



T.C.

**KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**

**PELET MAKİNASI PARAMETRELERİNİ SCADADAN
PHP ARACILIĞI İLE ANDROİDE AKTARILMASI**

Mahmut KERTMEN

YÜKSEK LİSANS TEZİ

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Temmuz-2019

KONYA

Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Mahmut Kertmen tarafından hazırlanan “Pelet Makinası Parametreleri Scadadan Php Aracılığı ile Androide Aktarılması” adlı tez çalışması 18/07/2019 tarihinde aşağıdaki jüri tarafından oy birliği ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Elektrik ve Elektronik Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS olarak kabul edilmiştir.

Jüri Üyeleri

Başkan

Prof. Dr. Cemil SUNGUR

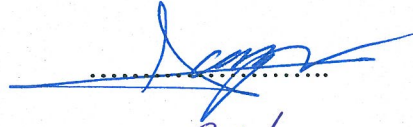
Danışman

Doç. Dr. Nurettin ÇETİNKAYA

Üye

Dr. Öğr. Üyesi Mümtaz MUTLUER

İmza







Yukarıdaki sonucu onaylarım.

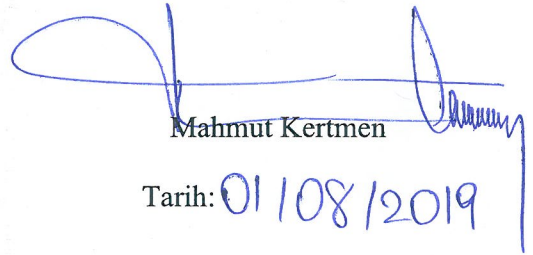
LEE Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.


Mahmut Kertmen
Tarih: 01/08/2019

ÖZET

YÜKSEK LİSANS TEZİ

PELET MAKİNASI PARAMETRELERİNİ SCADADAN PHP ARACILIĞI İLE ANDROİDE AKTARILMASI

Mahmut KERTMEN

**Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalı**

Danışman: Doç. Dr. Nurettin ÇETİNKAYA

2019, 63 Sayfa

Jüri

**Prof. Dr. Cemil SUNGUR
Doç. Dr. Nurettin ÇETİNKAYA
Dr. Öğr. Üyesi Mümtaz MUTLUER**

Dünyadaki canlı popülasyonu günden güne artmaktadır. Bu yüzden dünyadaki üretimlerin seri ve hızlı yapılması için sanayi devrimi ile beraber gerekli adımlar atılmıştır. Ağır makine tesisleri, insanoğlu tarafından gerçekleştirilen işleri ele almış ve bu sayede üretimin artmasına neden olmuştur. Teknolojik gelişmeler ağır makinaların çalışma prensibini günden güne değiştirmiştir. Buhar ile çalışan makineler elektrik enerjisi ile çalışmaya başlamış olup kontroller kumanda sistemine göre gerçekleştirilmiştir. Bilgisayar teknolojisinin gelişmesi bu kontrol sistemini başka bir boyuta taşımış olup PLC ve SCADA sistemini endüstri tesislerin karşısına getirmiştir. Günümüzde bilgi paylaşımı bu gelişmeler neticesinde çok önemlidir. Bu talep doğrultusunda SCADA üzerine düşen bilgilerin hosting aracılığı ile android bir programa ve bir html sayfasına aktarılması düşünülmüştür. Üzerinde düşünülen sistem ile beraber SCADA ile kontrol edilen bir tesisin internet aracılığı ile daha esnek çalışması hedeflenmiştir. Yapılan çalışmada yem fabrikalarında sıklıkla kullanılan pelet makinası referans olarak alınmıştır.

Anahtar Kelimeler: SCADA, PLC, Siemens, Php, Android, Hosting

ABSTRACT

MS THESIS

**TRANSFERİNG PARAMETERS OF PELLET MİLL FROM SCADA TO
ANDROİD APPLİCATION WITH PHP**

Mahmut KERTMEN

**Konya Technical University
Institute of Graduate Studies
Department of Electrical and Electronic Engineering**

Advisor: Assoc. Prof. Dr. Nurettin ÇETİNKAYA

2019, 63 Pages

Jury

Prof. Dr. Cemil SUNGUR

Assoc. Prof. Dr. Nurettin ÇETİNKAYA

Assist. Prof. Dr. Mümtaz MUTLUER

On Earth live population increase day by day. Fort this reason, the necessary steps have been taken with the industrial revolution in order to maket he productions in the world fast and fast. Big machinery facilities have dealt with the works carried out by human beings and this has led to an increase in production. Technological developments have changed the working principle of big machines day by day. Steam powered machines started to work with electrical energy and controls were carried out according to control system. The development of computer technology has taken this control system to another dimension and brought the PLC and SCADA system to the industrial facilities. Today, information sharing is very important as a result of these developments. In accordance with this demand, information on SCADA is planned to be transferred to an android program and an html page via hosting. The system, which is controlled with SCADA, is aimed to be more flexible through internet. In the study, the pellet machine, which is frequently used in feed factories, is taken as a reference.

Keywords: SCADA, PLC, Siemens, Php, Android, Hosting

ÖNSÖZ

Sanayi devrimi ile beraber başlayan süreçte gelişen endüstriyel makineler üretim için daima önemli bir rol üstlenmiştir. Günden güne üretimdeki talebin artması ilerleyen süreçlerde endüstriyel makinaları geliştirmiş ve gelişen bu makinelerin kontrolü için dünyada çeşitli fikirlerin oluşmasına sebep olmuştur.

Elektrik ve elektronik alanında teknolojinin ilerlemelerin ortaya çıkması endüstri sektörü de olumlu noktada etkilemiştir. Endüstri’de kullanılan makinelerin buhar enerjisinden elektrik enerjine dönüştürülmesi zamanla sağlanmış ve proseslerin kontrol sistemi yine elektriksel kumanda sistemleri ile yapılması söz konusu olmuştur. Elektronik kart tasarımının gelişimi ve kart programlama yazılım tekniklerinin artması zamanla PLC modüllerin çıkmasına neden olmuş kontrolün daha kararlı olması sağlanmıştır. Bilgisayar ve bilgisayar programlarının gelişimi insan ara yüzü olan HMI sistemlerini endüstri sektöründe ön plana çıkarmış ve günümüzde kontrol sistemleri bilgisayar ve Linux tabanlı insan ara yüzü cihazlar ile kullanıcıyı karşı karşıya getirmiştir.

Endüstriyel gelişmelerin ışığı ile beraber insanlığı son on yıl içerisinde hayatının önemli bir bölümünü kapsayan android cihazlar ve internet dünyası bu tez çalışmasında endüstri proseslerin daha kararlı çalışabilmesi için SCADA sistemlerinin yanına yardımcı olarak düşünülmüştür. Bununla alakalı olarak Siemens’in Simatic Manager programı ve Wincc Explorer programı tez çalışmasında kullanılmıştır. Siemens’in SCADA programı olan Wincc Explorer üzerinde yazılan bir Script vasıtası ile veri alışverişi için hosting üzerinde bir aktarma sağlanmıştır. Kullanılan modemlerin IP adresi üzerinden internet sayfasına ve android programına veri akışının sağlandığı sistemde oluşturulan android programından ve internet sayfasından sistemin kontrol edilmesi sağlanmıştır. Tez çalışmasında referans olarak yem fabrikalarında sıklıkla kullanılan pelet makinası tercih edilmiştir. Düşünülen tez ile beraber makine sistemlerinin daha kararlı ve kontrollü çalışması üzerinde bir öneride bulunulmak istenmiştir.

Mahmut KERTMEN

KONYA-2019

İÇİNDEKİLER

ÖZET	iv
ABSTRACT	v
ÖNSÖZ	vi
İÇİNDEKİLER	vii
SİMGELER VE KISALTMALAR	x
1. GİRİŞ	1
2. KAYNAK ARAŞTIRMASI	4
3. PELET MAKİNASI	8
4. PLC	10
4.1. Genel Tanım	10
4.2. PLC Cihazlarının Temel Yapısı.....	10
5. SCADA	13
5.1. Genel Tanım	13
5.2. SCADA Sisteminin Yapısı	14
5.2.1. MTU.....	14
5.2.2. RTU	16
5.2.2.1. İletişim ünitesi.....	16
5.2.2.2. Merkezi işlem ünitesi	17
5.2.2.3. Giriş çıkış izolasyon ünitesi.....	17
5.2.2.4. Kullanıcı arabirimi.....	17
5.2.2.5. Test birimi	17
5.2.2.6. Güç kaynağı.....	18
5.2.3. İletişim sistemleri.....	18
6. WEB PROGRAMLAMA DİLLERİ	20
6.1. HTML	20
6.1.1. HTML ana tagları	21
6.1.1.1. <!DOCTYPE>	21
6.1.1.2. <html >	21
6.1.1.3. <head>	21
6.1.1.4. <title>	21
6.1.1.5. <body >.....	22

6.1.2. HTML meta tagları	22
6.1.2.1. Content-type	23
6.1.2.2. Content-language.....	23
6.1.2.3. Description	23
6.1.2.4. Keywords.....	24
6.1.2.5. Robots.....	24
6.1.3.Önemli bazı HTML tagları	25
6.1.3.1. <!-- -->	25
6.1.3.2. <a>.....	25
6.1.3.3. <button>	25
6.1.3.4. <canvas >.....	25
6.1.3.5. <div >	25
6.1.3.6. <figure >	26
6.1.3.7. <footer >	26
6.1.3.8. <form >	26
6.1.3.9. <h1>,<h2>,<h3>,<h4>,<h5>,<h6>	26
6.1.3.10. <hr>	26
6.1.3.11. 	26
6.1.3.12. <label >.....	27
6.1.3.13. <map >.....	27
6.1.3.14. <menu>.....	27
6.1.3.15. <menuitem>.....	27
6.1.3.16. <nav>.....	27
6.1.3.17. <p >.....	28
6.2. Php	28
6.2.1. Php tarihsel gelişim süreci	28
6.2.2. Php çalışma prensibi	29
6.3. MySQL	30
6.3.1. XAMPP.....	31
6.4.Android İşletim Sistemi	32
7.MATERYAL VE YÖNTEM.....	34
7.1. Simatic Manager	34
7.1.1. Fonksiyon bloğu	34
7.1.2. Fonksiyon.....	35
7.1.3. Database.....	35
7.2. Simatic Manager Programını Simülasyona Alma	36
7.3. PLC'den SCADA'ya Veri Aktarımı	36
7.4. Wincc Explorer	37
7.4.1. Wincc Explorer'in ana unsurları.....	38
7.4.1.1.Computer	38
7.4.1.2.Tag management	38
7.4.1.2.1.Structure	39
7.4.1.3.Graphics desinger	39
7.4.1.4.Menus and toolbars.....	39
7.4.1.5.Alarm loging.....	39
7.4.1.6.Report desinger.....	40
7.4.1.7. Text library	40
7.4.1.8. Text distributor	40
7.4.1.9. User administrator	40
7.4.1.10. Picture tree manager	41
7.4.1.10. Global script	41

7.4.1.11. Life beat monitöring	42
7.4.1.12. Cross referance	42
7.5. SCADA'yı Simülasyona Alma	42
7.6. PLC'den SCADA'ya Veri Aktarımı	43
7.7. SCADA'dan Hosting'e Veri Aktarımı.....	43
7.8.Hosting'den SCADA'ya Veri Aktarımı.....	48
7.9.Hosting ve Domain	49
7.10.Codeigniter Klasörünü Hosting Serverına Aktarılması	49
7.11.Veritabanının Web Sayfasını Beslemesi.....	50
7.12.Web Sayfalarından Android'e Veri Gönderme	51
7.13.Json	52
7.14.Web Sayfasından SCADA'ya Veri Gönderme.....	52
8.ARAŞTIRMA SONUÇLARI VE TARTIŞMA.....	54
8.1. Araştırma Sonucunda Çalışmanın Avantajı ve Dezavantajı.....	54
8.1.1. Araştırma sonucunda çalışmanın avantajı	54
8.1.2. Araştırma sonucunda çalışmanın dezavantajı.....	54
9. SONUÇLAR VE ÖNERİLER.....	55
KAYNAKLAR	56
EKLER	58
ÖZGEÇMİŞ	63

SİMGELER VE KISALTMALAR

Kısaltmalar

SCADA:	Supervisory Control and Data Acquisition
PLC:	Programmable Logic Controllers
HMI:	Human Machine Interface
Vbs:	Visual Basic Script
PID:	Proportional Integral Derivative
PHP:	Hypertext Preprocessor
MYSQL:	My Structured Query Language
SILAR:	Sequential İyonik Layer Adsorpsiyon ve Reaksiyonu
RPG:	Role Playing Game
RTS:	Real Time Strategy
Ms SQL:	Microsoft Structured Query Language
AJAX:	Asynchronous Javascript And Hmi
CPU:	Central Processing Unit
ms:	Milisaniye
24V:	24 Volt
RAM:	Random Accesible Memory
EEPROM:	Electrically Erasable Programmable Read-Only Memory
DCS:	Distributed Control System
AER:	Australian Energy Regulator
MTU:	Master Transfer Unit
RTU:	Remote Transfer Unit
HTML:	HyperText Markup Language
DB:	Database
PC:	Personal Computer
FB:	Fonksiyon Blođu
FC:	Fonksiyon
OB:	Organizasyon Blođu

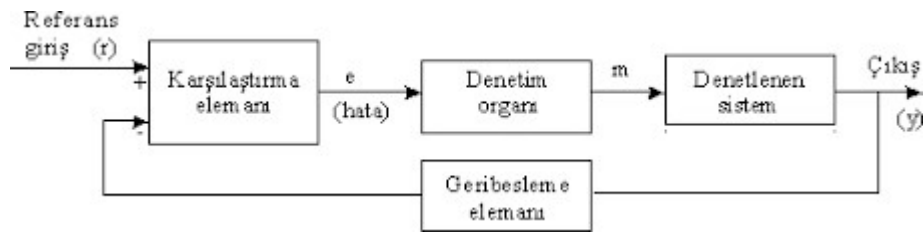
1. GİRİŞ

Kontrol sistemleri insanlığın var olduğu günden bugüne kadar ki süreçte üzerinde durulduğu fizik tabiridir. Başta kainatta olmak üzere dünya üzerindeki fiziksel olayların hepsi bir kontrol sistemine bağlı olarak gerçekleşmektedir. Güneş'in geliş açısından dolayı baharın gelmesi ve çiçeklerin açması, manyetik alanların farklılığından dolayı belirli zamanlarda ayın güneşin önüne geçmesi gibi örnekler evrendeki kontrol sistemlerine örnek verilebilecek olaylardan başlıcalarıdır.

İnsanoğlunun teknolojik hamleler neticesinde kontrol sistemleri dünyada farklı konular üzerine ele alınmaya başlanmıştır. Bu konulardan elektriksel kontrol elektrik enerjisinin binalara, fabrikalara ve taşınabilir cihazların üzerine girmesi nedeniyle daha bir önem kazanmıştır. En basit örnek ile vermek gerekirse bir evdeki kısa devre sonucu da sigortaların atması, fabrikalardaki aşırı yük çeken bir motorun yanmaması için atan motor koruma aslında elektriksel kontrolün hayatımızdaki ne denli olduğunu göstermektedir.

Günümüzde kontrol sistemleri üzerindeki çalışmalar neticesinde kontrol sistemleri kararlı ve kararsız olmak üzere ikiye ayrılmaktadır. Kontrol sistemlerindeki kararlı yapı sistemin çıkışından aldığı bilgileri başlangıç noktasına götürerek sistemin kontrollü bir yapıya bürünmesini sağlar. Kararsız yapılarda ise bu söz konusu değildir. Zira kararsız yapılarda sistemin çıkışından bilgiler alınmaz ve bunun neticesinde oluşabilecek olaylar kontrolsüz bir şekilde gerçekleşir.

Elektriksel kontrol sistemlerinde kararlı yapılar genellikle sahadan dönen bilgiler neticesinde meydana gelmektedir. Şekil 1.1. geri besleme diyagramında da gösterildiği gibi endüstriyel sektörde motorun çalıştığını belirten kontaktörün yardımcı kontağından gelen bilgi, sahadaki valfin çalıştığını belirten sensör bu kararlı yapıya örnek teşkil eder.



Şekil 1.1. Kararlı sistem kontrol şeması örneği (Anonim)

Endüstri sektöründe makinaların çalışmasını sağlayan enerji elektrik enerjisidir. Özellikle elektrik enerjisinin harcama konusunda konutlara göre endüstri sektörü daha ciddi bir noktadır. Enerjinin kontrollü olarak kullanılması ortaya çıkabilecek arızalara ve ciddi problemlere karşı önem teşkil etmektedir. Zira kullanılan makinaların ücretleri yüksektir. Bu sebeple oluşabilecek elektriksel arızaları en aza indirebilmek için güvenlik önlemleri sıklaştırılmaktadır ve sahadan daha fazla bilgi alınması istenmektedir. Örneğin Şekil 1.2’ de gösterildiği gibi bir motorun aşırı akım çekmesi motorun yanmasına sebep olur ki bu nedenle güvenlik önlemi için motor koruma konulmuştur. Yine sahadaki arızanın aşırı akım çekilmesinden kaynaklı olduğunu anlayabilmek için de motor korumanın üzerindeki yardımcı kontaklardan gelen bilgilere bakılmaktadır. Endüstri sektöründeki kontroller geçmişten bugüne sahadan gelen bilgiler neticesinde yapılmıştır. Bir motorun çalışması ya da bir valfin çekmesi bir komutla sağlanmış olup yine bu komutlar sahadan alınan feed back bilgileri ile kararlı bir yapı haline getirilmesi öngörülmüştür.



Şekil 1.2. Kararsız kontrol sistemi sonucu oluşan motor yanması (Anonim)

Yazılım ve elektronik sektörünün gelişmesi aynı zamanda endüstri sektöründeki kontrol mekanizmalarının gelişmesine ve değişmesine sebep olmuştur. Özellikle ağır saha şartlarına uyum sağlayabilen elektronik kartlar yazılım ile birleşmesi sonucunda daha kararlı sisteme girilmesine olanak sağlamıştır. Kontrolün insandan yazılıma geçişi

ile beraber sahada meydana gelebilecek operatör kaynaklı arızalar daha aza indirilmiş olup daha etkili bir kontrol gerçekleştirilmiştir.

Endüstri sektöründeki teknolojik gelişmeler neticesinde günden güne kontrol daha kararlı bir hale getirilmektedir. Yapılan çalışmalar kontrol sisteminin saha olaylarından etkilenmeyen elektronik kartlara ve bilgisayarlara bırakılması yönündedir. Bu çalışma ile beraber kontrol sistemi daha geniş yelpaze üzerinde durulması hedeflenmiştir. Bu sebeple SCADA üzerinde olan bir sistemin internet sayfasına ve aynı zamanda bir android uygulamasına aktarılması hedeflenip bu hedefle kontrolü sağlanması öngörülmüştür.



2. KAYNAK ARAŞTIRMASI

Bu tez çalışmasında otomasyon sektöründe öncü firmalardan biri olan Siemens'in Simatic Manager programı kullanılmıştır. Bu programı kullanmak için Siemens tarafından hazırlanmış olan "Working with STEP 7" kitapçığından yararlanılmış olup yazılım yine bu kitapçık üzerindeki bilgiler neticesinde tasarlanmıştır. Kitapçık haricinde ayrıca Siemens PLC programlama ekosistemine uygun kaynaklardan da yararlanılmıştır. Siemens' in hazırlamış olduğu kitapçık incelenecek olursa, yazılım fonksiyon blokları, fonksiyon, data blokları ve organizasyon bloklarından oluşmaktadır. Bu nesnelere birbirleri arasında içten içe geçerek ve birbirlerini çağırarak yazılım gerçekleştirilmektedir.

Bu tez çalışmasında kullanılan başka kaynaklardan bir diğeri "Wincc V7.4 Simatic Hmi Wincc V7.4 Getting Started" kaynağıdır. SCADA ara yüzü hazırlamada ve nesnelere gelen bilgileri neticesinde flaşlama ya da renklendirme işlemi yapmasını sağlamaktadır. Ayrıca SCADA ile PLC arasındaki haberleşmeyi sağlamayı yine bu kitapçık üzerindeki bilgiler neticesinde projede yarar sağlanmıştır.

SCADA üzerindeki bilgilerin sanal ortama aktarılması için kullanılan bir başka kaynak ise yine Siemens firması tarafından hazırlanan "Wincc V7.4 Wincc: Scripting" kitapçığıdır. Veri tabanı oluşturmada ve verilerin sanal bilgisayar hosting üzerine aktarılmasında Vbs dili kullanılması gerek duyulmuştur. Bununla beraber Verinin tekrardan sanal ortamdan SCADA'ya aktarılması için Vbs dili kullanılması öngörülmüştür.

Karayel, 2013 yılında yaptığı yüksek lisans tez çalışmada prototip olarak şebekeye bağlı olmayan küçük ölçekte bir hidroelektrik santralinin otomasyonunu gerçekleştirmiştir. PLC kullanarak gerçekleştirmiş olduğu proseste PID değerlerine bağlı olarak bilgisayar üzerinden sistem kontrolünü sağlamasını hedeflemiştir (Karayel, 2013).

Şumnulu, 2005 yılında yaptığı yüksek lisans tez çalışmasında Php, MySQL kodları yazılımında ister Windows tabanlı formatlarda isterse de server tabanlı uygulamalarda kodların çalışıp çalışmadığı test etmek amaçlı bir editör gerçekleştirmiştir. Özellikle günümüzde Php'nin uygulama sıklığından dolayı bu editörü gerçekleştirmesinde etken olmuştur (Şumnulu, 2005).

