDQ.RecordCount = 0 then exit;

    with d.Query1 do begin
        close;
        sql.Clear;
        sql.Add('select * FROM elementanim WHERE
olcumElemanID='+d.olcumElemanDQ.fieldbyname('id').AsString);
        open;

        if d.Query1.RecordCount > 0 then begin
            ShowMessage('Bu Ölçüm Elemanı Kart-Element Tanımlarında Kullanıldığından Silinemez!');
            exit;
        end;
        if (MessageDlg('Ölçüm Elamanı karşılıklarıyla birlikte silinecektir.'+#13#10+'Yine de silmek
istediğinizden emin misiniz?', mtConfirmation, [mbYes, mbNo], 0) = mrNo) then exit;
        close;
        sql.Clear;
        sql.Add('DELETE from olcumkarsilik WHERE
olcumElemanID='+d.olcumElemanDQ.FieldByName('id').AsString);
        ExecSQL;
        close;
        sql.Clear;
        sql.Add('DELETE from olcumelemanı WHERE
id='+d.olcumElemanDQ.FieldByName('id').AsString);
        ExecSQL;
        end;
        d.olcumElemanDQ.Refresh;
    end;

end.

```

Ölçme Elemanı Ekle/Düzeltil Ekranı

```

unit olcumElemaniEkleFrm;

interface

uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
    RzButton, RzShellDialogs, Mask, RzEdit, StdCtrls, ExtCtrls, RzPanel, shellapi,
    jpeg;

```

```

type
  TolcumElemaniEkle = class(TForm)
    RzToolBar1: TRzToolBar;
    RzToolButton1: TRzToolButton;
    RzToolButton2: TRzToolButton;
    pnlArama: TRzPanel;
    Label4: TLabel;
    aciklama: TRzMemo;
    Label5: TLabel;
    elemanBilgi: TRzEdit;
    OpenFileDialog1: TRzOpenDialog;
    RzBitBtn1: TRzBitBtn;
    RzBitBtn2: TRzBitBtn;
    ekleDuzelt: TLabel;
    resimPaneli: TPanel;
    olcumElemanResmi: TImage;
    resimYolu: TLabel;
    procedure RzBitBtn1Click(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure RzBitBtn2Click(Sender: TObject);
    procedure RzToolButton1Click(Sender: TObject);
    procedure RzToolButton2Click(Sender: TObject);
    procedure elemanBilgiKeyPress(Sender: TObject; var Key: Char);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  olcumElemaniEkle: TolcumElemaniEkle;

implementation

uses dataModul;

{$R *.DFM}

procedure TolcumElemaniEkle.RzBitBtn1Click(Sender: TObject);
var
  Kaynak,Hedef:String;
begin
  OpenFileDialog1.InitialDir := ExtractFilePath(Application.ExeName)+'Resimler\';

  if not OpenFileDialog1.Execute then exit;

  Kaynak := OpenFileDialog1.FileName;
  Hedef := ExtractFilePath(Application.ExeName)+'Resimler\' + ExtractFileName(Kaynak);

  if Hedef <> Kaynak then begin
    if FileExists(Hedef) then begin
      ShowMessage('Aynı isimle başka bir dosyanız Mevcut');
      exit;
    end;
  end;

```

```

CopyFile(PChar(Kaynak),PChar(Hedef),false);
end;

olcumElemanResmi.Picture.LoadFromFile(hedef);
resimYolu.Caption := Hedef;

end;

procedure TolcumElemaniEkle.FormActivate(Sender: TObject);
begin
if (resimYolu.Caption = "") then
    resimPaneli.Caption := 'Resim Yok'
else
    resimPaneli.Caption := "";
end;

procedure TolcumElemaniEkle.RzBitBtn2Click(Sender: TObject);
begin
if resimPaneli.Caption = 'Resim Yok' then exit;
if MessageDlg('Resmi kaldırmak istediğinizden emin misiniz?',mtconfirmation,[mbytes,mbno],0) =
mrno then exit;

olcumElemanResmi.Picture := nil;
resimYolu.Caption := "";
resimPaneli.Caption := 'Resim Yok';

end;

procedure TolcumElemaniEkle.RzToolButton1Click(Sender: TObject);
var
query,Harf,mysqlResimYolu:string;
sayac:integer;
begin
if elemanBilgi.Text = " then begin
    ShowMessage ('Ölüm Elemanı Belirtilmemiş!');
    elemanBilgi.SetFocus;
    exit;
end;

mysqlResimYolu := "";
for sayac:=1 to length(resimYolu.Caption) do begin
    Harf := copy(resimYolu.Caption,sayac,1);

    if Harf = '\' then
        mysqlResimYolu := mysqlResimYolu+'\'';

    mysqlResimYolu := mysqlResimYolu+Harf;
end;

with d.Query1 do begin
    close;
    sql.Clear;
    if ekleDuzelt.Caption ='ekle' then

```

```

        query := 'INSERT INTO olcumelemanı SET '
    else
        query := 'UPDATE olcumelemanı SET ';

    query := query +' elemanID='+d.elemanDQ.fieldbyname('id').Text+',
    elemanBilgi="'+elemanBilgi.Text+'",aciklama="'+aciklama.Text+'", resim="'+mysqlResimYolu+'";

    if ekleDuzelt.Caption ='duzelt' then query := query +' WHERE
    id='+d.olcumElemanDQ.fieldbyname('id').AsString;
    SQL.Add(query);
    ExecSQL;
    end;
    d.olcumElemanDQ.Refresh;
    close;

end;

procedure TolcumElemaniEkle.RzToolButton2Click(Sender: TObject);
begin
    close;
end;

procedure TolcumElemaniEkle.elemanBilgiKeyPress(Sender: TObject;
    var Key: Char);
begin
    if (Key='i') or (Key='İ') then Key := 'I';
    if (Key='l') or (Key='L') then Key := 'L';

end;

end.

```

Ölçüm Karşılık Ekranı

```

unit olcumKarsilikFrm;

interface

uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
    Grids, DBGrids, ExtCtrls, Mask, RzEdit, StdCtrls, RzPanel, RzButton;

type
    TolcumKarsilik = class(TForm)
        RzToolbar1: TRzToolbar;
        pnlArama: TRzPanel;
        aciklama: TRzMemo;
        resimPaneli: TPanel;
        olcumElemanResmi: TImage;
        DBGrid1: TDBGrid;
        RzToolButton3: TRzToolButton;
        RzToolButton4: TRzToolButton;
        RzToolButton5: TRzToolButton;
        elemanBilgi: TLabel;
        procedure RzToolButton2Click(Sender: TObject);
    end;

```

```

    procedure RzToolButton3Click(Sender: TObject);
    procedure RzToolButton5Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    olcumKarsilik: TolcumKarsilik;

implementation

uses dataModul, olcumKarsilikEkleFrm;

{$R *.DFM}

procedure TolcumKarsilik.RzToolButton2Click(Sender: TObject);
begin
    if d.olumKarsilikDQ.RecordCount > 0 then begin
        olcumKarsilikEkle.olumenDeger.Text := d.olumKarsilikDQolumenDeger.Text;
        olcumKarsilikEkle.olumKarsiligi.Text := d.olumKarsilikDQolumKarsiligi.Text;
        olcumKarsilikEkle.ekleDuzelt.Caption := 'duzelt';
        olcumKarsilikEkle.ShowModal;
    end;
end;

procedure TolcumKarsilik.RzToolButton3Click(Sender: TObject);
begin
    olcumKarsilikEkle.olumenDeger.Text := '';
    olcumKarsilikEkle.olumKarsiligi.Text := '';
    olcumKarsilikEkle.ekleDuzelt.Caption := 'ekle';
    olcumKarsilikEkle.ShowModal;

end;

procedure TolcumKarsilik.RzToolButton5Click(Sender: TObject);
begin

    if d.olumKarsilikDQ.RecordCount = 0 then exit;

    if (MessageDlg('Seçili Karşılığı silmek istediğinizden emin misiniz?', mtConfirmation, [mbYes,
mbNo], 0) = mrNo) then exit;

    with d.Query1 do begin
        close;
        sql.Clear;
        sql.Add('DELETE FROM olcumkarsilik WHERE
id='+d.olumKarsilikDQ.fieldbyname('id').AsString);
        ExecSQL;
    end;
    d.olumKarsilikDQ.Refresh;
end;
end.

```

Ölçüm Karşılığı Ekle/Düzeltil Ekranı

```
unit olcumKarsilikEkleFrm;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Mask, RZEdit, RzPanel, RzButton, ExtCtrls;

type
  TolcumKarsilikEkle = class(TForm)
    RzToolBar1: TRzToolBar;
    RzToolButton1: TRzToolButton;
    RzToolButton2: TRzToolButton;
    pnlArama: TRzPanel;
    Label5: TLabel;
    ekleDuzelt: TLabel;
    Label1: TLabel;
    olculenDeger: TRZEdit;
    olcumKarsiligi: TRZEdit;
    procedure RzToolButton2Click(Sender: TObject);
    procedure RzToolButton1Click(Sender: TObject);
    procedure olculenDegerKeyPress(Sender: TObject; var Key: Char);
    procedure FormActivate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  olcumKarsilikEkle: TolcumKarsilikEkle;

implementation

uses dataModul;

{$R *.DFM}

procedure TolcumKarsilikEkle.RzToolButton2Click(Sender: TObject);
begin
  close;
end;

procedure TolcumKarsilikEkle.RzToolButton1Click(Sender: TObject);
var
  query:string;
begin
  if olculenDeger.Text = '' then begin
    ShowMessage ('Ölçülen Değer Belirtilmemiş!');
    olculenDeger.SetFocus;
    exit;
  end;
end;
```