Kayabaşı, 2016 yılında yaptığı yüksek lisans tez çalışmasında android tabanlı uygulamaların içerisindeki kötücül yazılımlara karşı uygulamaların tespit edilmesi ve api uzantılı dosyaların denetlenmesini hedeflemiştir. Drebin veri tabanı ile gerçekleştirdiği sistemde statik analiz yöntemi kullanmış ve sınıflandırma ile gerçekleştirdiği sistemin en yüksek performansı göstermesini sağlamıştır (Kayabaşı, 2016).

Kırmızıgül, 2014 yılında yaptığı yüksek lisans tez çalışmasında metro hatları için SCADA sistemleri için ayrıca yeni bir model önerisi gerçekleştirmiştir. Çalışmasında İstanbul Anadolu yakasında inşaatı devam eden SCADA metotları incelenmiş ve bunun sonucunda yeni metotları uygun bir birleşim sağlanarak önerilmiştir (Kırmızıgül, 2014).

Salğar, 2010 yılında gerçekleştirmiş olduğu çalışmada doğalgaz otomasyonu hakkında bir çalışma planlamıştır. Gerçekleştirmiş olduğu çalışma ile beraber çelik hat üzerinde bulunan doğalgaz istasyonlarının üzerindeki bilgileri tek bir ana kontrol merkezinde toplanılması sağlanmış ve bunun üzerinde yapılan çalışma ile daha etkin, daha verimli ve daha esnek bir kontrol hedeflemiştir. Gerçeğe uygun problemler üzerinde çalışmalar gerçekleştirilmiştir (Salğar, 2010).

Yaşar, 2018 yılında gerçekleştirmiş olduğu yüksek lisans tez çalışmada ardışık iyonik tabaka adsorpsiyon ve Reaksiyonu (SILAR) methodunu kolaylaştıracak bir PLC kontrol cihazı gerçekleştirmiştir. İnce filmlerin elde edilmesi amacıyla kullanılmakta olan SILAR methoduna etki eden parametreler ayrı ayrı olarak yüksek lisans tez çalışmasında incelenmiş ve insan hatası faktörleri kaldırmak hedeflenmiştir (Yaşar, 2018).

Uslu, 2015 yılında yaptığı çalışma ile HTML5 tabanlı oyunlar için yapay zeka kütüphanesi gerçekleştirmiştir. Bu incelemede aksiyon, RPG, RTS, macera, spor gibi oyunlar üzerine yönelmiş sanal oyun tarihi hakkında bilgi verilmiştir. Çalışma sonucunda yapay oyun zeka kütüphanesi için DignityAI geliştirilerek üzerinde bir oyun tasarlanmıştır (Uslu, 2015).

Sönmez, 2017 yılında yapmış olduğu çalışmada Açık kaynak kodlar ile öğrenci takip sistemi tasarımı gerçekleştirmiştir. Php, ASP.Net, Ms SQL ve My SQL gibi açık kod kaynaklı programlar ile iki sistem oluşturulmuştur. İki sistem arasındaki performanslar incelenmiş olup meydana getirilecek olan sistemlerin küçük ve orta ölçekli olarak hangisinin tercih edileceğinden bahsedilmiştir (Sönmez, 2017).

Bıçek, 2016 yılında yapış olduđu çalıřma ile beraber Android web tabanlı program vasıtasıyla işitme engellileri ilgilendirecek bir çalıřma üzerinde durmuřtur. Sesli ders takip sistemi ile beraber işitme engellilerin hayatını kolaylařtırmaya yönelik olan çalıřma AJAX yazılımı ile web sayfalarına aktarılması sađlanmıřtır. Tez kapsamındaki çalıřma dersi takip eden ve dersi tekrar etmesine olanak sunacak řekilde planlanmıř ve tasarlanmıřtır (Bıçek, 2016).

Kulga, 2017 yılında gerçekteřirmiş olduđu tez çalıřması ile beraber Tıbbi ultrason cihazları hakkında bir çalıřma üzerine gitmiřtir. Yapmış olduđu bu çalıřma ile beraber ultrason cihazlarından meydana gelen ultrasonik dalgaların eğitim düzeyinde anlatılmasına olanak sađlamayı hedeflemiřtir (Kulga, 2017).

řen, 2018 yılında gerçekteřirmiş olduđu tez çalıřması ile beraber içme suyu tesislerinde SCADA sistemlerine ait sezgisel yaklařımları üzerine tespit arayıřlarına giriřmiřtir. Özellikle SCADA sistemlerinde yeni yeni ortaya çıkan sezgisel yöntemle insan paydasını en aza indirebilmek için yaptıđı arařtırmada su ve enerji sistemlerinde sezgisel yaklařımlar ile ne kadar yakın bir oranda sonuç varılabileceđi hesaplamasına gidilmiřtir (řen, 2018).

Üstünsoy, 2018 yılında gerçekteřirmiş olduđu tez çalıřması ile günümüz teknolojisindeki hızlı enerji harcamalarına dikkat çekerek SCADA üzerinde enerji takibi ve tasarrufu hakkında bir çalıřma planlamıřtır. Özellikle alışveriř merkezlerinde ve řehir hayatındaki enerji harcamaları hakkında tek bir merkezden toplama yaparak raporlama ile enerji tasarrufu hakkında bir yöntem ortaya atmıřtır (Üstünsoy, 2018).

Asubay, 2018 yılında gerçekteřirmiş olduđu tez çalıřması ile rölelerin arıza bilgileri, seri portlar üzerinden haberleřtirilmesi sađlanmıř ve kesicilerin konum bilgileri tařınmıřtır. Tez çalıřmasında özellikle oluřacak bir elektriksel arızada yardımcı rölelerin devre dıřı bırakılması sađlanmıřtır. Amacı itibariyle SCADA sistemleri üzerinden elektriksel arızaları en aza indirmek istemiřtir (Asubay, 2018).

Sarıkahya, 2013 yılında gerçekteřirmiş olduđu tez çalıřmasında oksijen sıvılařtırma plantine uygulanarak yük atma sistemi gerçekteřirmiřtir. Tez çalıřmasında özellikle yeni nesil yük atma yöntemi Intelligent load shedding yöntemini SCADA ve PLC sistemine eklemek istemiřtir. Amacı itibariyle SCADA sistemleri üzerinden Intelligent load shedding yöntemi sayesinde jeneratör üzerindeki yük miktarının azaldıđı

ve toparlanma süresinin daha kısa sürede gerçekleştiğinden bahsetmiştir (Sarıkahya, 2013).

Başar, 2014 yılında gerçekleştirmiş olduğu tez çalışmasında Afrika ile Arabistan levhalarının kalınlaşmasını takip etmek üzerine gerçekleştirmiştir. Hali hazırda ülkemizin doğu bölümündeki levha tabakasının kalınlaşmasını takip etmek için ALAGÖZ projesi ile GPS ve GPSS sayesinde web yazılımı projenin dökülmesi sağlanmıştır (Başar, 2013).

Boşnak, 2016 yılında gerçekleştirmiş olduğu tez çalışmasında döküm fabrikalarında kullanılan indüksiyon fırını ile ilgili olarak PLC sistemi üzerine bir çalışma gerçekleştirmiştir. Döküm fabrikalarında gerçekleşen çelik ürünlerinin kalitesini artırmak için gerçekleştirdiği sistemde PLL algoritma üzerinde daha çok durmuştur (Boşnak,2016).

Özer, 2016 yılında gerçekleştirmiş olduğu tez çalışmasında bir endüstriyel tesis üzerinde SCADA izlenmesi yapmıştır. Gerekli olan bilgiler ışında prosesi çalıştıran çalışmada debi, sıcaklık, basınç, seviye, ağırlık gibi proses elemanlarını SCADA üzerinde takip etmiş ve olumsuzluklara karşı önlem almasını sağlamıştır (Özer, 2016).

Karan, 2017 yılında gerçekleştirmiş olduğu tez çalışmasında hemzemin geçitlerde yaşanan kazalara yön çevirerek kazaların en aza indirmeye ile ilgili bir çalışma meydana getirmiştir. Ülkemizdeki tren kazalarının yaklaşık %28 hemzemin geçiti ile ilgili olduğunu belirttiği çalışmada hemzemin geçitlerdeki kazayı minimuma indirmek için Emiralem İstasyonunda bir PLC ile uygulama gerçekleştirmiştir (Karan, 2017).

3. PELET MAKİNASI

İnsan popülasyonu günden güne arttığı bir ortamda her besin kaynaklarında olduğu gibi hayvancılık sektöründe bu bağlamda gelişmekte ve kendini yeni yerlerde görmek zorunda kalmıştır. Zira modern tarım ve hayvancılığa geçiş bunu bir sonucu olarak gösterilebilmektedir. Hayvancılıkta yem otlaklık alandan endüstri sektöründe laboratuvar ortamında belirtilen rasyonla hazırlanmış yemlere doğru bir yönelim söz konusudur. Özellikle günümüz hayvan yemlerinde dünya düzeyinde sıklıkla tüketilen yem endüstri sektöründe hazırlanan pelet yemi olarak görülüyor. Birden fazla hammaddenin belirli rasyonlar neticesinde karıştırılıp tozdan katı sert halde dönüştürülmüş yeme pelet yemi bu yemi ortaya çıkararak makinaya da pelet ya da pres makinası denir.

Pelet Makinesi proses olarak üç ana bölümden meydana gelmektedir. Bunlar besleyici, kondisyoner, pelet presi kısmıdır. Makinede besleyici bölümü malın kondisyonerlere veren bölümdür. Besleme bölümünde kullanılacak olan motora takılacak invertör ile pelet presini döndüren ana motor akımlarına göre malın gelme hızını ayarlanmaktadır. Kondisyoner besleyiciden gelen kuru malın belirli bir oranda buhar verilerek nemlenmesi için kullanılmaktadır. Verilecek olan buhar mal üzerindeki sıcaklığa göre ayarlanmaktadır. Son olarak Pelet pres kısmı nemli toz halinde gelen malın pelet haline getirildiği bölümdür. Burada iki adet pelet diski bulunmaktadır. Pelet diskleri tek kayışla iki adet ana motor yardımı ile hareket edilmesi sağlanmaktadır.

Pelet makinesinden çıkan pelet yemi üzerinde bulunan sıcaklığı atması için daha sonra proses içinde soğutma makinesine aktarılmaktadır. Bu aktarma işlemi ile beraber soğutucudaki peletin sıcaklığı 70 C' den 20 C' ye doğru düşmesi meydana gelmektedir. Soğutucunun pelet soğutma işlemi için fan olarak kabul edilen bir makinadan yararlanılmaktadır. Fan üzerinde gerçekleşen soğutma işlemi sırasında peletin yanı sıra toz olarak soğutucuya düşen malın fanın dışına çıkmaması gerekmektedir. Bu yüzden fanın altında siklon olarak adlandırılan bir makine kullanılmaktadır. Siklon hava kilidi olan bir motor vasıtası ile sıcak olan havayı fabrikadan atmayı sağlarken ve toz olarak kalan yemi ise tekrardan fabrikanın içine kazandırmaktadır.



Şekil 3.1. Pelet Makinesi (Anonim)

Pelet makinası aslında bir insanın boğaz yapısına benzemektedir. Zira pelet makinası nemsiz ortamda gelen toz halindeki bir malı makinanın dışına çıkarması mümkün değildir. Bu sebeple makine üzerine gelen malın diskler üzerine gelmeden önce nemlendirilmesi gerektirmektedir. Malın nemlendirilmesini sağlayan oransal vanadır. Yalnız nemlenen malın nem oranı ve sıcaklığı oldukça önemlidir. Makinanın üzerinde nemi yüksek mal çamur yaparak makinanın zorlanmasına sebep olmaktadır. Ayrıca makine üzerine gelen mal miktarının fazlalığı ise yine makinayı sırtlayan motorların yüksek amperde akım çekmesine sebep olmaktadır. Bundan dolayı makine üzerine gelen malın nem oranı ve gelen mal miktarını iyi ayarlanması gerekmektedir. Yani makine üzerindeki motorların akımlarına göre malın geliş hızı ve makinadaki malın nemine göre buharın verilmesi makinanın verimli çalışmasını sağlayacaktır.

4. PLC

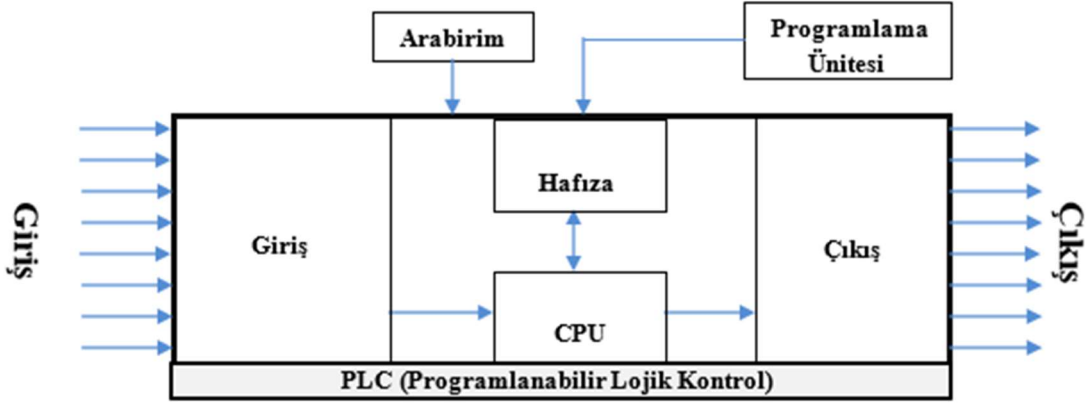
4.1. Genel Tanım

PLC tabiri Programmable Logic Controller İngilizce kelime öbeğinin kısaltmasıyla meydana gelmiştir. Ağır çalışma şartları içerisinde kullanılmak için üretilen PLC, endüstri ve makine otomasyonu açısından günümüzde vazgeçilmez teknolojik bir unsur haline gelmiştir. Son yıllarda Siemens, Festo, Omron, Mitsubishi, Fatek ve LG gibi birçok firma kendi ürettikleri PLC cihazlarını sahaya sürmüş ve bu sürülen cihazlarla endüstriyel sorunlara çözüm aramaya çalışmışlardır. Saha üzerinde endüstriyel otomasyonu sağlayan kumanda elemanları PLC ürünün ortaya çıkması ile işçiliğin azalması ve daha net kumanda devrelerinin ortaya çıkmasına sebep olmuştur. Endüstri kumanda sistemi üzerindeki etkisi nedeniyle PLC daima teknolojik gelişmelere açık olan bir cihaz grubu haline gelmiştir (Karayel, 2013).

Endüstriyel kumanda sistemlerinde geçmişte meydana gelen ihtiyaçlar ve sektördeki yetersizlikler genellikle bu sektörün gelişimine katkıda bulunmuştur. Analog kontrolle başlayan süreç yetersizlikler ve ihtiyaçları karşılayamama sebeplerinden dolayı dijital kontrol sistemleriyle çalışan bilgisayar olarak tabir ettiğimiz kontrol sistemlerine geçişine neden olmuştur. Ancak endüstriyel ortamdaki gelişimi sağlayan asıl olay mikroişlemci olarak isimlendirilen entegrelerin ortaya çıkmasıyla meydana gelmiştir. Zira bu entegreler vasıtasıyla programlama gerçekleştirilmiş olup daha esnek kontrol üniteleri ortaya konulması sağlanmıştır (Karayel, 2013).

4.2. PLC Cihazlarının Temel Yapısı

Bu tez çalışmasında Siemens'in S7-300 serisi olarak bilinen PLC cihazının üzerinde bir çalışma yapılacaktır. Bundan dolayı Siemens S7-300 PLC cihazını ve bu cihazı programlanmasını sağlayan Simatic Manager programı incelenecektir.



Şekil 4.1. PLC Temel Yapısı (Karayel, 2013)

Giriş kısmı, PLC cihazına sahadan gelen bilgilerin tam olarak toplandığı yerdir. CPU olarak adlandırılan mikro işlemciye ulaşmadan önce sahadan gelen bilgiler bu bölgeden karta aktarılır (Karan, 2017). Burada bulunan optakuplor vasıtası ile sahadan gelen bilgiler mikro işlemciye zarar vermeden ulaştırılmaktadır. PLC cihazları piyasada bulunan diğer mikro işlemci cihazlarından ayıran en önemli özelliği sahada meydana gelen parazit bilgilere karşı bir filtreleme özelliğidir. PLC cihazlar arasında değişen değerler olsa da 15 ms yakın bir değerde filtreleme özelliği yine giriş kısmında bulunmaktadır (Karayel, 2013). PLC üretim amacı endüstri sektörüne hitap etmesi sebebi ile PLC kartında kullanılan gerilim voltajı 24 V olarak belirlenmiştir. Bu yüzden saha üzerinde dolaşan her sensör PLC cihazının giriş kısmına 24 V olarak dönüş yapmaktadır. Buradan anlaşılacağı üzere PLC üzerinde lojik 1 olarak 24 V lojik 0 ise 0 V olarak kabul görmüştür. Bu tabir pnp giriş çıkış ünitelerine bağlı olarak dizayn edilmiş olan PLC cihazlarına aittir. Bazı PLC cihazları ise npn olup lojik 1 olarak 0 V lojik 0 olarak ta 24 V olarak görmektedir.

Endüstriyel ortamlarda kullanılan programlanabilir kontrol cihazı olan PLC her programlanabilir kartta bulunabileceği gibi hafıza kısmı bulunmaktadır. PLC cihazlarında RAM ve EEPROM olarak iki hafıza bölümü bulunmaktadır. RAM bölümü PLC cihazlarında içeriği kolaylıkla değişebilen hafıza kısmıdır. RAM CPU'da işlenmeden önce giriş kısmından gelen bilgileri tutulmasını sağlayan kısımdır. Yani CPU'nun bekleme odasıdır. EEPROM olarak belirtilen bellek ise RAM belleğinin tam tersi bir durumdadır. Zira EEPROM sahadan gelen bilgileri daima hafızasında tutan

bellek kısmıdır. Sahada oluşabilecek elektrik kesintilerine bağlı olarak önemli bilgiler EEPROM hafızası sayesinde saklanmaktadır (Yaşar, 2018).

PLC cihazlarının beyni olarak kabul gören CPU kısmı, hafıza bölümü üzerinde bulunan bilgilere göre karta çıkış veren bölümdür (Boşnak, 2016). Giriş kısmından gelen bilgileri toplayan RAM ya da EEPROM hafıza kısımları CPU'nun işlem hızına göre CPU'ya veri aktarımı yapmakta ve bu yapılan verilere göre CPU içinde işlemler yaparak çıkış ünitesine doğru veri gönderimi bu kısımda gerçekleşmektedir. Birbirinden farklı PLC grubunun CPU çalışma hızı farklı olsa da genellikle bu zaman dilimi 0.8 ms ile 0.15 ms arasında değişiklik gösterir (Karayel, 2013).

PLC cihazlarının bir diğer bölümü ise çıkış kısmıdır. Bu kısım genellikle teknolojinin gelişmesiyle zaman içerisinde farklılıklar görmemize neden olan bölümdür. İlk ortaya çıkan çıkış bölümleri röleli çıkış bölümleridir. Zamanla sistemin ihtiyacı ve teknolojik gelişmeler röleli çıkış olan sisteme ek olarak transistörlü çıkış eklenmesine neden olmuştur. Transistörlü çıkış yapan PLC cihazları endüstri sistemine uygun olarak 24 V bir çıkış gerilimi ortaya çıkarmaktadır. Röleli çıkış yapan PLC cihazları ise bu çıkış gerilimi 24 V ve 220 V ayarındadır. Günümüz teknolojisinde PLC güvenlik mantığı sebebiyle PLC röle çıkışları 24 V olarak üretilmektedir (Özer, 2016).

5. SCADA

5.1. Genel Tanım

SCADA (Supervisory Control and Data Acquisition) kelime öbeğinin kısaltılmasıyla meydana gelmiştir. İngilizce olan bu tabir Türkçeye dönüştürüldüğünde Denetimli Kontrol ve Veri Toplama olarak adlandırılmaktadır. Büyük ve geniş alan üzerindeki cihazların saha bilgilerini bir merkeze toplamak amacıyla meydana getirilen SCADA kurulduğu temel alarm ekseni olmuştur (Salğar, 2010).

SCADA sistemleri yapısal olarak incelendiği vakit kontrol edilen bölgelerin gösterdiği değişikliklere göre vücut bulduğu söylenebilir. Zira SCADA üzerinde fabrika ve buna benzer birleşik proseslerin kontrolü yapılacağına gibi trafo merkezleri gibi dağılık proseslerin kontrolü de gerçekleştirilmektedir (Kırmızıgül, 2014).

Dağılık olan sistemlerin tek bir merkezde kontrolünü sağlayan endüstriyel sistemler SCADA sisteminin bir diğer türevi olan DCS sistemleridir. SCADA ile DCS sistemleri arasındaki fark saha üzerinde toplanan bilgilerin merkeze aktarılması esnasında aracı olan haberleşme yöntemidir. DCS sistemlerinde gsm baz istasyonlarını kullanılmaktadır. SCADA sistemlerinde ise haberleşme protokolleri sonucunda doğmuş olan haberleşme kabloları yönündedir (Kırmızıgül, 2014).

SCADA tarihsel gelişimi incelenecek olursa güç endüstrisi üzerine çalışan AER firması tarafından 1971 yılında ilk olarak adının anıldığı belirtilebilir. Bu ad üzerinde çalışmaların ilk ürünü ise 1973 yılında Power Industry Computer Applications adı altındaki konferanstaki sunum ile tanıtılmıştır. Bu sunumda AER firması ile ortaya çıkan SCADA Fisher Corporation adındaki bir şirket tarafından DC2 bilgisayarı ile de ilk SCADA sistemini gün yüzüne çıkarmıştır (Karayel, 2013).

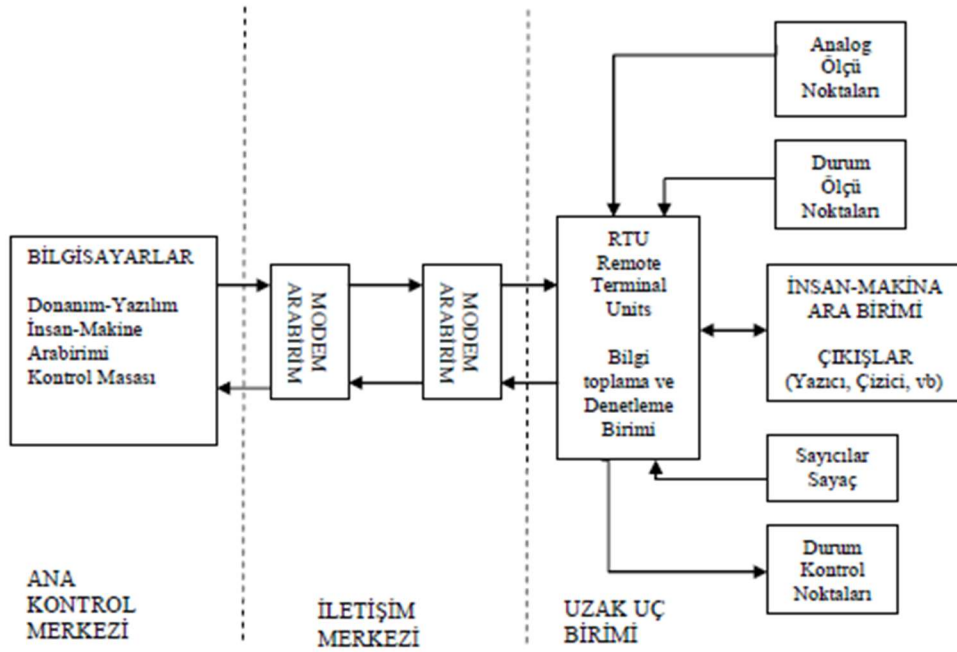
SCADA işlevsel olarak birçok özelliği bulunmaktadır. Bu özelliklerin hepsi daha önceden de belirtildiği üzere alarm temelinden geliştirilmiştir. Arıza takibi, reçete kontrolü, anlık analog değer takibi, trend ve grafik gösterimi, alınan verilerin veri tabanına aktarımı ve kaydedilmesi SCADA tarafından ön planda bulunan hizmetler olarak geçmektedir (Sarıkahya, 2013).

SCADA sistemlerinin uygulama alanları birden fazla noktada bulunmaktadır. Geniş alanlar üzerine kurulan sistemlerden daha küçük bölgesel proseslerin çoğu SCADA sayesinde kontrolü hayat bulmaktadır. Günümüzde Kimya Enstitüsünde,

Doğalgaz hatlarında, Ev otomasyonunda ve dahi trafik kontrolü gibi birçok noktada SCADA kullanılmaktadır (Üstünsoy, 2018).

5.2. SCADA Sisteminin Yapısı

SCADA temel yapısı üç ana kısımdan meydana gelmiştir. Bunlar MTU, RTU ve iletişim sistemleridir (Şen, 2018).



Şekil 5.1. SCADA sisteminin genel yapısı (Karayel, 2013)

5.2.1. MTU

Master Transfer Unit kelime öbeğinin kısaltması sonucunda oluşan MTU Türkçe karşılığı kontrol merkezidir. SCADA sisteminin çalışmasında kontrol edilen proses ile kullanıcı olan kişiler arasında ara yüzün meydana geldiği bölümdür ve SCADA sisteminde yönetici konumunda görev edinmektedir. Uzak uç birim olan RTU'dan gelen bilgileri arayüz ekranına aktarmasında, operatörden sisteme komut göndermesinde ve ayrıca SCADA'nın işlevlerini uygulamasıyla görevlidir (Kırmızıgül, 2014).

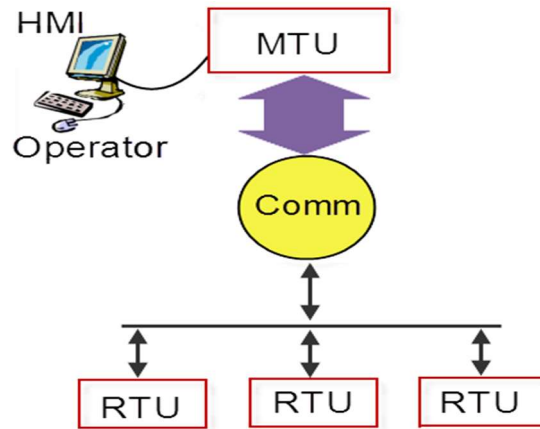
MTU'nun donanımsal yapısı kontrol edilen proseslerin yapısına göre değişiklikler göstermektedir. Genellikle MTU donanım yapısından istenen kritik süreçlerde anlık

olarak sistem bilgilerin kontrolü hızlı bir şekilde yapılanması sağlaması yönündedir. Ayrıca gerçek zamanlı bilgi takibi yapmak ve oluşabilecek bir arıza sürecinde operatörün müdahil olması gibi durumlar MTU donanımsal yapısında bazı değişiklikler oluşmasına neden olmaktadır. Bu değişiklikler kontrol bilgisayarı ile operatör sayısı arasında gerçekleşmektedir (Karayel, 2013).

Endüstriyel otomasyonda kullanılan SCADA sistemlerinde standart donanımlar genellikle operatörler, bilgisayarlar, HMI ve veri tabanlarından meydana gelmektedir. Bu süreçte ek olarak kontrol edilen sistem üzerinde manuel kullanıcılar için ayrıca mimik ekran olarak adlandırılan pano üstü kontrol tablosunda eklenebilmektedir.

Kontrol merkezinin yazılımsal sahadan gelen bilgilerin ve saha gönderilecek olan komutların neticesinde gerekli yazılımlar sonucunda sistemin işletilmesinin sağlandığı yapıdır. Bu yapıda bilgilerin ara yüzde gösterimi bilgiler ışığında alarm gösteriminin sağlanmasında, bilgilerin kaydedilmesi, kaydedilen bilgilerin bilgisayar localhostuna aktarılmasında ve reçete oluşturulması görev edinmektedir (Karayel, 2013).

Kontrol merkezi olarak adlandırılan MTU uzak uç birimler arasında haberleşmesi neticesinde görevini icra etmektedir. Bunun neticesinde kontrol merkezi üzerindeki haberleşme yapısı RTU ile gerçekleşmesi dağınık ve birleşik olan sistem proseslerine değişiklik göstermektedir. Genel olarak birleşik sistemlerde haberleşme yapısı LAN olarak kabul gören yerel haberleşme kabloları ile sağlanmaktadır (Karayel, 2013).



Şekil 5.2. SCADA MTU ve RTU yapısı (Anonim)

5.2.2. RTU

RTU, Remote Terminal Unit kelime öbeğinin kısaltmasıdır. Yaptığı görevler gözetilerek ele alındığında RTU'nun Türkçe karşılığı Uzaktan Bilgi Toplama ve Denetleme Birimi olarak çevrilebilir (Asubay, 2018).

RTU'lar yeteneklerine ve programlanabilir yapılarına göre iki grupta incelenmektedir. Programlama konusunda pasif olan RTU'lar temel yapı olarak ana programlanabilir RTU'ya sadece veri gönderme ve veri alma işlemini gerçekleştirmektedir. Programlama konusunda aktif olan RTU'lar ise diğer RTU grubunun görevlerini dışında programlama yeteneklerinde söz konusu olduğundan dolayı komutları algılama işlevleride söz konusudur (Karayel, 2013).

RTU'lar SCADA sistemleri açısından vazgeçilmez bir yapıda bulunmaktadır. Zira SCADA sistemlerinde RTU'lar sayesinde uzak olan sistemlerin kontrolü daha kolay olmakta ve sistem üzerindeki yükü hafifletilmesi sağlanmaktadır. Bundan dolayı bir sistem tasarlandığı vakit sistemin daha kolay olması açısından SCADA sistemlerinde RTU'lar daha etkili olarak ortaya konulmalıdır (Karayel, 2013).

RTU'lar altı adet bölüm üzerinden birleşmesiyle ortaya çıkmaktadır. Bunlar iletişim ünitesi, ana işlem ünitesi, giriş ve çıkış izalasyon ünitesi, kullanıcı arabirim ünitesi, test ünitesi ve güç kaynağı ünitesidir. Ana işlem ünitesi sadece programlama yapabilen RTU'larda bulunmaktadır (Karayel, 2013).

5.2.2.1. İletişim ünitesi

İletişim sistemleri SCADA üzerinde MTU ve RTU arasındaki haberleşme bağlantısını yapan önemli bir SCADA elemanıdır. Bu eleman birbirinden farklı uzak uç birimi ile haberleşmesi yanı sıra bu haberleşme sonucunda uzak uç birimlerine bağlanan farklı ekipmanların haberleşmesi de söz konusudur. Günümüzde farklı cihazlar için farklı haberleşme yöntemleri geliştirmiş ve üzerinde durulmuştur (Kırmızıgül, 2014).

RTU'da herhangi bir haberleşme metodundan gelen bilgiler RTU'nun iletişim ünitesinde RTU modülünün anlayabileceği kodlaşmaya dönüşür ve buna bağlı olarak merkezi işlem ünitesine bu ünitelerden gelen bilgilere göre işlem yaptırılması sağlanır (Kırmızıgül, 2014).

5.2.2.2. Merkezi işlem ünitesi

Merkezi işlem birimi saha üzerinden gelen bilgileri ya da SCADA üzerinden gelen komutları algılamasını yapan ve değerlendirmesine göre giriş çıkış izolasyon ünitesine komut gönderen RTU ünitesidir. Merkezi işlem ünitesi yine saha üzerinden gelen bilgileri sahadaki bilgiler değişinceye kadar RTU'nun hafızasında tutma işlemini yapması nedeniyle de ayrıca bir önemli görev yapmaktadır (Kırmızıgül, 2014).

5.2.2.3. Giriş çıkış izolasyon ünitesi

Birçok sahadan gelen bilgileri ve yine SCADA üzerinden gelen komut bilgilerinin merkezi işlem ünitesinden değerlendirildikten sonra sahada işleme konulmasını sağlayan ünedir. Bu ünite sahadan dönüş yapan 24V kullanılması yerine ayrı olarak sahadan çekilen 24V RTU üzerinden aktarımı sağlanmaktadır. Yani saha üzerinde meydana gelebilecek herhangi bir kısa devre ve gürültüden kartı etkilenmemesi için izolasyon kullanılması tercih edilmiştir (Kırmızıgül, 2014).

5.2.2.4. Kullanıcı arabirimi

Kullanıcı arabirimi RTU'nun içerisinde diğer birimlere karşı tıpkı bir kontrol merkezi gibi üniteleri incelemekte ve yönetmesi üzerinde görev edinmektedir. Teknolojik gelişmeler ve gereken taleplere cevap vermek için farklı bölgelerdeki proseslerin kontrolünü sağlayabilmek amacıyla RTU'ların içerisinde üreticiler zamanla kullanıcı arabirimi yükleyerek RTU'nun kendi içerisinde kontrol edebilmesini hedeflemişlerdir (Kırmızıgül, 2014).

5.2.2.5. Test birimi

RTU'da oluşabilecek herhangi bir arızanın ve görevini eksik yerine getirmesini sebebiyle test birimleri geliştirilmiştir. Genel görevleri arızaları takip etmek olan test üniteleri arızaların giderilmesi için RTU'lara gerekli yönlendirme yapmasında

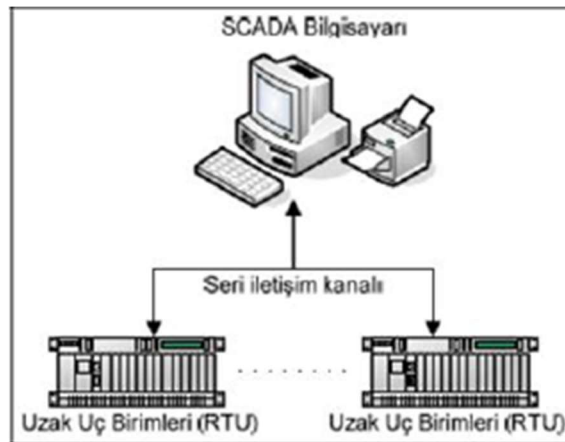
sağlamaktadır. Ayrıca RTU ile MTU arasında arıza dahi olsa haberleşmenin kesilmemesi için sistem içerisinde gerekli önlemleri almakla da yükümlüdür (Kırmızıgül, 2014).

5.2.2.6. Güç kaynağı

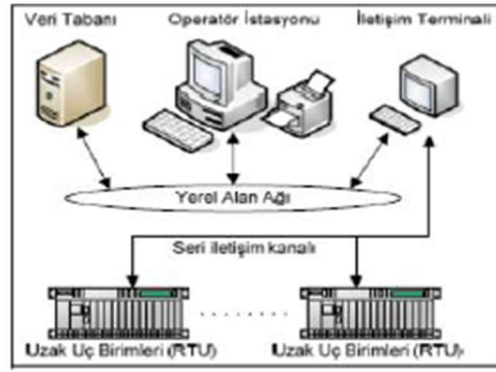
RTU'nun diğer bölümlerinin çalışmasını sağlayan bölümdür. Zira RTU üzerindeki elektriksel beslemeyi sağlamaktadır. Genel itibariyle RTU kartları 24 V olarak beslenmektedir. Ancak güç kaynağı beslemeleri 12 V'a düşebileceği gibi diğer güç elektroniği farkı ile tasarlanan RTU kartlarına mütakip 48 V'a kadar sistemin besleyici çıkmaktadır (Kırmızıgül, 2014).

5.2.3. İletişim sistemleri

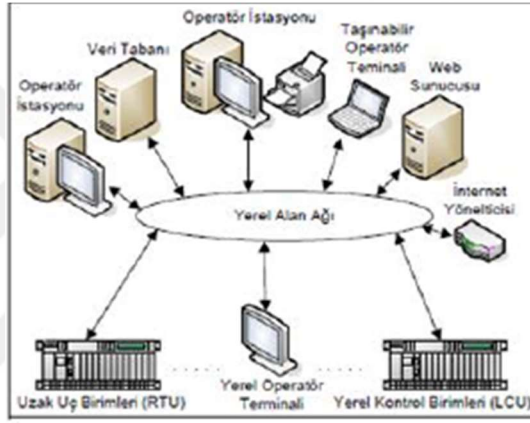
SCADA teknolojisinde MTU ile RTU arasındaki birbirleri arasındaki bağlantıyı sağlayan iletişim sistemleri SCADA'nın var olduğu dünden bugüne kadar üç aşamada kendini göstermiştir. Bu aşamalardan ilki sadece MTU ile RTU arasında gerçekleşen monolog bir haberleşme iken ilerleyen süreçlerin getirdikleri ve yazılım dünyasındaki yenilikler sonucunda dialog yani ikincil nesil haberleşme gelişimi ile beraber bilgisayar üzerinde veri tabanına MTU'dan bilgi gelmesi sağlanmıştır. Üçüncül nesil veri tabanı haberleşmesinde ise ikincil nesil haberleşmenin gelişimi ayrı bir periyoda girmiş olup RTU üzerindeki verilerin MTU'ya uğramadan direk olarak veri tabanına aktarımı sağlanmaktadır (Karayel, 2013).



Şekil 5.3. Birincil nesil SCADA iletişim sistemleri (Karayel, 2013)



Şekil 5.4. İkincil nesil SCADA iletişim sistemleri (Karayel, 2013)



Şekil 5.5. Üçüncül nesil SCADA iletişim sistemleri (Karayel, 2013)

6. WEB PROGRAMLAMA DİLLERİ

6.1. HTML

HTML, “HyperText Markup Language” kelime öbeğinin kısaltılmış halidir. Belgelerin, resimlerin birbirleri arasında hangi biçimde bağlanacaklarını ve nasıl biçimde konumlanacaklarını belirlemede kullanılan HTML kod sistemi internet üzerinde görsel etkileşimler yaratabilmek amacıyla kullanılmaktadır. HTML aslında bir programlama dili değildir. Programlama dilleri genellikle dış veriye ulaşmak amacıyla yazılmaktadır. HTML ise programlama dillerine göre verinin kendisidir.

HTML yapı itibariyle web teknolojisi ile oluşturulan ses ve resim formatlarına aktif alan oluşturmayı sağlamaktadır. Webspacede adlandırılan bu alanlar tipik ana sayfa ve diğer sayfalar arasında link olarak çalışmaktadır. Linkli alanlar webspacede teknolojisinde iki alanda incelenmektedir. Bunlar internal ve external webspacede alanlarıdır. Internal webspacede alanları sayfalar arasında geçişi sağlamak amacıyla kullanılmaktadır. External webspacede alanları ise bir internet sayfasından başka bir internet sayfası arasındaki geçiş amacıyla kullanılmaktadır.

HTML, tarihsel gelişimi ilk olarak 1980 yılındaki CERN araştırmacılarının bilgileri ve dökümanları birbirleri ile paylaşımaya açık hale getirebilmek amacıyla Tim Berners-Lee yaptığı çalışmaya dayanmaktadır (Başar, 2014). ENQUIRE isimli sistemi CERN laboratuvarında uygulayan Berners-Lee 1990 yılında World Wide Web sistemini meydana getirerek HTML işaretleme dilinin temellilerini atmıştır. 1990 yılından günümüze 5 tane HTML versiyonu oluşturulmuştur. Son versiyon HTML 5 üzerindedir. CSS sayfa stili oluşturma dilleri üzerinde etkili rol oynayan HTML’in en son versiyonu HTML 5 Youtube gibi sitelerde ve Chrome, Safari, Opera gibi tarayıcılarda kısmi olarak desteklenmektedir (Uslu, 2015).

HTML tag unsurlarının bir araya gelmesiyle çalışmaktadır. Taglar “<” “>” işaretlerinin arasında tanımlanmaktadır. Tag yazma biçimi Şekil 6.1’de gösterilmiştir.

`<p>Merhaba dünya</p>`

Şekil 6.1. HTML Tag Örneği

6.1.1. HTML ana tagları

HTML ana tagları sayfa objeleri oluşturmak için kullanılan tagların bir arada düzenli bir şekilde çalışmasını sağlamaktadır. Ana tag unsurları 5 adettir. Bunlar <!DOCTYPE>, <html>, <head>, <title> ve <body> etiketidir.

6.1.1.1. <!DOCTYPE>

<!DOCTYPE> tagı, internet tarayıcılar tarafından açılan sayfaların hangi türde olduğunu belirten tag unsurudur. Bu tag ile web sayfaları hangi versiyon ile yazıldığı hakkında internet tarayıcılarına yön vermektedir.

6.1.1.2. <html >

<html > tagı, internet tarayıcılar tarafından açılan html sayfasını olduğunu belirtmek amacıyla kullanılmaktadır.

6.1.1.3. <head>

<head> tagı, HTML işaretleme dilinde sayfanın hem fiziki bilgilerinin hem de yazılımsal bilgilerinin bulunduğu bölümdür. İnternet sayfalarının gerekli bilgilerini yazmak için kullanıla meta tagları, CSS stillerinin kullanabilmek için açılan <style> tagları ve internet üzerinden çekilecek harici bağlantıların hepsi <head> tagları içerisinde tanımlanmaktadır.

6.1.1.4. <title>

<title> tagı, HTML işaretleme dilinde sayfanın başlık metnini oluşturan etikettir. Bu tag aynı zamanda internet üzerinde araştırma yapılan sayfaların daha kolay bulunmasına olanak sağlamaktadır.

6.1.1.5. <body >

<body> tagi, HTML üzerinde sayfaların tasarlanması için gereken tag grubunun yazıldığı bölümdür.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title></title>
5 </head>
6 <body>
7
8 </body>
9 </html>
```

Şekil 6.2. Sublime Text Editöründeki HTML Ana Tagları

6.1.2. HTML meta tagları

HTML 'de yazılan internet sayfalarının internet tarayıcılar tarafından tanınması için internet sayfası yazılımında gerekli olmaktadır. Tarayıcılar aranan nesnelere hakkındaki bilgileri ve gerekli dökümanları internet siteleri hakkında gerekli tanımlamalardan almaktadır. Bu tanımlamalar tarayıcılar için bir protokol olması sebebiyle zamanla HTML taglarında meta olarak tanımlanan taglar ortaya çıkmıştır.

Meta tagları internet sayfaları hakkında fiziksel özellikleri belirtmekte kullanılmaktadır. HTML kod diziliminde <head> tagları arasına yazılmaktadır. Sayfaların hangi konu hakkında olduğu, hangi dil yapısında meydana geldiği sayfa içeriğinin responsive olup olmadığını, hangi bölgelerde sayfanın aktif olduğu gibi birçok tanım üzerinden internet tarayıcılarına bilgi vermektedir.

6.1.2.1. Content-type

Content-type, sayfa içerisinde kullanılan karakter kümelerinin hangi türe ait olduğunu belirtmekte kullanılır. Kullanılmadığı takdirde internet tarayıcıları tarafından bu sayfaların karakter kümesini ISO-8859-1 olarak kabul edilmektedir. Türkçe karakter kümesi utf-8 olarak tanımlanmıştır. Bu yüzden content-type meta tagları Türkçe internet siteleri için ayrıca kullanılmalıdır.

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

Şekil 6.3. Content type meta tagı yazım biçimi

6.1.2.2. Content-language

Content-Language sayfanın dili hakkında bilgi vermektedir. İnternet tarayıcılarının dil hakkında tarama yaparken özellikle site üzerinde yarar sağlamaktadır. Bu meta tagı sayesinde ilgili konu eğer o dil ile aranıyorsa bu sayfanın ön plana çıkması daha kolay şekilde meydana gelmektedir.