```

if olcumKarsiligi.Text = " then begin
    ShowMessage ('Ölçüm Karşılığı Belirtilmemiş!');
    olcumKarsiligi.SetFocus;
    exit;
end;
with d.Query2 do begin
    close;
    sql.Clear;
    if ekleDuzelt.Caption ='ekle' then
        query := 'select * from olcumkarsilik where
olcumElemanID='+d.olcumElemanDQ.fieldbyname('id').AsString+' and
olculenDeger='+olculenDeger.Text
    else
        query := 'select * from olcumkarsilik where
id<>'+d.olcumKarsilikDQ.fieldbyname('id').AsString+' and
olcumElemanID='+d.olcumElemanDQ.fieldbyname('id').AsString+' and
olculenDeger='+olculenDeger.Text;
    sql.Add(query);
    open;
    if RecordCount > 0 then begin
        ShowMessage (olculenDeger.Text+' Karşılığı daha önceden belirlenmiş!!!');
        olculenDeger.SetFocus;
        exit;
    end;
    close;
    sql.Clear;
    if ekleDuzelt.Caption ='ekle' then
        query := 'INSERT INTO olcumkarsilik SET '
    else
        query := 'UPDATE olcumkarsilik SET ';
        query := query +' olcumElemanID='+d.olcumElemanDQ.fieldbyname('id').AsString+',
olculenDeger='+olculenDeger.Text+', olcumKarsiligi='+olcumKarsiligi.Text;
        if ekleDuzelt.Caption ='duzelt' then query := query +' WHERE
id='+d.olcumKarsilikDQ.fieldbyname('id').AsString;
        SQL.Add(query);
        ExecSQL;
    end;
    d.olcumKarsilikDQ.Refresh;
    close;
end;

procedure TolcumKarsilikEkle.olculenDegerKeyPress(Sender: TObject;
var Key: Char);
begin
    if not (Key in [#8, '0'..'9']) then Key := #0;

end;

procedure TolcumKarsilikEkle.FormActivate(Sender: TObject);
begin
    olculenDeger.SetFocus;
end;
end.

```

Kart-Eleman Tanımları Ekranı

```
unit kartElemanTanimFrm;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, RzCmbxBx, RzPanel, RzButton, ExtCtrls, Grids, DBGrids;

type
  TkartElemanTanim = class(TForm)
    RzToolBar1: TRzToolBar;
    RzToolButton1: TRzToolButton;
    RzToolButton2: TRzToolButton;
    RzToolButton3: TRzToolButton;
    DBGrid1: TDBGrid;
    procedure RzToolButton1Click(Sender: TObject);
    procedure elemanDoldur;
    procedure RzToolButton2Click(Sender: TObject);
    procedure RzToolButton3Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  kartElemanTanim: TkartElemanTanim;

implementation

uses dataModul, elemanEkleFrm, kartElemanEkleFrm;

{$R *.DFM}
procedure TkartElemanTanim.elemanDoldur;
begin
  with d.elemanDQ do begin
    if not Active then open;

    kartElemanEkle.elemanID.items.Clear;
    kartElemanEkle.elemanID.Values.Clear;
    First;
    while not Eof do begin
      kartElemanEkle.elemanID.AddItemValue(fieldbyname('adi').Text,fieldbyname('id').Text);
      Next;
    end;
  end;
end;
end;
```

```

procedure TkartElemanTanim.RzToolButton1Click(Sender: TObject);
begin
    kartElemanEkle.kartNo.Text := "";
    kartElemanEkle.portNo.Text := "";
    kartElemanEkle.aciklama.Text := "";
    elemanDoldur;
    kartElemanEkle.elemanID.Text := "";
    kartElemanEkle.maxKapanmaSuresi.Text := "";
    kartElemanEkle.ekleDuzelt.Caption := 'ekle';
    kartElemanEkle.elemanID.OnChange (Sender);

    kartElemanEkle.showmodal;

end;

procedure TkartElemanTanim.RzToolButton2Click(Sender: TObject);
begin
    kartElemanEkle.kartNo.ItemIndex := d.kartElemanDQ.fieldbyname('kartNo').AsInteger-1;
    kartElemanEkle.portNo.ItemIndex := d.kartElemanDQ.fieldbyname('portNo').AsInteger-1;
    kartElemanEkle.aciklama.Text := d.kartElemanDQ.fieldbyname('aciklama').Text;
    kartElemanEkle.maxKapanmaSuresi.Text
:=d.kartElemanDQ.fieldbyname('maxKapanmaSuresi').Text;
    kartElemanEkle.kartNoKapat.ItemIndex := d.kartElemanDQ.fieldbyname('kartNoKapat').AsInteger-
1;
    kartElemanEkle.portNoKapat.ItemIndex := d.kartElemanDQ.fieldbyname('portNoKapat').AsInteger-
1;
    elemanDoldur;
    kartElemanEkle.elemanID.ItemIndex :=
kartElemanEkle.elemanID.Items.IndexOf(kartElemanTanim.DBGrid1.Columns[2].Field.Text);
    kartElemanEkle.elemanID.OnChange (Sender);
    kartElemanEkle.ekleDuzelt.Caption := 'duzelt';
    kartElemanEkle.showmodal;

end;

procedure TkartElemanTanim.RzToolButton3Click(Sender: TObject);
begin
    with d.Query1 do begin
        if d.kartElemanDQ.FieldByName('hareketDurumu').AsInteger=0 then begin
            close;
            sql.Clear;
            sql.Add('select * from elemantanim WHERE
elemanID='+d.kartElemanDQ.FieldByName('elemanID').AsString+' and
id!='+d.kartElemanDQ.FieldByName('id').AsString);
            open;

            if d.Query1.RecordCount = 0 then begin
                close;
                sql.Clear;
                sql.Add('select * from cikiskurallari where
elemanID1='+d.kartElemanDQ.FieldByName('elemanID').AsString+' or '+'
'elemanID2='+d.kartElemanDQ.FieldByName('elemanID').AsString+' or '+'
'elemanID3='+d.kartElemanDQ.FieldByName('elemanID').AsString+' or '+'
'elemanID4='+d.kartElemanDQ.FieldByName('elemanID').AsString+' or '+'
'elemanID5='+d.kartElemanDQ.FieldByName('elemanID').AsString);

```

```

open;

if d.Query1.RecordCount > 0 then begin
    ShowMessage('Bu eleman Çıkış Elemanları Kurallarında kullanılmış olduğundan
    Silinemez'+#13#10+'Öncelikle kurallardan silinmeli'+!!!');
    exit;
end;
end;
end else begin
    close;
    sql.Clear;
    sql.Add('select * from cikiskurallari WHERE
    cikisElemanID='+d.kartElemanDQ.FieldByName('id').AsString);
    open;

    if d.Query1.RecordCount > 0 then begin
        ShowMessage ('Bu eleman Çıkış Kurallarında kullanıldığından silinemez'+#13#10+'Öncelikle
        kurallardan silinmeli'+!');
        exit;
    end;
end;

if (MessageDlg('Tanımlı Elemanı silmek istediğinizden emin misiniz?', mtConfirmation, [mbYes,
mbNo], 0) = mrNo) then exit;
close;
sql.Clear;
sql.Add('DELETE from elemantanım WHERE
id='+d.kartElemanDQ.FieldByName('id').AsString);
ExecSQL;
end;
d.kartElemanDQ.Refresh;
end;

end.

```

Kart-Eleman Tanımı Ekle/Düzeltilme Ekranı

```

unit kartElemanEkleFrm;

interface

uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
    StdCtrls, RzCmboBx, RzPanel, RzButton, ExtCtrls, RzEdit, jpeg, Mask;

type
    TkartElemanEkle = class(TForm)
        RzToolBar1: TRzToolBar;
        RzToolButton1: TRzToolButton;
        RzToolButton2: TRzToolButton;
        pnlArama: TRzPanel;
        Label2: TLabel;
        Label1: TLabel;
        Label3: TLabel;
        ekleDuzelt: TLabel;
    end;

```

```

    elemanID: TRzComboBox;
    portNo: TRzComboBox;
    kartNo: TRzComboBox;
    Label4: TLabel;
    aciklama: TRzMemo;
    olcumElemanLbl: TLabel;
    olcumElemanID: TRzComboBox;
    olcumElemanResmi: TImage;
    olcumElemaniAciklama: TRzMemo;
    maxKapanmaSuresi: TRzEdit;
    sn: TLabel;
    maxKapanmaSuresiLbl: TLabel;
    kartNoLbl: TLabel;
    portNoLbl: TLabel;
    portNoKapat: TRzComboBox;
    kartNoKapat: TRzComboBox;
    procedure RzToolButton1Click(Sender: TObject);
    procedure RzToolButton2Click(Sender: TObject);
    procedure elemanIDChange(Sender: TObject);
    procedure olcumElemanIDChange(Sender: TObject);
    procedure maxKapanmaSuresiKeyPress(Sender: TObject; var Key: Char);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    kartElemanEkle: TkartElemanEkle;

implementation

uses dataModul, kartElemanTanimFrm;

{$R *.DFM}

procedure TkartElemanEkle.RzToolButton1Click(Sender: TObject);
var
    query,cikisDurumu,kartNoOff,portNoOff,cport,cportOff:string;
begin
    if (kartNo.ItemIndex<0) then begin
        ShowMessage('Kart No Belirtilmeli');
        kartNo.SetFocus;
        exit;
    end;
    if (portNo.ItemIndex<0) then begin
        ShowMessage('Port No Belirtilmeli');
        portNo.SetFocus;
        exit;
    end;
    if (elemanID.ItemIndex<0) then begin
        ShowMessage('Eleman Belirtilmeli');
        elemanID.SetFocus;
    end;
end;

```

```

    exit;
end;

if (elemanID.Value='1') and (portNo.Text<>'5') then begin
    ShowMessage('Rüzgar Hızı İçin Yanlızca 5 Numaralı Port Kullanılabilir');
    portNo.SetFocus;
    exit;
end;

cport := inttostr((kartNo.ItemIndex+1)*16);
cport := inttostr(strtoint(cport)+portNo.ItemIndex+1);

cikisDurumu := '0';

kartNoOff:="";
portNoOff:="";
cportOff:="";

with d.Query1 do begin
    close;
    sql.Clear;
    sql.Add('select hareketDurumu,inOu from eleman WHERE id='+elemanID.Value);
    Open;

    if d.Query1.RecordCount > 0 then begin
        if (fieldbyname('inOu').AsInteger = 0) and (portNo.Text>'5') then begin
            ShowMessage('Giriş için yalnızca 1-5 Numaralı Portlar Kullanılabilir');
            portNo.SetFocus;
            exit;
        end;
        if (fieldbyname('inOu').AsInteger = 1) and (portNo.Text<'6') then begin
            ShowMessage('Çıkış için yalnızca 6-15 Numaralı Portlar Kullanılabilir');
            portNo.SetFocus;
            exit;
        end;
        With d.Query2 do begin
            close;
            sql.Clear;
            query := 'SELECT * FROM elemantanım WHERE '+'
                '((kartNo='+kartNo.Text+' and portNo='+portNo.Text+') or '+'
                '(kartNoKapat='+kartNo.Text+' and portNoKapat='+portNo.Text+')) '+';
            if ekleDuzelt.Caption <>'ekle' then
                query := query +' and id<>'+d.kartElemanDQ.fieldbyname('id').Text;
            sql.Add(query);
            open;

            if d.Query2.RecordCount > 0 then begin
                ShowMessage(kartNo.Text+' nolu Kartın '+portNo.Text+' nolu Portu Daha Once
                Kullanılmış');
                portNo.SetFocus;
                exit;
            end;
        end;
    end;
end;

```

```

if (fieldbyname('inOu').AsInteger = 1) then begin
  if (fieldbyname('hareketDurumu').AsInteger=2) then begin
    if (fieldbyname('inOu').AsInteger = 0) and (portNoKapat.Text>'5') then begin
      ShowMessage('Giriş için yalnızca 1-5 Numaralı Portlar Kullanılabilir');
      portNoKapat.SetFocus;
      exit;
    end;
    if (fieldbyname('inOu').AsInteger = 1) and (portNoKapat.Text<'6') then begin
      ShowMessage('Çıkış için yalnızca 6-15 Numaralı Portlar Kullanılabilir');
      portNoKapat.SetFocus;
      exit;
    end;

    if (kartNo.Text=kartNoKapat.Text) and (portNo.Text=portNoKapat.Text) then begin
      ShowMessage('Açma ve Kapatma Portları Aynı Olamaz');
      portNoKapat.SetFocus;
      exit;
    end;

    kartNoOff := kartNoKapat.Text;
    portNoOff := portNoKapat.Text;
    cportOff := inttostr((kartNoKapat.ItemIndex+1)*16);
    cportOff := inttostr(strtoint(cportOff)+portNoKapat.ItemIndex+1);

    With d.Query2 do begin
      close;
      sql.Clear;
      query := 'SELECT * FROM elemantanim WHERE '+
        '((kartNo='+kartNoKapat.Text+' and portNo='+portNoKapat.Text+') or '+
        '(kartNoKapat='+kartNoKapat.Text+' and portNoKapat='+portNoKapat.Text+')) ';
      if ekleDuzelt.Caption <> 'ekle' then
        query := query + ' and id<>'+d.kartElemanDQ.fieldbyname('id').Text;
      sql.Add(query);
      open;

      if d.Query2.RecordCount > 0 then begin
        ShowMessage(kartNoKapat.Text+' nolu Kartın '+portNoKapat.Text+' nolu Portu Daha
Once Kullanılmış');
        portNoKapat.SetFocus;
        exit;
      end;

    end;

    end;
    end;
    cikisDurumu := fieldbyname('hareketDurumu').AsString;

  end else begin
    if (olcumElemanID.ItemIndex<0) then begin
      ShowMessage('Ölçüm Elemanı Belirtilmeli');
      olcumElemanID.SetFocus;
      exit;
    end;
  end;
end;

```

```

close;
sql.Clear;
if ekleDuzelt.Caption ='ekle' then
    query := 'INSERT INTO elemanTanim SET '
else
    query := 'UPDATE elemanTanim SET ';

    query := query +' kartNo='+kartNo.Text+',portNo='+portNo.Text+',
maxKapanmaSuresi="'+maxKapanmaSuresi.Text+"',
hareketDurumu="'+cikisDurumu+"',kartNoKapat="'+kartNoOff+"',portNoKapat="'+portNoOff+"',cNo
Kapat="'+cportOff+"',cNo='+cport+', elemanID="'+elemanID.Value+"',
olcumElemanID="'+olcumElemanID.Value+"', aciklama="'+aciklama.Text+"';

    if ekleDuzelt.Caption ='duzelt' then query := query +' WHERE
id='+d.kartElemanDQ.fieldbyname('id').AsString;
    SQL.Add(query);
    ExecSQL;
end;
d.kartElemanDQ.close;
d.kartElemanDQ.open;
close;
end;

procedure TkartElemanEkle.RzToolButton2Click(Sender: TObject);
begin
    close;
end;

procedure TkartElemanEkle.elemanIDChange(Sender: TObject);
var
    secilenIndex,indexSay,hareketDurumu:integer;
    elemanTipi,olcumElemani,Query:String;
begin
    if kartElemanEkle.elemanID.ItemIndex < 0 then begin
        maxKapanmaSuresi.Visible := false;
        maxKapanmaSuresiLbl.Visible := false;
        sn.Visible := false;
        portNoKapat.Visible := false;
        portNoLbl.Visible := false;
        kartNoKapat.Visible := false;
        kartNoLbl.Visible := false;
        kartElemanEkle.olcumElemanID.Visible := false;
        kartElemanEkle.olcumElemanLbl.Visible := false;
        kartElemanEkle.Height := 180;

        exit;
    end;

    if (kartElemanEkle.kartNo.ItemIndex < 0) or (kartElemanEkle.portNo.ItemIndex < 0) then begin
        ShowMessage('Öncelikle Kart ve Port Numarası Seçilmeli');
        elemanID.ItemIndex := -1;
        kartNo.SetFocus;
        exit;
    end;

```

```

end;
secilenIndex := -1;
with d.Query2 do begin
  close;
  SQL.Clear;
  sql.Add('select * from eleman e WHERE e.id='+kartElemanEkle.elemanID.Value+' limit 0,1');
  open;

  elemanTipi := d.Query2.fieldbyname('inOu').Text;
  try
    hareketDurumu := d.Query2.fieldbyname('hareketDurumu').AsInteger;
  except
  end;

  kartElemanEkle.olcumElemanID.items.Clear;
  kartElemanEkle.olcumElemanID.Values.Clear;
  close;
  SQL.Clear;
  Query := 'select e.adi,e.inOu,t.* from eleman e '+
    ' LEFT JOIN elemantanim t ON e.id=t.elemanID'+
    ' WHERE e.id='+kartElemanEkle.elemanID.Value+' and
t.kartNo='+kartElemanEkle.kartNo.Value+
    ' and t.portNo='+kartElemanEkle.portNo.Value;
  sql.Add(Query);
  open;
  olcumElemani := fieldbyname('olcumElemanID').Text;
  try
    secilenIndex := fieldbyname('hareketDurumu').AsInteger;
  except
    secilenIndex := -1;
  end;

  if (elemanTipi = '0') then begin
    kartElemanEkle.olcumElemanID.Visible := true;
    kartElemanEkle.olcumElemanLbl.Visible := true;
    kartElemanEkle.Height := 265;
    maxKapanmaSuresi.Visible := false;
    maxKapanmaSuresiLbl.Visible := false;
    sn.Visible := false;
    portNoKapat.Visible := false;
    portNoLbl.Visible := false;
    kartNoKapat.Visible := false;
    kartNoLbl.Visible := false;

    close;
    SQL.Clear;
    Query := 'select *,e.birimi from olcumeleman o, eleman e WHERE e.id=o.elemanID and
elemenID='+kartElemanEkle.elemanID.Value+' order by elemanBilgi';
    sql.Add(Query);
    open;

    First;
    indexSay := -1;
    while not Eof do begin
      indexSay := indexSay + 1;

```

```

        if (olcumElemani = d.Query2.fieldbyname('id').Text) then secilenIndex := indexSay;

kartElemanEkle.olcumElemanID.AddItemValue(fieldbyname('elemanBilgi').Text,fieldbyname('id').Text);
    Next;
end;
kartElemanEkle.olcumElemanID.ItemIndex := secilenIndex;
end else begin
    kartElemanEkle.olcumElemanLbl.Visible := false;
    kartElemanEkle.olcumElemanID.Visible := false;
    if hareketDurumu > 1 then begin
        maxKapanmaSuresi.Visible := true;
        maxKapanmaSuresiLbl.Visible := true;
        sn.Visible := true;
        portNoKapat.Visible := true;
        portNoLbl.Visible := true;
        kartNoKapat.Visible := true;
        kartNoLbl.Visible := true;
        kartElemanEkle.Height := 250;
    end else begin
        maxKapanmaSuresi.Visible := false;
        maxKapanmaSuresiLbl.Visible := false;
        sn.Visible := false;
        portNoKapat.Visible := false;
        portNoLbl.Visible := false;
        kartNoKapat.Visible := false;
        kartNoLbl.Visible := false;
        kartElemanEkle.Height := 180;
    end;

end;

kartElemanEkle.olcumElemanID.OnChange(Sender);
end;
end;

procedure TkartElemanEkle.olcumElemanIDChange(Sender: TObject);
begin
    if olcumElemanID.Value<>" then begin
        kartElemanEkle.olcumElemanResmi.Visible := true;
        kartElemanEkle.olcumElemaniAciklama.Visible := true;
        kartElemanEkle.olcumElemanLbl.Visible := true;
        with d.Query2 do begin
            close;
            SQL.Clear;
            sql.Add('select * from olcumeleman WHERE id='+olcumElemanID.Value+' limit 0,1');
            open;

            olcumElemaniAciklama.Text:=fieldbyname('aciklama').Text;
            if fieldbyname('resim').Text<>" then begin
                if FileExists(fieldbyname('resim').Text) then
                    olcumElemanResmi.Picture.LoadFromFile(fieldbyname('resim').Text);
            end else
                kartElemanEkle.olcumElemanResmi.Picture := nil;
        end;
    end;
end;

```