```
<meta http-equiv="Content-Language" content="tr" />
```

Şekil 6.4. Content Language meta tagı yazım biçimi

6.1.2.3. Description

Description, sayfanın hangi konu hakkında olduğu üzerine internet tarayıcılarına bilgi vermektedir. Bu tag ile beraber internet tarayıcılarının arama sayfalarında internet sitelerinin alt bölmesinde bulunan açıklama kısmını bu taglar vasıtasıyla ekrana yansıtılmaktadır.

```
<meta name="description" content="Spor haberleri hakkında 7 gün 24 saat sitemize ulaşabilirsiniz " />
```

Şekil 6.5. Description meta tagı yazım biçimi

6.1.2.4. Keywords

Keywords, sayfaların hangi anahtar kelimeler ile anılması gerektiği konusunda bilgi içermektedir. İnternet üzerinde arama yapan kullanıcının daha kolay sayfaya erişebilme hedefi üzerine bu meta tagı oluşturulmuştur.

```
<meta name="keywords" content="Spor,Futbol,Basketbol" />
```

Şekil 6.6. Keywords meta tagı yazım biçimi

6.1.2.5. Robots

Robots, tarayıcıların sayfalar üzerinde nasıl hareket edeceğini belirlemede kullanılmaktadır. Sayfaların yüksek index puanında ya da düşük index puanında tutulmasını sağlayan meta tagıdır. Bu şekilde sayfaların tarayıcılar üzerinde bulunmasını kolaylaştırıp zorlaştırılması sağlanabilmektedir.

```
<meta name="robots" content="index, follow" />
```

Şekil 6.7.Robots indexleme meta tagının yazım biçimi