```

end;
end else begin
    kartElemanEkle.olcumElemanResmi.Visible := false;
    kartElemanEkle.olcumElemanResmi.Picture := nil;
    kartElemanEkle.olcumElemaniAciklama.Visible := false;
    kartElemanEkle.olcumElemaniAciklama.Text := "";
end;

end;

procedure TkartElemanEkle.maxKapanmaSuresiKeyPress(Sender: TObject;
    var Key: Char);
begin
    if not (Key in [#8, '0'..'9']) then Key := #0;

end;

end.

```

Otomatik Kontrol Tanımları Ekranı

```

unit cikisTanimFrm;

interface

uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
    Grids, DBGrids, RzButton, ExtCtrls, RzPanel, StdCtrls, RzCmboBx, Mask,
    RzEdit, RzRadChk;

type
    TcikisKuralTanim = class(TForm)
        RzToolBar1: TRzToolBar;
        RzToolButton1: TRzToolButton;
        RzToolButton2: TRzToolButton;
        RzToolButton3: TRzToolButton;
        DBGrid1: TDBGrid;
        Panel1: TPanel;
        elemanID: TRzComboBox;
        Label2: TLabel;
        procedure RzToolButton1Click(Sender: TObject);
        procedure olcumElemanDoldur;
        procedure RzToolButton2Click(Sender: TObject);
        function isaretSartKarsiligi(gelenSart:string):integer;
        procedure elemanIDChange(Sender: TObject);
        procedure DBGrid1DrawColumnCell(Sender: TObject; const Rect: TRect;
            DataCol: Integer; Column: TColumn; State: TGridDrawState);
        procedure RzToolButton3Click(Sender: TObject);
    private
        { Private declarations }
    public

```

```

    { Public declarations }
end;

var
    cikisKuralTanim: TcikisKuralTanim;

implementation

uses dataModul, cikisKuralEkleFrm;

{$R *.DFM}

function TcikisKuralTanim.isaretSartKarsiligi(gelenSart:string):integer;
begin
    if gelenSart='' then Result := 0;
    if gelenSart!='' then Result := 1;
    if gelenSart='>' then Result := 2;
    if gelenSart='>=' then Result := 3;
    if gelenSart='<' then Result := 4;
    if gelenSart='<=' then Result := 5;

end;

procedure TcikisKuralTanim.olcumElemanDoldur;
var
    sayac:integer;
begin
    with d.Query2 do begin
        close;
        sql.Clear;
        sql.Add('select * from eleman WHERE inOu="0" and aktif="1" ');
        open;

        for sayac := 1 to 5 do begin
            TRzComboBox(cikisKuralEkle.FindComponent('elemanID'+inttostr(sayac))).items.Clear;
            TRzComboBox(cikisKuralEkle.FindComponent('elemanID'+inttostr(sayac))).Values.Clear;
        end;
        while not Eof do begin
            for sayac := 1 to 5 do begin
                TRzComboBox(cikisKuralEkle.FindComponent('elemanID'+inttostr(sayac))).AddItemValue(fieldbyname('adi').Text,fieldbyname('id').Text);
            end;
            Next;
        end;

        close;
        sql.Clear;
        sql.Add('select * from elemantanim WHERE id='+cikisKuralTanim.elemanID.Value);
        open;

        if d.Query2.RecordCount > 0 then begin
            cikisKuralEkle.cikisData.items.Clear;
            cikisKuralEkle.cikisData.Values.Clear;
            if (FieldByName('hareketDurumu').AsInteger=1) then begin
                cikisKuralEkle.cikisData.AddItemValue('Kapat','1');
            end;
        end;
    end;
end;

```

```

        cikisKuralEkle.cikisData.AddItemValue('Aç','2');
    end else begin
        cikisKuralEkle.cikisData.AddItemValue('Tam Kapat','3');
        cikisKuralEkle.cikisData.AddItemValue('%10 Aç','4');
        cikisKuralEkle.cikisData.AddItemValue('%20 Aç','5');
        cikisKuralEkle.cikisData.AddItemValue('%30 Aç','6');
        cikisKuralEkle.cikisData.AddItemValue('%40 Aç','7');
        cikisKuralEkle.cikisData.AddItemValue('%50 Aç','8');
        cikisKuralEkle.cikisData.AddItemValue('%60 Aç','9');
        cikisKuralEkle.cikisData.AddItemValue('%70 Aç','10');
        cikisKuralEkle.cikisData.AddItemValue('%80 Aç','11');
        cikisKuralEkle.cikisData.AddItemValue('%90 Aç','12');
        cikisKuralEkle.cikisData.AddItemValue('Tam Aç','13');
    end;
end;
end;
end;

procedure TcikisKuralTanim.RzToolButton1Click(Sender: TObject);
var
    sayac:integer;
begin
    if elemanID.Value<'0' then begin
        ShowMessage('Kural Ekleyebilmek için Öncelikle bir Çıkış Elemanı Kuralı Seçmelisiniz');
        elemanID.SetFocus;
        exit;
    end;
    cikisKuralEkle.kuralAdi.Text := "";

    for sayac:=1 to 5 do begin
        TRzComboBox(cikisKuralEkle.FindComponent('elemanID'+inttostr(sayac))).ItemIndex := -1;
        TRzComboBox(cikisKuralEkle.FindComponent('sart'+inttostr(sayac))).ItemIndex := 0;
        TRzEdit(cikisKuralEkle.FindComponent('deger'+inttostr(sayac))).Text := "";
        TLabel(cikisKuralEkle.FindComponent('birimi'+inttostr(sayac))).Caption := "";

    end;
    cikisKuralEkle.ve1.Checked :=false;
    cikisKuralEkle.cikisData.ItemIndex := -1;
    cikisKuralEkle.aktif.ItemIndex := 0;
    cikisKuralEkle.elemanBilgi.Caption := elemanID.Text;
    cikisKuralEkle.ekleDuzelt.Caption := 'ekle';
    cikisKuralEkle.cikisElemanID.Caption := elemanID.Value;

    olcumElemanDoldur;
    cikisKuralEkle.ShowModal;
end;

procedure TcikisKuralTanim.RzToolButton2Click(Sender: TObject);
var
    sayac,itemSay:integer;
    secElemanID,secSart:TRzComboBox;
    secDeger:TRzEdit;
    secVe : TRzCheckBox;

begin

```

```

if d.cikisKurallariDQ.RecordCount = 0 then exit;

if elemanID.Value<'0' then begin
    ShowMessage('Kural DüzenliyeBilmek içim Öncelikle bir Çıkış Elemanı Kuralı Seçmelisiniz');
    elemanID.SetFocus;
    exit;
end;

olcumElemanDoldur;

cikisKuralEkle.ve1.Checked := false;
for sayac:=1 to 5 do begin
    TLabel(cikisKuralEkle.FindComponent('birimi'+inttostr(sayac))).Caption:='';
    TRzEdit(cikisKuralEkle.FindComponent('deger'+inttostr(sayac))).Text :='';
end;

with d.cikisKurallariDQ do begin

    cikisKuralEkle.kuralAdi.Text := fieldByName('kuralAdi').Text;

    for sayac:=1 to 5 do begin
        secElemanID :=TRzComboBox(cikisKuralEkle.FindComponent('elemanID'+inttostr(sayac)));
        secSart := TRzComboBox(cikisKuralEkle.FindComponent('sart'+inttostr(sayac)));
        secDeger := TRzEdit(cikisKuralEkle.FindComponent('deger'+inttostr(sayac)));
        secVe := TRzCheckBox(cikisKuralEkle.FindComponent('ve'+inttostr(sayac-1)));

        if fieldByName('elemanID'+inttostr(sayac)).Text<>" then begin
            if sayac>1 then begin
                secVe.Enabled := true;
                secVe.checked := true;
            end;

            for itemSay:=0 to secElemanID.Count do begin
                secElemanID.ItemIndex := itemSay;
                if secElemanID.Value = fieldByName('elemanID'+inttostr(sayac)).Text then Break;
            end;

            with d.query1 do begin
                close;
                sql.clear;
                sql.add('select birimi from eleman WHERE
id='+d.cikisKurallariDQ.fieldByName('elemanID'+inttostr(sayac)).text);
                open;

                TLabel(cikisKuralEkle.FindComponent('birimi'+inttostr(sayac))).Caption:=fieldbyname('birimi').Text
                ;
                end;

            end else
                secElemanID.ItemIndex := -1;

            if fieldByName('sart'+inttostr(sayac)).Text<>" then
                secSart.ItemIndex := isaretSartKarsiligi(fieldByName('sart'+inttostr(sayac)).text);

```