```
<meta name="robots" content="noindex,nofollow" />
```

Şekil 6.8.Robots indexlememe meta tagının yazım biçimi

6.1.3.Önemli bazı HTML tagları

6.1.3.1. <!-- -->

Sayfa kaynağında açıklama olarak kullanılmaktadır. İleride yazılan kod dizilimin daha kolay bir şekilde hatırlanması için not alınmasını için görev verilmiştir.

6.1.3.2. <a>

Sayfa içerisinde bir link oluşturmayı sağlamaktadır. Bu linkler vasıtasıyla başka sayfalara geçişler yapılmaktadır.

6.1.3.3. <button>

İnternet sayfalarında input özelliği oluşturularak kullanıcılara sunulan giriş etiketidir. Üzerinde yapılacak gerekli düzenlemeler ile farklı biçimlerde düğme oluşturulmaktadır.

6.1.3.4. <canvas >

Sayfada bir tuval alanı oluşturmayı sağlamaktadır. Özellikle oyunlar, grafikler, animasyonlar gibi 3 boyutlu ve 2 boyutlu birçok alanda kullanılmaktadır. <canvas> etiketi Java script destekli çalışabilmektedir. Ayrıca bu etiketi kullanabilmek için internet tarayıcınızın HTML 5 uyumlu olması gerekmektedir.

6.1.3.5. <div >

Sayfa tasarımında tasarım parçalara ayrılmasını sağlayan etikettir. Tabloların kullanılmadığı sayfa tasarımlarında tabloların yerine görev almaktadır.

6.1.3.6. <figure >

Birçok medya grubu içeriğini belirlemek amacıyla kullanılmaktadır. Standart olarak figüre etiketi fotoğraf objelerini kapsayan etiket biçiminde görevlendirilmektedir. <figure> etiketi medya grubu etiketi olduğu için sıklıkla resim etiketi olan , video etiketi olan <video> gibi birçok etiket ile beraber kullanılmaktadır.

6.1.3.7. <footer >

Sitenin alt bölümünde bulunan içerik kısmını belirtmek amacıyla kullanılmaktadır. Sayfayı bitiren etiket olma özelliği olarak ta belirtilmektedir. Copyright içeriği yine bu bölümde belirtilmektedir.

6.1.3.8. <form >

İnternet sitesinde bulunan kullanıcıların işlem yapması için giriş etiketleri form etiketinin altında işlem yapmaktadır. Bu etikette <input>, <textarea>, <button>, <select>, <option> gibi etiketleri kullanabilmek için bir sektör oluşturmaktadır.

6.1.3.9. <h1>,<h2>,<h3>,<h4>,<h5>,<h6>

Başlık tanımlamayı sağlamaktadır. <h1> en önemli başlık olarak geçmektedir. <h6> ya gidildikçe başlığın önem düzeyi düşmektedir.

6.1.3.10. <hr>

Yatay çizgi oluşturmayı sağlayan etikettir. Çizgi üzerinde boyut, renk, uzunluk gibi nitelikler yine bu etikette tanımlanmaktadır.

6.1.3.11.

Resim tanımlamak için kullanılan etikettir. Resim içerisinde yine boyut, konum bu etiket içerisinde ayarlanmaktadır.

6.1.3.12. <label >

<input> etiketi için başlık tanımlamaktadır. Bu tag ayrıca kullanıcıların giriş yaptığı tag grubları için radiobutton, checkbox gibi kontrolleri etkin hale getirmek için kullanılmaktadır.

6.1.3.13. <map >

Harita etiketi olarak tanımlanmaktadır. <map> etiketi name özelliği sayesinde usemap özelliklerini eşitleyerek resim ile harita arasında bir ilişki oluşturmaktadır. Ayrıca <map> etiketi tıklanabilir alan oluşturmada <area> etiketi ile beraber kullanılmaktadır.

6.1.3.14. <menu>

Komut listesi oluşturmada ya da başka bir ifade ile komut menüsü oluşturmak için kullanılmaktadır. <menu> etiketi içerik menüleri, araç çubukları gibi komut listelerinden oluşmaktadır.

6.1.3.15. <menuitem>

<menu> etiketinde kullanılacak olan öğeleri belirtmekte kullanılmaktadır. Yazılan html yazılımlara ilave olarak eklenen Javascript ile beraber gelişmiş menü öğeleri yapılabilmektedir.

6.1.3.16. <nav>

<nav> etiketi menü içerisinde link vermeyi sağlayan etikettir. Menü ve diğer sayfalar arasındaki geçiş yapmak amacıyla kullanılmaktadır.

6.1.3.17. <p >

<p> etiketi html sayfa oluşumunda bir paragraf ya da yazı kümesi oluşturmak amacıyla kullanılmaktadır. Yine bu etiket üzerinde gerekli değişiklikler ile beraber yazı kümesinin stilini değiştirmekte mümkündür.

6.2. Php

Statik olan web tasarımı sayfaların dinamik hale getirilmesi sayfaların kullanıcıları tarafından işlem yapılabilmesi için gerekli olan bir husustur. Sunucu ile etkileşim halinde olan internet sayfalarda sitedeki kullanıcıların bilgi girişi yapması, internet sayfalarından bilgi çekebilmesi statik web sayfaların dinamik hale getiren gerekli programlar ile mümkündür (Başar, 2014). Bu programlar arasında günümüzde en sıklıkla kullanılan programlama dili Php programlama dilidir.

6.2.1. Php tarihsel gelişim süreci

Php, Web program geliştiricileri tarafından kullanılan ve web sayfalarının dinamik hale getirilmesi için üzerinde çalışılan bir script dilidir. Günümüzde incelendiği vakit yazılım uzmanları tarafından web geliştirmesi yönünde %67 oranında Php dili tercih edilmektedir (Kulga, 2017). Tarihsel süreç incelendiğinde Php dili ilk olarak yazılım dünyasının karşısına 1994 yılında çıkmaktadır. Rasmus Lerdorf adındaki bir internet sayfa tasarımcısının hazırlamış olduğu internet sitesinin görüntüleme miktarını belirlemek amacıyla C dili ile yazmış olduğu yazılım ilerleyen süreçte Php Tools 1.0 sürümü ile yazılım dünyasının karşısına gelmiştir (Şumnulu, 2005).

Rasmus Lerdorf 1995 yılında Php Tools 1.0 sürümü geliştirebilmek ve kütüphane kullanıcıların daha kolay hareket edebilmesini sağlayabilmek amacıyla 1995 yılında kütüphane içine veri tabanları ile etkileşimini kolaylaştıran başka bir taslak meydana getirmiştir. 1995 yılında ayrıca Lerdorf veri tabanı çalışmaları yanına ek olarak açık kaynak kod oluşturmayı sağlamış olup bu sayede oluşan hataların daha kısa sürede çözülmesini hedeflemiştir. 1995 yılında Lerdorf'un yapmış olduğu bu çalışmalar tarihe Php Tool 2.0 olarak geçmesi ile sonuçlanmıştır (Sönmez, 2014).

Binlerce kullanıcıya çıkan ve popüler bir dil olma yolunda ilerleyen Php 1997 yılında temel ayrıştırma motorunda oluşan ayrışma sorunu neticesinde Lerdorf Andi Gutmans ve Zeev Suraski ile ortaklaşarak daha ileri seviye bir program yazılması yönünde çalışmalarına devam etmiştir. Bu çalışmalar neticesinde 1998 yılında Php yeni bir sürüm olan Php 3.0 versiyonunun yayınlanması meydana gelmiştir. Ayrıca Php dilinin kısaltması değiştirilmiş olup kısaltmanın yeni açılımı Hypertext Preprocessor olarak ortaya konulmuştur (Sönmez, 2014).

Günümüzde Php dilinin kullanımını kolaylaştıran en önemli gelişme 2000 senesinde meydana gelen Php 4.0 sürümü ile ortaya çıkmıştır. Php dili üzerinde Zend motorunun temel alınması ve dilin tek kişi ile geliştirilmesinden çıkarıp Unix tabanlı geliştirecilere açık hale getirilmesi bugün milyonlarca yazılımcıya ulaşmasını sağlayan en önemli nokta olarak gösterilmektedir. Şu an Php sürümü 6.0 da bulunmaktadır. Php 4.0 sürümünden son sürümüne kadarki süreçte güvenlik hataları ve diğer hatalar üzerinde düzenlemeler yine Php 4.0 sürümü ile meydana gelen yenilikler neticesinde oluşturulmuştur (Sönmez, 2014).

6.2.2. Php çalışma prensibi

Php bir script dili olarak tanımlandığından dolayı çalışma prensibinde uzantılar önemli bir rol oynamamaktadır. Herhangi bir editör veya programa ihtiyaç olmadan yazılan Php çalıştırılabilmesi için gerekli duyduğu tek şey web sunucularıdır. Web sunucuları, tasarlanmış bir internet sayfasının tarayıcıda gösterilmesini sağlayan yayınlama aygıtıdır. Php gibi birçok yazılım dilini internet sayfalarının çalışmasını sağlayan sunucuların iki aşamada çalışma mantığını anlatmak mümkündür. Birinci aşama tarayıcıların arama sekmesine yazılan URL adresini alma işlemidir. İkinci aşama ise aldığı adrese göre yazılan HTML ve Php yazılım dilinde okunmasını sağlamak ve sunucudan istenen adresi yansıtmasını sağlamaktır (Başar, 2014).

Web sunucuları genel olarak istenen dosyayı HTML dosyası olarak kabul eder ve tarayıcılara yine HTML sayfaları olarak geri dönmelerini sağlar. Ancak eğer HTML sayfalarında `<?php >` ile karşılaşılan bir kod algılsa web sunucuları HTML kod okuma modundan Php mod okuma moduna dönüş yapmaktadır. Php üzerinde eğer 'echo' çıkış komutu görürse web sunucuları php modunda tarayıcıya o değeri gönderir ve bu sonsuz döngü devam eder (Başar, 2014).

`<p>Merhaba dünya</p>`

Şekil 6.9.HTML modu

`<?php echo"<p>Merhaba dünya </p>"; ?>`

Şekil 6.10.Php modu

Php Windows, Linux, OS ve Solaris gibi birçok işletim sisteminde çalışmasını sağlayabilen açık kod sistemidir. Yine Php'nin çalışabilmesi için Apache, IIS ve Zeus gibi birçok Php motoru bu işletim sistemlerinde çalışmasına yardımcı olmaktadır.

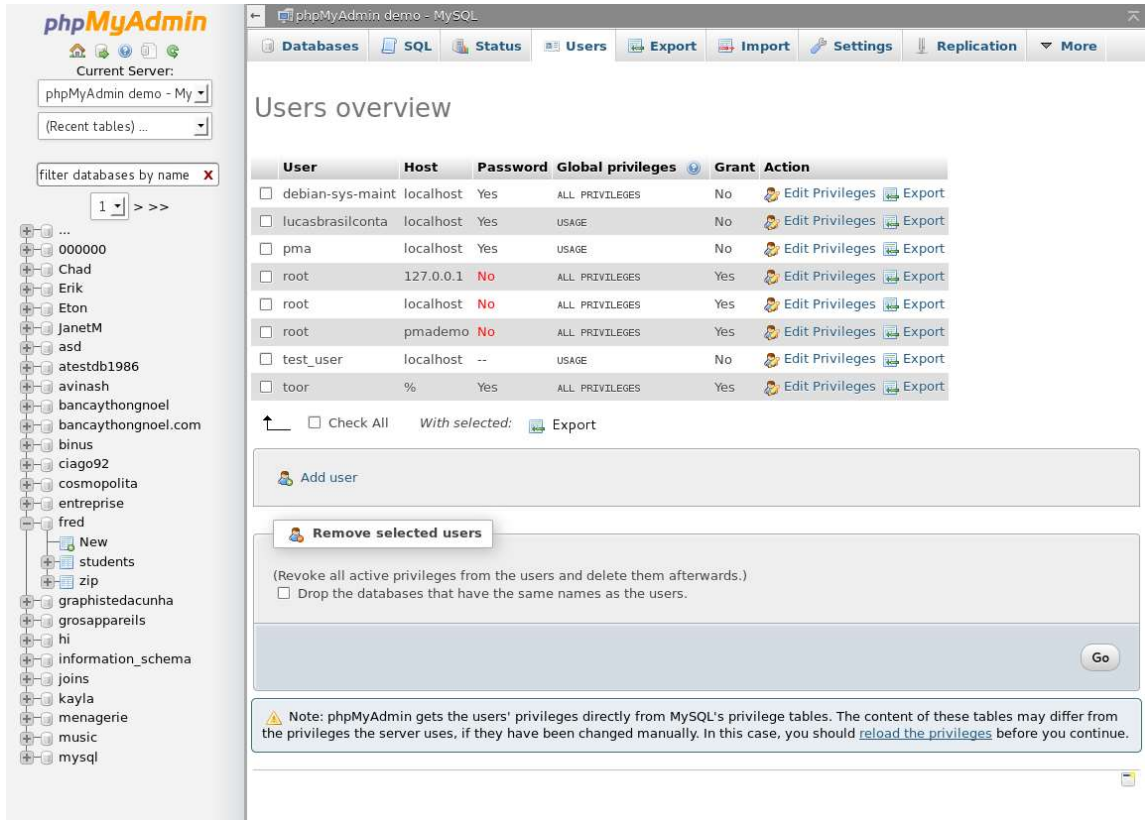
Php tüm web sunucuları ile entegreli bir halde çalışmasa dahi genel olarak yaygın olarak kullanılan web sunucularıyla yüksek performansta çalışmaktadır.

6.3. MySQL

Bilgisayar sistemlerinde birden çok bilginin beraber tutulması için kullanılan veri tabanı yerel ya da global tabanda bilgilere kolayca ulaşılmasını sağlayan bir bulut teknolojisi aygıtıdır. Günümüzde birçok veri tabanı bulunmaktadır. Oracle, PostgreSQL, Interbase, Mysql ve MsSQL sistemlerden birkaçıdır. Performans ve kullanım kolaylığından dolayı MySQL, Apache ve Php birçok yazılım bileşenlerine göre daha fazla tercih edilmektedir (Sönmez, 2014).

MySQL iki lisansta bulunan özgür bir yazılımdır. GPL yani Genel Kamu Lisansı ile sınırlı ve sınırsız yapılara bürünmektedir. MySQL performans açısından ticari olmayan yazılımlarda özellikle Linux, Windows ve IOS gibi işletim sistemlerinde kullanıcılara esnek çalışma alanı sunmaktadır. ASP Net, Php, Go ve Python gibi birçok dille uyumlu olarak çalışmaktadır (Sönmez, 2014).

PhpMyadmin MySQL olarak yazılan değişken değerlerinin kaydedilmesini sağlayan sütun bölümüdür. Şekil 6.11' de ekran görüntüsü gösterilmektedir. Kısaca MySQL veritabanı yazılımının ara yüzü olarakta belirtilebilmektedir (Sönmez, 2014).



The screenshot shows the phpMyAdmin interface for MySQL. The main content area is titled "Users overview" and displays a table of users. The table has the following columns: User, Host, Password, Global privileges, Grant, and Action. The data rows are as follows:

User	Host	Password	Global privileges	Grant	Action
<input type="checkbox"/> debian-sys-maint	localhost	Yes	ALL PRIVILEGES	No	Edit Privileges Export
<input type="checkbox"/> lucasbrasilconta	localhost	Yes	USAGE	No	Edit Privileges Export
<input type="checkbox"/> pma	localhost	Yes	USAGE	No	Edit Privileges Export
<input type="checkbox"/> root	127.0.0.1	No	ALL PRIVILEGES	Yes	Edit Privileges Export
<input type="checkbox"/> root	localhost	No	ALL PRIVILEGES	Yes	Edit Privileges Export
<input type="checkbox"/> root	pmademo	No	ALL PRIVILEGES	Yes	Edit Privileges Export
<input type="checkbox"/> test_user	localhost	--	USAGE	No	Edit Privileges Export
<input type="checkbox"/> toor	%	Yes	ALL PRIVILEGES	Yes	Edit Privileges Export

Below the table, there are buttons for "Check All", "With selected:", and "Export". There are also buttons for "Add user" and "Remove selected users". A note at the bottom of the interface reads: "Note: phpMyAdmin gets the users' privileges directly from MySQL's privilege tables. The content of these tables may differ from the privileges the server uses, if they have been changed manually. In this case, you should reload the privileges before you continue."

Şekil 6.11. MySQL Ekran Görüntüsü (Anonim)

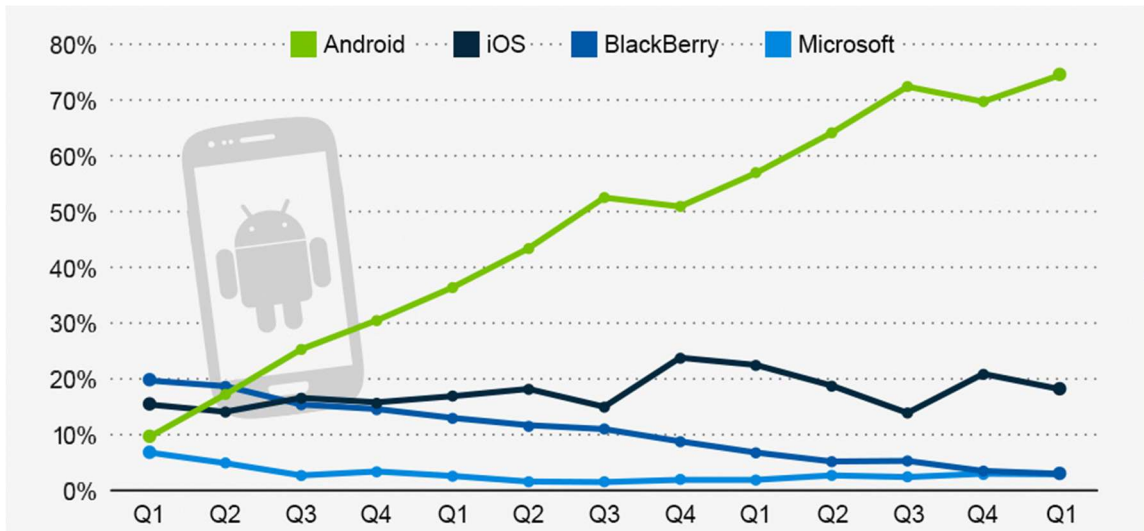
6.3.1. XAMPP

Yazılım geliştirme uzmanlarının çalıştıkları projeleri yerel bir web sunucusu oluşturabilmek amacıyla kullandıkları apache datacı aygıtıdır. Kelime açılımına bakılırsa XAMPP her harfinin bir anlam içerdiği görülmektedir. Öyle ki X ifadesi çapraz bir yapıdan meydana geldiğini anlatmaktadır. A harfi yine Apache motorunu anlatmakta olup M harfi MySQL veri tabanını, P harfleri ise biri Php yazılımını diğeri ise Perl programlama dilini anlatmaktadır.

Çapraz platform olarak tanımlanmasını sağlayan X harfi farklı işletim sistemlerinde yazılan yazılımın aynı seviye ve performasta yazıldığını belirtmeyi sağlamaktadır. Perl olarak bilinen programlama dili ise pek sıklıkla kullanılsa dahi program içerisinde metin ve görüntü tanımlamada güçlü üst düzey bir programlama dili olarak bilinmektedir.

6.4. Android İşletim Sistemi

Mobil işletim sistemleri günümüzde bilgisayar işletim sistemlerine göre daha fazla kullanılmaktadır. Kullanıcıların istek ve önerilerine göre açık kaynak kod sistemi ile yazılan Mobil işletim sistemleri Linux tabanlı bir geliştirmeler ışığında Android adı verilen bir işletim sistemine dönüşümüne neden olmuştur.

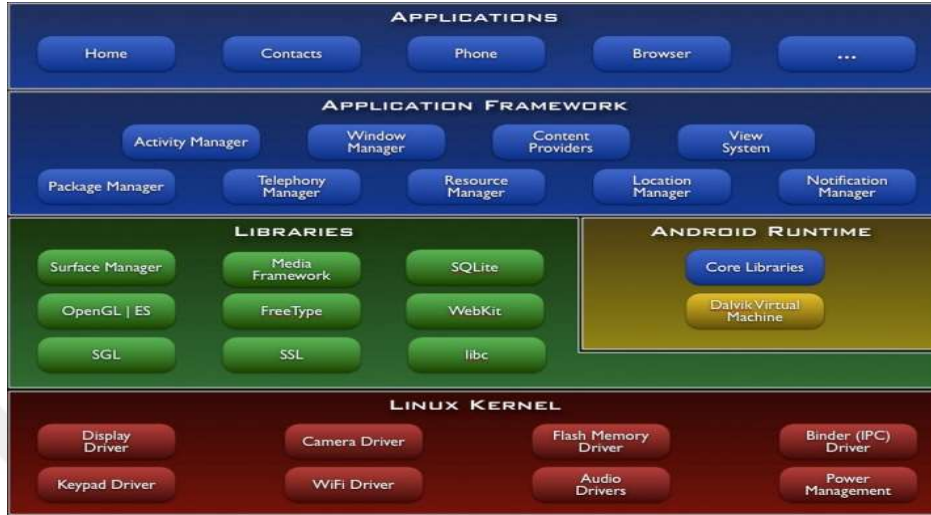


Şekil 6.12. 2010 ile 2013 yılları arası mobil işletim sistemi dağılımı (Anonim)

Mobil cihazlar için kullanılan açık kaynak kod ve geniş kütüphane yapısı olarak bilinen Android işletim sistemi günümüz gelişmeler neticesinde herhangi bir işlemci üzerinde çalışabilecek noktaya getirilmiştir. Mimari olarak dört katmandan oluşmakta olan android işletim sistemi beş uygulama parçası ile beraber kullanıcı ara yüzü meydana getirmektedir (Kayabaşı, 2016).

Android işletim sistemleri oluşturulmasını sağlayan Linux, mobil cihazların donanım driverları, bluetooth ve Wi-fi gibi yazılımlar Android işletim sistemi mimarisinin en alt katmanını oluşturmaktadır. İkinci katmanda Android kütüphaneleri ile beraber Android yürütücüsü olan çekirdek kütüphaneleri ve Dalvik Sanal Makinası bulunmaktadır. Dalvik kütüphanesi Java sanal makinesine benzeyen bir android uygulamalarının yürütülmesini sağlayan bileşendir. Her android uygulama içerisinde bu sanal makine bulunup her uygulama için ayrı ayrı bir alan (sandboxing) ortaya çıkarmasını sağlar. Üçüncü katman uygulama çatısı olarak bilinen uygulamalarında

kullanabileceği servisler bulunmaktadır. Dördüncü ve en son katman ise telefon görüşmesi yapma, mesaj atma, internette araştırma yapma gibi ana kullanıcı uygulamaları bulunmaktadır (Biçek, 2016).



Şekil 6.13. Android İşletim Sistemlerinin Katmanları (Kayabaşı, 2016)



Şekil 6.14. Uygulamadaki Android Sayfa Tasarımı

7. MATERYAL VE YÖNTEM

Bu uygulamada endüstri prosesinin SCADA haricinde yeni kontrol panelleri talebi neticesinde android ve internet sayfası kontrolüne sunulması gerçekleştirilmiştir. Bu çalışmanın oluşturulmasındaki ana sebep endüstri proseslerin kontrolünü kolaylaşmasını sağlamaktır. Uygulama Simatic Manager, Wincc Explorer ve Hosting olmak üzere üç ana bölümden meydana gelmektedir.

7.1. Simatic Manager

Simatic Manager programı Siemens tarafından PLC cihaz grubunu programlayabilmek için geliştirilmiş hali hazırda versiyonları yenilenen paket programıdır. Manager 2011 yılında Tia Portal programı adı altında Siemens'in yeni çıkarmış olduğu paket programına karşın kullanımının kolay olması ve daha hızlı olması sebebiyle otomasyon sektöründe halen tercih edilmektedir.

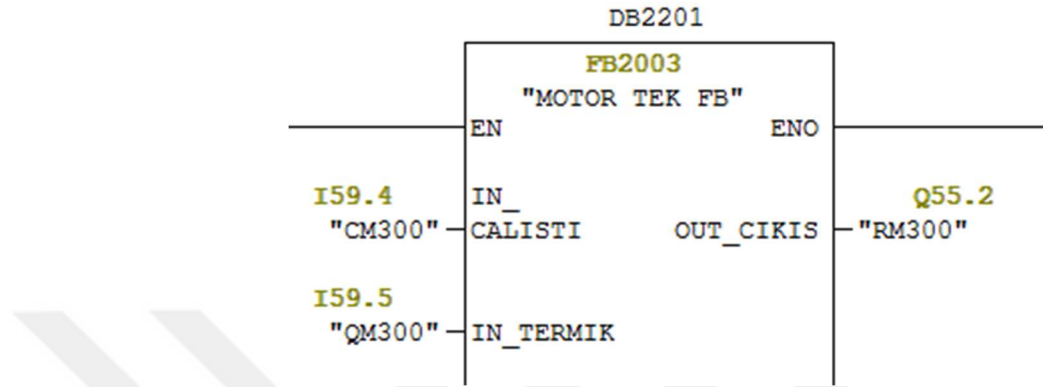
Program blok mantığında çalışmaktadır. Her blok birbirleri içerisinden geçerek en son organizasyon bloğu diye tabir edilen bloğun içine yazılmaktadır. PLC cihazındaki işlemci ise bu organizasyon bloğuna girerek baştan aşağıya doğru tarama yapmaktadır ve yazılan program sonucunda ortaya çıkan sonuçları PLC cihazının çıkış bölümüne yazmaktadır.

7.1.1. Fonksiyon bloğu

Fonksiyon bloğu yazılım içerisindeki en küçük yapı birimidir. Zira burada yazılan yazılımlar genellikle saha üzerindeki elemanların çalışmasını sağlayan parametrelerini bulundurmaktadır. Örneğin saha üzerinde bir asenkron motorun çalışmasını sağlayan parametreler motor bloğu olarak belirlenen fonksiyon bloğunda toplanır ve bu programda yazılan yazılımlara göre motorun çalışması sağlanır. Fonksiyon bloğu program yapısı içerisinde zorunlu olarak yazılacak bir program nesnesi değildir. Ancak programın kendi içerisinde daha düzenli olması ve her saha elemanın daha emniyetli çalışması için fonksiyon bloğu çoğu kullanıcı tarafından tercih edilmektedir.

Fonksiyon blokları aslında saha üzerindeki kullanılan elemanların bilgilerinin kaydeden database sayesinde çıkış vermesini sağlayan nesne grubudur (Working with

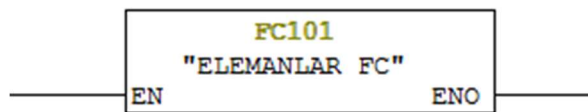
STEP7, 2010). Bu nedenle program içerisinde oluşturulmuş bir fonksiyon bloğu kendisine ait veri tabanı olarak belirtilen database program nesnelere bulundurmaktadır. Şekil 7.1.'de gösterildiği gibi veri tabanları saha üzerindeki toplanan en son gelen bilgileri hafızada tutmakta ve fonksiyon bloklarının işleyişini sağlamaktadır.



Şekil 7.1. Fonksiyon Bloğu Kesiti ve Database Görünümü

7.1.2. Fonksiyon

Fonksiyon program yapısı içerisinde çoğu kullanıcı tarafından yazılımın yazıldığı bölümdür. Ayrıca kullanıcılar tarafından fonksiyonlar fonksiyon blokları olarak adlandırılan program nesnelere çağrıldığı bölümdür (Working with STEP7, 2010). Organizasyon blokları içerisinde yazılan fonksiyonlar Şekil 7.2.'deki gibi bulunmaktadır.




Şekil 7.2. Fonksiyonun Organizasyon Bloğundaki Görünümü

7.1.3. Database

Program yapısı içerisinde database yani veri tabanının önemi ayrıdır. Zira database fonksiyon bloklarında anlatıldığı gibi blok parametrelerinin üzerindeki en son veri değerlerinin kaydedilmesinde kullanılırken ayrıca program içerisinde kullanılan

parametrelerdeki bilgilere isim vererek bu bilgilerin program içerisinde başka bölümlerde kullanılmasına da olanak sunmaktadır (Working with STEP7, 2010).

7.2. Simatic Manager Programını Simülasyona Alma

Siemens endüstri paket programlarında yazılımcıların yazmış olduğu yazılımları daha kolay kontrol edebilmeleri için simülasyon kısmına ayrı bir önem vermektedir. Program üzerinde herhangi bir makinanın senaryosunu deneyebilmek ve yazılan senaryonun verdiği reaksiyonunu gözlemleyebilmek için kullanılan simülasyon birçok programcı tarafından kullanılmaktadır. Simatic Manager programında simülasyon moduna geçebilmek için ilk ara yüzünde  ikonuna tıklamak gereklidir. İkona tıklanmasıyla açılan faceplate vasıtasıyla yazılan program Siemens'in bir PLC cihazı üzerinde çalıştırmak yerine PLCSIM yani sanal PLC üzerinde çalışmasına olanak sağlamaktadır.

Programı Simatic Manager'ın sanal PLC ara yüzüne yüklerken dikkat edilecek iki önemli husus vardır. Bunlardan birinci Simatic Manager programı içerisindeki Set PG/PC İnterface ayarının PLCSIM.TCP.1 olarak belirlenmesidir. Programdaki bu ayar kullanılan PLC cihazı ile programın yüklü olduğu bilgisayar arasında yapılacak haberleşme tercihinin seçilmesi sağlamaktadır. Yapılan çalışma simülasyonda gerçekleşen bir sistem olduğu için PLCSIM.TCP.1 haberleşme yöntemi tercih edilmiştir.

7.3. PLC'den SCADA'ya Veri Aktarımı

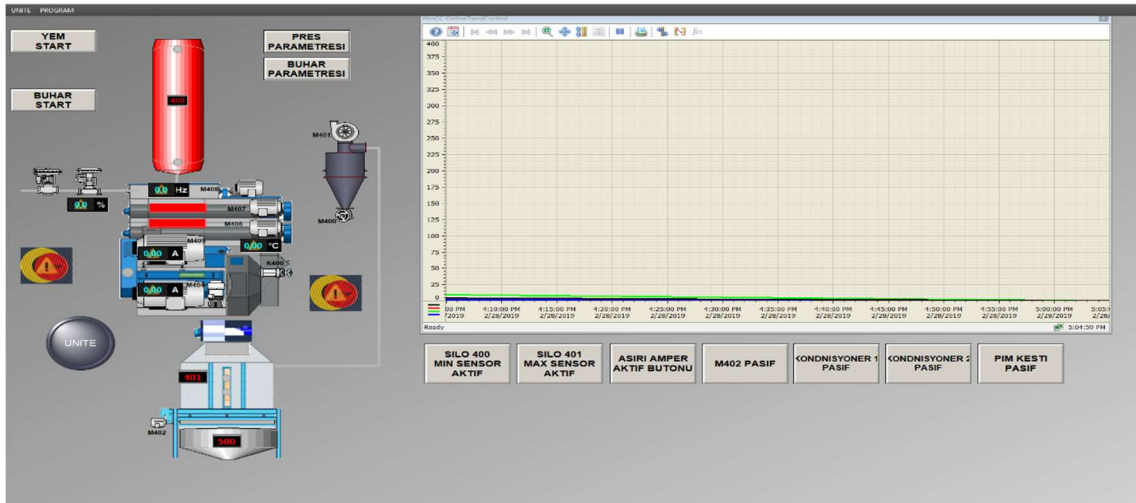
PLC ile SCADA arasındaki haberleşme tag olarak tabir edilen bilgi bitleri arasında yapılmaktadır. SCADA PLC tarafından aldığı taglara göre belirtilen görsellikleri sağlamaktadır. Siemens endüstri paket programında bu tagların bir arada daha kolay olarak veri transferi sağlayabilmek amacıyla DB adı verilen tag kümeleri meydana getirmiştir. PLC üzerinde yüklenen yazılımlar sahadan gelen bilgiler neticesinde parametrelerin aldıkları değerler DB'lerde tutulması sağlanmaktadır. Uygulama için yazılan simülasyon programında yine SCADA'ya bilgi haberleşmesi DB kümelerinden yararlanılmıştır. Şekil 7.3' de programda kullanılan DB içi gösterilmiştir.

Address	Name	Type	Initial value
0.0		STRUCT	
+0.0	M400_CALISMA_BILGISI	BOOL	FALSE
+0.1	M401_CALISMA_BILGISI	BOOL	FALSE
+0.2	M402_CALISMA_BILGISI	BOOL	FALSE
+0.3	M404_CALISMA_BILGISI	BOOL	FALSE
+0.4	M405_CALISMA_BILGISI	BOOL	FALSE
+0.5	M406_CALISMA_BILGISI	BOOL	FALSE
+0.6	M407_CALISMA_BILGISI	BOOL	FALSE
+0.7	M408_CALISMA_BILGISI	BOOL	FALSE
+1.0	K400_ACIK_BILGISI	BOOL	FALSE
+1.1	OV2_ACIK_BILGISI	BOOL	FALSE
+1.2	YEM_START	BOOL	FALSE
+1.3	BUHAR_START	BOOL	FALSE
+1.4	UNITE_START	BOOL	FALSE
+1.5	UNITE_STOP	BOOL	FALSE
+1.6	UNITE_BEKLE	BOOL	FALSE
+1.7	UNITE_DEVAM	BOOL	FALSE
+2.0	UNITE_ACIL_STOP	BOOL	FALSE
+2.1	SIL0_400_MIN	BOOL	FALSE
+2.2	SIL0_401_MAX	BOOL	FALSE
+2.3	KODNISYONER_KAPAK_1	BOOL	FALSE
+2.4	KODNISYONER_KAPAK_2	BOOL	FALSE
+2.5	PIM_KESTI	BOOL	FALSE
+4.0	MOTOR_AKIM_1	REAL	8.000000e+001
+8.0	MOTOR_AKIM_2	REAL	8.000000e+001
+12.0	BESLEYICI_CALISMAYI_KEST	BOOL	FALSE
+12.1	HATA	BOOL	FALSE
+12.2	K400_KLAPES_PRES_PASIF	BOOL	FALSE
+12.3	KODNISYONER1_KAPAK_ACIK	BOOL	FALSE
+12.4	KODNISYONER2_KAPAK_ACIK	BOOL	FALSE
+12.5	MOTOR_SARMA	BOOL	FALSE
+12.6	RESET	BOOL	FALSE
+12.7	SIL0_400_BOS	BOOL	FALSE
+13.0	SIL0_401_DOLU	BOOL	FALSE
+14.0	SICAKLIK	REAL	7.000000e+000
+18.0	BUHAR	REAL	7.000000e+000
+22.0	BESLEME	REAL	8.000000e+000
+26.0	ACIL_STOP	BOOL	FALSE
=28.0		END_STRUCT	

Şekil 7.3. Uygulamadaki Database Objesinin Görünümü

7.4. Wincc Explorer

Siemens tarafından 1994 yılından itibaren PC sistemler için geliştirmiş olduğu Wincc Explorer insan ara yüzü sistemidir. SCADA sistemi olarak kullanılan Wincc Explorer panel programlarına göre daha gelişmiş bir yapıda bulunmaktadır. Wincc Explorer programı Şekil 7.4’de gösterilmektedir.



Şekil 7.4. Uygulamadaki SCADA Ara yüzü

7.4.1. Wincc Explorer'ın ana unsurları

7.4.1.1.Computer

Computer bölümü, programın kurulduğu bilgisayar ile uyumlu çalışabilmesi için gerekli yapılandırma ayarların bulunduğu bölümdür. Bu bölümde programın kurulduğu bilgisayar ile program arasındaki ad uyumluluğunun ayarı, bilgisayar açıldığında programın çalışır duruma geçme ayarı, bilgisayar içerisinde çalışma saatinin çekilme ayarı, açılan sayfaların monitöre aktarım biçim ayarı, programda menü toolbar izni ayarı ve program içerisinde mouse hareket şeklinin belirlenme ayarı bulunmaktadır (Wincc V7.4 Simatic Hmi Wincc V7.4 Getting Started, 2016).

7.4.1.2.Tag management

Tag Management program içerisinde yapılan gerekli ayarlar neticesinde haberleştiği PLC cihazındaki data bilgilerine ad verilmesini sağlayan bölümdür. Yine SCADA ile sistemin haberleşme ayarları bu bölümde yapılmaktadır. Haberleşme ayarları bölümünde SCADA için gerekli olan Can Bus, TCP/IP, MPI, gibi haberleşme seçenekleri bulundurmaktadır. Ayrıca program içerisinde kullanılacak olan tagların internal veya extarnel olarak kullanılmasını sağlayacak olan ayarlarda Tag Managment bölümünün içerisinde bulunmaktadır (Wincc V7.4 Simatic Hmi Wincc V7.4 Getting Started, 2016).

7.4.1.2.1.Structure

Tag Management bölümünün alt sekmesi olan görev yapan Structure, aynı görev yapan elemanların tag yapılarını ayrı ayrı yazmak yerine gruplaştırarak yazılmasını sağlamaktadır. Structure sayesinde tagların yazılması daha kolay olmaktadır ve kontrolü daha esnek olmaktadır.

7.4.1.3.Graphics desinger

Graphics Desinger bölümü, SCADA içerisindeki görselliklerin oluşturulmasında gerekli olan sayfaların tasarımının yapıldığı bölümdür. Bu bölümde sayfalardaki obje flashlandırma, obje renklendirme gibi işlemler yapılmaktadır (Wincc V7.4 Simatic Hmı Wincc V7.4 Getting Started, 2016).

7.4.1.4.Menus and toolbars

Programda tasarlanan sayfaların geçişi için gerekli olan menülerin oluşturulduğu bölümdür. Bu bölümde tasarlanan menüleri aktif etmek için computer bölümündeki menü toolbar yapılandırma ayarları düzenlemek gerekmektedir (Wincc V7.4 Simatic Hmı Wincc V7.4 Getting Started, 2016).

7.4.1.5.Alarm loging

Sistemde oluşabilecek hataların operatörleri uyarabilmesi için hata mesajlarının hatalarla etiketlenilmesi bu bölümde gerçekleştirilmektedir. Ayrıca bölüm içerisinde gelen hata bayrak bilgilerinin tarih zaman ayarları, bilgilerin biçimlerine göre filtrelenmesi gibi ayarları da yine alarm loging bölümünde yapılmaktadır (Wincc V7.4 Simatic Hmı Wincc V7.4 Getting Started, 2016).

7.4.1.6. Report desinger

Report Desinger, program içerisinde tutulan verilerin bir düzen içerisinde yazıcıya aktarılmasına olanak sağlayan bölümdür. Genellikle program içerisinde raporlama işleminin yapılmasında kullanılmaktadır. Bu özellik verilerin SCADA üzerinde tutulmasını ve istenildiği takdirde yazıcının üzerinden kullanıcıların verileri almasını sağlamak yönündedir (Wincc V7.4 Simatic Hmı Wincc V7.4 Getting Started, 2016).

7.4.1.7. Text library

Tasarlanan SCADA programındaki kelimelerin sözlüğü olarak görev almaktadır. SCADA üzerinde dil seçimi oluşturulmak istenirse farklı diller arasındaki dönüşüm bu bölüm ile beraber sağlanmaktadır (Wincc V7.4 Simatic Hmı Wincc V7.4 Getting Started, 2016).

7.4.1.8. Text distributor

SCADA üzerinde oluşturulan bir proje bu bölüm yardımıyla .txt ve .csv dosya uzantılarına dönüştürülmektedir. Özellikle dışarıdan SCADA programına aktarılması gereken taglar için kullanılmaktadır.

7.4.1.9. User administrator

Her programda olabileceği gibi SCADA'da bazı bölümlerin kontrolünü kişiye özel yapmak gerekebilmektedir. Bu bölüm sayesinde bazı komutların ve girişlerin değişimlerini kontrol eden kişilere karşı özelleştirme yapılabilmektedir. Admin ve User olmak üzere iki adet farklı kullanıcı tipi ile yönetilecek sayfanın önemine göre kullanıcı girişleri için sayfalara şifreler konulmaktadır. Admin genellikle sayfaların tümüne hakim olan kişi anlamına gelirken User ise daha az yetkiye sahip kişi anlamına gelmektedir.

7.4.1.10. Picture tree manager

Sayfalar arasındaki hiyerarşinin belirlenmesini sağlayan bölümdür. Bu bölümde gerekli düzenlemeler ile beraber sayfaların birbirlerinin arasında geçiş için kullanılacak olan butonlar otomatik olarak oluşmaktadır.

7.4.1.10. Global script

Wincc Explorer programında mantık kuramlarının yetersiz kaldığı yerlerde devreye giren yazılım bölümüdür. Program üzerinde iki farklı dilde script türü bulunmaktadır. Bunlar Vbs Script ve C Script'tir.

Programda kullanılan yazılım dilleri gerçek zamanda belirlenen dilimlere göre çalışmaktadır. Cycle süreleri olarak tanımlana bu zaman dilimlerine göre yazılan yazılımlar görevlerini daima tekrar ederek tamamlanmaktadır (Wincc V7.4 Simatic Hmi Wincc V7.4 Getting Started, 2016).