```

    if fieldByName('deger'+inttostr(sayac)).Text<>" then
        secDeger.Text := fieldByName('deger'+inttostr(sayac)).Text;

end;

if fieldByName('cikisData').AsInteger <3 then
    cikisKuralEkle.cikisData.ItemIndex := fieldByName('cikisData').AsInteger-1
else
    cikisKuralEkle.cikisData.ItemIndex := fieldByName('cikisData').AsInteger-3;

cikisKuralEkle.elemanBilgi.Caption := elemanID.Text;
cikisKuralEkle.ekleDuzelt.Caption := 'duzelt';
cikisKuralEkle.cikisElemanID.Caption := elemanID.Value;

    if fieldbyname('aktif').AsString ='1' then
        cikisKuralEkle.aktif.ItemIndex :=0
    else
        cikisKuralEkle.aktif.ItemIndex :=1;
end;
cikisKuralEkle.ShowModal;

end;

procedure TcikisKuralTanim.elemanIDChange(Sender: TObject);
begin
    with d.cikisKurallariDQ do begin
        close;
        sql.Clear;
        if elemanID.Value = '-1' then
            sql.Add('select * from cikiskurallari WHERE aktif="1"')
        else if elemanID.Value = '-2' then
            sql.Add('select * from cikiskurallari WHERE aktif="0"')
        else if elemanID.Value = '-3' then
            sql.Add('select * from cikiskurallari')
        else
            sql.Add('select * from cikiskurallari WHERE cikisElemanID="'+elemanID.Value+'");

        open;
    end;
end;

procedure TcikisKuralTanim.DBGrid1DrawColumnCell(Sender: TObject;
    const Rect: TRect; DataCol: Integer; Column: TColumn;
    State: TGridDrawState);
var
    deger :string;
begin
    deger := "";
    if datacol=1 then begin
        if Column.Field.AsString='0' then
            deger:='Pasif'
        else if Column.Field.AsString='1' then
            deger:='Aktif';
        DBGrid1.Canvas.TextRect(Rect,Rect.Left+4,Rect.top+2,deger);
    end;
end;

```

```

end;
end;

procedure TcikisKuralTanim.RzToolButton3Click(Sender: TObject);
begin
    if d.cikisKurallariDQ.RecordCount = 0 then exit;

    if (MessageDlg('Seçili Kuralı Silmek istediğinizden emin misiniz?', mtConfirmation, [mbYes,
mbNo], 0) = mrNo) then exit;

    with d.Query1 do begin
        close;
        sql.Clear;
        sql.Add('DELETE FROM cikiskurallari WHERE
id='+d.cikisKurallariDQ.fieldbyname('id').AsString);
        ExecSQL;
    end;

    d.cikisKurallariDQ.Refresh;
end;
end.

```

Otomatik Kontrol Kuralları Ekle/Düzeltilme Ekranı

```

unit cikisKuralEkleFrm;

interface

uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
    RzButton, RzEdit, StdCtrls, Mask, RzCmboBx, RzPanel, ExtCtrls, RzRadChk,
    RzBckgnd, RzLabel, dxCntner, dxEditor, dxExEdtr, dxEdLib;

type
    TcikisKuralEkle = class(TForm)
        RzToolbar1: TRzToolbar;
        RzToolButton1: TRzToolButton;
        RzToolButton2: TRzToolButton;
        pnlArama: TRzPanel;
        Label2: TLabel;
        ekleDuzelt: TLabel;
        Label4: TLabel;
        kuralAdi: TRzEdit;
        aktif: TRzComboBox;
        elemanBilgi: TLabel;
        elemanID1: TRzComboBox;
        sart1: TRzComboBox;
        Label1: TLabel;
        Label3: TLabel;
        Label5: TLabel;
        RzSeparator1: TRzSeparator;
        ve1: TRzCheckBox;
        elemanID2: TRzComboBox;
        sart2: TRzComboBox;
        RzSeparator2: TRzSeparator;
    end;

```

```

ve2: TRzCheckBox;
elemanID3: TRzComboBox;
sart3: TRzComboBox;
RzSeparator3: TRzSeparator;
ve3: TRzCheckBox;
RzSeparator4: TRzSeparator;
birimi1: TLabel;
birimi2: TLabel;
birimi3: TLabel;
birimi4: TLabel;
birimi5: TLabel;
cikisData: TRzComboBox;
Label6: TLabel;
deger1: TRzEdit;
deger2: TRzEdit;
deger3: TRzEdit;
deger4: TRzEdit;
deger5: TRzEdit;
elemanID5: TRzComboBox;
sart5: TRzComboBox;
ve4: TRzCheckBox;
sart4: TRzComboBox;
elemanID4: TRzComboBox;
cikisElemanID: TLabel;
procedure ve1Click(Sender: TObject);
procedure elemanID1Change(Sender: TObject);
procedure RzToolButton1Click(Sender: TObject);
procedure deger1KeyPress(Sender: TObject; var Key: Char);
procedure RzToolButton2Click(Sender: TObject);
function sartIsaretKarsiligi(gelenSart:string):String;
procedure kuralAdiKeyPress(Sender: TObject; var Key: Char);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    cikisKuralEkle: TcikisKuralEkle;

implementation

uses dataModul, cikisTanimFrm;

{$R *.DFM}
function TcikisKuralEkle.sartIsaretKarsiligi(gelenSart:string):String;
begin
    if gelenSart='eşittir' then Result := '=';
    if gelenSart='eşit değildir' then Result := '!=';
    if gelenSart='büyüktür' then Result := '>';
    if gelenSart='büyük veya eşittir' then Result := '>=';
    if gelenSart='küçüktür' then Result := '<';
    if gelenSart='küçük veya eşittir' then Result := '<=';
end;

```

```

procedure TcikisKuralEkle.ve1Click(Sender: TObject);
var
sayac:integer;
begin
if TRzCheckBox(FindComponent(TRzCheckBox(Sender).name)).Checked then begin
    TRzComboBox(FindComponent('elemanID'+inttostr(TRzCheckBox(Sender).Tag+1))).Enabled :=
true;
    TRzComboBox(FindComponent('sart'+inttostr(TRzCheckBox(Sender).Tag+1))).Enabled := true;
    TRzEdit(FindComponent('deger'+inttostr(TRzCheckBox(Sender).Tag+1))).Enabled := true;
    if FindComponent('ve'+inttostr(TRzCheckBox(Sender).Tag+1))<>nil then
        TRzCheckBox(FindComponent('ve'+inttostr(TRzCheckBox(Sender).Tag+1))).Enabled := true;
end else begin
    for sayac:= TRzCheckBox(Sender).Tag+1 to 5 do begin
        TRzComboBox(FindComponent('elemanID'+inttostr(sayac))).Enabled := false;
        TRzComboBox(FindComponent('sart'+inttostr(sayac))).Enabled := false;
        TRzEdit(FindComponent('deger'+inttostr(sayac))).Enabled := false;
        if FindComponent('ve'+inttostr(sayac))<>nil then begin
            TRzCheckBox(FindComponent('ve'+inttostr(sayac))).Enabled := false;
            TRzCheckBox(FindComponent('ve'+inttostr(sayac))).Checked := false;
        end;
    end;
end;

end;

procedure TcikisKuralEkle.elemanID1Change(Sender: TObject);
var
    query:string;
    birimLabel : TLabel;
begin
    birimLabel := TLabel( FindComponent( 'birimi'+inttostr(TRzComboBox(Sender).Tag) ) );
    with d.query1 do begin
        close;
        sql.clear;
        query := 'select birimi from eleman WHERE id='+TRzComboBox(Sender).value;
        sql.add(query);
        open;
        birimLabel.Caption:=fieldbyname('birimi').Text;
    end;
end;

procedure TcikisKuralEkle.RzToolButton1Click(Sender: TObject);
var
sayac:integer;
query:String;
secilenEleman,secilenSart:TRzComboBox;
secileninDeger:TRzEdit;
begin
if kuralAdi.Text = " then begin
    ShowMessage ('Kural Adı Belirtilmemiş!');
    kuralAdi.SetFocus;
    exit;

```

```

end;
with d.Query1 do begin
  close;
  sql.Clear;
  if ekleDuzelt.Caption ='ekle' then
    query := 'INSERT INTO cikiskurallari SET'
  else
    query := 'UPDATE cikiskurallari SET';

  query := query+' kuralAdi="'+kuralAdi.Text+'";

for sayac:= 1 to 5 do begin
  secilenEleman := TRzComboBox(cikisKuralEkle.FindComponent('elemanID'+inttostr(sayac)));
  secilenSart := TRzComboBox(cikisKuralEkle.FindComponent('sart'+inttostr(sayac)));
  secileninDeger := TRzEdit(cikisKuralEkle.FindComponent('deger'+inttostr(sayac)));
  if secilenEleman.Enabled then begin
    if secilenEleman.ItemIndex < 0 then begin
      ShowMessage ('Ölçüm elemanı belirtilmemiş!');
      secilenEleman.SetFocus;
      exit;
    end;
    if secileninDeger.Text = " then begin
      ShowMessage ('Ölçüm değeri belirtilmemiş!');
      secileninDeger.SetFocus;
      exit;
    end;
    query := query + ', elemanID'+IntToStr(sayac)+'='+secilenEleman.Value+',
sart'+IntToStr(sayac)+'='+sartIsaretKarsiligi(secilenSart.Text)+'',
deger'+IntToStr(sayac)+'='+secileninDeger.Text;
  end else
    query := query + ', elemanID'+IntToStr(sayac)+'=null, sart'+IntToStr(sayac)+'=null,
deger'+IntToStr(sayac)+'=null';
  end;

  query := query+', cikisElemanID="'+cikisElemanID.Caption+'";

  if cikisData.ItemIndex < 0 then begin
    ShowMessage ('Yapılacak Hareket Belirtilmemiş!');
    cikisData.SetFocus;
    exit;
  end;
  query := query+', cikisData="'+cikisData.Value+'";
  query := query+', aktif="'+aktif.Value+'";

  if ekleDuzelt.Caption ='duzelt' then query := query +' WHERE
id='+d.cikisKurallariDQ.fieldbyname('id').AsString;
  sql.Add (query);
  ExecSQL;
end;
d.cikisKurallariDQ.Refresh;
close;
end;

procedure TcikisKuralEkle.deger1KeyPress(Sender: TObject; var Key: Char);

```

```

begin
  if not (Key in [#8, '0'..'9', DecimalSeparator]) then Key := #0;
end;

procedure TcikisKuralEkle.RzToolButton2Click(Sender: TObject);
begin
  close;
end;

procedure TcikisKuralEkle.kuralAdiKeyPress(Sender: TObject; var Key: Char);
begin
  if (Key='i') or (Key='İ') then Key := 'I';
  if (Key='ı') or (Key='I') then Key := 'I';
end;

end.

```

Elle Kontrol Ekranı

```

unit elleKontrolFrm;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, RzCmboBx, RzPanel, RzButton, ExtCtrls;

type
  TelleKontrol = class(TForm)
    RzToolbar1: TRzToolbar;
    RzToolButton1: TRzToolButton;
    otoKontrol: TRzToolButton;
    pnlArama: TRzPanel;
    Label6: TLabel;
    cikisData: TRzComboBox;
    elemanID: TRzComboBox;
    Label1: TLabel;
    elemanBilgi: TLabel;
    RzToolButton3: TRzToolButton;
    cNo: TLabel;
    cNoKapat: TLabel;
    maxKapanmaSuresi: TLabel;
    procedure RzToolButton3Click(Sender: TObject);
    procedure elemanIDChange(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure RzToolButton1Click(Sender: TObject);
    procedure otoKontrolClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  elleKontrol: TelleKontrol;

```

implementation

uses dataModul, anaFormFrm, bekleFrm;

{ \$R *.DFM }

```
procedure TelleKontrol.RzToolButton3Click(Sender: TObject);
begin
  close;
end;
```

```
procedure TelleKontrol.elemanIDChange(Sender: TObject);
var
  deger:string;
begin
```

```
  with d.Query2 do begin
    close;
    sql.Clear;
    sql.Add('select * from elemantanım WHERE id='+elleKontrol.elemanID.Value);
    open;
```

```
  if d.Query2.RecordCount > 0 then begin
    cNo.Caption := FieldByName('cNo').AsString;
    cNoKapat.Caption := FieldByName('cNoKapat').AsString;
    maxKapanmaSuresi.Caption := FieldByName('maxKapanmaSuresi').AsString;
```

```
    elleKontrol.cikisData.items.Clear;
    elleKontrol.cikisData.Values.Clear;
    if (FieldByName('hareketDurumu').AsInteger=1) then begin
      elleKontrol.cikisData.AddItemValue('Kapat','1');
      elleKontrol.cikisData.AddItemValue('Aç','2');
    end else begin
      elleKontrol.cikisData.AddItemValue('Tam Kapat','3');
      elleKontrol.cikisData.AddItemValue('%10 Aç','4');
      elleKontrol.cikisData.AddItemValue('%20 Aç','5');
      elleKontrol.cikisData.AddItemValue('%30 Aç','6');
      elleKontrol.cikisData.AddItemValue('%40 Aç','7');
      elleKontrol.cikisData.AddItemValue('%50 Aç','8');
      elleKontrol.cikisData.AddItemValue('%60 Aç','9');
      elleKontrol.cikisData.AddItemValue('%70 Aç','10');
      elleKontrol.cikisData.AddItemValue('%80 Aç','11');
      elleKontrol.cikisData.AddItemValue('%90 Aç','12');
      elleKontrol.cikisData.AddItemValue('Tam Aç','13');
    end;
```

```
  end;
  close;
  sql.Clear;
  sql.Add('select * from cikisDurum WHERE tanimliElemanID='+elemanID.Value+' and id=(select
max(id) from cikisDurum WHERE tanimliElemanID='+elemanID.Value+')');
  open;
```

```
  elemanBilgi.Caption := "";
  if d.Query2.RecordCount > 0 then begin
```

```

deger := "";
if FieldByName('cikisData').AsString='0' then
    deger:='Giriş Elemanı'
else if FieldByName('cikisData').AsString='1' then
    deger:='Kapalı'
else if FieldByName('cikisData').AsString='2' then
    deger:='Açık'
else if FieldByName('cikisData').AsString='3' then
    deger:='Tam Kapalı'
else if FieldByName('cikisData').AsString='4' then
    deger:='%10 Açık'
else if FieldByName('cikisData').AsString='5' then
    deger:='%20 Açık'
else if FieldByName('cikisData').AsString='6' then
    deger:='%30 Açık'
else if FieldByName('cikisData').AsString='7' then
    deger:='%40 Açık'
else if FieldByName('cikisData').AsString='8' then
    deger:='%50 Açık'
else if FieldByName('cikisData').AsString='9' then
    deger:='%60 Açık'
else if FieldByName('cikisData').AsString='10' then
    deger:='%70 Açık'
else if FieldByName('cikisData').AsString='11' then
    deger:='%80 Açık'
else if FieldByName('cikisData').AsString='12' then
    deger:='%90 Açık'
else if FieldByName('cikisData').AsString='13' then
    deger:='Tam Açık'
else
    deger:='Tanımsız';

elemanBilgi.Caption := 'Durumu: '+deger;

if FieldByName('kuralID').AsInteger > 0 then begin
    elemanBilgi.Caption := elemanBilgi.Caption + ' (Otomatik)';
    otoKontrol.Enabled := false;
end else begin
    elemanBilgi.Caption := elemanBilgi.Caption + ' (Elle)';
    if FieldByName('kuralID').AsInteger = 0 then otoKontrol.Enabled := true;
end;

end;
end;
end;

procedure TelleKontrol.FormActivate(Sender: TObject);
begin
    elleKontrol.elemanID.SetFocus;
end;

procedure TelleKontrol.RzToolButton1Click(Sender: TObject);
var
    query:string;
    sayac:integer;

```

```

beklemeZamani:real;
begin
  if elemanID.ItemIndex < 0 then begin
    ShowMessage ('Çıkış İşlemi Seçilmemiş!');
    elemanID.SetFocus;
    exit;
  end;

  if cikisData.ItemIndex < 0 then begin
    ShowMessage ('Yapılacak Hareket Belirtilmemiş!');
    cikisData.SetFocus;
    exit;
  end;

  with application do if findcomponent('bekleForm')=NIL then CreateForm(TbekleForm,bekleForm);
  bekleForm.Show;
  bekleForm.Update;

  if anaForm.ComPort.Connected then begin
    sayac :=0;
    while (sayac<3000) do begin
      Application.ProcessMessages;
      Sleep(20);
      sayac := sayac+1;
      if not anaForm.ComPort.Connected then begin
        anaForm.ComPort.Connected := true;
        anaForm.Timer1.Enabled := false;
        Break;
      end;
    end;
  end else begin
    anaForm.ComPort.Connected := true;
    anaForm.Timer1.Enabled := false;
  end;

  if not anaForm.ComPort.Connected then begin
    bekleForm.Hide;
    bekleForm.free;
    ShowMessage ('Seri İletişim Portu Meşgul Olduğundan İşleminiz Yapılamadı!');
    exit;
  end;

  with d.Query2 do begin
    close;
    sql.Clear;
    query := 'SELECT max(islemZamani) as islemZamani FROM cikisDurum WHERE
    tanimliElemanID='+elemanID.Value+' and cikisData='+cikisData.Value+' and id=(select max(id)
    from cikisDurum WHERE tanimliElemanID='+elemanID.Value+')';
    sql.Add(query);
    open;
  end;

  if d.Query2.FieldName('islemZamani').Text ='' then begin

```

```

anaForm.comPortOkunan := -1;
if cikisData.Value='1' then begin
  anaForm.ComPort.WriteStr(cNo.Caption);
end else if cikisData.Value='2' then begin
  anaForm.ComPort.WriteStr(chr(strtoint(cNo.Caption)+128));
end else if cikisData.Value='3' then begin
  anaForm.ComPort.WriteStr(chr(strtoint(cNoKapat.Caption)+128));
  sayac :=0;
  while (sayac<100) do begin
    Application.ProcessMessages;
    Sleep(10*strtoint(maxKapanmaSuresi.Caption));
    sayac:=sayac+1;
  end;
  anaForm.ComPort.WriteStr(cNoKapat.Caption);
end else if cikisData.Value>'3' then begin
  anaForm.ComPort.WriteStr(chr(strtoint(cNoKapat.Caption)+128));
  sayac :=0;
  while (sayac<100)do begin
    Application.ProcessMessages;
    Sleep(10*strtoint(maxKapanmaSuresi.Caption));
    sayac:=sayac+1;
  end;
  anaForm.ComPort.WriteStr(cNoKapat.Caption);

  sayac :=0;
  while (sayac<100)do begin
    Application.ProcessMessages;
    Sleep(10);
    sayac:=sayac+1;
  end;
  anaForm.ComPort.WriteStr(chr(strtoint(cNo.Caption)+128));

  sayac:=strtoint(maxKapanmaSuresi.Caption);
  beklemeZamani := round((sayac-3)*10/100*10);
  sayac :=0;
  while (sayac<100)do begin
    Application.ProcessMessages;
    Sleep( round(beklemeZamani)) ;
    sayac:=sayac+1;
  end;
  anaForm.ComPort.WriteStr(cNo.Caption);
end;

sayac :=0;
while (sayac<150) do begin
  Application.ProcessMessages;
  Sleep(20);
  sayac := sayac+1;
end;
end;