```

Sub InvertirBitEnVariable (Byval TagName, Byval bit)

    Dim Tag
    Set Tag = HMIRuntime.Tags(TagName)

    If Tag.Read And (2 ^ bit) Then           'Bit es True
        Tag.Write(Tag.Read And Not (2 ^ bit)) 'Reset
    Else                                     'Bit is False
        Tag.Write(Tag.Read Or (2 ^ bit))     'Set
    End If

End Sub

Sub InvertirBitEnVariableWait (Byval TagName, Byval bit)

    Dim Tag
    Set Tag = HMIRuntime.Tags(TagName)

    If Tag.Read(1) And (2 ^ bit) Then       'Bit es True
        Tag.Write(Tag.Read(1) And Not (2 ^ bit)),1 'Reset
    Else                                     'Bit is False
        Tag.Write(Tag.Read(1) Or (2 ^ bit)),1   'Set
    End If

End Sub

```

Şekil 7.5. SCADA'daki Vbs diline bir örnek

7.4.1.11. Life beat monitöring

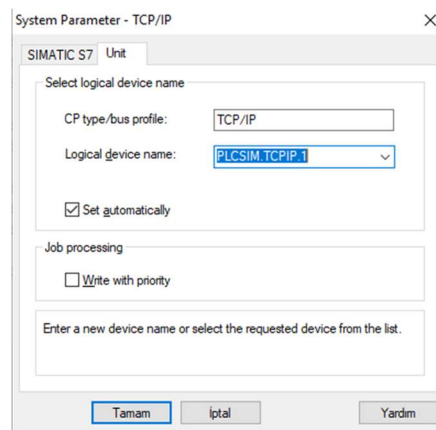
Programın ET modülleri ve diğer cihazlar arasındaki haberleşme durumunu denetlemektedir. Haberleşme üzerinde oluşabilecek kopmaların hangi noktada meydana geldiği konusunda bu bölüm gözlemleyiciye yardımcı olmaktadır. RTU ile MTU arasındaki etkileşim yine bu bölümden sağlanarak gerçekleşmektedir.

7.4.1.12. Cross reference

Programda tag tarayıcısı olarak geçmektedir. Oluşturulan sayfalar içerisindeki objelerin hangi tag etiketine bağlı olduğunu bu tarayıcı tarafından yardım olarak bulunmaktadır. Böylelikle sayfalarda kullanılmayan tagların bulunması taglara hakim olunması daha da bu bölüm kolaylaştırmaktadır.

7.5. SCADA'yı Simülasyona Alma

Wincc Explorer programı çekeceği verilerin hangi driver ile aktarılacağına göre çalışmaktadır. Bilgisayar üzerinde bulunan Ethernet Driverları, TS Adapter IE, MPI gibi seçenekler bulunabileceği Wincc Explorer'da simülasyon durumunda PLCSIM.TCPIP.1 seçeneği bulunmaktadır. Tag Management bölümünde bulunan System Parameters'den ayarlanan haberleşme driverları sistemin durumuna göre değişiklik meydana gelmektedir. Şekil 7.6' da yapılan çalışama için ayarlanan System Parameters bölümü gösterilmektedir.



Şekil 7.6. System Parameters Ayarı ile Simülasyona Sistemin çekilmesi

7.6. PLC'den SCADA'ya Veri Aktarımı

PLC'den SCADA'ya veri çekme işlemi yine endüstriyel haberleşme protokolleri ile meydana gelmektedir (Wincc V7.4 Simatic Hmı Wincc V7.4 Getting Started, 2016). Uygulama üzerinde kullanılan program olan Wincc Explorer PLC'den gelen verileri Tag Management'ta oluşturulan grup bölmesine yazılmasıyla eşleşmektedir. Grup bölümünde yazılan tag isimleri tag adresleri ile birebir olup parametrik değerler yine SCADA'nın bu bölümünden SCADA sayfalarına ve diğer uygulamalarına aktarılmaktadır. Şekil 7.7' te uygulama için yazılan yazılımın PLC üzerinden çekilen verilerin Tag Management grubu gösterilmektedir.

Tags [GONDERIM]							
Name	Comment	Data type	Length	Format adaptation	Connection	Group	Address
1	ACIL_STOP	Binary Tag	1		PLC	GONDERIM	DB110,D26.0
2	BESLEME	Floating-point number 32-bit IEEE	4	FloatToFloat	PLC	GONDERIM	DB110,D022
3	BESLEYICI_CALIS_KESTI	Binary Tag	1		PLC	GONDERIM	DB110,D12.0
4	BUHAR_DEGERI	Floating-point number 32-bit IEEE	4	FloatToFloat	PLC	GONDERIM	DB110,DD18
5	BUHAR_START	Binary Tag	1		PLC	GONDERIM	DB110,D1.3
6	HATA_OLUSTU	Binary Tag	1		PLC	GONDERIM	DB110,D12.1
7	K400_ACIK_BILGISI	Binary Tag	1		PLC	GONDERIM	DB110,D1.0
8	K400_KLAPE_PASIFE_GECTI	Binary Tag	1		PLC	GONDERIM	DB110,D12.2
9	KODNISYONER1_KAPAK	Binary Tag	1		PLC	GONDERIM	DB110,D2.3
10	KODNISYONER1_KAPAK_ACTI	Binary Tag	1		PLC	GONDERIM	DB110,D12.3
11	KODNISYONER2_KAPAK	Binary Tag	1		PLC	GONDERIM	DB110,D2.4
12	KODNISYONER2_KAPAK_ACTI	Binary Tag	1		PLC	GONDERIM	DB110,D12.4
13	M400_CALISMA_BILGISI	Binary Tag	1		PLC	GONDERIM	DB110,D0.0
14	M401_CALISMA_BILGISI	Binary Tag	1		PLC	GONDERIM	DB110,D0.1
15	M402_CALISMA_BILGISI	Binary Tag	1		PLC	GONDERIM	DB110,D0.2
16	M404_CALISMA_BILGISI	Binary Tag	1		PLC	GONDERIM	DB110,D0.3
17	M405_CALISMA_BILGISI	Binary Tag	1		PLC	GONDERIM	DB110,D0.4
18	M406_CALISMA_BILGISI	Binary Tag	1		PLC	GONDERIM	DB110,D0.5
19	M407_CALISMA_BILGISI	Binary Tag	1		PLC	GONDERIM	DB110,D0.6
20	M408_CALISMA_BILGISI	Binary Tag	1		PLC	GONDERIM	DB110,D0.7
21	MOTOR_SARMA_YAPTI	Binary Tag	1		PLC	GONDERIM	DB110,D12.5
22	MOTOR1_AKIM	Floating-point number 32-bit IEEE	4	FloatToFloat	PLC	GONDERIM	DB110,DD4
23	MOTOR2_AKIM	Floating-point number 32-bit IEEE	4	FloatToFloat	PLC	GONDERIM	DB110,DD8
24	OV2_ACIK_BILGISI	Binary Tag	1		PLC	GONDERIM	DB110,D1.1
25	PIMKESTI	Binary Tag	1		PLC	GONDERIM	DB110,D2.5
26	RESET_ORTAYA_CIKTI	Binary Tag	1		PLC	GONDERIM	DB110,D12.6
27	SICAKLIK_DEGERI	Floating-point number 32-bit IEEE	4	FloatToFloat	PLC	GONDERIM	DB110,DD14
28	SIL0_400_MIN	Binary Tag	1		PLC	GONDERIM	DB110,D2.1
29	SIL0_401_MAX	Binary Tag	1		PLC	GONDERIM	DB110,D2.2
30	SIL0400_BOS	Binary Tag	1		PLC	GONDERIM	DB110,D12.7
31	SIL0401_DOLU	Binary Tag	1		PLC	GONDERIM	DB110,D13.0
32	UNITE_ACIL_STOP	Binary Tag	1		PLC	GONDERIM	DB110,D2.0
33	UNITE_BEKLE	Binary Tag	1		PLC	GONDERIM	DB110,D1.6
34	UNITE_DEVAM	Binary Tag	1		PLC	GONDERIM	DB110,D1.7
35	UNITE_START	Binary Tag	1		PLC	GONDERIM	DB110,D1.4
36	UNITE_STOP	Binary Tag	1		PLC	GONDERIM	DB110,D1.5
37	YEM_START	Binary Tag	1		PLC	GONDERIM	DB110,D1.2

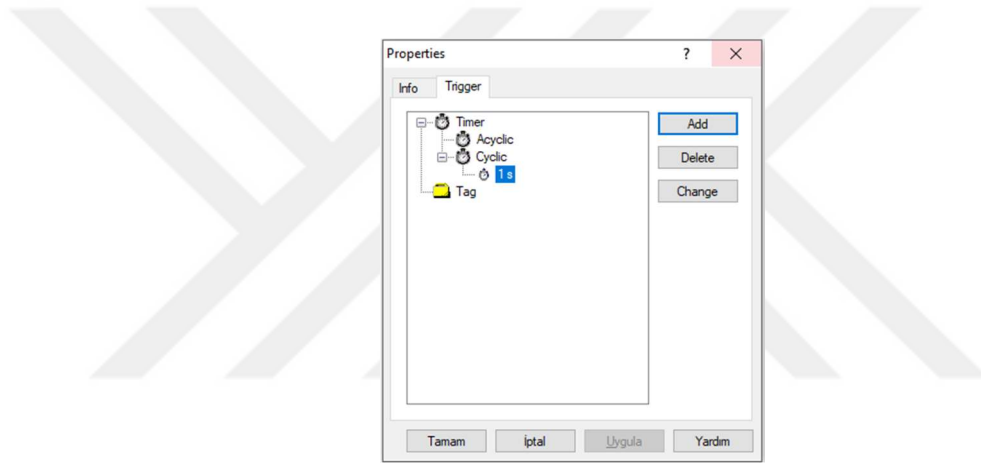
Şekil 7.7. Uygulamadaki Tag Management Görünümü

7.7. SCADA'dan Hosting'e Veri Aktarımı

PLC'den SCADA'ya düşen bilgiler bir hosting üzerine aktarımı için Wincc Explorer programı içerisinde bulunan Vbs yazılım dilinden faydalanılmıştır. Global Script bölmesinden seçilen Vbs script dili cycle süresi ayarları ise yine global script bölmesinden yapılmaktadır. Bu ayarı yapabilmek için herhangi bir script dili içinde iken

Ctrl+T tuşları tıklamak yeterlidir. Ortaya çıkan Properties penceresinden istenilirse herhangi bir tag ile ya da herhangi bir timer ile script dil cycle süresini ayarlamak mümkündür (Wincc V7.4 Wincc: Scripting, 2016). Şekil 7.8’ de Properties penceresini gösterilmektedir.

Yazılan yazılımda ilk olarak yazılım içinde kullanılacak olan değişkenlerin belirtilmesi hedeflenmiştir. Bu sonucunda android uygulamasına ve html uygulamasına gönderilecek olan verilerin isimlerini değişken isimlere yazılması uygun görülmüştür. Şekil 7.9’da görülmekte olan script kısmı değişken isim tanımlamayı göstermektedir. Dim adındaki bu tanımlamalar Vbs yazılım dili için sembol tanımlama olarak geçmektedir (Wincc V7.4 Wincc: Scripting, 2016).



Şekil 7.8. Veri Alışverişi İçin Gerekli Olan Cycle Süresinin Belirlenmesi

```

Dim M400_CALISMA_BILGISI
Dim M401_CALISMA_BILGISI
Dim M402_CALISMA_BILGISI
Dim M404_CALISMA_BILGISI
Dim M405_CALISMA_BILGISI
Dim M406_CALISMA_BILGISI
Dim M407_CALISMA_BILGISI
Dim M408_CALISMA_BILGISI
Dim K400_ACIK_BILGISI
Dim OV2_ACIK_BILGISI
Dim YEM_START
Dim BUHAR_START
Dim UNITE_START
Dim UNITE_STOP
Dim UNITE_BEKLE
Dim UNITE_DEVAM
Dim UNITE_ACIL_STOP
Dim SILO_400_MIN
Dim SILO_401_MAX
Dim KODNISYONER1_KAPAK
Dim KODNISYONER2_KAPAK
Dim PIMKESTI
Dim BESLEYICI_CALIS_KESTI
Dim HATA_OLUSTU
Dim K400_KLAPE_PASIFE_GECTI
Dim KODNISYONER1_KAPAK_ACTI
Dim KODNISYONER2_KAPAK_ACTI
Dim MOTOR_SARMA_YAPTI
Dim RESET_ORTAYA_CIKTI
Dim SILO400_BOS
Dim SILO401_DOLU
Dim MOTOR1_AKIM
Dim MOTOR2_AKIM
Dim SICAKLIK_DEGERI
Dim BUHAR_DEGERI
Dim BESLEME

```

Şekil 7.9. Uygulamadaki Scriptte Kullanılan Tanımlamalar

SCADA gerçek zamanlı bir programlama türüdür. Bu yüzden gelen tagların anlık olarak değişmesi android uygulamasına ve internet sayfasına düşecek olan bilgilerin anlık olarak bilgi talep etmesine neden olacaktır. “*HMIRuntime.Tags(“”).Read*” komutu SCADA’nın gerçek zamanlı biçimine geçtiğinde sistemdeki bilgileri cycle süresine bağlı olarak okumasına olanak sağlayacaktır (Wincc V7.4 Wincc: Scripting, 2016). Şekil 7.10’deki yazılım ile beraber tagların cycle sorgulama işlemi gerçekleşmektedir.

```

M400_CALISMA_BILGISI = HMIRuntime.Tags("M400_CALISMA_BILGISI").Read
M401_CALISMA_BILGISI = HMIRuntime.Tags("M401_CALISMA_BILGISI").Read
M402_CALISMA_BILGISI = HMIRuntime.Tags("M402_CALISMA_BILGISI").Read
M404_CALISMA_BILGISI = HMIRuntime.Tags("M404_CALISMA_BILGISI").Read
M405_CALISMA_BILGISI = HMIRuntime.Tags("M405_CALISMA_BILGISI").Read
M406_CALISMA_BILGISI = HMIRuntime.Tags("M406_CALISMA_BILGISI").Read
M407_CALISMA_BILGISI = HMIRuntime.Tags("M407_CALISMA_BILGISI").Read
M408_CALISMA_BILGISI = HMIRuntime.Tags("M408_CALISMA_BILGISI").Read
K400_ACIK_BILGISI = HMIRuntime.Tags("K400_ACIK_BILGISI").Read
OV2_ACIK_BILGISI = HMIRuntime.Tags("OV2_ACIK_BILGISI").Read
YEM_START = HMIRuntime.Tags("YEM_START").Read
BUHAR_START = HMIRuntime.Tags("BUHAR_START").Read
UNITE_START = HMIRuntime.Tags("UNITE_START").Read
UNITE_STOP = HMIRuntime.Tags("UNITE_STOP").Read
UNITE_BEKLE = HMIRuntime.Tags("UNITE_BEKLE").Read
UNITE_DEVAM = HMIRuntime.Tags("UNITE_DEVAM").Read
UNITE_ACIL_STOP = HMIRuntime.Tags("UNITE_ACIL_STOP").Read
SILO_400_MIN = HMIRuntime.Tags("SILO_400_MIN").Read
SILO_401_MAX = HMIRuntime.Tags("SILO_401_MAX").Read
KODNISYONER1_KAPAK = HMIRuntime.Tags("KODNISYONER1_KAPAK").Read
KODNISYONER2_KAPAK = HMIRuntime.Tags("KODNISYONER2_KAPAK").Read
PIMKESTI = HMIRuntime.Tags("PIMKESTI").Read
BESLEYICI_CALIS_KESTI = HMIRuntime.Tags("BESLEYICI_CALIS_KESTI").Read
HATA_OLUSTU = HMIRuntime.Tags("HATA_OLUSTU").Read
K400_KLAPE_PASIFE_GECTI = HMIRuntime.Tags("K400_KLAPE_PASIFE_GECTI").Read
KODNISYONER1_KAPAK_ACTI = HMIRuntime.Tags("KODNISYONER1_KAPAK_ACTI").Read
KODNISYONER2_KAPAK_ACTI = HMIRuntime.Tags("KODNISYONER2_KAPAK_ACTI").Read
MOTOR_SARMA_YAPTI = HMIRuntime.Tags("MOTOR_SARMA_YAPTI").Read
RESET_ORTAYA_CIKTI = HMIRuntime.Tags("RESET_ORTAYA_CIKTI").Read
SILO400_BOS = HMIRuntime.Tags("SILO400_BOS").Read
SILO401_DOLU = HMIRuntime.Tags("SILO401_DOLU").Read
MOTOR1_AKIM = HMIRuntime.Tags("MOTOR1_AKIM").Read
MOTOR2_AKIM = HMIRuntime.Tags("MOTOR2_AKIM").Read
SICAKLIK_DEGERI = HMIRuntime.Tags("SICAKLIK_DEGERI").Read
BUHAR_DEGERI = HMIRuntime.Tags("BUHAR_DEGERI").Read
BESLEME = HMIRuntime.Tags("BESLEME").Read

```

Şekil 7.10. Uygulamadaki Tanımlamaların Okunmasını Sağlayan Yazılım

Anlık olarak gelen bilgiler internet üzerinde bulunan hosting vasıtası ile veritabanına yerleştirilmesi sağlanmaktadır (Wincc V7.4 Wincc: Scripting, 2016). Bunun için Şekil 7.11’ daki yazılım uygulanmaktadır. SQL veritabanı bilgileri driver, port, database, UID, password gibi aşağıdaki hosting bilgilerinin doğrulanmasına bağlı olarak gönderilmektedir.

```

strConnectionString= "DRIVER={MySQL ODBC 5.3 ANSI Driver}; SERVER=rattle.guzelhosting.com;
PORT=3306; DATABASE=mahmutke_yuksekk_lisans_mahmut; UID=mahmutke_bilgeamc_mahmut; PASSWORD=Mahmut.12345;
Option=3"

```

Şekil 7.11. Bilgisayar ile Hosting Domain İnternet Vasıtası ile Bağlantı Yazılımı

Şekil 7.12’ de bulunmakta olan yazılımlarda hosting veri tabanına gönderilecek olan bilgilerin veri tabanı ile eşleşmesi sağlanmaktadır. Burada gönderilecek olan veri

tabanı bilgileri Şekil 7.10' da gösterilmektedir. Şekil 7.13'de ise bu gönderilecek bilgilerin değerleri belirtilmektedir.

```
Dim urunSql
urunSql = "INSERT INTO scada (" &
" `M400_CALISMA_BILGISI`, " &
" `M401_CALISMA_BILGISI`, " &
" `M402_CALISMA_BILGISI`, " &
" `M404_CALISMA_BILGISI`, " &
" `M405_CALISMA_BILGISI`, " &
" `M406_CALISMA_BILGISI`, " &
" `M407_CALISMA_BILGISI`, " &
" `M408_CALISMA_BILGISI`, " &
" `K400_ACIK_BILGISI`, " &
" `OV2_ACIK_BILGISI`, " &
" `YEM_START`, " &
" `BUHAR_START`, " &
" `UNITE_START`, " &
" `UNITE_STOP`, " &
" `UNITE_BEKLE`, " &
" `UNITE_DEVAM`, " &
" `UNITE_ACIL_STOP`, " &
" `SILO_400_MIN`, " &
" `SILO_401_MAX`, " &
" `KODNISYONER1_KAPAK`, " &
" `KODNISYONER2_KAPAK`, " &
" `PIMKESTI`, " &
" `BESLEYICI_CALIS_KESTI`, " &
" `HATA_OLUSTU`, " &
" `K400_KLAPE_PASIFE_GECTI`, " &
" `KODNISYONER1_KAPAK_ACTI`, " &
" `KODNISYONER2_KAPAK_ACTI`, " &
" `MOTOR_SARMA_YAPTI`, " &
" `RESET_ORTAYA_CIKTI`, " &
" `SILO_400_BOS`, " &
" `SILO_401_DOLU`, " &
" `MOTOR1_AKIM`, " &
" `MOTOR2_AKIM`, " &
" `SICAKLIK_DEGERI`, " &
" `BUHAR_DEGERI`, " &
" `BESLEME`)" &
```

Şekil 7.12. Uygulamadaki Veritabanına Gönderilecek Olan Bilgilerin Tanımlamaları

```
"VALUES (" &
M400_CALISMA_BILGISI & ", " &
M401_CALISMA_BILGISI & ", " &
M402_CALISMA_BILGISI & ", " &
M404_CALISMA_BILGISI & ", " &
M405_CALISMA_BILGISI & ", " &
M406_CALISMA_BILGISI & ", " &
M407_CALISMA_BILGISI & ", " &
M408_CALISMA_BILGISI & ", " &
K400_ACIK_BILGISI & ", " &
OV2_ACIK_BILGISI & ", " &
YEM_START & ", " &
BUHAR_START & ", " &
UNITE_START & ", " &
UNITE_STOP & ", " &
UNITE_BEKLE & ", " &
UNITE_DEVAM & ", " &
UNITE_ACIL_STOP & ", " &
SILO_400_MIN & ", " &
SILO_401_MAX & ", " &
KODNISYONER1_KAPAK & ", " &
KODNISYONER2_KAPAK & ", " &
PIMKESTI & ", " &
BESLEYICI_CALIS_KESTI & ", " &
HATA_OLUSTU & ", " &
K400_KLAPE_PASIFE_GECTI & ", " &
KODNISYONER1_KAPAK_ACTI & ", " &
KODNISYONER2_KAPAK_ACTI & ", " &
MOTOR_SARMA_YAPTI & ", " &
RESET_ORTAYA_CIKTI & ", " &
SILO400_BOS & ", " &
SILO401_DOLU & ", " &
"REPLACE(' " & MOTOR1_AKIM & " ', ' ', ' ', ' ') " &
"REPLACE(' " & MOTOR2_AKIM & " ', ' ', ' ', ' ') " &
"REPLACE(' " & SICAKLIK_DEGERI & " ', ' ', ' ', ' ') " &
"REPLACE(' " & BUHAR_DEGERI & " ', ' ', ' ', ' ') " &
"REPLACE(' " & BESLEME & " ', ' ', ' ', ' ') " &
");"
```

Şekil 7.13. Uygulamadaki Veritabanına Gönderilecek Olan Bilgilerin Değerler

7.8.Hosting’den SCADA’ya Veri Aktarımı

Hosting’ten SCADA’ya veri aktarımı yine SCADA üzerinde kullanılan Vbs Script ile yapılmıştır. Veri göndermekten farklı olarak veri çekmek için farklı bir veritabanı kullanılan sistemde Html ya da android üzerinden gönderilen veriler SCADA üzerinden PLC Database ögesine düşmektedir. Script üzerinde aynı şekilde bir tag ile ya da zaman vasıtası ile cycle yapılmakta olan yazılım bu süreyi Ctrl+T ile ayarlanması sağlanmaktadır (Wincc V7.4 Wincc: Scripting, 2016). Şekil 7.14’ de bulunmakta olan yazılımda hosting üzerinden SCADA’ya veri gönderilmesini sağlamaktadır.