with d.Query2 do begin
  close;
  sql.Clear;

```

```

        query := 'insert into cikisDurum set tanimliElemanID='+elemanID.Value+',
cikisData='+cikisData.Value+', kuralID=0,
islemZamani="'+anaForm.TarihToMySQL(datetimetostr(anaForm.simdi))+'"';
        sql.Add(query);
        ExecSQL;
    end;

    elemanID.OnChange(Sender);

    bekleForm.Hide;
    bekleForm.free;
    anaForm.ComPort.Connected := false;
    d.sonDurumlar.Refresh;
    anaForm.Timer1.Enabled := true;
    close;

end;

procedure TelleKontrol.otoKontrolClick(Sender: TObject);
var
    query,sonCikisData:string;

begin
    with d.Query2 do begin
        close;
        sql.Clear;
        sql.Add('select * from cikisDurum WHERE kuralID=0 and tanimliElemanID='+elemanID.Value+'
and id=(select max(id) from cikisDurum WHERE tanimliElemanID='+elemanID.Value+') ');
        open;

        if d.Query2.RecordCount > 0 then begin
            sonCikisData := fieldByName('cikisData').AsString;

            close;
            sql.Clear;
            query := 'insert into cikisDurum set tanimliElemanID='+elemanID.Value+',
cikisData='+sonCikisData+', kuralID=-1,
islemZamani="'+anaForm.TarihToMySQL(datetimetostr(anaForm.simdi))+'"';
            sql.Add(query);
            ExecSQL;
        end;
    end;
    otoKontrol.Enabled := false;
    close;
end;
end.

```

Grafik Kriter Seçim Ekranı

```
unit grafikSecFrm;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Mask, RzEdit, RzCmboBx, RzButton, ExtCtrls, RzPanel, dxExEdtr,
  dxEdLib, dxCntner, dxEditor, ToolEdit, ComCtrls;

type
  TgrafikSec = class(TForm)
    RzToolBar1: TRzToolBar;
    RzToolButton1: TRzToolButton;
    RzToolButton2: TRzToolButton;
    GroupBox1: TGroupBox;
    Label3: TLabel;
    elemanID: TRzComboBox;
    basZaman: TdxTimeEdit;
    bitZaman: TdxTimeEdit;
    basTarih: TDateTimePicker;
    bitTarih: TDateTimePicker;
    Label1: TLabel;
    Label2: TLabel;
    procedure RzToolButton2Click(Sender: TObject);
    procedure RzToolButton1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  grafikSec: TgrafikSec;

implementation

uses dataModul, grafikFrm;

{$R *.DFM}

procedure TgrafikSec.RzToolButton2Click(Sender: TObject);
begin
  close;
end;

procedure TgrafikSec.RzToolButton1Click(Sender: TObject);
begin
  if elemanID.ItemIndex < 0 then begin
    ShowMessage('Lütfen İncelenecek Elemanı Seçiniz');
    elemanID.SetFocus;
  end;
  exit;
end;
```

```

with d.grafikDQ do begin
    close;
    sql.clear;
    sql.add(' select olcumZamani as Zaman, ortalama as Deger from elemanortalama where
eleman='+elemanID.Value);
    open;
end;
grafikForm.grafikTitle.Caption := elemanID.text+' '+datetostr(basTarih.DateTime)+'
'+FormatDateTime('hh:mm',basZaman.time)+' - '+datetostr(bitTarih.DateTime)+'
'+FormatDateTime('hh:mm',bitZaman.time)+' Arası Verileri';
grafikForm.showmodal;
end;

end.

```

EK 3. WEB PROGRAMI KODLARI

global.php:

```
<?php

$hostname = "localhost";
$username = "sera";
$password = "sifresera";
$dbName = "seraotomasyonu";

@MYSQL_CONNECT($hostname,$username,$password) OR DIE("<br><center><strong>!!!
Database Bağlantısı Kurulamadı !!!</strong></center>");

@mysql_select_db($dbName) or DIE("<br><center><strong>!!! Database Bağlantısı Kurulamadı
!!!</strong></center>");

?>
```

index.php:

```
<?php
include ("global.php");

function authenticate() {
    header('WWW-Authenticate: Basic realm="Test Authentication System"');
    header('HTTP/1.0 401 Unauthorized');
    echo "Giriş İptal Edildi\n";
    exit;
}

if ( ($_SERVER['PHP_AUTH_USER']!= 'admin') or
(MD5($_SERVER['PHP_AUTH_PW'])!= '8f10d078b2799206cfe914b32cc6a5e9')){
    authenticate();
}

?>
<html>
<head>
<link rel="stylesheet" type="text/css" href="styles.css" />
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">

<script type="text/javascript" language="JavaScript1.2" src="js/jquery.js"></script>
<script type="text/javascript" language="JavaScript1.2" src="js/jquery.fastserialize.js"></script>

</head>
<body background="images/bg.png" >
<center>
<table width="700" border="0" background="images/tablobg.png">
<tr height="100">
<td colspan="2">&nbsp;
```

```

</td>
</tr>
<tr>
<td valign="top">

<table width="100%" id="okunanDeger" border="0" cellspacing="0" cellpadding="0">
<tbody id="okunanBaşlık">
<tr>
<td colspan="5" class="Bold cizSag cizSol cizUst cizAlt">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Son
Ölçme İşlemleri</td>
</tr>
<tr>
<td class="etiket cizSol cizSag cizAlt">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Ölçme İşlemi</td>
<td class="etiket cizSag cizAlt">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Max</td>
<td class="etiket cizSag cizAlt">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Min</td>
<td class="etiket cizSag cizAlt">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Ölçüm</td>
<td class="etiket cizSag cizAlt">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Son Ölçüm</td>
</tr>
</tbody>
<tbody id="okunanDegerler">
</tbody>
<tbody id="okunanDegerlerYukleniyor" style="display:none">
<tr bgcolor="#DDDDDD" height="100">
<th>Yükleniyor<br></th>
</tr>
</tbody>
<tbody id="okunanDegerlerTaslak" style="display:none">
<tr height="18">
<td class="baslik cizSol cizSag cizAlt">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td>
<td class="kucukYazi cizSag cizAlt">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td>
<td class="kucukYazi cizSag cizAlt">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td>
<td class="kucukYazi cizSag cizAlt Bold">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td>
<td class="kucukYazi cizSag cizAlt">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td>
</tr>
</tbody>
</table>
<br>

<table width="100%" id="sonDurum" border="0" cellspacing="0" cellpadding="0">
<tbody id="sonDurumBaslik">
<tr height="18">
<td colspan="4" class="Bold cizSag cizSol cizUst cizAlt">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Çıkış
İşlemleri Durumu</td>
</tr>
<tr height="18">
<td class="etiket cizSag cizSol cizAlt">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Çıkış İşlemi</td>
<td class="etiket cizSag cizAlt">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;İşlem Zamanı</td>
<td class="etiket cizSag cizAlt">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Durumu</td>
<td class="etiket cizSag cizAlt">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Kontrol</td>
</tr>
</tbody>
<tbody id="sonDurumlar">
</tbody>
<tbody id="sonDurumlarYukleniyor" style="display:none">

```

```

        <tr bgcolor="#DDDDDD" height="100">
            <th >Yükleniyor<br></th>
        </tr>
    </tbody>
    <tbody id="sonDurumlarTaslak" style="display:none">
        <tr height="18" >
            <td class="baslik cizSol cizSag cizAlt">&nbsp;</td>
            <td class="kucukYazi cizSag cizAlt">&nbsp;</td>
            <td class="kucukYazi cizSag cizAlt Bold">&nbsp;</td>
            <td class="kucukYazi cizSag cizAlt">&nbsp;</td>
        </tr>
    </tbody>
</table>
</td>
<td rowspan="2" valign="top">
<table>
<tr height="18">
    <td align="center" class="Bold">
        Sistem Zamanı<br>
        <span id="sistemZamani" class="kucukYazi Bold"></span><br>
        <br>
        <br>
    </td>
</tr>
</table>

<table cellpadding="0" cellspacing="0" border="0">
<tr height="161">
    <td height="161" width="161" background="images/pusula.gif" >
        <div id="tasiyici" style="display: block; width:161px; height:161px;
overflow:hidden;">
            
        </div>
    </td>
</tr>
<tr height="18">
    <td align="center">
        <span id="ruzgarHizi" class="etiket Bold"></span>
    </td>
</tr>
</table>

</td></tr></table>
</center>

<script language="javascript" type="text/javascript">
    var aci, w, h, $resim;

    $( function() {
        aci=0;
        $resim=$("#yonGoster").css("display","block");

        h=$resim.height();

```

```

        w=$resim.width();
    } );

function rotateElement(element,angle)
{
    // Rotating the element, depending on the browser:
    var rotate = 'rotate('+angle+'deg)';

    if(element.css('MozTransform')!=undefined)
        element.css('MozTransform',rotate);

    else if(element.css('WebkitTransform')!=undefined)
        element.css('WebkitTransform',rotate);

    // A version for internet explorer using filters, works but is a bit buggy (no surprise
here):
    else if(element.css("filter")!=undefined)
    {
        var cos = Math.cos(Math.PI * 2 / 360 * angle);
        var sin = Math.sin(Math.PI * 2 / 360 * angle);
        sin1=-1*sin;

        element.css("filter","progid:DXImageTransform.Microsoft.Matrix(M11="+cos+",M12="+sin
1+",M21="+sin+",M22="+cos+",SizingMethod='auto expand',FilterType='nearest neighbor')");

        element.css("marginLeft",-Math.floor((element.width()-w)/2));
        element.css("marginTop",-Math.floor((element.height()-h)/2));
    }

}

var okunanDegerleriCagir=function () {
    $("#okunanDegerlerYukleniyor").show();
    $("#okunanDegerler").hide();
    $("#okunanDegerler").html("");
    $.getJSON("okunanDegerleriDoldur.php", { istekZamani: new Date().getTime() },
function(gelenBilgi){
    if (gelenBilgi) {
        if (gelenBilgi.kayitSayisi>0){
            $(gelenBilgi.satirlar).each( function(){
                $("#okunanDegerlerTaslak * TD:nth-
child(1)").html('&nbsp;'+this.elemanAdi);
                $("#okunanDegerlerTaslak * TD:nth-
child(2)").html('&nbsp;'+this.enYuksek);
                $("#okunanDegerlerTaslak * TD:nth-
child(3)").html('&nbsp;'+this.enDusuk);
                $("#okunanDegerlerTaslak * TD:nth-
child(4)").html('&nbsp;'+this.ortalama);
                $("#okunanDegerlerTaslak * TD:nth-
child(5)").html('&nbsp;'+this.olcumZamani);
                $("#okunanDegerlerTaslak").html() );
                if (this.elemanID==1) {
                    $("#ruzgarHizi").html(this.ortalama);
                }
            });
        }
    }
});

```

```

        if (this.elemanID==2) {
            rotateElement( $resim, this.elemanOrt );
        }
    });
    $("#okunanDegerlerYukleniyor").hide();
    $("#okunanDegerler").show();
}
}
});
var sonDurumlariCagir=function () {
    $("#sonDurumlarYukleniyor").show();
    $("#sonDurumlar").hide();
    $("#sonDurumlar").html("");
    $.getJSON("sonDurumlariDoldur.php", { istekZamani: new Date().getTime() },
function(gelenBilgi){
    if (gelenBilgi) {
        if (gelenBilgi.kayitSayisi>0){
            $(gelenBilgi.satirlar).each( function(){
                $("#sonDurumlarTaslak * TD:nth-
child(1)").html('&nbsp;'+this.elemanAdi);
                $("#sonDurumlarTaslak * TD:nth-
child(2)").html('&nbsp;'+this.islemZamani);
                $("#sonDurumlarTaslak * TD:nth-
child(3)").html('&nbsp;'+this.cikisData);
                $("#sonDurumlarTaslak * TD:nth-
child(4)").html('&nbsp;'+this.islemiYapan);
                $("#sonDurumlar").append(
                $("#sonDurumlarTaslak").html() );
            });
            $("#sonDurumlarYukleniyor").hide();
            $("#sonDurumlar").show();
        }
    }
});
var timer=setInterval("sistemKontrol()",10000);
var timer2=setInterval("sistemZamaniYaz()",1000);

function sistemKontrol(){
    okunanDegerleriCagir();
    sonDurumlariCagir();
}
function sistemZamaniYaz(){
    var currentTime = new Date()
    var month = currentTime.getMonth() + 1
    var day = currentTime.getDate()
    var year = currentTime.getFullYear()
    var hour=currentTime.getHours();
    var min=currentTime.getMinutes();
    var sec=currentTime.getSeconds();
    if (day<=9) { day="0"+day; }
    if (month<=9) { month="0"+month; }

```

```

        if (hour<=9) { hour="0"+hour; }
        if (min<=9) { min="0"+min; }
        if (sec<=9) { sec="0"+sec; }
        $("#sistemZamani").html(day + "/" + month + "/" + year + "
"+hour+":"+min+":"+sec);
    }

    okunanDegerleriCagir();
    sonDurumlariCagir();
    sistemZamaniYaz();

</script>

</body>
</html>

```

okunanDegerleriDoldur.php:

```

<?php
header("Content-Type: text/javascript; charset=iso-8859-9");

function authenticate() {
    header('WWW-Authenticate: Basic realm="Test Authentication System"');
    header('HTTP/1.0 401 Unauthorized');
    echo "Giriş İptal Edildi\n";
    exit;
}

if ( ($_SERVER['PHP_AUTH_USER']!= 'admin') or
(MD5($_SERVER['PHP_AUTH_PW'])!= '8f10d078b2799206cfe914b32cc6a5e9')){
    authenticate();
}

include ("global.php");
$sql="select e.id as elemanID,e.adi,CONCAT(eo.enYuksek,' ',e.birimi) as
enYuksek,CONCAT(eo.enDusuk,' ',e.birimi) as enDusuk,CONCAT(eo.ortalama,' ',e.birimi) as
ortalama, eo.olcumZamani as olcumZamani,eo.ortalama as ort from eleman e
left join elemanortalama eo ON eo.eleman=e.id and eo.id=(select max(id) from elemanortalama where
eleman=e.id)
where e.aktif='1' and inOu='0'
group by e.id
";

$olcumGelen=mysql_query($sql);
@$olcumSay=mysql_num_rows($olcumGelen);

echo "{";
echo "\"kayitSayisi\":'$olcumSay'";
    $ilkSatirGecildi=false;
    echo "\"satirlar\":[";
    while ($olcum=@mysql_fetch_array($olcumGelen)) {
        if ($ilkSatirGecildi) echo ","; else $ilkSatirGecildi =true;
        echo "{";
        echo "\"elemanAdi\":'\".($olcum[\"adi\"]).\"\",";
        echo "\"elemanID\":'\".($olcum[\"elemanID\"]).\"\",";
        echo "\"elemanOrt\":'\".($olcum[\"ort\"]).\"\",";
    }
}

```

```

        echo
        "\enYukse\":"\".(($Solcum["enYukse"])?$Solcum["enYukse"]:"<center>-</center>")."\",";
        echo "\enDusuk\":"\".(
        ($Solcum["enDusuk"])?$Solcum["enDusuk"]:"<center>-</center>")."\",";
        echo "\ortalama\":"\".(
        ($Solcum["ortalama"])?$Solcum["ortalama"]:"<center>-</center>")."\",";
        echo "\olcumZamani\":"\".(($Solcum["olcumZamani"]) ?
        (substr($Solcum["olcumZamani"],8,2)."/".substr($Solcum["olcumZamani"],5,2)."/".substr($Solcum["olcu
        mZamani"],0,4)." ".substr($Solcum["olcumZamani"],11)):"<center>-</center>")."\",";
        echo "}";
    }
    echo "]";
echo "}";
?>

```

sonDurumlariDoldur.php:

```

<?php
    header("Content-Type: text/javascript; charset=iso-8859-9");

function authenticate() {
    header("WWW-Authenticate: Basic realm="Test Authentication System");
    header("HTTP/1.0 401 Unauthorized");
    echo "Giriş İptal Edildi\n";
    exit;
}

    if ( ($_SERVER['PHP_AUTH_USER']!= 'admin') or
(MD5($_SERVER['PHP_AUTH_PW'])!= '8f10d078b2799206cfe914b32cc6a5e9')){
        authenticate();
    }

    include ("global.php");
    $sql="select e.adi, cd.islemZamani as 'islemZamani',cikisData,IF(kuralID is null,"
    IF (kuralID>0, 'Otomatik' , 'Elle') ) as 'islemiYapan'
    from eleman e
    left join elemantanim et ON et.elemanID=e.id
    left join cikisdurum cd ON cd.tanimliElemanID=et.id and cd.id=(select max(id) from cikisdurum
    where tanimliElemanID=et.id)
    WHERE e.inOu='1' and e.aktif='1'
";

    $sonDurumGelen=mysql_query($sql);
    @$sonDurumSay=mysql_num_rows($sonDurumGelen);

    echo "{";
    echo "\kayitSayisi\":"$sonDurumSay",";
    $ilkSatirGecildi=false;
    echo "\satirlar\":"[";
    while ($sonDurum=@mysql_fetch_array($sonDurumGelen)) {
        if ($ilkSatirGecildi) echo ","; else $ilkSatirGecildi =true;
        echo "{";
        echo "\elemanAdi\":"\".($sonDurum["adi"])."\",";
        echo "\islemiYapan\":"\".($sonDurum["islemiYapan"])."\",";
    }
}

```

```

        echo "\"islemZamani\":\":".((\$sonDurum["islemZamani"]) ?
(substr(\$sonDurum["islemZamani"],8,2)."/".substr(\$sonDurum["islemZamani"],5,2)."/".substr(\$sonD
urum["islemZamani"],0,4)." ".substr(\$sonDurum["islemZamani"],11)):"<center>-</center>")."\",";
        echo "\"cikisData\":\":".( durumGetir(\$sonDurum["cikisData"])
).\\"";
        echo "}";
    }
    echo "]";
echo "}";

function durumGetir(\$gelenCikisData){
    \$donusBilgi="";
//    (is_numeric(\$gelenCikisData))? \$gelenCikisData=\$gelenCikisData:\$gelenCikisData="";
    switch (\$gelenCikisData) {
        case "0":
            \$donusBilgi = "Giriş Elemanı";break;
        case "1":
            \$donusBilgi = "Kapalı";break;
        case "2":
            \$donusBilgi = "Açık";break;
        case "3":
            \$donusBilgi = "Tam Kapalı";break;
        case "4":
            \$donusBilgi = "%10 Açık";break;
        case "5":
            \$donusBilgi = "%20 Açık";break;
        case "6":
            \$donusBilgi = "%30 Açık";break;
        case "7":
            \$donusBilgi = "%40 Açık";break;
        case "8":
            \$donusBilgi = "%50 Açık";break;
        case "9":
            \$donusBilgi = "%60 Açık";break;
        case "10":
            \$donusBilgi = "%70 Açık";break;
        case "11":
            \$donusBilgi = "%80 Açık";break;
        case "12":
            \$donusBilgi = "%90 Açık";break;
        case "13":
            \$donusBilgi = "Tam Açık";break;
        default:
            \$donusBilgi = "Tanımsız";
    }
    return \$donusBilgi;
}

?>

```