```
'strConnectionString= "DRIVER={MySQL ODBC 5.3 ANSI Driver}; SERVER=% VeritabanıIPAdresi %"; PORT=% Ve
'strConnectionString= "DRIVER={MySQL ODBC 5.3 ANSI Driver}; SERVER=daydream.guzelhosting.com; PORT=3306
strConnectionString= "DRIVER={MySQL ODBC 5.3 ANSI Driver}; SERVER=localhost; PORT=3306;
DATABASE=mahmutke_yuksekk_lisans_mahmut; UID=root; PASSWORD=''; Option=3"
'strConnectionString= "DRIVER={MySQL ODBC 5.3 ANSI Driver}; SERVER=rattle.guzelhosting.com; PORT=3306;
'strConnectionString= "DRIVER={MySQL ODBC 5.3 ANSI Driver}; SERVER=rattle.guzelhosting.com; PORT=3306;
HMIRuntime.Trace "strConnectionString= "& strConnectionString & vbCrLf

Set objConnection = CreateObject("ADODB.Connection")
objConnection.ConnectionString = strConnectionString
objConnection.Open

Dim uyariSql
uyariSql = "select * from uyari order by id desc limit 8"

HMIRuntime.Trace uyariSql & vbCrLf

Set objRecordset = CreateObject("ADODB.Recordset")
Set objCommand = CreateObject("ADODB.Command")

With objCommand
.ActiveConnection = objConnection
.CommandText = uyariSql
End With

Set objRecordset = objCommand.Execute
Do Until objRecordSet.EOF
'Wscript.Echo "Acil Stop Degeri: " & objRecordSet.Fields("ACIL_STOP").Value
HMIRuntime.Tags("ACIL_STOP").Write(objRecordSet.Fields("ACIL_STOP").Value)
HMIRuntime.Tags("SANAL_ORTAM_RESETLEME").Write(objRecordSet.Fields("SANAL_ORTAM_RESETLEME").Value)
HMIRuntime.Tags("SANAL_ORTAM_START").Write(objRecordSet.Fields("SANAL_ORTAM_START").Value)
HMIRuntime.Tags("SANAL_ORTAM_STOP").Write(objRecordSet.Fields("SANAL_ORTAM_STOP").Value)
HMIRuntime.Tags("SANAL_ORTAM_DEVAM").Write(objRecordSet.Fields("SANAL_ORTAM_DEVAM").Value)
HMIRuntime.Tags("SANAL_ORTAM_BEKLE").Write(objRecordSet.Fields("SANAL_ORTAM_BEKLE").Value)
HMIRuntime.Tags("SANAL_ORTAM_YEM_ST").Write(objRecordSet.Fields("SANAL_ORTAM_YEM_ST").Value)
HMIRuntime.Tags("SANAL_ORTAM_BUH_ST").Write(objRecordSet.Fields("SANAL_ORTAM_BUH_ST").Value)
objRecordSet.MoveNext
Loop
```

Şekil 7.14. Fonksiyonun Organizasyon Bloğundaki Görünümü

“HMIRuntime.Trace” komutu yukarda bağlanılan hosting domainden veri kümesi çekmek amacı ile kullanılmaktadır. Burada cycle süresine bağlı olarak çekilen anlık veriler Şekil 7.14’ deki program “HMIRuntime.Tags("").Read.Write(objRecordSet.Fields().Value)” komutu sayesinde SCADA üzerinde bulunan değişkenlere bilgiler girilmektedir (Wincc V7.4 Wincc: Scripting, 2016).

7.9.Hosting ve Domain

Hosting sanal bilgisayar olarak tanımlanmaktadır. İnternet çağı ile beraber internet sitelerinin milyonlarca takipçilere ulaştığı günümüzde sitelerin alışkanlıklarını ve sitelerin çökmemesini sağlayabilmek amacıyla kullanılmaktadır. Tıpkı bilgisayar üzerinde yapabildiğiniz işleri daha gelişmiş olarak hosting havuzlarında yapabilmek mümkündür.

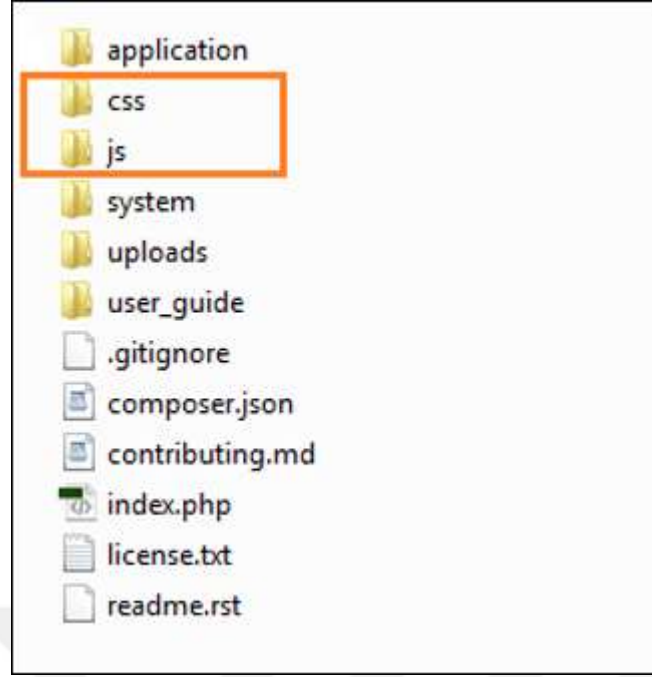
Proje üzerinde güzelhosting.com sitesinde faydalanmıştır. Bilgisayarda yerel olarak localhost üzerine düşen bilgiler Şekil 7.11’de bulunan yazılım sayesinde hosting havuzuna düşülmesini gerçekleştirilmiştir.

Web sitelerinin ulaşımı sağlayabilmek amacıyla kullanılan uzantılara domain denilmektedir. İnternete ulaşabilmek için IP (Internet Protokol) adresi olarak adlandırılan alan adı gerekmektedir. Bu adres 3 nokta ile birbirleri arasında ayrılmıştır. Domain bu IP adres bilgilerini bize internet sitesi ismi olarak göstermeyi sağlamaktadır. Domain com., net., org., gibi uzantılar dillerinde bulunmaktadır. Günümüzde domain uzantı belirlemedeki yetki ODTÜ idaresinde gerçekleşmektedir.

7.10.Codeigniter Klasörünü Hosting Serverına Aktarılması

Web sayfa tasarımı için gerekli olan bütün dosyaların toplandığı klasöre Codeigniter denir. Codeigniter klasörleri statik sayfa tasarımında ve dinamik sayfa biçimlendirilmesinde html web sayfaları için görev almaktadır. Codeigniter klasörü herhangi bir bilgisayar serverında çalışabilecek dosya türüdür. Yani dosya yerel bir bilgisayarda çalışabileceği gibi sanal bir bilgisayar olan hosting üzerinde çalışabilmesi mümkündür.

Uygulamada internet ortamı veri alışverişi gerçekleşeceğinden dolayı sanal bilgisayar olan hosting kullanılmıştır. Hosting domainde kullanıcıya paylaşım yapabilecek klasörleri ayrı tutmaktadır ve bu klasörleri public_html klasörü adı altında bir klasörde barındırmaktadır. Yani web sayfasının hosting serverında çalışabilmesi için codeigniter klasörü hosting serverının içersindeki public_html klasörüne aktarılması gerekmektedir. Bu aktarım sayesinde web sayfası kullanıcılara açılmaktadır.



Şekil 7.15. Codeigniter Klasörünün Örnek Gösterimi

7.11.Veritabanının Web Sayfasını Beslemesi

Codeigniter klasörünün içerisinde bulunan database.php adında bir yazılım ile hosting bilgisayarına aktarılan veri tabanını yine hosting bilgisayarında bulunan web sayfasına aktarımı gerçekleştirilmiştir. Şekil 7.16’da web sayfası ile veri tabanı arasındaki bilgi alışverişi için yazılan yazılım gösterilmektedir.

Veri tabanı ile web sayfası arasındaki bilgi transferi için gerekli olan bağlantı küme elemanları hostname, username, password ve database’dır. Bunun haricinde geri kalan bağlantı elemanları ise web sayfalarındaki biçim ayarlarında kullanılmak için bulunmaktadır. Hostname kullanılan hosting domain adresini belirtmek için kullanılmaktadır. Uygulamada güzelhosting firmasının hosting aygıtı kullanılmıştır. Her hosting firmasının kullanıcılarına verdikleri bir kullanıcı ismi ve bir şifre bulunmaktadır. Yapılan ayarlama da kullanıcı ismi olarak mahmutke_bilgeamc_mahmut şifre ise Mahmut.12345 olarak belirlenmiştir. Kullanılacak olan hesapta hangi veritabanından veri çekileceği yönündeki belirtme ise database ayarı ile gerçekleşmektedir. Uygulamada veri transferi mahmutke_yuksekk_lisans_mahmut veritabanı kullanılmıştır.

```

$db['default'] = array(
    'dsn' => '',
    'hostname' => 'rattle.guzelhosting.com',
    'username' => 'mahmutke_bilgeamc_mahmut',
    'password' => 'Mahmut.12345',
    'database' => 'mahmutke_yuksek_lisans_mahmut',
    'dbdriver' => 'mysqli',
    'dbprefix' => '',
    'pconnect' => FALSE,
    'db_debug' => (ENVIRONMENT !== 'production'),
    'cache_on' => FALSE,
    'cachedir' => '',
    'char_set' => 'utf8',
    'dbcollat' => 'utf8_general_ci',
    'swap_pre' => '',
    'encrypt' => FALSE,
    'compress' => FALSE,
    'stricton' => FALSE,
    'failover' => array(),
    'save_queries' => TRUE
);

```

Şekil 7.16. Web Sayfasının Hosting Domaine Bağlantısının Sağlanması

7.12. Web Sayfalarından Android'e Veri Gönderme

Web sayfalarından Android işletim sistemine gönderilecek olan bilgiler contaigner klasörü içerisindeki web sayfalarının dinamik hale getirilmesini sağlayan bir php uzantılı yazılım aracılığı ile gerçekleştirilmiştir. Veri tabanından çekilen bilgiler web sitesinin arka planındaki betik dilin üzerine düştükten sonra Şekil 7.17'de gösterilen yazılımdaki database ve json kütüphaneleri yardımı ile Android işletim sistemine aktarımı gerçekleştirilmiştir.

```

$q = $this->db->query("select * from scada order by id desc limit 1");
foreach($q->result_array() as $k){
    echo json_encode(array("M400_CALISMA_BILGISI"=>$k["M400_CALISMA_BILGISI"], "M401_CALISMA_BILGISI"=>$k["M401_CALISMA_BILGISI"],
    "M402_CALISMA_BILGISI"=>$k["M402_CALISMA_BILGISI"], "M404_CALISMA_BILGISI"=>$k["M404_CALISMA_BILGISI"],
    "M405_CALISMA_BILGISI"=>$k["M405_CALISMA_BILGISI"], "M406_CALISMA_BILGISI"=>$k["M406_CALISMA_BILGISI"],
    "M407_CALISMA_BILGISI"=>$k["M407_CALISMA_BILGISI"], "M408_CALISMA_BILGISI"=>$k["M408_CALISMA_BILGISI"],
    "M408_CALISIVOR"=>$k["M408_CALISIVOR"], "K400_ACIK_BILGISI"=>$k["K400_ACIK_BILGISI"],
    "OV2_ACIK_BILGISI"=>$k["OV2_ACIK_BILGISI"], "YEM_START"=>$k["YEM_START"],
    "BUHAR_START"=>$k["BUHAR_START"], "UNITE_START"=>$k["UNITE_START"],
    "UNITE_STOP"=>$k["UNITE_STOP"], "UNITE_BEKLE"=>$k["UNITE_BEKLE"],
    "UNITE_DEVAM"=>$k["UNITE_DEVAM"], "UNITE_ACIL_STOP"=>$k["UNITE_ACIL_STOP"],
    "SILO_400_MIN"=>$k["SILO_400_MIN"], "SILO_401_MAX"=>$k["SILO_401_MAX"],
    "KODNISONER1_KAPAK"=>$k["KODNISONER1_KAPAK"], "KODNISONER2_KAPAK"=>$k["KODNISONER2_KAPAK"],
    "PIMKESTI"=>$k["PIMKESTI"], "BESLEVICI_CALIS_KESTI"=>$k["BESLEVICI_CALIS_KESTI"],
    "HATA_OLUSTU"=>$k["HATA_OLUSTU"], "K400_KLAPE_PASIFE_GECTI"=>$k["K400_KLAPE_PASIFE_GECTI"],
    "KODNISONER1_KAPAK_ACTI"=>$k["KODNISONER1_KAPAK_ACTI"], "KODNISONER2_KAPAK_ACTI"=>$k["KODNISONER2_KAPAK_ACTI"],
    "MOTOR_SARMA_YAPTI"=>$k["MOTOR_SARMA_YAPTI"], "RESET_ORTAYA_CIKTI"=>$k["RESET_ORTAYA_CIKTI"],
    "SILO_400_BOS"=>$k["SILO_400_BOS"], "SILO_401_DOLU"=>$k["SILO_401_DOLU"],
    "MOTOR1_AKIM"=>$k["MOTOR1_AKIM"], "MOTOR2_AKIM"=>$k["MOTOR2_AKIM"],
    "BESLEME"=>$k["BESLEME"], "BUHAR_DEGERI"=>$k["BUHAR_DEGERI"], "SICAKLIK_DEGERI"=>$k["SICAKLIK_DEGERI"]
    ));
}
}

```

Şekil 7.17. Web Sayfasına Düşen Verilerin Android'e Aktarımı

$\$q = \$this->db->query("select * from scada order by id desc limit 1");$ q değerine db yani database kütüphanesinin jquery yardımı ile çekilen veri tabanı atanması sağlanmıştır. Burada daha sonra $foreach(\$q->result_array() as \$k)$ komutu ile q değişkeni k değişkenine atanarak tek gelen veriler parçalara ayrılmıştır.

Parçalara ayrılan veriler $json_encode(array)$ komutu ile daha kolay bir şekilde android işletim sistemine aktarılması hedeflenmiştir. Bu aktarma işlemi sırasında veri tabanından çekilen verilerin değerleri $\$k[""]$ komut ile alınması sağlanmıştır. Verilerin değeri en son olarak Json komutu ile değişken ismi verilerek tanımlanmıştır. Verilerin değeri ve verilerin değişkenleri php yazılımı aşağıda gösterilmiştir.

```
json_encode(array("M400_CALISMA_BILGISI"=>$k["M400_CALISMA_BILGISI"],
```

Şekil 7.18. Json Kütüphanesindeki Veri Bilgisi ve Veri Değeri

7.13.Json

Json kütüphaneleri veri alışverişini sağlamak ve veri alışverişi esnasında akışkanlığı oluşturabilmek amacı ile kullanılmaktadır. Genel olarak Json kütüphaneleri Xml kodlarının uzunluğundan daha kısa ve anlaşılır bir kod türüne dönüştürmek amacıyla oluşturulmuştur. Java ile geliştirilmiş olan bir kütüphanedir. Farklı yazılım türleri arasında geçiş yapmaya olanak sağlamaktadır.

7.14.Web Sayfasından SCADA'ya Veri Gönderme

Web sayfasından SCADA'ya veri göndermek için yine db ve json kütüphaneleri kullanılmıştır. Uygulama prosesi neticesinde gerekli olan butonlar için kullanılan web sayfasından SCADA'ya veri çekme için buton objeleri için ayrıca bir script yazılmıştır. Şekil 7.19' de web sayfasında click butonu olarak kullanılması için php ile bir fonksiyon yazılımı tasarlanmıştır. $\$q = \$this->db->query("select * from uyari order by id desc limit 1");->row();$ komutu web sayfasındaki oluşturulmuş olan verileri uyari adı altındaki veri tabanına göndermek amacı ile kullanılmaktadır. Şekil 7.20 de ise SCADA'ya gönderilecek olan verilerin değer alması amacı ile ilgili bir yazılım gösterilmektedir. Buradaki yazılım da php ile bir fonksiyon oluşturularak web sayfasında bir toggle butonu oluşturulması

hedeflenmiştir. *function(e)* fonksiyonu en son halinde bulunan buton değerini ters olan diğer değere alınmasını sağlamıştır. Yani buton 1 ise 0 değerine 0 ise 1 değerine aktarımı Şekil 7.20' deki yazılım sayesinde aktarımı sağlanmıştır.

```

1 <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
2   <div class="card" style="padding: 45px;">
3     <?php
4     $q = $this->db->query("select * from uyari order by id desc limit 1")->row();
5     ?>
6     <div class="row">
7       <?php
8
9         if($q->SANAL_ORTAM_RESETLEME == 1){
10          |
11          ?>
12          <div class="col-md-3 text-center">
13            <button class="btn btn-success" id="SANAL_ORTAM_RESETLEME_stop"> Sanal Ortam Resetleme</button>
14          </div>
15        <?php
16        }else {
17          ?>
18
19          <div class="col-md-3 text-center ">
20            <button class="btn btn-danger" id="SANAL_ORTAM_RESETLEME_baslat" > Sanal Ortam Resetleme</button>
21          </div>
22
23        <?php
24        }
25      ?>
26    </div>

```

Şekil 7.19. Fonksiyonun Organizasyon Bloğundaki Görünümü(Karayel, 2013)

```

27 </script>
28 // SANAL_ORTAM_RESETLEME BAŞLAT
29 $("#SANAL_ORTAM_RESETLEME_baslat").click(function(e){
30   e.preventDefault();
31   $.ajax({
32     url: "<?php echo base_url(); ?>index.php/yonetici/SANAL_ORTAM_RESETLEME_baslat",
33     type:'get',
34     success: function(data){
35       location.href="<?php echo base_url(); ?>index.php/yonetici/";
36     }
37   });
38 });
39
40 $("#SANAL_ORTAM_RESETLEME_stop").click(function(e){
41   e.preventDefault();
42   $.ajax({
43     url: "<?php echo base_url(); ?>index.php/yonetici/SANAL_ORTAM_RESETLEME_stop",
44     type:'get',
45     success: function(data){
46
47     location.href="<?php echo base_url(); ?>index.php/yonetici/";
48   }
49 });
50 });
51
52 </script>

```

Şekil 7.20. Fonksiyonun Organizasyon Bloğundaki Görünümü

8. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

8.1. Araştırma Sonucunda Tezin Avantajı ve Dezavantajı

8.1.1. Araştırma sonucunda tezin avantajı

Araştırma sonucunda gelişime daima açık ve büyüyen bir sektör olan endüstriyel sektörünün SCADA'dan başka bir arayüzde ve işletim sisteminde kontrolü sağlanmıştır. Bu gelecek dünyadaki kontrol esnekliğini sağlanmasında önemli bir unsurdur. Ayrıca her şeyin bilgisayar düzleminden akıllı telefonlara indiği çağda herhangi bir sistem sahibinin sistemi hakkındaki bilgilere ulaşabilmesi açısından çalışma ayrıca bir avantaja sahiptir.

Çalışma sayesinde ayrıca verilerin Hosting düzlemine düşmesi sağlanmıştır. Bu sayede verilerin kayıt edilmesi sağlanmıştır. Sistemde meydana gelen olayların kayıt edilmesi sistemin arızalarının tespit edilmesiyle çalışmada ayrıca bir yeri olacaktır.

8.1.2. Araştırma sonucunda tezin dezavantajı

Araştırma sonucunda tezin en büyük dezavantajı sisteme müdahil olan personel sayısının artışıdır. Sistemi birden fazla kişinin kontrol etmesi sistemdeki kararsızlığa neden olacaktır. Bunun ile ilgili olarak sistem içerisinde bir admin ve personel olmak üzere izin sistemi oluşturulması önerilebilir. Ayrıca sistemdeki ikinci problem ise internetten kaynaklanmaktadır. Zira internet sistemi ile haberleşmenin sağlanması haberleşme protokollerine göre daha güvensiz ve süreksizdir. Hosting üzerine düşen bilgilerin kesilmesi veya sistemi kontrol eden akıllı telefonunun internetinin kesilmesi sistemin kararlılığını etkileyen bir diğer unsurdur.

9. SONUÇLAR VE ÖNERİLER

Endüstriyel sektörlerde kullanılan sistemler için düşünülmüş olan çalışmada SCADA telemetri sistemleri ek olarak android ve internet tabanlı uygulamalar ile kontrol ele alınmıştır. Endüstriyel proses olarak yem fabrikalarında sıklıkla kullanılan pelet makinalarının prosesi referans olarak belirlenmiştir.

Siemens endüstriyel paket programlarından Step 7 Simatic Manager programı ile Wincc Explorer programları tercih edilmiştir. Wincc Explorer programına anlık gelen bilgilerin ışığında program üzerinde yazılan script ile beraber veriler hosting üzerinde kurulan veritabanına aktarılmıştır. Veritabanı ile hosting arasındaki bağlantı için gerekli olan yazılım Şekil 7.11’ de gösterilmiştir. Şekil 7.11’deki yazılımda da görüldüğü üzere bir bilgisayarın modeme nasıl bağlanıyorsa aynı şekilde bir hostinge bağlantının gerçekleşmesi için bir şifre, UID, port adresi, database dosyasının gerekli olduğu belirtilmiştir.

Hosting veritabanına aktarılan bilgilerin anlık olarak html sayfasına düşmesinde için sayfanın body bölümüne cycle süresi eklenilmesi düşünülmüştür. Bu sayede sayfanın arka planına gelen bilgilerin taze tutulması planlanmıştır. İnternet sayfası güvenlik açısından belirtilen modemlere bağlı olarak çalışmaktadır. Yani kullanılan modemden başka bir modem ile internet sitesine ulaşılmak mümkün olmamaktadır. Bu da çalışmada yöneticilik kavramı oluşturması sağlanmıştır.

İnternet sayfası ile beraber android uygulamasından da kontrol edilen sistem için android studio programı kullanılmıştır. Bu sayfa tasarımında veriler internet sitesi sayfası üzerinden çekilmiştir. İnternet sayfası gibi aynı şekilde android uygulamasına cycle süresi eklenmiştir.

İnternet ve akıllı cihaz çağına geçtiğimiz bu günlerde endüstriyel gelişmeler bu ışıkta kendini bulmaktadır. Bu sebeple beraber endüstri 4.0 ve 5.0 uygulamalar içerisinde artık endüstriyel HMI, SCADA sistemlerine ek olarak ara yüz programlar için internet sayfası ve android uygulamalarının eklenmesi söz konusudur. Bu çizelgede duruma bakılırsa eğer endüstriyel kütüphanelerin artık android ve ios uygulamalar için yazılım geliştiricileri tarafından desteklenmesi görüşü hakimdir.

KAYNAKLAR

- Anonymous, 2010, Simatic Working with STEP 7, 1-110.
- Anonymous, 2016, Simatic Hmı Wincc V7.4 Getting Started, 1-216.
- Anonymous, 2016, Wincc V7.4 Wincc: Scripting, 1-850.
- Asubay, M.B., 2018, SCADA Sistemlerinin Tanıtımı ve Kullanılan Haberleşme Protokolleri, Bitlis Eren Üniversitesi, Türkiye –Bitlis, 1-73
- Başar, D., 2014, Web tabanlı GPS/GNSS proje veri yönetim sistemi tasarımı ve uygulaması, İstanbul Teknik Üniversitesi, Türkiye –İstanbul, 1-39
- Bıçek, E., 2016, Android ve Web Tabanlı Bir Uygulama ile İşitme Engellilere Yönelik Sesli Dersi Yazılı Takip Sistemi, Türkiye –Bitlis, 1-73
- Boşnak, İ., 2016, PLC ile PLL Kontrol İndüksiyon Isıtma Uygulaması, Karabük Üniversitesi, Türkiye –Karabük, 1-72
- Karan, R., 2017, PLC Tabanlı Hemzemin Geçit Koruma Sisteminin Tasarlanması ve Simülasyonu, Gazi Üniversitesi, Türkiye –Ankara, 1-72
- Karayel, M., 2013, Mikrotip hidroelektrik santraller için PLC tabanlı SCADA sistem otomasyonu ve RTU/PLC ile frekans ve gerilim regülasyonunun gerçekleştirilmesi, Gazi Üniversitesi, Türkiye - Ankara, 1-70
- Kayabaşı, G., 2016, İzin tabanlı statik analiz yöntemi ile android uygulamaların sınıflandırılması, Gazi Üniversitesi, Türkiye –Ankara, 1-51
- Kırmızıgül, M. E., 2014, Metro hatlarında SCADA sistemlerinin entegrasyonu için bir model önerisi, Bahçeşehir Üniversitesi, Türkiye –İstanbul, 1-71
- Kulga, U.B., 2018, Php Applications, K-wave Simulations and Experimental Studies for Medical Ultrasound, Orta Doğu Teknik Üniversitesi, Türkiye –Ankara, 1-124
- Salğar, G., 2010, Doğalgaz SCADA otomasyon uygulamalarında ana kontrol merkezi ile haberleşmeyi sağlayan SCADA otomasyon sistemlerinin; cicode SCADA ve PLC programlarındaki mevcut parametreler yardımı ile mevcut parametreler yardımı ile bilgisayar destekli yeni ara yüzler oluşturulması, Sütçü İmam Üniversitesi, Türkiye –Kahramanmaraş, 1-93
- Sarıkahya, M., 2013, SCADA ile enerji izleme ve otomasyon, Gazi Üniversitesi, Türkiye –Ankara, 1-75
- Sönmez, Y., 2017, Açık kaynak programlar ile internet üzerinden öğrenci takip sistemi tasarımı ve performans analizi, İstanbul Kemerburgaz Üniversitesi, Türkiye –İstanbul, 1-44

- Şen, C., 2016, İçme Suyu Tesislerinde SCADA Sistemlerine Sezgisel Yaklaşımların Uygulanması, Sakarya Üniversitesi, Türkiye –Sakarya, 1-166
- Şumnulu, H., 2015, HTML / PHP /MySQL kod editörü, Ege Üniversitesi, Türkiye - İzmir, 1-120
- Özer, A., 2016, Endüstriyel Sistemlerde PLC ve SCADA Uygulaması, Marmara Üniversitesi, Türkiye – İstanbul, 1-86
- Uslu, B., 2015, HTML5 tabanlı oyunlar için yapay zeka kütüphanesi, Süleyman Demirel Üniversitesi, Türkiye - Isparta, 1-110
- Üstünsoy, F., 2018, Orta Ölçekli Bir Tesisin SCADA ile Enerji Yönetimi, Gazi Üniversitesi, Türkiye –Ankara, 1-63
- Yaşar, U. Ş., 2018, PLC kontrollü SILAR cihazı yapımı, Erzincan Üniversitesi, Türkiye – Erzincan, 1-44

EKLER

EK-1 Vbs ile SCADA'dan Hosting Üzerindeki Veritabanına Veri Transferi

Option Explicit
Function action

Dim M400_CALISMA_BILGISI
Dim M401_CALISMA_BILGISI
Dim M402_CALISMA_BILGISI
Dim M404_CALISMA_BILGISI
Dim M405_CALISMA_BILGISI
Dim M406_CALISMA_BILGISI
Dim M407_CALISMA_BILGISI
Dim M408_CALISMA_BILGISI
Dim K400_ACIK_BILGISI
Dim OV2_ACIK_BILGISI
Dim YEM_START
Dim BUHAR_START
Dim UNITE_START
Dim UNITE_STOP
Dim UNITE_BEKLE
Dim UNITE_DEVAM
Dim UNITE_ACIL_STOP
Dim SILO_400_MIN
Dim SILO_401_MAX
Dim KODNISYONER1_KAPAK
Dim KODNISYONER2_KAPAK
Dim PIMKESTI
Dim BESLEYICI_CALIS_KESTI
Dim HATA_OLUSTU
Dim K400_KLAPE_PASIFE_GECTI
Dim KODNISYONER1_KAPAK_ACTI
Dim KODNISYONER2_KAPAK_ACTI
Dim MOTOR_SARMA_YAPTI
Dim RESET_ORTAYA_CIKTI
Dim SILO400_BOS
Dim SILO401_DOLU
Dim MOTOR1_AKIM
Dim MOTOR2_AKIM
Dim SICAKLIK_DEGERI
Dim BUHAR_DEGERI
Dim BESLEME

M400_CALISMA_BILGISI = HMIRuntime.Tags("M400_CALISMA_BILGISI").Read
M401_CALISMA_BILGISI = HMIRuntime.Tags("M401_CALISMA_BILGISI").Read
M402_CALISMA_BILGISI = HMIRuntime.Tags("M402_CALISMA_BILGISI").Read
M404_CALISMA_BILGISI = HMIRuntime.Tags("M404_CALISMA_BILGISI").Read
M405_CALISMA_BILGISI = HMIRuntime.Tags("M405_CALISMA_BILGISI").Read
M406_CALISMA_BILGISI = HMIRuntime.Tags("M406_CALISMA_BILGISI").Read
M407_CALISMA_BILGISI = HMIRuntime.Tags("M407_CALISMA_BILGISI").Read
M408_CALISMA_BILGISI = HMIRuntime.Tags("M408_CALISMA_BILGISI").Read
K400_ACIK_BILGISI = HMIRuntime.Tags("K400_ACIK_BILGISI").Read
OV2_ACIK_BILGISI = HMIRuntime.Tags("OV2_ACIK_BILGISI").Read
YEM_START = HMIRuntime.Tags("YEM_START").Read
BUHAR_START = HMIRuntime.Tags("BUHAR_START").Read
UNITE_START = HMIRuntime.Tags("UNITE_START").Read
UNITE_STOP = HMIRuntime.Tags("UNITE_STOP").Read
UNITE_BEKLE = HMIRuntime.Tags("UNITE_BEKLE").Read
UNITE_DEVAM = HMIRuntime.Tags("UNITE_DEVAM").Read
UNITE_ACIL_STOP = HMIRuntime.Tags("UNITE_ACIL_STOP").Read
SILO_400_MIN = HMIRuntime.Tags("SILO_400_MIN").Read
SILO_401_MAX = HMIRuntime.Tags("SILO_401_MAX").Read
KODNISYONER1_KAPAK = HMIRuntime.Tags("KODNISYONER1_KAPAK").Read

```

KODNISYONER2_KAPAK = HMIRuntime.Tags("KODNISYONER2_KAPAK").Read
PIMKESTI = HMIRuntime.Tags("PIMKESTI").Read
BESLEYICI_CALIS_KESTI = HMIRuntime.Tags("BESLEYICI_CALIS_KESTI").Read
HATA_OLUSTU = HMIRuntime.Tags("HATA_OLUSTU").Read
K400_KLAPE_PASIFE_GECTI = HMIRuntime.Tags("K400_KLAPE_PASIFE_GECTI").Read
KODNISYONER1_KAPAK_ACTI = HMIRuntime.Tags("KODNISYONER1_KAPAK_ACTI").Read
KODNISYONER2_KAPAK_ACTI = HMIRuntime.Tags("KODNISYONER2_KAPAK_ACTI").Read
MOTOR_SARMA_YAPTI = HMIRuntime.Tags("MOTOR_SARMA_YAPTI").Read
RESET_ORTAYA_CIKTI = HMIRuntime.Tags("RESET_ORTAYA_CIKTI").Read
SILO400_BOS = HMIRuntime.Tags("SILO400_BOS").Read
SILO401_DOLU = HMIRuntime.Tags("SILO401_DOLU").Read
MOTOR1_AKIM = HMIRuntime.Tags("MOTOR1_AKIM").Read
MOTOR2_AKIM = HMIRuntime.Tags("MOTOR2_AKIM").Read
SICAKLIK_DEGERI = HMIRuntime.Tags("SICAKLIK_DEGERI").Read
BUHAR_DEGERI = HMIRuntime.Tags("BUHAR_DEGERI").Read
BESLEME = HMIRuntime.Tags("BESLEME").Read

```

```

Dim objConnection
Dim objCommand
Dim objRecordset
Dim strConnectionString
Dim isEmriSql

```

```

'strConnectionString= "DRIVER={MySQL ODBC 5.3 ANSI Driver}; SERVER=" & VeritabaniIPAdresi &"; PORT=" &
VeritabaniPort &"; DATABASE=EMS_ " & firma &"; UID=" & VeritabaniKullaniciAdi &"; PASSWORD=" & VeritabaniSifresi &";
Option=3"

```

```

'strConnectionString= "DRIVER={MySQL ODBC 5.3 ANSI Driver}; SERVER=daydream.guzelhosting.com;
PORT=3306; DATABASE=bilgeamc_mahmut; UID=bilgeamc_mahmut; PASSWORD=Mahmut.12345; Option=3"

```

```

'strConnectionString= "DRIVER={MySQL ODBC 5.3 ANSI Driver}; SERVER=localhost; PORT=3306;
DATABASE=mahmutke_yukse lisans_mahmut; UID=root; PASSWORD="; Option=3"

```

```

strConnectionString= "DRIVER={MySQL ODBC 5.3 ANSI Driver}; SERVER=rattle.guzelhosting.com; PORT=3306;
DATABASE=mahmutke_yukse lisans_mahmut; UID=mahmutke_bilgeamc_mahmut; PASSWORD=Mahmut.12345; Option=3"

```

```

'strConnectionString= "DRIVER={MySQL ODBC 5.3 ANSI Driver}; SERVER=daydream.guzelhosting.com;
PORT=3306; DATABASE=bilgeamc_mahmut; UID=bilgeamc_mahmut; PASSWORD=Mahmut.12345; Option=3"

```

```

HMIRuntime.Trace "strConnectionString= " & strConnectionString & vbCrLf

```

```

Set objConnection = CreateObject("ADODB.Connection")
objConnection.ConnectionString = strConnectionString
objConnection.Open

```

```

Dim urunSql
urunSql = "INSERT INTO scada (" & _
"M400_CALISMA_BILGISI'," & _
"M401_CALISMA_BILGISI'," & _
"M402_CALISMA_BILGISI'," & _
"M404_CALISMA_BILGISI'," & _
"M405_CALISMA_BILGISI'," & _
"M406_CALISMA_BILGISI'," & _
"M407_CALISMA_BILGISI'," & _
"M408_CALISMA_BILGISI'," & _
"K400_ACIK_BILGISI'," & _
"OV2_ACIK_BILGISI'," & _
"YEM_START'," & _
"BUHAR_START'," & _
"UNITE_START'," & _
"UNITE_STOP'," & _
"UNITE_BEKLE'," & _
"UNITE_DEVAM'," & _
"UNITE_ACIL_STOP'," & _
"SILO_400_MIN'," & _
"SILO_401_MAX'," & _

```

```

"KODNISYONER1_KAPAK',"&_
"KODNISYONER2_KAPAK',"&_
"PIMKESTI',"&_
"BESLEYICI_CALIS_KESTI',"&_
"HATA_OLUSTU',"&_
"K400_KLAPE_PASIFE_GECTI',"&_
"KODNISYONER1_KAPAK_ACTI',"&_
"KODNISYONER2_KAPAK_ACTI',"&_
"MOTOR_SARMA_YAPTI',"&_
"RESET_ORTAYA_CIKTI',"&_
"SILO_400_BOS',"&_
"SILO_401_DOLU',"&_
"MOTOR1_AKIM',"&_
"MOTOR2_AKIM',"&_
"SICAKLIK_DEGERI',"&_
"BUHAR_DEGERI',"&_
"BESLEME')"&_

"VALUES (" &_
M400_CALISMA_BILGISI & ","&_

M401_CALISMA_BILGISI & ","&_
M402_CALISMA_BILGISI & ","&_
M404_CALISMA_BILGISI & ","&_
M405_CALISMA_BILGISI & ","&_
M406_CALISMA_BILGISI & ","&_
M407_CALISMA_BILGISI & ","&_
M408_CALISMA_BILGISI & ","&_

K400_ACIK_BILGISI & ","&_

OV2_ACIK_BILGISI & ","&_
YEM_START & ","&_
BUHAR_START & ","&_
UNITE_START & ","&_
UNITE_STOP & ","&_
UNITE_BEKLE & ","&_
UNITE_DEVAM & ","&_

UNITE_ACIL_STOP & ","&_

SILO_400_MIN & ","&_
SILO_401_MAX & ","&_
KODNISYONER1_KAPAK & ","&_
KODNISYONER2_KAPAK & ","&_
PIMKESTI & ","&_
BESLEYICI_CALIS_KESTI & ","&_
HATA_OLUSTU & ","&_

K400_KLAPE_PASIFE_GECTI & ","&_

KODNISYONER1_KAPAK_ACTI & ","&_
KODNISYONER2_KAPAK_ACTI & ","&_
MOTOR_SARMA_YAPTI & ","&_
RESET_ORTAYA_CIKTI & ","&_
SILO400_BOS & ","&_
SILO401_DOLU & ","&_
"REPLACE(" & MOTOR1_AKIM & ",";';!'),"&_
"REPLACE(" & MOTOR2_AKIM & ",";';!'),"&_
"REPLACE(" & SICAKLIK_DEGERI & ",";';!'),"&_
"REPLACE(" & BUHAR_DEGERI & ",";';!'),"&_
"REPLACE(" & BESLEME & ",";';!')"&_

");"

```

```

HMIRuntime.Trace urunSql & vbCrLf

Set objRecordset = CreateObject("ADODB.Recordset")
Set objCommand = CreateObject("ADODB.Command")

With objCommand

.ActiveConnection = objConnection

.CommandText = urunSql

End With

Set objRecordset = objCommand.Execute

Set objCommand = Nothing
objConnection.Close
Set objRecordset = Nothing
Set objConnection = Nothing
End Function

```

EK-2 Vbs ile Hosting Üzerindeki Veritabanından SCADA'ya Veri Transferi

```

Option Explicit
Function action

Dim objConnection
Dim objCommand
Dim objRecordset
Dim strConnectionString

'strConnectionString= "DRIVER={MySQL ODBC 5.3 ANSI Driver}; SERVER=" & VeritabaniIPAdresi & "; PORT=" &
VeritabaniPort & "; DATABASE=EMS_ " & firma & "; UID=" & VeritabaniKullaniciAdi & "; PASSWORD=" & VeritabaniSifresi & ";
Option=3"
'strConnectionString= "DRIVER={MySQL ODBC 5.3 ANSI Driver}; SERVER=daydream.guzelhosting.com;
PORT=3306; DATABASE=bilgeamc_mahmut; UID=bilgeamc_mahmut; PASSWORD=Mahmut.12345; Option=3"
'strConnectionString= "DRIVER={MySQL ODBC 5.3 ANSI Driver}; SERVER=localhost; PORT=3306;
DATABASE=mahmutke_yuksekk_lisans_mahmut; UID=root; PASSWORD=" & VeritabaniSifresi & "; Option=3"
strConnectionString= "DRIVER={MySQL ODBC 5.3 ANSI Driver}; SERVER=rattle.guzelhosting.com; PORT=3306;
DATABASE=mahmutke_yuksekk_lisans_mahmut; UID=mahmutke_bilgeamc_mahmut; PASSWORD=Mahmut.12345; Option=3"
'strConnectionString= "DRIVER={MySQL ODBC 5.3 ANSI Driver}; SERVER=rattle.guzelhosting.com; PORT=3306;
DATABASE=mahmutke_yuksekk_lisans_mahmut; UID=mahmutke_bilgeamc_mahmut; PASSWORD=Mahmut.12345; Option=3"
HMIRuntime.Trace "strConnectionString=" & strConnectionString & vbCrLf

Set objConnection = CreateObject("ADODB.Connection")
objConnection.ConnectionString = strConnectionString
objConnection.Open

Dim uyariSql
uyariSql = "select * from uyari order by id desc limit 1"

HMIRuntime.Trace uyariSql & vbCrLf

Set objRecordset = CreateObject("ADODB.Recordset")
Set objCommand = CreateObject("ADODB.Command")

With objCommand
.ActiveConnection = objConnection
.CommandText = uyariSql

End With

```

```
Set objRecordset = objCommand.Execute
Do Until objRecordSet.EOF
    'Wscript.Echo "Acil Stop Degeri: " & objRecordSet.Fields("ACIL_STOP").Value
    HMIRuntime.Tags("ACIL_STOP").Write(objRecordSet.Fields("ACIL_STOP").Value)

HMIRuntime.Tags("SANAL_ORTAM_RESETLEME").Write(objRecordSet.Fields("SANAL_ORTAM_RESETLEME").Value)
HMIRuntime.Tags("SANAL_ORTAM_START").Write(objRecordSet.Fields("SANAL_ORTAM_START").Value)
HMIRuntime.Tags("SANAL_ORTAM_STOP").Write(objRecordSet.Fields("SANAL_ORTAM_STOP").Value)
HMIRuntime.Tags("SANAL_ORTAM_DEVAM").Write(objRecordSet.Fields("SANAL_ORTAM_DEVAM").Value)
HMIRuntime.Tags("SANAL_ORTAM_BEKLE").Write(objRecordSet.Fields("SANAL_ORTAM_BEKLE").Value)

HMIRuntime.Tags("SANAL_ORTAM_YEM_ST").Write(objRecordSet.Fields("SANAL_ORTAM_YEM_ST").Value)

HMIRuntime.Tags("SANAL_ORTAM_BUH_ST").Write(objRecordSet.Fields("SANAL_ORTAM_BUH_ST").Value)

    objRecordSet.MoveNext
Loop

Set objCommand = Nothing
objConnection.Close
Set objRecordset = Nothing
Set objConnection = Nothing

End Function
```


ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Mahmut Kertmen
Uyruğu : Türkiye Cumhuriyeti
Doğum Yeri ve Tarihi : Konya – 29.03.1994
Telefon : 05459029683
e-mail : mahmut.kertmen.01@gmail.com

EĞİTİM

Derece	Adı, İlçe, İl	Bitirme Yılı
Lise	: Cemil Keleşoğlu Lisesi, Merkez, Konya	2007 - 2011
Üniversite	: Selçuk Üniversitesi, Selçuklu, Konya	2012 - 2016
Yüksek Lisans	: Selçuk Üniversitesi, Selçuklu, Konya	2016 - Devam

İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
2017-2019	Arskon Otomasyon Ltd.Şti.	Otomasyon Mühendisi

UZMANLIK ALANI

Wincc Explore: **Çok iyi**
 Simatic : **İyi**
 AutoCAD : **orta**
 Asp : **orta**
 Word, Excel : **İyi**

YABANCI DİLLER

Yabancı Dil : **İngilizce**
 Okuma : **İyi**
 Yazma : **İyi**
 Konuşma : **orta**

YAYINLAR

KERTMEN, Mahmut, ÇETİNKAYA, Nurettin 2019, " Transfer Of Pelet Machine Parameters From Scada To Android Through Php", International Journal Of Engineering Sciences & Research Technology, Vol. 8, Iss.4. pp:48-58